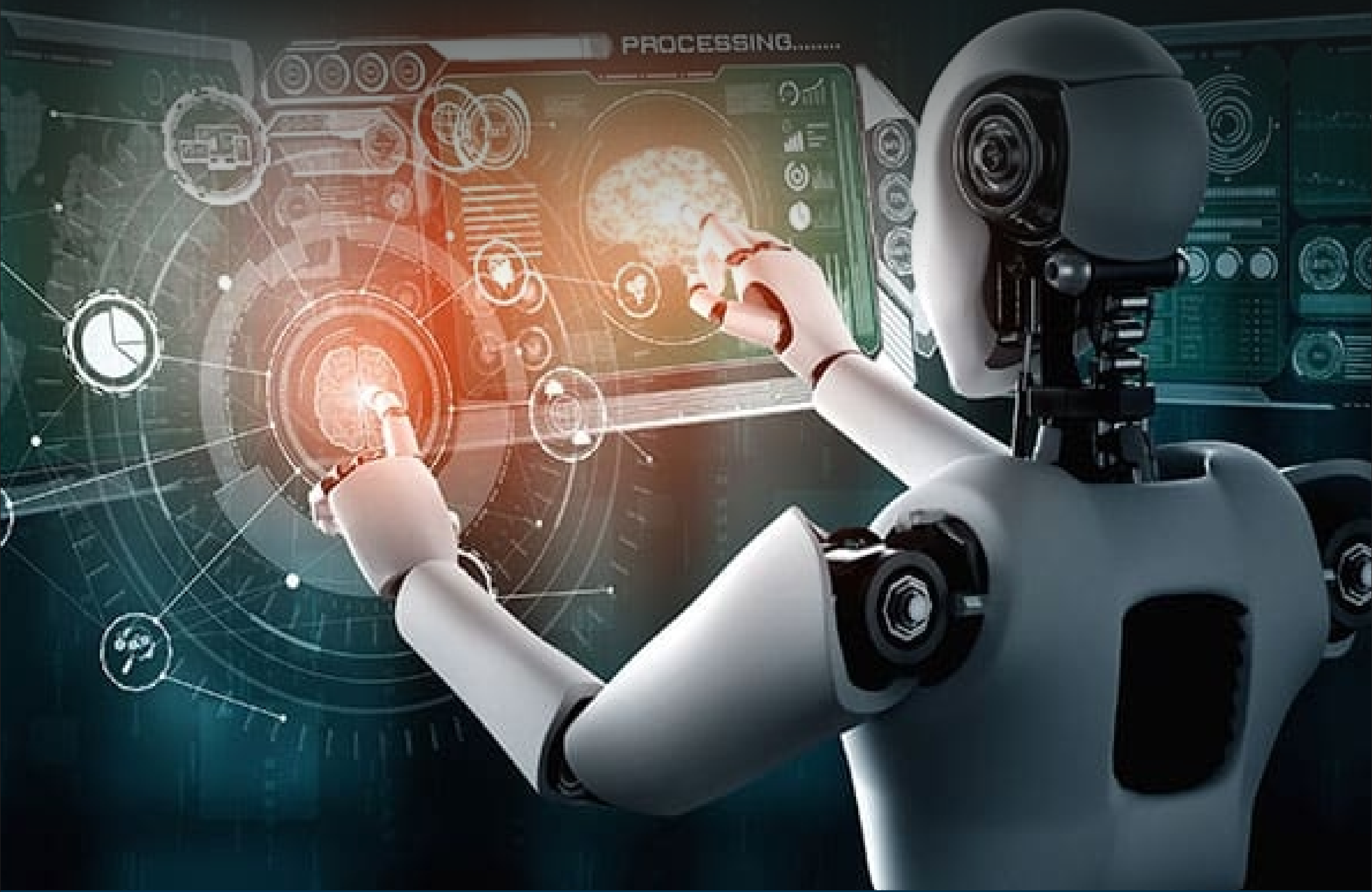
YAYASAN PRIMA AGUS TEKNIK

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

AI (Artificial Intelligence) dan PEMECAHAN MASALAH

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

AI (Artificial Intelligence) dan PEMECAHAN MASALAH



YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-50-8 (PDF)
9 786347 227508

AI (Artificial Intelligence) dan PEMECAHAN MASALAH

Penulis :
Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

ISBN : 978-634-7227-50-8 (PDF)

Editor :
Dr. Joseph Teguh Santoso, S.Kom., M.Kom.

Penyunting :
Dr. Mars Caroline Wibowo. S.T., M.Mm.Tech

Desain Sampul dan Tata Letak :
Irdha Yunianto, S.Ds., M.Kom

Penebit :
Yayasan Prima Agus Teknik Bekerja sama dengan
Universitas Sains & Teknologi Komputer (Universitas STEKOM)

Anggota IKAPI No: 279 / ALB / JTE / 2023

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. 08122925000
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, atas rahmat dan karunia-Nya sehingga buku "***AI (Artificial Intelligence) Dan Pemecahan Masalah***" ini dapat selesai disusun. Buku ini bertujuan memberikan pemahaman komprehensif mengenai teknik-teknik pemecahan masalah dalam ranah kecerdasan buatan melalui pendekatan yang sistematis dan aplikatif.

Dalam buku ini, pembaca diajak untuk mengenal berbagai teknik dan langkah pemecahan masalah yang dikembangkan dalam bidang kecerdasan buatan, mulai dari langkah-langkah Polya yang klasik hingga pendekatan modern yang menggabungkan analisis jendela manusia dan solusi mesin terbaik. Setiap bab menghadirkan sejumlah permasalahan yang khas dan aplikatif, seperti permasalahan misionaris dan kanibal, teka-teki keledai merah, sudoku, hingga kriptografi dan algoritma Monte Carlo. Dengan pendekatan yang terstruktur serta didukung program-program yang dapat dimainkan, buku ini diharapkan mampu memperkaya wawasan para pembaca, baik akademisi, praktisi, maupun mahasiswa yang tertarik mendalami kecerdasan buatan dan teknik pemecahan masalah.

Bab 1 membuka dengan pendahuluan yang menjelaskan tujuan, latar belakang, serta kontribusi buku ini dalam pengembangan ilmu kecerdasan buatan dan metode pemecahan masalah. Bab ini membekali pembaca dengan konteks yang diperlukan untuk memahami isi buku selanjutnya. Bab 2 membahas teori dasar pemecahan masalah, menguraikan lima langkah Polya yang klasik serta berbagai teknik pemecahan masalah, termasuk konsep jendela manusia dan kriteria evaluasi solusi. Bab ini berfungsi sebagai landasan teori yang kokoh.

Bab-bab 3 hingga 17 menyajikan studi kasus dan permasalahan khas yang sering dijumpai dalam kecerdasan buatan, seperti permasalahan misionaris dan kanibal, teka-teki keledai merah, kubus Rubik, sudoku, dan kriptografi. Masing-masing bab membahas teknik pemecahan, analisis solusi manusia dan mesin, serta menyediakan program yang dapat digunakan untuk eksperimen praktis.

Bab 18 mengakhiri dengan refleksi terhadap studi jendela manusia dan pelajaran penting yang diperoleh, ditutup dengan retrospektif dan arah karya masa depan dalam bidang ini. Buku ini juga dilengkapi daftar pustaka sebagai referensi lebih lanjut.

Dengan penyajian yang terstruktur dan menyeluruh ini, kami harapkan buku ini dapat menjadi sumber inspirasi dan bimbingan bagi akademisi, praktisi, dan mahasiswa yang hendak mendalami bidang kecerdasan buatan dan pemecahan masalah.

Kami menyadari masih banyak kekurangan, oleh karena itu masukan yang membangun sangat kami nantikan. Semoga buku ini memberikan manfaat luas dan menyemangati perkembangan teknologi kecerdasan buatan di Indonesia.

Selamat dan Semangat Membaca...

Semarang, Oktober 2025
Penulis

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM.

DAFTAR ISI

BAB 1
PENDAHULUAN

Saat ini, siswa memiliki banyak sekali teknologi yang tersedia untuk membantu mereka memecahkan masalah—baik di sekolah maupun dalam kehidupan. Sayangnya, sebagian besar siswa menggunakan teknologi terutama untuk hiburan dan jejaring sosial. Dalam hal penggunaan teknologi untuk tujuan pendidikan, sebagian besar aplikasinya terbatas pada pencarian jawaban atau solusi cepat untuk tugas. Faktanya, informasi tentang hampir setiap mata pelajaran begitu mudah tersedia dan mudah diakses sehingga selama dua dekade terakhir, bukti upaya siswa untuk memecahkan masalah yang membutuhkan pemikiran kritis menjadi agak langka.

Pemecahan masalah, analisis isu, dan pengambilan keputusan (yang dulu dianggap standar bagi lulusan perguruan tinggi) semakin menjadi keterampilan yang paling dicari di pasar kerja. Dengan demikian, mengembangkan keterampilan ini akan bermanfaat tidak hanya dalam pendidikan tetapi juga dalam pekerjaan dan bidang kehidupan lainnya. Namun, orang sering kali kewalahan oleh masalah yang kompleks dan cenderung tersesat di dalamnya. Cara sistematis untuk mengajarkan siswa cara memahami suatu masalah, menggunakan teknik yang tepat sesuai dengan jenis masalah spesifik, jelas diperlukan.

1.1 TUJUAN DAN MAKSUD BUKU INI

Tujuan buku ini adalah untuk mengumpulkan, mengklasifikasikan, dan mempelajari solusi manusia dan mesin untuk beberapa masalah klasik yang sering diajukan dalam matematika, ilmu komputer, dan Kecerdasan Buatan (AI). Secara keseluruhan, kami mengklasifikasikan dan memandang masalah-masalah ini sebagai masalah AI karena karakteristik khususnya: masalah-masalah ini tidak sepele, setidaknya bagi manusia; melibatkan logika; mungkin melibatkan pencarian dan memori; solusi untuk masalah-masalah ini melibatkan teknik-teknik tertentu, seperti deduksi, inferensi, penyelesaian submasalah, dan pengenalan pola, yang dapat diterapkan dan diperluas ke masalah-masalah lain; masalah-masalah ini pada umumnya sudah dikenal dan telah teruji oleh waktu; beberapa dari masalah-masalah ini telah digunakan sebagai landasan uji untuk teknik-teknik pemecahan masalah AI—misalnya, The 15 Puzzle, Mastermind, The Knight's Tour Problem, The Red Donkey Puzzle, dan Cryptarithms; dan masalah-masalah ini cocok untuk mempelajari aktivitas pemecahan masalah manusia. Tujuan jangka panjang yang lebih tinggi dari pekerjaan kami adalah pengembangan teori pemecahan masalah. Permasalahan yang kami pilih untuk diselidiki meliputi:

1. **Permasalahan Misionaris dan Kanibal: Dalam permasalahan ini**, sekelompok tiga misionaris dan tiga kanibal harus diangkut dengan aman dari tepi timur sungai ke tepi barat, menggunakan perahu yang hanya dapat mengangkut dua penumpang sekaligus. Syaratnya adalah, dalam kondisi apa pun, jumlah kanibal tidak boleh melebihi jumlah

misionaris. Tujuannya adalah agar semua orang dapat mencapai seberang sungai dengan perjalanan perahu sesedikit mungkin.

2. **Permasalahan 12 Koin**: Dalam permasalahan ini, terdapat 12 koin, dan kita harus menentukan 1 koin yang beratnya berbeda dari 11 koin lainnya hanya dengan tiga kali penimbangan, menggunakan timbangan.
3. **Kriptaritma:** Beberapa contohnya adalah $\text{KIRIM} + \text{LEBIH BANYAK} = \text{UANG}$ atau $\text{DONALD} + \text{GERALD} = \text{ROBERT}$. Tujuan dari soal-soal ini adalah menemukan angka unik (dari 0 hingga 9) yang mewakili huruf-huruf alfabet yang memenuhi persamaan.
4. **Teka-teki Keledai Merah:** Ini adalah teka-teki balok geser di mana balok persegi berukuran $2'' \times 2''$ harus dipindahkan dari atas papan berukuran $4'' \times 5''$ ke tengah bawah. Potongan lain dalam teka-teki ini meliputi empat balok vertikal berukuran $1'' \times 2''$, empat balok persegi berukuran $1'' \times 1''$, dan satu balok horizontal berukuran $2'' \times 1''$.
5. **Teka-teki 15:** Ini adalah teka-teki balok geser lainnya di mana 15 balok persegi satuan (bernomor 1 hingga 15) harus disusun ulang dalam urutan menaik searah jarum jam pada petak berukuran $4'' \times 4''$.
6. **Masalah Tur Ksatria:** Teka-teki ini melibatkan pencarian jalur di papan catur sedemikian rupa sehingga seorang ksatria catur mengunjungi setiap petak di papan tepat satu kali. Variasi yang lebih ketat dari masalah ini mengharuskan ksatria untuk kembali ke petak asal pada langkah terakhirnya. Biasanya ksatria memulai dari petak di sudut kiri bawah papan (dikenal sebagai A1).
7. **Mastermind:** Ini adalah permainan papan pemecah kode dua pemain di mana salah satu pemain (disebut pemecah kode) harus memecahkan kode atau menebak kode yang dibuat oleh pemain lain (disebut pembuat kode) dalam maksimal 10 tebakan. Kode ini diwakili oleh serangkaian pasak berwarna yang disusun dalam urutan tertentu.
8. **Masalah Monty Hall:** Masalah ini melibatkan menebak pintu mana dari tiga pintu yang memiliki hadiah (misalnya, mobil) di belakangnya. Dilemanya terletak pada apakah akan tetap pada pilihan pertama Anda atau berganti pintu setelah pembawa acara memberi tahu Anda pintu mana dari tiga pintu yang tidak memiliki hadiah di belakangnya.
9. **Kubus Rubik:** Teka-teki balok geser populer ini adalah kubus tiga dimensi yang terdiri dari kubus-kubus kecil dengan warna berbeda di setiap permukaannya. Tujuannya adalah menyusun kubus-kubus kecil ini sehingga kubus tersebut secara keseluruhan hanya memiliki satu warna di setiap permukaannya.
10. **Dilema Narapidana:** Di sini kami mempertimbangkan serangkaian masalah yang muncul dari teori permainan klasik yang melibatkan skema optimal tentang bagaimana narapidana bisa mendapatkan hukuman minimal, tergantung pada apakah mereka kooperatif atau tidak dalam pengakuan pribadi mereka.
11. **Masalah Lain-lain:** Kami juga telah memilih lima masalah yang lebih kecil yang bergenre sama dengan sepuluh masalah yang disajikan dalam buku ini, tetapi dalam

skala yang lebih kecil. Kelima masalah tersebut terutama melibatkan pengembangan abstraksi grafis yang kuat dan pemahaman masalah, merancang skema, menggunakan probabilitas, dan logika.

Buku ini berfokus pada solusi untuk masalah-masalah klasik ini menggunakan strategi pemecahan masalah spesifik seperti pencarian, reduksi masalah, deduksi, dan rumus matematika. Ia juga meneliti bagaimana orang biasanya mencoba memecahkan masalah dan menyajikan ide tentang cara menganalisis masalah dan memecahnya agar lebih mudah dikelola dan dipecahkan.

1.2 LATAR BELAKANG DAN KARYA SEBELUMNYA

Pemecahan masalah manusia merupakan topik penting yang telah dipelajari secara luas dan merupakan dasar bagi Kecerdasan Buatan dan "sepupunya" dari psikologi: ilmu kognitif. Dari perspektif kami, sebelum kami mengembangkan dan menggunakan mesin untuk memecahkan masalah bagi kami, pertama-tama kami harus memiliki pemahaman yang mendalam tentang bagaimana manusia berpikir dan memecahkan masalah.

Sebagian besar inspirasi untuk karya kami berasal dari buku George Polya, "How to Solve It". Buku kecil ini terbukti inovatif dan teruji oleh waktu dalam membantu para pemecah masalah untuk menjadi lebih sadar akan aktivitas mereka dan bagaimana mereka dapat lebih berhasil. (Lihat Bab 2 untuk informasi lebih lanjut tentang karya Polya.)

Allen Newell dan Herbert Simon telah berkontribusi secara signifikan terhadap topik pemecahan masalah dan penelitian mereka dirangkum dalam buku *Human Problem Solving*. Menurut Newell, Shaw, dan Simon, suatu masalah memiliki serangkaian jalur, yang hanya satu (atau beberapa) di antaranya mengarah ke tujuan yang diinginkan. Menemukan jalur yang tepat adalah memecahkan masalah. Masalah juga didefinisikan sebagai selisih antara keadaan yang ada dan keadaan yang diinginkan. Newell dan Simon memberikan beberapa contoh pemecahan masalah, seperti memecahkan labirin atau teka-teki silang, menemukan kombinasi brankas, dan menerjemahkan satu bahasa ke bahasa lain. Penelitian mereka mencakup topik-topik seperti representasi lingkungan eksternal dalam bentuk internal, strategi seperti pembuatan subtujuan dan submasalah untuk mencapai solusi, bekerja mundur dari suatu tujuan, dan menggunakan heuristik untuk menemukan jalur terbaik menuju solusi. Mereka juga menekankan proses mengandalkan abstraksi sederhana dari ruang masalah sebagai cara untuk mendekati masalah yang lebih kompleks.

Cerveny, Garrity, dan Sanders membahas dua jenis pemecahan masalah. Salah satunya adalah pemecahan masalah linear, yang melibatkan pelaksanaan serangkaian langkah untuk mencapai suatu tujuan, dan yang lainnya adalah pemecahan masalah konkuren, yang bersifat iteratif dan melibatkan perbandingan keadaan saat ini dengan keadaan ideal dan pelaksanaan langkah-langkah yang diperlukan untuk mencapai keadaan ideal tersebut.

Gunzelmann dan Anderson menyatakan bahwa dua langkah awal yang penting untuk memecahkan masalah adalah memahami masalah dan merepresentasikan masalah dalam bentuk internal. Mereka juga menekankan pentingnya perencanaan dalam pemecahan masalah. Manusia cenderung meningkatkan tingkat perencanaan jika dihadiahi dengan solusi yang lebih cepat.

Menemukan jalan menuju tujuan terkadang tidak cukup. Karena kendala seperti waktu, daya komputasi, dan sumber daya lainnya, pemecahan masalah secara efisien dengan pemanfaatan sumber daya yang optimal diperlukan. Menurut Anzai dan Simon, beberapa strategi tersedia untuk memecahkan masalah, beberapa lebih efisien daripada yang lain. Efisiensi suatu strategi bergantung pada berbagai faktor seperti kecepatan, beban memori, dan kemudahan penyimpanan informasi dalam memori, antara lain.

1.3 KONTRIBUSI BUKU INI

1. Klasifikasi permasalahan yang dibahas.
2. Pembahasan teknik-teknik pemecahan masalah umum yang berlaku untuk permasalahan yang dibahas.
3. Penerapan strategi pemecahan masalah yang tepat untuk permasalahan yang dipilih sesuai dengan domain permasalahan masing-masing.
4. Studi perilaku pemecahan masalah manusia dalam lingkungan pemecahan masalah.
5. Rekomendasi strategi dan pendekatan yang tepat dalam urutan yang tepat sesuai jenis permasalahan.
6. Pemaparan konsep "Jendela Manusia" Donald Michie untuk memecahkan permasalahan tipe AI.

BAB 2
PEMECAHAN MASALAH

Pemecahan masalah merupakan keterampilan penting yang membantu kita dalam segala aspek kehidupan. Oleh karena itu, penting bagi kita untuk mengikuti pendekatan sistematis dalam memecahkan masalah. Karya Polya merupakan terobosan karena menghasilkan pengembangan urutan lima langkah pemecahan masalah, yang pada dasarnya dapat diterapkan secara universal, terlepas dari domainnya.

2.1 LIMA LANGKAH POLYA UNTUK PEMECAHAN MASALAH

1. Memahami masalah.
2. Mengembangkan algoritma (rencana).
3. Mengimplementasikan algoritma (pengodean dalam ilmu komputer).
4. Menguji algoritma (debugging).
5. Merevisi algoritma (kemungkinan kembali ke langkah 1 hingga 4).

Hal yang luar biasa tentang lima langkah Polya adalah bahwa langkah-langkah tersebut dapat diterapkan hanya dengan sedikit revisi pada permasalahan matematika, pemrograman komputer, teknik, ilmu manajemen, bisnis, dan banyak lagi!

1. Pahami Masalah

Yang dimaksud Polya adalah jangan repot-repot mencoba memecahkan masalah sebelum Anda meluangkan waktu untuk membiasakan diri. Pelajari esensi masalahnya. Cobalah untuk mengklasifikasikan masalahnya.Apa jenis masalahnya? Bisakah masalah tersebut dipecah menjadi beberapa bagian? Bagaimana representasi yang baik untuk masalah tersebut? Apakah masalah tersebut terkait dengan masalah lain yang pernah Anda temui atau pecahkan?

Bahkan saat kami menulis bagian ini, kami menemukan MASALAH. Laptop terus-menerus mati. Beberapa komputer diketahui memiliki masalah dengan catu dayanya. Jadi kami bertanya-tanya, apa penyebabnya? Catu daya (salah satu dari dua catu daya yang tersedia tidak tepat untuk mesin tersebut)? Jadi kami mencoba kedua catu daya tersebut. Tidak ada perubahan. Cukup jelas bahwa catu daya bukanlah masalahnya. Setiap beberapa menit laptop mati, dan itu bukan pertanda baik untuk menyelesaikan buku ini sesuai tenggat waktu kami. Salah satu solusi yang kami coba adalah berhenti bekerja lebih awal dan membiarkan mesin "mendingin" semalaman. Tampaknya membaik di pagi hari, tetapi setelah sekitar satu jam beroperasi, mati lagi.

Apa saja gejala unik lain dari masalah ini?

Kami telah mencatat bahwa dalam beberapa minggu terakhir laptop terasa cukup panas (di kiri atas, di atas keyboard) di area ventilasi kipas.

Postmortem Masalah: Sekitar seminggu kemudian, motherboard dan kipas komputer diganti dalam masa garansi. Mesinnya sekarang berfungsi dengan baik.

2. Kembangkan Algoritma

Jelas, pemecahan masalah tanpa berpikir tidaklah efektif. Misalnya, dalam catur, kita mengatakan lebih baik bermain dengan rencana, bahkan rencana yang buruk sekalipun, daripada tidak punya rencana sama sekali. Kita bekerja dengan definisi algoritma berikut:

"Rangkaian langkah yang terdefinisi dengan baik, mungkin berulang, dan terbatas untuk memecahkan masalah."

Setelah memperhatikan masalah yang dijelaskan dalam kotak di atas, dan dipadukan dengan pengalaman sebelumnya (juga dijelaskan di atas), dan mengingat bahwa kita pernah melihat seorang anak muda memecahkan masalah dengan komputernya hanya dengan membukanya dan memasang kembali kipas, sebuah ide muncul:

Tersedia kipas yang sangat kecil (5 inci) yang terkadang digunakan selama hari-hari musim panas untuk pendinginan lokal yang cepat. Mungkin kipas itu bisa digunakan untuk mendinginkan laptop.

3. Terapkan Algoritma

Di sinilah Anda harus bertindak. Di kepala kita, kita memiliki lusinan ide dan rencana sepanjang waktu. Rencana tanpa implementasi menjadi seperti "kenangan buruk kecil" atau pengingat bahwa kita harus menyimpan daftar, catatan, dan sebagainya, dan tidak hanya menyimpan semuanya di kepala kita. Namun, Anda tidak dapat melakukan semuanya sekaligus, dan manusia bukanlah mesin. Kita membutuhkan semacam keteraturan dalam hidup dan aktivitas kita—suatu arahan—dan dengan demikian sebuah RENCANA yang memungkinkan kita menilai efektivitas solusi masalah yang diusulkan.

Kipas kecil 5 inci telah berputar selama sekitar 45 menit, dan laptop masih berfungsi dengan baik. Area di kiri atas masih panas, tetapi tidak sepanas sebelumnya (hanya menggunakan indra peraba). Apakah ini berarti masalah telah terpecahkan?

4. Uji Algoritma

Pengujian merupakan langkah yang sangat penting dalam pemecahan masalah. Kita semua ingin dapat berkata, "Masalahnya telah terpecahkan!" Seberapa besar keputusan itu didasarkan pada keinginan, emosi, ego, atau hanya keinginan untuk melanjutkan? Bagi orang-orang yang berorientasi pada tujuan, ambisius, teliti, obsesif, atau sekadar bertekad, membuat keputusan untuk menerima atau menolak suatu solusi bisa sangat membebani. Oleh karena itu, kita harus menjawab pertanyaan-pertanyaan berikut:

Apakah solusinya berhasil? Apakah berhasil sepenuhnya? Apakah berhasil sebagian? Sekarang, seberapa yakin Anda bahwa Anda telah memecahkan masalah? Berapa banyak lagi usaha yang diperlukan untuk memastikan bahwa masalah tersebut telah terpecahkan? Lebih dari satu setengah jam kemudian, solusi kipas angin tersebut tampaknya berhasil.

5. Merevisi Algoritma

Berdasarkan hasil di atas, dapatkah kita membuat keputusan tentang bagaimana melanjutkannya? Apakah kita kembali ke Tahap 1, di mana kita mempertimbangkan kembali pemahaman kita tentang masalah tersebut? Atau apakah kita mempertimbangkan solusi/rencana/algoritma alternatif (Tahap 2)? Apakah kita telah mengimplementasikan solusi atau algoritma yang direncanakan dengan benar?

Selain solusi "kipas", solusi alternatif telah dipertimbangkan, seperti menggunakan laptop lain atau mencadangkan semua berkas kunci yang dimaksud. Kita dapat menggunakan rencana alternatif ini, tergantung pada hasil algoritma/pengujian saat ini.

Sebuah masalah telah menyediakan beberapa data yang diberikan, beberapa variabel yang tidak diketahui, beberapa kondisi, keadaan awal, dan keadaan tujuan. Tujuannya adalah menggunakan data yang diberikan untuk menemukan variabel yang tidak diketahui sambil berpindah dari keadaan awal ke keadaan tujuan. Selama tahap awal pemecahan masalah, tujuan pemecah masalah adalah untuk memahami masalah dan merepresentasikannya dalam bentuk internal.

Salah satu langkah kunci dalam memfasilitasi **pemahaman** masalah adalah merepresentasikan masalah dalam format yang secara jelas menyajikan berbagai status masalah (tahapan) dari solusi yang diusulkan dan transisi di antara keduanya. Hal ini seringkali mengurangi kompleksitas masalah. Beberapa masalah mungkin paling baik diungkapkan secara grafis, seperti diagram atau sketsa grafis, sementara yang lain mungkin lebih baik direpresentasikan dalam format pohon pencarian.

Aspek penting dari **representasi pengetahuan** adalah tingkat detail. Representasi pengetahuan dapat bersifat ekstensional atau intensional. *Representasi ekstensional* menunjukkan setiap langkah, kasus, atau contoh, sedangkan *representasi intensional* singkat dan implisit—misalnya, sebuah rumus. Kita perlu memilih representasi yang paling baik mewakili masalah dan membantu kita memahami serta menyelesaikannya dengan mudah.

Menurut Polya, langkah penting dalam memahami masalah adalah ***mengisolasi bagian-bagian utama***—yaitu, data, kondisi, dan hal-hal yang tidak diketahui dalam suatu masalah. Bagian-bagian ini diperiksa secara individual, dalam kaitannya satu sama lain, dan dalam kaitannya dengan keseluruhan masalah. Masing-masing pendekatan ini membantu kita memahami masalah dengan lebih baik.

Setelah kita memahami masalahnya, kita melanjutkan ke **perencanaan** untuk menyelesaikannya. Ini adalah cara untuk mengelola kompleksitas masalah. Telah diamati bahwa jika perencanaan menghasilkan solusi yang lebih cepat, tingkat perencanaan akan meningkat.

Salah satu cara untuk menangani masalah yang kompleks adalah dengan *memecahkan versi masalah yang lebih sederhana*. Solusi dan metode yang digunakan untuk masalah yang lebih sederhana kemudian dapat digunakan untuk merencanakan masalah yang lebih sulit.

Memecahkan masalah melibatkan pencarian melalui sejumlah besar kemungkinan yang terdiri dari serangkaian jalur yang mengarah ke keadaan tujuan. Kita juga dapat membuat subtujuan dan menyelesaikannya untuk mencapai tujuan akhir. Algoritma dan heuristik adalah dua cara dasar untuk memecahkan masalah.

Algoritma dalam ilmu komputer biasanya membutuhkan komputasi intensif dan melibatkan pencarian menyeluruh dalam ruang keadaan. Oleh karena itu, algoritma akan selalu mengarah ke keadaan tujuan dan 100% benar. Pendekatan heuristik untuk pemecahan masalah lebih cocok untuk manusia dengan keterbatasan mereka dalam hal kapasitas memori dan kecepatan pemrosesan.

Heuristik adalah aturan praktis yang memungkinkan seseorang memilih satu atau beberapa jalur yang memungkinkan melalui ruang keadaan yang lebih mungkin mengarah ke keadaan tujuan. Meskipun heuristik biasanya menghasilkan solusi dengan pemrosesan yang relatif lebih sedikit, hal ini tidak selalu menjamin solusi dan terkadang bahkan dapat membawa kita ke arah yang salah.

2.2 TEKNIK PEMECAHAN MASALAH

Sekarang kita akan membahas beberapa teknik pemecahan masalah. Banyak buku dapat ditulis tentang teknik-teknik ini, tetapi tujuan kita di sini hanyalah untuk memperkenalkan dan membedakan teknik-teknik ini satu sama lain dengan harapan pemecah masalah akan menganggapnya bermanfaat.

- **Induksi**

  Dalam matematika, induksi sering digunakan sebagai teknik yang tidak terlalu akurat. Sayangnya, ketika digunakan di dunia nyata, teknik ini belum tentu akurat.

  Contoh 1: Saya menemukan seekor gagak hitam. Saya menemukan 100 gagak hitam. Setiap gagak yang saya temukan berwarna hitam. Oleh karena itu, gagak berikutnya yang saya temukan akan berwarna hitam. (Hal ini belum tentu benar.)

  Contoh 2: Setiap kali sebuah tim unggul 3-0 dalam seri playoff bisbol, tim tersebut menang. Hal ini telah terjadi 39 kali dalam sejarah, tetapi pada tahun 2004, Yankees unggul 3-0 melawan Boston Red Sox—dan mereka kalah!

- **Deduksi**

  Deduksi melibatkan upaya untuk mencapai kesimpulan dari beberapa fakta yang secara logis saling mengikuti. Seringkali, deduksi melibatkan informasi umum yang darinya kita mencoba mencapai kesimpulan spesifik

  Contoh 1: Modus *ponens yang terkenal* adalah contoh umum:

  Semua manusia fana. Socrates adalah manusia. Oleh karena itu, Socrates fana.

  Contoh 2: Terjadi perampokan di 365 85th Street, Apartemen 4B, antara pukul 16.00 dan 18.00. John Stealer adalah satu-satunya orang yang terlihat (oleh kamera keamanan) masuk dan meninggalkan apartemen tersebut antara jam-jam tersebut. Dengan demikian, logis untuk berasumsi (dengan deduksi) bahwa John Stealer adalah pencurinya.

  Contoh 3: Dalam kedokteran, kita menggunakan deduksi untuk menentukan penyakit apa yang diderita pasien. Misalnya, jika seorang pasien mengalami demam, sakit kepala, dan sakit tenggorokan, kemungkinan besar ia akan dipastikan menderita flu karena ia memiliki gejala "seperti flu".

- **Abduksi**

  Abduksi adalah upaya untuk membuat deduksi yang tidak tepat. Hal ini dibahas dalam buku John Sowa, Conceptual Structures. Ia memberikan contoh berikut:

  Contoh 1: Semua Quaker adalah pasifis. Richard Nixon adalah seorang Quaker. *Modus ponens* menyiratkan bahwa Nixon adalah seorang pasifis. (Hal ini belum tentu benar.)

Contoh 2: Cokelat adalah makanan cinta. Saya sedang jatuh cinta. Oleh karena itu, cokelat yang saya makan membuat saya jatuh cinta.

Contoh 3: Merokok menyebabkan kanker. Tom menderita kanker. Oleh karena itu, Tom adalah seorang perokok.

- **Empiris**

  Istilah *empiris* sering digunakan dengan istilah *data* dan *bukti*—misalnya, "data empiris menunjukkan" atau "bukti empiris menunjukkan." Ini mengacu pada data atau bukti yang tersedia atau telah diamati.

  Contoh 1: Bukti empiris menunjukkan bahwa merokok menyebabkan kanker.

  Contoh 2: Data empiris menunjukkan bahwa pengemudi yang terganggu menyebabkan sebagian besar kecelakaan kendaraan bermotor. Hal ini menunjukkan bahwa mengirim pesan teks saat mengemudi—sesuatu yang mengganggu pengemudi—seharusnya ilegal (*modus* ponens).

- **Pencacahan Ekstrem**

  Ini adalah upaya untuk memecahkan masalah dengan mempertimbangkan setiap kemungkinan dalam ruang pencarian. Hal ini tentu saja tidak efisien.

  Contoh 1: Di kelas, Anda dapat mengetahui siapa pemilik kertas ujian yang tidak ditandatangani dengan pencacahan ekstrem—misalnya, dengan memeriksa semua kertas ujian yang *memiliki* tanda tangan.

  Contoh 2: Kemungkinan untuk "menang", "seri", dan "kalah" dalam tic-tac-toe dapat dengan mudah ditentukan melalui enumerasi lengkap.

  Contoh 3: Semua kemungkinan untuk bermain catur secara teoritis dapat diidentifikasi melalui enumerasi lengkap, tetapi karena terdapat 10^{42} permainan yang diperkirakan masuk akal, hal itu tidak mungkin dilakukan dalam praktik.

- **Kontradiksi**

  Kontradiksi adalah teknik pemecahan masalah yang sering digunakan untuk membuktikan sesuatu itu benar atau tidak mungkin. Teknik ini didasarkan pada upaya untuk menunjukkan bahwa lawan dari suatu argumen tidak dapat hidup berdampingan dengan argumen itu sendiri.

  Contoh 1: Diberikan beberapa premis seperti A > B, maka A = B tidak mungkin benar pada saat yang bersamaan.

  Contoh 2: Jika seseorang dicurigai melakukan kejahatan tetapi memiliki alibi (misalnya, bukti bahwa ia berada di luar negeri pada saat kejahatan terjadi), maka itu akan menjadi pembuktian dengan kontradiksi.

- **Rekursif**

  Permasalahan atau solusi rekursif melibatkan penggambaran suatu masalah (atau fungsi) dalam dirinya sendiri. Teknik pemecahan masalah rekursif bersifat halus dan elegan. Teknik ini melibatkan penerapan langkah-langkah yang sama berulang-ulang tetapi untuk "argumen" yang berbeda. Karena alasan ini, program rekursif jauh lebih singkat daripada program iteratif.

$$\textbf{Contoh 1}: \mathbf{Fact(N)} = \mathbf{N} \circ \mathbf{Fact(N-1)};\ \mathbf{Fact(1)} = \mathbf{1}$$

Di sini, fungsi Faktorial didefinisikan dalam dirinya sendiri tetapi diterapkan pada argumen yang berbeda.

Contoh 2: Untuk menyelesaikan masalah Menara Hanoi dengan tujuh cakram, seseorang hanya perlu menyelesaikan masalah untuk enam cakram. Untuk menyelesaikan Menara Hanoi, yang perlu dilakukan hanyalah menyelesaikan masalah untuk lima cakram. Proses ini berulang hingga kondisi keluar terpenuhi. Inilah rekursi yang sedang bekerja. Ini adalah semacam teknik bagi-dan-kuasai. Namun, perlu diingat bahwa harus ada kondisi "keluar"—misalnya, ketika hanya ada satu cakram yang tersisa (cakram terbesar), cakram tersebut harus dipindahkan ke patok gawang.

- **Membagi dan Menaklukkan**

  Membagi dan menaklukkan adalah teknik pemecahan masalah yang didasarkan pada gagasan memecah masalah menjadi bagian-bagian yang lebih kecil. Hanya sedikit masalah yang dapat dipecahkan "sekaligus", jadi akan lebih baik jika masalah tersebut dipecah menjadi komponen-komponen yang logis.

  Contoh 1: Seseorang tidak bisa begitu saja duduk dan menulis seluruh buku, tetapi seseorang dapat memulai dengan kerangka, beberapa kata, frasa, kalimat lengkap, dan akhirnya sebuah paragraf. Hampir semua pemecahan masalah melibatkan suatu bentuk membagi dan menaklukkan.

  Contoh 2: Mengemudi 20 jam langsung ke Miami hampir mustahil. Namun, membagi perjalanan menjadi empat segmen, masing-masing berdurasi 5 jam, jauh lebih mudah dikelola.

- **Memecahkan Submasalah**

  Mirip dengan membagi dan menaklukkan, teknik ini melibatkan identifikasi komponen-komponen spesifik dari suatu masalah dan menyelesaikannya. Komponen-komponen tersebut mungkin independen satu sama lain (sehingga tidak ada ketergantungan), tetapi setiap komponen merupakan bagian dari keseluruhan masalah yang sama.

  Contoh 1: Jika kita perlu mengidentifikasi semua faktor suatu bilangan, mencari faktor ganjil terlebih dahulu, kemudian faktor genap, merupakan cara untuk menemukan submasalah yang independen.

  Contoh 2: Jika suatu solusi dapat berada di salah satu dari empat kuadran, menyelesaikan satu kuadran berarti menyelesaikan submasalah.

- **Menyelesaikan Subtujuan**

  Hal ini berbeda dengan menyelesaikan submasalah karena untuk menyelesaikan masalah utama, Anda harus mampu menyelesaikan masalah yang lebih kecil yang merupakan bagian dari masalah yang lebih besar.

  Contoh 1: Sebuah bangunan tidak dapat dibangun tanpa mempersiapkan fondasinya.

  Contoh 2: Dalam kasus Masalah 12 Koin, Anda tidak dapat menyelesaikan 12 koin sampai Anda dapat menyelesaikan 4 koin.

Contoh 3: Dalam kasus Teka-teki Keledai Merah, Anda tidak dapat membawa potongan keledai ke tujuannya sampai ia dapat melewati garis horizontal, yang menghalanginya.

- **Reduksi Masalah**

  Hal ini melibatkan pengubahan masalah menjadi masalah yang lebih sederhana atau bentuk yang lebih sederhana.

  Contoh 1: Kita menggunakan aljabar untuk menyederhanakan persamaan.

  Contoh 2: Dalam kriptoaritma, jika kita memecahkan karakter paling kiri, kita akan memiliki lebih sedikit kemungkinan untuk karakter lainnya.

- **Memecahkan Masalah Terkait**

  Ketika kita "selaras" dengan suatu masalah, kita dapat mengidentifikasi dan memecahkan masalah terkait.

- **Menggunakan Pengetahuan Sebelumnya**

  Ketika kita menduga bahwa masalah pada laptop adalah laptop tersebut terlalu panas (laptop tersebut terus bekerja dalam keadaan "dingin"), tebakan tersebut didasarkan pada pengetahuan dan pengalaman sebelumnya. Seperti kata pepatah, "pengalaman itu penting." Pengalaman tidak boleh diremehkan. Semakin banyak Anda dapat menggunakan pengetahuan dan pengalaman Anda untuk mendukung pemecahan masalah, semakin singkat dan efektif prosesnya.

  Contoh 1: Saat membuat keputusan tentang rute mana yang akan diambil ke tujuan tertentu, sebuah jalan tertentu dihindari karena diketahui sering terjadi lalu lintas padat atau sedang ada perbaikan jalan.

  Contoh 2: Mengetahui sisi miring suatu segitiga membantu memecahkan masalah mencari luas jajargenjang.

  Contoh 3: Setelah mengikuti kuliah seorang profesor sebelumnya, Anda lebih familiar dengan gaya mengajar dan soal ujiannya.

- **Hasilkan dan Uji**

  Metode pemecahan masalah ini secara harfiah merujuk pada penamaannya. Solusi dihasilkan berdasarkan batasan pada suatu masalah. Contoh umum adalah Masalah Delapan Ratu, di mana ratu-ratu "ditempatkan" di papan catur (komponen "generasi") dan kemudian "diuji" untuk mengetahui apakah mereka memenuhi batasan masalah.

  Contoh 1: Dalam Masalah Delapan Ratu, seorang ratu baru ditempatkan di baris dan kolom baru pada setiap "generasi".

  Contoh 2: Seorang pria mencatat bahwa ia meninggalkan mobilnya di dekat lift di lantai lima gedung parkir. Informasi ini membantunya menghasilkan solusi potensial untuk masalah menemukan mobilnya di gedung parkir ketika ia kembali.

- **Memaksakan Struktur: Hirarki**

  Mampu menyusun masalah selalu lebih baik daripada harus menyelesaikannya tanpa pengetahuan sebelumnya. Struktur dapat membantu mengidentifikasi subtujuan dan submasalah yang dapat dan harus diselesaikan.

Contoh 1: Dengan memiliki struktur perusahaan, jelas siapa yang bertanggung jawab atas apa.

Contoh 2: Dengan mengembangkan pohon masalah, jelas bagaimana cara melanjutkan dari satu komponen masalah ke komponen lainnya.

Contoh 3: Peta membantu mengidentifikasi titik-titik dalam perjalanan dengan tujuan mencapai suatu tujuan.

- **Memecahkan Masalah Analog (Analogi)**

  Pemecahan masalah adalah berpikir. Berpikir seharusnya melibatkan pencarian pola dan hubungan antar masalah. Misalnya, jika kita menyadari bahwa pendekatan tertentu terhadap suatu situasi telah berhasil sebelumnya, kemungkinan besar kita akan menggunakan pendekatan yang sama (atau serupa) lagi di masa mendatang.

  Dalam biografinya, Juara Dunia Catur Mikhail Botvinnik (1948–1963) menceritakan bagaimana pemerintah Rusia hanya akan mendukung penelitian catur komputernya (dengan komputer canggih) ketika ia dapat memecahkan beberapa masalah pembangkit listrik (ia adalah seorang insinyur listrik), dengan menggunakan komputer yang lebih canggih.

  Contoh 1: Mengalihdayakan bantuan. Jika pengalihdayaan telah berhasil sebelumnya dalam situasi masalah (misalnya, Anda membutuhkan desain sampul buku), kemungkinan besar Anda akan mengalihdayakan lagi.

  Contoh 2: Makan pizza sebelum tidur menyebabkan refluks asam. Oleh karena itu, Anda harus menghindari makan apa pun yang mengandung saus tomat sebelum tidur (yaitu, saus tomat berkontribusi terhadap refluks asam).

- **Munculkan "Ide Cemerlang"**

  Ini mirip dengan ungkapan "Berpikir di luar kotak". Jangan berpikir dengan cara konvensional. Terkadang hal ini dapat dibantu dengan menjauh dari masalah—mirip dengan "inkubasi" di bawah ini. Ide cemerlang tidaklah murah, dan sulit untuk memahami bagaimana seseorang dapat menemukan ide cemerlang. Bagi sebagian orang, ide cemerlang mereka muncul saat mandi. Kuncinya adalah bersikap rileks agar Anda dapat berpikir bebas tanpa tekanan. Ide yang Anda hasilkan harus dengan cara tertentu mendukung tujuan pemecahan masalah Anda.

  Contoh 1: Mobil Anda sudah tua, dan Anda ingin membeli yang baru. Lebih masuk akal untuk membeli mobil bekas yang berusia sekitar dua hingga lima tahun dan jarak tempuhnya rendah. Melakukan hal ini menghemat uang tetapi tetap menghasilkan solusi yang layak.

  Contoh 2: Seringkali dalam hidup, kita mendapati bahwa kita membuat penilaian cepat yang tidak berhasil. Meskipun dalam situasi tertentu (lihat di bawah—intuisi), mengikuti insting Anda adalah hal yang baik, terkadang lebih baik untuk menunda membuat keputusan apa pun (misalnya, "inkubasi" atau "menunda-nunda"). Atau seperti yang terkadang kita katakan dalam instruksi catur, terkadang langkah terbaik adalah tidak melakukan langkah apa pun. Ini dapat membantu dalam "menemukan ide cemerlang."

- **Intuisi**

  Intuisi bisa menjadi metode pemecahan masalah yang sangat ampuh. Intuisi bukanlah sesuatu yang kebetulan, dan tentu saja terbentuk dengan bantuan pengalaman sebelumnya. Intuisi dapat berasal dari naluri, yang mungkin muncul setelah menghadapi masalah atau situasi yang serupa. Intuisi harus dibedakan dari tebakan yang tidak berdasar. Biasanya ada dasar intuisi yang mungkin sulit diidentifikasi atau diungkapkan.

  Contoh 1: "Intuisi saya mengatakan bahwa sebaiknya Anda tidak menghadapinya." Saran ini biasanya didasarkan pada fakta, kebijaksanaan, dan pengalaman sebelumnya.

  Contoh 2: "Coba substitusikan L dengan 1 di ruas kanan persamaan." Saran seperti ini tidak datang dari kurangnya pengetahuan, tetapi lebih sering dari pengetahuan atau pengalaman sebelumnya.

- **Pertimbangkan Kondisi Batas**

  Dalam ilmu komputer, penting untuk menyelesaikan masalah untuk semua nilai X. Oleh karena itu, pengujian untuk kasus-kasus ketika X sangat besar atau sangat kecil penting untuk mendapatkan solusi yang umum dan lengkap.

  Contoh 1: Pertimbangkan ketika $X = 0$.

  Contoh 2: Pertimbangkan ketika X sangat besar.

- **Bekerja Mundur (Analisis Retrograde): Penelusuran Mundur**

  Terkadang penting untuk dapat merekonstruksi langkah-langkah yang mengarah ke situasi masalah (misalnya, dalam kasus kehilangan sesuatu atau dalam analisis kecelakaan).

  Contoh 1: Bidak catur apa yang jatuh dari kotak itu?

  Contoh 2: Apa penyebab kecelakaan mobil?

- **Inkubasi**

  Ini adalah pendekatan yang sangat direkomendasikan untuk pemecahan masalah. Pada dasarnya, pendekatan ini menyarankan untuk "tidur sambil memikirkan masalah." Terkadang, menjauh dari masalah adalah hal terbaik yang dapat Anda lakukan. Kenali kapan Anda telah cukup lama mengerjakan suatu masalah, lalu menjauhlah sejenak darinya. Meskipun Anda secara mental "menjauh" dari masalah tersebut, alam bawah sadar Anda mungkin masih mengerjakannya.

  Contoh 1: Jika Anda telah mengerjakan suatu masalah selama beberapa jam dan masih buntu, mungkin cara yang paling efektif adalah mengistirahatkannya, minum secangkir kopi (bahkan mungkin beristirahat beberapa jam), lalu kembali lagi nanti atau bahkan keesokan harinya.

- **Dekomposisi (Kelas Ekivalensi): Kasus**

  Kadang-kadang, suatu masalah dapat diselesaikan dengan memecahnya menjadi beberapa kasus yang berbeda. Jika hal ini dapat dilakukan, masalah besar dipecah menjadi masalah-masalah yang lebih kecil yang lebih mudah dikelola. Hal ini mirip dengan memecahkan submasalah.

Contoh 1: Masalah manufaktur yang besar dipecah menjadi beberapa tahap yang berurutan dan dapat dicapai.

Heuristik

Salah satu gagasan terpenting yang muncul dari karya Polya adalah gagasan heuristik atau "aturan praktis". Sebenarnya, heuristik merupakan teknik pemecahan masalah yang sangat ampuh dan digunakan secara universal. Itulah sebabnya kami menganggapnya layak mendapat bagian tersendiri. Judea Pearl menganggap topik ini begitu penting sehingga ia menulis buku lengkap tentangnya. Ketika kita mempertimbangkan bagaimana kita menjalani kehidupan sehari-hari, tampak jelas bahwa kita memang spesies yang dikendalikan secara heuristik. Heuristik juga memainkan peran penting dalam proses pengambilan keputusan dan dalam kehidupan spesies lain.

Heuristik dapat dipandang sebagai "atribut", "fitur", "faktor", "karakteristik", atau bahkan "firasat" yang berkontribusi pada proses pengambilan keputusan dalam pemecahan masalah, tetapi heuristik bukan sekadar keistimewaan atau takhayul. Mari kita pertimbangkan beberapa contoh:

Contoh 1: Dalam bisbol, ketika seorang pemain pemukul bertubuh kecil dan ringan, pemain outfield akan cenderung bergerak ke dalam, meskipun ia mungkin memiliki kekuatan yang cukup besar. Para pemain outfield menerapkan heuristik alami yang *biasanya* berhasil—tetapi terkadang tidak.

Contoh 2: Heuristik untuk mengemudi di jalan raya menyatakan bahwa jika lalu lintas padat dan kendaraan bergerak sangat lambat, Anda harus tetap di jalur kanan. Ini biasanya membantu. Selain itu, jika Anda berbicara di telepon sambil mengemudi (bebas genggam), Anda harus tetap di jalur kanan.

Contoh 3: Sebelum mengajar kelas, sebaiknya Anda datang ke kantor setidaknya dua jam sebelumnya untuk memeriksa materi yang mungkin Anda perlukan dan meninjaunya.

Contoh 4: Hindari jam sibuk! Usahakan untuk menghindari berkendara ke New York City antara pukul 07.30 dan 09.00, dan hindari berkendara keluar New York City antara pukul 16.00 dan 18.00.

Contoh 5: Aturan Brook: Berapa pun waktu yang dikatakan programmer diperlukan untuk menyelesaikan suatu pekerjaan, kalikan dengan tiga. Ini karena orang-orang tidak pandai memperkirakan waktu dan upaya yang dibutuhkan untuk menyelesaikan suatu pekerjaan—yang disebut Brooks sebagai "estimasi tanpa nyali".

Contoh 6: Saat mengembangkan program komputer, pertimbangkan data kasus umum serta kasus batas—misalnya, ketika $\mathrm{N} = 0$ dan N adalah angka yang sangat besar.

Contoh 7: Untuk Masalah Tur Ksatria, ketika diberi pilihan langkah, selalu bergerak ke tepi. Heuristik ini akan berfungsi hampir sepanjang waktu.

Contoh 8: Dalam bisbol, lebih mudah bagi pemain luar yang masuk untuk menangkap bola yang melayang daripada pemain dalam yang keluar. Oleh karena itu, pemain dalam biasanya harus tunduk kepada pemain luar.

Contoh 9: Ada heuristik "Bersikaplah agresif tetapi jangan bodoh." Kelima kata tersebut mewakili heuristik yang sangat berarti dan dapat mencakup banyak domain. Misalnya, dalam sepak bola, itu berarti agresi tanpa melakukan penalti. Dalam catur, itu berarti bermain secara aktif dengan bidak Anda tetapi tidak memberikan bidak Anda. Dalam bisbol, itu bisa berarti mencoba mencuri base tanpa dilempar keluar.

Konsep yang sama dapat digeneralisasikan ke banyak situasi dalam kehidupan. Ada kalanya Anda harus tegas atau "agresif," tetapi jangan terlalu agresif hingga tindakan Anda menjadi "bodoh." Hal ini berlaku untuk situasi sosial atau bahkan akademis. Katakanlah Anda sedang mendengarkan kuliah di kelas dan sangat tidak setuju dengan profesor. Jika Anda dapat menyampaikan maksud Anda dengan cara yang kuat, fasih, dan intelektual, tanpa bersikap kasar atau terlalu keras, dan Anda memiliki bukti pendukung yang valid untuk pandangan Anda, maka Anda lebih mungkin dianggap serius daripada seseorang yang hanya sangat agresif, blak-blakan, dan tidak sopan.

Contoh 10: Jika Anda memiliki sesuatu yang mengganggu Anda dan sudah larut malam, seringkali lebih baik "menundanya" daripada mencoba menyelesaikannya saat Anda lelah. Ini adalah heuristik "akal sehat". Faktanya, saat Anda lelah, Anda cenderung membuat keputusan yang buruk. Terkadang Anda mungkin telah melakukan atau mengatakan hal-hal saat lelah yang telah lama Anda sesali. Setelah tidur malam yang nyenyak, suasana hati Anda biasanya lebih terorganisir, optimis, dan siap menghadapi tantangan hari baru.

Contoh 11: "Jangan lakukan hari ini apa yang bisa Anda tunda sampai besok." Sungguh heuristik yang buruk! Intinya, "Tunda sampai besok apa yang perlu Anda lakukan hari ini." Namun, itu jelas merupakan ide yang buruk dan sudah tidak asing lagi bagi mereka yang terbiasa menunda-nunda.

Konsep ini bisa saja merupakan perluasan dari Contoh 10, tetapi kami tidak menganjurkannya, juga bukan itu yang ingin kami sampaikan di sini. Jangan samakan pekerjaan dengan "masalah." Jika ada pekerjaan yang sudah Anda kerjakan, bukan ide yang buruk untuk mengerjakannya sebanyak mungkin selagi pikiran Anda masih berfungsi dengan baik dan Anda tidak terlalu lelah. Dengan kata lain, selesaikan apa yang bisa Anda kerjakan dalam sekali duduk. Selalu ada biaya tambahan yang terlibat dalam meninggalkan proyek. Semakin lama Anda meninggalkan proyek, semakin banyak waktu yang Anda habiskan untuk mencoba mendapatkan kembali status, ide, dan rencana akhir Anda. Ya, jika Anda dapat menyelesaikan bagian-bagian kecil dari sebuah proyek dalam jadwal kerja yang sistematis dan teratur, itu sangat baik (inkubasi bahkan mungkin terjadi), tetapi itu tidak lebih baik daripada benar-benar menyelesaikan sesuatu. Meskipun demikian, selalu merupakan ide yang baik untuk meninjau pekerjaan yang telah Anda lakukan—baik di bawah tekanan waktu maupun saat Anda lelah. Laporan penting dari Institute of Medicine, *To Err Is Human*, mengaitkan banyak kesalahan manusia dengan kelelahan semata. Seseorang harus mampu mengenali kelelahan dan kebutuhan untuk berhenti bekerja ketika diperlukan.

Heuristik Tambahan untuk Pemecahan Masalah

Di bagian ini, kami menyajikan beberapa heuristik tambahan yang telah digunakan secara efektif oleh para pemecah masalah.

1. **Carilah masalah yang lebih sederhana dan terkait.** Hal ini niscaya akan menghasilkan pemahaman yang lebih baik tentang masalah itu sendiri.
2. **Bekerja mundur dari spesifik ke umum.** Artinya, gunakan fakta-fakta untuk mencoba menyimpulkan sifat masalah tersebut. Dalam AI, ini disebut *backward chaining*.
3. **Kata maju dari umum ke khusus.** Intinya, kita mencoba menggunakan generalisasi untuk mencapai kesimpulan yang spesifik. Misalnya, katakanlah kita menemukan satu siswa nakal di sebuah sekolah. Kemudian kita menemukan beberapa siswa nakal di sekolah tersebut. Dengan demikian, semua siswa yang kita temui adalah siswa nakal. Jadi, ketika kita menemukan siswa baru di sekolah itu, kita mungkin akan berasumsi bahwa ia adalah siswa nakal. Tentu saja, penalaran ini tidak tepat, dan dapat menimbulkan prasangka yang tidak beralasan.
4. **Persempit kondisinya.** Lihat apakah masalah dapat dipahami dan dipecahkan dengan lebih baik jika kondisinya lebih terbatas.
5. **Perluas kondisi untuk melihat apakah hal itu akan membuat masalah lebih mudah dipahami dan dipecahkan.** Metode #4 dan #5 dapat membantu dalam pemahaman masalah.
6. **Lihat apakah ada contoh tandingan yang akan membantah hipotesis Anda untuk suatu solusi.**
7. **Ubah mode konseptual.** Ini seperti inkubasi. Lihat apakah Anda dapat melihat masalah dari perspektif yang berbeda. Ini dapat mencakup menyatakan kembali masalah, menggambar diagram, atau memilih representasi alternatif.
8. **Cobalah untuk memperkirakan jawaban yang diajukan.** Selalu merupakan ide yang baik untuk menentukan apakah Anda berada di "kisaran yang tepat."
9. **Pertimbangkan apakah semua data telah digunakan.** Biasanya merupakan ide yang baik untuk melihat bagaimana data yang berbeda memengaruhi suatu solusi.
10. **Cari pola.** Pola dapat memberikan wawasan untuk solusi masalah.

2.3 JENDELA MANUSIA

Solusi untuk masalah harus berada di dalam Jendela Manusia (Gambar 2.1). Jendela Manusia adalah wilayah yang dibatasi oleh kemampuan memori manusia dan keterbatasan komputasi. Dengan demikian, solusi harus dapat dieksekusi dalam batasan memori manusia dan harus mudah dipahami. Lebih lanjut, solusi harus 100% benar dan memiliki ukuran butir yang dapat dikelola (yaitu, solusi tidak boleh terlalu intensif memori atau terlalu intensif komputasi).

Jendela Manusia menggambarkan keterbatasan kemampuan otak manusia untuk memproses informasi dan kebutuhan solusi Kecerdasan Buatan untuk berada dalam batasannya. Almarhum Donald Michie dianggap sebagai penemu gagasan ini. Agar suatu solusi berada dalam batasan Jendela Manusia, solusi tersebut harus ideal dalam hal *kebenaran, ukuran butir, kemampuan eksekusi,* dan *pemahaman*.

Gambar 2.1 Solusi Jendela Manusia.

Ketepatan suatu solusi menentukan apakah solusi tersebut 100% benar atau tidak. Solusi untuk masalah kompleks yang salah tidak akan sesuai dengan Jendela Manusia.

Ukuran butir solusi mengacu pada batasan komputasi dan memori manusia dan Jendela Manusia. Gagasan utamanya adalah bahwa solusi untuk masalah dengan kompleksitas yang cukup (misalnya, masalah tipe AI) terbatas dalam hal jumlah detail terkait komputasi dan memori yang dibutuhkan manusia untuk mengeksekusi solusi dan memahaminya. Solusi untuk masalah kompleks harus memiliki ukuran butir yang dapat dikelola—yaitu, tidak terlalu intensif komputasi (ukuran butir besar) atau intensif memori (misalnya, basis data, ukuran butir kecil).

Keterekseksusian suatu solusi diukur dari seberapa mudah seseorang dapat menemukan solusinya, dan **pemahaman** suatu solusi diukur dari seberapa baik seseorang dapat memahaminya. Meskipun kriteria ini menilai seberapa baik suatu solusi sesuai dengan Jendela Manusia, eksekusi dan pemahaman tidak selalu bergantung satu sama lain. Misalnya, beberapa solusi untuk masalah dapat dieksekusi tetapi belum tentu dipahami. Jenis solusi lain mungkin mudah dipahami orang, tetapi mereka mungkin tidak dapat mengeksekusinya.

Selain kriteria di atas, penting juga untuk mempertimbangkan tingkat detail solusi untuk masalah kompleks dalam hal representasi pengetahuannya dan menentukan apakah solusi tersebut **ekstensional** (eksplisit, detail, dan panjang) atau **intensional** (implisit, singkat, dan padat). Representasi ekstensional biasanya akan menampilkan setiap kasus, setiap contoh, dari suatu informasi, sementara representasi intensional seringkali singkat (misalnya, rumus atau ekspresi yang merepresentasikan suatu informasi).

Buku ini menyajikan solusi untuk masalah-masalah yang dipilih, yang mencakup bagian-bagian tentang Jendela Manusia untuk masing-masing masalah. Di bagian ini, berbagai solusi manusia akan dibandingkan berdasarkan serangkaian kriteria Jendela Manusia.

Kopec telah membahas beberapa masalah yang dipilih dalam mata kuliahnya dan telah mengumpulkan sejumlah solusi dan sumber daya untuk studi dan penyelesaiannya. Untuk setiap masalah kompleks di atas, kami telah menemukan dan menganalisis solusi yang telah dikembangkan. Di antara solusi-solusi tersebut, kami telah memeringkat masing-masing berdasarkan penilaian kami tentang bagaimana solusi-solusi tersebut memenuhi kriteria di atas. Dengan menggunakan justifikasi dan opini kami ini, kami telah mengevaluasi solusi-solusi yang dapat dianggap sebagai "Most Human Window Compatible (MHWC)" dan "Least Human Window Compatible (LHWC)" untuk setiap permasalahan kompleks. Kami juga telah mencoba menemukan solusi untuk permasalahan yang dapat dianggap "terbaik" untuk sebuah mesin dengan mengevaluasi efisiensi runtime dan ruang dari algoritma-algoritma tertentu. Terakhir, Christopher Pileggi juga telah mencoba menentukan cara terbaik untuk merepresentasikan solusi-solusi ini.

Kami melihat *solusi masalah* dari sudut pandang berikut:

1. Solusi intensional versus ekstensional
2. Pilihan representasi pengetahuan
3. Kesesuaian dalam hal memenuhi batasan Jendela Manusia (yaitu, ketepatan, ukuran butir, eksekusi, dan pemahaman)
4. Metode pemecahan masalah
5. Fleksibilitas
6. Cara penyampaian
7. Kemungkinan optimalitas

Menemukan solusi yang paling dan paling tidak kompatibel dengan Jendela Manusia untuk masalah kompleks berdasarkan faktor-faktor ini dapat bermanfaat dalam beberapa cara. Misalnya, menemukan solusi terbaik dan paling sesuai untuk manusia dapat sangat bermanfaat bagi studi dan pemahaman pemecahan masalah. Hal yang sama berlaku untuk menemukan solusi optimal untuk masalah mesin. Solusi-solusi ini tidak hanya akan berlaku untuk masalah yang dimaksudkan, tetapi juga dapat menyediakan gerbang untuk memecahkan masalah manusia dan komputasi lainnya. Solusi-solusi ini juga dapat digunakan sebagai alat pengajaran yang efektif.

2.4 KRITERIA JENDELA MANUSIA DAN PERINGKAT SOLUSI

Untuk setiap masalah kompleks dalam buku ini, kami memeringkat solusi yang dipertimbangkan berdasarkan kriteria berikut:

Intensional versus Ekstensional: Solusi untuk suatu masalah akan termasuk dalam kategori intensional atau ekstensional. Solusi *ekstensional* adalah solusi yang eksplisit, terperinci, dan biasanya sangat panjang, sementara solusi *intensional* adalah solusi yang biasanya implisit, singkat, dan padat. Kami menunjukkan apakah suatu solusi bersifat ekstensional atau intensional.

Intensionalitas/Ekstensionalitas: Meskipun menentukan apakah suatu solusi bersifat intensional atau ekstensional adalah keharusan, hal itu tidak menjadikan kategori tersebut hitam dan putih. Solusi intensional dapat diukur berdasarkan seberapa intensionalnya, dan solusi ekstensional dapat diukur berdasarkan seberapa ekstensionalnya. Misalnya, perhatikan dua solusi ekstensional untuk Masalah Misionaris dan Kanibal: grafik dengan gambar dan tabel dengan petunjuk. Grafik dengan gambar dapat dianggap lebih ekstensional daripada tabel. Bahkan, kita dapat mengatakan bahwa suatu solusi bahkan lebih ekstensional jika, misalnya, melibatkan pengamatan langsung para misionaris dan kanibal menyeberangi tepi sungai. Kita akan memberi setiap solusi peringkat antara 1 dan 10 untuk menilai seberapa intensional atau ekstensional suatu solusi, dengan 10 sebagai tingkat intensionalitas atau ekstensionalitas tertinggi.

Pilihan Representasi Pengetahuan: Saat menganalisis dan memeringkat solusi untuk mengidentifikasi mana yang dianggap "paling sesuai", penting untuk memperhatikan bagaimana setiap solusi direpresentasikan. Dengan menganalisis peringkat dan memperhatikan bagaimana suatu masalah direpresentasikan, kesimpulan dapat ditarik tentang representasi mana (misalnya, tabel, pohon, pseudocode, dll.) yang dapat dianggap ideal untuk manusia. Untuk setiap solusi, kita akan menunjukkan pilihan representasi yang digunakan dan menganalisis pengaruh serta atributnya.

Memenuhi Kendala Jendela Manusia: Hal ini penting untuk menentukan apakah suatu solusi berada dalam Jendela Manusia atau tidak. Artinya, apakah solusi tersebut memenuhi setiap batasan berikut? Untuk setiap solusi, kami akan menjawab pertanyaan ini dengan "ya" atau "tidak". Namun, untuk menjawab pertanyaan ini sepenuhnya, pertama-tama kami harus menganalisis setiap kriteria berikut:

- **Ketepatan:** Untuk memahami suatu solusi, solusi tersebut harus benar. Jika tidak, solusi tersebut pada dasarnya tidak bermakna. Untuk setiap solusi, kami menunjukkan apakah solusi tersebut benar, salah, atau benar tetapi dengan siklus. Artinya, solusi tersebut memberikan jawaban yang benar tetapi dengan beberapa langkah tambahan (misalnya, penelusuran balik).
- **Ukuran Butir:** Pertukaran antara jumlah memori dan komputasi yang diperlukan untuk menjalankan suatu solusi sangat penting agar solusi tersebut berada dalam batasan Jendela Manusia. Terlalu banyak atau terlalu sedikit salah satu dari keduanya dapat membuat solusi menjadi sangat tidak jelas sehingga tidak dapat dipahami (ukuran butir terlalu kecil) atau tidak dapat dieksekusi (ukuran butir terlalu besar). Untuk setiap solusi, kami memeringkat aspek ini berdasarkan ukuran butirnya. *Sangat Kecil* berarti tingkat memori yang dibutuhkan untuk menjalankan solusi terlalu tinggi, sementara *Sangat Besar* berarti tingkat komputasi yang dibutuhkan untuk menjalankan solusi terlalu tinggi. *Ideal* berarti terdapat keseimbangan antara seberapa banyak seseorang perlu menghafal dan berhitung agar dapat memahami dan menjalankan solusi.
- **Eksekutabilitas:** Eksekutabilitas suatu solusi adalah ukuran seberapa baik seseorang dapat benar-benar menjalankannya. Jumlah langkah yang dibutuhkan, sumber daya yang dibutuhkan (misalnya, tabel), dan latar belakang yang dibutuhkan (misalnya, matematika) hanyalah beberapa faktor yang menentukan apakah seseorang dapat menjalankan solusi tertentu atau tidak. Setiap solusi diberi peringkat berdasarkan eksekutabilitasnya dari 1 hingga 10, 10 berarti sangat dapat dieksekusi.
- **Komprehensibilitas:** Meskipun suatu solusi mungkin dapat dieksekusi, hal itu tidak selalu berarti seseorang dapat memahaminya. Komprehensibilitas seseorang terhadap suatu solusi adalah seberapa baik ia dapat memahaminya. Hal ini dapat ditentukan oleh faktor-faktor seperti gambar, warna, ukuran solusi, organisasi data, dan sebagainya. Karena kami tidak bereksperimen dengan subjek manusia untuk analisis Jendela Manusia, penilaian didasarkan pada pengalaman kami dengan orang-orang dan sikap mereka terhadap masalah yang kompleks. Basis penilaian kami akan berkisar antara pemecah masalah yang mahir dan bukan pemecah masalah, dan kami akan menentukan rata-rata yang memungkinkan di antara keduanya. Setiap solusi diberi peringkat dari 1 hingga 10, 10 berarti sangat mudah dipahami.

Metode Pemecahan Masalah: Sebagaimana kita harus memperhatikan pilihan representasi suatu solusi, penting juga untuk memperhatikan metode pemecahan masalah yang digunakan. Meskipun terdapat beragam metode yang dapat dipilih, rata-rata orang mungkin lebih menyukai satu metode daripada metode lainnya. Misalnya, seseorang mungkin merasa jauh lebih mudah untuk memecahkan masalah dengan memecahnya menjadi subtujuan daripada menggunakan pendekatan top-down. Dengan demikian, membandingkan peringkat solusi dengan metode yang digunakan untuk menyelesaikannya dapat menentukan metode mana yang lebih disukai seseorang. Untuk setiap solusi, metode yang digunakan untuk menyelesaikannya ditunjukkan.

Fleksibilitas Solusi: Ada kemungkinan bahwa suatu solusi tertentu dapat direpresentasikan secara berbeda. Fleksibilitas suatu solusi adalah seberapa mudah solusi tersebut dapat direpresentasikan dengan cara lain, tanpa mengubah metode pemecahan masalah yang digunakan atau mengurangi kesesuaiannya dalam hal memenuhi batasan Jendela Manusia. Namun, solusi dapat diubah agar lebih sesuai, terutama dalam hal pemahamannya. Misalnya, gambar hitam-putih mungkin lebih mudah dipahami jika berwarna. Setiap solusi diberi peringkat fleksibilitasnya dari 1 hingga 10, dengan 10 berarti sangat fleksibel.

Cara Penyampaian: Atribut penting lain dari sebuah solusi adalah seberapa "portabel"-nya. Misalnya, pertimbangkan dua solusi yang sepenuhnya mudah dipahami—satu yang dapat disajikan dengan tangan dan satu yang hanya dapat dijalankan melalui pembuat gambar di komputer. Solusi yang dapat digambar dengan tangan lebih portabel karena dapat digunakan di mana saja untuk memecahkan masalah. Di sisi lain, hal ini dapat memengaruhi kemampuan solusi untuk memenuhi batasan Jendela Manusia. Setiap solusi dinilai berdasarkan kemungkinan cara penyampaiannya.

Optimalitas yang Mungkin: Sifat yang membuat suatu solusi patut dipuji secara keseluruhan adalah optimalitasnya. Artinya, solusi tersebut dijalankan dalam langkah sesedikit mungkin. Hal ini tidak hanya mengurangi ukuran masalah, tetapi juga memberi tahu kita apa solusi terbaik yang mungkin diperlukan. Kita menunjukkan apakah suatu solusi optimal atau tidak.

Bersamaan dengan pemeringkatan ini, masing-masing dari 10 masalah ini diperingkat berdasarkan dua faktor tambahan:

Tingkat Kesulitan: Ini mengukur seberapa sulit kita memandang setiap masalah berdasarkan studi kita terhadap solusinya. Peringkat didasarkan pada seberapa sulit masalah yang kami temukan setelah mencoba menyelesaikannya dan seberapa sulit menurut kami bagi orang lain. Setiap solusi diberi peringkat dari 1 hingga 10, dengan 10 berarti sangat sulit.

Kompleksitas: Tidak hanya penting untuk mengukur seberapa mudah atau sulit bagi seseorang untuk memahami suatu solusi, tetapi juga penting untuk mengukur seberapa mudah atau sulit bagi mesin. Kompleksitas setiap masalah dari solusi mesin terbaik ditentukan dalam notasi Big-Oh.

2.5 KLASIFIKASI

Tabel 2.1 mengklasifikasikan masalah berdasarkan jenis metode pemecahan masalah yang digunakan. Setiap masalah dibahas berdasarkan latar belakang historis, pilihan representasi yang sesuai, teknik pemecahan masalah yang digunakan, solusinya, pendekatan pemecahan masalah manusia, analisis Jendela Manusia, dan solusi mesin terbaik.

Tabel 2.1 Klasifikasi Masalah.

NO	SOAL	JENIS MASALAH
1	Soal Misionaris dan Kanibal	Logika, pemenuhan kendala, pencarian
2	Soal 12 Koin	Logika, pengoptimalan informasi
3	Kriptaritmetika	Kombinatorika, pemenuhan kendala, penyelesaian submasalah
4	Teka-teki Keledai Merah	Teka-teki blok geser, pencarian, kepuasan kendala, pemecahan submasalah
5	Teka-teki ke-15	Teka-teki blok geser, pencarian, heuristik, pemenuhan kendala
6	Masalah Tur Ksatria	Pencarian, heuristik
7	Dalang	Logika, pengoptimalan informasi
8	Masalah Monty Hall	Logika, probabilitas
9	Kubus Rubik	Subtujuan, pola, submasalah
10	Dilema Tahanan	Teori permainan, Ekuilibrium Nash, Optimalitas Pareto, skema
11	a. Kartu/Koin dalam Kegelapan b. Sepuluh Bajak Laut dan Emas Mereka c. Masalah Jabat Tangan Halmos d. Masalah Kursi Pesawat e. Masalah Ulang Tahun	Probabilitas, logika Skema, logika Permutasi, logika Probabilitas, logika Probabilitas, logika

BAB 3
PERMASALAHAN MISIONARIS DAN KANIBAL

3.1 PENDAHULUAN

Permasalahan Misionaris dan Kanibal adalah teka-teki penyeberangan sungai dan merupakan varian dari Permasalahan Suami Cemburu (varian lainnya adalah Permasalahan Saudara Laki-laki dan Perempuan), yang pertama kali muncul dalam manuskrip abad kesembilan yang awalnya berjudul *Propositiones ad Acuendos Juvenes* (Permasalahan untuk Mengasah Anak Muda). Permasalahan penyeberangan sungai biasanya melibatkan pengangkutan benda dari satu tepi sungai ke tepi sungai lainnya, menggunakan perahu yang dapat mengangkut sejumlah benda. Tujuannya adalah mengangkut benda-benda tersebut secara optimal (dengan jumlah perjalanan sesedikit mungkin) sambil memenuhi beberapa *batasan*.

Sekitar abad ketiga belas, Permasalahan Suami Cemburu menjadi populer di seluruh Eropa utara. Permasalahan ini melibatkan suami dan istri sebagai entitas, dengan batasan bahwa jumlah suami tidak boleh melebihi jumlah istri di kedua tepi sungai. Pada akhir abad kesembilan belas, masalah ini bangkit kembali dan dikenal sebagai Masalah Misionaris dan Kanibal. Aturannya persis sama dengan Masalah Suami yang Cemburu, tetapi istri digantikan dengan misionaris dan suami digantikan dengan kanibal.

Saat ini, Masalah Misionaris dan Kanibal banyak digunakan dalam bidang Kecerdasan Buatan. Masalah ini dianggap penting karena beberapa alasan. Salah satunya adalah karena merupakan contoh masalah mainan, yang merupakan masalah yang tampaknya tidak memadai dan tidak layak dipelajari atau dipecahkan tetapi membantu menjelaskan teknik pemecahan masalah yang lebih umum. Alasan lainnya adalah karena penggunaannya dalam studi representasi masalah. Ini, seperti yang ditunjukkan oleh ilmuwan komputer Saul Amarel, merupakan contoh utama dari fakta bahwa "pilihan representasi yang tepat mampu memberikan efek spektakuler pada efisiensi pemecahan masalah".

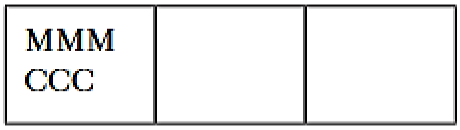
Secara keseluruhan, Masalah Misionaris dan Kanibal adalah masalah yang relatif sederhana. Solusi apa pun optimal jika dapat memindahkan keenam misionaris dan kanibal ke tepi seberang dalam 11 langkah dengan tetap mematuhi batasan masalah. Dari analisis masalah, dapat disimpulkan bahwa ada empat cara untuk mencapai keadaan tujuan dalam 11 langkah.

Permasalahannya dinyatakan sebagai berikut: Diberikan tiga misionaris dan tiga kanibal di tepi barat sungai, temukan cara untuk mengangkut semua orang dengan aman ke seberang sungai. Batasan masalah tersebut adalah:

1. Perahu hanya dapat mengangkut dua orang pada satu waktu.
2. Pada suatu titik waktu (keadaan), jumlah kanibal tidak boleh melebihi jumlah misionaris.

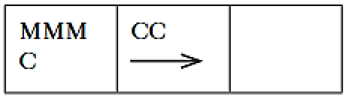
3.2 MEMILIH REPRESENTASI YANG TEPAT

Permasalahan ini cukup mudah dipahami dan dapat diselesaikan oleh orang dewasa maupun anak-anak. Namun, memilih urutan transformasi yang tepat dari keadaan awal ke keadaan tujuan, dengan tetap memperhatikan batasan permasalahan, dapat menjadi tantangan. Pengalaman telah menunjukkan bahwa pemilihan representasi sangat erat kaitannya dengan keberhasilan penyelesaian masalah. Representasi tidak hanya harus menggambarkan dengan jelas berbagai keadaan dalam permasalahan, tetapi juga pergerakan dan transisi dari satu keadaan ke keadaan lainnya. Gambar berikut menunjukkan salah satu cara untuk merepresentasikan keadaan awal:

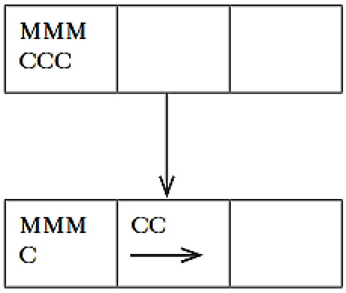


Pada gambar ini, ketiga kotak masing-masing mewakili tepi barat, sungai, dan tepi timur. M berarti misionaris, dan C berarti kanibal.

Karena perahu awalnya diam, perahu tersebut tidak ditampilkan. Saat perahu bergerak, perahu tersebut dapat diwakili oleh panah yang menunjukkan arah pergerakannya. Dengan demikian, gambar berikut mewakili dua kanibal yang diangkut dari tepi barat sungai ke tepi timur:



Transisi dapat ditunjukkan sebagai berikut:



Keadaan bermasalah dapat direpresentasikan dengan cara lain. Misalnya, keadaan awal dapat direpresentasikan sebagai serangkaian tiga angka, dengan angka pertama mewakili jumlah misionaris, angka kedua jumlah kanibal, dan angka ketiga mewakili ada atau tidaknya perahu. Namun, ini hanya akan merepresentasikan keadaan di satu sisi sungai.

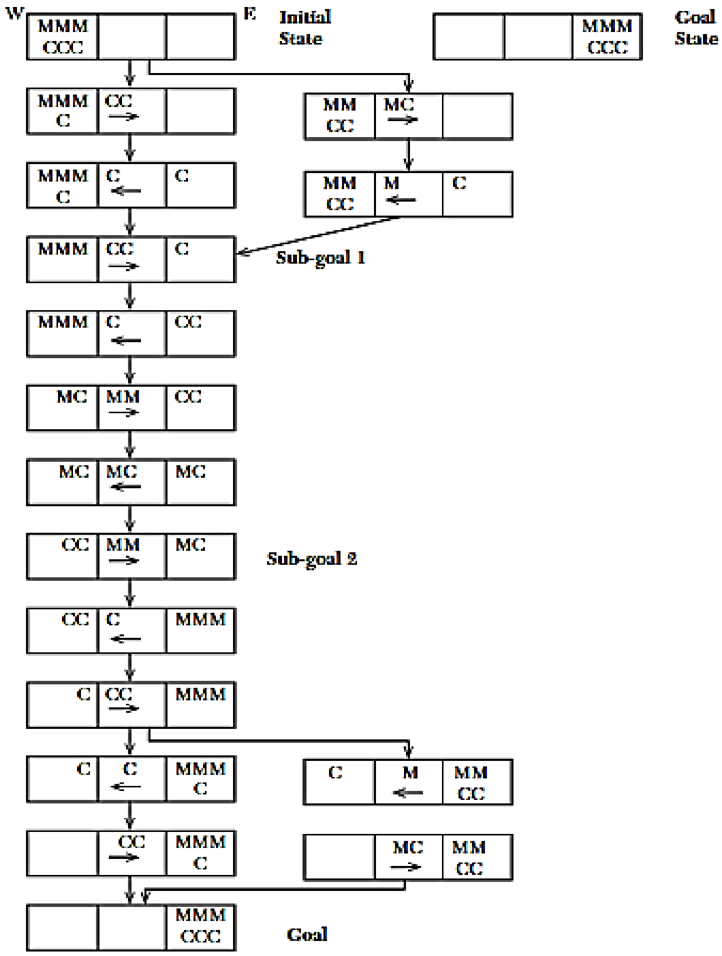
Misalnya, 331 merepresentasikan keadaan awal di tepi barat dengan tiga misionaris, tiga kanibal, dan sebuah perahu. Demikian pula, keadaan di tepi barat setelah mengangkut dua kanibal ke sisi lain dapat direpresentasikan sebagai 310.

Transisinya dapat ditunjukkan sebagai:

331 310

Jelas bahwa representasi pertama lebih ekstensional dibandingkan dengan yang kedua. Kita dapat menggunakan deskripsi verbal atau notasi formal untuk menggambarkan keadaan, gerakan, dan transisi antar gerakan, tetapi notasi grafis terlihat lebih transparan dan memudahkan visualisasi keadaan permasalahan.

Dari keadaan awal, mungkin tampak ada banyak kemungkinan gerakan. Namun, ketika kita menggali lebih dalam dan mempertimbangkan batasan permasalahan, dapat dilihat bahwa dua entitas harus berada di dalam perahu untuk membawa perahu kembali ke tepi barat. Selain itu, untuk menghindari batasan jumlah kanibal yang melebihi misionaris, baik dua kanibal atau satu misionaris dan satu kanibal dapat diangkut pada langkah pertama. Langkah-langkah selanjutnya yang mengarah ke keadaan aman ditunjukkan pada Gambar 3.1



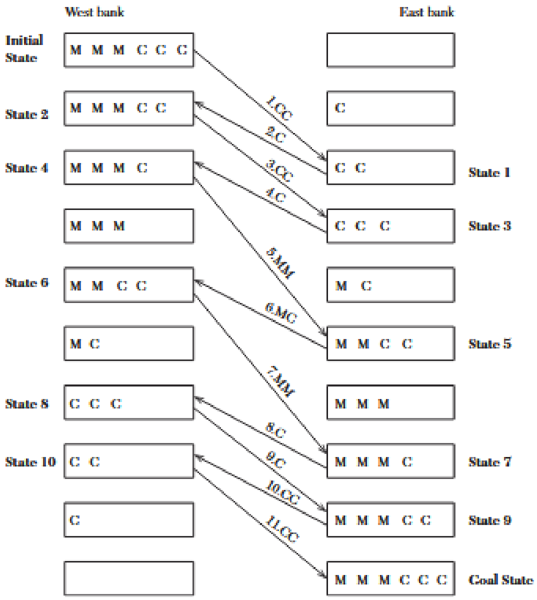
Gambar 3.1 Diagram Pohon

3.3 DIAGRAM TRANSISI KEADAAN

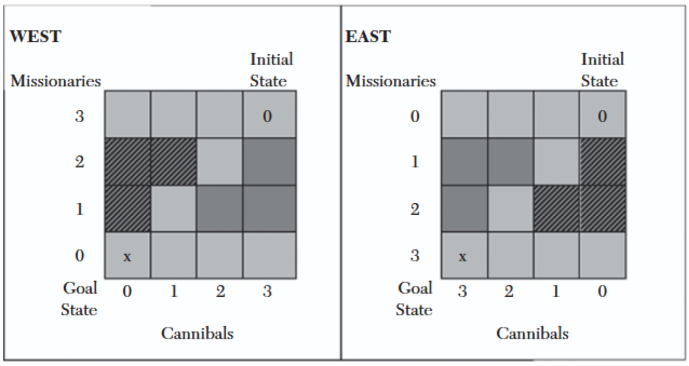
Salah satu cara untuk merepresentasikan permasalahan ini dan berbagai langkahnya adalah dalam format pohon pencarian, seperti yang ditunjukkan pada Gambar 3.1. Minimal sebelas langkah diperlukan untuk mencapai keadaan tujuan.

Cara lain untuk merepresentasikan solusi ini adalah *diagram transisi keadaan* (Gambar 3.1). Kotak di sisi kiri menggambarkan semua keadaan aman di tepi barat, dan kotak di sisi kanan menggambarkan keadaan yang sesuai di tepi timur. Hanya keadaan yang aman untuk kedua sisi sungai yang ditampilkan. Panah mewakili transisi dari satu keadaan ke keadaan lain serta arah perahu.

Langkah-langkah yang direpresentasikan pada Gambar 3.2juga dapat diterjemahkan ke dalam *matriks ruang keadaan masalah*, seperti yang ditunjukkan pada Gambar 3.3. Kedua matriks tersebut menggambarkan himpunan keadaan masalah di kedua tepi sungai. Sel yang diarsir lebih gelap menunjukkan keadaan tidak aman di mana jumlah kanibal melebihi jumlah misionaris, dan sel yang diarsir lebih terang menunjukkan keadaan aman. Sel-sel bergaris menggambarkan keadaan yang sesuai dengan keadaan tidak aman di seberang sungai. Misalnya, memiliki dua misionaris dan satu kanibal di tepi timur merupakan keadaan yang aman. Namun, ketika Anda mempertimbangkan keadaan yang sesuai di tepi barat (yaitu, satu misionaris dan dua kanibal), Anda tahu bahwa itu adalah keadaan yang tidak aman untuk tepi barat, sehingga keadaan ini tidak valid dan harus dihindari. Oleh karena itu, solusinya melibatkan transisi dari keadaan awal (sel bertanda 0) ke keadaan tujuan (sel bertanda X).

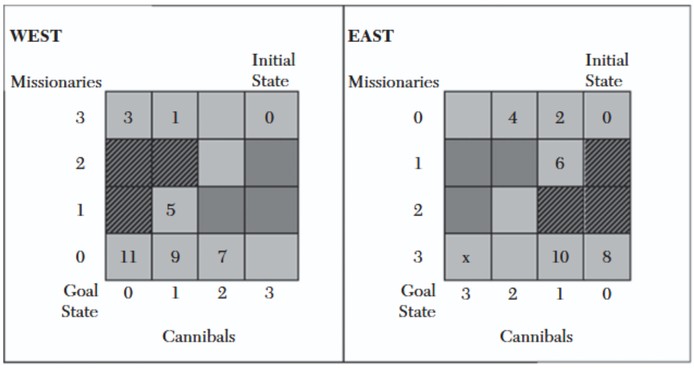


Gambar 3.2 Diagram Transisi Keadaan.



Gambar 3.3 Matriks Ruang Keadaan Masalah.

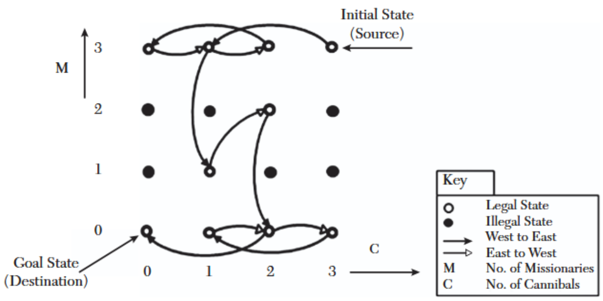
Pergerakan diberi nomor dan disebutkan dalam sel yang mewakili keadaan setelah pergerakan dilakukan. Pergerakan yang dilakukan di tepi barat (yaitu, keberangkatan perahu) direpresentasikan dengan angka ganjil dalam matriks berlabel BARAT, dan pergerakan yang dilakukan di tepi timur direpresentasikan dengan angka genap dalam matriks berlabel TIMUR. Solusi dalam bentuk matriks ditunjukkan pada Gambar 3.4.



Gambar 3.4 Solusi Menggunakan Matriks Ruang Keadaan Masalah.

Cara lain untuk merepresentasikan matriks ini digambarkan pada Gambar 3.5. Gambar ini merepresentasikan ruang keadaan masalah tepi barat setiap saat. Pergerakan bernomor ganjil (dari tepi barat) direpresentasikan dengan panah berujung hitam, dan pergerakan bernomor genap (dari tepi timur) direpresentasikan dengan panah berujung putih.

Gambar 3.5 membantu kita memahami aspek yang sangat penting dari permasalahan ini: *simetri dalam pembalikan waktu*. Ini berarti bahwa pergerakan dari awal dan akhir solusi serupa dan berlawanan. Artinya, jika kita bekerja mundur dari solusi, kita memindahkan kanibal dan misionaris dari tepi timur ke tepi barat. Sifat ini dapat membantu kita melakukan *pencarian dua arah* dari permasalahan ini dan dengan demikian mengurangi pencarian hingga dua kali lipat.



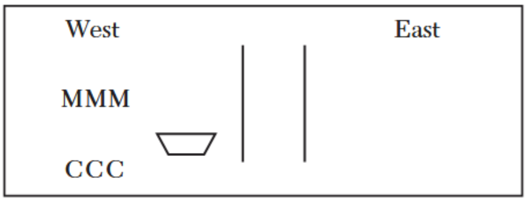
Gambar 3.5 Transisi Keadaan dalam Ruang Keadaan Masalah.

3.4 PEMECAHAN MASALAH MANUSIA

Sebagai bagian dari penelitian kami, kami mengamati siswa yang mencoba memecahkan Masalah Misionaris dan Kanibal. Mereka menghabiskan sekitar 20 menit untuk menyelesaikannya, dan beberapa berhasil menemukan solusinya. Kebanyakan dari mereka menggunakan diagram yang digunakan instruktur saat memperkenalkan masalah. Beberapa mencoba representasi lain, tetapi representasi bergambar tampaknya yang paling mudah mereka pahami.

Sebagian besar siswa menyadari bahwa subtujuan utama dari masalah ini adalah untuk mengisolasi para misionaris dan kanibal terlebih dahulu. Kedua, karena sifat keadaan awal, wajar untuk memindahkan semua kanibal ke seberang sungai terlebih dahulu. Namun, itu hanyalah solusi sementara. Langkah selanjutnya adalah memindahkan semua misionaris ke seberang sungai. Namun, ini melibatkan pemindahan para kanibal kembali ke keadaan semula untuk menjemput para misionaris.

Gambar 3.6 menunjukkan diagram yang digunakan instruktur saat ia mempresentasikan masalah.



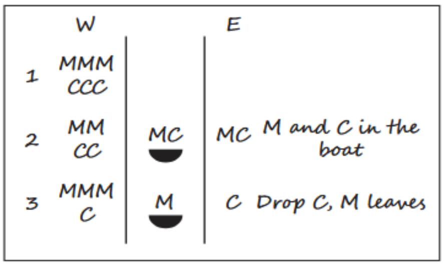
Gambar 3.6 Diagram Instruktur Untuk Masalah Misionaris Dan Kanibal.

Berikut adalah cara tiga siswa menyelesaikan masalah tersebut.

Siswa 1:

Siswa menggunakan diagram yang mirip dengan diagram instruktur (Gambar 3.7). Diagram tersebut merepresentasikan keadaan setelah setiap gerakan.0 Penjelasan tertulis untuk setiap gerakan disertakan dalam diagram.

Siswa memulai dengan dua entitas di atas perahu, tetapi menyadari bahwa yang satu harus tetap di belakang dan yang lainnya harus kembali untuk menjemput orang lain.
Siswa memulai dengan M dan C, tetapi meninggalkan solusi setelah dua langkah.



Gambar 3.7 Solusi Parsial Siswa 1

Percobaan berikutnya dilakukan dengan memindahkan dua C terlebih dahulu, tetapi kembali membatalkan percobaan setelah tiga langkah. Pada percobaan ketiga, dua M dipindahkan terlebih dahulu, tetapi hal ini gagal memenuhi batasan yang diberikan oleh soal pada langkah kedua.

Siswa kembali ke percobaan pertama dan mencoba lagi, kali ini mencapai langkah keenam dan berhasil memindahkan satu C dan satu M ke seberang sungai. Waktu yang diberikan telah habis (Gambar 3.8).

	W	E	
0	MMM CCC		
1	MM CC	MC	M and C leave
2	MMM CC	C	M leaves
3	MMM	CCC	CC leaves
4	MMM C	CC	C leaves
5	MC	MM CC	MM leave
6	MM CC		MC leave

Gambar 3.8 Solusi Siswa 1.

Kesimpulan

Siswa mencoba tiga kemungkinan langkah pertama, dan setelah mencapai jalan buntu, ia belajar tentang langkah yang baik dan buruk. Pada percobaan ketiga, siswa telah mengevaluasi langkah terbaik dari tiga kemungkinan langkah pertama. Pada percobaan keempat, siswa melakukan langkah ini dan melanjutkan lebih jauh, dengan hati-hati menghindari langkah yang melanggar batasan soal. Siswa telah mengumpulkan pengetahuan ini dari tiga percobaan coba-coba sebelumnya.

Siswa pertama-tama mencoba memahami soal dengan menggunakan teknik coba-coba dan merencanakan percobaan berikutnya.

Siswa 2:

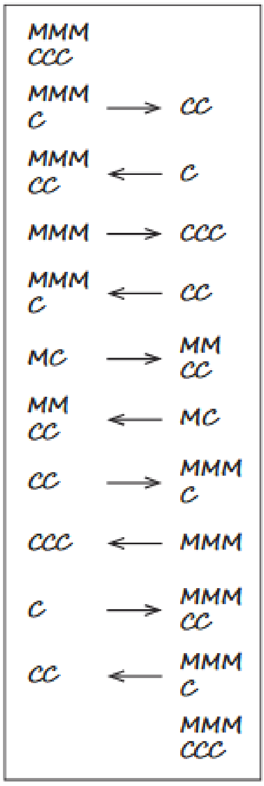
Siswa menggunakan huruf M dan C untuk masing-masing mewakili misionaris dan kanibal, dan menggunakan panah untuk menggambarkan arah perahu.

Tidak ada representasi eksplisit lain dari gerakan tersebut yang ditampilkan.

Pertama, semua kanibal dipindahkan ke tepi timur sungai.

Kemudian semua kanibal dikembalikan ke tepi barat, dan semua misionaris dibawa ke tepi timur, dengan hati-hati menghindari pelanggaran batasan masalah.

Akhirnya, setelah semua misionaris dipindahkan ke tepi timur, para kanibal dipindahkan satu per satu ke tepi timur (Gambar 3.9).



Gambar 3.9 Solusi Siswa 2.

Kesimpulan

Siswa menyelesaikan soal tepat waktu. Tampaknya ia menyadari bahwa salah satu subtujuannya adalah memisahkan C dari M dengan memindahkan semua M atau semua C terlebih dahulu. Namun, saat memindahkan M, beberapa M dan beberapa C terkadang harus dipindahkan bersama untuk menjaga keseimbangan agar jumlah C tidak melebihi jumlah M. Siswa menyadari bahwa memindahkan C kembali ke tepi barat diperlukan agar semua M berada di tepi timur.

Siswa 3:

Siswa 3 menggunakan format tabel dengan tiga kolom untuk merepresentasikan permasalahan dan solusinya. Kolom pertama menggambarkan negara bagian di tepi barat, kolom kedua menggambarkan negara bagian di tepi timur, dan kolom ketiga menggambarkan langkahnya.

Siswa memulai dengan memindahkan seorang misionaris dan seorang kanibal pada langkah pertama, lalu memindahkan semua kanibal ke tepi timur.

Setelah ini, siswa melewati beberapa langkah, mencapai langkah kedua hingga terakhir, lalu langkah terakhir, dan tidak menunjukkan langkah-langkah tersebut dalam tabel.

Dua langkah terakhir merupakan kebalikan dari dua langkah pertama (Gambar 3.10).

3M 3C		MC →
2M 2C	M C	M ←
3M 2C	C	2C →
3M	3C	
M C	2M 2C	
	3M 3C	

Gambar 3.10 Solusi Siswa 3.

Kesimpulan

Mungkin saja siswa tersebut mampu mengidentifikasi bahwa solusi ini memiliki simetri pembalikan waktu sehingga melompat ke beberapa langkah terakhir dan mencoba mencari tahu atau menyerah mencari tahu langkah-langkah di antaranya. Tampaknya ada upaya untuk menggunakan pencarian dua arah, karena keadaan akhir disebutkan dengan beberapa langkah di tengah yang hilang. Mungkin saja selain mencari dari keadaan awal menuju keadaan tujuan, siswa tersebut bekerja mundur menuju tujuan.

Analisis Solusi

Kami menyadari akan lebih mudah untuk menyelesaikan masalah jika kita memisahkan misionaris dan kanibal. Ini dapat dilakukan dengan memindahkan semua misionaris atau semua kanibal ke seberang sungai terlebih dahulu. Mencoba memindahkan semua misionaris terlebih dahulu tidak mungkin karena setelah memindahkan dua di antaranya, kita masuk ke keadaan yang tidak aman. Memindahkan semua kanibal ke seberang sungai dimungkinkan. Ini dicapai setelah empat langkah pertama. Namun, setidaknya satu kanibal harus kembali untuk menjemput para misionaris. Sekarang kita dapat mulai memindahkan para misionaris ke seberang sungai. Langkah keenam sangat penting. Langkah ini menjaga keseimbangan dengan memindahkan seorang misionaris dan seorang kanibal ke tepi barat untuk mendapatkan misionaris terakhir dari sisi tersebut pada langkah ketujuh, sekaligus memastikan kondisi aman di tepi timur.

Perhatikan bahwa langkah ini tampaknya menjauhkan dari solusi (yaitu, langkah ini menyebabkan jarak antara kondisi saat ini dan kondisi tujuan bertambah). Empat langkah berikutnya mengembalikan kanibal yang tersisa ke tepi timur.

Melalui analisis, ditemukan bahwa orang-orang membagi masalah menjadi tiga atau empat submasalah utama. Jelas dari solusi sebelumnya bahwa empat langkah pertama melibatkan pemindahan semua kanibal ke tepi timur. Tiga langkah berikutnya melibatkan pemindahan kembali kanibal secara hati-hati untuk mengambil semua misionaris, sambil menyeimbangkan jumlah misionaris dan kanibal di kedua sisi sungai dengan hati-hati. Terakhir, empat langkah terakhir kembali memindahkan semua kanibal ke tepi timur.

3.5 ANALISIS SOLUSI HUMAN WINDOW

Setelah melakukan riset, kami telah mengumpulkan sejumlah solusi untuk Masalah Misionaris dan Kanibal untuk digunakan dalam analisis kami terhadap Human Window. Di bagian ini, kami menyajikan temuan dan analisis kami atas temuan tersebut. Kami juga menyajikan solusi yang telah kami tentukan sebagai yang paling dan paling tidak kompatibel dengan Human Window, serta penjelasan mengapa solusi tersebut dianggap demikian.

Saat menganalisis solusi (Tabel 3.1), kita dapat melihat adanya beragam kemungkinan representasi yang dapat digunakan untuk mendeskripsikan dan menyelesaikan masalah ini. Satu detail penting yang perlu diperhatikan adalah bahwa setiap solusi bersifat ekstensional. Hal ini mungkin disebabkan oleh ruang keadaan masalah yang relatif kecil, yang memungkinkan untuk menampilkan solusi dalam gambar, grafik, tabel, dan sebagainya dengan ukuran yang wajar. Hanya ada 16 keadaan hukum secara total, dan hanya keadaan hukum tersebut yang diperlukan untuk mendeskripsikan solusi. Alasan lainnya adalah karena masalah tersebut bukan bagian dari masalah yang lebih besar. Artinya, akan selalu ada tiga misionaris dan tiga kanibal dalam masalah tersebut, dan angka-angka ini tidak bervariasi.

Kita dapat melihat bahwa mayoritas solusi berupa tabel atau representasi grafis. Hal ini kemungkinan besar karena ini adalah masalah yang melibatkan visualisasi, karena terdapat lingkungan dengan dua jenis entitas yang berinteraksi dengannya. Selain itu, grafik dan tabel membantu menyediakan visualisasi yang meminimalkan kebutuhan akan deskripsi.

Secara keseluruhan, solusi-solusi tersebut memiliki peringkat yang beragam untuk setiap kategorinya. Beberapa lebih dapat dieksekusi daripada dipahami dan beberapa lebih mudah dipahami daripada dieksekusi. Namun, solusi yang berisi gambar atau grafik umumnya lebih sesuai dengan Jendela Manusia. Dapat juga dilihat bahwa solusi-solusi tersebut memiliki peringkat ekstensionalitas yang lebih tinggi, sementara solusi yang direpresentasikan oleh teks atau tabel memiliki peringkat yang relatif rendah. Terakhir, solusi-solusi ini menggunakan representasi grafis/visual sebagai teknik pemecahan masalah utamanya, sementara solusi lainnya, terutama yang direpresentasikan oleh tabel, lebih menggunakan pendekatan top-down.

Kesulitan
2/10

kerumitan
O(N)

Nama	Int atau Ext?	Intens./Ekstens.	Repr.	Jendela Manusia?	Benar?	Ukuran Butir	Eksekusi	Keterpahaman	Metode Pemecahan Masalah	Fleksibilitas	Mode Konversi	Optimal?	Total
Aturan Kendala	Ekst.	1/10	Tabel	Y	Y	Kecil	6/10	2/10	Kepuasan Kendala	5/10	Pembuat Tabel; Manual	Y	14/40
Diagram Copin	Ekst.	5/10	Diagram Grafik	Y	Y	Kecil	9/10	6/10	Top-Down; Repr. Grafik	8/10	Grafik; Gambar; Pembuat; Manual	Y	28/40
Permainan	Ekst.	9/10	Gambar	Y	Y	Ideal	10/10	9/10	Top-Down; Repr. Visual	10/10	Gambar; Pembuat; Manual	Y	38/40
Diagram Logika	Ekst.	3/10	Diagram Status	Y	Y	Kecil	7/10	5/10	Repr. Grafik	7/10	Grafik; Gambar; Pembuat; Manual	Y	22/40
Tabel Ma	Ekst.	1/10	Tabel	Y	Y	Ideal	9/10	9/10	Top-Down; Repr. Tabel	9/10	Pembuat Tabel; Manual	Y	22/40
O-G	Ekst.	6/10	Deskripsi/Gambar	Y	Y	Ideal	10/10	9/10	Top-Down; Repr. Visual	9/10	Dokumen Teks; Gambar; Pembuat; Manual	Y	32/40
Tabel Prolog	Ekst.	2/10	Tabel	Y	Y	Ideal	10/10	7/10	Top-Down; Repr. Tabel	8/10	Pembuat Tabel; Manual	Y	27/40
Simetris	Ekst.	7/10	Gambar Grafik	Y	Y	Ideal	10/10	8/10	Repr. Grafik	10/10	Grafik; Gambar; Pembuat	Y	35/40

Simbol Teks	Ekst.	1/10	Gambar Teks	Y	Y	Ideal	10/10	7/10	Top-Down; Repr. Visual	9/10	Dokumen Teks; Manual	Y	27/40
Z	Ekst.	2/10	Grafik/Grid	Y	Y	Ideal	7/10	3/10	Repr. Grafik	7/10	Grafik; Gambar; Pembuat; Manual	Y	29/40

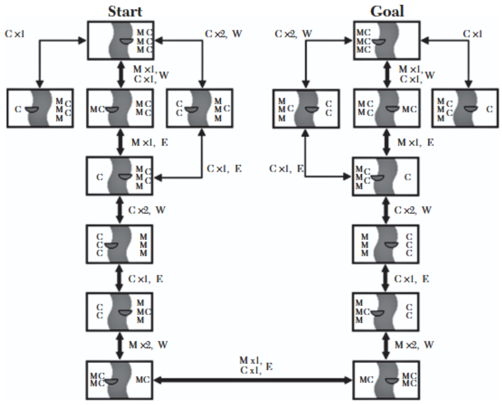
Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tabel 3.1 Pemeringkatan Solusi Masalah Misionaris dan Kanibal Berdasarkan Jendela Manusia.

Solusi yang Paling Kompatibel dengan Jendela Manusia

Solusi yang Paling Kompatibel dengan Jendela Manusia (MHWC) adalah "Solusi Simetris" (Gambar 3.11). (Contoh yang baik dari solusi ini dapat ditemukan di.) Solusi ini merupakan penggambaran ruang keadaan dari masalah sebagai gambar grafis. Titik-titik pada grafik ini merupakan representasi keadaan dari masalah tersebut. Representasi ini menunjukkan visual lingkungan, yang menunjukkan sungai, perahu, tepi sungai tempat perahu terdekat, dan huruf-huruf yang mewakili misionaris dan kanibal. Masing-masing komponen ini terpisah dari yang lain, dan lingkungan digambar ulang sepenuhnya di setiap keadaan. Terakhir, sebuah kunci disediakan untuk menunjukkan arti dari inisial. Setiap detail ini membuat representasi keseluruhan sangat mudah dipahami.



Keterangan:

M = Misionaris

W = Tepi Barat

C = Kanibal

E = Tepi Timur

Contohnya, $M \times 1, W$

Seorang misionaris melakukan perjalanan ke tepi barat

Gambar 3.11 "Solusi Simetris."

Tepi grafik menunjukkan keadaan mana yang berdekatan satu sama lain. Mereka juga diberi bobot yang menunjukkan berapa banyak misionaris dan/atau kanibal yang harus ditempatkan di perahu untuk berpindah di antara setiap negara bagian. Ini dilakukan dengan menggunakan angka dan inisial. Hal ini membuat solusinya sangat mudah dieksekusi.

Solusi ini menunjukkan semua negara bagian yang sah dan bagaimana mereka dicapai, termasuk negara bagian yang tidak mengarah pada solusi. Ini memberikan gambaran seluruh ruang negara bagian, mengungkapkan setiap solusi yang mungkin untuk masalah tersebut, dan memungkinkan seseorang untuk memecahkan masalah dengan cara apa pun yang dianggapnya tepat. Salah satu detail paling menarik tentang solusi ini adalah bahwa ia mengungkapkan metode pemecahan masalah lainnya: penggunaan subtujuan. Ketika Anda memeriksa solusi ini lebih lanjut, Anda akan melihat bahwa bagian pertama dari solusi adalah bayangan cermin yang tepat dari bagian kedua. Ini menunjukkan bahwa ketika bagian pertama dari masalah terpecahkan, bagian kedua dapat dipecahkan dengan mereplikasi langkah-langkah untuk bagian pertama secara terbalik dan menempatkan misionaris dan kanibal di tepi yang terbalik. Solusi ini menggambarkan simetri ini dengan menyajikan grafik sebagai gambar simetris. Simetri ini sekali lagi menunjukkan manfaat penggunaan pencarian dua arah untuk menyelesaikan masalah ini. Itulah sebabnya solusi ini merupakan solusi MHWC, meskipun tidak mendapatkan peringkat keseluruhan tertinggi dalam tabel.

Solusi yang Paling Tidak Kompatibel dengan Jendela Manusia

Solusi yang merupakan Solusi yang Paling Tidak Kompatibel dengan Jendela Manusia (Least Human Window–Compatible/LHWC) dalam penilaian kami adalah "Solusi Aturan Kendala" (Tabel 3.2). Disebut demikian karena menggunakan pemenuhan kendala sebagai metode pemecahan masalahnya. Untuk solusi ini, guna menemukan keadaan tujuan, seseorang harus mengikuti serangkaian kendala, atau "Aturan Produksi" (lihat Bagian 3.6). Aturan-aturan ini sesuai dengan kendala yang harus dipatuhi ketika mencoba memecahkan Masalah Misionaris dan Kanibal.

Ide utama di balik penggunaan metode ini untuk menemukan solusi adalah (1) untuk setiap langkah, lihat keadaan saat ini; (2) lihat aturan yang diberikan untuk melihat mana yang dapat diikuti; (3) terapkan aturan ini; dan (4) tunjukkan keadaan baru serta aturan yang digunakan. Proses ini harus dilanjutkan hingga keadaan tujuan tercapai.

Meskipun aturan-aturan ini secara efisien mencapai keadaan tujuan dengan langkah sesedikit mungkin (pendekatan ideal untuk program komputer), solusi seperti ini mungkin terlalu rumit bagi manusia, terutama untuk masalah yang relatif kecil seperti Masalah Misionaris dan Kanibal. Manusia lebih menyukai representasi visual dan intuitif, terutama ketika terdapat kendala. Dengan menganalisis Tabel 3.2 lebih lanjut, kita dapat melihat bahwa solusinya terbagi menjadi dua bagian: tabel Aturan Produksi ini dan tabel yang menggambarkan contoh solusi ketika mengikuti aturan-aturan ini. Kita akan menganalisis keduanya.

Tabel solusi contoh menunjukkan kemungkinan transisi keadaan jika Aturan Produksi diterapkan pada Masalah Misionaris dan Kanibal. Setiap langkah diwakili oleh baris, dan menunjukkan keadaan saat ini dan aturan yang diterapkan untuk mencapai keadaan tersebut. Saat menganalisis tabel, masalah yang mungkin muncul dengan metode ini adalah menelusuri solusi secara umum. Untuk sepenuhnya memahami bagaimana solusi dilakukan, seseorang harus menghafal setiap aturan, yang terlalu sulit, atau terus-menerus merujuk kembali ke tabel untuk menemukan aturan yang sesuai, yang bisa sangat membosankan.

Selain tabel contoh yang diberikan pada gambar, memahami Aturan Produksi mungkin sulit. (Contoh aturan ini dapat ditemukan di.) Hal ini karena aturan tersebut tampak mirip dengan "pernyataan jika" yang umum digunakan oleh ilmuwan komputer. Rata-rata orang mungkin tidak dapat memahami implikasi sebenarnya di balik serangkaian pernyataan "jika". Namun, implementasi Aturan Produksi merupakan solusi yang lebih baik untuk mesin daripada manusia (lihat Bagian 3.6).

Selain itu, untuk solusi secara umum, para misionaris dan kanibal dapat dilihat diwakili oleh variabel, dan bank serta perahu diwakili oleh fungsi yang menggunakan variabel tersebut sebagai parameter. Sekali lagi, seorang ilmuwan komputer atau matematikawan mungkin dapat memahami hal ini lebih baik daripada rata-rata orang.

Keterangan

$\mathrm{x} = \mathrm{Misionaris}\ (\mathrm{x} = 0, 1, 2, \mathrm{atau}\ 3)$

$\mathrm{y} = \mathrm{Kanibal}\ (\mathrm{y} = 0, 1, 2, \mathrm{atau}\ 3)$

$\mathrm{W}(x, y)$ = Jumlah misionaris dan/atau kanibal di tepi barat

$\mathrm{E}(x, y)$ = Jumlah misionaris dan/atau kanibal di tepi timur

$\mathrm{B}(x, y)$ = Jumlah misionaris dan/atau kanibal di atas perahu dan menyeberangi sungai untuk mencapai tepi seberang

Aturan Kendala

NO. ATURAN	BATASAN
1	$\mathrm{B}(2,0)$ jika dan hanya jika: ($x - 2 \geq y \,\|\, x - 2 = 0$ di salah satu tepi) && ($x + 2 \geq y$ di tepi yang lain)
2	$\mathrm{B}(1,0)$ jika dan hanya jika: ($x - 1 \geq y \,\|\, x = 0$ di salah satu tepi) && ($x + 1 \geq y$ di tepi yang lain)
3	$\mathrm{B}(1,1)$ jika dan hanya jika: ($x - 1 \geq y - 1 \,\|\, x = 0$ di salah satu tepi) && ($x + 1 \geq y + 1 \,\|\, x = 0$ di tepi yang lain)
4	$\mathrm{B}(0,1)$ jika dan hanya jika: ($y - 1 < x \,\|\, x = 0$ di salah satu tepi) && ($y + 1 \leq x \,\|\, y = 0$ di tepi yang lain)
5	$\mathrm{B}(0,2)$ jika dan hanya jika: ($y - 2 \leq x \,\|\, x = 0$ di salah satu tepi) dan ($y + 2 \leq x \,\|\, x = 0$ di tepi yang lain)

Keadaan Awal: W(0, 0) E(3, 3)

Keadaan Tujuan: W(3, 3) E(0, 0)

Contoh

LANGKAH	KEADAAN	ATURAN YANG DITERAPKAN
0	W(0, 0); E(3, 3)	(Awal)
1	W(1, 1); E(2, 2); B(1, 1)	3
2	W(0, 1); E(3, 2); B(1, 0)	2
....		
10	W(2, 2); E(1, 1); B(1, 0)	2
11	W(3, 3); E(0, 0); B(1, 1)	3

Tabel 3.2 "Solusi Aturan Kendala."

3.6 SOLUSI MESIN TERBAIK

Bagi komputer, Masalah Misionaris dan Kanibal umumnya mudah dipecahkan. Solusi terbaik untuk masalah ini adalah solusi yang menemukan solusi dengan cepat dan membutuhkan ruang yang sangat sedikit, biasanya merepresentasikan ruang keadaan masalah dengan pohon pencarian, di mana keadaan awal adalah akar dari pohon tersebut. Bersamaan dengan pencarian dua arah, algoritma pencarian seperti pencarian **breadth-first**, pencarian **depth-first**, dan **iterative-deepening** telah terbukti sangat efektif dalam menyelesaikan masalah ini. Hal ini karena ruang keadaan untuk masalah ini sangat kecil, terutama ketika kendala terpenuhi, sehingga mengurangi kedalaman dan faktor percabangan pohon.

Selain itu, serangkaian Aturan Produksi digunakan untuk membantu memandu pencarian ini ke keadaan tujuan. Aturan Produksi adalah kendala yang harus diikuti untuk menyelesaikan masalah dengan benar (yaitu, tanpa melanggar aturan). Aturan-aturan ini pada dasarnya adalah pernyataan "jika-maka". Menurut Robertson, "Beberapa peneliti mengklaim mampu memprediksi sejauh mana aturan yang berlaku di satu area dapat mentransfer [pengetahuan spesifik domain] ke area lain". Contoh aturan ini dapat dilihat pada Tabel 3.2.

Dengan aturan yang diterapkan dan batasan yang diterapkan, skenario terburuk untuk algoritma pencarian apa pun adalah pohon dengan hanya 16 simpul (status legal). Oleh karena itu, kompleksitas keseluruhan dari Masalah Misionaris dan Kanibal hanya berada pada orde **O(n)**.

3.7 MASALAH TERKAIT

Karena Missionaries and Cannibals termasuk dalam kategori masalah penyeberangan sungai, masalah lain dalam kategori yang sama sangat erat kaitannya. Salah satu masalah ini dikenal sebagai "Masalah Kereta Tengah Malam", yang juga dikenal sebagai "Masalah Penyeberangan Berbahaya". Masalah ini termasuk dalam kategori teka-teki penyeberangan sungai. Kendala dasarnya sama, yaitu jembatan hanya dapat menampung dua orang pada satu waktu, dan satu orang harus membawa sesuatu kembali setelah menyeberangi jembatan. Namun, kendala lainnya adalah mereka semua harus menyeberangi jembatan dalam jangka waktu yang ditentukan. Masalah ini melibatkan empat orang yang mencoba menyeberangi jembatan larut malam hanya dengan satu senter. Senter harus digunakan untuk menyeberangi jembatan dan harus dibawa kembali dari sisi yang berlawanan. Setiap orang memiliki kecepatan berjalan yang berbeda, dan orang-orang yang menyeberangi jembatan harus berjalan dengan kecepatan orang yang paling lambat. Dan mereka semua harus menyeberang dalam waktu kurang dari waktu tertentu.

Masalah terkait lainnya adalah "Masalah Suami Cemburu" yang disebutkan secara singkat di awal bab ini. Premisnya hampir sama dengan Masalah Misionaris dan Kanibal, di mana tiga pasang suami istri mencoba menyeberangi sungai dengan perahu yang hanya mampu mengangkut dua orang setiap saat. Kendalanya adalah seorang istri tidak boleh berada di hadapan suami lain tanpa suaminya sendiri juga di sisinya. Jika tidak, hal itu akan membuat suami istri tersebut cemburu. Kendala inilah yang menjadi nama masalahnya. Solusinya sama dengan Masalah Misionaris dan Kanibal.

3.8 PROGRAM YANG DAPAT DIMAINKAN

Beberapa situs web memiliki permainan yang dikembangkan dalam Flash untuk Masalah Misionaris dan Kanibal di mana seseorang dapat mencoba dan menggunakan berbagai teknik pemecahan masalah. Permainan ini merupakan representasi yang sangat ekstensional; pemecah masalah sebenarnya dapat memanipulasi objek dan menggunakan teknik coba-coba untuk menemukan solusinya. Lebih lanjut, seseorang harus menggambar representasi beberapa kali sebelum sampai pada solusi jika mencoba memecahkan masalah ini secara manual. Dengan demikian, permainan semacam itu membuat siswa tertarik pada masalah seperti ini.

Oleh karena itu, pilihan representasi, metode coba-coba, perencanaan, penyelesaian subtujuan, dan simetri adalah beberapa teknik yang membantu dalam memecahkan masalah seperti Masalah Misionaris dan Kanibal.

BAB 4
PERMASALAHAN 12 KOIN

4.1 PENDAHULUAN

Permasalahan 12 Koin termasuk dalam kelas permasalahan koin palsu yang melibatkan pencarian koin palsu di antara sekelompok koin. D. Schell di *American Mathematical Monthly* edisi Januari 1945, yang mengharuskan pencarian koin dengan berat kurang di antara sekelompok delapan koin menggunakan neraca balok hanya dua kali. Ada banyak variasi permasalahan koin palsu, beberapa di antaranya menyediakan koin standar tambahan untuk perbandingan dan yang lainnya mengungkapkan apakah koin ganjil tersebut lebih berat atau lebih ringan daripada koin lainnya.

Untuk Permasalahan 12 Koin dalam bab ini, mengingat 1 dari 12 koin memiliki berat yang berbeda, tantangannya adalah menemukan koin ganjil tersebut hanya dalam tiga kali penimbangan dan juga menentukan apakah koin tersebut lebih berat atau lebih ringan daripada koin lainnya.

Data yang diberikan untuk permasalahan ini adalah:

Hanya ada satu koin ganjil.

Koin ganjil memiliki berat yang berbeda dari koin lainnya.

Yang tidak diketahui adalah:

Koin manakah yang ganjil?

Apakah koin ganjil lebih berat atau lebih ringan daripada koin lainnya?

Kendalanya adalah:

Hanya tiga kali penimbangan yang diperbolehkan.

4.2 MENYELESAIKAN MASALAH YANG LEBIH KECIL

Kita dapat mengubah masalah ini menjadi lebih sederhana dengan mengurangi jumlah koin dan jumlah penimbangan. Mengikuti nasihat Polya, "Jika Anda tidak dapat menyelesaikan suatu masalah, maka jika ada masalah yang lebih mudah yang dapat Anda selesaikan, temukanlah," kita mencari masalah dengan koin yang lebih sedikit.

Mari kita mulai menggunakan *teknik reduksi masalah*. Jika kita mulai hanya dengan 2 koin, mustahil untuk menemukan koin mana yang ganjil kecuali kita tahu apakah koin ganjil tersebut lebih berat atau lebih ringan daripada yang lain atau jika kita memiliki koin standar untuk dibandingkan. Dengan demikian, masalah 2 koin bukanlah masalah minimal (tanpa koin standar tambahan atau pengetahuan tentang berat relatif koin ganjil terhadap koin yang baik). Oleh karena itu, kita dapat memeriksa apakah soal 3 koin merupakan versi yang lebih mudah dari Soal 12 Koin. Jika kita mengurangi jumlah koin menjadi 3, kita mungkin mendapatkan ide yang dapat diterapkan untuk menyelesaikan Soal 12 Koin.

Dalam soal 3 koin, jika kita mengetahui bahwa koin ganjil tersebut beratnya kurang atau lebih, hanya diperlukan satu kali penimbangan untuk mengidentifikasi koin ganjil tersebut. Jika tidak, pertama-tama kita harus membandingkan koin pertama dan kedua. Jika keduanya sama,

kita tahu bahwa koin ketiga ganjil. Penimbangan koin ketiga sekali lagi dengan salah satu dari dua koin pertama akan memberi tahu kita apakah koin ganjil tersebut lebih berat atau lebih ringan daripada dua koin lainnya. Namun, jika timbangannya tidak sama pada penimbangan pertama, artinya salah satu dari dua koin pertama ganjil dan koin ketiga tidak ganjil. Bagaimanapun, setelah hasil penimbangan pertama, soal 3 koin menjadi serupa dengan "soal 2 koin dengan koin standar", sehingga koin ganjil dan berat relatifnya dapat ditemukan dalam total dua kali penimbangan. Solusi untuk masalah 3 koin dengan koin bernama A, B, dan C ditunjukkan pada Tabel 4.1.
Dengan demikian, kita dapat menyimpulkan hal berikut: *Koin dengan berat yang sama adalah koin yang baik dan dapat dipisahkan dari koin yang belum ditimbang, dan kelompok koin yang tersisa ini berisi koin ganjil.*

Tabel 4.1 Solusi Untuk Masalah 3 Koin

PENIMBANGAN 1 A ? B	PENIMBANGAN 2 A ? C	HASIL
A = B (Jadi, C ganjil)	A = C	Tidak mungkin karena salah satu dari mereka ganjil
	A > C	C ganjil dan ringan
	A < C	C ganjil dan berat
A > B (Jadi, A atau B ganjil)	A = C	B ganjil dan ringan
	A > C	A ganjil dan berat
	A < C	Tidak mungkin karena HANYA salah satu dari mereka yang ganjil
A < B (Jadi, A atau B ganjil)	A = C	B ganjil dan berat
	A > C	Tidak mungkin karena HANYA salah satu dari mereka yang ganjil
	A < C	A ganjil dan ringan

Ketika timbangan tidak sama, salah satu sisi timbangan berisi koin ganjil.

Setelah koin ganjil terdeteksi, kita perlu mengetahui beratnya relatif terhadap koin yang baik untuk menentukan apakah koin tersebut lebih berat atau lebih ringan daripada koin lainnya.

Beralih ke masalah 4 koin yang sedikit lebih rumit, jika kita membagi koin menjadi dua kelompok yang masing-masing terdiri dari 2 koin dan membandingkan berat kelompok-kelompok ini, kita akan tahu mana dari kedua kelompok yang lebih berat (atau lebih ringan). Namun, kita tidak tahu kelompok mana yang berisi koin ganjil, kita juga tidak tahu apakah koin ganjil itu berat atau ringan. Dengan demikian, sangat sedikit informasi yang diperoleh dari penimbangan pertama. Lebih lanjut, dua penimbangan lagi diperlukan untuk membandingkan koin-koin di setiap kelompok secara individual untuk mengidentifikasi koin ganjil.

Alternatifnya, kita dapat mengambil 3 koin pertama dan menyelesaikannya dalam dua penimbangan seperti yang kita lakukan untuk masalah 3 koin. Jika kedua kelompok memiliki berat yang sama, kita tahu koin keempat ganjil, dan penimbangan satu kali lagi dengan salah satu koin baik yang diketahui dapat memberi tahu kita apakah koin keempat kelebihan berat atau kekurangan berat. Dengan demikian, masalah 4 koin dapat diselesaikan dalam tiga kali penimbangan.

Masalah ini membantu kita menyimpulkan bahwa *membagi koin menjadi kelompok-kelompok yang terdiri dari tiga koin memberikan lebih banyak informasi pada penimbangan pertama, dibandingkan dengan membagi menjadi kelompok-kelompok yang terdiri dari dua koin*—informasi tentang kelompok yang hanya berisi koin-koin yang baik. Hal ini karena setelah penimbangan pertama, kita mengetahui bahwa baik dua kelompok pertama maupun kelompok ketiga semuanya berisi koin-koin yang baik, sehingga memisahkan koin-koin yang baik dari kelompok-kelompok yang berisi koin ganjil, dan dengan demikian mengurangi masalah menjadi masalah dengan koin yang lebih sedikit.

Jika kita mencoba menyelesaikan masalah 6 koin dengan pengetahuan yang diperoleh dari dua masalah sebelumnya, kita dapat melanjutkan sebagai berikut: Membagi koin-koin menjadi tiga kelompok yang masing-masing berisi 2 koin. Setelah membandingkan dua kelompok pertama, kita mengetahui bahwa koin ganjil berada di salah satu dari dua kelompok pertama jika beratnya tidak sama atau bahwa koin ganjil berada di kelompok ketiga jika dua kelompok pertama memiliki berat yang sama pada penimbangan pertama.

Dalam kedua kasus tersebut, penimbangan satu kali lagi pada kelompok ganjil (atau kemungkinan ganjil) dengan salah satu kelompok baik akan memberi tahu apakah koin ganjil tersebut berat atau ringan, dan penimbangan ketiga akan membantu mengidentifikasi koin ganjil dalam kelompok ganjil (Tabel 4.1).

Tabel 4.1 Solusi Untuk Masalah 6 Koin

PENIMBANGAN 1 AB ? CD	PENIMBANGAN 2 AB ? EF	PENIMBANGAN 3	HASIL
AB = CD (E atau F ganjil)	AB = EF	Tidak mungkin karena HANYA salah satunya ganjil	
	AB > EF	E = F	Tidak mungkin karena HANYA salah satunya ganjil
		E > F	F ganjil dan ringan
		E < F	E ganjil dan ringan
	AB < EF	E = F	Tidak mungkin karena HANYA salah satunya ganjil
		E > F	E ganjil dan berat
		E < F	F ganjil dan berat
AB > CD	AB = EF	C = D	Tidak mungkin karena HANYA salah satunya ganjil
		C > D	D ganjil dan ringan
		C < D	C ganjil dan ringan
	AB > EF	A = B	Tidak mungkin karena HANYA salah satunya ganjil
		A > B	A ganjil dan berat
		A < B	B ganjil dan berat
	AB < EF	Tidak mungkin karena HANYA salah satunya ganjil	
AB < CD	AB = EF	C = D	Tidak mungkin karena HANYA salah satunya ganjil

		C > D	C ganjil dan berat
		C < D	D ganjil dan berat
	AB > EF	Tidak mungkin karena HANYA salah satunya ganjil	
	AB < EF	A = B	Tidak mungkin karena HANYA salah satunya ganjil
		A > B	B ganjil dan ringan
		A < B	A ganjil dan ringan

Dari masalah 6 koin, kita dapat menyimpulkan hal berikut:

Ketika kita membagi koin menjadi kelompok-kelompok yang masing-masing terdiri dari 3 koin, kita dapat mengetahui dari penimbangan kedua apakah koin ganjil tersebut berat atau ringan dan kelompok mana yang berisi koin ganjil tersebut.

Hal ini karena pada akhir dua penimbangan, kita mengetahui berat relatif masing-masing kelompok terhadap dua kelompok lainnya, yang memisahkan kelompok ganjil dan memberi tahu kita apakah koin ganjil tersebut berat atau ringan.

Masalah 9 koin merupakan kasus khusus, karena kita membagi koin-koin menjadi tiga kelompok yang masing-masing terdiri dari 3 koin. Dengan demikian, setelah membandingkan salah satu dari tiga kelompok dengan dua kelompok lainnya dalam dua penimbangan, kita mengetahui kelompok ganjil dan berat relatif koin ganjil tersebut. Oleh karena itu, masalah ini disederhanakan menjadi masalah 3 koin ketika berat relatif koin ganjil diketahui. Dengan demikian, satu penimbangan lagi menghasilkan koin ganjil.

Sebagai alternatif, kita dapat menganggap 3 koin di masing-masing dari tiga kelompok sebagai satu koin besar. Masalahnya kemudian dapat disederhanakan menjadi masalah 3 koin, dan setelah dua kali penimbangan, kita tahu apakah ketiga kelompok atau koin besar itu ganjil dan apakah berat atau ringan. Dengan demikian, sekarang kita tahu kelompok 3 koin yang berisi koin ganjil yang sebenarnya. Masalah ini disederhanakan menjadi masalah 3 koin di mana kita tahu bahwa koin ganjil itu berat atau ringan, sehingga satu kali penimbangan sudah cukup untuk menemukan koin ganjil. Teknik ini dikenal sebagai *rekursi*, di mana suatu masalah terus dipecah menjadi submasalah yang, ketika mencapai kasus dasar atau elementer, dapat diselesaikan secara langsung.

Kita dapat menyimpulkan bahwa seiring bertambahnya jumlah koin, kita perlu membandingkan koin sebanyak mungkin pada dua penimbangan pertama untuk menyederhanakan masalah menjadi masalah yang sekecil mungkin. Kita baru saja melihat bagaimana masalah 9 koin disederhanakan menjadi masalah 3 koin setelah dua kali penimbangan.

4.3 SOLUSI MASALAH 12 KOIN

Dalam sebuah observasi yang dilakukan di kelas AI, salah satu siswa menemukan solusi untuk Masalah 12 Koin yang ditunjukkan pada Tabel 4.2.

Tabel 4.2 Solusi Untuk Masalah 12 Koin

PENIMBANGAN 1 ABCD ? EFCH	PENIMBANGAN 2	PENIMBANGAN 3	HASIL
ABCD = EFCH	ABC = IJK	A = L	Tidak mungkin karena salah satunya ganjil
		A > L	L ganjil dan ringan
		A < L	L ganjil dan berat
	ABC > IJK	I = J	K ganjil dan ringan
		I > J	J ganjil dan ringan
		I < J	I ganjil dan ringan
	ABC < IJK	I = J	K ganjil dan berat
		I > J	I ganjil dan berat
		I < J	J ganjil dan berat
ABCD > EFCH	**AB**E = **CD**F	G = H	Tidak mungkin karena salah satunya ganjil
		G > H	H ganjil dan ringan
		G < H	G ganjil dan ringan
	ABE > **CD**F	A = B	F ganjil dan ringan
		A > B	A ganjil dan berat
		A < B	B ganjil dan berat
	ABE < **CD**F	C = D	E ganjil dan ringan
		C > D	C ganjil dan berat
		C < D	D ganjil dan berat
ABCD < EFCH	AB**E** = CD**F**	G = H	Tidak mungkin karena salah satunya ganjil
		G > H	G ganjil dan berat
		G < H	H ganjil dan berat
	AB**E** > CD**F**	C = D	E ganjil dan berat
		C > D	D ganjil dan ringan
		C < D	C ganjil dan ringan
	AB**E** < CD**F**	A = B	F ganjil dan berat
		A > B	B ganjil dan ringan
		A < B	A ganjil dan ringan

Dalam solusi ini, 12 koin dibagi menjadi tiga kelompok, masing-masing berisi 4 koin. Pada penimbangan pertama, kita mengetahui bahwa kelompok koin ketiga, yang belum ditimbang, berisi koin ganjil, atau salah satu dari dua kelompok pertama berisi koin ganjil. Dalam kasus pertama, masalah disederhanakan menjadi masalah 4 koin. Dengan menghilangkan satu koin dari kelompok ini, masalah dapat diselesaikan sebagai masalah 3 koin dalam dua penimbangan berikutnya. Jika koin keempat yang dihilangkan ternyata ganjil (pada penimbangan kedua), koin tersebut dapat ditimbang dengan koin yang diketahui baik untuk memeriksa apakah berat atau ringan pada penimbangan keempat.

Pada kasus kedua, kita mengetahui berat relatif antara dua kelompok pertama dan dapat menggunakan pengetahuan ini untuk penimbangan berikutnya. Kemudian, kita membagi kelompok yang lebih berat menjadi masing-masing 2 koin dan meletakkannya di kedua sisi timbangan. Kemudian, kita mengambil 2 koin dari kelompok yang lebih ringan dan menempatkan satu koin di kedua sisi timbangan, lalu melakukan penimbangan kedua. Ini akan mengurangi jumlah koin ganjil yang dicurigai menjadi 3 koin yang mungkin jika timbangan tidak seimbang atau 2 koin yang mungkin (2 koin sisanya dari kelompok yang lebih ringan) jika timbangan seimbang. Bagaimanapun, satu kali penimbangan lagi sudah cukup untuk mendeteksi koin ganjil. Solusi ini, meskipun benar, tidak dapat diperluas ke sejumlah koin.

Kita dapat menggeneralisasi Masalah 12 Koin menjadi masalah n koin. Ditemukan bahwa jumlah penimbangan yang diperlukan untuk menemukan koin ganjil di antara n koin dapat disimpulkan dari rumus berikut yang dikembangkan oleh Bennet Manvel dalam artikelnya "Counterfeit Coin Problems" di $Mathematics\ Magazine$:

Satu koin yang beratnya lebih ringan dalam satu set yang berisi k koin dapat ditemukan dalam ?$\log_3 k$? kali penimbangan.

Jika kita hanya tahu bahwa koin palsu memiliki berat yang berbeda, maka penimbangan $[\log_3(2k+3)]$ diperlukan.

Diperlukan $[\log_3(2k+1)]$ penimbangan jika koin standar disediakan.

*Harap dicatat bahwa $[x]$ berarti "batas atas" x, atau bilangan bulat terkecil yang lebih besar atau sama dengan x, di mana x adalah bilangan riil.

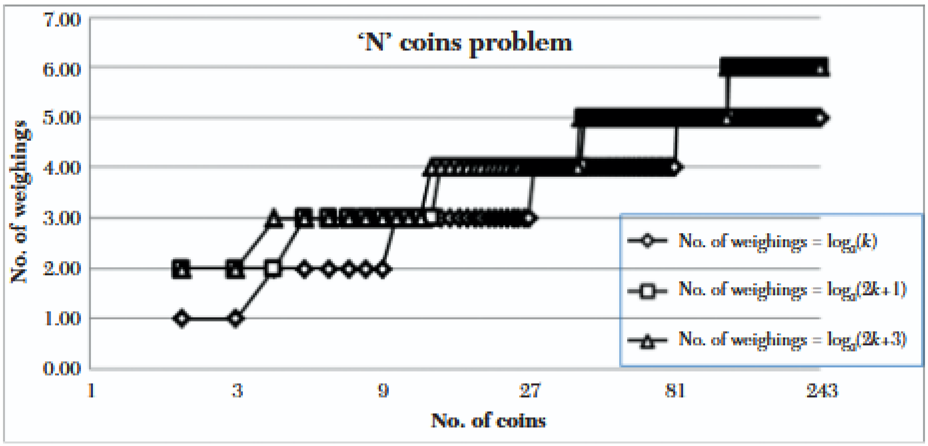
Pada Tabel 4.3, kami mentabulasi jumlah penimbangan untuk beberapa soal n koin menggunakan rumus untuk tiga kasus yang disebutkan di atas.

NO. JUMLAH KOIN = k	**KASUS 1:** MENCARI KOIN **LEBIH RINGAN** DALAM SATU SET k KOIN **Jumlah penimbangan = $[\log_3(k)]$**	**KASUS 2:** MENCARI KOIN BERBEDA BERAT **(GANJIL)** JIKA KOIN STANDAR TERSEDIA **Jumlah penimbangan = $[\log_3(2k+1)]$**	KASUS 3: MENCARI KOIN BERBEDA BERAT **(GANJIL)** JUMLAH PENIMBANGAN = $[\log_3(2k+3)]$
3	1	2	2
4	2	2	3
6	2	3	3
9	2	3	3
11	3	3	3
12	3	3	3
13	3	3	4
39	4	4	4
40	4	4	5
120	5	5	5

Tabel 4.3 Penimbangan Koin Ganjil dari Koin (Berbagai Kasus)

Grafik pada Gambar 4.1 mengilustrasikan penerapan rumus pada Tabel 4.3.

Yang perlu diperhatikan adalah bagaimana laju pertumbuhan log3 tampaknya mendefinisikan fungsi bertahap.



Gambar 4.1 Gambaran Grafis Beberapa Masalah Koin

Solusi lain yang diciptakan Jack Wert tersedia di $http://www.cut-the-knot.org/blue/OddCoinProblems.shtml$ dan $http://www.cut-the-knot.org/blue/$

$OddCoinProblemsShort.shtml$. Ini adalah solusi untuk Masalah 12 Koin yang dapat diperluas ke sejumlah koin $(3^n - 3)/2$.

Bagilah koin-koin tersebut menjadi tiga kelompok yang sama besar, masing-masing berisi empat koin. Mari kita beri nama koin-koin tersebut A, B, C, D, E, F, G, H, I, J, K, dan L.

Bagilah masing-masing dari ketiga kelompok tersebut menjadi subkelompok utama yang terdiri dari 3 koin dan kemudian 1 koin.

Letakkan dua kelompok koin pertama yang masing-masing terdiri dari 4 koin di kedua sisi timbangan dan catat hasilnya. Ini adalah penimbangan pertama.

Kemudian, putar subgrup utama—dengan kata lain, pindahkan subgrup utama dari kelompok pertama dari sisi kiri timbangan, pindahkan subgrup utama dari kelompok kedua dari sisi kanan ke kiri, dan pertahankan subgrup utama dari kelompok ketiga di sisi kanan timbangan. Pertahankan koin-koin tunggal dari dua kelompok pertama sebagaimana adanya di timbangan. Ini adalah penimbangan kedua.

Penimbangannya dijelaskan pada Tabel 4.4.

Tabel 4.4 Konfigurasi Koin dalam Dua Penimbangan

	LEMPENG KIRI	**LEMPENG KANAN**
Penimbangan 1 (W1)	A, B, C, D	E, F, G, D
Penimbangan 2 (W2)	E, F, G, H	I, J, K, H

Masing-masing penimbangan memiliki tiga kemungkinan hasil:

= berarti kedua timbangan sama.

< berarti wadah kanan lebih berat daripada wadah kiri.

> berarti wadah kiri lebih berat daripada wadah kanan.

Tabel 4.5 menunjukkan sembilan kemungkinan hasil dari kedua penimbangan.

Tabel 4.5 Kemungkinan Hasil Setelah Dua Penimbangan

KASUS	HASIL W1	HASIL W2	KOIN ANEH YANG MUNGKIN	APAKAH KOIN ANEH LEBIH BERAT ATAU LEBIH RINGAN?	LANGKAH BERIKUTNYA	BERAPA KALI PENIMBANGAN LAGI YANG DIBUTUHKAN?	JUMLAH TOTAL PENIMBANGAN
1	=	=	I	Belum bisa dipastikan.	Bandingkan koin I dengan salah satu koin baik untuk mengetahui berat relatifnya.	1	3
2	<	<	D atau H	Belum bisa dipastikan.	Bandingkan koin D atau H dengan salah satu koin baik untuk mengetahui berat relatif koin aneh.	1	3
3	>	>	D atau H	Belum bisa dipastikan.		1	3
4	=	<	I, J, atau K	Lebih berat	Masalah ini direduksi menjadi masalah 3 koin, di mana berat relatif koin aneh sudah diketahui.	1	3
5	=	>	I, J, atau K	Lebih ringan		1	3
6	<	=	A, B, atau C	Lebih ringan		1	3
7	>	=	A, B, atau C	Lebih berat		1	3
8	<	>	D, E, atau F	Lebih berat		1	3
9	>	<	D, E, atau F	Lebih ringan		1	3

Aspek kunci dari solusi ini adalah upaya *memaksimalkan jumlah informasi yang diperoleh dalam jumlah penimbangan paling sedikit*. Dalam dua penimbangan pertama, 11 dari 12 koin ditimbang. Penimbangan kemudian dikonfigurasi sedemikian rupa sehingga setelah dua penimbangan, permasalahan direduksi menjadi permasalahan yang lebih kecil yang dapat diselesaikan hanya dalam satu penimbangan lagi, bergantung pada sembilan kemungkinan hasil. Selain itu, kita dapat memanfaatkan informasi yang diperoleh dari dua penimbangan sebelumnya pada penimbangan terakhir.

Menurut solusi ini, untuk menyelesaikan permasalahan koin $(3^n - 3)/2$ secara umum, langkah-langkah berikut harus diikuti:

Bagilah koin-koin tersebut menjadi tiga kelompok utama yang sama. Masing-masing dibagi menjadi (n – 1) subkelompok yang terdiri dari $3^{\mathrm{n}-2}, 3^{\mathrm{n}-3}, \ldots, 3^0 \mathrm{koin}$.

Letakkan dua kelompok utama pertama di setiap sisi timbangan. Ini adalah penimbangan pertama.

Putar hanya subgrup terbesar, seperti yang dijelaskan sebelumnya dalam solusi Masalah 12 Koin. Ini adalah penimbangan kedua.

Jika kondisi timbangan berubah, seperti yang terlihat pada enam kasus terakhir dari solusi Masalah 12 Koin di atas, subgrup yang berisi koin ganjil akan teridentifikasi, dan permasalahan akan direduksi menjadi permasalahan yang lebih kecil.

Lanjutkan rotasi subgrup terbesar berikutnya hingga terjadi perubahan kondisi timbangan atau hingga hanya ada satu koin di setiap sisi timbangan. Kemudian putar subgrup tersebut dan koin ganjil, dan berat relatifnya akan ditemukan.

4.4 PEMECAHAN MASALAH MANUSIA

Masalah ini dipresentasikan kepada sekitar 20 mahasiswa pascasarjana yang mempelajari Kecerdasan Buatan. Masalah ini dipresentasikan dalam dua tahap: sebagai masalah 6 koin dan kemudian sebagai Masalah 12 Koin. Pilihan representasi yang digunakan sebagian besar mahasiswa adalah pohon pencarian, dengan cabang-cabang pohon yang berbeda menandakan hasil penimbangan yang berbeda.

Sebagian besar mahasiswa mampu menyelesaikan masalah 6 koin dalam tiga kali penimbangan, dengan salah satu mahasiswa menyimpulkan bahwa jika berat relatif koin ganjil diketahui, hanya dua kali penimbangan yang cukup.

Kelas tersebut kesulitan dengan Masalah 12 Koin karena lebih sulit daripada masalah 6 koin. Sebagian besar mahasiswa mampu menyelesaikan masalah dalam empat kali penimbangan, dan satu mahasiswa menyelesaikannya dalam minimal tiga kali penimbangan. (Solusi mahasiswa ini ditunjukkan pada Tabel 4.5.) Informasi yang diperoleh dari berat relatif dua kelompok koin yang dibandingkan pada penimbangan pertama digunakan untuk mempersempit jumlah koin ganjil yang mungkin.

Dari analisis Masalah 12 Koin, dengan mereduksinya menjadi masalah-masalah yang lebih kecil, kita belajar lebih banyak tentang masalah tersebut dan mendapatkan petunjuk tentang cara menyelesaikan masalah aslinya. Selain itu, menguraikan masalah yang lebih besar menjadi bagian-bagian kecil dan menyelesaikan masalah-masalah yang lebih kecil secara rekursif membantu menyelesaikan masalah yang lebih besar. Kita telah melihat bahwa teknik reduksi masalah dan rekursi berguna dalam menyelesaikan Masalah 12 Koin.

4.5 ANALISIS JENDELA MANUSIA TERHADAP SOLUSI

Saat menganalisis solusi (Tabel 4.6), kita dapat melihat beberapa kesamaan dalam pemeringkatannya. Pertama, perhatikan bahwa masing-masing solusi bersifat ekstensional. Solusi yang kita temukan spesifik untuk Masalah 12 Koin. Karena solusinya relatif kecil (tiga pembobotan), metode apa pun dapat menggambarkan masalah secara keseluruhan. Namun, jika tabel tersebut berisi solusi untuk *masalah umum n koin*, kemungkinan besar solusi tersebut akan diklasifikasikan sebagai intensional.

Secara keseluruhan, pemeringkatan ekstensionalitas solusi tersebut beragam. Solusi yang menggunakan gambar nyata yang menggambarkan koin dan timbangan umumnya berperingkat lebih tinggi, sementara solusi yang direpresentasikan oleh teks berperingkat lebih rendah. Namun, tidak banyak solusi yang ditemukan menggunakan gambar sebagai sarana representasi.

Pilihan representasi pengetahuan yang digunakan untuk solusi merupakan campuran antara tabel dan teks. Solusi teks memberikan gambaran umum tentang setiap kemungkinan hasil untuk ketiga penimbangan. Tabel menyederhanakan penjelasan ini dengan penggunaan dan pengaturan simbol. Secara umum, setiap solusi menggunakan metode tiga penimbangan secara serupa dan menekankan submasalah yang dihasilkannya.

Tabel 4.6 Pemeringkatan 12 Solusi Masalah Koin Berdasarkan Jendela Manusia

KESULITAN
3/10

KERUMITAN
$O(n(\log_3 n))$

NAMA	INT ATAU EXT?	INTENS./EKSTENS.	REPR.	JENDELA MANUSIA?	BENAR?	UKURAN BUTIR	EKSEKUSI	KETERPAHAMAN	METODE PEMECAHAN MASALAH	FLEKSIBILITAS	MODE KONVERSI	OPTIMAL?	TOTAL
B & Y	Ekst.	2/10	Deskripsi Teks	Y	Y	Ideal	8/10	8/10	Rep. Vis.; Subtujuan	8/10	Txt Editor	Y	26/40
Duncanson	Ekst.	4/10	Deskripsi Teks	Y	Y	Ideal	7/10	6/10	Rep. Vis.; Subtujuan	9/10	Txt Editor; Tangan	Y	26/40
FA	Ekst.	6/10	Tble	Y	Y	Ideal	10/10	7/10	Rep. Vis.; Subtujuan	7/10	Txt Editor; Tangan	Y	30/40
Tabel Generik	Ekst.	7/10	Tble	Y	Y	Ideal	10/10	8/10	Rep. Tbl.; Subtujuan	8/10	Pembuat Tbl; Tangan	Y	33/40
Logically Enumerated	Ekst.	3/10	Deskripsi Teks	Y	Y	Kecil	7/10	5/10	Rep. Vis.; Subtujuan	5/10	Pembuat Gbr Grafik; Tangan	Y	20/40
LogicFlow	Ekst.	5/10	Diag. Logika	Y	Y	Kecil	9/10	6/10	Rep. Vis.; Subtujuan	6/10	Pembuat Gbr	Y	27/40
Poskitt	Ekst.	7/10	Teks; Tble	Y	Y	Ideal	8/10	5/10	Rep. Vis.; Subtujuan	7/10	Pembuat Tbl; Txt Editor; Tangan	Y	27/40

Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?

Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Pemeringkatan yang berkaitan dengan kesesuaian solusi dengan Jendela Manusia juga beragam. Secara umum, setiap solusi diberi peringkat tinggi untuk tingkat eksekusi. Secara keseluruhan, langkah-langkah dari ketiga pembobotan mudah diikuti, apa pun representasi yang digunakan. Namun, tingkat pemahaman solusi beragam. Solusi yang menggunakan representasi teks umumnya berperingkat lebih rendah, sementara solusi yang direpresentasikan oleh tabel berperingkat lebih tinggi. Terakhir, perlu dicatat bahwa setiap solusi menggunakan teknik pemecahan masalah umum yang sama, yaitu pencarian subtujuan dengan menggunakan representasi visual.

Solusi yang Paling Kompatibel dengan Jendela Manusia

Solusi yang dinilai sebagai MHWC adalah "Solusi Tabel Generik" (Tabel 4.7). Solusi ini sama dengan yang ada di Bagian 4.3. Solusi ini direpresentasikan sebagai tabel yang berisi empat kolom. Tiga kolom pertama digunakan untuk menunjukkan kemungkinan hasil yang dapat dihasilkan timbangan selama salah satu dari tiga penimbangan, dengan setiap kolom mewakili penimbangan yang berbeda. Hasil ini berisi huruf yang mewakili koin dan operator kondisional yang mewakili hasil penimbangan. Koin (huruf) yang dikelompokkan bersama adalah koin yang ada di sisi timbangan tertentu. Kolom terakhir menunjukkan hasil penimbangan ketiga, sehingga menyimpulkan apakah koin tersebut berat atau ringan, atau jika penimbangan tidak memungkinkan.

Solusi ini mudah dipahami karena ringkas dan kolom-kolom dalam tabel menyusun solusi dengan baik. Solusi ini juga sangat mudah dieksekusi. Ketika hasil penimbangan pertama ditemukan, kemungkinan hasil yang tersisa untuk penimbangan kedua berada tepat di sebelah hasil tersebut di kolom berikutnya. Kedekatan ini juga berlaku untuk penimbangan kedua dan ketiga. Hal ini memungkinkan solusi ditemukan dengan mudah dan cepat, dan mengurangi informasi yang diperlukan dalam tabel secara keseluruhan.

Meskipun solusinya mudah dipahami, pemahaman secara keseluruhan dapat ditingkatkan jika terdapat gambar kecil timbangan aktual di samping setiap kemungkinan hasil untuk menunjukkan apakah timbangan seimbang atau tidak. Selain itu, pemahaman lebih lanjut dapat dilakukan jika huruf-huruf tersebut dilingkari, yang akan menunjukkan dengan lebih baik bahwa huruf-huruf tersebut sebenarnya mewakili koin.

Solusi yang Paling Tidak Kompatibel dengan Jendela Manusia

Solusi yang dinilai sebagai LHWC adalah "Solusi yang Dihitung Secara Logis" (Tabel 4.8). Solusi ini memberikan deskripsi tekstual tentang apa yang harus dilakukan pada setiap penimbangan, dengan hasil tertentu.

Tabel 4.7 "Solusi Tabel Generik"

PENIMBANGAN 1 ABCD ? EFGH	PENIMBANGAN 2	PENIMBANGAN 3	HASIL
ABCD = EFGH	ABC = IJK	A = L	Tidak mungkin karena salah satunya adalah ganjil
		A > L	L ganjil dan ringan
		A < L	L ganjil dan berat
	ABC > IJK	I = J	K ganjil dan ringan
		I > J	J ganjil dan ringan
		I < J	I ganjil dan ringan
	ABC < IJK	I = J	K ganjil dan berat
		I > J	I ganjil dan berat
		I < J	J ganjil dan berat
ABCD > EFGH	ABE = CDF	G = H	Tidak mungkin karena salah satunya adalah ganjil
		G > H	H ganjil dan ringan
		G < H	G ganjil dan ringan
	ABE > CDF	A = B	F ganjil dan berat
		A > B	A ganjil dan berat
		A < B	B ganjil dan berat
	ABE < CDF	C = D	E ganjil dan ringan
		C > D	C ganjil dan ringan
		C < D	D ganjil dan ringan
ABCD < EFGH	ABE = CDF	G = H	Tidak mungkin karena salah satunya adalah ganjil
		G > H	G ganjil dan berat
		G < H	H ganjil dan berat
	ABE > CDF	C = D	E ganjil dan berat
		C > D	D ganjil dan berat
		C < D	C ganjil dan berat
	ABE < CDF	A = B	F ganjil dan ringan
		A > B	B ganjil dan ringan
		A < B	A ganjil dan ringan

Meskipun solusi ini menunjukkan setiap kemungkinan hasil dan mendemonstrasikan cara menyelesaikan masalah dengan jumlah penimbangan paling sedikit, representasi tabel mungkin lebih baik daripada representasi deskriptif. Hal ini karena solusi khusus untuk Masalah 12 Koin dapat langsung ditemukan di tabel, sementara solusi tersebut mungkin sulit ditemukan di tengah deskripsi yang panjang.

Secara khusus, solusi ini mungkin kurang mudah dipahami, karena deskripsi dikelompokkan dalam paragraf-paragraf kecil. Di dalam paragraf-paragraf tersebut terdapat representasi koin, pengelompokan koin, dan skala, yang mungkin dianggap tidak efisien. Setiap koin diberi nama yang sesuai dengan kelompok tertentu, seperti X1, X2, Z3, dan seterusnya.

Untuk menunjukkan bahwa koin-koin tersebut termasuk dalam kelompok tertentu pada skala, koin-koin tersebut digabungkan menggunakan simbol "–".

Terakhir, kelompok-kelompok tersebut ditunjukkan berada di sisi kiri atau kanan skala dengan pemisahan menggunakan operator kondisional. Meskipun ini berfungsi untuk "Solusi Tabel Generik," ini mungkin tidak berfungsi dalam jenis representasi ini.

- Coin Groups: $\{X1, X2, X3, X4\}$; $\{Y1, Y2, Y3, Y4\}$; $\{Z1, Z2, Z3, Z4\}$
- Weigh $X1 - X2 - X3 - X4$ against $Y1 - Y2 - Y3 - Y4$.

  1st Weighing: If $X1 - X2 - X3 - X4 = Y1 - Y2 - Y3 - Y4$

  The counterfeit must be in group Z. Take any of the X or Y coins as a control coin (i.e. X1).

  Weigh $Z1 + Z2$ against $Z3 + X1$.

  - 2nd Weighing: $Z1 + Z2 = Z3 + X1$.

    Z1 or Z2 is heavy or Z3 is light. Weigh Z1 against Z2.

    (or if $Z1 + Z2 \neq Z3 + X1$, substitute heavy for light and vice versa in the following).

    ♦ 3rd Weighing: $Z1 = Z2$

    <u>Z3 is light.</u>

    ♦ 3rd Weighing: $Z1 > Z2$

    <u>Z1 is heavy.</u>

    ♦ 3rd Weighing: $Z1 < Z2$

    <u>Z2 is heavy.</u>

  - 2nd Weighing: $Z1 + Z2 \neq Z3 + X1$.

    Z4 must be the counterfeit.

    Weigh it against any of the control coins to determine whether if is heavy or light (i.e. X1)

    ♦ 3rd Weighing: $Z4 < X1$

    <u>Z4 is light.</u>

    ♦ 3rd Weighing: $Z4 > X1$

    <u>Z4 is heavy.</u>

- 1st weighing: $X1 - X2 - X3 - X4 < Y1 - Y2 - Y3 - Y4$ (analogous holds if $X1 - X2 - X3 - X4 > Y1 - Y2 - Y3 - Y4$; switch heavy for light and vice versa).

One of the X's is lighter than the rest, or one of the Y's is heavier than the rest.

All the Z's can be taken as control coins.

Weigh $X1 - X2 - Y1 - Y2$ against $X3 - Y3 - Z1 - Z2$.

- **2nd Weighing: $X1 - X2 - Y1 - Y2 > X3 - Y3 - Z1 - Z2$**

  Y1 or Y2 is heavy, or X3 is light.

  Weigh Y1 against Y2:

  ♦ **3rd weighing: $Y1 > Y2$**

  <u>Y1 is heavy.</u>

  ♦ **3rd weighing: Y1 < Y2**

  <u>Y2 is heavy.</u>

  ♦ **3rd weighing: Y1 = Y2**

  <u>X3 is light.</u>

- **2nd Weighing: $\mathbf{X1 - X2 - Y1 - Y2 \; = \; X3 - Y3 - Z1 - Z2}$**

$\mathbf{X4}$ is light or $\mathbf{Y4}$ is heavy. Weigh $\mathbf{X4}$ against $\mathbf{Z1}$.

- ♦ **3rd weighing: $\mathbf{Y1}$ > $\mathbf{Y2}$**

  $\mathbf{Y1}$ is heavy.
- ♦ **3rd weighing: $\mathbf{Y1}$ < $\mathbf{Y2}$**

  $\mathbf{Y2}$ is heavy.

Tabel 4.8 “Solusi yang Dihitung Secara Logika”

4.6 SOLUSI MESIN TERBAIK

Secara umum, tidak ada kategori algoritma atau heuristik khusus yang diperlukan agar mesin dapat menyelesaikan Masalah 12 Koin. Algoritma apa pun yang mengimplementasikan metode umum penyelesaian Masalah 12 Koin dalam tiga penimbangan dijamin akan menemukan solusi optimal. Jika diimplementasikan dengan benar, teknik yang sama dapat digunakan untuk menyelesaikan masalah n koin apa pun dalam penimbangan $[\log_3(2k+3)]$. Implementasi algoritma yang tepat akan menemukan solusi untuk masalah n koin dalam waktu maksimal $\mathbf{O(nlog_3n)}$. n mewakili jumlah koin dalam masalah, dan $log_3 n$ mewakili jumlah penimbangan. $log_3 n$ bukan sekadar $\log_3 \mathrm{n}$ karena algoritma harus menemukan cara untuk mengelompokkan setiap koin pada setiap penimbangan. Meskipun beberapa koin tidak digunakan dalam penimbangan kedua dan ketiga dan pasti akan menangani kurang dari n koin, jumlah koin tersebut masih dalam orde n.

BAB 5
KRIPTARITMA

5.1 PENDAHULUAN

Kriptaritma adalah jenis teka-teki matematika di mana angka-angka digantikan oleh huruf alfabet atau simbol lainnya. Kriptaritma pertama kali muncul pada tahun 1864 di *majalah American Agriculturist*. Pada tahun 1931, Simon Vatriquant memperkenalkan istilah *kriptoaritmetika* (*kriptaritme* dalam bahasa Prancis) dengan nama samaran MINOS di *Sphinx*, sebuah majalah matematika rekreasi Belgia. Pada tahun 1955, J. A. H. Hunter menciptakan istilah *alfametika* untuk menunjukkan kriptoaritma yang huruf-hurufnya membentuk kata atau frasa yang bermakna.

Dalam Kecerdasan Buatan, kriptoaritma dipelajari sebagai kelas **masalah pemenuhan kendala**, yang tujuannya adalah untuk menetapkan nilai yang benar (digit) yang termasuk dalam suatu domain (0 hingga 9 dalam kasus bilangan desimal) ke sekumpulan variabel (karakter alfabet), sambil memenuhi kendala (hasil operasi aritmatika harus benar) dari masalah.

Kendala-kendala tersebut adalah sebagai berikut:

Setiap karakter alfabet hanya mewakili satu digit, dan setiap digit diwakili oleh tepat satu karakter alfabet.

Representasi numerik suatu kata tidak dimulai dengan angka nol.

Ketika karakter alfabet diganti dengan digit, hasil penjumlahannya benar secara aritmatika.

5.2 TEKNIK PEMECAHAN MASALAH

Deduksi, pembangkitan dan pengujian, penelusuran balik, penyelesaian submasalah, dan penyelesaian persamaan linear adalah beberapa teknik yang dapat membantu dalam memecahkan teka-teki kriptoaritmatika. Biasanya, soal itu sendiri berisi beberapa petunjuk yang memberikan lebih banyak informasi dan petunjuk untuk memecahkan huruf atau simbol lain. Dengan demikian, kita dapat membangun basis data informasi yang dapat membantu menemukan lebih banyak petunjuk dan akhirnya menemukan solusinya. Misalnya, mudah untuk menemukan 0 dan 9 dalam suatu persamaan karena 0 yang ditambahkan ke angka apa pun adalah angka yang sama, dan 9 yang ditambahkan ke suatu angka juga merupakan angka yang sama dengan carryover jika ada carryover dari kolom sebelumnya. Selain itu, ketika dua digit desimal ditambahkan, carryover adalah maksimum 1, sehingga digit tambahan dalam penjumlahan berarti digit tersebut adalah 1. Lebih lanjut, ketika digit yang mewakili salah satu dari dua operan yang ditambahkan berasal dari ujung ekstrem domain nilai yang mungkin (misalnya, 0, 1, atau 9), relatif lebih mudah untuk menguraikan operan lainnya jika kita tahu apakah penjumlahan menghasilkan carryover atau tidak. Dengan demikian, kita dapat menentukan hasil secara logis dari informasi yang diberikan. Dalam bab ini, kami mempertimbangkan dua masalah yang terkenal.

Masalah 1:

$$\mathbf{SEND} + \mathbf{MORE} = \mathbf{MONEY}$$

Ini mungkin teka-teki kriptoaritmetika paling klasik. Teka-teki ini diperkenalkan oleh Henry Dudeney di *Majalah Strand* pada tahun 1924. Soal ini juga merupakan teka-teki alfametika, karena kata-kata dalam teka-teki membentuk frasa yang masuk akal *KIRIM LEBIH BANYAK UANG*. Salah satu karakteristik penting dari soal ini adalah digit tambahan dalam penjumlahan (yaitu, M), karena merupakan petunjuk penting. Ciri lainnya adalah beberapa huruf, seperti M, E, dan N, diulang, sehingga semakin cepat diuraikan, semakin cepat soal dapat diselesaikan.

	S	E	N	D
+	M	O	R	E
M	O	N	E	Y

Masalah 2:

$$\mathbf{DONALD} + \mathbf{GERALD} = \mathbf{ROBERT}\ (\mathbf{Clue}: \mathbf{D} = \mathbf{5})$$

Ini adalah masalah lain yang terkenal, dan telah menjadi subjek penelitian Newell dan Simon tentang pemecahan masalah manusia menggunakan kriptoaritmatika.

	D	O	N	A	L	D
+	G	E	R	A	L	D
	R	O	B	E	R	T

Perhatikan petunjuk bahwa $\mathrm{D} = 5$. Ini sangat membantu dalam menemukan informasi lebih lanjut tentang semua variabel yang terhubung ke D oleh berbagai kendala—yaitu, T, carryover ke kolom kedua, G, dan R.

5.3 SOLUSI YANG DITAWARKAN KRIPTANIA

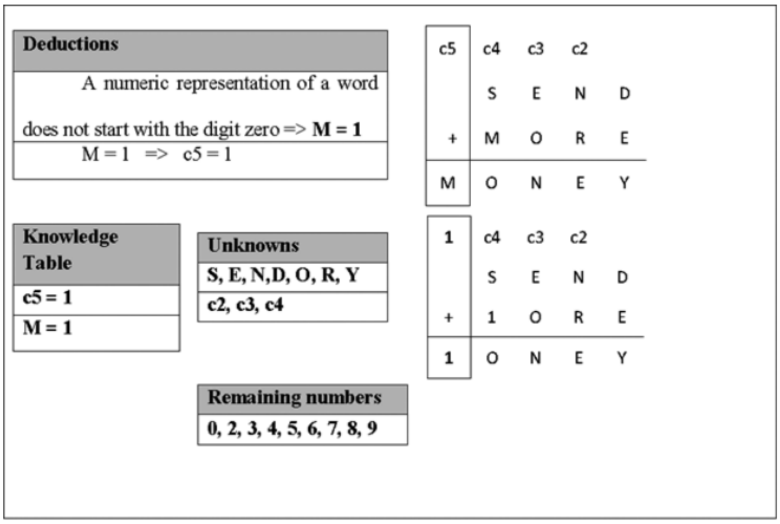
Masalah 1:

$$\mathbf{SEND} + \mathbf{MORE} = \mathbf{MONEY}$$

Berikut adalah solusi yang sangat mirip dengan banyak solusi lain yang tersedia daring. Karena jumlahnya memiliki satu digit lebih banyak daripada dua angka yang dijumlahkan, kami memilih untuk mulai menyelesaikan soal dari sisi kiri. Kami akan merujuk kolom-kolom tersebut sebagai pertama, kedua, ketiga, dan seterusnya, dimulai dari sisi kanan. Saat kami mulai menemukan petunjuk untuk solusi tersebut, kami terus menambahkannya ke Tabel Pengetahuan dan merujuknya untuk mendapatkan petunjuk tambahan.

$$
\begin{array}{ccccc}
c5 & c4 & c3 & c2 & \\
 & \mathrm{S} & \mathrm{E} & \mathrm{N} & \mathrm{D} \\
+ & \mathrm{M} & \mathrm{O} & \mathrm{R} & \mathrm{E} \\
\hline
\mathrm{M} & \mathrm{O} & \mathrm{N} & \mathrm{E} & \mathrm{Y}
\end{array}
$$

Jelaslah dari soal bahwa $\mathbf{M} = \mathbf{1}$, karena merupakan sisa dari kolom sebelumnya dan juga karena penjumlahannya tidak dapat dimulai dengan angka 0 di depan. Gambar 5.1 memperlihatkan hal ini.

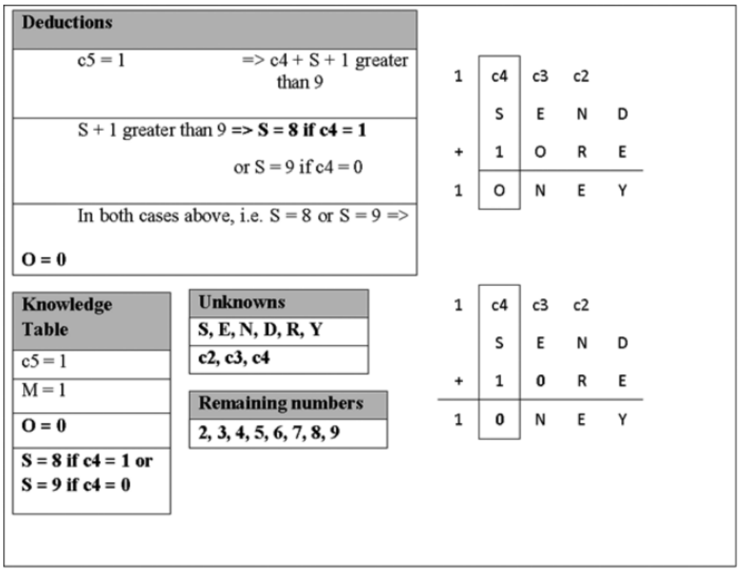


Gambar 5.1 Langkah 1

Dengan asumsi tidak ada carryover dari kolom ketiga, $S + 1 = 10$, maka S hanya bisa bernilai 9, yang membuktikan $\mathrm{O} = 0$.

Namun, jika ada carryover dari kolom ketiga, $1 + S + 1 = 10$—yaitu, $S + 2 = 10$, sehingga $\mathrm{S} = 8$, yang menghasilkan $\mathrm{O} = 0$, atau $\mathrm{S} = 9$, yang menghasilkan $\mathrm{O} = 1$, yang tidak mungkin karena M sudah bernilai 1. Jadi, jika ada carryover, $\mathrm{S} = 8$.

Bagaimanapun, $\mathbf{O} = \mathbf{0}$. Gambar 5.2 **mengilustrasikan bagaimana kita mencapai kesimpulan untuk $\mathbf{O} = \mathbf{0}$, dan $\mathbf{S} = \mathbf{8}$ atau $\mathbf{9}$**.



Gambar 5.2 Langkah 2

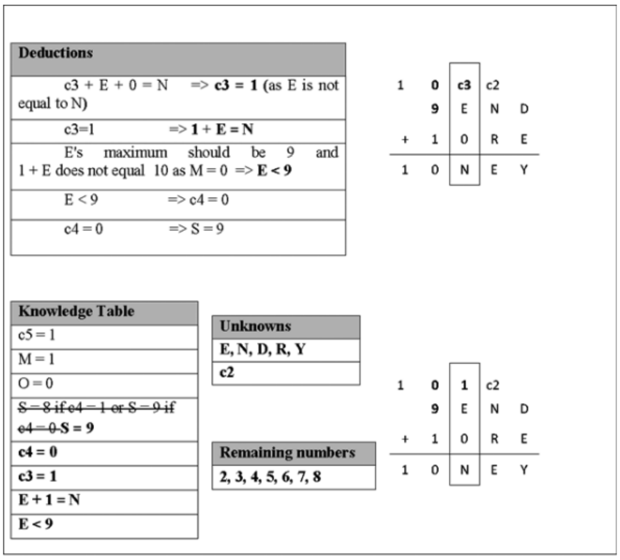
Kita lanjutkan dengan berpindah ke kolom ketiga dari kanan. Melihat Gambar 5.3, kita melihat bahwa $\mathrm{E} + 0 = \mathrm{N}$, yang tidak mungkin karena apa pun yang ditambahkan ke nol adalah dirinya sendiri. Ini berarti $\mathrm{c3} = 1$. Oleh karena itu, $E + 1 = N$. Nilai maksimum E adalah 9, dan jika $E = 9$, $N = 0$, yang tidak mungkin karena $\mathrm{O} = 0$. Ini berarti E tidak sama dengan 9 dan $\mathrm{E} < 9$. Jadi, $\mathrm{N} < 10, \mathrm{c4} = 0$, dan $\mathbf{S} = \mathbf{9}$.

Melanjutkan ke kolom keempat, kita dapat melihat pada Gambar 5.4 bahwa $N + R = E$ dan $\mathrm{c3} = 1$, jadi $N + R = E + 10$. Persamaan ini valid jika $\mathrm{c2} = 0$.

Selain itu, kita tahu bahwa $E + 1 = N$. Dengan mensubstitusikan ini ke persamaan di atas, kita mendapatkan $\mathrm{E} + 1 + \mathrm{R} = \mathrm{E} + 10$, jadi $\mathrm{R} = 9$. Tetapi $\mathrm{S} = 9$, jadi asumsi kita bahwa kolom $\mathrm{c2} = 0$ *salah*. Oleh karena itu, persamaan yang benar adalah

$1 + N + R = E + 10$

Artinya, $1 + E + 1 + R = E + 10$, yang menghasilkan $\mathbf{R} = \mathbf{8}$.



Gambar 5.3 Langkah 3

Melanjutkan ke kolom terakhir, kita memiliki $D + E = Y$, yang menghasilkan carryover ke kolom berikutnya (lihat Gambar 5.4). Jadi, $\mathrm{D} + \mathrm{E} = 10$. Sekarang, angka 0, 1, 8, dan 9 diambil.

Dengan demikian, $12 = \mathrm{D} + \mathrm{E} = 17$ dan

D dan E dapat berupa 2, 3, 4, 5, 6, atau 7.

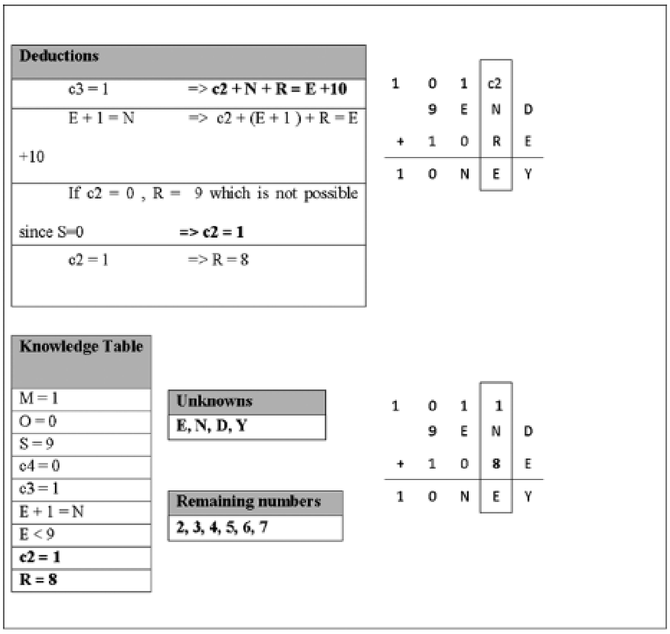
Kombinasi angka-angka di atas berikut ini memiliki jumlah antara 12 dan 17

$5 + 7 = 12$

$6 + 6 = 12$

$6 + 7 = 13$

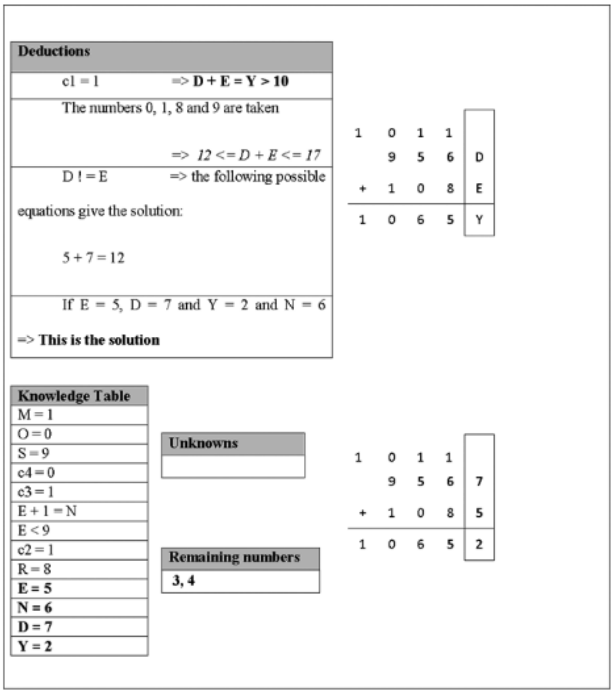
$7 + 7 = 14$



Gambar 5.4 Langkah 4

Karena D tidak sama dengan E (kendala permasalahan), hanya persamaan pertama dan ketiga yang valid. Selain itu, karena E lebih sering muncul dalam permasalahan ini, mari kita coba nilai 5, 6, dan 7 untuk E terlebih dahulu (lihat Gambar 5.5).

Jika kita menetapkan $\mathrm{E} = 5, \mathrm{D} = 7, \mathrm{Y} = 2$, dan $\mathrm{N} = 6$, kita dapat melihat bahwa masing-masing penetapan ini valid, dan dengan demikian, telah mencapai solusi.



Gambar 5.5 Langkah 5

Dengan mencoba $E = 6$, kita mendapatkan $D = 7, Y = 3$, dan $N = 7$, yang tidak mungkin karena $D = 7$.

Dengan mencoba $E = 7$ juga, kita mendapatkan $D = 6, Y = 3$, dan N = 8, yang tidak mungkin karena kita telah membuktikan $R = 8$.

Demikian pula, dengan mencoba $D = 5$, kita mendapatkan $E = 7$, sehingga $N = 8$, yang lagi-lagi tidak valid karena $R = 8$.

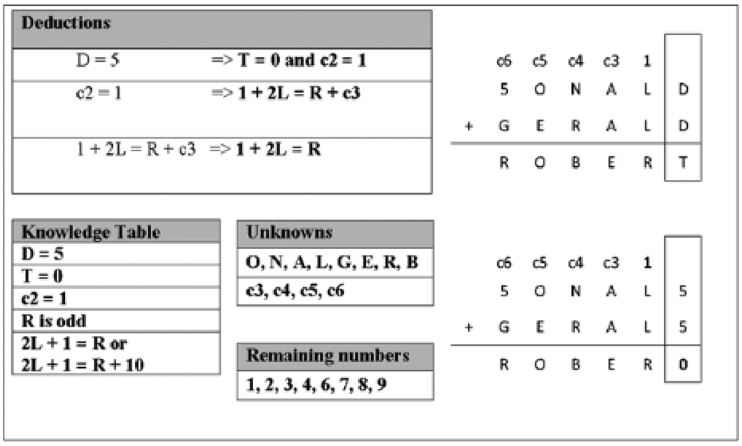
Dengan demikian, kita menyimpulkan bahwa masalah telah terpecahkan dan tidak ada solusi lain.

Masalah 2:

Kita menamai carryover $c2, c3, \ldots, c6$ untuk kolom $2, 3, \ldots, 6$.

	c6	c5	c4	c3	c2	
	D	O	N	A	L	D
+	G	E	R	A	L	D
	R	O	B	E	R	T

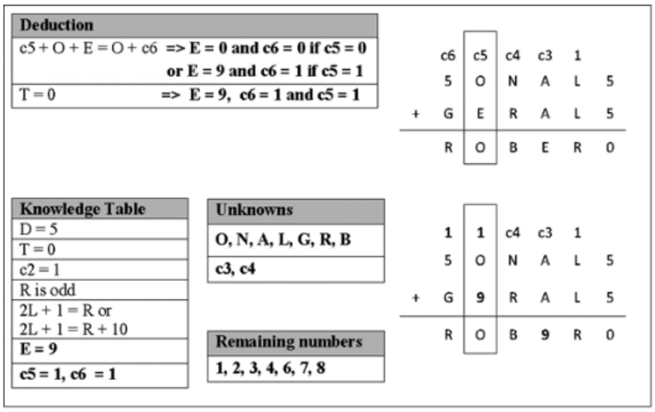
Dengan mengganti petunjuk tersebut, kita mendapatkan $T = 0$ dengan carryover ke kolom kedua. Jadi, $c2 = 1$ dan $1 + L + L = R$, sehingga **R ganjil** (lihat Gambar 5.6).



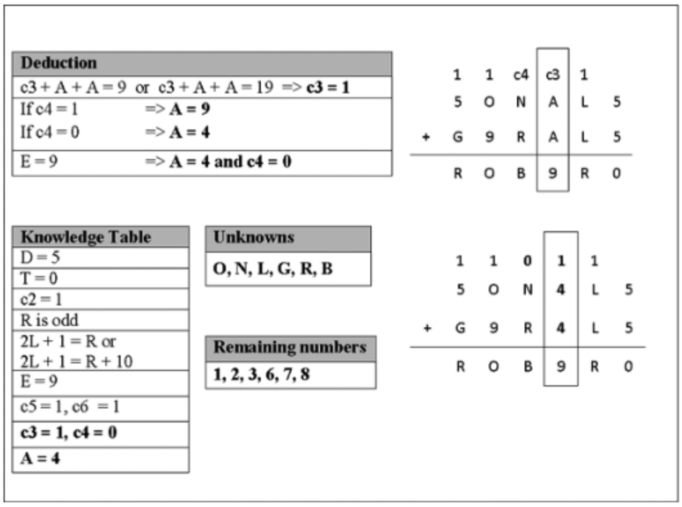
Gambar 5.6 Langkah 1

Beralih ke kolom kelima, $O + E = O$, yang berarti jika tidak ada carryover, $E = 0$, dan jika ada carryover, $E = 9$ (lihat Gambar 5.6).

Karena T sudah 0, $\mathbf{E = 9}$ dan dengan demikian **mengonfirmasi adanya carryover dari kolom keempat dan carryover ke kolom keenam—yaitu**, $c5 = c6 = 1$.



Gambar 5.7 Langkah 2

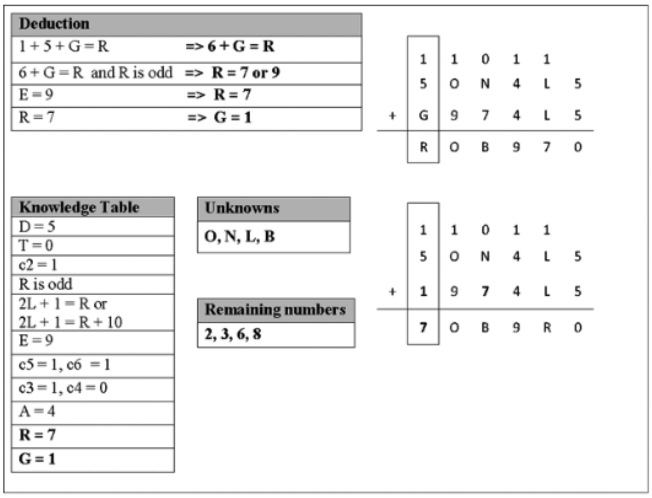


Gambar 5.8 Langkah 3

Karena $A + A = 9$ (yang merupakan bilangan ganjil), terdapat carryover dari kolom kedua ke kolom ketiga (lihat Gambar 5.8). Selain itu, tidak ada carryover dari kolom ketiga ke kolom keempat karena itu berarti $A + A + 1 = 9$, sehingga $A = 9$. Namun, angka 9 sudah diambil. Jadi **tidak ada carryover dari kolom ketiga ke kolom keempat**, sehingga $\mathbf{A = 4}$.

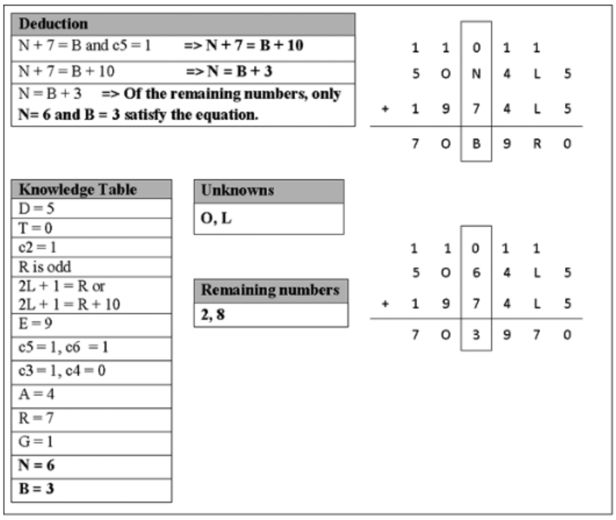
Beralih ke kolom keenam, kita mendapatkan $5 + \mathrm{G} = \mathrm{R}$ (lihat Gambar 5.9).

Karena R ganjil, $\mathrm{R} = 7$ atau $\mathrm{R} = 9$. Namun $\mathrm{E} = 9$, maka $\mathrm{R} = 7$, sehingga $\mathrm{G} = 1$ karena terdapat carryover dari kolom kelima.

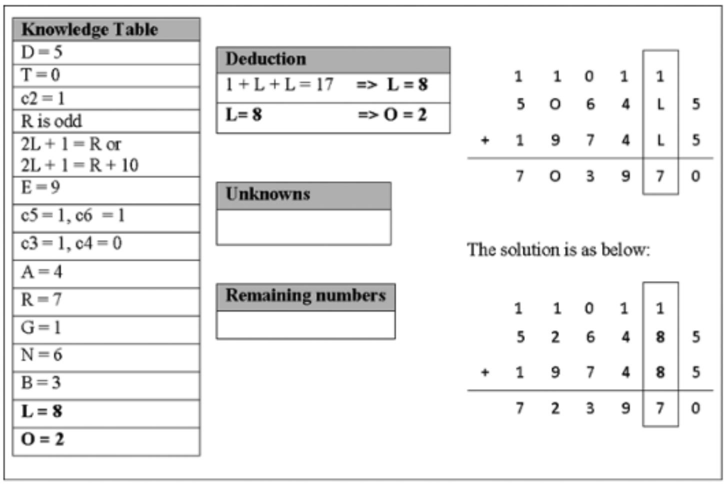


Gambar 5.9 Langkah 4

Pada kolom keempat, $\mathrm{N} + 7 = \mathrm{B}$ dengan carryover ke kolom kelima (lihat Gambar 5.10). Selain itu, karena tidak ada carryover dari kolom ketiga, N bisa berupa 2, 3, 4, 6, atau 8. Untuk menghasilkan carryover, N harus lebih besar dari 2. $\mathrm{N} = 3$ atau 4 menghasilkan $\mathrm{B} = 0$ atau 1, yang keduanya sudah ditetapkan. $\mathrm{N} = 6$ menghasilkan $\mathrm{B} = 3$, dan $\mathrm{N} = 8$ menghasilkan $\mathrm{B} = 5$, yang sekali lagi tidak valid. Jadi, $\mathbf{N = 6 \text{ dan } B = 3}$.



Gambar 5.10 Langkah 5



Gambar 5.11 Langkah 6

Sekarang kita juga memiliki beberapa data baru untuk kolom kedua: $1 + L + L = 7$ (lihat Gambar 5.11). Karena ada carryover ke kolom ketiga, $L = 8$ sehingga $\text{jumlahnya menjadi} = 1 + 8 + 8 = 17$. Satu-satunya digit yang tersisa adalah 2, dan alfabet yang tersisa adalah O. Jadi, $O = 2$.

Dari dua solusi di atas, jelas bahwa kita juga telah menghitung nilai carryover, yang juga dikenal sebagai variabel bantu. Meskipun variabel bantu tidak secara langsung menjadi bagian dari solusi, variabel bantu membantu dalam proses pemecahan masalah.

Mari kita perhatikan Soal 1: $\text{KIRIM} + \text{LEBIH BANYAK} = \text{UANG}$.

Variabel-variabel tersebut adalah S, E, N, D, M, O, R, dan Y.

Domain untuk variabel-variabel ini adalah 0 hingga 9.

Variabel bantu adalah $c1, c2, c3, \text{dan } c4$ (carryover).

Domain untuk variabel bantu adalah 0, 1.

Kendala-kendalanya adalah sebagai berikut:

Semua variabel mewakili angka yang berbeda.

Angka-angka tersebut tidak dimulai dengan angka 0.

$S = c4$

$c3 + S + M = O + 10^\circ c4$

$c2 + E + O = N + 10^\circ c3$

$c1 + N + R = E + 10^\circ c2$

$D + M = Y + 10^\circ c1$

Jika kita melakukan pencarian solusi secara menyeluruh melalui ruang keadaan masalah, kita akan segera menyadari bahwa ini merupakan tugas yang sangat berat, karena jika suatu masalah terdiri dari 10 variabel yang berbeda, terdapat $10! = 3.628.800$ kemungkinan penugasan. Namun, seperti yang telah diamati sebelumnya, jika kita dapat menguraikan beberapa variabel menggunakan petunjuk yang melekat dalam masalah, seperti digit tambahan dalam penjumlahan atau angka 0 atau 9, kita dapat mengurangi jumlah kemungkinan kombinasi—tetapi hanya sampai batas tertentu.

Sebagai alternatif, kita dapat merumuskan masalah sebagai pohon pencarian dengan mempertimbangkan salah satu nilai yang ditemukan pada awalnya sebagai keadaan awal dan keadaan tujuan, yaitu situasi di mana nilai setiap variabel dalam solusi diketahui. Kita juga dapat membagi masalah menjadi submasalah. Dalam hal ini, kita dapat menganggap kolom sebagai submasalah dan mempertimbangkan penyelesaian satu submasalah pada satu waktu dengan menghitung nilai variabel dalam kolom tersebut.

Namun, sia-sia untuk mempertimbangkan kolom secara terpisah karena carryover dari jumlah dalam satu kolom memengaruhi jumlah di kolom berikutnya, sehingga kolom-kolom tersebut saling terhubung. Hubungan antara kolom-kolom ini—atau, lebih spesifik, antara variabel-variabel—terlihat jelas dari daftar kendala dan dapat dilihat lebih jelas melalui *hipergraf kendala*, yang dikembangkan George Luger untuk masalah ini. Oleh karena itu, kita harus mempertimbangkan kendala yang menghubungkan satu kolom dengan kolom lainnya. Jika kita perhatikan lebih dekat, variabel bantu (yaitu, carryover)lah yang menghubungkan dua kolom. Untungnya, domain variabel bantu ini sangat kecil, karena nilainya 1 atau 0. Jadi, jika kita mengidentifikasi variabel bantu ini dengan benar, kita mungkin dapat memangkas pohon pencarian dan mengurangi upaya dan waktu pencarian. Selain itu, kita dapat mengasumsikan salah satu nilai legal untuk variabel bantu dan mencari nilai untuk variabel yang tersisa di kolom tertentu.

Heuristik ini, yang melibatkan pertimbangan nilai variabel sesedikit mungkin, disebut *heuristik nilai tersisa minimum* (*MRV*). Dengan demikian, setelah menerapkan "teknik pembangkitan dan pengujian" pada variabel dengan MRV, domain variabel lain di kolom tersebut juga berkurang. Kita juga mempertimbangkan nilai kedua yang mungkin untuk nilai bantu dan membandingkan domain variabel yang dihasilkan di kolom tersebut.

Misalnya, pada Soal 1, perhatikan kolom 4 setelah mensubstitusi $\mathrm{M} = \mathrm{c4} = 1$:

$\mathrm{c3} + \mathrm{S} + 1 = \mathrm{O} + 10$

Jika $\mathrm{c3} = 1, \mathrm{S} + 1 = \mathrm{O} + 10$

Jadi, $\mathrm{S} + 1 = 10$, dan satu-satunya nilai legal yang mungkin untuk S adalah 9 dan untuk O adalah 0.

Demikian pula, jika $\mathrm{c3} = 1, \mathrm{S} + 2 = 10$, dan satu-satunya nilai legal yang mungkin untuk S adalah 8 dan untuk O adalah 0.

Pertanyaannya sekarang adalah apakah urutan dua kemungkinan nilai $\mathrm{c3}$ perlu dipertimbangkan atau tidak. Dalam kasus ini, urutannya tidak penting karena domain S direduksi menjadi hanya satu nilai legal (yaitu, 8 atau 9) dalam kedua kasus (untuk kedua kemungkinan nilai c3)—yaitu, 1 atau 0, berturut-turut. Dalam kasus di mana pengujian nilai probabilitas untuk suatu variabel menghasilkan domain yang terlalu kecil atau nol untuk variabel tetangganya, variabel tersebut dipertimbangkan terakhir. Heuristik ini, yang memilih nilai yang mengesampingkan nilai terkecil yang mungkin untuk variabel tetangga, disebut *nilai pembatas terkecil* (LCV).

Dengan demikian, ketika menggunakan heuristik MRV, kita dapat menemukan nilai O dan dapat mereduksi domain S menjadi dua nilai: 8 dan 9. Hal ini mencegah kita dari keharusan menguji semua nilai lain dari domain untuk variabel-variabel tersebut, yang nantinya akan gagal.

Setelah memilih LCV untuk suatu variabel, kita dapat mulai menurunkan kemungkinan nilai legal untuk variabel yang tersisa dan melanjutkan lebih jauh pada pohon pencarian menuju solusi. Teknik ini disebut *pemeriksaan maju*. Dengan cara ini, jika kita mencapai jalan buntu atau keadaan di mana tidak ada nilai legal yang tersisa untuk suatu variabel, kita menelusuri kembali ke penugasan terakhir. Teknik ini disebut *pelacakan mundur*.

5.4 PEMECAHAN MASALAH MANUSIA

Jika kita menganalisis solusi untuk Masalah 1, kita menyadari bahwa solusi tersebut dimulai dengan menetapkan nilai 1 untuk M, yang disimpulkan dari kendala 2 dan 3. Kemudian pada kolom 4, variabel bantu $c3$ dipilih sebagai MRV, karena domainnya lebih kecil dibandingkan dengan dua variabel lainnya. Dengan demikian, kita hanya memiliki dua kemungkinan nilai untuk huruf S dan hanya satu nilai untuk O. Setelah substitusi $O = 0$ pada kolom 3, hubungan antara E dan N ditemukan, yang membantu menemukan nilai R. Setelah menguraikan semua variabel bantu yang tersisa, domain variabel pada kolom pertama direduksi menjadi hanya tiga angka, karena pencarian menyeluruh menggunakan kombinasi ketiga nilai tersebut, dan penerapan kendala mengarah pada solusi.

Solusi untuk Masalah 2 dimulai dengan jalur yang sama, yaitu substitusi petunjuk dan deduksi nilai/paritas variabel terkait. Solusinya berlanjut dengan menemukan nilai 9 dan nilai sebagian besar variabel bantu. Selain itu, kolom-kolom diamati satu per satu, substitusi pengetahuan yang diperoleh.

5.5 ANALISIS JENDELA MANUSIA TERHADAP SOLUSI

Secara keseluruhan, setiap solusi yang kami temukan mengikuti konsep umum yang sama: menunjukkan (setidaknya) kriptoaritma yang telah dienkripsi dan didekripsi sepenuhnya dan memberikan deskripsi langkah demi langkah tentang bagaimana solusi tersebut ditemukan.

Saat menganalisis solusi yang kami temukan (Tabel 5.1), kami dapat melihat bahwa semuanya direpresentasikan menggunakan teks. Hal ini karena masalah aritmatika matematika, beserta penalaran di baliknya, lebih mudah dipahami melalui kata-kata daripada gambar atau diagram. Oleh karena itu, pilihan representasi cenderung tidak berperan dalam menentukan solusi terbaik. Sebaliknya, pengaturan, detail, dan deskripsi langkah-langkah memainkan faktor utama dalam mengidentifikasi solusi yang dianggap paling kompatibel.

Sebagian besar solusi bersifat ekstensional, terutama terdiri dari metode untuk menyelesaikan masalah **KIRIM LEBIH BANYAK UANG** dan masalah **DONALD GERALD ROBERT**. Mereka dianggap ekstensional karena menjelaskan prosedur langkah demi langkah untuk menyelesaikan salah satu dari dua kriptaritma ini secara keseluruhan. Solusi intensional yang ditemukan, seperti "Solusi Primer," merangkum kemungkinan kendala dan teknik aritmatika untuk menyelesaikan kriptaritma apa pun, bahkan untuk yang melibatkan aritmatika selain penjumlahan.

Peringkat untuk eksekusi dan pemahaman umumnya beragam. Karena setiap solusi menggunakan teks sebagai representasi sarana, solusi tersebut dianalisis lebih mendalam, terutama dalam hal organisasi, detail, dan deskripsi langkah-langkah, seperti yang disebutkan sebelumnya. Namun, dapat dilihat bahwa solusi intensional umumnya berperingkat lebih rendah. Hal ini kemungkinan besar karena mereka menggunakan teknik aritmatika umum yang harus diingat untuk menyelesaikan kriptaritma, alih-alih solusi kriptaritma yang tersedia bagi pemecah masalah, seperti dalam kasus solusi ekstensional.

Tabel 5.1 Pemeringkatan Solusi Kriptaritmetika Berdasarkan Jendela Manusia

KESULITAN
8/10

KERUMITAN
O((n – (constrs + N/A))!)

NAMA	INT ATAU EXT?	INTENS. /EKSTENS.	REPR.	JENDELA MANUSIA?	BENAR ?	UKURAN BUTIR	EKSEKUSI	KETERPAHAMAN	METODE PEMECAHAN MASALAH	FLEKSIBILITAS	MODE KONVERSI	OPTIMAL?	TOTAL
Ariels	Ekst.	6/10	Tks	Y	Y	Ideal	7/10	6/10	Pemuasan Kendala	9/10	Editor Tks; Tangan	Y	28/40
Code	Int.	7/10	Kode/Algotm	Y	Y	Ideal	6/10	4/10	Pemuasan Kendala; Kekuatan Brutal	8/10	Editor Tks; Tangan	N	25/40
Donald	Ekst.	9/10	Tks	Y	Y	Kecil	5/10	5/10	Pemuasan Kendala	9/10	Editor Tks; Tangan	Y	28/40
Tabel Pengetahuan	Ekst.	8/10	Tks dng Gbr	Y	Y	Ideal	10/10	10/10	Pemuasan Kendala	9/10	Editor Tks; Tangan	Y	37/40
M and Cs	Ekst.	8/10	Tks dng Tabel	Y	Y	Ideal	9/10	8/10	Pemuasan Kendala	8/10	Editor Tks; Pembuat Tbl; Tangan	Y	33/40
Pavalli	Ekst.	9/10	Tks	Y	Y	Ideal	7/10	9/10	Pemuasan Kendala	9/10	Editor Tks; Tangan	Y	34/40
Primer	Int.	4/10	Tks	Y	Y dng Siklus	Sangat Besar	6/10	5/10	Pemuasan Kendala; Kekuatan Brutal	8/10	Editor Tks; Tangan	N	23/40

Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?

Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tidak mengherankan, setiap solusi menggunakan pemenuhan kendala sebagai teknik pemecahan masalah. Metode ini, bersama dengan bantuan penyelesaian subtujuan, sangat penting bagi kriptaritmetika. Sedangkan untuk solusi intensional, mereka menggunakan pemenuhan kendala yang disertai dengan penggunaan metode *brute force*. Hal ini karena, tidak seperti masalah SEND MORE MONEY, kriptaritma umumnya tidak unik. Oleh karena itu, ada kemungkinan bahwa satu huruf atau simbol dapat mewakili beberapa nilai yang mungkin, dan bukan digit tertentu. Untuk menyelesaikan jenis kriptaritma ini, metode brute force harus digunakan.

Terakhir, penting juga untuk dicatat bahwa fleksibilitas solusi ini memiliki peringkat yang cukup tinggi. Karena direpresentasikan menggunakan teks, solusi ini dapat disajikan dalam berbagai cara.

Solusi yang Paling Kompatibel dengan Jendela Manusia

Solusi yang dinilai sebagai MHWC adalah "Solusi Tabel Pengetahuan". Solusi ini sama dengan solusi yang disajikan di Bagian 5.3. Solusi ini sederhana, namun detail, untuk kriptoaritma SEND MORE MONEY. Kita dapat menganggapnya sebagai MHWC karena beberapa alasan. Salah satunya adalah karena solusi ini dibagi menjadi beberapa langkah, sehingga sangat mudah dipahami. Alasan lainnya adalah karena solusi ini membagi kolom-kolom persamaan menjadi sub-masalah, sehingga sangat mudah dieksekusi.

Alasan ketiga adalah karena persamaan ini diulang secara konsisten di setiap langkah, menunjukkan nilai terkini pada langkah tersebut dalam solusi dan kolom yang sedang dianalisis. Selain itu, simbol dan nilai carryover juga ditampilkan. Hal ini mencegah perlunya menghafal atau menuliskan nilai huruf dan carryover, serta membuat pembaca tetap terlibat.

Alasan keempat adalah tingkat detail deskripsinya. Solusi ini memiliki tingkat detail yang tepat karena menjelaskan secara tepat mengapa huruf-huruf tersebut diberi nilai tertentu. Selain itu, bahasa deskripsinya sangat eksplisit, sehingga bahkan mereka yang tidak memiliki latar belakang matematika tingkat lanjut pun dapat memahami solusinya dengan baik. Alasan terakhir, dan paling substansial, mengapa solusi ini dianggap sebagai solusi MHWC adalah karena penggunaan Tabel Pengetahuan.

Tabel-tabel ini menampilkan nilai huruf, deduksi, dan variabel yang tidak diketahui yang tersisa pada setiap langkah. Seiring berjalannya solusi, lebih banyak elemen ditambahkan atau dihapus dari tabel-tabel ini. Tabel-tabel ini penting karena merupakan cara yang ringkas dan padat untuk mengorganisasikan data. Hal ini membuat solusi ini sangat mudah dipahami dan dieksekusi.

Solusi yang Paling Tidak Kompatibel dengan Jendela Manusia

```
DONALD + GERALD = ROBERT
Solution:
c1 c2 c3 c4 c5
D O N A L D
G E R A L D
-----------
R O B E R T

Steps:
1) T is even (D + D = an even number)

2) D ≥ 5 ⟹ c5 = 1
  D ≤ 4 ⟹ c5 = 0

3) L ≥ 5 ⟹ c4 = 1
  L ≤ 4 ⟹ c4 = 0
4) A ≥ 5 ⟹ c3 = 1
  A ≤ 4 ⟹ c3 = 0

5) c5 = 1 ⟹ R is odd
  c5 = 0 ⟹ R is even (L + L)
  R is not 0 (R and L cannot both be 0)

6) c4 = 1 ⟹ E is odd
  c4 = 0 ⟹ E is even (A + A = even)
  E not 0 (E and A cannot both be 0)

7) c2 = 0 ⟹ E = 0
  c2 = 1 ⟹ E = 9 and c3 + N + R > 9

8) c1 + D + G ≤ 9 (no carry)
  R ≤ D
  R ≤ G
          …
          …
          …
          …
          …
```

Gambar 5.12 “Solusi Donald” (Disingkat)

Solusi yang dinilai sebagai solusi LHWC adalah “Solusi Donald” (Gambar 6.8). Ini adalah solusi untuk kriptaritma DONALD GER-ALD ROBERT, dengan representasi yang mirip dengan yang disajikan dalam. Solusi ini terdiri dari fungsi ekivalensi yang secara implisit dan ringkas menunjukkan nilai potensial huruf dan carryover pada setiap langkah. Meskipun solusi ini disusun sebagai langkah-langkah, solusi ini menunjukkan bagaimana nilai-nilai tersebut diperoleh hanya dengan penjelasan matematis singkat dan sedikit deskripsi. Faktor-faktor ini berpotensi menurunkan eksekusi dan pemahaman solusi, terutama bagi mereka yang tidak familiar dengan kriptaritma. Meskipun demikian, solusi ini mungkin cocok bagi mereka yang memiliki kecenderungan matematis dan/atau logis yang tinggi.

Fitur lain yang mungkin menjadi masalah dengan solusi ini adalah fakta bahwa, tidak seperti "Solusi Tabel Pengetahuan", kriptaritma tidak diulang di antara langkah-langkah. Hal ini semakin mengurangi jumlah panduan yang diberikan kepada pembaca dan dengan demikian dapat semakin mengurangi pemahaman solusi.

Terakhir, Anda mungkin memperhatikan bahwa solusi dimulai dengan menganalisis huruf-huruf di kolom paling kanan, bukan kolom paling kiri. Seperti yang terlihat pada solusi sebelumnya dalam bab ini, kita biasanya mulai dengan menganalisis huruf-huruf di kolom paling kiri, lalu bergerak menuju kolom paling kanan. Pendekatan ini lebih efisien karena kemungkinan akan menentukan nilai variabel carryover lebih awal dalam proses. Namun, memulai dari kolom paling kanan dapat menghasilkan lebih banyak langkah, seperti yang terjadi pada "Solusi Donald". Oleh karena itu, hal ini juga dapat mengurangi eksekusi solusi.

5.6 SOLUSI MESIN TERBAIK

Hingga saat ini, kami hanya menganalisis kriptoaritma yang nilainya berisi hingga 10 kemungkinan nilai (yaitu, digit 0 hingga 9). Artinya, kami telah menyelesaikan persamaan yang nilainya berbasis 10, yang pada dasarnya merupakan sistem bilangan yang kita gunakan setiap hari. Namun, kemungkinan penggunaan basis yang berbeda sudah pasti dipertimbangkan.

Untuk sebuah mesin, menentukan apakah sebuah kriptoaritma memiliki solusi bersifat **NP-lengkap** ketika digeneralisasi ke basis sembarang. Ini berarti tidak ada algoritma yang diketahui yang dapat menjamin solusi untuk sebuah kriptoaritma dalam jangka waktu yang dapat diterima seiring bertambahnya jumlah digit yang tidak diketahui (basis) (namun, solusi dapat diverifikasi dalam waktu polinomial). Untuk setiap kriptoaritma yang nilainya berbasis n, waktu proses terburuk untuk menemukan solusi tersebut adalah $\mathbf{O}(\mathbf{n}!)$. Hal ini terjadi ketika semua n digit digunakan dalam persamaan dan ketika tidak ada kendala yang dapat disimpulkan dengan penggunaan aturan aritmatika matematika. Namun, kemungkinan besar masalah kriptoaritmatika tidak akan memuat semua nilai n-nya dan akan memuat kendala matematika dalam bentuk tertentu. Dengan demikian, waktu eksekusi rata-rata adalah $\mathbf{O}((\mathbf{n}-(\mathbf{kendala}+\mathbf{digit\ yang\ tidak\ digunakan}))!)$.

Untungnya, sebagian besar masalah kriptoaritmatika yang dianalisis cenderung memuat huruf dengan nilai dalam basis 10 (misalnya, masalah KIRIM LEBIH BANYAK UANG), karena ini adalah basis yang biasa kita gunakan. Untuk kriptoaritma dalam basis 10, kita mengganti n dengan 10 dan dengan demikian menemukan bahwa waktu eksekusi kasus terburuk adalah $\mathbf{O}(\mathbf{10}!)$ dan kasus rata-ratanya adalah $\mathbf{O}((\mathbf{10}-(\mathbf{kendala}+\mathbf{digit\ yang\ tidak\ digunakan}))!)$.

Meskipun hal ini sangat mengurangi ruang masalah, 10! masih relatif besar. Bagi sebuah mesin, salah satu algoritma paling efisien untuk menemukan solusi bagi kriptoaritma adalah **algoritma genetik paralel**. Algoritma ini pada dasarnya sama dengan algoritma genetik pada umumnya (yaitu, "Solusi Kode") tetapi dieksekusi dengan bantuan beberapa prosesor. Dalam sebuah percobaan yang dilakukan di Universitas Shahid Chamran, waktu yang dibutuhkan untuk memecahkan kriptoaritma dengan algoritma genetik paralel meningkat secara proporsional seiring dengan bertambahnya jumlah variabel yang tidak diketahui. Setelah melakukan percobaan ini pada kriptoaritma basis 10 yang berisi semua 10 variabel, dibutuhkan waktu 3,5 detik untuk menemukan solusi menggunakan algoritma generik paralel, sedangkan algoritma depth-first membutuhkan waktu sedikit lebih dari satu jam.

5.7 MASALAH TERKAIT

Masalah yang berkaitan dengan kriptoaritma adalah "masalah jarum jam". Ini bukanlah masalah kriptoaritma itu sendiri, tetapi proses penyelesaiannya sangat mirip dengan cara seseorang menyelesaikan masalah kriptoaritma. Masalah ini dijelaskan sebagai berikut: Tanggal dan waktu ditulis dalam format $JJ: BB: DD\ DD/MM$, sehingga 6 menit 34 detik lewat pukul 5 sore tanggal 25 Juli akan ditulis sebagai 17:06:34 25/07. Dengan menggunakan semua digit 0-9 hanya sekali, berapa waktu dan tanggal paling awal yang dapat dihasilkan oleh jam dan berapa yang paling akhir?

Solusi untuk masalah ini mengharuskan pemain untuk memahami batasan-batasan pada masalah tersebut. Kendala pertama adalah bahwa digit pertama untuk bulan hanya bisa 0 atau 1, dan jika 1, maka digit berikutnya hanya bisa 0 atau 2, karena hanya ada 12 bulan dan digit tidak dapat diulang. Kendala kedua adalah bahwa digit pertama untuk tanggal hanya bisa 0, 1, 2, atau 3, dan jika 3, hanya 0 atau terkadang 1 untuk digit kedua, karena itu hanya ada 30 atau 31 hari untuk sebagian besar bulan, kecuali Februari. Kendala ketiga adalah pada jam di mana digit pertama hanya bisa 0, 1, atau 2, dan jika "2," maka hanya "0," "1," atau "3" untuk digit kedua, karena hanya ada 24 jam dalam sehari dan jam 12 biasanya ditulis sebagai "00" yang tidak akan tersedia karena digit yang berulang. Kendala keempat akan dibangun di atas semua kendala sebelumnya di mana angka 6, 7, 8, atau 9 hanya dapat berada pada digit kedua dalam salah satu dari 5 kategori tanggal dan waktu. Ini karena alasan yang disebutkan di atas, dan sebagai tambahan, hanya boleh ada maksimal 59 menit atau detik karena begitu mencapai 60, akan terjadi pengaturan ulang ke "00". Oleh karena itu, hanya mungkin keempat angka ini dapat muncul hanya untuk digit kedua dari empat dari lima kategori. Kendala terakhir adalah prioritas kategori. Yaitu, bulan diutamakan daripada tanggal, kemudian diikuti jam, menit, dan terakhir detik. Mengikuti semua kendala ini, pada akhirnya akan mengarah pada solusi 17:48:59 26/03 untuk waktu dan tanggal paling awal, dan 17:56: 09 untuk yang paling akhir. Cobalah.

5.8 PROGRAM YANG DAPAT DIMAINKAN

Saat ini, terdapat banyak aplikasi yang tersedia di iPhone dan ponsel Android yang telah menjaga minat terhadap kriptoaritmetika tetap hidup. Dengan munculnya komputer modern beberapa dekade yang lalu, orang-orang mulai menerbitkan kriptoaritma yang dihasilkan komputer yang terlalu rumit untuk dipecahkan oleh manusia, yang menyebabkan penurunan minat terhadap permainan ini. Kriptaritma yang berada di dalam Jendela Manusia menyenangkan untuk dipecahkan dan membantu meningkatkan keterampilan pemecahan masalah kita. Namun, mencatat tabel pengetahuan, status terkini masalah, dan berbagai kemungkinan variabel hanya dengan menggunakan pena dan kertas bisa menjadi pekerjaan yang membosankan. Di sinilah memainkan aplikasi dan permainan komputer yang berisi kriptoaritma dapat membantu. Kita dapat memilih nilai variabel berdasarkan logika dan heuristik, sementara komputer dapat melakukan perhitungan kompleks dan manajemen memori. Salah satu aplikasi tersebut dapat ditemukan di *http://www.letsget-wordy.com/#am_app*.

Gambar diterbitkan dengan izin dari Thomas L. Peterson, Eximietate, Inc.

BAB 6
TEKA-TEKI KELEDAI MERAH

6.1 PENDAHULUAN

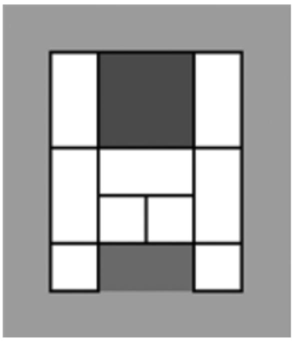
Selain dikenal sebagai variasi dari Dad's Puzzler, yang mungkin merupakan teka-teki tertua yang diketahui dari jenisnya, **Teka-teki Keledai Merah** tercatat sebagai teka-teki balok geser terlaris ketiga (setelah The 15 Puzzle dan Dad's Puzzler) dalam buku *Sliding Piece Puzzles* karya L. Edward Hordern. Kemunculan pertama teka-teki ini tercatat di Prancis, di mana ia dikenal sebagai $L'\hat{A}ne\ Rouge$ (Keledai Merah) dan sangat populer. Versi bahasa Inggris dari teka-teki ini, yang dikenal sebagai Keledai Merah, muncul pada awal tahun 1930-an, sementara di Polandia dikenal sebagai *Klocki* (balok kayu) dan di Jepang sebagai *Hakoiri musume* (Putri dalam Kotak).

Teka-teki Keledai Merah adalah jenis teka-teki balok geser yang terdiri dari beberapa petak (atau balok) yang dibatasi oleh suatu area tertentu di mana petak-petak tersebut harus disusun dalam susunan tertentu. Secara khusus, Teka-teki Keledai Merah dibatasi oleh area persegi panjang berukuran $4'' \times 5''$ dan terdiri dari 10 balok: satu balok persegi berukuran $2'' \times 2''$ (dikenal sebagai kepingan Keledai), empat balok vertikal berukuran $1'' \times 2''$, satu balok horizontal berukuran $2'' \times 1''$, dan empat balok persegi berukuran $1'' \times 1''$. Batasan pada teka-teki ini adalah sebagai berikut:

Balok-balok hanya dapat meluncur di dalam area persegi panjang yang dibatasi.

Balok-balok tersebut tidak dapat diangkat dari area yang akan diganti.

Susunan balok pada keadaan awal teka-teki ditunjukkan pada Gambar 6.1. Pada keadaan awalnya, balok berukuran $2'' \times 2''$, yang merupakan balok terbesar, berada di bagian tengah atas area yang dibatasi, dan tujuannya adalah untuk memindahkannya ke bagian tengah bawah. Kami juga memberi nomor pada blok-blok tersebut sehingga dapat dengan mudah dirujuk dan solusinya dipahami dengan jelas.



Gambar 6.1 Teka-Teki Keledai Merah

6.2 SOLUSI TEKA-TEKI KELEDAI MERAH

Pada bagian ini, kami mempelajari solusi yang terdiri dari 81 langkah, serupa dengan yang diberikan oleh Martin Gardner dalam *Scientific American* edisi Maret 1964. Untuk Teka-teki Keledai Merah, solusi apa pun yang mencapai kondisi tujuan dalam 81 langkah dianggap optimal, karena ini diketahui sebagai jumlah langkah terkecil yang diperlukan untuk menyelesaikan teka-teki. Sebuah permainan daring (tersedia di *www.bsswebsite.me.uk*) digunakan untuk memainkan teka-teki dan menunjukkan beberapa tahapan penting dalam penyelesaiannya. Berikut solusinya dengan angka yang mewakili balok:

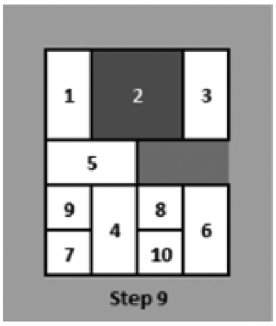
10 (setengah), 6, 5, 7 (bawah), 4
9 (setengah), 7, 4, 5, 8 (atas, kiri)
10, 4, 8 (kanan, bawah), 5, 10
8, 6, 4, 9, 7, 5, 8 (bawah, kiri), 4, 6, 3
2, 1, 10, 8, 4, 1, 2, 3, 6, 7, 8 (kiri, atas), 5, 1, 4, 7
2, 10, 8 (atas, kanan), 7, 4
1, 9 (kiri, bawah), 2, 10 (bawah, kiri), 3
6, 2, 10, 8 (setengah), 7, 4, 1, 9, 10 (bawah), 2
6, 3, 7, 8, 4, 1, 2, 8, 7, 3
6, 8 (kanan, atas), 5, 10, 9
2, 7, 8, 5, 9 (atas, kanan), 2

Setelah beberapa kali percobaan dan setelah memahami permasalahan, kami menyadari bahwa ruang kosong yang tercipta dengan memindahkan balok-balok persegi satuan harus dibentuk menjadi persegi panjang horizontal atau vertikal agar balok-balok yang lebih besar dapat bergerak. Dengan demikian, balok-balok persegi kecil lebih fleksibel dibandingkan dengan balok-balok yang lebih besar, karena dapat digunakan untuk menciptakan ruang bagi balok-balok yang lebih besar, baik dalam arah horizontal maupun vertikal. Lebih lanjut, salah satu pengamatan utama adalah bahwa balok horizontal di tengah menghalangi pergerakan balok keledai dan balok vertikal. Oleh karena itu, sebaiknya balok ini dipindahkan dari tengah ke samping. Dengan demikian, salah satu subtujuan penting dari permasalahan ini adalah memindahkan balok horizontal ke sudut atau samping, menjauh dari tengah, sehingga memungkinkan pergerakan bebas balok-balok lainnya.

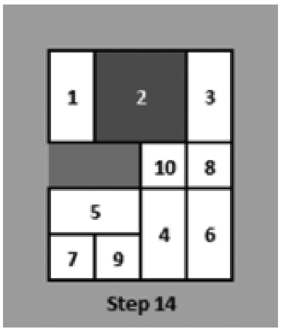
Langkah-langkah berikut mengarah ke keadaan yang ditunjukkan pada Gambar 6.2:

10 (setengah), 6, 5, 7 (turun), 4, 9 (setengah), 7, 4, dan 5

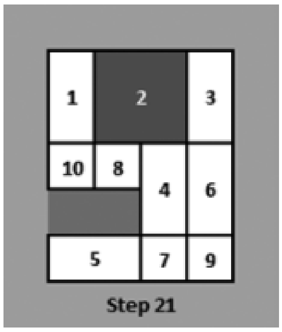
Dalam keadaan ini, kita dapat memindahkan balok 5 ke samping. Setelah lima langkah lagi, balok 5 bergerak satu baris ke bawah dengan bantuan balok-balok kecil seperti yang ditunjukkan pada Gambar 6.3 dan setelah langkah ke-21, balok 5 berada di baris terbawah (Gambar 6.4).



Gambar 6.2 Langkah 9

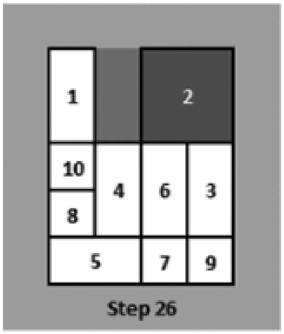


Gambar 6.3 Langkah 14

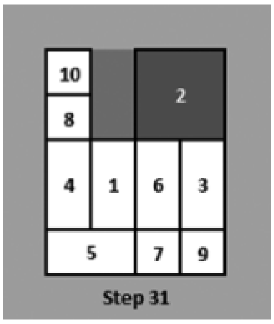


Gambar 6.4 Langkah 21

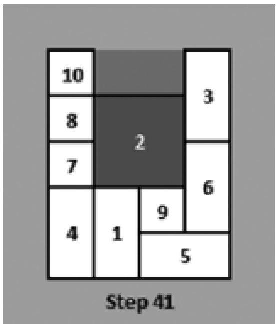
Kita sekarang memiliki ruang untuk memindahkan balok vertikal sehingga balok 2 dapat bergerak secara lateral, seperti yang terlihat pada Gambar 6.5 setelah gerakan ke-26. Namun, balok persegi kecil perlu berdekatan dengan balok 2 untuk membantunya bergerak karena balok vertikal masih membatasi pergerakan balok 2. Balok persegi kecil tersebut berdekatan dengan balok keledai pada Langkah 31 (Gambar 6.6).



Gambar 6.5 Langkah 26

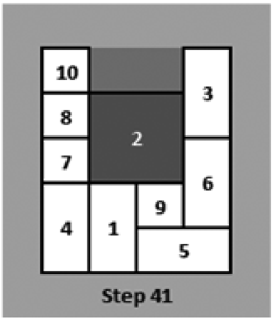


Gambar 6.6 Langkah 31

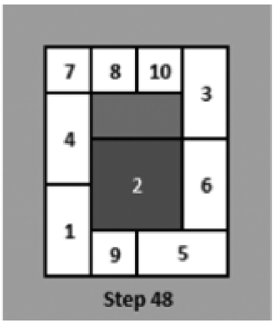


Gambar 6.7 Langkah 41

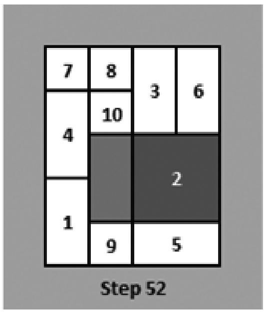
Pada Langkah 41 (Gambar 6.7 dan Gambar 6.8), blok horizontal telah berpindah ke pojok kanan bawah, memungkinkan blok vertikal 4 dan 1 bergerak lebih jauh ke bawah, sehingga memberi ruang bagi salah satu dari dua blok persegi satuan yang tersisa untuk bergerak ke atas. Sangat penting untuk menjaga salah satu blok persegi satuan tetap berada di bawah untuk menghindari terciptanya ruang persegi satuan, yang dapat menghambat pergerakan blok yang lebih besar. Dengan tiga blok persegi kecil di atas, blok vertikal 4 dan 1 dapat dipindahkan ke sisi kiri sehingga blok keledai dapat bergerak satu baris ke bawah, seperti yang ditunjukkan pada Langkah 48 (Gambar 6.9). Pengamatan terhadap kondisi saat ini menunjukkan bahwa blok keledai masih terkekang oleh blok vertikal di kedua sisi, dan perlu untuk memindahkannya ke atas agar blok keledai dapat bergerak bebas lebih jauh ke bawah.



Gambar 6.8 Langkah 41

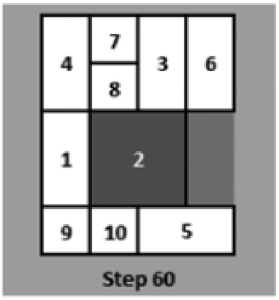


Gambar 6.9 Langkah 48

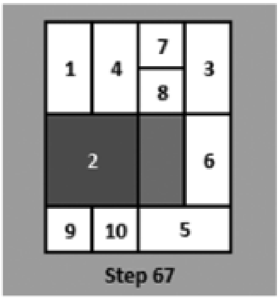


Gambar 6.10 Langkah 52

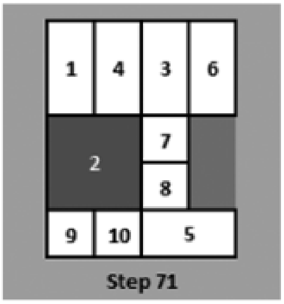
Langkah 48 hingga 71 berfokus pada pemindahan semua balok vertikal ke atas, yang merupakan subtujuan penting lainnya. Ini akan membersihkan jalur balok keledai menuju tujuannya di bawah. Setelah Langkah 48, balok kecil 8 dan 10 dapat diatur untuk menciptakan ruang vertikal, memungkinkan balok 6 bergerak ke atas. Kemudian balok keledai dapat dengan mudah meluncur ke kanan untuk memberi ruang bagi balok vertikal 1 dan 4 untuk bergerak ke atas, seperti yang terlihat pada Langkah 52 (Gambar 6.10). Pada titik ini, balok 10 dapat dipindahkan kembali ke bawah. Seperti yang terlihat pada Langkah 60 dan seterusnya (Gambar 6.11 hingga Gambar 6.15), balok 6 harus bergerak turun sementara untuk menurunkan balok persegi satuan 7 dan 8 sementara balok keledai meluncur ke kiri.



Gambar 6.11 Langkah 60

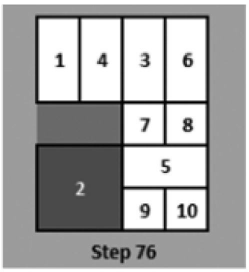


Gambar 6.12 Langkah 67

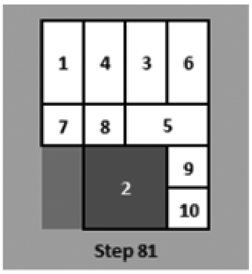


Gambar 6.13 Langkah 71

Dengan keempat balok persegi kecil yang berdekatan dengan balok 2 pada Langkah 71, balok keledai dapat meluncur ke baris paling bawah dengan menggeser balok 5, 9, dan 10. Langkah terakhir dicapai pada langkah ke-81, ketika balok keledai bergerak langsung ke tengah.



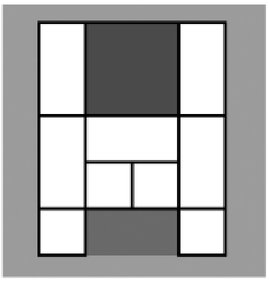
Gambar 6.14 Langkah 76



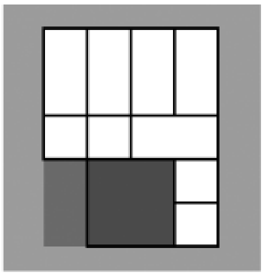
Gambar 6.15 Langkah 81

Pencarian Dua Arah

Jika kita mengetahui keadaan tujuan, pencarian dua arah akan membantu dengan mencari dari keadaan awal dan keadaan tujuan hingga bertemu di tengah jalan ketika jalur solusi tersedia.

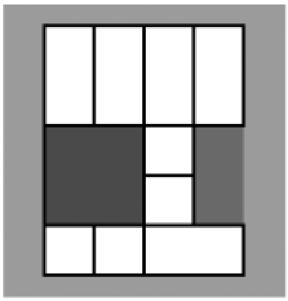


Gambar 6.16 Keadaan Awal



Gambar 6.17Keadaan Tujuan

Keadaan awal dan keadaan tujuan ditunjukkan masing-masing pada Gambar 6.16 dan Gambar 6.17. Pengamatan yang cermat terhadap keadaan tujuan menunjukkan bahwa tonggak penting saat bekerja mundur adalah ketika blok keledai dipindahkan ke kiri dan satu baris ke atas, sementara blok horizontal bergerak ke kanan bawah. Ini adalah keadaan yang direpresentasikan pada Langkah 71 dalam solusi sebelumnya.



Gambar 6.18 Subtujuan 1 Saat Mencari Mundur Dari Keadaan Tujuan

Kembali ke keadaan awal, kita dapat menganggap keadaan pada Gambar 6.18 sebagai keadaan tujuan antara dan mengupayakannya. Dengan demikian, subtujuan baru kita dalam pencarian maju adalah memindahkan blok keledai dan blok horizontal dua baris di bawah posisi mereka saat ini. Subtujuan kedua, jika dicapai secara optimal, dapat dilakukan dalam 21 langkah, seperti yang ditunjukkan pada Gambar 6.18. Dengan demikian, teknik pencarian analisis mean-ends dapat digunakan untuk mengurangi perbedaan antara keadaan saat ini dan keadaan tujuan. Kose (2012) memecahkan teka-teki Klotski menggunakan pencarian dua arah dalam 116 langkah.

6.3 PEMECAHAN MASALAH MANUSIA

Masalah ini terlalu rumit untuk diselesaikan dalam 20 menit, terutama tanpa alat simulasi seperti permainan sungguhan atau program yang dapat dimainkan. Oleh karena itu, beberapa solusi yang tersedia daring dipelajari, alih-alih solusi yang dikerjakan oleh sekelompok siswa. Berdasarkan pengalaman kami dengan masalah ini, teka-teki ini sangat sulit, dan tanpa strategi, seseorang akan terjebak di jalan buntu, yang mengakibatkan cukup banyak pengulangan.

Solusi yang dipelajari di Bagian 6.2 membagi masalah menjadi beberapa subtujuan:

- Memindahkan balok horizontal ke bawah untuk memberi ruang bagi balok keledai untuk bergerak ke tengah
- Memindahkan balok persegi kecil ke atas untuk membantu memindahkan balok vertikal keluar dari jalur balok keledai ke tengah
- Memindahkan semua balok vertikal ke atas setelah balok keledai berada di tengah

6.4 ANALISIS JENDELA MANUSIA TERHADAP SOLUSI

Meskipun Teka-Teki Keledai Merah pernah sangat populer, kami tidak dapat menemukan banyak solusi untuk analisis kami. Namun, solusi yang kami temukan sangat mirip satu sama lain.

Saat menganalisis Tabel 6.1, semua solusi diberi peringkat serupa di hampir semua bidang. Hal ini karena, seperti yang akan ditunjukkan oleh peringkat fleksibilitas, hanya ada beberapa cara untuk merepresentasikan Teka-Teki Keledai Merah. Hal ini dikarenakan teka-teki ini memiliki ukuran dan bentuk blok yang tetap, yang merupakan fitur utama yang harus dimiliki setiap representasi.

Dari Tabel 6.1, kita dapat melihat bahwa setiap solusi direpresentasikan secara ekstensional. Hal ini karena masing-masing menggunakan gambar teka-teki yang sebenarnya dan menunjukkan setiap pergerakan blok yang dilakukan untuk mencapai keadaan akhir. Untuk masalah sesulit Teka-Teki Keledai Merah, lebih baik menunjukkan teka-teki yang sebenarnya daripada mendeskripsikan solusi dengan kata-kata. Akibatnya, ekstensionalitas dan eksekusi setiap solusi relatif sama, dan masing-masing menggunakan teknik pemecahan masalah representasi visual yang sama.

Atribut dengan perbedaan terbesar dalam pemeringkatan adalah pemahaman. Meskipun setiap solusi menggunakan representasi yang sama, semuanya memiliki properti tambahan yang membedakannya satu sama lain, termasuk warna blok, penomoran, dan gambar. Oleh karena itu, pemahaman suatu solusi dapat menjadi faktor penentu apakah solusi tersebut harus dianggap MHWC atau tidak.

Kesulitan
9/10

kerumitan
$\mathbf{O(b^{d/2})}$

Nama	Int atau Ext?	Intens./Ekstens.	Repr.	Jendela Manusia?	Benar?	Ukuran Butir	Eksekusi	Keterpahaman	Metode Pemecahan Masalah	Fleksibilitas	Mode Konversi	Optimal?	Total
Donkey	Ekst.	6/10	Gbr	Y	Y	Ideal	9/10	9/10	Rep. Vis.; Subtujuan	7/10	Pembuat Gbr	Y	31/40
Gnome	Ekst.	7/10	Gbr	Y	Y	Ideal	9/10	5/10	Rep. Vis.; Subtujuan	5/10	Pembuat Gbr	Y	26/40
Hollow	Ekst.	5/10	Gbr	Y	Y	Ideal	8/10	4/10	Rep. Vis.; Subtujuan	5/10	Pembuat Gbr; Tangan	N	22/40
Klajok	Ekst.	7/10	Gbr	Y	Y	Ideal	9/10	6/10	Rep. Vis.; Subtujuan	5/10	Pembuat Gbr	Y	27/40
Neon	Ekst.	9/10	Gbr	Y	Y	Ideal	9/10	8/10	Rep. Vis.; Subtujuan	5/10	Pembuat Gbr	Y	31/40

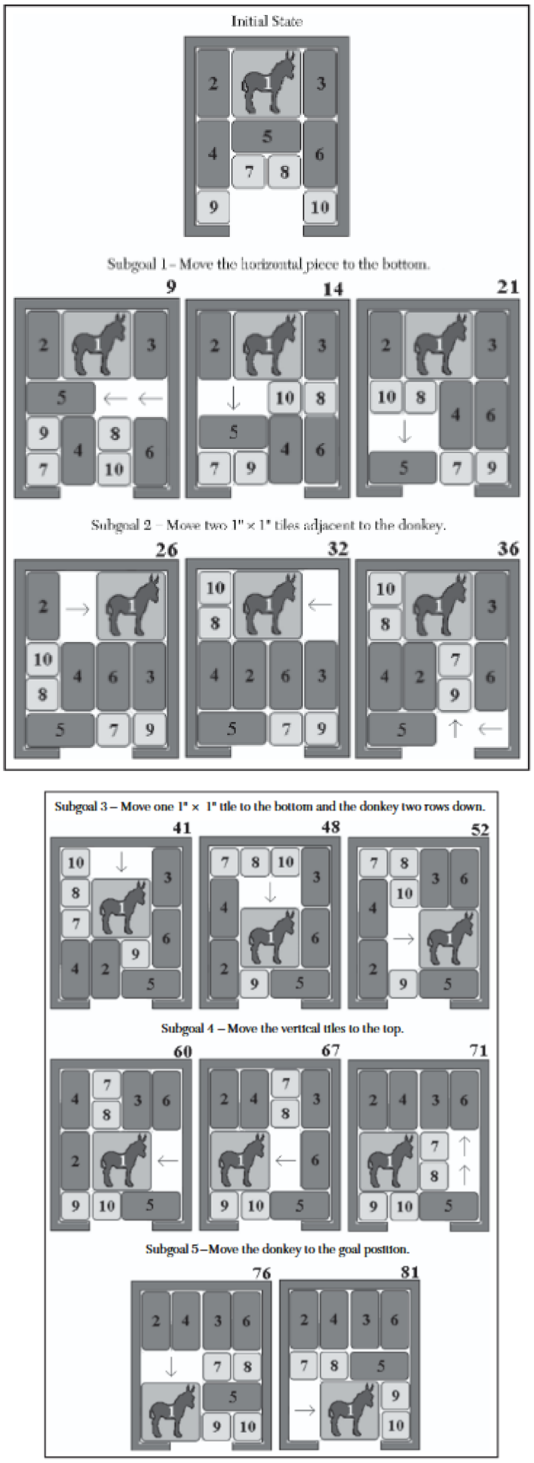
Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tabel 6.1 Peringkat Solusi Teka-Teki Keledai Merah Berdasarkan Jendela Manusia

Solusi yang Paling Kompatibel dengan Jendela Manusia

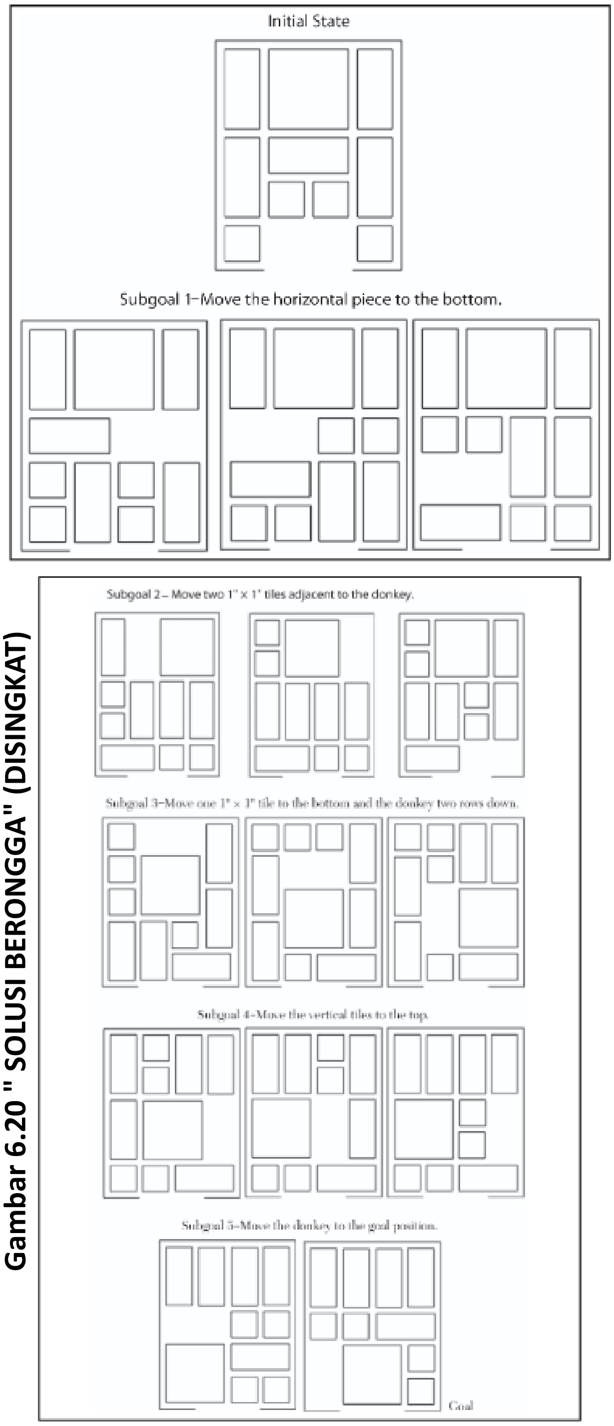
Solusi yang kami anggap sebagai MHWC adalah "Solusi Keledai" (Gambar 6.19). Contoh representasi jenis ini dapat ditemukan di. Alih-alih menampilkan masing-masing dari 81 langkah, solusi ini telah dipersingkat, hanya menampilkan subtujuan penting. Solusi ini memiliki banyak kualitas yang membuatnya sangat mudah dipahami. Setiap petak direpresentasikan dengan arsiran dan nomor yang berbeda, sehingga mudah dibedakan satu sama lain. Hal ini terutama berlaku untuk potongan keledai, yang memiliki gambar keledai asli. Selain itu, di atas setiap gambar dalam solusi terdapat nomor yang mewakili nomor langkahnya. Terakhir, kualitas yang sangat meningkatkan pemahaman secara keseluruhan adalah penyertaan tanda panah. Tanda panah menunjukkan petak mana yang dipindahkan dari keadaan sebelumnya untuk mencapai keadaan saat ini dan berapa banyak unit yang dipindahkan. Hal ini mencegah perlunya membandingkan gambar untuk dua langkah berturut-turut untuk menentukan petak mana yang telah dipindahkan. Gambar langkah saat ini dapat dilihat secara mudah, dan susunan langkah sebelumnya dapat ditentukan darinya.



Gambar 6.19 "Solusi Keledai" (Disingkat)

Solusi yang Paling Tidak Kompatibel dengan Jendela Manusia

Solusi yang dinilai sebagai LHWC adalah “Solusi Berongga” (Gambar 6.20). Saat menganalisis solusi ini, terlihat bahwa gambar yang mewakili teka-teki hanyalah garis hitam tipis yang disatukan sedemikian rupa untuk membentuk gambar teka-teki, termasuk ubin dan batasnya. Meskipun ubin dan batas diberi jarak yang cukup efisien untuk membedakan satu sama lain, fakta bahwa semua ubin terlihat hampir sama dapat membuatnya sulit dibedakan satu sama lain, yang mungkin mengurangi pemahaman solusi. Hal ini terutama berlaku ketika mencoba mengidentifikasi ubin yang dipindahkan dari langkah sebelumnya untuk mencapai susunan langkah saat ini. Selain itu, warna latar belakang sama dengan warna ubin. Ini menghasilkan efek ubin berongga dan dapat membuat gambar tampak seperti sekumpulan garis hitam tipis, sehingga membuat komponen teka-teki semakin tidak dapat dibedakan satu dengan yang lain.



Gambar 6.20 " SOLUSI BERONGGA" (DISINGKAT)

6.5 SOLUSI MESIN TERBAIK

Saat ini, tidak ada heuristik khusus yang dapat digunakan untuk memecahkan Teka-Teki Keledai Merah. Demikian pula, belum ada algoritma yang diketahui dapat memecahkan teka-teki ini dalam langkah sekecil mungkin. Namun, dua solusi paling terkenal untuk memecahkan teka-teki ini adalah pencarian dua arah [6] dan pencarian breadth-first sederhana.

Teka-Teki Keledai Merah memiliki ruang keadaan yang relatif besar, terdiri dari 25.955 kemungkinan susunan blok, beserta 47.151 kemungkinan langkah. Ketika menganalisis nilai-nilai ini dalam struktur pohon, kemungkinan susunan petak dapat dilihat sebagai simpul pohon, sedangkan pergerakan petak dapat dilihat sebagai busur. Karena Teka-Teki Keledai Merah dapat memiliki maksimal tiga kemungkinan langkah per susunan petak, faktor percabangan pohon yang dihasilkan oleh pencarian sangat kecil. Namun, mengingat banyaknya kemungkinan gerakan dan susunan ubin, kedalaman yang dihasilkan bisa menjadi sangat besar.

Dalam makalahnya "Solving Klotski," Karl Wiberg menunjukkan manfaat menggunakan pencarian breadth-first untuk memecahkan Teka-Teki Keledai Merah. Seperti implementasi pencarian breadth-first lainnya, metode ini menganalisis setiap simpul pada kedalaman k tertentu dari pohon sebelum menganalisis setiap simpul pada kedalaman $k + 1$ di pohon. Pencarian ini dapat menemukan solusi dalam 116 langkah.

Metode pencarian lain yang lebih canggih yang dapat digunakan untuk memecahkan Teka-Teki Keledai Merah adalah pencarian dua arah, yang dikembangkan oleh Erdal Kose. Menariknya, pencarian ini mampu menemukan solusi dalam 116 langkah, sama seperti pencarian breadth-first [6]. Namun, pencarian ini lebih canggih karena sangat mengurangi jumlah simpul yang perlu dikunjungi dan diperluas. Hal ini karena kedua ujung pencarian dimulai dengan simpul induk dan bertemu satu sama lain pada langkah tengah yang sama (lihat Bagian 6.2).

Karena solusi ditemukan secara efisien melalui penggunaan algoritma pencarian, kompleksitas Teka-Teki Keledai Merah dapat dievaluasi dalam bentuk pohon pencarian. Dalam kasus terburuk, waktu proses dan ruang yang dibutuhkan untuk pencarian breadth-first adalah $\mathrm{O}(\mathrm{b}^{\mathrm{d}})$, dengan b adalah faktor percabangan dan d adalah kedalaman pohon. Namun, untuk pencarian dua arah, waktu proses dan ruang yang dibutuhkan adalah $\mathbf{O}(\mathbf{b}^{\mathbf{d/2}})$. Hal ini menunjukkan bahwa pencarian dua arah lebih efisien daripada pencarian breadth-first, meskipun keduanya menemukan solusi dalam jumlah langkah yang sama.

6.6 MASALAH TERKAIT

Teka-teki yang sama dengan nama dan gambar yang berbeda pada setiap kepingnya tersedia di berbagai negara dengan nama yang berbeda pula. Namun, pada dasarnya ini adalah permainan yang sama. Di sisi lain, berbagai versi teka-teki balok geser di mana terdapat keping yang terjebak dan tugasnya adalah memindahkan balok ke area tertentu dalam teka-teki, terkait dengan masalah ini. Ada banyak versi berbeda dari jenis permainan ini yang ditawarkan di toko aplikasi di sebagian besar ponsel. Perlu dicatat bahwa keping-keping yang terjebak ini bervariasi ukurannya jika dibandingkan dengan Red Donkey Puzzle versi asli. Keping-keping ini dapat ditemukan di aplikasi yang dikembangkan untuk platform seluler.

6.7 PROGRAM YANG DAPAT DIMAINKAN

Banyak program yang dapat dimainkan untuk Red Donkey Puzzle tersedia daring, di mana Anda dapat memainkan teka-teki tersebut, membiasakan diri dengan masalahnya, dan belajar membedakan antara langkah yang baik dan yang buruk.

Akan sangat sulit untuk menghafal semua gerakan, jadi Anda harus menggambar semua gerakan di selembar kertas. Namun, menggambar pohon pencarian bisa sulit dan membosankan, serta membutuhkan banyak ruang. Oleh karena itu, memecahkan teka-teki di komputer lebih disukai. Teka-teki fisik, seperti yang terbuat dari kardus atau balok kayu, dan teka-teki virtual, seperti permainan di komputer atau ponsel pintar, hanyalah beberapa contoh.

Ini adalah bentuk representasi masalah yang paling ekstensional, karena pemecah masalah dapat memanipulasi keadaan masalah secara fisik atau virtual.

BAB 7
TEKA-TEKI 15

7.1 PENDAHULUAN

Teka-teki 15 diciptakan oleh Noyes Chapman, seorang kepala kantor pos yang tinggal di Canastota, New York. Ini adalah contoh lain dari teka-teki balok geser. Namun, tidak seperti Teka-teki Keledai Merah, yang terdiri dari ubin-ubin dengan ukuran berbeda, ubin-ubin Teka-teki 15 semuanya berbentuk persegi dan berukuran sama. Saat ini, teka-teki ini dikenal sebagai teka-teki geser paling awal yang dibuat.

Pada tahun 1874, Chapman mengembangkan teka-teki yang melibatkan "16 balok bernomor yang harus disusun dalam baris-baris yang terdiri dari empat balok." Tujuan dari teka-teki ini adalah untuk menyusun balok-balok ini sehingga angka-angka pada baris balok berjumlah 34. Pada tahun 1879, versi baru dan lebih baik dari Teka-teki 15 menyebar ke seluruh Amerika Serikat setelah putra Chapman memperkenalkan teka-teki tersebut [1]. Akhirnya, teka-teki ini populer di seluruh Amerika Serikat, dan pada tahun 1880, teka-teki ini telah memicu kegilaan teka-teki. Teka-teki ini akhirnya sampai ke Kanada dan kemudian ke Eropa. Namun, pada bulan Juli 1880, popularitasnya mulai memudar.

Pada suatu ketika, Chapman gagal mematenkan teka-teki ini. Kemungkinan besar, hal ini terjadi karena teka-teki ini terlalu mirip dengan "Puzzle-Blocks" karya Ernest Kinsey yang dipatenkan pada tahun 1878.

Penemuan The 15 Puzzle seringkali secara keliru dikaitkan dengan pemain catur, komposer catur, dan matematikawan rekreasi Amerika, Samuel Lloyd. Hingga kematiannya, Lloyd bersikeras bahwa ia telah menemukan teka-teki ini pada tahun 1891, dan pada suatu saat, ia menghidupkan kembali popularitasnya ketika ia memperkenalkan The 14-15 Puzzle, yang sangat mirip, kecuali dua petaknya yang ditukar (lihat Bagian 7.2). Namun, terlepas dari klaim-klaim ini, sebenarnya Chapman-lah yang menemukan The 15 Puzzle.

Teka-teki 15 terdiri dari 15 petak persegi bernomor 1 hingga 15, dibatasi dalam area tertutup berukuran $4” \times 4”$. Ke-15 petak tersebut mengisi 15 ruang di area tersebut, sementara ruang ke-16 dibiarkan kosong sehingga setiap petak memiliki ruang terbatas untuk bergerak. Tujuannya adalah untuk memposisikan ulang petak-petak tersebut agar ditempatkan dalam urutan numerik per baris.

7.2 TEKNIK PEMECAHAN MASALAH

Berbeda dengan soal-soal yang telah kita analisis sejauh ini, Teka-teki 15 tidak memiliki pengaturan awal yang telah ditentukan sebelumnya. Oleh karena itu, setiap susunan petak, termasuk susunan tujuan, dapat dianggap sebagai keadaan awal.

Terdapat $16!\ ^\circ(2{,}09 \times 10^{13})$ susunan tempat petak-petak tersebut dapat ditempatkan. Namun, menurut sebuah studi oleh Woosly Johnson dan William Story, hanya terdapat 16!/2 kemungkinan posisi di mana susunan tujuan benar-benar dapat dicapai. Bahasa Indonesia: Pada tahun 1879, Johnson dan Story meneliti The 15 Puzzle dan menemukan bahwa semua teka-teki dengan susunan paritas genap dapat dipecahkan, sedangkan yang berparitas ganjil tidak. Yaitu, jika jumlah pertukaran ubin (atau permutasi) yang diperlukan untuk mengubah susunan tertentu menjadi susunan yang dapat dipecahkan adalah genap, maka susunan tertentu itu sendiri dapat dipecahkan. Jika ganjil, itu tidak dapat dipecahkan. Lebih dari satu dekade setelah penemuan ini, Sam Lloyd menyajikan variasi masalah tersebut kepada publik yang melibatkan penyelesaian The 15 Puzzle di mana ubin bernomor 14 dan 15 ditukar. Ini disebut The 14-15 Puzzle. Karena ini terdiri dari satu pertukaran, itu adalah paritas ganjil, dan karena itu tidak dapat dipecahkan.

Meskipun jumlah posisi yang memungkinkan lebih rendah dari yang diharapkan, $16!/2(1{,}05 \times 10^{13})$ kemungkinan susunan ubin awal masih banyak! Namun, menemukan solusi untuk The 15 Puzzle bisa sangat mudah. Teknik pemecahan masalah yang paling berharga untuk digunakan adalah memecahkan subtujuan. Subtujuannya adalah untuk mencapai baris yang lengkap, dan setelah itu tercapai, seseorang dapat melanjutkan ke langkah berikutnya. Namun, seperti yang akan Anda lihat (di Bagian 7.3), ada situasi khusus, seperti memecahkan baris keempat, di mana baris ketiga yang telah lengkap harus *dicabut*.

Secara umum, The 15 Puzzle adalah jenis teka-teki n, dengan nilai n mewakili bilangan bulat lebih besar dari 1. Aturan dan tujuan dari setiap teka-teki n sama dengan The 15 Puzzle. Satu-satunya perbedaan adalah bahwa teka-teki ini terdiri dari n petak persegi bernomor 1 hingga n yang ditempatkan dalam area tertutup $sqrt(n\ +1)\ \times\ sqrt(n\ +\ 1)$, dengan lokasi ke-n dibiarkan sebagai ruang petak kosong. Namun, seberapa pun besar masalahnya, teknik pemecahan masalah yang sama dapat digunakan. Mungkin hanya dibutuhkan lebih banyak langkah untuk menyelesaikannya. Dalam buku ini, kami melakukan analisis hanya pada Teka-Teki 15.

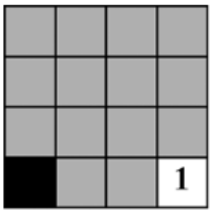
7.3 SOLUSI TEKA-TEKI 15

Solusi yang sangat umum untuk Teka-Teki 15 melibatkan pemfokusan dan penyelesaian petak individual satu per satu dalam urutan numerik. Melakukan hal ini tidak hanya akan menyelesaikan teka-teki secara teratur, tetapi juga memungkinkan penyelesaian subtujuan, seperti yang disebutkan di Bagian 7.2. Secara spesifik, subtujuan adalah penyelesaian satu baris. Dengan pengecualian langkah terakhir, setelah satu baris diselesaikan, tidak ada petak di baris tersebut yang perlu dipindahkan lagi.

Karena terdapat begitu banyak kemungkinan susunan ubin awal, solusi ini akan menunjukkan cara umum untuk menyelesaikan setiap ubin dan oleh karena itu tidak akan didasarkan pada susunan awal.

Menyelesaikan Baris Pertama

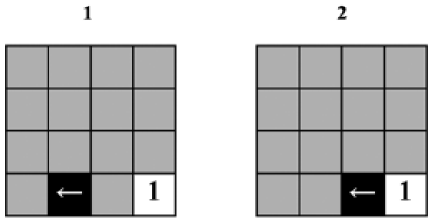
Langkah pertama adalah memindahkan ubin 1 ke pojok kiri atas papan. Langkah ini relatif mudah dilakukan. Karena ubin-ubin lainnya belum diposisikan dengan benar, cara pemindahannya tidak menjadi masalah selama ubin 1 ditempatkan pada posisi yang benar. Misalkan teka-teki disusun seperti pada Gambar 7.1.



Gambar 7.1 Keadaan Awal

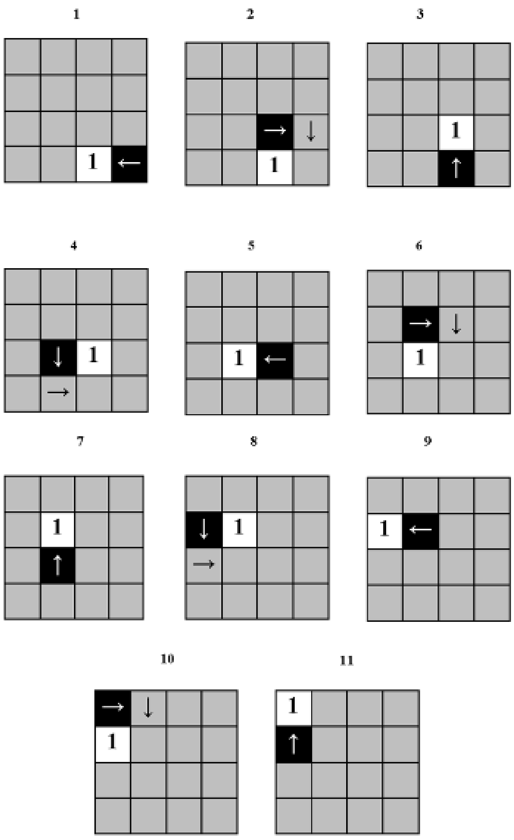
Kotak hitam mewakili ruang ubin kosong, dan kotak abu-abu mewakili ubin yang tidak akan kita bahas pada tahap ini.

Pertama, susun ubin-ubin tersebut sehingga ruang kosong berada di dekat ubin 1 (Gambar 7.1).



Gambar 7.2 Pindahkan Ruang Kosong Ke Ubin 1

Sekarang, pindahkan ubin 1 ke pojok kiri atas (Gambar 7.3).

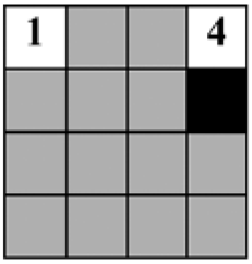


Gambar 7.3 Pindahkan Ubin 1 Ke Sudut Kiri Atas

Pada dasarnya, ide umum di balik pemindahan ubin target adalah memindahkan semua ubin yang diabaikan sementara di sekitarnya hingga ruang kosong berada di arah umum posisi yang tepat relatif terhadap ubin target. Kemudian, gunakan ruang kosong tersebut untuk memindahkan ubin target ke posisi tujuannya. Lanjutkan langkah ini hingga ubin target berada di posisi yang tepat.

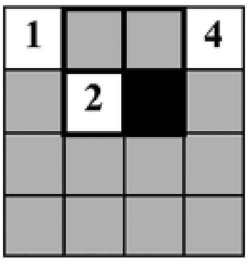
Gambar 7.3 menunjukkan ide umum di balik pemindahan ubin di seluruh papan. Mulai sekarang, diagram selanjutnya hanya akan menunjukkan gerakan yang signifikan untuk memposisikan ubin tertentu. Untuk menyegarkan kembali cara memindahkan ubin di sekitar papan, lihat kembali Gambar 7.3.

Selanjutnya, kita memindahkan ubin 4 ke sudut kanan atas papan puzzle. Sama seperti yang kita lakukan dengan ubin 1, ubin ini dapat dengan mudah diposisikan hanya dengan memindahkan ubin yang diabaikan sementara. Posisikan seperti yang ditunjukkan pada Gambar 7.4.



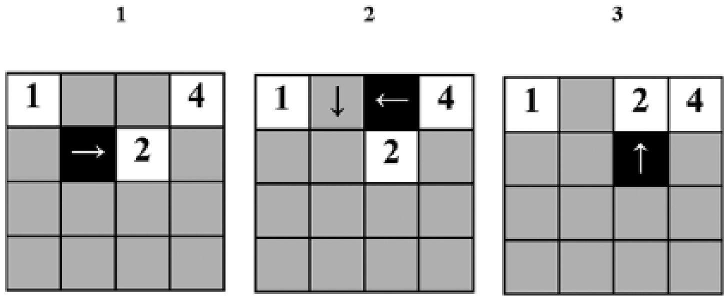
Gambar 7.4 Posisi Ubin 4

Sekarang kita posisikan ubin 2 di sebelah kanan ubin 4. Meskipun posisi ubin 4 dan 2 sangat berdekatan, ubin 4 tidak perlu digeser dari posisi aslinya untuk menempatkan ubin 2 di sebelahnya. Hal ini dapat dilakukan dengan memindahkan ubin 2 dan ruang kosong ke posisi mana pun di "kotak" tengah yang terletak di antara ubin 1 dan 4. Misalnya, susunan ubin yang memadai akan seperti yang ditunjukkan pada Gambar 7.5.



Gambar 7.5 "Kotak" Tengah

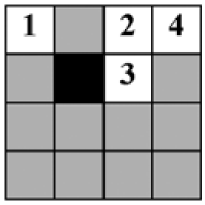
"Kotak" tengah yang kita maksud ditunjukkan dengan batas yang dicetak tebal. Sekarang, lakukan gerakan seperti pada Gambar 7.6.



Gambar 7.6 Memposisikan Ubin 2

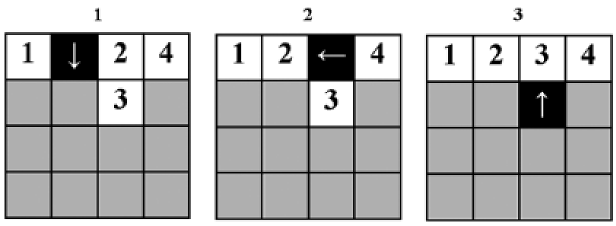
Pada titik ini, Anda mungkin menyadari bahwa kita belum menempatkan ubin 2 pada posisi yang benar. Namun, menempatkannya sementara di sini akan memudahkan kita untuk memposisikan ubin 3.

Sekarang, atur ubin 3 dan slot kosong sehingga berada pada posisi yang ditunjukkan pada Gambar 7.7.



Gambar 7.7 3 Posisi Pilihan Ubin

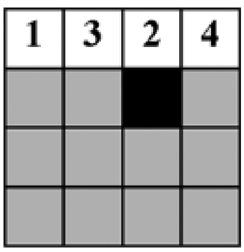
Ini seharusnya sangat mudah, karena ubin 1, 2, atau 4 tidak perlu dipindahkan untuk mencapai kondisi ini. Sekarang, lakukan pemindahan seperti pada Gambar 7.8.



Gambar 7.8 Memposisikan Ubin Ke-3

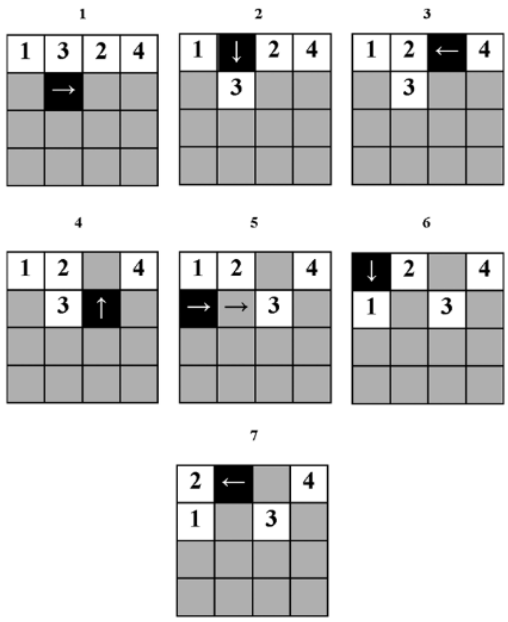
Teknik Sudut

Setelah memposisikan ubin ke-2, baris pertama mungkin terlihat seperti Gambar 7.9.



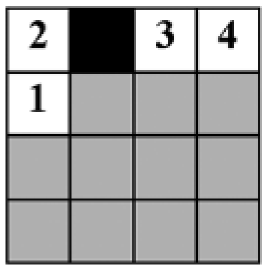
Gambar 7.9 3 Kasus Khusus Ubin

Dalam pengaturan ini, teknik yang digunakan pada Gambar 7.8 tidak dapat digunakan. Sebagai gantinya, pastikan tempat kosong berada di bawah ubin ke-2 dan lakukan gerakan pada Gambar 7.10.



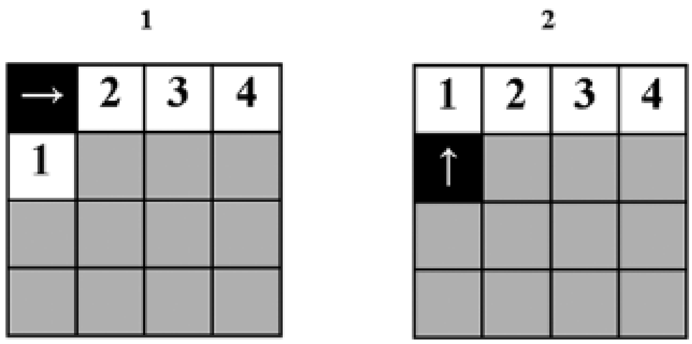
Gambar 7.10 Memposisikan 3 Ubin (Kasus Khusus)

Sekarang putar ubin-ubin di dalam "kotak" tersebut sehingga tersusun seperti pada Gambar 7.11.



Gambar 7.11 Ubin 1 Dan 2 Di Sudut

Kemudian, posisikan kembali ubin 1 dan 2 pada tempatnya, seperti pada Gambar 7.12.



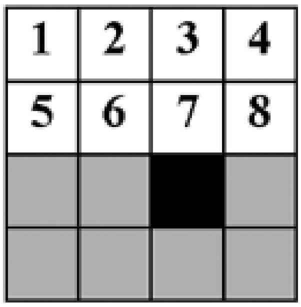
Gambar 7.12 Memposisikan Ulang Ubin 1 Dan 2

Kita sekarang telah menyelesaikan apa yang kita sebut teknik sudut. Dinamakan demikian karena kita telah memindahkan ubin 1 dan 2 melewati sudut papan untuk memberi ruang bagi ubin 3 agar dapat bergerak ke posisinya. Teknik ini akan digunakan lagi untuk menyelesaikan baris ke-4.

Jadi sekarang kita telah menyelesaikan baris pertama. Selanjutnya, tidak ada ubin di baris teratas yang perlu dipindahkan lagi dan akan tetap di tempatnya selama sisa penyelesaian.

Menyelesaikan Baris Kedua

Tujuan selanjutnya adalah menyelesaikan baris kedua. Untuk melakukannya, ikuti teknik yang sama yang digunakan untuk menyelesaikan baris pertama. Pindahkan ubin 5 ke kiri atas, ubin 8 ke kanan atas, ubin 6 ke kiri ubin 8, dan gunakan teknik yang sama yang digunakan untuk menyelesaikan ubin 3 untuk menyelesaikan ubin 7. Jika dikerjakan dengan benar, teka-teki akan terlihat seperti Gambar 7.13.

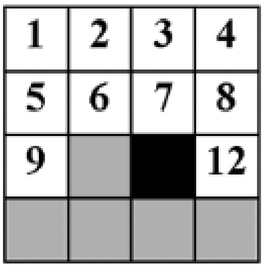


Gambar 7.13 Baris 2 Terselesaikan

Pada titik ini, ubin di baris kedua tidak perlu diposisikan ulang. Kita sudah setengah jalan, menyisakan dua baris lagi untuk dipecahkan. Seperti yang Anda lihat, ruang yang tersedia semakin berkurang, dan semakin sedikit ruang untuk memindahkan ubin. Menempatkan tujuh ubin terakhir akan sedikit lebih sulit.

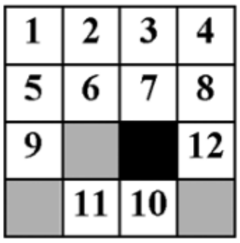
Menyelesaikan Baris Ketiga

Selanjutnya, tujuan selanjutnya adalah menyelesaikan baris ketiga. Pertama, pindahkan ubin ke-9 ke posisi paling kiri, lalu pindahkan ubin ke-12 ke posisi paling kanan, seperti pada Gambar 7.14.



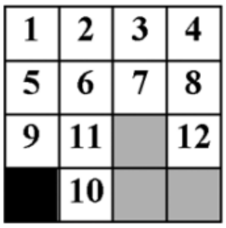
Gambar 7.14 Posisikan Ubin 9 Dan 12

Selanjutnya, kita posisikan ubin 10 dan 11 secara bersamaan. Untuk melakukan ini, penting agar ubin 10 dan 11 serta slot kosong ditempatkan di kotak tengah. Gambar 7.15 menunjukkan kemungkinan pengaturan.



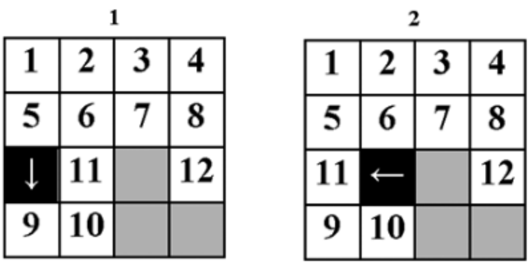
Gambar 7.15 "Kotak" Tengah Yang Dapat Diterima

Jika demikian, cukup memutar ubin di "kotak" dua kali akan melengkapi baris ketiga. Namun, hal ini tentu tidak selalu terjadi. Misalnya, Anda mungkin memiliki posisi seperti pada Gambar 7.16.



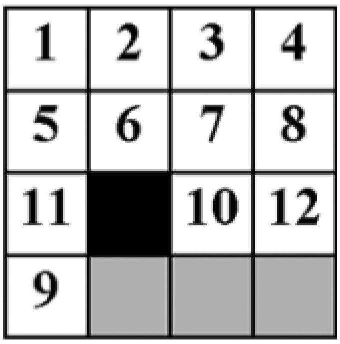
Gambar 7.16 Posisi Ubin Yang Buruk

Perhatikan bagaimana kita memiliki ruang kosong di bawah ubin nomor 9. Hal ini dapat diterima jika salah satu ubin yang diabaikan sementara berada di sebelah kanannya, tetapi sayangnya hal itu tidak terjadi. Dengan pengaturan ini, mustahil untuk menyelesaikan teka-teki tanpa memindahkan ubin nomor 9. Ikuti langkah-langkah pada Gambar 7.17.



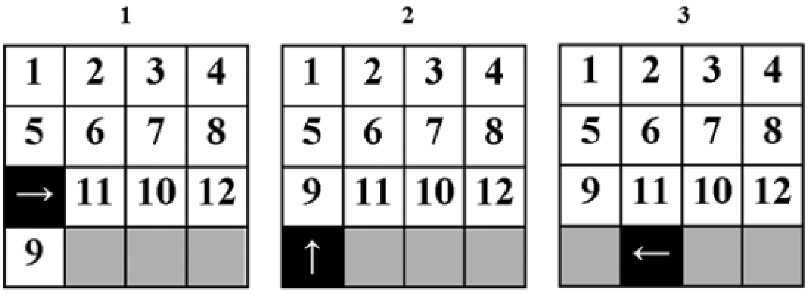
Gambar 7.17 Memindahkan Ubin 9

Sekarang, putar ubin di kotak tengah sehingga salah satu ubin yang diabaikan sementara berada di sebelah ubin 9 dan slot kosong berada di atasnya, seperti yang ditunjukkan pada Gambar 7.18.



Gambar 7.18 Putar "Kotak" Tengah

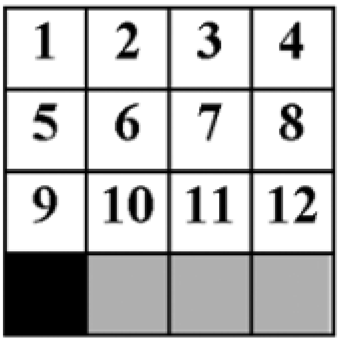
Lalu lakukan gerakan pada Gambar 7.19.



Gambar 7.19 Mendapatkan "Kotak" Tengah Yang Memadai

Sekarang kita telah mendapatkan kebutuhan kita. Demikian pula, jika slot kosong dimulai di bawah petak 12, lakukan refleksi langkah-langkah di atas pada sisi kanan puzzle. Pada dasarnya, sangat penting bahwa petak 9 dan 12 untuk sementara waktu mengabaikan petak di bawahnya untuk melengkapi baris ketiga.

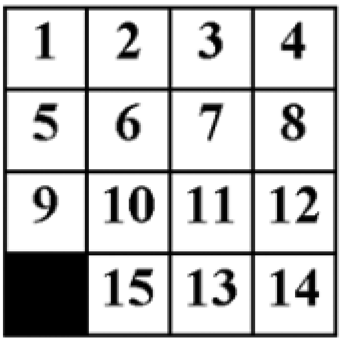
Seperti yang Anda lihat, mungkin juga petak 10 dan 11 ditempatkan secara tidak nyaman dan saling menempati posisi target. Apakah ini terlihat familier? Cukup gunakan "teknik sudut" untuk memposisikannya dengan tepat. Jika dilakukan dengan benar, baris ketiga akan lengkap, seperti yang ditunjukkan pada Gambar 7.20.



Gambar 7.20 Baris Ketiga Selesai

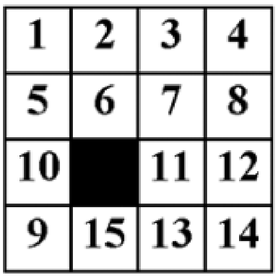
Menyelesaikan Baris Keempat

Kita hanya perlu menyelesaikan satu baris. Pada titik ini, teka-teki terpecahkan atau posisinya salah, seperti pada Gambar 7.21.



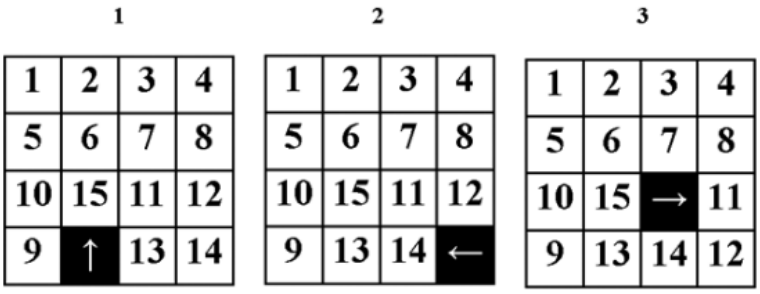
Gambar 7.21 Baris Terakhir Yang Salah

Misalkan ubin di baris terakhir diposisikan secara salah. Hingga titik ini, kita telah mengikuti strategi dasar yang mencegah perlunya memindahkan ubin yang termasuk dalam baris yang sudah selesai. Untuk melanjutkan menemukan solusi, aturan tersebut harus dilanggar. Untuk menangani pengaturan ini, kita mulai dengan sekali lagi menggunakan teknik sudut. Pertama, pindahkan ubin 9 dan 10 agar sejajar vertikal. Ini melibatkan pemindahan ubin 9 ke baris terakhir dan pemindahan ubin 10 ke posisi di mana ubin 9 seharusnya berada (Gambar 7.22).



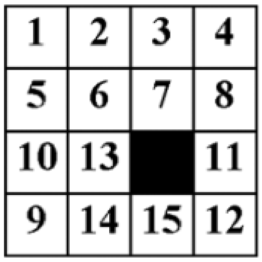
Gambar 7.22 Ubin 9 Dan 10 Yang Digeser

Kemudian lakukan hal yang sama pada ubin 11 dan 12 (Gambar 7.23).



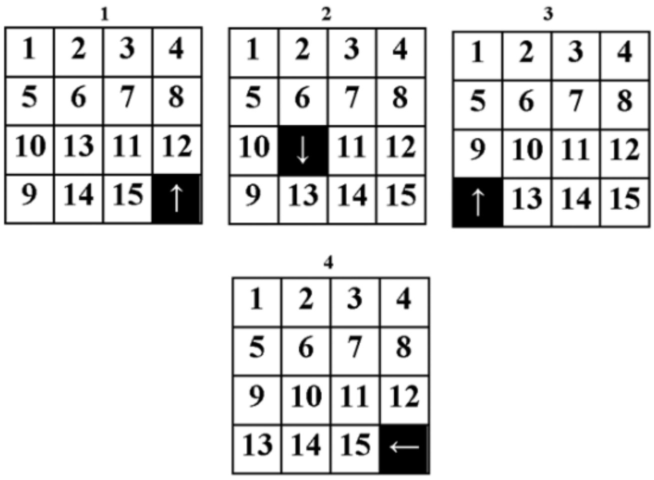
Gambar 7.23 Menggeser Ubin 11 Dan 12

Pada titik ini, ubin di "kotak" tengah seharusnya hanya ubin yang berada di baris terakhir (yaitu, 13, 14, dan 15). Susun ubin-ubin tersebut seperti pada Gambar 7.24.



Gambar 7.24 Putar Ubin "Persegi" Tengah

Terakhir, lakukan gerakan pada Gambar 7.25.



Gambar 7.25 Menyelesaikan Baris Terakhir

Akhirnya, teka-teki terpecahkan!

7.4 ANALISIS JENDELA MANUSIA DARI SOLUSI

Seperti halnya Teka-teki Keledai Merah, kami belum menemukan banyak solusi untuk analisis Jendela Manusia kami untuk Teka-teki 15. Selain itu, sebagian besar solusi identik dengan solusi yang dibahas di Bagian 7.3 dan satu sama lain. Oleh karena itu, masing-masing solusi menggunakan metode dan representasi pemecahan masalah yang serupa. Setiap solusi menunjukkan metode umum yang menggunakan deskripsi, beserta gambar visual sebagai panduan, untuk menyelesaikan Teka-teki 15, terlepas dari susunan awalnya. Oleh karena itu, solusi-solusi tersebut dapat dianggap intensional karena sifat umum ini. Namun, intensionalitasnya relatif rendah karena sebagian bersifat ekstensional (yaitu, diagram).

Saat menganalisis peringkat solusi untuk pemahaman dan eksekusi (Tabel 7.1), kita dapat melihat bahwa solusi-solusi tersebut umumnya beragam. Oleh karena itu, fitur-fitur yang memengaruhi kategori peringkat inilah yang menentukan solusi mana yang dianggap sebagai MHWC.

Satu-satunya solusi intensional yang menggunakan teknik yang agak berbeda dari yang lain adalah "Solusi In-Place". Sub-tujuan yang diperlukan untuk menyelesaikannya sama, kecuali bahwa solusi ini menjamin bahwa tidak ada ubin yang mencapai posisi yang benar akan bergerak lagi.

Terakhir, satu-satunya solusi yang diklasifikasikan sebagai ekstensional adalah "Solusi Optimal".

Kesulitan
4/10

kerumitan
$O(b^{d/2})$

Nama	Int atau Ext?	Intens./ Ekstens.	Repr.	Jendela Manusia?	Benar?	Ukuran Butir	Eksekusi	Keterpahaman	Metode Pemecahan Masalah	Fleksibilitas	Mode Konversi	Optimal?	Total
Di tempat	Int.	3/10	Desk. dng Vis.	Y	Y dng siklus	Kecil	5/10	7/10	Sub-tujuan	7/10	ET; PG; Tangan	N	22/40
Jurgen I	Int.	4/10	Desk. dng Vis.	Y	Y	Ideal	4/10	6/10	Rep. Vis.; Subtujuan	8/10	ET; PG; Tangan	N	22/40
Optimal atau Terbaik	Ekst.	3/10	Desk. dng Vis.	Y	Y	Ideal	9/10	5/10	Repr. Visual	9/10	ET; Tangan	Y	26/40
Nyata	Int.	9/10	Desk. dng Vis.	Y	Y	Ideal	9/10	7/10	Rep. Vis.; Subtujuan	7/10	ET; PG; Tangan	N	32/40
Ubin	Int.	5/10	Desk. dng Vis.	Y	Y	Ideal	10/10	9/10	Rep. Vis.; Subtujuan	7/10	ET; PG; Tangan	N	33/40
Bagaimana-W	Int.	5/10	Desk. dng Vis.	Y	Y	Ideal	6/10	8/10	Rep. Vis.; Subtujuan	7/10	ET; PG	Y	26/40

Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tabel 7.1 Peringkat 15 Solusi Teka-Teki Berdasarkan Jendela Manusia

Solusi yang Paling Sesuai dengan Jendela Manusia

Solusi yang dipertimbangkan dalam MHWC adalah "Solusi Ubin", yang sama dengan yang ada di Bagian 7.3. Solusi ini menyediakan metode langkah demi langkah tentang cara menyelesaikan 15 Teka-Teki dengan susunan awal apa pun. Pada setiap langkah, deskripsi sederhana diberikan tentang ubin mana yang harus dipindahkan dan bagaimana melakukannya. Di setiap langkah, gambar teka-teki itu sendiri disediakan, yang berisi ubin yang diarsir atau diberi nomor. Ubin yang diarsir tidak penting pada langkah tertentu dan dapat diabaikan, sedangkan ubin bernomor penting atau telah ditempatkan dengan benar. Gambar-gambar ini digunakan untuk menunjukkan seperti apa tampilan teka-teki setelah setiap langkah dieksekusi. Terakhir, panah digunakan untuk menunjukkan bagaimana ubin tertentu dipindahkan pada suatu langkah. Petunjuk singkat namun tepat sasaran ini, beserta gambarnya, membuat solusinya sangat mudah dipahami dan dieksekusi.

Untuk menyempurnakan solusi ini lebih jauh, gambar asli teka-teki dapat digunakan, alih-alih kisi atau tabel. Ini akan membuat solusinya lebih ekstensional. Solusi yang menggunakan pendekatan ini adalah "Solusi Teka-Teki Nyata".

Solusi yang Paling Tidak Kompatibel dengan Jendela Manusia

Solusi yang dinilai sebagai LHWC adalah "Solusi Optimal" (Tabel 7.2). Representasi serupa dikembangkan oleh B. MacKenzie. Disebut demikian karena menunjukkan cara menyelesaikan Teka-Teki 15 dengan langkah sesedikit mungkin. Solusi optimal mungkin layak untuk mesin dengan algoritma yang cerdas atau presisi, tetapi bagi manusia, menemukan solusi optimal untuk masalah yang tidak sepele biasanya sangat sulit, bahkan mustahil.

Menemukan solusi optimal untuk Teka-Teki 15 bergantung pada susunan awal petak. Dari semua 16!/2 kemungkinan susunan petak awal, hanya 16 yang dapat diselesaikan dalam 43 langkah dengan pergerakan beberapa petak, yang dianggap optimal. Jika tidak, jumlah minimal gerakan ubin tunggal yang diperlukan untuk memecahkan teka-teki dapat rata-rata sekitar 80 langkah. Namun, seperti yang telah kita lihat, solusi optimal umumnya tidak banyak berhubungan dengan Jendela Manusia. Meskipun solusi ini menunjukkan metode yang sangat efisien dalam memecahkan Teka-teki 15 dalam jumlah langkah paling sedikit, kita hanya akan menganalisis representasi solusinya.

Bagian ini menganalisis 1 dari 16 kemungkinan susunan yang dapat dipecahkan dalam 43 langkah, yang optimal untuk Teka-teki 15. Solusi ini dapat dianggap sebagai LHWC karena berbagai alasan. Salah satunya adalah karena representasinya. Semua yang ada dalam solusi ini berupa teks, termasuk teka-teki itu sendiri. Teka-teki tersebut direpresentasikan sebagai kumpulan angka yang menunjukkan ubin. Ubin-ubin tersebut disusun sedemikian rupa untuk mewakili teka-teki yang sebenarnya (yaitu, ada empat baris dan empat angka dalam satu baris). Namun, ini mungkin tidak dapat dipahami seperti gambar sebenarnya dari teka-teki itu sendiri.

Alasan lain mengapa ini bisa disebut LHWC adalah karena cara langkah-langkahnya dilakukan. Setiap langkah terdiri dari huruf dan angka. Huruf tersebut mewakili arah di mana ubin harus dipindahkan relatif terhadap ruang kosong, dan angka tersebut mewakili jumlah ubin yang harus bergerak ke arah itu. Misalnya, "U2" berarti dua ubin pertama yang berada di bawah ruang kosong saat ini harus bergerak ke atas ke ruang itu. Representasi ini bisa membingungkan, terutama karena tidak ada representasi teka-teki yang sesuai dengan setiap langkah. Akibatnya, seseorang kemungkinan besar tidak akan dapat menentukan langkah mana yang harus dilacak kembali jika ia membuat kesalahan dan harus memulai dari awal.

Terakhir, ini dapat dianggap LHWC karena solusi ini bersifat ekstensional. Untuk solusi ini, semua langkah yang memecahkan teka-teki ini dari awal hingga akhir ditampilkan, sehingga diklasifikasikan sebagai ekstensional. Meskipun solusinya sangat dapat dieksekusi (karena menyediakan langkah-langkah untuk memecahkan seluruh susunan teka-teki), ini adalah solusi untuk hanya satu susunan ubin tertentu. Jika seseorang mencoba memecahkan Teka-teki 15 dengan susunan yang berbeda, ia mungkin tidak memiliki pengetahuan yang dibutuhkan untuk melakukannya.

Puzzle Solution

1	5	9	13
2	6	10	14
3	7	11	15
4	8	12	X

X = vacant position (empty tile space)

Moves (direction of X):

1. L1	11. R1	21. R3	31. R2	41. R2
2. U1	12. U1	22. D1	32. U1	42. D2
3. R1	13. R2	23. L3	33. R1	43. R1
4. U2	14. U1	24. D1	34. D2	
5. L3	15. L3	25. R2	35. L3	
6. D3	16. D1	26. U2	36. U1	
7. R2	17. R3	27. L1	37. R3	
8. U2	18. U1	28. D3	38. D2	
9. L2	19. L3	29. L1	39. L3	
10. D2	20. U1	30. U2	40. U2	

Key:
L = left
R = right
U = up
D = down

Tabel 7.2“Solusi Optimal”

7.5 SOLUSI MESIN TERBAIK

Dalam bidang Kecerdasan Buatan, solusi untuk Teka-teki 15, dan teka-teki n secara umum, biasanya dimodelkan dan dipelajari dengan penerapan algoritma pencarian yang dibantu oleh heuristik. Meskipun pencarian sederhana seperti pencarian breadth-first dapat menemukan solusi optimal untuk Teka-teki 15, jumlah ruang dan waktu yang dibutuhkan untuk mencari setiap kemungkinan susunan sangatlah besar, bahkan dengan pencarian dua arah. Oleh karena itu, sebaiknya gunakan pencarian yang menggunakan heuristik sebagai panduan dan untuk membatasi pilihan yang buruk.

Heuristik memberikan algoritma pencarian perkiraan di mana kemungkinan tujuannya, sehingga sangat mengurangi jumlah posisi teka-teki yang perlu dicari. Pencarian yang paling umum digunakan untuk menemukan solusi optimal untuk Teka-teki 15 adalah **pencarian $\mathbf{A}$°** yang dilengkapi dengan **heuristik yang dapat diterima**. Pencarian A° menggabungkan **pencarian biaya seragam**, yang mengukur jumlah langkah yang diperlukan untuk mencapai susunan teka-teki tertentu dari susunan awal, dengan heuristik yang dapat diterima, yang mencegah pencarian melebih-lebihkan tujuan. Kedua sifat ini, jika praktis beroperasi, memungkinkan pencarian A° untuk menemukan solusi optimal.

Pencarian A° dapat menggunakan beberapa heuristik yang dapat diterima untuk memecahkan Teka-teki ke-15. Salah satunya adalah **heuristik terbaik-pertama**, yang mempertahankan antrean simpul terbuka dan mengejar simpul di depan antrean terlebih dahulu tetapi menyimpan dalam memori simpul lain pada antrean untuk mengejar jalur lain yang mungkin. Yang lainnya adalah **heuristik jarak Manhattan**, yang menggunakan geometri taksi untuk mengukur seberapa jauh petak tertentu dari posisi tujuannya. Namun, heuristik ini lambat dan mungkin masih menghasilkan terlalu banyak status, menghabiskan banyak memori. Salah satu solusi untuk masalah ini adalah penggunaan **pencarian A° yang memperdalam secara iteratif**. Pencarian ini mengurangi jumlah simpul yang biasanya dicari oleh pencarian A°, sehingga mengurangi kebutuhan ruang. Namun, pencarian ini masih lambat.

Heuristik lain, dan yang lebih baik, yang dapat digunakan adalah **heuristik basis data pola terpisah**. Heuristik ini bekerja dengan menemukan jumlah "jumlah gerakan yang diperlukan untuk menyusun petak dengan baik" dalam pola struktur data tertentu. Struktur ini adalah antrean minimum prioritas yang dikombinasikan dengan tabel hash. Dengan heuristik ini, kurang dari 100.000 simpul diperluas, dan setiap susunan 15 Puzzle dapat diselesaikan dalam waktu kurang dari 10 detik.

Seperti The Red Donkey Puzzle, waktu proses dan ruang terburuk yang diperlukan untuk menyelesaikan setiap susunan puzzle berada dalam orde $\mathbf{O}\left(\mathbf{b}^{\mathbf{d}/2}\right)$ saat menggunakan pencarian dua arah. Mengingat jumlah petak yang dapat bergerak terbatas, faktor percabangannya kecil. Namun, mengingat banyaknya kemungkinan susunan, kedalamannya bisa menjadi sangat besar. Untungnya, karena pencarian A* dengan heuristik basis data pola terpisah sangat mengurangi jumlah simpul yang diperluas, solusi dapat ditemukan dengan kompleksitas yang lebih rendah daripada $\mathrm{O}\left(\mathrm{b}^{\mathrm{d}/2}\right)$.

Untuk teka-teki n umum, menemukan solusi untuk susunan ubin yang sah dapat dilakukan dalam waktu polinomial. Namun, masalah untuk selalu menemukan solusi optimal untuk susunan tertentu adalah **NP-Hard**.

7.6 MASALAH TERKAIT

Beberapa soal yang berkaitan dengan Teka-Teki Lima Belas memerlukan urutan angka 1-15 yang berbeda, seperti mengurutkan berdasarkan kolom atau membalik urutannya. Namun, solusi dasarnya tidak berubah dalam situasi ini, yaitu menempatkan angka 1 di tempat yang benar terlebih dahulu, diikuti oleh angka dua dan angka-angka selanjutnya. Beberapa soal serupa lainnya memiliki ukuran yang berbeda dari teka-teki 15, seperti Teka-Teki Delapan pada papan 3x3. Soal-soal ini dapat ditemukan di aplikasi yang dikembangkan untuk platform seluler karena termasuk dalam genre teka-teki klasik yang sangat populer.

Salah satu penggunaan populer Teka-Teki Lima Belas adalah untuk menguji efisiensi algoritma pencarian heuristik. Teka-teki ini telah terbukti menjadi tolok ukur standar yang baik untuk mengevaluasi algoritma pencarian heuristik karena sifat-sifatnya yang dipahami dengan baik. Untuk menyelesaikan Teka-Teki Lima Belas, jarak petak dari keadaan awal ke keadaan yang diinginkan dihitung. Jarak ini disebut heuristik Jarak Manhattan. Jarak dihitung berdasarkan jumlah langkah yang dibutuhkan oleh sebuah petak untuk bergerak secara horizontal dan vertikal, untuk mencapai tujuannya. Algoritma pencarian heuristik menghitung angka ini dan memilih langkah dengan jumlah langkah terkecil yang dibutuhkan untuk setiap petak untuk mencapai keadaan yang diinginkan. Satu set Lima Belas Teka-teki yang disebut "Korf 100 Set" digunakan untuk menguji algoritma karena merupakan set pertama yang digunakan untuk tujuan demonstrasi. Ini telah digunakan berulang kali dalam pengujian serupa berikutnya yang dilakukan oleh peneliti algoritma lainnya.

Algoritma heuristik tercepat yang ditulis dalam C++ dapat menyelesaikan semua Lima Belas Teka-teki dalam set tersebut dalam waktu sekitar 540 detik dan membutuhkan total memori sebesar 27GB. Karena Lima Belas Teka-teki dapat dengan mudah dipahami oleh semua orang, ini adalah alat yang sangat efektif untuk digunakan untuk menguji algoritma heuristik.

7.7 PROGRAM YANG DAPAT DIMAINKAN

Game Puzzle 15 yang dapat dimainkan tersedia di $http://migo.sixbit.org/puzzles/fifteen/$. Dengan satu klik, pemain dapat memindahkan ubin ke ruang kosong. Selain itu, pemain juga dapat melakukan gerakan multi-ubin. Terakhir, terdapat opsi untuk mencoba Puzzle 14-15 jika Anda mau. Semoga berhasil!

BAB 8
PERMASALAHAN TUR KSATRIA KUDA

8.1 PENDAHULUAN

Tur kuda adalah serangkaian langkah yang dilakukan oleh kuda catur, dimulai dari petak awal mana pun di papan catur (biasanya di pojok kiri bawah), sehingga setiap petak di papan dikunjungi tepat satu kali. Dalam hal ini, perlu dipertimbangkan bahwa kuda dibatasi pada langkah yang biasanya dilakukannya di papan catur (yaitu, bentuk "L").

Permasalahan Tur Kuda melibatkan pencarian urutan langkah yang tepat bagi kuda untuk menyelesaikan tur kuda. Permasalahan ini biasanya diselesaikan pada petak standar berukuran $8” \times 8”$, mirip dengan papan catur standar. Namun, tur kuda telah dipelajari pada petak umum berukuran $m \times n$.

Konsep Permasalahan Tur Kuda berasal dari sekitar tahun 2200 SM, tetapi referensi paling awal yang diketahui adalah sekitar tahun 900 M di India. Ini pertama kali diperkenalkan dalam sebuah karya Sansekerta tentang poetika berjudul Kavyalankara, oleh penyair Kashmir Rudrata. Dalam Kavyalankara, pola The Knight's Tour disajikan sebagai figur puitis yang rumit yang disebut *turagapadabandha*, yang diterjemahkan menjadi "pengaturan dalam langkah-langkah kuda." Tur ksatria itu "tampaknya disajikan oleh serangkaian suku kata pada kotak-kotak yang masuk akal ketika dibaca dalam urutan tur." Bait yang sama dalam empat baris delapan suku kata dapat dibaca dengan mengikuti jalan ksatria dalam tur, dengan setiap suku kata dianggap sebagai kotak di papan catur.

Selama bertahun-tahun, Masalah Tur Ksatria berkembang ketika para matematikawan mulai tertarik. Leonhard Euler dan H. C. Von Warnsdorff membantu mengembangkan solusi penting.

Secara umum, ada dua jenis tur ksatria. Salah satunya adalah tur terbuka, yang hanya melibatkan ksatria mengunjungi setiap kotak hanya sekali. Yang lainnya adalah tur tertutup (atau reentrant), yang mirip dengan tur terbuka, kecuali kuda harus dapat kembali ke petak awalnya setelah mengunjungi semua petak di papan catur. Ini adalah variasi dari masalah NP-Lengkap yang terkenal yang disebut Masalah Siklus Hamilton. Intinya adalah bahwa untuk suatu graf tertentu, mustahil untuk menentukan secara apriori apakah Siklus Hamilton dari graf tersebut (misalnya, tur kuda tertutup dalam kasus ini) dimungkinkan.

Matematikawan Allen Schwenk mengajukan teorema terbukti bahwa papan catur berukuran $m \times n$ dengan $m \leq n$ memiliki tur kuda tertutup kecuali satu atau lebih dari tiga kondisi berikut berlaku: m dan n keduanya ganjil; $m = 1, 2, atau\ 4$; atau $m = 3$ dan $n = 4, 6, atau\ 8$.

8.2 TEKNIK PEMECAHAN MASALAH

Banyak solusi untuk Masalah Tur Ksatria telah ditemukan pada papan berukuran $8” \times 8”$ dengan menggunakan beberapa teknik pemecahan masalah. Salah satunya adalah *enumerasi lengkap*, tetapi itu akan sangat tidak efisien karena papan berukuran $8” \times 8”$ berisi

4×10^{51} kemungkinan urutan. Teknik pemecahan masalah yang lebih efisien adalah *bagi dan taklukkan*, di mana papan catur dibagi menjadi papan-papan yang lebih kecil yang diselesaikan secara individual. Metode yang paling efisien adalah menyelesaikan subtujuan, dan ini juga merupakan teknik yang paling umum digunakan. Solusi yang kami temukan menggunakan beberapa metode subtujuan yang lebih populer.

Salah satu metode adalah metode yang relatif mudah yang ditemukan oleh Abraham de Moivre. Ketika memulai dari petak sudut, kuda pertama-tama akan mengunjungi semua petak di sekitar tepi papan dan kemudian mengunjungi semua petak yang tersisa di tengah. Heuristiknya adalah pada setiap titik pilihan, kuda terlebih dahulu mengunjungi petak yang paling dekat dengan tepi (yaitu, petak dengan derajat masuk terkecil). Metode serupa digunakan dalam solusi yang disajikan dalam bab ini (lihat Bagian 8.3).

Metode populer kedua adalah *petak dan wajik*. Solusi ini pada dasarnya membagi papan catur menjadi empat subpapan berukuran $4'' \times 4''$. Di setiap subpapan ini, empat pola dibentuk dengan kuda selama perjalanan: belah ketupat, persegi, dan bayangan cermin untuk masing-masing petak. Metode ini digunakan dalam "Solusi D&S".

Metode ketiga adalah *teselasi*. Dasar dari metode ini adalah menggerakkan kuda pada lintasan yang menelusuri bentuk empat poligon yang, jika digabungkan, membentuk teselasi. Teselasi adalah sekelompok bentuk yang dapat saling berhimpitan sempurna pada suatu bidang tanpa tumpang tindih. Dengan metode ini, kuda dapat memulai dari petak mana pun di papan dan membentuk poligon tempat petak tersebut berada. Setelah menyelesaikan bentuk tersebut, kuda dapat berpindah ke petak lain yang merupakan bagian dari salah satu dari tiga poligon lainnya dan berjalan di sepanjang lintasan untuk menelusuri bentuknya. Perjalanan selesai setelah kuda menelusuri bentuk keempat poligon tersebut. Jika dilakukan dengan benar, metode teselasi dapat menghasilkan perjalanan tertutup. Metode ini digunakan dalam "Solusi Teselasi", yang dikembangkan oleh Daniel E. Thomasson.

8.3 SOLUSI TUR KUDA

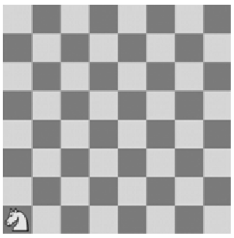
Bagian ini menyajikan solusi yang relatif sederhana, dengan langkah-langkah dan aturan yang mudah diikuti dan diingat. Metode ini didasarkan pada metode yang dikembangkan oleh Abraham de Moivre, di mana kuda bergerak di tepi luar papan catur. Dengan menggunakan metode ini, kita masih dapat mencapai putaran tertutup. Namun, ada titik-titik tertentu dalam solusi di mana heuristik harus dipatahkan. Pendekatan ini serupa dengan yang dikembangkan oleh Andrew Joseph, mantan mahasiswa Dr. Kopec.

Putaran kuda tertutup dapat dicapai dengan membuat kuda mengikuti dua aturan umum:

Aturan 1: *Bergerak sedekat mungkin dengan tepi luar papan.*

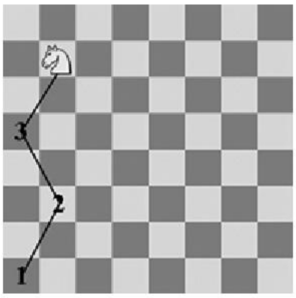
Aturan 2: *Ketika ada kesempatan, petak sudut harus diambil.*

Untuk solusi ini, kuda kita mulai dari sudut kiri bawah papan (Gambar 8.1).

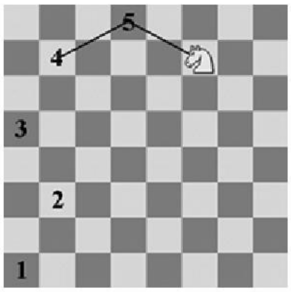


Gambar 8.1 Keadaan Awal (Kotak Kunjungan 1)

Ide umum pertama adalah bergerak dalam gerakan spiral mengelilingi papan, yang merupakan sesuatu yang akan dilakukan kuda di sebagian besar solusi. Kita akan mulai dengan melintasi kotak luar papan dengan gerakan searah jarum jam (Gambar 8.2 dan Gambar 8.3).

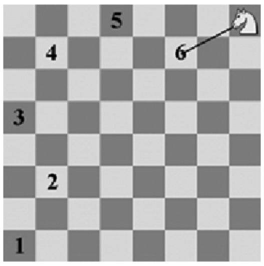


Gambar 8.2 Mengunjungi Kotak 2–4



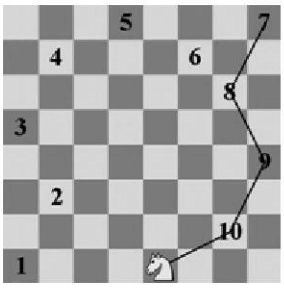
Gambar 8.3 Mengunjungi Kotak 5 Dan 6

Pada titik ini, kita memiliki kesempatan untuk bergerak ke pojok kanan atas papan catur. Kita akan memanfaatkan kesempatan ini dan mengarahkan kuda ke sana (Gambar 8.4).



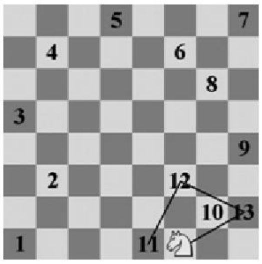
Gambar 8.4 Sudut Kanan Atas (Kotak 7)

Sekarang lanjutkan dengan Aturan 1 seperti biasa (Gambar 8.5).



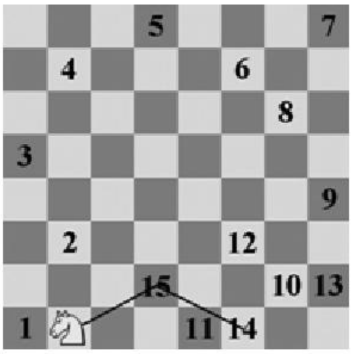
Gambar 8.5 Mengunjungi Kotak 8–11

Mendarat di kotak 11 adalah salah satu dari dua kunjungan yang sangat penting untuk diperhatikan, karena aturannya sekarang akan dihapus sementara. Lakukan langkah yang ditunjukkan pada Gambar 8.6.

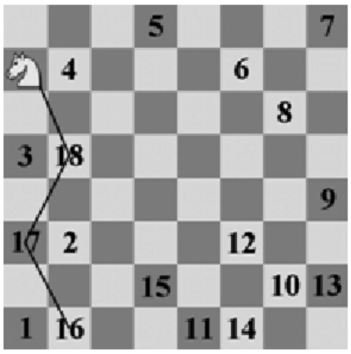


Gambar 8.6 Langgar Aturan—Pertama Kali (Mengunjungi Kotak 12–14)

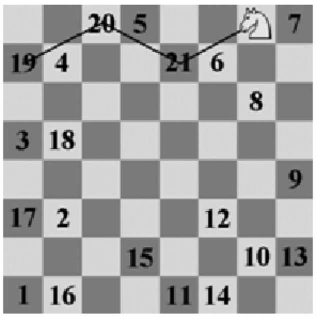
Meskipun rangkaian langkah ini tidak membantu kuda mengunjungi pojok kanan bawah papan, mengunjungi kotak 12 hingga 14 akan membantu kuda dengan langkah selanjutnya. *Untuk solusi "favorit" ini, kami menyarankan agar pemecah masalah cukup menghafal pengecualian ini, karena akan diperlukan ketika kita melakukan manuver serupa di kotak 36*. Sampai saat itu, teruslah mengikuti aturan (Gambar 8.7 hingga Gambar 8.10).



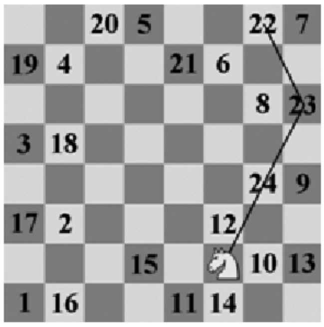
Gambar 8.7 Mengunjungi Kotak 15 Dan 16



Gambar 8.8 Mengunjungi Kotak 17–19

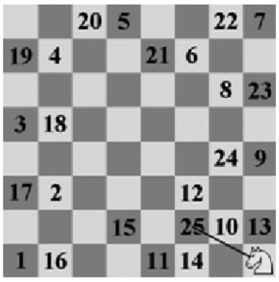


Gambar 8.9 mengunjungi kotak 20–22



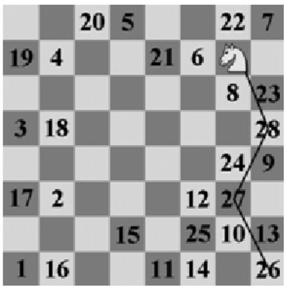
Gambar 8.10 Mengunjungi Kotak 23–25

Sekali lagi, kita memiliki kesempatan untuk mengunjungi kotak sudut. Jadi, kita akan mengambilnya (Gambar 8.11).

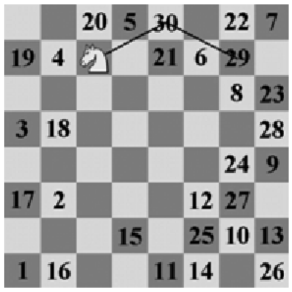


Gambar 8.11 Pojok Kanan Bawah (Kotak 26)

Sekarang lanjutkan mengikuti Aturan 1. Namun, kita tidak akan melanjutkan gerakan spiral searah jarum jam pada titik ini. Sebaliknya, kuda sekarang akan mulai melintasi papan dengan gerakan berlawanan arah jarum jam (Gambar 8.12 dan Gambar 8.13).

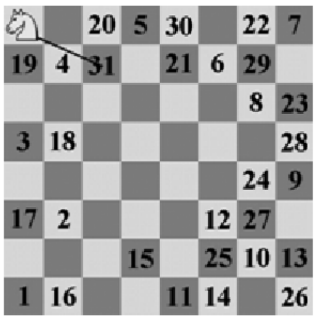


Gambar 8.12 Mengunjungi Kotak 27–29



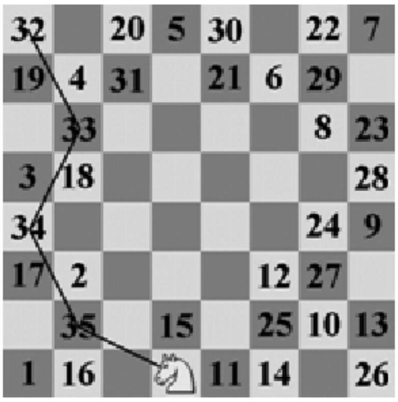
Gambar 8.13 Mengunjungi Kotak 30 Dan 31

Kita sekarang memiliki kesempatan untuk mengisi sudut terakhir di kiri atas papan. Kita akan memanfaatkannya (Gambar 8.14).



Gambar 8.14 Sudut Kiri Atas (Kotak 32)

Lalu kita lanjutkan (Gambar 8.15).



Gambar 8.15 MENGUNJUNGI KOTAK 33–36

Setelah mencapai kotak 36, kita akan sekali lagi melanggar aturan. Pindahkan kuda ke kotak yang ditunjukkan pada Gambar 8.16.

Gambar 8.16 Melanggar Aturan—Kedua Kalinya (Mengunjungi Kotak 37 Dan 38)

Pada titik ini, lanjutkan mengikuti aturan, dengan kuda sekali lagi melintasi kotak terluar papan searah jarum jam, seperti yang ditunjukkan pada **Gambar 8.17** hingga Gambar 8.20.

Gambar 8.17 Kotak Kunjungan 39–41

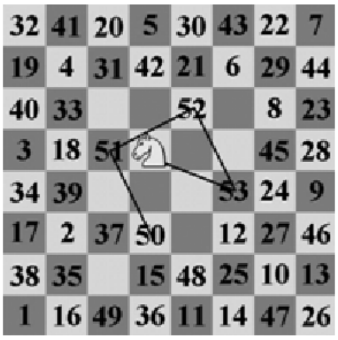
Gambar 8.18 Kotak Kunjungan 42–44

Gambar 8.19 Kotak Kunjungan 45–47

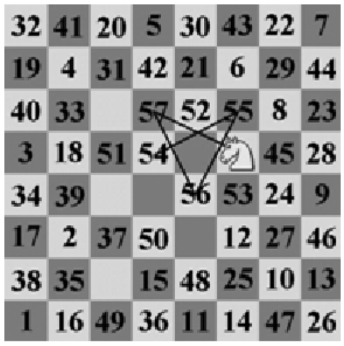
Gambar 8.20 Kotak Kunjungan 48–50

Pada titik ini, kita telah menyelesaikan penjelajahan kuda di sekitar petak terluar papan. Anda akan melihat bahwa salah satu petak belum dikunjungi. Ini karena petak tersebut telah dicadangkan agar kuda dapat menyelesaikan perjalanan tertutup.

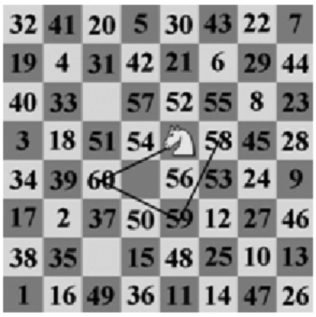
Sekarang kuda harus mengunjungi setiap petak tengah yang tersisa. Ide umumnya sekarang adalah mempertahankan gerakan selanjutnya searah jarum jam (Gambar 8.21 hingga Gambar 8.23).



Gambar 8.21 Kotak Pusat Kunjungan 51–54

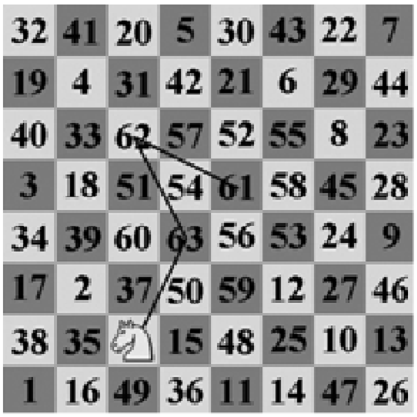


Gambar 8.22 Kotak Pusat Kunjungan 55–58



Gambar 8.23 Kotak Pusat Kunjungan 59–61

Sisanya sekarang sudah jelas (Gambar 8.24).



Gambar 8.24 Keadaan Sasaran (Kotak Kunjungan 62–64)

Dengan demikian, kita telah menyelesaikan perjalanan kuda kita. Selain itu, karena kita dapat memindahkan kuda ke posisi pertamanya dari posisi terakhirnya, perjalanan ini merupakan perjalanan tertutup.

8.4 PEMECAHAN MASALAH MANUSIA

Pada musim semi 2013, Masalah Tur Ksatria dipresentasikan kepada beberapa mahasiswa. Mereka tidak diamati ketika diberikan masalah tersebut, sehingga tidak pasti apakah para mahasiswa memperoleh solusi mereka dengan cara yang adil atau tidak. Mayoritas mahasiswa mengambil pendekatan yang serupa dengan "Solusi Sederhana". Mereka mulai dengan menggerakkan kuda mengelilingi kotak-kotak terluar papan. Kemudian, ketika mereka mencapai titik di mana mereka tidak dapat melanjutkan, mereka melintasi beberapa kotak tengah sehingga mereka dapat mencapai kotak terluar dan melanjutkan perjalanan ke kotak-kotak terluar sekali lagi. Akhirnya, setelah semua kotak terluar dikunjungi, mereka memindahkan kuda ke tengah dan mengunjungi kotak-kotak yang tersisa.

Seorang mahasiswa menggunakan pendekatannya sendiri setelah menemukan solusi setelah beberapa "coba-coba". Sebagaimana dijelaskan dalam uraiannya, pendekatan utamanya adalah "pertama-tama menghilangkan kotak sudut [dan] kemudian kotak tengah." Namun, ia tidak mengikuti heuristik ini secara spesifik. Pertama, ia menelusuri kotak-kotak luar, mencoba mengisi setiap sudutnya. Kemudian, ia memasuki pusat dan mengisi setiap sudut secara acak hingga setiap kotak telah dikunjungi. Pendekatan semacam itu mungkin berhasil untuk tur terbuka (mengunjungi setiap kotak tanpa mempertimbangkan di mana kita memulai dan mengakhiri), tetapi kemungkinan besar tidak akan berhasil untuk tur tertutup, karena kotak-kotak transisi ke kotak-kotak tepi terluar mungkin sudah dikunjungi.

8.5 ANALISIS JENDELA MANUSIA DARI SOLUSI

Kami menemukan beberapa solusi untuk Masalah Tur Ksatria. Hal ini tidak mengherankan, karena terdapat begitu banyak metode untuk menemukan tur dan ini merupakan masalah yang relatif populer. Saat menganalisis solusi (Tabel 8.1), kita dapat melihat bahwa terdapat campuran antara solusi ekstensional dan intensional. Hal ini karena solusi untuk Masalah Tur Ksatria dapat dinyatakan dalam dua cara: solusi dapat memberikan langkah-langkah umum dari suatu metode yang digunakan untuk menemukan tur kuda, mengklasifikasikannya sebagai intensional, dan solusi juga dapat direpresentasikan dengan menunjukkan bagaimana suatu metode digunakan untuk menemukan tur langkah demi langkah, menggunakan diagram aktual papan catur dan kuda, mengklasifikasikannya sebagai ekstensional.

Dari Tabel 8.1, kita dapat melihat bahwa sebagian besar solusi bersifat ekstensional. Selain itu, solusi-solusi tersebut memiliki peringkat yang relatif tinggi dalam ekstensionalitas. Untuk solusi Tur Ksatria ekstensional, semakin tampak seperti papan catur sungguhan, semakin tinggi ekstensionalitasnya. Kita dapat melihat bahwa masing-masing solusi menggunakan metode dan representasi pemecahan masalah yang serupa.

Mereka juga memiliki peringkat yang relatif identik dalam hal pemahaman. Pemahaman solusi bergantung pada tingkat detail yang digunakan untuk representasinya (misalnya, papan catur, lintasan, penomoran). Dari peringkat tersebut, terlihat bahwa tidak ada solusi yang sangat mudah dipahami. Namun, ada kemungkinan bahwa, ketika menggabungkan atribut spesifik dari setiap solusi, representasi MHWC dapat dihasilkan.

Terakhir, kita dapat melihat bahwa peringkat untuk eksekusi solusi umumnya beragam. Jelas bahwa eksekusi solusi bergantung pada metode yang digunakan untuk menemukan tur ksatria. Meskipun ada begitu banyak metode yang dapat digunakan untuk menemukan tur, beberapa dapat dilakukan jauh lebih mudah daripada yang lain. Oleh karena itu, salah satu faktor yang dapat menentukan apakah suatu solusi merupakan MHWC adalah eksekusinya, dan dengan demikian menemukan metode "terbaik" untuk menyelesaikan Masalah Tur Ksatria.

Kesulitan
5/10

kerumitan
O(n)

Nama	Int atau Ext?	Intens./Ekstens.	Repr.	Jendela Manusia?	Benar?	Ukuran Butir	Eksekusi	Keterpahaman	Metode Pemecahan Masalah	Fleksibilitas	Mode Konversi	Optimal?	Total
Algoritma	Int.	8/10	Kode	Y	Y dng siklus	Ideal	8/10	7/10	Kembali dg Heuristik	7/10	Editor Teks	N	30/40
D&S	Int.	6/10	Repr. Vis. dg Deskripsi	Y	Y	Ideal	9/10	6/10	Sub-tujuan	7/10	Editor Teks; Pembuat Gbr	Y	28/40
Green	Ekst.	8/10	Grid/Tabel; Gbr	Y	Y	Ideal	5/10	7/10	Repr. Vis.; Sub-tujuan	8/10	Pembuat Gbr/Tabel	Y	28/40
Narayan	Ekst.	7/10	Grid/Tabel; Gbr	Y	Y	Ideal	8/10	7/10	Repr. Vis.; Sub-tujuan	7/10	Pembuat Gbr/Tabel	Y	29/40
Semi-Magic	Ekst.	7/10	Grid/Tabel; Gbr	Y	Y	Ideal	4/10	7/10	Repr. Vis.; Sub-tujuan	7/10	Pembuat Gbr/Tabel	Y	25/40
Simple	Ekst.	8/10	Grid/Tabel; Gbr	Y	Y	Ideal	8/10	9/10	Repr. Vis.; Sub-tujuan	8/10	Pembuat Gbr/Tabel	Y	33/40

Tessellation	Ekst.	8/10	Grid/Tabel; Gbr	Y	Y	Kecil	10/10	7/10	Repr. Vis.; Sub-tujuan	8/10	Pembuat Gbr/Tabel	Y	33/40
Wiki	Ekst.	7/10	Grid/Tabel; Gbr	Y	Y	Ideal	8/10	7/10	Repr. Vis.; Sub-tujuan	8/10	Pembuat Gbr/Tabel	Y	30/40

Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tabel 8.1 Peringkat Solusi Masalah Tur Kuda Berdasarkan Jendela Manusia

Solusi yang Paling Sesuai dengan Jendela Manusia

Solusi yang kami analisis sebagai solusi MHWC adalah "Solusi Sederhana". Solusi ini sama dengan yang disajikan di Bagian 8.3. Solusi ini sangat mudah dipahami. Solusi ini direpresentasikan menggunakan kisi, yang membuat ruang keadaan tampak seperti papan catur. Selain itu, seperti pada papan catur, warna kotak berganti antara gelap dan terang, yang membuat setiap kotak dapat diidentifikasi satu sama lain. Selain itu, entitas yang mewakili kuda adalah figur dari bidak kuda yang sebenarnya. Ini merupakan pengingat bahwa solusi harus mengikuti batasan langkah yang sama seperti kuda. Hal ini sangat membantu bagi mereka yang belum familiar dengan catur. Setelah setiap langkah, sebuah garis yang mewakili jalur yang telah ditempuh kuda sejauh ini digambar, menghubungkan posisi kuda sebelumnya dengan posisinya saat ini. Garis-garis ini tetap ada untuk setiap langkah solusi. Gambar akhir dalam solusi ini merangkum keseluruhan tur. Ini menunjukkan seluruh jalur yang diambil kuda, serta angka-angka pada setiap kotak yang menunjukkan urutan langkah yang dilakukan ksatria dalam menyelesaikan putaran.

Meskipun solusinya mudah dipahami, representasinya dapat ditingkatkan untuk meningkatkan pemahaman keseluruhannya. Misalnya, atribut lain yang dapat ditambahkan adalah huruf dan angka untuk menunjukkan nama baris dan kolom, mirip dengan papan catur sungguhan. Hal ini dapat membuat solusi lebih mudah dilacak, dan urutan serta kotak tertentu dapat dilacak, dihafal, atau sekadar diidentifikasi.

Yang menjadikan solusi ini MHWC adalah metode yang digunakan. Metode ini tidak hanya sangat mudah dieksekusi, tetapi juga membutuhkan sedikit atau tanpa memori untuk dijalankan, yang ideal bagi kebanyakan manusia. Meskipun ada beberapa aturan yang harus diikuti dan beberapa lokasi di mana heuristik harus dilanggar, aturan-aturan tersebut relatif mudah diingat.

Anda mungkin memperhatikan dari Tabel 8.1 bahwa "Solusi Tessellasi," yang dikembangkan oleh Daniel E. Thomasson (lihat Bagian 8.2), memiliki eksekusi yang lebih tinggi daripada "Solusi Sederhana." Ini karena metode tessellasi tidak hanya menghasilkan tur tertutup, tetapi juga menjamin bahwa tur akan dihasilkan setelah menggerakkan kuda di sekitar papan sehingga menelusuri empat poligon. Di mana pun kuda mulai di papan, jika keempat bentuk itu dilacak, tur akan dibuat. Dengan demikian, eksekusi solusinya sangat tinggi. Sayangnya, untuk menjalankan metode ini, diperlukan memori yang cukup banyak, karena seseorang harus menghafal keempat bentuk dan kotak mana yang sesuai dengan bentuk tersebut. Oleh karena itu, ukuran butiran solusinya dianggap kecil. Namun, "Solusi Sederhana," menyediakan langkah-langkah yang mudah diingat.

Solusi Jendela yang Paling Tidak Kompatibel dengan Manusia

Solusi yang dinilai sebagai LHWC adalah "Solusi Semi-Ajaib" (Gambar 8.25). Solusi ini juga dikembangkan oleh Daniel E. Thomasson. Metode yang digunakan untuk menghasilkan tur ksatria ini adalah metode kuadrat dan wajik (lihat Bagian 8.2), meskipun secara implisit digunakan.

Solusi ini menunjukkan konsep "Tur Semi-Ajaib", yaitu tur ksatria yang menghasilkan pembentukan *persegi semi-ajaib*. Matriks $n \times n$ yang terdiri dari nilai-nilai dari 1 hingga $n2$ disebut *persegi ajaib* jika angka-angka pada setiap baris, kolom, dan diagonal berjumlah satu angka yang dikenal sebagai *konstanta ajaib*. Jika nilai-nilai pada setiap baris dan kolom berjumlah angka ini, tetapi diagonalnya tidak, maka matriks tersebut dianggap sebagai kuadrat semi-ajaib.

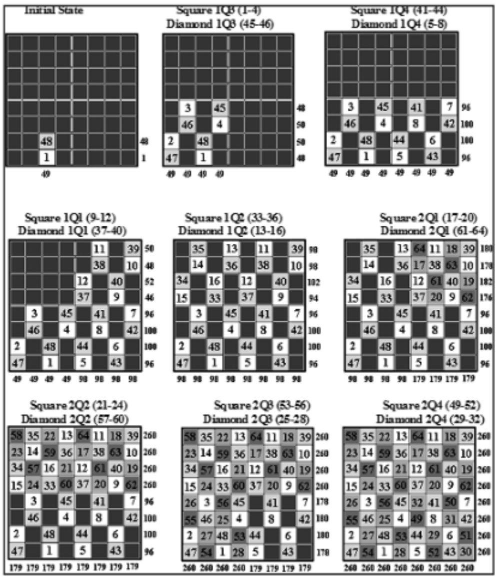
Menemukan tur kuda dapat diibaratkan seperti menghasilkan kotak berukuran $8” \times 8”$ yang nilainya berkisar antara 1 hingga 82. Setiap kotak di papan catur dapat diberi nilai yang sesuai dengan nomor langkah kuda tersebut mendarat di kotak tersebut. Setelah tur selesai, jika nilai kotak di setiap baris, kolom, dan diagonal berjumlah sama, seluruh papan catur dianggap sebagai kotak ajaib. Namun, telah terbukti bahwa kotak ajaib tidak dapat dibentuk oleh tur kuda pada papan berukuran $8” \times 8”$. Namun, dimungkinkan untuk menemukan tur yang menghasilkan kotak semi-ajaib. Kotak semi-ajaib pertama ditemukan oleh Leonhard Euler, di mana setiap baris dan kolom papan catur berjumlah konstanta ajaib 260.

"Solusi Semi-Sihir" menunjukkan metode pembentukan kotak semi-sihir dengan konstanta ajaib yang sama, yaitu 260. Dengan menggunakan metode kotak dan wajik, dua tur ksatria simultan dihasilkan: satu dimulai dari kotak 1 dan bertambah secara bertahap, dan yang lainnya berkurang secara bertahap. Kedua tur selalu mengunjungi kotak-kotak yang vertikal satu sama lain, dengan jumlah langkah yang berjumlah 49 dan 81. Meskipun solusi ini menghasilkan tur ksatria yang lengkap dan tur semi-sihir serta mengungkap simetri di baliknya, kami akan menganalisis representasinya dalam konteks Jendela Manusia, seolah-olah solusi ini dimaksudkan untuk ditunjukkan kepada audiens mana pun (yaitu, mereka yang belum familiar dengan Masalah Tur Ksatria). Solusi ini juga dipersingkat untuk menghemat ruang, karena berisi total 32 langkah.

Solusi ini mungkin tidak terlalu mudah dieksekusi oleh manusia karena terlalu banyak tujuan yang harus diingat. Metode kotak dan wajik simultan diulang untuk keempat kuadran, dan nilai kolomnya berjumlah 98. Setelah itu, proses yang sama dilanjutkan, kecuali langkah-langkahnya harus diberi nomor agar berjumlah 81. Untuk melakukan solusi ini, seseorang harus mengingat dengan tepat bentuk mana yang akan dihasilkan pada setiap langkah dan harus mengingat urutan pembentukannya.

Solusi ini mungkin tidak terlalu mudah dieksekusi oleh manusia karena terlalu banyak tujuan yang harus diingat. Metode kotak dan wajik simultan diulang untuk keempat kuadran, dan nilai kolomnya berjumlah 98. Setelah itu, proses yang sama dilanjutkan, kecuali langkah-langkahnya harus diberi nomor agar berjumlah 81. Untuk melakukan solusi ini, seseorang harus mengingat dengan tepat bentuk mana yang akan dihasilkan pada setiap langkah dan harus mengingat urutan pembentukannya.

Solusinya mungkin juga tidak terlalu mudah dipahami. Solusi ini tidak memiliki kuda untuk diikuti dalam perjalanan, juga tidak ada garis yang menunjukkan jalurnya. Solusi ini hanya menunjukkan jumlah kotak tempat kuda akan mendarat selama langkah tertentu. Hal ini dapat membuat solusi sulit diikuti dan dapat mengalihkan seseorang dari fakta bahwa ini adalah solusi untuk Masalah Perjalanan Ksatria. Terakhir, kotak-kotak tersebut diarsir secara berbeda untuk menunjukkan bentuk tertentu (yaitu, wajik dan kotak) yang sedang dibentuk. Campuran ini dapat membuat solusi sulit dipahami, bahkan untuk gambar yang menunjukkan perjalanan ksatria yang telah selesai.



Gambar 8.25 "Solusi Semi-Ajaib" (Dipersingkat)

8.6 SOLUSI MESIN TERBAIK

Banyak algoritma dan heuristik telah dikembangkan untuk menemukan jalur kuda. Salah satu algoritma yang dapat digunakan adalah *algoritma bagi-dan-kuasai*, yang membagi papan catur menjadi bidak-bidak yang lebih kecil, menemukan urutan untuk bidak-bidak tersebut, dan menggabungkannya (lihat Bagian 8.2). Terdapat juga solusi jaringan saraf tiruan, di mana papan catur diimplementasikan sebagai jaringan saraf tiruan dan setiap kotak direpresentasikan sebagai neuron.

Mengingat terdapat sekitar 4×10^{51} kemungkinan urutan pada papan berukuran 8” x 8”, sebaiknya komputer dipandu untuk menemukan jalur kuda melalui penggunaan heuristik agar tidak perlu menganalisis setiap urutan tersebut. Salah satu heuristik yang paling efisien dan umum digunakan adalah yang didasarkan pada Aturan Warnsdoff, yang dikembangkan oleh H. C. Warnsdoff. Aturannya adalah menganalisis petak-petak yang dapat dikunjungi kuda dari posisinya saat ini, dan petak mana pun yang memberikan kemungkinan penerus paling sedikit bagi kuda adalah petak yang harus ditujunya.

Namun, kelemahan Aturan Warnsdoff adalah tidak selalu berhasil menemukan tur lengkap, karena biasanya berujung pada jalan buntu. Menggunakan heuristik untuk menemukan tur kuda terkadang dapat mengakibatkan kegagalan, terutama ketika tur harus ditemukan di papan catur yang lebih besar. Seiring nilai n untuk papan catur $n \times n$ menjadi besar, tingkat keberhasilan Aturan Warnsdoff menurun. Salah satu metode untuk mengatasi masalah ini adalah menggabungkan heuristik dengan *algoritma lacak balik*. Algoritma ini akan mengikuti heuristik Warnsdoff seperti biasa, tetapi ketika kuda mencapai jalan buntu, algoritma tersebut akan kembali ke langkah sebelumnya dan mencoba menemukan tur sekali lagi. Jenis algoritma yang sama ini digunakan dalam "Solusi Algoritma".

Metode kedua yang lebih efisien telah dianalisis oleh Luis Paris. Algoritma ini mengimplementasikan heuristik Warnsdoff dengan penambahan heuristik kedua yang, jika terjadi seri antara petak yang mungkin dikunjungi seorang ksatria pada langkah tertentu, akan "memilih petak bernilai terendah yang paling dekat dengan salah satu dari empat sudut papan, terlepas dari n." Menurut analisis yang dilakukan oleh Paris, heuristik ini selalu berhasil menemukan tur, tidak peduli seberapa besar papan caturnya. Algoritma ini tidak menggunakan backtracking dan membutuhkan waktu $\mathbf{O}(\mathbf{n}^2)$.

George Koltanowski, seorang Master Catur Internasional dengan ingatan eidetik, terkenal dengan ceramah "Tur Ksatria"-nya. Salah satu rekan penulisnya mengenang AS Terbuka di Portland, Oregon, pada tahun 1987, di mana ia memberikan demonstrasi dengan meminta penonton untuk menyebutkan nomor seri pada uang dolar, nama-nama pemain bisbol terkenal, dan sebagainya, pada petak-petak di papan catur. Koltanowski kemudian akan membelakangi penonton dan mengamati papan selama beberapa menit. Kemudian ia akan berbalik menghadap penonton, melakukan tur ksatria yang sempurna, sambil membacakan semua informasi yang telah diberikan untuk setiap petak. Tidak mengherankan, Koltanowski juga memegang rekor resmi dunia untuk Catur Buta—56 papan—yang berlangsung di San Francisco pada tahun 1960.

8.7 SOAL TERKAIT

Perpanjangan dari soal yang dibahas dalam bab ini adalah Tur Ksatria pada papan tak terhingga. Agar hal ini terjadi, tempatkan kuda di tengah papan. Lakukan satu langkah dua petak ke bawah dan satu petak ke kanan untuk memulai, lalu teruslah berputar mengelilingi pusat hingga setiap petak telah dikunjungi. Kemudian, lakukan langkah lain seperti langkah pertama untuk memperluas perimeter luar tur dan teruslah berputar mengelilingi bagian dalam hingga setiap blok di perimeter tersebut telah dikunjungi. Sekali lagi, lakukan langkah seperti langkah pertama, dan terus ulangi langkah ini untuk papan tak terbatas ini.

8.8 PROGRAM YANG DAPAT DIMAINKAN

Permainan tur kuda yang dapat dimainkan tersedia di $http://www.kongregate.com/games/evgenykarataev/knights - tour$. Dalam permainan ini, Anda dapat mengendalikan kuda dan memilih petak mana yang akan dikunjungi relatif terhadap posisinya saat ini. Petak yang dapat dikunjungi dan yang telah dikunjungi akan ditunjukkan. Selain itu, papan catur dan kuda ditampilkan dalam bentuk tiga dimensi, memberikan kesan kepada pemain seperti menggunakan papan catur sungguhan.

BAB 9
MASTERMIND

9.1 PENDAHULUAN

Mastermind adalah permainan papan yang diciptakan pada tahun 1970 oleh Mordecai Meirowitz, seorang kepala kantor pos dan pakar telekomunikasi Israel. Setelah ditolak oleh beberapa perusahaan mainan terkemuka, hak cipta Mastermind akhirnya dibeli oleh Invicta Plastics Ltd, sebuah perusahaan kecil Inggris, dan dirilis pada tahun 1972. Menurut G. Darby, "Lebih dari 50 juta kopi kemudian, permainan ini masih dipasarkan hingga saat ini".

Mastermind adalah jenis permainan "pemecahan kode". Dalam permainan Mastermind asli, seorang pemain, yang dikenal sebagai pemecah kode, harus menebak kode rahasia yang dipilih oleh pemain lain, yang dikenal sebagai pembuat kode. Kode ini berupa urutan empat pasak yang dapat berupa enam warna berbeda, dengan pasak berwarna sama diperbolehkan dalam urutan yang sama. Pemecah kode harus memecahkan kode ini dengan menentukan warna dan urutan pasak yang tepat. Untuk melakukannya, pemecah kode harus membuat serangkaian paling banyak 10 tebakan. Setelah setiap tebakan, pembuat kode memberi pemecah kode skor dalam bentuk dua nilai: jumlah pasak yang berwarna tepat dan berada di posisi yang benar dan jumlah pasak yang berwarna tepat tetapi tidak pada posisi yang benar. Angka-angka ini masing-masing diwakili oleh pasak hitam dan putih kecil. Jika dalam 10 tebakan pemecah kode tidak dapat menentukan urutan yang benar, pembuat kode menang.

Sejak dirilis pada tahun 1972, telah ada berbagai macam versi alternatif dari permainan asli. Satu versi disebut Mastermind44, yang mencakup enam warna, lima pasak untuk satu urutan, dan sebanyak tiga pembuat kode. Versi lain adalah Ultimate Mastermind, yang mencakup delapan warna dan lima pasak untuk satu urutan.

9.2 TEKNIK PEMECAHAN MASALAH

Seperti halnya The 15 Puzzle, Mastermind tidak memiliki status awal yang spesifik. Namun, tidak seperti soal-soal yang telah kami analisis dalam buku ini, soal ini tidak memiliki status tujuan yang spesifik. Selain itu, untuk soal-soal lainnya, kami mencoba menemukan metode untuk mencapai tujuan, sementara di Mastermind kami mencoba menemukan tujuan itu sendiri. Kita dapat mengatakan bahwa The Knight's Tour Problem serupa karena tidak memiliki tujuan yang ditetapkan secara spesifik. Namun, kita tahu bahwa ksatria harus mengunjungi semua 64 petak, jadi masalahnya tidak sama.

Dalam permainan Mastermind asli, terdapat total $6 \times 6 \times 6 \times 6 = 1.296$ kemungkinan kombinasi, dan pemecah kode harus menentukan satu urutan warna yang benar! Dengan hanya 10 tebakan yang tersedia, kuncinya adalah mengurangi jumlah kemungkinan kombinasi sebanyak mungkin dengan setiap tebakan. Hal ini dapat dilakukan dengan menganalisis skor yang diberikan pada setiap tebakan dan menggunakan deduksi untuk menentukan pasak berwarna mana yang akan digunakan dan mana yang tidak, serta urutan penempatannya. Oleh karena itu, *deduksi* merupakan teknik pemecahan masalah yang paling berharga untuk digunakan dalam memecahkan urutan pasak yang tepat dan memenangkan Mastermind. Deduksi logis semacam ini juga dilakukan untuk menyelesaikan soal n koin guna menentukan cara terbaik membagi jumlah koin untuk setiap penimbangan.

9.3 SOLUSI MASTERMIND

Sebelum memulai solusi, mari kita bahas beberapa notasi:

- Alih-alih menggunakan warna, pasak akan direpresentasikan sebagai huruf dari A hingga F. Notasi ini lebih baik daripada menggunakan nama warna sebenarnya karena versi permainan yang berbeda mungkin menggunakan pasak dengan warna yang berbeda.
- Urutan pasak akan direpresentasikan dengan menempatkan empat huruf ini dalam satu baris. Misalnya, ABBC adalah urutan empat pasak yang berisi warna pasak A, warna pasak B, warna pasak B, dan warna pasak C, dalam urutan yang tepat.
- Skor akan direpresentasikan sebagai serangkaian angka dalam tanda kurung: (x, y). Variabel x akan mewakili jumlah pasak yang ditebak dengan benar dan berada di lokasi yang benar (yaitu, pasak hitam kecil), dan variabel y akan mewakili jumlah pasak yang ditebak dengan benar tetapi berada di posisi yang salah (yaitu, pasak putih kecil). Misalnya, (2, 1) berarti 2 pasak memiliki warna dan lokasi yang benar dalam urutan akhir, dan 1 pasak lainnya memiliki warna yang benar tetapi tidak berada di lokasi yang benar dalam urutan akhir. Urutan ini memiliki skor total 3.

Contoh tebakan tunggal mungkin terlihat seperti Gambar 9.1.

●ABBC	(2,1)

Gambar 9.1 Contoh Tebakan

Karena tidak ada tujuan spesifik yang ingin dicapai, tidak ada solusi umum untuk memenangkan Mastermind. Namun, dengan mengambil langkah-langkah berikut, pemecah kode dapat menyelesaikan deret tersebut dalam empat hingga enam tebakan:

1. Tebakan pertama harus selalu deret AABB. Ini bukan hanya tebakan paling sederhana, tetapi juga sangat penting karena, bahkan dalam kasus terburuk, tebakan ini sangat mengurangi jumlah kemungkinan yang tersisa (dalam kasus terburuk, tebakan ini mengurangi 1.296 kemungkinan tebakan menjadi 256).
2. Anda harus menggunakan skor tebakan pertama untuk membuat asumsi tentang pasak mana dari tebakan sebelumnya (yaitu, pasak A dan B) yang harus digunakan untuk tebakan berikutnya. Misalnya, jika skornya total 0, pasak A dan B tidak boleh digunakan lagi untuk permainan ini. Jika skornya total 1 atau 2, gunakan A dan B pada tebakan berikutnya untuk menentukan A atau B mana yang ada dalam deret tersebut. Jika skor totalnya 3, gunakan dua pasak A dan satu pasak B pada tebakan berikutnya. Terakhir, jika skornya 4, artinya keempat pasak berada dalam urutan tersebut dan harus disusun dalam urutan yang benar.

3. Ulangi langkah 2 untuk tebakan berikutnya hingga semua warna pasak telah digunakan (yaitu, jika kita tidak mendapatkan skor 4 sebelum itu). Pada titik ini, seharusnya sudah ada informasi yang cukup dari semua tebakan yang telah dibuat untuk menggunakan deduksi dan menyelesaikan urutan tersebut.

Pada awalnya, langkah-langkah di atas mungkin tampak terlalu luas dan tidak jelas. Namun, kami akan memberikan lima contoh agar Anda dapat memahaminya dengan lebih baik.

Contoh 1

Mari kita mulai dengan contoh yang mudah. Urutan tujuannya adalah CEDF. Kita mulai dengan tebakan pertama yang diperlukan dan mendapatkan Gambar 9.2.

1.	AABB	(0,0)

Gambar 9.2 Contoh 1—Tebakan 1

Dari sini, kita tahu bahwa baik A maupun B tidak ada dalam deret tersebut. Oleh karena itu, kita akan membuat tebakan serupa menggunakan warna pasak C dan D, sehingga jumlah pilihannya semakin berkurang (Gambar 9.3).

2.	CCDD	(2,0)
1.	AABB	(0,0)

Gambar 9.3 Contoh 1—Tebakan 2

Kita sekarang memiliki skor 2. Lebih baik lagi, kita memiliki dua pasak yang berada di posisi yang benar. Namun, karena kita tidak tahu pasak mana yang benar, kita perlu membuat asumsi. Kita dapat memilih dua pasak apa pun, tetapi sebaiknya kedua warna tersebut dimasukkan dalam tebakan berikutnya. Kita akan mempertahankan C pertama dan D pertama dalam urutan tersebut, dan, karena ada dua tempat yang tersedia, kita akan menambahkan dua pasak E (Gambar 9.4).

			Asumsi:
3.	CEDE	(3,0)	C_D_
2.	CCDD	(2,0)	
1.	AABB	(0,0)	

Gambar 9.4 Contoh 1—Tebakan 3

Skor kita meningkat menjadi 3, dengan ketiga pasak berada di posisi yang benar. Dari sini, kita dapat menyimpulkan tiga hal: asumsi kita tentang penyertaan dan lokasi pasak C dan D benar; pasak E ada dalam urutan; dan dengan satu pasak masih salah dan satu warna tersisa, pasak F juga harus ada dalam urutan. Ini berarti salah satu pasak E harus dihilangkan. Kita akan berasumsi bahwa E di posisi 2 benar dan E di posisi 4 harus diganti dengan F (Gambar 9.5).

	Asumsi:

4.	CEDF	(4,0)*	CEDF
3.	CEDE	(3,0)	
2.	CCDD	(2,0)	
1.	AABB	(0,0)	

Gambar 9.5 Contoh 1—Tebakan 4

Dengan demikian, kita telah menyelesaikan deret tersebut. Tentu saja, tebakan 4 adalah tebakan yang beruntung. Jika kita berasumsi bahwa E pada posisi 4 benar, bukan posisi 2, kita akan mendapatkan Gambar 9.6.

			Asumsi
4.	CFDE	(2,2)	CEFE

Gambar 9.6 Contoh 1—Tebakan Alternatif 4

Namun, dari informasi ini, kita dapat dengan mudah menentukan solusinya, dan kita hanya perlu satu tebakan tambahan untuk menyelesaikannya. Saat melihat skor, kita dapat melihat bahwa keempat pasak berada dalam urutan akhir, dengan dua pasak di posisi yang benar dan dua di posisi yang salah. Karena kita sudah tahu bahwa pasak C dan D berada di tempat yang benar, kita dapat dengan mudah menyimpulkan bahwa pasak E dan F harus ditukar.

Contoh 2

Urutan tujuannya adalah EAFA. Seperti biasa, kita membuat tebakan pertama yang diperlukan (Gambar 9.7).

1.	AABB	(1,1)

Gambar 9.7 Contoh 2—Tebakan 1

Kita melihat bahwa kita memiliki skor 2, dengan satu pasak di posisi yang benar dan satu di posisi yang salah. Skor 2 untuk tebakan pertama menyulitkan untuk menentukan A atau B mana yang benar. Mari kita asumsikan bahwa kedua warna berada di urutan terakhir, dan kita akan menempatkan keduanya di tebakan berikutnya. Mari kita asumsikan pasak A pertama berada di tempat yang benar dan salah satu pasak B tidak pada tempatnya. Pindahkan B ke posisi 2 dan tambahkan dua warna berikutnya ke urutan tersebut (Gambar 9.8).

			Asumsi:
2.	ABCC	(0,1)	AB_ _
1.	AABB	(1,1)	

Gambar 9.8 Contoh 2—Tebakan 2

Skor kita sekarang berkurang menjadi 1, dengan satu pasak berada di luar posisi. Ini memberi kita dua kesimpulan: C tidak berada dalam deret akhir, dan baik dua pasak A maupun dua pasak B berada dalam solusi akhir, tetapi tidak keduanya. Kita mengetahui hal ini karena skor 2 kita pada tebakan pertama telah berkurang menjadi 1. Untuk tebakan 3, kita akan mengasumsikan bahwa dua A berada dalam deret tersebut. Sekarang kita akan membuat beberapa asumsi lagi berdasarkan asumsi ini. Karena skor kita saat ini menunjukkan bahwa kita memiliki satu pasak yang tidak pada posisinya, kita tahu bahwa A tidak mungkin berada di posisi 1. Selain itu, dari melihat skor kita pada tebakan 1, salah satu pasak A pasti berada di posisi 2, karena terdapat pasak hitam. Oleh karena itu, mari kita asumsikan bahwa satu A berada di posisi 2. A kedua dapat ditempatkan di posisi 3 atau 4, jadi mari kita asumsikan berada di posisi 3. Terakhir, kita sekarang akan memperkenalkan warna pasak D (Gambar 9.9).

			Asumsi:
3.	DAAD	(1,1)	_AA _
2.	ABCC	(0,1)	
1.	AABB	(1,1)	

Gambar 9.9 Contoh 2—Tebakan 3

Kita sekali lagi mendapatkan skor 2, dengan satu pasak pada tempatnya dan satu pasak yang tidak pada tempatnya. Sayangnya, dengan tebakan ini, kita tidak dapat langsung mengambil kesimpulan. Tebakan ini dapat berarti salah satu dari dua hal:

1. Kedua A berada dalam urutan tersebut, dan D tidak berada di posisi tersebut. Kita hanya perlu memindahkan A yang berada di posisi 3 ke posisi 4.
2. D berada dalam urutan tersebut, dan asumsi kita tentang memasukkan A salah, sehingga dua B tetap berada di urutan terakhir.

Mari kita lanjutkan dengan #1 dan asumsikan bahwa A adalah warna yang tepat. Dari skor kita, kita dapat berasumsi bahwa A di posisi 2 berada di tempatnya karena kita memiliki pasak hitam. Kita juga dapat berasumsi bahwa A berada di posisi 4, seperti yang telah disebutkan sebelumnya. Akhirnya, dua posisi lainnya akan digantikan oleh pasak E (Gambar 9.10).

			Asumsi:
4.	EAEA	(3,0)	_A_A
3.	DAAD	(1,1)	
2.	ABCC	(0,1)	
1.	AABB	(1,1)	

Gambar 9.10 Contoh 2—Tebakan 4

Tebakan ini sangat membantu. Kita sekarang sampai pada tiga kesimpulan:

1. D tidak termasuk dalam deret, karena skor kita meningkat, meskipun kita telah menghilangkannya.
2. Dengan demikian, dua A berada dalam deret dan berada di tempat yang tepat.
3. Satu E berada dalam deret, dan berada di posisi yang tepat.

Dengan F sebagai satu-satunya patok yang belum kita gunakan, kita biasanya menyimpulkan bahwa F adalah patok terakhir yang hilang. Namun, masih mungkin bahwa patok A ketiga merupakan bagian dari deret (kita belum membantahnya). Jadi untuk tebakan kita selanjutnya, mari kita asumsikan bahwa patok E yang benar adalah yang berada di posisi 1 dan posisi 3 berisi patok F (Gambar 9.11).

			Asumsi:
5.	EAEA	(4,0)*	E*F*
4.	EADA	(3,1)	
3.	DAAD	(1,1)	
2.	ABCC	(0,1)	
1.	AABB	(1,1)	

Gambar 9.11 Contoh 2—Tebakan 5

Dengan demikian, kita telah menemukan deret kita. Jika posisi 3 memiliki A lain, bukan F, kita hanya perlu satu tebakan tambahan untuk menentukannya.

Contoh 3

Deret tujuannya adalah EABC. Seperti biasa, tebakan pertama ditunjukkan pada Gambar 9.12.

1.	AABB	(2,0)

Gambar 9.12 Contoh 3—Tebakan 1

Kita mulai dengan skor 2, yang berarti mungkin diperlukan beberapa tebakan untuk menentukan A dan/atau B mana yang ada dalam deret tersebut. Untuk tebakan berikutnya, kita asumsikan bahwa pasak A dan pasak B berada dalam deret tersebut. Selain itu, karena kita memiliki dua pasak hitam, mari kita asumsikan bahwa pasak A berada di posisi 1 dan pasak B berada di posisi 3. Dua posisi sisanya adalah C (Gambar 9.13).

			Asumsi:
2.	ABCC	(2,1)	A_B_
1.	AABB	(2,0)	

Gambar 9.13 Contoh 3—Tebakan 2

Dengan tebakan ini, skor kita meningkat menjadi 3 dengan penambahan pasak putih. Karena peningkatan ini dari tebakan terakhir, kita dapat membuat kesimpulan yang jelas bahwa setidaknya ada satu pasak C dalam deret tersebut. Kita juga dapat menggunakan logika untuk menentukan kesimpulan yang kurang jelas.

Misalkan kita mempertahankan asumsi bahwa A dan B berada di posisi yang benar. Dengan ini, kita harus mengatakan bahwa ada satu pasak C dalam deret tersebut dan pasak tersebut mewakili pasak putih. Namun, ini tidak masuk akal. Berpikir bahwa menukar kedua pasak C akan menempatkan satu pasak di posisi yang benar tidaklah realistis. Dengan kata lain, menukar kedua pasak C tidak dapat tiba-tiba menghasilkan pasak hitam, karena tidak akan ada yang berubah. Dengan inferensi logis tunggal ini, kita dapat sampai pada empat kesimpulan:

1. Hanya ada satu pasak C dalam deret tersebut.
2. C mewakili pasak hitam dan berada di posisi yang benar.
3. Berdasarkan #1 dan #2 serta skor untuk tebakan ini, A dan B berada dalam deret tersebut.
4. Satu A atau B diwakili oleh pasak hitam, dan yang lainnya diwakili oleh pasak putih. Artinya, satu berada di posisi yang benar, dan yang lainnya berada di posisi yang salah.

Pertama, mari kita asumsikan pasak C berada di posisi 2. Kita tahu bahwa A dan B ada dalam deret tersebut, dan sekarang kita harus menentukan di mana pasak A dan B harus ditempatkan. Dengan menganalisis skor dari tebakan 1, yang memiliki dua pasak hitam, kita dapat mengasumsikan bahwa pasak A hanya dapat berada di posisi 1 atau 2, dan B hanya dapat berada di posisi 3 atau 4. Berdasarkan kesimpulan #4, satu A atau B akan tetap di tempatnya, sementara yang lain akan dipindahkan ke posisi berikutnya yang memungkinkan. Untuk tebakan kita selanjutnya, mari kita asumsikan A diwakili oleh pasak hitam dan berada di tempatnya, sedangkan B diwakili oleh pasak putih dan harus dipindahkan ke posisi berikutnya yang memungkinkan, yaitu 4. Terakhir, posisi 3 akan diberi pasak D, seperti yang ditunjukkan pada Gambar 9.14.

			Asumsi:
3.	ACDB	(0,3)	AC_B
2.	ACBC	(2,1)	
1.	AABB	(2,0)	

Gambar 9.14 Contoh 3—Tebakan 3

Kita tetap mendapatkan skor 3, tetapi sekarang setiap pasak berada di luar posisinya. Ini belum tentu kabar buruk, karena sekarang kita dapat membuat empat kesimpulan:

1. Pasak C berada di posisi 4.
2. Pasak B berada di posisi 3.
3. Pasak A berada di posisi 2.
4. Berdasarkan fakta bahwa skor tidak meningkat dari tebakan sebelumnya, pasak D tidak berada dalam deret tersebut.

Kesimpulan #1 hingga #3 mudah ditentukan karena pasak A, B, dan C hanya memiliki satu dari dua kemungkinan, seperti yang kita tentukan dari tebakan kedua. Karena kita salah memilih untuk ketiga pasak, sekarang kita tahu posisi yang benar untuk masing-masing pasak. Dengan informasi ini, kita sekarang dapat menyusun tebakan berikutnya. Posisi 2 akan memiliki pasak A, posisi 3 akan memiliki pasak B, dan posisi 4 akan memiliki pasak C. Posisi yang tersisa akan memiliki jenis pasak berikutnya yang belum kita gunakan, yaitu E (Gambar 9.15).

	Asumsi:

4.	CEDF	(4,0)*	_***
3.	CEDE	(0,3)	
2.	CCDD	(2,1)	
1.	AABB	(2,0)	

Gambar 9.15 Contoh 3—Tebak 4

Dan dengan demikian, kita telah menemukan deretnya.

Contoh 4

Urutan tujuannya adalah FBFF. Untuk contoh terakhir, kita memiliki tujuan yang agak tidak biasa, karena sekarang kita berurusan dengan tiga pasak dengan jenis yang sama.

Kita mulai dengan tebakan pertama kita yang biasa (Gambar 9.16).

1.	AABB	(0,1)

Gambar 9.16 Contoh 4—Tebak 1

Kita mulai dengan skor 1, dengan satu pasak yang tidak pada tempatnya. Kita asumsikan bahwa A ada dalam deret dan tidak pada tempatnya. Pertama-tama kita asumsikan A berada di posisi 3. Tiga posisi tersisa akan diisi dengan pasak C (Gambar 9.17).

			Asumsi
1.	CCAC	(0,0)	_ _A_
2.	AABB	(0,1)	

Gambar 9.17 Contoh 4—Tebakan 2

Dengan tebakan ini, kita dapat membuat dua kesimpulan: pasak A tidak ada dalam deret, jadi pasak B pasti ada, dan pasak C tidak ada dalam deret.

Mengingat tebakan 1 memiliki skor 1 dengan pasak putih, pasak B hanya dapat berada di posisi 1 atau 2. Pertama-tama, kita akan mengasumsikan bahwa pasak B berada di posisi 1. Posisi lainnya akan diberi pasak D (Gambar 9.18).

			Asumsi
3.	BDDD	(0,1)	B_ _ _
2.	CCAC	(0,0)	
1.	AABB	(0,1)	

Gambar 9.18 Contoh 4—Tebakan 3

Dengan tebakan ini, kita sampai pada dua kesimpulan: B berada di posisi 2, dan D tidak berada dalam deret.

Untuk tebakan berikutnya, kita akan menempatkan pasak B di posisi 2. Sisanya akan diberi pasak E (Gambar 9.19).

	Asumsi:

4.	EBEE	(1,0)	_B_ _
3.	BDDD	(0,1)	
2.	CCAC	(0,0)	
1.	AABB	(0,1)	

Gambar 9.19 Contoh 4—Tebakan 4

Kita sekarang memiliki pasak hitam, yang berarti kesimpulan kita benar dan pasak B sebenarnya terletak di posisi 2. Selain itu, kita juga dapat membuat dua kesimpulan lagi: pasak E tidak berada dalam urutan tersebut, dan posisi yang tersisa adalah pasak F, karena merupakan jenis pasak terakhir yang tersedia (Gambar 9.20).

			Asumsi:
5.	FBFF	(4,0)*	****
4.	EBEE	(1,0)	
3.	BDDD	(0,1)	
2.	CCAC	(0,0)	
4.	AABB	(0,1)	

Gambar 9.20 Contoh 4—Tebakan 5

Kita telah menemukan urutannya.

9.4 PEMECAHAN MASALAH MANUSIA

Permainan Mastermind didistribusikan secara digital kepada mahasiswa untuk menganalisis bagaimana rata-rata orang mendekati permainan dan metode yang digunakan untuk menemukan solusinya. Permainan ini didistribusikan kepada sembilan mahasiswa, dan tidak memiliki aturan asli. Dalam versi ini, terdapat delapan warna, dan tidak diperbolehkan adanya warna duplikat.

Untuk tebakan pertama, dua mahasiswa memilih empat warna pertama yang disajikan sebagai pilihan, sementara sisanya memilih empat warna acak. Namun, kita tidak dapat menentukan apakah ada mahasiswa yang akan memilih tebakan AABB jika mereka memainkan permainan aslinya, meskipun mereka mungkin tetap memilih empat warna pertama.

Terlepas dari apa yang dipilih mahasiswa sebagai tebakan pertama mereka, semua mahasiswa cukup berhasil dalam permainan. Tujuh mahasiswa menggunakan deduksi dan logika yang tepat untuk menemukan pasak yang tepat dan urutannya. Satu mahasiswa mampu menemukan pasak yang benar tetapi kesulitan menentukan urutannya yang benar. Akhirnya, siswa ini berhasil menemukan deret yang benar dalam tujuh tebakan. Pengecualian lainnya berhasil menyimpulkan urutan pasak yang benar, tetapi kurang berhasil dalam menentukan pasak yang benar. Siswa ini menggunakan pasak yang sama yang dapat dinalar keluar dari deret akhir pada tebakan sebelumnya. Akibatnya, siswa ini membutuhkan sembilan tebakan untuk menemukan deret yang benar.

Rata-rata, kita dapat melihat bahwa para siswa telah menyelesaikan deret tersebut dengan jumlah tebakan yang cukup sedikit. Satu siswa menyelesaikan deret tersebut dalam empat tebakan, tiga siswa dalam lima tebakan, dua siswa dalam enam tebakan, dua siswa dalam tujuh tebakan, dan satu siswa dalam sembilan tebakan. Modusnya adalah lima tebakan, yang juga merupakan rata-rata jumlah tebakan yang diperlukan untuk menyelesaikan solusi yang disajikan di Bagian 9.3.

9.5 ANALISIS JENDELA MANUSIA TERHADAP SOLUSI

Sayangnya, data untuk analisis Jendela Manusia untuk masalah ini tidak mencukupi. Selain itu, tidak ditemukan cukup solusi yang dapat digunakan untuk membantu analisis ini. Yang kami temukan hanyalah deskripsi kata untuk strategi umum dalam menyelesaikan Mastermind, tetapi tidak ada solusi konkret. Hal ini mungkin karena, seperti yang telah disebutkan sebelumnya, permasalahan ini tidak memiliki tujuan spesifik, sehingga hanya kiat dan strategi umum yang dapat digunakan untuk membantu menemukan tujuan tersebut. Namun, hal ini memberi kami kesempatan untuk menggunakan analisis Jendela Manusia dari permasalahan lain dalam buku ini untuk menentukan solusi seperti apa yang dapat dianggap sebagai MHWC untuk Mastermind.

Karena permasalahan Mastermind tidak memiliki satu tujuan spesifik yang ingin dicapai, sebaiknya gunakan solusi umum yang intensional, karena ini akan membantu menemukan tujuan akhir dari permainan Mastermind, apa pun tujuannya. Seperti "Solusi Ubin", dikombinasikan dengan contoh-contoh seperti yang ada di Bagian 9.3, pendekatan serupa dapat diterapkan dengan solusi umum untuk Mastermind. Solusi ini dapat berisi satu atau lebih contoh yang menunjukkan metode langkah demi langkah untuk menyelesaikan urutan tujuan tertentu. Lebih lanjut, harus ada cukup contoh agar setiap skenario yang mungkin terjadi dapat dijelaskan. Hal ini akan memberikan solusi peringkat eksekusi yang tinggi. Selain itu, untuk meningkatkan pemahaman solusi, gambar permainan Mastermind yang sebenarnya dapat digunakan di setiap langkah untuk memberikan gambaran seperti apa seharusnya permainan tersebut.

9.6 SOLUSI MESIN TERBAIK

Solusi terbaik untuk permainan Mastermind untuk sebuah mesin sejauh ini dikenal sebagai *Algoritma Lima Tebakan*, yang dikembangkan dan dipresentasikan pada tahun 1977 dalam sebuah makalah berjudul "Komputer sebagai Mastermind" oleh Donald Knuth. Knuth mendasarkan algoritmanya pada sebuah studi yang ia lakukan terhadap permainan Mastermind asli. Algoritma ini disebut Algoritma Lima Tebakan karena menjamin bahwa setiap solusi dapat ditemukan dalam lima tebakan atau kurang.

Algoritma ini cukup canggih dan rumit. Berikut penjelasan singkatnya:

1. Buat himpunan P dari kemungkinan yang tersisa. Pada awalnya, P selalu 1.296.
2. Buat tebakan pertama AABB (tebakan pertama seperti yang digunakan pada Bagian 9.3). Hapus semua kemungkinan dari P yang akan memberikan skor lebih rendah jika merupakan jawabannya. 3. Untuk setiap 1.296 kemungkinan tebakan, hitung berapa

banyak kemungkinan dari P yang akan dihilangkan untuk setiap kemungkinan skor. Skor tebakan adalah yang terkecil dari nilai-nilai tersebut.

3. Mainkan tebakan dengan skor tertinggi.
4. Ulangi langkah #2 hingga #4 hingga urutannya ditemukan.

Kami mengulang catatan sejarah dari Knuth :

Sebuah permainan yang sangat mirip dengan Mastermind, yang disebut "Bulls and Cows", telah populer di Inggris selama bertahun-tahun. Perbedaannya adalah semua digit kode dalam Bulls and Cows harus berbeda, tetapi digit 0 hingga 9 diperbolehkan. Versi permainan ini menjadi demonstrasi komputer yang populer setelah Frank H. King memperkenalkan programnya pada bulan Agustus 1968 di Universitas Cambridge.

F.H. King, The Game of M00, Universitas Cambridge, Laboratorium Komputer, memorandum tertanggal Februari 1976.

9.7 MASALAH TERKAIT

Seperti yang telah disebutkan sebelumnya, salah satu permainan yang sangat erat kaitannya dengan Mastermind adalah Bulls and Cows. Permainan ini dapat dimainkan dengan angka 4 digit dari 0-9, atau dapat dimainkan dengan kata 4 huruf yang valid. Versi angka dijelaskan di sini, tetapi idenya sama dengan versi kata. Pemilih akan menuliskan angka 4 digit, dan penebak akan mencoba menebak angka tersebut. Penebak diberikan umpan balik setelah setiap tebakan, banteng menunjukkan bahwa angka tersebut benar dan berada di posisi yang tepat, sementara sapi berarti angka tersebut benar tetapi berada di posisi yang berbeda (salah). Permainan berakhir ketika penebak menebak angka yang benar. Premisnya sama dengan Mastermind, kecuali bahwa ini adalah permainan yang dimainkan dengan angka, atau huruf. Solusi untuk permainan ini sangat mirip dengan Mastermind.

9.8 PROGRAM YANG DAPAT DIMAINKAN

Game Mastermind yang dapat dimainkan tersedia di http://www.gray-man.de/mastermind/index_en.php4. Untungnya, game ini mengikuti aturan permainan aslinya, jadi mengikuti contoh di Bagian 9.3 dapat membantu. Selain itu, pasak-pasak yang berbeda tidak hanya diwarnai secara individual, tetapi juga diberi nomor tambahan.

BAB 10
PERMASALAHAN MONTY HALL

10.1 PENDAHULUAN

Permasalahan Monty Hall adalah salah satu permasalahan yang paling terkenal dan banyak dipelajari dalam bidang teka-teki. Permasalahan ini merupakan teka-teki berbasis probabilitas di mana peluang hasil yang menguntungkan harus dibandingkan sebelum dan sesudah suatu peristiwa terjadi, dan kemudian situasi yang lebih baik dari kedua situasi tersebut harus ditentukan.

Permasalahannya adalah sebagai berikut: Dalam sebuah acara kuis, pembawa acara meminta seorang kontestan untuk memilih satu dari tiga pintu, dengan asumsi salah satu pintu memiliki mobil di belakangnya. Masing-masing dari dua pintu lainnya memiliki seekor kambing di belakangnya. Setelah kontestan memilih sebuah pintu, pembawa acara membuka salah satu dari dua pintu lainnya untuk menunjukkan seekor kambing di belakangnya. Kontestan kemudian ditanya apakah ia ingin beralih ke pintu yang belum dibuka atau tetap pada pilihan awalnya. Menurut Anda, apakah kontestan memiliki pilihan menang yang lebih baik jika ia beralih?

Awalnya, tampaknya pergantian pintu tidak berpengaruh karena setelah pembawa acara membuka pintu dengan seekor kambing, kita masih berasumsi bahwa peluang pintu-pintu yang tersisa yang belum dibuka memiliki mobil adalah ½, peluang yang sama dengan pilihan awal. Karena peluang menang tampak sama untuk kedua pintu yang tersisa, tampaknya tidak ada manfaat dalam pergantian pintu.

Anehnya, jawaban yang benar adalah kontestan memiliki peluang lebih baik untuk memenangkan mobil jika ia beralih ke pintu yang lain. Kami akan menjelaskan mengapa hal ini benar di bagian selanjutnya.

10.2 TEKNIK PEMECAHAN MASALAH

Jelas bahwa salah satu dari dua pilihan tersebut hanya memberikan peluang atau probabilitas untuk memenangkan mobil dan tidak menjamin kemenangan. Oleh karena itu, analisis berbasis probabilitas adalah kunci untuk memecahkan masalah ini. Selain itu, masalah ini melibatkan pengambilan keputusan. Awalnya, kontestan harus memilih sebuah pintu, yang kemungkinan besar merupakan pilihan acak karena ketiga pintu memiliki probabilitas yang sama (1/3) untuk memiliki mobil di belakangnya. Namun setelah terungkap bahwa salah satu pintu memiliki seekor kambing, situasinya berubah. Kontestan kini memiliki lebih banyak informasi daripada ketika ia memilih satu dari tiga pintu. Karena keputusan akhir kontestan untuk tetap memilih atau mengganti pintu yang dipilih bergantung pada pilihan awal dan tindakan tuan rumah, kami membuat pohon keputusan untuk menganalisis masalah tersebut. Versi tabel dari pohon keputusan ditunjukkan dalam **Error! Sumber referensi tidak ditemukan**.

Selain itu, karena masalah ini memiliki ruang keadaan yang kecil dalam hal semua kemungkinan hasil dari kombinasi keputusan awal kontestan, pilihan pintu tuan rumah, dan keputusan akhir kontestan, kami dapat melakukan enumerasi lengkap dari ruang keadaan solusi seperti yang ditunjukkan dalam **Error! Sumber referensi tidak ditemukan**.

Error! Sumber referensi tidak ditemukan. menunjukkan semua kemungkinan skenario untuk teka-teki ini. Angka 1, 2, dan 3 mewakili tiga pintu. Kami mempertimbangkan tiga kemungkinan mobil berada di belakang setiap pintu satu per satu. Dalam masing-masing tiga kasus yang disebutkan di atas, kami selanjutnya mempertimbangkan tiga kasus atau kemungkinan untuk pilihan awal.

PINTU DENGAN MOBIL	PILIHAN AWAL	BENAR/ SALAH	PILIHAN HOST	PILIHAN AKHIR	HASIL
1	1	Benar	2 atau 3	Tetap	Menang
				Pindah	Kalah
	2	Salah	3	Tetap	Kalah
				Pindah	Menang
	3	Salah	2	Tetap	Kalah
				Pindah	Menang
2	1	Salah	3	Tetap	Kalah
				Pindah	Menang
	2	Benar	1 atau 3	Tetap	Menang
				Pindah	Kalah
	3	Salah	1	Tetap	Kalah
				Pindah	Menang
3	1	Salah	2	Tetap	Kalah
				Pindah	Menang
	2	Salah	1	Tetap	Kalah
				Pindah	Menang
	3	Benar	1 atau 2	Tetap	Menang
				Pindah	Kalah

Tabel 10.1 MASALAH MONTY HALL

Untuk setiap kemungkinan pilihan awal, pembawa acara akan memilih pintu yang memiliki kambing, sehingga mengurangi jumlah pintu yang mungkin memiliki mobil menjadi hanya dua, bukan tiga. Hal ini membuat kontestan memiliki dua pilihan: tetap pada pilihan awal atau beralih ke pintu yang tidak dibuka oleh pembawa acara.

Pilihan akhir yang dibuat oleh kontestan dan hasil selanjutnya ditunjukkan pada dua kolom terakhir. Jelas bahwa tetap pada pilihan awal menjamin kemenangan jika pilihan awal benar. Dan probabilitas pilihan awal benar adalah 1/3 karena hanya satu dari tiga pintu yang memiliki mobil di belakangnya. Sebaliknya, probabilitas pilihan awal salah adalah 2/3, jadi dalam hal ini beralih mengarah pada peluang menang yang lebih baik.

Kita bisa melihatnya dengan cara lain. Katakanlah, alih-alih 3 pintu, terdapat 10 pintu dengan mobil di belakang 1 pintu dan kambing di belakang 9 pintu. Kontestan memilih sebuah pintu, dan pembawa acara mengungkapkan 8 pintu yang memiliki kambing. Dalam hal ini, apakah kontestan harus tetap pada pilihan awalnya atau beralih?

Jelas bahwa pada awalnya, probabilitas membuka pintu yang benar adalah 1/10, yang juga berarti probabilitas membuka pintu dengan kambing adalah 9/10. Karena kita lebih mungkin membuka pintu dengan kambing pada awalnya, beralih lebih mungkin menghasilkan kemenangan—dalam hal ini, peluang menangnya adalah 1/2.

10.3 SOLUSI MONTY HALL

Solusi seperti yang dijelaskan di bagian sebelumnya adalah Anda memiliki peluang menang lebih besar jika beralih. Ini karena Anda memiliki peluang dua kali lipat untuk mendapatkan pintu yang salah dengan pilihan awal Anda. Tuan rumah selalu membuka pintu (yang tidak Anda pilih) dengan seekor kambing di belakangnya (seperti yang dijelaskan dalam soal). Jadi setelah kejadian ini terjadi, kita tahu mobil berada di belakang salah satu dari dua pintu lainnya dan dengan demikian ada peluang 50% mobil berada di belakang salah satu dari dua pintu lainnya. Jadi, tetap menggunakan pilihan awal berarti peluang menang $1/3 = 33{,}3\%$, dan beralih berarti peluang menang $2/3 = 66\%$. Hal ini ditunjukkan dengan jelas dalam **Error! Sumber referensi tidak ditemukan**.

10.4 PEMECAHAN MASALAH MANUSIA

Masalah Monty Hall pertama kali muncul di kolom Marilyn vos Savant di *Majalah Parade*. Vos Savant, yang menjawab pertanyaan dan memecahkan teka-teki yang dikirimkan pembaca, pernah tercatat dalam *Guinness Book of World* Records sebagai orang dengan IQ tertinggi di dunia.

Soal ini menjadi sangat populer karena terinspirasi dari acara permainan populer *Let's Make a Deal* dan dinamai sesuai nama pembawa acaranya, Monty Hall. Menanggapi pertanyaan apakah lebih baik tetap menggunakan pilihan awal atau beralih, Vos Savant menyatakan bahwa beralih adalah strategi yang lebih baik. Jawabannya memicu sekitar 10.000 tanggapan, termasuk dari banyak doktor, yang bersikeras bahwa jawabannya salah. Kisah ini menarik begitu banyak perhatian hingga dimuat di *New York Times*.

10.5 MASALAH TERKAIT

Sebuah soal yang berkaitan erat dengan Masalah Monty Hall adalah "Paradoks Kotak" karya Bertrand Russell. Berikut pernyataan soalnya: Terdapat 3 kotak identik yang masing-masing memiliki 2 laci. Satu kotak berisi koin emas di setiap laci, kotak lainnya memiliki koin emas di satu laci dan koin perak di laci lainnya, sementara kotak terakhir memiliki koin perak di setiap laci. Jika sebuah kotak dipilih secara acak dan sebuah laci terbuka dan sebuah koin emas ditemukan, temukan probabilitas laci lainnya juga berisi koin emas. Jawabannya sama dengan Masalah Monty Hall, 2/3. Satu penjelasan yang mudah dipahami adalah bahwa ada 2 koin emas dan 1 koin perak yang dapat dipilih karena hanya ada 2 kotak berisi koin emas, sehingga probabilitasnya adalah emas adalah 2/3.

Masalah lain yang terkait dengan Masalah Monty Hall adalah "Masalah Tiga Tahanan." Situasinya adalah bahwa tiga tahanan berada di penjara menunggu eksekusi dan salah satu dari mereka akan diampuni. Penjaga tahu siapa yang akan diampuni tetapi tidak diizinkan untuk mengatakan siapa. Tahanan A meminta penjaga untuk memberi tahu dia yang mana dari 2 tahanan yang tersisa yang akan dieksekusi karena hanya 1 tahanan yang dapat diampuni. Penjaga menjawab B akan dieksekusi dan A memberi tahu C tentang hal itu. Tentukan probabilitas bahwa C adalah orang yang akan diampuni. Jawabannya sama dengan Masalah Monty Hall, 2/3. Alasannya sebenarnya juga sama dengan Masalah Monty Hall.

BAB 11
KUBUS RUBIK

11.1 PENDAHULUAN

Kubus Rubik adalah salah satu teka-teki paling populer sepanjang masa. Kubus ini merupakan teka-teki kombinasi tiga dimensi yang diciptakan pada tahun 1974 oleh Ernö Rubik, seorang arsitek dan profesor Hungaria. Kubus ini terdiri dari 26 kubus yang lebih kecil. Ketika mempertimbangkan salah satu kubus kecil ini sebagai satu unit, kita dapat mengatakan bahwa keseluruhan kubus memiliki dimensi $3'' \times 3'' \times 3''$, dengan 1 kubus kecil hilang dari pusatnya. Bersama-sama, 9 kubus kecil yang membentuk salah satu sisi keseluruhan kubus dapat diubah menjadi "lapisan" yang memiliki kemampuan untuk berputar sambil tetap pada posisinya. Rubik menciptakan kubus tersebut dalam upaya untuk memahami bagaimana balok dapat bergerak secara independen tanpa [seluruh struktur] berantakan.

Ketika kubus tersebut diurai, masing-masing dari enam sisinya akan memiliki satu warna. Inilah keadaan tujuan kubus tersebut. Untuk mencapai keadaan ini, setiap kubus yang lebih kecil harus dipindahkan ke posisi tertentu. Namun, hal ini tidak semudah kedengarannya. Sebuah kubus kecil tidak dapat dipindahkan secara individual. Seluruh lapisan kubus harus diputar untuk memposisikannya dengan benar. Sayangnya, hal ini pasti akan memposisikan ulang kubus-kubus kecil lainnya, meskipun kubus-kubus tersebut sudah berada di posisi yang benar. Oleh karena itu, algoritma khusus harus digunakan untuk memposisikan kubus-kubus yang lebih kecil dengan benar tanpa memindahkan kubus lainnya. tidak pada tempatnya. Setelah mengacak model tersebut, Rubik menghabiskan waktu sebulan untuk mencoba-coba memecahkan teka-teki itu sendiri.

Pada tahun 1975, Rubik mengajukan paten untuk karyanya dan mulai mencari mitra untuk membantunya memasarkannya. Ia menandatangani kontrak dengan produsen mainan Hungaria, Politechnika, untuk memproduksi kubus tersebut secara massal. Pada tahun 1977, versi pertama kubus tersebut muncul di toko-toko mainan di Budapest dengan nama *Büvös Kocka*, yang juga dikenal sebagai Magic Cube. Akhirnya, kubus tersebut menjadi populer di seluruh Hungaria. Pada tahun 1980, kubus tersebut dibawa ke pasar internasional dan dijual dengan nama paten, Rubik's Cube, yang membuatnya terkenal di seluruh dunia. Saat ini, teka-teki tersebut dimiliki oleh Rubik's Brand Ltd. dan masih sangat populer. Para pemecah teka-teki Rubik's Cube, atau "kubis," masih berlatih dan meneliti teka-teki tersebut hingga hari ini. Beberapa dari mereka begitu berdedikasi sehingga mereka mampu memecahkan rekor dengan memecahkan kubus tersebut dalam hitungan detik.

Kubus Rubik asli berukuran $3'' \times 3'' \times 3''$. Sejak saat itu, beberapa variasi telah diperkenalkan. Dua variasi yang paling populer adalah Rubik's Revenge (juga dikenal sebagai Master Cube), yang merupakan kubus berukuran $4'' \times 4'' \times 4''$, dan Professor's Cube, yang merupakan kubus berukuran $5'' \times 5'' \times 5''$.

11.2 TEKNIK PEMECAHAN MASALAH

Teknik pemecahan masalah yang paling berharga untuk Rubik's Cube adalah *memecahkan subtujuan*. Ada banyak metode untuk memecahkan Rubik's Cube dan masih banyak lagi yang menunggu untuk ditemukan. Namun, terlepas dari langkah-langkah yang terdapat dalam metode tersebut dan seberapa berbedanya satu sama lain, metode-metode tersebut kemungkinan besar akan membutuhkan penggunaan subtujuan dalam penyelesaiannya. Secara spesifik, subtujuan adalah menempatkan sekumpulan kubus kecil tertentu pada posisi yang tepat.

Seperti yang telah disebutkan sebelumnya, terlepas dari apakah kubus-kubus tersebut berada pada posisi yang tepat atau tidak, tidak dapat dihindari bahwa kubus-kubus yang lebih kecil akan dipindahkan setelah memutar lapisan-lapisannya dalam upaya untuk memposisikan kubus-kubus kecil lainnya dengan benar. Untuk mengatasi hal ini, "algoritma" tertentu harus digunakan. Dalam konteks Rubik's Cube, algoritma mengacu pada urutan rotasi lapisan yang memindahkan kubus-kubus kecil tertentu ke lokasi tertentu dan pada akhirnya tidak memindahkan kubus-kubus kecil lainnya yang telah diposisikan dengan benar.

Secara umum, solusi untuk Rubik’s Cube dapat dilakukan sebagai berikut:

1. Gunakan algoritma untuk mengatur sekumpulan kubus kecil tertentu ke posisi yang benar.
2. Gunakan algoritma lain untuk mengatur sekumpulan kubus kecil lainnya ke posisi yang benar tanpa memindahkan kubus yang telah disusun pada langkah sebelumnya. (Tidak masalah jika kubus dipindahkan selama algoritma, asalkan kubus dikembalikan ke posisi yang benar setelah algoritma selesai.)
3. Ulangi langkah 2 hingga setiap kubus yang lebih kecil tersusun pada posisi yang benar dan kubus terpecahkan.

Lebih dari satu algoritma dapat digunakan pada subtujuan tertentu—dengan kata lain, satu algoritma dapat digunakan untuk mengatur ulang orientasi kubus yang lebih kecil, sementara algoritma lain memposisikannya dengan benar.

11.3 SOLUSI DENGAN KUBUS RUBIK

Meskipun ada banyak cara untuk menyelesaikan Kubus Rubik standar 3” × 3” × 3”, bagian ini menjelaskan salah satu metode termudah dan terpopuler. Metode ini menggunakan tujuh subtujuan untuk menyelesaikan kubus tersebut. Namun, sebelum kita mulai, penting untuk memberikan beberapa notasi.

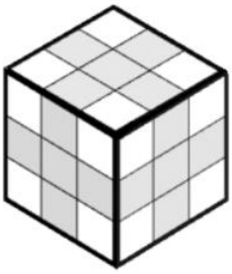
Komponen Kubus

Bagian-Bagian

Seperti yang telah disebutkan sebelumnya, Kubus Rubik terdiri dari 26 kubus yang lebih kecil. Namun, kubus-kubus tersebut tidak identik, karena terdapat tiga jenis kubus yang lebih kecil.

Sudut (berwarna putih) (Gambar 11.1)

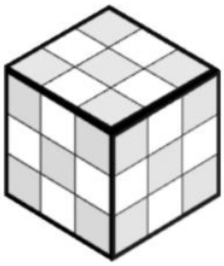
Terdapat total 8 kubus sudut, masing-masing terdiri dari tiga warna.



Gambar 11.1 Potongan Sudut

Tepi (Gambar 11.2)

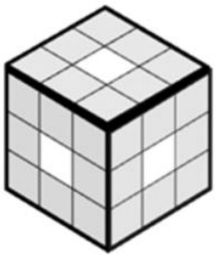
Terdapat total 12 potongan tepi, masing-masing terdiri dari dua warna.



Gambar 11.2 Potongan Tepi

Pusat (Gambar 11.3)

Terdapat total 6 potongan tengah, masing-masing terdiri dari satu warna. Potongan tengah merupakan potongan yang sangat penting untuk diperhatikan saat menyelesaikan kubus, karena potongan ini mengidentifikasi warna sisi tertentu. Hal ini karena potongan tengah tidak akan pernah berubah posisi relatif terhadap potongan tengah lainnya.



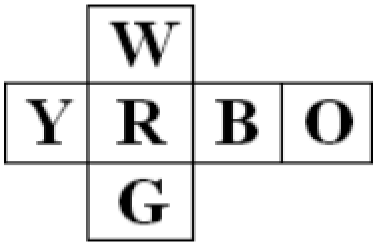
Gambar 11.3 Potongan Tengah

Untuk solusi ini, kita akan menamai setiap kubus kecil berdasarkan warna dan jenisnya. Selain itu, kita akan menyebutnya sebagai potongan, bukan kubus kecil. Misalnya, potongan tepi terdiri dari warna merah dan biru. Kita akan menyebutnya potongan tepi Merah-Biru.

Sisi (Gambar 11.4)

Layaknya kubus pada umumnya, terdapat total enam sisi (atau sisi) pada Kubus Rubik. Setiap sisi diidentifikasi dengan warna yang berbeda, dan warna tersebut ditentukan oleh warna kubus kecil di tengahnya. Konfigurasi warna dapat berbeda dari satu kubus ke kubus lainnya, jadi untuk solusi di sini, kita menggunakan tampilan dua dimensi kubus dan warnanya:

R = Merah
W = Putih
Y = Kuning
B = Biru
G = Hijau
O = Oranye

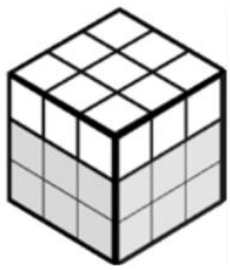


Gambar 11.4 Sisi Kubus

Lapisan

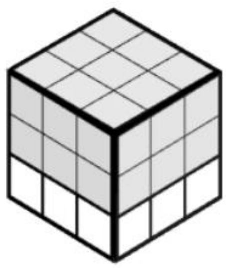
Yang dimaksud dengan *lapisan* adalah bagian-bagian kubus yang dapat diputar. Satu lapisan terdiri dari 9 kubus kecil yang dapat diputar hingga 360 derajat secara dua dimensi. Kubus memiliki total enam lapisan, dan masing-masing akan dilambangkan dengan satu huruf:

Lapisan Atas (atau Atas) = **U** (Gambar 11.5)



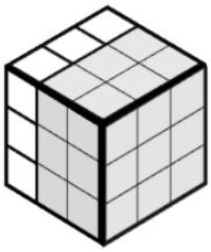
Gambar 11.5 Lapisan Atas

Lapisan Bawah (atau Bawah) = **D** (Gambar 11.6)



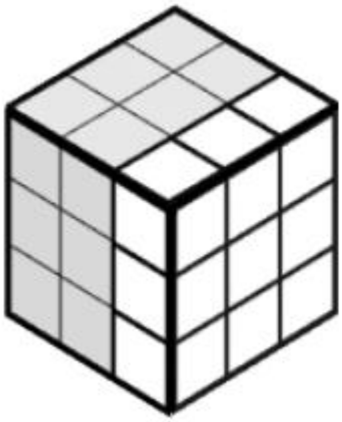
Gambar 11.6 Lapisan Bawah

Lapisan Kiri = **L** (Gambar 11.7)



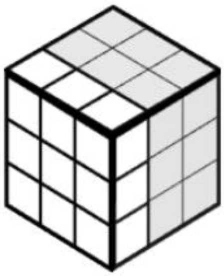
Gambar 11.7 Lapisan Kiri

Lapisan Kanan = **R** (Gambar 11.8)



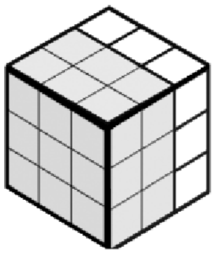
Gambar 11.8 Lapisan Kanan

Lapisan Depan = **F** (Gambar 11.9)



Gambar 11.9 Lapisan Depan

Lapisan Belakang = **B** (Gambar 11.10)



Gambar 11.10 Lapisan Belakang

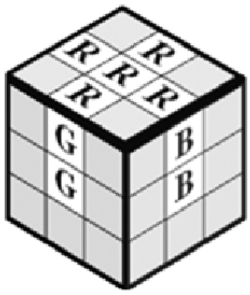
Layer-layer diberi nama relatif terhadap sisi kubus yang dianggap sebagai sisi depan dan atas. Misalnya, jika kita memilih Sisi Putih sebagai Sisi Depan dan Sisi Merah sebagai Sisi Atas, maka layer yang dianggap sebagai Sisi Kiri akan berbeda dari layer yang dianggap sebagai Sisi Kiri, misalnya, jika Sisi Biru dianggap sebagai Sisi Depan.

Denotasi ini akan digunakan untuk menunjukkan bagaimana layer tertentu akan diputar dalam suatu algoritma. Ketika sebuah huruf ditampilkan, artinya layer yang bersangkutan akan diputar 90 derajat searah jarum jam. Lebih lanjut, tanda apostrof (') setelah huruf tersebut menunjukkan bahwa layer yang sama akan diputar 90 derajat berlawanan arah jarum jam. Rangkaian huruf-huruf ini akan membentuk sebuah algoritma, dan harus diikuti sesuai urutan yang diberikan. Berikut adalah contoh algoritma:

U D F'

Algoritma ini menunjukkan bahwa Layer Atas harus digerakkan searah jarum jam, kemudian Layer Bawah, dan terakhir Layer Depan harus digerakkan berlawanan arah jarum jam. Anda akan melihat bahwa semakin banyak kubus kecil yang ditempatkan, algoritma yang digunakan untuk menyelesaikan kubus yang tersisa menjadi lebih kompleks. Untuk memulai solusi ini, kita pilih Sisi Merah sebagai Sisi Atas. Sisi Depan akan ditentukan nanti.

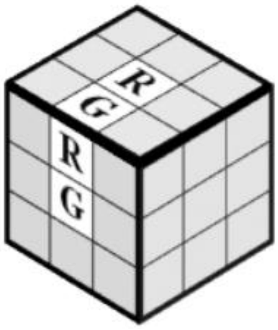
Subtujuan 1: Persilangan Atas



Gambar 11.11 Subtujuan 1

Subtujuan pertama (Gambar 11.11) adalah menempatkan 4 kubus tepi teratas pada posisi yang tepat. Hasilnya, Sisi Atas akan terlihat seperti tanda silang merah.

Mencapai konfigurasi ini cukup mudah. Kasus terburuk terjadi ketika sebuah keping tepi berada di sisi yang salah. Misalnya, mari kita asumsikan bahwa tepi Merah-Hijau adalah bagian dari Sisi Biru. Kita ingin meletakkannya pada posisi yang tepat di Sisi Hijau. Hal pertama yang harus dilakukan adalah memutar lapisan yang berisi Sisi Biru hingga tepi ini berada di Lapisan Bawah. Selanjutnya, putar Lapisan Bawah hingga keping tepi ini berada pada Sisi Hijau yang sesuai. Terakhir, putar lapisan dengan Sisi Hijau dua kali, dan tepi Merah-Hijau seharusnya berada pada posisi yang tepat. Bisa saja tepi Merah-Biru berada pada posisi yang tepat sejak awal, dan gerakan kita menggesernya. Untungnya, mudah untuk mengembalikan tepi Merah-Biru ke tempatnya. Setelah memutar layer dengan Sisi Biru secukupnya hingga tepi Merah-Hijau berada di Lapisan Bawah, putar Lapisan Bawah sekali sehingga tepi Merah-Hijau menjauh dari layer dengan Sisi Biru, lalu putar layer tersebut hingga Sisi Merah-Biru kembali ke tempatnya. Anda mungkin akan mendapatkan Gambar 11.12.



Gambar 11.12 Tepi Terbalik

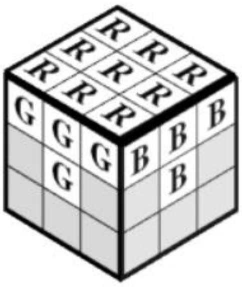
Perhatikan masalah ini. Tepi Merah-Hijau memiliki warna yang terbalik. Untuk memperbaikinya, algoritma tertentu harus digunakan.

Pertama, anggap Sisi Depan sebagai Sisi Hijau, dengan Sisi Atas tetap sebagai Sisi Merah. Selanjutnya, lakukan gerakan berikut:

F’ U L’ U’

Algoritma ini akan mengubah orientasi tepi Merah-Hijau ke konfigurasi yang benar, sambil tetap menjaga tiga tepi atas lainnya tetap pada tempatnya.

Subtujuan 2: Sudut Atas



Gambar 11.13 Subtujuan 2

Subtujuan pada Gambar 11.13 tidak hanya akan menempatkan 4 bagian sudut atas di tempat yang tepat, tetapi juga akan melengkapi Red Face dan seluruh Top Layer!

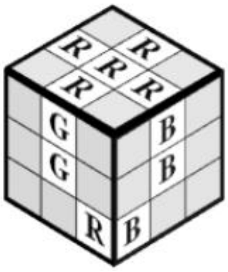
Misalnya kita ingin memulai dengan menempatkan sudut Merah-Biru-Hijau di tempat yang tepat. Langkah pertama kita adalah membawa sudut tersebut ke bawah jika belum ada. Pertama, temukan salah satu sisi tempat sudut Merah-Biru-Hijau berada dan sebut itu Sisi Depan. Dalam kasus kita, katakanlah terletak di antara Yellow Face dan White Face. Kita dapat memilih Front Face sebagai Yellow Face atau White Face. Mari kita pilih White Face sebagai Front Face. Sudut Merah-Hijau-Biru sekarang dapat ditempatkan di sudut kiri atas Front Face. Untuk mencapai titik terendah tanpa menggeser tepi Merah-Putih, jalankan algoritma ini:

F D’ F’

Jika kita memilih Sisi Depan sebagai Sisi Kuning, sudut Merah-Hijau-Biru akan berada di sisi kiri atas, dan kita harus menjalankan algoritma ini:

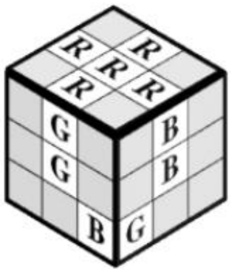
F’ D F

Algoritma mana pun yang dijalankan, sudut target kita sekarang seharusnya berada di Lapisan Bawah, dan tepi yang kita posisikan di sub-tujuan 1 seharusnya tetap berada di Lapisan Atas. Putar Lapisan Bawah hingga bagian sudut berada di antara Sisi Hijau dan Biru. Pada titik ini, kita dapat memiliki salah satu dari tiga kemungkinan keadaan yang ditunjukkan pada Gambar 11.14 hingga Gambar 11.16



Gambar 11.14 keadaan 1

Gambar 11.15 keadaan 2



Gambar 11.16 keadaan 3

Keadaan 1: Jadikan Muka Depan yang baru sebagai muka yang mengandung warna yang tepat (yang dalam kasus ini adalah Muka Biru). Sudut target kita sekarang seharusnya menjadi bagian dari sudut kiri bawah Muka Depan. Kemudian jalankan algoritma berikut:

L D’ L’

Tahap 2: Seperti pada Tahap 1, buatlah Permukaan Depan yang baru dengan warna yang tepat (dalam hal ini adalah Permukaan Hijau). Namun, sudut target sekarang seharusnya berada di sudut kanan bawah Permukaan Depan. Kemudian, jalankan algoritma berikut:

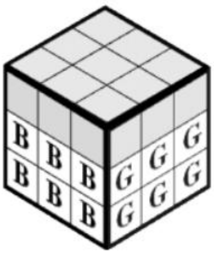
R’ D R

Tahap 3: Dalam orientasi ini, kita tidak dapat menempatkan sudut Merah-Hijau-Biru dengan tepat pada posisinya. Kita perlu mengubah orientasi sudut tersebut agar sesuai dengan Tahap 1 atau Tahap 2. Mari kita ubah ke Tahap 1. Pertama, pilih Permukaan Depan baru yang akan membuat sudut target berada di sudut kiri bawahnya (dalam hal ini Permukaan Hijau). Kemudian, jalankan algoritma berikut:

L D’D’ L’ D

Setelah melakukan ini, bagian sudut akan kembali berada di antara Permukaan Hijau dan Biru dan akan berorientasi seperti yang terlihat pada Tahap 1. Sekarang, cukup jalankan algoritma Tahap 1. Lakukan metode ini untuk keempat sudut, dan Anda akan mencapai subtujuan 2.

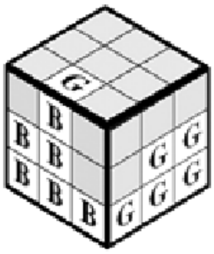
Subtujuan 3: Lapisan Tengah



Gambar 11.17 Subtujuan 3

Subtujuan 3 (Gambar 11.17) menempatkan keempat kubus sisi tengah pada posisi yang tepat. Di akhir subtujuan ini, lapisan tengah Kubus Rubik akan terpecahkan sepenuhnya. Sebelum kita mulai, penting untuk membalik kubus tersebut. Oleh karena itu, Lapisan Atas yang baru adalah Sisi Oranye, yang sekarang menjadikan Sisi Merah sebagai Lapisan Bawah.

Pertama, cari tepi pada Lapisan Atas yang tidak memiliki warna yang sama dengan sisi atas (dalam hal ini Oranye). Katakanlah kita menemukan tepi Biru-Hijau. Katakanlah juga bahwa warna vertikal tepi ini adalah Biru, sedangkan warna horizontalnya adalah Hijau. Putar Lapisan Atas hingga sisi vertikal potongan tepi tersebut sesuai dengan sisinya. Dalam hal ini, putar Lapisan Atas hingga tepi Biru-Hijau berada di Sisi Biru, yang sekarang akan dianggap sebagai Sisi Depan. Kubus sekarang akan terlihat seperti Gambar 11.18.



Gambar 11.18 Kondisi Pra-Algoritma

Perhatikan warna teratas dari potongan tepi ini. Dari Sisi Depan, lihat warna Sisi Kiri dan Kanan. Warna yang cocok dengan warna teratas tepi target akan menentukan algoritma mana yang akan digunakan selanjutnya. Dalam hal ini, warna Hijau di atas tepi Biru-Hijau cocok dengan Sisi Hijau, yang terletak di Lapisan Kanan. Oleh karena itu, gunakan algoritma berikut:

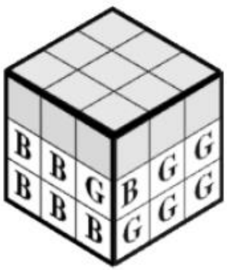
U R U’ R’ U’ F’ U F

Setelah menjalankan algoritma ini, tepi Biru-Hijau akan ditempatkan pada posisi yang sesuai di baris tengah.

Atau, jika Sisi Hijau berada di Lapisan Kiri, kita akan menggunakan algoritma ini:

U’ L’ U L U F U’ F’

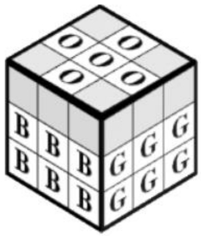
Salah satu masalah yang mungkin terjadi adalah potongan tepi akan ditempatkan pada posisi yang benar tetapi orientasinya salah, seperti pada Gambar 11.19.



Gambar 11.19 Orientasi Yang Salah

Untuk memperbaiki masalah ini, pertama-tama harus dipindahkan kembali ke Lapisan Atas. Hal ini dilakukan dengan menggunakan salah satu dari dua algoritma di atas untuk menempatkan potongan tepi lain di tempat tersebut. Khususnya, tepi ini harus berwarna Oranye, karena akan tetap diganti dan tidak akan salah menempatkan tepi penting lainnya. Setelah dipindahkan dari baris tengah, gunakan algoritma di atas dengan tepat untuk mengembalikannya ke orientasi yang benar. Lakukan ini untuk keempat tepi agar berhasil menyelesaikan lapisan tengah.

Subtujuan 4: Persilangan Atas Baru

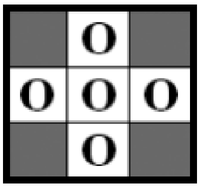


Gambar 11.20 Subtujuan 4

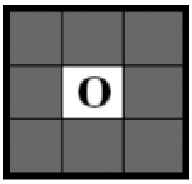
Kita telah berhasil menyelesaikan dua pertiga dari keseluruhan kubus.

Namun, menyelesaikan Lapisan Atas ini membutuhkan empat subtujuan. Subtujuan berikutnya adalah membuat tanda silang pada Permukaan Atas yang sekarang (Gambar 11.20). Potongan sisi mana yang digunakan tidak menjadi masalah selama bagian atas sisi berwarna Oranye dan terdapat tanda silang Oranye.

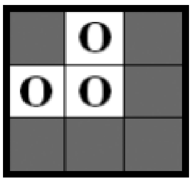
Setelah subtujuan sebelumnya, Permukaan Atas akan disusun dengan salah satu dari empat cara yang ditunjukkan pada Gambar 11.21 hingga Gambar 11.24.



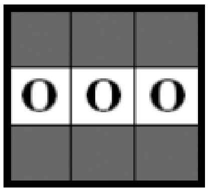
Gambar 11.21 negara bagian 1



Gambar 11.22 negara bagian 2



Gambar 11.23 negara bagian 3



Gambar 11.24 negara bagian 4

Pada keempat kondisi ini, jangan perhatikan warna sudutnya, apakah berwarna Oranye atau tidak. Selain itu, kondisi ini tetap harus dianggap sebagai Sisi Atas, bukan Sisi Depan. Dalam menjalankan algoritma berikut, Anda harus memilih Sisi Depan agar Sisi Atas tampak seperti salah satu gambar ini ketika Anda melihat kubus dari atas.

Kondisi 1: Persilangan sudah terbentuk, dan Anda dapat melanjutkan ke subtujuan 6.

Kondisi 2: Satu-satunya bagian dari persilangan adalah bagian tengahnya. Gunakan algoritma berikut:

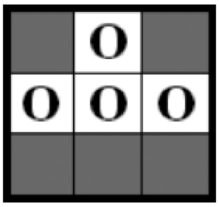
F U R U’ R’ F’

Kondisi 3: Hanya tiga keping persilangan yang berada di tempat yang tepat dan membentuk sesuatu seperti sudut siku-siku atau huruf "L" terbalik. Gunakan algoritma yang sama seperti Kondisi 2 untuk konfigurasi ini.

Kondisi 4: Sekali lagi, hanya tiga keping persilangan yang berada di tempat yang tepat, tetapi kali ini membentuk garis lurus. Untuk konfigurasi ini, gunakan algoritma berikut:

F R U R' U' F'

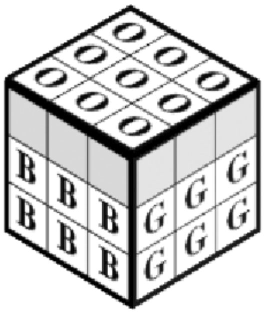
Selain itu, Anda mungkin menemukan bahwa Top Face terlihat seperti Gambar 11.25.



Gambar 11.25 keadaan alternatif

Ini pada dasarnya seperti kombinasi Keadaan 3 dan 4. Untuk konfigurasi ini, gunakan salah satu dari dua algoritma di atas dan konfigurasinya akan kembali ke salah satu keadaan sebelumnya. Ketika ini terjadi, gunakan algoritma yang sesuai dengan keadaan tersebut. Ini melengkapi subtujuan ini.

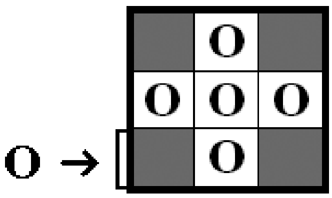
Subtujuan 5: Sudut-sudut Puncak Baru



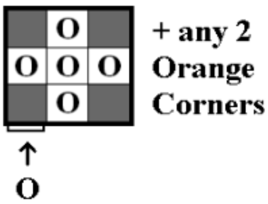
Gambar 11.26 Subtujuan 5

Subtujuan 5 (Gambar 11.26) adalah membuat sudut-sudut Sisi Atas berwarna Oranye, yang akan melengkapi Sisi Oranye. Sekali lagi, tidak masalah apakah sudut-sudut tersebut berada di tempat yang benar atau tidak, asalkan bagian atasnya berwarna Oranye.

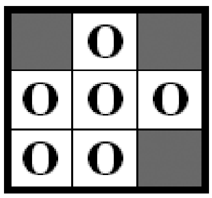
Setelah subtujuan sebelumnya, Sisi Atas (Oranye) akan memiliki salah satu dari tiga konfigurasi yang ditunjukkan pada Gambar 11.27 hingga Gambar 11.29.



Gambar 11.27 negara bagian 1



Gambar 11.28 negara bagian 2



Gambar 11.29 negara bagian 3

Sekali lagi, ini bukan Muka Depan, melainkan Muka Atas. Namun, pastikan Anda memilih Muka Depan agar konfigurasi ini sesuai dengan tampilan Muka Atas saat melihat kubus dari atas.

Untuk setiap keadaan, algoritma berikut akan digunakan:

R U R’ U R U U R’

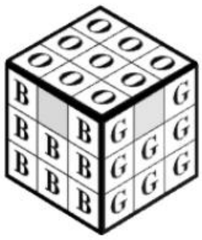
Keadaan 1: Hanya tanda silang yang ada. Penting bagi Anda untuk memposisikan kubus agar bagian pojok kiri bawah berwarna Oranye di sisi kiri luarnya, seperti yang digambarkan pada diagram. Dengan kata lain, saat melihat Muka Kiri, Anda akan melihat Oranye di pojok kanan atasnya.

Keadaan 2: Salah satu dari dua sudut di Muka Atas berwarna Oranye. Tidak masalah bagaimana kedua sudut tersebut diorientasikan. Yang penting adalah bagian pojok kiri bawah berwarna Oranye di sisi bawah luarnya, seperti yang digambarkan pada diagram. Dengan kata lain, saat melihat Muka Depan, pojok kiri atas harus berwarna Oranye.

Keadaan 3: Bersama dengan tanda silang, terdapat satu sudut Oranye. Inilah kondisi yang kita cari. Arahkan kubus sehingga bagian sudut ini berada di sudut kiri bawah Sisi Atas. Kemudian gunakan algoritma di atas.

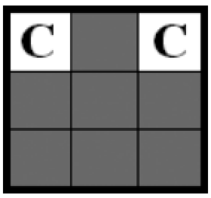
Setelah algoritma ini diterapkan secara terus-menerus pada kondisi ini, seluruh Sisi Atas pada akhirnya akan memiliki warna yang sama, dan sub-tujuan akan tercapai.

Sub-tujuan 6: Penempatan Sudut yang Benar

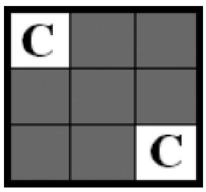


Gambar 11.30 Sub-Tujuan 6

Subtujuan 6 (Gambar 11.30) adalah memposisikan keempat keping sudut Lapisan Atas pada orientasi yang benar. Pada titik ini, akan ada tepat dua sudut pada orientasi yang benar. Dari tampilan atas kubus, Lapisan Atas akan disusun dengan salah satu dari dua cara yang ditunjukkan pada Gambar 11.31 dan Gambar 11.32.



Gambar 11.31 keadaan 1 (berdekatan)



Gambar 11.32 keadaan 2 (diagonal)

"C" menunjukkan keping sudut pada orientasi yang Benar. Keping-keping ini ditandai demikian karena hal ini dapat diterapkan pada dua sudut acak mana pun di Lapisan Atas. Untuk melihat sudut mana yang berada pada orientasi yang benar, seluruh kubus harus dianalisis. Untungnya, melakukan ini merupakan tugas yang mudah. Pertama, analisis keempat keping sudut sebagaimana adanya. Jika Anda tidak melihat bahwa mereka berada pada posisi yang benar, putar Lapisan Atas 90 derajat dan lihat kembali. Jika Anda masih tidak dapat menentukannya, putar Lapisan Atas 90 derajat sekali lagi. Pada akhirnya, akan menjadi jelas sudut mana yang berada pada orientasi yang benar. Namun, penting bahwa setelah Anda menemukan sudut yang benar, Anda harus memutar Lapisan Atas agar mereka ditempatkan pada lokasi yang benar relatif terhadap seluruh kubus.

Untuk subtujuan ini, algoritma berikut harus digunakan dalam kedua kasus:

R’ F R’ B B R F’ R’ B B R R U’

Keadaan 1: Ada kemungkinan bahwa dua keping sudut yang berorientasi benar akan berada tepat di samping satu sama lain. Pertama, periksa kembali untuk memastikan posisi kubus dan keping-keping tersebut benar. Jika tidak, Anda harus memutar Sisi Atas hingga posisinya benar. Hal ini mudah ditentukan, karena keduanya bersebelahan dan memiliki dua warna yang sama. Kemudian, pilih Sisi Depan sehingga ketika Anda melihat Sisi Atas dari atas, akan tampak seperti pada Gambar 11.31 (yaitu, keping-keping sudut atas). Saat melihat Sisi Depan, keping-keping ini seharusnya berada di bagian paling belakang kubus di bagian atas.

Keadaan 2: Ada kemungkinan juga kubus-kubus dengan orientasi yang benar berada tepat di seberang satu sama lain pada Lapisan Atas. Pada titik ini, kita perlu membawa kubus ke Keadaan 1. Jika keping-keping sudut dengan orientasi yang benar tampak seperti pada Gambar 11.32, cukup gunakan algoritma di atas. Posisi Lapisan Atas atau sisi mana yang dipilih sebagai Sisi Depan tidak menjadi masalah, asalkan Sisi Atas yang kita gunakan tetap seperti itu (dalam hal ini Sisi Oranye). Pada titik ini, kubus Anda akan siap untuk subtujuan terakhir.

Subtujuan 7: Tepi Atas

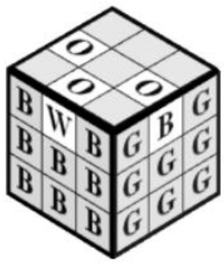
Gambar 11.33 Subtujuan 7 (Kondisi Tujuan)

Subtujuan terakhir, subtujuan 7, adalah memposisikan dengan benar empat keping tepi Lapisan Atas yang tersisa (Gambar 11.33). Ada dua algoritma yang dapat menyelesaikan kubu. Pada titik ini, kubus dapat berada dalam salah satu dari dua kondisi:

Kondisi 1: Satu kubus tepi berada di posisi yang benar. Mungkin saja salah satu tepi Lapisan Atas sudah berada di posisi yang benar. Hal pertama yang harus dilakukan adalah memeriksa kubus dan menganalisis seluruh Lapisan Atas untuk melihat tepi mana yang berada di posisinya. Hal ini mudah ditentukan karena tepi yang berada di posisi yang benar akan menyelesaikan salah satu Sisi Sisi. Kemudian, orientasikan kubus sehingga sisi yang telah selesai menjadi bagian dari apa yang sekarang kita anggap sebagai Lapisan Belakang.

Langkah selanjutnya adalah menentukan algoritma mana yang akan digunakan. Untuk melakukannya, kita harus menganalisis keping tepi yang salah dan bagaimana posisinya dalam kaitannya satu sama lain. Pertama, perhatikan dua sisi yang salah, sebaiknya satu sisi di Lapisan Depan dan satu sisi di Lapisan mana pun yang bersebelahan dengannya (misalnya, Lapisan Kiri atau Lapisan Kanan). Pada titik ini, Anda akan melihat bahwa salah satu sisi harus digeser ke kiri atau kanan agar posisinya benar.

Misalnya, kita telah menemukan Sisi Biru sebagai bagian dari Lapisan Depan. Asumsikan konfigurasi kita seperti pada Gambar 11.34.



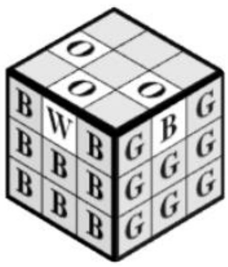
Gambar 11.34 Tepi Oranye-Biru Ke Kanan

Kita melihat bahwa potongan tepi Oranye-Biru perlu dipindahkan ke Lapisan Depan dari Lapisan Kanan, yang berarti kita perlu menggesernya ke kiri. Oleh karena itu, gunakan algoritma berikut:

F F U L R’ F F L’ R U F F

Ini akan menggeser ketiga potongan tepi target searah jarum jam relatif terhadap Lapisan Atas, sehingga kita dapat menyebutnya algoritma "searah jarum jam".

Mungkin juga kubus tersebut memiliki konfigurasi seperti pada Gambar 11.35.



Gambar 11.35 Tepi Oranye-Hijau Ke Kanan

Sekarang, kita melihat bahwa tepi Oranye-Hijau perlu dipindahkan dari Lapisan Depan ke Lapisan Kanan dan menggesernya ke kanan. Oleh karena itu, kita menggunakan apa yang kita sebut algoritma "berlawanan arah jarum jam":

F F U’ L R’ F F L’ R U’ F F

Ini akan menggeser ketiga keping tepi target berlawanan arah jarum jam. Menggunakan salah satu algoritma ini sekali saja akan menempatkan tepi-tepi ini pada posisi yang benar.

Keadaan 2: Tidak ada tepi Lapisan Atas yang benar. Mungkin saja keempat tepi Lapisan Atas tidak berada pada posisi yang salah. Jika demikian, gunakan salah satu dari dua algoritma di atas. Algoritma ini tidak akan menempatkan keempat tepi pada posisi yang benar, tetapi akan menempatkan satu tepi pada posisi yang benar, sehingga kubus berada pada Keadaan 1. Pada titik ini, Kubus Rubik akan terpecahkan sepenuhnya.

11.4 PEMECAHAN MASALAH MANUSIA

Sayangnya, kami belum dapat mengumpulkan subjek uji untuk menerapkan analisis Pemecahan Masalah Manusia kami pada Kubus Rubik. Namun, jika Anda mengenal seseorang yang bertekad untuk memecahkan teka-teki, teka-teki itu kemungkinan besar adalah Kubus Rubik. Mungkin hanya karena penasaran, tetapi jika Anda meletakkan Kubus Rubik di ruangan yang berisi beberapa orang, seseorang (bahkan mungkin beberapa orang) akan mengambilnya dan mencoba memecahkannya. Bahkan jika Anda sedang bersama seorang teman dan kita sedang mengobrol, ia akan mengambilnya dan mulai memecahkannya, baik secara sadar maupun tidak sadar.

Secara pribadi, kami belum pernah melihat seseorang memecahkan kubus tersebut. Jika kami mengamati seseorang yang mencoba, ia selalu membuat kesalahan fatal dengan mencoba memecahkan satu sisi pada satu waktu. Pertama, tanpa mencoba menggunakan algoritma apa pun, mereka kesulitan memecahkan satu sisi. Begitu mereka memecahkannya, mereka mencoba memecahkan sisi kedua, hanya untuk menemukan bahwa sisi pertama mereka menjadi tidak terpecahkan. Lalu mereka menyerah begitu saja. Seandainya mereka mencoba memposisikan kubus-kubus kecil dengan benar, alih-alih hanya sisi-sisinya, mungkin mereka akan lebih berhasil.

11.5 ANALISIS JENDELA MANUSIA TERHADAP SOLUSI

Kesulitan
5/10

kerumitan
O(**n**)

Nama	Int atau Ext?	Intens./ Ekstens.	Repr.	Jendela Manusia?	Benar?	Ukuran Butir	Eksekusi	Keterpahaman	Metode Pemecahan Masalah	Fleksibilitas	Mode Konversi	Optimal?	Total
Adam	Eks.	3/10	Teks dgn Gambar	Y	Y	Ideal	8/10	4/10	Sub-tujuan	5/10	Editor Gbr/Teks	T	20/40

Animasi	Eks.	9/10	Teks dgn Gambar	Y	Y	Ideal	9/10	8/10	Sub-tujuan	7/10	Editor Gbr/Teks	T	33/40
C&P	Eks.	4/10	Teks dgn Gambar	Y	Y	Ideal	8/10	6/10	Sub-tujuan	6/10	Editor Gbr/Teks; Tangan	T	24/40
Gambar Kartun	Eks.	5/10	Teks dgn Gambar	Y	Y	Ideal	8/10	7/10	Sub-tujuan	7/10	Editor Gbr/Teks; Tangan	T	27/40
Musuh	Eks.	8/10	Teks dgn Gambar	Y	Y	Ideal	7/10	9/10	Sub-tujuan	7/10	Editor Gbr/Teks	T	31/40
Kubus Ajaib	Eks.	7/10	Teks dgn Gambar	Y	Y	Ideal	7/10	8/10	Sub-tujuan	7/10	Editor Gbr/Teks; Tangan	T	29/40
Pendek	Eks.	6/10	Teks dgn Gambar	Y	Y	Ideal	2/10	3/10	Sub-tujuan	6/10	Editor Gbr/Teks; Tangan	T	17/40
Kubus Sederhan a	Eks.	7/10	Teks dgn Gambar	Y	Y	Ideal	8/10	9/10	Sub-tujuan	7/10	Editor Gbr/Teks	T	31/40
Kotak	Eks.	7/10	Teks dgn Gambar	Y	Y	Ideal	6/10	4/10	Sub-tujuan	7/10	Editor Gbr/Teks	T	24/40

Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tabel 11.1 Peringkat Solusi Kubus Rubik Berdasarkan Jendela Manusia

Karena Kubus Rubik merupakan teka-teki yang sangat populer dengan begitu banyak metode untuk menyelesaikannya, wajar saja jika terdapat begitu banyak solusi untuknya. Namun, terlepas dari kenyataan bahwa begitu banyak solusi yang ada, representasinya pada dasarnya identik.

Saat menganalisis solusi, dapat dilihat bahwa terdapat tiga fitur utama untuk masing-masing solusi yang membentuk representasinya:

Penggambaran Kubus: Baik untuk menunjukkan susunan tujuan, langkah selanjutnya, atau susunan yang harus dihindari, solusi umumnya menampilkan gambar kubus itu sendiri. Menampilkan gambar kubus akan menunjukkan gambaran yang jelas tentang seperti apa seharusnya atau tidak seharusnya kubus tersebut pada langkah tertentu. Hal ini membuat solusi sangat mudah dipahami, terutama bagi mereka yang menggunakan kubus fisik, karena mereka cukup mencocokkan kubus mereka dengan gambar. Kubus dapat dideskripsikan menggunakan kata-kata, tetapi hal itu akan membuat segalanya menjadi sangat rumit.

Deskripsi Teks: Meskipun gambar bermanfaat, penting juga bahwa sebuah solusi memuat deskripsi. Ini akan membantu menekankan dan menjelaskan fitur-fitur penting dari sebuah solusi, seperti notasinya, sisi-sisi kubus yang perlu difokuskan pada langkah tertentu, dan apa yang harus diperhatikan atau dihindari pada suatu langkah. Deskripsi tekstual memberikan semacam panduan dalam menyelesaikan solusi. Solusi yang hanya berisi gambar tanpa deskripsi akan membuatnya kurang terorganisir dan lebih sulit diikuti, sehingga mungkin kurang mudah dipahami.

Notasi Gerakan: Fitur yang tampaknya paling berbeda di antara solusi-solusi ini adalah representasi lapisan-lapisan kubus yang berputar ketika menjelaskan algoritma. Representasi yang paling umum digunakan adalah gambar kubus yang sebenarnya, dengan indikasi lapisan mana yang perlu diputar dan ke arah mana. Umumnya, ini tampaknya merupakan representasi terbaik. Representasi lainnya adalah kotak kecil dengan garis-garis yang mewakili lapisan dan panah pada garis-garis yang mewakili arah rotasi (yaitu), seperti  pada "Solusi Adam" dan "Solusi C&P". Terakhir, representasi lain menggunakan huruf yang mewakili lapisan yang akan diputar dan apostrof untuk menunjukkan bahwa lapisan tersebut akan digerakkan berlawanan arah jarum jam, seperti pada "Solusi Gambar Kartun", "Solusi Singkat", dan solusi yang disajikan dalam bab ini (Bagian 11.3).

Fitur-fitur ini merupakan faktor penentu utama seberapa komprehensif suatu solusi. Seperti yang dapat Anda lihat dari tabel, peringkatnya umumnya beragam. Beberapa solusi memiliki lebih banyak diagram dan tanpa teks, yang lain memiliki terlalu banyak teks dan terlalu sedikit diagram, beberapa tepat, dan seterusnya. Agar suatu solusi mendapat peringkat tinggi dalam hal pemahaman, solusi tersebut harus memiliki jumlah teks dan diagram yang tepat. Namun, notasi pergerakan dapat bervariasi, tergantung pada bagaimana solusi tersebut disampaikan.

Algoritma dan subtujuan yang paling umum digunakan adalah yang digunakan dalam solusi yang disajikan dalam bab ini. Namun, beberapa solusi menyediakan subtujuan yang serupa tetapi dengan algoritma yang berbeda. Solusi lainnya memberikan algoritma dan subtujuan yang sama sekali berbeda. Kriteria inilah yang menentukan peringkat eksekusi suatu solusi. Algoritma yang lebih panjang dan lebih kompleks yang digunakan umumnya lebih sulit untuk dijalankan dengan benar dan mungkin memerlukan lebih banyak memori.

Terakhir, tidak ada solusi yang dianalisis yang optimal. Karena solusi ditemukan dengan mencapai subtujuan tertentu, jumlah langkah yang diperlukan untuk mencapai subtujuan ini kemungkinan besar lebih banyak daripada jumlah langkah tersingkat yang memungkinkan. Solusi yang lebih maju kemungkinan besar optimal, karena menunjukkan algoritma yang lebih presisi (lihat Bagian 11.6).

Solusi yang Paling Kompatibel dengan Jendela Manusia

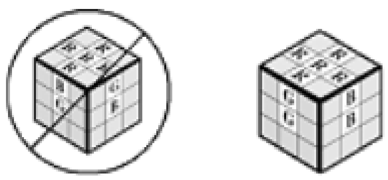
Solusi yang dinilai sebagai MHWC adalah "Solusi Animasi", yang dikembangkan oleh Ryan Heise. Solusi ini mengandung animasi, sehingga sulit untuk direproduksi di atas kertas. Namun, solusi ini dapat ditemukan di referensi. Solusi ini menggunakan subtujuan, langkah, dan algoritma yang sama dengan solusi yang digambarkan di bagian ini. Solusi ini menggambarkan subtujuan sehingga mudah dipahami, sementara deskripsinya singkat dan menjelaskan secara jelas cara mencapai subtujuan ini, sehingga solusi ini sangat mudah dipahami.

Solusi ini juga mencakup setiap kasus yang mungkin terjadi pada subtujuan tertentu dan cara menyelesaikannya dengan benar. Fitur paling menonjol dari solusi ini, dan salah satu alasan utama solusi ini dianggap sebagai MHWC, adalah fakta bahwa Penggambaran Kubus dan Notasi Gerakan digabungkan. Solusi ini menggambarkan gambar tiga dimensi kubus untuk menunjukkan susunan subtujuan akhir dan seperti apa kubus tersebut pada kasus-kasus tertentu, sehingga meningkatkan pemahaman dan ekstensionalitas solusi. Fitur paling menonjol dari gambar-gambar ini adalah animasinya, sehingga ketika diklik, lapisan-lapisannya berputar untuk menunjukkan bagaimana algoritma tertentu dijalankan sesuai perintah. Hal ini semakin meningkatkan pemahaman dan eksekusi solusi.

Untuk lebih meningkatkan solusi ini, langkah-langkah setiap algoritma harus digambarkan bersama dengan kubus-kubus yang dianimasikan. Ini akan memberi pemecah pilihan antara menonton animasi dan mengikuti langkah-langkah algoritma sebagaimana umumnya diikuti.

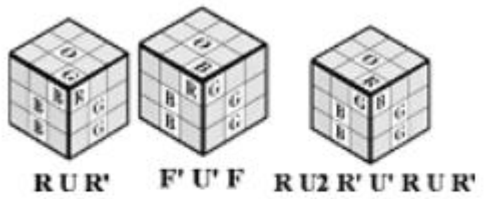
Solusi Jendela yang Paling Tidak Kompatibel dengan Manusia

Langkah 1: Salib



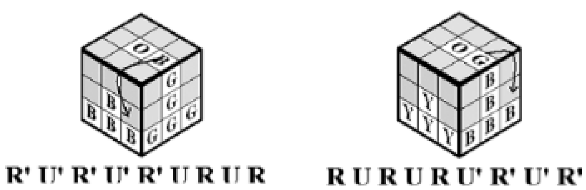
Langkah 2: Sudut Lapisan Bawah

Setiap algoritma memindahkan sudut dari lapisan atas ke lapisan bawah tepat di bawahnya.



Langkah 3: Lapisan Tengah

• Pertama, posisikan tepi dari lapisan atas ke lapisan tengah di bagian depan.
• Kedua, posisikan tepian dari lapisan paling atas ke lapisan tengah di bagian belakang.



Langkah 4: Tepi

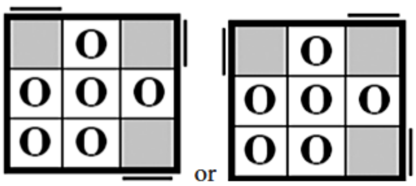
Jika potongan Jeruk membentuk garis di atas, terapkan algoritma pertama. Kemudian, terapkan algoritma kedua.

F (R U R' U') F' U2 F (U R U' R') F'

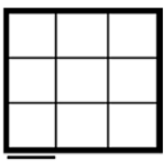
Langkah 5: Sudut

Lihat lapisan atas. Algoritma: R U R’ U R U2 R’

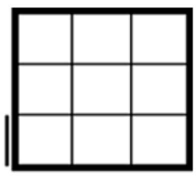
Jika SATU sudut berwarna Oranye, pindahkan ke kiri bawah. Kemudian terapkan algoritmanya.



Jika DUA sudut berwarna Oranye, putar lapisan atas hingga stiker Kuning berada di kiri bawah, menghadap ke depan. Kemudian terapkan algoritmanya.

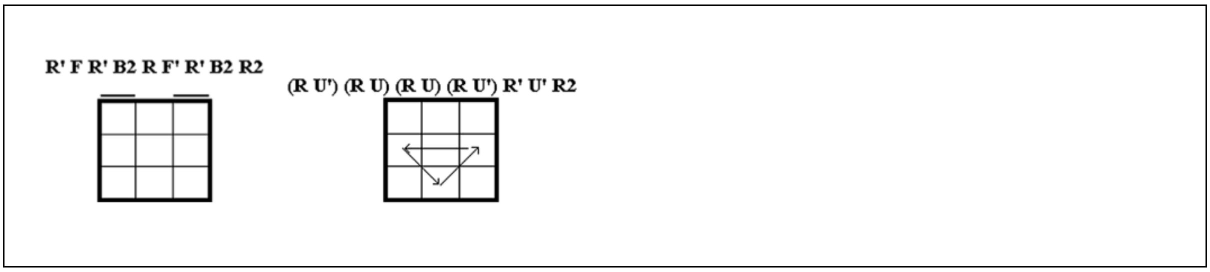


Jika NOL sudut berwarna Oranye, putar lapisan atas hingga stiker Oranye berada di kiri bawah, menghadap ke kiri. Kemudian terapkan algoritmanya.



Langkah 6: Permutasi

Putar lapisan atas hingga warna yang sama berada di belakang. Jika Anda tidak memilikinya, terapkan algoritma untuk mendapatkannya.



Gambar 11.36 “Solusi Singkat”

Solusi yang dinilai sebagai LHWC adalah “Solusi Singkat” (Gambar 11.36). (Representasi serupa dapat ditemukan dalam referensi.) Meskipun dimaksudkan sebagai referensi bagi mereka yang sudah memahami solusinya, kita akan memandang solusi ini dalam konteks Jendela Manusia seolah-olah ditujukan bagi mereka yang belum tahu cara memecahkan teka-teki.

Solusi ini juga menggunakan subtujuan umum seperti yang disajikan di Bagian 11.3. Namun, tidak seperti “Solusi Animasi” dan solusi yang disajikan di Bagian 11.3, solusi ini terlalu singkat. Subtujuan dijelaskan secara singkat dan kurang detail, yang mungkin dapat mengurangi pemahaman dan eksekusi. Bagian ini juga menunjukkan kemungkinan susunan lapisan untuk subtujuan dan algoritma yang digunakan untuk menyelesaikannya, tetapi gagal menunjukkan setiap kemungkinan kasus, yang juga dapat mengurangi eksekusi.

Terakhir, penting untuk dicatat bahwa subtujuan pertama tidak menggambarkan deskripsi atau algoritma apa pun. Subtujuan ini hanya menunjukkan susunan akhir dan susunan yang harus dihindari. Jika seseorang baru dalam memecahkan Kubus Rubik, ia tidak akan tahu algoritma mana yang harus digunakan.

11.6 SOLUSI MESIN TERBAIK

Kubus Rubik memiliki sekitar 43 kuintiliun kemungkinan susunan, jadi sejak awal kita dapat mengabaikan segala jenis algoritma brute force. Meskipun jumlahnya sangat besar, kita telah menunjukkan bahwa kubus dapat dipecahkan dalam jumlah langkah yang wajar dengan menyelesaikan subtujuan. Pertanyaannya sekarang adalah seberapa efisien sebuah mesin dapat memecahkan kubus tersebut?

Pada tahun 1981, matematikawan Morwen Thistlethwaite mengembangkan algoritma kompleks yang dapat memecahkan kubus dengan 52 langkah atau kurang. Karena membutuhkan memori yang cukup besar, algoritma ini lebih efisien untuk sebuah mesin. Berbeda dengan algoritma yang telah kita lihat di bab ini, algoritma Thistlethwaite "bekerja pada semua bagian secara bersamaan, membatasinya pada kemungkinan yang semakin sedikit hingga hanya tersisa satu posisi yang memungkinkan untuk setiap bagian". Terdapat empat fase:

$\mathrm{G0} = \{\mathrm{L}, \mathrm{R}, \mathrm{F}, \mathrm{B}, \mathrm{U}, \mathrm{D}\}$

$\mathrm{G1} = \{\mathrm{L}, \mathrm{R}, \mathrm{F}, \mathrm{B}, \mathrm{U2}, \mathrm{D2}\}$

$\mathrm{G2} = \{\mathrm{L}, \mathrm{R}, \mathrm{F2}, \mathrm{B2}, \mathrm{U2}, \mathrm{D2}\}$

$\mathrm{G3} = \{\mathrm{L2}, \mathrm{R2}, \mathrm{F2}, \mathrm{B2}, \mathrm{U2}, \mathrm{D2}\}\ \mathrm{G4} = \{1\}$

Kelompok-kelompok ini mewakili subtujuan yang ingin dicapai. Subtujuan-subtujuan ini digambarkan sebagai himpunan, yang mewakili setiap kemungkinan susunan yang dapat dicapai, dengan batasan tertentu pada pergerakan lapisan. Huruf-huruf dalam tanda kurung mewakili batasan-batasan ini. Setiap huruf mewakili satu lapisan kubus, dan huruf-huruf dengan eksponen di sebelahnya dibatasi hanya untuk putaran 180 derajat.

Untuk mencapai subtujuan tertentu, kubus harus disusun di salah satu posisi yang terdapat dalam himpunan posisi subtujuan berikutnya. Misalnya, katakanlah kubus tersebut berada di salah satu susunan himpunan G1. Ini berarti bahwa untuk mencapai subtujuan G2, kubus perlu diposisikan ulang ke posisi yang terdapat dalam subtujuan G2, yang pada dasarnya berarti posisi di mana lapisan Depan, Belakang, Atas, atau Bawah tidak perlu lagi diputar 90 derajat. Namun, untuk mencapai posisi tersebut, kita harus mengikuti aturan G1, dan setiap kali permukaan Atas atau Bawah akan dipindahkan, keduanya harus diputar 180 derajat, bukan 90 derajat. Untuk menentukan apakah subtujuan berikutnya terpenuhi atau tidak, posisi kubus saat ini dicocokkan dengan tabel.

Meskipun algoritma Thistlethwaite telah disempurnakan, kelemahan terbesarnya adalah jumlah fase yang membuat tabel harus cukup besar. Salah satu orang yang mengatasi masalah ini dan menyempurnakan algoritmanya adalah Herbert Kociemba. Pertama, ia menyederhanakan algoritma menjadi hanya dua fase:

$\mathrm{G0} = \mathrm{U, D, R, L, F, B}$

$\mathrm{G1} = \mathrm{U, D, R^2, L^2, F^2, B^2}$

$\mathrm{G2} \ = \ \{1\}$

Konsekuensi dari pengurangan jumlah fase adalah peningkatan jumlah langkah yang tersedia dan ukuran tabel per fase secara signifikan. Untuk mengatasi hal ini, ia memutuskan untuk "membuat tabel pemangkasan dan menggunakan IDA* untuk menyelesaikan setiap fase," yang secara signifikan mengurangi ruang keadaan. Untuk menyempurnakan algoritma lebih lanjut, Kociemba menambahkan fitur yang melanjutkan pencarian IDA* bahkan setelah solusi untuk fase tertentu ditemukan. Hal ini memungkinkan algoritma untuk menemukan solusi dengan jumlah langkah terpendek yang mungkin.

Meskipun algoritma Kociemba efisien, itu tidak optimal. Pada tahun 1995, Michael Reid menganalisis algoritma ini dan menemukan bahwa dibutuhkan paling banyak 29 langkah untuk memecahkan kubus dengan algoritma. Namun, pada tahun 2010, Tomas Rokicki, Herbert Kociemba, Morley Davidson, dan John Dethridge membuktikan bahwa Rubik's Cube dapat dipecahkan dalam paling banyak 20 langkah, tidak peduli apa posisinya. Meskipun tidak ada algoritma khusus yang dapat memecahkan kubus dalam 20 langkah, ini dibuktikan dengan melakukan penelitian ekstensif untuk memecahkan semua 43 kuintiliun kemungkinan pengaturan.

Dengan menggunakan pemrograman matematika sederhana dan dengan mempartisi pengaturan ini ke dalam set tertentu, dibutuhkan sekitar 35 tahun CPU untuk memecahkan setiap posisi Rubik's Cube, menemukan bahwa hanya dibutuhkan paling banyak 20 langkah untuk memecahkan kubus dari pengaturan apa pun. Angka ini sekarang disebut sebagai "Angka Tuhan" karena merupakan jumlah langkah optimal, yang menunjukkan bahwa

"Algoritma Tuhan" akan menghasilkan 20 langkah. Istilah "Algoritma Tuhan" mengacu pada algoritma yang memecahkan Kubus Rubik, atau teka-teki kombinasi lainnya, dalam jumlah langkah tersingkat. Beberapa algoritma telah mendekati, tetapi belum ada yang mendapatkan gelar tersebut.

Terlepas dari algoritma mana yang digunakan untuk memecahkan Kubus Rubik, kompleksitas untuk setiap algoritma dalam kasus terburuk ditemukan sebesar $\boldsymbol{\theta}(\mathbf{n2}/\mathbf{log}(\mathbf{n}))$, menurut sebuah makalah penelitian tentang algoritma untuk Kubus Rubik berjudul "Algoritma untuk Memecahkan Kubus Rubik".

11.7 PROGRAM YANG DAPAT DIMAINKAN

Permainan Kubus Rubik yang dapat dimainkan tersedia di $http://activeden.net/item/cube-assembler/full\_screen\_preview/3131480$. Permainan ini menggambarkan kubus tiga dimensi dengan lapisan-lapisan yang dapat diputar hanya dengan sekali usap mouse. Selain itu, kubus itu sendiri dapat diputar sesuka hati dengan menggeser salah satu dari dua bilah gulir, yang memungkinkan pemain untuk melihat kubus secara penuh. Selain itu, pemain dapat memilih antara menyelesaikan kubus $2'' \times 2'' \times 2''$ yang mudah hingga kubus $5'' \times 5'' \times 5''$ yang lebih sulit. Terakhir, pemain memiliki opsi untuk mengubah warna permukaan kubus menjadi warna apa pun yang diinginkannya.

BAB 12
DILEMA TAHANAN

12.1 PENDAHULUAN

Dalam film *The Postman Always Rings Twice*,** kedua tokoh utama jatuh cinta dan memutuskan untuk "menghilangkan" suami dari tokoh wanita tersebut. Polisi tidak memiliki cukup bukti untuk menjatuhkan hukuman. Namun, setelah pembunuhan tersebut, kedua kekasih tersebut ditangkap oleh polisi dan ditahan untuk diinterogasi di ruang interogasi terpisah, di mana masing-masing ditawari proposisi yang sama: "Katakan yang sebenarnya dan dakwa pasanganmu, dan kami akan bersikap lunak padamu." Menyerahkan permintaan ini disebut "membelot." Kedua pelaku tahu bahwa kaki tangan mereka ditawari kesepakatan yang sama. Apa yang harus dilakukan masing-masing tahanan? Situasi ini menarik karena tidak ada satu orang pun yang mengetahui pikiran pasangannya (mereka berada di sel isolasi). Dalam dunia yang ideal, keduanya akan tetap setia satu sama lain (sesuai kesepakatan awal mereka dalam melakukan kejahatan) dan tanpa adanya bukti lebih lanjut dan kesaksian pendukung yang memberatkan mereka, kemungkinan besar akan dihukum dengan kejahatan yang lebih ringan. Namun, jika salah satu kaki tangan membelot, maka yang lain tentu akan lebih baik dengan membelot juga, daripada menjalani hukuman pembunuhan.

Dilema Tahanan ini pertama kali dirumuskan dalam istilah teori permainan oleh Merrill Floyd dan Melvin Dresher di RAND Corporation pada tahun 1950. Pada saat itu, sebagian besar negara terlibat dalam Perang Dingin dengan dua negara adidaya nuklir: Uni Soviet dan Amerika Serikat. Haruskah kedua negara ini bekerja sama satu sama lain dan berupaya menuju perlucutan senjata bersama (sambil menyadari setiap saat bahwa pihak lain mungkin mengingkari), atau haruskah masing-masing terus menciptakan persenjataan baru yang lebih mematikan? Inilah dilema yang dihadapi planet kita selama empat dekade, dari akhir Perang Dunia II hingga akhirnya berakhirnya Perang Dingin pada tahun 1989 dengan runtuhnya Tembok Berlin. Dilema Tahanan dengan tepat memodelkan era ketidakpercayaan tersebut. Dilema ini juga dapat dimodelkan oleh **matriks imbalan** pada Gambar 12.1. Matriks imbalan menentukan imbalan bagi setiap pemain untuk setiap kombinasi tindakan oleh kedua peserta permainan.

Asumsikan bahwa kedua pemain (yaitu, para tahanan) dalam permainan ini rasional dan ingin meminimalkan hukuman penjara mereka. Setiap tahanan memiliki dua pilihan (seperti yang dilakukan kedua tokoh dalam *The Postman Always Rings Twice*): bekerja sama dengan rekan kriminal mereka dan tetap diam atau membelot dengan mengaku kepada polisi demi hukuman yang lebih ringan.

Tahanan B

Tahanan A		Bekerja Sama (tetap diam)	Bekerja Sama (tetap diam)
	Bekerja Sama (tetap diam)	A: 1 Tahun B: 1 Tahun	A: 10 Tahun B: 0 Tahun
	Mengkhianati (menghianati pasangan)	A: 0 Tahun B: 10 Tahun	A: 5 Tahun B: 5 Tahun

Gambar 12.1 Matriks Imbalan Untuk Dilema Tahanan

Anda mungkin memperhatikan bahwa permainan ini berbeda dalam satu aspek penting dari permainan yang dibahas sebelumnya dalam buku ini. Untuk menentukan arah tindakan dalam permainan dan teka-teki lain yang dibahas, Anda perlu mengetahui arah tindakan lawan Anda. Misalnya, jika Anda berada di posisi kedua dalam permainan tic-tac-toe, Anda perlu mengetahui di mana pemain lain meletakkan X awal. Hal ini tidak berlaku dalam Dilema Tahanan. Misalkan Anda adalah pemain A dan memilih untuk membelot. Namun, pemain B memutuskan untuk tetap setia dan memilih strategi kerja sama. Dalam hal ini, keputusan Anda tidak menghasilkan hukuman penjara, berbeda dengan hukuman satu tahun jika Anda berdua memilih untuk bekerja sama. Jika rekan Anda memilih untuk membelot, hasil Anda tetap lebih baik jika Anda memilih untuk membelot. Dalam teori permainan, membelot adalah **strategi yang dominan**. Karena Anda berasumsi bahwa lawan Anda dalam permainan ini rasional, ia akan memilih strategi yang sama.

Strategi {Mengkhianati, Mengkhianati} yang digunakan oleh kedua partisipan disebut sebagai **Ekuilibrium Nash**. Perubahan strategi oleh salah satu pemain akan menghasilkan imbalan yang lebih kecil bagi mereka (yaitu, hukuman penjara yang lebih lama).

Seperti yang ditunjukkan pada Gambar 12.1, jika setiap pemain bertindak lebih berdasarkan keyakinan daripada rasionalitas (keyakinan bahwa pasangan mereka akan tetap setia), maka total imbalan akan melebihi total 10 tahun penjara yang diberikan oleh Ekuilibrium Nash {Cacat, Cacat}. Strategi {Bekerja Sama, Bekerja Sama} ini menghasilkan hasil terbaik dalam hal total imbalan bagi kedua pemain. Strategi optimal ini disebut **Pareto Optimal**. Perlu dicatat bahwa Dilema Tahanan bukanlah permainan zero-sum. *Mengapa tidak*? Dalam permainan semacam itu, Ekuilibrium Nash tidak selalu sesuai dengan permainan Pareto Optimal.

12.2 DILEMA TAHANAN YANG BERULANG

Jika Anda memainkan Dilema Tahanan hanya sekali, membelot adalah strategi yang mendominasi bagi kedua pemain. Dalam versi lain dari permainan ini, Anda bermain berulang kali—n kali—selama masih ada ingatan tentang tindakan sebelumnya. Ketika memiliki lebih dari satu giliran, setiap pemain tidak akan mudah untuk *membelot*, karena mereka tahu bahwa balas dendam dari lawan akan segera datang. Salah satu strateginya adalah memulai dengan satu kerja sama untuk memberi lawan kesempatan bertindak dengan *penuh belas kasih*. Jika lawan tetap memilih untuk membelot, Anda dapat melawan dengan terus-menerus membelot. Jika lawan akhirnya memilih untuk bekerja sama, Anda dapat kembali ke kebijakan yang *lebih murah hati*. Menariknya, dengan Dilema Tahanan dan iterasi atau variasinya, kita dapat mengaitkan kata-kata dengan angka yang mewakili perilaku. Perilaku dapat mencakup, misalnya, pengkhianatan, pembelotan, kerja sama, kepercayaan, balas dendam, kemurahan hati, kasih sayang, dan sebagainya. Konsep-konsep ini dibahas secara lengkap dalam risalah Robert Bram yang luar biasa, "Theory of Moves". Latihan di akhir bab ini membahas permainan dua orang lainnya yang serupa dengan Dilema Tahanan.

12.3 APLIKASI DI BERBAGAI BIDANG

Dilema Tahanan memungkinkan kita untuk menangani dan menggambarkan situasi masalah dalam ilmu sosial—misalnya, ekonomi, ilmu politik, psikologi, filsafat—dengan cara numerik sehingga dapat dipahami dengan lebih baik dan solusinya dapat direpresentasikan secara efektif dengan cara yang mudah. Selain itu, melalui "Dilema Tahanan Berulang", generasi-generasi masalah yang sedang dipertimbangkan dapat disimulasikan. Perbedaan antara Dilema Tahanan dan masalah lain yang telah kita bahas hingga saat ini adalah bahwa Dilema ini berkaitan dengan pertimbangan "Bagaimana jika?". Dengan kata lain, tindakan dan respons terhadapnya dipertimbangkan dan dievaluasi serealistis mungkin dengan "sistem matriks". Bab 13 menyajikan masalah "Sepuluh Bajak Laut dan Emas Mereka." Masalah ini mirip dengan Dilema Tahanan, karena untuk menentukan tindakan apa yang harus diambil, Anda harus mempertimbangkan sejumlah kemungkinan "langkah" dan skenario beberapa langkah ke depan.

Berikut ini adalah masalah-masalah umum yang telah diajukan dalam buku Lucci dan Kopec. Pertimbangkan apakah masalah-masalah tersebut merupakan contoh dari masalah yang pada dasarnya sama dengan Dilema Tahanan. Jika Anda menyimpulkan bahwa suatu masalah setara dengan Dilema Tahanan, rancanglah matriks hasil yang sesuai. Berikan komentar tentang keberadaan ekuilibrium Nash dan Pareto Optimal dalam setiap kasus.

Contoh 1: Linux adalah versi Unix yang dikembangkan di bawah lisensi GPL. Berdasarkan perjanjian ini, Anda diberikan perangkat lunak bebas dan dapat mempelajari kode sumbernya, yang dapat dimodifikasi. Anda dapat membelot dengan menyimpan penyempurnaan ini untuk diri sendiri, atau Anda dapat bekerja sama dengan mendistribusikan versi kode yang telah diperbaiki. Faktanya, kerja sama dipaksakan dengan melarang pendistribusian hanya kode sumber tanpa penyempurnaan Anda.

Solusi: Dalam masalah ini, angka yang lebih tinggi berarti lebih banyak keberhasilan.

Jika Anda **bekerja sama/berkolaborasi** dengan mendistribusikan kode sumber beserta penyempurnaan Anda, terdapat manfaat bersama. Ini pada dasarnya merupakan variasi dari Dilema Tahanan. Jika Anda melakukan kesalahan dengan tidak melakukan perbaikan atau mendistribusikan perangkat lunak secara ilegal, tindakan ini masing-masing direpresentasikan oleh kata "bekerja sama/cacat" (10, 0) atau "cacat/bekerja sama" (0, 10). Terakhir, jika Anda mendistribusikan perangkat lunak DAN tidak melakukan perbaikan apa pun, Anda melakukan "cacat/cacat", yang berarti Anda adalah pencuri dan pembohong.

UNIX dan GPL	Bekerja Sama	Membelot
Bekerja Sama	(5, 5)	(10, 0)
Membelot	(0, 10)	(1, 1)

Tabel 12.1 Contoh 1

Ekuilibrium Nash direpresentasikan oleh (0, 10) dan (10, 0) seperti pada Dilema Tahanan awal.

Contoh 2: Perusahaan rokok pernah diizinkan untuk beriklan di Amerika Serikat. Jika hanya satu perusahaan yang memutuskan untuk beriklan, peningkatan penjualan pasti akan terjadi. Namun, jika dua perusahaan meluncurkan kampanye iklan, iklan mereka pada dasarnya akan saling meniadakan, dan tidak ada peningkatan pendapatan yang dihasilkan.

Solusi:

Ini mirip dengan Dilema Tahanan awal, tetapi dengan sedikit perbedaan. Jika kedua perusahaan beriklan, Ekuilibrium Nash adalah jika salah satu perusahaan beriklan. Jika kedua perusahaan beriklan atau tidak ada perusahaan yang beriklan, maka hasilnya adalah status quo (1, 1).

IKLAN ROKOK	BERIKLAN	TIDAK BERIKLAN
Beriklan	(1, 1)	(10, 1)
Tidak Beriklan	(1, 10)	(1, 1)

Tabel 12.2 Contoh 2

Contoh 3: Di Selandia Baru, kotak koran dibiarkan tidak terkunci. Seseorang dapat dengan mudah mencuri koran (cacat). Tentu saja, jika semua orang melakukan ini, tidak akan ada koran yang tersisa. Jelas ini adalah contoh Dilema Tahanan Berulang, karena meskipun pada awalnya tampak bahwa koran yang "dicuri" tidak banyak berpengaruh, seiring waktu tidak akan ada koran yang tersisa untuk dijual.

Contoh 4 (Tragedi Tanah Milik Bersama): Sebuah desa memiliki n petani, dan padang rumput terbatas. Masing-masing petani ini dapat memutuskan untuk memelihara seekor domba. Setiap petani memperoleh beberapa manfaat dari domba-domba ini dalam bentuk wol dan susu. Namun, padang rumput umum (tanah milik bersama) akan sedikit dirugikan oleh domba yang merumput di sana.

Solusi: Solusinya dijelaskan di bawah ini. Solusi ini pertama kali dijelaskan oleh Hardin dalam *The Tragedy of the Commons*.

Ketika beberapa individu bertindak secara independen dan mereka mengutamakan kepentingan pribadi mereka, mereka akan menghabiskan sumber daya bersama. Hal ini akan terjadi meskipun tidak menguntungkan siapa pun dalam jangka panjang. Awalnya dirumuskan dalam konteks domba dan padang rumput biasa, hal ini dapat dengan mudah diterapkan pada banyak teka-teki kontemporer. Iklim global adalah salah satu contohnya. Memproduksi produk memang lebih murah jika sedikit atau tanpa perhatian diberikan pada pengendalian polusi dan emisi, tetapi tentu saja merugikan kesehatan bumi dalam jangka panjang. Contoh yang bagus untuk skenario ini tersedia di *www.skepticalscience.com*.

12.4 MASALAH TERKAIT

Masalah yang tampaknya paling mirip dengan Dilema Tahanan adalah permainan Chicken. Ini adalah situasi di mana dua pengemudi berada di jalur tabrakan satu sama lain. Mereka diberi pilihan untuk menjauh (menghindar) atau tetap di jalur. Jika keduanya tetap di jalur, kemungkinan terburuk, yaitu tabrakan, akan terjadi. Menariknya, meskipun kedua peserta mungkin menyadari kemungkinan terburuk dan kemungkinan terbaik, beberapa orang tetap memilih untuk tidak bekerja sama. Hal ini mungkin disebabkan oleh alasan bahwa kedua pengemudi akan berpikir bahwa pengemudi lain akan menjauh untuk menghindari tabrakan.

Hal ini juga mengingatkan kita pada jenis "eskalasi" yang dapat terjadi ketika ada peringatan nuklir palsu. Masing-masing pihak dapat meningkatkan tingkat responsnya dalam situasi seperti itu tanpa mengetahui tingkat reaksi pihak lain. Ini adalah keadaan yang sangat berbahaya.

BAB 13
SUDOKU

13.1 PENDAHULUAN

Sudoku adalah teka-teki logika yang menampilkan kotak angka 9×9 dengan sembilan sub-kotak 3×3. Setiap baris, kolom, dan sub-kotak harus berisi semua digit dari $1 - 9$, tanpa digit yang muncul dua kali di area tersebut. Permainan siap pakai dilengkapi dengan beberapa kotak yang sudah terisi, dan tujuannya adalah untuk mengisi seluruh kotak. Pembuat teka-teki tertentu dapat mengontrol tingkat kesulitannya; semakin banyak "petunjuk" yang diberikan, semakin mudah permainan tersebut dipecahkan. Namun, teka-teki yang "tersusun dengan baik" akan selalu hanya memiliki satu solusi unik, bahkan teka-teki yang dimulai dengan lebih sedikit kotak terisi.

Asal mula paling mendasar dari teka-teki Sudoku dapat ditelusuri kembali ke Teka-teki "Kotak Ajaib" yang ditemukan di Tiongkok sejak 650 SM, yang terdiri dari kotak angka berbentuk persegi, di mana setiap baris, kolom, dan diagonalnya harus sama dengan angka yang sama. Teka-teki Kotak Ajaib akhirnya menyebar ke Persia, Arab, dan India, bahkan sampai ke Eropa.

Matematikawan Leonard Euler mengembangkan Kotak Ajaib yang lebih kompleks, yang disebut Kotak Yunani-Romawi. Penggunaan angka serta persyaratan apa pun terkait diagonal dihilangkan, digantikan dengan menggabungkan huruf Yunani dan Romawi di setiap kotak sedemikian rupa sehingga semuanya muncul sekali di setiap baris dan kolom. Konsep ini sangat mirip dengan Sudoku saat ini. Kotak Yunani-Romawi yang lengkap mungkin terlihat seperti ini:

$A\alpha$	$B\gamma$	$C\delta$	$D\beta$
$B\beta$	$A\delta$	$D\gamma$	$C\alpha$
$C\gamma$	$D\alpha$	$A\beta$	$B\delta$
$D\delta$	$C\beta$	$B\alpha$	$A\gamma$

Gambar 13.1 Kotak Yunani-Romawi

Pada tahun 1979 di New York, "Dell Puzzle Magazine" menerbitkan sebuah permainan yang disebut "Number Place" yang memperkenalkan kisi angka 9×9 dengan sentuhan tambahan berupa penambahan kendala sub-kisi 3×3, serta persyaratan bahwa hanya ada satu solusi unik. Konsepnya secara populer dikaitkan dengan Howard Garnes.

Selama beberapa dekade berikutnya, permainan ini mendapatkan popularitas yang luar biasa di Jepang, di mana ia dicetak dan diedarkan oleh majalah "Nikoli" dengan nama Su Doku, yang diterjemahkan sebagai "angka tunggal" (yaitu setiap angka hanya muncul sekali). Nikoli juga memperkenalkan kriteria "simetri," persyaratan bahwa pengaturan awal permainan harus berupa bentuk yang akan mempertahankan bentuk dasarnya meskipun diputar. Hal ini membuatnya lebih menarik secara estetika serta umumnya lebih mudah untuk dipecahkan. Mungkin sifat bahasa Jepang, yang kurang cocok untuk teka-teki silang dan teka-teki kata lainnya, yang menyebabkan permainan angka begitu menarik.

Seorang hakim Selandia Baru bernama Wayne Gould jatuh cinta pada Sudoku setelah melihatnya di sebuah publikasi Jepang, dan ia berperan penting dalam mendorong "Times" London untuk mulai menerbitkannya pada tahun 2004. Hasilnya, Sudoku menjadi sangat populer di Inggris, dan dengan cepat menyebar ke Amerika Serikat dan seluruh dunia.

Sebelum membahas Sudoku secara mendalam, mari kita definisikan istilah-istilah yang akan kita gunakan untuk menggambarkan posisi dan entitas dalam teka-teki ini: "Kisi" adalah keseluruhan lingkungan permainan 9×9. "Sel" adalah unit kotak terkecil tempat sebuah angka ditempatkan. "Baris" dan "kolom" sudah cukup jelas, masing-masing berisi 9 sel. Setiap baris akan diberi label angka $1 - 9$(i), dan setiap kolom akan diberi label angka 1 - 9(j). Sebuah sel tertentu dapat dideskripsikan berdasarkan posisi (i, j)-nya. Terdapat 9 "blok", yang terdiri dari sub-kisi yang masing-masing berisi 3×3 sel dan merupakan larik 3×3. Sebuah blok akan dipanggil berdasarkan lokasi sel kiri atasnya. "Sub-baris" dan "sub-kolom" adalah segmen dari baris atau kolom yang berisi 3 sel dan berada dalam satu blok. Terakhir, "pita" mengacu pada baris blok, dan "tumpukan" mengacu pada kolom blok.

13.2 ANALISIS MATEMATIKA

Selama bertahun-tahun, diketahui bahwa terdapat teka-teki Sudoku yang sah yang diinisialisasi hanya dengan 17 sel terisi, sementara belum pernah ada yang berhasil menyusun teka-teki 16 petunjuk dengan solusi unik. Oleh karena itu, diduga bahwa teka-teki semacam itu tidak ada. Baru-baru ini, (1 Januari 2012) dalam sebuah pembuktian yang diunggah daring oleh Gary McGuire dari University College Dublin, dan disetujui oleh para matematikawan dalam sebuah konferensi beberapa hari kemudian (7 Januari), ditunjukkan bahwa jumlah minimum petunjuk yang dibutuhkan untuk menyelesaikan sebuah teka-teki adalah 17; teka-teki dengan 16 petunjuk atau kurang tidak dapat memiliki solusi unik. Kebanyakan teka-teki koran memiliki sekitar 25 petunjuk, dengan tingkat kesulitan teka-teki yang menurun seiring bertambahnya petunjuk.

Strategi pembuktian ini adalah dengan membuat algoritma yang dapat memeriksa kisi Sudoku lengkap untuk semua susunan "$\mathrm{n} - \mathrm{petunjuk}$" yang menghasilkan kisi tersebut sebagai solusi uniknya. Idenya adalah untuk mengulangi setiap kemungkinan teka-teki Sudoku yang terisi penuh, dan menerapkan algoritma untuk $\mathrm{n} = 16$ untuk melihat apakah ada kisi lengkap yang merupakan solusi unik dari 16 petunjuk. Algoritma tersebut menggunakan gagasan "himpunan-hitting" yang, singkatnya, mencari "himpunan yang tidak dapat dihindari," atau susunan angka yang dapat disusun ulang, untuk menghasilkan petunjuk yang memaksa himpunan tersebut untuk tetap di tempatnya. Baik algoritma itu sendiri maupun aplikasinya dalam mencari semua kisi Sudoku yang mungkin adalah perhitungan yang tidak mungkin dilakukan dengan tangan. Oleh karena itu, komputer dilibatkan untuk tugas tersebut.

Meskipun demikian, dibutuhkan jutaan jam inti pada superkomputer untuk menyimpulkan bahwa tidak ada teka-teki 16 petunjuk.

Masalah matematika menarik lainnya yang terkait dengan Sudoku, yang juga menggunakan bantuan komputer dalam penyelesaiannya, adalah pertanyaan tentang berapa banyak kisi Sudoku yang unik dan lengkap yang ada. Dengan menggunakan teknik penghitungan dasar, bahkan orang awam dapat membuat perkiraan kasar tentang jumlah kemungkinan cara untuk mengisi kisi Sudoku yang kosong. Dimulai dengan asumsi (yang jelas sederhana) bahwa setiap sel memiliki **tidak lebih dari** sembilan kemungkinan, ini menghasilkan total "batas atas" sebesar 9^{81}, atau 2×10^{77}, atau, dengan kata lain, dua ratus quattuorvigintillion. Jumlah tersebut tidak jauh dari jumlah atom di alam semesta, yang diperkirakan antara 10^{79} dan 10^{80}!

Namun, ketika kita menerapkan batasan baris, kolom, dan blok,* hal itu memungkinkan kita untuk memulai dengan sembilan kemungkinan untuk setiap kotak awal yang sembarang (mari kita asumsikan kiri atas untuk kenyamanan). Sekarang kita dapat membuat 3 kisi, masing-masing mewakili fokus pada satu batasan. Jumlah kemungkinan untuk setiap sel akan direpresentasikan oleh angka di dalam sel tersebut.

Kisi 1 (Gambar 13.2) akan memiliki semua sel teratas yang direpresentasikan sebagai sembilan kemungkinan, berkurang satu per satu saat Anda melanjutkan ke bawah setiap kolom. Dengan demikian:

9	9	9	9	9	9	9	9	9
8	8	8	8	8	8	8	8	8
7	7	7	7	7	7	7	7	7
6	6	6	6	6	6	6	6	6
5	5	5	5	5	5	5	5	5
4	4	4	4	4	4	4	4	4
3	3	3	3	3	3	3	3	3
2	2	2	2	2	2	2	2	2
1	1	1	1	1	1	1	1	1

Gambar 13.2 Kotak 1

Kotak 2 (Lihat Gambar 13.3) akan memiliki sembilan kemungkinan untuk setiap sel paling kiri, berkurang satu seiring Anda melanjutkan baris.

9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1

9	8	7	6	5	4	3	2	1
9	8	7	6	5	4	3	2	1

Gambar 13.3 kisi 2

Terakhir, kisi 3 (Lihat Gambar 13.4) akan mengisi setiap blok (dimulai dengan sembilan opsi di kiri atas setiap blok dan berkurang seiring Anda menelusuri setiap sub-baris di dalam blok).

9	8	7	9	8	7	9	8	7
6	5	4	6	5	4	6	5	4
3	2	1	3	2	1	3	2	1
9	8	7	9	8	7	9	8	7
6	5	4	6	5	4	6	5	4
3	2	1	3	2	1	3	2	1
9	8	7	9	8	7	9	8	7
6	5	4	6	5	4	6	5	4
3	2	1	3	2	1	3	2	1

Gambar 13.4 Kisi 3

Untuk kisi terakhir kita (Gambar 13.5), kita menetapkan setiap sel dengan angka terkecil yang ditemukan di tiga kisi pertama untuk sel tersebut, yang mewakili jumlah kemungkinan untuk sebuah sel dengan mempertimbangkan aturan paling ketat yang berlaku di dalamnya.

9	8	7	6	5	4	3	2	1
6	5	4	6	5	4	3	2	1
3	2	1	3	2	1	3	2	1
6	6	6	6	5	4	3	2	1
5	5	4	5	5	4	3	2	1
3	2	1	3	2	1	3	2	1
3	3	3	3	3	3	3	2	1
2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1

Gambar 13.5 Kisi 4

Untuk estimasi kasar terakhir, kami mengalikan angka-angka pada kisi di atas untuk menemukan jumlah total kemungkinan. Hasilnya kira-kira $1{,}5 \times 10^{31}$, angka yang jauh lebih mudah dipahami, dan jauh lebih mendekati nilai yang benar.

Pada tahun 2005, Bertram Felgenhauer dan Frazer Jarvis menghitung jumlah total cara pengisian kisi Sudoku yang sebenarnya sesuai dengan semua aturan. Perhitungan yang digunakan cukup rumit, dan mereka menemukan bahwa jumlah kemungkinan kisi Sudoku 9×9 adalah $\mathrm{N} = 6.670.903.752.021.072.936.960$ yang kira-kira sama dengan $6{,}671 \times 10^{21}$.

Analisisnya melibatkan pengisian blok kiri atas, dan menghitung berapa banyak cara yang memungkinkan untuk mengisi sisa pita atas, dan akhirnya setiap tumpukan, untuk melengkapi kisi. Angka ini kemudian dikalikan dengan jumlah kemungkinan untuk blok kiri atas.

Sinopsis singkat teknik yang digunakan: Mulailah dengan fakta bahwa terdapat 9! kemungkinan untuk blok kiri atas. Ini berasal dari penggunaan aturan perkalian dasar, yang memungkinkan sembilan kemungkinan untuk sel pertama dan mengurangi satu untuk setiap sel berikutnya. Dengan melihat salah satu pengaturan tersebut, gunakan 6 diambil 3 sekaligus (yaitu 20) untuk menghitung berapa banyak cara enam angka yang tersisa untuk baris atas dapat dibagi menjadi dua sub-baris kosong. Dua dari 20 angka tersebut disebut baris atas "murni" (dengan 4, 5, 6 di satu sub-baris dan 7, 8, 9 di sub-baris lainnya), dan 18 lainnya disebut "campuran." Untuk setiap baris teratas murni, terdapat (3!)6 cara untuk mengisi pita teratas, dan untuk setiap baris teratas campuran, terdapat $3 \times (3!)^6$ opsi. Ini menghasilkan $2 \times (3!)^6 + 18 \times 3 \times (3!)^6 = 2.612.736$ kemungkinan cara untuk mengisi pita teratas untuk setiap blok kiri atas yang telah diselesaikan.

Menghitung semua penyelesaian grid untuk setiap pita teratas tersebut masih akan menjadi tugas yang monumental, sehingga para programmer mulai menyederhanakan angka tersebut. Mereka berhasil mempertimbangkan berbagai permutasi dan operasi yang akan menghasilkan jumlah set solusi yang sama (misalnya, jika seseorang menukar blok tengah atas dan kiri atas, solusinya akan memerlukan pertukaran seluruh tumpukan tersebut, tetapi jika tidak, hasilnya akan sama persis). Hal ini memungkinkan mereka untuk mengurangi jumlah total pita teratas yang perlu dipertimbangkan menjadi hanya 44. Sebuah program komputer membantu menentukan jumlah solusi untuk setiap kemungkinan tersebut. Setiap hasil tersebut dikalikan dengan jumlah pita teratas yang berbagi himpunan solusi tersebut, dan angka ini kemudian dikalikan dengan 9! (jumlah kemungkinan untuk blok kiri atas yang telah diselesaikan) dengan total 6.670.903.752.021.072.936.960! (tanda seru, bukan faktorial!)

Namun, terdapat "solusi simetris", yang pada dasarnya serupa satu sama lain dalam struktur, tetapi beberapa operasi sederhana dapat dilakukan pada opsi pertama yang mengubahnya menjadi opsi lain. Penggunaan program komputer juga dicantumkan di sini, untuk menghitung kelas-kelas simetri dan menggunakannya untuk menyempurnakan total. Melalui metode ini, dihitung lebih lanjut bahwa terdapat 3.546.146.300.288 kisi Sudoku yang **pada dasarnya berbeda**.

13.3 TEKNIK DAN STRATEGI PEMECAHAN MASALAH

Jadi, bagaimana cara menyelesaikan kotak yang terisi sebagian (seperti yang terdapat di majalah dan surat kabar)? Algoritma paling sederhana yang dapat dipikirkan siapa pun di jalan akan melibatkan coba-coba dan brute force. Isi sel pertama yang tersedia dengan angka terkecil yang tersedia, lanjutkan ke sel berikutnya, dan ulangi lagi. Jika Anda mengalami kesulitan, kembali ke sel terakhir yang dimaksud dan coba angka yang berbeda sebelum melanjutkan. Metode ini dijamin akan menemukan solusi, jika ada. Namun, mencoba memecahkan teka-teki Sudoku dengan cara ini secara manual pasti akan sangat berantakan dan membutuhkan waktu yang sangat lama.

Sebagai gantinya, mari kita mulai dengan beberapa aturan dasar yang memungkinkan kita mengisi sel kosong, kemudian kita dapat beralih ke teknik yang lebih luas yang memungkinkan kita mengeliminasi beberapa angka, dan akhirnya, kita dapat menganalisis strategi atau algoritma lengkap untuk menerapkan aturan-aturan tersebut dalam upaya menyelesaikan seluruh kotak. Aturan-aturan berikut adalah aturan kami sendiri, meskipun terinspirasi oleh aturan-aturan yang terdapat di SudokuDragon.com dan juga di: http://www.math.cornell.edu/~mec/Summer2009/meerkamp/Site/Solving_any_Sudoku_II.html.

Aturan Satu-Satunya Angka - Ini mungkin aturan yang paling sederhana. Aturan ini berfokus pada satu sel dan memungkinkan kita mengisinya jika hanya ada satu angka yang dapat diisi (berdasarkan sisa baris, kolom, blok, atau kombinasi dari beberapa angka di atas). Untuk mengilustrasikan hal ini, perhatikan bahwa sel (1,1) hanya memiliki satu kemungkinan angka yang dapat diisi. Angka tersebut adalah 1, karena 6 dan 7 sudah ada di baris tersebut.

				6			7	
2								
3								
5								
4								
9								
8								

Gambar 13.6 Aturan Satu-Satunya Angka

Aturan Satu-Satunya Sel - Alih-alih berfokus pada sel tertentu, fokus kita adalah pada angka tertentu. Kita memperluas cakupan ke seluruh baris, kolom, atau blok. Di dalam area tersebut, kita melihat apakah ada angka yang dibatasi pada sel tertentu. Misalnya, angka 3 mungkin hanya memiliki 3 tempat kosong dalam satu baris, 2 di antaranya dapat dihilangkan berdasarkan pertimbangan kolom dan/atau blok. Cari baris atau kolom yang cukup penuh dan Anda mungkin dapat menghilangkan angka dari dua dari tiga subbagian, dan pada subbagian ketiga, angka tersebut mungkin dibatasi pada salah satu dari tiga sel.

Dalam contoh ini, angka 3 dipaksa masuk ke sel (3,1) menggunakan aturan satu-satunya sel, karena subkolom bawah penuh, subkolom tengah memiliki angka 3 di bloknya, dan sudah ada angka 3 di baris atas.

					3			

1								
6	3							
4								
1								
5								

Gambar 13.7 Aturan Sel Tunggal

Aturan Dua dari Tiga - Penggunaan Aturan Sel Tunggal yang lebih luas: Strateginya adalah memindai seluruh pita atau tumpukan sekaligus. Periksa apakah ada angka yang sudah muncul di dua dari tiga baris atau kolom. Jika ada angka tersebut, angka tersebut sekarang dipersempit menjadi satu sub-baris atau sub-kolom. (Hanya satu baris/kolom yang tersedia dan hanya satu blok juga). Seringkali, angka tersebut dibatasi pada salah satu dari tiga sel tersebut.

Dalam contoh ini, di pita teratas, angka terakhir 1 dibatasi pada baris 2, dan selanjutnya ke sub-baris paling kanan. Akhirnya, kita melihat angka tersebut dibatasi pada sel (2,9).

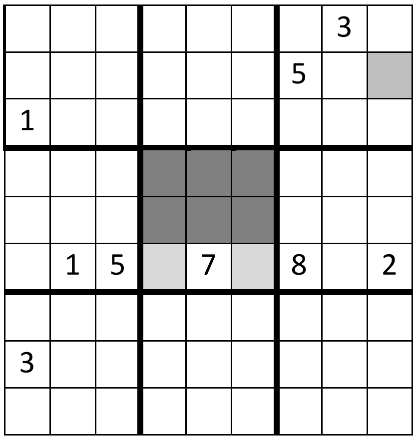
				1				
						5		
1								
							1	

Gambar 13.8 Aturan Dua Dari Tiga

Beberapa aturan sebenarnya tidak mengisi sel, melainkan berfungsi untuk mengeliminasi beberapa angka guna mempersempit pilihan untuk sebuah sel atau sekelompok sel:

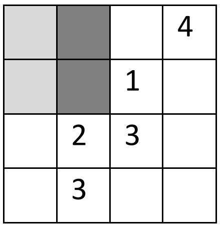
Aturan Pengecualian Sub-Grup - Dalam satu baris atau kolom, terkadang seseorang dapat membatasi angka tertentu ke salah satu dari tiga sub-bagian. Hal ini, pada gilirannya, mendiskualifikasi angka tersebut untuk ditempatkan di tempat lain di blok tersebut. Hal ini sebenarnya memungkinkan seseorang untuk mengisi sel yang sebelumnya memiliki dua pilihan.

Dalam contoh ini, untuk baris 6, angka 3 dieliminasi dari sub-baris pertama dan ketiga, sehingga mempersempitnya ke baris tengah, atau sel yang diarsir abu-abu. Ini mendiskualifikasi angka 3 dari sel merah.



Gambar 13.9 Aturan Pengecualian Subgrup

Aturan Pengecualian Kembar - Jika n elemen dipersempit menjadi n sel (misalnya 2 ke 2 atau 3 ke 3, dst.), maka elemen bebas yang tersisa akan dipersempit ke sel terbuka lainnya.



Gambar 13.10 Aturan Pengecualian Kembar

Aturan ini jauh lebih mudah diilustrasikan dengan kotak 4x4. Perhatikan bahwa dua angka 2 dan 3 didiskualifikasi dari sel yang diarsir merah dan dipaksa masuk ke dua sel abu-abu, sehingga warna merah hanya tersedia untuk 1 dan 4. (Sekarang kita sebenarnya dapat mengisi kedua sel merah karena 4 tidak dapat masuk ke (1,2) dan 1 tidak dapat masuk ke (2,2)!)

Memecahkan Teka-Teki yang Lengkap

Sebelum memulai, saya suka menuliskan angka 1 sampai 9 di suatu tempat di sisi halaman untuk mencatat nilai yang masih tersedia. Setelah setiap baris, kolom, dan kotak berisi angka-angka tersebut, angka tersebut dapat dicoret dari daftar.

Tahap Satu

Sekarang kita siap untuk memulai. Selalu mulai menggunakan aturan pertama ("hanya angka" dan "hanya sel", menerapkan "dua dari tiga" untuk membuat "hanya sel" lebih efektif). Pindai kisi dengan cermat menggunakan salah satu aturan, dan ketika tidak ada lagi sel yang dapat diisi, beralihlah ke aturan lainnya. Jika ada sel yang terisi, Anda dapat beralih kembali ke aturan pertama, karena mungkin akan menghasilkan lebih banyak jawaban karena kisi sudah lebih lengkap. Lanjutkan beralih bolak-balik hingga Anda telah mengisi sel sebanyak mungkin.

Tahap Dua

Banyak teka-teki Sudoku dapat diselesaikan sepenuhnya hanya dengan menggunakan aturan di atas. Jika kisi Anda penuh, selamat! Jika tidak, gunakan pensil untuk menandai bagian bawah setiap sel kosong dengan semua kemungkinan angka yang ada di sana. Kita sekarang dapat menerapkan dua aturan terakhir, "pengecualian subgrup" dan "pengecualian kembar". Aturan pengecualian ini dapat digunakan untuk mencoret kemungkinan yang tertulis di beberapa sel, dan seringkali hal ini akan menyebabkan lebih banyak sel yang terpecahkan. Perlu diingat bahwa setiap kali sel baru terisi, tahap pertama dapat ditinjau kembali untuk melihat apakah ada aturan dasar yang relevan. Bahkan, sebagian besar teka-teki dapat diselesaikan sepenuhnya pada tahap ini.

Tahap Tiga

Jika ada sel yang masih kosong, kita sekarang harus meninjau kembali algoritma yang sangat sederhana yang telah kita mulai, sambil memodifikasinya agar lebih praktis dan efisien. Pertama, kita mulai dengan menggunakan metode lain untuk mengisi sebagian besar kisi. Hal itu sendiri secara drastis mengurangi jumlah kemungkinan untuk sel yang tersisa, dan pada Tahap Dua kita mengurangi jumlah tersebut lebih jauh lagi. Sekarang, alih-alih mengisi setiap sel secara membabi buta dengan angka yang tersedia dan melanjutkan hingga muncul masalah, kita dapat melanjutkan dengan beberapa pertimbangan. Jika satu sel memiliki dua kemungkinan dan yang lain memiliki lebih banyak, selalu mulai dengan sel dengan opsi paling sedikit. Selain itu, Anda harus sangat berhati-hati agar semua pekerjaan tetap rapi dan memantau kemajuannya. Saya rasa menggunakan beberapa pensil dengan warna berbeda sangat membantu.

Setelah sel dipilih untuk dikerjakan, pilih salah satu kemungkinannya secara acak dan catat dengan pensil. Jika Anda sekarang dapat melengkapi kisi, Anda sudah selesai! Jika masih ada lebih dari satu kemungkinan di sel, Anda harus mencari cabang baru, membuat tebakan baru, dan melanjutkan (sebaiknya dengan warna yang berbeda). Jika suatu saat Anda mengalami kebuntuan, Anda telah memastikan bahwa angka terakhir yang Anda pilih secara acak bukanlah pilihan dan Anda harus menghapus semua entri tentatif dari warna saat ini dan mencoba angka lain. (Jika Anda belum berada di level pertama menebak, Anda mungkin telah menghabiskan semua kemungkinan tanpa solusi, yang dalam hal ini Anda harus kembali dan mengubah level sebelumnya.) Meskipun tahap terakhir ini terdengar lambat dan membosankan, Anda telah melakukan segala yang mungkin sebelumnya untuk mempersiapkan dan menyederhanakan semuanya semaksimal mungkin, dan kabar baiknya adalah Anda dijamin (pada akhirnya) akan menemukan solusinya!

13.4 EKSPERIMEN KEHIDUPAN NYATA

Untuk mendapatkan gambaran tentang bagaimana teka-teki Sudoku dan solusi di atas diterapkan dalam kehidupan nyata, ditemukan seorang subjek yang tidak terlalu sering bermain dan belum pernah mencoba memecahkannya. Tujuan eksperimen ini adalah untuk mengamati percobaan awal dan melihat bagaimana subjek mencoba memecahkan teka-teki tersebut. Subjek kemudian akan membaca bagian sebelumnya dan baru setelah itu mencoba teka-teki lain dengan tingkat kesulitan yang setara. Kemudian akan dilihat apakah ada peningkatan yang signifikan selama percobaan.

Subjek tersebut adalah seorang perempuan berusia 23 tahun yang tampaknya cukup cerdas. Ia diberi "teka-teki hari ini" dari *sudokutoday.com* yang memiliki peringkat untuk kategori "mudah". Setelah mengerjakan selama sekitar dua puluh menit, dan menemui beberapa jalan buntu, ia berasumsi bahwa ia memerlukan semacam metodologi coba-coba yang memakan waktu untuk melanjutkan. Pada titik itu, ia memutuskan untuk berhenti setelah menyelesaikan sekitar setengah dari kisi-kisi.*

Sangat menarik melihat bagaimana ia secara alami dan spontan menggunakan kombinasi aturan "hanya sel" dan "hanya angka". Ini sebenarnya cukup intuitif, karena aturan-aturan ini menguraikan cara-cara paling dasar untuk mengisi sel, yang merupakan tujuan langsung di setiap langkah permainan. Namun, ia kurang lebih terus maju tanpa gambaran yang jelas tentang apa yang ia lakukan, yang menyebabkan beberapa kesalahan di awal, seperti mengisi sel dengan angka yang tidak sepenuhnya dipaksakan pada posisi tersebut.

Ia sangat ingin mendapatkan panduan sehingga ia segera membaca bagian sebelumnya dan menyatakan bahwa itu sangat masuk akal baginya. Karena ia belum memecahkan teka-teki asli pada percobaan pertamanya, ia diberi salinan teka-teki yang sama yang belum terisi pada percobaan keduanya. Perbedaannya sangat drastis. Kali ini, ia secara sistematis dan akurat mengisi sebagian besar sel dengan bergantian di antara beberapa aturan pertama, lalu menuliskan kemungkinan yang tersisa demi kejelasan dan berhasil menyelesaikan teka-teki dalam waktu sekitar dua puluh lima menit.

Berikut adalah penggambaran langkah demi langkah teka-teki yang digunakan dalam percobaan dan solusinya:

Ini adalah teka-teki kosong. Pada langkah berikutnya, kami akan menunjukkan bagaimana kesembilan angka sembilan dapat diisi menggunakan "aturan dua dari tiga".

				6		5		
	9	7			4		8	
6			2				7	
	4		3		9	7		
9								5
		2	1		5		3	
	3				2			9
	8		7			6	5	
		6		4				

Gambar 13.11 Teka-Teki Kosong

				6		5	9	
	9	7			4		8	
6			2	9			7	
	4		3		9	7		
9								5
		2	1		5	9	3	
	3				2			9
	8	9	7			6	5	
		6	9	4				

Gambar 13.12 Langkah 1

Misalnya, kita bisa mulai dengan pita tengah. Dua angka 9 yang ada berada di 2 baris teratas dan 2 kotak kiri. Hal ini membatasi angka 9 yang tersisa ke baris paling kanan bawah. Satu sel telah terisi, dan kolom paling kanan sudah memiliki angka 9, sehingga hanya tersisa sel (6,7).

Dengan hanya menggunakan "aturan dua dari tiga", kita dapat menyelesaikan semua angka enam, tujuh, dan sembilan, seperti yang ditunjukkan. Aturan ini telah digunakan untuk saat ini.

				6	7	5	9	
	9	7			4		8	6
6			2	9			7	
	4		3		9	7	6	
9	7				6			5
	6	2	1	7	5	9	3	
7	3		6		2			9
	8	9	7			6	5	
		6	9	4				7

Gambar 13.13 Langkah 2

Perhatikan bahwa baris ke-6 hanya memiliki angka 4 dan 8 yang hilang. Aturan "hanya sel" standar menempatkan angka 4 di (6,9). Aturan "hanya angka" menempatkan angka 8 di sel (6,1) yang tersisa.

				6	7	5	9	
	9	7			4		8	6
6			2	9			7	
	4		3		9	7	6	
9	7				6			5
8	6	2	1	7	5	9	3	4
7	3		6		2			9
	8	9	7			6	5	
		6	9	4				7

Gambar 13.14 Langkah 3

"Hanya sel" juga dapat digunakan pada kolom 8 untuk memasukkan angka "4" ke dalam (7,8), pada kotak kanan atas untuk memasukkan angka "4" ke dalam (3,7), dan pada baris 8 untuk memasukkan angka 4 ke dalam (8,1) dan angka 2 ke dalam (8,9).

				6	7	5	9	
	9	7			4		8	6
6			2	9		4	7	
	4		3		9	7	6	
9	7				6			5
8	6	2	1	7	5	9	3	4
7	3		6		2		4	9
4	8	9	7			6	5	2
		6	9	4				7

Gambar 13.15 Langkah 4

Kita sekarang dapat mengisi seluruh kotak dengan beralih di antara aturan "hanya sel", "hanya angka", dan "dua dari tiga". Mengisi beberapa kemungkinan "jika kita buntu" mungkin akan membantu, meskipun untuk permainan seperti ini hal itu sebenarnya tidak perlu.

2	1	4	8	6	7	5	9	3
3	9	7	5	1	4	2	8	6
6	5	8	2	9	3	4	7	1
1	4	5	3	2	9	7	6	8
9	7	3	4	8	6	1	2	5
8	6	2	1	7	5	9	3	4
7	3	1	6	5	2	8	4	9
4	8	9	7	3	1	6	5	2
5	2	6	9	4	8	3	1	7

Gambar 13.16 Langkah 5

Eksperimen ini dengan jelas menunjukkan bahwa meskipun seseorang mungkin sangat cerdas, sifat kompleks teka-teki Sudoku membuatnya sangat sulit dipecahkan tanpa pendekatan yang jelas dan sistematis. Hal ini juga menunjukkan bahwa pendekatan yang kami kembangkan di atas cukup sederhana untuk dipelajari dengan relatif cepat, dan tampaknya berhasil ketika digunakan.

13.5 ALGORITMA UNTUK SOLUSI KOMPUTER

Telah ditentukan bahwa algoritma yang paling cocok untuk digunakan komputer cukup mirip dengan "algoritma sederhana" yang disebutkan sebelumnya, yaitu menelusuri semua sel, menetapkan setiap sel ke angka pertama yang dapat dimasukinya, dan melanjutkan dengan cara tersebut di seluruh kisi. Jika macet, ia kembali ke sel terakhir yang dapat diubah dan mencoba angka yang berbeda. Meskipun metode ini terlalu lambat dan rumit untuk dipertimbangkan manusia, metode ini sangat tepat untuk komputer karena kesederhanaan dan efektivitasnya. Jenis algoritma ini dikenal sebagai "backtracking" dan diimplementasikan dengan menelusuri setiap sel kisi secara rekursif dan memeriksa apakah akan menghasilkan solusi. Contoh program yang menggunakan backtracking dapat ditemukan di CD-ROM.

13.6 ANALISIS JENDELA MANUSIA TERHADAP SOLUSI

KESULITAN
5/10

KERUMITAN
$O(n^m)$

NAMA	INT ATAU EXT?	INTENS./EKSTENS.	REPR.	JENDELA MANUSIA?	BENAR?	UKURAN BUTIR	EKSEKUSI	KETERPAHAMAN	METODE PEMECAHAN MASALAH	FLEKSIBILITAS	MODE KONVERSI	OPTIMAL?	TOTAL
Tips	Int.	9/10	Dese. dng. Kisi	Y	Y	Kecil	3/10	6/10	CS; Deduksi; Repr. Vis.	8/10	Generator Kisi; Tangan	N	26/40
Frekuensi	Ekst.	9/10	Dese. dng. Kisi	Y	Y	Ideal	9/10	8/10	CS; Deduksi; Repr. Vis.	8/10	Generator Kisi; Tangan	N	34/40
Tidak Berpikir	Int.	8/10	Dese. dng. Kisi	Y	Y	Ideal	10/10	9/10	Konstr. Puas.; Repr. Vis.	8/10	Generator Kisi; Tangan	N	35/40
Arsir Silang	Int.	5/10	Dese. dng. Kisi	Y	Y	Ideal	9/10	9/10	CS; Deduksi; Repr. Vis.	7/10	Generator Kisi; Tangan	N	30/40
Sederhana	Int.	9/10	Dese. dng. Kisi	Y	Y	Sangat Kecil	8/10	5/10	Deduksi; Repr. Vis.	6/10	Generator Kisi; Tangan	N	28/40
Acak	Int.	10/10	Dese. dng. Kisi	Y	Y	Kecil	5/10	5/10	Deduksi; Repr. Vis.	8/10	Generator Kisi; Tangan	N	28/40
Pensil Duluan	Int.	10/10	Dese. dng. Kisi	Y	Y	Ideal	6/10	4/10	Deduksi; Repr. Vis.	7/10	Generator Kisi; Tangan	N	27/40
Langkah demi Langkah	Ekst.	9/10	Dese. dng. Kisi	Y	Y	Ideal	10/10	10/10	CS; Deduksi; Repr. Vis.	9/10	Generator Kisi; Tangan	N	35/40

Kata kunci

Int or Ext?: Intensional atau Ekstensional?—Apakah solusinya bersifat intensional atau ekstensional?
Intly/Extly: Intensionalitas/Ekstensionalitas—Seberapa intensional atau ekstensional solusinya?
Rep.: Representasi—Bagaimana solusi tersebut direpresentasikan?
HW?: Jendela Manusia (Human Window)—Apakah solusi tersebut ada di dalam Jendela Manusia?
Corr?: Kebenaran—Apakah solusinya benar?
Grn Sz: Ukuran Butir (Grain Size)—Berapa banyak komputasi (besar) atau memori (kecil) yang dibutuhkan untuk menyelesaikan solusi tersebut?
Exec: Kemampuan Eksekusi—Seberapa bisa dijalankan solusi ini?
Compr: Komprehensibilitas—Seberapa mudah solusi ini dipahami?
Prob. Solv. Mthd: Metode Pemecahan Masalah—Metode apa yang digunakan dalam solusi untuk memecahkan masalah?
Flex: Fleksibilitas—Seberapa fleksibel solusi ini (yaitu, dapatkah solusi ini direpresentasikan dengan cara lain)?
Mode of Conv: Mode Penyampaian—Dengan cara apa solusi ini dapat direproduksi/direplikasi?
Opt?: Optimal—Apakah solusinya optimal?

Tabel 13.1Pemeringkatan Solusi Sudoku Berdasarkan Jendela Manusia

Teka-teki Sudoku tidak memiliki satu metode khusus untuk mendapatkan solusi. Oleh karena itu, terdapat banyak teknik dan metode pemecahan masalah yang tersedia untuk dianalisis berdasarkan Jendela Manusia. Akibatnya, kita dapat melihat bahwa pemeringkatan antara setiap solusi yang kita peroleh (Tabel 13.1) bervariasi berdasarkan sebagian besar kriteria.

Pertama, mari kita analisis kriteria yang diberi peringkat serupa di antara semua solusi dalam tabel. Saat melihat solusinya, kita dapat melihat bahwa semuanya direpresentasikan menggunakan deskripsi yang disertai kisi-kisi. Hal ini tidak mengherankan, karena teka-teki Sudoku adalah kisi-kisi 9×9 dan berbagai teknik yang digunakan untuk menyelesaikan teka-teki hanya dapat dijelaskan secara detail. Salah satu teknik yang paling umum digunakan di antara semua solusi, misalnya, adalah "crosshatching" yang memanfaatkan nilai yang diberikan teka-teki untuk menentukan nilai kotak kosong di kuadran tertentu. Teknik lain yang umum digunakan disebut "penciling in", yang menunjukkan kemungkinan pilihan untuk setiap kotak agar dapat melacak angka yang tersisa tanpa harus mengingatnya. Agar dapat dipahami, teknik-teknik seperti ini harus dijelaskan secara deskriptif disertai gambar.

Kedua, perlu dicatat bahwa metode pemecahan masalah yang digunakan dalam solusi-solusi ini utamanya adalah pemenuhan kendala dan deduksi. Hal ini juga tidak mengejutkan karena menemukan solusi untuk setiap teka-teki Sudoku melibatkan proses yang sama: menyelesaikan kotak-kotak yang tersisa dengan proses eliminasi melalui kendala nilai yang diberikan dan aturan-aturan Sudoku.

Selanjutnya, kita dapat mengamati bahwa sebagian besar solusi bersifat intensional, sementara sangat sedikit yang bersifat ekstensional. Solusi intensional menyediakan berbagai teknik yang dapat diterapkan pada teka-teki apa pun, tergantung pada kondisi teka-teki tersebut. Akibatnya, Ukuran Butir untuk solusi-solusi ini relatif kecil secara rata-rata mengingat pemecah harus mengingat teknik-teknik ini untuk menyelesaikan teka-teki. Selain itu, tingkat eksekusinya relatif rendah karena sebagian besar solusi intensional tidak mengikuti prosedur langkah demi langkah. Solusi ekstensional, di sisi lain, menyediakan panduan langkah demi langkah tentang cara memecahkan satu teka-teki tertentu. Solusi-solusi ini berperingkat lebih tinggi di sebagian besar kriteria dan memiliki Ukuran Butir ideal karena lebih lugas daripada solusi intensional.

Namun, meskipun berperingkat lebih tinggi, solusi ekstensional mungkin bukan pendekatan terbaik karena sebagian besar solusi hanya berlaku untuk satu solusi di antara banyak kemungkinan teka-teki Sudoku. Oleh karena itu, mungkin lebih baik untuk menyediakan langkah-langkah dan teknik umum yang dapat diterapkan pada solusi teka-teki apa pun. Setelah menganalisis solusi dari tabel dan peringkatnya secara detail, solusi ideal yang memungkinkan adalah solusi yang menyediakan teknik umum yang dapat digunakan untuk memecahkan teka-teki Sudoku apa pun. Namun, teknik-teknik ini seharusnya hanya membutuhkan sedikit memori. Selain itu, solusi ideal dapat mendemonstrasikan teknik-teknik ini dengan memecahkan teka-teki langkah demi langkah sebagai contoh, sehingga membuat solusi lebih mudah dieksekusi.

Akhirnya, kita dapat melihat bahwa peringkat untuk pemahaman solusi relatif beragam. Karena semua solusi direpresentasikan menggunakan kisi-kisi (misalnya, gambar teka-teki Sudoku) dan deskripsi, pemahaman solusi kemungkinan besar diukur melalui detail kisi-kisi di setiap langkah dan seberapa jelas deskripsinya. Dengan demikian, solusi ideal mungkin memiliki kisi-kisi yang jelas dan berlabel yang mewakili status teka-teki di setiap langkah, disertai deskripsi teknik yang sederhana namun ringkas dan bagaimana teknik-teknik tersebut diterapkan.

Solusi Jendela Paling Kompatibel dengan Manusia

SUDOKU

	6		1		4		5	
		8	3		5	6		
2								1

8			4		7			6
		6				3		
7			9		1			4
5								2
		7	2		6	9		
	4		5		8		7	

Langkah 1: Isi "Kotak-kotak yang Hilang"

123 456 789	6	123 456 789	1	123 456 789	4	123 456 789	5	123 456 789
123 456 789	123 456 789	8	3	123 456 789	5	6	123 456 789	123 456 789
2	123 456 789	123 456 789	123 456 789	123 456 789	123 456 789	123 456 789	123 456 789	1
8	123 456 789	123 456 789	4	123 456 789	7	123 456 789	123 456 789	6
123 456 789	123 456 789	6	123 456 789	123 456 789	123 456 789	3	123 456 789	123 456 789
7	123 456 789	123 456 789	9	123 456 789	1	123 456 789	123 456 789	4
5	123 456 789	123 456 789	123 456 789	123 456 789	123 456 789	123 456 789	123 456 789	2
123 456 789	123 456 789	7	2	123 456 789	6	9	123 456 789	123 456 789
123 456 789	4	123 456 789	5	123 456 789	8	123 456 789	7	123 456 789

Masukkan angka 1 sampai 9 dalam pola 'tic-tac-toe' di setiap kotak yang kosong.

Langkah 2: Hapus "Baris"

23 789	6	23 789	1	23 789	4	23 789	5	23 789
12 4 79	12 4 79	8	3	12 4 79	5	6	12 4 79	12 4 79
2	3 456 789	3 456 789	3 456 789	3 456 789	3 456 789	3 456 789	3 456 789	1
8	123 5 9	123 5 9	4	123 5 9	7	123 5 9	123 5 9	6
12 45 789	12 45 789	6	12 45 789	12 45 789	12 45 789	3	12 45 789	12 45 789
7	23 56 8	23 56 8	9	23 56 8	1	23 56 8	23 56 8	4
5	13 46 789	13 46 789	13 46 789	13 46 789	13 46 789	13 46 789	13 46 789	2

13 45 8	13 45 8	7	2	13 45 8	6	9	13 45 8	13 45 8
123 6 9	4	123 6 9	5	123 6 9	8	123 6 9	7	123 6 9

Angka pertama (di sudut kanan atas) adalah '6'. Hapus setiap '6' yang ditulis dengan pensil di baris tersebut

Langkah 3: Hapus "Kolom"

39	6	23 9	1	23 789	4	278	5	3 789
149	12 79	8	3	12 4 79	5	6	12 49	79
2	35 789	3 459	678	3 456 789	39	45 78	3 46 89	1
8	123 5 9	123 59	4	123 5 9	7	125	123 9	6
149	125 789	6	78	12 45 789	29	3	12 4 89	5 789
7	23 58	235	9	23 56 8	1	258	23 68	4
5	13 789	13 49	678	13 46 789	39	14 78	13 46 89	2
134	13 58	7	2	13 45 8	6	9	13 48	358
13 69	4	123 9	5	123 6 9	8	12	7	39

Sekarang, untuk angka '6' yang sama, hapus '6' yang ditulis dengan pensil di setiap kotak dalam kolom yang sama. Lakukan hal yang sama untuk semua angka lain yang tersedia di semua kolom.

Langkah 4: Hapus "Kuadran"

39	6	23 9	1	2 789	4	278	5	3 789
149	179	8	3	2 79	5	6	2 49	79
2	35 79	3 459	678	6 789	9	4 78	34 89	1
8	123 59	123 59	4	235	7	125	12 9	6
149	125 9	6	8	258	2	3	12 89	5 789
7	23 5	235	9	23 568	1	258	28	4
5	13 89	139	7	134 79	39	148	13 468	2

13	138	7	2	13 48	6	9	13 48	358
13 69	4	123 9	5	13 69	8	1	7	3

Hapus semua '6' yang ditulis dengan pensil di kuadran yang sama dengan '6' awal. Lakukan hal yang sama untuk semua angka lain yang tersedia di semua kuadran.

Langkah 5: Temukan Jawaban

39	6	23 9	1	2 789	4	278	5	3 789
149	179	8	3	2 79	5	6	2 49	79
2	35 79	3 459	678	6 789	9	4 78	34 89	1
8	123 59	123 59	4	235	7	125	12 9	6
149	125 9	6	8	258	2	3	12 89	5 789
7	23 5	235	9	23 568	1	258	28	4
5	13 89	139	7	134 79	39	148	13 468	2
13	138	7	2	13 48	6	9	13 48	358
13 69	4	123 9	5	13 69	8	1	7	36

Langkah 6: Eliminasi Lain

39	6	23 9	1	2 789	4	278	5	3 789
1	7	8	3	2 79	5	6	2 49	79
2	5	4	6	6 789	9	4 78	3	1
8	123 59	123 59	4	3	7	125	12 9	6
4	125 9	6	8	5	2	3	12 89	7
7	23 5	235	9	6	1	258	28	4
5	13 89	139	7	1	9	148	6	2
13	138	7	2	13 48	6	9	13 48	5
6	4	2	5	13 69	8	1	7	3

Lihatlah nilai-nilai tersisa yang ditulis dengan pensil di setiap baris. Jika Anda menemukan angka yang hanya muncul sekali dalam satu baris, maka itulah jawaban untuk kotak tersebut. Hal yang sama berlaku untuk kolom dan kuadran.

Gambar 13.17 Solusi "Tanpa Berpikir"

Solusi yang dinilai sebagai MHWC adalah "Solusi Tanpa Berpikir" (Gambar) (Contoh yang baik dari solusi ini dapat ditemukan di.) Solusi ini menggunakan satu teknik, yaitu menandai setiap kotak kosong dengan semua nilai yang mungkin, kemudian mengeliminasi kemungkinan tersebut menggunakan deduksi berdasarkan nilai yang diberikan dan aturan Sudoku.

Secara umum, solusinya adalah sebagai berikut:

1. Tandai angka 1 hingga 9 dengan pensil di semua kotak kosong dalam teka-teki.
2. Hilangkan salah satu nilai yang ditandai dengan pensil berdasarkan nilai awal yang diberikan dalam teka-teki. Jika hasilnya adalah kotak kosong dengan satu nilai yang ditandai dengan pensil, maka nilai kotak tersebut adalah nilai placeholder tersebut.
3. Telusuri semua kotak kosong untuk melihat apakah ada nilai placeholder yang tersisa yang unik untuk suatu baris, kolom, atau kuadran. Nilai placeholder yang unik tersebut berarti bahwa nilai kotak kosong tersebut adalah nilai yang tersisa.
4. Ulangi langkah 2 dan 3. Nilai baru yang diperoleh akan memungkinkan Anda untuk menghilangkan lebih banyak nilai placeholder. Ulangi langkah ini hingga nilai semua kotak kosong ditemukan.

Solusi ini dianggap sebagai MHWC karena merupakan solusi intensional yang menyediakan teknik lugas yang dapat diterapkan pada soal Sudoku apa pun dan menjamin penyelesaian teka-teki, apa pun statusnya. Meskipun teknik ini membosankan, teknik ini hanya membutuhkan sedikit memori atau bahkan tanpa memori (atau tanpa berpikir seperti namanya) untuk dieksekusi, sehingga menghasilkan Grain Size yang ideal. Selain itu, deskripsi di setiap langkah ringkas dan gambar detail teka-teki disediakan di setiap langkah yang menunjukkan statusnya saat ini, sehingga meningkatkan eksekusi dan pemahaman solusi secara keseluruhan.

Solusi Jendela yang Paling Tidak Kompatibel dengan Manusia

Memulai

		7	9					1
	2	3	8			6	7	
		6		2	7			
	7	8		5				
	5		2		6		3	
				1		9	5	
			6	3		8		
	8	4			9	2	1	
2					1	3		

Untuk menyelesaikan teka-teki Sudoku reguler, tempatkan angka ke dalam setiap kotak diagram sehingga setiap baris mendatar, setiap kolom, dan setiap blok dalam diagram besar (ada 9 blok) akan berisi setiap angka dari 1 sampai 9. Dengan kata lain, tidak ada angka yang boleh muncul lebih dari satu kali dalam baris, kolom, atau blok mana pun.
...
Proses eliminasi dasar yang digunakan pada contoh di atas menghasilkan Penyelesaian Langsung. Cara terbaik untuk mengerjakan teka-teki Sudoku adalah dengan menyelesaikan Penyelesaian Langsung terlebih dahulu, karena itu yang paling mudah.

...

Menemukan kandidat untuk dieliminasi adalah tempat teknik pemecahan tingkat lanjut—disebut Deduksi—berperan. Di bawah ini adalah beberapa teknik deduktif yang berbeda:

MEMULAI

7	2	1	369	34 69	34	34 59	45	8
5	34 659	34 69	1	234 69	8	23 49	7	349
39	8	349	239	7	5	6	124	13 49
239	5	23 79	239	1	234	23 47	8	6
8	139	239	7	23 49	6	12 34	124	5
4	136	23 67	5	8	23	12 37	9	137
12 39	139	8	4	5	12 37	179	6	179
136	7	346	8	36	9	145	145	2
12 69	14 69	5	26	26	127	8	3	14 79

Dalam locked candidate, sebuah nilai harus muncul pada persimpangan blok tertentu dan baris atau kolom, sehingga dapat dihapus sebagai kandidat dari sisa blok dan baris atau kolom tersebut. Dalam contoh, angka 6 di Baris A hanya dapat berada di Kuadran 2 (dengan kata lain, tidak ada sel di Baris A pada Kuadran 1 atau Kuadran 3 yang memiliki 6 sebagai kandidat). Oleh karena itu, angka 6 dapat dieliminasi sebagai kandidat dari semua sel lain di Kuadran 2—artinya, angka 6 dapat dihapus dari Baris B, Kolom 5.

NAKED PAIR

579	379	37	1	2	4	678	36 78	358
125 79	8	237	567	69	79	4	123 67	135
6	147	247	57	8	3	9	127	15
3	69	1	4	5	2	68	689	7
79	2	67	3	69	8	1	5	4
4	5	8	67	1	79	3	69	2
78	37	9	2	4	1	5	378	6
127	17	5	8	3	6	27	4	9
28	346	23 46	9	7	5	28	123 8	138

Dalam naked pair, dua sel dalam satu baris, kolom, atau blok masing-masing berisi dua kandidat yang sama, dan hanya kandidat tersebut. Jika naked pair muncul dalam satu baris, kolom, atau kuadran, dua kandidat tersebut dapat dihapus dari semua sel lain di baris, kolom, atau blok tersebut. Dalam contoh, satu-satunya kandidat yang mungkin untuk sel di Baris I, Kolom 1 dan Baris I, Kolom 7 adalah angka 2 dan 8, membentuk sebuah 'naked pair.' Karena 2 dan 8 harus ada di sel tersebut, maka tidak ada sel lain di Baris I yang bisa berupa 2 atau 8 (jika tidak, kita tidak dapat memberi nilai pada keduanya). Oleh karena itu, angka 2 dan 8 dapat dihapus dari semua sel lain di Baris I.

Gambar 13.18 "Solusi Tips"

Solusi yang dinilai sebagai LHWC adalah "Solusi Tips" (Gambar) (Contoh yang baik dari solusi ini dapat ditemukan di.) Solusi ini merupakan kumpulan teknik yang dapat diterapkan pada teka-teki Sudoku apa pun untuk menemukan nilai yang hilang. Teknik-teknik tersebut meliputi Locked Candidate, Hidden Pair, X-Wing, Swordfish, dll. Pemilihan teknik tertentu untuk diterapkan pada teka-teki sangat bergantung pada kondisi teka-teki saat ini. Dengan kata lain, teknik yang dapat digunakan pada setiap langkah ditentukan oleh posisi nilai awal dan nilai yang baru ditemukan yang tersedia pada langkah tersebut.

Solusi ini dinilai sangat intensional karena implisitnya solusi tersebut secara keseluruhan. Namun, solusi ini dianggap sebagai LHWC karena beberapa alasan. Pertama, seperti yang telah disebutkan sebelumnya, faktor penentu apakah teknik yang diberikan dapat dijalankan atau tidak bergantung pada kondisi teka-teki saat ini. Karena jumlah teknik yang disediakan solusi ini terbatas, teka-teki mungkin berada dalam kondisi di mana tidak ada teknik yang dapat diterapkan, sehingga menurunkan eksekusi solusi. Kedua, solusi ini tidak menyediakan prosedur langkah demi langkah tentang cara mencapai tujuan akhir teka-teki. Sebaliknya, teknik-teknik ini diterapkan pada teka-teki yang berbeda, sehingga semakin menurunkan eksekusi solusi. Ketiga, teknik yang disediakan rumit, dan mungkin kurang ideal bagi mereka yang belum berpengalaman dengan Sudoku, sehingga menurunkan pemahaman solusi secara keseluruhan. Terakhir, karena solusi ini mengharuskan pemecah tidak hanya mengingat teknik, tetapi juga kondisi teka-teki yang perlu dicapai agar teknik-teknik ini dapat diterapkan, Ukuran Butir solusi diperingkat Kecil.

Salah satu cara untuk meningkatkan solusi ini adalah dengan menyediakan prosedur langkah demi langkah tentang cara memecahkan satu teka-teki sebagai contoh dari awal hingga akhir, sambil mendemonstrasikan teknik-teknik yang mungkin berlaku pada waktu yang tepat dalam prosedur tersebut. Hal ini dapat meningkatkan eksekusi solusi secara keseluruhan. Selain itu, menggabungkan teknik-teknik kompleks ini dengan teknik yang mirip dengan "Solusi Tanpa Berpikir" akan memungkinkan pemecah teka-teki untuk menyelesaikan teka-teki meskipun berada dalam kondisi di mana solusi kompleks tidak dapat diterapkan, sehingga semakin meningkatkan eksekusi solusi tersebut.

BAB 14
PEWARNAAN PETA DAN BILANGAN KROMATIK

14.1 PENDAHULUAN

Masalah Pewarnaan Peta telah menjadi daya tarik bagi para matematikawan selama hampir dua abad. Matematikawan dan astronom August Ferdinand Möbius telah menyebutkan masalah ini dalam kuliahnya sejak tahun 1840. Pertanyaannya adalah sebagai berikut: Diberikan sebuah peta dengan sejumlah entitas, masing-masing berbentuk acak, berapa jumlah warna minimum yang dapat digunakan untuk mewarnai setiap entitas sehingga tidak ada dua warna yang berbagi batas? Bilangan ini disebut Bilangan Kromatik. Menarik untuk dicatat bahwa masalah yang berpusat pada peta ini jarang relevan bagi para pembuat peta, karena penggunaan pewarnaan minimum tidak kondusif untuk menghasilkan peta yang estetis, jelas, dan mudah dibaca.

Hipotesis paling awal yang tercatat diajukan pada tanggal 23 Oktober 1852, ketika Francis Guthrie memperhatikan bahwa hanya empat warna unik yang diperlukan untuk mewarnai peta wilayah-wilayah di Inggris. Ini kemudian dikenal sebagai "Teorema Empat Warna." Pada saat itu, saudara laki-laki Guthrie, Frederick, adalah seorang mahasiswa matematikawan terkenal Augustus De Morgan di University College di London. Francis bertanya kepada Frederick mengenai masalah tersebut, yang kemudian membawanya ke De Morgan. Dalam kata-kata De Morgan sendiri:

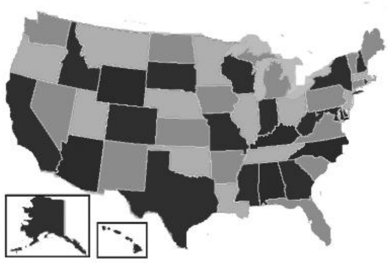
"Seorang mahasiswa saya [Guthrie] meminta saya hari ini untuk memberinya alasan untuk sebuah fakta yang tidak saya ketahui sebagai fakta — dan belum tahu. Dia mengatakan bahwa jika suatu bangun dibagi berapa pun dan kompartemennya diwarnai berbeda sehingga bangun dengan bagian mana pun dari garis batas yang sama diwarnai berbeda — empat warna mungkin diinginkan tetapi tidak lebih — berikut adalah kasusnya di mana empat warna diinginkan. Pertanyaan tidak dapat menemukan keharusan untuk lima atau lebih... ".

Pada tahun 1854 masalah tersebut diajukan di majalah sastra London "The Athenaeum" oleh seorang "F.G." yang mungkin salah satu dari Guthrie bersaudara. De Morgan sendiri kembali mengajukan pertanyaan tersebut di majalah yang sama pada tahun 1860.

Permasalahan ini dapat dinyatakan secara lebih ringkas dan matematis dalam istilah teori graf. Sebuah graf berisi titik-titik yang disebut simpul yang terhubung dengan garis yang disebut *sisi*. Graf planar adalah graf yang tidak ada sisi yang dipaksa berpotongan. Setiap peta dapat diubah menjadi graf planar dengan memungkinkan entitas direpresentasikan sebagai simpul dan sisi di antara keduanya merepresentasikan batas. Dengan menggunakan istilah-istilah ini, Teorema Empat Warna dapat dinyatakan kembali sebagai berikut: "setiap graf planar dapat diwarnai empat", yang berarti bahwa empat warna dapat diberikan pada simpul sedemikian rupa sehingga tidak ada sisi yang menghubungkan dua simpul dengan warna yang sama.

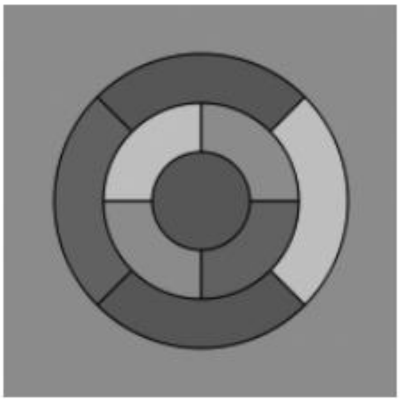
14.2 MENGILUSTRASIKAN TEOREMA

Gambar 14.1 menunjukkan peta Amerika Serikat yang hanya diisi dengan empat warna. Meskipun kurang sedap dipandang, setiap wilayah terdefinisi dengan jelas, tanpa dua warna yang saling berbatasan.



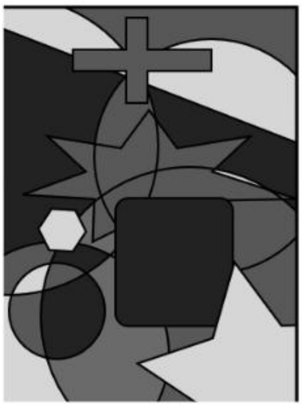
Gambar 14.1 Peta Amerika Serikat

Gambar 14.2 menunjukkan bahwa bahkan bentuk dan wilayah yang tidak biasa pun sesuai dengan Teorema Empat Warna.



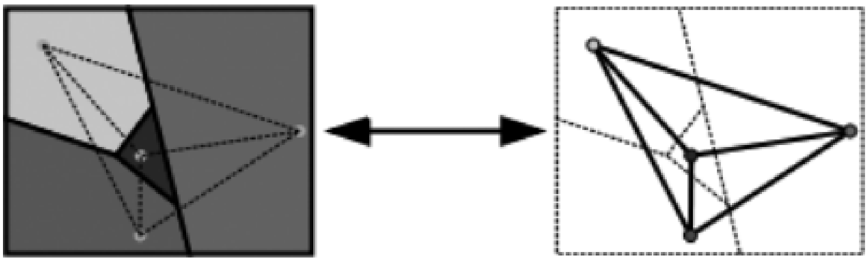
Gambar 14.2 Bentuk-Bentuk Tidak Biasa 1

Gambar 14.3 adalah contoh lain dari peta kompleks dengan bentuk-bentuk yang sangat tidak biasa yang dapat diwarnai hanya dengan empat warna.



Gambar 14.3 Bentuk-bentuk Tidak Biasa 2

Gambar 14.4 menggambarkan sebuah "peta" empat wilayah dengan lima batas, dan mengilustrasikan bagaimana peta tersebut dapat dengan mudah diubah menjadi sebuah graf dengan empat titik sudut dan lima sisi. Perhatikan bahwa graf tersebut planar (tidak ada sisi yang berpotongan).



Gambar 14.4 Peta Lima Batas

14.3 UPAYA AWAL PEMBUKTIAN

De Morgan sendiri percaya bahwa dasar Teorema ini dapat dinyatakan secara sederhana dan logis sebagai fakta tentang empat wilayah mana pun, dan ia tidak percaya bahwa dasar tersebut dapat diturunkan dari sesuatu yang lebih mendasar. Seperti yang ia nyatakan, "Saya segera menyimpulkan hal berikut - yang awalnya tidak masuk akal - kemudian tentu saja benar - kemudian menjadi aksiomatik - karena saya tidak dapat membuatnya bergantung pada apa pun yang saya lihat lebih jelas."

Selama beberapa dekade berikutnya, terdapat beberapa upaya yang gagal untuk membuktikan Teorema Empat Warna. Pada tahun 1879, Alfred Kempe mengajukan satu bukti yang diduga dan diterima secara luas. Bukti ini kemudian diikuti oleh bukti lain pada tahun 1880, yang diberikan oleh Peter Guthrie Tait. Setiap bukti yang salah berdiri tanpa tertantang selama sebelas tahun, hingga pada tahun 1890 Percy Heawood menunjukkan bukti Kempe salah, dan pada tahun 1891 Julius Peterson menunjukkan bukti Tait salah. Namun, sepanjang proses akademis yang mulia ini dalam mengejar kebenaran, ada beberapa penemuan yang berguna dan akurat. Ini termasuk penemuan rantai Kempe, bukti untuk Teorema Lima Warna, generalisasi Teorema Empat Warna ke jenis permukaan lainnya, dan penemuan (oleh Tait) bahwa Teorema Empat Warna setara dengan pernyataan bahwa jenis grafik tertentu (disebut snark) harus non-planar, membuka bidang penelitian baru dalam teori grafik.

Baru pada paruh kedua abad kedua puluh kemajuan nyata pada bukti yang berhasil benar-benar dimulai. Pada tahun enam puluhan dan tujuh puluhan, matematikawan Jerman Heinrich Heesch mengembangkan penggunaan metode yang disebut *discharging* untuk digunakan bersama dengan program komputer. Singkatnya, pelepasan muatan melibatkan penetapan "muatan" ke titik-titik dalam graf planar dan perancangan serta implementasi "fungsi muatan" dengan aturan yang memungkinkan pembuktian bahwa semua graf dari kategori tertentu memuat beberapa subgraf dari daftar yang telah disusun sebelumnya. Ia juga memperluas konsep reduksibilitas (bahwa suatu konfigurasi tertentu dapat digantikan dengan sesuatu yang lebih kecil tanpa memengaruhi bilangan kromatiknya) dan mengembangkan metode berbasis komputer untuk mengujinya. Namun, karena keterbatasan teknis dan finansial, ia tidak dapat melakukan proses tersebut sendiri.

Pada tanggal 21 Juni 1976, Kenneth Appel dan Wolfgang Haken dari University of Illinois mengumumkan bahwa mereka akhirnya memecahkan pertanyaan yang telah membingungkan dunia matematika selama lebih dari satu abad. Metode yang mereka gunakan didasarkan pada gagasan bahwa jika ada grafik planar yang membutuhkan lebih dari 4 warna, contoh tandingan minimal yang berisi daerah sesedikit mungkin harus ada, dan beberapa ketidakmungkinan kemudian harus benar. Dua konsep yang digunakan untuk mengembangkan bukti ini dengan kontradiksi adalah "himpunan yang tidak dapat dihindari" dan "konfigurasi yang dapat direduksi." Yang pertama adalah himpunan konfigurasi yang, jika ada grafik minimal yang tidak dapat diwarnai empat, itu harus sesuai dengan sesuatu dalam kelompok ini. Namun, "konfigurasi yang dapat direduksi" adalah peta yang dapat direduksi menjadi peta yang lebih kecil sedemikian rupa sehingga nomor kromatik tidak terpengaruh. Bukti Appel dan Haken dilanjutkan dengan mengembangkan "himpunan tak terelakkan" dari 1476 konfigurasi, dan kemudian membuktikan bahwa semua 1476 konfigurasi tersebut dapat direduksi, yang pada dasarnya menunjukkan bahwa tidak ada graf planar yang tidak dapat diwarnai empat.

Aspek baru dan inovatif dari bukti ini, bagaimanapun, bukanlah pada strateginya. Melainkan pada cara di mana bukti itu benar-benar dihitung. Karena meskipun himpunan tak terelakkan dibuat dan diperiksa oleh tangan manusia (mengisi lebih dari 400 halaman mikrofis), membuktikan bahwa setiap konfigurasi dapat direduksi memerlukan penggunaan komputer dan itupun membutuhkan lebih dari 1000 jam waktu komputasi! Tentu saja, berita tentang bukti ini menggemparkan dunia. Di satu sisi, salah satu teka-teki matematika yang paling terkenal dan sulit dipahami akhirnya terpecahkan. Di sisi lain, meskipun masalahnya sendiri begitu sederhana sehingga anak sekolah mana pun dapat memahaminya, solusinya begitu rumit sehingga sebagian darinya tidak dapat benar-benar diperiksa dengan tangan, dan bagian yang tampaknya bisa, masih sangat panjang dan membosankan. Dalam kata-kata Appel dan Haken sendiri:

"Hal ini membuat pembaca menghadapi 50 halaman berisi teks dan diagram, 85 halaman berisi hampir 2500 diagram tambahan, dan 400 halaman mikrofis yang berisi diagram lebih lanjut dan ribuan verifikasi individual atas klaim yang dibuat dalam 24 lema di bagian utama teks. Selain itu, pembaca diberi tahu bahwa fakta-fakta tertentu telah diverifikasi dengan menggunakan sekitar seribu dua ratus jam waktu komputer dan akan sangat memakan waktu untuk memverifikasi dengan tangan. Makalah-makalah tersebut agak mengintimidasi karena gaya dan panjangnya dan hanya sedikit matematikawan yang membacanya secara detail".

Namun, selama bertahun-tahun, bukti tersebut tampaknya telah bertahan dalam ujian waktu, dan meskipun masih ada yang menentangnya, bukti tersebut kurang lebih telah diterima sebagai valid. Karya besar setebal 741 halaman yang diterbitkan Appel dan Haken pada tahun 1989 jelas membantu dengan mengklarifikasi bukti secara ketat dan membelanya terhadap banyak klaim ketidaksempurnaannya. Referensi untuk Appel dan Haken.

Dalam beberapa tahun terakhir, telah terjadi penyempurnaan lebih lanjut pada bukti yang membawanya semakin dekat ke penerimaan luas. Pada tahun 1996, Neil Robertson, Daniel Sanders, Paul Seymour, dan Robin Thomas, dari Georgia Tech dan Universitas Ohio, mengembangkan bukti yang mirip dengan milik Appel dan Haken, namun disederhanakan untuk menampilkan serangkaian konfigurasi yang tak terelakkan yang hanya terdiri dari 633 konfigurasi yang terbukti dapat direduksi. Meskipun versi ini tampak jauh lebih baik, masih hampir mustahil untuk diperiksa secara manual, dan oleh karena itu masih rentan terhadap kritik bahwa kebenaran bukti tersebut bergantung pada program komputer. Robertson, Sanders, Seymour, dan Thomas mengajukan bukti alternatif pada tahun 2001 dengan membuktikan Teorema Snark, yang, sebagaimana telah disebutkan sebelumnya, setara dengan Teorema Empat Warna.

Bagian terakhir dari teka-teki ini terisi pada tahun 2005, ketika Benjamin Werner dan Georges Gonthier menggunakan asisten pembuktian terkomputerisasi bernama Coq untuk benar-benar memverifikasi himpunan tak terelakkan dan bagian redusibilitas dari pembuktian yang ada. Dalam kata-kata Gonthier sendiri:

"Pekerjaan kami dapat dilihat sebagai langkah pamungkas dalam upaya klarifikasi ini, yang sepenuhnya menghilangkan dua mata rantai terlemah dari pembuktian: verifikasi manual argumen kombinatorial, dan verifikasi manual bahwa program komputer khusus mengisi bagian-bagian argumen tersebut dengan benar. Untuk mencapai hal ini, kami telah menulis skrip pembuktian formal yang mencakup bagian matematika dan komputasional dari pembuktian. Kami telah menjalankan skrip ini melalui sistem pemeriksaan pembuktian Coq, yang secara mekanis memverifikasi kebenarannya dalam segala hal. Oleh karena itu, meskipun kebenaran pembuktian kami masih bergantung pada operasi yang benar dari beberapa komponen perangkat keras dan perangkat lunak komputer (prosesor, sistem operasinya, pemeriksa pembuktian Coq, dan kompiler Ocaml yang mengompilasinya), tidak satu pun dari komponen ini yang spesifik untuk pembuktian Teorema Empat Warna. Semuanya sudah siap pakai, memiliki tujuan yang lebih umum, dan dapat (dan memang) diuji secara ekstensif pada banyak pekerjaan lain, mungkin jauh lebih banyak daripada yang dapat dipikirkan oleh seorang matematikawan yang sedang meninjau sebuah manuskrip bukti."

Kita dapat melihat bahwa proses pembuktian, bagaimanapun cara penafsirannya, adalah proses yang panjang dan rumit.

14.4 PERISTIWA MENUJU DEFINISI DAN SOLUSI MASALAH EMPAT WARNA.

Identifikasi dan Definisi Masalah

1840	Matematikawan dan astronom August Ferdinand Möbius menyebutkan masalah ini dalam kuliahnya.
1852	Francis Guthrie menyadari bahwa hanya empat warna unik yang diperlukan untuk mewarnai peta wilayah-wilayah di Inggris; hipotesis ini diajukan kepada

	matematikawan ternama Augustus De Morgan dan kemudian dikenal sebagai "Teorema Empat Warna".
1854	Masalah ini diajukan di majalah sastra London "The Athenaeum" oleh seseorang bernama "F.G." (kemungkinan Francis Guthrie). De Morgan kembali mengajukan masalah yang sama pada tahun 1860.

Upaya Pembuktian yang Berulang Kali Gagal

1879	Alfred Kempe mengajukan satu dugaan pembuktian yang diterima secara luas.
1880	Upaya lain oleh Peter Guthrie Tait.
1890	Percy Heawood menunjukkan bahwa pembuktian Kempe salah.
1891	Julius Peterson menunjukkan bahwa pembuktian Tait salah.

Akibat Akibat Pembuktian

- Pembuktian untuk Teorema Lima Warna
- Penemuan Rantai Kempe
- Generalisasi Teorema Empat Warna ke jenis permukaan lainnya
- Penemuan (oleh Tait) bahwa Teorema Empat Warna setara dengan pernyataan bahwa jenis graf tertentu (disebut snark) harus non-planar, membuka bidang penelitian baru dalam teori graf.

Upaya Pembuktian di Paruh Kedua Abad ke-20

1960-an dan 1970-an Matematikawan Jerman Heinrich Heesch mengembangkan metode yang disebut *discharging* untuk digunakan bersama dengan program komputer. Heesch juga memperluas konsep redusibilitas (bahwa suatu konfigurasi tertentu dapat digantikan dengan sesuatu yang lebih kecil tanpa memengaruhi bilangan kromatiknya) dan mengembangkan metode berbasis komputer untuk mengujinya.

Pada 21 Juni 1976, Kenneth Appel dan Wolfgang Haken dari Universitas Illinois mengumumkan bahwa mereka akhirnya memecahkan pertanyaan yang telahmembingungkan dunia matematika selama lebih dari satu abad. Metode yang digunakan didasarkan pada gagasan bahwa jika terdapat graf planar yang membutuhkan lebih dari 4 warna, sebuah contoh tandingan minimal yang memuat daerah sesedikit mungkin harus ada, dan beberapa kemustahilan kemudian haruslah benar. Komputer kemudian digunakan untuk melakukan pembuktian melalui kontradiksi ini.

Pembuktian Terverifikasi

1989	Appel dan Haken menerbitkan sebuah karya besar setebal 741 halaman yang secara ketat mengklarifikasi pembuktian tersebut dan membelanya terhadap berbagai klaim ketidaksempurnaannya. Bukti tersebut mengembangkan "himpunan tak terelakkan" yang terdiri dari 1476 konfigurasi, dan dilanjutkan (dengan menggunakan komputer) untuk menunjukkan bahwa semuanya dapat

	direduksi. Oleh karena itu, semua graf planar dapat diwarnai empat warna. Lebih dari 1000 jam komputasi diperlukan untuk melakukan perhitungan.
1996	Neil Robertson, Daniel Sanders, Paul Seymour, dan Robbin Thomas, dari Georgia Tech dan Universitas Ohio, mengembangkan bukti yang serupa dengan Appel dan Haken, namun disederhanakan untuk menampilkan himpunan tak terelakkan yang hanya terdiri dari 633 konfigurasi yang terbukti dapat direduksi.
2001	Robertson, Sanders, Seymour, dan Thomas mengemukakan bukti alternatif pada tahun 2001 dengan membuktikan Teorema Snark, yang, sebagaimana disebutkan sebelumnya, setara dengan Teorema Empat Warna.
2005	Benjamin Werner dan Georges Gonthier (Penelitian Microsoft, Cambridge) menggunakan asisten pembuktian terkomputerisasi yang disebut Coq untuk benar-benar memverifikasi rangkaian yang tidak dapat dihindari dan bagian redusibilitas dari pembuktian yang ada.

14.5 CONTOH KODE DARI BUKTI

Untuk memberikan gambaran singkat kepada pembaca tentang kompleksitas dan kerumitan program komputer yang ditulis oleh Robertson, Sanders, Seymour, dan Thomas, kami telah menyertakan prototipe fungsi untuk segmen "Reducibility" dari bukti [11]:

```
/* function prototypes */
#ifdef PROTOTYPE_MAX

void testmatch(long, char *, long[], char *, long);
void augment(long, long[], long, long **, long[16][16][4], char *, char
*, long *, long, long, long, char *, long *, long);
void checkreality(long, long **, char *, char *, long *, long, long,
long, char *, long *, long);
long stillreal(long, long[], long, char *, long);
long updatelive(char *, long, long *);
void strip(tp_confmat, tp_edgeno);
long ininterval(long[], long[]);
void findangles(tp_confmat, tp_angle, tp_angle, tp_angle, long[]);
long findlive(char *, long, tp_angle, long[], long, long);
void checkcontract(char *, long, tp_angle, tp_angle, long[], long[]);
void printstatus(long, long, long, long);
void record(long[], long[], long, long[][5], char *, long *, long);
long inlive(long[], long[], long, char *, long);
long ReadConf(tp_confmat, FILE *, long *);
void ReadErr(int, char[]);
#else
void testmatch();
void augment();
void checkreality();
long stillreal();
long updatelive();
void strip();
long ininterval();
void findangles();
```

```
long findlive();
void checkcontract();
void printstatus();
void record();
long inlive();
long ReadConf();
void ReadErr();
#endif
```

BAB 15
KRIPTOGRAFI

Bab yang membahas solusi komputer untuk permasalahan Kecerdasan Buatan tidak akan lengkap tanpa membahas kriptografi. Kriptografi adalah penyandian dan penguraian pesan dalam kode rahasia atau cipher; versi modernnya mengacu pada pengodean dan penguraian kode informasi terkomputerisasi. Praktik ini telah ada dalam berbagai bentuk selama ribuan tahun, terutama digunakan untuk komunikasi kerajaan atau militer. Namun, belakangan ini, kriptografi telah menjadi sesuatu yang dimanfaatkan oleh sebagian besar warga dunia, meskipun di balik layar, dalam aktivitas berbasis komputer sehari-hari seperti mengirim surel dan transaksi daring.

15.1 PENDAHULUAN

Contoh informasi terenkripsi paling awal yang diketahui ditemukan di Mesir, berasal dari sekitar tahun 2000 SM. Hieroglif ini menghiasi makam raja dan penguasa, menceritakan kisah rakyat mereka dengan cara terkode agar terkesan lebih agung dan penting. Contoh lain kriptologi kuno ditemukan dalam tulisan-tulisan Ibrani Yeremia, yang berasal dari sekitar tahun 700 SM, dan juga digunakan di seluruh karya Kabbalah awal dalam Alkitab. Metode ini dikenal sebagai "Atbash", dan bekerja sebagai berikut: huruf pertama alfabet ditukar dengan huruf terakhir, huruf kedua dengan huruf kedua dari belakang, dan seterusnya, hingga huruf terakhir yang sesuai dengan huruf pertama. Meskipun berasal dari bahasa Ibrani, ini merupakan teknik pertukaran yang sederhana dan efektif yang dapat diterapkan pada hampir semua bahasa lain.

Penggunaan kriptografi pertama yang diketahui untuk transmisi rahasia informasi penting, seperti pada masa perang, terjadi pada awal abad ke-5 SM di tangan bangsa Sparta. Mereka mengembangkan perangkat yang disebut "skytale", sebuah batang kayu yang dililitkan erat dengan sepotong bahan tipis dalam bentuk spiral. Pesan tersebut kemudian ditulis pada bahan tersebut memanjang ke bawah pada tongkat, sehingga hanya dapat diuraikan ketika dibungkus ulang pada batang dengan lebar yang sama; jika tidak, pesan tersebut akan tampak seperti omong kosong. Metode enkripsi Yunani lainnya dikenal sebagai kotak Polypius, atau "papan catur", di mana huruf-huruf ditempatkan dalam kotak dan diwakili oleh nomor baris dan kolomnya.

Contoh historis pertama kriptologi yang digunakan untuk urusan politik adalah oleh Julius Caesar di Roma pada abad pertama SM. Metodenya menggunakan offset tiga, dan setiap huruf ditukar dengan satu huruf tiga posisi kemudian dalam alfabet, pada akhirnya berputar ke awal. Teknik serupa apa pun, bahkan jika offsetnya bukan tiga, disebut sandi Caesar.

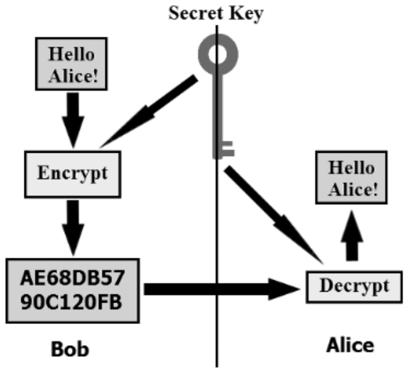
Orang-orang Arab pada pergantian milenium pertama Masehi-lah yang merevolusi bidang ini dengan mengambil pendekatan ilmiah terhadap kriptografi. Ini termasuk pengenalan kriptoanalisis, proses menganalisis materi berkode dan mereproduksi teks asli. Cendekiawan Muslim seperti Ahmad al-Qalqashandi menulis secara ekstensif tentang subjek ini, dan mengembangkan teknik-teknik penting yang memanfaatkan frekuensi huruf yang mungkin dari karakter teks biasa untuk mencocokkannya dengan karakter sandi yang sesuai berdasarkan frekuensinya dalam kode. Cukup mengherankan, metode yang kuat ini relevan bahkan hingga saat ini.

Beberapa abad berikutnya menyaksikan lompatan kualitatif dan kuantitatif yang besar dalam studi kriptografi. Metode seperti *sandi polialfabetik* menambahkan kompleksitas eksponensial pada teks berkode. Kemajuan teknologi menyediakan mesin yang mampu mengubah teks biasa melalui beberapa lapisan enkripsi untuk menghasilkan kode yang sangat terselubung. Di sisi lain, komunikasi melalui telegraf dan radio membuat pesan jauh lebih rentan terhadap intersepsi. Episode politik dan militer yang terkenal dan berdampak luas berkisar pada pesan terenkripsi dan "pemecahannya." Beberapa insiden yang lebih terkenal termasuk plot Babington yang menyebabkan eksekusi Mary Queen of Scots, dan telegraf Zimmerman yang membawa Amerika ke Perang Dunia I, dan pemecahan oleh Sekutu dari mesin enkripsi Enigma Jerman serta sistem sandi "Ungu" Jepang, yang berkontribusi tak terkira terhadap kemenangan Sekutu dalam Perang Dunia II. Contoh-contoh ini berfungsi untuk menggarisbawahi kekuatan dan pentingnya pesan terenkripsi, serta kenyataan yang selalu ada bahwa kerentanan dapat dan akan dieksploitasi. Dengan demikian, sangat penting untuk memberikan perhatian maksimal pada data sensitif. Fokus bab ini akan menjadi kriptografi modern sebagai solusi untuk masalah yang kompleks dan global yang menyangkut hampir setiap individu, yang muncul ketika mereka ingin mengirimkan informasi sensitif dengan aman dan teratur melalui media yang sangat publik seperti World Wide Web.

15.2 ENKRIPSI KUNCI SIMETRIK

Internet telah merevolusi dunia yang kita kenal. Hanya dalam beberapa dekade, Bumi kita yang luas telah menyusut hingga hampir semua hal hanya berjarak beberapa klik mouse. Tumpukan data tersebar di domain publik dengan kecepatan yang luar biasa. Hal ini sangat berbahaya, karena data ini mencakup informasi pribadi yang dapat bersifat pribadi dan sensitif. Bahkan, para ahli keamanan percaya bahwa ancaman keamanan terbesar terhadap pertukaran seperti transaksi e-commerce terjadi pada tingkat komunikasi internet. Alat paling dasar untuk melawan ancaman ini adalah penggunaan enkripsi.

Strategi enkripsi paling sederhana dikenal sebagai "Enkripsi Kunci Simetris". Idenya sederhana dan berjalan sebagai berikut: katakanlah Alice dan Bob ingin mengomunikasikan pesan yang aman. Mereka menyepakati sebuah kunci, yang mereka rahasiakan sebelumnya. Bob menerapkan kunci tersebut ke pesan menggunakan algoritma tertentu yang telah diatur sebelumnya, dan hasilnya adalah teks sandi. Ketika Alice menerima pesan ini, ia menggunakan kunci yang sama untuk memulihkan pesan tersebut. Hal ini dapat dijelaskan secara matematis. Misalkan m mewakili pesan asli, k kunci, c teks sandi, E algoritma enkripsi, dan D algoritma dekripsi. Kita sekarang dapat mengatakan bahwa Bob melakukan $c = E(k, m)$ dan Alice mengambil pesan asli menggunakan $m = D(k, c)$. Sebuah gambar dapat mengilustrasikan proses ini lebih lanjut:



Gambar 15.1 Enkripsi Kunci Simetris

Sistem enkripsi modern menggunakan rangkaian digit biner sebagai kunci. Sebagai contoh, ambil pesan yang berisi huruf "A" yang direpresentasikan oleh nilai ASCII binernya, yaitu 01000001. Salah satu cara untuk mengodekannya adalah dengan mengalikannya dengan kunci biner 8 bit, misalnya 01010101. Jika setiap byte ditransformasikan dengan cara yang sama, penerima teks sandi yang dihasilkan dapat dengan mudah mendekode pesan tersebut dengan kunci 8 bit yang sama.

Kompleksitas dan karenanya ketahanan sistem kunci simetris berbanding lurus dengan panjang kunci biner. Pada contoh sebelumnya, kunci dapat dengan mudah ditemukan menggunakan metode brute force, karena hanya ada 2^8 atau 256 kemungkinan. Akibatnya, sebagian besar sistem menggunakan kunci minimal 56 bit. Beberapa menggunakan hingga 512 digit biner, sehingga menghasilkan 2512 kemungkinan untuk diperiksa. Diperkirakan bahwa semua komputer di dunia yang digabungkan membutuhkan waktu 10 tahun untuk menemukan kunci yang sebenarnya.

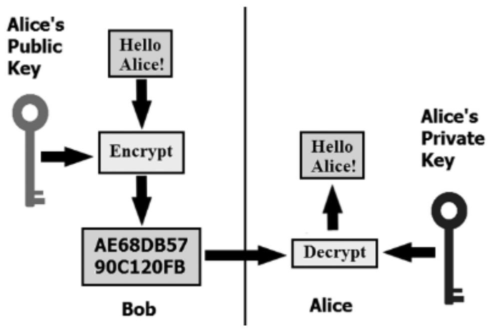
Standar asli untuk skema enkripsi kunci simetris dikembangkan pada awal 1970-an di IBM, bekerja sama erat dengan Badan Keamanan Nasional (NSA). Dikenal sebagai Standar Enkripsi Data, atau DES, standar ini diterbitkan sebagai standar yang disetujui pemerintah federal untuk transfer data sensitif yang aman pada tahun 1977. DES menggunakan kunci 56 digit, yang diterapkan pada data asli dalam blok 64 bit menggunakan algoritma enkripsi yang kompleks. Algoritma tersebut melibatkan 16 putaran pemrosesan, selain permutasi awal dan akhir, semuanya setelah membagi blok menjadi dua bagian 32 bit, untuk memungkinkan pemrosesan bergantian yang dikenal sebagai "Skema Feistal." Pusat perhatian yang dicapai oleh penerbitan standar enkripsi yang disetujui NSA bertanggung jawab untuk melontarkan kriptografi dari bidang yang tidak jelas yang sebagian besar terbatas pada pemerintah, menjadi topik akademis yang menarik dan diteliti dengan baik.

Sejak awal, ada kontroversi mengenai keamanan DES, yang terutama berasal dari ukuran kunci yang relatif kecil. Pada tahun 1990-an, sebuah proyek menunjukkan kemampuan untuk memulihkan kunci DES dalam 22 jam dan 15 menit. Pada tahun 1999, Federal Institute of Standards and Technology (FIST) menegaskan kembali DES sebagai aman, meskipun dengan rekomendasi untuk menggunakan "Triple DES" yang telah ditingkatkan. Namun, pada saat yang sama, FIST sedang melakukan proses seleksi terbuka untuk enkripsi simetris yang baru dan lebih baik, yang disebut AES, atau Advanced Encryption Standard. Entri yang menang dikenal sebagai "Rijndael Cipher," yang diciptakan oleh Joan Daemen dan Vincent Rijmen. Pembahasan komprehensif tentang cara kerja matematika dari algoritma ini mencakup 46 halaman dalam publikasi oleh FIST, dan berada di luar kedalaman dan keluasan buku ini. Cukuplah untuk mengatakan bahwa algoritma ini bekerja dengan sangat baik, dan pada tahun 2001 ditegaskan sebagai standar federal baru untuk semua pesan aman.

Kelemahan mencolok yang melekat dalam enkripsi kunci simetris adalah masalah berbagi kunci untuk memulai. Dalam contoh Bob dan Alice yang kita mulai, secara teoritis mereka dapat berbagi kunci secara langsung, sebelum komunikasi apa pun. Namun, dalam praktiknya, di hampir semua aplikasi enkripsi di dunia nyata, kedua pihak tidak akan dan tidak dapat melakukan hal ini. Lalu, bagaimana kunci tersebut dapat dikomunikasikan dengan aman? Hal ini memang tidak mungkin dilakukan hingga munculnya "Enkripsi Kunci Publik".

15.3 ENKRIPSI KUNCI PUBLIK

Enkripsi kunci publik menyediakan konsep untuk transisi data yang aman tanpa perlu mentransfer kunci bersama secara aman. Mari kita kembali ke contoh Bob dan Alice. Bob masih ingin menyampaikan pesan kepada Alice. Namun, kali ini, Alice memiliki dua kunci. Satu dapat diakses oleh publik dan disebut *kunci publik*, dan yang kedua ia simpan sendiri, disebut *kunci rahasia*. Kedua kunci tersebut terkait dalam arti bahwa jika sebuah pesan dikodekan dengan suatu algoritma enkripsi menggunakan kunci publik, informasi asli dapat direproduksi menggunakan algoritma dekripsi dengan kunci rahasia. Dalam metode ini, mutlak diperlukan bahwa kunci rahasia tidak dapat diturunkan hanya dari pengetahuan tentang kunci publik saja, dan teks sandi tidak dapat didekodekan hanya menggunakan kunci publik. Jika kondisi ini terpenuhi, Bob dapat mengenkripsi pesannya menggunakan kunci publik Alice yang telah dipublikasikan, yakin bahwa hanya Alice yang dapat menguraikan pesan tersebut dengan kunci rahasianya. Secara matematis, kita akan kembali menyatakan m sebagai pesan asli, c sebagai teks sandi, E sebagai algoritma enkripsi, dan D sebagai algoritma dekripsi. Namun, sekarang Alice memiliki kunci publik pk dan kunci privat (rahasia) sk. Kita sekarang mengatakan bahwa Bob melakukan $c = E(pk, m)$ dan Alice mengambil pesan asli menggunakan $m = D(sk, c)$. Untuk memperjelas lebih lanjut dengan gambar:



Gambar 15.2 Enkripsi Kunci Publik

Hal ini membawa kita pada masalah kunci itu sendiri. Kita memerlukan serangkaian fungsi enkripsi agar enkripsi itu sendiri dapat dengan mudah dihitung menggunakan kunci publik. Membalikkan proses dan mendapatkan pesan asli dari teks sandi menggunakan kunci yang sama pastilah mustahil secara praktis. Fungsi semacam ini disebut *fungsi satu arah*. Kita juga memerlukan beberapa informasi tambahan, kunci rahasia kita, yang memungkinkan komputasi invers yang efisien untuk mendapatkan data asli. Ini disebut *informasi pintu jebakan*. Fungsi satu arah yang berisi informasi pintu jebakan dikenal sebagai fungsi *pintu jebakan*.

Pada bulan November 1976, Whitfield Diffie dan Martin Hellman dari Universitas Stanford menerbitkan sebuah makalah inovatif berjudul "Arah Baru dalam Kriptografi" yang memperkenalkan konsep enkripsi kunci publik. Di dalamnya, mereka menyajikan sistem kunci publik yang berfungsi yang memanfaatkan kesulitan dalam mengekstraksi logaritma diskrit. Faktanya, ini adalah masalah matematika yang saat ini belum memiliki solusi. Diffie dan Hellman juga memperkenalkan gagasan *tanda tangan digital* menggunakan kunci publik dan rahasia. Motivasinya jelas --- sama pentingnya untuk mentransfer data dengan aman, penerima mungkin ingin mengonfirmasi bahwa data ini memang dikirim oleh pengirim yang dimaksud. Tanda tangan digital akan menggunakan kunci publik dan rahasia yang sama seperti yang digunakan untuk enkripsi, meskipun secara terbalik. Untuk tanda tangan, Bob akan menggunakan kunci privatnya untuk "mendekripsi" pesan tanda tangan menjadi omong kosong. Sekarang siapa pun dapat menggunakan kunci publik Bob untuk "mengenkripsi" omong kosong tersebut dan menciptakan kembali pesan tanda tangan asli, membuktikan bahwa Bob benar-benar pengirimnya. Ini sangat mirip dengan tanda tangan manual, di mana hanya penanda tangan yang memiliki sarana untuk melakukannya, dan semua orang dapat dengan mudah membuktikan keasliannya. Agar kunci publik dan rahasia mampu melakukan tanda tangan digital sekaligus enkripsi, fungsi yang digunakan harus bekerja dua arah. Artinya, kunci publik dan rahasia harus mampu mengubah data sedemikian rupa sehingga hanya kunci lainnya yang dapat membalikkannya. Metode enkripsi yang diperkenalkan Diffie dan Hellman tidak berfungsi sebagai tanda tangan digital yang efektif. Bahkan, mereka mengusulkan pencarian fungsi tersebut sebagai pertanyaan terbuka.

15.4 ENKRIPSI RSA

Pada tahun 1978, hanya 2 tahun setelah Diffie dan Hellman menantang dunia akademis untuk menemukan sistem kunci publik yang mampu mengenkripsi sekaligus menandatangani secara digital, sebuah solusi diumumkan. Ron Rivest, Adi Shamir, dan Leonard Adleman dari MIT menggemparkan komunitas akademis dan sekitarnya dengan sistem kriptografi kunci publik yang elegan yang menggunakan perkalian dua bilangan prima besar sebagai dasar formulasi kunci. Metode mereka kemudian dikenal sebagai RSA, akronim untuk tiga nama belakang mereka. Arah tunggal fungsi-fungsi ini merupakan akibat dari kesulitan besar yang dihadapi dalam faktorisasi prima bilangan yang sangat besar. Bagi mereka yang tertarik dengan deskripsi matematis yang lebih detail, saya mengutip sebagian pengantar solusi yang jelas dan ringkas dari para pionir (diambil dari "Metode untuk Memperoleh Tanda Tangan Digital dan Kriptosistem Kunci Publik"):

"Untuk mengenkripsi pesan M dengan metode kami, menggunakan kunci enkripsi publik (e, n), lakukan sebagai berikut. (Di sini e dan n adalah pasangan bilangan bulat positif.) Pertama, nyatakan pesan sebagai bilangan bulat antara 0 dan $n - 1$ (.). Kemudian, enkripsi pesan dengan menaikkannya ke pangkat eth modulo n. Artinya, hasilnya (cipherteks C) adalah sisa ketika Me dibagi dengan n. Untuk mendekripsi cipherteks, naikkan ke pangkat d, lagi-lagi modulo n (.). Dengan demikian, kunci enkripsi adalah pasangan bilangan bulat positif (e, n). Demikian pula, kunci dekripsi adalah pasangan bilangan bulat positif (d, n). Setiap pengguna menjadikan kunci enkripsinya publik, dan merahasiakan kunci dekripsi yang terkait. Bagaimana Anda harus memilih enkripsi dan Kunci dekripsi, jika Anda ingin menggunakan metode kami? Pertama, Anda menghitung n sebagai hasil perkalian dua bilangan prima p dan q: $n = p \cdot q$. Bilangan prima ini adalah bilangan prima "acak" yang sangat besar. Meskipun Anda akan membuat n publik, faktor p dan q akan secara efektif disembunyikan dari orang lain karena kesulitan yang sangat besar dalam memfaktorkan n. Ini juga menyembunyikan cara d dapat diturunkan dari e. Kemudian, Anda memilih bilangan bulat d sebagai bilangan bulat acak yang besar yang relatif prima terhadap $(p - 1) \cdot (q - 1)$. Artinya, periksa apakah d memenuhi: fpb($d, (p - 1) \cdot (q - 1)) = 1$ ("fpb" berarti "pembagi persekutuan terbesar"). Bilangan bulat e akhirnya dihitung dari p, q, dan d menjadi "invers perkalian" dari d, modulo $(p - 1) \cdot (q - 1)$. Dengan demikian, kita memiliki $e \cdot d = 1 \,(mod\,(p - 1) \cdot (q - 1))$."

Enkripsi kunci publik menggunakan RSA bekerja dengan sangat baik dan memecahkan masalah yang dihadapi dengan enkripsi kunci simetris. Namun, enkripsi kunci publik jauh kurang efisien dibandingkan kunci simetris yang lebih ramping, dan karena itu, enkripsi itu sendiri tidak optimal untuk transfer data dalam jumlah besar. Solusinya adalah sistem hibrida yang disebut Amplop Digital. Enkripsi kunci publik digunakan untuk mengomunikasikan kunci bersama untuk digunakan dalam dekripsi kunci simetris dari teks sandi, sehingga menggabungkan kedua metode untuk menyediakan kriptosistem yang lengkap, aman, dan praktis.

Hampir empat dekade telah berlalu sejak diperkenalkannya enkripsi kunci publik dan RSA. Sejak awal, para penulis mengakui bahwa mustahil untuk membuktikan bahwa suatu sistem tidak dapat dipecahkan, dan hanya setelah suatu metode mampu menahan semua kemungkinan serangan untuk jangka waktu yang cukup lama, barulah sistem tersebut dapat dianggap aman. RSA bergantung pada kesulitan matematika dalam memfaktorkan bilangan yang sangat besar menjadi kelipatan prima -- semakin besar bilangannya, semakin sulit untuk difaktorkan. Bahasa Indonesia: Dalam makalah mereka yang diterbitkan pada tahun 1977, Rivest, Shamir, dan Adleman memperkirakan bahwa komputer akan membutuhkan waktu $4{,}2 \times 10^{25}$ tahun untuk memfaktorkan angka 500 digit menggunakan algoritma tercepat yang diketahui, dan mereka merekomendasikan penggunaan kunci minimal 200 digit. Namun, seiring berjalannya waktu, kemampuan komputer telah tumbuh secara eksponensial, memungkinkan faktorisasi prima yang lebih cepat. Pada akhir 1990-an, beberapa proyek menunjukkan bahwa adalah mungkin untuk memecahkan kunci 512 bit dalam hitungan beberapa bulan. Pada tahun 2010, angka RSA terbesar yang pernah dipecahkan adalah 768 bit, dan itu dilakukan dalam dua langkah: satu membutuhkan waktu setengah tahun pada 80 prosesor, yang lainnya membutuhkan hampir dua tahun pada ratusan mesin. Meskipun ini masih sangat lambat dan tidak praktis, agak mengecewakan bahwa kita sedang menuju pada lintasan menuju pemecahan angka yang lebih besar lebih cepat dari sebelumnya. Memang, rekomendasi saat ini untuk ukuran kunci oleh Laboratorium RSA adalah 1024 bit untuk penggunaan standar dan 2048 bit jika keamanan dan umur panjang ekstra diinginkan.

Ada juga kelemahan potensial lainnya. Salah satu contoh yang memengaruhi sejumlah kecil aplikasi RSA melibatkan persyaratan bahwa bilangan prima yang digunakan untuk menghitung kunci harus sepenuhnya acak. Para peneliti menemukan bahwa dalam sekitar 0,2% kasus, sistem untuk menghasilkan bilangan prima acak gagal melakukannya secara efektif. Berbagai serangan matematika lainnya secara teoritis mungkin terjadi dalam kondisi tertentu. Namun, kerentanan ini tidak dianggap signifikan. Bahkan masalah pertama mengenai peningkatan cepat daya komputasi yang tersedia untuk menghitung faktorisasi prima, sama sekali tidak mendekati perhatian yang sah untuk masa mendatang yang dapat diperkirakan, selama kuncinya cukup panjang.

Enkripsi kunci publik menggunakan RSA telah melewati ujian waktu dan muncul tanpa cedera. Ini adalah salah satu sistem enkripsi paling populer yang ada, dengan lebih dari satu miliar instalasi di seluruh dunia pada tahun 2000. Ini tertanam dalam ratusan produk perangkat lunak populer, contoh penting adalah Windows. Bahkan sistem enkripsi yang lebih baru dan banyak digunakan di internet, seperti TLS dan SSL, sebenarnya menggunakan RSA sebagai bagian dari algoritmanya. Sistem ini digunakan oleh situs web besar seperti Google, YouTube, dan Facebook. Ternyata Hellman dan Diffie cukup profetik dalam kalimat pembuka makalah mereka tentang metode kunci publik: "Kita saat ini berada di ambang revolusi dalam kriptografi."

15.5 PERMASALAHAN MENGENAI SISTEM KRIPTOGRAFI RSA **

Permasalahan yang terus-menerus muncul dalam perancangan sistem Kriptografi adalah semakin panjang dan rumit kuncinya, semakin aman kunci tersebut tetapi membutuhkan lebih banyak upaya untuk menguraikan dan menanganinya.

Terdapat trade-off yang terus-menerus antara keamanan, kemudahan penggunaan, dan faktor ekonomi.

Keseimbangan antara kesederhanaan dan kemudahan penanganan, versus kompleksitas dan keamanan yang lebih tinggi yang dibarengi dengan biaya penanganan yang lebih besar, juga merupakan permasalahan yang terus berlanjut. Enkripsi yang lebih kompleks membutuhkan lebih banyak komunikasi yang meningkatkan biaya. Oleh karena itu, terdapat motivasi yang besar untuk mengembangkan sistem enkripsi yang efektif dan tidak dapat dipecahkan.

Sistem RSA adalah salah satu sistem yang telah menjadi standar kriptografi selama lebih dari 30 tahun.

Serangan terhadap Enkripsi Kunci Publik menjadi perhatian utama, terutama karena transformasi enkripsinya diketahui publik. Oleh karena itu, penyerang dapat memulai "serangan teks biasa terpilih" dan yang lebih kuat lagi adalah "serangan teks sandi terpilih". Keamanan lingkungan tempat skema enkripsi kunci publik akan digunakan menentukan seberapa mudah satu serangan teks sandi dapat dilakukan.

Skema enkripsi kunci publik yang efektif mengasumsikan bahwa pengirim pesan akan memiliki cara untuk mendapatkan salinan autentik (autentikasi) dari kunci publik penerima yang dituju. Jika tidak demikian, maka skema enkripsi tersebut rentan terhadap serangan peniruan identitas (spoofing).

Sejumlah skema enkripsi kunci publik mengasumsikan bahwa pesan yang akan dienkripsi maksimal berukuran tetap (panjang bit). Jika pesan teks biasa lebih panjang dari panjang bit maksimum yang ditentukan, pesan tersebut akan dipecah menjadi "blok" berukuran tetap. Blok-blok ini kemudian dapat dienkripsi secara independen.

Contoh Implementasi RSA

Contoh 1: Ringkasan Algoritma Pembangkitan Kunci untuk Enkripsi Kunci Publik RSA

Setiap entitas "A" harus melakukan hal berikut:

1. Menghasilkan dua bilangan prima acak (dan berbeda) yang besar, p dan q, masing-masing berukuran kurang lebih sama.
2. Hitung $\mathrm{n} = \mathrm{pq}$ dan $ø = (\mathrm{p} - 1) * (\mathrm{q} - 1)$
3. Pilih bilangan bulat acak e, $1 < \mathrm{e} < ø$, sehingga FPB $(\mathrm{e}, ø) = 1$.
4. Gunakan Algoritma Euclidean untuk menghitung bilangan bulat unik $\mathrm{d}, 1 < \mathrm{d} < ø$, sehingga $\mathrm{ed} = 1$ (mod ø)
5. Kunci publik A adalah (n, e); kunci privat A adalah d.

Contoh Algoritma RSA 2 (sederhana)

- Pilih $\mathrm{p} = 3$ dan $\mathrm{q} = 11$
- Hitung $\mathrm{n} = \mathrm{p} * \mathrm{q} = 3 * 11 = 33$
- Hitung $\varphi(\mathrm{n}) = (\mathrm{p} - 1) * (\mathrm{q} - 1) = 2 * 10 = 20$
- Pilih e sedemikian rupa sehingga $1 < \mathrm{e} < \varphi(\mathrm{n})$ dan e serta n koprima. Misalkan $\mathrm{e} = 7$
- Hitung nilai untuk d sehingga $(\mathrm{d} * \mathrm{e})$ % $\varphi(\mathrm{n}) = 1$. Salah satu solusinya adalah $\mathrm{d} = 3$ $[(3 * 7)$ % $20 = 1]$
- Kunci publik adalah $(\mathrm{e}, \mathrm{n}) \; => \; (7, 33)$
- Kunci privat adalah $(\mathrm{d}, \mathrm{n}) \; => \; (3, 33)$
- Enkripsi $m = 2$ adalah $c = 27$ % $33 = 29$
- Dekripsi c = 29 adalah m = 293 % 33 = 2

https://www.cs.utexas.edu/~mitra/honors/soln.html

Keamanan dan Serangan terhadap RSA

Sejak diperkenalkan pada tahun 1978, Algoritma RSA telah mendapatkan banyak perhatian dan tanggung jawab, sehingga wajar jika terdapat sejumlah upaya untuk menyerang dan membobol keamanannya. Pada bagian ini, kami akan menjelaskan secara singkat beberapa metode yang telah digunakan untuk menyerang Algoritma RSA.

1. **Faktor** – Tantangan bagi penyerang adalah memulihkan (mendekripsi) teks biasa dari teks sandi yang sesuai – dan belum ada algoritma efisien yang diketahui untuk

mencapai hal ini. Oleh karena itu, hal ini dikenal sebagai Masalah RSA (RSAP). Dengan demikian, "ketika menghasilkan kunci RSA, sangat penting bahwa bilangan prima p dan q dipilih sedemikian rupa sehingga pemfaktoran $\mathbf{n = pq}$ tidak layak secara komputasi." Eksponen enkripsi kecil (yaitu $e = 3$) dapat mengalami hal ini karena terdapat algoritma yang dapat dengan mudah mengekspos teks biasa. Solusi yang mungkin untuk masalah ini adalah menambahkan bitstring pseudo-acak sebelum enkripsi. Teknik ini disebut "salting" pesan.

Serangan lain yang dapat terjadi meliputi:

2. **Memilih eksponen dekripsi kecil** (mungkin rapuh, seperti halnya eksponen enkripsi kecil). Solusi untuk masalah ini adalah memilih batasan struktural pada pesan teks biasa.
3. **Serangan Berputar**, atau intinya, upaya untuk memfaktorkan n. Karena pemfaktoran n dianggap sulit diatasi, serangan semacam itu dianggap tidak menimbulkan ancaman serius.
4. **Pesan yang Tidak Disembunyikan** - meskipun beberapa pesan akan selalu tidak disembunyikan, jumlahnya umumnya akan sangat kecil.

Beberapa upaya untuk mengamankan RSA dan mengatasi kemajuan dalam daya komputasi meliputi:

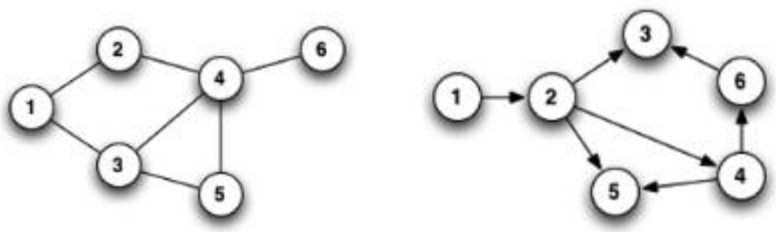
1. **Ukuran modulus minimal 512 bit**. Bahkan, pada tahun 1996, untuk mengatasi teknik faktorisasi prima yang kuat yang dikembangkan oleh Carl Pomerance yang dikenal sebagai Quadratic Sieve, modulus dengan ukuran minimal 768 bit direkomendasikan.
2. **Selisihnya, $\mathbf{p - q}$ tidak boleh terlalu kecil.**
3. **P dan Q harus merupakan "bilangan prima yang kuat", yaitu**
   a) $p - 1$ memiliki faktor prima yang besar, r
   b) $p + 1$ memiliki faktor prima yang besar, dan
   c) $r - 1$ memiliki faktor prima yang besar.

BAB 16
JALAN ACAK PADA GRAF DAN METODE MONTE CARLO

16.1 PENDAHULUAN

Bab ini merupakan versi revisi dan populer dari makalah, yang merupakan karya bersama Paula Whitlock dan Wen-Ju Cheng. Dalam bab ini, kami membahas hubungan antara jalan acak yang muncul dari beberapa area aplikasi yang berbeda. Pada prinsipnya, kami menghubungkan studi jalan acak pada graf dan graf berarah (digraf) dengan jalan acak yang muncul dari permasalahan tertentu yang diselesaikan menggunakan metode Monte Carlo. Kami mencatat bahwa setiap proses ini, meskipun dapat dilakukan secara efisien oleh komputer, terlalu intensif komputasi untuk dilakukan secara manual. Misalnya, jalan acak yang efisien (didefinisikan di bawah) pada graf berukuran wajar dapat memerlukan jutaan atau miliaran langkah. Ini mungkin berarti hanya beberapa detik di komputer, tetapi bertahun-tahun bagi manusia!

Pertama, mari kita definisikan terminologi kita. **Jalan acak** pada graf berlangsung dengan memilih tetangga simpul mana yang akan dikunjungi selanjutnya secara acak dan seragam. Misalnya, jika sebuah simpul memiliki tiga sisi keluar ke tiga tetangga, probabilitas untuk mengikuti setiap sisi adalah 1. Dalam **graf tak berarah** (juga disebut **graf**), sisi yang menghubungkan simpul bersifat "dua arah" (lihat Gambar 16.1 di sebelah kiri), sementara dalam **graf berarah** (**digraf**), sisinya bersifat "satu arah", sehingga sisi pada suatu simpul dapat berupa **sisi masuk** atau **sisi keluar** (lihat Gambar 16.1 di sebelah kanan).



Gambar 16.1 Sebuah Graf Tak Berarah Di Sebelah Kiri Dan Sebuah Graf Berarah Di Sebelah Kanan.

Derajat suatu simpul adalah jumlah sisi yang berinsiden pada simpul itu. Dalam suatu digraf, suatu simpul memiliki **derajat masuk** yang memberikan jumlah sisi masuk, dan **derajat keluar** yang memberikan jumlah sisi keluar. Suatu **simpul seimbang** jika derajat masuknya sama dengan derajat keluarnya dan suatu **digraf seimbang** jika setiap simpulnya seimbang. **Waktu tumbukan** suatu simpul adalah waktu yang diharapkan (rata-rata) yang dibutuhkan untuk jalan acak yang dimulai dari simpul awal yang dibedakan untuk mengunjungi ("menabrak") simpul itu. **Waktu cakupan** adalah panjang rata-rata jalan acak yang mengunjungi semua simpul. Akhirnya, **waktu pencampuran** adalah waktu rata-rata yang dibutuhkan agar jalan tersebut konvergen ke **distribusi stasioner**, yaitu, waktu ketika probabilitas mengunjungi simpul mana pun pada langkah berikutnya stabil. Misalnya, sebuah jalan acak pada digraf di sebelah kanan kiri Gambar 16.1 mungkin berlanjut dengan mengunjungi simpul 1, lalu simpul 2, lalu simpul 4, dan akhirnya simpul 5. Karena simpul 5 tidak memiliki sisi keluar, jalan acak berakhir di sana.

Algoritma Monte Carlo adalah algoritma yang biasanya digunakan untuk mensimulasikan perilaku sistem lain. Ini bukanlah metode yang eksak, melainkan prosedur heuristik. Algoritma ini menggunakan keacakan dan statistik untuk mendapatkan hasil. Secara umum, proses komputasi menggunakan bilangan acak semu untuk menghasilkan suatu hasil. Proses ini mungkin membuat kesalahan satu sisi, yang berarti mungkin berpikir tidak ada solusi untuk suatu masalah (misalnya, teka-teki Sudoku) padahal sebenarnya ada. Di sisi lain, tidak ada kesalahan ketika solusi ditemukan.

Terakhir, **Rantai atau Proses Markov** adalah jalan acak dengan serangkaian **keadaan** dan serangkaian **probabilitas transisi** antar keadaan. Jalan acak pada suatu graf adalah Rantai Markov. Sebaliknya, setiap Rantai Markov diskret bersesuaian dengan jalan acak pada **grafik berbobot**, di mana setiap simpul mewakili keadaan dan setiap sisi mewakili kemungkinan transisi, dengan bobot yang sesuai dengan probabilitas transisi itu. Dalam hal ini probabilitas mengikuti sisi dari suatu simpul mungkin tidak seragam. Misalnya, sebuah simpul mungkin memiliki dua sisi keluar dengan probabilitas $\frac{1}{3}$ dan $\frac{2}{3}$. Jika transisi **reversibel**, kita mendapatkan grafik berbobot, dan jika tidak, kita mendapatkan digraf berbobot. Jika digraf yang dihasilkan **terhubung kuat** (Anda dapat memperoleh dari simpul mana pun ke simpul lain mana pun) rantai Markov disebut **ergodis**. Definisi waktu memukul, waktu menutupi, dan waktu mencampur didefinisikan serupa dengan definisi untuk grafik. Kami telah menunjukkan cara mengubah rantai Markov "bagus" tertentu menjadi digraf yang tidak berbobot. Mari kita perjelas dengan contoh sederhana.

Misalkan Anda memiliki "pohon telepon" di mana setelah menerima panggilan, Anda menelepon semua orang di daftar Anda (misalnya, untuk memberi tahu bahwa hari ini tidak ada sekolah). Alih-alih menelepon semua orang di daftar Anda, Anda melempar dadu untuk menentukan satu orang di daftar Anda yang akan dihubungi. Ini setara dengan berjalan secara acak pada grafik. Waktu tutup telepon mengukur seberapa cepat pesan menyebar ke seluruh kota. Waktu tekan telepon untuk seseorang mengukur berapa lama sebelum mereka menerima panggilan. Waktu pencampuran adalah berapa lama waktu yang dibutuhkan hingga peluang orang berikutnya yang dihubungi adalah orang tertentu tetap sama.

16.2 APLIKASI TEORITIS

Aplikasi teoretis utama untuk studi lintasan acak pada graf adalah pada teori kompleksitas komputasi, yaitu klasifikasi himpunan permasalahan berdasarkan sumber daya komputasi yang digunakan. Terdapat hubungan yang erat antara lintasan acak pada graf dan kelas kompleksitas terbatas ruang, himpunan permasalahan yang dapat diselesaikan menggunakan memori komputer yang terbatas. Permasalahan utama yang terbuka dan telah terpecahkan di bidang kelas kompleksitas terbatas ruang logaritma, himpunan permasalahan yang dapat diselesaikan oleh program yang hanya menggunakan memori proporsional dengan jumlah bit yang dibutuhkan untuk menuliskan panjang masukan sebagai angka, semuanya dapat dinyatakan kembali sebagai permasalahan dalam pencarian graf dan, khususnya, pencarian graf acak $[1, 2, 3, 4, 5, 7]$. Lintasan acak dapat dengan mudah dibatasi pada penggunaan memori ruang logaritma selama pencarian graf, karena hanya perlu mengingat jumlah simpul saat ini, yang membutuhkan logaritma n bit. Sebaliknya, pencarian deterministik naif harus mengingat beberapa riwayat simpul yang dikunjungi, menggunakan setidaknya ruang linear dalam ukuran grafik. Aplikasi praktis dari jalan acak pada grafik telah untuk masalah perkolasi dan jaringan listrik.

Setiap kelas kompleksitas terbatas ruang log yang dipelajari dengan baik dapat diubah menjadi kelas masalah pencarian grafik yang setara. Model yang digunakan oleh para ahli teori kompleksitas adalah mesin Turing (pembaca yang tidak terbiasa dengan mesin Turing dapat melihat mesin ini hanya sebagai jenis program khusus yang berjalan pada komputer yang sangat primitif.) $\mathbf{L}$ (komputasi oleh mesin terbatas ruang log deterministik) sesuai dengan pencarian grafik yang tidak berarah, $\mathbf{RL}$ dan $\mathbf{BPL}$ (komputasi oleh mesin serupa yang menggunakan bit acak) sesuai dengan pencarian acak dari digraf berperilaku baik tertentu, dan $\mathbf{NL}$ (komputasi oleh mesin serupa yang menggunakan nondeterminisme, mirip dengan mesin NP) sesuai dengan pencarian digraf umum. Seseorang mungkin bertanya mengapa tidak menggunakan bit pseudo-acak atau quasi-acak saja untuk program probabilistik ini? Dalam teori kompleksitas, diasumsikan bit "benar-benar" acak. Mengganti bit yang benar-benar acak dengan generator bilangan pseudo acak yang menggunakan benih acak berukuran $\mathrm{O}(\log \mathrm{n})$ akan menunjukkan bahwa $\mathrm{L} = \mathrm{BPL}$. Mungkin mengejutkan, belum ada yang mampu melakukan ini. Hasil terbaik sejauh ini adalah apa yang disebut generator Nissan-Widgerson yang dapat mengelabui mesin Turing (program) ruang $\mathrm{O}(\log \mathrm{n})$ acak dengan menggunakan benih benar-benar acak dari bit $\mathrm{O}\left(\log^{3/2} \mathrm{n}\right)$. Telah ditunjukkan juga bahwa RL dapat disimulasikan secara deterministik menggunakan ruang yang sedikit lebih kecil $\left(\mathrm{O}\left(\log^{4/3} \mathrm{n}\right)\right)$, tetapi masih lebih besar daripada $\left(\mathrm{O}\left(\log^{4/3} \mathrm{n}\right)\right)$ asli.

Banyak yang diketahui tentang jalan acak pada grafik tak berarah dan hasil ini mendahului penemuan oleh Omer Reingold. Reingold menunjukkan bahwa setiap graf tak berarah dapat dicari secara deterministik hanya dengan menggunakan ruang kerja O(log n). Untuk mengapresiasi betapa mengejutkannya hasil ini, perlu diingat bahwa dibutuhkan log n bit hanya untuk menuliskan jumlah satu simpul! Cara lain untuk melihat hasil ini adalah bahwa rantai Markov reversibel diskrit tertentu dapat disimulasikan secara deterministik, tanpa kehilangan efisiensi. Dalam beberapa hal, semua rantai Markov yang digunakan dalam simulasi komputer bersifat diskrit karena umumnya menggunakan bilangan floating point presisi terhingga. Meskipun motivasi teoretis kompleksitas untuk mempelajari lintasan acak pada graf tak berarah telah menguap, hasil yang diperoleh masih berlaku untuk rantai Markov dan memberikan hasil yang menghubungkan lintasan acak pada graf tak berarah dengan rantai Markov reversibel. Mengikuti pembuktian Reingold bahwa pencarian graf tak berarah berada di L, fokus penelitian telah bergeser ke lintasan acak pada graf berarah dalam upaya untuk memperjelas hubungan antara kelas kompleksitas L, RL, NL, dan lainnya.

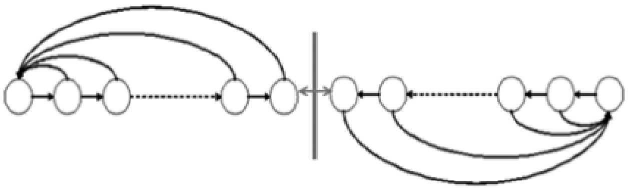
Berikut ini, kami merangkum beberapa hasil yang diketahui pada jalan acak pada grafik yang tidak diarahkan dan diarahkan, dan kemudian kami menyajikan beberapa hasil baru kami. Hasil baru menghubungkan waktu penutup dan pencampuran jalan acak dengan properti struktural grafik yang mengukur seberapa tidak seimbangnya mereka. Kami memperluas definisi kami tentang ketidakseimbangan digraf ke rantai Markov dan menunjukkan bagaimana ketidakseimbangan memengaruhi kinerja. Dalam beberapa kasus diskrit, misalnya dalam simulasi anil seperti yang diterapkan pada teka-teki Sudoku, jalan acak dapat mengambil waktu yang diharapkan secara eksponensial, tergantung pada ukuran masukan, untuk mencapai keadaan tujuan. Salah satu alasannya adalah karena jumlah keadaan yang mungkin sangat besar dibandingkan dengan ukuran masukan. Ketidakseimbangan lain terjadi jika beberapa probabilitas transisi keadaan secara eksponensial lebih kecil (dalam ukuran masukan) daripada yang lain. Ini tentu saja bukan properti yang diinginkan, dan sebagian besar aplikasi mencari rantai Markov dengan probabilitas yang tidak terlalu kecil atau terlalu besar. Dalam praktiknya, jika probabilitas penerimaan suatu pergerakan sangat kecil, random walk tidak akan mencapai kemajuan yang memadai menuju konvergensi.

Sebaliknya, jika probabilitas penerimaan suatu pergerakan terlalu besar, hal ini menyiratkan pengambilan sampel ruang keadaan yang buruk oleh random walk. Selain itu, pembulatan sering digunakan untuk membatasi presisi representasi probabilitas. Kami secara formal mendefinisikan properti ini dan menyebut rantai Markov yang dihasilkan sebagai terikat dengan baik. Tentu saja, bahkan tanpa adanya dua kondisi ketidakseimbangan ini, rantai Markov mungkin masih memiliki waktu pencampuran dan waktu penutupan yang eksponensial.

16.3 RANDOM WALK PADA GRAF

Hubungan antara random walk pada graf tak berarah dan jaringan listrik telah dipelajari dengan baik. Doyle dan Snell menyatakan hasil dasar yang menunjukkan korespondensi antara random walk pada graf tak berarah terhubung dan jaringan listrik. Seseorang dapat mengubah graf menjadi jaringan listrik dengan mengganti setiap sisi dengan resistor 1 ohm. Chandra dan rekan rekan. menunjukkan bahwa jika kita menyuntikkan jumlah arus yang tepat, waktu tumbukan dan waktu penutup dari jalan acak pada grafik G sebanding dengan resistansi dan tegangan dalam jaringan listrik yang sesuai. Dalam pengertian yang sangat tepat, aliran listrik (sebenarnya potensial listrik) melalui jaringan ini mencerminkan "aliran" probabilitas melalui grafik selama jalan acak.

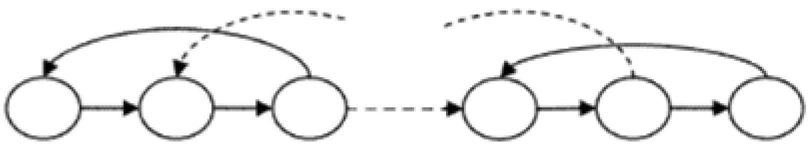
Ada dua ukuran penting lainnya yang telah digunakan untuk membatasi waktu pencampuran jalan acak pada grafik. Seseorang dapat menghitung angka yang disebut "**konstanta Cheeger**" dari grafik yang mengukur setiap hambatan pada aliran probabilitas melalui grafik. Ukuran ini telah diadaptasi ke digraf dan memberikan batas langsung pada waktu pencampuran jalan acak pada digraf. Pada Gambar 16.2 kita melihat grafik dengan hambatan dan karenanya waktu pencampuran yang sangat lambat.



Gambar 16.2 Sebuah Digraf Dengan Leher Botol Yang Diilustrasikan Oleh Segmen Garis Vertikal.

Pengukuran kedua melibatkan matriks yang disebut graf **Laplacian**, yang terkait dengan matriks ketetanggaan graf tersebut. Sebuah angka dapat dihitung dari matriks ini, yang disebut **ekspansi spektral**, dan matriks ini menentukan seberapa cepat sebuah random walk "menyebar". Hal ini juga memberikan batasan langsung pada waktu pencampuran. Sekali lagi, pengukuran ini dapat diadaptasi untuk kasus terarah dan memberikan batasan pada waktu pencampuran untuk digraf. Kedua pengukuran ini, untuk kasus tak terarah, telah diterapkan pada waktu pencampuran untuk rantai Markov reversibel.

Masalahnya adalah pengukuran ini hanya membatasi waktu pencampuran dan bukan waktu penutup. Sebuah graf dapat mengalami pencampuran yang cepat (dalam waktu polinomial) tetapi memiliki waktu penutup (eksponensial) yang buruk. Waktu penutup penting ketika seseorang menggunakan metode acak untuk mencari solusi suatu masalah. Seseorang harus memastikan bahwa Anda mengunjungi semua simpul dengan probabilitas tinggi untuk meminimalkan kemungkinan Anda melewatkan solusi. Misalnya, pertimbangkan pohon biner lengkap dengan kedalaman n. Ini bercampur dengan cepat (dalam waktu linier) karena tidak ada hambatan pada aliran probabilitas, tetapi waktu penutupnya eksponensial. Jika solusi terletak pada satu titik daun (leaf node) tunggal, maka seseorang harus menjalankan jalan acak (random walk) untuk jumlah langkah eksponensial untuk memastikan menemukannya dengan probabilitas tinggi. Diketahui juga bahwa digraf yang seimbang (balanced digraphs) memiliki waktu cakupan yang baik. Karena alasan ini, dan untuk pertimbangan teoretis kompleksitas, kami mengidentifikasi sebuah properti struktural dari digraf yang mengukur seberapa tidak seimbangnya mereka dan memberikan batasan pada waktu cakupan.



Gambar 16.3 Sebuah Graf Yang Tampak Hampir Seimbang Dengan Waktu Tutup Yang Sangat Buruk.

Pada Gambar 16.3, kami mengilustrasikan keluarga digraf dengan waktu tutup yang sangat buruk. Setiap graf terdiri dari rantai simpul dengan sisi maju, dan dimulai dengan simpul kedua, setiap simpul memiliki sisi yang mundur dua simpul. Jalan acak pada graf ini memiliki masalah bahwa untuk setiap langkah maju, seseorang cenderung mundur dua langkah! Ternyata waktu tutup untuk graf ini, seiring bertambahnya jumlah simpul n, bertambah seiring dengan penjumlahan n bilangan Fibonacci pertama, yang merupakan eksponensial dalam n.

Namun, penambahan satu sisi berarah dari simpul kedua ke simpul terakhir membuat waktu tutup menjadi polinomial dalam n (cepat). Perhatikan juga bahwa graf tersebut hanya memiliki dua simpul yang tidak seimbang: simpul kedua dan simpul kedua terakhir dalam rantai. Hal ini menunjukkan bahwa jumlah simpul yang tidak seimbang bukanlah ukuran ketidakseimbangan yang baik untuk waktu tutup.

Dalam kami menunjukkan bahwa ukuran ketidakseimbangan yang benar adalah jumlah simpul asimetris, yang didefinisikan sebagai berikut. Pertama-tama kami menentukan jumlah minimal sisi yang dapat dihilangkan dari digraf sehingga menjadi seimbang. Ini disebut **subgraf seimbang maksimal**. Kami kemudian mempertimbangkan sisi-sisi dari grafik asli yang tidak berada dalam subgraf seimbang ini. Setiap titik di mana setidaknya satu dari sisi-sisi ini insiden padanya sebagai sisi keluar adalah **titik asimetris**. Subgraf seimbang maksimal mungkin tidak unik, tetapi jumlah titik asimetris adalah sama tidak peduli yang mana yang dipilih. Dengan sedikit kerja, seseorang dapat memverifikasi bahwa setengah dari titik dalam grafik contoh kami adalah asimetris, oleh karena itu perilaku buruk.

Pentingnya ukuran ini ditunjukkan dalam, di mana kami menunjukkan bahwa kondisi yang diperlukan untuk digraf yang terhubung kuat dengan derajat keluar terbatas untuk memiliki waktu penutup yang buruk adalah memiliki jumlah titik asimetris yang relatif besar. Khususnya, jika jumlah simpul asimetris kecil ($\mathrm{O}(\log \mathrm{n})$), maka waktu tutupnya bersifat polinomial dan karenanya layak. Di sisi lain, jika jumlah ini tumbuh secara asimtotik lebih cepat dari log n (misalkan $\sqrt{n}$), maka waktu tutupnya mungkin superpolinomial dan karenanya jalan acak pada graf instans besar mungkin tidak layak bahkan untuk komputer tercepat sekalipun. Bahkan, dapat dibuktikan bahwa terdapat digraf dengan √n simpul asimetris yang memiliki waktu tutup eksponensial.

16.4 RANTAI MARKOV DAN METODE MONTE CARLO

Sebuah Rantai Markov dibatasi dengan baik jika jumlah keadaan dibatasi oleh polinomial dalam ukuran masukan dan probabilitas transisinya tidak terlalu kecil maupun terlalu besar. Kami menunjukkan dalam bahwa kami dapat mentransformasikan setiap Rantai Markov keadaan-n ergodis yang dibatasi dengan baik, M, menjadi digraf terhubung kuat GM yang ekuivalen (tidak terbobot), sehingga suatu lintasan acak pada GM berkorespondensi persis dengan Rantai Markov. Kami kemudian mengidentifikasi konstanta Cheeger dan ekspansi spektral M dengan angka-angka ini untuk GM. Hal ini segera memberi kita batasan untuk waktu pencampuran Rantai Markov ini. Sebagai contoh, angka-angka ini dapat digunakan untuk membuktikan bahwa Rantai Markov mengalami pencampuran yang cepat, yaitu, bahwa Rantai Markov tersebut konvergen ke distribusi stasioner dalam waktu polinomial.

Kami juga mendefinisikan gagasan keadaan asimetris untuk Rantai Markov. Kami membuktikan dalam bahwa jika rantai Markov M, suatu keadaan n ergodis yang berbatas baik, memiliki sejumlah kecil keadaan asimetris ($\mathrm{O}(\log \mathrm{n})$), maka rantai tersebut memiliki waktu tutup polinomial. Hal ini menunjukkan bahwa metode Monte Carlo untuk menyelesaikan masalah optimasi, berdasarkan M, akan layak. Karena waktu pencampuran selalu lebih kecil daripada waktu tutup, hal ini juga menunjukkan bahwa M mengalami pencampuran yang cepat.

Waktu pencampuran sangat penting dalam apa yang disebut metode Markov Chain Monte Carlo (MCMC). Misalnya, ketika mensimulasikan sistem fisik, seseorang sering kali ingin menggunakan sampel acak dari suatu distribusi probabilitas, alih-alih hanya menggunakan bilangan acak yang terdistribusi seragam. Dalam MCMC, seseorang mencapai hal ini dengan membangun rantai Markov M yang memiliki distribusi probabilitas yang diinginkan sebagai distribusi stasioner atau "keseimbangan". Seseorang mengambil sampel dari distribusi tersebut dengan melakukan perjalanan acak yang cukup panjang menggunakan M. Sampel tersebut hanyalah keadaan di mana perjalanan tersebut berakhir. Jadi, pertanyaan yang relevan adalah "Berapa lama perjalanan tersebut seharusnya untuk mendapatkan sampel yang baik?" Ini hanyalah pertanyaan tentang waktu pencampuran rantai Markov M. Hasil kami kembali menunjukkan bahwa jika jumlah keadaan asimetris M tidak terlalu besar, maka M akan memiliki waktu penutupan polinomial. Ini menyiratkan bahwa M mengalami pencampuran yang cepat dan karenanya metode MCMM berdasarkan M layak digunakan. Jika tidak, sebaiknya pilih Rantai Markov lain untuk MCMM!

Jadi, sebagai kesimpulan, ketika kami memulai investigasi kami terhadap random walk pada graf yang dimotivasi oleh penerapannya pada teori kompleksitas, kami menemukan bahwa hasil kami meluas ke area metode Monte Carlo yang lebih praktis. Metode-metode ini dapat diterapkan secara luas dalam sains, tetapi membutuhkan komputer yang canggih dan oleh karena itu tidak berbasis manusia.

BAB 17
BERBAGAI MASALAH

Bab ini berisi lima masalah kecil dari berbagai bidang, seperti probabilitas, logika, dan matematika. Kami yakin masalah-masalah ini cukup menarik untuk menarik minat dan bernilai bagi setiap pemecah masalah yang berdedikasi.

17.1 KARTU/KOIN DALAM GELAP

Kartu/Koin dalam Gelap adalah teka-teki matematika dan memiliki dua varian: satu dengan setumpuk kartu dan satu dengan satu set 100 koin.

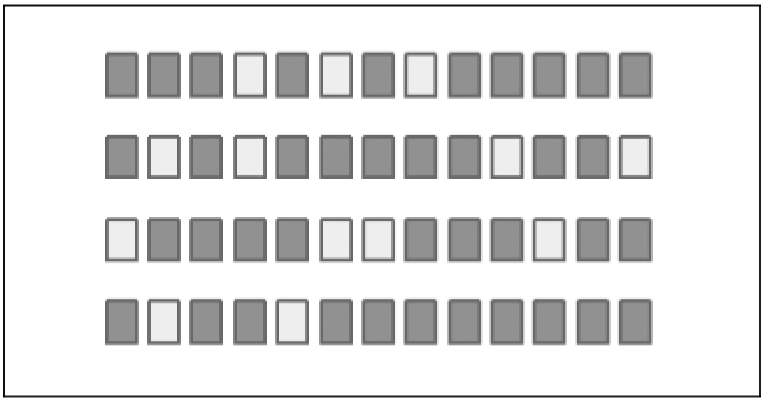
Kartu dalam Gelap

Masalahnya adalah sebagai berikut: Anda berada di ruangan gelap dan diberikan setumpuk 52 kartu. Anda diberi tahu bahwa 13 dari 52 kartu tersebut terbuka dan tersebar di seluruh tumpukan. Anda diminta untuk membagi tumpukan menjadi dua tumpukan yang terdiri dari jumlah kartu terbuka yang sama. Anda tidak dapat membedakan kartu terbuka dan kartu tertutup hanya dengan merabanya, dan karena ruangan gelap, Anda tidak dapat melihat kartu-kartu tersebut. Bagaimana Anda dapat menyelesaikan tugas ini?

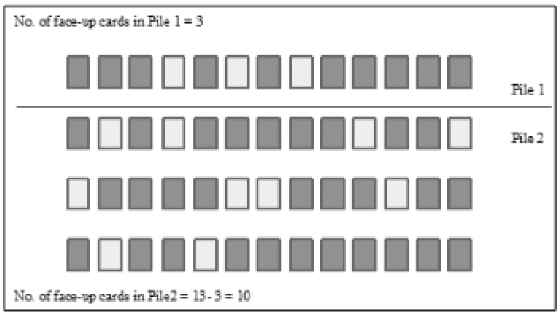
Solusi

Ada 13 kartu terbuka. Jika kita membagi seluruh tumpukan kartu secara acak menjadi dua tumpukan, 13 kartu terbuka akan dibagi di antara kedua tumpukan. Misalkan, jika tumpukan pertama memiliki n kartu terbuka, maka tumpukan kedua akan memiliki $13 - n$ kartu terbuka. Jadi, jika tumpukan pertama memiliki $13 - n$ kartu terbuka, bukan n kartu terbuka, kedua tumpukan akan memiliki jumlah kartu terbuka yang sama.

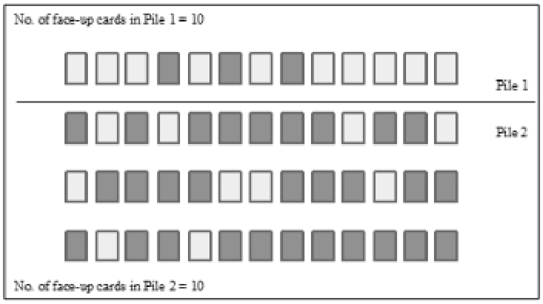
Jadi, jika tumpukan pertama memiliki total 13 kartu, maka akan ada $13 - n$ kartu tertutup, dan jika kita membalik semua kartu di tumpukan pertama, kita akan memiliki $13 - n$ kartu terbuka. Jadi, solusinya adalah membagi kartu menjadi dua tumpukan, satu dengan 13 kartu dan satu dengan 39 kartu, lalu membalik semua kartu di tumpukan pertama. Gambar 17.1 hingga Gambar 17.3 menunjukkan solusi ini.



Gambar 17.1 Tumpukan Kartu Yang Diletakkan Di Atas Meja



Gambar 17.2 Dua Tumpukan Kartu



Gambar 17.3 Setelah Membalik Semua Kartu Di Tumpukan 1

Versi Koin dari Soal

Versi koin dari soal ini memiliki 100 koin di atas meja di ruangan gelap. Dari 100 koin tersebut, 10 koin memiliki kepala menghadap ke atas, dan tujuannya adalah membagi koin-koin tersebut menjadi dua tumpukan dengan jumlah koin yang sama menghadap ke atas.

Solusinya mirip dengan versi kartu dari soal ini. Kita memilih 10 koin apa pun untuk membentuk tumpukan pertama dan membalik semuanya. Sisa 90 koin membentuk tumpukan kedua.

Jika tumpukan pertama awalnya berisi n koin dengan kepala menghadap ke atas, tumpukan kedua akan memiliki $10 - n$ koin menghadap ke atas. Setelah semua koin di tumpukan pertama dibalik, tumpukan pertama juga akan memiliki $10 - n$ koin.

17.2 SEPULUH BAJAK LAUT DAN EMAS MEREKA

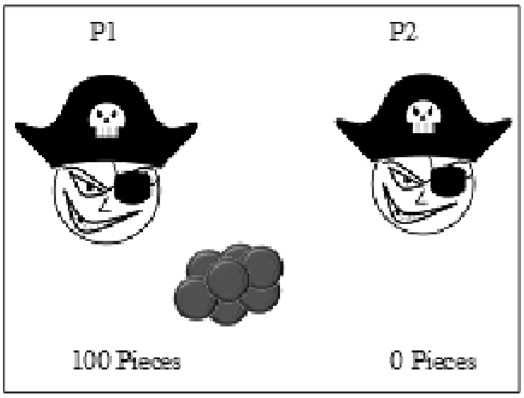
Masalah Sepuluh Bajak Laut dan Emas Mereka telah diajukan selama lebih dari 20 tahun, tetapi menarik perhatian ketika beberapa matematikawan serius (Ian Stewart) dan ilmuwan komputer (Stephen M. Omohumdro) menanganinya dengan beberapa variasi pada akhir 1990-an. Masalahnya adalah sebagai berikut: Sepuluh bajak laut menemukan harta karun terpendam berisi 100 keping emas. Tantangannya adalah membagi emas tersebut dengan cara yang diinginkan berdasarkan beberapa batasan (aturan). Batasan pertama adalah Bajak Laut 1 adalah bajak laut utama; Bajak Laut 2 adalah yang kedua yang bertanggung jawab; Bajak Laut 3 adalah yang ketiga terkuat; dan seterusnya. Para bajak laut memiliki skema untuk membagi uang. Mereka sepakat bahwa bajak laut pertama (P1) akan mengajukan proposal tentang bagaimana uang tersebut akan dibagi. Jika 50% atau lebih bajak laut setuju dengan sistem P1, sistem tersebut akan diberlakukan. Jika tidak, maka P1 akan terbunuh, dan bajak laut terkuat berikutnya menjadi bajak laut utama. Sekarang, lagi-lagi dengan satu bajak laut lebih sedikit, prosesnya berulang. Bajak laut utama yang baru, P2, kini mengusulkan proses baru untuk membagi emas. Proses ini akan melalui pemungutan suara, dengan 50% suara dibutuhkan agar usulan pemimpin tersebut lolos; kurang dari 50% mengakibatkan kematian bajak laut utama.

Semua bajak laut sangat rakus dan licik, sehingga mereka akan menolak proposal jika itu berarti mereka akan mendapatkan lebih banyak emas jika proposal tersebut gagal, dan dengan demikian seorang bajak laut utama terbunuh. Mereka tidak akan pernah memilih proposal yang akan memberi mereka lebih sedikit emas atau tidak sama sekali.

Ketika Anda pertama kali mendengar masalah ini dan mempertimbangkan kemungkinan solusi, Anda mungkin berpikir sebagai berikut: Katakanlah bajak laut utama (P1) ingin bersikap adil dan mengambil 19 keping emas, menawarkan 81 keping emas kepada 9 bajak laut yang tersisa (masing-masing 9 keping). Masalahnya adalah ia membutuhkan 4 bajak laut untuk setuju dengannya. Dan jika 4 bajak laut melihat cara untuk mendapatkan lebih banyak emas, mereka tidak akan memilih untuk pergi bersamanya.

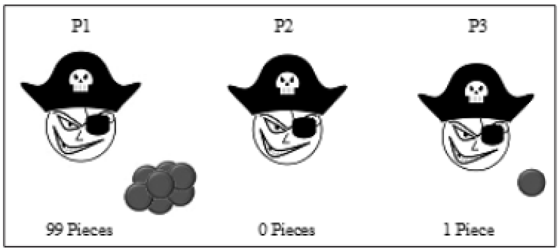
Bajak laut pemimpin mengambil 19 dan sisanya mengambil 11 tampaknya cukup adil, tetapi 4 bajak laut yang memberikan suara akan melihat cara yang lebih baik bagi mereka. Jika P1 tereliminasi (tidak ada suara yang mendukung), maka skenario yang bagus mungkin adalah P2 mendapat 12 dan yang lainnya mendapat $8 \times 11 (= 88)$. Ini tampaknya sangat masuk akal dan adil, tetapi bajak laut tidak dikenal adil. Jadi para bajak laut mungkin mengeliminasi P2, melihat skenario seperti $\mathrm{P3} = 16$, dan yang lainnya mendapat $7 \times 12 (= 84)$. Namun, akan ada bajak laut yang melihat cara bagi mereka untuk mendapatkan lebih banyak emas untuk diri mereka sendiri yang akan menggunakan hak veto mereka untuk kemungkinan mencapai hal ini.

Dengan pendekatan ini, kita dapat melihat logika berikut dapat menghasilkan beberapa hasil nyata: Bayangkan jika hanya ada dua bajak laut, P1 dan P2 (Gambar 17.4). P1 ingin mengambil semua emas (100 keping), meninggalkan P2 tanpa emas dan tidak memiliki hak untuk menentukannya (aturan 50%).

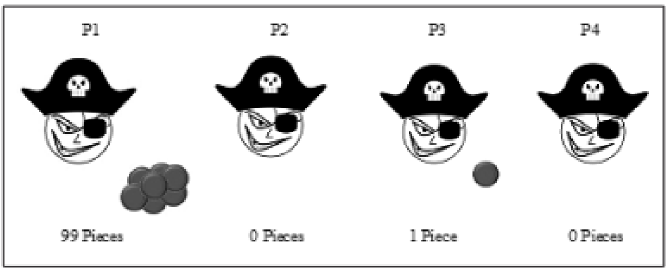


Gambar 17.4 Dua Bajak Laut

Sekarang mari kita perhatikan situasi dengan tiga bajak laut: P1, P2, dan P3 (Gambar 17.5). Bajak laut tidak bodoh. Mereka akan menyadari bahwa jika mereka memilih menentang Anda, P2 akan mendapatkan semua emas dan P3 tidak akan mendapatkan emas setelah Anda pergi. Mengetahui hal ini, P3 akan menerima tawaran Anda berupa 1 keping emas, tanpa emas untuk P2. P2 mungkin tidak akan senang, tetapi proposal Anda akan disetujui. Dengan empat bajak laut, proposal yang sama (P3 mendapatkan 1 koin) akan disetujui, dengan P3 memberi Anda satu suara yang Anda butuhkan untuk disetujui (50%) (Gambar 17.6).



Gambar 17.5 Tiga Bajak Laut



Gambar 17.6 Empat Bajak Laut

Logika yang sama berlaku untuk hingga 10 bajak laut, di mana Anda akan memberikan satu keping emas kepada P3, P5, P7, dan P9 dan menyimpan 99 keping lainnya untuk Anda sendiri. Keempat bajak laut tersebut akan mendukung proposal Anda, karena jika tidak, mereka tidak akan menerima emas jika proposal Anda gagal.

Sekitar waktu artikel Stewart di tahun 1999, Steven Omohundro mengangkat masalah ini ke tingkat yang lebih tinggi dengan menanyakan apa yang terjadi jika tidak hanya ada 10 bajak laut tetapi sebanyak 200 atau 500? Ia menyimpulkan bahwa dengan 200 bajak laut, P200 akan terus menawarkan apa pun kepada bajak laut bernomor ganjil dari P1 hingga P199 dan satu koin emas kepada bajak laut bernomor genap P2 hingga P198.

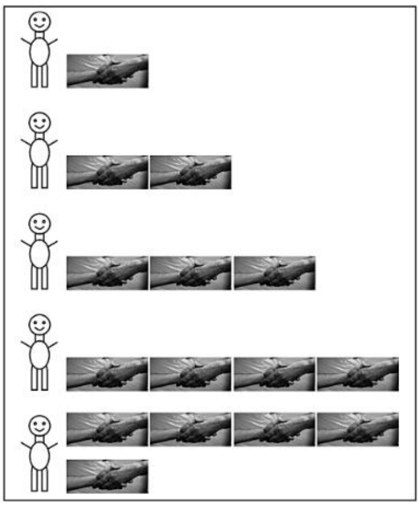
Namun, situasinya menjadi rumit jika kita melampaui 200 bajak laut, karena P201 dan P202 akan mendapati diri mereka kekurangan "kekuatan suap" yang cukup untuk mendapatkan dukungan. Masalah menjadi semakin rumit saat kita mempertimbangkan 203, 204,..., 207 bajak laut. Kesimpulan Omohundro, yang dibagikan Ian Stewart dalam artikel di bawah ini, adalah bahwa ketika kita mendekati 500 bajak laut, sebuah pola yang terus berlanjut tanpa henti muncul.

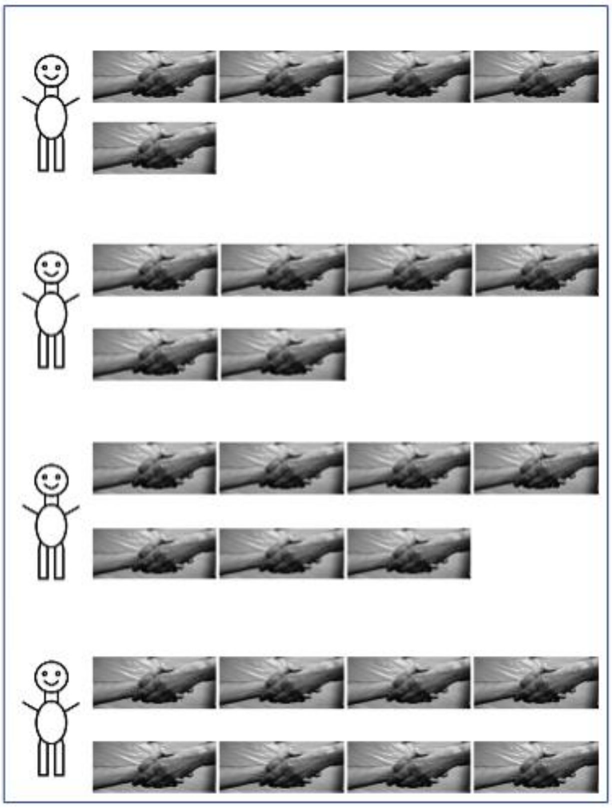
Bajak laut yang dapat mengajukan proposal yang menguntungkan (selalu tidak memberi diri mereka apa pun dan menyuap 100 rekan bajak laut) dipisahkan satu sama lain oleh barisan bajak laut yang semakin panjang yang akan dibuang ke laut apa pun proposal yang mereka buat—dan dengan demikian suara mereka dijamin untuk proposal bajak laut yang lebih ganas. Bajak laut yang terhindar dari nasib ini adalah P201, P202, P204, P208, P216, P232, P264, P328, P456, dan seterusnya—bajak laut yang jumlahnya sama dengan 200 ditambah pangkat 2.

17.3 MASALAH JABATAN TANGAN HALMOS

Paul Halmos adalah seorang matematikawan Hungaria yang terkenal. Ia menulis biografi yang menarik tentang Jon von Neumann yang hebat, yang secara umum dianggap berjabat tangan dengan John von Neumann, yang secara umum dianggap sebagai pengembang arsitektur komputer serial yang masih digunakan hingga saat ini. Ia juga mengembangkan masalah logika berikut, yang disebut **Masalah Jabat Tangan Halmos**.

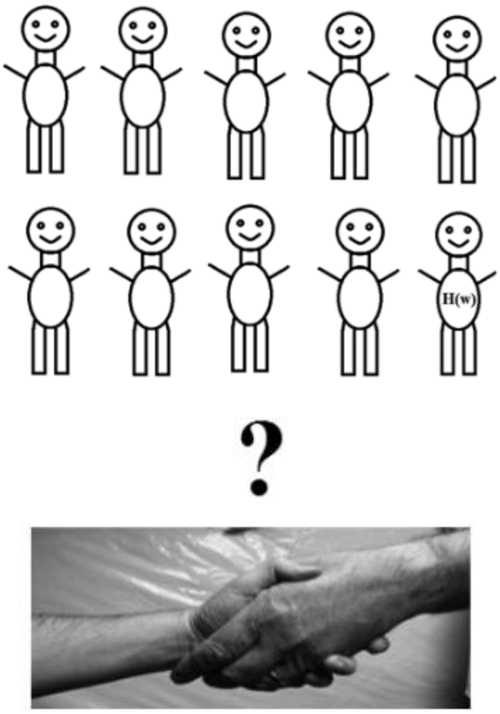
Seperti biasa, para akademisi terkadang menghadiri pesta makan malam. Halmos dan istrinya menghadiri pesta tersebut bersama empat pasangan lainnya. Selama jamuan koktail, beberapa dari mereka yang hadir berjabat tangan tetapi dengan cara yang tidak sistematis, tanpa berusaha untuk menjabat tangan semua tamu. Tentu saja, tidak ada yang menjabat tangannya sendiri, tidak ada yang berjabat tangan dengan pasangannya, dan tidak ada yang berjabat tangan dengan orang yang sama lebih dari sekali. Selama makan malam, Halmos bertanya kepada masing-masing dari sembilan orang yang hadir (termasuk istrinya sendiri) berapa banyak tangan yang telah dijabat oleh orang tersebut. Dalam kondisi yang diberikan, kemungkinan jawaban berkisar antara 0 hingga 8 tangan yang telah dijabat. Halmos memperhatikan bahwa setiap orang memberikan jawaban yang berbeda. Satu orang mengaku tidak berjabat tangan dengan siapa pun, satu orang berjabat tangan tepat dengan satu orang lain, satu orang berjabat tangan tepat dengan dua orang, dan seterusnya hingga satu orang yang mengaku berjabat tangan dengan semua orang yang hadir, kecuali pasangannya—totalnya ada 8 jabat tangan. Singkatnya, dari 10 orang yang hadir, responden menjawab 0 hingga 8 kali berjabat tangan (Gambar 17.7).





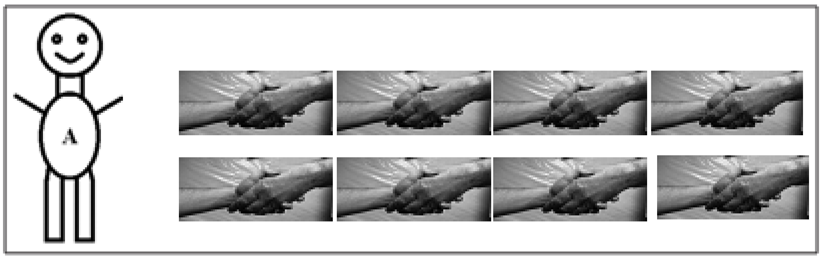
Gambar 17.7 Sepuluh Orang Dan Jabat Tangan

Nah, inilah pertanyaannya: *Berapa banyak tangan yang dijabat istri Halmos?*



Gambar 17.8 Berapa Banyak Orang Yang Berjabat Tangan Dengan Istrinya?

Mari kita lihat apa yang sudah kita ketahui (Gambar 17.8). Kita akan mulai dengan mempertimbangkan orang yang mengaku telah berjabat tangan sebanyak 8 kali dan menyebut orang ini A. Tangan siapa yang dijabat A? Tentu saja, A berjabat tangan dengan semua orang yang hadir, kecuali pasangannya (yang kita sebut A(w)). Jadi, semua orang selain pasangan A {A(w)} berjabat tangan setidaknya dengan satu tangan—yaitu, tangan A (Gambar 17.9).



Gambar 17.9 Orang A—8 Jabat Tangan

Jadi, siapa orang yang berjabat tangan 0 kali? Karena semua orang kecuali pasangan A berjabat tangan setidaknya satu kali, orang yang berjabat tangan 0 kali pastilah pasangan A, A(w) (Gambar 17.9).

Gambar 17.10 Istri Orang A—0 Jabat Tangan

Sekarang perhatikan orang yang berjabat tangan 7 kali, yang kita sebut B. B tidak berjabat tangan dengan dirinya sendiri, pasangannya {B(w)}, atau A(w) (karena A(w) tidak berjabat tangan). Hanya tersisa 7 orang lainnya, sehingga B berjabat tangan dengan semua orang kecuali B dan pasangannya sendiri {B(w)} (Gambar 17.11). Dan masing-masing orang ini juga berjabat tangan dengan A, artinya semua orang ini berjabat tangan setidaknya dengan dua orang: A dan B. Jadi, satu-satunya orang tersisa yang mungkin berjabat tangan hanya dengan 1 tangan adalah pasangan B, B(w) (Gambar 17.12). Sama seperti orang yang berjabat tangan 8 kali menikah dengan orang yang berjabat tangan 0 kali, orang yang berjabat tangan 7 kali menikah dengan orang yang hanya berjabat tangan 1 kali. Kita sekarang dapat melihat pola yang berkembang dalam solusi ini.

Gambar 17.11 Orang B—7 Jabat Tangan

Gambar 17.12 Istri Orang B—1 Jabat Tangan

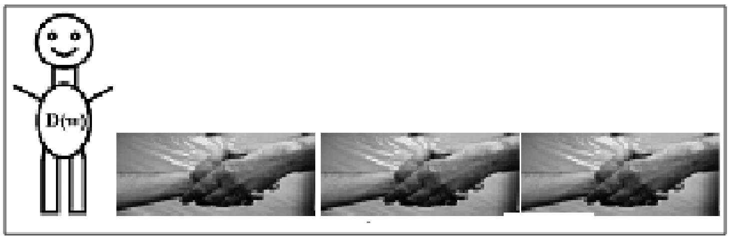
Analisis serupa dapat diterapkan pada orang yang berjabat tangan 6 kali, C, yang tidak mungkin berjabat tangan dengan dirinya sendiri atau pasangannya, A(w) (yang berjabat tangan 0 kali), atau B(w) (yang hanya berjabat tangan dengan A) (Gambar 17.13). Jadi, C berjabat tangan dengan semua orang lainnya, dan karena C juga berjabat tangan dengan A dan B, satu-satunya orang yang tersisa yang hanya dapat berjabat tangan dengan 2 kali adalah pasangan C, {C(w)} (Gambar 17.14).

Gambar 17.13 Orang C—6 Jabat Tangan

Gambar 17.14 Istri Orang C—2 Jabat Tangan

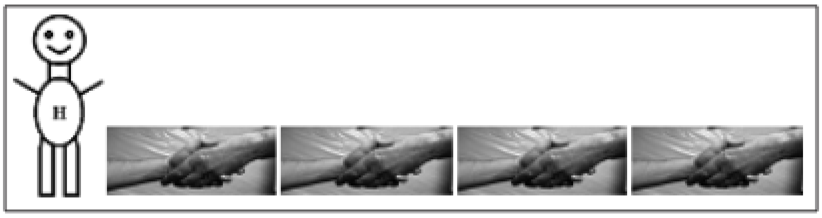
Menerapkan logika yang sama sekali lagi menunjukkan bahwa orang yang berjabat tangan sebanyak 5 kali, D, menikah dengan orang yang berjabat tangan sebanyak 3 kali, {D(w)} (Gambar 17.15 dan 219).

Gambar 17.15 Orang D—5 Shake



Gambar 17.16 Istri Orang D—3 Shake

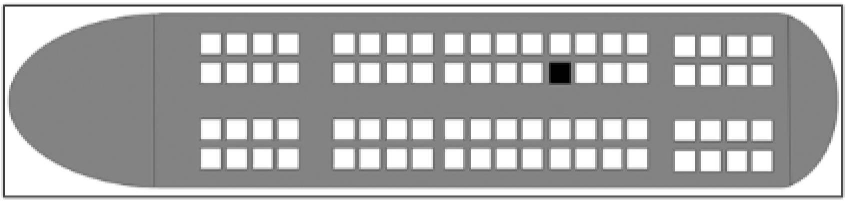
Apa yang tersisa dari ini? Kita tahu bahwa semua orang yang mungkin memberikan jawaban 0, 1, 2, 3, 5, 6, 7, dan 8 semuanya menikah dengan orang lain yang menjawab pertanyaan Halmos. Satu-satunya orang yang tidak menikah dengan salah satu orang yang menjawab pertanyaan tersebut adalah orang yang menjawab 4—dan itu pasti istri Halmos {H(w)}. Jadi, istri Halmos (H(w)) menjabat 4 tangan, dan dapat disimpulkan bahwa Halmos (H) juga melakukannya (Gambar 17.17 dan Gambar 17.18).



Gambar 17.17 Halmos—4 Jabat Tangan

Gambar 17.18 Istri Halmos—4 Jabat Tangan

Permasalahan ini mengilustrasikan bahwa dengan menggunakan *inferensi logis* dan mengingat sedikitnya jumlah permutasi untuk permasalahan ini, dimungkinkan untuk *menghitung secara lengkap* berapa banyak jabat tangan untuk setiap orang yang menghadiri pesta. Selain itu, mengingat sedikitnya jumlah permutasi yang mungkin, dimungkinkan untuk menggunakan sistem notasi sederhana dan/atau grafik untuk merepresentasikan solusi.



Gambar 17.19 Kursi Pesawat

17.4 MASALAH KURSI PESAWAT ACAK

Masalah Kursi Pesawat Acak adalah jenis masalah probabilitas dan logika yang hasilnya mengejutkan dan tak terduga (Gambar 17.19). Berikut adalah salah satu rumusan masalahnya: Seratus orang mengantre di ruang tunggu untuk naik pesawat yang memiliki 100 kursi tersedia. Orang pertama di antrean ini kehilangan boarding pass-nya. Tanpa mengetahui kursi yang ditentukan, ia secara acak memilih kursi. Akibatnya, untuk setiap orang yang memasuki pesawat, hanya ada dua kemungkinan: kursi yang ditentukan tersedia atau tidak. Jika kursinya tidak tersedia, ia memilih kursi kosong secara acak. Berdasarkan informasi ini, ketika orang ke-100 akhirnya memasuki pesawat, berapa probabilitas ia akan menemukan kursi yang ditentukan kosong?

Masalah ini membutuhkan logika yang cukup untuk menentukan solusinya. Meskipun solusinya telah terungkap, masih belum sepenuhnya jelas bagaimana cara mendapatkannya. Namun, mereka yang memahami hukum probabilitas mungkin memiliki pemahaman yang lebih baik tentang cara mendekati masalah ini dan mengapa solusinya tepat.

Solusi

Ada peluang 50% bahwa penumpang ke-100 akan mendapatkan tempat duduk di kursi yang telah ditentukan. Dengan begitu banyak orang yang memasuki pesawat, dan penumpang ke-100 masuk terakhir, jawaban ini tampaknya tidak masuk akal! Namun, dengan pendekatan analitis dan sedikit logika, solusinya dapat dengan mudah dipahami.

Cara umum untuk memecahkan teka-teki ini adalah dengan menentukan secara matematis peluang setiap orang duduk di kursi yang telah ditentukan, tetapi perhitungannya cepat menjadi rumit. Namun, kita dapat dengan mudah mendapatkan jawaban kita dengan menganalisis kemungkinan yang dapat terjadi dan dampak yang disebabkan oleh kemungkinan tersebut.

Untuk solusi ini, kita akan menyebut seorang penumpang Pn, di mana n mewakili posisi orang tersebut dalam antrean. Misalnya, orang yang berada di urutan kelima adalah P5.

Kita mulai dengan membuat dua pengamatan:

1. *Jika salah satu dari 99 orang pertama duduk di kursi yang ditentukan P100*, *P100 TIDAK AKAN* bisa duduk di kursinya. Pengamatan ini seharusnya jelas, karena orang lain yang duduk di kursinya berarti dia tidak bisa duduk di sana.
2. *Jika salah satu dari 99 orang pertama duduk di kursi P1, P100 AKAN bisa duduk di kursinya*. Pengamatan ini tidak begitu jelas. Untuk memperjelas, katakanlah, misalnya, P1 duduk di kursi P2. Ini berarti P2 tidak bisa duduk di kursi yang ditentukan, jadi dia harus duduk di kursi lain. Secara potensial, P2 akan duduk di kursi lain yang dipilihnya. Katakanlah P2 duduk di kursi P3. Maka P3 akan berada dalam kesulitan yang sama dengan P2 dan, tanpa kursi yang ditentukan, akan duduk secara acak di mana pun dia mau. Kemudian katakanlah P3 duduk di kursi Pn. Semua penumpang antara P3 dan $\mathrm{P}n$ akan duduk di kursi yang telah ditentukan, tetapi $\mathrm{P}n$ akan mendapati kursinya terisi dan oleh karena itu akan secara acak duduk di kursi penumpang lain yang kosong. Ini akan berlanjut hingga seorang penumpang—misalnya, P50—duduk di kursi P1. Setelah ini terjadi, penumpang yang tersisa tidak akan mendapati kursi mereka terisi dan

sekarang dapat dengan bebas duduk di kursi yang telah ditentukan. P51 akan duduk di kursi yang telah ditentukan, P52 akan duduk di kursi yang telah ditentukan, dan seterusnya. Oleh karena itu, P100 akan duduk di kursinya sendiri.

Jadi untuk meringkas kedua pengamatan ini, jika kursi P100 terisi, P100 tidak akan mendapatkan kursinya. Jika kursi P1 terisi, P100 akan mendapatkan kursinya. Oleh karena itu, hanya kedua faktor ini yang menentukan apakah P100 akan mendapatkan kursinya atau tidak. Sedangkan untuk kursi-kursi lainnya, kita dapat melihat dari pengamatan kita bahwa keduanya sama sekali tidak berpengaruh dalam menentukan kursi terakhir P100. Ketika penumpang lain memilih kursi yang bukan milik P1 atau P100, mereka hanya menunda penentuan akhir.

Namun, penting untuk dipahami bahwa probabilitas seorang penumpang secara acak memilih kursi tertentu adalah sama persis untuk setiap kursi. Misalnya, ketika P1 memasuki pesawat, ia memiliki 100 kursi untuk dipilih dan oleh karena itu memiliki peluang 1/100 untuk duduk di kursinya sendiri dan peluang 1/100 untuk duduk di kursi P100. Jika ia kebetulan duduk di kursi P2, maka P2 akan memiliki 99 kursi untuk dipilih. Oleh karena itu, akan ada peluang 1/99 untuk duduk di kursi P1 dan peluang 1/99 untuk duduk di kursi P100. Umumnya, jika terdapat n kursi tersisa di pesawat, penumpang berikutnya dalam antrean akan memiliki peluang $1/n$ untuk duduk di kursi P1 dan peluang $1/n$ untuk duduk di kursi P100. Oleh karena itu, peluang setiap penumpang duduk di kursi P1 atau kursi P100 akan selalu sama.

Untuk meringkas solusi ini:

1. P100 akan mendapatkan tempat duduk yang ditentukan jika ada penumpang lain yang duduk di tempat duduk P1.
2. P100 tidak akan mendapatkan tempat duduk yang ditentukan jika ada penumpang lain yang duduk di tempat duduk P100.
3. 1 dan 2 adalah dua pilihan yang menentukan tempat duduk P100, dan keduanya selalu memiliki peluang yang sama untuk terjadi.

Oleh karena itu, P100 akan memiliki peluang 50% untuk duduk di tempat duduk yang ditentukan jika P1 secara acak duduk di mana saja.

17.5 MASALAH ULANG TAHUN

Masalah Ulang Tahun (atau Paradoks Ulang Tahun) adalah masalah probabilitas lain yang penyelesaiannya mengandung hasil yang agak tak terduga. Masalah awalnya adalah sebagai berikut: Misalkan Anda memiliki sekelompok orang acak di sebuah ruangan. Berapa banyak orang yang Anda butuhkan dalam kelompok ini agar ada lebih dari 50% peluang bahwa dua orang ini memiliki tanggal lahir yang sama (bulan dan tanggal)? Tidak seperti Masalah Kursi Pesawat Acak, masalah ini membutuhkan pemahaman yang lebih baik tentang hukum probabilitas.

Solusi

Jawabannya adalah 23. Dalam kelompok yang terdiri dari 23 orang acak, terdapat peluang 50,76% bahwa dua orang atau lebih memiliki tanggal lahir yang sama! Meskipun ini tampak tidak masuk akal pada awalnya, hal ini akan menjadi lebih jelas ketika kita menyajikan beberapa teori probabilitas.

Pertanyaannya menanyakan tentang probabilitas terkait dua orang acak dalam suatu kelompok. Kesalahpahaman umum adalah bahwa masalah ini menanyakan peluang bahwa dua orang tertentu akan memiliki tanggal lahir yang sama. Perhitungan langsung untuk menyelesaikan masalah ini relatif rumit. Namun, untuk menyederhanakan masalah, pertama-tama kita akan menghitung probabilitas tidak ada dua orang yang memiliki tanggal lahir yang sama [1].

Misalkan tanggal 29 Februari tidak ada dan hanya ada 365 hari dalam setahun, yang berarti total ada 365 kemungkinan tanggal lahir. Katakanlah kita hanya memiliki 1 orang, yang akan kita sebut Orang A. Karena Orang A hanya satu orang, tanggal lahirnya bisa salah satu dari 365 hari tersebut, dan tidak ada kemungkinan tanggal lahirnya akan sama dengan orang lain. Oleh karena itu, ada probabilitas $\mathbf{365/365 = 1 = 100}\%$ bahwa Orang A tidak akan memiliki tanggal lahir yang sama dengan orang lain.

Sekarang mari kita tambahkan orang kedua ke dalam kelompok kita, yaitu Orang B. Karena Orang A sudah ada dalam kelompok, maka kemungkinan tanggal lahir Orang B berkurang satu sehingga B tidak memiliki tanggal lahir yang sama dengan tanggal lahir orang lain dalam kelompok tersebut, yang pada dasarnya adalah tanggal lahir Orang A. Oleh karena itu, probabilitas Orang B tidak memiliki tanggal lahir yang sama dengan A adalah 364/365. Sekarang kita perlu mencari probabilitas bahwa kelompok tersebut tidak memiliki tanggal lahir yang sama. Penting untuk dicatat bahwa seseorang yang memiliki tanggal lahir tertentu tidak bergantung pada orang lain yang memiliki tanggal lahir yang sama. Oleh karena itu:

$$\mathbf{P(Tidak\ Ada) = (365/365) \times (364/365) = 0,9973 = 99,73}\%$$

Ini berarti ada probabilitas 99,73% bahwa tidak ada tanggal lahir yang sama dalam kelompok tersebut. Jika kita terus menambahkan Orang C, orang tersebut akan memiliki peluang 363/365 bahwa ia memiliki tanggal lahir yang sama dengan dua orang lainnya dan oleh karena itu:

$$\mathbf{P(Tidak\ Ada) = (365/365) \times (364/365) \times (363/365) = 0,9918 = 99,18}\%$$

Dari ketiga kasus ini, kita dapat melihat sebuah pola yang muncul. Dapat disimpulkan bahwa dalam kelompok dengan n orang, probabilitas tidak ada orang yang memiliki tanggal lahir yang sama adalah:

$$\mathbf{P(Tidak\ Ada) = [(365 \times 364 \times\times (365 - (\mathit{n} - 1)))] / (365\hat{}n)}$$

Dengan menyederhanakannya, kita peroleh:

$$\mathbf{P(TidakAda) = [365!/(365 - \mathit{n})!]/(365^{n})}$$

Setelah kita mengembangkan rumus umum, menghitung probabilitas dua orang atau lebih AKAN memiliki tanggal lahir yang sama dalam kelompok menjadi mudah. Yang perlu dilakukan hanyalah mengurangi 1 dari hasil rumus di atas. Oleh karena itu:

$$\mathbf{P(\sim TidakAda) = P(2\ atau\ lebih) = 1 - [365!/(365 - \mathit{n})!]/(365^{n})}$$

Menggunakan contoh sebelumnya di mana Orang C ditambahkan ke dalam kelompok, probabilitas dua orang atau lebih akan memiliki tanggal lahir yang sama adalah:

$$\mathbf{P(2\ atau\ lebih)\ =\ 1\ -\ 0,9918\ =\ 0,0082\ =\ 0,82\%}$$

Ini berarti terdapat peluang kurang dari 1% bahwa dua orang atau lebih akan memiliki tanggal lahir yang sama. Peluang ini tampak sangat kecil, tetapi dengan menggunakan rumus kita dan mengganti n dengan 23:

$$\mathbf{P(2\ atau\ lebih)\ =\ 1\ -\ [365!\ /\ (365\ -\ 23)!]\ /\ (36523)\ =\ 1\ -\ 0,4927\ =\ 0,5073\ =\ 50,73\%}$$

Dengan demikian, kita dapat menyimpulkan bahwa dalam kelompok yang terdiri dari 23 orang, terdapat peluang sekitar 50% bahwa setidaknya dua orang akan memiliki tanggal lahir yang sama.

17.6 TEKNIK PEMECAHAN MASALAH AI YANG LEBIH BARU

Pembelajaran Mendalam

Dalam wawancara yang menarik dengan Charlie Rose, Sebastian Thrun, [2] $***$ seorang tokoh terkemuka dalam pengembangan Sistem Robotik, mengungkapkan beberapa kemajuan luar biasa dalam sistem AI yang berada dalam jangkauan kita dalam waktu dekat. Ia menunjukkan bahwa fitur dan kekuatan dari apa yang sudah ada di ponsel jutaan orang, dapat dimanfaatkan untuk melakukan penelitian yang dapat dan akan sangat membantu umat manusia. Misalnya, ia membahas tentang bagaimana kamera pada ponsel ini sudah dapat digunakan untuk mengidentifikasi gambar yang menunjukkan adanya kanker kulit. Proses ini bisa sangat efisien hanya karena jumlah gambar yang dapat dihasilkan dengan cepat oleh siapa pun tanpa menggunakan teknologi yang lebih canggih daripada yang dapat ditemukan pada ponsel saat ini. Dan analisis temuan akan lebih efisien daripada pekerjaan yang dilakukan oleh seorang ahli onkologi yang melihat lebih sedikit gambar spesifik. Yang Thrun sarankan adalah mengingat teknologi yang sudah tersedia dan murah saat ini, banyak kemajuan yang dapat diharapkan dalam waktu dekat di sejumlah bidang, termasuk visi komputer, diagnosis medis, robotika, dll.

Selama kurang lebih 25 tahun terakhir, banyak bidang di mana AI telah mencapai kisah sukses telah dicapai dengan menerapkan analitik data besar dengan menggunakan jaringan saraf tiruan dan teknik pembelajaran mesin lainnya, termasuk deteksi wajah, analisis dokumen, mobil tanpa pengemudi, deteksi wajah, dan lain-lain.

Kita telah menyaksikan kemajuan pesat dalam Pemrosesan Bahasa Alami (NLP) ketika korpus bahasa dikombinasikan dengan pendekatan statistik (lihat Lucci dan Kopec, 2016, Bab 13, Pemrosesan Bahasa Alami). Tren keberhasilan menggabungkan pembelajaran mesin atau pendekatan berbasis pengetahuan dengan pendekatan statistik untuk hasil positif ini terus berlanjut. Masalah utama dengan pendekatan pembelajaran mesin adalah tidak adanya "fasilitas penjelasan". Oleh karena itu, pendekatan ini mencapai kinerja yang tinggi, tetapi tidak memiliki cara untuk mentransfer rahasia keberhasilannya kembali kepada manusia yang mungkin dapat menggunakannya dengan cara lain yang lebih efektif.

Selama kurang lebih 10 tahun terakhir, sebuah pendekatan baru untuk menggunakan jaringan saraf tiruan (neural network) untuk kinerja yang sukses, yang disebut "Pembelajaran Mendalam" (Deep Learning), telah dikembangkan seperti yang dijelaskan oleh Prof. Thrun di atas. Gagasan utamanya, secara global, adalah pengenalan pola, dan melalui ekstraksi pola yang efektif menggunakan mesin klasifikasi. Temuan-temuan baru-baru ini dalam ilmu saraf menunjukkan bahwa neokorteks tidak secara eksplisit melakukan pra-proses sinyal sensorik tetapi “memungkinkan sinyal-sinyal tersebut menyebar melalui hierarki modul-modul yang kompleks yang seiring waktu belajar untuk merepresentasikan observasi berdasarkan keteraturan yang ditunjukkannya.” Penemuan ini adalah motivasi utama di balik subbidang pembelajaran mesin mendalam, dengan tujuan berfokus pada "model komputasi untuk representasi informasi yang menunjukkan karakteristik yang mirip dengan neokorteks.

Jaringan Saraf Konvolusional (CNN)

Gagasan umum tentang jaringan saraf (yang telah ada sejak lebih dari 60 tahun yang lalu) adalah bahwa entah bagaimana mereka adalah model tentang bagaimana otak manusia bekerja dan ketika nilai-nilai tertentu melebihi "ambang aktivasi" neuron dirangsang dan diaktifkan. Berkutat pada model biologis ini, cara kerjanya, dan analogi matematis antara jaringan saraf dan otak manusia berada di luar tujuan atau kemampuan kita di sini. Meskipun demikian, dapat diterima bahwa jaringan saraf memiliki lapisan dan nilai yang disebarkan melalui lapisan-lapisan tersebut. Semakin banyak lapisan yang dilewati nilai, semakin "matang" dan "bermakna" mereka, alih-alih bersifat arbitrer.

Mungkin ada analogi yang berguna di sini untuk teknik pencarian (yang dianggap sebagai domain AI yang sangat penting) dan bagaimana teknik tersebut telah dikembangkan selama beberapa dekade. Keyakinan umum selalu bahwa jika pencarian dapat lebih mendalam dan nilai yang disebarkan oleh pencarian tersebut akurat, maka pencarian tersebut lebih bermanfaat – oleh karena itu, ia memprediksi masa depan. Gagasan yang sama, melalui analogi, dapat dikaitkan dengan lapisan-lapisan jaringan saraf. Mungkin semakin banyak lapisan tempat nilai-nilai dalam jaringan saraf dapat disebarkan, semakin "benar" dan bermakna nilai-nilai tersebut!?

Jadi, jaringan saraf dengan lebih banyak lapisan memberikan informasi yang lebih bermakna daripada jaringan dengan lebih sedikit lapisan. Dalam sejarah awal jaringan saraf, hambatan besar diyakini adalah tidak adanya lapisan propagasi balik, tetapi dengan daya komputasi yang lebih besar selama kurang lebih 30 tahun terakhir, masalah ini telah teratasi. Lebih lanjut, semakin banyak lapisan yang dapat diproses untuk mencapai ambang aktivasi, semakin efektif jaringan saraf tersebut. Jaringan saraf konvolusional dengan beberapa lapisan, terutama untuk memproses data gambar dan video dua dimensi, telah terbukti efektif, terutama bila dikombinasikan dengan pekerjaan sebelumnya dalam jaringan saraf tunda waktu (TDNN). Mereka mengurangi persyaratan komputasi pembelajaran dengan berbagi bobot dalam dimensi temporal, terutama bila diterapkan pada tujuan yang dimaksudkan dari pemrosesan ucapan dan deret waktu. Sebagaimana dinyatakan oleh, "mereka adalah pendekatan pembelajaran mendalam pertama yang benar-benar berhasil di mana banyak lapisan hierarki berhasil dilatih secara robust. Teknik yang disebut "penyaringan digital" membantu mendapatkan dan menyorot fitur-fitur penting yang invarian terhadap pergeseran, skala, dan rotasi, dengan memungkinkan akses ke fitur-fitur dasar seperti orientasi tepi atau sudut.

Makalah oleh Lecun dan rekan rekan. tentang klasifikasi digit tulisan tangan MNIST memberikan banyak wawasan tentang proses ini, baik dengan membuat satu peta per lapisan atau beberapa peta.

Keluaran membentuk peta fitur baru yang dilewatkan melalui urutan konvolusi, sub-sampling, dan alur fungsi lainnya. Proses ini dapat diulang beberapa kali.

Lapisan-lapisan berikutnya dapat menggabungkan satu atau lebih lapisan sebelumnya.

CNN menciptakan invariansinya terhadap translasi objek dengan menggabungkan nilai-nilai dari satu atau lebih lapisan sebelumnya. Ini disebut "penggabungan fitur" dan ini dibuat secara manual oleh penyelenggara jaringan. Penggabungan disetel, tidak dilatih atau dipelajari.

Para peneliti telah menerapkan CNN pada berbagai permasalahan pembelajaran mesin, termasuk deteksi wajah analisis dokumen, dan deteksi ucapan. Baru-baru ini, mereka dilatih dengan tujuan koherensi temporal untuk memanfaatkan koherensi bingkai ke bingkai yang ditemukan dalam video.

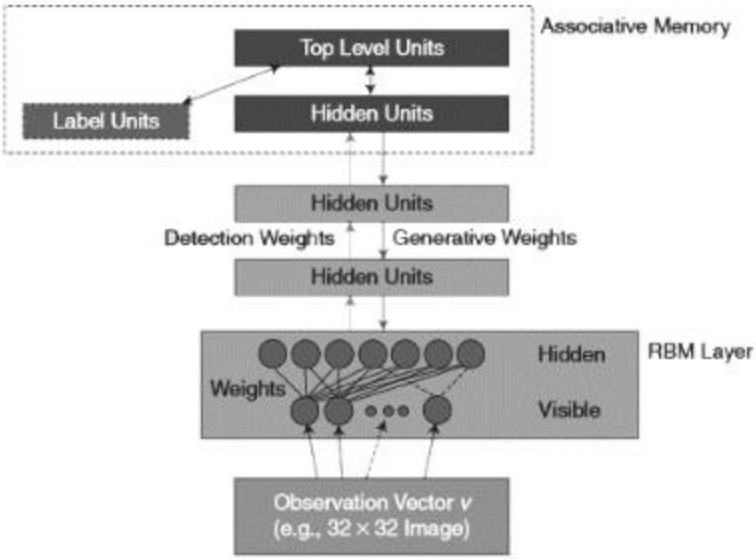
Jaringan Keyakinan Mendalam

Konsep jaringan saraf tiruan pembelajaran mendalam disempurnakan dengan Jaringan Keyakinan Mendalam (DBN), teknik pembelajaran mendalam lain yang juga diperkenalkan sekitar 10 tahun yang lalu. Ini adalah model generatif probabilistik, yang diperkenalkan oleh Geoffrey Hinton, yang berbeda dengan sifat diskriminatif jaringan saraf tiruan.

Keunggulan model generatif adalah menyediakan distribusi probabilitas gabungan atas data dan label yang dapat diamati, sehingga memfasilitasi estimasi nilai dalam kedua keadaan yang mungkin terjadi – yaitu, $\mathrm{P}(\mathrm{Observasi} \mid \mathrm{Label})$ maupun sebaliknya, $\mathrm{P}(\mathrm{Label} \mid \mathrm{Observasi})$. Jaringan saraf tiruan diskriminatif tradisional terbatas pada pendekatan yang terakhir – yaitu $\mathrm{P}(\mathrm{Label} \mid \mathrm{Observasi})$. Jaringan semacam itu mengatasi masalah yang muncul ketika propagasi balik diterapkan pada Jaringan Syaraf Tiruan berlapis dalam:

(1) Kebutuhan akan set data berlabel yang substansial untuk pelatihan
(2) Pembelajaran yang lambat (dengan waktu konvergensi yang lambat, dan
(3) Teknik pemilihan parameter yang tidak memadai yang cenderung menghasilkan optima lokal yang buruk.

DBN terdiri dari beberapa lapisan Mesin Boltzmann Terbatas (RBM), suatu jenis jaringan saraf tiruan tertentu (Lihat Gambar 17.20)



Gambar 17.20 Jaringan Saraf Tiruan Konvolusional

Disebut "terbatas" karena terbatas pada satu lapisan tampak dan satu lapisan tersembunyi dari suatu jaringan saraf tiruan. Unit-unit tersembunyi dilatih untuk menangkap korelasi data orde tinggi yang diamati pada unit-unit tampak. Oleh karena itu, dua lapisan teratas pada dasarnya membentuk memori asosiatif, dengan lapisan-lapisan DBN hanya terhubung oleh bobot generatif top-down. Kemudahan dalam mendapatkan bobot koneksi antar lapisan ini menjadikan RBM pilihan yang menarik dan efisien. Bobot generatif (pra-pelatihan awal) terjadi secara greedy tanpa pengawasan, lapis demi lapis, yang difasilitasi oleh apa yang disebut Hinton sebagai *divergensi kontras*. Selama fase pelatihan ini, vektor disajikan kepada unit tampak dan diteruskan ke unit tersembunyi. Akibatnya, masukan unit tampak ditemukan secara stokastik dalam upaya merekonstruksi masukan asli. Langkah-langkah bolak-balik ini dengan tujuan mengidentifikasi rekonstruksi satu langkah, dikenal sebagai Gibbs Sampling. Perbedaan korelasi antara aktivasi tersembunyi dan masukan yang terlihat membentuk dasar untuk pembaruan bobot. Karena satu langkah diperlukan untuk memperkirakan pembelajaran kemungkinan maksimum, waktu pelatihan berkurang secara signifikan. Setiap lapisan yang ditambahkan ke jaringan meningkatkan log-probabilitas data pelatihan. Oleh karena itu, data tersebut memperoleh daya representasional yang sebenarnya. Keberhasilan setiap proses pembelajaran mendalam bergantung pada ekspansi yang bermakna yang dipadukan dengan pemanfaatan data yang tidak berlabel.

Bobot diikat bersama di dua lapisan teratas sehingga keluaran dari lapisan bawah memberikan petunjuk referensi bagi lapisan atas untuk berasosiasi dengan isi memorinya. Data berlabel digunakan setelah pra-pelatihan bersama dengan back-propagation untuk meningkatkan kinerja diskriminatif DBN. "Pembelajaran" jaringan atas bobot ini menghasilkan kinerja yang lebih baik dibandingkan dengan back-propagation saja.

Arel, Rose, dan Karnowski merangkum:

"Hasil kinerja yang diperoleh ketika menerapkan DBN pada tugas pengenalan karakter tulisan tangan MNIST telah menunjukkan peningkatan yang signifikan dibandingkan jaringan umpan maju. Tak lama setelah DBN diperkenalkan, analisis yang lebih menyeluruh yang disajikan dalam memperkuat penggunaannya dengan tugas-tugas tanpa pengawasan serta masukan bernilai kontinu. Pengujian lebih lanjut dalam menggambarkan ketahanan DBN (serta arsitektur mendalam lainnya) pada masalah dengan variasi yang meningkat."

Sumber referensi tambahan yang sangat baik tentang DBN adalah, karya pendirinya, Geoffrey Hinton.

BAB 18
MENUJU TEORI PEMECAHAN MASALAH

18.1 STUDI JENDELA MANUSIA

Dengan demikian, studi perbandingan solusi Jendela Manusia mengungkapkan beberapa sifat penting yang dimiliki solusi yang baik, dengan kuncinya adalah *ekstensionalitas* dan *pilihan representasi yang baik*. Semakin visual atau grafis suatu representasi, semakin efektif representasi tersebut sebagai alat bantu bagi sebagian besar pemecah masalah. Animasi atau permainan yang dapat dimainkan merupakan beberapa representasi paling ekstensional yang dapat dinikmati orang. Kedua faktor ini juga memengaruhi *pemahaman* solusi—aspek penting lain dari solusi yang mendapat peringkat tertinggi dalam studi kami tentang Jendela Manusia. (Lihat Lampiran C pada CD.)

Sebuah teori tentang pentingnya Jendela Manusia dapat berfungsi sebagai jembatan antara era informasi (dengan data yang melimpah) dan alat bantu pengetahuan, yang dengannya pengetahuan dapat digunakan untuk pembelajaran dan pengambilan keputusan yang efektif. Hasil-hasil ini konsisten dengan paradigma pendidikan sains dan pemecahan masalah yang dikenal sebagai "epistemologi konstruktivis". Premis dasar aliran pembelajaran ini adalah bahwa mengetahui berarti melakukan. Jika Anda dapat melakukan sesuatu, maka Anda mengetahuinya. Guru terbaik adalah mereka yang tahu cara melakukan sesuatu dan paling efektif menjelaskan kepada orang lain apa yang mereka ketahui dengan kata-kata, grafik, dan metode.

18.2 PELAJARAN YANG DIPEROLEH

Pada bagian ini, kami membahas beberapa pelajaran yang dipetik dari mempelajari dan menganalisis tujuh permasalahan dalam buku ini sebagai titik awal yang memungkinkan untuk mengembangkan teori pemecahan masalah. Tabel 18.1 merangkum apa yang telah kita pelajari tentang pemecahan masalah dengan mengkaji 10 permasalahan dalam buku ini.

Tabel 18.1 Strategi Pemecahan Masalah

NO	SOAL	REPRESENTASI PENGETAHUAN	JENIS SOAL	STRATEGI PEMECAHAN MASALAH
1	Soal Misionaris dan Kanibal	Pohon pencarian, diagram transisi keadaan, matriks ruang keadaan masalah	Logika, masalah pemenuhan kendala	Memecahkan subtujuan, backtracking, simetri, pencarian
2	Soal 12 Koin	Pohon pencarian, tabel	Logika, matematika	Reduksi masalah, rekursi
3	Kriptaritma	Tabel pengetahuan, hipergraf kendala	Matematika, masalah pemenuhan kendala	Menghasilkan dan menguji, deduksi, aritmatika, backtracking, pemeriksaan maju, heuristik
4	Teka-Teki Keledai Merah	Bergambar	Teka-teki balok geser	Memecahkan subtujuan, pencarian dua arah
5	Teka-Teki 15 Koin	Grafis, baris demi baris	Teka-teki balok geser	Memecahkan submasalah; berbagai metode, termasuk pola terpisah, heuristik basis data

6	Masalah Tur Ksatria	Grafis; Teselasi;	Grafik—Masalah Siklus Hamilton	Pola dan heuristik
7	Dalang	Karakter atau Grafis	Logika, kombinatorika	Deduksi, Algoritma Lima Tebakan
8	Masalah Monty Hall	Bergambar	Logika, probabilitas	Bergambar, logika, probabilitas
9	Kubus Rubik	Bergambar	3 blok; pola	Subtujuan, langkah, algoritma
10	Dilema Tahanan	Tabel	Logika, masalah pemenuhan kendala	Deduksi, iteratif, berpikir strategis

Masalah Misionaris dan Kanibal

Masalah ini lebih sederhana dibandingkan dengan tiga masalah lainnya dan dapat diselesaikan dengan metode brute force atau trial and error. Namun, pelajaran terpenting adalah bahwa pilihan representasi sangat memengaruhi proses pemecahan masalah. Oleh karena itu, disarankan agar sebelum mencoba memecahkan masalah, siswa meluangkan waktu untuk membiasakan diri dengan masalah tersebut, memahami sifatnya, dan **memilih representasi masalah yang paling menyerupai representasi internal dan pemahaman siswa atau pemecah masalah tentang ruang masalah**. Misalnya, masalah ini selalu membuat kita menciptakan gambaran mental (grafik) dari suatu kejadian dalam dunia nyata; oleh karena itu, representasi bergambar menjadi lebih masuk akal. Semakin realistis representasi bergambar tersebut, semakin sesuai tampaknya dalam proses pemecahan masalah.

Aspek menarik dari masalah ini adalah bahwa keadaan tujuan sangat jelas. Keadaannya sama dengan keadaan awal, tetapi di seberang tepi sungai. Ini menunjukkan semacam simetri dalam hal waktu dan ruang. Dengan demikian, **memecahkan masalah secara mundur dari keadaan tujuan merupakan strategi yang berguna ketika masalah tersebut** *memiliki sifat simetri*.

Biasanya, dalam masalah di mana keadaan awal harus diubah menjadi keadaan tujuan, mengidentifikasi dan **merumuskan subtujuan** dapat membantu. Namun, hal ini juga bisa berbahaya karena orang cenderung hanya bergerak menuju subtujuan tanpa mempertimbangkan kemungkinan bergerak ke keadaan yang dianggap buruk—yaitu, keadaan yang menjauh dari tujuan dan mungkin merupakan satu-satunya cara untuk mencapai tujuan. Terkadang Anda harus mundur selangkah untuk maju. Hal ini terbukti dari diagram transisi keadaan yang menunjukkan bahwa **menjauh dari suatu tujuan untuk sementara** terkadang diperlukan. Namun, kemampuan untuk mengenali kapan perlu untuk menjauh (dan karenanya melangkah "mundur" dari jalur solusi) mungkin khusus bagi pemecah masalah ahli atau mereka yang sangat akrab (terlibat) dengan domain masalah. Akhirnya, salah satu strategi subjek uji pendahuluan, yaitu **menggunakan metode coba-coba untuk menguji berbagai gerakan guna meningkatkan pembelajaran dan perencanaan** langkah selanjutnya, juga merupakan alat yang penting.

Masalah 12 Koin

Masalah ini menggambarkan pentingnya menggunakan reduksi masalah jika memungkinkan. Masalah itu sendiri sangat sulit jika seseorang mencoba menyelesaikannya langsung tanpa melihat versi yang lebih kecil. Sebagaimana dibahas dalam Bab 4, kami memulai dari bentuk paling dasar dari masalah ini untuk memahami kendala masalah dan cara terbaik untuk membagi koin. Kesulitan dalam membagi koin menjadi dua kelompok yang sama terlihat jelas dari versi empat koin dari masalah ini. **Memecahkan versi reduksi dari masalah ini membantu kami memahami nuansa masalah dan membedakan antara penimbangan yang baik dan yang buruk.**

Lebih lanjut, salah satu teknik penting yang digunakan adalah **memaksimalkan jumlah pengetahuan yang diperoleh dari penimbangan sebelumnya**. Karena kendala dalam mencoba meminimalkan jumlah penimbangan, terkadang kami harus menemukan koin ganjil dan berat relatifnya dalam satu kali penimbangan.

Rekursi merupakan teknik penting lainnya yang digunakan karena kami telah memecahkan versi yang lebih kecil dan elementer dari Masalah 12 Koin. Rekursi membantu menyederhanakan masalah secara berkelanjutan hingga diubah ke bentuk paling elementer dan diselesaikan secara langsung.

Lebih lanjut, penggunaan rekursi memfasilitasi konversi solusi ekstensional untuk submasalah (misalnya, 3, 4, 5, 6 koin) menjadi rumus intensional yang ringkas dan dapat digeneralisasi untuk semua jumlah koin.

Kriptaritma

Karena sifat operasi aritmatika, terdapat beberapa petunjuk yang melekat pada kriptoaritma yang dapat dengan mudah ditemukan, seperti kemunculan angka 0, 1, dan 9. Petunjuk ini berfungsi sebagai titik awal untuk memecahkan masalah. Grafik yang merepresentasikan kendala masalah membantu mengungkap hubungan dan ketergantungan antar variabel masalah. Masalah seperti kriptografi membutuhkan penggunaan beragam teknik dalam berbagai tahap proses pemecahan masalah. Hal ini juga berlaku untuk masalah kehidupan nyata di mana satu teknik mungkin tidak menawarkan solusi yang lengkap.

Pada awalnya, petunjuk dan sisa-sisa dapat membantu mencapai kemajuan. Setelah itu, pemecah masalah cenderung mempersempit domain variabel, menghilangkan beberapa nilai yang pada akhirnya akan mengarah ke jalan buntu. Hal ini menghemat waktu dan menghindari perhitungan yang tidak perlu. Ketika domain beberapa variabel cukup rendah untuk memungkinkan pencarian yang menyeluruh, teknik pembangkitan dan pengujian dapat digunakan untuk memfasilitasi beberapa asumsi, melanjutkan, dan menelusuri kembali jika pencarian lebih lanjut tidak memungkinkan. Nilai-nilai tersebut dihilangkan, dan nilai-nilai yang tersisa diuji.

Selain itu, kolom-kolom dalam masalah dipertimbangkan satu per satu sebagai hasil dari strategi **pemecahan submasalah**. Lebih lanjut, struktur seperti **Tabel Pengetahuan** dapat sangat berguna dalam melacak variabel yang telah dipecahkan yang selanjutnya dapat membantu dalam menyimpulkan variabel yang tersisa. Pengalaman menunjukkan bahwa banyak siswa pemecah masalah menghambat kemajuan mereka karena tidak cukup terorganisir tentang apa yang telah mereka pastikan pada tahap tertentu dalam proses pemecahan masalah.

Teka-Teki Keledai Merah

Teka-teki ini melibatkan transformasi keadaan awal menjadi keadaan tujuan dan dengan demikian dapat dirumuskan sebagai serangkaian **subtujuan**. Subtujuan juga berfungsi sebagai tonggak pencapaian yang dapat **dilacak kembali** jika seseorang tersesat dalam teka-teki. Seperti yang telah disebutkan sebelumnya tentang Masalah Misionaris dan Kanibal, ada kalanya kita harus menjauh dari tujuan untuk sementara waktu agar dapat bergerak maju.

Jika keadaan tujuan yang tepat dari solusi diketahui, maka **pencarian mundur atau pencarian dua arah** dapat menjadi teknik yang berguna. Ini mirip dengan permainan akhir catur yang dikenal sebagai Akhiran Ratu dan Pion. Banyak pemain catur cenderung menghindari akhiran seperti itu karena mereka secara keliru berasumsi adanya faktor percabangan yang besar dan kompleksitas yang tinggi, padahal sebenarnya jumlah kemungkinan seringkali terbatas, yang mengarah ke pohon yang panjang dan sempit.

Selain itu, sebagaimana terlihat dari keempat permasalahan, memiliki model masalah yang dapat dimainkan (baik fisik maupun virtual) dapat sangat mengurangi waktu belajar, menjaga minat pemecah masalah tetap hidup, dan membantu menguji berbagai teori dan teknik. Lebih lanjut, pemecah masalah terbebas dari kalkulasi rumit yang merepotkan dan tugas-tugas yang membutuhkan banyak memori, sehingga dapat berkonsentrasi pada pengambilan keputusan.

Dalam konteks pemecahan masalah, strategi-strategi yang dibahas di atas merupakan dasar dari proses pengambilan keputusan manusia, tetapi pada saat yang sama, manusia dibatasi oleh keterbatasan kecepatan komputasi dan kapasitas memori mereka (lihat Lucci dan Kopec). Keterbatasan ini dapat diatasi dengan penggunaan komputer untuk membantu atau melengkapi proses pemecahan masalah manusia. Lebih lanjut, komputer dapat digunakan untuk menguji berbagai keputusan, hipotesis, atau ide dengan cepat guna memvalidasi kesesuaiannya dalam memecahkan masalah yang dimaksud, sehingga menghilangkan langkah-langkah yang tidak valid dan mengurangi waktu pencarian.

Teka-teki 15

Teka-Teki 15 telah ada selama hampir 150 tahun dan mungkin telah menjadi sasaran lebih banyak penerapan teknik pencarian AI daripada masalah lain dalam buku ini. Heuristik seperti "jumlah ubin yang tidak pada tempatnya" atau "Jarak Manhattan", yang dipadukan dengan algoritma seperti pencarian kedalaman-pertama, pencarian lebar-pertama, algoritma A*, dan yang terbaru, metode basis data pola terpisah, telah secara rutin menggunakan Teka-Teki 15 sebagai landasan uji. Kita telah belajar dari solusi MHWC bahwa pendekatan sistematis yang mengembangkan solusi baris demi baris sangat diterima dan mudah dipahami.

Masalah Tur Ksatria

Tur Ksatria juga merupakan masalah yang sangat tua yang telah menjadi subjek penelitian selama ribuan tahun. Fakta yang luar biasa adalah bahwa solusi harus diidentifikasi dari 4×10^{51} kemungkinan urutan. Sejumlah teknik telah digunakan untuk mengidentifikasi solusi, termasuk submasalah, enumerasi lengkap, heuristik, pola, dan lainnya. Ada juga beberapa pengingat tentang aturan 90-10 dalam AI. Artinya, beberapa heuristik menangani sebagian besar ruang masalah (90%), dan kemudian sejumlah heuristik (atau aturan) baru harus dikembangkan untuk menangani 10% dari ruang masalah. Dalam hal ini, heuristik yang selalu menyarankan pemilihan kotak tepi efektif hampir sepanjang waktu—hingga akhirnya tidak efektif lagi. Pemecah masalah harus mempelajari kasus-kasus khusus tersebut.

Mastermind

Mastermind adalah salah satu masalah logika yang paling umum yang telah diujicobakan pada algoritma dan metode komputer selama bertahun-tahun. Ini adalah masalah yang sangat logis dan intuitif yang dapat dipahami dengan mudah oleh orang-orang (bahkan anak-anak). Memahami masalah jauh lebih mudah daripada menentukan metode logis (atau pilihan) yang dapat digunakan untuk membuat tebakan sesedikit mungkin.

Donald Knuth, mungkin ilmuwan komputer terhebat sepanjang masa di Amerika Serikat, memutuskan untuk mengatasi masalah ini pada pertengahan 1970-an. Ia menghasilkan "Five-Guess Algorithm", yang tampaknya diterima sebagai kesimpulan akhir tentang subjek tersebut. Kami percaya bahwa salah satu kontribusi kami di sini bukan hanya analisis berbagai kemungkinan solusi, tetapi juga studi solusi tersebut dari perspektif Human Window, serta studi aktivitas pemecahan masalah subjek manusia untuk Mastermind.

Masalah Monty Hall

Buku karya James Surowiecki, *Wisdom of the Crowds*, membahas bagaimana ketika sekelompok orang (audiens) yang heterogen ditanyai pendapatnya, pendapat tersebut lebih cerdas (terinformasi, benar) daripada orang yang paling cerdas (berdasarkan IQ) di ruangan itu. Dalam masalah ini, sebagaimana dijelaskan oleh Yanofsky dalam buku terbarunya, *The Outer Limits of Reason*, pendapat sekitar 10.000 anggota Mensa berbeda dengan pendapat Marilyn vos Savant—dan mereka salah! Terkadang kemampuan untuk mengabstraksi dan merepresentasikan (menggambar) gambaran dari suatu solusi memiliki makna yang lebih penting daripada hal lainnya.

Kubus Rubik

Ini juga telah menjadi masalah yang umum selama bertahun-tahun. Sepertinya Kubus Rubik, yang mirip dengan ponsel, pesan teks, aplikasi, dan sebagainya, merupakan masalah yang lebih cocok untuk anak muda daripada orang dewasa. Kami yakin bahwa penyajian dan pembahasan solusi untuk "Kubus" di Bab 11 dapat memberikan wawasan bagi mereka yang sebelumnya tercengang olehnya. Tentu saja, tidak ada pembelajaran yang dapat diperoleh tanpa perhatian dan usaha. Ini adalah contoh klasik yang menggambarkan pentingnya kemampuan mengatasi masalah dengan mengidentifikasi dan memecahkan subtujuan.

Dilema Tahanan

Kemampuan untuk menerapkan masalah, ide, dan metode solusi dari Teori Permainan ke dalam permasalahan sosial yang praktis dan nyata sangat menarik bagi semua orang. Dilema Tahanan adalah salah satu permasalahan dari Teori Permainan yang dapat diterapkan dalam berbagai cara untuk permasalahan sosial-politik, situasi ekonomi, dan sebagainya.

Permasalahan Lain-lain

Lima permasalahan lain-lain di Bab 13 menunjukkan bahwa pemecahan masalah dalam banyak kasus merupakan masalah perspektif dan pengalaman. Permasalahan tidak perlu panjang atau rumit untuk menjadi menarik dan instruktif. Karya "Sepuluh Bajak Laut dan Emas Mereka", misalnya, menjadi landasan bagi pemikiran dan representasi solusi untuk Dilema Tahanan.

18.3 RETROSPEKTIF, KESIMPULAN, DAN KARYA MASA DEPAN

Donald Michie dan Danny Kopec pertama kali mengusulkan konsep Jendela Manusia. Konsep ini menyatakan bahwa solusi untuk masalah sulit seperti AI harus memenuhi kriteria tertentu agar dapat berada dalam batasan Jendela Manusia. Sebagaimana awalnya disajikan dalam tesis Shweta, solusi tersebut harus benar, mudah dipahami, dapat dieksekusi, dan memiliki ukuran butir yang moderat. Ia sangat gembira ketika sekitar 30 tahun kemudian, Chris Pileggi melakukan studi solusi Jendela Manusia pada Masalah Misionaris dan Kanibal, kriptoaritmatika, Masalah 12 Koin, Masalah Tur Ksatria, Teka-teki 15 Koin, dan Teka-teki Keledai Merah) untuk proyek tesis seniornya. Karya Pileggi sangat mengesankan dan merupakan penerapan pertama gagasan Michie pada permasalahan nyata dan umum serta solusinya.

Shweta Shetty menyusun dan mempelajari solusi-solusi untuk permasalahan dalam buku ini. Sebagaimana dapat kita lihat dari analisis Human Window atas solusi-solusi ini, manusia merasa nyaman dengan solusi yang grafis, tidak terlalu pendek atau terlalu panjang, mudah dipahami dan dieksekusi, dapat diuji, dan dapat diterapkan pada setiap kondisi permasalahan.

Lebih lanjut, kami kini memahami bahwa meskipun kami yakin telah mengidentifikasi solusi terbaik untuk serangkaian permasalahan AI yang umum, hal ini tidaklah cukup. Penelitian di masa mendatang perlu berfokus pada bagaimana solusi Human Window dapat disajikan sebaik mungkin kepada subjek manusia. Jelas, dengan generasi saat ini dan kemungkinan generasi mendatang, bahkan pendekatan grafis dengan aturan (di atas kertas) tidak akan cukup. Gagasan tentang partisipasi aktif dalam pembelajaran (pola) dan eksekusi (performing) diperlukan. Presentasi solusi kami dalam bentuk video (kemungkinan interaktif) atau animasi tampaknya menjadi langkah terbaik selanjutnya dalam melanjutkan penelitian ini.

Penelitian lebih lanjut diperlukan untuk melanjutkan pekerjaan ini dengan mendalami subjek studi proses pemecahan masalah manusia dengan menggunakan komputer sebagai alat eksperimen. Selain itu, proses berpikir subjek manusia harus direkam dan dianalisis untuk semua masalah tersebut, seperti yang dilakukan Luger untuk masalah kriptoaritma. Sebuah studi perbandingan proses pembelajaran manusia, dengan analisis seberapa baik solusi dipahami (baik sebelum maupun sesudah solusi ditemukan) tampaknya menjadi langkah penting dan penting untuk mendapatkan pemahaman yang lebih baik tentang proses pemecahan masalah dan bagaimana proses tersebut dapat berhasil.

DAFTAR PUSTAKA

Ahmad, S. F., Rahmat, M. K., Mubarik, M. S., Alam, M. M., & Hyder, S. I. (2021). Artificial intelligence and its role in education. Sustainability, 13(22), 12902.

Celik, I. (2023). Exploring the determinants of artificial intelligence (Ai) literacy: Digital divide, computational thinking, cognitive absorption. Telematics and Informatics, 83, 102026.

Farhud, D. D., & Zokaei, S. (2021). Ethical issues of artificial intelligence in medicine and healthcare. Iranian journal of public health, 50(11), i.

Gong, X., Li, Z., & Qiao, A. (2025). Impact of generative AI dialogic feedback on different stages of programming problem solving. Education and Information Technologies, 30(7), 9689-9709.

Habib, S., Vogel, T., Anli, X., & Thorne, E. (2024). How does generative artificial intelligence impact student creativity?. Journal of Creativity, 34(1), 100072.

Ikechukwu, I. B., Uwajumogu, B., & Echerenachukwu, O. (2024). Assessment of Artificial Intelligence Roles in Teaching and Learning (ICT) for Problem Solving in Secondary School in Imo State: Investigating the Prospect and Challenges. GASPRO International Journal of Eminent Scholars, 11(1), 108-123.

Jia, N., Luo, X., Fang, Z., & Liao, C. (2024). When and how artificial intelligence augments employee creativity. Academy of Management Journal, 67(1), 5-32.

Jiang, L., Wu, Z., Xu, X., Zhan, Y., Jin, X., Wang, L., & Qiu, Y. (2021). Opportunities and challenges of artificial intelligence in the medical field: current application, emerging problems, and problem-solving strategies. Journal of International Medical Research, 49(3), 03000605211000157.

Khaleel, M., Ahmed, A. A., & Alsharif, A. (2023). Artificial intelligence in engineering. Brilliance: Research of Artificial Intelligence, 3(1), 32-42.

Kim, S. W., & Lee, Y. (2022). The artificial intelligence literacy scale for middle school students. 한국컴퓨터정보학회논문지, 27(3), 225-238.

Kong, S. C., Zhu, J., & Yang, Y. N. (2025). Developing and validating a scale of empowerment in using artificial intelligence for problem-solving for senior secondary and university students. Computers and Education: Artificial Intelligence, 8, 100359.

Kortemeyer, G. (2023). Could an artificial-intelligence agent pass an introductory physics course?. Physical Review Physics Education Research, 19(1), 010132.

Li, S. (2024, September). Enhancing mathematical problem solving in large language models through tool-integrated reasoning and python code execution. In 2024 5th International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE) (pp. 165-168). IEEE.

Lucci, S., Musa, S. M., & Kopec, D. (2022). Artificial intelligence in the 21st century.

Malinka, K., Peresíni, M., Firc, A., Hujnák, O., & Janus, F. (2023, June). On the educational impact of chatgpt: Is artificial intelligence ready to obtain a university degree?. In Procee

Musleh Al-Sartawi, A. M., Hussainey, K., & Razzaque, A. (2022). The role of artificial intelligence in sustainable finance. Journal of Sustainable Finance & Investment, 1-6.

Neisser, U. (2024). General, academic, and artificial intelligence. In The nature of intelligence (pp. 135-144). Routledge.

Nguyen, T. L., & DANG, T. V. D. (2022). Critical factors affecting the adoption of artificial intelligence: An empirical study in Vietnam. The Journal of Asian Finance, Economics and Business (JAFEB), 9(5), 225-237.

Ngwenya, J. (2024). Learning with generative artificial intelligence in collaborative problem solving: a teaching and learning framework for entrepreneurship education (Master's thesis, J. Ngwenya).

Orrù, G., Piarulli, A., Conversano, C., & Gemignani, A. (2023). Human-like problem-solving abilities in large language models using ChatGPT. Frontiers in artificial intelligence, 6, 1199350.

Plantec, Q., Deval, M. A., Hooge, S., & Weil, B. (2023). Big data as an exploration trigger or problem-solving patch: Design and integration of AI-embedded systems in the automotive industry. Technovation, 124, 102763.

Rane, N. (2023). ChatGPT and similar generative artificial intelligence (AI) for smart industry: role, challenges and opportunities for industry 4.0, industry 5.0 and society 5.0. Challenges and Opportunities for Industry, 4.

Sorantin, E., Grasser, M. G., Hemmelmayr, A., Tschauner, S., Hrzic, F., Weiss, V., ... & Holzinger, A. (2022). The augmented radiologist: artificial intelligence in the practice of radiology. Pediatric radiology, 52(11), 2074-2086.

Su, J., & Zhong, Y. (2022). Artificial Intelligence (AI) in early childhood education: Curriculum design and future directions. Computers and Education: Artificial Intelligence, 3, 100072.

Urban, M., Děchtěrenko, F., Lukavský, J., Hrabalová, V., Svacha, F., Brom, C., & Urban, K. (2024). ChatGPT improves creative problem-solving performance in university students: An experimental study. Computers & Education, 215, 105031.

Waisberg, E., Ong, J., Masalkhi, M., Kamran, S. A., Zaman, N., Sarker, P., ... & Tavakkoli, A. (2023). GPT-4: a new era of artificial intelligence in medicine. Irish Journal of Medical Science (1971-), 192(6), 3197-3200.

Yu, M., Xu, J., Liang, W., Qiu, Y., Bao, S., & Tang, L. (2024). Improved multi-strategy adaptive Grey Wolf Optimization for practical engineering applications and high-dimensional problem solving. Artificial Intelligence Review, 57(10), 277.

Zhai, C., & Wibowo, S. (2023). A systematic review on artificial intelligence dialogue systems for enhancing English as foreign language students' interactional competence in the university. Computers and Education: Artificial Intelligence, 4, 100134.

Zhang, C., & Lu, Y. (2021). Study on artificial intelligence: The state of the art and future prospects. Journal of Industrial Information Integration, 23, 100224.

AI (Artificial Intelligence)

dan

PEMECAHAN MASALAH

Dr. Ir. Agus Wibowo, M.Kom, M.Si, MM

BIO DATA PENULIS

Penulis memiliki berbagai disiplin ilmu yang diperoleh dari Universitas Diponegoro (UNDIP) Semarang. dan dari Universitas Kristen Satya Wacana (UKSW) Salatiga. Disiplin ilmu itu antara lain teknik elektro, komputer, manajemen dan ilmu sosiologi. Penulis memiliki pengalaman kerja pada industri elektronik dan sertifikasi keahlian dalam bidang Jaringan Internet, Telekomunikasi, Artificial Intelligence, Internet Of Things (IoT), Augmented Reality (AR), Technopreneurship, Internet Marketing dan bidang pengolahan dan analisa data (komputer statistik).

Penulis adalah pendiri dari Universitas Sains dan Teknologi Komputer (Universitas STEKOM) dan juga seorang dosen yang memiliki Jabatan Fungsional Akademik Lektor Kepala (Associate Professor) yang telah menghasilkan puluhan Buku Ajar ber ISBN, HAKI dari beberapa karya cipta dan Hak Paten pada produk IPTEK. Sejak tahun 2023 penulis tercatat sebagai Dosen luar biasa di Fakultas Ekonomi & Bisnis (FEB) Universitas Diponegoro Semarang. Penulis juga terlibat dalam berbagai organisasi profesi dan industri yang terkait dengan dunia usaha dan industri, khususnya dalam pengembangan sumber daya manusia yang unggul untuk memenuhi kebutuhan dunia kerja secara nyata.

YAYASAN PRIMA AGUS TEKNIK

PENERBIT :
YAYASAN PRIMA AGUS TEKNIK
Jl. Majapahit No. 605 Semarang
Telp. (024) 6723456. Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

ISBN 978-634-7227-50-8 (PDF)
9 786347 227508