Robotika Raspberry Pi & Arduino

Memakai Python dan OpenCV

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

YAYASAN PRIMA AGUS TEKNIK

Robotika Raspberry Pi dan Arduino: Memakai Python dan OpenCV

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

ISBN : 9 786235 734774

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)

Redaksi :
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : penerbit_ypat@stekom.ac.id

Distributor Tunggal :
Universitas STEKOM
Jl. Majapahit no 605 Semarang
Telp. (024) 6723456
Fax. 024-6710144
Email : info@stekom.ac.id

KATA PENGANTAR

Puji syukur penulis sampaikan karena buku berjudul "*Robotika Raspberry Pi dan Arduino: Memakai Python dan OpenCV*" telah terselesaikan. Robotika itu sebenarnya mudah. Dalam buku ini, penulis memperkenalkan Anda pada bidang robotika. Buku ini akan memaparkan aspek fundamental dan tidak terlalu mendasar dari robotika. Anda akan mengelola perangkat keras, merakit dan menyolder papan sirkuit, menulis kode dalam dua bahasa pemrograman, menginstal dan mengkonfigurasi operating sistem Linux. Semua hal lain yang Anda lakukan dengan robot akan menjadi materi sangat berguna dan tersaji dalam buku ini.

Buku ini, juga di peruntukan para pemula yang baru mengenal elektronik dan IoT; dan yang belum pernah menggunakan Raspberry Pi atau Arduino secara terpisah, apalagi bersama-sama. Buku ini untuk para penghobi yang tertarik untuk belajar lebih banyak tentang bekerja dengan robot. Mungkin Anda telah membangun beberapa sirkuit dengan Arduino atau sistem hiburan rumah khusus dengan Raspberry Pi, dan sekarang Anda penasaran tentang apa yang dibutuhkan untuk membangun robot. Buku ini akan mengajarkan bagaimana kedua perangkat ini bekerja sama untuk memberikan kemampuan yang sangat kuat.

Buku ini untuk mahasiswa yang perlu belajar lebih banyak tentang teknologi; seseorang yang tidak memiliki waktu untuk membaca banyak buku berbeda tentang Arduino, Raspberry Pi, elektronik, atau pemrograman; seseorang yang mencari pengantar yang luas namun padat untuk beberapa dasar-dasar. Buku ini juga untuk mahasiswa yang ingin mengembangkan kemampuan dalam mengembangkan ilmu robotika, dan yang ingin bekerja dengan perangkat keras dan perangkat lunak yang lebih mirip dengan apa yang mungkin mereka lihat di perguruan tinggi atau di dunia profesional. Tidak ada asumsi yang dibuat tentang pengalaman atau latar belakang teknologi. Saat Anda membaca bab-babnya, Anda mungkin menemukan bagian-bagian yang sudah Anda kenal, dan pembaca dapat melompat ke depan. Tetapi jika Anda baru mengenal topik ini, penulis mencoba memberi pembaca pengantar yang cepat namun mudah.

Pembaca mulai dengan mempelajari tentang Raspberry Pi dan cara bekerja dengannya. Anda mengunduh dan menginstal sistem operasi Raspbian, lalu mengonfigurasi Pi untuk proyek kami. Tujuannya adalah untuk mengatur sistem Anda agar dapat dengan mudah mengakses robot Anda dan menulis kode Anda langsung di atasnya. Setelah Anda dapat mengakses Pi Anda dari jarak jauh, di Bab 3, Pembaca akan mempelajari pemrograman dengan Python. Penulis akan menunjukkan cara menulis sederhana program pada Raspberry Pi. Penulis juga membawa para pembaca melampaui dasar-dasar dan membahas beberapa topik menengah, seperti modul dan *Class*. Ini adalah salah satu bab terpanjang karena ada banyak materi untuk dibahas. Dari bab ini, Anda mempelajari cara menghubungkan Raspberry Pi dengan elektronik eksternal, seperti sensor dan LED, melalui header GPIO Pi.

Bab 4 membahas berbagai cara menangani pin pada header, beberapa fungsi melalui header, dan cara menggunakan sensor ultrasonik untuk mendeteksi objek. Di Bab 5, Anda menghubungkan Arduino ke Raspberry Pi. Penulis juga menunjukkan cara bekerja dengan Arduino IDE untuk menulis program. Dibab ini juga membahas komunikasi serial antara dua board dan bagaimana menyampaikan informasi bolak-balik di antara mereka. Penulis melakukan pengujian dan analisi bab ini menggunakan sensor ultrasonik yang sama yang digunakan dalam bab sebelumnya.

Bab 6 menjelaskan tentang menggerakan motor dengan Raspberry Pi Anda. Anda menggunakan papan khusus untuk mengontrol motor. Di sinilah penulis memperkenalkan keterampilan lain yang dibutuhkan dalam robotika: menyolder. Header dan terminal perlu disolder ke papan yang dipilih untuk tujuan ini. Hal yang menyenangkan tentang menyolder header dan blok terminal adalah sulit untuk merusak apa pun, dan Anda akan mendapatkan banyak latihan.

Bab 7 menguraikan tentang menyatukan semuanya. Anda membangun robot, dan penulis membahas beberapa karakteristik fisik robotika. Penulis akan membahas beberapa pertimbangan desain yang perlu Anda ingat saat mendesain sasis Anda sendiri. Meskipun penulis mencantumkan kit sasis khusus untuk proyek ini, Anda tidak perlu menggunakan yang sama. Bahkan, penulis mendorong Anda untuk menjelajahi pilihan lain untuk menemukan satu yang tepat untuk Anda.

Dalam Bab 8, penulis memperkenalkan jenis sensor lainnya—sensor IR, dan menunjukkan kepada Anda cara menggunakan algoritma kontrol yang sangat umum yang disebut pengontrol PID. Penulis menjelaskan tentang berbagai jenis sensor IR dan di mana Anda ingin menggunakannya. (Bab tentang kontrol PID membahas apa itu dan karakteristiknya). Bab 9 adalah tentang visi komputer, di mana Anda melihat kemampuan sebenarnya dari Raspberry Pi. Dalam bab ini, penulis membahas paket *open source* yang disebut OpenCV. Pada akhir Bab 9, robot kecil Anda akan mengejar bola di sekitar meja. Bab 10 akan menjadi bagian penutup buku ini. penulis memberikan beberapa tip yang mungkin diambil, dan memberkani gambaran sekilas tentang alur kerja dan peralatan yang digunakan. Akhir kata semoga buku ini berguna untuk para pembaca.

Semarang, Maret 2022

Penulis

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

DAFTAR ISI

BAB 1
PENGANTAR ROBOTIKA

Kata robotika bisa berarti banyak hal. Bagi sebagian orang, itu adalah sesuatu yang bergerak dengan sendirinya; seni kinetik adalah robotika. Bagi orang lain, robotika berarti sesuatu yang bergerak atau sesuatu yang dapat bergerak sendiri dari satu tempat ke tempat lain. Sebenarnya ada bidang yang disebut robotika seluler; penyedot debu otomatis, seperti Roomba atau Neato, termasuk dalam kategori ini. Bagi saya robotika berada di antara seni kinetik dan robotika seluler.

Robot adalah teknologi yang menerapkan logika untuk melakukan tugas secara otomatis. Ini adalah definisi yang cukup luas, tetapi robotika adalah bidang yang cukup luas. Ini dapat mencakup semuanya, mulai dari mainan anak-anak hingga kemampuan parkir paralel otomatis di beberapa mobil. Kami membangun robot seluler kecil di buku ini.

Banyak prinsip yang Anda paparkan dalam buku ini dapat dengan mudah dialihkan ke bidang lain. Bahkan, kita akan melalui seluruh proses pembuatan robot dari awal hingga akhir. Beberapa saat kemudian dalam bab ini, saya membahas proyek yang akan kita bangun. Pada saat itu, saya akan memberikan daftar bagian-bagian yang digunakan dalam buku ini. Bagian-bagian tersebut antara lain sensor, driver, motor, dan lain sebagainya. Anda dipersilakan untuk menggunakan apa pun yang Anda miliki karena, sebagian besar, semua yang kita lalui dalam buku ini dapat diterapkan pada proyek lain.

1.1 DASAR-DASAR ROBOTIKA

Saya ingin memberi tahu orang-orang yang baru mengenal robotika, atau hanya penasaran dengan robotika, bahwa robot terdiri dari tiga elemen.

- Kemampuan untuk mengumpulkan data
- Kemampuan untuk memproses, atau melakukan sesuatu dengan data yang dikumpulkan
- Kemampuan berinteraksi dengan lingkungan

Dalam bab-bab berikut, kami menerapkan prinsip ini untuk membangun robot bergerak kecil. Kami akan menggunakan pengukur jarak ultrasonik dan sensor inframerah untuk mengumpulkan data tentang lingkungan. Secara khusus, kita akan mengidentifikasi kapan ada objek yang harus dihindari, kapan kita akan keluar dari tepi meja, dan kontras antara meja dan garis yang akan kita ikuti. Setelah kami memiliki data ini, kami akan menerapkan logika untuk menentukan respons yang sesuai. Kami akan menggunakan Python di lingkungan Linux untuk memproses informasi dan mengirim perintah ke motor kami. Saya memilih Python sebagai bahasa pemrograman karena mudah dipelajari, dan Anda tidak perlu memiliki lingkungan pengembangan yang kompleks untuk membangun beberapa aplikasi yang cukup kompleks.

Interaksi kita dengan lingkungan hanya untuk mengontrol kecepatan dan arah motor. Ini akan memungkinkan robot kita bergerak bebas di atas meja atau lantai. Sebenarnya tidak banyak yang bisa dilakukan untuk mengendarai motor. Kami akan melihat dua cara untuk melakukannya: dengan driver motor yang dibuat untuk Raspberry Pi dan dengan pengontrol motor umum. Buku ini dimaksudkan untuk menantang. Saya membahas beberapa materi yang cukup rumit dan saya melakukannya dengan cepat. Tidak ada cara saya dapat memberikan liputan terperinci tentang topik-topik ini, tetapi saya berharap dapat membawa Anda ke robot fungsional pada akhir buku ini. Di setiap bab, saya mencoba memberi Anda lebih banyak sumber daya untuk menindaklanjuti topik yang dibahas. Anda akan berjuang di kali; Saya lakukan dan saya sering masih melakukannya.

Tidak semua orang akan tertarik pada semua mata pelajaran. Harapannya adalah Anda akan memperluas bidang yang paling menarik minat Anda di luar buku ini. Kegigihan terbayar. Di akhir buku, saya menambahkan sedikit tantangan lagi. Di Bab 9, kita mulai memanfaatkan kekuatan Raspberry Pi yang sebenarnya. Kami melihat visi komputer. Secara khusus, kami melihat paket open source yang disebut OpenCV (CV adalah singkatan dari computer vision). Ini adalah kumpulan utilitas umum dan sangat kuat yang membuat bekerja dengan gambar dan aliran video menjadi sangat mudah. Ini juga merupakan build enam jam pada versi terbaru dari Raspberry Pi. Untuk membuat segalanya sedikit lebih mudah dan lebih sedikit memakan waktu, saya telah tersedia untuk mengunduh versi sistem operasi dengan OpenCV yang sudah diinstal. Saya membahas ini lebih lanjut di Bab 2.

1.2 LINUX DAN ROBOTIKA

Linux adalah sistem operasi berbasis Unix. Ini sangat populer di kalangan programmer dan ilmuwan komputer karena sederhana dan lugas. Mereka tampaknya menikmati antarmuka terminal berbasis teks. Namun, bagi banyak orang lain, termasuk saya, Linux bisa sangat menantang. Jadi, mengapa saya memilih lingkungan ini untuk buku pengantar robotika? Jawaban atas pertanyaan itu ada tiga. Pertama, ketika Anda bekerja dengan robotika, Anda akhirnya harus berhadapan dengan Linux. Itu hanya fakta. Anda dapat melakukan banyak hal tanpa pernah mengetik satu perintah sudo, tetapi kemampuan Anda terbatas. Perintah sudo adalah singkatan dari super user do di Linux. Ini memberitahu sistem operasi bahwa Anda akan melakukan fungsi yang dilindungi yang memerlukan lebih dari akses pengguna umum. Anda akan mempelajari lebih lanjut tentang ini ketika kami mulai bekerja dengan Raspberry Pi.

Kedua, Linux itu menantang. Seperti yang saya nyatakan sebelumnya, buku ini akan menantang Anda. Jika Anda pernah bekerja di Linux sebelumnya, maka alasan ini tidak berlaku untuk Anda. Namun, jika Anda baru mengenal Linux, Raspberry Pi, atau bekerja di baris perintah, maka beberapa hal yang kami lakukan akan menantang. Dan itu bagus. Anda mempelajari sesuatu yang baru dan itu harus menjadi tantangan. Ketiga, dan ini yang paling penting, Raspberry Pi menggunakan Linux. Ya, Anda dapat menginstal sistem operasi lain di Pi, tetapi itu dirancang dan dimaksudkan untuk menggunakan Linux. Faktanya, Raspberry Pi memiliki cita rasa Linux sendiri yang disebut Raspbian. Ini adalah sistem operasi yang direkomendasikan, jadi itulah yang akan kami gunakan. Salah satu hal menyenangkan tentang

menggunakan sistem operasi bawaan, selain kemudahan penggunaannya, adalah banyak alat yang sudah terpasang dan siap digunakan.

Karena kami menggunakan Linux, kami akan menggunakan instruksi baris perintah secara ekstensif. Di sinilah sebagian besar pengguna baru memiliki masalah. Kode baris perintah dimasukkan melalui terminal. Raspbian memiliki antarmuka bergaya Windows yang akan kita gunakan, tetapi sebagian besar menggunakan terminal. Jendela terminal tersedia di antarmuka pengguna grafis (GUI), jadi kami akan menggunakannya. Namun, ketika kami mengatur Pi, kami akan mengaturnya untuk boot ke mode terminal secara default. Masuk ke GUI hanyalah perintah startx sederhana. Semua ini dibahas dalam Bab 2.

1.3 SENSOR DAN GPIO

GPIO adalah singkatan dari input/output tujuan umum. Ini mewakili semua berbagai koneksi ke perangkat. Raspberry Pi memiliki banyak pilihan GPIO: HDMI, USB, audio, dan sebagainya. Namun, ketika saya berbicara tentang GPIO dalam buku ini, saya biasanya mengacu pada header GPIO 40-pin. Header ini menyediakan akses langsung ke sebagian besar fungsi papan. Saya membahas ini di Bab 2. Arduino juga memiliki GPIO. Faktanya, orang dapat berargumen bahwa Arduino adalah semua GPIO dan tidak ada yang lain. Ini tidak jauh dari kebenaran mengingat bahwa semua koneksi lain ada untuk memungkinkan Anda berkomunikasi dengan dan memberi daya pada chip AVR di jantung Arduino.

Semua header dan koneksi GPIO ini ada sehingga kami dapat mengakses sensor di luar papan itu sendiri. Sensor adalah perangkat yang mengumpulkan data. Ada banyak jenis sensor yang berbeda, dan semuanya memiliki tujuan. Sensor dapat digunakan untuk mendeteksi tingkat cahaya, jangkauan suatu objek, suhu, kecepatan, dan lain sebagainya. Secara khusus, kami akan menggunakan header GPIO dengan pengintai ultrasonik dan detektor IR.

1.4 GERAK DAN KONTROL

Satu hal yang sebagian besar definisi robot memiliki kesamaan adalah bahwa ia harus dapat bergerak. Tentu, Anda dapat memiliki robot yang tidak benar-benar bergerak, tetapi perangkat jenis ini umumnya berada di bawah moniker IoT, Internet of Things. Ada banyak cara untuk menambahkan gerakan ke proyek Anda. Yang paling umum adalah penggunaan motor. Tetapi Anda juga dapat menggunakan solenoida, udara, atau tekanan air. Saya membahas motor lebih banyak di Bab 6. Meskipun dimungkinkan untuk menggerakkan motor langsung dari Raspberry Pi atau papan Arduino, itu sangat tidak dianjurkan.

Motor cenderung menarik lebih banyak arus daripada yang dapat ditangani oleh prosesor di papan. Sebagai gantinya, Anda disarankan menggunakan pengontrol motor. Seperti motor, pengontrol motor datang dalam berbagai bentuk. Papan kontrol motor yang akan kita gunakan diakses melalui header Raspberry Pi. Saya juga membahas cara mengemudikan motor dengan pengontrol motor ganda L298N.

1.5 RASPBERRY PI DAN ARDUINO

Kami akan menggunakan Raspberry Pi (lihat Gambar 1-1) bersama dengan Arduino (lihat Gambar 1-2) sebagai platform pemrosesan robot kami.

Gambar 1-1. Raspberry Pi 3 B+

Gambar 1-2. Arduino Uno

Raspberry Pi adalah komputer papan tunggal yang seukuran kartu kredit. Meskipun ukurannya kecil, ini adalah perangkat yang sangat mumpuni. Pi menjalankan versi Linux yang disesuaikan untuk bekerja pada prosesor ARM yang menggerakkannya. Ini menempatkan banyak fungsi ke dalam perangkat kecil yang mudah disematkan ke hal-hal seperti robot. Namun, meskipun ini adalah komputer yang hebat, ada beberapa tempat di mana ia tidak unggul.

Satu area adalah antarmuka dengan perangkat eksternal. Ini dapat bekerja dengan sensor dan perangkat eksternal, tetapi Arduino melakukan ini jauh lebih baik. Arduino adalah perangkat pemrosesan kecil lainnya yang tersedia dan mudah digunakan. Tidak seperti Raspberry Pi, bagaimanapun, ia tidak memiliki kapasitas untuk sistem operasi penuh. Daripada menjalankan mikroprosesor seperti ARM, ia menggunakan jenis chip yang berbeda yang disebut mikrokontroler. Perbedaannya adalah bahwa mikrokontroler dirancang khusus untuk berinteraksi dengan sensor, motor, lampu, dan semua jenis perangkat. Ini langsung berinteraksi dengan perangkat eksternal ini. Pi bekerja melalui banyak lapisan pemrosesan sebelum mencapai pin yang terhubung dengan perangkat.

Dengan menggabungkan Raspberry Pi dan Arduino, kami dapat memanfaatkan yang terbaik dari masing-masing. Raspberry Pi menawarkan kekuatan pemrosesan tingkat tinggi dari komputer lengkap. Arduino menyediakan kontrol mentah atas perangkat eksternal. Pi memungkinkan kita untuk memproses aliran video dari kamera USB sederhana; sedangkan Arduino memungkinkan kita untuk mengumpulkan informasi dari berbagai sensor, dan

menerapkan logika untuk memahami semua data itu, dan kemudian mengembalikan temuan ringkas ke Pi. Anda akan mempelajari lebih lanjut tentang Raspberry Pi di Bab 2. Nanti, Anda akan menghubungkan Arduino ke Pi dan belajar tentang memprogramnya, serta cara menyampaikan informasi bolak-balik antara Arduino dan Pi.

1.6 ULASAN PROYEK

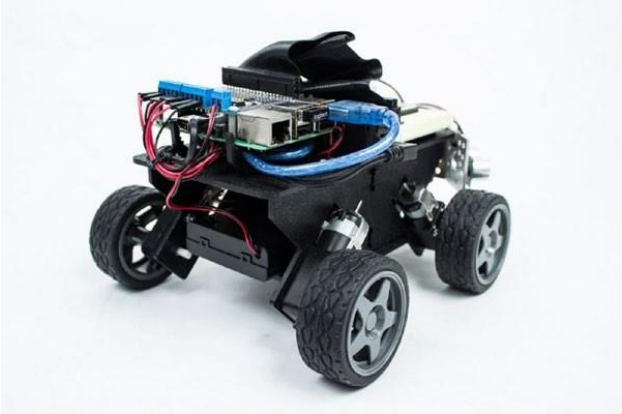
Dalam buku ini, kita akan membangun sebuah mobile robot kecil. Robot ini dirancang untuk mendemonstrasikan pelajaran yang Anda pelajari di setiap bab. Namun, sebelum kita benar-benar dapat membuat robot, kita perlu membahas banyak materi dan meletakkan dasar untuk pelajaran selanjutnya.

1.7 ROBOT

Robot yang akan kami buat adalah robot rover kecil beroda dua atau empat. Ini akan dapat mendeteksi rintangan dan tepi meja, dan mengikuti garis. Sasis yang saya pilih adalah robot beroda empat, tetapi ada desain lain yang cocok untuk proyek ini (lihat Gambar 1-3 dan 1-4).

Meskipun saya memberikan daftar bagian-bagian yang saya gunakan untuk proyek ini, Anda dipersilakan untuk menggunakan bagian apa pun yang Anda inginkan. Yang penting adalah mereka berperilaku dengan cara yang sama seperti yang saya sebutkan.

Gambar 1-3. Bagian depan robot kami menunjukkan sensor ultrasonik dan Pi T Cobbler di papan tempat memotong roti



Gambar 1-4. Bagian belakang robot kami menunjukkan Raspberry Pi dan papan kontrol motor

1.8 BILL OF MATERIAL (BOM)

Untuk sebagian besar, saya mencoba untuk membuat daftar materi se-generik mungkin. Ada beberapa item yang khusus vendor. Saya memilih mereka karena mereka menyediakan banyak fungsi dan kenyamanan. Pengendali motor DC & Stepper dan Pi T-Cobbler berasal dari pengecer online bernama Adafruit, yang merupakan sumber yang bagus untuk suku cadang, tutorial, dan inspirasi. Kit sasis berasal dari pengecer online bernama ServoCity, yang memproduksi banyak komponen mekanis untuk robotika.

Berikut ini adalah bagian khusus (ditunjukkan pada Gambar 1-5) yang kami gunakan dalam buku ini:

- Sasis robot Runt Rover Junior dari ServoCity.com
- Adafruit DC & Stepper Motor HAT untuk Raspberry Pi – Mini Kit PID: 2348
- GPIO Stacking Header untuk Pi A+/B+/Pi 2/Pi 3 – Ekstra-panjang 2×20 Pin PID: 2223 (memungkinkan penggunaan pelat tambahan dan Tukang sepatu untuk dipasang ke papan tempat memotong roti)
- Rakitan Pi T-Cobbler Plus – GPIO Breakout – Pi A+, B+, Pi 2, Pi 3, Zero PID: 2028

Gambar 1-5. Bagian sasis Runt Rover dan Pi T Cobbler, kabel pita, topi kontrol motor, dan tajuk yang diperpanjang

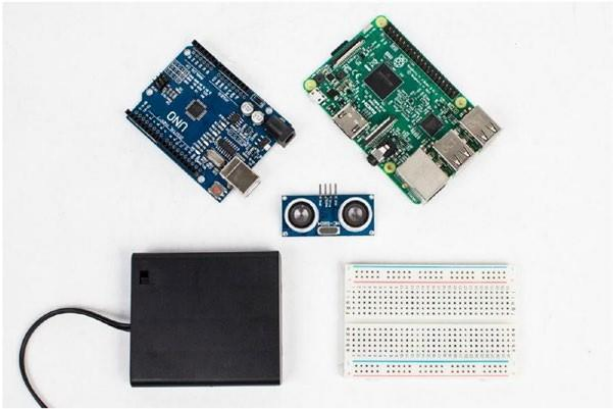
Bagian berikut (ditunjukkan pada Gambar 1-6) cukup umum dan dapat dibeli dari sebagian besar vendor:

- Raspberry Pi 3 – Model B – ARMv8 dengan RAM 1G
- Arduino Uno
- 4 × dudukan baterai AA dengan sakelar hidup/mati (menghidupkan motor)
- Paket Baterai USB – Kapasitas 2200 mAh – Output PID 5V 1A: 1959 (menghidupkan Raspberry Pi)
- Papan tempat memotong roti setengah ukuran
- Sensor ultrasonik – HC-SR04

  Anda mungkin ingin mendapatkan beberapa di antaranya. Seperti yang akan Anda temukan, sensor ultrasonik tidak dapat diandalkan pada sudut tertentu, dan bagus untuk memiliki susunannya. Saya menggunakan setidaknya tiga di sebagian besar proyek saya.
- Kumpulan kabel jumper (lihat Gambar 1-7)

Anda membutuhkan jumper pria-ke-pria dan jumper pria-wanita. Ini adalah ide yang baik untuk mendapatkan mereka dalam beberapa warna. Hitam dan merah digunakan untuk memberi daya pada perangkat Anda. Koleksi warna lain membantu Anda memahami sirkuit Anda. Untungnya, Anda bisa mendapatkan jumper dari semua jenis yang terbuat dari kabel pita warna-warni.

- Kabel USB untuk Arduino Anda
- Kabel micro USB untuk Raspberry Pi
- Pengisi daya telepon USB umum, lebih disukai untuk ponsel cerdas atau tablet modern yang dapat menyediakan daya 2 amp
- TV HDMI atau monitor komputer
  Sebagian besar monitor komputer tidak memiliki port HDMI. Namun, Anda bisa mendapatkan konverter HDMI-ke-DVI yang memungkinkan Anda menggunakan monitor yang ada.
- Keyboard dan mouse USB (saya suka kombinasi keyboard dan touchpad nirkabel Logitech K400, tetapi ada banyak pilihan di luar sana)
- Komputer yang terhubung ke jaringan
- Wi-Fi atau kabel Ethernet untuk Pi



Gambar 1-6. Bagian umum: Raspberry Pi, Arduino Uno, sensor ultrasonik, tempat baterai, dan papan tempat memotong roti

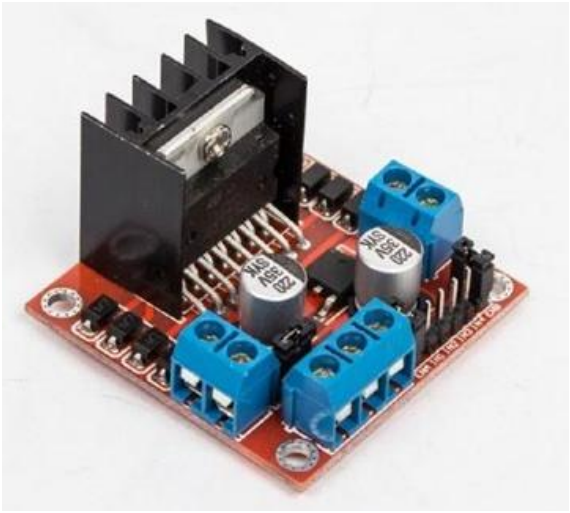
Gambar 1-7. Jumper dalam bentuk kabel pita. Tarik apa yang Anda butuhkan

Anda tidak perlu menjadi mewah dengan monitor dan keyboard. Setelah Anda membaca Bab 2, tempat kami menginstal dan mengkonfigurasi Raspberry Pi , Anda tidak lagi membutuhkannya. Saya memiliki beberapa keyboard nirkabel karena saya biasanya memiliki beberapa proyek sekaligus. Untuk monitor, saya cukup menggunakan salah satu monitor komputer saya dengan adaptor HDMI-to-DVI.

Jika Anda tidak menggunakan kit sasis dengan motor dan roda yang disertakan, Anda juga memerlukan suku cadang berikut (lihat Gambar 1-8):

- Hobby gearmotor – 200 RPM (pasangan)
- Roda – 65mm (ban karet, pasang)

Gambar 1-8. Motor dan roda gigi DC



Gambar 1-9. Modul pengontrol motor ganda L298N hadir dalam berbagai jenis, tetapi pada dasarnya berfungsi sama

Jika Anda tidak ingin menggunakan Adafruit Motor dan Stepper Hat, Anda juga dapat menggunakan hampir semua pengontrol motor, meskipun masing-masing memiliki antarmuka dan kode yang berbeda. Pilihan yang umum dan cukup populer adalah L298N Dual Motor Controller (lihat Gambar 1-9).

Ada beberapa persediaan lain yang saya simpan karena digunakan di hampir setiap proyek. Di Bab 7, kami merakit robot; Anda juga memerlukan pita pemasangan busa dua sisi,

ikatan ritsleting 4 inci, dan Velcro berperekat. Saat Anda melanjutkan dalam robotika, Anda akan sering beralih ke item ini. Bahkan, Anda mungkin ingin membeli berbagai ukuran dasi zip. Percayalah kepadaku.

1.9 RINGKASAN

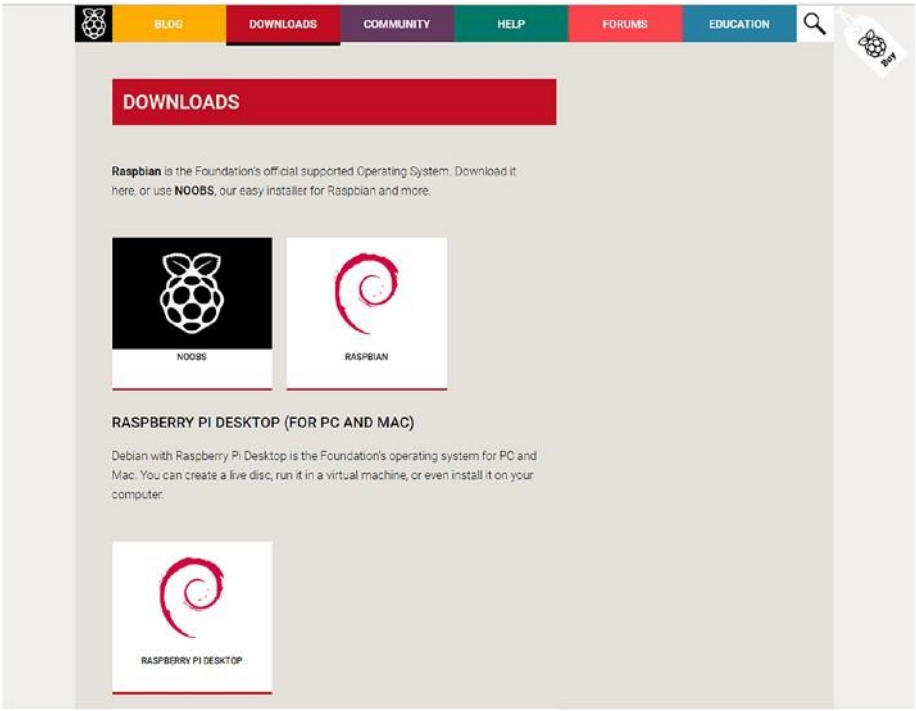
Memulai robotika tidak perlu sulit. Hal ini menantang, namun. Buku ini merupakan pengantar untuk beberapa keterampilan yang perlu Anda kembangkan jika Anda ingin berhasil di bidang ini. Robot yang kami buat memperkenalkan Anda ke Raspberry Pi, Linux, Arduino, sensor, dan visi komputer. Keterampilan ini dengan mudah ditingkatkan menjadi robot yang lebih besar dan proyek serupa lainnya.

BAB 2
PENGANTAR RASPBERRY PI

Tujuan dari buku ini adalah untuk menantang Anda untuk membangun sebuah robot sederhana yang akan diperluas dari waktu ke waktu. Buku ini dimaksudkan untuk menjadi sulit; Namun, itu tidak terlalu sulit atau tidak perlu rumit. Anda akan mengalami banyak komplikasi di sepanjang jalan, tetapi pemasangan sistem operasi pada Raspberry Pi Anda tidak perlu menjadi salah satunya.

2.1 MENGUNDUH DAN MENGINSTAL RASPBIAN

Pada dasarnya, ada dua metode untuk menginstal sistem operasi (OS) pada Pi Anda. Yang pertama melibatkan mengunduh gambar Raspbian terbaru, menulisnya ke kartu SD, dan pergi dari sana. Metode ini memerlukan penginstalan paket perangkat lunak pihak ketiga yang menulis gambar yang dapat di-boot pada kartu SD. Keuntungannya adalah dibutuhkan lebih sedikit ruang pada kartu SD Anda. Jika Anda menggunakan kartu SD minimal 8GB, ini mungkin berguna; jika Anda menjadi lebih besar, maka pertimbangan ini dapat diperdebatkan.



Gambar 2-1. Layar unduh Raspbian

Padahal pemasangan langsung tidak terlalu rumit (sebenarnya agak mudah), ada cara yang lebih mudah yang tidak melibatkan pemasangan perangkat lunak tambahan di sistem Anda. NOOBS (Perangkat Lunak Baru Di Luar Kotak) dirancang untuk membuat instalasi dan konfigurasi Raspberry Pi Anda lebih mudah. Hal ini memungkinkan Anda untuk memilih dari beberapa sistem operasi dan hanya menginstal. Namun, paket NOOBS tetap ada di kartu SD dan memakan ruang yang berharga. Itu memungkinkan Anda untuk kembali dan memperbaiki

OS Anda atau mengubah OS sepenuhnya, tetapi itu dapat ditangani secara manual dengan cukup mudah. Pada akhirnya, pilihan ada di tangan Anda. Saya akan membahas kedua opsi sehingga Anda dapat memilih jalur instalasi mana pun yang paling sesuai untuk Anda. Apapun pilihan yang Anda pilih, perjalanan Anda dimulai di halaman download Raspbian di www.raspberrypi.org/downloads/ (lihat Gambar 2-1).

2.2 RASPBIAN DENGAN OPENCV

Menjelang akhir buku ini, kami akan bekerja dengan visi komputer untuk menunjukkan kepada Anda mengapa Anda harus menggunakan Raspberry Pi daripada platform yang kurang mampu. Namun, untuk melakukan itu, Anda perlu menginstal OpenCV di Pi Anda. Sayangnya, tidak ada penginstal OpenCV sederhana untuk Raspberry Pi. Karena Pi berjalan pada prosesor ARM, paket harus dikompilasi dari kode sumber, yang merupakan proses enam jam. Untuk mempermudah Anda, saya mengkompilasi OpenCV di Raspbian Jesse dan membuat gambar yang dapat diunduh di https://github.com/jcicolani/Jesse-OpenCV. Anda masih harus melalui proses instalasi dan konfigurasi untuk menyesuaikan instalasi. Gambar menyertakan pengaturan default yang perlu Anda ubah (dengan beberapa pengecualian yang diperlukan untuk membuat build).

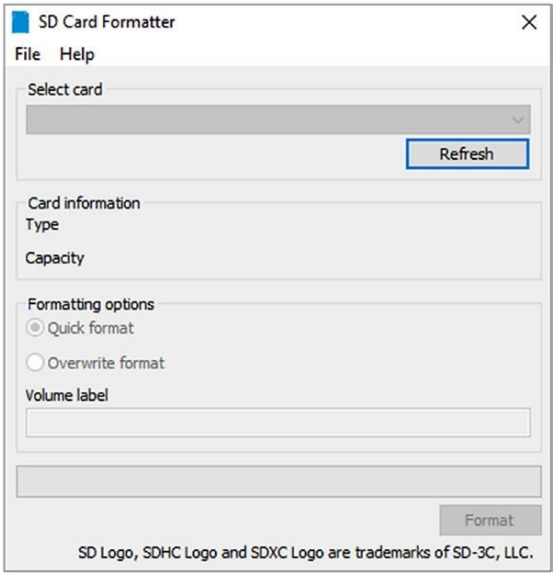
2.3 CARA "SULIT"

Metode yang lebih sulit menginstal image Raspbian OS langsung di kartu SD—siap untuk boot. Ini adalah metode yang saya gunakan karena sebenarnya tidak lebih rumit dari metode sebelumnya, dan memungkinkan saya untuk menggunakan versi yang tidak tersedia melalui NOOBS. Anda memiliki dua opsi untuk instalasi Raspbian Anda. Jessie adalah sistem operasi versi stabil terbaru; itu yang akan kita gunakan. Opsi pertama adalah Raspbian Jessie dengan PIXEL—GUI baru mereka yang dioptimalkan. Ini adalah unduhan 1,5GB, dan ini adalah gambar 4,2GB setelah didekompresi. Opsi kedua adalah Raspbian Jessie Lite, gambar minimal dengan unduhan 300MB yang jauh lebih kecil (1,4GB setelah dekompresi). Namun, minimal berarti tidak ada GUI, jadi semuanya dilakukan melalui baris perintah. Jika Anda penggemar Linux tanpa kepala, maka ini adalah opsi untuk Anda. Kami akan menggunakan instalasi yang lebih besar dengan PIXEL.

Jika Anda memiliki klien BitTorrent yang terinstal, klik Unduh Torrent. Ini jauh lebih cepat daripada mengunduh file .zip.

1. Navigasikan ke www.raspberrypi.org/downloads/.
2. Klik gambar Raspbian.
3. Pilih rasa Raspbian yang ingin Anda instal.
4. Setelah unduhan selesai, dekompresi file di suatu tempat yang mudah Anda temukan.
5. Unduh dan instal Win32 Disk Imager. Ini memungkinkan Anda untuk menulis file gambar yang baru saja Anda unduh ke kartu micro SD. Anda bisa mendapatkannya di https://sourceforge.net/projects/win32diskimager/.
6. Secara opsional, Anda mungkin juga ingin mengunduh SDFormatter untuk memastikan bahwa kartu SD Anda disiapkan dengan benar. Anda bisa mendapatkannya di www.sdcard.org/downloads/formatter_4/.

7. Masukkan kartu micro SD Anda ke pembaca kartu yang terhubung ke komputer Anda.
8. Jika sudah download dan install SDFormatter, buka. Anda akan melihat kotak dialog yang mirip dengan yang ditunjukkan pada Gambar 2-2.



Gambar 2-2. Pemformat Kartu SD

9. Pastikan Anda memilih drive yang mewakili kartu SD Anda. Anda akan memformatnya, jadi jika Anda memilih hal yang salah, itu akan menghapus apa pun yang Anda miliki di drive itu. Alat ini biasanya memilih yang benar secara default, tetapi periksa kembali. Sebaiknya putuskan sambungan perangkat penyimpanan eksternal lainnya.
10. Pastikan bahwa Format size adjustment diatur ke On. Ini menghapus partisi lain pada kartu dan menggunakan semuanya. Biarkan semua pengaturan lainnya pada default.
11. Klik Mulai. Ketika proses selesai, Anda siap untuk menginstal OS.
12. Untuk mem-flash gambar ke kartu SD, buka Win32 Disk Imager.
13. Di kolom file gambar, pilih gambar Raspbian yang Anda unduh. Anda dapat mengklik ikon folder file untuk menavigasi ke sana.
14. Pastikan kartu SD Anda dipilih di kotak drop-down perangkat. Sekali lagi, memilih perangkat yang salah dapat menyebabkan dunia yang terluka; jadi perhatikan.
15. Klik Tulis.
16. Setelah proses selesai, keluarkan kartu dari pembaca kartu Anda.
17. Masukkan kartu ke dalam pembaca kartu micro SD pada Raspberry Pi.

Kedengarannya panjang, tetapi sangat cepat dan mudah dilakukan. Selanjutnya, mari kita berjalan melalui proses instalasi NOOBS.

2.4 CARA “MUDAH”

Saya menyebut cara ini sebagai cara yang “mudah”, meskipun cara yang sulit sebenarnya cukup mudah. Yang membuatnya mudah adalah Anda tidak perlu menulis gambar secara langsung. Anda mungkin ingin memformat kartu, tetapi jika itu kartu baru, itu mungkin tidak perlu. Untuk membuatnya lebih mudah, jika Anda membeli Pi Anda sebagai bagian dari

starter kit, itu mungkin datang dengan NOOBS yang sudah diinstal pada kartu micro SD. Jika ini masalahnya, Anda dapat melewati beberapa langkah pertama.

Anda memiliki dua opsi: NOOBS dan NOOBS Lite. NOOBS menyertakan gambar Raspbian dengan unduhan, jadi Anda tidak perlu terhubung ke jaringan untuk mengunduh apa pun setelah ada di kartu SD Anda. Anda memiliki opsi untuk memilih OS lain, jika Anda mau, tetapi Anda harus menghubungkan Pi Anda ke jaringan agar NOOBS dapat mengunduhnya. NOOBS Lite tidak menyertakan gambar Raspbian penuh. Untuk tujuan kami, pilih instalasi NOOBS standar.

1. Klik gambar NOOBS pada halaman Download.
2. Pilih rasa NOOBS Anda. Jika Anda memiliki klien BitTorrent yang terinstal, klik Unduh Torrent. Ini jauh lebih cepat daripada mengunduh file .zip.
3. Secara opsional, Anda mungkin juga ingin mengunduh SDFormatter untuk memastikan bahwa kartu SD Anda disiapkan dengan benar. Anda bisa mendapatkannya di www.sdcard.org/downloads/formatter_4/.
4. Jika Anda mengunduh dan menginstal SDFormatter, buka.
5. Pastikan Anda memilih drive yang mewakili kartu SD Anda. Anda akan memformatnya, jadi jika Anda memilih hal yang salah, itu akan menghapus apa pun yang Anda miliki di drive itu. Alat ini biasanya memilih yang benar secara default, tetapi periksa kembali. Sebaiknya putuskan sambungan perangkat penyimpanan eksternal lainnya.
6. Pastikan bahwa Penyesuaian ukuran format diatur ke Hidup. Ini menghapus partisi lain pada kartu dan menggunakan semuanya. Biarkan semua pengaturan lainnya pada default.
7. Klik Mulai. Ketika proses telah selesai, Anda siap untuk menginstal OS.
8. Buka zip file NOOBS langsung ke kartu SD.
9. Keluarkan kartu dari pembaca kartu Anda.
10. Masukkan kartu ke dalam pembaca kartu micro SD pada Raspberry Pi.
11. Pada titik ini, Anda perlu menghubungkan Pi Anda untuk melanjutkan. Jadi, lanjutkan ke bagian "Menghubungkan Raspberry Pi" di bab ini. Setelah Anda menyelesaikan langkah-langkah tersebut, kembali ke bagian ini untuk melanjutkan penyiapan.
12. Saat Anda menghubungkan daya ke Raspberry Pi, itu boot ke layar instalasi NOOBS. Jika Anda menggunakan NOOBS Lite, Anda memiliki pilihan OS. Jika Anda menggunakan unduhan NOOBS standar, satu-satunya pilihan Anda adalah Raspbian (tidak apa-apa karena itulah yang kami gunakan).
13. Klik Raspbian untuk memastikan bahwa itu dipilih. Pastikan juga Anda memilih bahasa yang benar di bagian bawah layar (dalam kasus saya, ini adalah bahasa Inggris (AS)).
14. Klik tombol Instal di bagian atas layar.

Instalasi bisa memakan waktu agak lama, jadi silakan dan ambil secangkir kopi.

2.5 MENGHUBUNGKAN RASPBERRY PI

Sekarang kartu micro SD Anda siap digunakan, Anda perlu menghubungkan Raspberry Pi Anda. Jika Anda menggunakan Pi generasi pertama yang asli, ini sedikit lebih rumit. Namun,

setiap model setelah yang asli, menyertakan beberapa port USB dan konektor HDMI untuk mempermudah segalanya. Menghubungkan Pi sangat sederhana.

1. Hubungkan monitor Anda melalui kabel HDMI. Jika Anda menggunakan televisi kecil yang dilengkapi dengan sambungan komponen daripada HDMI, soket audio pada Pi adalah soket komponen empat kutub. Anda memerlukan konverter RCA-ke-3.5mm, biasanya dalam bentuk kabel, untuk melakukan ini.
2. Hubungkan keyboard dan mouse Anda ke port USB. Saya menggunakan kombinasi keyboard/touchpad nirkabel karena ringkas dan portabel.
3. Pastikan kartu micro SD Anda dengan Raspbian atau NOOBS terpasang di port micro SD di Pi. Pada dasarnya, ini adalah hard drive untuk perangkat kecil computer anda, jadi harus di tempat yang tepat. Itu tidak akan membaca OS melalui pembaca kartu SD yang terhubung ke salah satu port USB.
4. Jika Anda menggunakan kabel Ethernet, sambungkan ke port Ethernet. Anda juga dapat mencolokkan dongle Wi-Fi ke port USB. Jika Anda menggunakan Pi 3, seperti saya, Wi-Fi sudah terpasang.
5. Hubungkan daya 5V ke port micro USB. Port ini hanya untuk daya. Anda tidak dapat mengakses papan melalui USB.

Itu dia. Raspberry Pi Anda akan terlihat seperti yang ditunjukkan pada Gambar 2-3. Pi harus boot di monitor Anda. Jika Anda menginstal NOOBS, kembali ke langkah 10 dari instalasi Noobian untuk menyelesaikan proses instalasi.

Gambar 2-3. Koneksi Raspberry Pi

Sekarang setelah Anda terhubung dan boot, Anda harus masuk. Berikut ini adalah kredensial default untuk instalasi Raspbian:

- Nama pengguna: pi
- Kata sandi: raspberry

Tentu saja, nama pengguna dan kata sandi default tidak pernah aman. Jadi, agar teman keamanan siber Anda tidak kabur dengan robot Anda, salah satu hal pertama yang akan kami lakukan adalah mengubah kata sandi. Nanti di konfigurasi, kita akan mengubah username default.

2.6 MENGONFIGURASI PI ANDA

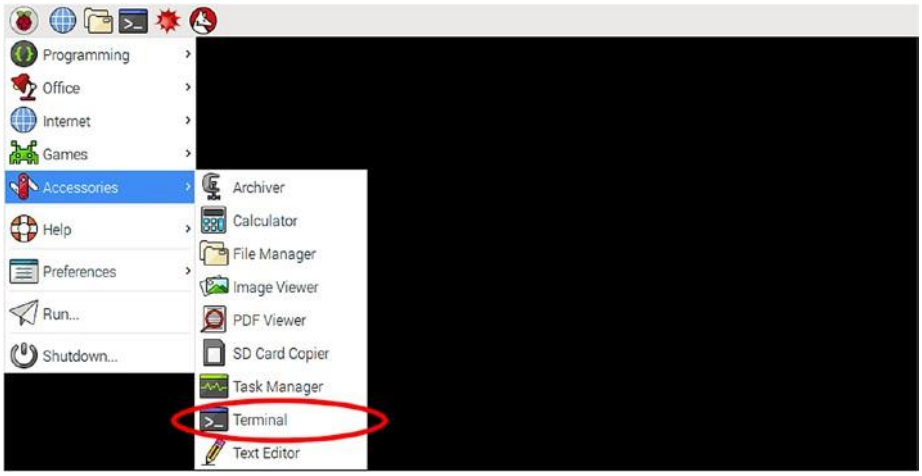
Sekarang setelah kita menyelesaikan instalasi awal, kita akan beralih ke sedikit penyesuaian. Pi memiliki beberapa fitur yang dapat Anda aktifkan, tergantung pada penggunaan khusus Anda. Awalnya, mereka tidak diaktifkan untuk mengurangi beberapa overhead yang diperlukan untuk menjalankan OS. Pengaturan konfigurasi yang akan kita terapkan adalah untuk keamanan dan kenyamanan.

2.7 MENGGUNAKAN RASPI-CONFIG

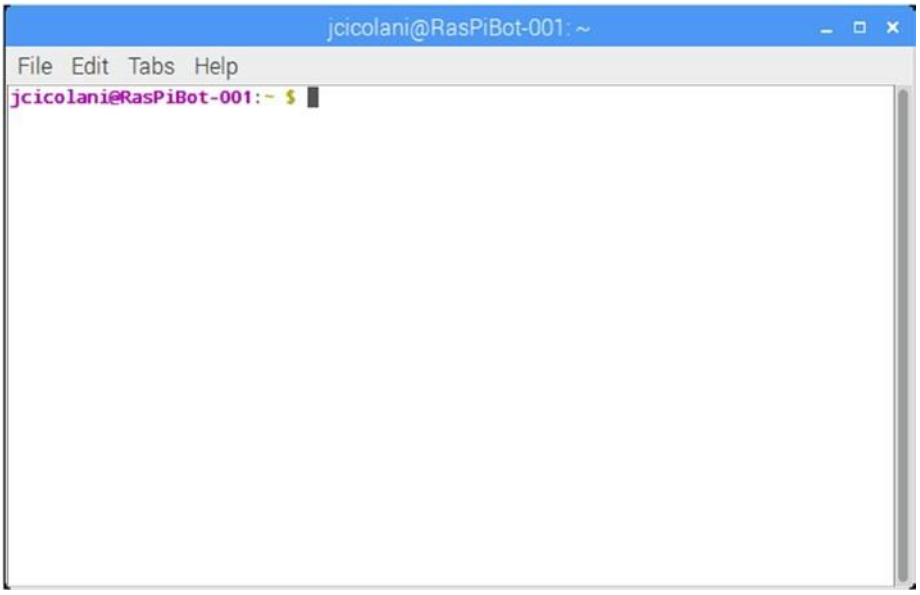
Untuk membuat penyesuaian, orang-orang baik di Raspberry Pi Foundation telah menyertakan utilitas yang disebut raspi-config. Terminal baris perintah diperlukan untuk menggunakannya. Satu perintah dimasukkan sekarang, tetapi saat kita bergerak maju dalam lokakarya, Anda akan menjadi lebih akrab dengan jendela terminal. Jika Anda baru mengenal Linux (yang menjadi dasar Raspbian), ini bisa sedikit menakutkan. Tidak perlu, dan saya akan melakukan yang terbaik untuk memudahkan Anda melakukannya. Tetapi Anda harus belajar cara mengatasinya. Anda dapat menemukan informasi lebih lanjut tentang utilitas raspi-config di
www.raspberrypi.org/documentation/configuration/raspi-config.md.

Pada titik ini, Anda seharusnya sudah boot ke Raspberry Pi Anda. Jika tidak, lakukan sekarang. Kami akan melakukan beberapa hal untuk mengonfigurasi Pi, dimulai dengan memperluas sistem file untuk memanfaatkan seluruh kartu SD. Secara default, Raspbian tidak menggunakan seluruh kartu SD, jadi kami ingin memberitahukannya. Jika Anda menggunakan NOOBS, ini telah dilakukan untuk Anda, jadi Anda dapat melewati langkah ini.

1. Klik ikon Raspberry Pi di bagian atas layar. Ini akan membuka daftar aplikasi.
2. Pilih Aksesoris => Terminal, seperti yang ditunjukkan pada Gambar 2-4. Saat dibuka, jendela terminal ditampilkan (lihat Gambar 2-5).

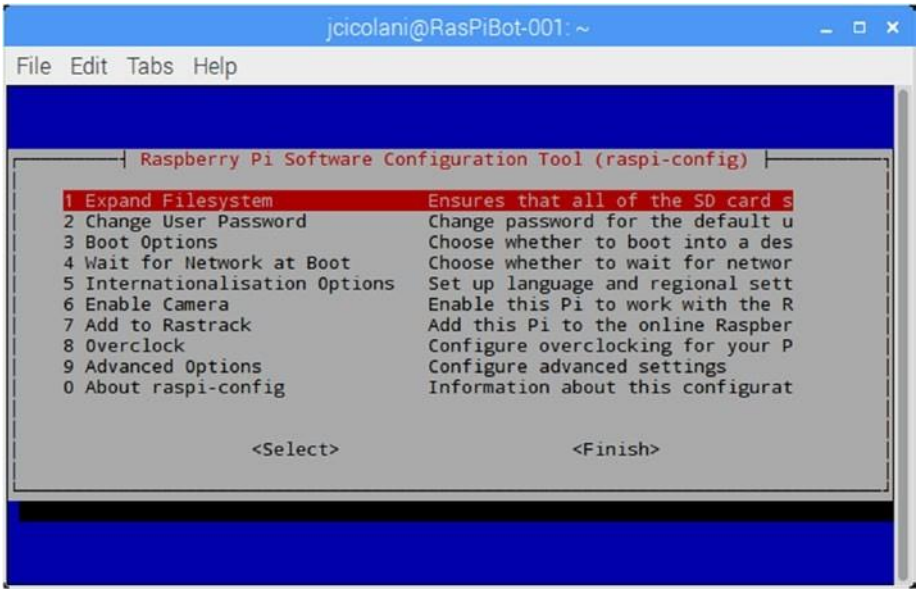


Gambar 2-4. Pilihan terminal dari daftar aplikasi. Ikon terminal juga ada di bilah akses cepat.



Gambar 2-5. jendela terminal

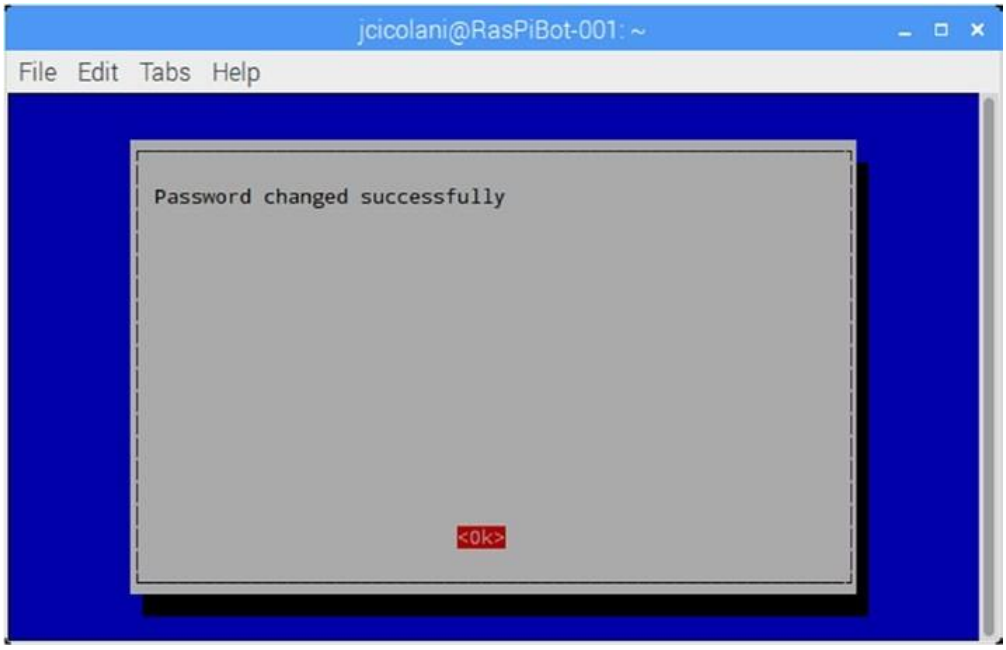
3. Ketik sudo raspi-config.
   Ini akan membuka Alat Konfigurasi Perangkat Lunak Raspberry Pi, seperti yang ditunjukkan pada Gambar 2-6.



Gambar 2-6. Layar raspi-config. Sebagian besar opsi tingkat OS dapat diatur di sini, termasuk mengaktifkan dan menonaktifkan layanan.

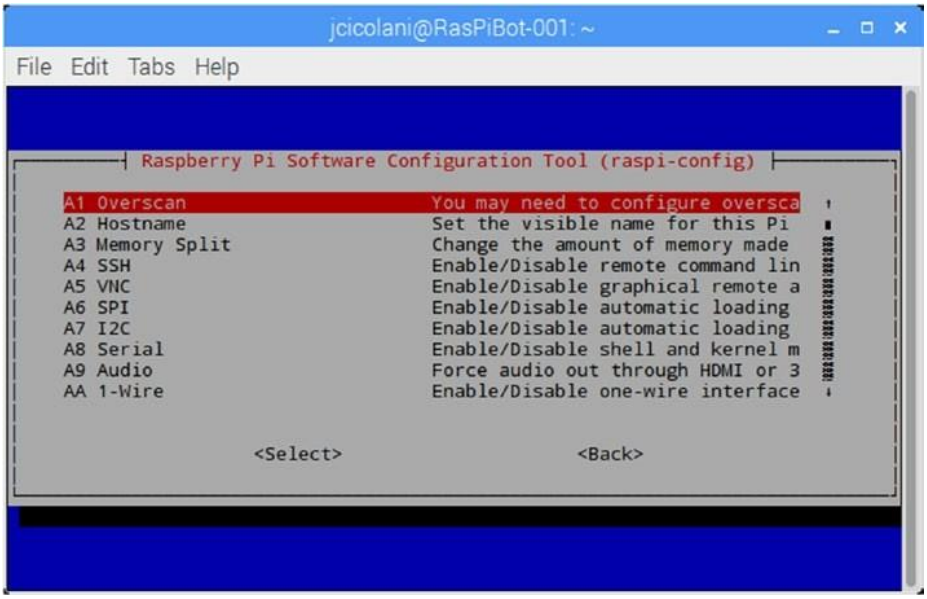
   Versi Raspbian yang lebih baru secara otomatis memperluas sistem file Anda saat pertama kali Anda memulai Pi. Kecuali Anda menggunakan versi Raspbian yang lebih lama, Anda seharusnya dapat melewati langkah berikutnya dan beralih ke mengubah kata sandi.
4. Pastikan Expand file system disorot.

5. Tekan Enter. Sistem memunculkan pesan tentang memperluas sistem file dan meminta Anda untuk reboot. (Kami akan reboot nanti, setelah kami melakukan sebagian besar perubahan.)
   Selanjutnya, kami akan mengubah kata sandi pengguna.
6. Pastikan Ubah kata sandi pengguna disorot.
7. Tekan Enter. Sistem menampilkan pesan yang mengatakan bahwa Anda akan dimintai kata sandi baru.
8. Tekan Enter. Ini menjatuhkan Anda ke terminal untuk memasukkan kata sandi baru.
9. Masukkan kata sandi baru Anda dan tekan Enter.
10. Konfirmasikan kata sandi baru Anda dan tekan Enter. Ini menampilkan konfirmasi bahwa kata sandi berhasil diperbarui (lihat Gambar 2-7).
11. Tekan Enter.
    Beberapa langkah selanjutnya mengaktifkan beberapa layanan yang akan kita gunakan nanti.
    Kami akan mulai dengan mengubah nama host Pi Anda menjadi sesuatu yang unik yang lebih mudah ditemukan di jaringan. Ini menjadi sangat penting ketika Anda berada di sebuah ruangan dengan 20 Raspberry Pis lainnya.



Gambar 2-7. Konfirmasi perubahan kata sandi di raspi-config

12. Pastikan opsi lanjutan disorot, lalu tekan Enter. Ini menampilkan antarmuka dan opsi lainnya (lihat Gambar 2-8).



Gambar 2-8. opsi lanjutan raspi-config. Nama host dan aktivasi layanan diakses di sini.

Nama host adalah bagaimana Raspberry Pi Anda muncul di jaringan. Anda ingin memberi Pi Anda nama yang unik, terutama ketika Anda mempertimbangkan berapa banyak dari mereka yang mungkin ada di jaringan pada waktu tertentu. Nama host harus bermakna bagi aplikasi dan unik.

13. Sorot Nama Inang dan tekan Enter.
14. Kotak dialog menjelaskan persyaratan untuk nama host. Itu harus hanya karakter alfanumerik: tidak ada simbol, tidak ada tanda hubung, dan tidak ada garis bawah. Tekan Enter untuk melanjutkan.
15. Masukkan nama host baru Anda dan tekan Enter.
    SSH memungkinkan kita untuk mengakses Pi melalui jendela terminal (klien SSH) dari komputer lain. Di Windows, Putty adalah klien SSH gratis yang sangat populer. SSH tidak menyediakan GUI. Semua interaksi dibuat menggunakan perintah terminal. Ini membantu jika Anda ingin cepat menjalankan suatu program, menginstal software, dan lain sebagainya. Saat Anda menjadi lebih akrab dengan terminal, Anda mungkin akan menemukan diri Anda sendiri menggunakan SSH untuk menghubungkan perintah sederhana, sementara memesan VNC (remote desktop) untuk tugas yang lebih terlibat seperti menulis program.
16. Kembali ke menu opsi lanjutan.
17. Pilih Aktifkan SSH dan tekan Enter.
18. Konfirmasikan bahwa Anda ingin mengaktifkan SSH dan tekan Enter.
19. Tekan Enter lagi untuk kembali ke menu.
    I2C adalah protokol komunikasi serial yang sangat populer di embedded system seperti Pi, Arduino, dan lain sebagainya. Ini memungkinkan komunikasi yang kuat dengan banyak perangkat dengan menggunakan sejumlah pin. Papan kontrol motor yang akan kita gunakan berkomunikasi melalui I2C. (Jika nanti Anda memilih untuk menambahkan papan lain, seperti papan kontrol servo, itu juga akan menggunakan

I2C.) Selama perangkat memiliki alamat yang berbeda, Anda dapat terus menumpuknya.

20. Kembali ke menu opsi lanjutan.
21. Pilih Aktifkan I2C dan tekan Enter.
22. Konfirmasikan bahwa Anda ingin mengaktifkan SSH, lalu tekan Enter.
23. Tekan Enter lagi untuk kembali ke menu.
    Karena kami juga berencana untuk menggunakan Raspberry Pi headless (tanpa monitor, keyboard, atau mouse yang terpasang), mari kita atur untuk boot ke konsol secara otomatis. Jangan khawatir; cukup mudah untuk meluncurkan GUI desktop saat Anda mau, seperti yang akan Anda lihat.
24. Buka opsi boot dan tekan Enter.
25. Pilih Konsol dan tekan Enter. Jika Anda yakin bahwa Anda akan menjadi satu-satunya yang mengakses Pi Anda secara langsung, Anda dapat memilih Console Autologin. Autologin tidak berlaku untuk sesi jarak jauh, hanya akses langsung dengan keyboard dan monitor.
26. Dengan semua pengaturan diperbarui, sorot Selesai dan tekan Enter.
27. Pi menanyakan apakah Anda ingin reboot. Pilih Ya dan tekan Enter.
    Pada titik ini, Pi Anda reboot. Ini mungkin memakan waktu beberapa menit, terutama jika Anda tidak menginstal melalui NOOBS dan Pi harus memperluas sistem file Anda.
    Ingat, kami mengatur Pi untuk boot ke konsol secara default. Karena beberapa langkah berikutnya sudah selesai melalui baris perintah, kita tidak perlu memuat GUI. Namun, mari kita tetap melakukannya sehingga Anda dapat melihat betapa mudahnya itu.
28. Ketik startx dan tekan Return.
    Anda sekarang berada di desktop GUI. Untuk keluar dari desktop, lakukan hal berikut:
    1. Klik menu program (raspberry di pojok kiri atas).
    2. Klik tombol daya.
    3. Pilih Keluar ke baris perintah.

    Anda sekarang harus kembali ke baris perintah.

2.8 PENGGUNA

Pengguna default pada setiap instalasi Raspbian adalah pi. Sebelumnya, kami mengubah kata sandi untuk membuatnya lebih aman. Namun, Anda mungkin tidak ingin selalu masuk sebagai pengguna pi. Ingat ketika saya mengatakan kami akan mulai menggunakan terminal lebih banyak? Nah, itu dimulai sekarang. Cara termudah untuk membuat dan mengelola pengguna adalah melalui baris perintah. Kami akan menjalani proses itu sekarang.

Mengamankan Root

Selain pengguna default, pi, ada pengguna default lain di Pi. Ini adalah pengguna root. Pengguna root pada dasarnya adalah pengguna administratif yang digunakan oleh mesin untuk menjalankan perintah tingkat rendah. Pengguna ini memiliki akses ke segalanya dan dapat melakukan apa saja karena, yah, ini adalah mesinnya. Tidak seperti pengguna pi default, bagaimanapun, root tidak memiliki kata sandi default. Ini tidak memiliki kata sandi.

Jadi, sementara kita mengonfigurasi dan mengamankan komputer untuk robot kita, mari kita lanjutkan dan berikan kata sandi kepada pengguna root.

1. Buka jendela terminal.
2. Ketik sudo passwd root. (Perhatikan bahwa passwd adalah perintah yang tepat dan bukan salah ketik.)
3. Masukkan kata sandi baru untuk pengguna root.
4. Masukkan kembali kata sandi untuk mengonfirmasi.

Pengguna root Anda sekarang diamankan, yang bagus karena Anda akan membutuhkannya untuk langkah konfigurasi selanjutnya.

Ubah Nama Pengguna Default

Hal pertama yang akan Anda lakukan adalah mengubah nama pengguna default menjadi sesuatu yang Anda pilih. Apa yang akan dilakukan adalah mengganti nama pengguna pi dengan nama pengguna Anda sendiri. Ini memberikan lapisan keamanan lain pada perangkat; sekarang, seseorang tidak hanya perlu mengetahui kata sandinya, mereka bahkan tidak akan memiliki nama pengguna default untuk digunakan. Itu juga mempertahankan beberapa izin khusus dan tidak terdokumentasi yang diberikan kepada pengguna default.

1. Keluar dari pengguna pi. Anda dapat melakukan ini melalui sistem menu atau hanya dengan mengetik logout di terminal.
2. Masuk dengan pengguna root Anda yang sekarang aman.
3. Ketik

   `usermod -l <newname> pi`

   <newname> adalah nama pengguna baru yang Anda pilih.

   Jangan sertakan < or > dalam perintah.
4. Untuk memperbarui nama direktori home, ketik

   `usermod -m -d /home/<newname> <newname>`

   Sekali lagi, <newname> adalah nama pengguna baru yang Anda gunakan pada langkah sebelumnya.
5. Keluar dari pengguna root dan masuk kembali dengan nama pengguna baru Anda.

Pada titik ini, Anda telah mengubah kredensial pengguna default untuk pengguna default dan pengguna root. Anda juga telah mengubah nama host. Ini adalah jumlah minimum yang diperlukan untuk mengamankan pi dan robot Anda. Raspberry Pi Anda sekarang sudah diatur, dikonfigurasi, dan siap digunakan. Ada satu hal lagi yang ingin kami lakukan sebelum melanjutkan ke bab berikutnya, dan itu adalah menyiapkan Pi Anda tanpa kepala.

Membuat mesin "tanpa kepala" berarti mengonfigurasinya sehingga Anda tidak perlu lagi menghubungkan monitor, keyboard, dan mouse ke mesin untuk mengoperasikannya. Ini biasanya dilakukan dengan dua cara: dengan sakelar KVM atau dengan menyetel atas akses jarak jauh. Pada robot seluler, menghubungkan KVM sebenarnya bukan pilihan. Faktanya, itu akan sedikit berbeda dari sekadar menghubungkan semuanya. Yang ingin kami lakukan adalah mengatur Pi sehingga kami dapat mengaksesnya dari jarak jauh melalui jaringan. Tapi pertama-tama, pastikan Anda terhubung ke jaringan Anda.

2.9 MENGHUBUNGKAN KE JARINGAN NIRKABEL

Ketika Anda awalnya menghubungkan Raspberry Pi Anda, Anda memiliki opsi untuk menghubungkan kabel Ethernet. Jika Anda melakukan ini, maka Anda sudah berada di jaringan. Namun, Anda masih ingin terhubung ke jaringan nirkabel Anda. Ini memungkinkan Anda untuk melakukan remote ke robot Anda saat sedang bergerak. Anda akan dapat mengirimkan perintah, memperbarui kode, dan bahkan melihat umpan video dari webcam yang terpasang di Bab 10.

Untuk melakukan koneksi ke jaringan nirkabel, Anda memerlukan koneksi Wi-Fi.

Jika Anda menggunakan Raspberry Pi 3, maka Anda sudah memilikinya; jika tidak, Anda perlu mendapatkan dongle Wi-Fi, sebaiknya dongle yang dapat ditenagai oleh port USB. Sebuah penelitian kecil di sini berjalan jauh.

1. Masuk ke antarmuka GUI dengan mengetik startx.
2. Klik ikon jaringan di kanan atas layar Anda. Ikon ini terlihat seperti Gambar 2-9.

Gambar 2-9. Ikon koneksi jaringan

3. Pilih jaringan nirkabel Anda dari daftar.
4. Masukkan kunci keamanan untuk jaringan Anda.
   Anda sekarang harus terhubung ke jaringan nirkabel Anda.

2.10 PERGI TANPA KEPALA

Anda tidak akan ingin membawa monitor, keyboard, dan mouse tambahan saat mengerjakan lokakarya ini. Untuk membuat hidup Anda jauh lebih mudah, mari kita atur agar Anda dapat mengakses Pi tanpa kepala.

2.11 AKSES JARAK JAUH

Ada dua cara untuk mendapatkan akses jarak jauh. Salah satu metode adalah dengan menggunakan SSH, yang memungkinkan Anda untuk terhubung ke perangkat jarak jauh menggunakan klien terminal. Metode lainnya adalah mengatur desktop jarak jauh.

Desktop Jarak Jauh Dengan Xrdp

Mari kita mulai dengan mengakses desktop dari jarak jauh dari komputer lain. Petunjuk berikut adalah untuk pengguna Windows. Sebagian besar instalasi Windows modern dilengkapi dengan Remote Desktop Connection yang sudah diinstal, yang akan kami gunakan untuk terhubung ke Pi setelah disiapkan.

Mari kita instal beberapa layanan di Pi: tightVNCserver dan xrdp. Secara teoritis, xrdp harus menginstal server VNC dengan sendirinya. Pada kenyataannya, tidak. Pada titik ini, Anda harus berada di baris perintah di Pi Anda.

1. Ketik `sudo apt-get install tightvncserver`.
2. Selesaikan instalasi.

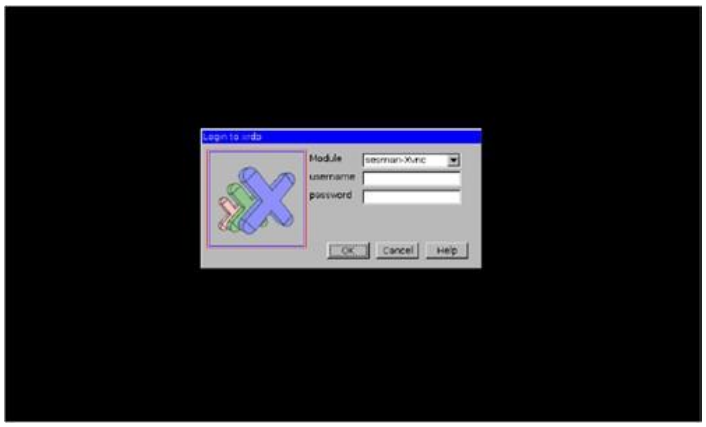
3. Ketik `sudo apt-get install xrdp`.
   Ketika instalasi selesai, Anda harus siap untuk pergi.

Untuk terhubung, lakukan hal berikut:

1. Di Pi, ketik `sudo ifconfig`.
2. Catat alamat Internet (inet addr) di blok eth0 jika Anda menggunakan kabel Ethernet, atau blok wlan0 untuk Wi-Fi.
3. Di laptop Anda, buka Remote Desktop Connection. Ini menampilkan kotak dialog koneksi, seperti yang ditunjukkan pada Gambar 2-10.

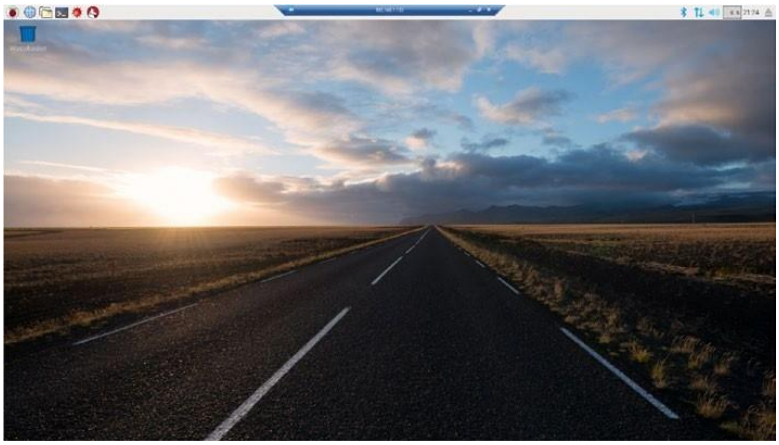
Gambar 2-10. Koneksi Desktop Jarak Jauh Windows

4. Masukkan addr inet dari Pi Anda.
5. Klik Hubungkan.
   Anda akan melihat layar desktop jarak jauh dengan formulir login xrdp (lihat Gambar 2-11).



Gambar 2-11. Layar login desktop jarak jauh XRDP

6. Masukkan nama pengguna dan kata sandi Anda.
7. Klik Oke. Ini membuka desktop dari Pi Anda (lihat Gambar 2-12).



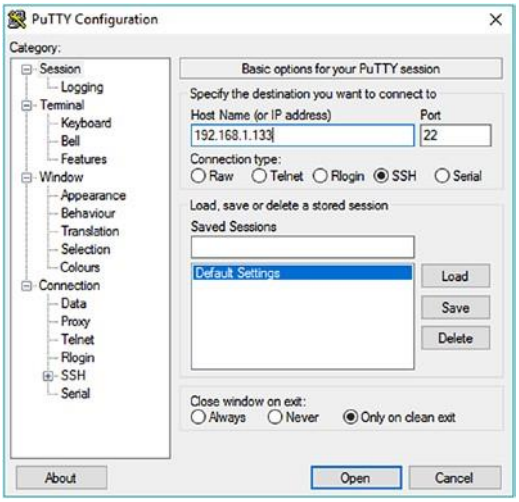
Gambar 2-12. Desktop Raspbian default dilihat melalui sesi desktop jarak jauh

Selama alamat IP Pi Anda tidak berubah, Anda tidak lagi memerlukan keyboard, mouse, atau monitor untuk menggunakan Pi Anda.

Ssh Dengan Putty

Klien SSH yang paling umum mungkin adalah Putty. Ini gratis untuk digunakan dan dapat diunduh dari www.chiark.greenend.org.uk/~sgtatham/putty/ download.html. File yang Anda unduh untuk Putty adalah file yang dapat dieksekusi, yang tidak perlu diinstal. Letakkan di desktop Anda atau di tempat yang mudah ditemukan. Untuk terhubung, lakukan hal berikut.

1. Buka antarmuka Putty (lihat Gambar 2-13).



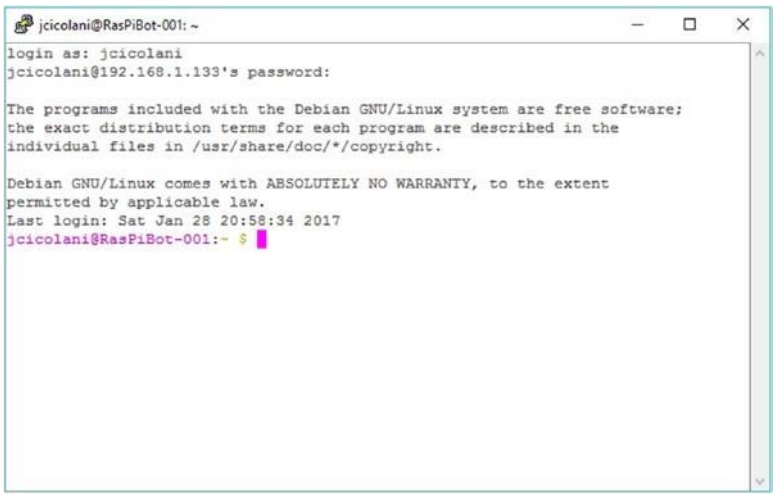
Gambar 2-13. Jendela konfigurasi Putty

2. Masukkan alamat IP Raspberry Pi Anda.
3. Klik Buka.
4. Anda mungkin akan mendapatkan peringatan keamanan, seperti yang ditunjukkan pada Gambar 2-14, tetapi kita tahu bahwa ini adalah koneksi yang tepat, jadi klik Ya.

Gambar 2-14. Peringatan keamanan pada koneksi SSH pertama dengan Putty

Jendela terminal terbuka, menanyakan nama pengguna dan kata sandi Anda.

5. Masukkan nama pengguna dan kata sandi Anda. Anda sekarang akan melihat prompt terminal, seperti yang ditunjukkan pada Gambar 2-15.



Gambar 2-15. Buka koneksi SSH

Itu dia. Anda sekarang terhubung melalui SSH ke Raspberry Pi Anda. Anda dapat memiliki beberapa koneksi sekaligus, tetapi tidak memiliki lebih dari yang Anda butuhkan. Beberapa koneksi berguna saat Anda bekerja dengan sesuatu seperti Sistem Operasi Robot (ROS). (Jangan khawatir, itu jauh.) ROS menjalankan banyak program melalui terminal. Masing-masing membutuhkan jendela terminalnya sendiri. Dengan Putty, Anda dapat memiliki koneksi terminal jarak jauh sebanyak yang Anda butuhkan.

Menemukan Perangkat Anda Di Jaringan

Untuk mengakses Pi Anda dari jarak jauh, Anda perlu mengetahui alamat IP-nya di jaringan. Umumnya, sakelar jaringan mempertahankan IP perangkat dari sesi ke sesi; Namun, ini tidak dijamin. Menemukan alamat IP perangkat Anda di jaringan bisa jadi rumit. Jika Anda di rumah dan memiliki akses ke panel admin router Anda, ini mungkin cara paling mudah untuk menemukan perangkat Anda. Cukup masuk ke router Anda, temukan daftar perangkat yang terhubung, dan gulir ke bawah hingga Anda menemukan nama host Raspberry Pi Anda.

Jika Anda perlu menemukan alamat IP tetapi jauh dari rumah, ada beberapa cara untuk melakukannya. Cara termudah adalah dengan menggunakan aplikasi Nmap di ponsel Anda. Saya menggunakan aplikasi bernama Fing di ponsel Android saya. Setelah ponsel terhubung ke jaringan Wi-Fi lokal, aplikasi memindai jaringan dan mencantumkan semua perangkat lain di jaringan itu. Anda cukup menggulir ke bawah daftar sampai Anda menemukan nama host Anda. Jika jaringan baru bagi Anda, Raspberry Pi Anda tidak akan terhubung secara otomatis. Situasi ini membuatnya sedikit rumit. Untuk mempermudah, bersiaplah sebelum Anda pergi. Saya pengguna Windows; jika tidak, Anda perlu mencari prosedur yang tepat untuk OS Anda. Saya melakukan operasi ini dengan laptop yang saya miliki saat bepergian. Ini memungkinkan saya untuk melakukan remote ke Pi cukup lama untuk terhubung ke Wi-Fi lokal dan mendapatkan alamat IP wlan0.

Perlu diingat bahwa IP diberikan oleh laptop. IP yang Anda dapatkan di akhir operasi ini kemungkinan besar tidak akan berfungsi dari mesin lain mana pun. Pada mesin Windows 7 atau yang lebih baru, Anda dapat melakukan langkah-langkah berikut untuk melakukan remote ke Pi Anda secara langsung untuk mendapatkan alamat IP-nya. Catat alamat IP jika Anda perlu terhubung langsung ke Pi untuk mengatur koneksi Wi-Fi baru. Anda akan memerlukan kabel Ethernet pendek, yang harus ditambahkan ke kit atau kotak peralatan Anda. Pastikan Anda dapat melihat Raspberry Pi Anda dengan pengaturan monitor, keyboard, dan mouse, atau melalui koneksi jarak jauh melalui jaringan Wi-Fi. Pi tidak dapat dihubungkan ke jaringan melalui kabel Ethernet karena port tersebut diperlukan untuk operasi ini.

1. Hubungkan kabel Ethernet ke laptop Anda.
2. Hubungkan ujung lainnya ke port Ethernet pada Pi Anda.
3. Buka jendela terminal pada Pi.
4. Ketik `sudo ifconfig`.
5. Cari inet addr di blok eth0.
6. Buka jendela terminal di laptop Anda. Anda dapat melakukan ini dengan mencari cmd di menu Start.
7. Ketik yang berikut di terminal Windows:
   `ping <your.Pis.IP.address>`
   <your.Pis.IP.address> adalah alamat IP eth0 dari Pi Anda.
8. Buka Remote Desktop Connection di laptop Anda.
9. Masukkan alamat IP dari Pi dan tekan Enter.

Anda sekarang harus memiliki koneksi jarak jauh dari laptop Anda langsung ke Raspberry Pi. Pastikan Anda menyimpan alamat IP ini di tempat yang dapat Anda temukan nanti. Koneksi Desktop Jarak Jauh harus mengingatnya, tetapi sebaiknya simpan juga di tempat lain. Sekarang setiap kali Anda mencoba untuk terhubung ke jaringan Wi-Fi baru, Anda dapat menggunakan kabel Ethernet untuk remote ke Pi Anda langsung dari laptop Anda. Setelah remote masuk, cukup pilih jaringan dari daftar jaringan yang tersedia dan masukkan kode sandi, jika ada.

2.12 RINGKASAN

Raspberry Pi dirancang untuk pembuat hobi. Komputer Linux kecil membuatnya sangat berguna untuk banyak jenis proyek yang berbeda, tetapi ini berarti Anda perlu belajar sedikit Linux. Pengembang di Raspberry Pi Foundation menyediakan versi Debian Linux yang mudah digunakan yang disebut Raspbian. Kami mengambil instalasi dasar selangkah lebih maju dengan mengonfigurasi akses jarak jauh. Ini memungkinkan Anda untuk mengakses robot Anda dari jarak jauh melalui jaringan Anda, yang berarti monitor dan keyboard tidak lagi diperlukan.

BAB 3
KURSUS SINGKAT DENGAN PYTHON

Tujuan buku ini adalah untuk menantang Anda membuat robot sederhana yang dapat dikembangkan dari waktu ke waktu. Hal ini dimaksudkan untuk menjadi sulit. Hal ini dimaksudkan untuk memberikan pengalaman langsung untuk membantu Anda melewati bagian tersulit dari pembelajaran robotika: diintimidasi oleh teknologi. Saya akan mengajari Anda beberapa dasar-dasar robotika dengan cara yang sama seperti saya belajar berenang—dengan dilemparkan ke dasar yang dalam sementara seseorang yang lebih berpengalaman mengawasi untuk memastikan Anda tidak tenggelam.

Jadi dengan itu, saya berharap Anda mengambil apa yang Anda alami dan menambahkannya melalui pembelajaran Anda sendiri. Saya akan membawa Anda ke arah yang benar, tetapi tidak akan ada banyak pegangan. Anda harus mengisi kekosongan dan mempelajari beberapa detail terutama beberapa di antaranya untuk aplikasi tertentu sendiri. Pengantar Python ini tidak berbeda. Saya akan menunjukkan cara menginstal alat, menggunakan editor, dan menulis beberapa program sederhana. Kita akan bergerak cepat melalui struktur program, sintaks dan masalah pemformatan, tipe data, dan variabel, dan langsung ke struktur kontrol dan beberapa aspek berorientasi objek dari Python. Jangan khawatir jika semua ini terdengar seperti celoteh tekno, Anda akan memahaminya sebelum akhir bab ini.

Di akhir bab ini, saya tidak berharap Anda dapat menulis program Anda sendiri. Apa yang saya harapkan adalah agar Anda mengetahui cara menulis kode, menggunakan dan merasa nyaman dengan editor, dan mengkompilasi dan menjalankan program. Yang terpenting, Anda harus dapat melihat kode orang lain dan dapat membacanya, memiliki pemahaman dasar tentang apa yang mereka coba lakukan, dan mengidentifikasi blok pembangun. Membedah kode orang lain penting untuk dipelajari dengan cepat. Salah satu premis buku ini adalah untuk tidak menemukan kembali roda. Sebagian besar dari apa yang akan Anda lakukan telah dilakukan sebelumnya, dan itu dapat ditemukan jika Anda melakukan sedikit pencarian. Mampu membaca dan memahami apa yang Anda temukan akan membantu Anda mencapai tujuan Anda.

Dalam hal sumber daya, berikut beberapa saran:

- Dukungan komunitas untuk Python sangat baik. Situs web Python adalah sumber yang sangat berharga untuk belajar dan berkembang dengan Python. Secara khusus, pastikan untuk memeriksa halaman pemula di www.python.org/about/getstarted/. Kami benar-benar mulai di sini di bagian berikutnya.
- Dapatkan sendiri satu atau dua buku bagus tentang Python. Buku ini membantu Anda memulai, tetapi ada banyak detail yang tidak akan dibahas. Cari buku tentang pola desain dengan Python dan berbagai cara membangun algoritme untuk membuat kode

Anda seefisien mungkin. Temukan buku yang membahas secara mendalam tentang aplikasi Anda.

- Jangan berpikir Anda harus belajar Python, atau topik lain apa pun dalam buku ini, sendirian. Ada komunitas besar di luar sana. Temukan pertemuan, klub, dan kelas lokal. Temukan ruang peretas lokal Anda. Saya jamin Anda akan menemukan seseorang di sana yang dapat membantu Anda.

3.1 IKHTISAR PYTHON

Python adalah bahasa pemrograman tingkat tinggi yang dibuat oleh Guido van Rossum pada akhir 1980-an. Ini telah menjadi bahasa tujuan umum yang sangat populer karena fleksibel, dan relatif mudah dipelajari dan dikodekan. Dalam banyak hal, Python adalah bahasa yang sangat pemaaf, yang memberikan kemudahan penggunaannya. Seperti yang akan Anda lihat nanti di bab ini, Python mengelola data dengan cara yang sangat intuitif bagi orang yang baru mengenal pemrograman. Dengan demikian, ini adalah alat yang sangat populer untuk mengajarkan dasar-dasar pemrograman. Cara anehnya menggunakan variabel untuk mengelola kumpulan data besar juga membuatnya sangat populer di bidang ilmu data yang sedang berkembang. Ilmuwan data dapat mengimpor volume data dan melakukan operasi pada kumpulan data dengan kode yang sangat sedikit. Tentu saja, Python memiliki kekhasan yang kami jelajahi lebih dalam saat kami mengerjakan bab ini.

3.2 MENGUNDUH DAN MENGINSTAL PYTHON

Pertama, mari kita bahas sedikit tentang versi. Pada dasarnya ada dua rasa Python: Python 2.7 dan Python 3. Dalam Python 3, pencipta Guido van Rossum memutuskan untuk membersihkan kode tanpa terlalu menekankan kompatibilitas mundur; oleh karena itu, beberapa kode untuk versi 2.7 tidak akan berfungsi di versi 3 dan sebaliknya. Python 3 adalah versi saat ini dan semuanya pada akhirnya akan pindah ke sana. Faktanya, pada titik ini, hampir semuanya memiliki. Dalam hal robotika, ketidaksepakatan besar adalah OpenCV, perpustakaan open source fungsi visi komputer, yang akan kita gunakan di Bab 9. Ada yang lain yang belum sepenuhnya bermigrasi, jadi Anda harus mencari tahu apa yang ingin Anda lakukan dan jika paket yang Anda butuhkan telah di-porting. Kami akan menggunakan Python 2.7 untuk proyek kami karena banyak contoh yang akan Anda temukan dalam penelitian Anda sendiri ada di 2.7.

Jika Anda menggunakan sistem Ubuntu atau Debian Linux, seperti Raspberry Pi, Anda sudah selesai. Alat Python sudah diinstal dan siap digunakan. Sebagian besar distribusi berbasis Debian menginstal Python sebagai bagian dari gambar dasar. Jika Anda mengikuti di Windows atau sistem operasi lain, Anda perlu menginstal Python.

1. Navigasikan ke www.python.org/about/getstarted/.
2. Klik **Download**s.
3. Jika Anda menggunakan Windows, klik **Unduh Python 2.7.13**.
4. Jika Anda menggunakan OS lain, pilih dari daftar di bawah **Mencari Python dengan OS yang berbeda?** Ini membawa Anda ke tautan unduhan yang sesuai.

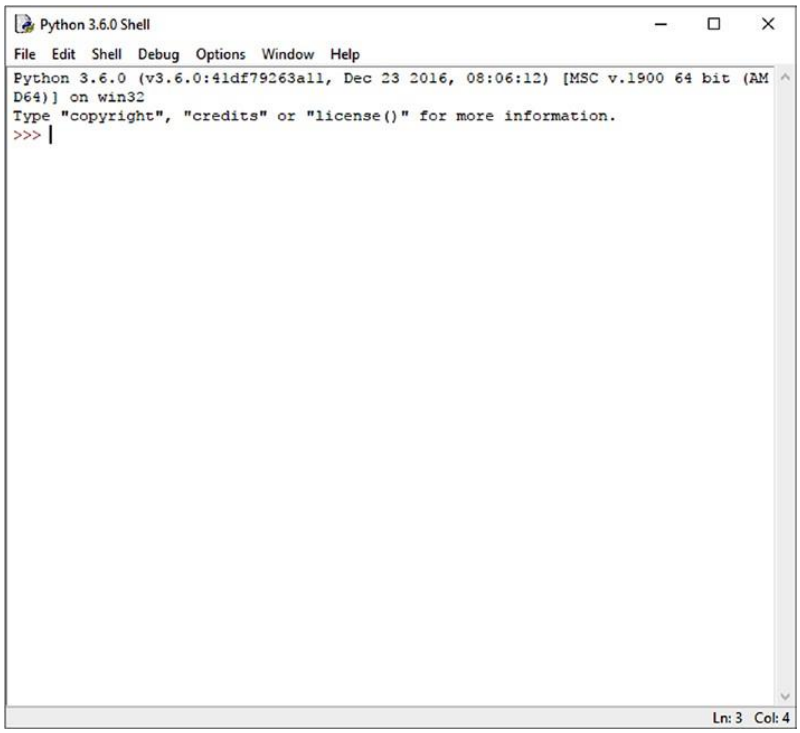
5. Jika Anda ingin menggunakan rilis Python yang lebih lama (untuk beberapa alasan aneh), Anda akan menemukan tautan yang sesuai dengan menggulir halaman ke bawah.
6. Setelah diunduh, jalankan penginstal dan ikuti petunjuk di layar.

3.3 ALAT PYTHON

Ada banyak alat untuk mendukung pengembangan Python Anda. Seperti kebanyakan bahasa pemrograman, kodenya hanyalah teks yang dapat ditulis dengan editor teks apa pun. Anda dapat menulis kode Python dengan Notepad di mesin Windows Anda. Saya tidak akan merekomendasikannya, tetapi Anda bisa melakukannya. Aplikasi seperti Notepad++ mengenali skrip Python berdasarkan ekstensi file dan kemudian menyorot kode yang sesuai. Pilihan Anda cukup luas. Namun, untuk latihan kami, kami akan menggunakan alat asli yang disertakan dengan setiap instalasi Python: shell Python dan editor Python.

3.4 CANGKANG PYTHON

Shell Python adalah antarmuka ke juru bahasa Python. Secara teknis, jika Anda penggemar baris perintah, Anda dapat meluncurkan jendela terminal dan memanggil juru bahasa Python. Namun, kami akan menggunakan antarmuka shell Python yang diinstal dengan Python, seperti yang ditunjukkan pada Gambar 3-1. Ini menyediakan antarmuka yang sangat bersih untuk melihat dan menjalankan perintah.



Gambar 3-1. Cangkang Python IDLE

Shell Python diluncurkan saat Anda membuka IDLE IDE asli. Bergantung pada siapa Anda bertanya, IDLE adalah singkatan dari lingkungan pengembangan terintegrasi atau

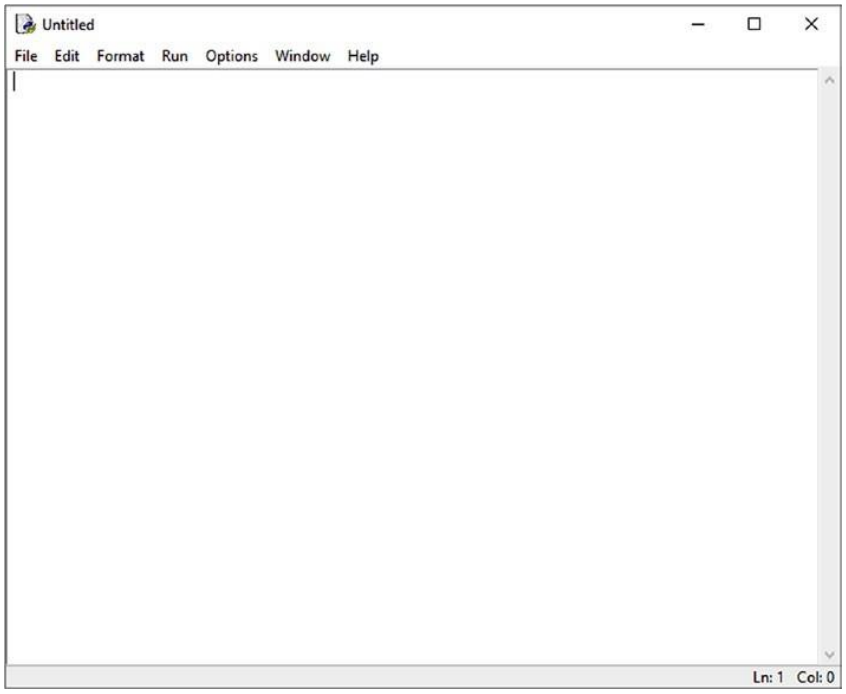
lingkungan pengembangan dan pembelajaran terintegrasi. Saya suka yang terakhir hanya karena itu lebih masuk akal bagi saya. Namun pada intinya, ini adalah antarmuka berjendela ke juru bahasa Python. Ini menawarkan beberapa fitur yang tidak akan Anda dapatkan di baris perintah sederhana.

- Fitur pengeditan sederhana, seperti temukan, salin, dan tempel
- Penyorotan sintaks
- Editor teks dengan penyorotan sintaks
- Sebuah debugger dengan stepping dan breakpoints

Karena kami akan sering menggunakan antarmuka ini di seluruh buku, akan lebih bijaksana untuk mempelajari lebih lanjut tentang antarmuka IDLE dan banyak alat yang disediakannya. Anda dapat memulai di halaman dokumentasi IDLE di https://docs. python.org/3/library/idle.html.

3.5 EDITOR PYTHON

IDLE memiliki aspek lain yang sangat penting: editor teks. Kami akan menggunakannya di seluruh buku ini untuk menulis program dan modul kami. Editor teks adalah aspek lain dari IDLE dan bukan program terpisah, meskipun selalu terbuka di jendela terpisah (lihat Gambar 3-2). Anda dapat menulis program Python di editor teks apa pun, dan ada banyak IDE yang mendukung Python. Seperti yang saya sebutkan di bagian sebelumnya, bagaimanapun, antarmuka IDLE memberikan banyak keuntungan.



Gambar 3-2. Editor file IDLE

Seperti yang akan Anda pelajari nanti, pemformatan sangat penting dalam Python. Dengan bahasa lain, seperti C dan Java, spasi tidak relevan dengan kompiler. Spasi, tab, baris

baru, dan baris kosong membuat kode lebih mudah dibaca orang. Dalam Python, bagaimanapun, lekukan menunjukkan blok kode. IDLE mengelola semua ini untuk Anda. Ini secara otomatis membuat indentasi kode Anda, mengurangi kemungkinan kesalahan sintaks karena indentasi yang tidak tepat. Ada juga beberapa alat untuk membantu Anda dengan kode Anda. Misalnya, saat Anda mengetik, IDLE menyajikan daftar kemungkinan pernyataan yang sesuai dengan posisi Anda dalam satu baris. Ada beberapa cara untuk mengaktifkan fitur ini. Banyak kali, itu secara otomatis terbuka saat Anda mengetik. Ini biasanya terjadi saat Anda berada di dalam panggilan fungsi dan hanya ada beberapa kemungkinan. Anda juga dapat memaksanya terbuka dengan menekan Ctrl-spasi saat mengetik. Ketika Anda melakukan ini, Anda melihat daftar pernyataan yang mungkin untuk dipilih. Ketika Anda memilih salah satu dari pernyataan ini, itu melengkapi kata untuk Anda dan memberi Anda pilihan lain jika tersedia, seperti parameter yang sesuai. Ini disebut calltips.

Calltips menampilkan nilai yang diharapkan untuk fungsi yang dapat diakses dan terbuka saat Anda mengetik "(" setelah nama fungsi. Ini menampilkan tanda tangan fungsi dan baris pertama docstring. Itu tetap terbuka sampai kursor dipindahkan ke luar fungsi atau penutup ")" diketik. Penyorotan konteks dilakukan melalui warna. Saat Anda mengetik kode, beberapa kata berubah warna. Warna memiliki makna dan merupakan cara visual yang cepat untuk memverifikasi bahwa Anda berada di jalur yang benar. Konteks yang disorot dengan cara ini adalah keluaran, kesalahan, keluaran pengguna dan kata kunci Python, kelas bawaan dan nama fungsi, nama mengikuti kelas dan def, string, dan komentar.

Mari kita lihat beberapa dari ini dalam tindakan.

1. Buka IDLE.
2. Klik **File => New File**. Ini akan membuka jendela editor teks baru.
3. Ketik **pr**.
4. Tekan Ctrl-spasi. Ini menampilkan daftar penyelesaian dengan kata cetak disorot.
5. Jenis (.
   Ini melakukan beberapa hal. Ini memilih teks yang disorot. Dalam hal ini cetak. Ini juga menampilkan calltip untuk fungsi cetak.
6. Ketik **"Hello World".**
7. Calltip ditutup setelah Anda mengetik penutupan).
8. Tekan Enter.
9. Simpan file ini sebagai hello_world.py.
10. Tekan F5 atau pilih **Run => Run Module** dari menu.

Di jendela shell Python, Anda akan melihat sesuatu seperti ini:

```
RESTART: D:/Projects/The Robot Group/Workshops/Raspberry Pi
Robot/hello_world.py
Hello World
```

Jika Anda melihat ini, maka kode Anda berhasil. Jika Anda menerima semacam kesalahan, kembali untuk memastikan bahwa kode Anda terlihat seperti ini:

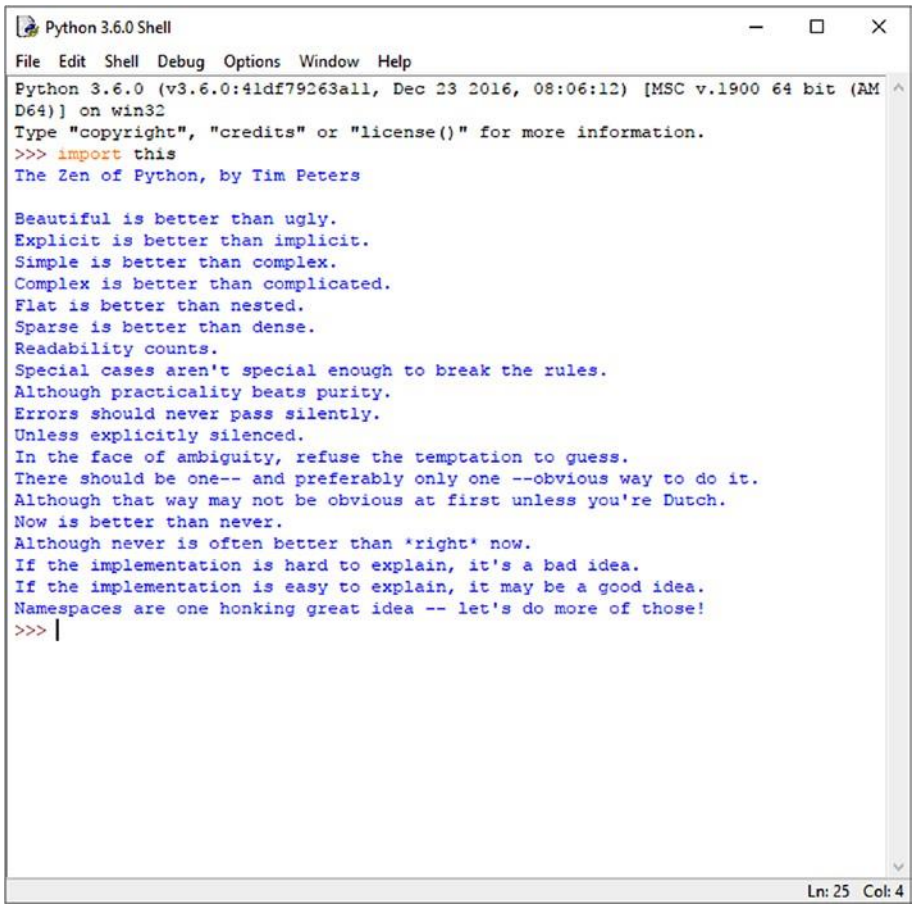
```
print("Hello World")
```

Omong-omong, Anda baru saja menulis dan menjalankan program Python pertama Anda. Jika dibandingkan dengan bahasa lain, Anda akan melihat kesederhanaan Python. Seringkali, beberapa baris dari operasi Python akan mengambil beberapa baris lagi di C, C++, atau Java.

3.6 ZEN DARI PYTHON

Tim Peters, kontributor lama untuk Python, menulis prinsip-prinsip yang mengatur di balik pengembangan Python. Saya pikir itu benar-benar berlaku untuk semua kode dan hampir semua yang kita lakukan dalam robotika, dan mungkin dalam kehidupan. Mereka tersimpan sebagai telur Paskah di Python IDE.

1. Buka IDLE.
2. Ketik **import this** dan tekan Enter.
3. Ini akan menampilkan teks yang ditunjukkan pada Gambar 3-3 (tapi tetap lakukan).



Gambar 3-3. Zen dari Python

3.7 MENULIS DAN MENJALANKAN PROGRAM PYTHON

Jika Anda mengikuti sebagaimana mestinya, maka Anda baru saja menulis dan menjalankan program Python pertama Anda. Jika Anda tidak mengikuti, jangan khawatir, kami akan melakukannya lagi sekarang tetapi dengan sedikit lebih banyak pemrograman.

3.8 HALO DUNIA

Mari tambahkan panggilan variabel sederhana. Kami akan berbicara tentang variabel dalam waktu dekat.

1. Buka `hello_world.py`.
2. Jika Anda salah satu dari pemberontak yang saya sebutkan sebelumnya, buka IDLE.
3. Klik **File => File Baru**.
4. Simpan file ini sebagai hello_world.py. Sekarang kita semua berada di halaman yang sama…
5. Jadikan programnya seperti ini:

```
message = "Hello World"
print(message)
```

6. Simpan filenya.
7. Tekan F5 atau pilih **Run => Run Module** dari menu.

Anda akan melihat output yang sama seperti sebelumnya:

```
RESTART: D:/Projects/The Robot Group/Workshops/Raspberry Pi
Robot/hello_world.py
Hello World
```

Yang kami lakukan di sini hanyalah memindahkan teks dari fungsi print ke dalam variabel, dan kemudian memberi tahu Python untuk mencetak konten variabel itu. Saya akan membahas variabel segera.

3.9 STRUKTUR DASAR

Sebelum kita mulai melihat spesifikasi program, kita harus terbiasa dengan struktur program Python. Kita akan melihat bagian yang berbeda dari sebuah program, bagaimana program diformat menggunakan lekukan, dan bagaimana Anda akan menambahkan konteks yang bermakna menggunakan komentar.

Bagian Program

Seperti yang Anda lihat, tidak banyak bagian yang diperlukan untuk program Python. Sebagian besar bahasa pemrograman mengharuskan Anda untuk membuat, paling tidak, semacam fungsi utama. Untuk Arduino itu adalah fungsi loop(). Dalam C++, ini adalah main(). Python tidak memiliki itu. Ia langsung menjalankan perintah apa pun yang ditemukannya saat ia menelusuri file. Namun, ini tidak berarti bahwa itu sepenuhnya linier. Saya membahas fungsi dan kelas nanti di lokakarya, tetapi ketahuilah bahwa penerjemah memindai file dan membangun fungsi dan kelas apa pun yang ditemukannya sebelum menjalankan perintah lainnya. Ini adalah salah satu hal yang membuat Python sangat mudah dipelajari. Itu tidak memiliki kerangka kerja yang kaku seperti yang Anda temukan di sebagian besar bahasa lain.

Oh, dan bagi Anda para ahli bahasa pemrograman, Python berjalan di antara bahasa skrip, di mana semuanya dieksekusi melalui juru bahasa, dan bahasa pemrograman. Beberapa kode dikompilasi menjadi executable seperti C dan C++. Faktanya, saat kita mulai membangun

modul, inilah yang terjadi. Namun, dalam buku ini, kami biasanya menjalankannya melalui juru bahasa.

Lekukan

Saat kami bekerja melalui lokakarya, program kami menjadi lebih kompleks. Secara khusus, kita akan mulai bekerja dengan blok kode, yang merupakan perintah yang dikelompokkan bersama untuk dieksekusi dalam suatu fungsi atau loop, atau sebagai bagian dari suatu kondisi. Struktur semacam ini sangat penting untuk menulis program yang efektif. Semua bahasa pemrograman memiliki sintaks untuk memformat blok kode. Bahasa berbasis C, termasuk Java, menggunakan tanda kurung kurawal {} untuk memuat blok kode. Python tidak melakukan ini. Python menggunakan lekukan. Blok kode diindentasi untuk menunjukkan bahwa mereka adalah blok terkait. Jika garis dalam blok tidak menjorok dengan benar, Anda akan mendapatkan kesalahan. Ini adalah salah satu alasan utama kami menggunakan IDLE. Ini secara otomatis mengelola lekukan. Itu tidak berarti bahwa Anda, sebagai pengguna, tidak dapat merusak program; itu hanya berarti jenis kesalahan ini sangat berkurang.

Saat kami maju melalui lokakarya, Anda akan melihat pentingnya lekukan. Sementara itu, ketahuilah bahwa itu penting. Terakhir, saya ingin membuat catatan singkat tentang indentasi dan editor. Editor teks menggunakan berbagai bentuk lekukan. Beberapa menggunakan karakter tab sementara yang lain menggunakan dua atau empat spasi. Anda tidak dapat melihat apa yang digunakan karena ini adalah karakter yang tidak terlihat. Hal ini menyebabkan frustrasi yang tiada habisnya ketika Anda berpindah dari satu editor ke editor lainnya. Editor yang lebih baik memungkinkan Anda memilih cara kerja tombol tab (dengan karakter tab atau dengan sejumlah spasi, biasanya empat). Secara default, IDLE menggunakan empat spasi.

Komentar

Mengomentari kode menjadi lebih penting dari waktu ke waktu. Ini adalah area di mana programmer terkenal kekurangannya. Mereka menggunakan komentar, tetapi mereka sering samar atau membuat asumsi tentang pengetahuan tentang program yang mungkin tidak berlaku untuk seseorang yang mengambil kode nanti. Ini tidak akan menjadi masalah jika mereka lebih baik tentang bentuk dokumentasi lainnya. Tapi, sayangnya, mereka tidak. Selain ratapan saya, berkomentar itu penting, terutama saat Anda sedang belajar. Jadi, biasakan menggunakan komentar. Gunakan mereka untuk menjelaskan apa yang Anda lakukan atau untuk memasukkan catatan kecil tentang logika.

Sebuah komentar di Python adalah setiap baris yang dimulai dengan #. Python mengabaikan semua yang mengikuti # hingga akhir baris; Misalnya:

```
# create a variable to hold the text
message = "Hello World"

# print the text stored in the variable
print(message)
```

Dalam kode sebelumnya, saya menambahkan dua baris komentar ke program hello_ world.py kami. Saya juga menambahkan baris kosong untuk membantu membuat kode sedikit lebih mudah dibaca. Jika Anda menyimpan dan menjalankan program ini, Anda akan mendapatkan output yang sama persis seperti yang Anda lakukan sebelumnya.

Anda juga dapat membuat blok komentar menggunakan tanda kutip tiga, """. Kompiler Python mengabaikan kode apa pun di antara kumpulan tanda ini. Anda membuka dan menutup blok dengan mereka. Ini memungkinkan Anda untuk menulis informasi sebanyak yang Anda inginkan dalam beberapa baris. Notasi ini sering digunakan untuk blok judul di awal file.

```
"""
Hello World
the simplest of all programs
By: Everyone who has written a program, ever
"""
```

Merupakan kebiasaan yang baik untuk hanya menguraikan kode Anda dengan komentar sebelum Anda menulisnya. Sebelum Anda mulai menulis, pikirkan tentang apa yang Anda perlukan untuk dilakukan oleh kode Anda dan bagaimana Anda akan melakukannya. Buat diagram alur atau cukup tuliskan langkah-langkah yang akan Anda ambil untuk mencapai tujuan Anda. Kemudian terjemahkan ini ke dalam serangkaian komentar di file Anda sebelum Anda menulis kode yang sebenarnya. Ini membantu Anda menyusun masalah dalam pikiran Anda dan meningkatkan aliran Anda secara keseluruhan. Jika Anda mengidentifikasi langkah yang Anda ulangi, kemungkinan Anda memiliki kandidat untuk suatu fungsi. Jika Anda menemukan bahwa Anda mereferensikan konsep terstruktur (seperti robot) yang ingin Anda tambahkan dengan properti dan fungsionalitas khusus, Anda memiliki kelas. Kemudian, saya membahas fungsi dan kelas secara lebih rinci.

3.10 MENJALANKAN PROGRAM

Seperti yang Anda lihat sebelumnya, ada beberapa cara untuk menjalankan program Python.

Dari IDLE, Anda cukup menekan F5. Anda harus memastikan bahwa file tersebut disimpan terlebih dahulu, tetapi ini menjalankan file Anda. Jika file tidak disimpan, Anda akan diminta untuk melakukannya. Ini sama seperti memilih Run Run Module dari menu bar. Jika sistem Anda dikonfigurasi dengan benar untuk menjalankan Python dari baris perintah, Anda juga dapat menjalankan program dari sana. Anda harus menavigasi ke lokasi file atau memiliki lokasi file lengkap dalam panggilan. Untuk mengeksekusi skrip Python dari baris perintah, Anda mengetik python diikuti dengan file yang akan dijalankan.

```
> python hello_world.py
> python c:\exercises\hello_world.py
> python exercises\hello_world.py
```

Ketiga perintah ini akan menjalankan program Hello World kami, meskipun dua di antaranya khusus untuk sistem operasi. Perintah pertama mengasumsikan bahwa Anda menjalankan file dari dalam direktori penyimpanannya. Perintah kedua menjalankan program di Windows, dengan asumsi itu disimpan dalam folder bernama latihan di root drive C:\. Perintah ketiga menjalankan program pada mesin Linux, dengan asumsi file disimpan dalam file bernama latihan di direktori home Anda.

3.11 PEMROGRAMAN DENGAN PYTHON

Di beberapa bagian berikutnya, kita menggunakan shell Python dan memasukkan perintah secara langsung. Beberapa saat kemudian, kita kembali menulis file program. Namun, untuk saat ini, semua yang kami lakukan dapat didemonstrasikan di jendela shell.

3.8 VARIABEL

Variabel pada dasarnya adalah wadah yang nyaman untuk menyimpan informasi. Dalam Python, variabel sangat fleksibel. Anda tidak perlu mendeklarasikan tipe. Jenis umumnya ditentukan ketika Anda menetapkan nilai untuk itu. Sebenarnya, Anda mendeklarasikan variabel dengan memberikan nilai padanya. Ini penting untuk diingat ketika Anda mulai dengan angka. Dalam Python, ada perbedaan antara 1 dan 1.0. Angka pertama adalah bilangan bulat dan angka kedua adalah float. Lebih lanjut tentang itu segera.

Berikut adalah beberapa aturan umum untuk variabel:

- Hanya boleh berisi huruf, angka, dan garis bawah
- Mereka peka huruf besar/kecil; misalnya, variabel tidak sama dengan Variabel. Itu akan menggigitmu nanti.
- Jangan gunakan kata kunci Python

Selain aturan keras dan cepat ini, berikut adalah beberapa tip:

- Buatlah nama variabel bermakna dengan karakter sesedikit mungkin
- Berhati-hatilah saat menggunakan huruf kecil L dan huruf besar O. Karakter ini terlihat sangat mirip dengan 1 dan 0, yang dapat menyebabkan kebingungan. Saya tidak mengatakan jangan menggunakannya; pastikan saja apa yang Anda lakukan jelas. Sangat tidak disarankan untuk menggunakannya sebagai nama variabel satu karakter.

3.9 TIPE DATA

Python adalah bahasa yang diketik secara dinamis. Ini berarti bahwa tipe data yang disimpan dalam variabel tidak diperiksa sampai program dijalankan, bukan saat sedang dikompilasi. Ini memungkinkan Anda untuk menunda menetapkan jenis sampai nilai ditetapkan. Namun, Python juga diketik dengan kuat, dan akan gagal jika Anda mencoba melakukan operasi yang tidak valid untuk tipe data tersebut. Misalnya, Anda tidak dapat melakukan operasi matematika pada variabel yang berisi string. Karena itu, penting untuk melacak jenis data yang menjadi referensi variabel Anda.

Saya akan berbicara tentang dua tipe data dasar: string dan angka. Kemudian saya akan membahas beberapa jenis yang lebih kompleks: tupel, daftar, dan kamus. Python memungkinkan Anda untuk mendefinisikan tipe Anda sendiri sebagai kelas. Saya akan membahas kelas menjelang akhir bab ini, karena ada beberapa konsep lain yang perlu kita bahas terlebih dahulu.

String

String adalah kumpulan dari satu atau lebih karakter yang terkandung dalam tanda kutip. Kutipan adalah cara Anda menunjukkan string. Misalnya, "100" adalah string; 100 adalah angka (lebih tepatnya bilangan bulat). Anda dapat menggunakan tanda kutip ganda atau tunggal (ingat saja apa yang Anda gunakan). Anda dapat menumpuk satu jenis kutipan di dalam yang lain, tetapi Anda akan mendapatkan kesalahan, atau lebih buruk lagi, hasil yang tidak terduga, jika Anda melewati tanda kutip Anda.

Berikut ini contoh kutipan ganda:

```
>>>print("This is text")
This is text
```

Berikut ini contoh kutipan tunggal:

```
>>>print('This is text')
This is text
```

Berikut adalah contoh tanda kutip tunggal di dalam tanda kutip ganda:

```
>>>print("'This is text'")
'This is text'
```

Berikut adalah contoh tanda kutip ganda di dalam tanda kutip tunggal:

```
>>>print('"This is text"')
"This is text"
```

Tanda kutip tiga digunakan untuk merentangkan beberapa baris dalam sebuah string. Berikut ini contoh kutipan rangkap tiga:

```
>>>print("""this
is
text""")
this
is
text
```

Anda dapat menggunakan tanda kutip tunggal sebagai apostrof jika Anda menghindarinya terlebih dahulu. Melarikan diri dari karakter berarti Anda memberi tahu penerjemah untuk melihat karakter sebagai karakter string daripada karakter fungsional.

Berikut adalah contoh dari kutipan melarikan diri:

```
>>>print('This won't work')
File "<stdin>", line 1
       print('this won't work')
                      ^
SyntaxError: invalid syntax
>>>print('This won\'t error')
This won't error
```

Anda dapat melakukan ini ke seluruh string dengan menjadikannya string mentah. Dalam contoh berikut ini, '/n' digunakan untuk pindah ke baris baru.
Berikut ini contoh string mentah:

```
>>>print('something\new')
Something\new
>>>print(r'something\new')
something/new
```

Manipulasi String

Ada banyak cara untuk memanipulasi string. Beberapa cukup mudah, seperti penggabungan—menambahkan string bersama. Namun, beberapa di antaranya sedikit mengejutkan. String diperlakukan sebagai daftar nilai karakter. Kemudian dalam bab ini, kita menjelajahi daftar secara lebih rinci. Namun, kita akan menggunakan beberapa ciri daftar untuk bekerja dengan string.

Karena string adalah daftar, yang mirip dengan array dalam bahasa lain, kita dapat mereferensikan karakter tertentu di dalam string. Seperti daftar, string diindeks nol. Ini berarti karakter pertama dari sebuah string berada pada posisi nol.

Dengan string, seperti daftar, karakter pertama ada di indeks [0]:

```
>>>robot = 'nomad'
>>>robot[0]n
```

Saat menggunakan angka negatif, indeks dimulai di akhir string dan bekerja mundur.

```
>>>robot[-1]t
```

Mengiris string memungkinkan Anda mengekstrak substring. Saat mengiris string, Anda memberikan dua angka yang dipisahkan oleh titik dua. Angka pertama adalah indeks awal dan yang kedua adalah indeks akhir.

```
>>>robot[0:3]nom
```

Perhatikan bahwa saat mengiris, indeks pertama inklusif dan indeks kedua eksklusif. Pada contoh sebelumnya, nilai pada indeks [0] dikembalikan, “r”; sedangkan nilai pada indeks [3] bukan, “o”.

Saat mengiris, jika Anda meninggalkan salah satu indeks, awal atau akhir string diasumsikan.

```
>>>robot[:3]nom
>>>robot[3:]ad
```

Menambahkan string bersama disebut concatenation. Dengan Python, Anda dapat dengan mudah menambahkan string bersama. Ini bekerja dengan literal string dan variabel string. Anda juga dapat mengalikan string untuk efek yang menarik. Anda dapat menambahkan dua string bersama-sama.

```
>>>print("Ro" + "bot")Robot
```

Anda dapat menambahkan variabel string bersama-sama, seperti ini:

```
>>>x = "Ro"
>>>y = "bot"
>>>z = x + y
>>>print(z)Robot
```

Anda dapat menambahkan variabel string dan literal.

```
>>>print(x + "bot")Robot
```

Anda dapat mengalikan string literal.

```
>>>print(2 * "ro" + "bot")rorobot
```

Perkalian string, bagaimanapun, hanya bekerja pada literal. Ini tidak akan bekerja pada variabel string.

Saya sarankan meluangkan waktu untuk menjelajahi ini dan metode manipulasi string lainnya. Untuk informasi lebih lanjut, kunjungi

https://docs. python.org/3/tutorial/introduction.html#strings

https://docs.python.org/3.1/library/stdtypes.html#string-methods.

Angka

Angka dalam Python datang dalam beberapa rasa, yang paling umum adalah bilangan bulat dan float. Integer adalah bilangan bulat, sedangkan float adalah desimal. Python juga menggunakan tipe Boolean yang memiliki nilai satu atau nol. Ini sering digunakan sebagai bendera atau negara bagian, di mana satu berarti "aktif" dan nol berarti "mati". Boolean adalah subkelas bilangan bulat dan diperlakukan sebagai bilangan bulat saat melakukan operasi. Seperti yang Anda harapkan, Anda dapat melakukan operasi matematika dengan tipe angka. Umumnya, jika Anda melakukan aritmatika dengan satu jenis, hasilnya adalah jenis itu. Matematika menggunakan bilangan bulat biasanya menghasilkan bilangan bulat. Namun, jika Anda melakukan pembagian dengan bilangan bulat, hasilnya adalah float. Matematika dengan

pelampung menghasilkan pelampung. Jika Anda melakukan aritmatika dengan kedua jenis, hasilnya adalah float.

Menambahkan dua bilangan bulat menghasilkan bilangan bulat.

```
>>>2+3
5
```

Menambahkan dua pelampung menghasilkan pelampung.

```
>>>0.2+0.3
0.5
```

Menambahkan float dan integer menghasilkan float.

```
>>>1+0.5
1.5
```

Pengurangan dan perkalian bekerja dengan cara yang sama.

```
>>>3-2
1
>>>3-1.5
1.5
>>>2*3
6
>>>2*0.8
1.6
```

Pembagian selalu menghasilkan float.

```
>>>3.2/21.6
>>>3/21.5
```

Operator ** menghasilkan angka pertama yang dipangkatkan ke yang kedua.

```
>>>3**2
9
```

Namun, ada satu tangkapan dengan float Python. Penerjemah terkadang menghasilkan jumlah tempat desimal yang tampaknya berubah-ubah. Ini ada hubungannya dengan bagaimana notasi floating-point disimpan dalam Python dan bagaimana matematika dilakukan di dalam interpreter.

```
>>>0.2+0.1
0.30000000000000004
```

Untuk informasi lebih lanjut tentang anomali ini, kunjungi https://docs.python.org/3/tutorial/floatingpoint.html.

Daftar

Daftar adalah kumpulan item dalam urutan tertentu. Dalam bahasa lain, mereka umumnya dikenal sebagai array. Anda dapat memasukkan apa pun yang Anda inginkan ke dalam daftar. Nilai yang disimpan dalam daftar tidak harus memiliki tipe data yang sama. Namun, jika Anda mencampur tipe data dalam daftar, pastikan Anda tahu tipe mana yang Anda dapatkan saat menggunakannya.

Anda bekerja dengan daftar ketika Anda bekerja dengan string. String pada dasarnya adalah daftar karakter. Dengan demikian, pengindeksan dan pengirisan juga berfungsi dengan daftar.

Daftar dibuat menggunakan tanda kurung siku [].

```
>>> robots = ["nomad","Ponginator","Alfred"]
>>> robots
['nomad', 'Ponginator', 'Alfred']
```

Seperti string, daftar diindeks nol. Ini berarti elemen pertama dalam daftar berada di posisi 0, yang kedua di posisi 1, dan seterusnya. Anda dapat mengakses elemen individual dari daftar dengan memanggil indeks atau lokasinya di dalam daftar.

```
>>>robots[0]
'nomad'
>>>robots[-1]
'Alfred'
```

Daftar juga dapat diiris. Ketika daftar diiris, hasilnya adalah daftar baru yang berisi subset dari daftar asli.

```
>>>robots[1:3]
['Ponginator','Alfred']
```

Sangat mudah untuk menambah, mengubah, dan menghapus anggota daftar menggunakan pemotongan dan penggabungan.

Contoh ini menambahkan anggota ke daftar.

```
>>>more_bots = robots+['Roomba','Neato','InMoov']
>>>more_bots
['nomad', 'Ponginator', 'Alfred', 'Roomba', 'Neato', 'InMoov']
```

Contoh ini mengubah anggota dalam daftar.

```
>>>more_bots[3] = 'ASIMO'
>>>more_bots
['nomad', 'Ponginator', 'Alfred', 'ASIMO', 'Neato', 'InMoov']
```

Contoh ini menghapus anggota daftar.

```
>>>more_bots[3:5] = []
>>>more_bots
['nomad', 'Ponginator', 'Alfred', 'InMoov']
```

Tetapkan anggota daftar ke variabel:

```
>>>a,b = more_bots[0:2]
>>>a
'nomad'
>>>b
'Ponginator'
```

Ada sejumlah metode yang secara otomatis disertakan dalam daftar. Misalnya, Anda dapat memaksa huruf pertama nama menjadi huruf besar.

```
>>> print(robots[0].title())Nomad
```

Seperti yang saya sebutkan, daftar dapat berisi semua jenis data, termasuk daftar lainnya. Bahkan, ketika kita mulai bekerja dengan visi komputer, kita akan sering menggunakan daftar daftar untuk menyimpan data gambar.

Daftar adalah aspek yang sangat kuat dan penting dari Python. Kunjungi https://docs.python.org/3/tutorial/introduction.html#lists untuk meluangkan waktu menjelajahi daftar.

Tuple

Anda akan sering mendengar istilah tuple saat bekerja dengan Python. Tuple hanyalah jenis daftar khusus yang tidak dapat diubah. Pikirkan tupel sebagai daftar konstanta, atau daftar konstanta. Anda mendeklarasikan Tuple menggunakan tanda kurung daripada tanda kurung siku. Tuple tidak dapat diubah, yang berarti bahwa sekali tuple telah dibuat, tupel tidak dapat diubah. Untuk mengubah isi sebuah tuple, sebuah tuple baru harus dibuat. Ini dilakukan dengan menggunakan teknik slicing yang sama yang kami gunakan untuk string dan list.

```
>>> colors = ("red","yellow","blue")
>>> colors
('red', 'yellow', 'blue')
>>>colors2 = colors[0:2]
>>>colors2 ('red','yellow')
```

Perhatikan bahwa kami menggunakan notasi daftar saat mengiris tupel, `colors[0:2]` bukan `colors(0:2)`. Hasil irisan masih berupa tuple.

Tuple, bagaimanapun, dapat diganti.

```
>>>colors2 = (1,2,3)
>>>colors2(1,2,3)
```

Mereka juga dapat diganti dengan tupel kosong.

```
>>>colors2 = ()
>>>colors2
()
```

Kamus

Kamus mirip dengan daftar kecuali memungkinkan Anda memberi nama item Anda dalam daftar. Ini dilakukan dengan menggunakan pasangan kunci/nilai. Kunci menjadi indeks untuk nilai. Ini memungkinkan Anda untuk menambahkan beberapa struktur yang berarti ke daftar Anda. Mereka berguna untuk menyimpan daftar parameter atau properti.
Kamus dideklarasikan menggunakan kurung kurawal daripada kurung siku.

```
>>> Nomad = {'type':'rover','color':'black','processor':'JetsonTX1'}
>>> print(Nomad['type'])Rover
```

Anda bekerja dengan kamus dengan cara yang sama seperti Anda menggunakan array. Kecuali daripada memberikan nomor indeks, Anda memberikan kunci untuk mengakses elemen.
Ada beberapa hal yang perlu diketahui tentang kamus sebelum Anda menggunakannya.

- Kunci harus berupa nilai yang tidak dapat diubah, seperti angka atau string. Tuple juga dapat digunakan sebagai kunci.
- Sebuah kunci tidak dapat didefinisikan lebih dari satu kali dalam kamus. Seperti variabel, nilai kunci adalah yang terakhir ditetapkan.

```
>>>BARB = {'type':'test-bed','color':'black','type':'wheeled'}
>>>BARB
{'color':'black','type':'wheeled'}
```

- Dalam contoh ini, nilai 'tipe' pertama ditimpa oleh yang kedua.
- Kamus dapat disarangkan sebagai nilai dalam kamus lain. Dalam contoh berikut, saya telah menyematkan definisi untuk proyek robotika saya yang sedang berjalan, Nomad ke dalam kamus robots.txt saya.

```
>>>myRobots = {'BARB':'WIP','Nomad':Nomad,'Llamabot':'WIP'}
>>>myRobots
{'BARB':{'color':'black','type':'wheeled'},'Nomad':{'color':'
black','type':'wheeled'},'Llamabot':'WIP'}
```

Tentu saja, kamus tidak akan terlalu berguna jika Anda tidak dapat memperbarui dan memanipulasi nilai yang terkandung di dalamnya. Membuat perubahan pada kamus mirip dengan membuat perubahan pada daftar. Satu-satunya perbedaan nyata adalah Anda menggunakan kunci daripada posisi untuk mengakses berbagai elemen.

Untuk memperbarui nilai, gunakan kunci untuk mereferensikan nilai yang akan diubah.

```
>>>myRobots['Llamabot'] = 'Getting to it'
>>>myRobots
```

```
{'BARB':{'color':'black','type':'wheeled'},'Nomad':{'color':'
black','type':'wheeled'},'Llamabot':'Getting to it'}
```

Pasangan kunci/nilai dapat dihapus dengan pernyataan del.

```
>>>del myRobots['Llamabot']
>>>myRobots
{'BARB':{'color':'black','type':'wheeled'},'Nomad':{'color':'
black','type':'wheeled'}}
```

Kamus dapat disalin dengan metode salin dari kelas kamus. Untuk mengakses metode salin, mulailah dengan nama kamus dan tambahkan .copy() di akhir.

```
>>>workingRobots = myRobots.copy()
>>>workingRobots
{'BARB':{'color':'black','type':'wheeled'},'Nomad':{'color':'
black','type':'wheeled'}}
```

Untuk menambahkan satu kamus ke akhir kamus lainnya, gunakan metode update.

```
>>>otherRobots = {'Rasbot-pi':'Pi-bot frombook','spiderbot':'broken'}
>>>myRobots.update(otherRobots)
>>>myRobots
{'BARB': {'color':'black','type':'wheeled'},'Nomad':
{'color':'black','type':'wheeled'},'Rasbot-pi':'Pi-bot
frombook','spiderbot':'broken'}
```

Tidak Ada Jenis

Ada tipe data khusus yang sangat penting ketika bekerja dengan kelas dan objek yang diimpor dari sumber lain. Ini adalah tipe none, yang merupakan placeholder kosong. Ini digunakan ketika kita ingin mendeklarasikan suatu objek tetapi mendefinisikannya nanti. Ini juga digunakan untuk mengosongkan objek. Anda akan melihat tipe none beraksi nanti di bab ini, saat kita membahas kelas. Sementara itu, ketahuilah bahwa itu ada dan pada dasarnya adalah tempat penampung kosong.

3.14 CATATAN AKHIR TENTANG VARIABEL

Saat Anda mengerjakan contoh di bagian ini, Anda bekerja dengan variabel. Perhatikan bagaimana sebuah variabel menerima nilai apa pun yang Anda berikan dan dengan senang hati mengembalikan apa yang Anda tetapkan. Jika Anda menetapkan daftar ke variabel, itu mengembalikan daftar—tanda kurung siku dan semuanya. Hal yang sama berlaku untuk tupel, kamus, string, dan angka. Apa pun yang Anda tetapkan untuk itu adalah persis apa yang Anda dapatkan kembali. Kami melihat ini beraksi ketika kami menumpuk satu kamus di dalam kamus lain. Dengan hanya menambahkan nama kamus ke dalam definisi yang lain, kami menyematkan semua nilai ke dalam kamus baru.

Mengapa Saya Menunjukkan Hal Ini?

Nanti di buku ini, ketika kita mulai bekerja dengan fungsi dan kelas, Anda akan menetapkan struktur data yang kompleks ke variabel Anda. Penting untuk diketahui bahwa apa pun yang Anda tetapkan ke variabel adalah isi variabel tersebut, dan Anda dapat menerapkan metode atau fungsi apa pun yang sesuai untuk tipe data tersebut.

```
>>> robots = ["nomad","Ponginator","Alfred"]
>>> robots
['nomad', 'Ponginator', 'Alfred']
>>> myRobot = robots[0]
>>> myRobot'nomad'
>>> myRobot.capitalize()'
Nomad'
```

Kami menggunakan metode kelas String pada variabel string kami, myRobot. Metode adalah fungsionalitas yang kita berikan ke kelas. Karena tipe data adalah kelas bawaan, kita dapat menggunakan metode dari kelas itu pada variabel kita. Saya akan membahas metode secara lebih rinci ketika kita mulai bekerja dengan kelas menjelang akhir bab ini.

3.15 STRUKTUR KONTROL

Di bagian ini, kita akan menjelajahi cara menambahkan struktur ke kode Anda. Daripada hanya melangkah melalui program dan mengeksekusi setiap baris kode seperti yang ditemui, Anda mungkin menginginkan lebih banyak kontrol. Struktur kontrol ini memungkinkan Anda untuk mengeksekusi kode hanya ketika kondisi tertentu ada, dan untuk melakukan blok kode beberapa kali. Untuk sebagian besar, akan lebih mudah untuk memandu Anda melalui konsep-konsep ini daripada mencoba menggambarkannya.

Pernyataan If

Pernyataan if memungkinkan Anda untuk menguji suatu kondisi sebelum Anda mengeksekusi blok kode. Kondisi dapat berupa nilai atau persamaan apa pun yang bernilai benar atau salah.

Potongan kode berikutnya ini melewati daftar robot dan menentukan apakah robot tersebut Nomad.

```
>>> for robot in robots:
      if robot=="Nomad":
            print("This is Nomad"
      else:
            print(robot + " is not Nomad")
This is Nomad
Ponginator is not Nomad
Alfred is not Nomad
```

Ponginator bukan Nomad Alfred bukan Nomad

Sekali lagi, perhatikan lekukan saat Anda mengetik. IDLE mengindentasi level lain setelah setiap baris yang diakhiri dengan titik dua, yang seharusnya setiap baris yang menunjukkan blok baru, seperti pernyataan loop dan kondisional. Penting juga untuk dicatat bagaimana kami menguji kesetaraan. Satu tanda sama dengan berarti penugasan. Tanda sama dengan ganda memberitahu penerjemah untuk membandingkan kesetaraan.

```
>>> myRobot = "Nomad"
>>> myRobot == "Ponginator"False
>>> myRobot == "Nomad"True
```

Berikut daftar komparatornya:

Setara	==
Tidak sama	!=
Kurang dari	<
Lebih besar dari	>
Kurang dari atau sama dengan	<=
Lebih besar dari atau sama dengan	>=

Anda juga dapat menggunakan dan dan atau untuk menguji beberapa kondisi.

```
>>> for robot in robots:
        if robot == "Ponginator" or robot == "Alfred":
            print("These aren't the droids I'm looking for.")
These aren't the droids I'm looking for.
These aren't the droids I'm looking for.
```

Ini bukan droid yang saya cari. Ini bukan droid yang saya cari. Perbandingan juga sering digunakan untuk menentukan apakah suatu objek ada atau mengandung nilai. Pada dasarnya, suatu kondisi bernilai benar jika tidak bernilai salah, 0, atau tidak sama sekali. Ini sangat berguna saat Anda ingin mengeksekusi sepotong kode hanya jika ada objek, seperti saat Anda menginisialisasi sensor atau koneksi melalui port serial atau jaringan.

Loop

Saat Anda bekerja dengan robotika, ada kalanya Anda ingin mengulang satu blok kode. Apakah akan melakukan serangkaian instruksi pada kumpulan objek, atau untuk mengeksekusi blok kode selama kondisi ada, Anda perlu menggunakan loop. Loop memungkinkan Anda mengulangi blok kode untuk melakukan tugas beberapa kali. Ada dua jenis perulangan: perulangan for dan perulangan while. Masing-masing menyediakan fungsionalitas khusus yang sangat penting untuk menulis program yang efisien.

Untuk lingkaran

A for loop melakukan blok kode untuk setiap elemen dalam daftar. Kumpulan nilai—sebuah tupel, daftar, atau kamus disediakan untuk perulangan for. Itu kemudian beralih melalui daftar dan mengeksekusi kode yang terkandung dalam blok kode. Ketika kehabisan elemen dalam koleksi, loop akan keluar dan baris kode berikutnya di luar blok for dieksekusi. Seperti pernyataan if, Anda meletakkan kode yang ingin Anda jalankan sebagai bagian dari

loop ke dalam blok yang ditandai dengan lekukan. Penting untuk memastikan bahwa Anda memiliki lekukan yang benar, atau Anda akan mendapatkan kesalahan. Saat Anda memasukkan ini ke dalam shell Python, perhatikan apa yang dilakukannya dengan lekukan.

Setelah Anda memasukkan perintah cetak dan menekan Enter, Anda perlu menekan Enter lagi agar shell tahu bahwa Anda sudah selesai.

```
>>> for robot in robots:
print(robot)

Nomad
Ponginator
Alfred
```

Program memasuki daftar robot dan menarik nilai pertama, Nomad, yang kemudian dicetak. Karena ini adalah baris terakhir dalam blok, interpreter kembali ke daftar dan mengekstrak nilai berikutnya. Ini berulang hingga tidak ada lagi nilai dalam daftar. Pada saat ini, program keluar dari loop. Saya cenderung menggunakan kata jamak untuk nama daftar saya. Ini memungkinkan saya untuk menggunakan bentuk tunggal dari nama tersebut untuk merujuk item dalam daftar untuk satu lingkaran. Misalnya, setiap elemen dalam Tuple Robots adalah robot. Jika Anda ingin mengulang elemen dalam kamus, Anda ingin menyediakan dua variabel untuk menyimpan elemen individual. Anda juga perlu menggunakan metode item dari kelas kamus. Ini memungkinkan Anda untuk mengakses setiap pasangan kunci/nilai secara bergantian.

```
>>>for name,data in Nomad.items():
print(name + ': ' + data)
color: black
type: wheeled
```

Anda dapat menggunakan fungsi enumerate untuk menambahkan nilai numerik berurutan ke output dari for loop.

```
>>>for num,robot in enumerate(robots):print(num,robot)
(0, 'Nomad')
(1, 'Ponginator')
(2, 'Alfred')
```

Loop Sementara

Sedangkan loop for mengeksekusi blok kode untuk setiap elemen dalam daftar, loop while mengeksekusi blok kode selama kondisinya bernilai true. Ini sering digunakan untuk mengeksekusi kode beberapa kali atau saat sistem dalam keadaan tertentu. Untuk mengulang kode beberapa kali, Anda menggunakan variabel untuk menyimpan nilai integer. Dalam contoh berikut, kami memberi tahu program untuk menjalankan kode selama nilai variabel count kami kurang dari lima.

```
>>> count = 1
>>> while count < 5:
        print(count) count =
        count+1
1
2
3
4
```

Kita mulai dengan mendeklarasikan sebuah variabel, menghitung, untuk menahan bilangan bulat kita, dan kita menetapkan nilai 1. Kita memasuki loop dan nilai 1 kurang dari 5, sehingga kode mencetak nilai ke konsol. Kami kemudian menambah nilai count dengan 1. Karena ini adalah pernyataan terakhir dalam loop, dan nilai terakhir yang dievaluasi oleh kondisi while kurang dari 5, kode kembali ke klausa while. Nilai count, sekarang 2, masih kurang dari 5, sehingga kode dieksekusi lagi. Proses ini berulang sampai nilai count adalah 5. Lima tidak kurang dari lima, sehingga interpreter keluar dari loop tanpa mengeksekusi kode di blok. Jika kita lupa menambah variabel count, itu akan menghasilkan loop terbuka. Karena count sama dengan 1, dan kita tidak pernah menaikkannya, nilai count selalu sama dengan 1. Satu kurang dari lima, jadi kode tidak akan pernah berhenti dieksekusi dan kita perlu menekan Ctrl-C untuk mengakhirinya. Ini juga digunakan sebagai jenis loop utama, mengeksekusi kode secara terus menerus.

Anda melihat yang berikut ini di banyak program:

```
while(true):
```

Benar selalu bernilai benar; oleh karena itu, kode di dalam blok ini terus dijalankan hingga program ditutup. Ini disebut loop terbuka karena tidak ada yang dekat dengannya. Untungnya, ada cara mudah untuk keluar dari loop terbuka. Jika Anda berada dalam loop terbuka, atau Anda hanya ingin keluar dari program secara sewenang-wenang, tekan Ctrl-C. Ini menyebabkan program segera keluar. Anda akan sering menggunakan ini. Teknik ini juga dapat digunakan untuk membuat program menunggu kondisi tertentu terpenuhi. Misalnya, jika kita memerlukan koneksi serial agar tersedia sebelum melanjutkan, pertama-tama kita akan memulai perintah koneksi. Kemudian kita bisa menunggu koneksi selesai sebelum melanjutkan dengan menggunakan beberapa seperti ini:

```
while(!connected):
    pass
```

Tanda seru, juga disebut bang, melambangkan tidak. Jadi, dalam kasus ini, dengan asumsi variabel terhubung berisi koneksi serial yang bernilai benar ketika dibuat, kami memberi tahu program untuk mengeksekusi kode yang terdapat dalam blok while selama tidak terhubung. Dalam hal ini, kode yang kami perintahkan untuk dieksekusi disebut pass,

yang merupakan perintah kosong. Ini digunakan ketika Anda tidak benar-benar ingin melakukan apa pun, tetapi Anda membutuhkan sesuatu di sana. Jadi, kami memberi tahu sistem ini: "Saat Anda tidak terhubung, jangan lakukan apa pun, dan ulangi sampai Anda terhubung."

3.16 FUNGSI

Fungsi adalah blok kode yang telah ditentukan sebelumnya yang dapat kita panggil dari dalam program untuk melakukan tugas. Kami telah menggunakan fungsi print() sepanjang bab ini. Perintah print() adalah fungsi bawaan dalam Python. Ada banyak fungsi yang telah ditentukan dalam Python dan lebih banyak lagi yang dapat ditambahkan menggunakan modul. Untuk informasi lebih lanjut tentang fungsi yang tersedia, lihat Pustaka Standar Python di https://docs.python.org/3/library/index.html. Akan ada banyak kali Anda ingin membuat fungsi Anda sendiri.

Fungsi melayani beberapa tujuan.

Paling sering, Anda menggunakan fungsi untuk memuat kode yang ingin Anda jalankan di seluruh program Anda. Setiap kali Anda mendapati diri Anda mengulangi rangkaian operasi yang sama di seluruh kode Anda, kemungkinan Anda memiliki kandidat untuk suatu fungsi. Fungsi juga banyak digunakan sebagai bentuk housekeeping. Mereka dapat digunakan untuk memindahkan proses panjang ke tempat lain selain program utama Anda. Ini dapat membuat kode Anda lebih mudah dibaca. Misalnya, Anda dapat menentukan tindakan untuk robot Anda sebagai fungsi. Ketika suatu kondisi terpenuhi dalam kode utama Anda, Anda cukup memanggil fungsi itu. Bandingkan dua blok kode semu ini:

```
while(true):
      if command==turnLeft:
            /*
            Lengthy list of instructions to turn left
            */
        if command==turnRight:
            /*
              Lengthy list of instructions to turn right
            */
      /* etc. */
```

Dan

```
 while(true):
        if command==turnLeft:
              turnLeft()
        if command==turnRight:
              turnRight()
/* etc. */
```

Pada blok pertama, kode untuk menggerakkan robot terdapat pada pernyataan if. Jika dibutuhkan 30 baris kode untuk berbelok ke kiri (tidak mungkin, tetapi bersabarlah), kode utama Anda akan lebih panjang 30 baris. Jika berbelok ke kanan membutuhkan jumlah kode yang sama, Anda akan memiliki 30 baris lagi semuanya harus Anda lalui untuk menemukan baris yang Anda cari. Ini menjadi sangat membosankan. Di blok kedua, kode untuk berbelok dipindahkan ke fungsi yang terpisah. Fungsi ini didefinisikan di tempat lain dalam program, atau seperti yang akan Anda pelajari saat kita membahas pustaka dan modul, fungsi ini bisa berada di file lain. Ini membuatnya lebih mudah untuk menulis dan membaca.

Mendefinisikan Fungsi

Untuk menentukan fungsi Anda sendiri, Anda membuat nama fungsi dan blok kode yang berisi operasi yang ingin Anda lakukan. Definisi dimulai dengan kata kunci def, diikuti dengan nama fungsi, tanda kurung, dan titik dua.

Mari kita buat fungsi sederhana:

```
>>> def hello_world():
      message = "Hello World"
      print(message)
>>> hello_world()Hello World
```

Dalam kode ini, kami membuat fungsi sederhana yang hanya mencetak pesan "Hello World". Sekarang, kapan pun kita ingin mencetak pesan itu, kita cukup memanggil fungsi itu.

```
>>> hello_world()
Hello World
```

Untuk membuat hal-hal sedikit lebih menarik, kami dapat menyediakan fungsi dengan data untuk digunakan. Ini disebut argumen.

Melewati Argumen

Seringkali, kami ingin memberikan informasi ke fungsi yang akan dikerjakan atau dijalankan. Untuk menyediakan informasi, kami memberikan fungsi satu atau lebih variabel untuk menyimpan informasi ini, yang disebut argumen.

Mari kita buat fungsi baru yang menyapa pengguna.

```
>>> def hello_user(first_name, last_name):
          print("Hello " + first_name + " " + last_name + "!")
>>> hello_user("Jeff","Cicolani")
Hello Jeff Cicolani!
```

Di sini kami membuat fungsi baru yang disebut hello_user. Kami memintanya untuk mengharapkan menerima dua informasi: nama depan dan nama belakang pengguna. Nama variabel dalam definisi fungsi berisi data yang ingin kita gunakan. Fungsi hanya mencetak salam menggunakan dua argumen yang kami sediakan.

Nilai Dasar

Anda dapat membuat nilai default untuk argumen hanya dengan menetapkan nilai saat Anda mendeklarasikan fungsi.

```
>>> def favorite_thing(favorite = "robotics"):
      print("My favorite thing in the world is "+ favorite)
>>> favorite_thing("pie")
My favorite thing in the world is pie
>>> favorite_thing()
My favorite thing in the world is robotics
```

Perhatikan bahwa untuk kedua kalinya kami memanggil fungsi, kami tidak menyertakan nilai. Jadi, fungsi tersebut hanya menggunakan nilai default yang kita tetapkan saat kita membuat fungsi.

Mengembalikan Nilai

Terkadang kita tidak hanya ingin fungsi melakukan sesuatu sendiri. Terkadang kita ingin itu memberi nilai kembali kepada kita. Ini berguna untuk memindahkan penghitungan umum ke suatu fungsi, atau jika kita ingin fungsi memvalidasi bahwa itu berjalan dengan benar. Banyak fungsi bawaan dan yang berasal dari perpustakaan eksternal mengembalikan 1 jika fungsi berhasil dan 0 jika gagal. Untuk mengembalikan nilai, cukup gunakan kata kunci kembali diikuti oleh nilai atau variabel yang ingin Anda kembalikan. Ingatlah bahwa return keluar dari fungsi dan memberikan nilai ke baris yang memanggil fungsi. Jadi, pastikan Anda tidak melakukan apa pun setelah pernyataan pengembalian.

```
>>> def how_many(list_of_things):
      count = len(list_of_things)
      return count
>>> how_many(robots)
3
```

Pernyataan kembali dapat mengembalikan lebih dari satu nilai. Untuk mengembalikan lebih dari satu nilai, pisahkan masing-masing dengan koma. Fungsi menempatkan nilai-nilai ke dalam tupel yang dapat diuraikan oleh kode panggilan.

```
>>> def how_many(list_of_things):
      count = len(list_of_things)
      return count, 1
>>> (x, y) = how_many(robots)
>>> x
3
>>> y
1
```

3.17 MENAMBAHKAN FUNGSI MELALUI MODUL

Modul pada dasarnya adalah kumpulan fungsi dalam file yang dapat Anda sertakan dalam program Anda. Ada banyak modul untuk membuat hidup Anda lebih mudah. Banyak modul disertakan sebagai bagian dari instalasi Python standar. Lainnya tersedia untuk diunduh dari berbagai pengembang. Jika Anda tidak dapat menemukan apa yang Anda cari, Anda dapat membuat modul kustom Anda sendiri.

Mengimpor Dan Menggunakan Modul

Mengimpor modul itu mudah. Seperti yang Anda lihat, Anda cukup menggunakan kata kunci import diikuti dengan nama modul. Ini memuat semua fungsi modul itu untuk Anda gunakan. Sekarang, untuk menggunakan salah satu fungsi dari modul, Anda harus memasukkan nama modul diikuti dengan fungsinya.

```
>>> import math
>>> math.sqrt(9)
3.0
```

Beberapa paket berukuran sangat besar, dan Anda mungkin tidak ingin mengimpor semuanya. Jika Anda mengetahui fungsi spesifik yang Anda butuhkan dalam program Anda, Anda hanya dapat mengimpor bagian modul tersebut. Ini mengimpor fungsi sqrt dari modul matematika. Jika Anda mengimpor hanya fungsinya, Anda tidak perlu mengawali fungsi dengan nama modul.

```
>>> from math import sqrt
>>> sqrt(9)
3.0
```

Terakhir, Anda dapat memberikan alias untuk modul dan fungsi yang Anda impor. Ini menjadi sangat berguna ketika Anda mengimpor modul dengan nama yang cukup panjang. Dalam contoh ini, saya hanya malas:

```
>>> import math as m
>>> m.sqrt(9)
3.0
>>> from math import sqrt as s
>>> s(9)
3.0
```

Modul Bawaan

Pustaka inti Python menyediakan banyak fungsi untuk program dasar. Namun, ada lebih banyak fungsi yang tersedia, yang ditulis oleh pengembang dan peneliti lain. Tapi sebelum kita masuk ke dunia modul pihak ketiga yang luar biasa, mari kita lihat apa yang ada di Python.

Buka instance IDLE dan ketik berikut ini:

```
>>> import sys
>>> sys.builtin_module_names
```

Anda harus mendapatkan output yang terlihat seperti ini:

```
('_ast', '_bisect', '_codecs', '_codecs_cn', '_codecs_hk',
'_codecs_iso2022', '_codecs_jp', '_codecs_kr', '_codecs_tw', '_collections',
'_csv', '_datetime', '_functools', '_heapq', '_imp', '_io', '_json',
'_locale', '_lsprof', '_md5', '_multibytecodec', '_opcode', '_operator',
'_pickle', '_random', '_sha1', '_sha256', '_sha512', '_signal', '_sre',
'_stat', '_string', '_struct', '_symtable', '_thread', '_tracemalloc',
'_warnings','_weakref','_winapi','array','atexit','audioop','binascii',
'builtins', 'cmath', 'errno','faulthandler', 'gc', 'itertools', 'marshal',
'math', 'mmap','msvcrt', 'nt', 'parser', 'sys', 'time', 'winreg', 'xxsubtype',
'zipimport','zlib')
```

Ini adalah daftar modul yang dibangun ke dalam Python dan tersedia untuk digunakan sekarang.

Untuk mendapatkan informasi lebih lanjut tentang modul, Anda dapat menggunakan bantuan () fungsi. Ini mencantumkan semua modul yang saat ini diinstal dan terdaftar dengan Python. (Perhatikan bahwa saya harus memotong daftar untuk dicetak.)

```
>>> help('modules')
```

Mohon tunggu sebentar sementara saya mengumpulkan daftar semua modul yang tersedia...

```
AutoComplete            _random     errno           pyexpat
AutoCompleteWindow      _sha1       faulthandler    pylab
AutoExpand              _sha256     filecmp         pyparsing
Bindings                _sha512     fileinput       pytz C
allTipWindow            _signal     fnmatch         queue
...
Enter any module name to get more help.Or, type "modulesspam" to search for
modules whose name or summary contain the string "spam"
```

Anda juga dapat menggunakan fungsi help() untuk mendapatkan informasi tentang modul tertentu. Pertama, Anda perlu mengimpor modul. Sekali lagi, daftar berikut telah dipotong untuk singkatnya.

```
>>> import math
>>> help(math)
Help on built-in module math:
NAME
    math
DESCRIPTION
    This module is always available.      It provides access to the
    mathematical functions defined by the C standard.
```

```
 FUNCTIONS
    acos(...)
        acos(x)
        Return the arc cosine (measured in radians) of x.
 ...
 FILE
(built-in)
```

Anda dapat mempelajari lebih banyak tentang modul bawaan ini di situs dokumentasi Python di https://docs.python.org/3/py-modindex.html.

Modul Yang Diperpanjang

Selain modul bawaan yang Anda dapatkan dengan setiap instalasi Python, ada banyak ekstensi yang dapat Anda tambahkan yang disebut paket. Untungnya, orang-orang baik di Python telah menyediakan metode untuk belajar tentang paket pihak ketiga. Kunjungi https://pypi.python.org/pypi untuk informasi lebih lanjut.

Setelah Anda menemukan paket yang ingin atau perlu Anda instal untuk aplikasi Anda, cara termudah untuk menginstalnya adalah dengan menggunakan PIP. Pada Python 2.7.9 dan Python 3.4, binari PIP disertakan dalam unduhan. Namun, karena paket terus berkembang, Anda mungkin perlu memutakhirkannya. Jika semuanya terinstal dan dikonfigurasi dengan benar, Anda harus dapat melakukan ini dari baris perintah.

1. Buka jendela terminal.
2. Di Windows, ketik
   ```
   python -m pip install -U pip
   ```
3. Di Linux atau macOS, ketik
   ```
   pip install -U pip
   ```

Setelah selesai, Anda siap menggunakan PIP. Ingatlah bahwa Anda akan menjalankan PIP dari terminal, bukan dari dalam shell Python. Untuk demonstrasi ini, kami akan menginstal paket yang digunakan untuk merencanakan rumus matematika. matplotlib adalah paket yang sangat populer untuk memvisualisasikan data menggunakan Python. Penggunaan sebenarnya dari paket ini berada di luar cakupan lokakarya ini. Untuk informasi lebih lanjut tentang menggunakan matplotlib, lihat situs web mereka di https://matplotlib.org.

Untuk menginstal paket baru, ketik

```
pip install matplotlib
```

Ini menginstal perpustakaan matplotlib untuk Anda gunakan.

Modul Kustom

Jika Anda memiliki beberapa fungsi yang selalu Anda gunakan (umumnya disebut sebagai fungsi pembantu), Anda dapat menyimpannya dalam file bernama `myHelperFunctions.py`. Anda kemudian dapat menggunakan perintah impor untuk membuat fungsi-fungsi ini tersedia di program lain.

Secara umum, Anda menyimpan file modul khusus untuk diimpor di lokasi file yang sama dengan program yang sedang Anda kerjakan. Ini adalah cara termudah dan terbaik untuk memastikan bahwa kompiler dapat menemukan file. Dimungkinkan untuk menyimpan file di tempat lain, tetapi kemudian Anda menyertakan path lengkap untuk file tersebut atau membuat perubahan pada variabel path sistem. Untuk saat ini, simpan file modul apa pun yang Anda buat di direktori kerja Anda (lokasi yang sama sebagai program yang sedang Anda kerjakan). Ini membantu Anda menghindari sakit hati tambahan.

Sampai sekarang, kami telah menggunakan shell IDLE . Mari buat file modul khusus, lalu impor ke program lain.

1. Buka IDLE.
2. Klik **File ➤ New File**. Ini akan membuka jendela editor teks baru.
3. Di jendela file baru, klik **File ➤ Save** dan beri nama myHelperFunctions.py.
4. Masukkan kode berikut:
   ```
   def hello_helper():
   print(″I'm helper. I help.″))
   ```
5. Simpan file.
6. Klik File New File untuk membuat file kode baru.
7. Ketik berikut ini:
   ```
   import myHelperFunctions myHelperFunctions.hello_helper(
   ```
8. Simpan file sebagai hello_helper.py di direktori yang sama dengan yang Anda simpan myHelperFunctions.py.
9. Tekan F5 atau pilih **Run ➤ Run Module** dari menu.
10. Di jendela shell, Anda akan melihat ini:
    ```
    I'm helper. I help.
    ```

3.18 KELAS

Sekarang kita sampai pada hal-hal yang baik: kelas. Kelas tidak lebih dari representasi logis dari entitas fisik atau abstrak dalam kode Anda; misalnya robot. Kelas robot membuat kerangka kerja yang menggambarkan robot fisik ke program. Bagaimana Anda menggambarkannya sepenuhnya terserah Anda, tetapi itu diwakili dalam bagaimana Anda membangun kelas. Representasi ini abstrak dengan cara yang sama seperti kata robot mewakili abstraksi dari konsep robot. Jika kita berdiri di ruangan yang penuh dengan robot dan saya berkata, "Berikan saya robotnya," tanggapan Anda kemungkinan besar adalah, "Robot yang mana?" Ini karena istilah robot berlaku untuk setiap robot yang ada di dalam ruangan. Tapi, jika saya mengatakan, "Berikan saya Nomad," Anda akan tahu robot spesifik yang saya bicarakan. Nomad adalah contoh robot.

Ini adalah bagaimana kelas digunakan. Anda mulai dengan mendefinisikan kelas. Anda melakukan ini dengan membangun abstraksi dari entitas yang ingin Anda wakili; dalam hal ini robot. Saat Anda ingin mendeskripsikan robot tertentu, Anda membuat instance kelas yang berlaku untuk robot tersebut.

Ada banyak hal yang harus dipelajari tentang kelas, tetapi berikut ini adalah hal-hal

penting yang perlu Anda ketahui.

- Sebuah kelas terdiri dari fungsi-fungsi yang disebut metode. Metode adalah fungsi dalam kelas yang melakukan pekerjaan. Misalnya, Anda mungkin memiliki metode di kelas robot yang disebut drive_forward(). Dalam metode ini, Anda menambahkan kode untuk membuat robot bergerak maju.
- Sebuah metode selalu membutuhkan parameter self. Parameter ini adalah referensi ke instance kelas.
- self selalu menjadi parameter pertama dari sebuah metode.
- Setiap kelas harus memiliki metode khusus yang disebut _init. Metode _init dipanggil ketika sebuah instance dibuat, dan itu menginisialisasi instance kelas tersebut. Dalam metode ini, Anda melakukan apa pun yang perlu terjadi agar kelas berfungsi. Paling sering, Anda mendefinisikan atribut untuk kelas.
- Atribut dari sebuah kelas adalah variabel di dalam kelas yang menjelaskan beberapa fitur. Misalnya, untuk kelas robot, kami ingin memberi nama beberapa atribut fungsional, seperti arah dan kecepatan. Ini dibuat dalam metode init.

Ada beberapa jenis metode:

- *Metode mutator:* Metode ini mengubah nilai di dalam kelas. Misalnya, setter adalah jenis metode mutator yang menetapkan nilai atribut.
- *Metode Accessor:* Metode ini mengakses atribut di dalam kelas.
- *Metode Helper:* Ini termasuk sejumlah metode yang melakukan pekerjaan di dalam kelas. Misalnya, metode _init_ wajib adalah jenis pembantu disebut konstruktor. Metode pembantu adalah segala sesuatu yang melakukan pekerjaan di dalam kelas, umumnya untuk metode lain; misalnya, metode yang memformat string sebelum output.

Membuat Kelas

Sebelum Anda mempelajari dan mulai menulis kode, saya sarankan Anda meluangkan sedikit waktu untuk merencanakan apa yang akan Anda buat. Ini tidak perlu menjadi rencana ekstensif yang menghilangkan setiap detail, tetapi ada baiknya untuk memiliki setidaknya garis besar kasar tentang apa yang akan Anda bangun sebelum Anda membangunnya.

Perencanaan

Cara termudah untuk membuat rencana adalah pada selembar kertas, tetapi jika Anda lebih suka digital, editor teks favorit Anda juga dapat melakukannya. Anda ingin membuat daftar atau garis besar kelas. Kelas contoh kami adalah untuk robot beroda yang disimulasikan, jadi kami ingin membuat daftar atribut yang menggambarkan robot kami, dan kemudian membuat daftar tindakan yang akan dilakukan robot. Ini adalah metode kami.

Kelas Robot Sampel Awal

- Atribut
    - Nama
    - Keterangan
    - Warna primer
    - Pemilik

- Metode
    - Berkendara ke depan
    - Berkendara mundur
    - Belok kiri
    - Belok kanan

Saat Anda menulis garis besar Anda, bayangkan bagaimana Anda akan menggunakan setiap metode. Informasi apa, jika ada, yang Anda perlukan untuk itu? Informasi apa, jika ada, yang akan dikembalikan? Jika metode Anda mengharapkan informasi dalam bentuk parameter, apakah ada nilai default? Jika demikian, apakah Anda ingin memesan? kemampuan untuk mengubah nilai default secara terprogram? Dari pengalaman saya, jawaban untuk pertanyaan terakhir ini hampir selalu ya. Jadi, dengan mengingat pertanyaan-pertanyaan ini, mari kita tinjau kembali garis besarnya.

Kelas Robot Sampel Awal

- Atribut
    - Nama
    - Keterangan
    - Warna primer
    - Pemilik
    - Kecepatan default (default: 125)
    - Durasi default (default: 100)
- Metode
    - Berkendara ke depan (parameter: kecepatan) (kembali: tidak ada)
    - Berkendara mundur (parameter: kecepatan) (kembali: tidak ada)
    - Belok kiri (parameter: durasi) (kembali: tidak ada)
    - Belok kanan (parameter: durasi) (kembali: tidak ada)
    - Atur kecepatan (parameter: kecepatan baru) (kembali: tidak ada)
    - Atur durasi (parameter: durasi baru) (kembali: tidak ada)

Seperti yang Anda lihat, setelah meninjau kembali garis besar, kami menambahkan beberapa atribut baru dan beberapa metode baru. Kecepatan default memegang nilai integer antara 0 dan 255. Kemudian dalam buku ini, kami menggunakan nilai ini untuk mengatur kecepatan pengontrol motor kami. Kecepatan setengahnya adalah 125. Durasi default adalah jumlah waktu robot bergerak dalam milidetik. Nilai 100 adalah sekitar 1/10 detik. Kami juga menambahkan dua metode untuk mengatur nilai dari dua atribut ini. Dalam sebagian besar bahasa pemrograman, atribut bersifat pribadi, yang berarti atribut tersebut hanya dapat diakses dari kode yang terdapat di dalam kelas. Dengan demikian, Anda membuat metode get() dan set() untuk melihat dan mengubah nilai. Dalam Python, atribut bersifat publik dan dapat diakses atau diubah dengan panggilan `class.attribute` sederhana. Atribut Python tidak dapat dijadikan pribadi; namun, tradisi dalam Python adalah memberi awalan atribut yang ingin Anda jadikan pribadi dengan garis bawah. Ini menunjukkan kepada pengembang lain bahwa atribut harus diperlakukan sebagai pribadi dan tidak dimodifikasi di luar metode

kelas.

Jadi, sebenarnya, metode kecepatan yang ditetapkan dan durasi yang ditetapkan tidak sepenuhnya diperlukan. Jika kita ingin menunjukkan bahwa atribut ini dimaksudkan untuk menjadi pribadi dan hanya boleh diperbarui dengan metode, maka kita mendahului nama dengan garis bawah, seperti ini:

```
_speed
_duratio
```

Anda dapat membuat kelas di mana saja dalam kode Anda. Apa yang membuat kelas sangat berguna adalah bahwa mereka merangkum fungsionalitas yang memungkinkan Anda untuk dengan mudah memindahkannya dari satu proyek ke proyek berikutnya. Untuk alasan ini, umumnya lebih baik membuat kelas sebagai modulnya sendiri dan mengimpornya ke dalam kode Anda. Itulah yang akan kami lakukan di sini.

Mari kita membangun kelas robot kita dan kemudian menggunakannya.

1. Buat file Python baru dan simpan sebagai `robot_sample_class.py`.
   Kami akan mulai dengan mendeklarasikan kelas kami dan membuat fungsi konstruktor yang diperlukan, init . Saat ini, yang perlu kita lakukan adalah menginisialisasi atribut dan memindahkan nilai dari parameter ke atribut. Perhatikan bahwa kami telah mendeklarasikan nilai default untuk kecepatan dan durasi masing-masing sebagai 125 dan 100.
2. Masukkan kode berikut:

```
class Robot():
        """
        A simple robot class
        This multi-line comment is a good place
        to provide a description of what the classis.
        """
        # define the initiating function.
        # speed = value between 0 and 255
        # duration = value in milliseconds
        def_init_(self, name, desc, color, owner,speed = 125, duration =
        100):
              # initilaizes our robotself.name = name
              self.desc = desc
              self.color = color
              self.owner = owner
              self.speed = speed
              self.duration = duration
```

Setelah inisialisasi selesai, mari kita lihat penulisan metode kita. Seperti disebutkan, metode hanyalah fungsi yang terkandung dalam kelas yang melakukan pekerjaan di dalam

kelas. Karena kami tidak memiliki robot untuk dikendalikan saat ini, kami hanya mencetak pesan konfirmasi ke shell untuk mensimulasikan robot kami.

```
        def drive_forward(self):
                # simulates driving forward print(self.name.title() + "
                is driving" +
                        " forward " + str(self.duration) +"
                        milliseconds")
        def drive_backward(self):
                # simulates driving backward print(self.name.title() + "
                is driving" +
                        " backward " + str(self.duration) +"
                        milliseconds")
        def turn_left(self):
        # simulates turning left
                print(self.name.title() + " is turning " +" right " +
                        str(self.duration) +
                        " milliseconds")
        def turn_right(self):
                # simulates turning right print(self.name.title() + " is
                turning " +
                        " left " + str(self.duration) +" milliseconds")
        def set_speed(self, speed):
# sets the speed of the motorsself.speed = speed
                print("the motor speed is now " +str(self.speed))
        def set_duration(self, duration):# sets duration of travelself.
                duration = duration
print("the duration is now " +str(self. duration))
```

3. Simpan file.
   Sekarang setelah kita membuat kelas Robot baru, kita akan menggunakannya untuk mendefinisikan Nomad sebagai Robot.
4. Buat file Python baru dan simpan sebagai `robot_sample.py`.
   Kita akan mulai dengan mengimpor kode robot_sample_class, dan kemudian menggunakannya untuk membuat robot baru bernama Nomad.
5. Masukkan kode berikut:

```
import robot_sample_class
my_robot = Robot("Nomad", "Autonomous rover",black", "Jeff Cicolani")
```

   Menggunakan definisi kelas untuk membuat instance baru dari kelas disebut

instantiasi. Perhatikan bahwa kami tidak memberikan nilai untuk dua parameter terakhir, kecepatan dan durasi. Karena kami memberikan nilai default untuk parameter ini, kami tidak perlu memberikan nilai selama instantiasi. Jika kami tidak memberikan nilai default, kami akan mendapatkan kesalahan ketika kami mencoba menjalankan kode.

Dengan instance robot baru kami, mari kita lakukan beberapa pekerjaan dengannya.

```
print("My robot is a " + my_robot.desc + " called " +my_robot.name)
my_robot.drive_forward()
my_robot.drive_backward()
my_robot.turn_left()
my_robot.turn_right()
my_robot.set_speed(255)
my_robot.set_duration(1000)
```

6. Simpan filenya.
7. Tekan F5 untuk menjalankan program.

   Di jendela shell Python, Anda akan melihat sesuatu seperti ini:

```
>>> ======================RESTART==================
>>>
My robot is an autonomous rover called Nomad
Nomad is driving forward 100 milliseconds
Nomad is driving backward 100 milliseconds
Nomad is turning left 100 milliseconds
Nomad is turning right 100 milliseconds
the motor speed is now 255
the duration is now 1000
```

3.19 GAYA

Sebelum kita menyelesaikan bab ini, saya ingin berbicara tentang menata kode Anda. Kami telah melihat bahwa lekukan itu penting dan harus memenuhi pedoman ketat untuk menunjukkan blok kode dan sebagainya. Tetapi ada beberapa area di mana Anda dapat memengaruhi keputusan penataan gaya yang tidak terlalu penting. Tentu saja, ada tradisi dalam komunitas Python yang direkomendasikan. Ada beberapa praktik terbaik yang disarankan oleh pembuat dan pengembang utama Python. Anda dapat membaca semua saran mereka di Panduan Gaya Python di www.python.org/dev/peps/pep-0008/. Saya sarankan membaca panduan gaya dan mempraktikkan saran mereka sebelum Anda mengembangkan beberapa kebiasaan yang sangat buruk (seperti yang saya lakukan). Untuk saat ini, mari fokus pada bagaimana Anda memberi nama variabel, fungsi, dan kelas Anda.

Garis Kosong

Meninggalkan baris kosong di antara blok kode untuk pemisahan visual yang logis adalah ide yang bagus. Itu membuat kode Anda lebih mudah dibaca.

Mengomentari

Tulis komentar dalam kode Anda. Lakukan dengan sering dan bertele-tele. Ketika Anda kembali untuk membaca kode Anda nanti (untuk men-debug atau menggunakannya kembali untuk proyek lain), Anda akan ingin tahu apa yang Anda pikirkan ketika kode itu ditulis, dan apa yang Anda coba lakukan dengannya. Jika kode Anda berhasil keluar ke alam liar, di mana orang lain membaca atau mengulasnya, mereka juga membutuhkan komentar. Python adalah komunitas, dan kode sering dibagikan. Kode yang dikomentari dan dijelaskan dengan baik sangat dihargai.

Konvensi Penamaan

Bagaimana Anda memberi nama variabel, fungsi, dan kelas Anda adalah keputusan pribadi. Lakukan apa yang paling nyaman bagi Anda. Python adalah bahasa yang peka terhadap huruf besar/kecil. Menggunakan huruf kapital di satu tempat dan bukan di tempat lain menciptakan dua variabel berbeda dan frustrasi berjam-jam.

Nama variabel umum tidak dibahas dalam panduan gaya, meskipun konvensinya adalah menggunakan penamaan huruf besar-kecil. Nama dengan huruf besar campuran dimulai dengan karakter huruf kecil, tetapi setiap kata dalam nama menggunakan huruf kapital; misalnya myRobots. Fungsi dan modul harus huruf kecil. Untuk membuatnya lebih mudah dibaca, gunakan garis bawah di antara kata-kata. Jadi fungsi hello world kami bernama hello_world. Kelas harus diberi nama menggunakan CapWords. Sesuai dengan namanya, CapWords menggunakan huruf kapital pada huruf pertama setiap kata, termasuk karakter pertama pada namanya. Gaya ini lebih dikenal sebagai kasus unta. Terakhir, daftar dan koleksi lainnya harus dimajemukkan. Ini adalah indikator bahwa variabel mewakili lebih dari satu objek. Misalnya, robot adalah daftar robot. Jika kami menangani item individual dalam daftar, itu akan terlihat seperti ini:

```
robot = robots[0]
```

3.20 RINGKASAN

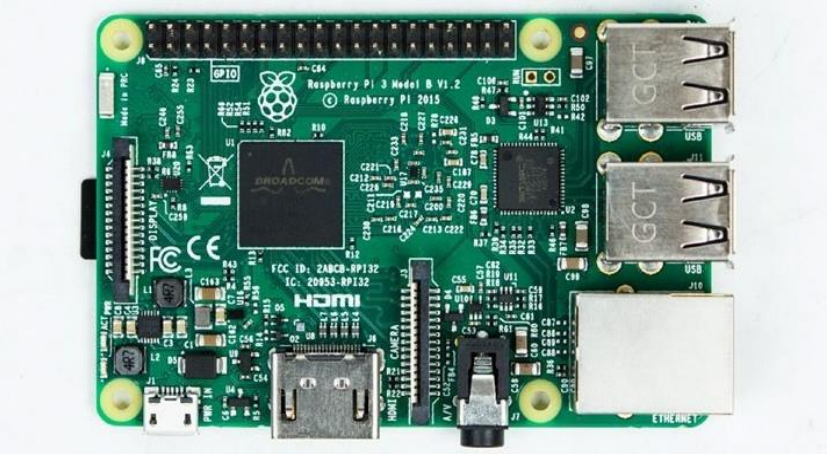
Kami menggunakan Python di seluruh buku ini. Ini adalah bahasa yang sangat sederhana untuk dipelajari, dan menyediakan banyak fitur canggih. Banyak pengembang perangkat lunak berpikir bahwa Python lambat. Tetapi di mana lambat di beberapa area, itu lebih dari menghabiskan waktu di area lain, seperti yang akan Anda lihat ketika kita mulai bekerja dengan computer vision di Bab 9.

BAB 4
RASPBERRY PI GPIO

Bab sebelumnya memperkenalkan perangkat keras Raspberry Pi, dan Anda telah mempelajari cara menggunakan Python untuk memprogramnya. Anda menginstal sistem operasi, mengonfigurasinya untuk penggunaan Anda, dan mengatur akses jarak jauh sehingga Anda dapat memprogram Pi tanpa menghubungkan keyboard, mouse, dan monitor langsung ke sana. Anda mempelajari struktur dasar program Python, sintaksis, dan cukup banyak tentang bahasa untuk mulai menulis program. Selanjutnya, Anda akan belajar cara menggunakan antarmuka GPIO Raspberry Pi untuk berinteraksi dengan dunia fisik. Ini sangat penting untuk robotika karena begitulah cara prosesor mendeteksi apa yang terjadi di sekitarnya dan merespons rangsangan dari luar. Tanpa kemampuan untuk mendeteksi dan bertindak di dunia fisik, segala jenis otonomi cerdas tidak mungkin terjadi.

4.1 RASPBERRY PI GPIO

Ada beberapa cara untuk terhubung ke Raspberry Pi. Sejauh ini yang paling sederhana adalah melalui salah satu port USB yang terpasang di papan. Port USB menyediakan empat koneksi serial yang melaluinya Anda dapat mengakses komponen luar, seperti keyboard dan mouse yang kami gunakan untuk menyiapkan Pi. Namun, port USB memerlukan perangkat keras khusus untuk mengubah perintah serial menjadi sinyal yang diperlukan untuk mengoperasikan perangkat. Raspberry Pi memiliki metode yang lebih langsung untuk menghubungkan ke perangkat eksternal: header GPIO.



Gambar 4-1. Raspberry Pi dengan header 40-pin

GPIO adalah antarmuka antara elektronik dan seluruh dunia. Header umumnya mengacu pada satu set pin di papan yang memungkinkan akses ke fungsi tertentu. Header GPIO adalah sepasang baris 20-pin yang berjalan di sepanjang salah satu tepi papan (lihat Gambar 4-1), yang disebut sebagai header 40-pin. Sangat penting untuk

dicatat bahwa header menyediakan koneksi langsung ke elektronik di papan tulis. Tidak ada penyangga atau fitur keamanan yang terpasang pada pin ini. Ini berarti bahwa jika Anda menghubungkan sesuatu secara tidak benar atau menggunakan voltase yang salah, kemungkinan Anda akan merusak Pi Anda.

Berikut ini adalah hal-hal yang perlu Anda perhatikan sebelum bekerja dengan header:

- Meskipun Raspberry Pi ditenagai dengan adaptor mikro USB 5 volt, elektroniknya 3,3 volt. Ini berarti Anda perlu memperhatikan voltase yang digunakan sensor.
- Ada dua tegangan yang diberikan pada pin GPIO: 5V dan 3.3V. Hati-hati yang mana yang Anda gunakan, terutama jika mencoba memberi daya pada Pi melalui GPIO.
- Dimungkinkan untuk memberi daya pada Raspberry Pi melalui salah satu pin GPIO 5V; namun, perlindungan dan pengaturan sirkuit tidak disediakan. Jika Anda memberikan terlalu banyak tegangan, atau ada lonjakan arus, papan mungkin rusak. Jika Anda harus menggunakan pin GPIO untuk memberi daya pada board, pastikan untuk menyediakan regulator eksternal.
- Ada dua skema penomoran untuk header GPIO: board dan BCM. Ini berarti ada dua cara berbeda untuk mereferensikan pin dari kode Anda; yang Anda putuskan untuk digunakan umumnya terserah Anda. Anda hanya perlu mengingat skema mana yang Anda pilih.

4.2 PENOMORAN PIN

Seperti yang saya sebutkan, ada dua skema penomoran untuk header 40-pin: board dan BCM. Penomoran papan hanya memberi nomor pin secara berurutan. Pin 1 adalah yang paling dekat dengan kartu micro SD, dan pin 2 adalah pin yang berdekatan yang paling dekat dengan tepi luar Pi. Penomoran berlanjut dengan cara ini, dengan pin bernomor ganjil di baris dalam dan pin bernomor genap di luar. Pin 40 adalah pin di tepi papan, dekat port USB.

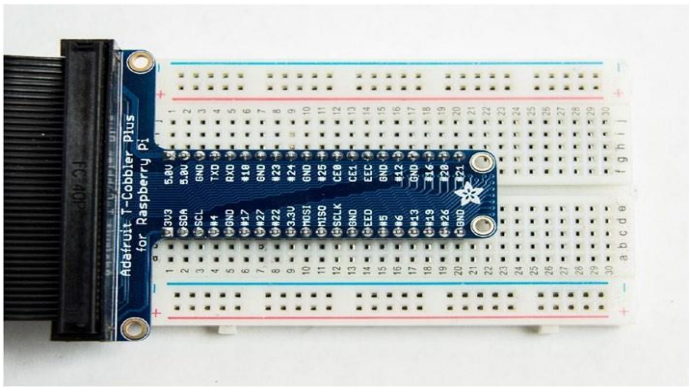
Penomoran BCM hampir tidak semudah itu. BCM adalah singkatan dari Broadcom, produsen SoC (system on a chip) yang menggerakkan Pi. Pada Raspberry Pi 2, prosesornya adalah BCM2836; pada Raspberry Pi 3, ini adalah BCM2837. Penomoran BCM mengacu pada nomor pin chip Broadcom, yang dapat bervariasi antar versi. BCM2836 dan BCM2837 memiliki pin-out yang sama, sehingga tidak ada perbedaan antara Pi 2 dan Pi 3.

Untuk menghubungkan komponen elektronik ke header 40-pin, kami akan menggunakan Adafruit T-Cobbler Plus dan papan tempat memotong roti. T-Cobbler memiliki informasi pin yang tertera di papan untuk referensi cepat; namun, T-Cobbler menggunakan penomoran BCM. Jadi, kita akan menggunakan penomoran BCM.

4.3 MENGHUBUNGKAN KE RASPBERRY PI

Ada beberapa cara untuk menghubungkan pin dari header ke perangkat lain.

Kontroler motor yang akan kita gunakan adalah contoh papan yang berada tepat di atas header. Dalam terminologi Raspberry Pi, papan ini disebut sebagai topi atau piring. Pilihan lain adalah menghubungkan langsung ke pin menggunakan jumper. Bagi banyak orang, ini adalah metode yang disukai selama pembuatan prototipe. Saya lebih suka metode ketiga, yaitu menggunakan papan lain dari Adafruit yang disebut Pi Cobbler. Ada beberapa versi tukang sepatu, tapi saya lebih suka Adafruit T-Cobbler Plus untuk Raspberry Pi (lihat Gambar 4-2). Papan ini dirancang untuk dipasang ke papan tempat memotong roti melalui kabel pita. Ini menggunakan header 40-pin yang dikonfigurasi tegak lurus dengan pin yang dicolokkan ke papan tempat memotong roti. Ini memindahkan lampiran kabel pita dari papan tempat memotong roti dan memungkinkan akses yang lebih baik ke lubang.



Gambar 4-2. T-Cobbler dipasang di papan tempat memotong roti

Salah satu keuntungan menggunakan tukang sepatu adalah bahwa pin breakout ditandai dengan jelas. Ketika kami mulai membangun sirkuit kami, akan sangat mudah untuk melihat dengan tepat apa yang Anda kaitkan. Ini juga memudahkan untuk mengidentifikasi pin mana yang digunakan untuk kode Anda. Saat Anda mendeklarasikan pin 21 sebagai pin keluaran, Anda akan tahu persis pin mana yang ada di papan tulis.

4.4 KETERBATASAN GPIO

Ada beberapa hal yang perlu diingat saat Anda bekerja dengan GPIO. Pertama, Raspberry Pi yang kita atur bukanlah perangkat real-time. Debian Linux adalah sistem operasi lengkap dengan banyak lapisan abstraksi dari perangkat keras. Ini berarti bahwa perintah ke perangkat keras tidak langsung. Sebaliknya, perintah dilewatkan melalui beberapa operasi sebelum dan setelah CPU mengirimkannya ke papan. Python beroperasi di lapisan abstraksi lain. Masing-masing lapisan ini memperkenalkan tingkat kelambatan tertentu. Ini umumnya tidak terlihat oleh kami, tetapi dapat membuat perbedaan besar dalam operasi robot. Ada distribusi Debian yang lebih real time, dirancang untuk aplikasi industri, tetapi versi Raspbian standar yang kami gunakan bukan salah satunya. Kedua, tidak ada input analog pada Pi. Ya, ada satu, tetapi itu dibagi dengan port serial, yang kemungkinan akan kita gunakan nanti untuk hal lain. Jadi, lebih baik untuk menerima bahwa tidak ada pin input analog. Anda akan

melihat mengapa ini penting di Bab 5. Ketiga, Pi hanya memiliki dua pin berkemampuan PWM. PWM adalah singkatan dari Pulse Width Modulation, yaitu bagaimana kita mengirim sinyal yang bervariasi ke perangkat eksternal. Ini berarti hanya ada dua pin pada header yang dapat mensimulasikan output analog. Kedua pin ini juga dibagi dengan output audio Pi, yang tidak optimal.

Kabar baiknya adalah ada solusi sederhana untuk semua masalah ini, yaitu cukup dengan memperkenalkan mikrokontroler eksternal yang real time, menawarkan beberapa input analog, dan menyediakan lebih dari dua output PWM. Kami akan menggunakan ini dengan Arduino di Bab 5. Arduino pada dasarnya adalah papan prototipe untuk mikrokontroler seri AVR AT. Chip ini terhubung langsung ke perangkat keras dan tidak memiliki lapisan abstraksi yang Anda temukan di sebagian besar prosesor SoC, seperti yang ada di Pi. Ada keuntungan lain menggunakan Arduino, yang saya bahas di Bab 5.

4.5 MENGAKSES GPIO DENGAN PYTHON

Perangkat keras hanyalah bagian dari persamaan. Kami akan menggunakan keterampilan Python baru kami untuk memprogram perilaku yang kami inginkan. Untuk melakukan itu, kami akan menggunakan perpustakaan RPi.GPIO. Anda akan ingat dari Bab 3 bahwa perpustakaan adalah kumpulan kelas dan fungsi yang menyediakan fungsionalitas tambahan. Dalam robotika, perangkat keras, sensor, atau komponen lain yang baru sering kali memiliki pustaka untuk memungkinkan Anda menggunakannya dengan lebih mudah. Terkadang perpustakaan bersifat generik, seperti RPi.GPIO; di lain waktu, perpustakaan dibuat untuk perangkat tertentu. Misalnya, kami akan menggunakan perpustakaan khusus untuk papan pengontrol motor di Bab 7. Saat Anda menambahkan lebih banyak perangkat keras ke robot Anda, Anda sering harus mengunduh perpustakaan baru dari situs web produsen. Anda akan melihat ini beraksi ketika kita mulai bekerja dengan pengontrol motor.

Pustaka GPIO menyediakan objek dan fungsi untuk mengakses pin GPIO. Raspbian hadir dengan perpustakaan yang diinstal, jadi itu harus siap untuk digunakan. Untuk informasi lebih lanjut tentang cara menggunakan paket, kunjungi https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/.

Untuk menggunakan perpustakaan GPIO, kita perlu melakukan dua hal: mengimpor paket dan kemudian memberi tahu mode mana yang akan kita gunakan untuk mengakses pin. Seperti yang saya diskusikan sebelumnya, ada dua mode board dan BCM yang pada dasarnya memberi tahu sistem referensi penomoran mana yang akan digunakan. Mode papan mereferensikan penomoran pada header P1 dari Raspberry Pi. Karena penomoran ini tetap konstan, untuk kompatibilitas mundur, Anda tidak perlu mengubah penomoran pin dalam kode Anda, berdasarkan revisi papan.

Sebaliknya, mode BCM mengacu pada penomoran pin dari Broadcom SoC, yang berarti bahwa pada versi Pi yang lebih baru, tata letak pin mungkin saja berubah. Untungnya, tata letak pin ini tidak berubah antara BCM2836 yang digunakan di Pi 2,

dan BCM2837 yang digunakan di Pi3. Untuk tujuan kami, kami akan menggunakan mode BCM—hanya karena itulah yang diilustrasikan pada T-Cobbler. Setiap program yang menggunakan header GPIO menyertakan dua baris kode berikut:

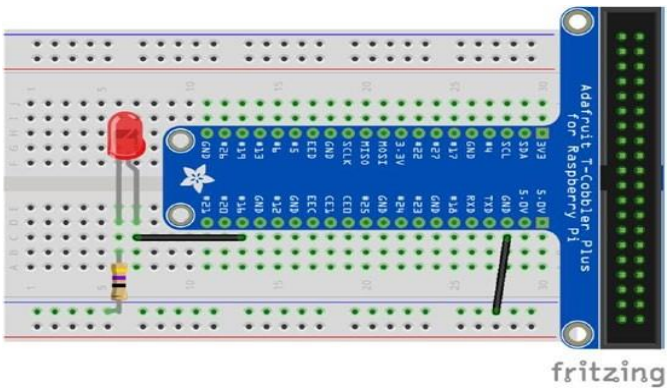
```
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
```

4.6 OUTPUT SEDERHANA: CONTOH LED

Contoh paling sederhana adalah versi perangkat keras "Hello World" yang ada di mana-mana LED yang berkedip. Proyek GPIO pertama kami adalah menghubungkan LED ke Pi dan menggunakan skrip Python untuk membuat LED berkedip. Mari kita mulai dengan menghubungkan sirkuit. Untuk melakukan ini, Anda memerlukan papan tempat memotong roti, T-Cobbler, LED, resistor 220ohm (Ω), dan dua potong kawat pendek untuk digunakan sebagai jumper.

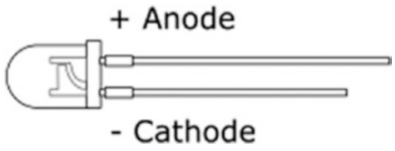
Menghubungkan Sirkuit

1. Pasang T-Cobbler seperti yang ditunjukkan pada Gambar 4-3. Satu baris pin harus berada di kedua sisi celah di papan. Penempatannya terserah Anda; namun, saya biasanya memasangnya sedemikian rupa sehingga header kabel pita terlepas dari papan. Ini memungkinkan akses maksimum ke papan tempat memotong roti.



Gambar 4-3. Tata letak sirkuit untuk contoh LED

2. Hubungkan resistor 220Ω antara rel ground dan rel 5 lubang kosong.
3. Hubungkan katoda LED ke rel yang sama dengan resistor. Katoda adalah pin yang paling dekat dengan sisi datar LED. Pada beberapa LED, pin ini lebih pendek dari pin lainnya (lihat Gambar 4-4).



Gambar 4-4. Polaritas LED

4. Hubungkan anoda LED ke rel 5-pin kosong lainnya.
5. Hubungkan jumper dari rel anoda ke rel yang terhubung ke pin 16 pada T-Cobbler.

6. Hubungkan jumper dari rel arde yang dihubungkan dengan LED dan rel yang terhubung ke salah satu pin arde T-Cobbler.

Jika Anda ingin menguji LED sebelum beralih ke kode, Anda dapat memindahkan jumper dari pin 16 ke salah satu pin 3.3V. Jika Pi Anda dihidupkan, LED akan menyala. Pastikan Anda memindahkan jumper kembali ke pin 16 sebelum melanjutkan.

Menulis Kode

Kode untuk proyek ini sangat sederhana. Itu ditulis dalam Python 3. Meskipun kode berfungsi di kedua versi, salah satu baris tidak akan berfungsi di Python 2.7. Secara khusus, garis cetak di bagian akhir menggunakan parameter akhir, yaitu tidak cocok. Jika Anda menggunakan Python 2.7, Anda harus menghilangkan parameter ini. Parameter akhir menggantikan default /n yang ditambahkan ke setiap baris yang dicetak, dengan /r. /r adalah carriage return yang bertentangan dengan baris baru yang diwakili oleh /n. Ini berarti bahwa kursor kembali ke awal baris saat ini, dan teks baru akan menimpa karakter sebelumnya. Namun, itu tidak menghapus garis terlebih dahulu. Jadi kami menambahkan sejumlah ruang kosong ke akhir teks baru untuk memastikan bahwa semua teks sebelumnya benar-benar dihapus.

Perintah GPIO mengakses memori tingkat sistem. Semua perintah tingkat sistem harus dijalankan dengan pengguna super atau akses root. Ini berarti Anda perlu menjalankan Python dengan sudo atau memberi diri Anda izin root permanen, yang bisa berbahaya. Setelah kami menulis kode, kami akan mengeksekusi dari perintah. Kita harus membuat file tersebut dapat dieksekusi sebelum kita melakukan ini, tetapi itu mudah dilakukan dari terminal.

Untuk memulai, mari buat file Python 3 baru dengan menggunakan IDLE atau di terminal. Jika menggunakan IDLE, lakukan hal berikut:

1. Buka IDLE untuk Python 3.
2. Klik Baru **New**.
3. Simpan file sebagai gpio_led.py di folder proyek Anda.

Jika menggunakan terminal, lakukan hal berikut:

1. Buka jendela terminal.
2. Arahkan ke folder proyek Anda. Di Pi saya, itu
   ```
   $ cd ~/TRG-RasPi-Robot/code
   ```
3. Ketik **touch gpio_led.py**.
4. Ketik **idle3 gpio_led.py**.
   Ini membuka file kosong di IDLE IDE untuk Python 3.
5. Setelah file Anda dibuat dan Anda berada di editor IDLE, masukkan kode berikut:
   ```
   # GPIO example blinking LED
   # Import the GPIO and time libraries
   ```

```
import RPi.GPIO as GPIO
import time
# Set the GPIO mode to BCM and disable warnings
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
# Define pins
led = 16
GPIO.setup(led,GPIO.OUT)
# Make sure LED is off
GPIO.output(led,False)
# Begin Loopw
hile True:
      # Turn LED on
      GPIO.output(led,True)
      # Wait 1 second
      time.sleep(1)
      # Turn LED off
      GPIO.output(led,False)
      # Wait 1 second
      time.sleep(1)
```

6. Simpan filenya.

Selanjutnya, kita akan menggunakan terminal untuk membuat file tersebut dapat dieksekusi dan kemudian menjalankannya.

1. Buka jendela terminal baru dan arahkan ke folder proyek Anda.
2. Ketik **chmod +x gpio_led.py**. Ini membuat file dapat dieksekusi.
3. Untuk menjalankan kode, ketik **sudo python3 gpio_led.p**.

Itu dia: LED yang berkedip. Hello World.

Modulasi Lebar Pulsa (PWM)

Meskipun hanya ada dua pin PWM pada header GPIO Pi, dan kemungkinan besar Anda tidak akan menggunakannya, akan sangat berguna untuk mengetahui cara mengontrolnya dengan baik. Dua pin PWM di papan adalah 18 dan 19. Untuk contoh ini, kami akan mengatur LED untuk menggunakan pin 18 dan menyalakan LED.

Menghubungkan Sirkuit

Baiklah, ini adalah bagian yang rumit. Untuk mengatur sirkuit ini, Anda harus mengikuti petunjuk ini dengan sangat cermat. Gunakan sirkuit yang kami buat untuk latihan LED.

1. Pindahkan jumper dari pin 16 ke pin 18.

   Fiuh. Sekarang setelah kita melewati semua itu, mari kita membuat kode.

Menulis Kode

Buat file Python 3 baru.

Jika menggunakan IDLE, lakukan hal berikut:

1. Buka IDLE untuk Python 3.
2. Klik **New**.
3. Simpan file sebagai gpio_pwm_led.py di folder proyek Anda.

Jika menggunakan terminal, lakukan hal berikut:

1. Di jendela terminal, navigasikan ke folder proyek Anda. Di Pi saya, itu
   `$ cd ~/TRG-RasPi-Robot/code.`
2. Ketik **touch gpio_pwm_led.py**.
3. Ketik **idle3 gpio_pwm_led.py**.
   Ini membuka file kosong di IDLE IDE untuk Python 3.
4. Setelah file Anda dibuat dan Anda berada di editor IDLE, masukkan kode berikut:

```
# GPIO example blinking LED
# Import the GPIO and time libraries
      import RPi.GPIO as GPIO
      import time
# Set the GPIO mode to BCM and disable warnings
      GPIO.setmode(GPIO.BCM)
      GPIO.setwarnings(False)
# Define pins
      pwmPin = 18
      GPIO.setup(pwmPin,GPIO.OUT)
      pwm = GPIO.PWM(pwmPin,100)
# Make sure LED is off
      pwm.start(0)
# Begin Loop
      while True:
      count = 1
# begin while loop to brighten LED
      while count < 100:
      # set duty cycle
            pwm.ChangeDutyCycle(count)
      # delay 1/100 of a second
            time.sleep(0.01)
      # increment count
            count = count + 1
# begin while loop to dim LED
```

```
while count > 1:
        pwm.ChangeDutyCycle(count)
        time.sleep(0.01)
# set duty cycle
        pwm.ChangeDutyCycle(count)
# delay 1/100 of a second
        time.sleep(0.01)
# decrement count
        count = count - 1
```

5. Buka jendela terminal baru dan arahkan ke folder proyek Anda.
6. Ketik **chmod +x gpio_pwm_led.py** untuk membuat file dapat dieksekusi.
7. Untuk menjalankan kode, ketik **sudo python3 gpio_pwm_led.py**

LED Anda sekarang harus berdenyut. Untuk mengubah kecepatan pulsa, ubah nilai dalam pemanggilan fungsi `time.sleep()`.

4.7 MASUKAN SEDERHANA

Sekarang kita telah melihat betapa mudahnya mengirim sinyal, saatnya untuk mendapatkan beberapa informasi kembali ke Pi. Kami akan melakukan ini melalui dua contoh. Pertama, tombol tekan; dalam contoh ini, Pi diatur untuk mengambil input dari tombol tekan sederhana dan menunjukkan di terminal saat tombol telah ditekan. Contoh kedua menggunakan pengintai sonik untuk membaca jarak ke suatu objek. Outputnya akan ditampilkan di terminal.

Contoh Tombol Tekan

Bentuk input yang paling sederhana adalah tombol tekan. Anda menekan tombol, sirkuit ditutup, dan sesuatu terjadi. Untuk contoh input pertama kami, kami akan menghubungkan tombol tekan ke header GPIO. Pada dasarnya ada dua cara untuk menghubungkan tombol tekan. Anda dapat mengaturnya untuk memulai dalam keadaan rendah, yang berarti bahwa ketika tombol tidak ditekan, tidak ada sinyal yang masuk ke pin, dan tegangan pada pin dibaca sebagai "rendah" oleh prosesor. Anda juga dapat menghubungkannya dalam keadaan tinggi. Dalam konfigurasi ini, pin terbaca sebagai high, atau on, ketika tombol tidak ditekan. Saat tombol ditekan, pin dibawa ke kondisi rendah. Anda sering mendengar istilah menarik tinggi atau menarik rendah. Menarik pin tinggi atau rendah adalah metode yang memaksa pin ke status tinggi atau rendah. Dalam banyak aplikasi, ini dilakukan dengan menambahkan resistor ke rangkaian. Sebuah resistor yang terhubung antara pin logika dan tegangan menyebabkan pin dalam keadaan tinggi. Pin ditarik tinggi. Tombol tersebut kemudian dihubungkan ke ground. Saat tombol ditekan, tegangan mengalir melalui tombol ke ground, melewati pin. Dengan tidak ada tegangan yang masuk ke pin, pin akan masuk ke status rendah.

Sebaliknya, menghubungkan resistor antara pin logika dan ground, dan kemudian menghubungkan tombol antara pin dan sumber tegangan, pin ditarik ke

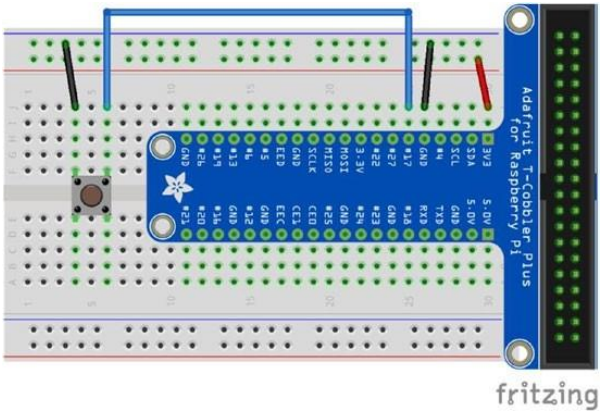
bawah. Saat tombol terbuka, tegangan sisa apa pun di pin ditarik ke ground, meninggalkan pin dalam kondisi rendah. Ketika tombol ditekan, tegangan diterapkan ke pin dan masuk ke status tinggi. Pin ditarik tinggi atau rendah untuk memastikan bahwa mereka berada dalam kondisi yang diharapkan saat tombol ditekan. Ini adalah cara untuk secara eksplisit memberi tahu sirkuit bagaimana perilakunya, dan ini umumnya merupakan praktik yang baik. Untungnya, Raspberry Pi memiliki sirkuit bawaan untuk mengakomodasi penarikan pin tinggi atau rendah. Ini berarti bahwa kita dapat menarik pin ke status yang tepat melalui kode, dan kita tidak perlu khawatir tentang menambahkan komponen tambahan. Untuk latihan ini, mari tarik pin tinggi-tinggi. Ketika tombol ditekan, pin menjadi rendah dan sebuah pesan dicetak ke jendela terminal.

Menghubungkan Sirkuit

Bagian-bagian berikut diperlukan untuk latihan ini:

- Sakelar tombol tekan taktil
- 4 jumper pria-ke-pria

1. Pasang T-Cobbler seperti yang ditunjukkan pada Gambar 4-5. Satu baris pin harus berada di kedua sisi celah di papan. Penempatannya terserah Anda; namun, saya biasanya memasangnya sehingga header kabel pita terlepas dari papan. Ini memungkinkan akses maksimum ke papan tempat memotong roti.



Gambar 4-5. Contoh tata letak sirkuit tombol tekan

2. Hubungkan tombol tekan taktil sehingga pin menjembatani celah di bagian tengah papan tempat memotong roti.
3. Hubungkan jumper antara pin 3.3V dan rel tegangan.
4. Hubungkan jumper lain antara pin arde dan rel arde.
5. Gunakan jumper lain untuk menghubungkan satu sisi sakelar taktil ke rel tanah.
6. Gunakan sisa jumper untuk menghubungkan pin tombol lainnya ke pin 17.

Sakelar taktil ini adalah tiang ganda, lemparan tunggal (DPST). Artinya, saat tombol ditekan, kedua pin di satu sisi celah papan tempat memotong roti terhubung. Pin di sisi lain celah membentuk sirkuit terpisah. Pastikan bahwa jumper akan

disematkan pada sisi yang sama.

Menulis Kode

Buat file Python 3 baru.

Jika menggunakan IDLE, lakukan hal berikut:

1. Buka IDLE untuk Python 3.
2. Klik **New**.
3. Simpan file sebagai gpio_button.py di folder proyek Anda.

Jika menggunakan jendela terminal, lakukan hal berikut:

1. Navigasikan ke folder proyek Anda. Di Pi saya itu
   ```
   $ cd ~/TRG-RasPi-Robot/code.
   ```
2. Ketik **touch gpio_button.py**.
3. Ketik **idle3 gpio_button.py**. Ini membuka file kosong di IDLE IDE untuk Python 3.
4. Masukkan kode berikut:

```
# GPIO example using an NC-SR04 ultrasonic rangefinder
# import the GPIO and time libraries
      import RPi.GPIO as GPIO
# Set the GPIO mode to BCM mode and disable warnings
      GPIO.setmode(GPIO.BCM)
      GPIO.setwarnings(False)
# Define pinbtn
      Pin = 20
      GPIO.setup(btnPin, GPIO.IN, pull_up_down = GPIO.PUD_UP)
# Begin while loop
      while True:
     btnVal = GPIO.input(btnPin)
     # If the pin is low, print to terminal
            if (btnVal == false):
            print('Button pressed')
```

5. Buka jendela terminal baru dan arahkan ke folder proyek Anda.
6. Ketik chmod +x gpio_button.py.
7. Untuk menjalankan kode, ketik sudo python3 gpio_button.py

Contoh Rangefinder Sonic

Untuk contoh ini, mari kita gunakan sensor ultrasonik HC-SR04 untuk menentukan jarak ke suatu objek. Anda akan membuat panggilan menjadi satu lingkaran yang memungkinkan kami mendapatkan pembacaan jarak yang konstan. Anda akan menggunakan perpustakaan yang digunakan pada contoh sebelumnya untuk mengakses pin GPIO. Latihan ini memperkenalkan Anda pada salah satu faktor utama yang harus diperhatikan dengan Pi

dan banyak perangkat lainnya: perbedaan tegangan antara sensor dan pin. Banyak sensor dirancang untuk bekerja pada 5 volt. Pi, bagaimanapun, menggunakan 3,3 volt dalam logikanya. Itu berarti semua pin I/O dirancang untuk bekerja dengan tegangan 3,3 volt. Menerapkan sinyal 5V ke salah satu pin ini dapat menyebabkan kerusakan parah pada Pi Anda. Pi memang menyediakan beberapa pin sumber 5V, tetapi kita perlu mengurangi sinyal balik menjadi 3,3 volt.

Menghubungkan Sirkuit

Kali ini, sirkuitnya sedikit lebih rumit. Betulkah. Perlu diingat bahwa sensor bekerja pada 5 volt. Pin GPIO Pi bekerja pada 3,3 volt. Memberi makan sinyal balik 5V ke pin 3.3V dapat merusak Pi. Agar hal itu tidak terjadi, mari tambahkan pembagi tegangan sederhana ke pin gema.

Mari kita lakukan beberapa matematika.

$$V_{out} = V_{in} * \frac{R1}{R1 + R2}$$

$$V_{out} = V_{in} * \frac{R2}{R2 + R1}$$

$$\frac{V_{out}}{V_{in}} = \frac{R2}{R1 + R2}$$

Kami memiliki 5 volt masuk dan ingin 3,3 volt keluar, dan kami menggunakan resistor 1kΩ sebagai bagian dari rangkaian. Jadi…

$$\frac{3.3}{5} = \frac{R2}{1000 + R2}$$

$$0.66 = \frac{R2}{1000 + R2}$$

$$0.66(1000 + R2) = R2$$

$$660 + 0.66R2 = R2$$

$$660 + 0.34R2$$

$$1941 = R2$$

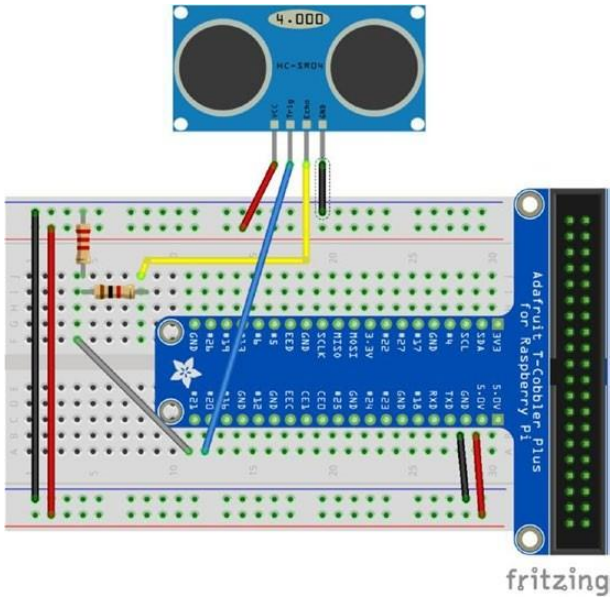
Berikut ini adalah daftar bagian:

- HC-SR04
- resistor 1kΩ
- resistor 2kΩ
  Anda dapat menggunakan dua resistor 1kΩ secara seri, atau resistor serupa yang lebih populer adalah 2.2kΩ. Itu yang akan kita gunakan.
- 4 jumper pria-wanita
- 4 jumper pria-ke-pria

Berikut pengaturannya.

1. Pasang T-Cobbler, seperti yang ditunjukkan pada Gambar 4-6. Satu baris pin harus

berada di kedua sisi celah di papan. Penempatannya terserah Anda; namun, saya biasanya memasangnya sehingga header kabel pita terlepas dari papan. Ini memungkinkan akses maksimum ke papan tempat memotong roti.



Gambar 4-6. Contoh tata letak rangkaian pengintai sonic

2. Pastikan jumper arde terpasang erat di antara pin arde dan rel arde.
3. Tambahkan jumper antara salah satu pin 5V dan power rail.
4. Gunakan jumper male-to-female untuk menghubungkan pin ground pada SR04 ke ground rail.
5. Hubungkan pin VCC atau 5V dari SR04 ke power rail.
6. Hubungkan pin trigonometri pada SR04 ke pin 20 T-Cobbler.
7. Hubungkan resistor 2kΩ dari rel 5-pin yang kosong ke rel ground.
8. Hubungkan resistor 1kΩ dari rel yang terhubung ke resistor 2kΩ ke rel 5-pin kosong lainnya.
9. Hubungkan jumper lain antara rel yang terhubung ke resistor 2kΩ dan pin 21 pada T-Cobbler.
10. Hubungkan pin gema SR04 ke rel tempat ujung lain dari resistor 1kΩ terhubung.

Itu melengkapi pengkabelan. Sekarang mari kita siapkan kodenya.

Menulis Kode

Pengintai ultrasonik HC-SR04 bekerja dengan mengukur waktu yang dibutuhkan pulsa ultrasonik untuk kembali ke sensor. Kami akan mengirimkan 10- mikrodetik pulsa dan kemudian mendengarkan pulsa untuk kembali. Dengan mengukur panjang pulsa yang dikembalikan, kita dapat menggunakan sedikit matematika untuk menghitung jarak dalam sentimeter. Jarak dihitung sebagai kecepatan × waktu. Diturunkan dari rumus kecepatan = jarak waktu. Di permukaan laut, suara merambat dengan kecepatan 343m per detik, atau 34.300cm per detik. Karena kita sebenarnya mengukur waktu yang dibutuhkan pulsa untuk mencapai target dan kembali, kita sebenarnya hanya membutuhkan setengah dari nilai itu.

Mari kita bekerja dengan rumus berikut:

$$\text{Jarak} = 17.150 \times \text{waktu}$$

Kode hanya mengirimkan pulsa 10µS, mengukur waktu yang diperlukan untuk kembali, menghitung perkiraan jarak dalam sentimeter, dan menampilkannya di jendela terminal.

Buat file Python 3 baru.
Jika menggunakan IDLE, lakukan hal berikut:

1. Buka IDLE untuk Python 3.
2. Klik **New**.
3. Simpan file sebagai `gpio_sr04.py` di folder proyek Anda. Jika menggunakan jendela terminal, lakukan hal berikut:

1. Navigasikan ke folder proyek Anda. Di Pi saya itu
   `$ cd ~/TRG-RasPi-Robot/code`
2. Ketik **touch gpio_sr04.py**.
3. Ketik **idle3 gpio_sr04.py**. Ini membuka file kosong di IDLE IDE untuk Python 3.
4. Masukkan kode berikut:

```
# GPIO example using an NC-SR04 ultrasonic rangefinder
# import the GPIO and time libraries
      import RPi.GPIO as GPIO
      import time
# Set the GPIO mode to BCM mode and disable warnings
      GPIO.setmode(GPIO.BCM)
      GPIO.setwarnings(False)
# Define pinstrig = 20
      echo = 21
      GPIO.setup(trig,GPIO.OUT)
      GPIO.setup(echo,GPIO.IN)
      print("Measuring distance")
# Begin while loop
      while True:
      # Set trigger pin low got 1/10 second
            GPIO.output(trig,False)
            time.sleep(0.1)
      # Send a 10uS pulse
            GPIO.output(trig,True)
            time.sleep(0.00001)
            GPIO.output(trig,False)
      # Get the start and end times of the return pulse
            while GPIO.input(echo)==0:
```

```
pulse_start = time.time()
while GPIO.input(echo)==1:
      pulse_end = time.time()
      pulse_duration = pulse_end - pulse_start
# Calculate the distance in centimeters
      distance = pulse_duration * 17150
      distance = round(distance, 2)
# Display the results. end = '\r' forces the output to the
same line
      print("Distance: " + str(distance) + "cm  ", end =
'\r')
```

5. Buka jendela terminal baru dan arahkan ke folder proyek Anda.
6. Ketik `chmod +x gpio_sr04.p.`
7. Untuk menjalankan kode, ketik **sudo python3 gpio_sr04.py**

4.8 RINGKASAN

Salah satu hal hebat tentang Raspberry Pi adalah header GPIO. Header 40-pin memungkinkan Anda untuk berinteraksi langsung dengan sensor dan perangkat lain. Selain GPIO sederhana yang biasa kami sambungkan ke LED, tombol, dan sensor ultrasonik, ada pin dengan fungsi khusus lainnya. Saya sarankan menjelajahi beberapa fungsi lain ini. Pin bertanda SCL, SDA, MISO, dan MOSI adalah koneksi serial yang memungkinkan Anda menggunakan sensor tingkat lanjut, seperti akselerometer dan GPS.

Saat bekerja dengan header GPIO, ada beberapa hal yang perlu Anda ingat.

- Untuk menjalankan skrip Anda, pertama-tama buat file tersebut dapat dieksekusi dengan menggunakan `chmod +x <filename>`
- Setiap kali Anda menjalankan skrip yang menggunakan pin GPIO, Anda harus menggunakan `sudo`.
- Perhatikan voltase yang digunakan oleh sensor Anda.
- Meskipun header dapat mensuplai 5 volt untuk perangkat, pin logikanya adalah 3,3 volt. Anda akan merusak Raspberry Pi Anda jika Anda tidak mengurangi sinyal yang datang dari sensor.
- Sirkuit pemisah tegangan seperti yang dibuat untuk sensor ultrasonic dapat digunakan untuk mengurangi sinyal 5V dari sensor menjadi 3,3 volt.
- Papan premade yang disebut pemindah level logika mengurangi tegangan.
- Raspberry Pi tidak memiliki pin analog yang berguna secara fungsional.
- Hanya memiliki dua saluran PWM. Masing-masing terhubung ke dua pin, sehingga mungkin terlihat seperti memiliki empat pin PWM yang dapat digunakan, tetapi sebenarnya tidak.

Di bab berikutnya, kami menghubungkan papan Arduino ke Raspberry Pi kami. Arduino adalah mikrokontroler yang dirancang untuk IO. Hanya itu yang dilakukannya, tetapi ia melakukannya dengan baik. Dengan menggabungkan kedua papan ini, kami tidak hanya mengatasi kekurangan Pi, tetapi juga menambah manfaat lainnya.

BAB 5
RASPBERRY PI DAN ARDUINO

Dalam Bab 4, kami menggunakan pin GPIO pada Raspberry Pi untuk berinteraksi dengan LED dan sensor ultrasonik. Sering kali, ini cukup untuk melakukan apa yang ingin Anda lakukan. Namun, saya juga membahas beberapa kekurangan dari Raspberry Pi GPIO dan kemungkinan kebutuhan untuk memperluas kemampuan Pi untuk mengatasi kekurangan ini. Dalam bab ini, kami memperkenalkan mikrokontroler ke robot kami. Mikrokontroler adalah perangkat, biasanya dalam bentuk chip, dirancang untuk bekerja secara langsung dengan komponen lain melalui pin input dan output. Setiap pin terpasang ke sirkuit mikrokontroler dan melayani tujuan tertentu.

Karena pin terhubung langsung ke bagian dalam mikrokontroler yang sensitif, sirkuit tambahan umumnya diperlukan untuk membuatnya aman untuk bekerja dengannya. Banyak produsen menyediakan papan evaluasi untuk memungkinkan pengembang membangun prototipe dan perangkat bukti konsep dengan cepat. Salah satu papan tersebut sebenarnya dikembangkan oleh pengembang, bukan oleh produsen chip, dan tersedia untuk umum. Karena kemudahan penggunaan, dokumentasi yang cukup, dan dukungan komunitas yang luar biasa, perangkat ini dengan cepat menjadi favorit komunitas hobi. Saya berbicara tentang Arduino, tentu saja.

Kami membahas banyak informasi tentang Arduino: cara menginstal perangkat lunak, menulis program (disebut sketsa), dan memuat program tersebut ke papan Arduino. Kami juga membahas cara membuat papan Raspberry Pi dan Arduino Anda saling berbicara. Ini menambahkan lebih banyak kemampuan secara eksponensial ke robot Anda. Namun sebelum kita masuk ke Arduino, mari kita ulas beberapa kekurangan dari Raspberry Pi.

5.1 GPIO RASPBERRY PI DALAM ULASAN

Secara khusus, mari kita bicara tentang kurangnya pin analog dan modulasi lebar pulsa (PWM) yang memadai.

5.2 PEMROSESAN REAL-TIME ATAU NEAR REAL-TIME

Pemrosesan waktu nyata adalah kemampuan sistem untuk berinteraksi langsung dengan GPIO dan perangkat eksternal. Sangat penting untuk aplikasi CNC atau aplikasi lain di mana respons segera diperlukan. Dalam istilah robotika, perlu untuk sistem loop tertutup di mana respons langsung terhadap rangsangan diperlukan. Contoh yang baik adalah detektor tepi untuk robot seluler. Anda ingin robot berhenti bergerak sebelum meluncur sendiri di atas tebing atau di tepi meja. Meluangkan waktu yang diperlukan untuk memproses melalui banyak lapisan abstraksi dari sistem operasi untuk mencapai logika untuk menentukan berhenti dan kemudian

mengirim sinyal melalui banyak lapisan ke pengontrol motor dapat menjadi bencana besar. Dan, jika OS menunda operasi atau hang, robot akan dengan senang hati jatuh ke kehancurannya, tidak pernah lebih bijaksana. Sebaliknya, Anda ingin robot Anda segera berhenti. Meskipun ada varian Linux yang memfasilitasi pemrosesan hampir real-time, ini adalah sistem operasi khusus dan instalasi Raspbian yang kami gunakan bukan salah satunya.

5.3 MASUKAN ANALOG

Kami telah melihat input digital bekerja pada Pi. Faktanya, kami menggunakan pengintai ultrasonik untuk mendeteksi jangkauan ketika pin digital dihidupkan dan dimatikan (menjadi tinggi, lalu rendah). Dengan sedikit matematika, kami dapat mengubah sinyal itu menjadi data yang berguna. Itu adalah sinyal digital; itu hanya mendeteksi ketika pin memiliki tegangan tinggi, dan kemudian ketika pin yang sama memiliki tegangan rendah. Ada banyak jenis sinyal analog; tidak hanya tinggi atau rendah, putih atau hitam, atau hidup atau mati, tetapi juga rentang nilai atau nuansa abu-abu untuk menggunakan analogi hitam/putih. Ini sangat berguna ketika Anda menggunakan sensor yang mengukur intensitas atau tingkat sesuatu. Sensor cahaya yang menggunakan fotoresistor adalah salah satu contohnya. Saat intensitas cahaya berubah, begitu juga resistansi, dan karena itu tegangan, pada sensor. Perangkat yang disebut analog-to-digital converter (ADC) mengubah sinyal analog tersebut menjadi nilai digital yang dapat digunakan oleh program.

Raspberry Pi memiliki satu pin analog. Ini tidak terlalu berguna, terutama jika dikaitkan dengan fungsi lain yang mungkin digunakan board—dalam hal ini, komunikasi serial. Jika kami mendedikasikan pin ke input analog, kami tidak akan dapat menggunakan saluran serial itu. Bahkan jika kami tidak berencana untuk menggunakan saluran serial tertentu, satu input analog memiliki penggunaan yang sangat terbatas.

5.4 KELUARAN ANALOG

Output analog pada dasarnya mirip dengan input analog. Dengan latihan LED yang kami lakukan sebelumnya, kami menggunakan sinyal digital untuk menyalakan dan mematikan LED. Analog memungkinkan kita untuk mengubah kecerahan, atau intensitas LED. Namun, sistem digital, seperti komputer atau mikroprosesor, tidak dapat membuat sinyal analog yang sebenarnya. Ini menyesuaikan frekuensi dan durasi sinyal digital. Durasi sinyal digital disebut sebagai pulsa. Menyesuaikan seberapa sering pulsa aktif dalam periode waktu tertentu, dan panjang pulsa itu, disebut modulasi lebar pulsa, atau PWM. Ketika kami mengukur sinyal dari pengintai ultrasonik, kami sebenarnya mengukur pulsa yang dikembalikan dari perangkat.

Raspberry Pi memiliki empat pin PWM yang tersedia. Namun, keempat pin tersebut hanya terhubung ke dua proses PWM. Jadi, ini berarti kami hanya memiliki dua saluran PWM yang tersedia untuk digunakan. Dan sekali lagi, ini tidak berguna seperti yang kita inginkan. Dengan prosesor real-time, kami dapat mensimulasikan PWM dengan perangkat lunak. Namun,

seperti yang dibahas sebelumnya, Raspberry Pi bukanlah sistem waktu nyata. Jadi, kita perlu mencari solusi lain jika kita menginginkan lebih dari dua saluran PWM.

5.5 ARDUINO UNTUK MENYELAMATKAN

Untungnya, ada kelas perangkat yang dirancang, secara khusus, untuk mengelola input dan output dari berbagai jenis, secara real time. Ini adalah mikroprosesor. Ada banyak jenis mikroprosesor di luar sana. Beberapa yang lebih umum, dan mudah digunakan, adalah AVR ATTiny dan prosesor ATMega.

Namun, ini adalah chip, dan kecuali Anda terbiasa bekerja dengannya, chip tersebut bisa sulit diakses dan digunakan. Untuk membuat perangkat ini lebih mudah digunakan, pabrikan membuat apa yang dikenal sebagai papan pengembangan. Papan ini menghubungkan pin pada chip ke header yang lebih mudah diakses untuk pembuatan prototipe. Mereka juga menambahkan elektronik yang diperlukan untuk menggunakan pin, seperti pengatur tegangan, resistor pull-up, tutup filter, dioda, dan sebagainya. Jadi, pada akhirnya, yang harus Anda lakukan adalah menghubungkan elektronik spesifik Anda ke perangkat dan membuat prototipe produk Anda. Beberapa tahun yang lalu, sekelompok insinyur di Italia berkumpul dan melakukan sesuatu yang belum pernah terjadi sebelumnya. Mereka mengembangkan papan pengembangan mereka sendiri di sekitar chip AVR ATMega, membuat desain terbuka untuk umum (perangkat keras terbuka), dan kemudian memasarkannya ke penghobi dan pelajar. Mereka menyebut papan ini Arduino. Gambar 5-1 menunjukkan Arduino Uno yang khas. Saya yakin itu memiliki konsekuensi yang diinginkan untuk menjadi standar de facto di komunitas hobi dan pembuat.

Gambar 5-1. Arduino Uno

Kami akan menggunakan Arduino Uno dengan Raspberry Pi kami. Mengapa? Pertama, ini adalah prosesor waktu nyata. Arduino berkomunikasi langsung dengan pin dan periferal yang terpasang. Tidak ada lag karena OS atau lapisan program abstraksi. Kedua, ini menyediakan lebih

banyak pin untuk digunakan. Diantaranya adalah enam pin analog dan enam pin PWM berbasis hardware yang kami tambahkan. Ini "berbasis perangkat keras" karena papannya real-time dan kami dapat mensimulasikan sinyal PWM (melalui perangkat lunak) pada salah satu pin (ada 20, omong-omong). Dan itu hanya Arduino Uno. Ada versi papan Arduino yang lebih besar yang disebut Mega. Mega memiliki 54 pin digital dan 16 pin analog. Itu adalah total 70 pin kebaikan IO.

Arduino adalah perangkat keras terbuka, yang berarti bahwa desainnya tersedia bagi siapa saja untuk membangunnya. Dengan demikian, Anda menemukan banyak versi berbeda dari banyak produsen berbeda pada banyak titik harga. Ini adalah contoh utama Anda mendapatkan apa yang Anda bayar. Jika Anda baru memulai, saya sarankan menghabiskan sedikit lebih banyak untuk mendapatkan papan yang lebih andal. Kemudian, saat Anda mengembangkan pemahaman yang lebih baik, dan toleransi yang lebih tinggi untuk pemecahan masalah, Anda dapat bereksperimen dengan papan yang lebih murah.

5.6 MENGGUNAKAN ARDUINO

Arduino sangat mudah diprogram dan digunakan. Banyak orang terintimidasi oleh prospek bekerja dengan elektronik dan perangkat keras pemrograman. Tapi, ada alasan mengapa begitu banyak orang memulai robotika dan IoT dengan Arduino. Menghubungkan perangkat ke Arduino sangat mudah, terutama dengan penggunaan add-on yang disebut perisai. Memprogram Arduino juga sangat mudah. Arduino menyediakan antarmuka untuk memprogram papan yang disebut, cukup sederhana, Arduino. Atau lebih tepatnya, itu adalah Arduino IDE (lingkungan pengembangan terintegrasi). Arduino IDE menggunakan cita rasa pemrograman C yang juga disebut Arduino. Seperti yang Anda lihat, perangkat keras, perangkat lunak, dan lingkungan pengembangan secara konseptual adalah hal yang sama. Ketika Anda berbicara tentang pemrograman Arduino tidak ada perbedaan antara perangkat lunak dan perangkat keras karena satu-satunya tujuan perangkat lunak adalah untuk berinteraksi dengan perangkat keras.

Sepanjang bab ini, Anda harus menginstal Arduino IDE dan Arduino terhubung ke komputer Anda. Diasumsikan bahwa instruksi dan latihan instalasi dijalankan di Raspberry Pi Anda, tetapi sejujurnya, instalasi di komputer lain sama mudahnya. Jadi, jika Anda lebih nyaman mengerjakan sesuatu selain Pi, atau Anda tidak ingin melakukan remote ke satu Pi, Anda dapat melakukan semua latihan di PC atau laptop Anda.

5.7 MEMASANG ARDUINO IDE

Sebelum kami menghubungkan Arduino ke Raspberry Pi kami, kami ingin menginstal perangkat lunak dan driver. Untungnya, ini sangat mudah. Menginstal Arduino IDE juga menginstal driver apa pun yang diperlukan untuk bekerja dengan Pi.

Memasang Arduino IDE

1. Buka jendela terminal.
2. Ketik **sudo apt-get install Arduino.**
3. Jawab ya untuk setiap petunjuk.
4. Ambil minum. Ini mungkin memakan waktu yang sangat lama.

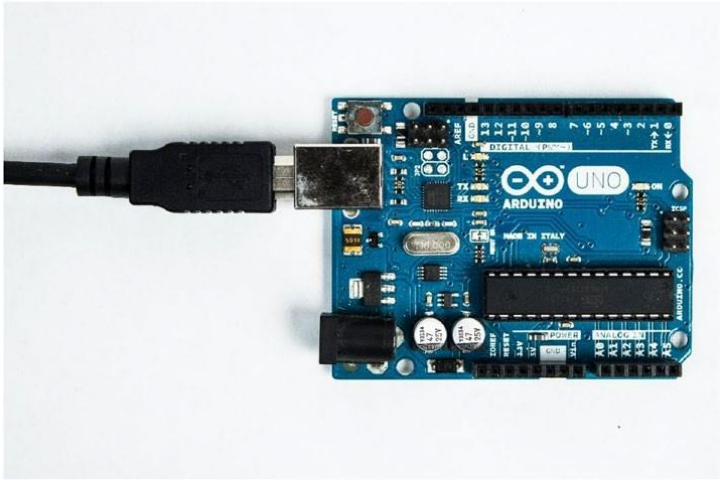
Ketika proses instalasi selesai, ia menambahkan Arduino IDE ke menu pemrograman Anda.

5.8 MENGHUBUNGKAN ARDUINO

Ketika saya awalnya menguraikan bagian buku ini, maksud saya adalah menyediakan banyak cara untuk menghubungkan Arduino ke Raspberry Pi. Namun, untuk menggunakan apa pun selain port USB memperkenalkan lapisan kompleksitas lain dan detail Linux yang berada di luar cakupan pengantar ini. Ini pada dasarnya melibatkan memberi tahu Pi bahwa Anda mengaktifkan pin UART, dan kemudian menonaktifkan sejumlah fungsi asli yang menggunakan saluran ini. Ini adalah proses yang tidak perlu untuk dilalui, terutama karena ada empat port USB yang siap digunakan, dan jika Anda membutuhkan lebih banyak, Anda selalu dapat menambahkan hub USB.

Jadi, kita akan menggunakan koneksi USB sehingga kita bisa fokus pada pengenalan Arduino yang berkaitan dengan Pi.

Untuk menghubungkan Arduino, yang harus kita lakukan adalah menghubungkan kabel USB dari Raspberry Pi ke Arduino, seperti yang digambarkan pada Gambar 5-2. Tergantung pada produsen papan, Anda mungkin memerlukan kabel USB yang berbeda. Karena saya menggunakan Uno asli, saya menggunakan kabel USB A-to-B. Beberapa orang menggunakan kabel mini USB, dan yang lain menggunakan mikro USB.



Gambar 5-2. Kabel USB A ke B terhubung ke Arduino Uno

Itu dia. Karena papan Arduino ditenagai oleh Pi Anda melalui kabel USB, Anda tidak perlu menambahkan daya eksternal. Anda hampir siap untuk mulai menggunakan Arduino Anda. Selanjutnya, kita akan menguji Arduino Anda dengan program berkedip di mana-mana. Tapi

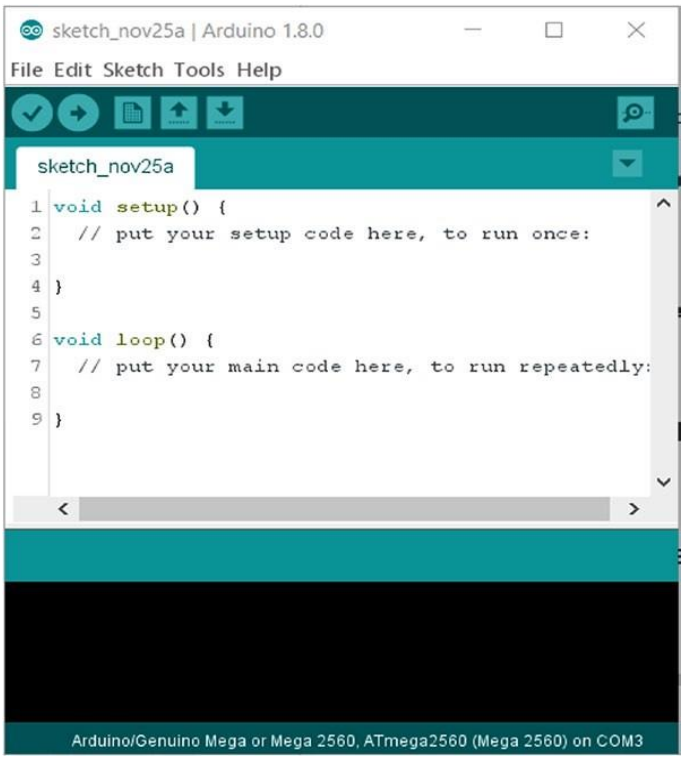
pertama-tama, mari kita lihat antarmukanya.

5.9 PEMROGRAMAN ARDUINO

Seperti yang saya katakan sebelumnya, memprogram Arduino sangat mudah. Namun, karena kita baru saja menghabiskan banyak waktu untuk mempelajari Python, sangat penting untuk memahami beberapa perbedaannya. Kami akan mulai dengan melihat antarmuka dan beberapa trik untuk menggunakannya. Kemudian kita akan menulis program kecil untuk mengilustrasikan anatomi dan sintaks bahasa. Semua ini untuk mempersiapkan Anda untuk bagian selanjutnya, di mana kita melihat lebih dalam bahasa pemrograman Arduino.

Arduino Ide

Saat pertama kali Anda membuka Arduino IDE, Anda akan disajikan dengan antarmuka yang sangat sederhana (lihat Gambar 5-3). Pengembang mengadopsi antarmuka bahasa Pemrograman dan IDE ketika mereka mengembangkan Arduino. Jika Anda pernah melakukan pengkodean di masa lalu, antarmuka ini akan tampak kekurangan fitur. Itu bertujuan dan sedikit menyesatkan.



Gambar 5-3. Arduino IDE

Meskipun antarmukanya sederhana, IDE ini ternyata sangat kuat. Yang terpenting, ini menyediakan kompilasi silang yang diperlukan untuk membuat kode Anda, yang ditulis di mesin Linux, Windows, atau Apple, untuk bekerja pada prosesor AVC yang jauh lebih sederhana.
Mari kita telusuri beberapa fitur dan operasi utama di Arduino IDE.

Ikon dan Menu

Menjadi Arduino dan berbeda, ikon di bilah alat di bagian atas antarmuka tidak seperti biasanya. Melihat Gambar 5-4 dan bergerak dari kiri ke kanan, ikon adalah kompilasi, upload, sketsa baru, buka, simpan, dan ke kanan adalah monitor serial.

Gambar 5-4. Bilah alat Arduino IDE

Dua ikon pertama sangat penting. Kompilasi memberitahu IDE untuk memproses kode Anda dan membuatnya siap untuk dimuat ke papan Arduino. Ini berjalan melalui kode Anda dan mencoba membangun program tingkat mesin akhir. Saat ini, ini mengidentifikasi kesalahan yang mungkin Anda masukkan. Arduino tidak menyediakan fungsionalitas debugging apa pun, jadi Anda sedikit bergantung pada fungsi kompilasi. Upload mengkompilasi sketsa dan kemudian mengunggahnya ke papan. Karena fungsi unggah menjalankan kompiler terlebih dahulu, Anda mendapatkan aktivitas kompilasi yang sama dengan fungsi kompilasi, tetapi, pada akhir proses, ia mencoba memuat kode yang dikompilasi ke papan Anda. Karena prosesor AVR hanya dapat menyimpan dan menjalankan satu program pada satu waktu, setiap kali Anda mengunggah ke papan Arduino, Anda menimpa apa pun yang sedang ada di sana. Ini tidak selalu diinginkan. Terkadang Anda mengkompilasi kode sebentar-sebentar untuk memeriksa sintaks dan memastikannya benar. Anda tidak akan selalu ingin memuat langkah-langkah perantara ini ke papan.

Namun, pada akhirnya, Anda perlu mengunggah sketsa Anda untuk melihat sesuatu terjadi. Mengkompilasi sketsa memastikan bahwa Anda memiliki kode yang berfungsi. Apakah kode tersebut melakukan apa yang Anda inginkan atau tidak adalah cerita lain. Anda tidak akan tahu ini sampai diunggah.

Membuat Sketsa Baru

Anda dapat membuat sketsa baru dengan mengeklik ikon Sketsa Baru di bilah alat atau dengan mengeklik File ➤ New dari menu. Membuat sketsa baru selalu membuka instance baru dari IDE. Apa pun yang Anda kerjakan di jendela sebelumnya masih ada di sana. Pertama kali Anda membuka Arduino IDE, Anda disajikan dengan kerangka sketsa baru. Ini juga yang Anda lihat saat membuatnya nanti. Setiap sketsa Arduino mengandung elemen-elemen ini. Operasi Sketsa Baru selalu mengisi IDE dengan kerangka kerja ini. Anda akan melihat elemen-elemen ini ketika kami menulis sketsa pertama kami.

Menyimpan Sketsa

Sebelum Anda dapat mengkompilasi atau menjalankan sketsa, Anda harus menyimpannya. Anda dapat menyimpan sketsa kapan saja, hanya saja harus dilakukan sebelum Anda dapat mengkompilasi atau mengunggahnya. Untuk menyimpan sketsa, klik ikon Simpan atau pilih File ➤ Save dari menu. Saat sketsa pertama kali disimpan, sistem secara otomatis

membuat folder proyek untuknya. Di sinilah file kode (dengan ekstensi ino) disimpan. File lain yang dibuat untuk proyek juga disimpan di folder ini. Ini penting ketika Anda bekerja dengan program yang lebih besar dan lebih kompleks, atau ketika Anda mulai memecah sketsa Anda menjadi beberapa tab di IDE.

Membuka Sketsa Keluar

Secara default, saat Anda membuka IDE, sketsa terakhir yang Anda kerjakan akan terbuka secara otomatis. Ini nyaman ketika Anda mengerjakan proyek yang sama untuk sementara waktu. Jika Anda perlu membuka sketsa lain, klik ikon Buka Sketsa di bilah menu atau pilih File ➤ Open. Atau, Anda juga dapat memilih File ➤ Open Terbaru. Ini mencantumkan beberapa sketsa terakhir yang Anda buka. Memilih salah satu file ini akan membukanya di instance baru IDE.

Pemilihan Papan Dan Pelabuhan

Sesuatu yang sangat penting untuk kompilasi dan pemuatan sketsa yang tepat adalah pemilihan papan dan port yang sesuai. Pemilihan papan dan port dilakukan dari menu Alat. Memilih papan memberi tahu kompiler versi Arduino mana yang Anda gunakan. Saat Anda berkembang dalam pengalaman Arduino, robotika, dan/atau IoT, Anda mungkin akan menggunakan papan yang berbeda. Salah satu hal hebat tentang Arduino IDE adalah fleksibilitasnya. Anda menemukan diri Anda menggunakan lingkungan yang akrab dan nyaman untuk memprogram banyak sekali papan yang berbeda oleh produsen yang berbeda. Ini memungkinkan adopsi Arduino sebagai standar de facto di komunitas pembuat.

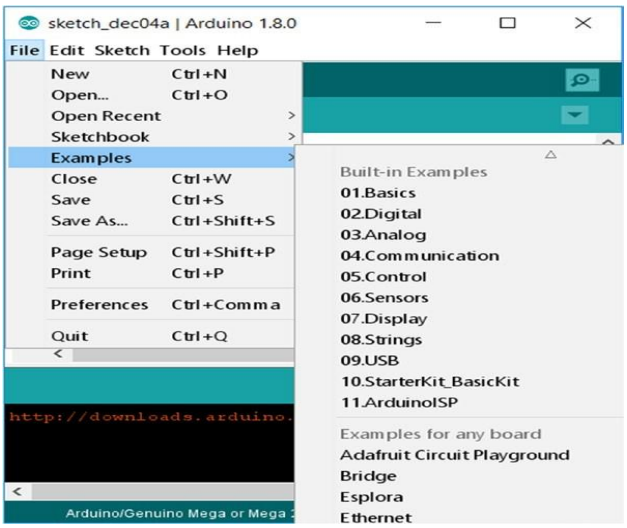
Untuk memilih papan dan port untuk robot Anda, pastikan Arduino Anda terhubung melalui USB, dan Arduino IDE diinstal dan dibuka.

1. Pilih **Tools ➤ Board** dari menu.
2. Pilih **Arduino/Genuino Uno** dari daftar papan yang tersedia.
3. Pilih **Tool ➤ Port** dari menu.
   Seharusnya ada satu entri dalam daftar yang mengatakan sesuatu seperti **Arduino/Genuino Uno on TTYAMC0**.
4. Pilih entri ini.

Pada titik ini, Arduino IDE harus siap untuk mengkompilasi dan memuat sketsa ke papan Anda. Kami akan menulis sketsa pertama kami untuk mengujinya segera.

Kecurangan Dengan Contoh

Saat Anda menginstal Arduino IDE, Anda juga menginstal kumpulan contoh sketsa (lihat Gambar 5-5). Ini adalah referensi yang sangat baik untuk mempelajari cara Anda di sekitar pengkodean Arduino. Saat Anda belajar, lihat sketsa ini untuk fungsionalitas yang mirip dengan apa yang ingin Anda capai.

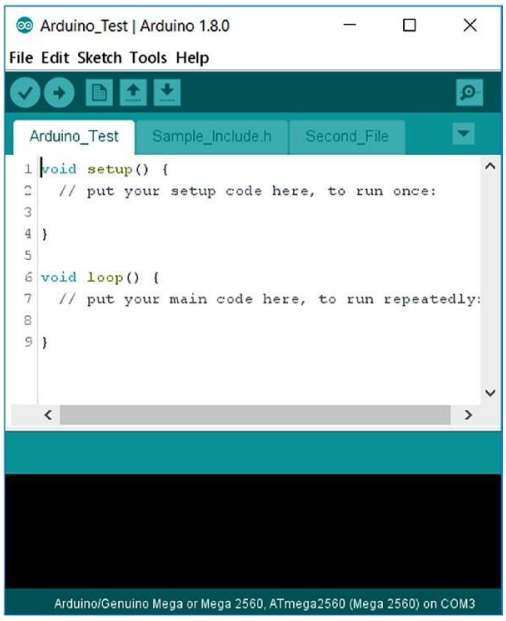


Gambar 5-5. Daftar kode contoh yang disertakan dengan instalasi dasar

Untuk melihat daftar contoh atau membukanya, klik File Contoh. Saat Anda menambahkan lebih banyak pustaka (seperti untuk sensor dan perangkat lain), Anda menambahkan ke daftar contoh ini. Jadi saat Anda memperluas kemampuan Anda sendiri, serta kemampuan robot Anda, pastikan untuk meninjau kembali contoh-contohnya.

Menggunakan Tab dan Banyak File

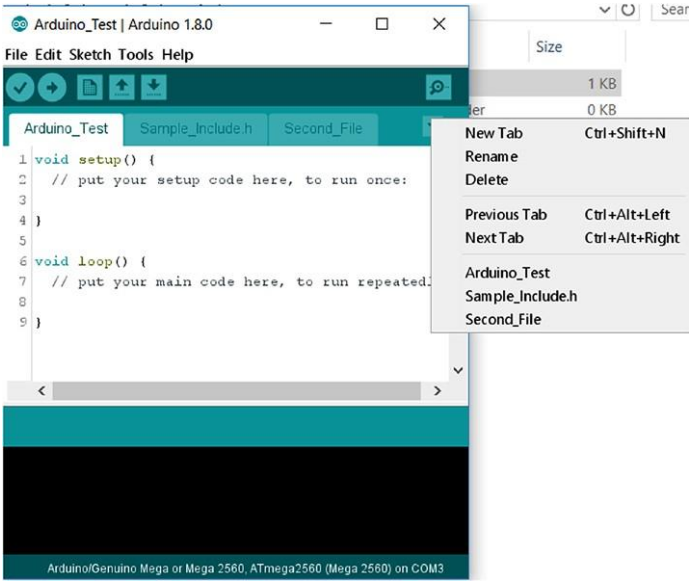
Ketika saya membahas menyimpan sketsa sebelumnya, mungkin tampak sedikit aneh bahwa folder proyek dibuat untuk satu file. Alasan untuk ini adalah proyek dapat terdiri dari lebih dari satu file. Anda dapat membuat beberapa file Arduino untuk sebuah proyek atau Anda mungkin ingin menyimpan file yang disertakan bersama dengan proyek Anda. Gambar 5-6 menunjukkan Arduino IDE dengan tiga tab terbuka.



Gambar 5-6. Arduino IDE dengan banyak tab

Saat Anda memiliki beberapa file kode dalam folder proyek, masing-masing akan muncul sebagai tab di Arduino IDE saat Anda membuka file di proyek itu. Ini memungkinkan Anda untuk dengan mudah menavigasi antara file saat Anda bekerja. Saat bekerja dengan tab dan banyak file, ada beberapa hal yang perlu diingat. Kode di tab yang dibuat melalui IDE dan disimpan sebagai file INO ditambahkan ke akhir file INO utama. Ini berarti setiap fungsi yang Anda buat di tab ini tersedia untuk digunakan di salah satu tab ini.

Namun, tab untuk file yang tidak dibuat di IDE, seperti tab untuk file yang disertakan, tidak tersedia hingga Anda menyertakan file dalam kode Anda. Ini bisa nyaman dan membuat frustrasi karena Anda perlu melacak dari file mana fungsi tertentu berasal. Saya akan menyentuh sedikit lebih banyak tentang memasukkan file nanti dalam bab ini ketika kita meninjau pengkodean di Arduino. Gambar 5-7 menunjukkan menu manajemen tab.



Gambar 5-7. Menu manajemen tab

Anda dapat membuat tab baru untuk membantu mengatur kode Anda. Saat Anda membuat tab baru, tab itu disimpan sebagai file baru di dalam file proyek Anda.

1. Buka Arduino IDE dan mulai file baru.
2. Simpan file untuk membuat file proyek baru.
3. Klik panah di bilah tab IDE.
4. Klik **New Tab**.
5. Pada dialog yang terbuka, masukkan nama untuk tab Anda. Ingatlah bahwa ini adalah nama file baru di folder proyek Anda.
6. Simpan filenya. Semua tab yang belum disimpan juga disimpan.

Setelah tab disimpan, Arduino secara otomatis membuat file baru untuk menyimpan kode di tab.

5.10 SKETSA

Program untuk Arduino disebut sketsa. Idenya adalah Anda hanya membuat sketsa kode seperti Anda membuat sketsa ide di serbet restoran. Dan, sejujurnya, terkadang memang seperti itu. Anda sedang menulis sketsa Arduino dalam bahasa yang disebut Pemrograman. Ini adalah versi tipis dari bahasa pemrograman C yang dirancang untuk membuat pengkodean lebih mudah. Arduino sebenarnya menggunakan versi modifikasi dari Pemrograman yang dibuat untuk papan Arduino. Ini pada dasarnya adalah serangkaian instruksi yang dikurangi yang dapat dijalankan oleh prosesor AVR.

Seperti Python, Anda dapat menambahkan perpustakaan saat Anda menambahkan fungsionalitas dan kompleksitas. Di C, kami menggunakan direktif include. Ini melayani tujuan yang sama seperti perintah impor di Python. Kita akan melihatnya sebentar lagi ketika kita melakukan komunikasi antara kedua dewan.

Halo Arduino

Untuk memahami perbedaan antara pemrograman Arduino dan Python, kami akan menulis program sederhana. Seperti bab tentang GPIO, program pertama adalah versi perangkat keras Hello World LED yang berkedip. Setelah Anda memuat program, Anda akan belajar lebih banyak tentang pemrograman, strukturnya, dan cara bekerja dengannya. Dalam bab GPIO, kami membangun sirkuit kecil dengan LED. Arduino, bagaimanapun, memiliki LED yang terpasang di papan yang tersedia untuk kita gunakan, jadi kita tidak perlu membongkar papan tempat memotong roti dulu. LED terpasang pada pin 13 pada UNO; mungkin berbeda pada versi lain.

1. Buka Arduino IDE dari menu pemrograman Anda.
2. Pastikan papan terhubung dan terdeteksi.
3. Pada menu Arduino IDE, buka Alat dan arahkan kursor ke **Board**. Anda akan melihat **Arduino Uno** dipilih.
4. Sekarang arahkan kursor ke Serial Port. Itu harus mengatakan sesuatu seperti /dev/ttyUSB0. Milik Anda mungkin berbeda jika Anda Pi menetapkan port yang berbeda. Intinya adalah bahwa ada sesuatu di sana dan itu diperiksa.
5. Tutup menu Tools dengan mengklik di suatu tempat di luar menu.
6. Masukkan kode berikut:

```
int ledPin = 13;
void setup() {
      pinMode(ledPin, OUTPUT);
}
void loop() {
      digitalWrite(ledPin, HIGH);
      delay(1000);
      digitalWrite(ledPin, LOW);
```

```
        delay(1000);
    }
```

7. Simpan file sebagai blink_test.
8. Klik ikon kotak centang untuk mengkompilasi sketsa.
9. Jika Anda mendapatkan kesalahan, pastikan Anda telah memasukkan kode dengan benar. Ingat, tidak seperti Python, Anda harus mengakhiri setiap baris dengan titik koma. Dan seperti Python, kapitalisasi penting.
10. Setelah semuanya terkompilasi dengan benar, klik ikon panah (menunjuk ke kanan). Ini mengunggah sketsa ke Arduino.

Tunggu beberapa detik saat sedang diunggah. Setelah itu, Anda akan melihat LED yang terhubung ke pin 13 berkedip. Selamat, Anda baru saja menyelesaikan program Arduino pertama Anda. Selain itu, Anda melakukannya dari Pi Anda.

Anatomi Sketsa

Sketsa yang baru saja kita tulis bukanlah yang paling rumit, tetapi menggambarkan struktur dasar sketsa Arduino.

```
int ledPin = 13;
```

Kami mulai dengan membuat variabel integer, yang disebut `ledPin`, dan memberinya nomor 13. Merupakan kebiasaan yang baik untuk memberi nama variabel yang bermakna, bahkan ketika programnya pendek dan hanya memiliki satu variabel.

```
void setup() {
```

Kami kemudian membuat fungsi yang disebut `setup()`. Fungsi ini dan fungsi `loop()` ada di setiap sketsa Arduino. Fungsi pengaturan adalah tempat Anda meletakkan kode persiapan, seperti membuka port serial atau, seperti yang kami lakukan di sketsa ini, tentukan bagaimana kita menggunakan pin. Fungsi setup berjalan sekali, di awal program.

```
pinMode(ledPin, OUTPUT);
```

Dalam fungsi pengaturan, kami memiliki satu perintah. Fungsi `pinMode` memberitahu compiler bagaimana sebuah pin digunakan. Dalam hal ini, kita mendeklarasikan `ledPin` (dengan nilai 13) sebagai pin output. Ini memberi tahu kompiler bahwa kami mengirim sinyal, dan kami tidak mengharapkan untuk menerima sinyal melalui pin ini. Kami kemudian menutup fungsi setup dengan tanda kurung penutup sebelum memulai fungsi `loop` kami.

```
void loop() {
```

Fungsi `loop` dijalankan terus menerus dan berulang-ulang sampai daya dilepas dari board, atau board direset. Ini setara dengan while true: perintah dengan Python. Kode apa pun dalam fungsi `loop` diulang secepat yang dapat dikelola oleh prosesor.

```
digitalWrite(ledPin, HIGH);
```

Dalam fungsi loop, kami memiliki kode untuk mengedipkan LED kami. Kita mulai dengan menyetel pin ke status tinggi dengan fungsi `digitalWrite`. Sekali lagi, kami memberikannya ledPin dan status yang ingin kami atur—dalam hal ini, TINGGI.

```
delay(1000);
```

Baris berikutnya menambahkan 1.000 milidetik, atau satu detik, penundaan sebelum menjalankan perintah berikutnya.

```
digitalWrite(ledPin, LOW);
```

Setelah penundaan satu detik, kami menyetel pin ke status rendah menggunakan perintah yang sama yang digunakan untuk menyetelnya tinggi, `digitalWrite`. Namun kali ini, kita melewatinya dengan konstan LOW.

```
delay(1000);
```

Sekali lagi, kami memperkenalkan penundaan satu detik. Karena ini adalah perintah terakhir dalam fungsi loop, setelah penundaan kita kembali ke awal fungsi loop. Ini berlanjut sampai kita mencabut Arduino atau mengunggah sketsa lain.

5.11 PENGANTAR SINGKAT BAHASA ARDUINO

Seperti yang telah dibahas sebelumnya, bahasa pemrograman Arduino berasal dari bahasa Pemrograman. Pemrograman, pada gilirannya, berakar pada C. Jika Anda terbiasa dengan pengkodean dalam C, Arduino mudah digunakan. Banyak fungsi, sintaks, dan pintasan berfungsi dengan baik di Arduino seperti di C. Untuk sisanya, Anda menangkap cukup cepat. Perlu diingat, Arduino bukan Python dan berperilaku jauh berbeda ketika Anda bekerja dengannya. Misalnya, Arduino jauh lebih tidak peduli dengan ruang putih dan pemformatan daripada Python, di mana lekukan digunakan untuk menunjukkan blok kode. Di C, blok kode didefinisikan menggunakan kurung kurawal {}. Karena itu, Anda tidak dapat mengabaikan ruang putih secara bersamaan. Ruang ekstra di awal baris dapat menyebabkan frustrasi tanpa akhir.

Perbedaan utama lainnya yang membuat frustrasi pemula dan programmer berpengalaman, adalah penghentian baris. Dengan Python, Anda cukup pindah ke baris berikutnya, tidak perlu terminator. Namun, di Arduino dan C, garis diakhiri dengan titik koma. Jika titik koma tidak ada di mana kompiler mengharapkannya, Anda mendapatkan kesalahan. Ini adalah kesalahan paling umum yang dilakukan pemula. Jika kode Anda tidak dapat dikompilasi, hal pertama yang harus dicari adalah titik koma yang hilang. Satu hal yang dibagikan antara Python dan Arduino adalah sensitivitas huruf. Penting untuk mengingat hal-hal penting. `intPin` tidak sama dengan `intpin`. Ini adalah hal kedua yang harus dicari jika kode Anda tidak dikompilasi dengan benar atau berperilaku seperti yang diharapkan.

5.12 TERMASUK FILE LAINNYA

Sama seperti Python, ada kalanya Anda perlu menyertakan file atau pustaka lain. Ini kemungkinan besar ketika Anda menambahkan sensor, motor, atau perangkat lain ke Arduino

dan perlu menambahkan perpustakaan perangkat ke kode Anda. Arduino menggunakan metode C dan C++ untuk menambahkan kode dari file eksternal melalui direktif `#include`. Baris berikut termasuk perpustakaan servo standar:

```
#include <Servo.h>
```

Seperti semua arahan, include memiliki sintaks yang sedikit berbeda. Perhatikan bahwa tidak ada titik koma di akhir baris ini. Titik koma menyebabkan kesalahan, dan kode tidak akan dikompilasi. Juga, kata kunci `include` didahului oleh # (hash).

5.13 VARIABEL DAN TIPE DATA

Seperti Python, Arduino memiliki semua tipe data umum, meskipun mereka mungkin bertindak sedikit berbeda. Salah satu perbedaan terbesar antara Python dan Arduino adalah Anda harus mendeklarasikan variabel Anda sebelum menggunakannya. Contoh yang baik adalah for loop.

Dengan Python, Anda bisa melakukan sesuatu seperti ini:

```
for i in range (0, 3):
```

Di C dan Arduino, loop terlihat seperti ini:

```
for (int i = 0; i < 3; i ++) {... }
```

Ini adalah pernyataan yang sangat berbeda. Saya menjelaskan sintaks for loop nanti di bab ini.

Hal utama yang harus diperhatikan di sini adalah bahwa dalam Python, variabel i dibuat tanpa tipe dan menjadi bilangan bulat ketika nilai pertama, 0, ditetapkan padanya. Di Arduino, Anda harus memberi tahu compiler apa variabel itu sebelum Anda memberikan nilai padanya; jika tidak, Anda menerima kesalahan yang mirip dengan ini:

```
Error: variable i not defined in this scope
```

Aturan untuk mendeklarasikan variabel sama dengan Python, dan praktik terbaiknya juga sama.

- Variabel hanya boleh berisi huruf, angka, dan garis bawah.
- Mereka peka huruf besar/kecil; `variable` tidak sama dengan `Variable`. Itu akan menggigitmu nanti.
- Jangan gunakan kata kunci Python.
- Buatlah nama variabel bermakna dengan karakter sesedikit mungkin.
- Berhati-hatilah saat menggunakan huruf kecil L dan huruf besar O, yang terlihat sangat mirip dengan 1 dan 0, dan dapat menyebabkan kebingungan. Saya tidak mengatakan jangan menggunakannya; pastikan saja apa yang Anda lakukan jelas. Menggunakannya sebagai nama variabel karakter tunggal sangat tidak disarankan.

Karakter Dan String

Senar datang dalam tiga rasa; karakter, string sebagai array karakter, dan string sebagai objek. Masing-masing ditangani dengan cara yang berbeda. Karakter (char) adalah karakter alfanumerik tunggal yang disimpan sebagai nilai numerik ASCII. Ingat, komputer bekerja pada 1s dan 0, dan semuanya akhirnya dipecah menjadi angka yang disimpan sebagai 1s dan 0s. Kode ASCII adalah nilai numerik yang mewakili individu karakter alfanumerik. Misalnya, huruf a sebenarnya adalah kode ASCII 97.

Bahkan karakter yang tidak terlihat memiliki representasi ASCII. Kode ASCII untuk carriage return adalah 13. Anda sering melihat ini ditulis menggunakan notasi yang sama dengan fungsi char, seperti `char(13)` Serangkaian karakter dapat ditangani dengan dua cara berbeda. penduduk metode asli untuk menangani string yang diwarisi dari C adalah array karakter. Anda mendeklarasikan jenis string seperti ini:

```
string someWord[7];
```

atau

```
string someWord[] = "Arduino";
```

Ini membuat string yang terdiri dari 10 karakter yang disimpan sebagai array. Kami akan mempelajari lebih lanjut tentang array segera, tetapi mereka kira-kira setara dengan daftar Python. Untuk mengakses karakter dalam string jenis ini, Anda menggunakan posisinya dalam array. Contoh `someWord[0]` mengembalikan karakter A.

Obyek String

Meskipun mungkin ada saatnya Anda ingin memanipulasi karakter dan string karakter dengan cara yang baru saja saya jelaskan, Arduino menyediakan cara yang jauh lebih nyaman untuk bekerja dengan string: objek `String`. Perhatikan huruf kapital S. Objek `String` menyediakan sejumlah metode bawaan untuk bekerja dengan teks dan mengubah nilai lain menjadi string. Banyak dari fungsi ini mudah dibuat ulang menggunakan manipulasi array sederhana. Objek String membuatnya lebih mudah; namun, jika Anda tidak berencana melakukan banyak manipulasi string, ini mungkin berlebihan. Contoh fungsi yang berguna untuk manipulasi string adalah `trim()`,`toUpperCase()`), dan `toLowerCase()`.

Ada beberapa cara untuk membuat objek `String`. Karena ini adalah sebuah objek, Anda harus membuat sebuah instance dari objek `String`. Sebuah objek umumnya dipakai dengan cara yang sama Anda mendeklarasikan variabel lain. Faktanya, karena semua tipe data pada dasarnya adalah objek, itu persis sama. Misalnya, ini adalah cara Anda memulai instance objek `String` yang disebut `myString`:

```
String myString;
```

atau

```
String myString = "Arduino";
```

Angka

Seperti Python, ada beberapa format angka yang tersedia. Yang paling umum adalah bilangan bulat (int) dan desimal (float). Anda terkadang menggunakan tipe Boolean dan beberapa lainnya.

Integer mewakili angka 16-bit antara -32.768 hingga 32.767. Sebuah integer unsigned dapat memiliki nilai positif antara 0 dan 65.535. Bilangan bulat panjang (panjang) adalah angka 32-bit dari –2.147.483.648 hingga 2.147.483.647. Jadi tergantung pada ukuran nomor yang Anda butuhkan, Anda memiliki beberapa opsi. Desimal, atau bukan bilangan bulat, disimpan sebagai tipe float. Float adalah bilangan 32-bit dari –3.4028235E+38 hingga 3.4028235E+38. Seperti Python, float di Arduino tidak asli dan hanya perkiraan. Tetapi mereka lebih tepat di Arduino daripada di Python.

Kode berikut menggambarkan cara membuat variabel angka di Arduino:

```
int myNumber;
int myNumber = 10;
long myLongInt;
long myLongInt = 123456;
float myFloat;
float myFloat = 10.1;
```

Pastikan untuk mencatat titik koma di akhir setiap baris. Setiap baris kode, kecuali blok kode, harus diakhiri dengan titik koma.

Array

Seperti disebutkan sebelumnya, array pada dasarnya sama dengan daftar di Python. Mereka dilambangkan dengan tanda kurung ([ ]). Mengatasi nilai dalam array bekerja persis seperti di Python. Array arduino juga berbasis nol, yang berarti nilai pertama dalam array berada pada posisi 0. Contoh berikut membuat array, mengulanginya, dan kemudian mengeluarkan beberapa nilai ke port serial.

1. Buat sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai `array_example`.
3. Perbarui kode agar terlihat seperti ini:

```
int numbers[5];
int moreNumbers[5] = {1,2,3,4,5};
void setup() {
      // put your setup code here, to run once:
Serial.begin(9600);
}
```

```
void loop() {
      // put your main code here, to run repeatedly:
for(int i = 0; i < 5; i++){
      Serial.println(numbers[i]);
      }
for(int i = 0; i < 5; i++){
      numbers[i] = moreNumbers[i];
      }
for(int i = 0; i < 5; i++){
      Serial.println(numbers[i]);
      }
      numbers[1] = 12;
      for(int i = 0; i < 5; i++){
      Serial.println(numbers[i]);
        }
      }
```

4. Simpan file.
5. Unggah sketsa ke Arduino Anda.
6. Klik **Tools ➤ Serial Monitor**.

5.14 STRUKTUR KONTROL

Seperti Python, Arduino menyediakan beberapa struktur untuk menambahkan beberapa kontrol ke kode Anda. Ini harus cukup akrab karena mereka sangat mirip dengan rekan-rekan mereka di Python. Tentu saja, sintaksnya berbeda, dan Anda harus memperhatikan titik koma dan tanda kurung.

if dan else

Ini umumnya dianggap sebagai konstruksi paling dasar. Ini hanya memungkinkan Anda untuk mengeksekusi kode berdasarkan hasil dari kondisi Boolean. Jika kondisi bernilai true, maka kode akan dieksekusi; jika tidak, program akan melewatkan kode dan menjalankan perintah berikutnya. Berikut adalah contoh pernyataan if:

```
if(val == 1){doSomething();}
```

Dalam contoh ini, kita hanya mengevaluasi isi dari variabel `val`. Jika `val` berisi bilangan bulat 1, maka kode di dalam tanda kurung akan dieksekusi; jika tidak, program akan melewatkan kode dan melanjutkan ke baris berikutnya. Seluruh klausa tidak perlu, dan paling sering tidak terbatas pada satu baris. Secara umum, bahkan jika kode di dalam tanda kurung terdiri dari satu

baris, saya memperluas pernyataan untuk menggunakan beberapa baris. Saya hanya merasa ini lebih mudah dibaca. Kode ini secara fungsional identik dengan contoh sebelumnya.

```
if(val == 1){
      doSomething();
      }
```

Anda dapat mengevaluasi beberapa nilai menggunakan pernyataan else, yang berfungsi persis seperti yang Anda harapkan. Anda hanya memberi tahu kompiler untuk mengevaluasi setiap kondisi berurutan jika kondisi sebelumnya bernilai salah.

```
if(val == 1){
      doSomething();
 }
else if(val == 2){
      doSomethingElse();
}
else if(otherVal == 3){
      doAnotherThing()
}
else {
      doAlternateThing();
}
```

Bagian pertama dari kode ini sama dengan contoh sebelumnya. Jika nilai val adalah 1, maka lakukan sesuatu. Jika kondisi ini salah, val bukan 1, lalu periksa untuk melihat apakah itu 2. Jika ya, lakukan hal lain. Jika itu juga tidak benar, maka periksa nilai otherVal. Jika itu adalah 3, maka lakukan hal lain. Terakhir, jika tidak ada kondisi sebelumnya yang benar, maka jalankan kode ini.

Pernyataan terakhir else tidak diperlukan. Anda dapat mengabaikan pernyataan ini, dan kode akan terus menjalankan kode apa pun yang mengikutinya. Pernyataan else terakhir adalah untuk kode yang hanya ingin Anda jalankan jika semua kondisi lainnya tidak benar. Juga, perhatikan pernyataan lain/jika kedua. Anda tidak harus mengevaluasi variabel yang sama untuk kondisi lain. Setiap operasi yang mengevaluasi benar atau salah adalah valid.

Sementara Loop

while loop berulang kali mengeksekusi blok kode selama kondisinya benar. Di Python, kami menggunakan ini untuk membuat loop berkelanjutan untuk mengeksekusi pemrograman kami secara konstan. Praktik itu tidak diperlukan di Arduino karena fungsi Loop() standar menyediakan fungsionalitas itu. Seperti pernyataan if, while mengevaluasi suatu kondisi. Jika kondisi bernilai true, blok kode akan dieksekusi. Setelah blok kode dieksekusi, ia mengevaluasi kondisinya lagi. Jika kondisi masih bernilai true, blok kode akan dieksekusi kembali. Ini berlanjut

sampai kondisi bernilai salah. Karena itu, sangat penting untuk memastikan bahwa nilai yang dievaluasi dalam kondisi diperbarui di blok kode.

Ini adalah contoh perulangan while:

```
int i = 0;
while(i < 3){
	doSomething();
	i++;
}
```

Dalam contoh ini, kita membuat bilangan bulat dengan nilai 0 sebelum kita masuk loop sementara. Pernyataan while mengevaluasi nilai i. Karena saat ini 0, yang kurang dari 3, ia mengeksekusi blok kode. Di dalam blok kode kita menaikkan nilai jika i. Pernyataan while mengevaluasi nilai lagi. Kali ini 1 yang masih kurang dari 3, jadi blok kode dieksekusi lagi. Ini berlanjut sampai nilai i bertambah menjadi 3. Karena 3 tidak kurang dari 3, loop while keluar tanpa mengeksekusi blok kode.

Seperti semua loop lainnya, loop while memblokir. Ini berarti selama kondisi bernilai benar, blok kode akan dieksekusi, mencegah kode lain dieksekusi. Fitur ini biasanya digunakan untuk mencegah kode berjalan hingga kondisi muncul untuk mencegah kesalahan atau hasil yang tidak diharapkan di kemudian hari. Misalnya, jika kode Anda memerlukan koneksi serial agar dapat dilanjutkan, Anda dapat menambahkan kode ini ke program Anda:

```
Serial.begin(9600);
while(!Serial){}
```

Fungsi Serial adalah bagian dari perpustakaan Arduino standar. Ini hanya memeriksa untuk melihat apakah koneksi serial tersedia. Jika koneksi serial telah dibuat, Serial mengevaluasi ke true. Namun, tanda seru (!) yang mendahuluinya berarti tidak. Jadi kami mengatakan, "Selama tidak ada koneksi serial, jalankan kode ini." Blok kode kosong, jadi tidak ada kode untuk dijalankan. Hasilnya adalah kode berhenti sampai koneksi serial tersedia.

Loop For

Seperti perulangan while, perulangan for mengeksekusi blok kode berulang kali hingga kondisi bernilai true. Perbedaan antara keduanya adalah bahwa for loop juga mendefinisikan dan mengubah variabel yang dievaluasi. Umumnya, ini hanya menyiapkan bilangan bulat untuk berfungsi sebagai penghitung, mengevaluasi nilai terhadap ambang batas yang ditetapkan, dan menambah nilai. Dengan setiap kenaikan, blok kode dieksekusi sampai kondisi tidak lagi bernilai benar; Misalnya:

```
for(int i = 0; i < 3; i++){
	doSomething();
}
```

Dalam contoh ini, kami mendeklarasikan bilangan bulat yang disebut i. Kami ingin melanjutkan untuk mengulang blok kode selama i kurang dari 3. Setiap kali kita menjalankan kode, kami menambah nilai i dengan 1 sampai nilai i adalah 3. Karena 3 tidak kurang dari 3, loop keluar tanpa mengeksekusi blok kode. Ini berguna ketika Anda ingin mengeksekusi sepotong kode beberapa kali. Anda juga dapat menggunakan nilai yang bertambah. Misalnya, jika kita ingin LED pada pin 13 memudar daripada hanya menyala, kita dapat membuat kode ini:

```
pinMode(11, OUTPUT);
for(int i = 0; i < 255; i++){
	analogWrite(11, i);
}
```

Pertama, kami memberi tahu Arduino bahwa kami ingin menggunakan pin 13 sebagai pin output.

Anda akan segera mempelajari lebih lanjut tentang bekerja dengan pin. Kemudian kita mengatur loop for untuk menaikkan nilai i dari 0 menjadi 254. Nilai i kemudian ditulis ke pin 13, mengatur nilai PWM. Jika Anda ingat dari bab yang sebelumnya, nilai PWM mengontrol kecerahan LED dengan menentukan seberapa sering pin tinggi dalam siklus tertentu. Jadi, kami memiliki LED yang meningkatkan kecerahan maksimumnya. Kami benar-benar menulis kode fading LED ketika kami mulai bekerja dengan pin.

Function

Seperti Python, Arduino memungkinkan Anda untuk memecah kode Anda menjadi bagian-bagian yang lebih kecil melalui fungsi. Fungsi Arduino sangat mirip dengan yang ada di Python. Sintaksnya, tentu saja, berbeda. Namun, dengan keduanya, Anda mendeklarasikan nama fungsi, mencantumkan parameter apa pun yang diperlukan, dan menyediakan blok kode untuk dijalankan saat fungsi dipanggil. Anda sudah akrab dengan sintaks fungsi Arduino. Baik blok setup dan loop dalam sketsa Arduino adalah fungsi. Satu-satunya perbedaan adalah bahwa ini adalah fungsi sistem yang secara otomatis dipanggil sesuai kebutuhan selama runtime. Jika Anda sudah familiar dengan C atau C++, mereka mirip dengan fungsi main() pada akar bahasa tersebut.

Anda menggunakan fungsi kapan saja Anda memiliki blok kode yang mungkin ingin Anda gunakan di lebih dari satu tempat. Dengan cara ini, Anda menulisnya hanya sekali, dan selalu melakukan hal yang sama terlepas dari mana Anda menyebutnya. Sintaks umum suatu fungsi terlihat seperti ini:

```
returnType functionName(parameterType parameterName){
	doSomething();
}
```

Mungkin lebih baik dan lebih mudah untuk memandu Anda melalui pembuatan dan penggunaan suatu fungsi. Dalam latihan ini, kita membuat fungsi sederhana yang menjumlahkan dua angka. Ini bukan fungsi yang sangat praktis, tetapi memberikan contoh bagaimana cara membuat fungsi.

1. Buat sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai `function_example`.
3. Perbarui kode menjadi ini:

```
int a = 1;
int b = 2;
int val;
int answer;
int add_vars(){
      val = a+b;
      return val;
}
int add_params(int p1, int p2){
      val = p1+p2;
      return val;
}
void printVal(){
     Serial.println(val);
}
void setup() {
       // put your setup code here, to run once:
       Serial.begin(9600);
}
void loop() {
       // put your main code here, to run repeatedly:
       add_vars();
       printVal();
       add_params(a,b);
       printVal();
       answer = add_vars();
       Serial.println(answer);
       a++;
       b++;
}
```

4. Unggah sketsa ke Arduino Anda.
5. Buka monitor serial dari menu Tools.

Dalam latihan ini, kami membuat tiga fungsi. Dua fungsi pertama mengembalikan nilai bertipe int. Karena itu, kami mendahului nama fungsi dengan tipe data int. Fungsi ketiga tidak mengembalikan data; itu hanya melakukan tugas, sehingga didahului dengan kekosongan. Fungsi pertama, `add_vars()`, menambahkan dua variabel global bersama-sama. Ini menekankan manfaat dan bahaya variabel global. Variabel global dapat dimanipulasi oleh kode apa pun dalam program Anda. Ini adalah metode mudah untuk melakukan tugas pada data yang sama, dan kemudian meneruskan data itu dari satu fungsi ke fungsi lainnya. Namun, Anda harus menyadari bahwa setiap perubahan yang Anda buat pada variabel berlaku di mana pun variabel tersebut digunakan.

Alternatif yang lebih aman untuk ini adalah menggunakan parameter dalam fungsi Anda. Dengan cara ini, Anda memiliki kontrol lebih karena Anda memberikan nilai. Fungsi kedua, `add_params()`, mendemonstrasikan ini. Parameter dibuat sebagai bagian dari deklarasi fungsi. Kami menyediakan tipe data untuk masing-masing dan nama variabel yang akan digunakan dalam fungsi. Jadi, ini persis seperti mendeklarasikan variabel, kecuali nilainya ditetapkan saat runtime ketika fungsi dipanggil. Fungsi terakhir tidak mengembalikan data dan tidak memerlukan parameter. Dalam kasus khusus ini, kami mencetak nilai variabel global val ke port serial.

5.15 BEKERJA DENGAN PIN

Tujuan utama Arduino adalah untuk berinteraksi dengan komponen lain, sensor, atau perangkat lain. Untuk melakukan ini, kita harus tahu cara berinteraksi dengan pin. Pin Arduino terhubung langsung ke prosesor AVR di jantungnya. Arduino menyediakan akses ke 14 pin digital, 6 pin analog, 6 pin PWM hardware, serial TTL, SPI, dan serial dua kabel. Saya menekankan pada PWM perangkat keras karena semua pin digital atau analog dapat digunakan untuk perangkat lunak PWM. Saya tidak membahas semua kemampuan ini dalam buku ini, tetapi saya menyarankan Anda meluangkan waktu untuk mempelajarinya. Kami akan melihat input dan output digital dan analog dasar Anda. Ini adalah fungsi yang paling sering Anda gunakan.

Sebelum Anda dapat menggunakan salah satu pin sebagai input atau output, Anda harus terlebih dahulu mendeklarasikan bagaimana Anda menggunakannya. Ini dilakukan dengan menggunakan fungsi pinMode(). Untuk melakukan ini, yang harus Anda lakukan adalah memberikan nomor pin dan mode. Misalnya, kode ini menetapkan pin 13 sebagai pin keluaran:

```
pinMode(13, OUTPUT);
```

Saya sering menggunakan variabel untuk nomor pin. Ini membuatnya lebih mudah untuk mengidentifikasi apa yang Anda lakukan dalam kode; Misalnya:

```
int servoPin = 11;
int LEDPin = 13;
```

Sekarang, ketika saya perlu mereferensikan pin, lebih mudah untuk memahaminya.

```
pinMode(LEDPin, OUTPUT);
```

Operasi Digital

Sekarang kita memiliki pin yang ditentukan, kita dapat mulai menggunakannya. Seperti halnya Python, Anda dapat mengaktifkan atau menonaktifkan pin dengan menyetelnya tinggi atau rendah. Ini dilakukan dengan menggunakan fungsi `digitalWrite()`, yang dengannya Anda memberikan nomor pin dan tinggi atau rendah; Misalnya:

```
digitalWrite(LEDPin, HIGH);
```

Dengan menggunakan contoh pinMode(), ini mengubah pin 13 menjadi tinggi, atau aktif. Demikian juga, Anda dapat mematikan pin dengan menyetelnya rendah. Di sisi lain, Anda dapat membaca status pin saat ini dengan `digitalRead()`. Untuk melakukan ini, pertama-tama Anda harus mengatur mode ke input.

```
int buttonPin = 3;
int val;
pinMode(buttonPin, INPUT);
val = digitalRead(buttonPin);
```

Cuplikan kode ini memberikan nilai 3 ke variabel buttonPin, dan kami membuat variabel untuk menyimpan hasilnya. Ini kemudian mengatur mode pin ke input sehingga kita dapat membacanya. Terakhir, kita membaca nilai pin 13 ke dalam variabel val.

Masukan Analog

Input analog bekerja sedikit berbeda; meskipun Anda dapat menggunakan pin IO apa pun untuk operasi digital, Anda hanya dapat menggunakan pin analog yang ditentukan untuk input analog. Seperti yang saya bahas dalam pengantar Python, mikrokontroler tidak dapat benar-benar melakukan analog. Dengan satu atau lain cara sinyal harus dikonversi antara analog dan digital. Dengan output analog, ini dilakukan melalui modulasi lebar pulsa (PWM). Untuk input analog, kami menggunakan konverter analog ke digital, atau ADC, untuk mengubah sinyal analog menjadi sinyal digital. Ini adalah fungsi perangkat keras, sehingga harus dilakukan pada pin tertentu. Dalam kasus Arduino Uno, pin ini adalah A0 hingga A5. Karena pin ini didedikasikan untuk input analog, mendeklarasikannya sebagai input tidak sepenuhnya diperlukan. Saya tetap menyarankan untuk melakukannya karena ini berfungsi sebagai indikasi bahwa pin ini sedang digunakan. Fungsi `analogRead()` digunakan untuk membaca pin; Misalnya:

```
val = analogRead(A0);
```

Ini memberikan nilai A0 ke variabel val. Ini adalah nilai integer antara 0 dan 1023.

Keluaran Analog (PWM)

PWM bekerja hampir sama seperti di Python. Pada pin yang ditentukan, Anda dapat memberikan nilai antara 0 dan 255 untuk memvariasikan output pin. Nilai 0 adalah ekuivalen analog dari rendah digital, atau mati; sedangkan nilai 255 analog dengan digital high, atau on. Dengan demikian, nilai 127 memberikan siklus kerja 50%, kira-kira sama dengan setengah daya. Dengan Arduino, Anda menggunakan `analogWrite()` untuk mengatur sinyal PWM pin. Pada Arduino Uno, pin PWM adalah 5, 11, 12, 15, 16, dan 17. Cuplikan kode berikut menetapkan output pin 11 menjadi sekitar 25%.

```
int PWMPin = 11;
pinMode(PWMPin, OUTPUT);
analogWrite(PWMPin, 64);
```

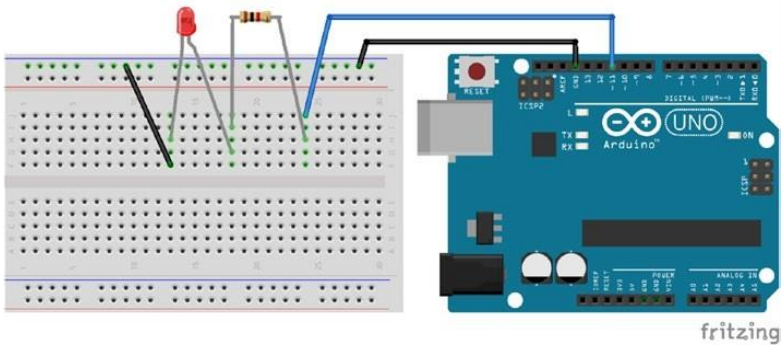
LED berdenyut

Dalam latihan ini, kita akan membuat pulsa LED. Pin 13 bukan pin PWM, jadi kami tidak akan dapat menggunakan LED bawaan kali ini, yang berarti sudah waktunya untuk membongkar papan tempat memotong roti dan beberapa jumper.

Sirkuit

Untuk menghubungkan rangkaian, kita memerlukan resistor 220 ohm, LED 5V, Arduino Anda, papan tempat memotong roti, dan beberapa jumper. Lihat Gambar 5-8 untuk menghubungkan latihan ini.

1. Hubungkan LED ke papan tempat memotong roti.
2. Hubungkan resistor sedemikian rupa sehingga salah satu ujungnya terhubung ke saluran yang digunakan bersama dengan pin panjang LED.
3. Hubungkan jumper dari pin lain dioda ke pin GND pada Arduino.
4. Hubungkan jumper dari ujung resistor yang lain ke pin 11 pada Arduino.



Gambar 5-8. Tata letak sirkuit latihan fade LED

Kode

Sebelumnya kita menggunakan `analogWrite()` dalam contoh for loop. Sekarang kita menulis kode untuk mengimplementasikan contoh di Arduino.

1. Buat sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai PWM_Example.
3. Perbarui kode menjadi ini:

```
int PWMPin = 11;
void setup() {
      // put your setup code here, to run once:
      pinMode(PWMPin, OUTPUT);
}
void loop() {
      // put your main code here, to run repeatedly:
      for(int i = 0; i < 255; i++){
            analogWrite(PWMPin, i);
      }
      for(int i = 255; i >= 0; i--){
            analogWrite(PWMPin, i);
      }
      delay(100);
}
```

4. Simpan dan unggah sketsa ke Arduino Anda.

LED pada papan tempat memotong roti sekarang harus berdenyut. Untuk mengubah laju pulsa, ubah nilai dalam fungsi tunda.

5.16 OBJEK DAN KELAS

Membuat objek dan kelas berada di luar cakupan buku ini. Anda sangat jarang, jika pernah, perlu membuatnya di dalam Arduino. Namun, Anda sering menggunakan objek atau kelas dari perpustakaan lain. Sebuah objek umumnya dipakai dengan cara yang sama seperti Anda mendeklarasikan variabel lain; Anda memberi tahu kompiler jenis objek diikuti dengan nama variabel yang merujuk padanya.

```
ObjectType variableName;
```

Setelah dideklarasikan, Anda memiliki akses ke semua properti dan metode kelas ini. Sebuah contoh yang baik dari ini adalah kelas Servo. Ini adalah standar perpustakaan dengan Arduino. Cuplikan kode berikut membuat objek servo dan melampirkannya ke pin 12:

```
#include <Servo.h>
Servo myServo;
```

```
myServo.attach(12);
```

Pertama, kami menyertakan perpustakaan Servo. Setelah perpustakaan Servo disertakan, kita dapat dengan mudah membuat instance dari kelas Servo. Dalam hal ini, kami membuat objek servo yang disebut myServo. Setelah objek dibuat, kita dapat menggunakan metode attach() untuk menetapkan pin 12 untuk mengontrol servo.

5.17 SERI

Ada beberapa saluran serial di Arduino. Kami menggunakan koneksi USB antara Raspberry Pi dan Arduino. Sejauh ini, ini adalah cara paling sederhana untuk berkomunikasi di antara keduanya.

Menghubungkan ke Serial

Untuk menggunakan komunikasi serial, Anda harus memulainya terlebih dahulu. Untuk melakukan ini, gunakan `Serial.begin(baudRate)`. Misalnya, baris ini memulai koneksi serial dengan baud rate 9600bps:

```
Serial.begin(9600);
```

Baud rate yang Anda pilih sepenuhnya terserah Anda dan kebutuhan Anda. Yang penting sesuai dengan baud rate komputer yang terhubung. Jadi, ketika Anda menginisialisasi koneksi serial pada Pi, Anda harus memastikan bahwa mereka cocok. Saya akan membahas membangun koneksi itu segera. Untuk memverifikasi koneksi serial berhasil, Anda dapat menanyakan kata kunci Serial. Serial adalah objek Boolean yang menunjukkan tersedia atau tidaknya koneksi serial. Jika koneksi tersedia, itu benar; jika tidak, itu salah. Sebenarnya ada beberapa cara untuk menggunakan Serial. Anda bisa gunakan itu sebagai kondisi Boolean untuk pernyataan if dan letakkan kode dependennya di blok kode pernyataan if. Atau, Anda dapat menggunakannya sebagai kondisi Boolean dari perulangan while. Berikut adalah dua metode untuk memeriksa koneksi serial. Jalankan kode hanya jika tersedia.

```
if(Serial){
      doSomething();
}
while(Serial){
      doSomething();
}
```

Blok pertama mengeksekusi kode jika koneksi serial tersedia, dan kemudian pindah ke kode mengikuti pernyataan if. Blok kedua menjalankan kode terus menerus, selama koneksi tersedia. Kode apa pun yang mengikuti loop while tidak akan berjalan sampai koneksi serial

diakhiri dan loop keluar. Alternatif ketiga adalah menghentikan semua kode saat koneksi tidak tersedia. Ini adalah while loop lain yang telah kita lihat sebelumnya.

```
while(!Serial){}
```

Ini menggunakan operator "tidak", atau tanda seru (!). Agar kondisi bernilai benar, kondisi tersebut tidak boleh memenuhi kriteria. Dalam hal ini, selama koneksi tidak tersedia, jalankan kode di blok. Tapi, karena tidak ada kode, itu hanya menghentikan program sampai tersedia.

Mengirim Data Serial

Banyak dari apa yang akan kita lakukan hanyalah mencetak ke port serial. Faktanya, itulah yang telah kami lakukan dalam contoh sebelumnya. Metode `Serial.println()` mengirimkan data dalam kurung ke port serial. Monitor serial di Arduino IDE memungkinkan Anda untuk melihat output ini. Untuk menulis data ke aliran serial, Anda biasanya menggunakan salah satu metode cetak serial. `Serial.print()` mencetak isi tanda kurung ke aliran serial tanpa terminator baris baru. Ini berarti bahwa semua yang Anda cetak menggunakan metode ini muncul pada baris yang sama di monitor serial. Metode `Serial.println()` menyertakan terminator baris baru. Jadi semua yang dicetak dengan metode ini diikuti oleh baris baru.

Menerima Data Serial

Tentu saja, port serial juga bekerja dengan cara lain. Anda dapat membaca aliran serial dari Pi menggunakan beberapa metode objek Serial. Banyak metode untuk membaca data dari serial untuk bekerja dengan byte individu. Ini bisa membingungkan dan tidak praktis jika Anda baru memulai. Jika Anda terbiasa dan nyaman bekerja dengan byte data individual, `Serial.read()`, `Serial.readByte()`, dan lainnya mungkin berguna. Namun, bukan itu yang akan kami gunakan. Untuk mempermudah, kita akan menggunakan metode `Serial.parseInt()` dan `Serial.readString()`.

Kedua metode ini melakukan bagian terbesar dari pekerjaan saat membaca dari aliran serial. `Serial.parseInt()` membaca aliran serial yang masuk dan kembali; namun, itu tidak mengurai bilangan bulat sekaligus. Saat Anda pertama kali memanggilnya, ia mengembalikan bilangan bulat pertama yang ditemuinya. Panggilan berikutnya mengembalikan bilangan bulat berikutnya. Setiap iterasi mengembalikan bilangan bulat berikutnya yang ditemukan sampai mencapai akhir baris.

Mari kita lihat bagaimana `parseInt()` bekerja. Dalam kode berikut, Arduino menunggu untuk menerima input dari serial stream. Itu kemudian beralih melalui input dan mem-parsing bilangan bulat, mencetak masing-masing pada baris baru.

1. Buka sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai `parseInt_example`.
3. Masukkan kode ini:

```
int val;
```

```
void setup() {
      // put your setup code here, to run once:
      Serial.begin(9600);
}
void loop() {
      // put your main code here, to run repeatedly:
      while(Serial.available() > 0){
         val = Serial.parseInt();
         Serial.println(val);
      }
}
```

4. Unggah sketsa ke Arduino Anda.
5. Buka monitor serial.
6. Di bidang entri data di bagian atas monitor serial, masukkan 1,2,3,4. Pastikan untuk memisahkan setiap nilai dengan koma.
7. Klik Kirim.

Monitor serial menulis setiap bilangan bulat pada baris baru. Jika Anda memasukkan karakter alfa, itu akan mencetak 0 karena itu adalah karakter alfanumerik tetapi bukan bilangan bulat. `Serial.readString()` membaca seluruh baris dari aliran serial sebagai rangkaian. Ini dapat ditetapkan ke variabel String untuk digunakan nanti. Metode ini bekerja dengan baik jika Anda mengirim informasi teks ke Arduino. Namun, lambat, dan Anda melihat jeda yang signifikan antara waktu baris dikirim dan waktu diterima, diproses, dan tersedia.

5.18 ARDUINO KE PI DAN KEMBALI LAGI

Anda perlu tahu sedikit tentang komunikasi serial karena begitulah cara kita berkomunikasi antara Raspberry Pi dan Arduino. Baik Pi dan Arduino bekerja dengan komunikasi serial secara berbeda. Saya tidak membahas serial di bab Python karena penting bahwa diskusi ini terjadi bersamaan dengan Arduino. Dengan demikian, Anda mungkin ingin melompat kembali ke Bab 3 untuk tinjauan singkat tentang Python setelah semua pengkodean Arduino itu. Saya telah berbicara tentang cara membuka koneksi serial di Arduino. Raspberry Pi hanya sedikit lebih rumit. Pertama, komunikasi serial bukan bagian dari kerangka default. Jadi, kita perlu menginstalnya. Setelah terinstal, kode kita perlu mengimpor perpustakaan serial. Setelah selesai, kami membuat turunan dari kelas serial, yang memberi kami akses ke metode yang kami butuhkan.

Menginstal PySerial

Fungsionalitas serial disediakan oleh paket PySerial. Untuk menggunakannya, Anda harus terlebih dahulu memastikan bahwa itu diinstal dalam implementasi Python Anda.

1. Di Raspberry Pi Anda, buka jendela terminal.
2. Ketik **python –m pip install pyserial.**
   Ini menginstal paket PySerial jika belum diinstal.
3. Ketik **python**.
   Ini memulai sesi Python baru di dalam terminal.
4. Ketik **import serial**.
   Ini memverifikasi versi PySerial Anda.

Sekarang setelah PySerial terinstal, kita dapat menggunakannya di program kita. Untuk menggunakan serial di Python, kita perlu mengimpor perpustakaan dan membuat koneksi. Cuplikan kode berikut kemungkinan ada di sebagian besar skrip yang berinteraksi dengan Arduino:

```
import serial
ser = serial.Serial('/dev/ttyAMC0', 9600)
```

Membuat koneksi serial dengan Python mirip dengan apa yang kami lakukan dengan Arduino. Perbedaan terbesar adalah kita menetapkan objek serial ke variabel; dalam hal ini, ser. Dalam panggilan inisiasi, kami menyediakan port tempat Arduino aktif serta baud rate yang kami hubungkan. Sekali lagi, pastikan ini sesuai dengan baud rate yang Anda tetapkan di Arduino. Jika ini tidak cocok, Anda mendapatkan karakter aneh dan hasil yang tidak terduga—jika Anda mendapatkan apa pun.

Mengirim Data ke Raspberry Pi

Ini bukan tentang mengirim data ke Pi tetapi tentang bagaimana Pi menerima data dan kemudian apa yang dilakukannya dengannya. Pendekatan paling sederhana untuk menerima data serial pada Pi adalah dengan menggunakan metode `readLine()` dari objek serial. Ini membaca byte dari aliran serial hingga mencapai karakter baris baru. Byte kemudian diubah menjadi string. Semua data yang dikirim pada baris disimpan dalam satu string. Bergantung pada cara Anda mengirim data dari Arduino, Anda mungkin perlu menggunakan metode `split()` untuk mengurai data menjadi tuple. Penting untuk dicatat bahwa metode `readLine()` terus membaca aliran serial sampai karakter baris baru diterima. Jika Anda tidak mengirimnya dari Arduino, Pi akan terus mencoba membaca data. Untuk membantu mencegah penguncian program Anda, Anda mungkin ingin mengatur interval waktu habis sebelum mencoba `readLine()`. Ini dapat dicapai dengan menambahkan parameter batas waktu saat Anda membuat koneksi. Baris kode berikut membuat koneksi serial dengan batas waktu satu detik:

```
ser = serial.Serial('/dev/ttyAMC0', 9600, timeout=1)
```

Metode pengiriman data yang saya sukai antara Pi dan Arduino adalah melalui serangkaian nilai yang dipisahkan koma. Bergantung pada kerumitan proyek, saya dapat melakukan pembacaan langsung, di mana setiap nilai yang diteruskan sesuai dengan variabel tertentu. Ini memiliki manfaat yang cukup sederhana. Yang harus saya lakukan adalah mengurai aliran serial menjadi bilangan bulat dan menetapkan setiap bilangan bulat, secara berurutan, ke variabel masing-masing untuk digunakan nanti.

Pada proyek yang lebih kompleks, saya dapat mengirim nilai berpasangan atau set bilangan bulat. Saat diurai, bilangan bulat pertama biasanya menunjukkan fungsi atau perangkat untuk pesan tersebut; yang kedua adalah nilai untuk ditetapkan ke variabel. Dari Arduino, saya cukup menulis nilai dan pemisah komanya dalam sejumlah perintah `Serial.print()` yang diakhiri dengan `Serial.print()` untuk memastikan bahwa baris diakhiri dengan benar. Di Pi, saya menggunakan metode `readLine()` untuk menangkap seluruh baris sebagai string tunggal kemudian menggunakan metode `split()` untuk mengurai string menjadi Tuple. Tuple dapat diuraikan lebih lanjut menjadi variabel individu sesuai kebutuhan.

Untuk mengilustrasikannya, mari buat program sederhana yang mengirimkan urutan angka dari Arduino ke Raspberry Pi setiap 500 milidetik. Ini cukup sering untuk tidak timeout. Di Pi, kami mengurai nilai-nilai itu dan menetapkannya ke variabel individual. Ini adalah kasus penggunaan umum untuk mengirim pembacaan sensor dari Arduino ke Pi. Untuk melakukan ini, kita harus menulis dua program: satu untuk Arduino dan satu untuk Pi. Mari kita mulai di Arduino.

1. Buat sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai `Arduino_to_Pi_example`.
3. Masukkan kode berikut:

```
int a = 1;
int b = 2;
int c = 3;
void setup() {
      // put your setup code here, to run once:
      Serial.begin(9600);
}
void loop() {
   // put your main code here, to run repeatedly:
   while(!Serial){};
   Serial.print(a); Serial.print(",");
   Serial.print(b); Serial.print(",");
   Serial.println(c);
   delay(500);
   a++;
```

```
    b++;
    c++;
}
```

4. Simpan dan unggah sketsa ke Arduino Anda.
5. Buka file Python baru di IDLE.
6. Simpan file sebagai Arduino_to_pi_example.py.
7. Masukkan kode berikut:

```
import serial
ser = serial.Serial('/dev/ttyACM0',9600,timeout=1)
while 1:
      val = ser.readline().decode('utf-8')
      parsed = val.split(',')
      parsed = [x.rstrip() for x in parsed]
      if(len(parsed) > 2):
            print(parsed)
            a = int(int(parsed[0]+'0')/10)
            b = int(int(parsed[1]+'0')/10)
            c = int(int(parsed[2]+'0')/10)
      print(a)
      print(b)
      print(c)
      print(a+b+c)
```

8. Simpan dan jalankan file tersebut.

Di jendela Shell IDLE, Anda akan melihat output yang mirip dengan ini:

```
['1','2','3']
1
2
3
6
```

Kami melakukan beberapa keajaiban Python di sini yang perlu kami tinjau. Pertama, kami mengimpor perpustakaan serial dan membuka koneksi serial. Setelah koneksi dibuka, kami memasuki perpetual while loop. Setelah itu, saya memperkenalkan beberapa elemen baru yang ingin saya jalani,

```
val = ser.readline().decode('utf-8')
```

Kami membaca baris berikutnya yang berasal dari aliran serial. Namun, string ini diambil sebagai string byte, yang bekerja secara berbeda dari string standar. Untuk membuatnya lebih mudah digunakan, kami menggunakan metode `decode()` untuk mengubah string dari string byte menjadi string standar. Ini memungkinkan kita untuk menggunakan metode kelas string untuk bekerja dengan garis.

```
parsed = val.split(',')
```

Selanjutnya, kami menguraikan string ke dalam daftar. Karena kami menggunakan koma untuk memisahkan nomor kami dari Arduino, berikan itu ke metode split(). Namun, sekarang elemen terakhir dalam daftar menyertakan akhir karakter baris /n/r. Kami tidak menginginkan karakter itu.

```
parsed = [x.rstrip() for x in parsed]
```

Baris ini membangun kembali daftar yang diurai tanpa karakter tambahan. Metode rstrip() menghapus spasi apa pun dari string. Jadi, yang dilakukan baris ini adalah mengulang setiap anggota daftar dan menerapkan metode rstrip(). Kami dibiarkan dengan daftar nomor sebagai string.

```
if(len(parsed) > 2)::
```

Salah satu tantangan yang akan kita hadapi dengan komunikasi serial antara dua papan adalah packet loss. Ini sangat umum ketika kami mengatur ulang Arduino, yang terjadi setiap kali kami membuat koneksi serial baru. Kehilangan ini menghasilkan karakter yang hilang dalam string serial. Untuk mengatasi tantangan ini dalam skrip ini kami menguji panjang daftar. Fungsi len() mengembalikan jumlah anggota dalam daftar. Karena kami tahu daftar kami harus berisi, setidaknya, tiga angka, kami hanya ingin menjalankan kode yang tersisa jika kondisi ini benar.

```
print(parsed)
```

Baris ini hanya mencetak daftar parsing ke jendela shell.

```
a = int(int(parsed[0]+'0')/10)
b = int(int(parsed[1]+'0')/10)
c = int(int(parsed[2]+'0')/10)
```

Bagian terakhir dari keajaiban Python dilakukan ketika kami menetapkan nilai ke variabel masing-masing. Garis-garis ini mencakup manipulasi string dan numerik. Untuk membaca apa yang terjadi di sini kita harus mulai dari tengah di mana kita menambahkan karakter '0' di akhir setiap anggota daftar. Kami melakukan ini karena, terlepas dari upaya kami sebelumnya,

mungkin masih ada string kosong dalam daftar. String kosong tidak dapat dikonversi ke bilangan bulat dan kode tidak akan dikompilasi. Dengan menambahkan 0, kami yakin bahwa ada nilai aktual di sana.

Kami kemudian mengubah string itu menjadi bilangan bulat. Namun, bilangan bulat itu sekarang memiliki 0 yang ditambahkan ke akhir, membuat 1 dibaca sebagai 10, dan seterusnya. Untuk menyesuaikannya, kami membagi angka itu dengan 10, yang menghasilkan float. Karena kita mencari bilangan bulat, kita harus mengonversi hasil akhir menjadi int. Bagian terakhir dari kode hanya mencetak nilai dari setiap variabel ke jendela shell. Baris terakhir disertakan untuk membuktikan bahwa kita beroperasi dengan bilangan bulat dan bukan string.

Mengirim Data Ke Arduino

Untuk mengirim data ke Arduino adalah masalah yang cukup sederhana, di sisi Arduino. Python adalah sentuhan yang lebih terlibat, namun. Menggunakan skenario yang sama seperti sebelumnya, kita perlu memasukkan nilai ke dalam sebuah tuple, dan kemudian menggunakan metode join() untuk menggabungkan nilai-nilai dalam tuple menjadi satu string. String itu kemudian ditulis ke koneksi serial. Di Arduino, yang harus kita lakukan adalah menggunakan `parseInt()` untuk memecah string menjadi tiga bilangan bulat independen, sekali lagi. Dalam latihan ini, kita akan mengirim tiga bilangan bulat ke Arduino. Dalam skenario dunia nyata, angka-angka ini mungkin mewakili warna atau kecerahan LED atau sudut untuk servo. Namun, akan sulit untuk memverifikasi apa yang terjadi di sisi Arduino karena kami tidak dapat menggunakan monitor serial. Untuk mengatasinya, kita akan meminta Arduino untuk menjumlahkan bilangan bulat dan mengembalikan hasilnya ke Pi. Ini berarti bahwa kedua papan membaca dan menulis ke aliran serial.

Sekali lagi, mari kita mulai dari sisi Arduino.

1. Buka sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai roundtrip_example.
3. Masukkan kode berikut:

```
int a = 0;
int b = 0;
int c = 0;
int d = 0;
void setup() {
      // put your setup code here, to run once:
      Serial.begin(9600);
}
void loop() {
      // put your main code here, to run repeatedly:
```

```
        while(!Serial){}
        if(Serial.available()>0){
           a = Serial.parseInt();
           b = Serial.parseInt();
           c = Serial.parseInt();
        }
        d = a+b+c;
          Serial.print(a); Serial.print(",");
          Serial.print(b); Serial.print(",");
          Serial.print(c); Serial.print(",");
          Serial.println(d);
          //delay(500);
}
```

4. Simpan sketsa dan unggah ke Arduino Anda.
5. Buka file Python baru di IDLE.
6. Simpan file sebagai roundtrip_example.py.
7. Masukkan kode berikut:

```
import serial
import time
ser = serial.Serial('/dev/ttyACM0',9600,timeout=1)
a = 1
b = 2
c = 3
while 1:
      valList = [str(a),str(b),str(c)]
      sendStr = ','.join(valList)
      print(sendStr)
      ser.write(sendStr.encode('utf-8'))
      time.sleep(0.1)
      recStr = ser.readline().decode('utf-8')
      print(recStr)
      a = a+1
      b = b+1
      c = c+1
```

8. Simpan dan jalankan file tersebut.

Di jendela shell Python, Anda akan melihat output seperti ini:

```
1,2,3
1,2,3,6
```

Output terus bertambah sampai Anda menghentikan program. Ada beberapa elemen baru di sini, tetapi, sebagian besar, itu tidak jauh berbeda dari yang kami lakukan sebelumnya. Mari kita lihat beberapa elemen baru. Perbedaan pertama adalah kita mengimpor perpustakaan waktu. Pustaka ini menyediakan banyak fungsi terkait waktu. Dalam latihan ini, kita tertarik pada fungsi `sleep()`. Fungsi `sleep()` menjeda pemrosesan selama beberapa detik yang disediakan. Seperti yang Anda lihat dalam kode kami, kami ingin menjeda pemrosesan selama 0,5 detik. Ini memberi kedua sisi waktu aliran serial untuk memproses buffer mereka. Jika Anda mengomentari baris itu dan menjalankan program lagi, Anda akan mendapatkan beberapa hasil yang menarik. Cobalah.

```
valList = [str(a),str(b),str(c)]
```

Di sini kita mengambil variabel kita dan memasukkannya ke dalam daftar. Pada langkah selanjutnya, ketika kita menggabungkan elemen menjadi satu string, bilangan bulat harus berupa string. Jadi, kami melanjutkan dan melakukan konversi di sini.

```
sendStr = ','.join(valList)
```

Selanjutnya, kami menggunakan metode join() dari kelas string untuk mengubah daftar menjadi string. Perhatikan bagaimana metode join() dilampirkan ke string ','. join adalah metode kelas string, bukan kelas daftar, jadi Anda harus memanggil itu dari sebuah string. Karena operasi sebenarnya bekerja pada daftar, bukan string, Anda harus menyediakan string agar berfungsi. Dalam hal ini, string yang disediakan adalah pemisah yang Anda inginkan di antara setiap anggota daftar. Ini bisa berupa karakter apa saja, tetapi agar parseInt() berfungsi di sisi Arduino, karakternya harus non-alfanumerik.

```
ser.write(sendStr.encode('utf-8'))
```

Perbedaan catatan lainnya adalah di mana kami mengirim data ke Arduino menggunakan metode write(). Ini bekerja seperti metode Serial.println() di Arduino. Perbedaan terbesar adalah Anda harus menyandikan string sebelum Anda dapat mengirimkannya.

5.19 PINGUINO

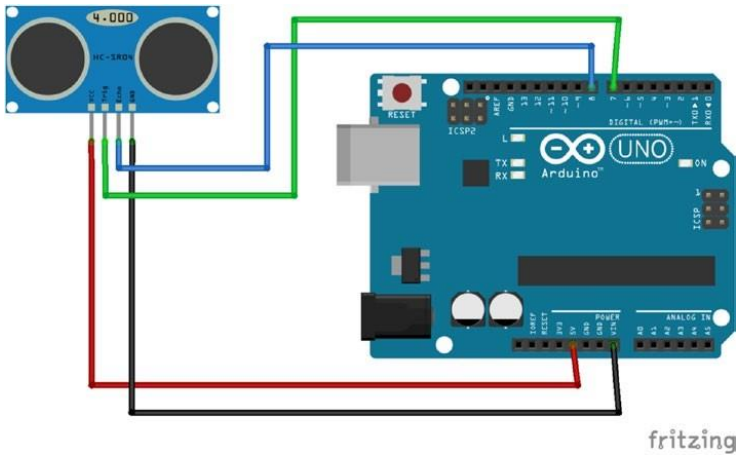
Kasus penggunaan umum untuk memasang satu atau lebih sensor untuk mendeteksi dunia di sekitar robot. Untuk latihan berikutnya, kita akan memasang pengintai ultrasonik HC-

SR04 di Arduino dan mengirim informasi jarak kembali ke Pi sebagai string serial. Untuk melakukan ini, kita perlu membuka koneksi serial antara dua papan. Arduino memicu sensor, dan, seperti di bengkel kami sebelumnya, membaca pulsa yang dikembalikan. Kami akan menghitung jarak, dan kemudian mengirimkan hasilnya ke Pi. Di sisi Pi, kita hanya akan memiliki program yang mendengarkan port serial dan kemudian mencetak apa pun yang dibacanya dari Arduino.

5.20 MENYIAPKAN SIRKUIT

Menyiapkan sirkuit tidak bisa lebih mudah. Faktanya, kami tidak menggunakan papan tempat memotong roti. Kami akan menghubungkan sensor langsung ke header Arduino (lihat Gambar 5-9).

1. Hubungkan VCC ke pin 5V pada Arduino.
2. Hubungkan GND ke salah satu pin GND di Arduino. Tidak masalah yang mana, tetapi ada dua yang berdekatan dengan pin 5V.
3. Hubungkan TRIG ke pin 7 pada Arduino.
4. Hubungkan ECHO ke pin 8 pada Arduino.



Gambar 5-9. Tata letak sirkuit latihan Pinguino

Kode

Kita perlu menulis kode untuk kedua papan agar ini berfungsi. Di Arduino, kami memicu sensor ultrasonik dan menangkap sinyal kembali. Kami kemudian akan mengubahnya menjadi sentimeter dan mencetak nilainya ke port serial. Pi membaca baris dari port serial dan mencetak hasilnya ke jendela shell Python.

Arduino

1. Buka jendela sketsa baru dan simpan sebagai `serial_test`.
2. Masukkan kode berikut:

```
int trig = 7;
int echo = 8;
int duration = 0;
int distance = 0;
void setup() {
      Serial.begin(9600);
      pinMode(trig, OUTPUT);
      pinMode(echo, INPUT);

      digitalWrite(trig,LOW);
}
void loop() {
      digitalWrite(trig, HIGH);
      delayMicroseconds(10);
      digitalWrite(trig, LOW);
      duration = pulseIn(echo, HIGH);
      distance = duration/58.2;

      Serial.write(distance);
      delay(500);
   }
```

Simpan dan unggah sketsa ke Arduino.

Raspberry Pi

1. Buka file IDLE baru dan simpan sebagai serial_test.py.
2. Masukkan kode berikut:

```
import serial
import time
ser = serial.Serial('/dev/ttyAMC0', 9600)
while 1:
      recSer = ser.readline().decode('utf-8')
      recSer.rstrip()
      distance = int(recSer + '0')/10
      print("Distance: " + str(distance) + "cm",end = '\r')
      time.sleep(0.5)
```

3. Simpan dan jalankan file tersebut.

Anda sekarang akan melihat teks di jendela shell Python dengan jarak dalam sentimeter. Kode ini mengeluarkan hasil dari sensor ultrasonik tunggal. Pada kenyataannya, robot Anda seharusnya memiliki tiga atau lebih sensor yang mengarah ke depan pada sudut yang berbeda. Pasalnya, sensor ultrasonik bekerja dengan baik selama rintangan berada tepat di depan robot. Jika robot mendekati dinding atau rintangan lain pada sudut miring, suara tidak memantul kembali ke sensor. Memiliki lebih dari satu sensor pada sudut yang berbeda memungkinkan robot untuk mendeteksi rintangan yang tidak langsung di depannya.

5.21 RINGKASAN

Menambahkan Arduino ke Raspberry Pi memberi Anda kemungkinan yang jauh lebih luas. Anda akan dapat menambahkan lebih banyak sensor dan LED daripada yang dapat Anda lakukan dengan Pi sendiri. Di antara manfaatnya adalah peningkatan jumlah input analog, lebih banyak output PWM, dan lebih banyak lagi output digital. Arduino sangat memprogram. Jika Anda sudah terbiasa dengan C atau C++, menulis untuk Arduino pasti sudah sangat familiar. Namun, sangat penting untuk mengingat perbedaan antara Arduino dan Python. Python tidak menggunakan karakter di akhir baris, tetapi Arduino mengakhiri setiap baris dengan titik koma. Ada sedikit lebih banyak sintaks yang terlibat dengan penulisan conditional dan loop. Dan blok kode terdapat dalam kurung kurawal. Lekukan tidak penting di Arduino, tetapi Python tidak akan dikompilasi jika lekukan Anda mati.

Terlepas dari perbedaan ini, ada beberapa hal yang membuat Arduino lebih mudah. Komunikasi serial tidak memerlukan banyak pengaturan dan perintah serial adalah bagian dari perpustakaan inti Arduino. Dengan Python, Anda harus mengimpor perpustakaan serial. Keduanya membuat penulisan ke port serial cukup sederhana. Python, bagaimanapun, membutuhkan penyandian dan penguraian kode ke utf-8 agar berguna. Selain itu, Arduino membuat penguraian angka dalam satu baris dari aliran serial menjadi mudah dengan metode parseInt(). Mendapatkan nomor dari string dengan Python membutuhkan sedikit manipulasi lembut. Saat Anda bekerja dengan Arduino, jangan lupa dukungan komunitasnya luar biasa. Sangat sedikit yang belum dilakukan dan didokumentasikan oleh orang lain. Juga ingat bahwa Anda memiliki hak sumber daya yang besar di IDE dalam bentuk kode contoh. Manfaatkan itu. Dan saat Anda menambahkan lebih banyak perpustakaan untuk lebih banyak perangkat, Anda menemukan lebih banyak contoh sketsa untuk membantu Anda.

BAB 6
MENGEMUDI MOTOR

Dalam Bab 4, kami menggunakan pin GPIO Raspberry Pi untuk mengontrol LED dan menerima informasi dari sensor ultrasonik. Di Bab 5, kita melihat Arduino dan mendiskusikan mengapa ini adalah pilihan yang lebih baik untuk fungsi GPIO umum. Kami menghubungkan pengintai ultrasonik dan LED ke Arduino dan belajar bagaimana melewatkan data antara dua papan. Tapi itu tidak berarti kita sudah selesai dengan header GPIO Raspberry Pi.Dalam bab ini, kita akan menggunakan pin GPIO untuk menghubungkan ke papan yang disebut driver motor, yang dirancang untuk berinteraksi dengan motor DC dan stepper. Saya akan membahas beberapa jenis motor yang berbeda, dan mendiskusikan apa itu pengemudi motor dan mengapa itu penting dalam apa yang kita lakukan. Kami akan menghubungkan motor DC ke pengontrol motor dan menulis program kecil untuk membuatnya berputar. Sebagai bagian dari contoh program, kita akan melihat bagaimana mengontrol kecepatan dan arah motor. Kami juga akan melihat beberapa properti dari pengontrol motor tertentu yang dipilih untuk bengkel.

Anda dapat memilih untuk tidak menggunakan pengontrol motor yang disarankan, jadi kami juga akan melihat alternatif umum: driver motor L298N. Papan driver, yang tersedia dari banyak produsen, dirancang untuk terhubung ke chip pengontrol H-bridge L298N di intinya. Tetapi karena papan ini mengandalkan sinyal PWM untuk mengatur kecepatan, kita harus menghubungkannya melalui Arduino. Saya akan membahas semua itu menjelang akhir bab ini. Pada akhir lokakarya ini, Anda akan memiliki komponen terakhir yang diperlukan untuk mulai membuat robot: gerak. Di Bab 7, kami akan menyatukan semuanya dengan kit sasis untuk membuat robot Anda bergerak.

6.1 MOTOR & PENGEMUDI

Sebelum beralih ke pengontrol motor, mari luangkan waktu sejenak untuk melihat apa yang kita kendalikan. Driver yang kami gunakan dirancang untuk motor DC sederhana, meskipun dapat juga digunakan untuk menggerakkan stepper. Mari kita lihat driver dan motor di bagian ini.

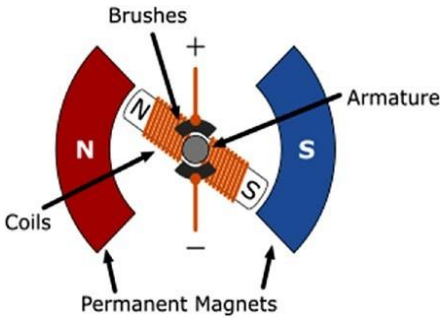
6.2 JENIS MOTOR

Motor mengubah energi listrik menjadi energi putaran. Mereka datang dalam berbagai jenis, dan mereka menggerakkan hampir semua yang bergerak. Jenis motor yang paling umum adalah motor DC sederhana, yang bahkan digunakan di banyak jenis motor lainnya. Misalnya, servomotor adalah perangkat yang menggabungkan motor DC dengan potensiometer, atau perangkat umpan balik lainnya, dan roda gigi untuk mengontrol gerakan yang tepat, baik itu sudut atau arah. Jenis motor lainnya termasuk stepper, yang menggunakan impuls listrik untuk mengontrol gerakan yang sangat tepat, dan motor tanpa biji, yang mengatur ulang

bagian khas motor DC untuk meningkatkan efisiensi.

Motor DC

Motor DC terdiri dari serangkaian kumparan dalam medan magnet. Ketika muatan listrik ditempatkan pada kumparan, itu menyebabkan kumparan berputar pada sumbu bersamanya. Motor sederhana memiliki kumparan yang diatur dan dipasang di sekitar poros tengah. Saat poros dan kumparan berputar, konektivitas listrik dipertahankan dengan sikat yang membuat kontak dengan poros. Poros, pada gilirannya, menonjol dari rakitan untuk menggunakan gaya rotasi untuk melakukan pekerjaan. Gambar 6-1 menunjukkan motor DC tipikal.

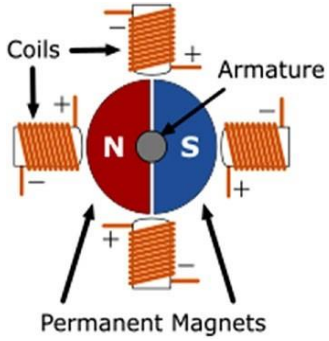


Gambar 6-1. operasi motor DC

Anda biasanya menemukan motor ini terpasang pada gearbox, sabuk, atau rantai yang berfungsi untuk memperkuat torsi motor dengan mengorbankan kecepatan rotasi. Hal ini dilakukan karena motor DC telanjang dapat menghasilkan banyak kecepatan, tetapi kecepatan mentah jarang berguna. Motor yang kami gunakan adalah jenis ini. Mereka adalah motor DC sederhana yang terpasang pada gear box.

Motor Tanpa Sikat

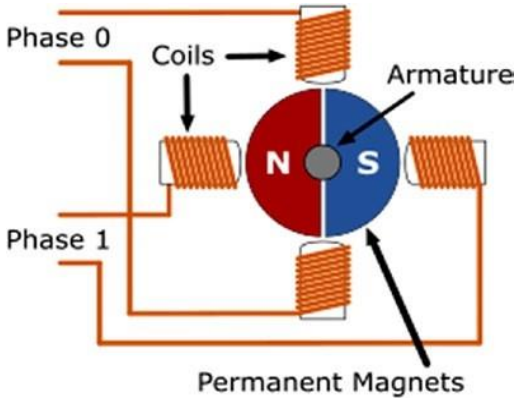
Jenis motor lain menggerakkan sambungan mekanis ke magnet. Kumparan tetap statis. Ketika muatan listrik diterapkan, magnet berputar di sekitar kumparan pada sumbu yang sama (lihat Gambar 6-2). Ini menghilangkan kebutuhan akan sikat, sehingga disebut motor tanpa sikat. Di dunia hobi, motor brushless paling sering dikaitkan dengan pesawat multirotor. Mereka juga digunakan secara luas di area lain di mana kecepatan dan efisiensi tinggi diperlukan, seperti di spindel CNC (dikontrol numerik komputer). Anda mungkin akrab dengan alat atau router Dremel; kedua perangkat ini adalah jenis spindel dan menggunakan motor brushless.



Gambar 6-2. Pengoperasian motor tanpa sikat

Motor Stepper

Semua motor yang saya bahas sejauh ini memiliki satu atau lebih kumparan yang bekerja dengan satu muatan listrik. Muatan itu bisa positif atau negatif, mengubah arah motor. Motor stepper berbeda. Stepper menggunakan beberapa kumparan dengan muatan yang berbeda (lihat Gambar 6-3), yang memecah putaran penuh menjadi beberapa langkah. Dengan memanipulasi muatan ini, kita dapat menyebabkan motor bergerak ke dan menahan posisi di salah satu langkah. Hal ini membuat motor ini sangat berguna untuk kontrol terbatas dalam aplikasi seperti mesin CNC, printer 3D, dan robotika.



Gambar 6-3. Pengoperasian motor stepper

Servo

Servo adalah motor yang bergerak ke sudut tertentu dan menahan posisi itu. Mereka umumnya memiliki rotasi maksimum 45 hingga 90 derajat di kedua arah. Mereka melakukan ini dengan menghubungkan potensiometer ke gigi keluaran akhir. Potensiometer memberikan umpan balik ke papan kontrol internal. Ketika servo menerima sinyal, biasanya dalam bentuk PWM, motor berputar hingga potensiometer dan sinyal seimbang. Gambar 6-4 menunjukkan servo hobi yang khas.

Gambar 6-4. Servomotor umum

Servo dengan limiter dan potensiometer dihilangkan disebut servos rotasi kontinu. Mereka digunakan dalam aplikasi di mana torsi diperlukan. Banyak robot digerakkan oleh servos rotasi terus menerus. Ini adalah contoh di mana satu hobi sangat menguntungkan yang lain. Servo hobi umum pada awalnya digunakan untuk pesawat RC hobi. Karena sebagian

besar penghobi tidak mampu membeli perangkat mahal untuk mengontrol kerajinan mereka, mereka menemukan cara untuk menurunkan harga secara signifikan. Ini, tentu saja, membantu kami para robotika hobi.

6.3 PROPERTI MOTOR

Ada beberapa hal yang perlu diingat tentang motor dalam proyek kami. Yang paling penting adalah sifat kelistrikan motor—khususnya tegangan dan arus.

Voltase

Anda sudah agak akrab dengan tegangan, yang merupakan ukuran listrik yang dibutuhkan untuk mengoperasikan perangkat. Pi ini didukung oleh 5 volt tetapi berjalan pada 3,3 volt. Arduino berjalan pada 5 volt, dipasok oleh port USB dari Pi. Motor yang kami gunakan berjalan pada 6 volt. Penting untuk menjaga tegangan ini tetap lurus. Jika Anda memasang 5 volt pada perangkat yang beroperasi pada 3,3 volt, Anda dapat merusak perangkat Anda. Ada perangkat yang dirancang khusus untuk membantu mengelola voltase di proyek Anda. Regulator tegangan (menaikkan atau menurunkan) mempertahankan tegangan konstan. Regulator 5V 7805 umum membutuhkan 6 hingga 12 volt dan mengubahnya menjadi 5 volt. Kelebihan energi dihamburkan sebagai panas, dan mereka bisa menjadi sangat panas.

Regulator tegangan sangat bagus untuk suplai tegangan, tetapi tidak banyak digunakan untuk menerjemahkan 5 volt dan 3,3 volt dalam perangkat. Untuk ini, kami menggunakan konverter level logika, yang memerlukan tegangan referensi dari kedua perangkat, tetapi menerjemahkan tegangan antar perangkat dengan aman. Jadi, sekarang Anda mengetahui perbedaan kebutuhan voltase perangkat Anda. Selanjutnya, kita melihat ampere.

Ampere

Arus listrik adalah ukuran arus, atau tekanan listrik yang dibutuhkan perangkat kita untuk beroperasi. Analogi yang paling umum adalah air melalui pipa, di mana tegangan adalah ukuran pipa dan ampere adalah jumlah air yang mengalir melaluinya. Saya sebenarnya suka mengubah analogi menggunakan tabung karet. Jika Anda mencoba untuk mendorong terlalu banyak air melalui tabung karet, hal-hal buruk terjadi. Di dunia elektronik, ini sering diukur dalam satuan miliampere yang lebih kecil, biasanya dicatat sebagai mA. Misalnya, port USB pada sebagian besar perangkat dibatasi hingga 800mA daya. Ini terjadi pada jumlah daya yang sama yang digunakan oleh motor yang kami pilih; namun, itu tidak memperhitungkan lonjakan daya.

Tegangan pada perangkat agak pasif. Perangkat menggunakan tegangan yang Anda berikan, tidak pernah mencoba menarik lebih banyak. Ampere justru sebaliknya. Sebuah perangkat haus akan arus listrik dan terus menarik apa yang dibutuhkannya sampai puas untuk melakukan pekerjaannya, atau melebihi pasokan yang tersedia. Komponen dan perangkat memiliki sejumlah daya yang mereka butuhkan untuk bekerja. Mereka juga memiliki jumlah kekuatan maksimum yang dapat mereka tahan. Karena daya listrik ekstra diubah menjadi panas, jika Anda melebihi arus listrik maksimum yang dapat ditoleransi

perangkat, perangkat menjadi panas dan mati. Terkadang secara spektakuler. Moral dari cerita ini adalah untuk “selalu memperhatikan arus yang Anda gambar.” Dan ini tidak hanya berlaku untuk motor. LED terkenal karena menarik banyak arus.

Motor dan Ampere

Motor adalah perangkat yang terkenal haus daya. Mereka terus-menerus berusaha memenuhi tujuan mereka: berputar. Ketika tidak ada beban, berat, atau hambatan pada motor, motor berputar dengan gembira pada penarikan arus minimumnya. Akan tetapi, mulailah menambah hambatan, dan motor menarik lebih banyak arus hingga ia mencapai maksimum yang dapat ditariknya, yang disebut arus stall. Stall current pada dasarnya adalah ampere motor ketika poros secara fisik ditahan untuk tidak bergerak.

Ketika sebuah motor mulai, dengan cepat mengubah arah, atau menghadapi terlalu banyak hambatan untuk berputar, daya yang dikonsumsinya meningkat secara drastis. Jika undian mendadak ini terlalu banyak untuk suplai, ada sesuatu yang rusak. Mari kita ambil sumber 800mA, seperti jack USB; jika motor tiba-tiba menarik 1 amp atau lebih besar, jack USB mungkin akan rusak.

6.4 PENGEMUDI MOTOR

Kebanyakan mikrokontroler, mikroprosesor, dan elektronik hanya dapat menangani sejumlah kecil arus. Jika seseorang menarik terlalu banyak arus, ia mulai terbakar. Karena motor biasanya dengan mudah melebihi arus maksimum ini, Anda biasanya tidak ingin menghubungkan motor dengan ukuran yang signifikan secara langsung ke prosesor Anda. Jadi, kita akan menggunakan perangkat yang disebut driver motor atau pengontrol motor. Pengendali motor dirancang untuk tujuan khusus ini. Ini menggunakan sinyal daya rendah dari mikrokontroler Anda untuk mengontrol arus dan/atau tegangan yang jauh lebih besar. Dalam kasus kami, kami menggunakan pengontrol motor untuk mengontrol 6 volt dengan 3,3 volt dari pin GPIO. Kami melakukan ini melalui serangkaian komponen yang memiliki toleransi arus 1,2A (1,200mA) yang jauh lebih besar dan dapat menangani lonjakan singkat hingga 3,0A (3.000mA) .

6.5 BEKERJA DENGAN PENGENDALI MOTOR

Mari kita lihat dua pengontrol motor. Yang pertama adalah HAT DC & Stepper Motor by Adafruit. Papan pengontrol ini dirancang khusus untuk dipasang ke Raspberry Pi. Kombinasi utilitas dan kenyamanan menjadikannya pilihan pilihan saya untuk proyek seperti milik kami. Pengendali motor lainnya adalah L298N, yang merupakan IC H-bridge. Meskipun L298N sebenarnya adalah komponen yang tersembunyi sebuah chip, ada banyak pabrikan yang telah membuat papan breakout yang nyaman. Jenis papan ini biasanya dirujuk ketika seseorang menyebutkan pengontrol motor L298N. Yang digunakan dalam buku ini adalah versi generik yang saya temukan di Amazon seharga Rp 75.000. Beberapa teman saya mengatakan bahwa saya membayar terlalu banyak untuk itu.

6.6 ADAFRUIT DC & STEPPER MOTOR HAT

Pengemudi motor dalam proyek ini adalah salah satu dari Adafruit tersedia di www. adafruit.com/products/2348. Informasi cara menggunakannya ada di https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi.

Faktanya, banyak dari apa yang akan kita bahas berasal dari situs web Adafruit ini. Ada beberapa alasan mengapa perangkat ini dipilih untuk robot kami; Tak terkecuali yang dipasang langsung ke Raspberry Pi, sehingga membatasi area yang dibutuhkan untuk memasang elektronik pada robot. Seperti yang akan Anda pelajari dengan cepat, ruang pemasangan sangat mahal di sebagian besar robot; terutama jika Anda mencoba membuatnya agak kompak. Berikut ini adalah beberapa alasan lain untuk menggunakan papan ini:

- Dapat mengontrol hingga empat motor DC atau dua motor stepper.
- Komunikasi ditangani melalui saluran serial I2C, yang memungkinkan banyak perangkat untuk ditumpuk (inilah sebabnya kami menggunakan pin yang lebih panjang pada header).
- Karena menggunakan I2C, ia memiliki modul PWM khusus untuk mengendalikan motor, jadi kita tidak harus bergantung pada PWM pada Pi yang tepat.
- Memiliki empat sirkuit kontrol motor H-Bridge dengan arus 1,2A, arus puncak 3,0A, dengan penghentian termal, dan dioda pelindung internal untuk melindungi papan Anda.
- Ada empat kontrol motor dua arah dengan kontrol kecepatan 8-bit (0 hingga 255).
- Ada koneksi yang mudah dengan penggunaan blok terminal.
- Ada library Python yang sudah jadi.

Beberapa Perakitan Diperlukan

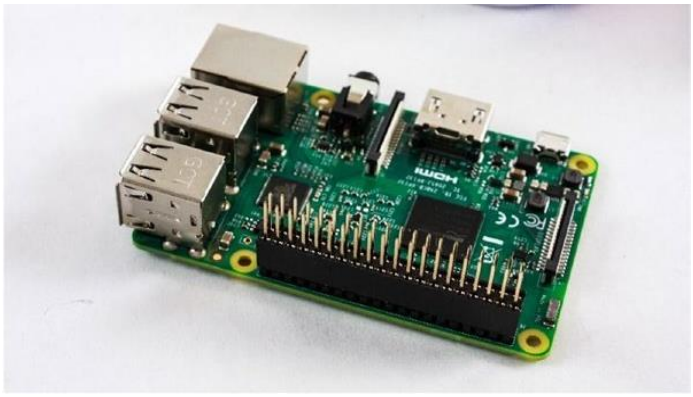
Papan datang dalam kit dan membutuhkan penyolderan. Jika Anda belum melakukannya, Anda perlu merakitnya sebelum melanjutkan proyek. Ingat, kami menentukan pin yang lebih panjang untuk header, jadi jangan gunakan pin yang disertakan dengan kit. Saatnya latihan menyolder. Ada banyak pin kecil (40 di antaranya) yang perlu disolder. Jika Anda tidak terbiasa dengan penyolderan, Anda perlu meluangkan waktu untuk mempelajari caranya. Meskipun sangat mudah, instruksi menyolder berada di luar cakupan buku ini. Ada banyak video bermanfaat yang tersedia di Internet. Saya juga sangat menyarankan agar Anda menemukan ruang pembuat lokal Anda. Pasti ada seseorang di sana yang bisa memberi Anda pelajaran singkat. Gambar 6-5 menunjukkan pengaturan penyolderan sederhana saya.

Gambar 6-5. Persiapan merakit motor HAT

Merakit HAT Motor sangat mudah, meskipun ada penyolderan yang terlibat. Anda dapat menemukan petunjuk rinci untuk perakitan di situs web Adafruit di https://learn.adafruit.com/adafruit-dc-and-stepper-motor-hat-for-raspberry-pi/assembly.

Untuk latihan ini, Anda membutuhkan besi solder dan solder. Saya merekomendasikan memiliki fluks yang berguna, serta sesuatu untuk menjaga ujung solder tetap bersih. Kembali di sekolah, kami menggunakan spons basah untuk membersihkan ujungnya, tetapi ada hal-hal yang lebih baik yang dibuat untuk pekerjaan itu sekarang. Raspberry Pi Anda juga akan membantu.

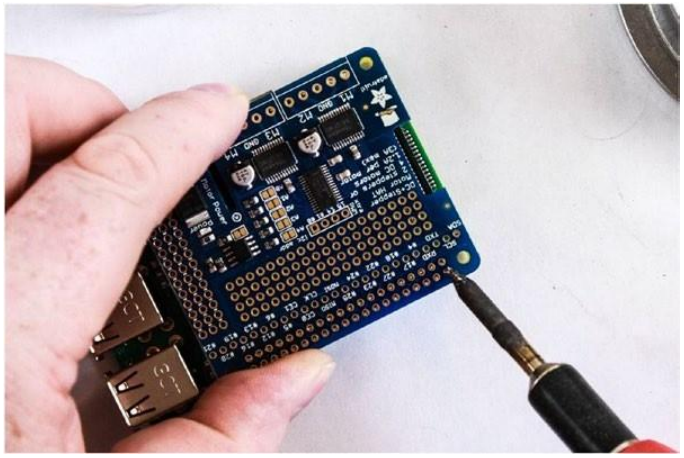
1. Pasang header yang diperluas ke header 40-pin Raspberry Pi (lihat Gambar 6-6). Ini membantu menstabilkan berbagai hal saat Anda menyolder.



Gambar 6-6. Header susun yang diperluas pada GPIO 40-pin Pi

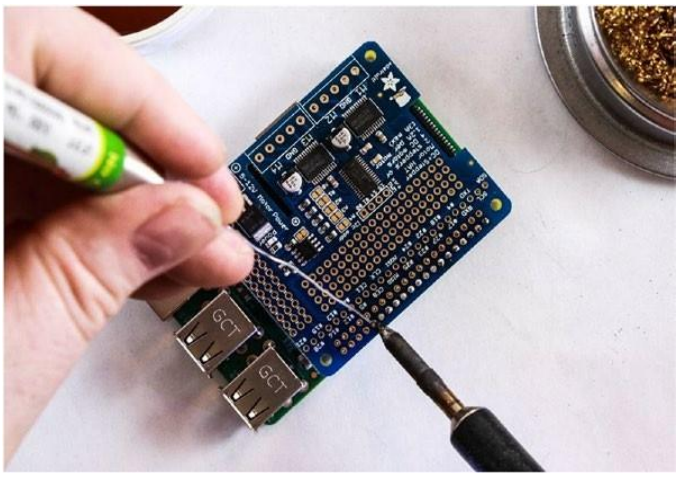
2. Pasang papan sirkuit Motor HAT ke header (lihat Gambar 6-7). Untuk membantu memegang papan pada sudut yang lebih baik untuk menyolder, Anda mungkin ingin meletakkan sesuatu untuk menopang sisi lainnya. Salah satu blok terminal bekerja dengan baik untuk ini.
3. Solder pin pertama.
4. Setelah pin pertama disolder, panaskan kembali dan sesuaikan papan agar terpasang dengan benar (lihat Gambar 6-8). Saat solder untuk pin mendingin, itu akan menahan papan pada sudut yang tepat saat Anda menyolder sisa pin. Jika Anda menopang papan dengan blok terminal atau sesuatu yang lain sehingga papan duduk tegak, Anda mungkin dapat melewati langkah ini.

Gambar 6-7. Papan sirkuit dipasang di header



Gambar 6-8. Menyesuaikan penempatan dan sudut papan

5. Solder sisa baris pertama (lihat Gambar 6-9). Anda menginginkan sambungan yang bagus, bersih, dan berkilau.

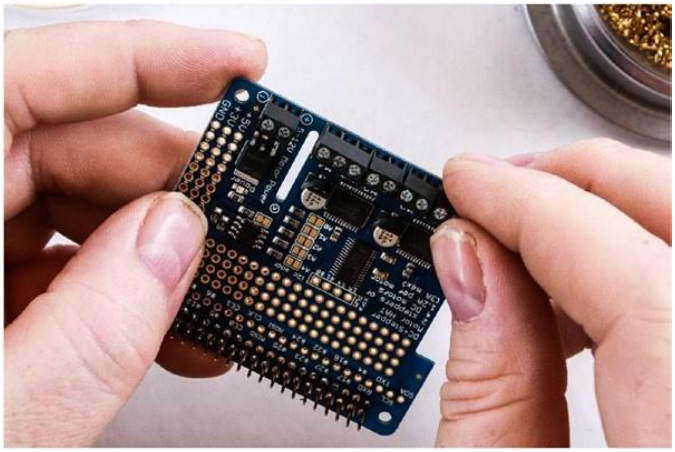


Gambar 6-9. Solder baris pertama pin

6. Putar papan 180 derajat dan solder baris kedua (lihat Gambar 6-10).

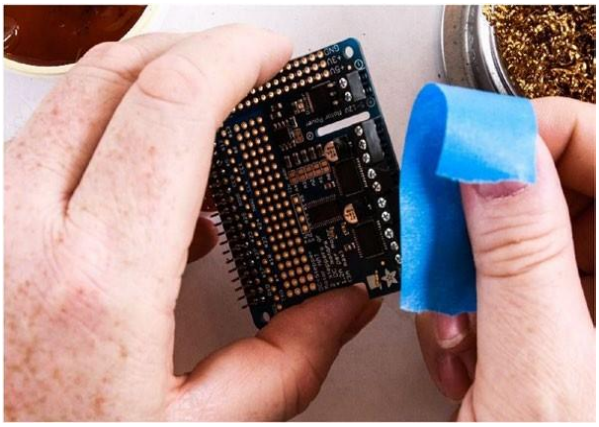
Gambar 6-10. Putar Pi dan solder pin yang tersisa

7. Lepaskan HAT dari Pi.
8. Pasang terminal sekrup ke papan (lihat Gambar 6-11).



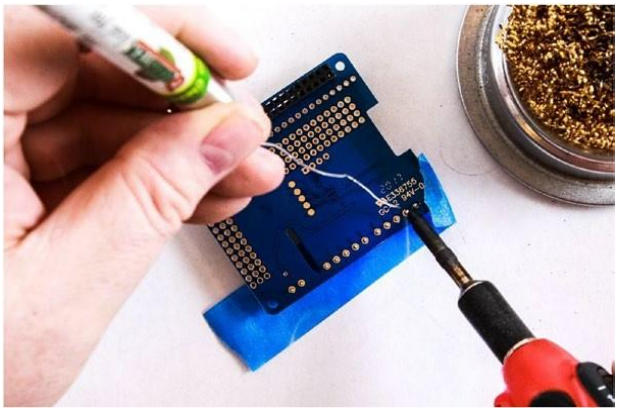
Gambar 6-11. Menambahkan blok terminal ke papan sirkuit

9. Gunakan selotip untuk menahan terminal pada tempatnya saat Anda membalik papan (lihat Gambar 6-12).



Gambar 6-12. Pita membantu menahan blok terminal di papan saat Anda membaliknya untuk menyoldernya ke tempatnya

10. Solder terminal pada tempatnya (lihat Gambar 6-13).



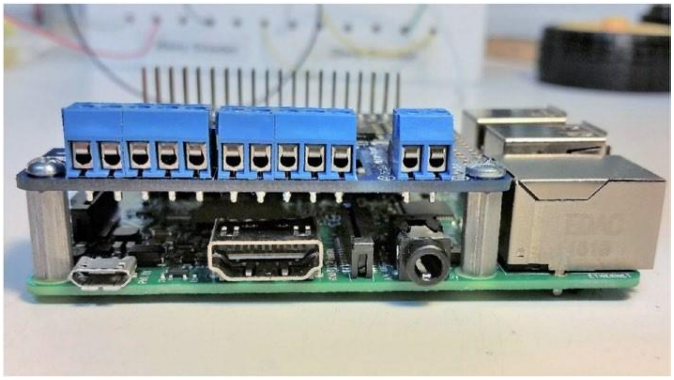
Gambar 6-13. Menyolder pin terminal

Setelah Anda melepaskan rekaman itu, Anda selesai. HAT Motor siap digunakan. Pasang HAT ke Pi. Anda ingin menyangga sisi dengan terminal sehingga tidak terjadi short di rumah HDMI (lihat Gambar 6-14).

Gambar 6-14. Papan yang telah selesai dipasang pada Raspberry Pi. Bagian dukungan oranye dicetak 3D.

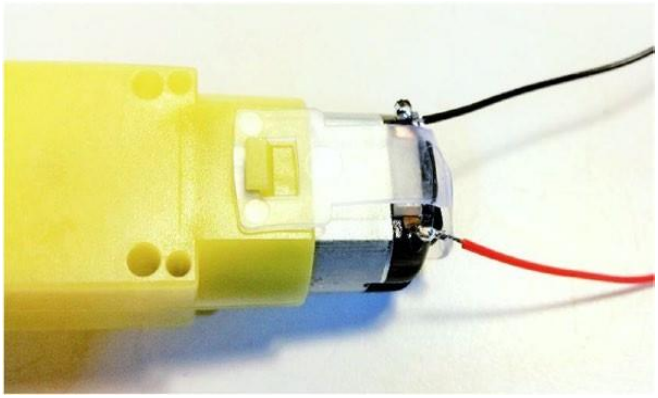
Menghubungkan Pengendali Motor

Menghubungkan HAT Motor cukup mudah. Cukup pasang papan pada header GPIO dari Pi. Namun, ada beberapa hal yang perlu diperhatikan. Pertama, berhati-hatilah untuk tidak menekuk salah satu pin pada Raspberry Pi atau HAT. Hal ini sangat mudah dilakukan. Pin header pada HAT Motor sangat rentan terhadap tekukan. Anda juga ingin berhati-hati untuk tidak menyingkat blok terminal. Anda akan melihat bahwa saat dipasang, sambungan solder sangat dekat dengan rumah logam sambungan HDMI (lihat Gambar 6-15). Ada dua solusi mudah untuk ini. Perbaikan pertama (dan inilah yang akan kami lakukan di bengkel sampai Anda dapat menerapkan solusi kedua) adalah dengan hanya menempatkan sepotong pita listrik di atas rumah logam dari micro USB dan Konektor HDMI dari Pi. Perbaikan kedua (disarankan) adalah mendapatkan beberapa offset untuk mendukung sisi HAT itu. Spacer dan sekrup juga akan melakukan pekerjaan itu. Intinya adalah Anda tidak ingin papan melorot dan bersentuhan dengan rumahan, yang mungkin akan menghasilkan pertunjukan cahaya singkat dan penghancuran Motor HAT dan Pi.



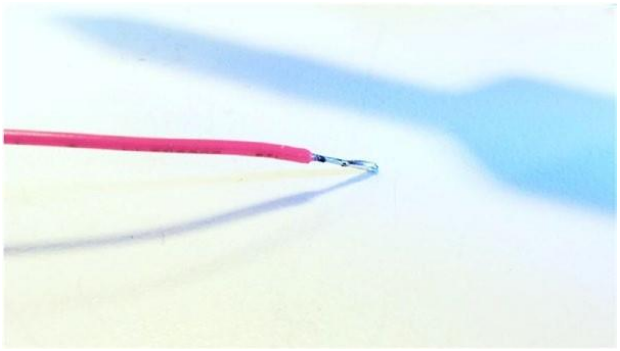
Gambar 6-15. Adafruit Motor HAT dipasang di Raspberry Pi

Setelah papan dipasang, dan diisolasi dengan aman dari korslet, saatnya untuk menghubungkan motor. Untuk tutorial pertama, kita hanya akan menggunakan satu motor. Bagian kedua dari kode mengontrol dua, jadi sebaiknya kita menghubungkannya sekarang. Tapi sebelum kita bisa melakukan itu, kita harus mempersiapkan motor kita. Sekarang, jika motor Anda dilengkapi dengan kabel yang terpasang, maka Anda berada di depan permainan. Jika tidak, Anda harus menyolder kabel ke motor Anda, seperti yang ditunjukkan pada Gambar 6-16. Saya cenderung menggunakan kabel hitam dan merah dengan ukuran yang sesuai untuk motor yang bersangkutan. Saya juga ingin memastikan bahwa kabel pada masing-masing motor cocok (kabel hitam menuju ke kutub yang sama di masing-masing motor, dan merah ke kutub yang sama di setiap motor). Dengan cara ini saya tidak perlu menebak-nebak bagaimana hal-hal terhubung nanti.



Gambar 6-16. Memimpin disolder ke terminal motor

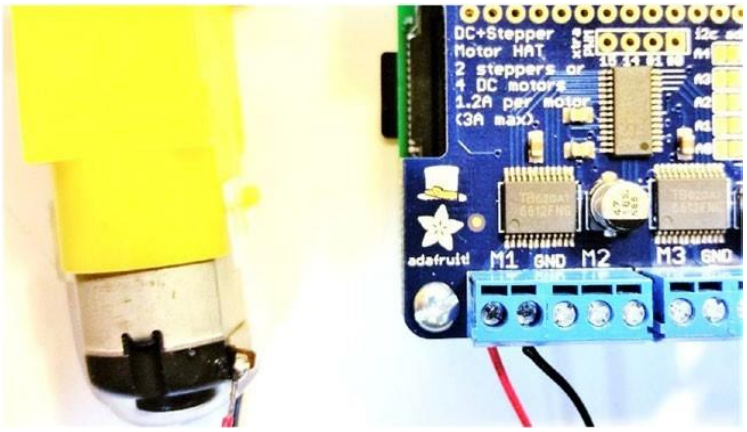
Dalam hal ini, saya menggunakan kawat untai 26AWG. Umumnya ada dua jenis kawat timah: terdampar dan padat. Solid lebih kaku dan sangat baik untuk jumper atau situasi di mana tidak akan ada banyak momen. Stranded wire terdiri dari beberapa kabel tipis yang ditempatkan dalam satu selubung. Ini lebih fleksibel dan ideal untuk aplikasi di mana kemungkinan akan ada pergerakan. Kawat terdampar sedikit lebih sulit untuk dikerjakan dan ujung-ujungnya yang masuk ke blok terminal harus dikalengkan, atau dilapisi dengan solder (lihat Gambar 6-17). Ini membuat ujung itu kaku, dan menghubungkan lebih baik di blok terminal.



Gambar 6-17. Timbal kaleng

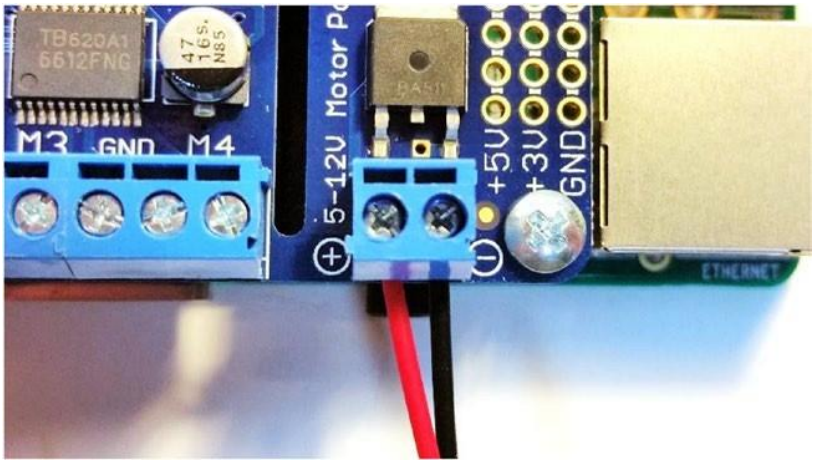
Langkah selanjutnya menghubungkan motor ke blok terminal.

1. Pastikan blok terminal terbuka dengan sekrup di dalam blok sampai ke atas. Pastikan untuk tidak melepas sekrup. Anda harus menggunakan obeng kepala Phillips yang cukup bagus.
2. Masukkan salah satu timah kaleng ke dalam lubang di sisi blok terminal bertanda M1 (lihat Gambar 6-18). Tidak masalah, kabel mana yang menuju ke port mana, selama kedua kabel tersebut menuju ke port yang berbeda untuk driver yang sama (dalam hal ini, M1) .
3. Kencangkan sekrup yang sesuai dengan lubang tempat Anda memasukkan kabel.
4. Ulangi prosedur untuk kabel kedua dari motor.



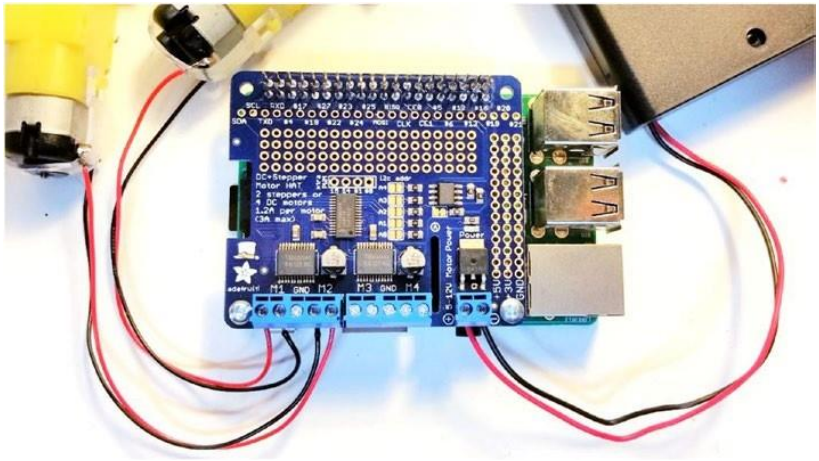
Gambar 6-18. Motor terhubung ke Motor HAT

Pada titik ini, jika Anda ingin, Anda juga dapat menghubungkan motor kedua. Saya cenderung membalik urutan lead karena mereka ditakdirkan untuk sisi berlawanan dari robot. Anda ingin perintah maju untuk memutar motor kiri ke satu arah dan motor kanan ke arah lain. Jika keduanya berputar ke arah yang sama secara elektrik, maka robot hanya berputar di tempat. Anda akan mengulangi prosedur untuk paket baterai empat AA ke terminal daya. Pastikan bahwa kabel merah mengarah ke sisi positif (+) dan hitam mengarah ke sisi negatif (–) (lihat Gambar 6-19).



Gambar 6-19. Paket baterai eksternal terhubung ke Motor HAT

Papan dan motor Anda akan terlihat seperti Gambar 6-20. Dengan kedua motor dan baterai terhubung, Anda siap untuk mulai coding!



Gambar 6-20. Koneksi selesai ke Motor HAT

Menggunakan Motor Hat

Dengan HAT Motor terpasang, dan motor serta catu daya motor Anda terhubung, saatnya untuk mem-boot Pi dan masuk.

Memasang Perpustakaan

Setelah Anda memulai dan terhubung ke Pi Anda, Anda perlu menginstal pustaka Python untuk Motor HAT. Ini tersedia dari situs Adafruit GitHub.

1. Buka jendela terminal.
2. Arahkan ke direktori kode Python Anda. Dalam kasus saya, ini adalah `TRG-RasPi_Robot`.
3. Masukkan yang berikut ini:
   ```
   git clone https://github.com/adafruit/Adafruit-Motor-HAT-Python-Library.git
   cd Adafruit-Motor-HAT-Python-Library
   ```
4. Instal pustaka pengembangan Python.
   ```
   sudo apt-get install python-dev
   ```
5. Instal perpustakaan Motor HAT.
   ```
   sudo python setup.py install
   ```

Pada titik ini, Pi Anda diperbarui dengan pustaka yang diperlukan, dan inilah saatnya untuk memulai dengan kode.

Kode

Sebelum kita mulai coding, ada satu catatan singkat tapi penting. Sebelumnya, kami menggunakan Python 3 untuk hampir semua hal. Untuk workshop ini, kita akan menggunakan Python 2.7. Mengapa? Nah, perpustakaan default yang disediakan oleh Adafruit ada di Python 2.7. Mungkin ada pustaka Python 3.x, tetapi kami menggunakan pustaka default untuk

lokakarya ini.

Seperti yang Anda temukan di lokakarya sebelumnya, kapan pun Anda menggunakan pin GPIO, Anda perlu menjalankan kode Anda sebagai pengguna super (sudo). Ada beberapa cara untuk melakukannya. Salah satu caranya adalah dengan menyimpan kode Python Anda, membuat file tersebut dapat dieksekusi, dan kemudian menjalankan program dengan sudo. Ini adalah cara yang tepat untuk kode yang akan Anda jalankan di masa mendatang. Untuk bengkel, kita ambil jalan pintas. Kami akan meluncurkan IDLE IDE menggunakan sudo dari baris perintah, yang membuat program apa pun berjalan dari instance IDLE itu berjalan dengan sudo.

Memutar Motor Tunggal

1. Buka jendela terminal. Anda dapat menggunakan yang sama yang digunakan untuk instalasi; tetapi selama IDLE terbuka, jendela terminal ini terkunci.
2. Ketik **sudo idle**.
3. Di IDLE IDE, buat file baru dan simpan sebagai `motors.py`.
4. Masukkan kode berikut:

```
from Adafruit_MotorHAT import Adafruit_MotorHAT as amhat
from Adafruit_MotorHAT import Adafruit_DCMotor as adcm
import time
# create a motor object
      mh = amhat(addr=0x60) m
      yMotor = mh.getMotor(1)
# set start speed
      myMotor.setSpeed(150)
      while True:
            # set direction
                  myMotor.run(amhat.FORWARD)
            # wait 1 second
                  time.sleep(1)
            # stop motor
                  myMotor.run(amhat.RELEASE)
            # wait 1 second
                  time.sleep(1)
```

5. Simpan file.
6. Tekan F5 untuk menjalankan program. Mari kita lihat kodenya.

Kita mulai dengan mengimpor objek yang kita butuhkan dari perpustakaan Adafruit_ MotorHAT dan menetapkan alias mereka sehingga kita tidak perlu menulis seluruh nama

setiap kali kita menggunakannya. Kami juga mengimpor perpustakaan waktu untuk penundaan kami nanti dalam kode.

```
from Adafruit_MotorHAT import Adafruit_MotorHAT as amhat
from Adafruit_MotorHAT import Adafruit_DCMotor as adcm
import time
```

Selanjutnya, kita membuat sebuah instance dari objek motor. Untuk melakukan ini, kami memberi tahu Python bahwa kami menggunakan Motor HAT yang terletak di alamat I2C default, 0x60. Kami kemudian membuat objek motor untuk motor yang dilampirkan ke M1, atau motor 1. Ini memungkinkan kami mengakses metode dan properti motor.

```
mh = amhat(addr=0x60)
myMotor = mh.getMotor(1)
```

Sebelum kita menyalakan motor, untuk program ini, kita mengatur kecepatan awal sedikit di atas setengah kecepatan.

```
myMotor.setSpeed(150)
```

Sekarang kita bungkus sisa kode penggerak motor dalam loop sementara. Selama nilai True adalah true, kode ini akan terus dieksekusi.

```
while True:
```

Kode untuk menggerakkan motor sangat sederhana. Kami menggerakkan motor ke depan selama satu detik, dan kemudian menghentikan motor selama satu detik. Program ini terus melakukan ini.

```
# set direction
myMotor.run(amhat.FORWARD)
# wait 1 second
time.sleep(1)
# stop motor
myMotor.run(amhat.RELEASE)
# wait 1 second
time.sleep(1)
```

Tekan Ctrl-C pada keyboard. Perhatikan bahwa program berakhir, tetapi motor terus berputar. Itu karena HAT Motor bebas berjalan. Ini berarti bahwa pengontrol melanjutkan dengan perintah terakhir yang diterima dari Pi. Jika kita tidak menyuruhnya berhenti, itu tidak akan terjadi. Sekarang kita akan melakukan sesuatu yang menarik; sesuatu yang belum pernah kita lakukan sebelumnya. Kami akan membungkus kode penggerak motor menjadi blok `try/except`. Ini adalah bagian dari kode yang memungkinkan kita untuk menangkap kesalahan yang terjadi, dan kemudian menanganinya dengan baik.

Dalam kasus khusus ini, kita akan menggunakan blok `try/except` untuk menangkap `KeyboardInterrupt` event. Event ini dipicu ketika kita menggunakan Ctrl-C untuk keluar dari program.

1. Ubah kode perulangan while menjadi seperti berikut:

```
try:
      while True:
            # set direction
                  myMotor.run(amhat.FORWARD)
            # wait 1 second
                  time.sleep(1)
            # stop motor
                  myMotor.run(amhat.RELEASE)
            # wait 1 second
                  time.sleep(1)
except KeyboardInterrupt:
      myMotor.run(amhat.RELEASE)
```

2. Jalankan programnya.
3. Biarkan berjalan sebentar, lalu tekan Ctrl-C. Motor sekarang akan berhenti ketika program keluar.

Python menangkap acara KeyboardInterrupt dan mengeksekusi baris kode terakhir sebelum keluar. Kode melepaskan motor, dan mematikannya.

Memutar Dua Motor

Memutar satu motor itu bagus dan keren, tetapi robot kami akan memiliki dua motor, dan kami ingin mereka beroperasi secara independen. Kami juga ingin mereka dapat mengubah kecepatan dan arah.

Untuk mengoperasikan beberapa motor, Anda hanya perlu membuat instance objek motor yang berbeda untuk setiap motor. Dengan asumsi bahwa Anda menghubungkan kedua motor Anda sebelumnya, kami membuat dua motor dan memberikan perintah untuk masing-masing motor. Kami juga mengubah kecepatan dan arah motor.

1. Buat file Python baru dari IDLE.
2. Simpan file sebagai `two_motors.py`.
3. Masukkan kode berikut:

```
from Adafruit_MotorHAT import Adafruit_MotorHAT as
amhat,Adafruit_DCMotor as adcm
import time
# create 2 motor objects
      mh = amhat(addr=0x60)
      motor1 = mh.getMotor(1)
```

```
        motor2 = mh.getMotor(2)
# set start speed
        motor1.setSpeed(0)
        motor2.setSpeed(0)
# direction variable
        direction = 0
# wrap actions in try loop
try:
        while True:
                # if direction = 1 then motor1 forward andmotor2
                backward
                # else motor1 backward and motor2 forward
        if direction == 0:
            motor1.run(amhat.FORWARD)
            motor2.run(amhat.BACKWARD)
        else:
                motor1.run(amhat.BACKWARD)
            motor2.run(amhat.FORWARD)
        # ramp up the speed from 1 to 255
        for i in range(255):
                j = 255-i
                motor1.setSpeed(i)
                motor2.setSpeed(j)
                time.sleep(0.01)
        # ramp down the speed from 255 to 1
        for i in reversed(range(255)):
                  j = 255-i
                  motor1.setSpeed(i)
                  motor2.setSpeed(j)
                  time.sleep(0.01)
        # wait half a second
        time.sleep(0.5)
        # change directions
        if direction == 0:
                  direction = 1
        else:
                direction = 0
```

```
# kill motors and exit program on ctrl-c
except KeyboardInterrupt:
        motor1.run(amhat.RELEASE)
        motor2.run(amhat.RELEASE)
```

4. Simpan file.
5. Tekan F5 untuk menjalankan program.

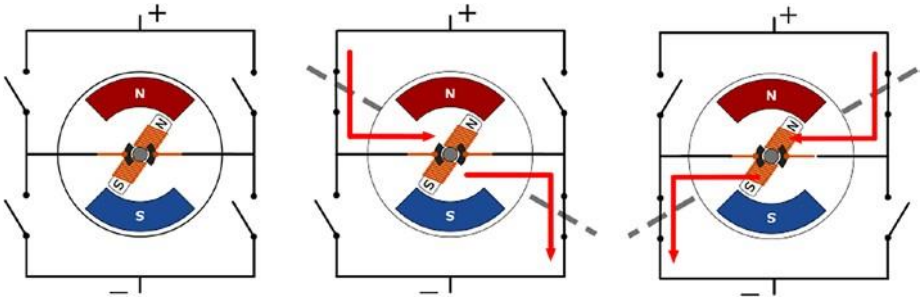
Untuk sebagian besar, kodenya sama. Kami menambahkan beberapa loop untuk menghitung hingga 255 dan mundur lagi. Kami membuat dua variabel untuk menyimpan nilai ini; yang kedua membalikkan nilai dengan mengurangkannya dari 255. Kami juga memiliki variabel untuk melacak arah putaran motor. Setelah kedua motor melaju naik dan turun lagi, kami mengubah arah dan melakukannya lagi. Kami menggunakan kode keluar yang sama seperti yang kami lakukan sebelumnya.

6.7 L298N DRIVER MOTOR GENERIK

L298N adalah chip pengontrol motor H-bridge yang umum. Beberapa produsen telah memasang chip di papan dan menambahkan semua elektronik pendukung yang diperlukan. Hasil akhirnya adalah pengontrol motor generik yang populer.

Pengontrol Motor H-Jembatan

Kontroler motor H-bridge adalah pengontrol motor paling umum yang akan Anda temui. Itu mendapat namanya dari bentuk H khas yang terlihat dalam skema. Sebuah H-jembatan pada dasarnya terdiri dari empat gerbang yang mengontrol aliran arus melalui motor. Tergantung pada bagaimana gerbang dibuka dan ditutup, Anda dapat mengontrol arah putaran motor. Pada L298N, ada dua pin aktif (satu untuk setiap motor) dan empat pin input. Motor kontrol in1 dan in2 pin 1, sedangkan motor kontrol in3 dan in4 2. Gambar 6-21 menunjukkan bagaimana gerbang diatur; in1 mengontrol S1 dan S4, dan in2 mengontrol S3 dan S2. Ketika in1 atau in2 tinggi, gerbang masing-masing ditutup. Ketika mereka rendah, gerbang terbuka. Ketika in1 tinggi dan in2 rendah, arus mengalir sehingga motor berputar searah jarum jam. Jika in1 rendah dan in2 tinggi, motor berputar berlawanan arah jarum jam. Jika kedua pin tinggi, motor tidak berputar, pada dasarnya mengerem. Jika kedua pin rendah, tidak ada arus yang mengalir melalui motor dan motor berputar bebas.



Gambar 6-21. Pengoperasian pengontrol motor jembatan-H

Itu meninggalkan pin aktif, enA dan enB, yang digunakan untuk mengatur kecepatan motor. Inilah sebabnya mengapa kami menggunakan PWM pada pin ini. PWM memungkinkan kita untuk memvariasikan kecepatan setiap motor. Jika kami menggunakan pin digital standar, kami dapat menghidupkan dan mematikan motor, tetapi itu akan menjadi daya penuh atau tanpa daya. PWM memungkinkan kita untuk memiliki kontrol lebih besar atas motor kita.

Menggunakan L298N

Ada beberapa cara untuk menggunakan L298N; masing-masing punya kelebihan dan kekurangan. Salah satu metode adalah menghubungkan pin ke Raspberry Pi, yang memiliki keunggulan dikendalikan langsung oleh Pi. Kekurangannya adalah Anda mungkin harus menggunakan konverter level logika karena pin Pi adalah 3,3 volt dan pengontrolnya adalah 5 volt. Juga, Anda kehilangan kemampuan untuk mengontrol kecepatan. Kontrol kecepatan membutuhkan PWM, dan seperti yang saya bahas di bab sebelumnya, itu adalah salah satu area yang diinginkan oleh Pi. Metode pilihan saya untuk menghubungkan ke L298N adalah melalui Arduino. Dengan cara ini, Anda memiliki kontrol kecepatan melalui PWM. Juga, karena Arduino dan pengontrol keduanya 5 volt, tidak perlu menggunakan konverter level logika. Tentu saja, kekurangannya di sini adalah Anda harus meneruskan instruksi motor melalui serial ke Arduino.

Kode Arduino

Untuk latihan ini, Arduino hanya akan bertindak sebagai pass-through untuk pengontrol motor. Kami akan membaca instruksi dari aliran serial dan meneruskan nilai-nilai itu ke pengontrol motor. Arduino tidak akan melakukan logika. Jika Anda menerapkan ini dalam skenario dunia nyata, Anda mungkin ingin sensor bertindak sebagai interupsi. Dengan membiarkan sensor mengganggu operasi normal, Anda dapat membangun beberapa pengaman ke dalam proyek.

1. Buka sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai L298N_passthrough.
3. Masukkan kode berikut:

```
int enA = 9;
int in1 = 8;
int in2 = 7;
int in3 = 5;
int in4 = 4;
int enB = 3;
int enAVal, in1Val, in2Val, in3Val, in4Val, enBVal;
void setup() {
      // put your setup code here, to run once:
      Serial.begin(9600);
      pinMode(enA, OUTPUT);
      pinMode(in1, OUTPUT);
```

```
    pinMode(in2, OUTPUT);
    pinMode(in3, OUTPUT);
    pinMode(in4, OUTPUT);
    pinMode(enB, OUTPUT);
  }
void loop() {
    // Only work if there is data in the serial buffer
    while(Serial.available() > 0){
    // Read the ints from the serial port
    enAVal = Serial.parseInt();
    in1Val = Serial.parseInt();
    in2Val = Serial.parseInt();
    // Only read the next three if there is data
    if(Serial.available() > 0){
    in3Val = Serial.parseInt();
    in4Val = Serial.parseInt();
    enBVal = Serial.parseInt();
    }
    // Write the values to the L298N
    analogWrite(enA, enAVal);
    digitalWrite(in1, in1Val);
    digitalWrite(in2, in2Val);
    digitalWrite(in3, in3Val);
    digitalWrite(in4, in4Val);
    analogWrite(enB, enBVal);
    // Purge any remaining data because we don't need it
    while(Serial.available() > 0){
      char x = Serial.read();
    }
  }
}
```

4. Simpan sketsa dan unggah ke Arduino.

Anda tidak akan melihat apa pun yang terjadi di Arduino. Apa yang telah kami lakukan adalah memuat Arduino dengan kode yang hanya membaca port serial dan meneruskan nilai yang dibaca ke L298N. Kami melakukan beberapa hal dalam kode ini yang ingin Anda perhatikan.

```
if(Serial.available() > 2){
```

Hal pertama yang harus diperhatikan adalah pernyataan if setelah kita membaca nilai untuk in2Val. Kode ini digunakan di kedua latihan yang akan datang. Latihan pertama hanya akan melewati tiga nilai. Yang kedua akan melewati enam nilai. Kami hanya membaca tiga nilai kedua jika ada; jika tidak, kita akan mendapatkan kesalahan. Untuk memastikan bahwa kami menghindari kesalahan, kami hanya ingin membaca tiga nilai berikutnya jika ada tiga atau lebih nilai untuk dibaca.

```
while(Serial.available() > 0){
      char x = Serial.read();
}
```

Di akhir sketsa, kami menambahkan loop while kecil. Jika kita punya apa pun yang tersisa di buffer serial setelah membaca keenam nilai, kita perlu menghapusnya sehingga tidak ada data yang tertinggal di buffer untuk siklus berikutnya. Blok ini hanya membaca semua byte yang tersisa dan menghapusnya dari buffer.

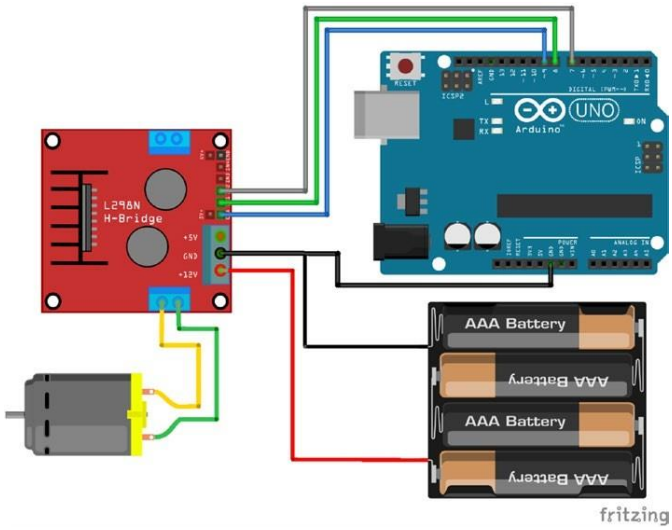
Menghubungkan L298N

Menghubungkan pengontrol motor sedikit lebih rumit daripada hanya mencolokkannya ke header. Kami akan terhubung melalui Arduino untuk memanfaatkan pin PWM. Seperti halnya Motor HAT, kami akan menyediakan pengontrol motor dengan daya eksternal dari empat baterai AAA. Ini memberikan 6 volt yang diinginkan motor, tanpa menggoreng Arduino.

Memutar Satu Motor

Pada latihan pertama dengan L298N, Anda mempelajari cara memutar satu motor. Kami mengatur kecepatan dan arah motor, mengubah arah, dan memvariasikan kecepatan. Gambar 6-22 menunjukkan sirkuit untuk latihan ini.

1. Hubungkan enA pada pengontrol motor ke pin 9 pada Arduino. Anda mungkin perlu melepas jumper.
2. Hubungkan in1 ke pin 8.
3. Hubungkan in2 ke pin 7.
4. Hubungkan pin ground pada Arduino ke ground post pada terminal sekrup. Ini kemungkinan pos tengah.
5. Hubungkan motor ke pengontrol motor dengan menghubungkan satu kabel ke out1 dan yang lainnya ke out2. Saat ini, tidak masalah prospek mana yang menuju ke pos keluaran mana.
6. Hubungkan kabel hitam dari baterai ke terminal ground pada L298N.
7. Hubungkan kabel merah dari baterai ke terminal positif. Biasanya diberi label + atau VCC.



Gambar 6-22. L298N kabel motor tunggal

8. Buka file baru di IDLE.
9. Simpan file sebagai L298N_1_motor_example.py.
10. Masukkan kode berikut:

```
import serial
import time
directon = 1
ser = serial.Serial("/dev/ttyACM0",9600,timeout=1)def
driveMotor(int speed, int drct):
      enA = speed
# determine direction
if drct == 1:
      in1 = 1
      in2 = 0
else if drct == -1:
      in1 = 0
      in2 = 1
else:
      in1 = 0
      in2 = 0
valList = str(enA) + ',' + str(in1) + ',' + str(in2)
serString = ','.join(valList)
ser.write(serString)
time.sleep(0.1)
while 1:
      # ramp up speed
```

```
        while motSpeed < 256:
                driveMotor(motSpeed, direction)
                motSpeed = motSpeed + 1
        # ramp down speed
        while motSpeed > 0:
                driveMotor(motSpeed, direction)
                motSpeed = motSpeed - 1
        # reverse direction
direction = -direction
```

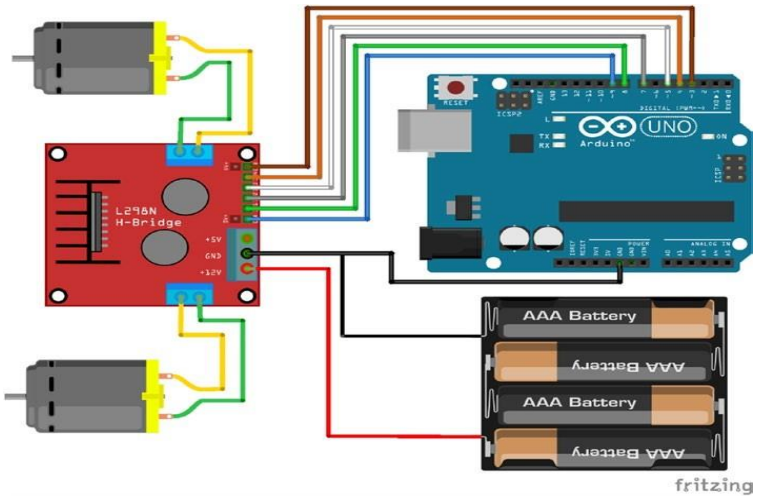
11. Simpan dan jalankan file

Motor harus mulai berputar, semakin cepat hingga mencapai kecepatan puncaknya. Pada saat itu, ia melambat hingga berhenti, berbalik arah, dan berulang. Ini berlanjut sampai Anda menekan Ctrl-C untuk menghentikan program.

Memutar Dua Motor

Selanjutnya, kami memutar dua motor. Pengaturan dan kodenya sangat mirip dengan yang baru saja kami lakukan, dengan motor tambahan. Anda harus sudah memiliki motor pertama yang terhubung. Jika tidak, selesaikan langkah 1 sampai 7 dari latihan sebelumnya. Mari kita angkat dengan penambahan motor kedua (lihat Gambar 6-23).

1. Hubungkan kabel dari pin 5 pada Arduino ke in3 pada pengontrol motor.
2. Hubungkan kabel dari pin 4 ke in4.
3. Hubungkan kabel dari pin 3 ke enB.
4. Hubungkan kabel dari motor kedua ke terminal out2. Sekali lagi, tidak terlalu penting dalam latihan ini lead mana yang menuju terminal mana. Kemudian, ketika Anda memasang motor ke robot, Anda ingin memastikan bahwa motor terhubung sehingga mereka berputar berlawanan satu sama lain. Untuk saat ini, bagaimanapun, kami hanya peduli bahwa mereka benar-benar berbalik.



Gambar 6-23. L298N dua kabel motor

5. Buka file baru di IDLE.
6. Simpan file sebagai L298N_2_motor_example.py.
7. Masukkan kode berikut:

```
import serial
import time
directon = 1
ser = serial.Serial("/dev/ttyACM0",9600,timeout=1)
def driveMotor(int motor, int speed, int drct):
      enA = speed
      # determine direction
      if drct == 1:
            in1 = 1
            in2 = 0
            in3 = 1
            in4 = 0
         else if drct == -1:
            in1 = 0
            in2 = 1
            in3 = 0
            in4 = 1
         else:
            in1 = 0
            in2 = 0
            in3 = 0
            in4 = 0
         valList = str(enA) + ',' + str(in1) + ',' + str(in2) +',' +
            str(in3) + ',' + str(in4) + ',' + str(enB)
         serString = ','.join(valList)
         ser.write(serString)
         time.sleep(0.1)
      while 1:
         # ramp up speed
         while motSpeed < 256:
            driveMotor(motSpeed, direction)
            motSpeed = motSpeed + 1
         # ramp down speed
         while motSpeed > 0:
            driveMotor(motSpeed, direction)
```

```
            motSpeed = motSpeed - 1
    # reverse direction
    direction = -direction
```

Kode ini tidak jauh berbeda dengan latihan sebelumnya. Yang kami lakukan hanyalah menambahkan variabel aktifkan dan input untuk motor kedua. Kedua motor harus berputar pada kecepatan yang sama. Mereka mempercepat, memperlambat, dan kemudian berbalik arah. Lihatlah kodenya dan tentukan cara membuat motor berputar secara independen satu sama lain.

6.8 RINGKASAN

Dalam bab ini, kita melihat jenis motor yang umum: DC, tanpa biji, stepper, dan servo. Kami merakit HAT Adafruit DC & Stepper Motor. (Sekarang Anda seharusnya cukup nyaman dengan besi solder.) Kemudian, Anda belajar bagaimana menghubungkan motor Anda ke sana dan membuatnya berputar. Kami juga melihat pengontrol motor generik yang umum. L298N bekerja sedikit berbeda karena arahnya diatur dengan mengubah status dua pin. Kami menghubungkan L298N melalui Arduino untuk memanfaatkan pin PWM untuk mengontrol kecepatan motor, serta arahnya. Kita bisa dengan mudah menghubungkan pin aktifkan ke pin digital out pada header Raspberry Pi GPIO. Namun, memiliki kontrol diskrit kecepatan motor adalah penting. Dalam bab yang akan datang, Anda akan melihat mengapa ini penting.

Pada titik ini, Anda memiliki semua informasi yang Anda butuhkan untuk membuat robot kecil sederhana. Anda telah belajar tentang pemrograman di Python dan Arduino. Anda telah bekerja dengan sensor untuk memungkinkan robot Anda mendeteksi lingkungannya. Dan akhirnya, Anda membuat motor Anda berputar, sehingga Anda memiliki gerakan. Logika, penginderaan, dan gerakan adalah inti dari setiap robot. Segala sesuatu yang lain adalah versi yang lebih maju dari elemen-elemen ini. Sekarang setelah Anda mengetahui semua yang Anda perlukan tentang robot, kami akan merakit kit sasis dan membuat robot. Setelah itu, kami mulai membuat robot kami lebih mampu dan lebih pintar. Kami mulai dengan sensor IR, beralih ke algoritma kontrol, dan kemudian memberikan mata robot. Yah, sebuah mata.

BAB 7
MERAKIT ROBOT

Di bab terakhir, kami membuat HAT Motor Adafruit, perangkat elektronik yang memungkinkan Anda mengontrol hingga empat motor DC dengan Raspberry Pi Anda. Kami juga melihat pengontrol motor generik yang kami jalankan melalui papan Arduino. Sekarang setelah Anda tahu cara membuat robot Anda bergerak, mari kita mulai membuatnya. Dalam bab ini, kita akan membangun robot kita. Sepanjang jalan, saya akan memberikan beberapa tip dan petunjuk yang saya ambil di build saya. Ada banyak hal kecil yang perlu dipertimbangkan saat merakit robot. Anda akan menemukan beberapa skenario aneh yang tidak Anda pertimbangkan. Yang paling diabaikan adalah kabel dan manajemen kabel. Hal-hal seperti urutan operasi dan komponen penempatan sangat penting. Keputusan yang dibuat di awal pembangunan dapat menyebabkan komplikasi di kemudian hari. Dengan memperhatikan hal-hal ini, Anda tidak perlu membongkar robot untuk memperbaiki kesalahan yang Anda buat sejak awal.

Membangun ini dipecah menjadi empat latihan terpisah. Kita akan mulai dengan membuat kit sasis Whippersnapper. Kemudian kita akan memasang elektronik, yang diikuti dengan pemasangan kabel. Terakhir, kita akan melihat pemasangan sensor ultrasonik. Dalam setiap latihan, saya akan menunjukkan beberapa hal yang perlu dipertimbangkan saat mengerjakan bangunan Anda sendiri.

7.1 MERAKIT SASIS

Untuk build ini, saya memilih untuk menggunakan kit yang tersedia secara komersial. Hal yang menyenangkan tentang kit adalah bahwa kit yang baik memiliki semua yang Anda butuhkan untuk memulai. Ada banyak pilihan di banyak titik harga yang berbeda dan dari banyak produsen yang berbeda. Banyak kit murah, umumnya ditemukan secara online dari penjual asing, kurang lengkap daripada yang lain. Seringkali, ini adalah kit untuk perangkat populer tetapi dirakit dengan sedikit pemikiran tentang bagaimana bagian-bagiannya berjalan bersama. Jadi, jika Anda akan membeli kit, pastikan kit memiliki semua perangkat keras dan bagian-bagiannya dirancang untuk bekerja sama.

7.2 MEMILIH BAHAN

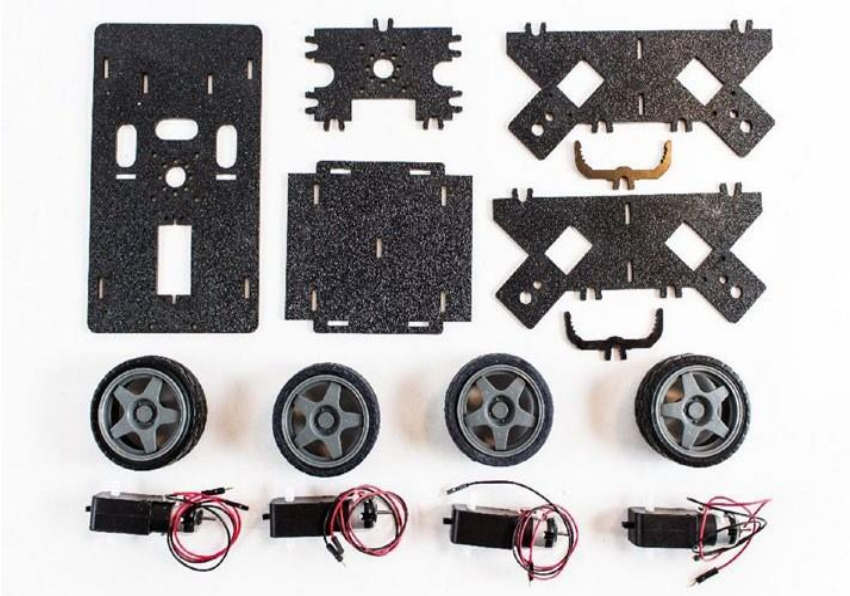
Bahan adalah hal lain yang perlu dipertimbangkan ketika memilih sasis. Sasis logam bagus. Ini cenderung lebih mahal daripada sasis plastik, tetapi juga cenderung jauh lebih tahan lama. Dalam hal kit plastik, ingatlah bahwa tidak semua plastik itu sama. Akrilik adalah bahan yang murah dan nyaman untuk digunakan; namun, itu bukan bahan yang tepat untuk sebagian besar aplikasi. Akrilik rapuh, tidak fleksibel, dan mudah tergores. Ketika pecah, biasanya pecah menjadi potongan-potongan yang tajam.

Juga bijaksana untuk diingat untuk tidak menggunakan akrilik dalam segala jenis

aplikasi gesekan tinggi karena cenderung pecah menjadi butiran saja yang memperkuat gesekan. Jika Anda akan menggunakan plastik, ABS adalah bahan yang lebih baik untuk digunakan. Seperti akrilik, ABS hadir dalam lembaran dan cukup murah. Tidak seperti akrilik, ini jauh lebih tahan lama. Tidak mudah retak atau pecah, dan lebih tahan gores. ABS dapat dibor dan lebih mudah digunakan daripada akrilik. Pilihan lain adalah polistiren. Styrene adalah bahan yang digunakan untuk model kit plastik. Jadi, jika Anda terbiasa bekerja dengan kit ini, maka styrene adalah pilihan yang mudah. Ini lebih fleksibel daripada akrilik atau ABS. Ini cenderung sedikit lebih mahal daripada yang lain, tetapi mudah untuk dikerjakan.

7.3 *THE WHIPPERSNAPPER*

Whippersnapper adalah kit komersial yang dibuat dengan lembaran ABS potong laser. Ini adalah bagian dari garis Runt Rover dari Actobotics, diproduksi oleh ServoCity. Saya telah bekerja dengan beberapa kit dari lini Actobotics, dan saya tahu mereka dirancang dengan baik, produk berkualitas. Selain kit robot, mereka menghasilkan garis besar suku cadang yang dirancang untuk bekerja sama. Semua hal ini berkontribusi pada pemilihan Whippersnapper (lihat Gambar 7-1) untuk dasar proyek ini. Tidak ada salahnya jika ini adalah sasis yang terlihat bagus dengan ruang untuk menampung semua barang elektronik dan menyisakan ruang untuk berkembang.



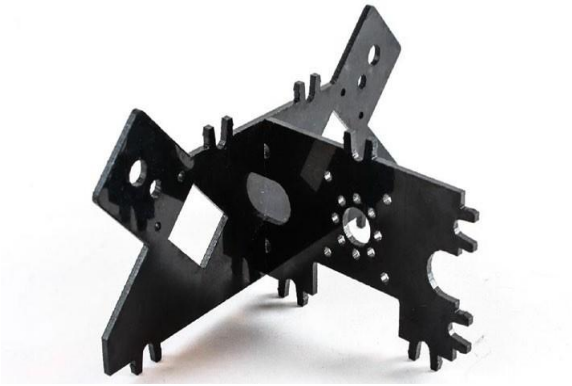
Gambar 7-1. Semua bagian Whippersnapper

Demi klarifikasi, Raspberry Pi akan dipasang di bagian belakang robot. Arduino akan berada di depan. Ini akan membuat pengkabelan sedikit lebih mudah.

Untuk memulai, saya suka meletakkan bagian-bagiannya. Ini membantu Anda memastikan bahwa semuanya ada di sana dan membuat Anda terbiasa dengan semua bagian. Kit ini terkunci bersama. Bahkan, satu-satunya alat yang Anda butuhkan adalah obeng Philips dan tang runcing. Saat menyatukan bagian-bagiannya, ketahuilah bahwa ukurannya pas dan perlu tenaga untuk menyatukan semuanya. Selama Anda menjaga bagian tetap lurus, mereka

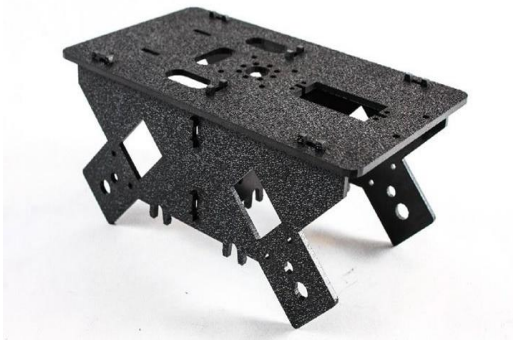
tidak akan pecah. Pertahankan pegangan yang kuat pada bagian tersebut dan berikan tekanan yang merata.

1. Pasang penyangga tengah ke salah satu sisinya. Pastikan sisi lapangan menghadap keluar. Perhatikan tab pada dukungan tengah. Sepasang tab tunggal menempel pada pelat bawah (lihat Gambar 7-2).



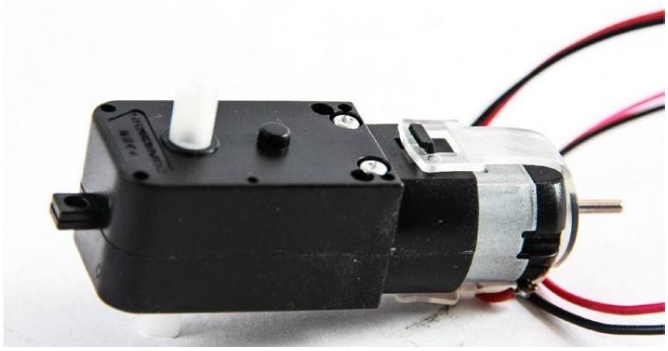
Gambar 7-2. Dukungan tengah melekat pada pelat luar

2. Pasang pelat samping kedua ke penyangga tengah. Sekali lagi, pastikan bahwa sisi lapangan berada di luar robot.
3. Pasang pelat atas ke rakitan. Ada enam set tab yang menempel pada pelat atas (lihat Gambar 7-3).



Gambar 7-3. Plat atas ditambahkan

Pada langkah selanjutnya, kami memasang motor. Di satu sisi motor ada pasak kecil (lihat Gambar 7-4), yang membantu menyelaraskan motor dan menahannya di tempatnya.

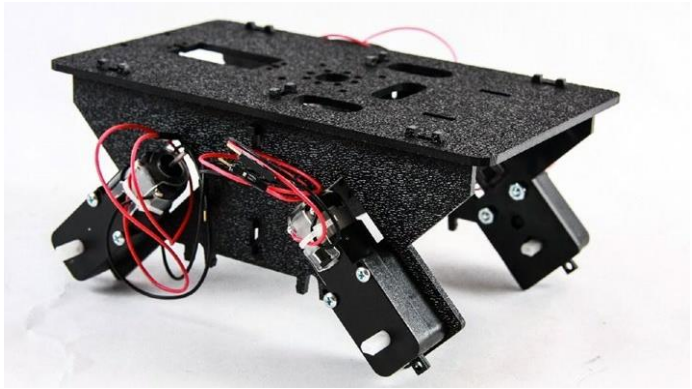


Gambar 7-4. Motor dengan tab

1. Pasang motor sehingga poros melewati lubang bawah dan pasak masuk ke lubang kedua.
2. Gunakan dua sekrup dan mur untuk menahan motor pada tempatnya (lihat Gambar 7-5). Meskipun tidak termasuk dalam kit, beberapa mesin cuci kunci split #4 akan baik digunakan di sini. Jika Anda tidak memilikinya, gunakan Loctite Threadlocker Blue pada mur. Tanpa sesuatu untuk menguncinya, mur akan berbunyi.

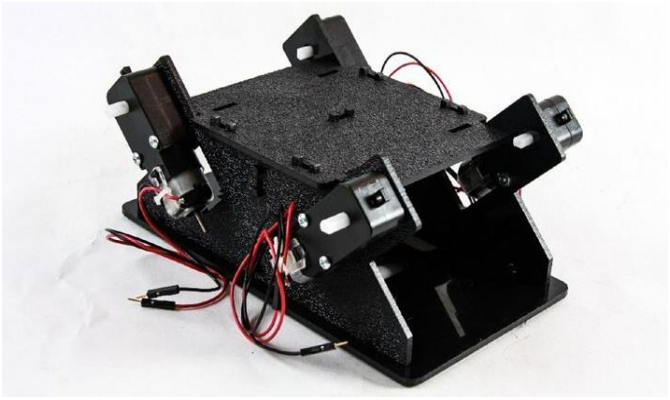
Gambar 7-5. Motor terpasang

3. Ulangi proses untuk masing-masing dari tiga motor yang tersisa (lihat Gambar 7-6).



Gambar 7-6. Semua motor terpasang

4. Balikkan sasis dan pasang pelat bawah. Ada lima set tab yang menahan pelat bawah (lihat Gambar 7-7).



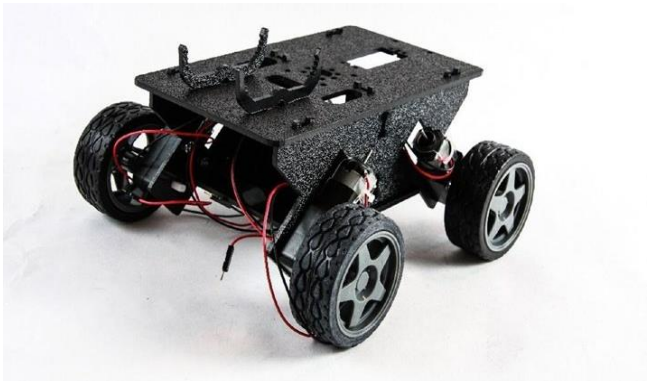
Gambar 7-7. Pelat bawah ditambahkan

5. Masukkan kabel untuk setiap motor ke dalam sasis melalui lubang di belakang motor (lihat Gambar 7-8). Sedikit pembersihan ini membuat kabel tidak tersangkut di roda atau tersangkut sesuatu.

Gambar 7-8. Kabel motor dimasukkan melalui lubang di belakang motor

6. Pasang klip elektronik ke pelat atas. Klip ini akan digunakan untuk memegang Raspberry Pi.
7. Masukkan kabel untuk motor depan melalui lubang di pelat penopang tengah.

Sasis sekarang siap untuk dipasangi elektronik. Sasis robot Anda akan terlihat seperti yang ditunjukkan pada Gambar 7-9.



Gambar 7-9. Whippersnapper yang sudah selesai

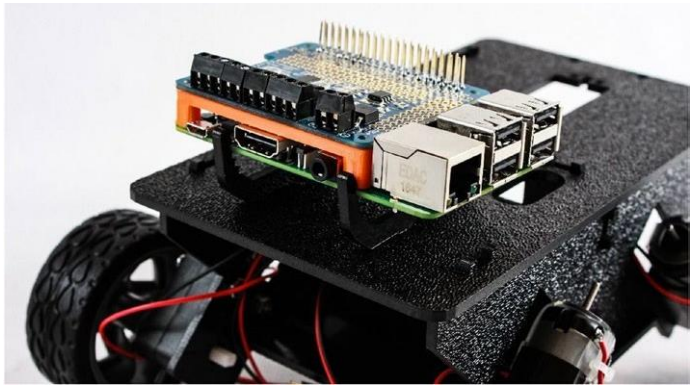
7.4 MEMASANG ELEKTRONIK

Selanjutnya, kami akan memasang elektronik ke sasis. Dimulai dengan Raspberry Pi, kami akan memasang setiap komponen, dengan Arduino dan papan tempat memotong roti yang dipasang ke depan.Selama bagian pembuatan ini, pita pemasangan dan pengikat ritsleting sering digunakan. Penempatan papan terserah Anda. Beberapa orang memasang beberapa elektronik di dalam sasis. Namun, saya menemukan pengaturan berikut ini yang paling cocok untuk saya.

Ini memungkinkan akses yang lebih mudah ke elektronik dan menghemat ruang di dalam untuk komponen tambahan.

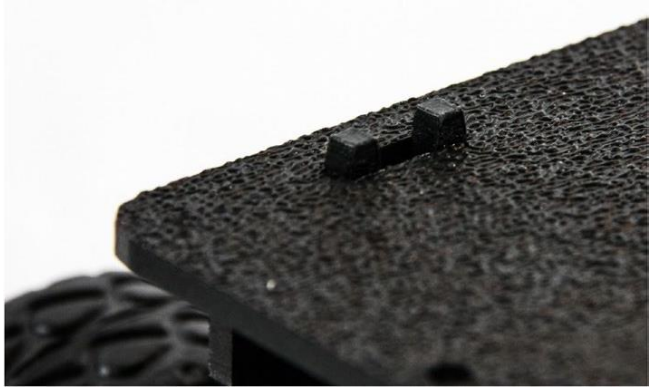
1. Pasang Raspberry Pi ke dalam klip di pelat atas (lihat Gambar 7-10). Pi harus dipegang

dengan kuat di tempatnya oleh duri atas.



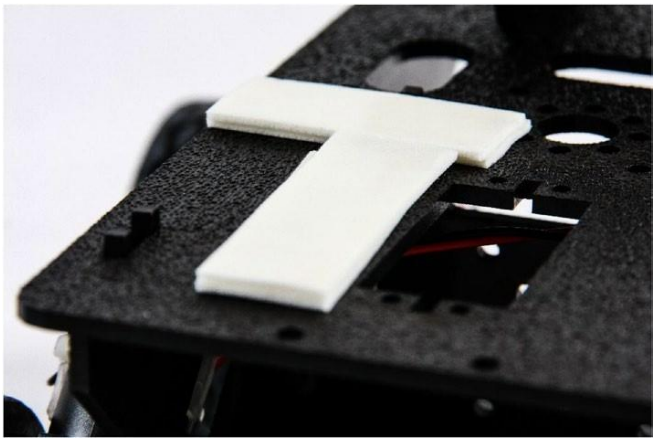
Gambar 7-10. Raspberry Pi dipasang di klip

Tab yang menyatukan sasis (lihat Gambar 7-11) membuat pemasangan Arduino dan papan tempat memotong roti menjadi suatu tantangan. Inilah salah satu alasan saya suka menggunakan pita pemasangan busa itu memberikan beberapa bantalan. Untuk menghapus tab, kita perlu menggandakan rekaman itu.



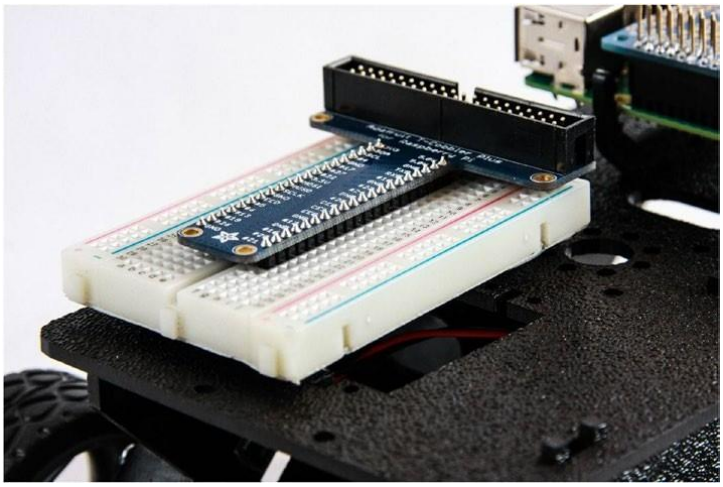
Gambar 7-11. Klip menonjol dari pelat atas

2. Tumpuk dua potong pita busa di atas satu sama lain dan letakkan di piring atas. Gunakan pita busa bertumpuk kedua untuk membentuk huruf T (lihat Gambar 7-12). Ini menambah stabilitas.



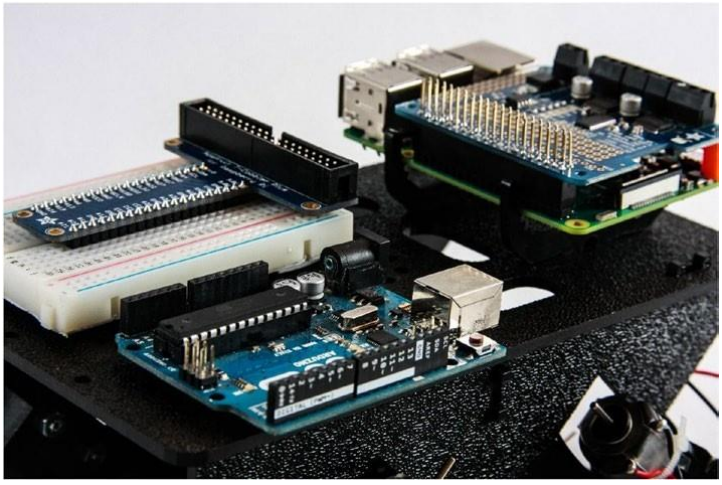
Gambar 7-12. Lapisan ganda pita pemasangan untuk papan tempat memotong roti

3. Lepaskan kertas pelindung dari bagian bawah papan tempat memotong roti dan tekan papan tempat memotong roti dengan kuat ke dalam pita berbentuk T di pelat atas (lihat Gambar 7-13).



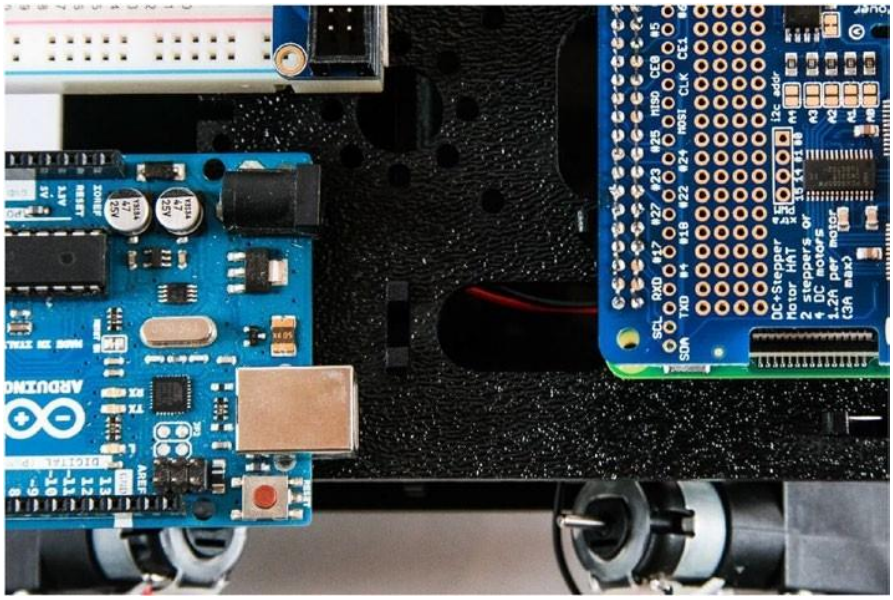
Gambar 7-13. Papan tempat memotong roti yang terpasang. Perhatikan bahwa T-cobbler telah dipindahkan ke depan untuk memberi ruang bagi power pack.

4. Ulangi prosedur untuk Arduino (lihat Gambar 7-14).



Gambar 7-14. Arduino dipasang pada lapisan ganda pita pemasangan

Saat memasang Arduino, ingatlah untuk meninggalkan ruang untuk kabel USB. Saya mengimbangi Arduino dari tengah sehingga colokan USB bersih dari Raspberry Pi (lihat Gambar 7-15).



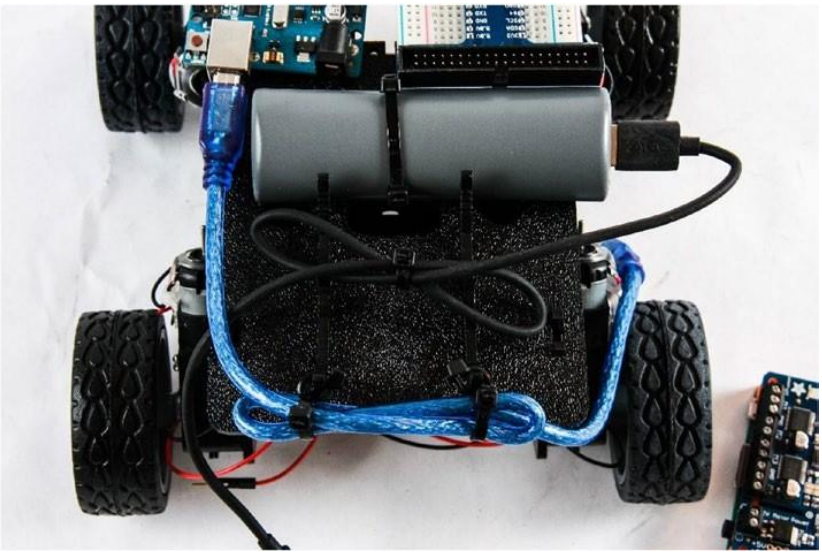
Gambar 7-15. Meninggalkan izin untuk kabel USB

5. Pasang 4 dudukan baterai AA di dalam sasis di bagian belakang. Pastikan untuk memasangnya sedemikian rupa sehingga memungkinkan akses ke baterai dan sakelar daya, jika ada. Saya menggunakan pita pemasangan busa untuk menahan milik saya di tempatnya.
6. Temukan tempat untuk memasang bank daya 5V dengan aman. Saya menemukan bahwa ruang antara papan tempat memotong roti dan Raspberry Pi bekerja dengan baik untuk bank daya kecil yang saya gunakan. Penempatan Anda akan ditentukan oleh faktor bentuk bank daya Anda.

Dengan elektronik di tempatnya, saatnya untuk mulai menyambungkan bagian-bagiannya.

7.5 PENGKABELAN

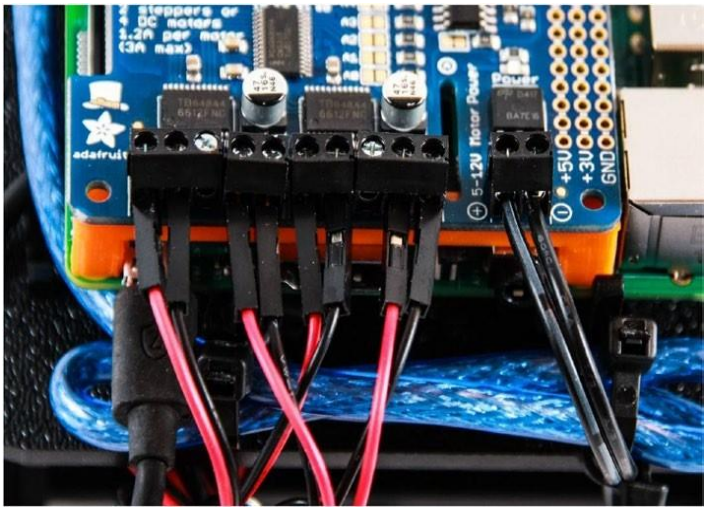
Tidaklah pantas untuk mencoba menulis bagian ini sebagai petunjuk langkah demi langkah. Bagaimana Anda menghubungkan robot Anda sepenuhnya terserah Anda. Setiap robot berbeda. Pengkabelan ditentukan oleh penempatan komponen, kabel yang Anda gunakan, dan preferensi pribadi. Sebagai gantinya, saya akan memandu Anda melalui bagaimana saya menghubungkan robot saya dan proses pemikiran di balik keputusan saya, dan menyertakan pertimbangan untuk proyek Anda. Saya lebih memilih untuk menjaga kabel saya rapi mungkin. Beberapa orang tidak terlalu memikirkan bagaimana mereka menjalankan kabel. Saya telah melihat beberapa kekacauan kusut di bawah selimut beberapa robot. Penting bagi saya untuk dapat mengakses bagian dengan mudah, dan ini termasuk kabel dan kabel.



Gambar 7-16. Kabel USB dibundel untuk kerapian

Kabel USB untuk menyalakan Pi dan menghubungkan ke Arduino sedikit lebih panjang daripada yang saya sukai untuk sebagian besar proyek. Ada banyak jenis kabel yang tersedia, termasuk yang memiliki colokan sudut kanan, yang membuat pemasangan kabel cukup mudah. Karena kabelnya agak panjang, saya menggunakan ikatan zip untuk mengikatnya menjadi sesuatu yang lebih kecil. Kabel yang lebih berat untuk Arduino kemudian diikat ke klip pemasangan untuk Pi. Kabel dari bank daya ke Pi terselip di bawah Pi (lihat Gambar 7-16).

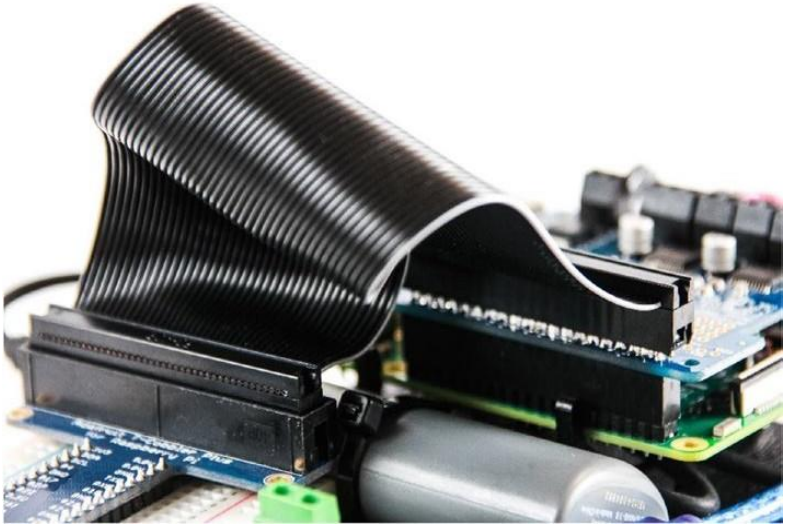
Selanjutnya, saya menghubungkan kabel dari motor ke Motor HAT. Motor HAT memiliki empat output untuk motor DC. Ada empat motor. Saya dapat memasang motor secara berpasangan ke dua keluaran yang berbeda: satu untuk sisi kiri dan satu untuk kanan; namun, motor kecil dan murah cenderung tidak terlalu konsisten dalam kecepatan. Meskipun dua motor menerima sinyal yang sama, tidak ada jaminan bahwa keduanya akan berputar pada kecepatan yang sama. Mampu menyesuaikan kecepatan setiap motor secara mandiri adalah fitur bagus yang saya manfaatkan. Jadi, setiap motor memiliki outputnya sendiri (lihat Gambar 7-17).



Gambar 7-17. Kabel motor dan baterai eksternal terhubung ke Motor HAT

Saya menyertakan pengganda untuk kecepatan masing-masing motor. Dengan sedikit penyetelan pengganda, saya bisa membuat motor berputar lebih konsisten. Setelah motor terhubung, saya menghubungkan daya. Saat Anda menghubungkan milik Anda, perhatikan polaritasnya. Sebagai standar, merah adalah positif dan hitam adalah negatif. Karena baterai saya dimodifikasi, kabelnya tidak berwarna merah dan hitam. Saya menggunakan voltmeter untuk menentukan polaritas kabel dan menghubungkannya dengan benar.

Kabel terakhir yang disambungkan adalah kabel pita untuk T-cobbler (lihat Gambar 7-18). Hanya ada satu cara untuk memasang kabel pita ke T-cobbler. Tab pada steker sejajar dengan celah pada steker. Pada Pi, pastikan kabel bergaris putih menuju ke pin 1. Untuk Pi, ini adalah pin yang paling dekat dengan sudut.



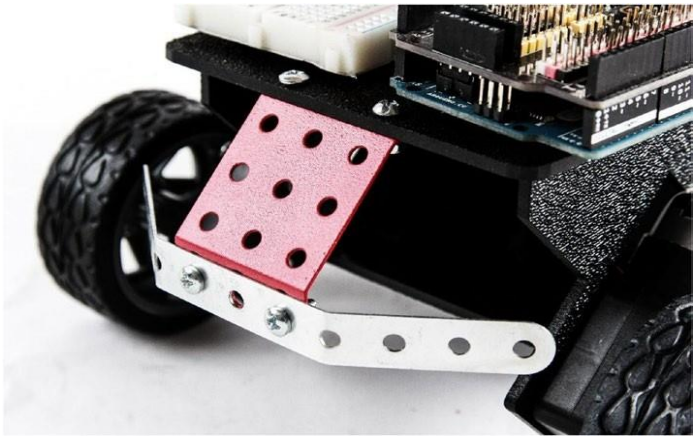
Gambar 7-18. Kabel pita menempelkan T-cobbler ke Pi. Perhatikan garis putih.

7.6 PEMASANGAN SENSOR

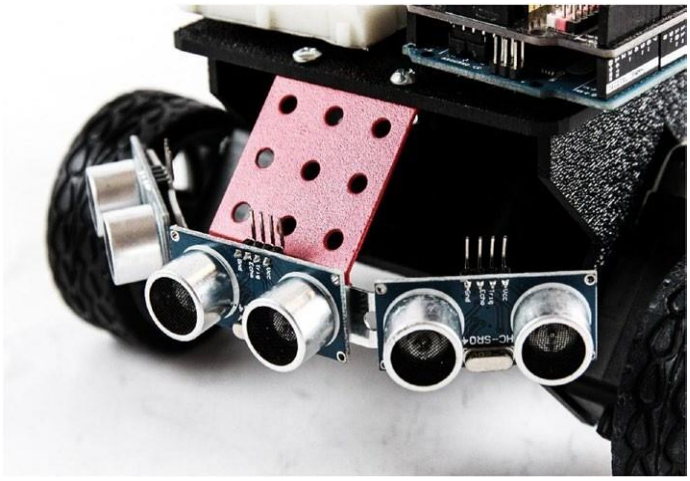
Di sinilah merakit robot membutuhkan kreativitas paling banyak. Sebagian besar sasis tidak dilengkapi dengan perangkat keras pemasangan untuk sensor. Jika ya, itu untuk sensor tertentu yang mungkin tidak Anda gunakan. Ada banyak pendekatan berbeda untuk memasang sensor. Saya menemukan hanya sedang dipersiapkan dengan sejumlah bahan yang berbeda cenderung bekerja dengan baik untuk saya. Ketika saya tumbuh dewasa, saya memiliki Erector Set. Jika Anda tidak akrab dengan Erector, mereka memproduksi mainan konstruksi yang mencakup sejumlah bagian logam: balok, braket, sekrup, mur, katrol, ikat pinggang, dan sebagainya.

Saya menghabiskan waktu berjam-jam membangun truk, traktor, pesawat, dan ya, bahkan di tahun 1980-an, robot. Bayangkan kegembiraan saya ketika mencari beberapa bagian umum untuk digunakan dalam sebuah proyek, saya menemukan Erector Set di toko hobi lokal saya. Saya bahkan lebih senang ketika saya menemukan bahwa salah satu toko perangkat keras kotak besar lokal menjual bagian-bagian individu di tempat sampah berbagai bagian mereka. Set Erector adalah sumber yang bagus untuk suku cadang kecil yang dibutuhkan di banyak proyek. Dalam hal ini, saya menggunakan salah satu balok dan braket

untuk memasang pengukur jarak ultrasonik (lihat Gambar 7-19).



Gambar 7-19. Sebuah braket dan balok dari Erector Set. Balok ditekuk untuk memberikan sudut bagi sensor.

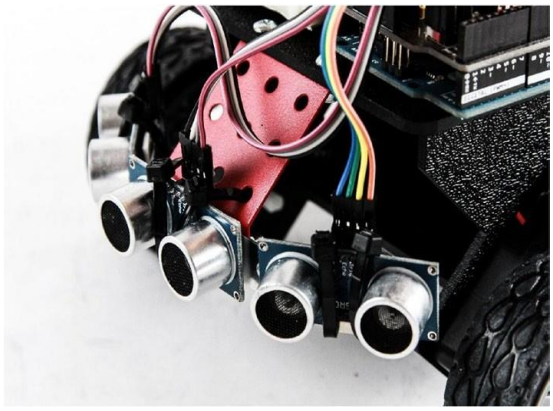


Gambar 7-20. Sensor ultrasonik dipasang

Dengan braket terpasang, saya menggunakan pita pemasangan untuk memasang sensor (lihat Gambar 7-20). Dalam kasus khusus ini, rekaman itu melayani dua tujuan. Pertama, ia memegang sensor ke logam. Tujuan kedua adalah isolasi. Elektronik di bagian belakang sensor terbuka; menempelkannya ke bagian logam berisiko menyebabkan korsleting. Pita pemasangan busa membuat isolator yang baik.

Satu hal yang saya pelajari adalah tidak mempercayai pita pemasangan saja untuk menahan sensor, terutama pada logam. Di masa lalu, pita itu lepas, menyebabkan sensor rusak. Solusinya adalah favorit saya yang lain: ikatan zip. Rekaman itu menahan sensor di tempatnya dan memberikan insulasi; namun, ikatan zip menambah keamanan dan kekuatan. Pada titik ini, saya cukup yakin bahwa segala sesuatunya tidak akan kemana-mana. Dengan sensor terpasang dengan kuat, hal terakhir yang harus dilakukan adalah menghubungkannya ke Arduino. Saya menggunakan jumper female-to-female dari sensor ke Arduino (lihat Gambar 7-21). Di Arduino, saya memasang pelindung sensor. Pelindung sensor menambahkan pin 5V dan ground ke masing-masing pin analog digital. Beberapa dari mereka bahkan

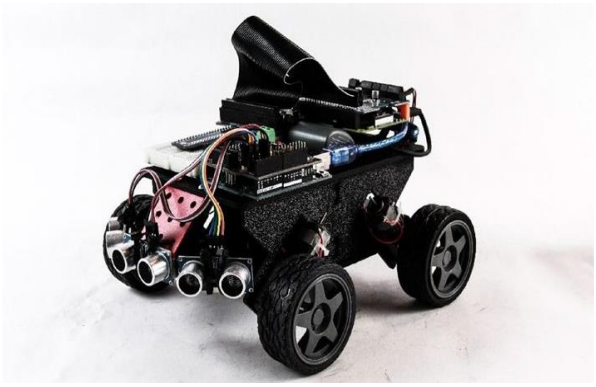
memiliki header khusus untuk perangkat serial atau nirkabel. Saya menggunakan yang sangat sederhana tanpa banyak header khusus. Pelindung sensor memudahkan pemasangan sensor dan perangkat lain.



Gambar 7-21. Pengukur jarak ultrasonik diamankan dengan ikatan zip dan disambungkan ke Arduino

7.7 ROBOT SELESAI

Dengan sensor terpasang, saya memiliki robot yang lengkap. Satu-satunya hal yang tersisa untuk dilakukan adalah menulis kode untuk membuatnya bergerak. Gambar 7-22 menunjukkan robot saya yang sudah selesai.



Gambar 7-22. Whippersnapper yang sudah jadi dengan elektronik

7.8 MEMBUAT ROBOT MOBILE

Saat ini, kami memiliki koleksi suku cadang yang sangat bagus. Tanpa perangkat lunak yang tepat, kita tidak benar-benar memiliki robot. Selanjutnya, saya menguraikan apa yang kita ingin robot lakukan. Kami akan mengubahnya menjadi perilaku, dan itu, pada gilirannya, menjadi kode yang diperlukan untuk menghidupkan robot kecil itu.

Rencana

Dalam bab-bab sebelumnya, kami bekerja dengan contoh-contoh yang mengilustrasikan berbagai topik. Karena ini adalah aplikasi pertama kita untuk robot yang bekerja, mari luangkan waktu sejenak untuk menguraikan apa yang kita ingin robot lakukan. Rencana ini didasarkan pada robot yang saya buat di awal bab ini. Diasumsikan bahwa ada tiga sensor ultrasonik dan empat motor yang beroperasi secara independen. Motor

dikendalikan melalui Motor HAT yang dipasang pada Pi. Sensor dioperasikan melalui Arduino.

Sensor

Seperti disebutkan, kami akan mengoperasikan tiga sensor ultrasonik. Sensor terhubung ke Arduino melalui pelindung sensor. Karena kita menggunakan serial untuk berkomunikasi dengan Pi, kita tidak dapat menggunakan pin 0 dan 1. Ini adalah pin yang digunakan oleh port serial. Jadi, sensor pertama kami, tengah, ada di pin 2 dan 3; sensor kiri ada di pin 4 dan 5; dan sensor kanan ada di pin 6 dan 7.

Sensor dipicu secara berurutan, dimulai dari tengah, diikuti oleh kiri, dan kemudian kanan. Setiap sensor menunggu sampai yang sebelumnya selesai sebelum memicu. Hasilnya dikirim kembali ke Pi dalam interval setengah detik sebagai string pelampung yang mewakili jarak dari setiap sensor dalam sentimeter.

Motor

Motor terhubung ke Motor HAT di Raspberry Pi. Setiap motor terhubung ke salah satu dari empat saluran motor pada pengontrol. Motor 1, motor kiri depan, terhubung ke M1. Motor 2, motor kiri belakang, terhubung ke M2. Motor 3, motor kanan depan, ada di M3. Dan, motor 4, motor kanan belakang, ada di M4. Penggerak robot menggunakan kemudi diferensial, juga disebut penggerak tangki atau kemudi selip. Untuk melakukan ini, motor kiri bergerak bersama dan motor kanan bergerak bersama. Saya menyebutnya sebagai saluran kiri dan kanan. Jadi, perintah yang sama dikirim ke M1 dan M2. Demikian juga, M3 dan M4 menerima perintah bersama. Kode memiliki pengganda untuk setiap motor. Pengganda diterapkan ke masing-masing motor untuk mengkompensasi perbedaan kecepatan. Implikasinya adalah kita perlu mengizinkan buffer untuk mengakomodasi perbedaan ini. Jadi, kecepatan tertinggi diatur ke nilai 200 dari 255. Awalnya, pengganda diatur ke 1. Anda perlu menyesuaikan pengganda agar sesuai dengan robot Anda.

Perilaku

Robot adalah penjelajah acak sederhana. Ini drive dalam garis lurus sampai mendeteksi hambatan. Kemudian menyesuaikan arahnya untuk menghindari menabrak rintangan. Ini tidak dimaksudkan untuk menjadi solusi yang sangat canggih, tetapi menggambarkan beberapa dasar dalam operasi robot.

Berikut adalah aturan untuk perilaku robot:

- Ini mendorong ke depan.
- Jika mendeteksi objek di sebelah kirinya, ia berbelok ke kanan.
- Jika mendeteksi objek di sebelah kanannya, ia berbelok ke kiri.
- Jika mendeteksi objek tepat di depannya, ia berhenti dan berbelok ke arah dengan jarak terjauh yang tersedia.
- Jika kedua arah memiliki jarak yang sama, atau kedua sensor sisi berada di luar nilai batas, robot berputar ke arah acak untuk waktu yang telah ditentukan sebelum melanjutkan.

Perilaku ini agak mendasar, tetapi harus menyediakan robot yang berkeliaran di rumah secara mandiri.

Kode

Kode dipecah menjadi dua bagian: untuk Arduino dan untuk Pi. Di Arduino, yang kami pedulikan hanyalah mengoperasikan sensor dan menyampaikan pembacaan kembali ke Pi pada interval yang telah ditentukan. Dalam hal ini, setiap 500 milidetik, atau setengah detik. Raspberry Pi menggunakan data yang masuk untuk mengeksekusi perilaku. Ia membaca dari port serial dan mem-parsing data ke dalam variabel. Variabel-variabel ini digunakan oleh Pi untuk menentukan tindakan selanjutnya. Tindakan ini diterjemahkan ke dalam instruksi untuk motor, yang kemudian dikirim ke pengontrol motor untuk dieksekusi.

Kode Arduino

Program ini hanya mengoperasikan tiga sensor ultrasonik di bagian depan robot. Kemudian mengembalikan nilai-nilai ini sebagai string float ke Raspberry Pi melalui koneksi serial. Kode pada dasarnya sama dengan contoh Pinguino di Bab 5. Perbedaannya adalah kita menggunakan tiga sensor, bukan satu.

1. Buka sketsa baru di Arduino IDE.
2. Simpan sketsa sebagai `robot_sensors`.
3. Masukkan kode berikut:

```
int trigMid = 2;
int echoMid = 3;
int trigLeft = 4;
int echoLeft = 5;
int trigRight = 6;
int echoRight = 7;
float distMid = 0.0;
float distLeft = 0.0;
float distRight = 0.0;
String serialString;
void setup() {
      // set the pinModes for the sensors
      pinMode(trigMid, OUTPUT);
      pinMode(echoMid, INPUT);
      pinMode(trigLeft, OUTPUT);
      pinMode(echoLeft, INPUT);
      pinMode(trigRight, OUTPUT);
      pinMode(echoRight, INPUT);
            // set trig pins to low;
            digitalWrite(trigMid,LOW);
            digitalWrite(trigLeft,LOW);
            digitalWrite(trigRight,LOW);
            // starting serial
```

```
        Serial.begin(115200);
    }
    // function to operate the sensors
    // returns distance in centimeters
    float ping(int trigPin, int echoPin){
        // Private variables, not available
        // outside the function
        int duration = 0;
        float distance = 0.0;
        // operate sensor
        digitalWrite(trigPin, HIGH);
        delayMicroseconds(10);
        digitalWrite(trigPin, LOW);
        // get results and calculate distance
        duration = pulseIn(echoPin, HIGH);
        distance = duration/58.2;
        // return the results
        return distance;
    }
    void loop() {
        // get the distance for each sensor
        distMid = ping(trigMid, echoMid);
        distLeft = ping(trigLeft, echoLeft);
        distRight = ping(trigRight, echoRight);
    // write the results to the serial port
    Serial.print(distMid); Serial.print(",");
    Serial.print(distLeft); Serial.print(",");
    Serial.println(distRight);
   // wait 500 milliseconds before looping
   delay(500);
}
```

4. Simpan sketsa dan unggah ke Arduino.

Arduino sekarang harus melakukan ping, tetapi karena tidak ada yang mendengarkan, kami belum benar-benar tahu. Selanjutnya, kita akan menulis kode untuk Raspberry Pi.

Kode Pi raspberry

Sekarang saatnya menulis kode yang berjalan di Raspberry Pi. Ini adalah program yang

cukup panjang, jadi saya akan memecahnya saat kita mulai. Sebagian besar ini akan terlihat sangat akrab. Ada beberapa perubahan di sana-sini untuk mengakomodasi logika, tetapi sebagian besar, kami telah melakukan ini sebelumnya. Setiap kali kita melakukan sesuatu yang baru, saya akan meluangkan waktu untuk memandu Anda melewatinya.

1. Buka IDLE untuk Python 2.7. Ingat, pustaka Adafruit belum berfungsi di Python 3.
2. Buat file baru.
3. Simpan sebagai pi_roamer_01.py.
4. Masukkan kode berikut. Saya menelusuri setiap bagian untuk memastikan bahwa Anda memiliki gagasan yang kuat tentang apa yang terjadi di sepanjang jalan.
5. Impor perpustakaan yang Anda butuhkan.

```
import serial
import time
import random
from Adafruit_MotorHAT import Adafruit_MotorHAT as amhat
from Adafruit_MotorHAT import Adafruit_DCMotor as adamo
```

6. Buat variabel motor dan buka port serial. Arduino diatur untuk berjalan pada baud rate yang lebih tinggi, sehingga Pi juga perlu berjalan pada baud yang lebih tinggi.

```
# create motor objects
motHAT = amhat(addr=0x60)
mot1 = motHAT.getMotor(1)
mot2 = motHAT.getMotor(2)
mot3 = motHAT.getMotor(3)
mot4 = motHAT.getMotor(4)
# open serial port
ser = serial.Serial('/dev/ttyACM0', 115200)
```

7. Buat variabel yang dibutuhkan. Banyak dari mereka yang mengapung karena kami bekerja dengan desimal.

```
# create variables
# sensors
distMid = 0.0
distLeft = 0.0
distRight = 0.0
# motor multipliers
 m1Mult  =  1.0
 m2Mult  =  1.0
 m3Mult  =  1.0
 m4Mult  =  1.0
```

```
# distance threshold
distThresh = 12.0
distCutOff = 30.0
```

8. Atur variabel yang dibutuhkan untuk mengatur motor. Anda akan melihat bahwa saya telah membuat sejumlah nilai default, dan kemudian menetapkan nilai tersebut ke variabel lain. Variabel `leftSpeed`, `rightSpeed`, dan `driveTime` seharusnya menjadi satu-satunya yang benar-benar kita ubah dalam kode. Selebihnya adalah untuk memberikan konsistensi sepanjang program. Jika Anda ingin mengubah kecepatan default, Anda cukup mengubah `speedDef`, dan perubahan tersebut diterapkan di mana-mana.

```
# speeds
speedDef = 200
leftSpeed = speedDef
rightSpeed = speedDef
turnTime = 1.0
defTime = 0.1
driveTime = defTime
```

9. Buat fungsi drive. Itu dipanggil dari dua tempat di dalam tubuh utama program. Karena ada banyak pekerjaan yang terlibat, lebih baik memecah kode menjadi blok fungsi yang terpisah.

```
def driveMotors(leftChnl = speedDef, rightChnl =
speedDef, duration = defTime):
      # determine the speed of each motor by multiplying
      # the channel by the motors multiplier
      m1Speed = leftChnl * m1Mult
      m2Speed = leftChnl * m2Mult
      m3Speed = rightChnl * m3Mult
      m4Speed = rightChnl * m4Mult
      # set each motor speed. Since the speed can be a
      # negative number, we take the absolute value
      mot1.setSpeed(abs(int(m1Speed)))
      mot2.setSpeed(abs(int(m2Speed)))
      mot3.setSpeed(abs(int(m3Speed)))
      mot4.setSpeed(abs(int(m4Speed)))
      # run the motors. if the channel is negative, run
      # reverse. else run forward
      if(leftChnl < 0):
```

```
            mot1.run(amhat.BACKWARD)
            mot2.run(amhat.BACKWARD)
        else:
            mot1.run(amhat.FORWARD)
            mot2.run(amhat.FORWARD)
        if (rightChnl > 0):
            mot3.run(amhat.BACKWARD)
            mot4.run(amhat.BACKWARD)
        else:
            mot3.run(amhat.FORWARD)
            mot4.run(amhat.FORWARD)

            # wait for duration
            time.sleep(duration)
```

10. Mulailah blok utama program dengan membungkus kode dalam blok try. Ini memungkinkan kita untuk keluar dari program dengan bersih. Tanpa itu dan blok kecuali yang sesuai, motor akan terus menjalankan perintah terakhir yang mereka terima.

```
try:
      while 1:
```

11. Lanjutkan blok utama dengan membaca port serial dan mem-parsing string yang diterima

```
# read the serial port
val = ser.readline().decode('utf=8')
print val
# parse the serial string
parsed = val.split(',')
parsed = [x.rstrip() for x in parsed]
# only assign new values if there are
# three or more available
if(len(parsed)>2):
      distMid = float(parsed[0] + str(0))
      distLeft = float(parsed[1] + str(0))
      distRight = float(parsed[2] + str(0))
```

12. Masukkan kode logika. Ini adalah kode yang menjalankan perilaku yang diuraikan sebelumnya. Perhatikan bahwa blok sensor tengah (yang menjalankan stop dan turn) ditulis di luar kode penghindaran rintangan kiri dan kanan. Ini dilakukan karena kami

ingin logika ini dievaluasi terlepas dari hasil kode kiri dan kanan. Dengan memasukkannya setelah kode lainnya, kode tengah menimpa salah satu nilai yang dibuat oleh kode kiri/kanan.

```
# apply cutoff distance
if(distMid > distCutOff):
      distMid = distCutOff
if(distLeft > distCutOff):
      distLeft = distCutOff
if(distRight > distCutOff):
      distRight = distCutOff
# reset driveTime
driveTime = defTime
# if obstacle to left, steer right by increasing
# leftSpeed and running rightSpeed negative defSpeed
# if obstacle to right, steer to left by increasing
# rightSpeed and running leftSpeed negative
if(distLeft <= distThresh):
      leftSpeed = speedDef
      rightSpeed = -speedDef
elif (distRight <= distThresh):
      leftSpeed = -speedDef
      rightSpeed = speedDef
else:
      leftSpeed = speedDef
      rightSpeed = speedDef
# if obstacle dead ahead, stop then turn toward most
# open direction. if both directions open, turn random
if(distMid <= distThresh):
      # stop
      leftSpeed = 0
      rightSpeed = 0
      driveMotors(leftSpeed, rightSpeed, 1)
      time.sleep(1)
      leftSpeed = -150
       rightSpeed = -150
       driveMotors(leftSpeed, rightSpeed, 1)
       # determine preferred direction. if distLeft >
       # distRight, turn left. if distRight > distLeft,
```

```
# turn right. if equal, turn random
dirPref = distRight - distLeft
if(dirPref == 0):
    dirPref = random.random()
if(dirPref < 0):
    leftSpeed = -speedDe
    frightSpeed = speedDef
elif(dirPref > 0):
    leftSpeed = speedDef
    rightSpeed = -speedDef
driveTime = turnTime
```

13. Panggil fungsi driveMotors yang kita buat tadi.

```
# drive the motors
driveMotors(leftSpeed, rightSpeed, driveTime)
```

14. Siram semua byte yang masih ada di buffer serial.

```
ser.flushInput()
```

15. Masukkan blok kecuali. Ini memungkinkan kita untuk mematikan motor ketika kita mengklik Ctrl-C sebelum kita keluar dari program.

```
except KeyboardInterrupt:
mot1.run(amhat.RELEASE)
mot2.run(amhat.RELEASE)
mot3.run(amhat.RELEASE)
mot4.run(amhat.RELEASE)
```

16. Simpan file.
17. Tekan F5 untuk menjalankan program.

Setelah selesai menonton robot kecil Anda berkeliaran di sekitar ruangan, tekan Ctrl-C untuk mengakhiri program. Selamat. Anda baru saja membuat dan memprogram robot bertenaga Raspberry Pi pertama Anda. Kami melakukan banyak hal dalam program ini—walaupun sebenarnya tidak ada yang belum pernah Anda lihat sebelumnya. Di bagian pertama program, kami mengimpor perpustakaan yang kami butuhkan dan membuat objek motor. Di bagian selanjutnya, kami mendefinisikan semua variabel kami. Bagian penting dari program ini adalah fungsi yang kita buat setelah variabel. Dalam fungsi ini kita menggerakkan motor. Kecepatan motor dan waktu penggerak dilewatkan sebagai parameter fungsi yang digunakan untuk mengatur kecepatan masing-masing motor. Kami menggunakan tanda kecepatan untuk menentukan arah motor. Setelah itu, kami memulai blok utama kami dengan membungkusnya dalam blok coba. Kami kemudian memasuki loop sementara, yang

memungkinkan program untuk mengulang tanpa batas.

Dalam loop while, kita mulai dengan membaca string serial, dan kemudian kita menguraikannya untuk mengekstrak tiga nilai float. Algoritme untuk mengonversi string menjadi float sedikit berbeda dari yang kami gunakan untuk mengonversi ke integer. Lebih khusus lagi, kita tidak harus membagi hasilnya dengan 10. Menambahkan 0 ke akhir desimal tidak mengubah nilainya, jadi kita bisa menggunakannya saat dikonversi. Pengukuran jarak menentukan tindakan robot selanjutnya. Blok if/elsif/else mengevaluasi nilai sensor. Jika sensor kiri atau kanan mendeteksi rintangan dalam ambang batas yang telah ditentukan, robot berputar ke arah yang berlawanan. Jika tidak ada halangan yang terdeteksi, robot terus maju. Sebuah blok if terpisah menentukan apakah sebuah rintangan berada tepat di depan robot. Jika ada halangan, robot berhenti dan kemudian berputar. Ini menggunakan nilai sensor kiri dan kanan untuk menentukan jalan mana yang harus ditempuh. Jika arah tidak dapat ditentukan, robot berputar ke arah yang acak.

Semua ini membutuhkan waktu, di mana Arduino dengan senang hati mengirim string serial dan mengisi buffer Pi. String ini harus dibersihkan sebelum melanjutkan. Kami menggunakan metode flushInput() dari objek serial untuk melakukan ini. Dengan cara ini, kami hanya bekerja dengan informasi terbaru. Terakhir, kita menggunakan blok exception untuk menangkap perintah interupsi keyboard. Ketika diterima, motor dilepaskan, menghentikannya. Kemudian program keluar.

7.9 RINGKASAN

Bab ini adalah tentang menyatukan semua yang kami pelajari sejauh ini menjadi robot yang berfungsi. Kami merakit kit sasis robot dan memasang semua elektronik. Setelah semuanya terpasang ke robot, kami menulis program untuk menjalankan robot. Itu adalah program roaming yang cukup sederhana. Saat Anda menjalankannya, robot baru Anda harus berkeliaran di ruangan dengan keberhasilan yang bervariasi, tergantung pada seberapa ramai furnitur ruangan itu. Dalam bab berikutnya, kami berupaya meningkatkan robot—menambahkan lebih banyak sensor, meningkatkan logika, dan menambahkan beberapa fungsionalitas tingkat yang lebih tinggi. Secara khusus, kami akan menambahkan kamera dan mempelajari cara menggunakan OpenCV untuk melacak warna dan mengejar bola.

BAB 8
BEKERJA DENGAN SENSOR INFRAMERAH

Untuk memulai bab ini, Anda harus memiliki robot yang berfungsi. Dalam bab-bab sebelumnya, penulis membahas semua yang perlu Anda ketahui untuk menginstal dan memprogram robot Anda. Anda telah bekerja dengan motor, sensor, dan komunikasi antara Raspberry Pi dan Arduino. Di Bab 3 dan Bab 5, Anda belajar bekerja dengan pengukur jarak ultrasonik menggunakan Python dan Arduino. Sisa buku ini memperkenalkan sensor baru, algoritma pemrosesan, dan visi komputer. Dalam bab ini, penulis bekerja dengan sensor inframerah (IR). Penulis melihat berbagai jenis sensor. Di akhir bab ini, penulis menggunakan serangkaian sensor IR untuk mendeteksi tepi permukaan dan garis.

8.1 SENSOR INFRAMERAH

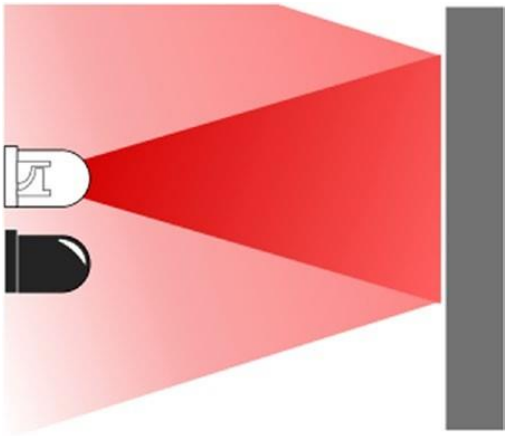
Sensor inframerah (IR) adalah setiap sensor yang menggunakan detektor cahaya, disetel untuk spektrum IR, untuk mendeteksi sinyal IR. Umumnya, sensor IR dipasangkan dengan LED pemancar IR untuk memberikan sinyal IR. Emisi dari LED diukur untuk intensitas atau kehadiran.

8.2 JENIS SENSOR IR

Inframerah cukup mudah digunakan. Karena itu, kami telah menemukan banyak cara berbeda untuk menggunakannya. Ada berbagai macam sensor IR yang tersedia. Banyak digunakan dalam aplikasi yang mungkin tidak Anda harapkan. Pintu otomatis, seperti yang terlihat di toko ritel, menggunakan jenis sensor yang disebut PIR, atau inframerah pasif, untuk mendeteksi gerakan. Jenis sensor ini digunakan untuk lampu otomatis dan sistem keamanan. Printer inkjet menggunakan sensor IR dan LED pemancar IR untuk mengukur pergerakan kepala cetak yang tepat. Remote control sistem hiburan Anda kemungkinan menggunakan LED inframerah untuk mengirimkan pulsa yang disandikan ke penerima IR. Kamera peka IR digunakan untuk jaminan kualitas di bidang manufaktur. Daftarnya terus berlanjut. Mari kita lihat beberapa jenis sensor IR yang berbeda.

Sensor Reflektansi

Sensor reflektansi mencakup semua sensor yang dirancang untuk mendeteksi sinyal yang dipantulkan dari target. Pengukur jarak ultrasonik adalah sensor pemantulan karena mendeteksi panjang gelombang suara yang dipantulkan benda di depannya. Sensor reflektansi IR bekerja dengan cara yang sama yaitu membaca intensitas radiasi IR yang dipantulkan dari suatu objek (lihat Gambar 8-1).



Gambar 8-1. Sensor reflektansi mengukur cahaya IR yang dikembalikan dari dioda IR

Varian dari jenis sensor ini dirancang untuk mendeteksi keberadaan sinyal IR. Sensor menggunakan ambang batas intensitas IR untuk menentukan apakah suatu objek berada di dekatnya atau tidak. Sensor mengembalikan sinyal rendah sampai ambang batas terlampaui, pada titik mana ia mengembalikan sinyal tinggi. Sensor ini umumnya dipasangkan dengan LED yang memancarkan, baik dalam konfigurasi pantul atau langsung.

Deteksi Garis dan Tepi

Detektor inframerah sering digunakan untuk membangun perangkat yang mendeteksi tepi pada garis atau langkan. Sensor ini digunakan untuk deteksi garis ketika kontras antara permukaan dan garis tinggi; misalnya garis hitam di atas meja putih. Ketika sensor berada di atas permukaan putih, sebagian besar sinyal IR dikembalikan ke sensor. Ketika sensor melewati garis gelap, lebih sedikit sinyal IR yang dikembalikan. Sensor ini biasanya mengembalikan sinyal analog yang mewakili jumlah cahaya yang dikembalikan. Dengan cara yang hampir sama, sensor dapat mendeteksi tepi permukaan. Ketika sensor berada di atas permukaan, sensor menerima lebih banyak sinyal IR. Ketika sensor berada di atas tepi, sinyal sangat berkurang, menghasilkan nilai yang rendah (lihat Gambar 8-2).

Gambar 8-2. Garis dan tepi dapat dideteksi oleh perbedaan cahaya yang dipantulkan

Beberapa sensor memiliki ambang yang dapat disesuaikan, memungkinkan mereka untuk memberikan sinyal digital. Ketika reflektansi berada di atas ambang batas, sensor dalam keadaan tinggi. Ketika reflektansi di bawah ambang batas, sensor rendah. Tantangan dengan jenis sensor ini adalah sulitnya melakukan dial di ambang batas yang tepat untuk mendapatkan hasil yang konsisten. Kemudian, bahkan jika Anda membuat mereka dipanggil untuk satu lingkungan, segera setelah kondisinya berubah, atau Anda mencoba

mendemonstrasikannya di sebuah acara, mereka harus dikalibrasi ulang. (Bukannya ini telah terjadi pada saya berulang kali.) Karena itu, penulis lebih suka menggunakan sensor analog, yang memungkinkan untuk menyertakan prosedur kalibrasi otomatis sehingga program dapat menetapkan ambangnya sendiri.

Pengintai

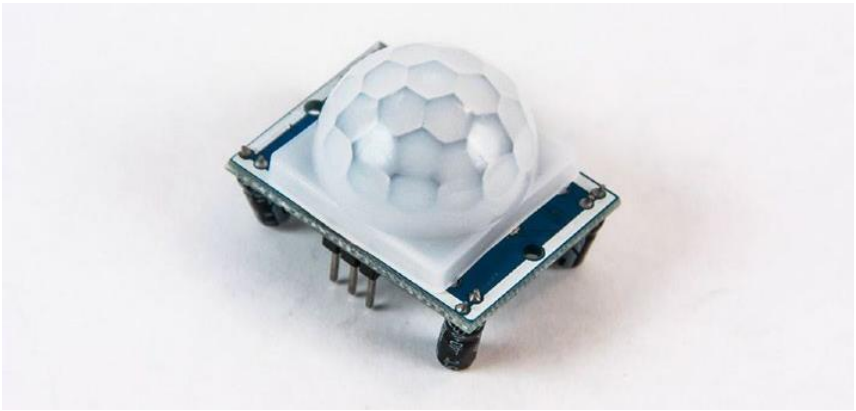
Sama seperti sensor jarak, pengukur jarak mengukur jarak ke suatu objek. Rangefinder menggunakan LED yang lebih kuat dengan sinar yang lebih sempit, yang digunakan untuk menentukan perkiraan jangkauan ke suatu objek. Tidak seperti pengukur jarak ultrasonik, pengukur jarak IR dirancang untuk mendeteksi rentang tertentu. Penting untuk mencocokkan sensor dengan aplikasi.

Sensor Interupsi

Sensor interupsi digunakan untuk mendeteksi keberadaan sinyal IR. Mereka biasanya dipasangkan dengan dioda pemancar dan dikonfigurasi untuk memungkinkan objek lewat antara emitor dan detektor. Ketika objek hadir dan menghalangi emitor, penerima mengembalikan sinyal rendah. Ketika objek tidak ada, dan penerima diizinkan untuk mendeteksi emitor, sinyalnya tinggi. Sensor ini sering digunakan dalam perangkat yang dikenal sebagai encoder. Encoder umumnya terdiri dari disk atau pita dengan bagian tembus dan transparan. Saat disk atau pita bergerak melewati sensor, sinyal terus bergerak dari tinggi ke rendah. Mikrokontroler, atau elektronik lainnya, kemudian dapat menggunakan sinyal bolak-balik ini untuk menghitung pulsa. Karena jumlah bagian transparan diketahui, gerakan dapat dihitung dengan keyakinan tinggi. Dalam bentuknya yang paling sederhana, sensor ini hanya dapat menyediakan pulsa untuk mikrokontroler untuk menghitung. Beberapa encoder menggunakan sejumlah sensor untuk memberikan informasi yang tepat tentang gerakan, termasuk arah.

Detektor Gerak PIR

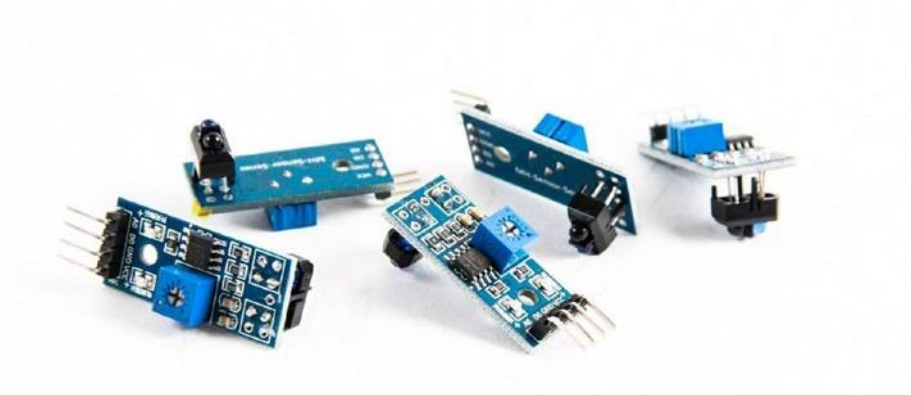
Sensor lain yang sangat umum dikenal sebagai detektor gerakan PIR (lihat Gambar 8-3). Sensor ini memiliki lensa segi yang memantulkan dan membiaskan radiasi IR yang dipancarkan atau dipantulkan oleh objek ke sensor IR di dalamnya. Ketika perubahan terdeteksi oleh sensor ini, sinyal tinggi dihasilkan. Sensor ini mengontrol pintu otomatis di toko bahan makanan lokal Anda, dan mengoperasikan lampu otomatis di rumah atau kantor Anda.



Gambar 8-3. Sensor PIR umum

8.3 BEKERJA DENGAN SENSOR IR

Seperti yang dibahas sebelumnya, ada beberapa cara untuk bekerja dengan sensor IR, tergantung pada jenis yang Anda gunakan. Untuk proyek kami, kami akan menggunakan lima sensor garis IR seperti yang ditunjukkan pada Gambar 8-4. Sensor yang digunakan adalah tipe analog. Sensor khusus yang kami gunakan sebenarnya dapat melakukan pembacaan analog dan digital. Ini memiliki potensiometer kecil yang menetapkan ambang; namun, seperti yang saya bahas sebelumnya dalam bab ini, ini sangat sulit untuk dial-in. Dalam penelitian ini menggunakan pembacaan analog, secara langsung, dan menghitung ambang batas dalam perangkat lunak.



Gambar 8-4. Sensor IR untuk mengikuti garis

8.4 MENGHUBUNGKAN SENSOR IR

Sensor yang saya gunakan untuk robot saya adalah varian 4-pin dari sensor IR 3-pin yang umum. Sensor 3-pin adalah digital dan menerapkan ambang batas ke sinyal analog sensor untuk mengembalikan sinyal tinggi atau rendah. Versi 4-pin menggunakan pengaturan ambang yang sama untuk mengembalikan sinyal digital, tetapi juga memiliki pin tambahan yang menyediakan pembacaan analog. Mari kita berjalan menggunakan kedua sinyal. Sensor yang saya gunakan sedikit berbeda dari kebanyakan. Mereka secara khusus dirancang untuk digunakan dalam aplikasi line-following. Dengan demikian, nilai kembalian dibalik. Ini berarti bahwa alih-alih memberikan angka tinggi saat pantulan tinggi, ia mengembalikan angka rendah. Dalam nada yang sama, sinyal digital juga terbalik. Nilai tinggi menunjukkan adanya garis dan nilai rendah menunjukkan ruang putih. Saat Anda menjalankan latihan berikutnya, jangan heran jika hasilnya berbeda. Kami mencari perilaku yang cukup konsisten.

Kami akan menghubungkan sensor 4-pin ke Arduino dan menggunakan monitor serial untuk melihat output dari sensor. Kita dapat menggunakan pin digital untuk sinyal tinggi/rendah dan pin analog untuk sensor analog, tetapi untuk mempermudah pengkabelan, kita menggunakan dua pin analog. Pin analog yang terhubung ke output digital digunakan dalam mode digital, sehingga berfungsi persis seperti pin lainnya. Karena Arduino sekarang sudah terpasang pada robot, mari gunakan pelindung sensor untuk koneksi. Juga, saya tidak akan melepaskan pengukur jarak ultrasonik. Sketsa untuk sensor IR tidak menggunakan pin tersebut, jadi tidak ada alasan untuk melepaskannya. Untuk latihan ini, Anda juga

membutuhkan permukaan uji. Selembar kertas putih dengan area hitam besar atau garis hitam tebal adalah yang terbaik. Karena sebagian besar kontes mengikuti garis menggunakan pita listrik hitam berukuran 3/4 inci untuk garis, menempelkan pita ini pada selembar kertas, papan poster putih, atau papan inti busa sangat ideal.

1. Menggunakan jumper *female-to-female,* sambungkan pin ground sensor ke pin ground header 3-pin A0.
2. Hubungkan pin VCC sensor ke pin tegangan dari 3-pin header A0. Ini adalah pin tengah.
3. Hubungkan pin analog ke pin sinyal untuk A0. (Pada sensor saya, pin analog diberi label A0.)
4. Hubungkan pin digital sensor ke pin sinyal A1. (Pada sensor saya, ini diberi label D0.)
5. Buat sketsa baru di Arduino IDE.
6. Simpan sketsa sebagai IR_test.
7. Masukkan kode berikut:

```
int analogPin = A0;
int digitalPin = A1;

float analogVal = 0.0;
int digitalVal = 0;

void setup() {
        pinMode(analogPin, INPUT);
        pinMode(digitalPin, INPUT);
        Serial.begin(9600);
}
void loop() {
        analogVal = analogRead(analogPin);
        digitalVal = digitalRead(digitalPin);

        Serial.print("analogVal: "); Serial.print(analogVal);Serial.print(" - digitalVal:
        "); Serial.
        println(digitalVal);
        delay(500);
}
```

8. Pindahkan sensor di atas area putih pada permukaan Anda. Sensor harus sangat dekat dengan permukaan tanpa menyentuhnya.
9. Catat nilai yang dikembalikan. (Saya mendapat nilai analog dalam kisaran 30 hingga 45. Nilai digital saya adalah 0.)
10. Gerakkan sensor di atas garis atau area hitam lain di permukaan.
11. Catat nilainya. (Saya mendapat nilai analog dalam kisaran 700 hingga 900. Nilai

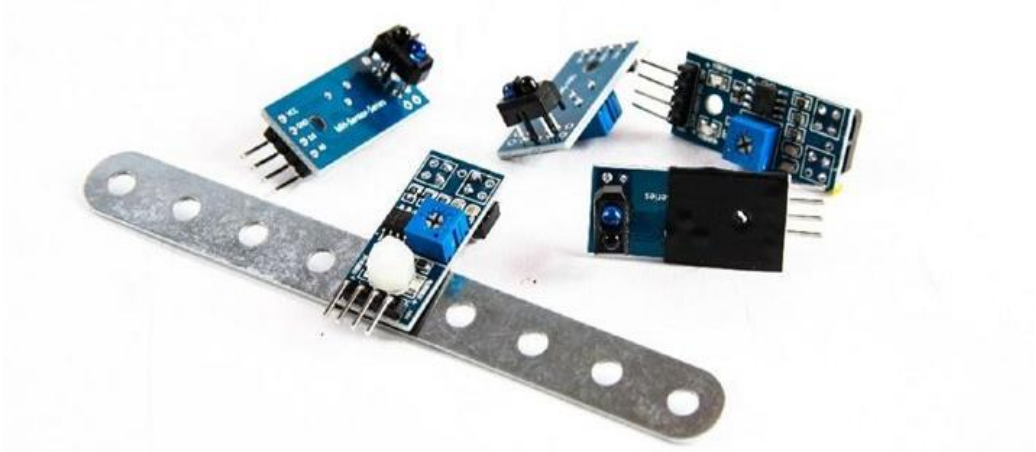
digitalnya adalah 1.)

Anda seharusnya menerima nilai yang sangat berbeda antara area terang dan gelap pada permukaan Anda. Anda dapat melihat bagaimana ini dengan mudah diterjemahkan ke dalam fungsi yang sangat berguna.

8.5 MEMASANG SENSOR IR

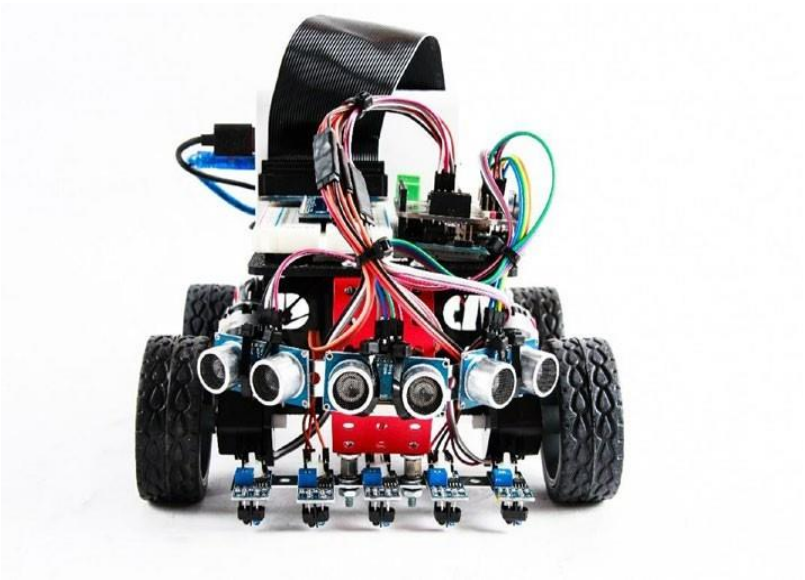
Selanjutnya, kita akan memasang sensor ke robot untuk melakukan sesuatu yang berguna. Sekali lagi, karena build Anda mungkin sangat berbeda dari rancangan, yang akan membahas apa yang dilakukan untuk menghubungkan sensor. Jika Anda telah setia mengikuti, maka Anda harus meniru apa yang telah sudah penulis lakukan. Jika tidak, maka di sinilah robotika mulai berkreasi. Anda perlu menentukan cara memasang sensor ke robot Anda. Lihatlah solusi untuk mendapatkan gambaran tentang apa yang Anda cari. Untuk memasang sensor, memutar (sekali lagi) ke bagian-bagian di Erector Set. Bagian-bagian ini sangat nyaman dan mudah digunakan. Dalam hal ini, menggunakan salah satu palang dan braket sudut yang sama dengan yang digunakan untuk memasang pengukur jarak ultrasonik. Bahkan, dengan menggunakan braket sudut, dengan memperluas rakitan itu untuk mendekatkan sensor IR ke tanah.

Saat mencoba memasang sensor IR, saya mengalami masalah. Lubang untuk memasang sensor berada di antara dua resistor pemasangan permukaan. Ini berarti kebuntuan logam kemungkinan akan menyebabkan hubungan pendek. Kebuntuan nilon di inventaris saya terlalu besar untuk diletakkan rata di ruang itu. Saya dapat menggunakan spacer dan sekrup panjang, tetapi spacer terlalu sempit dan tidak akan menempel lurus pada lubang di bilah pemasangan. Menambahkan mesin cuci membawa sensor terlalu dekat ke tanah. Solusinya adalah memasang sensor IR di atas bilah. Tantangannya adalah bahwa sambungan solder pin pasti akan pendek terhadap batang logam. Namun, hal itu mudah diatasi dengan memasang selotip listrik di bagian belakang sensor dan melubangi sekrup pemasangan (lihat Gambar 8-5).



Gambar 8-5. Memasang sensor IR pada batang. Pita listrik melindungi kabel dari korslet.

Setelah sensor dipasang, saya perlu menjalankan kabel dari sensor ke papan Arduino. Saya hanya menggunakan pin analog dari sensor, jadi saya perlu menggunakan satu pin logika pada Arduino untuk masing-masing. Jika saya menggunakan pin analog dan digital, saya memerlukan pin analog dan digital yang sesuai pada Arduino. Jadi, saya menggunakan pin A0 hingga A4. Untuk memastikan bahwa lead tercapai dengan benar, tanpa mengurangi ketegangan pada koneksi, saya menggunakan jumper *male-to-female* yang lebih pendek untuk memperpanjangnya. Sebuah pita kecil di sekitar sambungan dan sensor siap digunakan (lihat Gambar 8-6).



Gambar 8-6. Robot lengkap dengan sensor IR terpasang dan berkabel

8.6 KODE

Proyek ini, seperti yang terakhir, menggunakan Arduino sebagai perangkat GPIO. Mayoritas logika dilakukan oleh Raspberry Pi. Kami akan membaca sensor IR dalam interval 10 milidetik, 100 kali per detik. Nilai-nilai ini diteruskan ke Raspberry Pi untuk digunakan. Seperti yang Anda lihat di latihan sebelumnya, membaca sensor sangat mudah, jadi kode Arduino cukup ringan. Sisi Pi secara signifikan lebih kompleks. Pertama, kita harus mengkalibrasi sensor. Kemudian, setelah dikalibrasi, kita harus menulis algoritme yang menggunakan pembacaan dari sensor untuk menjaga robot tetap pada jalurnya. Ini mungkin lebih rumit dari yang Anda harapkan. Kemudian dalam bab ini kita melihat solusi yang baik, tetapi untuk saat ini, kita akan menggunakan pendekatan yang lebih langsung.

Kode Arduino

Kode Arduino sangat sederhana untuk aplikasi ini. Kami akan membaca setiap sensor dan mengirimkan hasilnya ke Pi melalui koneksi serial, 100 kali per detik. Namun, karena kita memerlukan pembacaan sensor lebih sering selama kalibrasi, kita perlu mengetahui kapan kalibrasi dijalankan karena kita ingin pembaruan terjadi 100 kali per detik untuk memastikan bahwa kita mendapatkan hasil yang baik.

1. Mulai sketsa baru di Arduino IDE.

2. Simpan sketsa sebagai line_follow1.
3. Masukkan kode berikut:

```
int ir1Pin = A0;
int ir2Pin = A1;
int ir3Pin = A2;
int ir4Pin = A3;
int ir5Pin = A4;

int ir1Val = 0;
int ir2Val = 0;
int ir3Val = 0;
int ir4Val = 0;
int ir5Val = 0;

void setup() {
        pinMode(ir1Pin, INPUT);
        pinMode(ir2Pin, INPUT);
        pinMode(ir3Pin, INPUT);
        pinMode(ir4Pin, INPUT);
        pinMode(ir5Pin, INPUT);

Serial.begin(9600);
}

void loop() {
      ir1Val = analogRead(ir1Pin);
      ir2Val = analogRead(ir2Pin);
      ir3Val = analogRead(ir3Pin);
      ir4Val = analogRead(ir4Pin);
      ir5Val = analogRead(ir5Pin);
      Serial.print(ir1Val); Serial.print(",");
      Serial.print(ir2Val); Serial.print(",");
      Serial.print(ir3Val); Serial.print(",");
      Serial.print(ir4Val); Serial.print(",");
      Serial.println(ir5Val);
      delay(100);
}
```

4. Simpan dan unggah sketsa.

Sketsa ini sangat sederhana. Yang kita lakukan hanyalah membaca masing-masing dari lima sensor dan mencetak hasilnya ke port serial.

Kode Python

Sebagian besar pemrosesan dilakukan pada Pi. Hal pertama yang perlu kita lakukan adalah mengkalibrasi sensor untuk mendapatkan nilai tinggi dan rendah. Untuk melakukan itu, kita perlu menyapu sensor bolak-balik melewati garis saat kita membaca nilai dari masing-masing sensor. Kami mencari nilai tertinggi dan terendah. Setelah kita melakukan beberapa operan melewati batas, kita harus memiliki nilai yang baik untuk dikerjakan. Dengan sensor yang dikalibrasi, saatnya untuk mulai bergerak. Dorong robot ke depan. Selama garis terdeteksi oleh sensor tengah, terus maju. Jika salah satu sensor di kiri atau kanan membaca garis, lakukan sedikit koreksi ke arah yang berlawanan untuk menyetel kembali. Jika salah satu sensor luar membaca garis, buat koreksi yang lebih dramatis. Ini membuat robot tetap mengikuti garis dan menangani belokan dengan mudah.

Untuk menjalankan kode ini dengan benar, buat baris untuk diikuti. Ada beberapa cara untuk melakukan ini. Jika Anda kebetulan memiliki lantai keramik putih, maka Anda bisa langsung menempelkan selotip listrik di atasnya. Pita listrik terangkat dari ubin tanpa merusaknya. Jika tidak, Anda dapat menggunakan lembaran kertas, papan poster, atau papan inti busa seperti yang digunakan untuk pameran sains. Sekali lagi, gunakan pita listrik untuk menandai garis. Pastikan untuk menambahkan beberapa kurva. Seperti halnya kode roamer, kita akan membahas ini di beberapa bagian. Kode yang kita tulis semakin panjang.

1. Buka file baru di IDLE IDE.
2. Simpan file sebagai line_follow1.py.
3. Impor perpustakaan yang diperlukan:

```
import serial
import time

from Adafruit_MotorHAT import Adafruit_MotorHAT as amhat
from Adafruit_MotorHAT import Adafruit_DCMotor as adamo
```

4. Buat objek motor. Untuk membuat kode lebih Pythonic, mari masukkan objek motor ke dalam daftar.

```
# create motor objects
motHAT = amhat(addr=0x60)
mot1 = motHAT.getMotor(1)
mot2 = motHAT.getMotor(2)
mot3 = motHAT.getMotor(3)
mot4 = motHAT.getMotor(4)

motors = [mot1, mot2, mot3, mot4]
```

5. Tentukan variabel yang diperlukan untuk mengendalikan motor. Sekali lagi, mari kita buat daftar.

```
# motor multipliers
motorMultiplier = [1.0, 1.0, 1.0, 1.0, 1.0]

# motor speeds
motorSpeed = [0,0,0,0]
```

6. Buka port serial.

```
# open serial port
ser = serial.Serial('/dev/ttyACM0', 9600
```

7. Tentukan variabel yang diperlukan. Seperti halnya motor, tentukan beberapa variabel sebagai daftar. (Ini terbayar nanti dalam kode. Saya berjanji.)

```
# create variables
# sensors
irSensors = [0,0,0,0,0]
irMins = [0,0,0,0,0]
irMaxs = [0,0,0,0,0]
irThesh = 50

# speeds
speedDef = 200
leftSpeed = speedDef
rightSpeed = speedDef
corMinor = 50
corMajor = 100
turnTime = 0.5
defTime = 0.01 driveTime = defTime
sweepTime = 1000 #duration of a sweep in milliseconds
```

8. Tentukan fungsi untuk menggerakkan motor. Meski mirip, kode ini berbeda dengan fungsi roamer.

```
def driveMotors(leftChnl = speedDef,rightChnl=speedDef,
duration = defTime):
        # determine the speed of each motor by multiplying
        # the channel by the motors multiplier
        motorSpeed[0] = leftChnl * motorMultiplier[0]
        motorSpeed[1] = leftChnl * motorMultiplier[1]
        motorSpeed[2] = rightChnl * motorMultiplier[2]
```

```
        motorSpeed[3] = rightChnl * motorMultiplier[3]
```

9. Ulangi daftar motor untuk mengatur kecepatan. Juga, ulangi daftar kecepatan motor.

```
# set each motor speed. Since the speed can be a
# negative number, we take the absolute value
for x in range(4):
        motors[x].setSpeed(abs(int(motorSpeed[x])))
```

10. Jalankan motor.

```
# run the motors. if the channel is negative, run
# reverse. else run forward
if(leftChnl < 0):
        motors[0].run(amhat.BACKWARD)
        motors[1].run(amhat.BACKWARD)
else:
        motors[0].run(amhat.FORWARD)
        motors[1].run(amhat.FORWARD)
if (rightChnl > 0):
        motors[2].run(amhat.BACKWARD)
        motors[3].run(amhat.BACKWARD)
else:
         motors[2].run(amhat.FORWARD)
         motors[3].run(amhat.FORWARD)

# wait for duration
time.sleep(duration)
```

11. Tentukan fungsi untuk membaca nilai sensor IR dari aliran serial dan menguraikannya.

```
def getIR():
        # read the serial port
        val = ser.readline().decode('utf-8')

        # parse the serial string
        parsed = val.split(',')
        parsed = [x.rstrip() for x in parsed]
```

12. Ulangi daftar irSensors untuk menetapkan nilai yang diuraikan, lalu hapus byte yang tersisa dari aliran serial.

```
if(len(parsed)==5):
        for x in range(5):
```

```
                irSensors[x] = int(parsed[x]+str(0))/10
    # flush the serial buffer of any extra bytes
    ser.flushInput()
```

13. Tentukan fungsi untuk mengkalibrasi sensor. Kalibrasi melewati empat siklus lengkap untuk membaca nilai minimum dan maksimum dari sensor.

```
    def calibrate():
        # set up cycle count loop
        direction = 1
        cycle = 0

        # get initial values for each sensor
        # and set initial min/max values getIR()

        for x in range(5):
            irMins[x] = irSensors[x]
            irMaxs[x] = irSensors[x]
```

14. Ulangi siklus lima kali untuk memastikan bahwa Anda mendapatkan empat pembacaan siklus penuh.

```
    while cycle < 5:
        #set up sweep loop
        millisOld = int(round(time.time()*1000))
        millisNew = millisOld
```

15. Selama waktu sweepTime, gerakkan motor dan baca sensor IR.

```
    while((millisNew-millisOld)<sweepTime):
        leftSpeed = speedDef * direction
        rightSpeed = speedDef * -direction
        # drive the motors
        driveMotors(leftSpeed, rightSpeed, driveTime)
        # read sensors
        getIR()
```

16. Perbarui irMins dan irMaxs jika nilai sensor di bawah atau di atas nilai irMins atau irMaxs saat ini.

```
    # set min and max values for each sensor
    for x in range(5):
        if(irSensors[x] < irMins[x]):
            irMins[x] = irSensors[x]
```

```
        elif(irSensors[x] > irMaxs[x]):
            irMaxs[x] = irSensors[x]

    millisNew = int(round(time.time()*1000))
```

17. Setelah satu siklus, ubah arah motor dan naikkan nilai siklus.

```
    # reverse direction
    direction = -direction

    # increment cycles
    cycle += 1
```

18. Setelah siklus selesai, dorong robot ke depan.

```
    # drive forward
    driveMotors(speedDef, speedDef, driveTime)
```

19. Tentukan fungsi followLine.

```
    def followLine():
        leftSpeed = speedDef
        rightSpeed = speedDef

        getIR()
```

20. Tentukan perilaku berdasarkan pembacaan senor. Jika garis terdeteksi oleh sensor paling kanan atau paling kiri, lakukan koreksi besar ke arah lain. Jika sensor kanan dalam atau kiri dalam mendeteksi garis, lakukan koreksi kecil ke arah lain; lain, mengemudi lurus.

```
    # find line and correct if necessary
    if(irMaxs[0]-irThresh <= irSensors[0]
    <= irMaxs[0]+irThresh):
        leftSpeed = speedDef-corMajor
    elif(irMaxs[1]-irThresh <= irSensors[1]
    <= irMaxs[1]+irThresh):
        leftSpeed = speedDef-corMinor
    elif(irMaxs[3]-irThresh <= irSensors[3]
    <= irMaxs[3]+irThresh):
        rightSpeed = speedDef-corMinor
    elif(irMaxs[4]-irThresh <= irSensors[4]
    <= irMaxs[4]+irThresh):
        rightSpeed = speedDef-corMajor
```

```
else:
        leftSpeed = speedDef
        rightSpeed = speedDef
# drive the motors
driveMotors(leftSpeed, rightSpeed, driveTime)
```

21. Masukkan kode untuk menjalankan program.

```
# execute program
try:
        calibrate()
 while 1:
        followLine()
        time.sleep(0.01)
except KeyboardInterrupt:
        mot1.run(amhat.RELEASE)
        mot2.run(amhat.RELEASE)
        mot3.run(amhat.RELEASE)
        mot4.run(amhat.RELEASE)
```

22. Simpan kodenya.
23. Tempatkan robot pada garis. Robot harus disejajarkan sehingga garis berjalan di antara roda kiri dan kanan, dan sensor tengah berada tepat di atasnya.
24. Jalankan programnya.

Robot Anda sekarang harus mengikuti garis, membuat koreksi jika mulai menyimpang dari garis. Anda mungkin perlu bermain dengan variabel corMinor dan corMajor untuk menyempurnakan perilaku.

Apa yang kami lakukan di sini dikenal sebagai kontrol proporsional. Ini adalah bentuk paling sederhana dari algoritma kontrol. Logika dasar di baliknya adalah jika robot Anda sedikit menyimpang, terapkan sedikit koreksi. Jika robotnya banyak keluar jalur, terapkan lebih banyak koreksi. Jumlah koreksi yang diterapkan pada robot ditentukan oleh seberapa besar kesalahannya.

Dengan kontrol proporsional saja, robot berusaha sangat keras untuk mengikuti garis. Bahkan mungkin berhasil; namun, Anda akan melihat bagaimana zig-zag di sepanjang garis. Perilaku ini dapat berkurang seiring waktu dan menjadi lancar; namun, ketika Anda memperkenalkan kurva, perilaku tidak menentu dimulai dari awal lagi. Kemungkinan besar, robot Anda melakukan koreksi berlebihan dan mengembara ke arah yang acak, meninggalkan garis jauh di belakang.

Ada cara yang lebih baik untuk mengontrol robot. Sebenarnya, ada beberapa cara yang lebih baik, semuanya dari bidang studi yang disebut loop kontrol. Loop kontrol adalah algoritma untuk meningkatkan respons mesin atau program. Kebanyakan dari mereka

menggunakan perbedaan antara keadaan saat ini dan keadaan yang diinginkan untuk mengontrol mesin. Perbedaan ini disebut kesalahan. Mari kita lihat sekaligus sistem kontrol tersebut selanjutnya.

8.7 MEMAHAMI KONTROL PID

Untuk mengontrol robot dengan lebih baik, Anda akan belajar tentang kontrol PID, dan saya akan mencoba membahasnya tanpa menjadi matematika yang berat. Kontroler PID adalah salah satu loop kontrol yang paling banyak digunakan karena keserbagunaan dan kesederhanaannya. Kami sebenarnya sudah menggunakan bagian dari pengontrol PID: kontrol proporsional. Bagian yang tersisa membantu kelancaran reaksi dan memberikan respon yang lebih baik.

8.8 KONTROL LOOP

Kontroler PID adalah anggota dari sekelompok algoritma yang disebut loop kontrol. Tujuan dari loop kontrol adalah menggunakan input dari proses yang diukur untuk membuat perubahan pada kontrol, atau kontrol, untuk mengkompensasi perbedaan antara keadaan saat ini dan keadaan yang diinginkan. Ada banyak jenis loop kontrol. Faktanya, loop kontrol adalah keseluruhan area studi yang disebut teori kontrol. Untuk tujuan kami, kami benar-benar hanya peduli tentang satu: proporsional, integral, dan turunan—atau PID.

Kontrol Proporsional, Integral, dan Derivatif

Menurut Wikipedia, “Pengontrol PID terus menghitung nilai kesalahan (e(t)) sebagai perbedaan antara setpoint yang diinginkan dan variabel proses yang diukur dan menerapkan koreksi berdasarkan istilah proporsional, integral, dan turunan. PID adalah inisialisasi untuk Proporsional-Integral-Derivatif, mengacu pada tiga istilah yang beroperasi pada sinyal kesalahan untuk menghasilkan sinyal kontrol.

Tujuan dari pengontrol adalah untuk menerapkan penyesuaian tambahan pada beberapa keluaran untuk mencapai hasil yang diinginkan. Dalam aplikasi kami, kami menggunakan umpan balik dari sensor IR untuk menerapkan perubahan pada motor kami. Perilaku yang diinginkan adalah robot yang tetap berada di tengah garis saat bergerak maju. Namun, proses ini dapat digunakan dengan sensor dan output apa pun; misalnya, PID digunakan dalam platform multirotor untuk tetap rata dan menjaga stabilitas. Sesuai dengan namanya, algoritma PID sebenarnya terdiri dari tiga bagian yaitu proporsional, integral, dan turunan. Setiap bagian adalah jenis kontrol; namun, jika digunakan secara independen, perilaku yang dihasilkan akan menjadi tidak menentu dan sulit diprediksi.

Kontrol Proporsional

Dalam kontrol proporsional, jumlah perubahan diatur sepenuhnya berdasarkan ukuran kesalahan. Semakin besar kesalahan, semakin banyak perubahan yang diterapkan. Kontrol proporsional murni akan mencapai keadaan kesalahan nol, tetapi mengalami kesulitan menghadapi perubahan drastis, yang menghasilkan osilasi berat.

Kontrol Integral

Kontrol integral tidak hanya mempertimbangkan kesalahan, tetapi juga waktu yang telah bertahan. Jumlah perubahan yang diterapkan untuk mengkompensasi kesalahan meningkat dari waktu ke waktu. Kontrol integral murni dapat membawa perangkat ke keadaan nol-kesalahan, tetapi bereaksi lambat dan cenderung overcompensate dan berosilasi.

Kontrol Derivatif

Kontrol derivatif tidak mempertimbangkan kesalahan, dan karena itu tidak pernah dapat membawa perangkat ke keadaan kesalahan nol. Namun, itu mencoba mengurangi perubahan kesalahan menjadi nol. Jika terlalu banyak kompensasi diterapkan, algoritme melampaui, dan kemudian menerapkan koreksi lain. Proses berlanjut dengan cara ini, menghasilkan pola koreksi yang terus meningkat atau menurun. Meskipun keadaan penurunan osilasi dianggap "stabil", algoritma tidak pernah mencapai keadaan kesalahan nol yang sebenarnya.

Menyatukan Mereka

Kontroler PID hanyalah jumlah dari tiga metode. Dengan menyatukannya, algoritme bertujuan untuk menghasilkan proses koreksi yang mulus yang membawa kesalahan ke nol. Waktu untuk sedikit matematika.

Mari kita mulai dengan mendefinisikan beberapa variabel.

e(t) adalah kesalahan dalam waktu, di mana (t) adalah waktu, atau saat ini.

K_p adalah parameter yang mewakili gain proporsional. Ketika kita mulai coding, ini adalah variabel proporsional.

K_i adalah parameter penguatan integral. Ini juga merupakan variabel.

K_d adalah parameter gain turunan. Dan Anda dapat menebaknya, variabel lain.

τmewakili nilai-nilai terintegrasi dari waktu ke waktu. Saya akan membahasnya.

Istilah proporsional pada dasarnya adalah kesalahan arus dikalikan dengan nilai K_p.

$$P_{out} = K_p e(t)$$

Bagian integral sedikit lebih rumit karena memperhitungkan semua kesalahan yang terjadi. Ini adalah jumlah kesalahan dari waktu ke waktu dan akumulasi koreksi.

$$I_{out} = K_i \int_0^t e(\tau)\,d\tau$$

Suku turunan adalah selisih antara kesalahan asli dan kesalahan saat ini dari waktu ke waktu, dan kemudian dikalikan dengan parameter turunan.

$$D_{out} = K_d \frac{de(t)}{dt}$$

Untuk menyatukan semuanya, persamaan PID kami terlihat seperti ini:

$$u(t) = \left(K_p e(t)\right) + \left(K_i \int_0^t e(\tau)\,d\tau\right) + \left(K_d \frac{de(t)}{dt}\right)$$

Itu saja untuk matematika. Untungnya, kita tidak harus menyelesaikannya sendiri. Python membuatnya sangat mudah. Namun, penting untuk memahami apa yang terjadi di dalam persamaan. Ada tiga parameter yang harus disesuaikan untuk menyempurnakan pengontrol PID. Dengan memahami bagaimana parameter ini digunakan, Anda akan dapat menentukan mana yang perlu disesuaikan dan kapan.

8.9 MENERAPKAN PENGONTROL PID

Untuk mengimplementasikan *controller,* kita perlu mengetahui beberapa hal. Apa hasil yang kita inginkan? Apa masukan kami? Apa keluaran kami? Tujuannya adalah untuk meningkatkan kinerja robot pengikut garis kami. Jadi, hasil yang kami inginkan adalah garis tetap berada di tengah robot saat bergerak maju. Input kami adalah sensor IR. Ketika sensor luar berada di atas area gelap (garis), kesalahannya dua kali lipat dari sensor dalam. Dengan cara ini, kita akan tahu apakah robot itu sedikit *off-center* atau banyak *off-center*. Selain itu, kedua sensor kiri akan memiliki nilai negatif, dan sensor kanan akan memiliki nilai positif, sehingga kita akan mengetahui arah mana yang mati. Akhirnya, output kami adalah motor. Lebih tepatnya, output kami adalah perbedaan kecepatan antara saluran motor kiri dan kanan.

Kode

Kode untuk latihan ini merupakan modifikasi dari kode sebelumnya. Padahal, kode Arduino tidak perlu diubah sama sekali. Ini adalah logika yang kami terapkan pada Raspberry Pi yang diperbarui.

Kode Pi raspberry

Kami akan memodifikasi kode line_follower1 untuk menggunakan PID daripada algoritma proporsional. Untuk melakukan itu, kita perlu memperbarui fungsi getIR untuk memperbarui variabel baru yang disebut sensorErr. Kami kemudian akan mengganti kode di dalam fungsi followLine dengan kode PID kami.

1. Buka file line_follower1 di IDLE IDE.
2. Dari menu file, pilih Save as, dan simpan file sebagai line_follower2.py.
3. Di bagian variabel, di bawah #sensors, tambahkan kode berikut:

```
# PID
sensorErr = 0
lastTime = int(round(time.time()*1000))
lastError = 0
target = 0
kp = 0.5
ki = 0.5
kd = 1
```

4. Buat fungsi PID.

```
def PID(err):
        # check if variables are defined before use
        # the first time the PID is called these variables will
        # not have been defined
        try: lastTime
        except NameError: lastTime = int(round(time.
        time()*1000)-1)

        try: sumError
        except NameError: sumError = 0

        try: lastError
        except NameError: lastError = 0

        # get the current time
        now = int(round(time.time()*1000))
        duration = now-lastTime

        # calculate the errorerror = target – err
        sumError += (error * duration)
        dError = (error - lastError)/duration

        # calculate PID
        output = kp * error + ki * sumError + kd * dError
        # update variables
        lastError = error
        lastTime = now

        # return the output value
        return output
```

5. Ganti fungsi followLine dengan ini:

```
def followLine():
        leftSpeed = speedDef
        rightSpeed = speedDef

        getIR()
```

```
prString = ''
for x in range(5):
        prString += ('IR' + str(x) + ': ' +str(irSensors[x]) + ' ')
print prString

    # find line and correct if necessary if(irMaxs[0]-irThresh
    <= irSensors[0]
    <= irMaxs[0]+irThresh):
        sensorErr = 2
    elif(irMaxs[1]-irThresh <= irSensors[1]
    <= irMaxs[1]+irThresh):
        sensorErr = 1
    elif(irMaxs[3]-irThresh <= irSensors[3]
    <= irMaxs[3]+irThresh):
        sensorErr = -1
    elif(irMaxs[4]-irThresh <= irSensors[4]
    <= irMaxs[4]+irThresh):
        sensorErr = -1
    else:
          sensorErr = 0
    # get PID results
    ratio = PID(sensorErr)

    # apply ratio
    leftSpeed = speedDef * ratio
    rightSpeed = speedDef * -ratio

    # drive the motors
    driveMotors(leftSpeed, rightSpeed, driveTime)
```

6. Simpan filenya.
7. Tempatkan robot pada garis.
8. Jalankan kodenya.

Sekali lagi, robot Anda harus mencoba mengikuti garis. Jika mengalami masalah saat melakukannya, mulailah bekerja dengan variabel Kp, Ki, dan Kd. Variabel-variabel ini perlu disesuaikan untuk hasil terbaik. Setiap robot berbeda.

8.10 RINGKASAN

Dalam bab ini, kami menambahkan beberapa sensor baru ke robot. Sensor IR diterapkan dalam aplikasi *line-following.* Mereka juga dapat digunakan untuk mendeteksi tepi

permukaan. Fungsi ini berguna jika Anda ingin mencegah robot Anda meluncur dari meja atau menuruni tangga.

Implementasi *line-following* pertama kami menggunakan kontrol proporsional dasar untuk mengarahkan robot. Ini adalah fungsional, tapi hampir tidak. Cara yang jauh lebih baik untuk melakukan ini adalah penggunaan loop kontrol yang disebut pengontrol PID, yang menggunakan beberapa faktor, termasuk kesalahan dari waktu ke waktu, untuk membuat koreksi lebih lancar. Anda mengetahui bahwa Anda dapat menyesuaikan pengaturan ID dengan menggunakan parameter PID yang direpresentasikan dalam kode kita, dengan variabel Kp, Ki, dan Kd. Dengan nilai yang tepat, osilasi dapat dihilangkan sepenuhnya, menyebabkan robot mengikuti garis dengan lancar.

BAB 9
PENGANTAR OPENCV

Kami telah menempuh perjalanan jauh sejak bab pertama memperkenalkan Raspberry Pi. Pada titik ini, Anda telah belajar tentang Pi dan Arduino. Anda telah belajar cara memprogram kedua papan. Anda telah bekerja dengan sensor dan motor. Anda telah membangun robot Anda dan memprogramnya untuk berkeliaran dan mengikuti garis. Namun, sejujurnya, Anda tidak benar-benar membutuhkan kekuatan Raspberry Pi. Bahkan, itu menjadi sedikit hambatan. Semua yang telah Anda lakukan dengan robot—roaming dan *line following,* Anda dapat melakukannya dengan baik dengan Arduino dan tanpa Pi. Sekarang saatnya untuk menunjukkan kekuatan sebenarnya dari Pi dan untuk mempelajari mengapa Anda ingin menggunakannya di robot Anda. Dalam bab ini, kita akan melakukan sesuatu yang tidak dapat Anda lakukan dengan Arduino saja. Kami akan menghubungkan kamera web sederhana dan mulai bekerja dengan apa yang umumnya dikenal sebagai visi komputer.

9.1 VISI KOMPUTER

Visi komputer adalah kumpulan algoritma yang memungkinkan komputer menganalisis gambar dan mengekstrak informasi yang berguna. Ini digunakan dalam banyak aplikasi, dan dengan cepat menjadi bagian dari kehidupan sehari-hari. Jika Anda memiliki ponsel cerdas, kemungkinan Anda memiliki setidaknya satu aplikasi yang menggunakan visi komputer. Sebagian besar kamera kelas menengah hingga kelas atas yang baru memiliki deteksi wajah bawaan. Facebook menggunakan visi komputer untuk deteksi wajah. Visi computer digunakan oleh perusahaan pelayaran untuk melacak paket di gudang mereka. Dan, tentu saja, ini digunakan dalam robotika untuk navigasi, deteksi objek, penghindaran objek, dan banyak perilaku lainnya.

Semuanya dimulai dengan sebuah gambar. Komputer menganalisis gambar untuk mengidentifikasi garis, sudut, dan area warna yang luas. Proses ini disebut ekstraksi fitur, dan ini adalah langkah pertama di hampir semua algoritma computer vision. Setelah fitur diekstraksi, komputer dapat menggunakan informasi ini untuk banyak tugas yang berbeda. Pengenalan wajah dilakukan dengan membandingkan fitur terhadap file XML yang berisi data fitur untuk wajah. File XML ini disebut cascades. Mereka tersedia untuk berbagai jenis objek, bukan hanya wajah. Teknik yang sama dapat digunakan untuk pengenalan objek. Anda cukup menyediakan aplikasi dengan informasi fitur untuk objek yang Anda minati.

Visi komputer juga menggabungkan video. Pelacakan gerak adalah aplikasi umum untuk visi komputer. Untuk mendeteksi gerakan, komputer membandingkan frame individu dari kamera stasioner. Jika tidak ada gerakan, fitur tidak akan berubah antar frame. Jadi, jika komputer mengidentifikasi perbedaan antara frame, kemungkinan besar ada gerakan. Pelacakan gerak berbasis visi komputer lebih andal daripada sensor IR, seperti sensor PIR yang dibahas di Bab 8.

Aplikasi computer vision terbaru yang menarik adalah augmented reality. Fitur yang diekstraksi dari aliran video dapat digunakan untuk mengidentifikasi pola unik pada permukaan. Karena komputer mengetahui polanya, ia dapat dengan mudah menghitung sudut permukaan. Model 3D kemudian ditumpangkan di atas pola. Model 3D ini bisa berupa sesuatu yang fisik, seperti bangunan, atau bisa juga berupa objek planar dengan teks dua dimensi. Arsitek menggunakan teknik ini untuk menunjukkan kepada klien seperti apa bentuk bangunan jika dibandingkan dengan kaki langit. Museum menggunakannya untuk memberikan informasi lebih lanjut tentang pameran atau seniman. Semua ini adalah contoh visi komputer dalam pengaturan modern. Tetapi daftar aplikasi terlalu besar untuk dibahas secara mendalam di sini, dan terus bertambah.

9.2 OPENCV

Hanya beberapa tahun yang lalu, visi komputer tidak benar-benar dapat diakses oleh penghobi. Itu membutuhkan banyak matematika yang berat dan pemrosesan yang bahkan lebih berat. Proyek computer vision umumnya dilakukan dengan menggunakan laptop, sehingga aplikasinya terbatas.

OpenCV telah ada untuk sementara waktu. Pada tahun 1999, Intel Research menetapkan standar terbuka untuk mempromosikan pengembangan visi komputer. Pada 2012, itu diambil alih oleh OpenCV Foundation nirlaba. Anda dapat mengunduh versi terbaru di situs web mereka. Namun, perlu sedikit usaha ekstra untuk menjalankannya di Raspberry Pi. Kami akan segera membahasnya. OpenCV ditulis secara native dalam C++; namun, ini dapat digunakan dalam C, Java, dan Python. Kami tertarik dengan implementasi Python. Karena library pengontrol motor kami tidak kompatibel dengan Python 3, kami perlu menginstal OpenCV untuk Python 2.7.

Menginstal OpenCV

Kami akan menginstal OpenCV pada Raspberry Pi. Anda ingin memastikan bahwa Raspberry Pi Anda dicolokkan ke pengisi daya daripada ke baterai, dan beri diri Anda banyak waktu untuk instalasi. Kami akan mengkompilasi OpenCV dari sumber, yang berarti bahwa kami akan mengunduh kode sumber dari Internet dan membangunnya langsung di Pi. Berhati-hatilah bahwa meskipun prosesnya tidak sulit, prosesnya memakan waktu lama dan melibatkan banyak perintah Linux. Saya biasanya memulai proses di malam hari dan membiarkan pembangunan terakhir berjalan dalam semalam.

1. Masuk ke Raspberry Pi Anda.
2. Buka jendela terminal pada Pi.
3. Pastikan Raspberry Pi sudah terupdate.
   sudo apt-get update
   sudo apt-get upgrade
   sudo rpi-update
   sudo reboot

4. Perintah ini menginstal prasyarat untuk membangun OpenCV.
   sudo apt-get install build-essential git cmake pkg-config
   sudo apt-get install libjpeg-dev libtiff5-dev
   libjasper-dev libpng12-dev
   sudo apt-get install libavcodec-dev
   libavformat-devlibswscale-dev libv4l-dev
   sudo apt-get install libxvidcore-dev
   libx264-devsudo apt-get install libgtk2.0-dev
   sudo apt-get install libatlas-base-dev gfortran

5. Unduh kode sumber OpenCV dan file kontribusi OpenCV. File yang dikontribusikan mengandung banyak fungsi yang belum dimasukkan ke dalam distribusi OpenCV utama.
   cd ~
   git clone https://github.com/Itseez/opencv.git
   cd opencv
   git checkout 3.1.0
   cd ~
   git clone https://github.com/Itseez/opencv_contrib.git
   cd opencv_contrib
   git checkout 3.1.0

6. Instal pustaka pengembangan Python dan pip.
   sudo apt-get install python2.7-dev
   wget https://bootstrap.pypa.io/get-pip.py
   sudo python get-pip.py

7. Pastikan NumPy sudah terinstal.
   pip install numpy

8. Siapkan kode sumber untuk kompilasi.
   cd ~/opencv
   mkdir build
   cd build
   *cmake -D CMAKE_BUILD_TYPE=RELEASE *
   *-D CMAKE_INSTALL_PREFIX=/usr/local *
   *-D INSTALL_C_EXAMPLES=OFF *
   *-D INSTALL_PYTHON_EXAMPLES=ON *
   *-D OPENCV_EXTRA_MODULES_PATH=~/opencv_contrib/modules *
   -D BUILD_EXAMPLES=ON .. ..

9. Sekarang mari kita kompilasi kode sumbernya. Bagian ini akan memakan waktu cukup lama.
   Beberapa orang mencoba memanfaatkan keempat inti di CPU ARM Raspberry Pi. Namun, saya menemukan ini rawan kesalahan, dan itu tidak pernah berhasil untuk saya. Saran saya adalah untuk menggigit peluru: biarkan Pi menentukan jumlah inti yang akan digunakan dan biarkan berjalan. Jika Anda ingin berani, Anda dapat memaksa Pi untuk menggunakan empat inti dengan menambahkan sakelar –j4 ke baris berikut
   make

10. Jika Anda mencoba sakelar –j4 dan gagal, sekitar jam empat, masukkan baris berikut:
    make clean
    make

11. Dengan kode sumber yang dikompilasi, Anda sekarang dapat menginstalnya.
    sudo make install
    sudo ldconfig

12. Uji instalasi dengan membuka baris perintah Python.
    python

13. Impor OpenCV.
    >>>import cv2

Anda sekarang harus memiliki versi operasi OpenCV yang diinstal pada Raspberry Pi Anda. Jika perintah impor tidak berfungsi, Anda perlu menentukan mengapa perintah itu tidak diinstal. Internet adalah panduan Anda untuk pemecahan masalah.

9.3 MEMILIH KAMERA

Sebelum kita benar-benar dapat menggunakan OpenCV untuk bekerja pada robot kita, kita perlu memasang kamera. Ada beberapa opsi dengan Raspberry Pi: Kamera Pi atau kamera web USB. Kamera Pi terhubung langsung ke port yang dirancang khusus untuk itu. Setelah terhubung, Anda harus masuk ke raspi-config dan mengaktifkannya. Keuntungan dari Kamera Pi adalah sedikit lebih cepat daripada kamera USB karena terhubung langsung ke papan. Itu tidak melalui bus serial USB. Ini memberikan sedikit keuntungan.

Sebagian besar Kamera Pi dilengkapi dengan kabel pita pendek 6 inci. Karena penempatan Raspberry Pi di robot kami, ini tidak cukup. Dimungkinkan untuk memesan kabel yang lebih panjang. Adafruit memiliki beberapa pilihan. Namun, untuk proyek ini, kami akan menggunakan kamera web sederhana. Kamera USB sudah tersedia di pengecer elektronik

mana pun. Ada banyak pilihan online, juga. Untuk aplikasi dasar ini, kita tidak memerlukan sesuatu yang sangat kuat. Kamera apa pun yang dapat memberikan gambar yang layak akan berhasil. Memiliki resolusi tinggi juga bukan masalah. Karena kami menjalankan kamera dengan sumber daya Raspberry Pi yang terbatas, resolusi yang lebih rendah sebenarnya akan membantu kinerja. Ingat, OpenCV menganalisis setiap piksel bingkai demi piksel. Semakin banyak piksel dalam sebuah gambar, semakin banyak pemrosesan yang harus dilakukan.

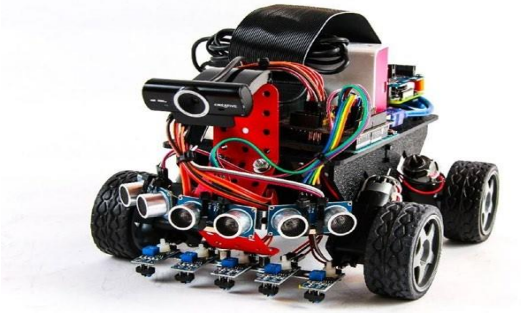
Untuk robot saya, saya memilih Live! Cam Sync HD by Creative (lihat Gambar 9-1), webcam HD dasar dengan mikrofon internal yang beroperasi melalui port USB 2 standar. Kami tidak memerlukan mikrofon untuk proyek ini, tetapi mungkin akan ada kebutuhan di masa mendatang. Ini menangkap video HD 720p, yang mungkin sedikit berlebihan untuk robot kami, tetapi jika ada penurunan kinerja, saya selalu mengurangi resolusi dalam perangkat lunak.

Gambar 9-1. Hidup Kreatif! Sinkronisasi Kamera HD

9.4 MEMASANG KAMERA

Sebagian besar webcam dipasang di atas monitor. Mereka biasanya memiliki penjepit lipat untuk memberikan dukungan bagi kamera saat berada di monitor. Sayangnya, penjepit ini biasanya dibentuk sebagai bagian dari bodi kamera dan tidak dapat dilepas tanpa merusaknya. Hal ini tentu berlaku dengan Live! Sinkronisasi Kamera. Jadi, sekali lagi, sedikit kreativitas ikut bermain. Braket yang saya gunakan untuk memasang sensor terlepas dari bagian depan robot pada sudut 45 derajat. Untuk membuat segalanya lebih mudah pada diri saya sendiri, saya memilih untuk tidak mengebor lubang di dudukan kamera.

Sebaliknya, saya menggunakan pita pemasangan tepercaya dan beberapa braket dari Erector Set. Saat saya memasangnya, saya ingin menaikkannya cukup tinggi dan sedikit mengarah ke bawah. Idenya adalah untuk memberikan pandangan terbaik ke depan, dan objek apa pun langsung di depannya. Saya juga ingin lensa sedekat mungkin dengan sumbu tengah untuk menjaga hal-hal lebih sederhana di sisi perangkat lunak. Gambar 9-2 menunjukkan robot setelah kamera dipasang.



Gambar 9-2. Kamera terpasang di robot

9.5 DASAR-DASAR OPENCV

OpenCV memiliki banyak kemampuan. Ini menawarkan lebih dari 500 perpustakaan dan ribuan fungsi. Ini adalah topik yang sangat besar—topik yang terlalu besar untuk dibahas dalam satu bab. Saya akan membahas dasar-dasar yang diperlukan untuk melakukan beberapa tugas sederhana pada robot Anda. Saya mengatakan tugas-tugas sederhana. Tugas-tugas ini hanya sederhana karena OpenCV mengabstraksi jumlah matematika yang monumental yang terjadi di latar belakang. Ketika saya mempertimbangkan keadaan robotika hobi beberapa tahun yang lalu, saya merasa luar biasa untuk dapat dengan mudah mengakses bahkan dasar-dasarnya.

Tujuannya adalah untuk membangun robot yang dapat mengidentifikasi bola dan bergerak ke arahnya pada akhir bab ini. Fungsi-fungsi yang saya cakup akan membantu kita mencapai tujuan itu. Saya sangat menyarankan untuk menghabiskan waktu membaca beberapa tutorial di situs web OpenCV (https://opencv.org). Untuk bekerja dengan OpenCV dalam kode Python Anda, Anda perlu mengimpornya. Dan, saat Anda melakukannya, Anda mungkin perlu mengimpor perpustakaan NumPy juga. NumPy menambahkan banyak fungsi matematika dan penanganan angka yang membuat bekerja dengan OpenCV lebih mudah.

Semua kode terkait gambar Anda harus dimulai dengan ini:

```
import cv2
import numpy as np
```

Dalam diskusi kode dalam bab ini, saya menganggap ini telah dilakukan. Fungsi yang diawali dengan cv2 adalah fungsi OpenCV. Jika diawali dengan np, itu adalah fungsi NumPy. Penting untuk membuat perbedaan ini jika Anda ingin memperluas apa yang Anda baca dalam buku ini. OpenCV dan NumPy adalah dua perpustakaan terpisah, tetapi OpenCV sering menggunakan NumPy.

9.6 BEKERJA DENGAN GAMBAR

Di bagian ini, Anda mempelajari cara membuka gambar dari file dan cara merekam video langsung dari kamera. Kami kemudian akan melihat cara memanipulasi dan menganalisis gambar untuk mendapatkan informasi yang dapat digunakan dari mereka. Secara khusus, kami akan bekerja tentang cara mengidentifikasi bola dengan warna tertentu dan melacak posisinya dalam bingkai.

Tapi pertama-tama, kami memiliki sedikit masalah ayam atau telur. Kita perlu melihat hasil manipulasi gambar kita di semua latihan. Untuk melakukan itu, kita harus mulai dengan cara menampilkan gambar. Itu adalah sesuatu yang akan kita gunakan ekstensif, dan sangat mudah digunakan. Tetapi saya ingin memastikan bahwa saya menutupinya terlebih dahulu, sebelum Anda mempelajari cara menangkap gambar.

Menampilkan Gambar

Sebenarnya sangat mudah untuk menampilkan gambar di OpenCV. Fungsi imshow()menyediakan fungsionalitas ini. Fungsi ini digunakan dengan gambar diam dan gambar video, dan penerapannya tidak berubah di antara keduanya. Fungsi

imshow()membuka jendela baru untuk menampilkan gambar. Saat Anda menyebutnya, Anda harus memberikan nama untuk jendela serta gambar atau bingkai yang ingin Anda tampilkan. Ini adalah poin penting tentang bagaimana OpenCV bekerja dengan video. Karena OpenCV memperlakukan video sebagai serangkaian frame individu, hampir semua fungsi yang digunakan untuk memodifikasi atau menganalisis gambar berlaku untuk video. Ini jelas termasuk imshow(). Jika kita ingin menampilkan gambar yang kita muat ke dalam variabel img, akan terlihat seperti ini:

cv2.imshow('img', img)

Dalam contoh ini, parameter pertama adalah nama jendela. Itu muncul di bilah judul jendela. Parameter kedua adalah variabel yang menahan gambar kita. Format untuk menampilkan video sama persis. Saya biasanya menggunakan variabel cap untuk pengambilan video, jadi kodenya akan terlihat seperti ini:

cv2.imshow('cap', cap)

Seperti yang Anda lihat, kodenya sama. Sekali lagi, ini karena OpenCV memperlakukan video sebagai rangkaian frame individu. Faktanya, pengambilan video bergantung pada loop untuk terus menangkap frame berikutnya. Jadi pada intinya, menampilkan gambar diam dari file dan bingkai individual dari kamera adalah hal yang sama persis. Ada satu elemen yang tersisa untuk menampilkan gambar. Untuk benar-benar menampilkan gambar, fungsi imshow() mengharuskan waitKey() juga dipanggil. Fungsi waitKey()menunggu jumlah milidetik yang ditentukan untuk menekan tombol keyboard. Banyak orang menggunakan ini untuk menangkap kunci Keluar. Saya biasanya memberikannya nol kecuali saya membutuhkan penekanan tombol.

cv2.waitKey(0)

Kami menggunakan imshow() dan waitKey()secara ekstensif di seluruh bab ini.

9.7 MENANGKAP GAMBAR

Ada beberapa sumber gambar yang diperlukan untuk bekerja dengan OpenCV, yang semuanya merupakan variasi dari dua faktor: file atau kamera, dan gambar diam atau video. Untuk sebagian besar, kami hanya peduli dengan video dari kamera karena kami menggunakan OpenCV untuk tujuan navigasi. Tetapi ada keuntungan untuk semua metode. Membuka file gambar diam adalah cara terbaik untuk mempelajari teknik baru, terutama saat Anda bekerja dengan aspek tertentu dari visi komputer. Misalnya, jika Anda mempelajari cara menggunakan filter untuk mengidentifikasi bola dengan warna tertentu, menggunakan gambar diam yang terdiri dari tiga bola berwarna berbeda (dan tidak ada yang lain) memungkinkan Anda untuk fokus pada tujuan tertentu tanpa harus khawatir tentang kerangka kerja yang mendasari untuk menangkap aliran video langsung. Oh, dan itu sedikit bayangan, jika Anda tidak memahaminya.

Teknik-teknik yang dipelajari dari menangkap gambar diam dengan kamera dapat diterapkan pada lingkungan hidup. Ini memungkinkan Anda untuk mengasah atau menyempurnakan kode dengan menggunakan gambar yang berisi elemen dunia nyata. Jelas,

merekam video langsung adalah apa yang kami cari untuk digunakan di robot. Aliran video langsung adalah apa yang akan kami gunakan untuk mengidentifikasi objek target kami dan kemudian menavigasi ke sana. Seiring bertambahnya pengalaman visi komputer Anda, Anda mungkin akan menambahkan deteksi gerakan atau metode lain ke repertoar Anda. Karena tujuan kamera pada robot adalah untuk mengumpulkan informasi lingkungan secara real time, video langsung diperlukan.

Video dari sebuah file juga sangat berguna untuk proses pembelajaran. Anda mungkin ingin merekam video langsung dari robot Anda dan menyimpannya ke file untuk analisis nanti. Katakanlah Anda sedang mengerjakan proyek robot Anda di waktu luang apa pun yang dapat Anda temukan sepanjang hari. Anda dapat membawa laptop Anda, tetapi membawa robot berkeliling adalah cerita yang berbeda. Jadi, jika Anda merekam video dari robot, Anda dapat mengerjakan algoritme visi komputer tanpa harus membawa robot.

Ingat, salah satu hal hebat tentang Python dan OpenCV adalah bahwa mereka sebagian besar diabstraksi dan platform independen. Jadi, kode yang Anda tulis di port mesin Windows ke Raspberry Pi Anda. Melakukan perjalanan bisnis dan mengharapkan waktu senggang di hotel? Pergi ke tempat keluarga untuk liburan dan ingin pergi sesekali? Menyelinap dalam pemrograman robot kecil selama jam makan siang Anda atau di antara kelas? Gunakan video yang direkam dengan instance lokal Python dan OpenCV, dan kerjakan algoritma deteksi Anda. Saat Anda tiba di rumah, Anda dapat mentransfer kode itu ke robot Anda dan mengujinya secara langsung. Di bagian ini, kami menggunakan tiga teknik pertama. Saya menunjukkan cara menyimpan dan membuka file video, tetapi sebagian besar, kami akan menggunakan gambar diam untuk mempelajari algoritme deteksi dan video langsung untuk mempelajari pelacakan.

Membuka File Gambar

OpenCV membuat bekerja dengan gambar dan file menjadi sangat mudah, terutama mengingat apa yang terjadi di latar belakang untuk memungkinkan operasi ini. Membuka file gambar tidak berbeda. Kami menggunakan fungsi imread() untuk membuka file gambar dari penyimpanan lokal. Fungsi imread() mengambil dua parameter: nama file dan jenis warna bendera. Nama file jelas diperlukan untuk membuka file. Bendera jenis warna menentukan apakah akan membuka gambar dalam warna atau skala abu-abu.

Mari kita buka dan tampilkan sebuah gambar. Saya akan menggunakan gambar tiga bola berwarna yang juga digunakan nanti di bab ini untuk mempelajari cara mendeteksi warna. Latihan ini dapat dilakukan di Pi atau di komputer Anda jika Anda telah menginstal Python dan OpenCV di dalamnya.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai *open_image.py.*
3. Masukkan kode berikut:

```
import cv2
img = cv2.imread('color_balls_small.jpg')
cv2.imshow('image',img)
```

cv2.waitKey(0)

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan python open_image.py dan tekan Return.

Sebuah jendela terbuka untuk menampilkan gambar tiga bola berwarna dengan latar belakang putih (lihat Gambar 9-3). Tekan sembarang tombol untuk menutupnya.

Gambar 9-3. Bola tiga warna

Karena cara IDLE berinteraksi dengan sistem GUI pada mesin berbasis Linux, jendela gambar tidak akan menutup dengan benar jika Anda menjalankan kode langsung dari IDLE. Namun, dengan menjalankan kode dari terminal, kami tidak memiliki masalah ini.

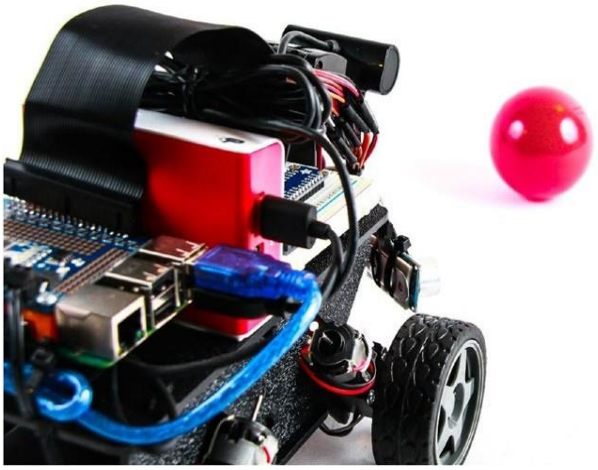
9.8 MENANGKAP VIDEO

Menangkap video dengan kamera Anda sedikit berbeda dengan membuka file. Ada beberapa langkah lagi untuk menggunakan video. Salah satu perubahannya adalah kita harus menggunakan loop untuk mendapatkan banyak frame; jika tidak, OpenCV hanya akan menangkap satu frame, yang bukan itu yang kita inginkan. Perulangan while terbuka umumnya digunakan. Ini menangkap video sampai kami secara aktif menghentikannya.

Untuk mempermudah pengujian, saya menempatkan bola langsung di depan kamera (lihat Gambar 9-4). Saat ini, kami hanya ingin mengabadikan gambar.

Untuk menangkap video dari kamera, kita akan membuat objek videoCapture() dan kemudian gunakan metode read() dalam satu lingkaran untuk menangkap frame. Metode read() mengembalikan dua objek: nilai kembalian dan bingkai gambar. Nilai yang dikembalikan hanyalah bilangan bulat yang memverifikasi keberhasilan atau kegagalan pembacaan. Jika pembacaan berhasil, nilainya adalah 1; jika tidak, pembacaan gagal dan mengembalikan 0. Untuk mencegah kesalahan yang menyebabkan kode Anda salah, Anda dapat menguji untuk

melihat apakah pembacaan berhasil.



Gambar 9-4. Bola diposisikan di depan robot untuk pengujian

Kami peduli dengan bingkai foto. Jika pembacaan berhasil, gambar dikembalikan. Jika tidak, maka objek null dikembalikan pada tempatnya. Karena objek null tidak dapat mengakses metode OpenCV, saat Anda mencoba mengubah atau memanipulasi gambar, kode Anda akan macet. Inilah mengapa ide yang baik untuk menguji keberhasilan operasi baca.

Melihat Kamera

Pada latihan berikutnya, kita menyalakan kamera video yang dipasang tadi untuk melihat video.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai view_camera.py.
3. Masukkan kode berikut:

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)
while(True):
        ret,frame = cap.read()

        cv2.imshow('video', frame)
        if cv2.waitKey(1) & 0xff == ord('q'):
                break
        cap.release()
        cv2.destroyAllWindows()
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder kerja Anda tempat skrip disimpan.
7. Ketik **sudo python view_camera.py.**

Ini akan membuka jendela yang menampilkan apa yang dilihat kamera Anda. Jika Anda menggunakan sesi desktop jarak jauh untuk bekerja pada Pi, Anda mungkin melihat pesan peringatan ini: Xlib: extension RANR missing on display :10. Pesan ini berarti bahwa sistem sedang mencari fungsionalitas yang tidak disertakan dalam vncserver. Itu bisa diabaikan. Jika Anda khawatir tentang kecepatan refresh gambar video, perlu diingat bahwa kami meminta banyak sekali Raspberry Pi saat kami menjalankan beberapa jendela melalui sesi desktop jarak jauh. Jika Anda menghubungkan monitor dan keyboard untuk mengakses Pi, itu berjalan lebih cepat. Pengambilan video bekerja lebih cepat jika Anda menjalankannya tanpa visualisasi.

Merekam Video

Merekam video adalah perpanjangan dari melihat kamera. Untuk merekam, Anda harus mendeklarasikan codec video yang akan Anda gunakan, dan kemudian mengatur objek VideoWriter yang menulis video yang masuk ke kartu SD. OpenCV menggunakan kode FOURCC untuk menunjuk codec. FOURCC adalah kode empat karakter untuk codec video. Anda dapat menemukan informasi lebih lanjut tentang FOURCC di www.fourcc.org.

Saat membuat objek VideoWriter, kita perlu memberikan beberapa informasi. Pertama, kita harus memberikan nama file untuk menyimpan video. Selanjutnya, kami menyediakan codec, diikuti dengan frame rate dan resolusi. Setelah objek VideoWriter dibuat, kita hanya perlu menulis masing-masing dari ke file menggunakan metode write() dari objek VideoWriter.

Mari rekam beberapa video feed robot kita. Kami akan menggunakan codec XVID untuk menulis ke file bernama test_video.avi. Daripada memulai dari awal, kami akan menggunakan kode pengambilan video dari latihan sebelumnya.

1. Buka file view_camera.py di IDLE IDE.
2. Pilih File ➤ Save as dan simpan file sebagai record_camera.py.
3. Perbarui kode. Berikut ini, baris baru dicetak tebal:

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)
fourcc = cv2.VideoWriter_fourcc(*'XVID')
vidWrite = cv2.VideoWriter('test_video.avi', \
                fourcc,  20,  (640,480))

while(True):
        ret,frame = cap.read()
        vidWrite.write(frame)

        cv2.imshow('video', frame)
        if cv2.waitKey(1) & 0xff == ord('q'):
```

```
            break

    cap.release()
    vidWrite.release()
    cv2.destroyAllWindows()
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder kerja Anda tempat skrip disimpan.
7. Ketik sudo python record_camera.py.
8. Biarkan video berjalan selama beberapa detik, lalu tekan Q untuk mengakhiri program dan menutup jendela.

Anda sekarang harus memiliki file video di direktori kerja Anda. Selanjutnya, kita akan melihat membaca video dari file. Ada beberapa item yang perlu diperhatikan dalam kode. Saat kami membuat objek VideoWriter, kami menyediakan resolusi video sebagai tuple. Ini adalah praktik yang sangat umum di seluruh OpenCV. Juga, kami harus melepaskan objek VideoWriter. Ini menutup file dari penulisan.

Membaca Video dari File

Memutar ulang video dari file sama persis dengan melihat video dari kamera. Satu-satunya perbedaan adalah bahwa alih-alih memberikan indeks ke perangkat video, kami memberikan nama file yang akan diputar. Kami akan menggunakan variabel ret untuk menguji akhir file video; jika tidak, kami akan mendapatkan kesalahan ketika tidak ada lagi video untuk diputar. Dalam latihan ini, kita hanya akan memutar ulang video yang telah kita rekam pada latihan sebelumnya. Kode tersebut akan terlihat sangat familiar.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai view_video.py.
3. Masukkan kode berikut:

```
import cv2
import numpy as np
cap = cv2.VideoCapture('test_video.avi')
while(True):
        ret,frame = cap.read()

        if ret:
            cv2.imshow('video', frame)
        if cv2.waitKey(1) & 0xff == ord('q'):
            break

cap.release()
cv2.destroyAllWindows()
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder kerja Anda tempat skrip disimpan.
7. Ketik **sudo python view_video.py.**

Sebuah jendela baru terbuka. Ini menampilkan file video yang kami rekam di latihan sebelumnya. Ketika akhir file tercapai, video berhenti. Tekan Q untuk mengakhiri program dan menutup jendela.

9.9 TRANSFORMASI GAMBAR

Sekarang setelah Anda mengetahui lebih banyak tentang cara mendapatkan gambar, mari kita lihat beberapa hal yang dapat kita lakukan dengan gambar tersebut. Kita akan melihat beberapa operasi yang sangat mendasar. Operasi ini dipilih karena mereka akan membantu kami mencapai tujuan kami melacak bola. OpenCV sangat kuat, dan memiliki lebih banyak kemampuan daripada yang saya sajikan di sini.

Membalik

Sering kali, penempatan kamera dalam sebuah proyek tidak ideal. Seringkali, saya harus memasang kamera secara terbalik, atau saya perlu membalik gambar karena satu dan lain alasan. Untungnya, OpenCV membuatnya sangat sederhana dengan metode flip(). Metode flip() mengambil tiga parameter: gambar yang akan dibalik, kode yang menunjukkan cara membaliknya, dan tujuan dari gambar yang dibalik. Parameter terakhir hanya digunakan jika Anda ingin menetapkan gambar yang dibalik ke variabel lain, tetapi Anda dapat membalik gambar di tempatnya. Sebuah gambar dapat dibalik secara horizontal, vertikal, atau keduanya dengan menyediakan flipCode. FlipCode adalah positif, negatif, atau nol. Nol membalik gambar secara horizontal, nilai positif membaliknya secara vertikal, dan angka negatif membaliknya di kedua sumbu. Lebih sering daripada tidak, Anda akan membalik gambar pada kedua sumbu untuk memutarnya secara efektif 180 derajat.

Mari kita gunakan gambar tiga bola yang kita gunakan sebelumnya untuk mengilustrasikan membalik bingkai.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai flip_image.py.
3. Masukkan kode berikut:

```
import cv2

img = cv2.imread('color_balls_small.jpg')
h_img = cv2.flip(img, 0)
v_img = cv2.flip(img, 1)
b_img = cv2.flip(img, -1)

cv2.imshow('image', img)
```

```
cv2.imshow('horizontal', h_img)cv2.imshow('vertical',
v_img)
cv2.imshow('both', b_img)
cv2.waitKey(0)
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan python flip_image.py dan tekan Return.

Empat jendela terbuka, masing-masing dengan versi file gambar yang berbeda. Tekan tombol apa saja untuk keluar.

Mengubah ukuran

Anda dapat mengubah ukuran gambar. Ini berguna untuk mengurangi sumber daya yang dibutuhkan untuk memproses gambar. Semakin besar gambar, semakin banyak memori dan sumber daya CPU yang dibutuhkan. Untuk mengubah ukuran gambar, kami menggunakan metode resize(). Parameternya adalah gambar yang Anda skala, dimensi yang diinginkan sebagai tupel, dan interpolasi. Interpolasi adalah metode matematika yang digunakan untuk menentukan bagaimana menangani penghapusan atau penambahan piksel. Ingat, ketika bekerja dengan gambar, Anda benar-benar bekerja dengan array multidimensi yang berisi informasi untuk setiap titik, atau piksel, yang membentuk gambar. Saat Anda mengurangi gambar, Anda menghapus piksel. Saat Anda memperbesar gambar, Anda menambahkan piksel. Interpolasi adalah metode dimana ini terjadi. Ada tiga opsi interpolasi. INTER_AREA paling baik digunakan untuk reduksi. INTER_CUBIC dan INTER_LINEAR keduanya bagus untuk memperbesar gambar, dengan INTER_LINEAR lebih cepat dari keduanya. Jika interpolasi tidak disediakan, OpenCV menggunakan INTER_LINEAR sebagai default untuk memperkecil dan memperbesar.

Gambar ketiga bola saat ini berukuran 800x533 piksel. Meskipun ukurannya tidak besar, kami akan membuatnya sedikit lebih kecil. Mari kita buat setengah dari ukuran saat ini untuk kedua sumbu. Untuk melakukan ini, kita akan menggunakan interpolasi INTER_AREA.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai resize_image.py.
3. Masukkan kode berikut:

```
import cv2
img = cv2.imread('color_balls_small.jpg')x,y = img.shape[:2]
resImg = cv2.resize(img, (y/2, x/2), interpolation =cv2.INTER_AREA)
cv2.imshow('image', img) cv2.imshow('resized',resImg)
cv2.waitKey(0)
```

4. Simpan file.
5. Buka jendela terminal.

6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan **python resize_image.py** dan tekan Return.

Dua jendela seharusnya terbuka. Yang pertama memiliki gambar asli. Yang kedua menampilkan gambar yang diperkecil. Tekan sembarang tombol untuk menutup jendela.

Bekerja dengan Warna

Warna jelas merupakan bagian yang sangat penting dalam bekerja dengan gambar. Dengan demikian, ini adalah bagian yang sangat menonjol dari OpenCV. Ada banyak hal yang bisa dilakukan dengan warna. Kami akan fokus pada beberapa elemen kunci yang diperlukan untuk mencapai tujuan akhir kami yaitu mengidentifikasi dan mengejar bola dengan robot.

Ruang Warna

Salah satu elemen kunci dalam bekerja dengan warna adalah ruang warna, yang menjelaskan bagaimana OpenCV mengekspresikan warna. Dalam OpenCV, warna diwakili oleh serangkaian angka. Ruang warna menentukan arti dari angka-angka tersebut.

Ruang warna default untuk OpenCV adalah BGR. Ini berarti setiap warna dijelaskan oleh tiga bilangan bulat antara 0 dan 255, yang sesuai dengan tiga saluran warna biru, hijau, dan merah, dalam urutan itu. Warna yang dinyatakan sebagai (255,0,0) memiliki nilai maksimum di saluran biru, dan hijau dan merah adalah nol. Ini mewakili biru murni. Mengingat ini, (0,255,0) berwarna hijau dan (0,0,255) berwarna merah. Nilai (0,0,0) mewakili hitam, tidak adanya warna apa pun, dan (255.255.255) putih.

Jika Anda pernah bekerja dengan grafik di masa lalu, BGR adalah kebalikan dari yang biasanya Anda gunakan. Sebagian besar grafik digital dijelaskan dalam istilah RGB merah, hijau, dan biru. Jadi, ini mungkin perlu sedikit membiasakan diri. Ada banyak ruang warna. Yang kami pedulikan adalah BGR, RGB, HSV, dan skala abu-abu. Kami telah membahas ruang warna default, BGR, dan ruang warna RGB umum. HSV adalah hue, saturation, dan value. Hue mewakili warna pada skala 0 hingga 180. Saturasi menunjukkan seberapa jauh putih warna dari 0 hingga 255. Nilai adalah ukuran seberapa jauh dari hitam warna dari 0 hingga 255. Jika saturasi dan nilainya adalah 0, warnanya abu-abu. Saturasi dan nilai 255 adalah versi rona paling terang.

Hue sedikit lebih rumit. Itu pada skala 0 hingga 180, di mana 0 dan 180 keduanya merah. Di sinilah pentingnya mengingat roda warna. Jika 0 dan 180 bertemu di bagian atas roda di tengah ruang merah, saat Anda bergerak searah jarum jam di sekitar roda, hue = 30 adalah kuning, hue = 60 adalah hijau, hue = 90 adalah teal, hue = 120 adalah biru , hue = 150 adalah ungu, dan hue = 180 membawa kita kembali ke merah. Salah satu yang paling sering Anda temui adalah skala abu-abu. Skala abu-abu persis seperti yang terdengar: versi hitam-putih dari sebuah gambar. Ini digunakan oleh algoritma deteksi fitur untuk membuat topeng. Kami menggunakannya saat kami memfilter objek. Untuk mengonversi gambar ke ruang warna yang berbeda, Anda menggunakan metode cvtColor. Dibutuhkan dua parameter: gambar dan konstanta ruang warna. Konstanta ruang warna dibangun ke dalam OpenCV. Mereka adalah COLOR_ BGR2RGB, COLOR_BGR2HSV, dan COLOR_BGR2GRAY. Apakah Anda

melihat pola di sana? Jika Anda ingin mengonversi dari ruang warna RGB ke ruang warna HSV, konstanta akan menjadi COLOR_RGB2HSV.

Mari kita ubah gambar kita dari tiga bola berwarna menjadi gambar skala abu-abu.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai gray_image.py.
3. Masukkan kode berikut:

```
import cv2
img = cv2.imread('color_balls_small.jpg')
grayImg = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)

cv2.imshow('img', img)
cv2.imshow('gray', grayImg)
cv2.waitKey(0)
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan python gray_image.py dan tekan Return.

Ini akan membuka dua jendela: satu dengan gambar berwarna asli dan satu lagi dengan versi skala abu-abu. Klik sembarang tombol untuk keluar dari program dan menutup jendela.

Filter Warna

Memfilter warna membutuhkan kode yang sangat sedikit, tetapi pada saat yang sama, ini bisa sedikit membuat frustrasi karena Anda biasanya tidak mencari warna tertentu tetapi rentang warna. Warna sangat jarang murni dan satu nilai. Inilah sebabnya mengapa kami ingin dapat beralih di antara ruang warna. Tentu, kita bisa mencari warna merah di BGR. Tetapi untuk melakukan itu, kita memerlukan rentang spesifik untuk masing-masing dari tiga nilai. Dan di mana itu akan berlaku dengan semua ruang warna, biasanya lebih mudah untuk memanggil rentang yang Anda butuhkan di ruang HSV.

Strategi yang digunakan untuk memfilter warna tertentu cukup mudah, tetapi ada beberapa langkah yang terlibat dan beberapa hal yang perlu diingat saat Anda melakukannya. Pertama, kita akan membuat salinan gambar di ruang warna HSV. Kemudian kami menerapkan rentang filter kami dan menjadikannya gambarnya sendiri. Untuk ini, kami menggunakan metode inRange(). Dibutuhkan tiga parameter: gambar tempat kita menerapkan filter, rentang bawah, dan rentang nilai atas. Metode inRange memindai semua piksel dalam gambar yang disediakan untuk menentukan apakah mereka berada dalam rentang yang ditentukan. Ini mengembalikan true, atau 1, jika demikian; jika tidak, ia kembali 0. Yang tersisa dari kita adalah gambar hitam-putih yang bisa kita gunakan sebagai topeng.

Selanjutnya, kita menerapkan mask menggunakan metode bitwise_and(). Metode ini

mengambil dua gambar dan mengembalikan area di mana piksel cocok. Karena bukan itu yang kami cari, kami perlu melakukan sedikit tipu daya. Untuk tujuan kami, bitwise_and membutuhkan tiga parameter: gambar 1, gambar 2, dan topeng. Karena kami ingin mengembalikan semua yang diungkapkan oleh topeng, gambar 1 dan gambar 2 keduanya menggunakan gambar asli kami. Kemudian kami menerapkan topeng kami dengan menetapkan parameter topeng. Karena kita meninggalkan beberapa parameter opsional, kita perlu menetapkan parameter mask secara eksplisit, seperti ini: mask = mask_image. Hasilnya adalah gambar yang hanya menunjukkan warna yang kita filter.

Cara termudah untuk menunjukkan ini adalah dengan berjalan melewatinya. Kode ini sebenarnya cukup sederhana setelah Anda tahu apa yang terjadi.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai blue_filter.py.
3. Masukkan kode berikut:

```
import cv2
img = cv2.imread("color_balls_small.jpg")
imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
        lower_blue = np.array([80,120,120])
        upper_blue = np.array([130,255,255])
        blueMask = cv2.inRange(imgHSV,lower_blue,upper_blue)
        res = cv2.bitwise_and(img, img, mask=blueMask)
        cv2.imshow('img', img) cv2.imshow('mask',
        blueMask)cv2.imshow('blue', res)
cv2.waitKey(0)
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan python blue_filter.pydan tekan Return.

Tiga jendela terbuka dengan versi berbeda dari gambar kita. Yang pertama adalah gambar biasa. Yang kedua adalah gambar hitam-putih yang bertindak sebagai topeng kita. Yang ketiga adalah gambar topeng terakhir. Hanya piksel di bawah area putih topeng yang ditampilkan. Mari luangkan waktu sejenak untuk menelusuri kode untuk memperjelas apa yang kita lakukan dan mengapa. Kami mulai seperti kami melakukan semua skrip kami, dengan mengimpor OpenCV dan NumPy, dan kemudian memuat gambar.

```
import cv2
import numpy as np
img = cv2.imread("color_balls_small.jpg")
```

Selanjutnya, kami membuat salinan gambar dan mengubahnya menjadi ruang warna HSV.

```
imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

Setelah berada di ruang warna HSV, lebih mudah untuk menyaring bola biru. Seperti

yang saya diskusikan, kita tahu biru murni memiliki nilai rona 120. Karena tidak mungkin objek yang kita filter berwarna biru murni, kita perlu memberinya rentang warna. Dalam hal ini, kami mencari semuanya dari 80, yang merupakan setengah jalan antara hijau dan biru, dan 130. Kami juga ingin memfilter warna yang tidak hampir putih atau hampir hitam, jadi kami menggunakan nilai 120 dan 255 untuk saturasi dan rentang nilai. Untuk memastikan bahwa rentang filter dalam format yang dipahami OpenCV, kami membuatnya sebagai array NumPy.

```
lower_blue = np.array([80,120,120])
upper_blue = np.array([130,255,255])
```

Dengan rentang filter yang ditentukan, kita dapat menggunakannya dengan metode inRange() untuk menentukan apakah piksel dalam versi HSV gambar kita berada dalam rentang biru yang kita cari. Ini membuat gambar topeng untuk mengecualikan semua piksel non-biru.

```
blueMask = cv2.inRange(imgHSV,lower_blue,upper_blue)
```

Selanjutnya, kita menggunakan bitwise_and() untuk menerapkan topeng kita. Karena kami ingin mengembalikan semua piksel dari gambar kami di dalam topeng kami, kami meneruskan gambar asli sebagai gambar 1 dan gambar 2. Ini membandingkan gambar dengan dirinya sendiri dan mengembalikan seluruh gambar, karena setiap piksel dalam gambar cocok dengan dirinya sendiri.

```
res = cv2.bitwise_and(img, img, mask=blueMask)
```

Terakhir, kami menampilkan gambar asli, topeng, dan gambar yang difilter. Kemudian kita menunggu tombol ditekan sebelum kita menutup jendela dan keluar dari program.

```
cv2.imshow('img', img)
cv2.imshow('mask', blueMask)
cv2.imshow('blue', res)
cv2.waitKey(0)
```

Seperti yang Anda lihat, setelah Anda mengetahui cara kerjanya, memfilter warna menjadi sangat mudah. Ini menjadi sedikit lebih rumit ketika Anda memfilter warna merah. Merah muncul di ujung rendah dan tinggi spektrum rona, jadi Anda harus membuat dua filter dan menggabungkan topeng yang dihasilkan. Ini dapat dengan mudah dilakukan dengan metode add() OpenCV, dan terlihat seperti ini:

```
combinedMask = cv2.add(redMask1, redMask2)
```

Pada akhirnya, Anda hanya memiliki gambar dengan piksel yang Anda cari. Untuk mata manusia, mudah dikenali sebagai kelompok terkait. Untuk komputer, tidak demikian. Secara asli, komputer tidak mengenali perbedaan antara piksel hitam dan biru. Di situlah deteksi gumpalan berperan.

9.10 GUMPALAN DAN DETEKSI GUMPALAN

Blob adalah kumpulan piksel yang serupa. Mereka bisa apa saja dari lingkaran monoton ke gambar jpeg. Untuk komputer, piksel adalah piksel, dan tidak dapat membedakan antara gambar bola dan gambar pesawat. Inilah yang membuat visi komputer begitu

menantang. Kami telah mengembangkan banyak teknik berbeda untuk mencoba mengekstrapolasi informasi tentang suatu gambar; masing-masing memiliki *trade-off* dalam hal kecepatan dan akurasi. Sebagian besar teknik menggunakan proses yang disebut ekstraksi fitur, yang merupakan istilah umum untuk kumpulan algoritme yang mengkatalogkan fitur luar biasa dalam suatu gambar, seperti garis, tepi, area warna yang luas, dan sebagainya. Setelah fitur ini diekstraksi, fitur tersebut dapat dianalisis atau dibandingkan dengan fitur lain untuk menentukan citra. Beginilah cara kerja fungsi seperti deteksi wajah dan deteksi gerakan.

Kita akan menggunakan metode yang lebih sederhana untuk melacak objek. Daripada mengekstrak fitur detail dan menganalisisnya, kita akan menggunakan teknik penyaringan warna dari bagian sebelumnya untuk mengidentifikasi area warna yang luas. Kami kemudian akan menggunakan fungsi bawaan untuk mengumpulkan informasi tentang grup piksel. Teknik yang lebih sederhana ini disebut deteksi gumpalan.

Menemukan Gumpalan

OpenCV membuat deteksi gumpalan cukup mudah, terutama setelah kami melakukan tugas berat menyaring semua yang tidak kami inginkan. Setelah gambar difilter, kita dapat menggunakan topeng untuk deteksi gumpalan bersih. Kelas SimpleBlobDetector dari OpenCV mengidentifikasi lokasi dan ukuran gumpalan. Kelas SimpleBlobDetector tidak sesederhana yang Anda kira. Ada sejumlah parameter bawaan yang perlu diaktifkan atau dinonaktifkan. Jika diaktifkan, Anda perlu memastikan bahwa nilainya berfungsi untuk aplikasi Anda.

Metode untuk mengatur parameter adalah SimpleBlobDetector_ Params().Metode untuk membuat detektor adalah SimpleBlobDetector_create().Anda meneruskan parameter ke metode create untuk memastikan semuanya diatur dengan benar. Setelah parameter ditetapkan dan detektor dibuat dengan benar, Anda menggunakan metode detect() untuk mengidentifikasi titik kunci. Dalam kasus detektor gumpalan sederhana, titik kunci mewakili pusat dan ukuran gumpalan yang terdeteksi.

Terakhir, kita menggunakan metode drawKeyPoints() untuk menggambar lingkaran di sekitar gumpalan kita. Secara default, ini menggambar lingkaran kecil di tengah gumpalan. Namun, sebuah bendera dapat dilewati yang menyebabkan ukuran lingkaran relatif terhadap ukuran gumpalan. Mari kita berjalan melalui sebuah contoh. Kami akan menggunakan kode filter dari latihan sebelumnya dan menambahkan deteksi gumpalan. Dalam latihan ini, kami memfilter bola biru di gambar kami. Kemudian kami menggunakan topeng untuk menemukan pusat bola dan menggambar lingkaran di sekitarnya.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai simple_blob_detect.py.
3. Masukkan kode berikut:

```
import cv2
import numpy as np
img = cv2.imread("color_balls_small.jpg")
imgHSV = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
```

```
# setup parameters
params = cv2.SimpleBlobDetector_Params()
params.filterByColor = False
params.filterByArea = False
params.filterByInertia = False
params.filterByConvexity = False
params.filterByCircularity = False

# create blob detector
det = cv2.SimpleBlobDetector_create(params)

lower_blue = np.array([80,120,120])
upper_blue = np.array([130,255,255])

blueMask = cv2.inRange(imgHSV,lower_blue,upper_blue)
res = cv2.bitwise_and(img, img, mask=blueMask)

# get keypoints
keypnts = det.detect(blueMask)

# draw keypoints
cv2.drawKeypoints(img, keypnts, img, (0,0,255),
        cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

cv2.imshow('img', img)
cv2.imshow('mask', blueMask)
cv2.imshow('blue', res)

# print the coordinates and size of keypoints to terminal
for k in keypnts:
        print k.pt[0]
        print k.pt[1]
        print k.size
cv2.waitKey(0)
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan python simple_blob_detect.py dan tekan Return.

Ini akan membuka tiga versi gambar. Namun, gambar aslinya sekarang memiliki lingkaran

merah yang digambar di sekitar bola biru. Di jendela terminal, kami telah mencetak koordinat pusat bola serta ukurannya. Bagian tengah bola digunakan nanti dalam bab ini ketika kita mulai melacak bola.

Parameter

Kelas SimpleBlobDetector membutuhkan beberapa parameter agar berfungsi dengan baik. Sangat disarankan agar semua opsi filter diaktifkan atau dinonaktifkan secara eksplisit dengan menyetel parameter yang sesuai ke True atau False. Jika sebuah filter diaktifkan, Anda juga perlu mengatur parameternya. Parameter default dikonfigurasi untuk mengekstrak gumpalan lingkaran gelap. Pada latihan sebelumnya, kami hanya menonaktifkan semua filter. Karena kami bekerja dengan gambar bola yang difilter, dan kami hanya memiliki satu gumpalan di gambar, kami tidak perlu menambahkan filter lain. Padahal Anda secara teknis dapat menggunakan parameter SimpleBlobDetector saja tanpa menutupi segala sesuatu yang lain, ini bisa menjadi sedikit lebih menantang dalam memanggil semua parameter untuk mendapatkan hasil yang kita inginkan. Juga, metode yang kami gunakan memungkinkan Anda sedikit lebih banyak wawasan tentang apa yang dilakukan OpenCV di latar belakang.

Penting untuk memahami cara kerja SimpleBlobDetector untuk mendapatkan gambaran yang lebih baik tentang cara filter digunakan. Ada beberapa parameter yang dapat digunakan untuk menyempurnakan hasil. Hal pertama yang terjadi adalah gambar diubah menjadi beberapa gambar biner dengan menerapkan ambang batas. minThreshold dan maxThreshold menentukan rentang keseluruhan, sedangkan thresholdStep menentukan jarak antara ambang batas. Setiap citra biner kemudian diproses untuk kontur menggunakan findContours(). Ini memungkinkan sistem untuk menghitung pusat setiap gumpalan. Dengan diketahui pusatnya, beberapa blob digabungkan menjadi satu grup menggunakan parameter minDistanceBetweenBlobs. Pusat grup dikembalikan sebagai titik kunci, seperti diameter keseluruhan grup. Parameter untuk setiap filter dihitung dan filter diterapkan.

9.11 FILTER

Berikut ini daftar filter dan parameter terkaitnya.

filterByColor

Filter ini untuk intensitas relatif dari setiap citra biner. Ini mengukur nilai intensitas di tengah gumpalan dan membandingkannya dengan parameter, blobColor. Jika mereka tidak cocok, gumpalan tidak memenuhi syarat. Intensitas diukur dari 0 hingga 255; 0 gelap dan 255 terang.

filterByArea

Ketika gumpalan individu dikelompokkan, area keseluruhannya dihitung. Filter ini mencari gumpalan antara minArea dan maxArea.

filterByCircularity

Lingkaran dihitung dengan rumus

$$\frac{4 * \pi * Area}{perimeter * perimeter}$$

Ini mengembalikan rasio antara 0 dan 1, yang dibandingkan dengan minCircularity dan maxCircularity. Jika nilainya berada di antara parameter ini, gumpalan disertakan dalam hasil.

filterByInertia

Inersia adalah perkiraan seberapa memanjang gumpalan itu. Ini adalah rasio antara 0 dan 1. Jika nilainya antara minInertiaRatio dan maxInertiaRatio, gumpalan dikembalikan dalam hasil keypoint.

filterByConvexity

Cembung adalah rasio dengan nilai antara 0 dan 1. Ini mengukur rasio antara kurva cembung dan cekung dalam gumpalan. Parameter konveksitas adalah minConvexity dan maxConvexity.

Pelacakan gumpalan Kita melihat di bagian sebelumnya bahwa koordinat x dan y dari pusat blob dikembalikan sebagai bagian dari keypoint, yang digunakan untuk melacak blob. Untuk melacak blob, Anda perlu menggunakan streaming video langsung dari kamera robot, lalu menentukan arti pelacakan untuk proyek Anda. Bentuk pelacakan yang paling sederhana adalah dengan memindahkan lingkaran yang dihasilkan dengan gumpalan.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai blob_tracker.py.
3. Masukkan kode berikut:

```
import cv2
import numpy as np

cap = cv2.VideoCapture(0)

# setup detector and parameters
params = cv2.SimpleBlobDetector_Params()
params.filterByColor = Falseparams.filterByArea =
True
params.minArea = 20000
params.maxArea = 30000
params.filterByInertia = False params.filterByConvexity =
False params.filterByCircularity = True
params.minCircularity = 0.5
params.maxCircularity = 1

det = cv2.SimpleBlobDetector_create(params)

# define blue
lower_blue = np.array([80,60,20])
upper_blue = np.array([130,255,255])
```

```
while True:
    ret, frame = cap.read()
    imgHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    blueMask = cv2.inRange(imgHSV,lower_blue,upper_blue)
    blur= cv2.blur(blueMask, (10,10))

    res = cv2.bitwise_and(frame, frame, mask=blueMask)

    # get and draw keypoint
    keypnts = det.detect(blur)

    cv2.drawKeypoints(frame, keypnts, frame, (0,0,255),
                cv2.DRAW_MATCHES_FLAGS_DRAW_RICH_KEYPOINTS)

      cv2.imshow('frame', frame)
      cv2.imshow('mask', blur)

      for k in keypnts:
          print k.size

      if cv2.waitKey(1) & 0xff == ord('q'):
      break

cap.release()
cv2.destroyAllWindows()
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan **sudo python blob_tracker.py** dan tekan Return.

Dua jendela terbuka: satu menampilkan topeng yang digunakan untuk memfilter warna dan satu lagi dengan aliran video. Sebuah lingkaran harus digambar di sekitar gumpalan. Saya mengaktifkan filterByArea dan filterByCircularity untuk memastikan bahwa saya hanya mendapatkan bola. Anda mungkin perlu melakukan penyesuaian pada parameter detektor untuk menyempurnakan filter Anda.

Bot Mengejar Bola

Anda sekarang tahu cara melacak gumpalan dengan webcam yang terpasang di robot. Di Bab 8, Anda mempelajari tentang algoritma untuk mengikuti garis yang disebut pengontrol PID. Apa yang terjadi ketika kita menggabungkan pengontrol PID dengan program pelacakan bola kita? Selanjutnya, mari kita programkan robot kecil itu untuk mengejar bola biru yang

sedang dilacaknya itu. Untuk melakukan ini, Anda akan menggunakan apa yang baru saja Anda pelajari tentang pelacakan gumpalan dan apa yang Anda pelajari di Bab 8.

Kontroler PID mengharapkan masukan berupa penyimpangan dari hasil yang diinginkan. Jadi, kita harus mulai dengan menentukan hasil yang diinginkan. Dalam hal ini, tujuannya hanya untuk menjaga bola di tengah bingkai. Jadi nilai kesalahan kita akan menjadi varians dari pusat, yang juga berarti kita perlu mendefinisikan pusat frame. Setelah pusat ditentukan, deviasi adalah masalah mengurangkan lokasi x bola dari lokasi x pusat. Kami juga akan mengurangi lokasi y bola dari lokasi y di tengah.

Sekarang kita dapat menggunakan dua pengontrol PID untuk menjaga agar bola tetap berada di tengah bingkai. Kontroler pertama mengarahkan robot. Saat bola bergerak pada sumbu x, penyimpangannya negatif atau positif. Jika positif, belok ke kiri. Jika negatif, belok ke kanan. Dengan cara yang sama, kita dapat menggunakan sumbu y untuk mengontrol kecepatan robot. Varians y positif mendorong robot ke depan, sedangkan varians negatif mendorongnya mundur.

1. Buka IDLE IDE dan buat file baru.
2. Simpan file sebagai ball_chaser.py.
3. Masukkan kode berikut:

```
import cv2
import numpy as np
import time

from Adafruit_MotorHAT import Adafruit_MotorHAT as amhat
from Adafruit_MotorHAT import Adafruit_DCMotor as adamo

# create motor objects
motHAT = amhat(addr=0x60)
mot1 = motHAT.getMotor(1)
mot2 = motHAT.getMotor(2)
mot3 = motHAT.getMotor(3)
mot4 = motHAT.getMotor(4)

motors = [mot1, mot2, mot3, mot4]

# motor multipliers
motorMultiplier = [1.0, 1.0, 1.0, 1.0, 1.0]

# motor speeds
motorSpeed = [0,0,0,0]

# speeds
```

```
speedDef = 100
leftSpeed = speedDef
rightSpeed = speedDef
diff= 0
maxDiff = 50
turnTime = 0.5

# create camera object
cap = cv2.VideoCapture(0)
time.sleep(1)

# PID
kp = 1.0
ki = 1.0
kd = 1.0
ballX = 0.0
ballY = 0.0

x = {'axis':'X',
        'lastTime':int(round(time.time()*1000)),
        'lastError':0.0,
        'error':0.0,
        'duration':0.0,
        'sumError':0.0,
        'dError':0.0,
        'PID':0.0}
y = {'axis':'Y',
        'lastTime':int(round(time.time()*1000)),
        'lastError':0.0,
        'error':0.0,
        'duration':0.0,
        'sumError':0.0,
        'dError':0.0,
        'PID':0.0}

# setup detector
params = cv2.SimpleBlobDetector_Params()

# define detector parametersparams.filterByColor =
Falseparams.filterByArea = True
```

```
params.minArea = 15000
params.maxArea = 40000
params.filterByInertia = False params.filterByConvexity =
False params.filterByCircularity = True
params.minCircularity = 0.5
params.maxCircularity = 1

# create blob detector object
det = cv2.SimpleBlobDetector_create(params)

# define blue
lower_blue = np.array([80,60,20])
upper_blue = np.array([130,255,255])

def driveMotors(leftChnl=speedDef, rightChnl=speedDef,
duration = defTime):

    # determine the speed of each motor by multiplying
    # the channel by the motors multiplier
    motorSpeed[0] = leftChnl * motorMultiplier[0]
    motorSpeed[1] = leftChnl * motorMultiplier[1]
    motorSpeed[2] = rightChnl * motorMultiplier[2]
    motorSpeed[3] = rightChnl * motorMultiplier[3]

    # set each motor speed. Since the speed can be a# negative number, we
    take the absolute value
    for x in range(4):
        motors[x].setSpeed(abs(int(motorSpeed[x])))

    # run the motors. if the channel is negative, run
    # reverse. else run forward
    if(leftChnl < 0):
        motors[0].run(amhat.BACKWARD)
        motors[1].run(amhat.BACKWARD)
    else:
        motors[0].run(amhat.FORWARD)
        motors[1].run(amhat.FORWARD)
    if (rightChnl > 0): motors[2].run(amhat.BACKWARD)
        motors[3].run(amhat.BACKWARD)
    else:
```

```
        motors[2].run(amhat.FORWARD)
        motors[3].run(amhat.FORWARD)

        def PID(axis):
            lastTime = axis['lastTime']
            lastError = axis['lastError']

            # get the current time
            now = int(round(time.time()*1000))
            duration = now-lastTime

            # calculate the error
            axis['sumError'] += axis['error'] * duration
            axis['dError'] = (axis['error'] - lastError)/duration

            # prevent runaway values
            if axis['sumError'] > 1:axis['sumError'] = 1
            if axis['sumError'] < -1: axis['sumError'] = -1

            # calculate PID
            axis['PID'] = kp * axis['error'] + ki *axis['sumError'] + kd *
            axis['dError']

            # update variables
            axis['lastError'] = axis['error']axis['lastTime'] = now

            # return the output valuereturn axis
def killMotors():
mot1.run(amhat.RELEASE)
mot2.run(amhat.RELEASE)
mot3.run(amhat.RELEASE)
mot4.run(amhat.RELEASE)

# main program
try:
    while True:
        # capture video frame ret,
        frame = cap.read()

        # calculate center of frame
```

```
    height, width, chan = np.shape(frame)
    xMid = width/2 * 1.0
    yMid = height/2 * 1.0

    # filter image for blue ball
    imgHSV = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)

    blueMask = cv2.inRange(imgHSV, lower_blue,
    upper_blue)
    blur = cv2.blur(blueMask, (10,10))

    res = cv2.bitwise_and(frame,frame,mask=blur)

    # get keypoints
    keypoints = det.detect(blur)
    try:
        ballX = int(keypoints[0].pt[0])
        ballY = int(keypoints[0].pt[1])
    except:
        pass

    # draw keypoints
    cv2.drawKeypoints(frame, keypoints, frame,(0,0,255),
            cv2.DRAW_MATCHES_FLAGS_DRAW
            _RICH_KEYPOINTS)

# calculate error and get PID ratio
xVariance = (ballX - xMid) / xMid
yVariance = (yMid - ballY) / yMid

x['error'] = xVariance/xMid
y['error'] = yVariance/yMid

x = PID(x)
y = PID(y)

# calculate left and right speeds
leftSpeed = (speedDef * y['PID']) + (maxDiff *x['PID'])
rightSpeed = (speedDef * y['PID']) - (maxDiff *x['PID'])
```

```
                # another safety check for runaway values
                if leftSpeed > (speedDef + maxDiff): leftSpeed
                = (speedDef + maxDiff)
                if leftSpeed < -(speedDef + maxDiff): leftSpeed
                = -(speedDef + maxDiff)
                if rightSpeed > (speedDef + maxDiff):
                rightSpeed = (speedDef + maxDiff)
                if rightSpeed < -(speedDef + maxDiff):
                rightSpeed = -(speedDef + maxDiff)

                # drive motors
                driveMotors(leftSpeed, rightSpeed, driveTime)
                # show frame
        ##              cv2.imshow('frame', frame)
        ##              cv2.waitKey(1)

except KeyboardInterrupt:
        killMotors()
        cap.release()
        cv2.destroyAllWindows()
```

4. Simpan file.
5. Buka jendela terminal.
6. Arahkan ke folder tempat Anda menyimpan file.
7. Masukkan sudo python ball_chaser.py dan tekan Return.

Setelah beberapa detik, robot Anda akan mulai bergerak maju. Jika ada bola biru di dalam bingkai, bola itu harus berbelok ke arahnya. Robot berusaha menjaga bola tetap berada di tengah bingkai.

Beberapa hal dalam kode ini sedikit berbeda dari cara kami melakukan sesuatu di masa lalu. Terutama, kami menempatkan nilai untuk sumbu x dan y ke dalam kamus. Kami melakukan ini untuk menjaga nilai tetap bersama ketika kami meneruskannya ke pengontrol PID, yang merupakan perubahan lain yang dibuat. Fungsi PID telah diperbarui untuk menerima satu parameter. Namun, parameter yang diharapkan adalah kamus. Itu ditugaskan ke variabel sumbu dalam fungsi. Semua referensi variabel kemudian diperbarui untuk menggunakan kamus. Hasilnya diperbarui dalam kamus sumbu, dan kemudian ditetapkan ke kamus yang sesuai di program utama.

Saya juga memastikan untuk menghilangkan penundaan yang akan mempengaruhi putaran utama atau kecepatan refresh kamera. Karena seluruh program ini berjalan dalam satu proses, tidak secepat jika kita memecah proses menjadi utas yang berbeda. Dengan

demikian, robot mungkin kehilangan bola dan mengembara.

9.12 RINGKASAN

Dalam bab ini, kami mulai memanfaatkan beberapa kemampuan menarik yang ditawarkan Raspberry Pi. Visi komputer memungkinkan kita untuk melakukan tugas yang jauh lebih kompleks daripada yang dapat kita lakukan dengan mikrokontroler saja. Untuk mempersiapkan bekerja dengan penglihatan, kami memasang webcam dasar pada robot. Ini mengambil pertimbangan khusus karena webcam ini tidak dirancang untuk dipasang. Tentu saja, solusi Anda mungkin berbeda dari milik saya, sehingga Anda dapat melatih kreativitas dalam memasang kamera. Setelah itu, kami siap untuk menginstal OpenCV.

OpenCV adalah platform visi komputer yang dikembangkan oleh komunitas open source yang membuat banyak fungsi visi menjadi sangat sederhana. Menginstal software di Raspberry Pi memakan waktu cukup lama, terutama karena kita harus mengkompilasinya dari kode sumber, dan meskipun kemampuannya mengesankan, Raspberry Pi tidak memiliki kekuatan pemrosesan laptop atau PC, sehingga membutuhkan waktu yang lama. sementara untuk mengkompilasi kode. Tetapi setelah dikompilasi dan diinstal, kami dapat melakukan beberapa hal menyenangkan. Kami mengerjakan beberapa latihan menggunakan gambar diam. Ini memungkinkan kami untuk mempelajari beberapa dasar OpenCV tanpa biaya pemrosesan video. Setelah kami mempelajari beberapa dasar-dasarnya, kami belajar untuk menarik video langsung dari kamera dan menerapkan pelajaran yang kami pelajari menggunakan gambar diam. Dengan menggunakan teknik penyaringan warna dan pelacakan gumpalan yang kami ambil di bab ini, kami memberi robot kami kemampuan untuk melihat dan mengikuti bola.

BAB 10
KESIMPULAN

Anda telah menempuh perjalanan jauh sejak Bab 1. Jika Anda baru mengenal robotika dan pemrograman, maka ini mungkin buku yang menantang. Itu dimaksudkan untuk menjadi, jadi selamat karena berhasil melewatinya. Semoga, Anda mengikuti dan membangun robot Anda sendiri dalam prosesnya. Sebagai rangkuman, di Bab 1, saya memperkenalkan beberapa konsep dasar robot dan membahas tujuan dari buku ini. Di Bab 2, kami mulai bekerja dengan Raspberry Pi dengan menginstal sistem operasi Raspbian dan mengonfigurasinya untuk akses jarak jauh. Bab 3 memperkenalkan Anda pada bahasa pemrograman Python. Di Bab 4, kami mulai bekerja dengan sensor menggunakan header GPIO Raspberry Pi. Dalam prosesnya, kami belajar sedikit tentang pemrosesan digital dan mendiskusikan beberapa keterbatasan.

Solusi untuk keterbatasan Pi diperkenalkan di Bab 5 ketika saya memberi Anda Arduino. Kami belajar bagaimana memprogram Arduino dan bagaimana mengirimkan data bolak-balik antara itu dan Pi. Di Bab 6, kami merakit HAT Motor dan mempelajari cara mengemudikan motor dengannya dan dengan pengontrol motor generik. Di Bab 7, kami akhirnya merakit robot. Di Bab 8, kami memasang sensor IR dan memprogram robot untuk mengikuti garis. Dan di Bab 9, kami mengeluarkan kekuatan Raspberry Pi untuk menggunakan visi komputer untuk menyaring warna dan melacak bola.

10.1 JENIS ROBOTIKA

Seperti yang saya bahas di Bab 1, robotika dapat berarti banyak hal yang berbeda. Itu benar-benar tergantung pada bagaimana Anda ingin mendefinisikannya. Untuk membantu mengaburkan definisi lebih jauh, banyak teknologi yang digunakan dalam robotika adalah mudah ditransfer ke Internet of Things (IoT). Perangkat keras, perangkat lunak, sensor, saluran komunikasi, dan sebagainya, sama di rumah otomatis Anda seperti di robot Anda. Pemrogramannya serupa dan hasilnya biasanya mempengaruhi dunia fisik. Jadi, intinya, IoT mengubah rumah, kantor, atau pabrik Anda menjadi robot.

Karena definisi yang luas ini, minat Anda pada robotika mungkin sangat berbeda dari minat saya. Misalnya, apakah Anda tertarik dengan bot meja kecil atau robot yang lebih besar? Apakah Anda terutama tertarik pada robot terestrial yang mengemudi di tanah, atau apakah Anda ingin peralatan otomatis Anda terbang? Atau, mungkin Anda tertarik untuk menjelajahi kedalaman laut dengan kapal selam robotik. Apakah Anda ingin bereksperimen dengan mobil otonom atau otomatisasi rumah dan IoT adalah pilihan Anda?

Mengetahui bidang yang kemungkinan akan Anda kejar menentukan alat yang Anda gunakan. Jika Anda membuat robot meja kecil, Anda mungkin tidak memerlukan tukang las. Bidang juga menentukan beberapa alat desain yang akan Anda gunakan. Misalnya, sebagian besar robot kecil seperti yang kami buat dalam buku ini dapat dirancang dengan cepat atau dengan pena dan kertas. Namun, jika Anda sedang membangun sesuatu yang lebih kompleks,

seperti hewan berkaki empat, Anda mungkin memerlukan perangkat lunak CAD.

10.2 ALAT

Alat datang dalam dua rasa dalam robotika: perangkat keras dan perangkat lunak. Saya tidak membahas alat perangkat keras fisik yang kemungkinan besar akan Anda gunakan karena jenis alat yang akan Anda gunakan bergantung pada jenis robotika yang Anda minati. Saya akan mendapatkan perangkat keras sebentar lagi.

Pertama, saya ingin berbicara sedikit tentang perangkat lunak. Perangkat lunak adalah satu area yang dibagikan di semua area robotika. Seperti kebanyakan hal dalam robotika, pilihan alat Anda sepenuhnya terserah Anda. Gunakan apa yang Anda rasa nyaman dan selesaikan pekerjaan.

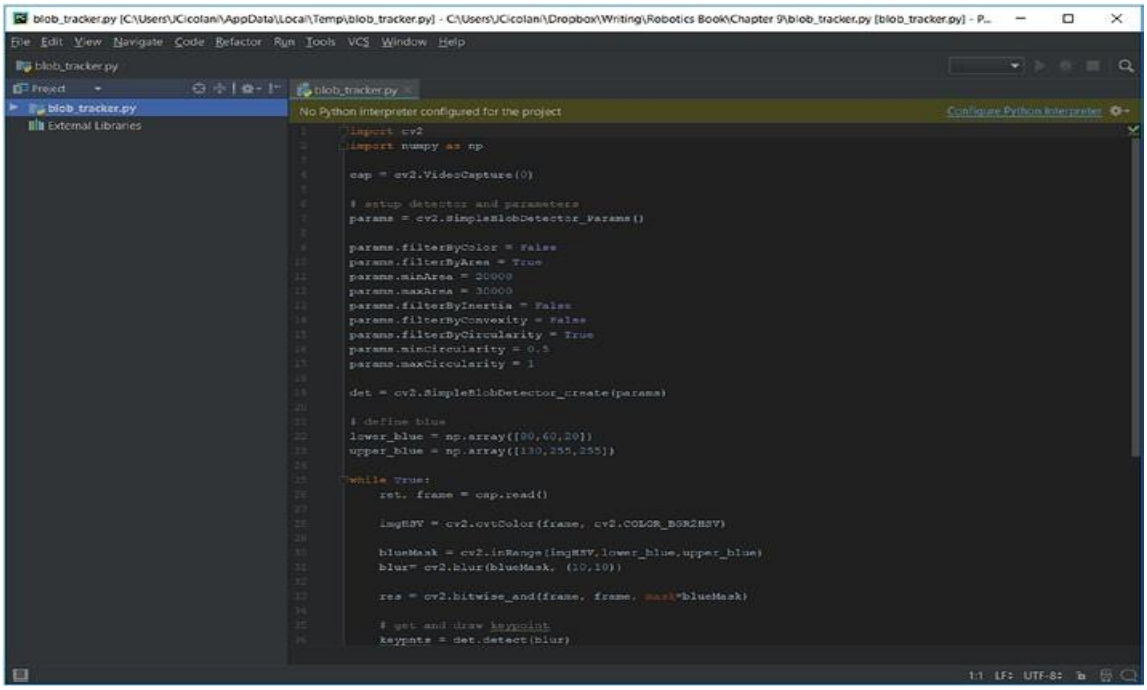
Perangkat lunak

Topik yang dibahas dalam buku ini jauh dari komprehensif. Masih banyak yang harus dipelajari tentang Linux, Python, Arduino, dan terutama OpenCV. Tujuannya adalah untuk memperkenalkan Anda pada beberapa konsep robotika, dan untuk membuat Anda terbiasa dan nyaman dengan beberapa alat.

Memilih IDE

IDE, atau lingkungan pengembangan terintegrasi, yang Anda gunakan terserah Anda. Ini adalah salah satu area yang dibagi di semua berbagai bidang. Ada banyak pilihan. Alat perangkat lunak yang kami gunakan asli dari Raspberry Pi dan Arduino. Dan dengan "asli" yang saya maksud adalah bahwa ini adalah alat yang dibangun ke dalam OS atau alat yang direkomendasikan untuk perangkat keras. Sebenarnya, di luar penulisan buku ini, saya tidak lagi menggunakan IDLE IDE. Alur kerja umum saya dimulai pada mesin Windows saya. Ketika kode bekerja seperti yang saya suka, saya mentransfernya ke Raspberry Pi untuk sentuhan akhir.

Alat pilihan saya untuk pemrograman Python adalah PyCharm(www.jetbrains.com/pycharm). Edisi Komunitas menawarkan semua fitur yang saya butuhkan untuk hampir semua proyek saya. Ini adalah IDE tingkat profesional yang membuat bekerja dengan Python lebih mudah (lihat Gambar 10-1). Ini tersedia untuk Windows dan Linux. Jadi, ketika saya mentransfer file ke Pi, saya dapat menggunakan alat yang sama untuk memperbarui kode, sesuai kebutuhan.



Gambar 10-1. IDE PyCharm

Spyder adalah IDE luar biasa lainnya untuk bekerja dengan Python. Itu disertakan dengan distribusi Anaconda dari Python, yang membuat instalasi sedikit lebih mudah. Ini menawarkan banyak alat yang ditujukan untuk komunitas ilmiah dan akademis. Anaconda sangat populer dengan banyak ilmuwan data yang bekerja dengan saya. Jika Anda tertarik untuk melihat Anaconda, Anda dapat menemukannya di www.anaconda.com. Atau, jika Anda ingin mencoba Spyder IDE, Anda dapat mengunduhnya di https://pythonhosted.org/spyder/. Juga, Microsoft Visual Studio adalah produk yang sangat kuat dan semakin mudah diakses. Sekali lagi, Edisi Komunitas mereka tersedia untuk diunduh dari situs web mereka (www.visualstudio.com). Sekali waktu, Visual Studio hanya untuk pengembang profesional. Bahkan ketika Microsoft mulai merilis Edisi Komunitas gratis, sulit bagi pemula dan penghobi untuk menggunakannya. Namun, beberapa rilis terakhir lebih ramah pengguna.

Salah satu hal menyenangkan tentang Visual Studio adalah dapat digunakan untuk sebagian besar kebutuhan pengembangan Anda. Itu memang memiliki kekurangannya. Misalnya, ini hanya tersedia untuk Windows. Ini masih memiliki sedikit kurva pembelajaran juga, tetapi ada banyak sumber daya yang tersedia untuk membantu. Sebagai IDE berbasis Windows, ia dikompilasi untuk Windows. Untungnya, Python bersifat lintas platform. Jadi, setelah Anda menulis kode, Anda dapat mentransfer file Python ke Raspberry Pi, membuat penyesuaian apa pun yang Anda perlukan untuk port serial dan seterusnya, lalu menjalankannya dari sana. Saya masih menggunakan Arduino IDE untuk sebagian besar pekerjaan Arduino saya. Ini hanya karena saya belum menemukan lingkungan mandiri yang lebih baik untuk bekerja. Visual Studio memiliki ekstensi yang memungkinkan Anda untuk mengembangkan kode Arduino dan kompilasi silang ke Arduino; meskipun melakukan kompilasi melalui Arduino IDE. Jadi, jika Anda mencari satu lingkungan untuk mengembangkan

proyek robotika Anda, Visual Studio mungkin merupakan pilihan yang baik.

Perangkat Lunak (Desain)

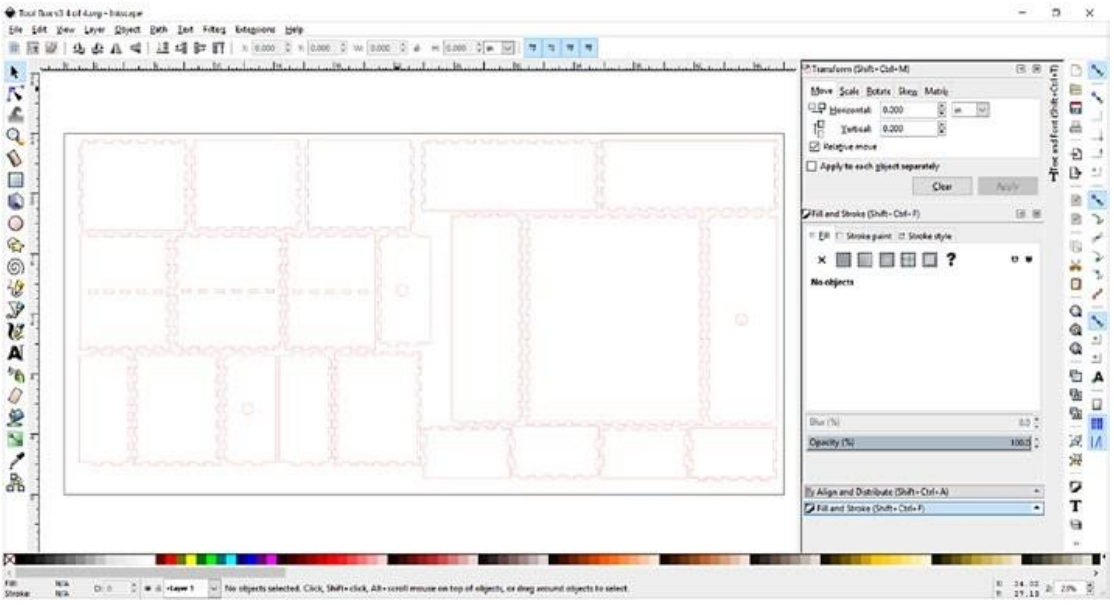
Banyak dari Anda mungkin tidak menggunakan perangkat lunak desain dengan frekuensi apa pun. Seperti yang lainnya, perangkat lunak yang digunakan untuk merancang berbagai bagian robot Anda akan bervariasi tergantung pada proyek Anda. Ini juga akan bervariasi tergantung pada anggaran Anda dan alat yang Anda gunakan untuk membuat robot Anda. Beberapa proyek, seperti kit atau desain orang lain, tidak memerlukan perangkat lunak desain sama sekali. Banyak proyek dan gaya bangunan lolos dengan pensil dan kertas sederhana. Jika Anda menggunakan bagian modular, Anda mungkin bisa lolos dengan daftar atau sketsa sederhana. Namun, untuk kustom apa pun, Anda mungkin memerlukan cara untuk mendesain sistem.

Gambar 2D

Perangkat lunak yang paling sederhana dan paling mudah digunakan adalah untuk desain 2D—atau datar. Alat-alat ini baik untuk merancang proyek yang dapat dibangun menggunakan bahan lembaran seperti MDF, karton, kayu lapis, atau lembaran akrilik. Jangan meremehkan apa yang dapat Anda lakukan dengan bahan datar. Proyek My Nomad dirancang dan dibangun menggunakan kayu lapis 1/4 inci. Ingatlah bahwa alat ini dirancang untuk seniman dan ilustrator, bukan untuk pekerjaan CAD yang presisi. Jadi beberapa fitur yang mungkin Anda harapkan tidak ada di sana. Misalnya, pengukuran yang tepat sulit dilakukan. Menggunakan kisi dan penggaris sangat membantu, tetapi jika Anda membutuhkan sudut atau panjang yang tepat, alat ini mungkin bukan yang terbaik untuk pekerjaan itu.

Salah satu alat desain 2D yang paling populer adalah proyek sumber terbuka yang disebut Inkscape (https://inkscape.org/en/). Inkscape sangat mudah digunakan, dan memiliki komunitas pengguna yang sangat besar. Ini gratis untuk diunduh dan digunakan, dan kaya dengan fitur. Ada juga banyak plugin yang dikembangkan komunitas. Salah satu favorit saya adalah pembuat kotak tab. Karena saya memiliki akses ke pemotong laser, saya menggunakan pembuat kotak bertab untuk mendesain kotak sederhana yang dapat saya potong dan jepret bersama-sama. Gambar 10-2 menunjukkan antarmuka Inkscape.

Ada juga program komersial yang tersedia. Adobe Illustrator (www.adobe.com/products/illustrator.html) dan Corel Draw (www.coreldraw.com/en/pages/ppc/coreldraw/) adalah dua pemimpin di bidang ini.

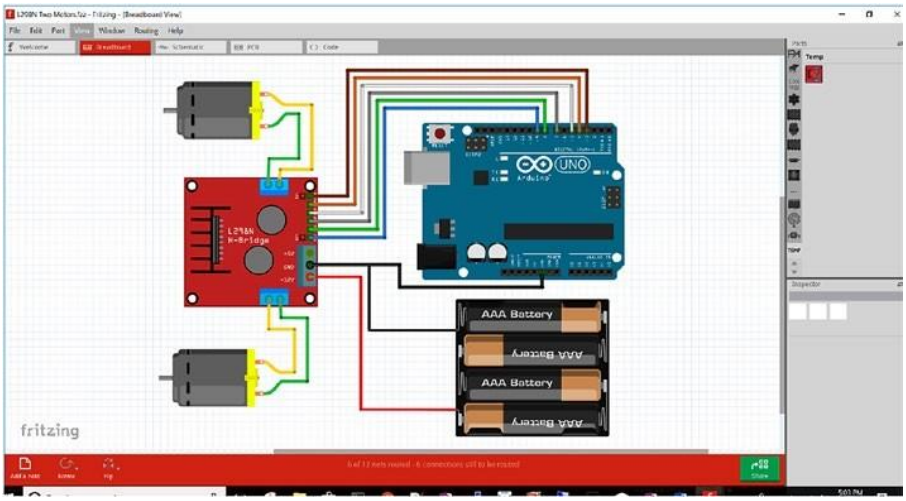


Gambar 10-2. Inkscape

Desain Papan Sirkuit

Pada titik tertentu, Anda mungkin perlu mendesain papan sirkuit atau pelindung Anda sendiri. Ini tidak serumit atau sesulit yang Anda bayangkan. Saat Anda semakin banyak bekerja dengan robotika, Anda akan menemukan rekomendasi untuk chip atau sirkuit tertentu. Seringkali, hanya dengan mencari secara online menyediakan tautan ke rangkaian contoh. Membuat ulang sirkuit ini dalam alat yang dirancang untuk itu memungkinkan Anda memesan papan. Ada banyak program yang dirancang untuk papan sirkuit. Faktanya, hampir setiap produsen papan sirkuit memilikinya.

Salah satu yang paling populer di komunitas hobi adalah Fritzing (http://fritzing.org/home/). Ini dikembangkan di University of Applied Science Potsdam di Jerman. Popularitasnya telah membuatnya dipecah menjadi organisasinya sendiri: Friends-of-Fritzing Foundation. Saya menggunakan perangkat lunak Fritzing untuk membuat diagram rangkaian dalam buku ini (lihat Gambar 10-3).



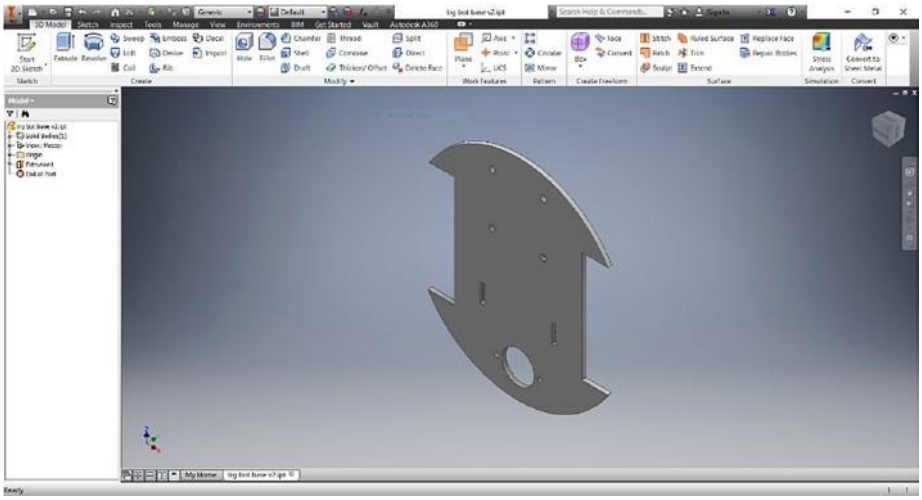
Gambar 10-3. Fritzing

Ada juga produk komersial yang tersedia; banyak dari mereka bebas menggunakan tambahan komunitas. Standar industri terkemuka adalah Eagle, yang sekarang dimiliki oleh Autodesk (www.autodesk.com/products/eagle/ Overview). Sebagian besar program lain mengimpor dan/atau mengekspor desain akhir dalam format Eagle yang populer.

Desain 3D

Jika Anda memiliki sasis dan suku cadang khusus, atau jika Anda menyukai pencetakan 3D, Anda memerlukan perangkat lunak CAD 3D. Sekali lagi, ada banyak tersedia. Tapi saya belum menemukan paket sumber terbuka atau gratis yang cocok dengan solusi komersial. Meskipun demikian, banyak dari solusi komersial memiliki versi pelajar yang ditawarkan secara gratis atau dengan harga yang lebih murah. SketchUp (www.sketchup.com) menawarkan versi gratis perangkat lunak yang dirancang untuk pembuat. Jika Anda belum pernah menggunakan program CAD sebelumnya, ini mungkin yang paling mudah dipelajari. Kontrolnya cukup intuitif dan ada banyak tutorial untuk membantu Anda mempelajari cara menggunakannya. Jika Anda pernah menggunakan CAD sebelumnya, maka ini mungkin bukan untuk Anda. Sebagian besar orang yang pernah bekerja dengan saya yang memiliki pengalaman CAD menganggap alat ini kurang intuitif. Itu karena tidak dirancang sebagai program CAD standar.

Bagi Anda yang lebih akrab dengan CAD, Autodesk menawarkan Fusion 360 (www.autodesk.com/products/fusion-360/overview) dengan biaya moderat. Perusahaan juga memberikan lisensi gratis kepada siswa untuk sebagian besar produk mereka, seperti Fusin 360, Inventor, dan sejumlah lainnya. Fusion 360 dan Inventor adalah program CAD profesional kelas komersial dengan sejumlah fitur, termasuk simulasi. Itu yang saya gunakan ketika saya perlu merancang sesuatu untuk robot saya atau proyek lain (lihat Gambar 10-4).



Gambar 10-4. Penemu Autodesk

Perangkat keras

Selain alat perangkat lunak yang saya jelaskan, Anda memerlukan beberapa alat yang sebenarnya. Pilihan alat Anda mungkin paling bergantung pada jenis robotika yang Anda minati, tetapi ada beberapa dasar yang harus dimiliki setiap kotak alat.

Alat Dasar

Di bagian ini, saya membahas alat yang mungkin Anda perlukan, terlepas dari bentuk robot atau proyek Anda, dan alat yang ada di kit dasar saya. Pertama, satu set tang yang baik adalah suatu keharusan. Anda membutuhkan ukuran dan jenis yang berbeda. Yang paling sering saya gunakan adalah set tang perhiasan. Saya juga sering menggunakan tang slip-joint. Pastikan set termasuk sepasang pemotong diagonal.

Selanjutnya dalam kit Anda, Anda ingin memiliki satu set obeng yang bagus. Banyak sekrup yang Anda gunakan berukuran kecil dan pas di tempat yang sempit. Pastikan ada berbagai hex head di set Anda. Seringkali, saya menemukan bahwa sekrup hex yang saya coba masukkan atau lepaskan berada di antara dua ukuran di set saya. Kepala bintang biasanya cocok dengan ini. Namun, berhati-hatilah, karena ada kemungkinan Anda bisa mencabut giginya. Dari sini, ada banyak alat lain yang bagus untuk dimiliki: pisau utilitas, array file, alat crimping, pemotong kawat flush, multimeter, kaliper, dan sebagainya. Anda akan mengumpulkan pilihan alat yang bagus.

Saya sangat menyarankan agar Anda membeli alat yang Anda butuhkan, daripada mencoba puas dengan apa yang Anda miliki. Menggunakan alat yang tepat untuk pekerjaan itu selalu memberikan hasil yang lebih baik. Dan, jika Anda meluangkan waktu untuk mendapatkan alat yang tepat, Anda akan memilikinya saat Anda membutuhkannya lagi. Anda juga membutuhkan stasiun solder. Itu tidak harus rumit. Setrika solder yang baik, tempat untuk fluks dan pembersih ujung Anda, dan satu set uluran tangan adalah yang Anda butuhkan.

Pastikan Anda memiliki tempat yang baik untuk menyimpan alat, dan cobalah untuk selalu mengembalikannya. Ini menghemat waktu Anda yang tak terhitung dari pencarian melalui kekacauan yang tak terhindarkan di ruang kerja Anda. Saya memiliki beberapa set alat. Satu set tinggal di meja kerja saya. Saya membeli sistem pegboard yang ringkas untuk menggantung sebagian besar peralatan saya. Apa yang tidak muat di papan pasak masuk ke laci khusus di bangku. Set lain ada di kotak peralatan yang saya tinggalkan bersama Nomad. Karena Nomad sering dibawa ke pertunjukan, dan segera kompetisi, saya ingin memastikan bahwa saya selalu memiliki apa yang saya butuhkan. Lebih sering daripada tidak, saya akhirnya membantu presenter lain di acara-acara karena mereka sering tidak siap.

Perangkat ketiga saya adalah perangkat mengambang. Saya menyimpannya di kotak peralatan yang mudah dipindahkan dari kamar ke kamar atau ke mobil saat saya menjelajah tanpa Nomad. Saya aktif di kancah robotika hobi lokal di Austin, dan ada baiknya untuk bersiap ketika seseorang membutuhkan tangan atau alat. Saya mencoba memastikan untuk selalu mengembalikan alat saya ke tempatnya ketika saya selesai menggunakannya. Ini memastikan bahwa lain kali saya meraih alat, itu ada di sana. Memang, saya tidak sekonsisten yang saya inginkan, tetapi ini adalah kebiasaan yang sangat baik untuk dilakukan.

Alat Khusus

Memiliki beberapa alat khusus yang lebih besar selalu membuat bangunan saya yang lebih besar lebih mudah. Gergaji pita dan mesin bor sangat berharga. Kecuali Anda berencana

untuk membuat beberapa robot yang sangat besar, versi benchtop dari kedua alat ini biasanya sudah cukup. Kombinasi benchtop belt/disk sander membantu membersihkan tepi atau membentuk bagian Anda. Selain semua alat ini, saya menggunakan alat yang lebih khusus. Sebagian besar, saya tidak memiliki alat ini di rumah. Tapi printer 3D cukup mudah didapat akhir-akhir ini; jika memungkinkan, memiliki satu atau dua di bengkel Anda adalah ide yang bagus. Saya juga menggunakan pemotong laser 120W, router CNC, dan pabrik CNC. Namun, itu bukan alat yang saya miliki di toko saya.

Ruang Praktek

Saya tidak memiliki pemotong laser dan pabrik CNC di toko rumah saya, seperti yang saya bayangkan sebagian besar dari Anda juga tidak memilikinya. Alat-alat ini besar dan mahal. Tapi, saya adalah anggota dari makerspace lokal di Austin: ATX Hackerspace (http://atxhs.org). Hackerspace adalah bengkel kerja sama tempat kami dapat mengumpulkan sumber daya kami untuk membeli beberapa mesin yang lebih besar. Apa yang tidak dimiliki *Hackerspace* sering kali dihosting oleh anggota untuk digunakan anggota lain.

Apa yang membuat ruang sangat berharga adalah komunitasnya. Makerspaces penuh dengan orang-orang yang suka membuat sesuatu. Orang-orang ini datang dari setiap lapisan masyarakat dan memiliki keterampilan individu. Ini adalah sumber yang sangat berharga ketika Anda mencoba melakukan sesuatu yang belum pernah Anda lakukan sebelumnya, atau Anda ingin mempelajari keterampilan baru, atau Anda ingin perspektif yang berbeda tentang suatu masalah, atau Anda hanya terjebak.

Saat ini, hampir setiap komunitas memiliki satu atau lebih ruang pembuat. Sumber daya yang tersedia bervariasi dari ruang ke ruang. Beberapa beroperasi di taman komersial, beberapa di sekolah, dan beberapa di luar garasi seseorang. Satu hal yang tetap konstan adalah komunitas. Jika Anda belum melakukannya, temukan ruang pembuat lokal Anda dan bergabunglah. Anda tidak akan menyesalinya.

10.3 RINGKASAN

Anda sekarang memiliki semua dasar yang Anda butuhkan untuk memulai robotika hobi. Jelas ada banyak lagi yang harus dipelajari dalam banyak topik. Tapi, Raspberry Pi dan Arduino akan membawa Anda jauh. Ingat, Anda tidak perlu mempelajari semuanya dalam ruang hampa. Ada komunitas besar di luar sana, dan itu berkembang setiap hari. Jangkau ruang pembuat lokal Anda untuk menemukan pembangun yang berpikiran sama. Jangan takut untuk bertanya. Jangan takut untuk melihat proyek orang lain untuk mendapatkan inspirasi. Manfaatkan kode sampel bila memungkinkan. Pada akhirnya, Anda akan menulis kode Anda sendiri, tetapi sampai saat itu, belajarlah dari mereka yang telah melakukannya.

Bidang robotika memang mengasyikkan. Fakta bahwa kita bisa masuk ke dalamnya, bereksperimen, dan belajar sangat fenomenal. Manfaatkan waktu ini. Yang terpenting, bersenang-senanglah. Semoga berhasil dan selamat membangun.

DAFTAR PUSTAKA

Agilent.(1999). Quadrature Decoder/Counter Interface ICs. Technical Data, Agilent Technologies, Inc., http://www.semiconductor.agilent.com

Asadi, A. (2014) Raspberry Pi for Beginners Second Revised Edition, Imagine Publishing Ltd, Bournemouth.

D. J. Norris, Beginning Artificial Intelligence with the Raspberry Pi. Appress, 2017.

D. Vaish, Python Robotics Projects_ Build smart and collaborative robots using Python. Puckt Publishing, 2018.

Gary Bradski, Adrian Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, "O'Reilly Media, Inc.". Copyright.September 2008, 1st Edition, Book ISBN: 978-0-596-51613-0

J. Costa Carlos, Aparicio Manuela, dan Caldeira Tiago. Teaching Programming through Open Source Robotics. (2016). 1-3

Johann Borenstein & Yoram Koren, Obstacle Avoidance with Ultrasonic Sensors, IEEE JOURNAL OF ROBOTICS AND AUTOMATION, VOL. 4, NO. 2, APRIL I988, pp. 213-218

Matt Richardson, Shawn Wallace, Getting Started with Raspberry Pi, 2nd Edition, Published by Maker Media, Inc., USA, 2014. Book ISBN: 978-1-457-18612-7

P. P. Ray, “IoRCar : IoT Supported Autonomic Robotic Movement and Control,” Int. Conf. Comput. Power, Energy, Inf. Commun. (ICCPEIC)ional Conf. Comput. power, energy, Inf. Commun., pp. 77–83, 2018.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., et al. (2011). Scikit-learn: Machine learning in python. Journal of machine learning research, 12(Oct), 2825–2830.

R. Laganière. OpenCV 2 Computer Vision Application Programming Cookbook. Packt Publishing 2011.

S. Pramono, “Pengendalian Robot Beroda Berbasis Arduino Uno R3 Menggunakan Koneksi Bluetooth,” J. Inform. SIMANTIK, vol. 1, no. 1, pp. 12–18, 2016.

Sanjana Prasad, P.Mahalakshmi, A.Jhon Clement Sunder, R.Swathi, “Smart Surveillance Monitoring System using Raspberry Pi and PIR Sensor”, International Journal of Computer Science and Information Technilogies, vol.5 (6), ISSN: 0975-9646, 2014

Upton, Eben, dan Gareth Halfacree. Raspberry Pi User Guide 2n edition. 2014

Robotika Raspberry Pi & Arduino

Memakai Python dan OpenCV

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. 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 978-623-5734-77-4 (PDF)

9 786235 734774

Robotika Raspberry Pi & Arduino

Memakai Python dan OpenCV

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

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