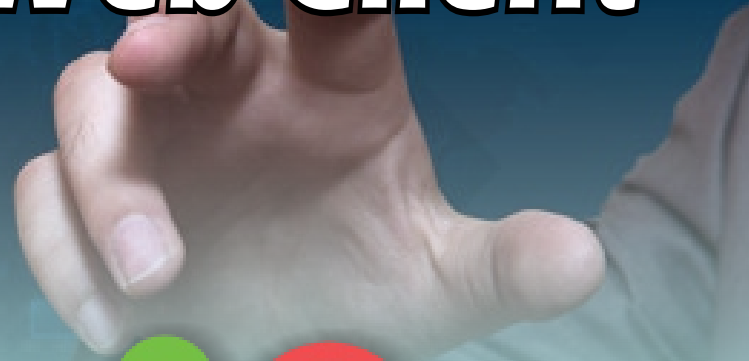


PEMROGRAMAN

Web Client



YAYASAN PRIMA AGUS TEKNIK

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

PEMROGRAMAN

Web Client



YAYASAN PRIMA AGUS TEKNIK

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

ISBN 978-602-52275-1-6



9 786025 227516

Perpustakaan Nasional RI. Data Katalog Dalam Terbitan (KDT)

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

PEMROGRAMAN WEB CLIENT

Perpustakaan Nasional RI, 2019.

xii, 272 hlm : 21 cm

ISBN : 9 786025 227516

1. Buku Ajar . I. Judul . II. Edisi . III. Perpustakaan Nasional

Desain Cover & Layout : Irdha Yunianto, S.Ds



YAYASAN PRIMA AGUS TEKNIK

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

Kata Pengantar

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa karena berkat rahmat Nya penulis dapat menulis buku Pemrograman Web Client ini.

Seperti yang telah kita ketahui bersama bahwa pengguna internet di Indonesia meningkat begitu pesat. Jumlah penduduk Indonesia saat ini mencapai lebih dari 262 juta orang. Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) lebih dari 50% atau sekitar 143 juta orang telah terhubung jaringan internet sepanjang tahun 2017. Terlebih lagi 49% pengguna internet di Indonesia adalah mereka yang berusia antara 19 hingga 34 tahun. Menurut Ofcom, ada indikasi bahwa penonton TV akan beralih ke media online. Hal ini mengindikasikan bahwa Internet adalah media utama di masa depan.

Di sisi lain, perkembangan teknologi website juga berkembang begitu cepat. Hingga saat ini yang paling populer digunakan adalah HTML5, CSS3 dan JavaScript. Dengan HTML5, CSS3, dan JavaScript pembuatan website menjadi lebih menarik dan interaktif. Ketiganya membantu kita untuk membuat aplikasi web yang lebih modern.

Melihat peningkatan jumlah pengguna internet dan perkembangan teknologi website yang terus berkembang, alangkah baiknya jika kita tidak hanya menjadi penonton namun juga berperan aktif dalam menciptakan website-website baru sebagai pilihan alternatif para pengguna internet yang terus meningkat. Pembuatan website telah dipermudah dengan berbagai macam CMS seperti wordpress, prestashop, joomla, dll. Selain itu, terdapat beberapa framework pembuatan website yang menerapkan metode MVC (Model, View, Controller) seperti Codeigniter, Laravel, Yii, dll. Meskipun pembuatan website telah banyak dipermudah dengan berbagai macam teknologi CMS dan framework yang canggih, namun mempelajari HTML5, CSS3, dan JavaScript masih menjadi syarat pendukung bila kita ingin mengubah atau mengedit tampilan bawaan yang ada.

Buku ini sangat cocok bagi anda yang ingin mempelajari Pemrograman Web Client dengan HTML5, CSS3, dan JavaScript. Saya menyadari bahwa buku ini belum sempurna dan masih memiliki banyak kekurangan. Tak lupa saya ucapkan terimakasih kepada semua pihak yang telah membantu dalam pembuatan buku ini. Saya berharap semoga buku ini dapat bermanfaat bagi pembaca dan menjadi amal kebaikan bagi segala pihak yang telah membantu dalam pembuatan buku ini.

Penulis

Daftar Isi

Kata Pengantar	1
1. Pengenalan HTML5 dan CSS3.....	7
1.1. Definisi HTML.....	8
1.2. Versi HTML.....	9
1.3. Definisi CSS.....	11
1.4. Versi CSS	12
2. DASAR-DASAR HTML	14
2.1. Heading dalam HTML	15
2.2. Paragraf dalam HTML	16
2.3. Link dalam HTML	16
2.4. Memasukkan Gambar dalam HTML	16
2.5. Tombol HTML.....	17
2.6. List dalam HTML.....	17
2.7. Tabel dalam HTML.....	18
2.8. Komentar dalam HTML.....	19
3. Hal Baru Dalam HTML5.....	20
3.1. Elemen-Elemen Struktur Baru dalam HTML5	21
3.2. Elemen Form Baru	22
3.3. Sintaks Atribut Baru.....	23
3.4. Grafik HTML5	23
3.5. Elemen Media Baru.....	23
4. Layout HTML5.....	24
4.1. Div	25
4.2. Header	25
4.3. Section.....	26
4.4. Article.....	26
4.5. Nav	27
4.6. Aside.....	28
4.7. Main	28
4.8. Footer	29
4.9. Membuat Layout HTML5	30
4.9.1. Menggabungkan Struktur Elemen HTML5	30

4.9.2.	Menambahkan CSS dalam Layout HTML5	32
5.	Dasar-Dasar CSS	38
5.1.	Syntax CSS.....	39
5.2.	Selector CSS.....	39
5.3.	Cara menggunakan CSS.....	41
5.4.	Memberi Warna dalam CSS.....	42
5.4.1.	Nama-Nama Warna dalam CSS.....	43
5.4.2.	Memberi Warna Tulisan	44
5.4.3.	Memberi Warna Border	44
6.	Property dalam CSS.....	46
6.1.	Property Background CSS	46
6.2.	Property Text Style.....	52
6.3.	Property Border	53
6.4.	Property Margin	56
6.5.	Property Padding.....	58
6.6.	Property Page Layout.....	59
6.7.	Property Interface.....	64
7.	Membuat Tampilan Website sederhana dengan HTML5 dan CSS3.....	65
7.1.	Membuat Header	65
7.2.	Membuat Footer	65
7.3.	Membuat Navigation.....	66
7.4.	Membuat Section dan Article.....	67
7.5.	Membuat Aside	68
7.6.	Membuat Sidebar	69
7.7.	Mengatur Tata Letak dengan Style CSS	70
7.8.	Menggabungkan Seluruh Kode.....	72
8.	Membuat Form dalam HTML5	77
8.1.	Membuat Form Teks	77
8.2.	Membuat Form Slider	78
8.3.	Membuat Form Spinboxes	79
8.4.	Membuat Form Tanggal.....	79
8.5.	Membuat Form Email	81

8.6.	Membuat form URL.....	82
8.7.	Form Warna.....	83
8.8.	Memberikan Petunjuk dengan Teks Placeholder.....	84
9.	Membuat Table Menarik dengan Pseudoclass	88
9.1.	Membuat Tabel	88
9.2.	Menambahkan CSS pada Tabel	89
9.2.1.	Memberi Warna yang Berbeda pada Setiap Row Tabel.....	90
9.2.2.	Membuat Rata Kanan dengan :nth-child	90
9.2.3.	Membuat Bold pada <i>Last Row</i> dengan: last-child	91
9.3.	Menggabungkan Seluruh Kode.....	92
10.	Canvas.....	95
10.1.	Membuat Canvas	95
10.2.	Menggambar Garis dalam Canvas.....	96
10.3.	Menggambar Lingkaran dalam Canvas.....	97
10.4.	Menulis Teks dalam Canvas.....	98
10.5.	Memberikan Gradien Warna dalam Canvas.....	98
11.	SVG dan MathML	100
11.1.	MathML.....	100
11.2.	SVG	101
11.2.1.	Membuat Lingkaran	104
11.2.2.	Membuat Persegi.....	105
11.2.3.	Membuat Ellipse	106
11.2.4.	Membuat Polygon	107
11.2.5.	Membuat Garis Terhubung	108
12.	Multimedia.....	109
12.1.	Elemen Video	109
12.1.1.	Format Video.....	109
12.1.2.	Width dan Controls	110
12.1.3.	Elemen Source.....	111
12.1.4.	Atribut Autoplay	112
12.1.5.	Atribut Muted.....	112
12.1.6.	Atribut Loop.....	112

12.1.7.	Atribut Preload	113
12.1.8.	Atribut Poster	114
12.2.	Elemen Audio	115
12.2.1.	Artibut Source	116
12.2.2.	Atribut Control	117
12.2.3.	Atribut Autoplay	117
12.2.4.	Atribut Loop	118
12.2.5.	Atribut Muted	118
12.2.6.	Atribut Preload	118
Referensi	1200

1. Pengenalan HTML5, CSS3 dan JavaScript

HTML5 dan CSS3 adalah trend baru dalam bahasa pemrograman berbasis web. Saat ini, hampir semua web browser seperti Google Chrome, Safari, Firefox, dan Opera sudah mendukung untuk bahasa pemrograman HTML5 dan CSS3.

HTML5 dan CSS3 bukan sekedar dua stantar baru yang diusulkan Worl Wide Web Consortium. Keduanya adalah kelanjutan iterasi teknologi yang setiap hari kita gunakan. Keduanya membantu kita untuk membuat aplikasi web yang modern. Sebelum kita mempelajari lebih dalam tentang HTML5 dan CSS3, ada baiknya kita mengetahui keunggulan dari HTML5 dan CSS3.

Beberapa kelebihan HTML5 dan CSS3 adalah:

- **Penulisan kode yang lebih efisien**

Jika dibandingkan dengan versi sebelumnya, HTML5 memiliki penulisan kode yang lebih efisien. Misalnya saja pada penulisan kode di awal dokumen. Jika pada HTML versi sebelumnya, dalam setiap dokumen HTML selalu diawali dengan kode HTML seperti berikut ini:

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"  
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd" >
```

Namun, pada HTML5, kode tersebut cukup ditulis dengan:

```
<!DOCTYPE html>
```

Penulisan kode ini jauh lebih efisien dibandingkan kode sebelumnya.

- **Dukungan multimedia yang lebih baik**

Tidak seperti versi sebelumnya, HTML5 tidak lagi membutuhkan aplikasi tambahan seperti flash untuk menjalankan video, audio atau multimedia yang lainnya. Hal ini memudahkan pengguna untuk menjalankan konten multimedia yang ada pada website yang kita bangun.

- **Form yang lebih baik**

HTML5 memiliki kontrol antarmuka pengguna yang lebih baik. Selama bertahun-tahun yang lalu, developer website menggunakan JavaScript dan CSS hanya untuk membuat slider, pemilih tanggal kalender, dan pemilih warna. Namun, dengan HTML5 hal tersebut tidak perlu dilakukan lagi. Hal itu telah didefinisikan sebagai elemen baru dalam HTML5, seperti drop-down, kotak centang, dan tombol radio. Anda akan belajar tentang cara menggunakannya di Bab Membuat Form.

- **Visual Efek yang baik**

Memberikan efek bayangan dan gradient pada teks dan gambar memberikan visual efek tersendiri pada halaman web. CSS3 memungkinkan Anda menambahkan bayangan dan gradien ke elemen tanpa menggunakan gambar latar belakang atau markup tambahan. Selain itu, Anda dapat menggunakan transformasi ke putaran sudut atau kemiringan elemen.

- **Markup yang lebih deskriptif**

Setiap versi HTML mengenalkan beberapa markup baru, namun sebelumnya belum pernah ada penambahan baru yang terkait langsung dengan penggambaran konten. Dalam HTML5 terdapat elemen-elemen yang secara langsung menggambarkan konten tersebut. Misalnya header, footer, navigation, section, article, dan lain lain.

- **Multimedia yang tidak bergantung pada plugin**

Kita tidak memerlukan Flash atau Silverlight untuk memutar video, audio, dan vector grafis lagi. Meskipun pemutar video berbasis Flash relatif mudah digunakan, namun dalam perangkat mobile Apple tidak dapat bekerja. Itu adalah pasar yang signifikan, jadi Anda harus belajar bagaimana menggunakan alternatif video non-Flash. Dalam buku ini, Anda akan melihat bagaimana menggunakan audio dan video HTML5 dengan pengganjilan yang efektif.

- **Aplikasi yang lebih baik**

Pengembang telah mencoba berbagai hal untuk membuat aplikasi interaktif yang lebih kaya di Web, mulai dari kontrol ActiveX hingga Flash. HTML5 menawarkan fitur luar biasa yang dalam beberapa kasus sepenuhnya menghilangkan kebutuhan akan teknologi pihak ketiga.

- **Aksesabilitas yang meningkat**

Elemen dalam HTML5 mendeskripsikan konten dengan jelas sehingga memudahkan programmer membaca layar untuk membaca konten. Sebagai contoh, sebuah navigasi situs akan lebih mudah untuk ditemukan dengan cukup mencari tag <nav> dalam <div> yang spesifik. Footer, sidebar, dan konten lainnya dapat dengan mudah ditemukan atau dilewati bersamaan.

1.1. Definisi HTML

HTML adalah bahasa standar untuk membuat website. *Hyper Text Markup Language* atau yang biasa disingkat HTML, menggambarkan struktur halaman web menggunakan *markup*. *Hypertext Markup Language* terdiri dari tiga kata, yaitu: ***Hypertext***, ***Markup*** dan ***Language***. Mari kita bahas satu per satu.

Kata **Hypertext** dari HTML menekankan pengertian bahwa text dalam HTML lebih dari sekedar teks ('hyper'-text). Maksudnya selain berfungsi sebagai teks biasa, sebuah teks dalam HTML juga bisa berfungsi sebagai penghubung suatu halaman ke halaman lain. Hal ini biasa dikenal dengan istilah link.

Kata selanjutnya adalah **Markup**. Kata *Markup* dapat diterjemahkan sebagai tanda atau penanda. Dalam bahasa HTML, digunakan tanda-tanda khusus seperti <h1>, <p>, atau dan lain-lain. Tanda ini diperlukan untuk mengatur format dan membuat struktur halaman web.

Kata terakhir adalah **Language**. Kata *Language* jika diterjemahkan berarti bahasa. Bagi anda yang pernah belajar bahasa pemrograman, disini HTML tidak menyebut dirinya sebagai *Programming Language*, tetapi hanya *Language* saja. Hal ini secara tidak langsung menyatakan bahwa HTML bukanlah sebuah bahasa pemrograman. HTML tidak memiliki struktur dasar seperti variabel, kondisi if ... else ..., function, atau class seperti layaknya sebuah bahasa pemrograman.

Elemen HTML adalah blok pembangun halaman HTML. Elemen HTML diwakili oleh tag yang melabeli potongan konten seperti "heading", "paragraph", "table", dan seterusnya. Browser kita tidak akan menampilkan tag HTML, namun menggunakannya untuk merender konten halaman.

Contoh:

- Deklarasi <!DOCTYPE html> mendefinisikan dokumen ini sebagai HTML5
- Elemen <html> adalah elemen root dari sebuah halaman HTML
- Elemen <head> berisi informasi meta tentang dokumen
- Elemen <title> menentukan judul untuk dokumen
- Elemen <body> berisi konten halaman yang terlihat
- Elemen <h1> mendefinisikan heading besar
- Elemen <p> mendefinisikan sebuah paragraf

1.2. Versi HTML

HTML5 adalah iterasi pengembangan dari HTML versi sebelumnya. Berikut ini adalah table versi HTML beserta tahun penemuannya.

Versi	Tahun
HTML	1991
HTML 2.0	1995
HTML 3.2	1997
HTML 4.01	1999
XHTML	2000
HTML5	2014

Pada akhir tahun 1991, Tim Berners-Lee mempublikasikan dokumen yang berjudul: HTML Tags. Dalam dokumen ini, Tim membuat konsep sederhana tentang HTML yang terdiri dari 18 element.

Daripada membuat standar bahasa baru, Tim Berners-Lee merancang HTML yang didasarkan kepada konsep bahasa markup yang dikenal dengan SGML (*Standard Generalized Markup Language*).

Pada tahun 1993, web browser lain yang digunakan untuk menampilkan HTML mulai bermunculan. Beberapa diantaranya adalah: Lynx, Mosaic dan Arena. Tetapi karena belum tersedia sebuah standar baku bagaimana aturan HTML ditulis, masing-masing web browser mendefinisikan versi HTML mereka sendiri. Untuk mengatasi hal ini, Tim Berners-Lee dan rekannya Dave Raggett, mengajukan sebuah proposal berjudul: *Hypertext Markup Language, Ver 1.0* kepada badan standarisasi IEFT (*Internet Engineering Task Force*). IEFT adalah badan standar internasional yang menangani arsitektur internet. Proposal inilah yang menjadi HTML versi 1.0. Namun draft HTML 1.0 ini gagal menjadi standar karena melewati batas waktu yang ditentukan oleh IEFT. Oleh karena itu, HTML versi 1.0 tidak pernah menjadi sebuah standar resmi.

Karena draft HTML 1.0 mengalami permasalahan, pada awal 1994 IEFT membentuk HTML *Working Group* atau yang biasa disingkat menjadi HTMLWG. HTMLWG bertujuan untuk menyempurnakan HTML yang sebelumnya diusulkan oleh Tim Berners-Lee dan Dave Raggett. Pada Juli 1994, HTML 2.0 resmi menjadi standar HTML pertama yang disahkan oleh IEFT. HTML 2.0 memiliki spesifikasi yang didasari kepada draft HTML 1.0 dengan penambahan beberapa fitur baru yang telah banyak digunakan pada web browser saat itu. Salah satu fitur barunya adalah penambahan tag `` untuk menampilkan gambar dalam browser.

Pada tahun 1995 terjadi perang browser antara Internet Explorer dengan Netscape. Era ini menyulitkan proses pembuatan web dengan HTML, karena programmer harus membuat 2 buah kode program untuk masing-masing web browser. Ini diperlukan karena sebuah tag HTML yang berjalan di Netscape, bisa jadi tidak tersedia di IE, dan sebaliknya.

Melihat kebutuhan akan standar HTML yang baru, pada pertengahan tahun 1995, Dave Raggett yang saat itu bekerja sebagai peneliti di perusahaan komputer Hewlett-Packard (HP), mencoba mengajukan draft HTML 3.0 kepada badan standarisasi IETF. Akan tetapi karena draft HTML 3.0 mengalami banyak perdebatan tentang fitur apa yang harus ada, dan fitur mana yang ingin dihapus. Akhirnya, HTML 3.0 gagal menjadi standar.

Nasib perkembangan HTML kemudian ditentukan oleh *World Wide Web Consortium* (W3C) yang didirikan oleh Tim Berners-Lee. W3C mencoba memperbaiki draft HTML 3.0 dan akhirnya pada Januari 1997, standar HTML 3.2 resmi dirilis oleh W3C. Salah satu fitur baru yang ditawarkan dalam HTML 3.2 adalah `<table>`. Selain itu, dalam HTML 3.2 juga diperkenalkan pemisahan antara konten dengan tampilan. Untuk membuat desain tampilan website, W3C memisahkannya dari HTML dan membuat fungsi tersendiri dikenal dengan CSS.

Pesatnya perkembangan internet dan HTML, membuat W3C langsung mengajukan draft HTML versi 4.0 pada akhir tahun 1997. Pada Desember 1998, standar HTML 4.0 resmi di rilis oleh W3C.

Setahun kemudian, pada Desember 1999, HTML 4.01 dirilis untuk menyempurnakan beberapa error yang terdapat pada HTML 4.0.

Perang web browser Netscape vs IE terus berkembang, keduanya berlomba menambahkan fitur baru selain standar HTML 4.01. Melihat hal ini, W3C kembali mengembangkan versi perbaikan dari HTML 4.01. W3C mencoba beralih ke konsep bahasa markup lain yaitu XML. Pada tahun 1999 perkembangan HTML menjadi beralih ke XHTML. XHTML adalah singkatan dari *eXtensible HyperText Markup Language*.

W3C berpendapat bahwa XHTML adalah bahasa web masa depan dan sebagai penerus dari HTML. W3C secara resmi mempublikasikan standar XHTML 1.0 pada Januari 2000. Segera setelah XHTML 1.0 dipublikasikan, W3C merilis standar XHTML 1.1 pada Mei 2001. Melanjutkan perkembangan standar XHTML, selama Agustus 2002 sampai dengan Juli 2006, W3C mencoba membuat draft untuk generasi XHTML berikutnya, yakni XHTML 2.0.

XHTML 2.0 adalah versi yang sepenuhnya baru, dan mencoba memutus rantai dengan HTML. XHTML 2.0 tidak lagi mendukung fitur-fitur yang sebelumnya ada pada HTML. Hal ini membuat mayoritas programmer web mengajukan protes. Menurut mereka, XHTML 2.0 tidak melihat kebutuhan web developer dan hanya berfokus kepada standar yang akan susah diimplementasikan. Masalah ini membuat pembahasan mengenai XHTML 2.0 mengalami perdebatan panjang dan terhenti.

Melihat arah XHTML yang tidak jelas dan “berhenti” pada draft XHTML 2.0, beberapa programmer dari Apple, Mozilla Foundation dan Opera Software mendirikan *Web Hypertext Application Technology Working Group*, yang disingkat menjadi WHATWG pada tahun 2004. WHATWG berkeinginan untuk membuat standar HTML yang didasarkan kepada kebutuhan programmer dan web browser. WHATWG kemudian merancang draft Web Forms 2.0 dan Web Apps 1.0 yang kemudian digabung menjadi HTML5.

WHATWG dan W3C saling bekerja sama untuk mengembangkan masa depan HTML. Pada Juli 2012, keduanya sepakat untuk ‘membagi tugas’. W3C mengembangkan standar tunggal HTML 5, sedangkan WHATWG mengerjakan apa yang disebut sebagai Living Standard HTML5. pada tanggal 28 Oktober 2014, standar HTML5 resmi dirilis oleh W3C.

1.3. Definisi CSS

CSS adalah singkatan dari *Cascading Style Sheets*. CSS digunakan untuk mempercantik tampilan halaman website. CSS berisi rangkaian instruksi untuk mengatur beberapa komponen dalam sebuah web sehingga akan lebih terstruktur dan seragam. Seperti yang telah dibahas sebelumnya, CSS mulai dikenal sejak HTML 3.2 diluncurkan.

Kata *Cascade* dari kepanjangan CSS diambil dari bahasa inggris, cascade yang berarti “air terjun kecil, riam, jeram, mengalir/berpancaran kebawah”. Dimana maknanya adalah: sesuatu yang mengalir dari atas ke bawah. Di dalam CSS, style atau aturan tampilan yang dibuat bisa saja saling

menimpa satu sama lain, tergantung dari posisinya dan ke-spesifikan kode CSS tersebut. Sebagai contoh, jika pada baris pertama kode CSS kita membuat perintah untuk mengubah warna paragraf menjadi biru, di baris kedua kita bisa menulis kembali perintah yang sama, tetapi kali ini mengubah warna tersebut menjadi merah.

Style sheet language adalah format bahasa khusus yang terdiri dari kumpulan kode untuk mengatur tampilan (style) dari sebuah dokumen. CSS dapat mengendalikan ukuran gambar, warna bagian tubuh pada teks, warna tabel, ukuran border, warna border, warna hyperlink, warna mouse over, spasi antar paragraf, spasi antar teks, margin kiri, kanan, atas, bawah, dan parameter lainnya. CSS adalah bahasa style sheet yang digunakan untuk mengatur tampilan dokumen. Dengan adanya CSS memungkinkan kita untuk menampilkan halaman yang sama dengan format yang berbeda.

1.4. Versi CSS

Pada Desember 1996, CSS 1.0 resmi dipublikasikan dan menjadi standar W3C. Namun standar CSS yang dikeluarkan oleh W3C ini hanyalah bersifat “rekomendasi”, artinya pembuat web browser boleh memilih apakah akan mengikuti standar ini atau tidak. Setidaknya butuh 3 tahun sejak CSS 1.0 diluncurkan hingga didukung penuh oleh web browser yang pada saat itu di dominasi oleh Internet Explorer dan Netscape. CSS 1.0 dikembangkan hanya dengan berpusat pada pemformatan dokumen HTML.

Sesat setelah CSS 1.0 di rilis pada November 1997, W3C langsung bekerja untuk mengembangkan versi penerus CSS. Akhirnya CSS 2.0 secara resmi menjadi standar W3C pada 12 May 1998. Namun standar CSS 2.0 ini cukup rumit dan susah diimplementasikan oleh web browser. Oleh karena itu segera diadakan revisi untuk perbaikan, yang dinamakan CSS 2.1. Setelah beberapa tahun pengembangan, pada 25 Februari 2004, CSS 2.1 direncanakan untuk dirilis secara resmi, dimana dalam tahapan status W3C sudah sebagai “kandidat rekomendasi” (Candidate Recommendation). Namun, karena satu dan lain hal, CSS 2.1 ditarik untuk direvisi ulang. Proses revisi ini membutuhkan waktu beberapa tahun dan diusulkan kembali menjadi candidate recommendation pada 19 Juli 2007. Sekali lagi, karena masih butuh penambahan dan revisi, spesifikasi CSS 2.1 lagi-lagi kembali diperbaiki dan juga makan waktu beberapa tahun. Akhirnya pada tanggal 12 April 2011, CSS 2.1 kembali masuk ke tahap Candidate Recommendation dan secara resmi menjadi standar pada 7 Juni 2011.

Seperti yang kita lihat, pengembangan CSS 2.1 yang bolak balik dari *working draft* ke *candidate recommendation* dan kembali lagi ke *working draft* membutuhkan waktu hampir 17 tahun, dan baru “selesai” pada tahun 2011. Dalam jangka waktu ini, web browser secara bertahap tetap mengimplementasikan standar CSS 2.1 yang belum selesai. Sehingga terdapat beberapa perbedaan dari satu web browser dengan web browser lainnya.

Memetik pelajaran dari “susahnya” membuat spesifikasi CSS 2.1, W3C memakai konsep baru dalam mengembangkan versi kelanjutan CSS, yang dimulai segera setelah CSS 2.1 selesai. Untuk CSS 3, W3C memutuskan memecah spesifikasi CSS menjadi modul-modul terpisah. Setiap modul kemudian dapat dikembangkan secara independen tanpa harus menunggu modul lainnya selesai.

Dengan cara seperti ini, fitur baru CSS bisa hadir dengan lebih cepat. Saat ini terdapat puluhan modul CSS 3, dimana masing-masingnya dikembangkan secara terpisah.

CSS1 dikembangkan berpusat pada pemformatan dokumen HTML, CSS2 dikembangkan untuk memenuhi kebutuhan terhadap format dokumen agar bisa ditampilkan di printer, sedangkan CSS3 adalah versi terbaru dari CSS yang mampu melakukan banyak hal dalam desain website. CSS2 mendukung penentuan posisi konten, downloadable, font huruf, tampilan pada layout tabel dan media tipe untuk printer. Kehadiran versi CSS yang kedua diharapkan lebih baik dari versi pertama dan kedua.

CSS3 juga dapat melakukan animasi pada halaman website, di antaranya animasi warna hingga animasi 3D. Dengan CSS3 desainer lebih dimudahkan dalam hal kompatibilitas websitenya pada *smartphone* dengan dukungan fitur baru yakni media query. Selain itu, banyak fitur baru pada CSS3 seperti: multiple background, border-radius, drop-shadow, border-image, CSS Math, dan CSS Object Model.

1.5. Definisi JavaScript

JavaScript adalah bahasa pemrograman yang memiliki ciri-ciri:

- tingkat tinggi (high-level),
- dinamis,
- tidak bertipe dan diproses secara interpreted.

JavaScript menggunakan standar spesifikasi ECMAScript. Bersama-sama dengan HTML dan CSS, JavaScript menjadi salah satu teknologi inti dari pembuatan konten halaman web (World Wide Web).

1.6. Versi JavaScript

Javascript lahir dan dikembangkan oleh Netscape Communications. Marc Andreessen, pendiri Netscape Communications, menyadari bahwa web seharusnya lebih dinamis. Kemudian, pada tahun 1995, Netscape merekrut Brendan Eich untuk mengimplementasikan bahasa pemrograman “Scheme” ke dalam web browser Netscape. Saat itu bahasa pemrograman Scheme cukup populer dan banyak digunakan oleh kalangan akademisi.

Pada Mei 1995, bahasa pemrograman

“Mocha” lahir. Bahasa pemrograman inilah yang diputuskan untuk digunakan Netscape. Nama

Mocha sendiri dipilih oleh Marc Andreessen.

Bahasa pemrograman Mocha dirilis pertama kali ke dalam versi beta Netscape Navigator 2.0 di bulan September 1995, tetapi dengan menggunakan nama baru: LiveScript. Kata ‘live’ dari LiveScript bisa bermakna bahwa bahasa ini akan menambah efek dinamis ke dalam HTML.

Umur dari LiveScript ternyata tidak lama. 3 bulan kemudian, tepatnya Desember 1995, hadir Netscape Navigator 2.0 beta 3 dengan sebuah bahasa baru: JavaScript. Sebenarnya ini bukanlah bahasa pemrograman baru, tapi perubahan nama dari LiveScript.

Nama JavaScript dipilih agar Netscape bisa ‘nompang tenar’ dari bahasa pemrograman JAVA milik Sun Microsystems, yang pada masa itu sangat populer di kalangan programmer. Netscape mendapat izin resmi menggunakan awalan “Java” dari Sun Microsystems.

Pilihan nama JavaScript inilah yang menjadi sumber kebingungan programmer hingga saat ini. Banyak yang menyangka JavaScript

Akhir-akhir ini, JavaScript berkembang sangat pesat, terutama sejak adanya Node.js. Dengan Node.js, JavaScript dapat berjalan sebagai bahasa pemrograman server. Fitur utamanya seperti non-blocking I/O model, dimana banyak pemrosesan bisa berjalan bersamaan, tanpa menunggu proses lain selesai. Selain itu juga banyak bermuculan framework JavaScript seperti AngularJS yang di kembangkan oleh Google dan ReactJS yang dikelola oleh FaceBook. Framework seperti ini digunakan untuk membuat website yang tidak berbentuk “website”, tetapi menyerupai aplikasi desktop yang dikenal sebagai Single-page Application (SPA). Beberapa contoh dari single-page application adalah google drive dan google doc. Jika anda membuka google drive atau google doc, halamannya akan tetap sama, tidak di reload seperti layaknya sebuah website.

2. DASAR-DASAR HTML

Pada bab ini kita akan belajar tentang dasar-dasar HTML. Salah satu cara paling sederhana untuk belajar menulis kode HTML adalah dengan menggunakan aplikasi notepad yang kemudian disimpan dalam format .html atau .htm.

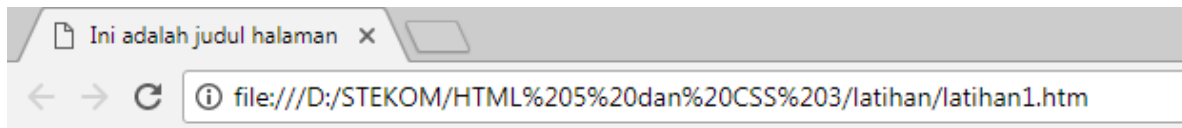
Ikuti empat langkah berikut ini untuk membuat file HTML pertama anda.

1. Buka aplikasi notepad atau notepad++.
2. Ketikkan kode HTML dalam notepad tersebut.

Berikut ini adalah salah satu contoh kode HTML:

```
<!DOCTYPE html>
<html>
<head>
  <title>Judul Halaman</title>
</head>
<body>
  <h1>Ini adalah contoh heading h1</h1>
  <p>Ini adalah contoh paragraf.</p>
</body>
</html>
```

3. Save file notepad yang telah dibuat dengan format .html.
4. Buka file tersebut dengan web browser yang kamu miliki.



Ini adalah contoh heading h1

Ini adalah contoh paragraf.

2.1. Heading dalam HTML

Dalam HTML, heading didefinisikan dengan tag <h1> hingga <h6>. <h1> mendefinisikan heading paling penting, dan <h6> digunakan untuk heading paling tidak penting.

```
<h1>Ini adalah heading 1</h1>
```

```
<h2>Ini adalah heading 2</h2>
```

```
<h3>Ini adalah heading 3</h3>
```

Jika kode diatas dijalankan, maka akan menampilkan seperti gambar berikut ini:

Ini adalah heading 1

Ini adalah heading 2

Ini adalah heading 3

2.2. Paragraf dalam HTML

Paragraf dalam HTML didefinisikan dengan tag <p>. Berikut ini adalah contoh penulisan paragraf dalam HTML:

```
<p>Ini adalah contoh paragraph dalam HTML.</p>  
<p>Ini adalah contoh paragraph dalam HTML.</p>
```

Jika kode diatas dijalankan, maka akan menampilkan seperti gambar berikut ini:

[Ini adalah contoh paragraph dalam HTML.](#)

[Ini adalah contoh paragraph dalam HTML.](#)

2.3. Link dalam HTML

Link dalam HTML didefinisikan dengan tag <a>. Tujuan link secara spesifik disebutkan dalam atribut href.

```
<a href="https://www.stekom.ac.id">Ini adalah link </a>
```

Jika kode diatas dijalankan, maka akan menampilkan seperti berikut ini:

[Ini adalah link](#)

2.4. Memasukkan Gambar dalam HTML

Untuk memasukkan gambar, dalam HTML digunakan tag . Sumber file gambar dituliskan dalam atribut src.

```

```

Kode diatas menandakan bahwa sumber file gambar bernama stekom.png berada dalam satu folder yang sama dengan tempat penyimpanan kode HTML tersebut. Berikut ini adalah hasil tampilan dari kode diatas:

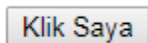


2.5. Tombol HTML

Tombol button dalam HTML didefinisikan dengan tag `<button>`

```
<button>Klik Saya</button>
```

Jika kode diatas dijalankan, maka akan menampilkan seperti gambar berikut ini:



2.6. List dalam HTML

List dalam HTML didefinisikan dengan tag `` atau `` yang kemudian diikuti dengan tag ``. Berikut ini adalah contoh list dalam HTML:

```
<ul>  
<li>Kopi</li>  
<li>Teh</li>  
<li>Susu</li>  
<li>Madu</li>
```


Jika kode diatas dijalankan, maka akan menampilkan seperti gambar berikut ini:

- Kopi
- Teh
- Susu
- Madu

2.7. Tabel dalam HTML

Dalam HTML, table didefinisikan dengan tag <table>. Setiap row didefinisikan dengan <tr>. Header table didefinisikan dengan <th> yang secara otomatis akan menjadi bold dan center. Data table atau cell didefinisikan dengan <td>.

Contoh:

```
<!DOCTYPE html>
<html>
<body>
<h2>Basic HTML Table</h2>
<table style="width:40%">
<tr>
  <th>Nama Depan</th>
  <th>Nama Belakang</th>
  <th>Umur</th>
</tr>
<tr>
  <td>Joni</td>
  <td>Siahaan</td>
  <td>20</td>
</tr>
<tr>
  <td>Eva</td>
  <td>anggraeni</td>
  <td>22</td>
</tr>
<tr>
  <td>Dodi</td>
  <td>Utomo</td>
  <td>19</td>
</tr>
```

```
</table>
</body>
</html>
```

Bila dijalankan dalam browser, akan memberikan tampilan sebagai berikut:

Tabel HTML Sederhana

Nama Depan	Nama Belakang	Umur
Joni	Siahaan	20
Eva	anggraeni	22
Dodi	Utomo	19

Pada bab lain dalam buku ini, kita akan mempelajari bagaimana cara mempercantik tampilan table dengan menambahkan kode CSS.

2.8. Komentar dalam HTML

Seperti pada bahasa pemrograman yang lain, HTML juga memiliki komen untuk mempermudah developer dalam mengembangkan kode programnya. Komentar berfungsi untuk memberikan keterangan terkait kode program yang telah dibuat. Dalam HTML, komen tersebut ditulis dengan

```
<!-- tulis komentar disini -->
```

Agar lebih mudah dipahami, silahkan tulis ulang kode dibawah ini kemudian save dengan nama komentar.html.

```
<!DOCTYPE html >
<html>
<head>
  <title>Belajar Tag Komentar (comment)</title>
</head>
<body>
  <!-- <p>Ini berada di dalam tag komentar, dan tidak akan tampil di browser</p> -->
  <p>Ini bukan komentar, dan akan tampil di browser</p>
</body>
</html>
```

3. Hal Baru Dalam HTML5

Beberapa elemen baru dalam HTML5 yang paling menarik adalah:

- Elemen semantik baru seperti `<header>`, `<footer>`, `<article>`, dan `<section>`.
- Atribut baru dari elemen form seperti *number*, *date*, *time*, *calendar*, dan *range*.
- Elemen grafis baru: `<svg>` dan `<canvas>`.
- Elemen multimedia baru: `<audio>` dan `<video>`.

Berikut ini adalah beberapa elemen HTML4 yang dihilangkan di HTML5:

Elemen yang dihilangkan	Pengganti
<acronym>	<abbr>
<applet>	<object>
<basefont>	CSS
<big>	CSS
<center>	CSS
<dir>	
	CSS
<frame>	
<frameset>	
<noframes>	
<strike>	CSS, <s>, or
<tt>	CSS

3.1. Elemen-Elemen Struktur Baru dalam HTML5

Berikut ini adalah beberapa struktur elemen baru dalam HTML5 yang tidak terdapat dalam versi sebelumnya:

Tag	Deskripsi
<article>	Mendefinisikan artikel dalam dokumen
<aside>	Mendefinisikan content aside dari page content
<bdi>	Memisahkan bagian teks yang mungkin memiliki format berbeda dari teks yang lain
<details>	Mendefinisikan detail tambahan yang pengguna bisa melihat atau menyembunyikannya
<dialog>	Mendefinisikan kotak dialog atau jendela
<figcaption>	Mendefinisikan keterangan untuk elemen <figure>
<figure>	Mendefinisikan konten mandiri
<footer>	Mendefinisikan footer untuk sebuah dokumen atau bagian

<header>	Mendefinisikan header untuk sebuah document atau bagian
<main>	Mendefinisikan content utama dari sebuah dokumen
<mark>	Mendefinisikan teks yang ditandai atau <i>dihighlight</i>
<menuitem>	Mendefinisikan sebuah item menu yang user dapat meramu dari sebuah <i>popup</i> menu
<meter>	Menentukan pengukuran skalar dalam rentang yang dikenal
<nav>	Mendefinisikan link navigasi
<progress>	Merepresentasikan progress sebuah tugas
<rp>	Mendefinisikan apa yang harus ditampilkan dalam browser yang tidak support anotasi ruby
<rt>	Mendefinisikan penjelasan/pengucapan karakter (untuk tipografi Asia Timur)
<ruby>	Mendefinisikan anotasi ruby (untuk tipografi Asia Timur)
<section>	Mendefinisikan bagian dalam document
<summary>	Mendefinisikan sebuah heading yang terlihat untuk sebuah elemen <details>
<time>	Mendefinisikan tanggal/waktu
<wbr>	Mendefinisikan sebuah jeda garis

Tabel diatas memberikan gambaran tentang beberapa elemen struktur baru dalam HTML5. Beberapa struktur baru dalam table diatas akan kita pelajari secara lebih detail pada bab lain dalam buku ini.

3.2. Elemen Form Baru

Berikut ini adalah elemen baru dalam form HTML5 yang tidak terdapat dalam versi HTML sebelumnya:

Tag	Deskripsi
<datalist>	Digunakan untuk menentukan daftar opsi yang diijinkan sebagai inputan
<output>	Menentukan hasil perhitungan

3.3. Sintaks Atribut Baru

HTML5 memungkinkan empat sintaks yang berbeda untuk atribut. Contoh ini menunjukkan sintaks yang berbeda yang digunakan dalam tag `<input>`:

Type	Contoh
Empty	<code><input type="text" value="Joni" disabled></code>
Unquoted	<code><input type="text" value=Joni></code>
Double-quoted	<code><input type="text" value="Joni Darda"></code>
Single-quoted	<code><input type="text" value='Joni Darda'></code>

Dalam HTML5, keempat sintaks diatas dapat digunakan, tergantung pada apa yang diperlukan untuk atribut.

3.4. Grafik HTML5

Terdapat dua hal baru dalam hal grafis pada HTML5, yaitu canvas dan svg. Berikut ini adalah table yang berisi tag grafis dalam HTML5 beserta penjelasannya:

Tag	Deskripsi
<code><canvas></code>	Menggambar grafik dengan cepat melalui scripting (biasanya JavaScript)
<code><svg></code>	Menggambar grafik vector scalable

Pada bab lain dalam buku ini, kita akan membahas secara tersendiri dan lebih mendalam tentang canvas dan svg. Dalam bab tersebut kita juga akan membahas tentang perbedaan canvas dan svg serta kelebihan dan kekurangannya.

3.5. Elemen Media Baru

Selain memberi dukungan penuh terhadap grafis, HTML5 juga memberikan dukungan terhadap media. Pada versi HTML sebelumnya, kita membutuhkan peran dari pihak ketiga (seperti adobe flash) untuk membuka media dalam browser. Kini, HTML5 telah memiliki elemen tersendiri untuk mendukung keperluan multimedia, sehingga tidak lagi membutuhkan peran serta pihak ketiga. Berikut ini adalah table yang berisi beberapa elemen baru dalam HTML5 yang mendukung dalam hal multimedia:

Tag	Deskripsi
-----	-----------

<audio>	Mendefinisikan konten suara
<embed>	Menentukan sebuah wadah untuk aplikasi eksternal (non-HTML)
<source>	Menentukan berbagai sumber media untuk elemen media (<video> dan <audio>)
<track>	Menentukan track teks untuk elemen media (<video> dan <audio>)
<video>	Menentukan video atau film

Pada bab lain dalam buku ini, akan dibahas beberapa elemen multimedia diatas secara lebih detail beserta contoh penerapannya.

4. Layout HTML5

Dalam bab ini kita akan membahas tentang cara membuat struktur halaman (layout) HTML menggunakan semantic tag HTML5. Sebelum kita belajar tentang layout HTML5 secara keseluruhan, terlebih dahulu kita perlu mempelajari bagian-bagian dari layout HTML5 secara terpisah. Setelah itu, pada bagian akhir dari bab ini kita akan belajar menggabungkan seluruh bagian-bagian layout HTML5 tersebut sekaligus menambahkan CSS untuk mengatur tata letak dan tampilan layout HTML5.

4.1. Div

Sudah menjadi standar di kalangan designer web untuk menggunakan tag <div> dalam merancang tampilan layout HTML5. Tag <div> adalah tag HTML yang tidak bermakna, atau dapat juga disebut sebagai tag “*generic*” yang dapat digunakan untuk keperluan apa saja. Tag <div> bertipe *block level element*, sehingga cocok digunakan untuk membagi layout halaman. Sebagai contoh, untuk membuat bagian header, tempat logo dan menu navigasi, kita bisa menggunakan kode HTML sebagai berikut:

```
<div>
  
<ul>
  <li><a href="index.html">Home</a></li>
  <li><a href="tentang-kami.html">Tentang Kami</a></li>
  <li><a href="kontak.html">Kontak</a></li></ul>
</div>
```

4.2. Header

Tag <header> ditujukan untuk bagian header. Tag ini berfungsi sebagai penanda bagian atas atau head dari sebuah halaman atau artikel. Tag ini cocok untuk menggantikan tag <div id="header">. Berikut adalah contoh kode HTMLnya:

```
<header>
<div id="logo">
  
</div>
<div id="menu_navigasi">
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="tentang-kami.html">Tentang Kami</a></li>
    <li><a href="kontak.html">Kontak</a></li></ul>
  </ul>
</div>
</header>
```

Selain itu, tag <header> juga dapat digunakan untuk bagian lain. Misalnya, dalam sebuah artikel, judul artikel kadang juga memiliki sub judul, dan kemudian diikuti dengan meta-data seperti tanggal artikel, kategori artikel, tag dan nama penulis. Seluruh bagian judul ini juga sesuai untuk tempat tag <header>. Berikut contoh penggunaannya:

```
<header>
<h1>Tutorial Memasak Mie Goreng</h1>
<h2>menu khas sajian anak kost</h2>
<time>15 April 2019</time>
  <span>Kategori: masakan</span>
  <span>Tag: anak kost, mie, telur</span>
```

```
<span>Penulis: Hardiansyah Putra</span>
</header>
```

Selain itu, penggunaan tag `<header>` juga dapat digunakan untuk bagian yang membutuhkan pembagian struktur. Bukan tidak mungkin di bagian footer dari sebuah website juga dibuat menggunakan tag `<header>`, sebagai tempat untuk judul dari bagian footer.

4.3. Section

Tag `<section>` digunakan untuk membuat sebuah bagian di dalam struktur halaman. Tag ini sebenarnya masih terkesan generik karena penggunaannya tidak hanya untuk bagian tertentu. Sebagai contoh, tag ini bisa digunakan untuk ‘membungkus’ bagian konten yang terdiri dari beberapa artikel, atau digunakan untuk memisahkan bagian sidebar. Sebagai contoh, kita bisa mengganti penggunaan tag `<div id="sidebar_1">` dengan tag `<section>`. Dengan demikian sidebar terdiri dari beberapa section:

```
<div id="sidebar">
  <section>
    <h1>Judul Sidebar 1</h1>
    <div id="isi_sidebar_1">
      <p>...isi sidebar 1...</p>
    </div>
  </section>
  <section>
    <h1>Judul Sidebar 2</h1>
    <div id="isi_sidebar_2">
      <p>...isi sidebar 2 ...</p>
    </div>
  </section>
</div>
```

Salah satu catatan dari spesifikasi HTML5, bahwa tag `<section>` bukanlah sebagai pengganti dari tag `<div>`. Jika kita ingin membuat struktur yang hanya berfungsi untuk ‘tempat’ CSS, tetap gunakan tag `<div>`.

4.4. Article

Sesuai dengan namanya, tag `<article>` digunakan untuk struktur artikel. Tag ini bisa digunakan untuk menggantikan bagian `<div id="artikel">`. Contohnya seperti berikut ini:

```
<section>
  <article>
    <header>
      <h1>Judul Artikel Tulis Disini</h1>
    </header>
    <p>....Tulis isi artikel disini...</p>
```

```
<p>....Tulis isi artikel disini...</p>
<p>....Tulis isi artikel disini...</p>
</article>
</section>
```

Dalam HTML5, tag `<article>` juga dapat digunakan untuk komentar user, sidebar dan bagian dari konten yang bisa berdiri sendiri. Pada awal perkenalannya, tag `<section>` dan tag `<article>` sering menjadi perdebatan. Kapan sebaiknya menggunakan tag `<section>` dan kapan menggunakan tag `<article>`. Secara garis besar, tag `<article>` adalah tag `<section>` untuk tujuan yang lebih spesifik. Jika bagian dari konten bisa berdiri sendiri dan tidak tergantung dengan bagian lain, maka gunakan tag `<article>`. Namun jika konten tersebut adalah bagian dari struktur yang lebih besar, gunakan tag `<section>`. Kedua tag ini juga bisa digunakan secara bersama-sama. Sebagai contoh, untuk artikel yang panjang, kita bisa memecahnya menjadi beberapa section. Berikut contoh kode HTML-nya:

```
<article>
  <h1>Jenis-Jenis Hewan </h1>
  <p>Berdasarkan jenis makanannya, hewan dibagi menjadi beberapa jenis:</p>
  <section>
    <h2>Hewan Herbivora</h2>
    <p>Penjelasan tentang hewan herbivora dijelaskan disini</p>
  </section>
  <section>
    <h2>Hewan Karnivora</h2>
    <p>Penjelasan tentang hewan karnivora dijelaskan disini</p>
  </section>
  <section>
    <h2>Hewan Omnivora</h2>
    <p>Penjelasan tentang hewan omnivora dijelaskan disini</p>
  </section>
</article>
```

4.5. Nav

Tag `<nav>` digunakan untuk mengelompokkan beberapa link yang membuat sebuah menu navigasi. Dengan kata lain, `<nav>` cocok digunakan sebagai “*container*” untuk menu. Berikut ini contoh penggunaan `<nav>`:

```
<nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="tentang-kami.html">Tentang Kami</a></li>
    <li><a href="kontak.html">Kontak</a></li>
  </ul>
</nav>
```

4.6. Aside

Tag `<aside>` digunakan untuk menyatakan bagian konten yang bukan merupakan konten utama, namun masih berhubungan. Dalam implementasinya, tag `<aside>` paling jelas digunakan untuk bagian sidebar. Berikut ini contoh penggunaan `<aside>`:

```
<aside>
  <section>
    <h1>Judul Sidebar 1</h1>
    <div id="isi_sidebar_1">
      <p>...isi sidebar 1...</p>
    </div>
  </section>
  <section>
    <h1>Judul Sidebar 2</h1>
    <div id="isi_sidebar_2">
      <p>...isi sidebar 2...</p>
    </div>
  </section>
  <section>
    <h1>Judul Sidebar 3</h1>
    <div id="isi_sidebar_3">
      <p>...isi sidebar 3...</p>
    </div>
  </section>
</aside>
```

Selain untuk sidebar, tag `<aside>` juga bisa diletakkan di dalam konten utama, yang ditujukan untuk memberi keterangan tambahan.

4.7. Main

Tag `<main>` digunakan untuk menandai bagian mana yang berperan sebagai konten utama. Dalam setiap halaman sebaiknya hanya terdapat 1 tag `<main>`. Web browser khusus seperti screen reader yang digunakan untuk pengguna disabilitas bisa mengabaikan konten lain dan langsung masuk ke bagian `<main>`. Dalam contoh kita, tag ini bisa digunakan sebagai pengganti tag `<div id="konten_utama">`

```
<main>
  <article>
    <header>
      <h1>Judul Artikel</h1>
      <p>Tanggal posting</p>
    </header>
    <section>
      <p>...isi artikel...</p>
      <p>...isi artikel...</p>
    </section>
  </article>
</main>
```

```
<p>...isi artikel...</p>
  </section>
<footer>
  <p>Ditulis oleh...</p>
  </footer>
</article>
</main>
```

4.8. Footer

Fungsi tag ini cukup jelas, dimana tag <footer> digunakan untuk menampung bagian footer dari sebuah halaman website. Berikut contohnya:

```
<footer>
<p>2019 All Rights Reserved.</p>
</footer>
```

Sama seperti tag <header>, tag <footer> tidak hanya digunakan untuk bagian akhir halaman, namun juga untuk bagian akhir dari artikel. Biasanya bagian ini berisi biodata tentang penulis artikel dan link-link yang berkaitan dengan artikel yang dibahas. Seperti contoh berikut ini:

```
<article>
  <header>
    <h1>Judul Artikel</h1>
    <p>Tanggal posting</p>
  </header>
  <section>
    <p>...isi artikel...</p>
    <p>...isi artikel...</p>
    <p>...isi artikel...</p>
  </section>
  <footer>
    <p>Ditulis oleh...</p>
  </footer>
</article>
```

Seringkali kita melihat dalam footer sebuah website terdapat link navigasi seperti Disclaimer, Term of User, Privacy Policy, dan lain-lain. Kita juga dapat menambahkan navigasi. Berikut ini adalah contoh kode footer dengan tambahan navigasi:

```
<footer>
  <div>
    <p> 2019. All Rights Reserved.</p>
  <nav>
    <ul>
```



```

        <li><a href="link-disclaimer.html">Disclaimer</a></li>
        <li><a href="link-term-of-use.html">Term of User</a></li>
        <li><a href="link-privacy-policy.html">Privacy Policy</a></li>
        <li><a href="link-tentang-kami.html">About Us</a></li>
    </ul>
</nav>
</div>
</footer>

```

4.9. Membuat Layout HTML5

Sebelumnya kita telah mempelajari bagian-bagian dari layout HTML5. Kali ini kita akan mempelajari bagaimana mengimplementasikan dan menggabungkan seluruh elemen-elemen tersebut menjadi satu kesatuan layout sebuah website. Untuk menjadikan itu kesatuan layout HTML5 memiliki tampilan yang menarik dan tertata, diperlukan CSS. CSS akan kita pelajari pada bab lain dalam buku ini.

4.9.1. Menggabungkan Struktur Elemen HTML5

Kali ini kita akan belajar menggabungkan elemen-elemen layout HTML5 menjadi sebuah kesatuan atau serangkaian kode dalam website. Berikut ini adalah contoh gabungan elemen-elemen HTML5 dari contoh yang telah dibahas sebelumnya:

```

<!DOCTYPE html>
<html>
<head>
<head>
  <meta charset="UTF-8">
  <title>Gabungan Kode Layout HTML5</title>
</head>
<body>
<div id="container">
<header>
  <div id="logo">
  </div>
  <nav>
  <ul>
    <li><a href="index.html">Home</a></li>
    <li><a href="tentang-kami.html">Tentang Kami</a></li>
    <li><a href="kontak.html">Kontak</a></li>
  </ul>
  </nav>
</header>

```

```

<main>
  <article>
    <header>
      <h1>Judul Artikel</h1>
      <p>Tanggal posting</p>
    </header>
    <section>
      <p>...isi artikel...</p>
      <p>...isi artikel...</p>
      <p>...isi artikel...</p>
    </section>
    <footer>
      <p>Ditulis oleh...</p>
    </footer>
  </article>
</main>

<aside>
  <section>
    <h1>Judul Sidebar 1</h1>
    <p>...isi sideabar 1...</p>
  </section>
  <section>
    <h1>Judul Sidebar 2</h1>
    <p>...isi sideabar 2...</p>
  </section>
  <section>
    <h1>Judul Sidebar 3</h1>
    <p>...isi sideabar 3...</p>
  </section>
</aside>

<footer>
  <div>
    <p> 2019. All Rights Reserved.</p>
    <nav>
      <ul>
        <li><a href="link-disclaimer.html">Disclaimer</a></li>
        <li><a href="link-term-of-use.html">Term of User</a></li>
        <li><a href="link-privacy-policy.html">Privacy Policy</a></li>
        <li><a href="link-tentang-kami.html">About Us</a></li>
      </ul>
    </nav>
  </div>

```

```
</div>
</footer>
</div>
</body>
</html>
```

Simpan kode diatas dalam file berformat .html. Jika file html tersebut dibuka dalam browser, maka akan memberikan tampilan seperti berikut ini:

- [Home](#)
- [Tentang Kami](#)
- [Kontak](#)

Judul Artikel

Tanggal posting

...isi artikel...

...isi artikel...

...isi artikel...

Ditulis oleh...

Judul Sidebar 1

....isi sidebar 1...

Judul Sidebar 2

....isi sidebar 2...

Judul Sidebar 3

....isi sidebar 3...

2019. All Rights Reserved.

- [Disclaimer](#)
- [Term of User](#)
- [Privacy Policy](#)
- [About Us](#)

Tampilan website diatas tentu saja tidak menarik dan tidak tertata dengan sebagaimana mestinya. Untuk melakukan penataan elemen-elemen layout HTML5, diperlukan CSS. Pada subbab selanjutnya kita akan belajar menambahkan CSS dalam layout HTML5 yang kita buat.

4.9.2. Menambahkan CSS dalam Layout HTML5

Kali ini kita akan menambahkan internal CSS dalam kode layout HTML5 yang telah kita buat sebelumnya. Internal CSS adalah kode CSS yang ditambahkan dalam satu file yang sama dengan

file HTMLnya. Cara menambahkan internal CSS adalah dengan menuliskan kode CSS diantara tag <style></style> dalam tag <head> pada HTML5.

```
<!DOCTYPE html>
<html>
<head>
<head>
  <meta charset="UTF-8">
  <title>Gabungan Kode Layout HTML5</title>
  <style>
  h1, p {
    margin: 0px;
  }
  #container {
    width: 960px;
    margin: 10px auto;
    background-color: #f2f2f2;
    height: 620px;
  }
  #container>header{
    width: 940px;
    float: left;
    margin: 10px;
    background-color: #d5d2d2;
    height: 150px;
  }
  #logo {
    width: 200px;
    height: 80px;
    float: left;
    background-color: #adacac;
    margin: 10px;
    clear: both;
  }
  nav {
    float: left;
    background-color: #adacac;
    margin: 0 10px;
    clear: both;
  }
  nav ul {
    list-style: none;
    margin: 0;
    padding: 0;
```

```

}
nav ul li {
  float: left;
}
nav ul a {
  text-decoration: none;
  color: black;
  padding: 10px 20px;
  height: 20px;
  display: block;
}
nav ul a:hover {
  color: white;
  background-color: #838181;
}
main{
  width: 700px;
  float: left;
  margin: 0px 10px 10px 10px;
  background-color: #d5d2d2;
  height: 370px;
}
article {
  width: 680px;
  float: left;
  margin: 10px;
  background-color: #adacac;
  height: 350px;
}
article header, article section, article footer {
  background-color: #838181;
  width: 640px;
  padding: 10px;
  margin: 10px 10px 0 10px;
  float: left;
  clear: both;
}
article header{
  margin-top: 20px;
}
aside{
  width: 230px;
  float: left;

```

```

background-color: #d5d2d2;
height: 370px;
}
aside section {
background-color: #838181;
width:190px;
padding: 10px;
margin: 10px 10px 0 10px;
height: 90px;
float: left;
clear: both;
}
footer {
width: 940px;
float: left;
margin: 0px 10px 10px 10px;
background-color: #d5d2d2;
height: 60px;
}
footer div {
margin: 30px 10px;
}
footer p {
float: left
}
footer nav {
float: right;
margin: -40px 0 0 0;
}
</style>
</head>
<body>
<div id="container">
<header>
<div id="logo">
</div>
<nav>
<ul>
<li><a href="index.html">Home</a></li>
<li><a href="tentang-kami.html">Tentang Kami</a></li>
<li><a href="kontak.html">Kontak</a></li>
</ul>
</nav>

```

```

</header>
<main>
  <article>
    <header>
      <h1>Judul Artikel</h1>
      <p>Tanggal posting</p>
    </header>
    <section>
      <p>...isi artikel...</p>
      <p>...isi artikel...</p>
      <p>...isi artikel...</p>
    </section>
    <footer>
      <p>Ditulis oleh...</p>
    </footer>
  </article>
</main>
<aside>
  <section>
    <h1>Judul Sidebar 1</h1>
    <p>...isi sideabar 1...</p>
  </section>
  <section>
    <h1>Judul Sidebar 2</h1>
    <p>...isi sideabar 2...</p>
  </section>
  <section>
    <h1>Judul Sidebar 3</h1>
    <p>...isi sideabar 3...</p>
  </section>
</aside>
<footer>
  <div>
    <p> 2019. All Rights Reserved.</p>
    <nav>
      <ul>
        <li><a href="link-disclaimer.html">Disclaimer</a></li>
        <li><a href="link-term-of-use.html">Term of User</a></li>
        <li><a href="link-privacy-policy.html">Privacy Policy</a></li>
        <li><a href="link-tentang-kami.html">About Us</a></li>
      </ul>
    </nav>
  </div>

```

```
</footer>
</div>
</body>
</html>
```

Bila kode diatas ditampilkan dalam browser, maka akan memberikan tampilan seperti berikut ini:



5. Dasar-Dasar CSS

CSS adalah singkatan dari Cascading Style Sheets, sebuah bahasa yang menggambarkan gaya dokumen HTML. CSS menjelaskan bagaimana elemen HTML harus ditampilkan. CSS dapat menghemat banyak pekerjaan karena dapat mengontrol tata letak beberapa halaman website sekaligus. Bab ini akan menjelaskan kepada anda dasar-dasar CSS.

Kode CSS dapat ditulis dalam file yang terpisah dengan HTML dengan format .css atau juga dapat ditulis dalam satu file yang sama dengan file HTML. Pertama-tama kita akan belajar CSS sederhana yang masih tergabung dalam satu file html. Untuk menuliskan kode CSS dalam file HTML dapat dilakukan dengan elemen <style>. Untuk lebih jelasnya, silahkan anda lihat contoh berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-color: lightblue;
}
h1 {
  color: white;
  text-align: center;
}
p {
  font-family: times new roman;
  font-size: 22px;
}
</style>
</head>
<body>
  <h1>CSS Pertamaku</h1>
  <p>Ini adalah tulisan dengan tipe paragraf.</p>
</body>
</html>
```

Berikut ini adalah tampilan dari kode diatas:

CSS Pertamaku

Ini adalah tulisan dengan tipe paragraf.

5.1. Syntax CSS

Sebuah aturan CSS terdiri dari sebuah selector dan sebuah declaration block. Misalnya saja pada CSS berikut ini:

```
h1 {  
  color: white;  
  text-align: center;  
}
```

Arti dari kode diatas adalah: temukan seluruh tag <h1> di dalam HTML, kemudian ubah warnanya menjadi putih, dan letak konten berada di tengah (center).

H1 disebut sebagai selector, dan semua yang ada dalam tanda kurung adalah *declaration block*. Selector digunakan untuk mencari/menentukan bagian mana dari HTML yang ingin di-style. *Declaration block* terdiri dari satu atau lebih *declaration* yang dipisahkan dengan tanda titik koma. Setiap declaration terdiri dari nama property CSS dan Value yang dipisahkan dengan titik dua. Setiap deklarasi CSS selalu diakhiri dengan tanda titik dua.

5.2. Selector CSS

Selector CSS digunakan untuk menemukan atau memilih elemen HTML berdasarkan nama elemen, id, class, atribut, atau yang lainnya.

1. Selector Elemen

Selector elemen memilih elemen berdasarkan nama elemen. Sebagai contoh, anda dapat memilih seluruh elemen <p> dalam HTML kemudian membuat seluruh teks dalam elemen <p> memiliki align center dan warna teks biru. Untuk melakukan hal tersebut, cukup dengan menuliskan kode css sebagai berikut ini:

```
p {  
  text-align: center;  
  color: blue;  
}
```

2. Selector Id

Selector id digunakan untuk memilih atribut id dalam HTML. Id elemen harus memiliki nama yang unik. Sehingga selector id digunakan untuk memilih elemen yang unik. Untuk memilih elemen dengan id yang spesifik, gunakan karakter hash (#) yang diikuti dengan nama id elemen. Berikut ini adalah contoh untuk melakukan seleksi pada elemen dengan id="gala1".

```
#gala1 {
  text-align: center;
  color: blue;
}
```

3. Selector Class

Selector class digunakan untuk memilih elemen dengan atribut class yang spesifik. Untuk memilih elemen dengan class yang spesifik, gunakan karakter titik yang diikuti dengan nama classnya. Contoh:

```
.center {
  text-align: center;
  color: blue;
}
```

Anda juga dapat menentukan bahwa hanya elemen HTML tertentu dengan class tertentu yang dipilih. Sebagai contoh, hanya elemen <p> dengan class="center" yang dipilih untuk diberi text dengan warna biru dan align center.

```
p.center {
  text-align: center;
  color: blue;
}
```

Dalam sebuah elemen HTML juga dapat merujuk ke lebih dari satu class. Sebagai contoh, dibawah ini adalah elemen <p> yang merujuk pada class="center" dan class="large".

```
<p class="center large">Paragraf ini merujuk pada dua kelas, yaitu kelas center dan large.</p>
```

Menggabungkan selector

Jika anda memiliki beberapa elemen dengan style yang sama, maka anda dapat melakukan penggabungan selector. Misalnya anda memiliki beberapa selector berbeda yang memiliki declaration yang sama seperti contoh berikut ini:

```

h2 {
  text-align: center;
  color: blue;
}
h3 {
  text-align: center;
  color: blue;
}
p {
  text-align: center;
  color: blue;
}

```

Maka anda dapat menggabungkan ketiga selector tersebut untuk meminimalisir penulisan code menjadi seperti berikut ini:

```

h2, h3, p {
  text-align: center;
  color: blue;
}

```

5.3. Cara menggunakan CSS

Terdapat tiga cara penggunaan CSS, yaitu:

- Inline
- Internal
- Eksternal

Berikut ini adalah pembahasan dari ketiga cara tersebut:

1. Inline Style

Seperti namanya, cara pertama untuk menggunakan CSS adalah dengan menuliskannya dalam satu baris yang sama dalam HTML. Berikut ini adalah contoh merubah warna dan margin kiri dari elemen <h1> dalam satu baris HTML yang sama.

```
<h1 style="color:red;margin-left:25px;">Ini adalah heading 1</h1>
```

2. Internal Style Sheet

Cara kedua adalah dengan menuliskan kode CSS pada satu file HTML yang sama namun berbeda baris. Hal ini dapat dilakukan dengan meletakkan kode CSS dalam elemen <style>.

```
<head>
<style>
body {
  background-color: linen;
}

h1 {
  color: maroon;
  margin-left: 35px;
}
</style>
</head>
```

3. Eksternal Style Sheet

Dengan Eksternal Style Sheet, anda dapat mengubah tampilan seluruh situs web dengan mengubah hanya satu file. Setiap halaman harus menyertakan referensi ke file style sheet eksternal di dalam elemen <link>. Elemen <link> masuk ke dalam bagian <head>: Berikut ini adalah contoh pemanggilan file eksternal style sheet ke dalam file html.

```
<head>
<link rel="stylesheet" type="text/css" href="mystyle.css">
</head>
```

Eksternal style sheet dapat ditulis dalam editor teks apa pun. File tidak boleh mengandung tag html apa pun. File lembar gaya harus disimpan dengan ekstensi .css.

Berikut ini adalah tampilan "mystyle.css"

```
body {
  background-color: lightblue;
}

h1 {
  color: navy;
  margin-left: 20px;
}
```









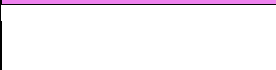

5.4. Memberi Warna dalam CSS

Sebelum kita melakukan pengaturan warna dalam web dengan CSS, terlebih dahulu ada baiknya jika kita mempelajari nama-nama warna yang dapat dituliskan dalam kode CSS.

5.4.1. Nama-Nama Warna dalam CSS

Terdapat 140 nama warna yang didukung oleh HTML5. Beberapa diantaranya adalah sebagai berikut ini:

Nama Warna	Kode Heksadesimal	Warna
<u>Aqua</u>	<u>#00FFFF</u>	
<u>Black</u>	<u>#000000</u>	
<u>Blue</u>	<u>#0000FF</u>	
<u>Brown</u>	<u>#A52A2A</u>	
<u>Chocolate</u>	<u>#D2691E</u>	
<u>DeepPink</u>	<u>#FF1493</u>	
<u>Fuchsia</u>	<u>#FF00FF</u>	
<u>Gold</u>	<u>#FFD700</u>	
<u>Indigo</u>	<u>#4B0082</u>	
<u>Ivory</u>	<u>#FFFFFF0</u>	
<u>Khaki</u>	<u>#F0E68C</u>	
<u>Lavender</u>	<u>#E6E6FA</u>	
<u>Maroon</u>	<u>#800000</u>	
<u>Olive</u>	<u>#808000</u>	
<u>OliveDrab</u>	<u>#6B8E23</u>	
<u>Orange</u>	<u>#FFA500</u>	
<u>OrangeRed</u>	<u>#FF4500</u>	
<u>Orchid</u>	<u>#DA70D6</u>	
<u>PaleGoldenRod</u>	<u>#EEE8AA</u>	
<u>PaleGreen</u>	<u>#98FB98</u>	

<u>PaleTurquoise</u>	<u>#AFEEEE</u>	
<u>PaleVioletRed</u>	<u>#DB7093</u>	
<u>PapayaWhip</u>	<u>#FFefd5</u>	
<u>Pink</u>	<u>#FFC0CB</u>	
<u>Red</u>	<u>#FF0000</u>	
<u>Silver</u>	<u>#C0C0C0</u>	
<u>Tomato</u>	<u>#FF6347</u>	
<u>Violet</u>	<u>#EE82EE</u>	
<u>White</u>	<u>#FFFFFF</u>	
<u>Yellow</u>	<u>#FFFF00</u>	

Selain nama-nama warna diatas, masih ada banyak nama warna lainnya yang didukung HTML5 dan CSS3. Selain menggunakan nama warna, kita juga dapat memberikan masukan berupa kode heksadesimal dalam memberikan pewarnaan. Kode heksadesimal tersebut memiliki format #RRGGBB. Diawali dengan tanda pagar, kemudian diikuti dengan nominal angka merah dua angka (red-red), hijau dua angka (green-green), dan biru dua angka (blue-blue).

5.4.2. Memberi Warna Tulisan

Berikut ini adalah contoh cara memberikan warna pada tulisan.

```
<h1 style="color:Tomato;">Halo dunia</h1>
<p style="color:Red;">Ini adalah warna merah</p>
<p style="color:Indigo;">Ini adalah warna indigo</p>
```

Jika kode diatas dijalankan maka akan memberikan tampilan seperti di bawah ini:

Halo dunia

Ini adalah warna merah

Ini adalah warna indigo

5.4.3. Memberi Warna Border

Untuk memberikan warna pada border, dapat dilakukan dengan cara berikut ini:

```
<h1 style="border:3px solid Brown;">Halo dunia warna Brown</h1>  
<h1 style="border:3px solid Maroon;">Halo dunia warna Maroon</h1>  
<h1 style="border:3px solid Gold;">Halo dunia warna Gold</h1>
```

Jika kode diatas dijalankan maka akan memberikan tampilan seperti di bawah ini:

Halo dunia warna Brown

Halo dunia warna Maroon

Halo dunia warna Gold

6. Property dalam CSS

Terdapat banyak sekali property dalam CSS3. Pada bab ini, kita akan membaginya menjadi beberapa subbab:

- Property Background
- Property Text Style
- Property Border
- Property Margin
- Property Padding
- Property Page layout
- Property Interface

Mari kita bahas satu per satu secara singkat.

6.1. Property Background CSS

Property background dalam CSS digunakan untuk menentukan efek latar belakang untuk elemen. Beberapa macam property background dalam CSS adalah:

- background-color
- background-image
- background-repeat
- background-position
- background-attachment

A. Background Color

Property background-color digunakan untuk memberi background warna pada sebuah elemen. Berikut ini adalah contoh penggunaan property background-color dalam CSS:

```
h1 {
  background-color: SlateBlue;
}
div {
  background-color: Violet;
}
p {
  background-color: yellow;
}
```

Untuk melihat hasilnya, CSS tersebut perlu dipanggil dalam sebuah file HTML. Berikut ini adalah contoh lengkap beserta kode HTMLnya:

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
h1 {
  background-color: SlateBlue;
}
div {
  background-color: Violet;
}
p {
  background-color: yellow;
}
</style>
</head>
<body>
<h1>Contoh background-color dalam CSS</h1>
  <div>
    Ini adalah teks dalam elemen div.
    <p>Ini adalah paragraf dalam elemen p yang juga berada dalam elemen
    div.</p>
    Ini juga masih termasuk dalam elemen div
  </div>
</body>
</html>

```

Kode diatas jika dijalankan maka akan memberikan tampilan seperti berikut ini:

Contoh background-color dalam CSS

Ini adalah teks dalam elemen div.

Ini adalah paragraf dalam elemen p yang juga berada dalam elemen div.

Ini juga masih termasuk dalam elemen div

B. Background Image

Property background-image digunakan untuk memberikan background berupa gambar dalam sebuah elemen. Dengan otomatis, gambar akan di repeat hingga menutupi seluruh bagian elemen. Contoh penerapannya adalah sebagai berikut ini:

```

body {
  background-image: url("kertas.gif");
}

```

Contoh penggunaannya adalah sebagai berikut ini:

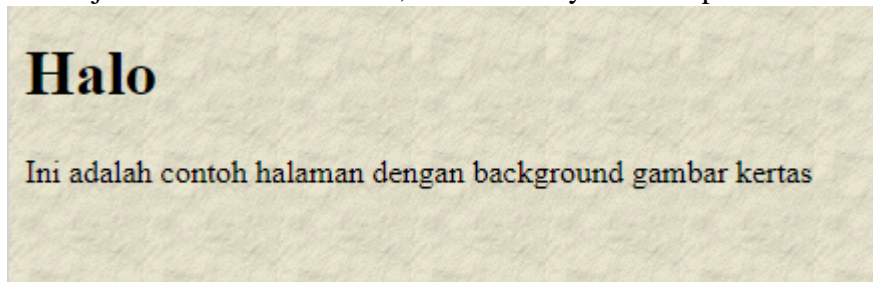
```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("kertas.gif");
}
</style>
</head>
<body>

<h1>Halo</h1>

<p>Ini adalah contoh halaman dengan background gambar kertas</p>

</body>
</html>
```

Bila dijalankan dalam browser, maka hasilnya akan seperti ini:



C. Background Repeat

Background-repeat adalah property dalam CSS yang berfungsi untuk mengulang pola background dalam halaman web. Terdapat tiga jenis value untuk background repeat:

- Repeat-x untuk mengulang background ke arah horisontal
- Repeat-y untuk mengulang background ke arah vertikal
- No repeat untuk membuat background hanya tampil satu kali

1. No Repeat

Ini digunakan agar background tidak mengalami perulangan.

```
body {
  background-image: url("gradien.png");
  background-repeat: no-repeat;
}
```

Berikut ini adalah kode lengkapnya:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
  background-image: url("gradien.png");
  background-repeat: no-repeat;

}
</style>
</head>
<body>

<h1>Halo</h1>
<p>ini adalah halaman dengan background no repeat </p>

</body>
</html>
```



Halo

ini adalah halaman dengan background no repeat

2. Repeat Horizontal

Ini digunakan untuk melakukan perulangan background ke arah horisontal.

```
body {
  background-image: url("gradien.png");
  background-repeat: repeat-x;
}
```

Contoh:

```
<!DOCTYPE html>
```

```

<html>
<head>
<style>
body {
  background-image: url("gradient_bg.png");
  background-repeat: repeat-x;
}
</style>
</head>
<body>

<h1>Halo</h1>
<p>ini adalah halaman dengan background repeat horisontal </p>

</body>
</html>

```



Halo

ini adalah halaman dengan background repeat horisontal

3. Repeat Vertikal

Ini digunakan untuk melakukan perulangan background ke arah vertikal. Berikut ini adalah contoh kode CSS nya:

```

body {
  background-image: url("gradien.png");
  background-repeat: repeat-y;
}

```

Berikut ini adalah contoh kode lengkap HTML beserta CSS nya:

```

<!DOCTYPE html>
<html>
<head>
  <style>
    body {
      background-image: url("gradien.png");
      background-repeat: repeat-y;
    }
  </style>

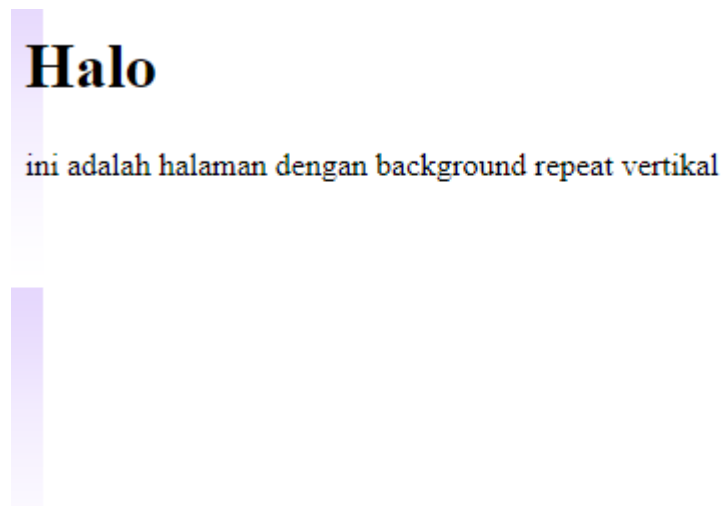
```

```

    </style>
</head>
<body>
    <h1>Halo</h1>
    <p>ini adalah halaman dengan background repeat vertikal </p>
</body>
</html>

```

Kode diatas bila ditampilkan dalam browser maka akan menjadi seperti ini:



Dari tampilan diatas terlihat bahwa gambar background gradien berulang ke bawah namun tidak ke samping.

D. Background Position

Property background position digunakan untuk mengatur posisi background agar sesuai dengan yang diinginkan. Sebagai contohnya, berikut ini adalah background berupa gambar pohon yang ingin diletakkan di sisi kanan atas halaman.

```

<!DOCTYPE html>
<html>
<head>
<style>
body {
    background-image: url("pohon.png");
    background-repeat: no-repeat;
    background-position: right top;
    margin-right: 250px;
}
</style>

```

```

</head>
<body>

<h1>Assalamualaikum</h1>
<p>Ini adalah contoh halaman web yang menggunakan property background
position.</p>
<p>Saat ini, gambar background hanya muncul satu kali di sisi kanan atas halaman.
</p>
<p>Dalam contoh ini kita juga dapat menambahkan margin di sisi kanan, sehingga
gambar background tidak akan mengganggu teks.</p>

</body>
</html>

```

Assalamualaikum

Ini adalah contoh halaman web yang menggunakan property background position.

Saat ini, gambar background hanya muncul satu kali di sisi kanan atas halaman.

Dalam contoh ini kita juga dapat menambahkan margin di sisi kanan, sehingga gambar background tidak akan mengganggu teks.



E. Background Attachment

Background attachment ini gunanya untuk membuat agar background berada pada posisi yang tetap (tidak bergerak) ketika kita melakukan scrolling pada browser kita.

```

body{
  background-image: url("pohon.png");
  background-repeat: no-repeat;
  background-position: right top;
  background-attachment: fixed;
}

```

6.2. Property Text Style

Property Text Style bisa digunakan untuk membuat gaya tulisan pada website. Berikut ini adalah beberapa property CSS yang berhubungan dengan Text Style:

- Font weight
Property ini digunakan untuk mengontrol ketebalan text. Contoh:
Selector {font-weight: 100}

Value bisa anda tulis dengan angka mulai dari seratus dan dilanjutkan dengan kelipatannya atau bisa anda tulis dengan kata 'Bold' atau yang lain.

- Font family

Property ini untuk mengatur font yang anda pakai. Contoh:

Selector {font-family: Arial, Helvetica, sans-serif}

Untuk jenis font sebaiknya menggunakan standard bawaan Windows, jika ingin terbaca oleh seluruh user.

- Font size

Property ini untuk mengatur ukuran font teks. Contoh:

Selector {font-size: 12px}

- Font variant

Property ini untuk mengatur font yang akan muncul apakah menggunakan huruf kecil, normal atau small caps. Contoh:

Selector {font-variant: small-caps}

- Font style

Property ini digunakan untuk mengatur style font yang digunakan dalam teks. Contoh:

Selector {font-style: italic}

- Color

Property ini digunakan untuk mengganti warna tulisan. Contoh:

Selector {color: #000000}

6.3. Property Border

Property ini digunakan untuk mengatur elemen yang mempunyai garis dengan berbagai variasi lebar, warna serta gaya. Anda bisa membuat suatu garis antara kiri-kanan-atas-bawah dengan berbeda style.

Contoh:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
  border: 5px solid red;
}
</style>
</head>
```



```
<body>
<h2>Property Border</h2>
<p> Ini adalah contoh property border.</p>
</body>
</html>
```

Bila kode diatas dijalankan, maka akan menampilkan seperti berikut ini:

Property Border

Ini adalah contoh property border.

Berikut ini propertynya:

- Border
Property ini untuk mengatur border secara keseluruhan. Contoh:
Selector {border: dotted}
Value dotted akan menghasilkan border berupa titik-titik.

Property Border

Ini adalah contoh property border.

Value lain: thin, thick, dashed, dotted, double, groove, hidden, inset, none, outset, ridge, solid

- Border-width
Property ini digunakan untuk mengatur lebar border secara keseluruhan. Contoh:
Selector {border-width: thin}
Value lain dapat diisi dengan thick atau medium.
- Border-top-width
Property ini untuk mengatur lebar border top. Contoh:
Selector {border-top-width: thin}
Value lain dapat diisi dengan thick atau medium.
- Border-right-width
Property ini untuk mengatur lebar border right. Contoh:

Selector {border-right-width: thin}

Value lain dapat diisi dengan thick atau medium.

- Border-bottom-width

Property ini digunakan untuk mengatur lebar border bottom. Contoh:

Selector {border-bottom-width: thin}

Value lain dapat diisi dengan thick atau medium.

- Border-left-width

Property ini digunakan untuk mengatur lebar border left. Contoh:

Selector {border-left-width: normal}

Value lain dapat diisi dengan thick atau medium.

- Border-color

Property ini digunakan untuk mengatur warna dari border. Contoh:

Selector {border-color: #333333}

Value lain anda bisa gunakan warna nama-nama warna atau kode heksadesimal warna yang awalnya dimulai dengan tanda pagar # kemudian diikuti oleh angka red-red-green-green-blue-blue.

- Border-style

Property ini digunakan untuk mengatur style dari border. Contoh:

Selector {border-style: normal}

Value lain dapat diisi dengan thin, thick, dashed, dotted, double, groove, hidden, inset, none, outset, ridge atau solid.

- Border-top

Property ini digunakan untuk mengatur border top. Contoh:

Selector {border-top: normal}

Value lain dapat diisi dengan thin, thick, dashed, dotted, double, groove, hidden, inset, none, outset, ridge atau solid.

- Border-right

Property ini digunakan untuk mengatur border right. Contoh:

Selector {border-right: normal}

Value lain dapat diisi dengan thin, thick, dashed, dotted, double, groove, hidden, inset, none, outset, ridge atau solid.

- Border-bottom

Property ini digunakan untuk mengatur border bagian bawah. Contoh:

Selector {border-bottom: normal}

Value lain dapat diisi dengan thin, thick, dashed, dotted, double, groove, hidden, inset, none, outset, ridge atau solid.

- Border-left

Property ini digunakan untuk mengatur border sebelah kiri. Contoh:

Selector {border-left: normal}

Value lain dapat diisi dengan thin, thick, dashed, dotted, double, groove, hidden, inset, none, outset, ridge atau solid.

6.4. Property Margin

Property Margin digunakan untuk mengatur ruang atau jarak antar elemen, misalnya antara elemen atas dan bawah, atau antara kiri dan kanan dll. Untuk lebih jelasnya, simak contoh berikut ini

```
<!DOCTYPE html>
<html>
<head>
<style>
div.contoh1 {
  width: 300px;
  background-color: yellow;
}

div.contoh2 {
  width: 300px;
  margin: 25px;
  background-color: lightblue;
}
</style>
</head>
<body>
<h2>Belajar Margin</h2>
<div class="contoh1">Ini adalah contoh elemen tanpa margin.</div>
<br>
<div class="contoh2">Ini adalah contoh elemen dengan margin 25px</div>
</body>
</html>
```

Bila kode diatas dijalankan dalam browser, maka akan muncul tampilan seperti berikut ini:

Belajar Margin

Ini adalah contoh elemen tanpa margin.

Ini adalah contoh elemen dengan margin 25px

Pada class contoh1, elemen tidak menggunakan margin sehingga tidak memiliki jarak dengan elemen lain. Pada class contoh2, memiliki margin sebesar 25px, sehingga memiliki jarak 25px baik pada sisi kanan, kiri, atas maupun bawah elemen. Berikut ini adalah beberapa macam property margin:

- Margin
Property ini digunakan untuk mengatur jarak secara keseluruhan dengan elemen lain.
Contoh:
Selector {margin: auto}
Value lain bisa ukuran px, em atau persen misalnya 30%.
- Margin-top
Property ini digunakan untuk mengatur jarak atas dengan elemen. Contoh:
Selector {margin-top: auto}
Value lain bisa ukuran px, em atau persen misalnya 30%.
- Margin-left
Property ini digunakan untuk mengatur jarak bagian kiri dengan elemen lain. Contoh:
Selector {margin-left: auto}
Value lain bisa ukuran px, em atau persen misalnya 30%.
- Margin-right
Property ini digunakan untuk mengatur jarak sebelah kanan dengan elemen lain. Contoh:
Selector {margin-right: auto}
Value lain bisa ukuran px, em atau persen misalnya 30%.
- Margin-bottom
Property ini digunakan untuk mengatur jarak bawah dengan elemen lain. Contoh:
Selector {margin-bottom: auto}
Value lain bisa ukuran px, em atau persen misalnya 30%.

6.5. Property Padding

Property padding adalah untuk mengatur ruang antara elemen dan konten. Untuk dapat memahami padding secara lebih jelas, berikut ini contohnya:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.contoh1 {
  width: 300px;
  background-color: yellow;
}

div.contoh2 {
  width: 300px;
  padding: 25px;
  background-color: lightblue;
}
</style>
</head>
<body>
<h2>Belajar Padding</h2>
<div class="contoh1">Ini adalah contoh elemen tanpa padding.</div>
<br>
<div class="contoh2">Ini adalah contoh elemen dengan padding 25px</div>
</body>
</html>
```

Bila kode diatas dijalankan maka akan menghasilkan tampilan berikut ini:

Belajar Padding

Ini adalah contoh elemen tanpa padding.

Ini adalah contoh elemen dengan padding
25px

Seperti yang dapat kita lihat pada contoh diatas, elemen yang tidak menggunakan padding tidak memiliki jarak antara elemen dengan konten. Sedangkan elemen yang menggunakan padding memiliki jarak dengan konten. Terdapat berbagai macam jenis property padding. Berikut ini adalah beberapa macam property padding:

- **Padding**
Property ini digunakan untuk mengatur ruang elemen dengan konten secara global.
Contoh:

Selector {padding: 7px}

Value lain bisa ukuran px, em atau persen misalnya 30%.

- **Padding-top**

Property ini digunakan untuk mengatur ruang atau jarak bagian atas antara konten dengan elemen. Contoh:

Selector {padding-top: 7px}

Value lain bisa ukuran px, em atau persen misalnya 30%.

- **Padding-left**

Property ini digunakan untuk mengatur ruang elemen bagian kiri dengan konten. Contoh:

Selector {padding-left: 7px}

Value lain bisa ukuran px, em atau persentase misalnya 25%

- **Padding-right**

Property ini untuk mengatur ruang elemen bagian kanan dengan konten. Contoh:

Selector {padding-right: 7px}

Value lain yang bisa digunakan adalah: em atau persentase misalnya 25%.

- **Padding-bottom**

Property ini untuk mengatur ruang elemen bagian bawah dengan konten. Contoh:

Selector {padding-bottom: 7px}

Value lain bisa ukuran px, em atau persen misalnya 25%.

6.6. Property Page Layout

Property Page Layout adalah property untuk mendesign suatu layout pada halaman web, ini property penting yang harus dikuasai karena dengan menguasai property ini maka anda dapat membuat suatu halaman web yang bagus dan tentunya harus didukung dengan kemampuan kreatifitas seseorang. HTML tidak mendukung tampilan layout sebaik CSS, dengan menguasai Property ini anda juga bisa merubah themes atau template pada script CMS.

Berikut ini propertynya:

- **Position**

Property ini digunakan untuk menentukan dimana letak suatu elemen dan juga untuk menempatkan elemen induk. Contoh:

Selector {position: absolute}

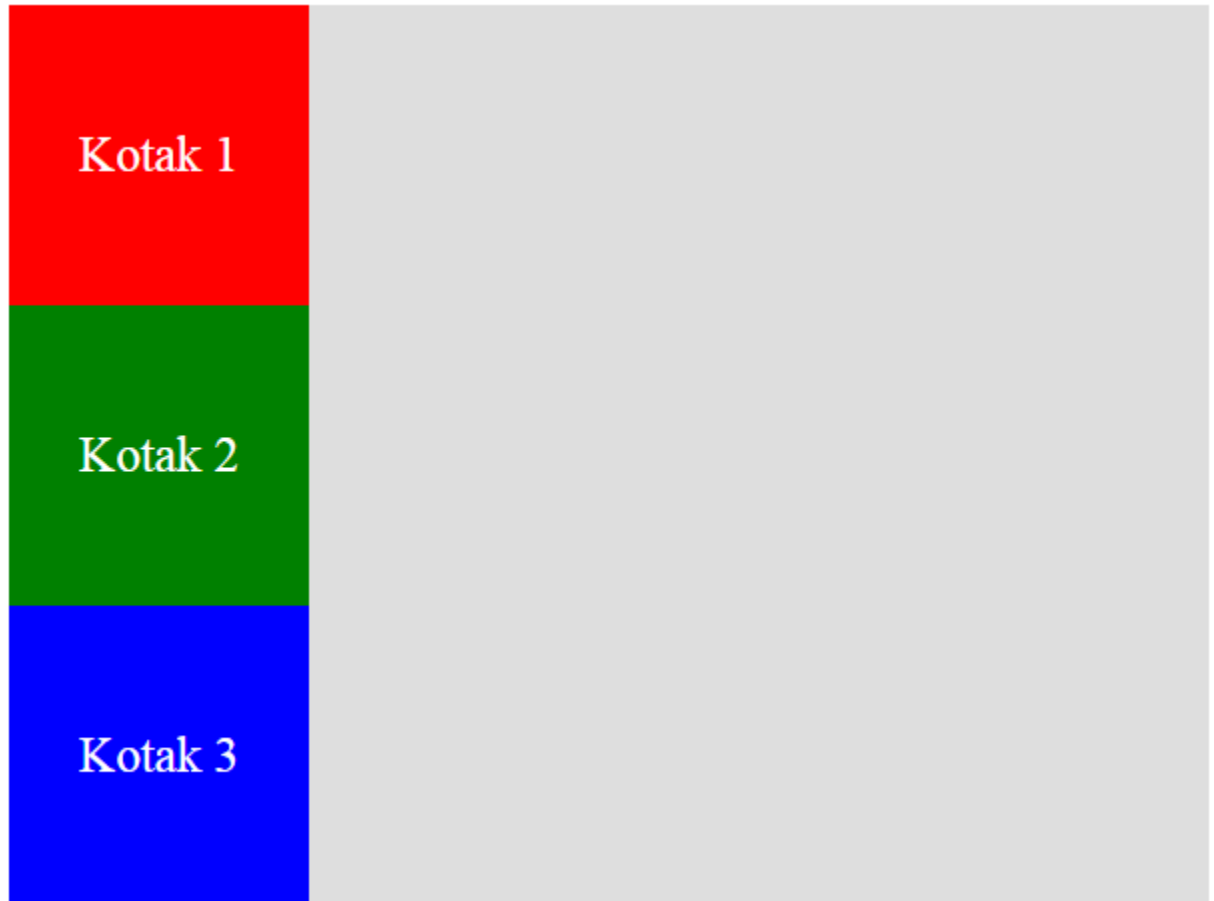
Value lain dapat diisi dengan fixed, relative, static, top, bottom, left dan right.

Nilai static untuk property position merupakan nilai *default*. Tanpa ditulis pun sebenarnya kita telah menggunakan property position: static. Penulisan property position: static juga berarti kita membuat sebuah element 'kembali' ke normal document flow. Berikut contohnya:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar CSS Position: Static</title>
  <style>
    .container {
      width: 600px;
      height: 450px;
      background-color: rgb(222, 222, 222);
      margin: 0 auto;
    }
    .satu, .dua, .tiga {
      width: 150px;
      height: 150px;
      font-size: 25px;
      line-height: 150px;
      text-align: center;
      color: white;
    }
    .satu {
      background-color: red;
      position: static;
    }
    .dua {
      background-color: green;
    }
    .tiga {
      background-color: blue;
    }
  </style>
</head>
<body>
  <div class="container">
    <div class="satu"> Kotak 1 </div>
    <div class="dua"> Kotak 2 </div>
    <div class="tiga"> Kotak 3 </div>
```

```
</div>  
</body>  
</html>
```

Bila kode diatas ditampilkan dalam browser maka akan memberikan tampilan seperti berikut ini:



- Top
Property ini digunakan untuk menentukan bagian atas dari suatu elemen ditempatkan.
Contoh:
Selector {top: absolute}
Value lain dapat diisi dengan auto, fixed, relative dan static.
- Left
Property ini digunakan untuk menentukan bagian kiri dari suatu elemen ditempatkan.
Contoh:
Selector {left: absolute}
Value lain dapat diisi dengan auto, fixed, relative dan static.

- Bottom

Property ini digunakan untuk menentukan bagian bawah dari suatu elemen ditempatkan.
Contoh:
Selector {bottom: absolute}
Value lain dapat diisi dengan auto, fixed, relative dan static.
- Right

Property ini digunakan untuk menentukan bagian kanan dari suatu elemen ditempatkan.
Contoh:
Selector {right: absolute}
Value lain dapat diisi dengan auto, fixed, relative dan static.
- Width

Property ini digunakan untuk menentukan lebar dari suatu elemen. Contoh:
Selector {width: 50%}
Value lain dapat diisi dengan satuan px (pixel), em atau auto.
- Min-width

Property ini digunakan untuk menentukan lebar minimal dari suatu elemen. Contoh:
Selector {min-width: 50%}
Value lain dapat diisi dengan none, auto, satuan px (pixel) atau em.
- Max-width

Property ini digunakan untuk menentukan lebar maksimal dari suatu elemen. Contoh:
Selector {max-width: 50%}
Value lain dapat diisi dengan satuan px (pixel), em, auto atau none.
- Height

Property ini digunakan untuk menentukan tinggi dari suatu elemen. Contoh:
Selector {height: 50%}
Value lain dapat diisi dengan satuan px (pixel), em, auto, atau none.
- Min-height

Property ini digunakan untuk menentukan tinggi minimal dari suatu elemen. Contoh:
Selector {min-height: 50%}
Value lain dapat diisi dengan px (pixel), em, auto atau none.
- Max-height

Property ini digunakan untuk menentukan tinggi maksimal dari suatu elemen. Contoh:

Selector {max-height: 50%}

Value lain dapat diisi dengan px, em, auto atau none.

- Z-index

Property ini digunakan untuk mengatur beberapa elemen yang posisinya overlap atau saling bertumpang tindih. Contoh:

Selector {z-index: auto}

Value lain dapat juga diisi dengan menggunakan bilangan bulat (integer).

- Visibility

Property ini digunakan untuk mengatur elemen apakah ditampilkan pada browser atau tidak. Contoh:

Selector {visibility: hidden}

Value lain dapat diisi dengan visible atau collapse.

- Overflow

Property ini digunakan untuk mengatur menampilkan konten yang tidak bisa ditampung oleh elemen. Contoh:

Selector {overflow: auto}

Value lain dapat diisi dengan hidden, scroll atau visible.

- Float

Property ini digunakan membuat efek elemen yang keluar dari flow dan ditempatkan disisi kiri atau kanan suatu elemen. Contoh:

Selector {float: left}

Value lain dapat diisi dengan right atau none.

- Clear

Property ini digunakan bersama dengan float dan disini ditentukan apakah elemen bisa menerima suatu float atau tidak. Contoh:

Selector {clear: both}

Value lain dapat diisi dengan left, right atau none.

- Clip

Property ini untuk mengatur cupilkan kecil suatu elemen untuk ditampilkan. Contoh:

Selector {clip: rect ('top', 'right', 'bottom', 'left')}

Value lain dapat diisi dengan both.

6.7. Property Interface

Property ini untuk mengatur tampilan pada web untuk membuat tampilan yang ditujukan untuk pengguna. Berikut ini propertynya:

- Cursor

Property ini digunakan untuk mengatur tampilan cursor yang akan dipakai oleh user pada browser. Contoh:

Selector {cursor: auto}

Value lain: url, crosshair, default, e-resize, help, move, n-resize, ne-resize, nw-resize, pointer, sw-resize, se-resize, text, wait, w-resize.

- Outline

Property ini digunakan untuk mengatur garis tepi dari elemen secara global. Contoh kodenya adalah sebagai berikut:

Selector {outline: #333333}

Untuk warna bisa menggunakan warna hexadecimal lain atau value lain seperti dashed, dotted, double, groove, hidden, inset, none, outside, ridge, solid, medium, thin atau thick.

- Outline-color

Properti ini digunakan untuk mengatur warna dari garis tepi suatu elemen. Contoh:

Selector {outline: #333333}

Untuk warna bisa menggunakan warna hexadecimal lain.

- Outline-style

Property ini digunakan untuk mengatur style dari garis tepi suatu elemen. Contoh:

Selector {outline: dashed}

Value lain dapat diisi dengan dotted, double, groove, hidden, inset, none, outside, ridge, solid, medium, thin atau thick.

- Outline-width

Property ini digunakan untuk mengatur seberapa lebar garis tepi suatu elemen. Contoh:

Selector {outline-width: medium}

Value lain dapat diisi dengan thick atau thin.

7. Membuat Tampilan Website sederhana dengan HTML5 dan CSS3

Pada bab ini kita akan belajar membuat tampilan website sederhana dengan HTML5 dan CSS3. Tampilan website yang akan kita buat akan memiliki tag HTML5 seperti header, footer, beberapa jenis navigasi (arsip, blogroll, dan tautan internal), dan tentu saja artikel atau posting.

Semua berawal dari format HTML5 berikut ini:

```
<!DOCTYPE html>
<html lang="en-US">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Judul Website</title>
  </head>
  <body>
  </body>
</html>
```

7.1. Membuat Header

Seperti yang telah kita bahas sebelumnya, bahwa dalam HTML5 tag untuk membuat header adalah <header>. Tidak perlu dibingungkan dengan h1, h2, atau h3, header dapat saja diisi dengan semua jenis konten seperti logo perusahaan, kotak pencarian, dan lain sebagainya. Namun, untuk saat ini kita contohkan hal yang sederhana dahulu yaitu hanya dengan mencantumkan judul blog saja.

```
<header id="page_header">
  <h1>Web Sederhana dengan HTML5 dan CSS3</h1>
</header>
```

Kita tidak dibatasi untuk hanya memiliki satu judul dalam satu halaman. Setiap artikel atau setiap halaman dapat memiliki header tersendiri. Oleh sebab itu, dalam contoh diatas, digunakan ID dalam header. Hal ini berguna untuk mengidentifikasi elemen header secara unik. Dengan adanya ID dalam header, dapat memudahkan kita untuk melakukan modifikasi dalam CSS maupun JavaScriptnya.

7.2. Membuat Footer

Anda pasti sebelumnya pernah melihat footer dalam sebuah situs web. Footer biasanya berisi informasi seperti tahun hak cipta dan informasi tentang siapa yang memiliki situs tersebut. Seperti yang telah kita bahas sebelumnya bahwa tag <footer> tidak hanya dipakai untuk mendefinisikan footer pada halaman website, tetapi juga dapat digunakan untuk mendefinisikan footer dalam artikel, dan lain sebagainya. Berikut ini adalah contoh footer sederhana dalam sebuah halaman website. Karena dalam sebuah website dapat memiliki banyak footer, pada contoh ini akan kami

berikan ID pada footer. Hal ini akan membantu dalam mengidentifikasi secara unik footer yang kita buat.

```
<footer id="page_footer">
  <p>&copy; 2018 HTML5 CSS3.</p>
</footer>
```

Contoh footer ini hanya berisi tahun hak cipta. Namun, seperti header, footer pada halaman website dapat mengandung elemen lain, termasuk elemen navigasi.

7.3. Membuat Navigation

Navigasi adalah hal yang sangat penting bagi keberhasilan sebuah situs web. Orang-orang tidak akan bertahan dalam website anda jika terlalu sulit bagi mereka untuk menemukan apa yang mereka cari. Jadi masuk akal jika navigasi memiliki tag HTML-nya sendiri.

Mari tambahkan bagian navigasi dalam header dokumen kita. Kita akan menambahkan tautan ke beranda blog, arsip, halaman yang mencantumkan nama kontributor dalam blog, dan tautan ke laman kontak.

```
<header id="page_header">
  <h1>Web Sederhana dengan HTML5 dan CSS3</h1>
  <nav>
    <ul>
      <li><a href="/">Latest Posts</a></li>
      <li><a href="archives">Archives</a></li>
      <li><a href="contributors">Contributors</a></li>
      <li><a href="contact">Contact Us</a></li>
    </ul>
  </nav>
  <link rel="stylesheet" href="style.css" type="text/css">
</header>
```

Seperti halnya header, halaman anda dapat memiliki banyak elemen navigasi. Biasanya anda sering mendapati di banyak situs, terdapat navigasi pada header dan footernya. Pada contoh blog yang akan kita buat kali ini, kita akan menambahkan navigasi pada bagian footernya. Berikut ini adalah contoh kode footer dengan navigasi ke Home, About, Terms of Services, dan Pivacy.

```
<footer id="page_footer">
<p>&copy; 2018 HTML5 CSS3.</p>
<nav>
  <ul>
    <li><a href="http://www.contoh.com/">Home</a></li>
    <li><a href="about">About</a></li>
```

```
        <li><a href="terms.html">Terms of Service</a></li>
        <li><a href="privacy.html">Privacy</a></li>
    </ul>
</nav>
</footer>
```

Kita akan mengubah tampilan navigasi footer yang telah kita buat dengan menggunakan CSS. Sehingga anda jangan terlalu khawatir dengan tampilan footer yang kita buat ini.

7.4. Membuat Section dan Article

Pada subbab ini, kita akan belajar membuat section dan article. Article akan diletakkan di dalam section. Berikut ini adalah cara penulisan tag section dengan id=posts:

```
<section id="posts">
</section>
```

Article adalah elemen yang cocok untuk memuat konten dalam website. Setiap artikel yang kita buat akan memiliki header, konten, dan footer. Kita dapat mendefinisikan artikel seperti berikut ini:

```
<article class="post">
<header>
    <h2>Apa saja kelebihan dari HTML5 dan CSS3?</h2>
    <p>Posted by admin on
    <time datetime="2018-10-01T14:39">October 1st, 2018 at 2:39PM</time>
    </p>
</header>
```

```
<p>
Teknologi web berkembang secara terus menerus. Perkembangan tersebut sampai saat ini membawa kita pada HTML5 dan CSS3. HTML5 dan CSS3 bukan sekedar dua stantar baru yang diusulkan Worl Wide Web Consortium. Keduanya adalah kelanjutan iterasi teknologi yang setiap hari kita gunakan. Keduanya membantu kita untuk membuat aplikasi web yang modern.
```

```
</p>
```

```
<p>
```

Sebelum kita mempelajari lebih dalam tentang HTML5 dan CSS3, ada baiknya kita mengetahui keunggulan dari HTML5 dan CSS3.

Beberapa kelebihan HTML5 dan CSS3 adalah:

- Dukungan multimedia yang lebih baik

- Tampilan yang lebih baik

- Form yang lebih baik

- Visual Efek yang baik


```
</p>
```

```
<footer>
  <p><a href="comments"><i>25 Comments</i></a> ...</p>
</footer>
</article>
```

Seperti yang telah kita bahas sebelumnya, kita dapat menggunakan elemen header dan footer di dalam artikel. Kita juga dapat membagi artikel yang kita buat menjadi beberapa section dengan elemen `<section>`.

7.5. Membuat Aside

Terkadang kita membutuhkan suatu hal yang ekstra dalam konten yang kita buat, seperti quote, diagram, dan beberapa tambahan lainnya. Anda dapat menggunakan elemen `<aside>` untuk keperluan tersebut.

```
<aside>
  <p>
    &quot;Belajar segala ilmu yang bermafaat adalah ibadah.&quot;
  </p>
</aside>
```

Contoh diatas adalah penggunaan `<aside>` untuk membuat quote yang letaknya berdekatan dengan konten utama. Secara keseluruhan, section, article dan aside bila digabung menjadi satu akan menjadi seperti berikut ini:

```
<section id="posts">
<article class="post">
<header>
  <h2>Apa saja kelebihan dari HTML5 dan CSS3?</h2>
  <p>Posted by admin on
  <time datetime="2018-10-01T14:39">October 1st, 2018 at 2:39PM</time>
  </p>
</header>
<aside>
  <p>
    &quot;Belajar segala ilmu yang bermafaat adalah ibadah.&quot;
  </p>
</aside>
<p>
```

Teknologi web berkembang secara terus menerus. Perkembangan tersebut sampai saat ini membawa kita pada HTML5 dan CSS3. HTML5 dan CSS3 bukan sekedar dua stantar baru yang diusulkan Worl Wide Web Consortium. Keduanya adalah kelanjutan iterasi teknologi yang setiap hari kita gunakan. Keduanya membantu kita untuk membuat aplikasi web yang modern.

</p>

<p>

Sebelum kita mempelajari lebih dalam tentang HTML5 dan CSS3, ada baiknya kita mengetahui keunggulan dari HTML5 dan CSS3.

Beberapa kelebihan HTML5 dan CSS3 adalah:

- Dukungan multimedia yang lebih baik

- Tampilan yang lebih baik

- Form yang lebih baik

- Visual Efek yang baik

</p>

<footer>

<p><i>25 Comments</i> ...</p>

</footer>

</article>

</section>

Sekarang, kita tinggal menambahkan sidebar.

7.6. Membuat Sidebar

Blog yang akan kita buat memiliki sidebar di sisi kanan yang berisi tautan ke arsip untuk blog. Jika Anda berpikir bahwa kami dapat menggunakan tag <aside> untuk menentukan bilah sisi blog kami, Anda salah. Anda bisa melakukannya dengan cara itu, tetapi itu bertentangan dengan semangat spesifikasi.

Selain itu dirancang untuk menunjukkan konten yang terkait dengan sebuah artikel. Ini adalah tempat yang baik untuk menampilkan tautan terkait, glosarium, atau kutipan penarikan. Untuk menandai bilah sisi kami yang berisi daftar arsip sebelumnya, kami hanya akan menggunakan tag bagian lain dan tag nav.

```
<section id="sidebar">
```

```
<nav>
```

```
<h3>Archives</h3>
```

```
<ul>
```

```
<li><a href="/09">September 2018</a></li>
```

```
<li><a href="/2018/08">Agustus 2018</a></li>
```

```
<li><a href="/2018/07">Juli 2018</a></li>
```

```
<li><a href="/2018/06">Juni 2018</a></li>
```

```
<li><a href="/2018/05">Mei 2018</a></li>
```

```
<li><a href="/2018/04">April 2018</a></li>
```

```
<li><a href="/2018/03">Maret 2018</a></li>
```



```
        <li><a href="2018/02">Februari 2018</a></li>
        <li><a href="2018/01">Januari 2018</a></li>
</ul>
</nav>
</section>
```

Selanjutnya kita dapat mulai menerapkan style pada elemen-elemen yang ada di dalam kode HTML yang telah kita buat.

7.7. Mengatur Tata Letak dengan Style CSS

Jika kita menampilkan kode HTML5 diatas tanpa menggunakan CSS3 maka tata letak tampilan yang muncul tidak sesuai dengan yang kita harapkan. Untuk mengatur tata letak dan style setiap bagian dari tampilan web diatas, perlu digunakan CSS. Untuk mengimplementasikan CSS dalam sebuah website, langkah pertama yang perlu dilakukan adalah membuat file baru bernama style.css.

Langkah berikutnya adalah, sisipkan kode berikut ini ke dalam tag <head> </head> file HTML5 yang telah kita buat sebelumnya.

```
<link rel="stylesheet" href="style.css" type="text/css">
```

Kode diatas berfungsi untuk memanggil file style.css yang kita gunakan untuk mengimplementasikan CSS3 dalam web tersebut.

Dalam file style.css yang telah kita buat, kita perlu menuliskan kode CSS untuk mengatur tampilan file HTML yang telah kita buat sebelumnya. Berikut ini adalah kode CSS3 yang perlu kita tuliskan dalam file style.css untuk mengatur body dalam HTML yang telah kita buat.

```
body{
    width:960px;
    margin:15px auto;
    font-family: Arial, "MS Trebuchet", sans-serif;
}
```

Kode CSS diatas digunakan untuk mengatur semua bagian dalam body website sehingga memiliki lebar tampilan 960px, batas kiri dan kanan 15px, dengan jenis font Arial, "MS Trebuchet", sans-serif.

```
p{
    margin:0 0 20px 0;
}
p, li{
    line-height:20px;
}
```

Kode diatas digunakan untuk mengatur margin atau jarak pada elemen <p>. Tahap selanjutnya, kita mendefinisikan header dengan ukuran lebar 100%.

```
header#page_header{
    width:100%;
}
```

Selanjutnya, kita buat kode CSS untuk membuat agar tampilan navigasi pada header dan footer yang kita buat sebelumnya berubah menjadi horizontal. Berikut ini adalah kode CSS untuk membuat agar list navigasi pada header dan footer berubah menjadi horizontal.

```
header#page_header nav ul, #page_footer nav ul{
    list-style: none;
    margin: 0;
    padding: 0;
}
#page_header nav ul li, footer#page_footer nav ul li{
    padding:0;
    margin: 0 20px 0 0;
    display:inline;
}
```

Selanjutnya kita perlu mengatur tampilan pada bagian posts. Dibawah ini adalah kode CSS untuk mengatur bagian posts agar bergeser ke kiri dan memiliki lebar 74% dari lebar layar keseluruhan.

```
section#posts{
    float: left;
    width: 74%;
}
```

Selanjutnya kita perlu mengatur tampilan pada bagian posts aside. Dibawah ini adalah kode CSS untuk mengatur bagian posts aside agar menempati posisi di sebelah kanan, memiliki lebar 35% dari lebar layar keseluruhan, memiliki margin di sebelah kiri sebesar 5%, memiliki ukuran font 20px dan memiliki tinggi 40px.

```
section#posts aside{
    float: right;
    width: 35%;
    margin-left: 5%;
    font-size: 20px;
    line-height: 40px;
}
```

Kita juga perlu menentukan posisi sidebar dan menentukan lebarnya. Kode CSS dibawah ini digunakan untuk mengatur posisi sidebar agar berada di sisi kiri dan memiliki lebar 25% dari total lebar pada layar.

```
section#sidebar{
    float: right;
    width: 25%;
}
```

Terakhir, kita perlu mengatur tampilan footer. Pada bagian-bagian sebelumnya, telah dilakukan banyak pengaturan float, baik float: left maupun float: right. Oleh sebab itu, pada bagian footer, perlu dilakukan pembersihan property float dengan menggunakan property clear. Property clear digunakan untuk menentukan elemen apa yang dapat berada di samping elemen tersebut. Jika kita menginginkan agar tidak terdapat elemen apapun baik di samping kiri maupun di samping kanan elemen footer, maka kita gunakan clear: both.

```
footer#page_footer{
    clear: both;
    width: 100%;
    display: block;
    text-align: center;
}
```

7.8. Menggabungkan Seluruh Kode

Mungkin masih ada beberapa pembaca merasa kebingungan bagaimana cara menggabungkan dan merangkai seluruh kode diatas menjadi satu kesatuan. Jika keseluruhan kode diatas digabungkan, maka berikut ini adalah gabungan dari keseluruhan kode HTML5 yang telah kita buat dalam bab ini:

```
<!DOCTYPE html>
<html lang="en-US">
<head>
    <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
    <title>Judul Website</title>
    <link rel="stylesheet" href="style.css" type="text/css">
</head>

<body>
<header id="page_header">
<h1>Web Sederhana dengan HTML5 dan CSS3</h1>
<nav>
```

```

        <ul>
        <li><a href="/">Latest Posts</a></li>
        <li><a href="archives">Archives</a></li>
        <li><a href="contributors">Contributors</a></li>
        <li><a href="contact">Contact Us</a></li>
        </ul>
    </nav>
</header>

<section id="sidebar">
<nav>
<h3>Archives</h3>
    <ul>
        <li><a href="2018/10">Oktober 2018</a></li>
        <li><a href="2018/09">September 2018</a></li>
        <li><a href="2018/08">Agustus 2018</a></li>
        <li><a href="2018/07">Juli 2018</a></li>
        <li><a href="2018/06">Juni 2018</a></li>
        <li><a href="2018/05">Mei 2018</a></li>
        <li><a href="2018/04">April 2018</a></li>
        <li><a href="2018/03">Maret 2018</a></li>
        <li><a href="2018/02">Februari 2018</a></li>
        <li><a href="2018/01">Januari 2018</a></li>
    </ul>
</nav>
</section>

<section id="posts">
<article class="post">
<header>
<h2>Apa saja kelebihan dari HTML5 dan CSS3?</h2>
<p>Posted by admin on
<time datetime="2018-10-01T14:39">October 1st, 2018 at 2:39PM</time>
</p>
</header>
    <aside>
        <p>
            &quot;Belajar segala ilmu yang bermafaat adalah ibadah.&quot;
        </p>
    </aside>
<p>
    Teknologi web berkembang secara terus menerus. Perkembangan tersebut sampai saat ini
    membawa kita pada HTML5 dan CSS3. HTML5 dan CSS3 bukan sekedar dua stantar baru yang

```

diusulkan World Wide Web Consortium. Keduanya adalah kelanjutan iterasi teknologi yang setiap hari kita gunakan. Keduanya membantu kita untuk membuat aplikasi web yang modern.

</p>

<p>

Sebelum kita mempelajari lebih dalam tentang HTML5 dan CSS3, ada baiknya kita mengetahui keunggulan dari HTML5 dan CSS3.

Beberapa kelebihan HTML5 dan CSS3 adalah:

- Dukungan multimedia yang lebih baik

- Tampilan yang lebih baik

- Form yang lebih baik

- Visual Efek yang baik

</p>

<footer>

<p><i>25 Comments</i> ...</p>

</footer>

</article>

</section>

<footer id="page_footer">

<p>© 2018 HTML5 CSS3.</p>

<nav>

Home

About

Terms of Service

Privacy

</nav>

</footer>

</body>

</html>

Kode diatas adalah kode HTML secara keseluruhan untuk struktur dalam blog yang kita buat. Simpan kode diatas dalam file berformat .html. Berikut ini adalah gabungan dari keseluruhan kode CSS3 yang telah kita buat dalam bab ini:

```
body{
```

```
width:960px;
```

```
margin:15px auto;
```

```
font-family: Arial, "MS Trebuchet", sans-serif;
```

```
}
```

```
p{
```

```
        margin:0 0 20px 0;
    }

    p, li{
        line-height:20px;
    }

    header#page_header{
        width:100%;
    }

    header#page_header nav ul, #page_footer nav ul{
        list-style: none;
        margin: 0;
        padding: 0;
    }

    #page_header nav ul li, footer#page_footer nav ul li{
        padding:0;
        margin: 0 20px 0 0;
        display:inline;
    }

    section#posts{
        float: left;
        width: 74%;
    }

    section#posts aside{
        float: right;
        width: 35%;
        margin-left: 5%;
        font-size: 20px;
        line-height: 40px;
    }

    section#sidebar{
        float: right;
        width: 25%;
    }

    footer#page_footer{
        clear: both;
    }
```

```
width: 100%;  
display: block;  
text-align: center;  
}
```

Simpan kode CSS3 diatas dalam file bernama style.css dan letakkan dalam satu folder yang sama dengan file HTML yang telah kita buat sebelumnya. Jika file HTML dijalankan dalam browser, maka akan memberikan tampilan seperti berikut ini:

Web Sederhana dengan HTML5 dan CSS3

[Latest Posts](#) [Archives](#) [Contributors](#) [Contact Us](#)

Apa saja kelebihan dari HTML5 dan CSS3?

Posted by admin on October 1st, 2018 at 2:39PM

Teknologi web berkembang secara terus menerus. Perkembangan tersebut sampai saat ini membawa kita pada HTML5 dan CSS3. HTML5 dan CSS3 bukan sekedar dua stantar baru yang diusulkan Worl Wide Web Consortium. Keduanya adalah kelanjutan iterasi teknologi yang setiap hari kita gunakan. Keduanya membantu kita untuk membuat aplikasi web yang modern.

"Belajar segala ilmu yang bermanfaat adalah ibadah."

Sebelum kita mempelajari lebih dalam tentang HTML5 dan CSS3, ada baiknya kita mengetahui keunggulan dari HTML5 dan CSS3. Beberapa kelebihan HTML5 dan CSS3 adalah:

- Dukungan multimedia yang lebih baik
- Tampilan yang lebih baik
- Form yang lebih baik
- Visual Efek yang baik

[25 Comments ...](#)

Archives

- [Oktober 2018](#)
- [September 2018](#)
- [Agustus 2018](#)
- [Juli 2018](#)
- [Juni 2018](#)
- [Mei 2018](#)
- [April 2018](#)
- [Maret 2018](#)
- [Februari 2018](#)
- [Januari 2018](#)

© 2018 HTML5 CSS3.

[Home](#) [About](#) [Terms of Service](#) [Privacy](#)

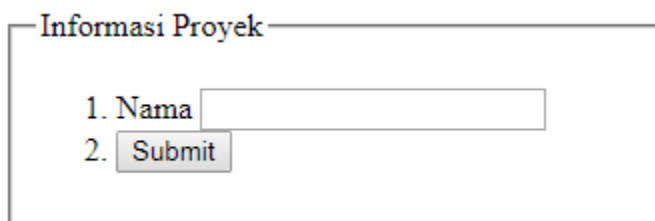
8. Membuat Form dalam HTML5

Jika anda pernah menggunakan HTML basic biasa, anda akan menemukan beberapa form standar seperti menu select, radio buttons, checkbox, dan lainnya yang memiliki keterbatasan dalam tampilan. Jika anda ingin membangun tampilan yang bagus, biasanya anda akan menambahkan HTML dengan jQuery, CSS, dan javascript. Untuk membuat form slider, calendar, spinboxes dan lainnya menggunakan HTML basic, hal itu akan sangat sulit untuk dilakukan. Namun, dengan HTML5, hal ini akan menjadi lebih mudah.

8.1. Membuat Form Teks

Form HTML yang paling sederhana dan paling sering digunakan adalah text box. Text box akan terlihat seperti kotak persegi panjang dimana pengguna bisa mengetik segala macam teks di dalam kotak ini. Tag `<input>` dengan atribut `type="text"` akan menampilkan text box untuk menginput tulisan. Berikut ini adalah contoh pembuatan text box:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Form Teks HTML5</title>
</head>
<body>
<form method="post" action="/projects/1">
<fieldset id="personal_information">
<legend>Informasi Proyek</legend>
  <ol>
    <li>
      <label for="name">Nama</label>
      <input type="text" name="name" id="name">
    </li>
    <li>
      <input type="submit" value="Submit">
    </li>
  </ol>
</fieldset>
</form>
</body>
</html>
```



The screenshot shows a browser window displaying a form titled "Informasi Proyek". The form is enclosed in a rectangular border. Inside the border, there is a list of two items. The first item is "1. Nama" followed by a text input field. The second item is "2. Submit" followed by a submit button.

8.2. Membuat Form Slider

Tag `<input>` dengan atribut `type="range"` akan menampilkan slider untuk menginput angka. Tampilan ini akan memudahkan pengguna karena tidak perlu mengetik teks, tapi cukup menggeser slider saja. Berikut contoh penggunaannya:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Form Slider HTML5</title>
</head>
<body>
<label for="priority">Prioritas</label>
<input type="range" min="0" max="10"
  name="priority" value="0" id="priority" >
</body>
</html>
```

Bila kode diatas dijalankan maka akan memberikan tampilan seperti berikut ini:



Akan tetapi seperti yang terlihat, objek form ini tidak menampilkan angka yang dipilih, sangat tidak nyaman digunakan karena kita tidak tahu berapa nilainya. Agar angka dari slider dapat ditampilkan, kita harus meminta bantuan JavaScript, seperti contoh berikut:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Form Slider HTML5</title>
</head>
<body>
  <form action="prosesform.php" method="get">
    <p>Umur: <input type="range" name="umur" min="0" max="99" step="5"
      onchange="updateNilaiSlider(this.value);">
    <span id="nilai_slider"></span></p>
    <p><input type="submit" value="Kirim Data"></p>
  </form>
  <script>
    function updateNilaiSlider(val) {
      document.getElementById('nilai_slider').innerHTML=val;
    }
  </script>
</body>
</html>
```

Bila kode diatas dijalankan maka akan memberikan tampilan seperti berikut ini:

Umur: 55

8.3. Membuat Form Spinboxes

Jika kita memiliki inputan form yang hanya boleh diisi dengan angka, bisa menggunakan tag `<input>` dengan atribut `type="number"`. Penggunaan objek form ini akan mengaktifkan proses validasi jika karakter yang diinput bukan angka. Selain itu, web browser juga menambahkan tombol up dan down di akhir text box. Tombol ini dikenal dengan sebutan spin button. Berikut ini adalah contoh form spinboxes:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Form Spinboxes HTML5</title>
</head>
<body>
  <label for="estimated_hours">Jam Estimasi </label>
  <input type="number" name="estimated_hours" min="0" max="1000"
    id="estimated_hours" >
</body>
</html>
```

Bila kode diatas dijalankan dalam web browser maka akan meberikan tampilan seperti berikut ini:

Jam Estimasi

8.4. Membuat Form Tanggal

Sebelum HTML5 diluncurkan, membuat form tanggal bukanlah sesuatu yang mudah. Kita perlu membuat tag `<select>` untuk pemilihan tanggal dan memerlukan 3 bagian terpisah. Hasilnya pun tidak terlalu menarik. Karena itu HTML5 menyediakan form untuk menginput tanggal dan waktu, tidak hanya satu, tetapi enam form sekaligus, yakni `date`, `time`, `datetime-local`, `datetime`, `month` dan `week`. Semuanya digunakan sebagai nilai atribut `type` untuk tag `<input>`.

Berikut adalah perbedaan ke-6 type input ini:

- Date: Input berupa tanggal dengan format: YYYY-MM-DD. Contohnya bila tanggal 25 April 2019 maka akan menjadi 2019-04-25.
- Time: Input berupa waktu dengan format: HH:MM, sebagai contoh jam 3:15 siang akan menjadi 15:15.
- Datetime-local: Input berupa tanggal dan waktu dengan format YYYY-MM-DDTHH:mm). Perhatikan bahwa tanda T memisahkan antara tanggal dan waktu. Sebagai contoh, 20 April 2019 pukul 3:15 siang akan menjadi 2019-04-20T15:15.
- Datetime: Input berupa tanggal dan waktu dengan format YYYY-MM-DD HH:mm:ss-HH:mm. Format ini menggunakan time-zone offset, akan tetapi datetime kebanyakan tidak didukung web browser, sebaiknya gunakan datetime-local.
- Month: Input berupa bulan dan tahun dengan format YYYY-MM, sebagai contoh bulan Maret 2019 akan menjadi: 2019-03.
- Week: Input berupa urutan minggu dalam 1 tahun, dengan format YYYY-Www. Nilai maksimal dari week bisa bernilai 52 atau 53, tergantung tahun. Sebagai contoh, Minggu keempat pada tahun 2015 akan menjadi 2015-W04.

Untuk membuat tanggal, tipe yang digunakan adalah date. Berikut ini adalah contoh kode HTML5 untuk membuat form tanggal:

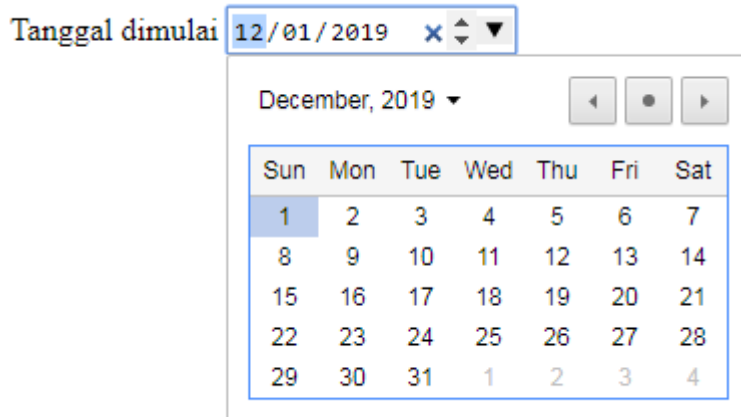
```

<!DOCTYPE html>
<html>
<head>
  <title>Belajar Form Date HTML5</title>
</head>
<body>
<label for="start_date">Tanggal dimulai</label>
<input type="date" name="start_date" id="start_date"
  value="2019-12-01" >

</body>
</html>

```

Bila kode diatas ditampilkan dalam browser maka akan memberikan tampilan seperti beriku ini:



8.5. Membuat Form Email

Email adalah salah satu isian form yang hampir selalu diperlukan dalam setiap form registrasi/login. Oleh karena hal inilah HTML5 menyediakan atribut `type="email"` untuk tag `<input>`. Objek form input type email tampak seperti text box reguler, akan tetapi web browser menambahkan proses validasi untuk memeriksa isian form agar memenuhi format alamat email. Sebuah alamat email yang valid, dimulai dengan beberapa karakter, kemudian diikuti oleh tanda '@' dan diakhiri dengan nama domain, seperti yahoo.com, atau gmail.com. Jika pengguna tidak menginput sesuai dengan format ini, form akan menampilkan pesan error ketika form di-submit. Berikut contoh penulisan tag input type email:

```

<!DOCTYPE html>
<html>
<head>
<meta charset="UTF-8">
  <title>Belajar Form Email HTML5</title>
</head>
<body>
  <form action="proses.php" method="get">
    <p>Email: <input type="email" name="email" multiple></p>
    <p><input type="submit" value="Kirim Data"></p>
  </form>
</body>
</html>

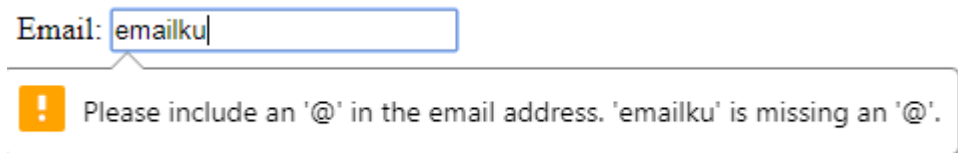
```

Bila kode diatas dijalankan dalam browser maka akan memberikan tampilan seperti berikut ini:

Email:

Kirim Data

Tag input type email juga mendukung atribut multiple. Atribut ini bisa digunakan untuk menginput beberapa alamat email yang dipisah dengan tanda koma. Selain validasi, input type="email" juga memudahkan proses input menggunakan mobile web browser, seperti Android. Ketika pengisian form, keyboard virtual android akan langsung menyediakan tombol '@' agar mudah dipilih. Jika anda memasukkan teks dalam form email tanpa tanda '@' maka sistem akan memberikan peringatan seperti pada gambar berikut ini:



Pesan error saat yang ditampilkan saat proses validasi berasal dari web browser. Jika web browser di-set menggunakan Bahasa Indonesia, pesan error ini juga akan ditampilkan dalam bahasa Indonesia. Jika settingan menggunakan Bahasa Inggris maka pesan error di tampilkan dalam bahasa Inggris.

8.6. Membuat form URL

Tag `<input>` dengan atribut `type="url"` menyediakan kotak input dengan proses validasi URL (alamat web). Secara garis besar, prosesnya mirip dengan input type email, yakni akan menampilkan pesan error jika pengguna menginput sesuatu yang bukan alamat URL. Berikut contoh penulisannya:

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar Form URL HTML5</title>
</head>
<body>
  <label for="url">URL</label>
  <input type="url" name="url" id="url">
</body>
</html>
```

Bila kode diatas ditampilakn dalam browser maka akan memberikan tampilan seperti berikut ini:

URL

Perlu dicatat juga bahwa alamat URL ini harus ditulis bagian protocol-nya, yakni bagian: http://. Jika tidak, pesan error tetap tampil. Sebagai contoh, jika yang diinput alamat adalah: www.stekom.ac.id, alamat tersebut dianggap tidak valid, kita harus menulisnya dengan lengkap: <http://www.stekom.ac.id>. Ketika objek form URL dibuka dengan mobile web browser (android/iOS), keyboard virtual akan menyembunyikan tombol spasi (yang tidak pernah digunakan pada URL), dan menggantinya dengan tombol seperti tanda titik (.), backslash (\) dan “.com”. Ini agar penulisan alamat URL menjadi lebih efisien.

8.7. Form Warna

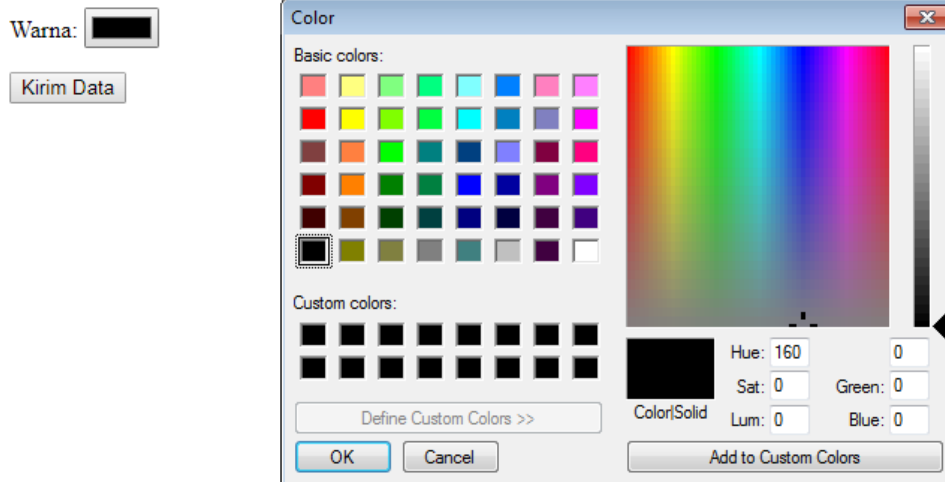
HTML5 menyediakan form khusus untuk memilih warna, yakni menggunakan atribut dengan type="color" pada tag <input>. Ketika dipilih, form ini akan menampilkan jendela warna yang tergantung dengan jenis web browser dan sistem operasi yang digunakan. Sebagai contoh, berikut adalah cara penggunaan input type color:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Form Warna HTML5</title>
</head>
<body>
  <form action="prosesform.php" method="get">
    <p>Warna kesukaan: <input type="color" name="warna"></p>
    <p><input type="submit" value="Kirim Data"></p>
  </form>
</body>
</html>
```

Kode diatas, jika dijalankan dalam browser maka akan memberikan tampilan seperti berikut ini:

Warna:

Jika kotak warna hitam di klik, maka akan memunculkan jendela pilihan waran seperti berikut ini:



Ketika objek form dipilih, nilai yang dikirim adalah angka heksadesimal warna dengan format #RRGGBB. Jika anda sudah pernah mempelajari CSS, tentunya tidak asing dengan format warna ini. R mewakili warna merah (Red), G warna hijau (Green) dan B adalah warna biru (Blue).

8.8. Memberikan Petunjuk dengan Teks Placeholder

Teks placeholder dalam sebuah form memberi petunjuk kepada pengguna tentang bagaimana mereka harus mengisi setiap kolom dalam form tersebut. Untuk menambahkan teks placeholder, tambahkan atribut placeholder ke setiap field masukan, seperti berikut ini:

```
<input id="email" type="email"
name="email" placeholder="nama@contoh.com" >
```

Bila kode diatas dijalankan dalam browser, maka akan memberikan tampilan seperti berikut ini:

Berikut ini adalah contoh kode tampilan form sign up dalam HTML5 yang menggunakan placeholder:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Placeholder HTML5</title>
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.2.1/jquery.min.js"></script>
  <link rel="stylesheet" href="style.css" type="text/css">
  <script src="jquery.placeholder.js">
  </script>
```

```

</head>
<body>

<script>
    function hasPlaceholderSupport() {
        var i = document.createElement('input');
        return 'placeholder' in i;
    }
</script>

<form id="create_account" action="/signup" method="post">
    <fieldset id="signup">
        <legend>Buat Akun Baru</legend>
        <ol>
            <li>
                <label for="first_name">Nama Depan</label>
                <input id="first_name" type="text"
                    autofocus="true"
                    name="first_name" placeholder="Joni" >
            </li>
            <li>
                <label for="last_name">Nama Belakang</label>
                <input id="last_name" type="text"
                    name="last_name" placeholder="Darda" >
            </li>
            <li>
                <label for="email">Email</label>
                <input id="email" type="email"
                    name="email" placeholder="nama@contoh.com" >
            </li>
            <li>
                <label for="password">Password</label>
                <input id="password" type="password" name="password"
                    value=""
                    autocomplete="off" placeholder="8-10 karakter" />
            </li>
            <li>
                <label for="password_confirmation">Konfirmasi
                Password</label>
                <input id="password_confirmation" type="password"
                    name="password_confirmation" value=""
                    autocomplete="off" placeholder="Ketikkan password anda lagi" />
            </li>
            <li><input type="submit" value="Sign Up"></li>
        </ol>
    </fieldset>

```



```
</form>
</body>
</html>
```

Kode diatas adalah kode HTML yang berfungsi sebagai kerangka website. Simpan kode diatas dalam file berformat .html. Dalam kode HTML5 diatas, terdapat kode seperti berikut ini:

```
<script>
function hasPlaceholderSupport() {
var i = document.createElement('input');
return 'placeholder' in i;
}
</script>
```

Kode diatas adalah kode jQuery untuk menampilkan peringatan disaat input salah. Selanjutnya, untuk mengatur tampilan dan tata letak, kita memerlukan CSS. Berikut ini adalah kode CSS yang diperlukan:

```
fieldset{
    width: 216px;
}
fieldset ol{
    list-style: none;
    padding:0;
    margin:2px;
}
fieldset ol li{
    margin:0 0 9px 0;
    padding:0;
}
fieldset input{
    display:block;
}
```

Simpan kode diatas dalam file bernama style.css. Letakkan dalam satu folder yang sama dengan file html yang telah dibuat sebelumnya. Bila file HTML yang telah kita buat dijalankan dalam browser maka akan memberikan tampilan seperti berikut ini:

Buat Akun Baru

Nama Depan

Nama Belakang

Email

Password

Konfirmasi Password

9. Membuat Table Menarik dengan Pseudoclass

Sebuah pseudoclass di CSS adalah cara untuk memilih elemen berdasarkan informasi yang berada di luar dokumen atau informasi yang tidak bisa diungkapkan menggunakan penyeleksi normal. Anda mungkin menggunakan pseudoclasses seperti: `hover` sebelum mengubah warna link saat pengguna melayang di atasnya. Pointer mouse mereka CSS3 memiliki beberapa pseudoclasses baru yang dibuat. Menemukan elemen lebih mudah.

9.1. Membuat Tabel

Sebelum mempelajari cara membuat table dengan Pseudoclass, terlebih dahulu kita belajar membuat table biasa tanpa style apapun.

```
<table >
  <tr>
    <th>Nama Barang</th>
    <th>Harga</th>
    <th>Jumlah</th>
    <th>Total</th>
  </tr>
  <tr>
    <td>Cangkir</td>
    <td>10.000</td>
    <td>5</td>
    <td>50.000</td>
  </tr>
  <tr>
    <td>Kaos</td>
    <td>20.000</td>
    <td>5</td>
    <td>100.000</td>
  </tr>
  <tr>
    <td>Stapler</td>
    <td>9.000</td>
    <td>4</td>
    <td>36.000</td>
  </tr>
  <tr>
    <td colspan="3">Subtotal</td>
    <td>186.000</td>
  </tr>
  <tr>
    <td colspan="3">Ongkos kirim</td>
    <td>12.000</td>
  </tr>
  <tr>
    <td colspan="3">Total biaya</td>
```

```

        <td>198.000</td>
    </tr>
</table>

```

Jika kode diatas dijalankan maka akan menghasilkan tampilan seperti berikut ini:

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

Tampilan table diatas tentu terlihat tidak menarik dilihat. Untuk mempercantik tampilan table agar terlihat lebih modern dan elegan, maka diperlukan CSS untuk mengatur style tampilan table.

9.2. Menambahkan CSS pada Tabel

Mari kita tambahkan kode css dengan cara membuat file baru bernama table.css yang diletakkan dalam satu folder yang sama dengan file table.html. Berikut ini adalah kode dalam table.css

```

table{
    width: 600px;
    border-collapse: collapse;
}
th, td{
    border: none;
}

```

Kode CSS diatas berfungsi untuk mengatur lebar table menjadi 600px. Jangan lupa untuk melakukan pemanggilan file CSS pada file html seperti yang telah kita pelajari pada bab sebelumnya. Hasil dari penambahan file css diatas adalah sebagai berikut:

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

Untuk memberikan warna background pada bagian kepala table (baris paling atas tabel), maka dapat ditambahkan kode berikut ini dalam file CSS yang telah kita buat sebelumnya:

```

th{
    background-color: #000;
    color: #fff;
}

```

Jika kode diatas telah ditambahkan dalam CSS, maka tampilan table akan berubah menjadi seperti berikut ini:

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

9.2.1. Memberi Warna yang Berbeda pada Setiap Row Tabel

Memberikan warna strip zebra pada setiap row dalam table adalah hal yang penting karena dapat membantu pengguna dalam melihat baris dalam table. Untuk melakukan hal tersebut, anda dapat menambahkan kode berikut ini dalam file css.

```
tr:nth-of-type(even){  
    background-color: #F3F3F3;  
}
```

```
tr:nth-of-type(odd){  
    background-color:#ddd;  
}
```

Jika kode diatas telah ditambahkan dalam CSS, maka tampilan table akan berubah menjadi seperti berikut ini:

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

9.2.2. Membuat Rata Kanan dengan :nth-child

Secara default, seluruh teks dalam table adalah rata kiri. Terkadang, kita membutuhkan teks dalam table berformat rata kanan, misalnya saat menulis angka. Untuk melakukan hal tersebut, kita dapat menggunakan nth-child. Untuk dapat melakukan hal tersebut, terlebih dahulu kita perlu mempelajari cara kerja nth-child. Secara umum format sintaks nth-child adalah sebagai berikut:

```
:nth-child(nomer) {  
    deklarasin css;  
}
```

Pada format `nth-child` diatas, nomer dapat diganti dengan angka, atau juga dapat diganti dengan rumus matematika dengan variable `n`. Variabel `n` akan otomatis menjadi angka yang teriterasi dari 0, 1, 2, dan seterusnya. Jika kita ingin mengakses seluruh row dalam tabel, maka dapat mengisikan variable `n` dalam sintaks tersebut, dengan contoh sebagai berikut:

```
table tr:nth-child(n)
```

Jika kita ingin mengakses seluruh row dalam table, kecuali pada row pertama table, maka dapat menggunakan kode berikut ini:

```
td:nth-child(n+2){  
    deklarasi css;  
}
```

Untuk membuat seluruh row dalam table memiliki teks dengan format rata kanan, kecuali pada row pertama dalam table, maka dapat digunakan kode berikut ini:

```
td:nth-child(n+2){  
    text-align: right;  
}
```

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

9.2.3. Membuat Bold pada *Last Row* dengan: `last-child`

Bagi seorang manajer, detail baris per baris setiap kolom dalam table seringkali tidak begitu penting. Sebagian besar manajer hanya ingin melihat baris paling bawah yang merupakan hasil akhir dari setiap table.

```
tr:last-child{  
    font-weight: bolder;  
}
```

Kode diatas digunakan untuk membuat baris terakhir dalam table memiliki tulisan bold (cetak tebal). Jika ingin membuat tulisan menjadi bold pada kolom terakhir, ganti `tr` pada kode diatas menjadi `td`. Kodenya akan menjadi seperti ini:

```
td:last-child{  
    font-weight: bolder;  
}
```

Kode diatas digunakan untuk membuat tulisan pada kolom terakhir sebuah table menjadi bercetak tebal (bold). Untuk memperbesar ukuran font baris terbawah dapat menggunakan kode berikut ini:

```
tr:last-child td:last-child{
    font-size:22px;
}
```

Bila kode diatas dirangkai dan dijalankan, maka akan memberikan tampilan seperti berikut ini:

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

9.3. Menggabungkan Seluruh Kode

Barangkali diantara pembaca masih ada yang bingung bagaimana cara merangkai kode yang telah kita buat sebelumnya menjadi satu kesatuan. Subbab ini akan membahas bagaimana gabungan seluruh kode yang telah kita buat diatas. Kali ini kita akan menggunakan metode eksternal CSS. Sehingga akan terdiri dari dua file yaitu satu file .html dan satu file .css. Berikut ini adalah kode yang disimpan dalam file .html:

```
<!DOCTYPE html>
<html>
<head>
    <link rel="stylesheet" href="tabel.css" type="text/css">
</head>
<body>
<table >
<tr>
    <th>Nama Barang</th>
    <th>Harga</th>
    <th>Jumlah</th>
    <th>Total</th>
</tr>
<tr>
    <td>Cangkir</td>
    <td>10.000</td>
    <td>5</td>
    <td>50.000</td>
</tr>
<tr>
```

```

        <td>Kaos</td>
        <td>20.000</td>
        <td>5</td>
        <td>100.000</td>
</tr>
<tr>
        <td>Stapler</td>
        <td>9.000</td>
        <td>4</td>
        <td>36.000</td>
</tr>
<tr>
        <td colspan="3">Subtotal</td>
        <td>186.000</td>
</tr>
<tr>
        <td colspan="3">Ongkos kirim</td>
        <td>12.000</td>
</tr>
<tr>
        <td colspan="3">Total biaya</td>
        <td>198.000</td>
</tr>
</table>
</body>
</html>

```

Berikut ini adalah gabungan kode CSS yang disimpan dalam file berformat .css. Simpan kode dibawah ini dalam file bernama tabel.css dan letakkan dalam satu folder yang sama dengan file .html diatas:

```

table{
width: 400px;
border-collapse: collapse;
}
th, td{
border: none;
}

th{
background-color: #000;
color: #fff;
}

tr:nth-of-type(even){
background-color: #F3F3F3;
}

```



```
}  
tr:nth-of-type(odd){  
background-color:#ddd;  
}
```

```
td:nth-child(n+2){  
text-align: right;  
}
```

```
tr:last-child{  
font-weight: bold;  
}
```

```
td:last-child{  
font-weight: bold;  
}
```

```
tr:last-child td:last-child{  
font-size:22px;  
}
```

Bila seluruh tahapan diatas dijalankan dengan benar, maka akan memberikan tampilan seperti berikut ini:

Nama Barang	Harga	Jumlah	Total
Cangkir	10.000	5	50.000
Kaos	20.000	5	100.000
Stapler	9.000	4	36.000
Subtotal			186.000
Ongkos kirim			12.000
Total biaya			198.000

10. Canvas

Elemen `<canvas>` digunakan untuk menggambar grafis secara real time dalam website. Canvas hanya digunakan sebagai tempat menggambar. Diperlukan JavaScript untuk menggambar suatu grafis dalam canvas. Meski begitu, Canvas memiliki beberapa metode untuk menggambar jalur, kotak, lingkaran, teks, dan menambahkan gambar.

10.1. Membuat Canvas

Dengan `width` menyatakan lebar canvas, dan `height` menyatakan tinggi canvas, berikut ini adalah format penulisan elemen canvas:

```
<canvas id="nama_canvas" width="350" height="150"></canvas>
```

Atribut `id` berfungsi sebagai ‘identitas’ atau ‘label’ dari tag `<canvas>`. Ini diperlukan karena canvas akan diakses menggunakan JavaScript. Atribut `width` dan `height` digunakan untuk mengatur tinggi dan lebar dari canvas. Bila dijalankan di browser, kode di atas hanya akan menampilkan layar putih. Hal tersebut dikarenakan warna dasar canvas sama dengan warna dasar tampilan browser yaitu berwarna putih.

Agar dapat melihat area canvas dengan jelas, kita berikan bingkai atau border pada area canvas dengan warna hitam. Berikut ini cara memberikan border pada area canvas:

```
<html>
<body>
<canvas id="canvasku" width="350" height="150" style="border: 1px solid #000000;">
</canvas>
</body>
</html>
```

Kode di atas digunakan untuk membuat sebuah canvas bernama `canvasku` dengan lebar 350, tinggi 150, tebal bingkai 1px, warna bingkai hitam. Bila kode di atas dijalankan di browser, maka akan muncul tampilan seperti berikut ini:



10.2. Menggambar Garis dalam Canvas

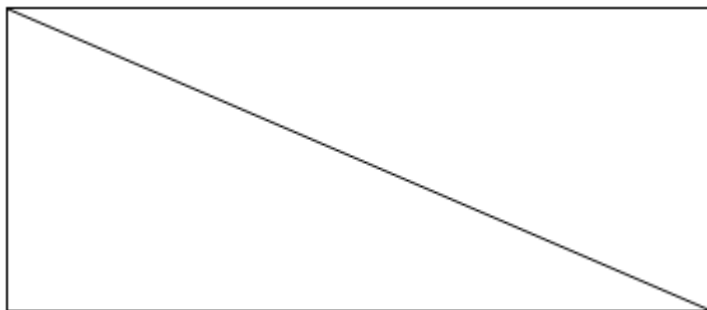
Untuk menggambar garis lurus dari koordinat tertentu ke koordinat tertentu dalam sebuah canvas, misalkan dari koordinat (X1, Y1) ke koordinat (X2, Y2) kita dapat menggunakan kode sebagai berikut ini:

```
var c = document.getElementById("nama_canvas");
var ctx = c.getContext("2d");
ctx.moveTo(X1, Y1);
ctx.lineTo(X2, Y2);
ctx.stroke();
```

Dalam HTML5, koordinat dimulai dari pojok kiri atas tampilan pada browser. Artinya, koordinat (0, 0) terletak di pojok kiri atas. Karena kode diatas adalah JavaScript, maka agar kode tersebut berjalan dalam file html, perlu diapit dengan elemen <script>. Berikut ini adalah contoh membuat garis lurus pada sebuah canvas bernama canvasku:

```
<!DOCTYPE html>
<html>
<body>
<canvas id="canvasku" width="350" height="150" style="border:1px solid #000000;">
</canvas>
<script>
    var c = document.getElementById("canvasku");
    var ctx = c.getContext("2d");
    ctx.moveTo(0, 0);
    ctx.lineTo(350, 150);
    ctx.stroke();
</script>
</body>
</html>
```

Bila kode diatas ditampilkan dalam browser, maka akan memberikan tampilan seperti berikut ini:



10.3. Menggambar Lingkaran dalam Canvas

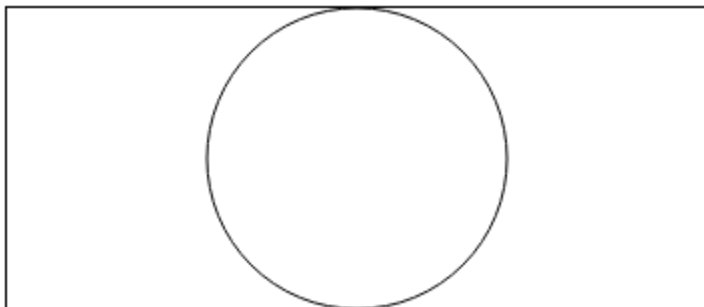
Untuk menggambar lingkaran penuh dengan pusat lingkaran berada pada koordinat (Xp, Yp) dan jari-jari r maka format penulisannya adalah sebagai berikut ini:

```
var c = document.getElementById("nama_canvas");
var ctx = c.getContext("2d");
ctx.beginPath();
ctx.arc(Xp, Yp, r, 0, 2 * Math.PI);
ctx.stroke();
```

Karena kode diatas adalah JavaScript, maka agar kode tersebut berjalan dalam file html, perlu diapit dengan elemen <script>. Berikut ini adalah contoh membuat lingkaran pada sebuah canvas bernama canvasku:

```
<!DOCTYPE html>
<html>
<body>
<canvas id="canvasku" width="350" height="150" style="border:1px solid #000000;">
</canvas>
<script>
    var c = document.getElementById("canvasku");
    var ctx = c.getContext("2d");
    ctx.beginPath();
    ctx.arc(175, 75, 75, 0, 2 * Math.PI);
    ctx.stroke();
</script>
</body>
</html>
```

Jika kode diatas dijalankan, maka akan muncul gambar sebagai berikut ini:



10.4. Menulis Teks dalam Canvas

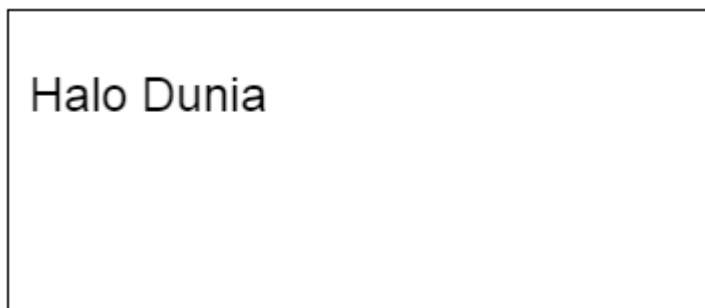
Kita dapat menambahkan teks dalam canvas dengan menggunakan perintah `context.font()` dan `context.fillText()`. Berikut ini adalah contohnya:

```
var c = document.getElementById("nama_canvas");
var ctx = c.getContext("2d");
ctx.font = "24px Arial";
ctx.fillText("Halo Dunia", 10, 50);
```

Perintah `context.font="25px Arial"` artinya: “pilih jenis font Arial sebesar 24 pixel”. Sedangkan perintah `context.fillText("Halo Dunia", 10, 50)` berarti perintah untuk menuliskan teks “Halo Dunia” dimulai dari koordinat 10, 50. Jadi, 10, 50 adalah koordinat posisi dimana teks tersebut mulai dituliskan. Berikut ini adalah kode HTML lengkapnya:

```
<!DOCTYPE html>
<html>
<body>
<canvas id="canvasku" width="350" height="150" style="border:1px solid #d3d3d3;">
Maaf, browser anda tidak support tag canvas HTML5.</canvas>
  <script>
    var c = document.getElementById("canvasku");
    var ctx = c.getContext("2d");
    ctx.font = "24px Arial";
    ctx.fillText("Halo Dunia", 10, 50);
  </script>
</body>
</html>
```

Bila kode diatas dijalankan dalam browser maka akan memberikan tampilan seperti berikut ini:



10.5. Memberikan Gradien Warna dalam Canvas

Untuk membuat gradient warna dalam canvas, dapat menggunakan kode sebagai berikut:

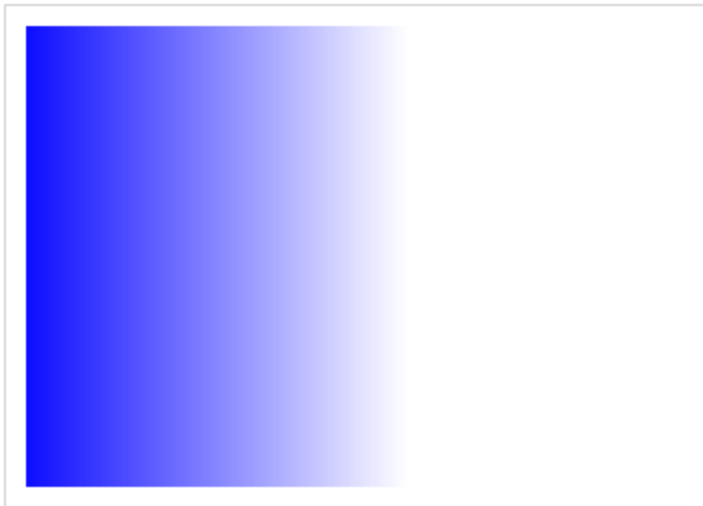
```
<!DOCTYPE html>
```

```
<html>
<body>
<canvas id="Canvasku" width="350" height="250" style="border: 1px solid #d3d3d3;">
  Browser anda tidak support tag canvas dalam HTML5.</canvas>
  <script>
    var c = document.getElementById("Canvasku");
    var ctx = c.getContext("2d");

    var grd = ctx.createLinearGradient(0,0,200,0);
    grd.addColorStop(0,"blue");
    grd.addColorStop(1,"white");

    ctx.fillStyle = grd;
    ctx.fillRect(10,10,350,230);
  </script>
</body>
</html>
```

Bila kode diatas dijalankan, maka akan menampilkan canvas yang telah diisi dengan gradient warna seperti berikut ini:



11. SVG dan MathML

Integrasi SVG dan MathML bisa jadi merupakan sebuah tonggak perjalanan menuju aplikasi web masa depan. Memasukkan format vector ke dalam browser sudah menjadi hal yang lama tertunda. Kehadiran SVG menjadi hal yang ditunggu-tunggu banyak orang.

11.1. MathML

MathML adalah sebuah tag dalam HTML yang digunakan untuk menuliskan rumus atau simbol matematika. Untuk menjelaskan seperti apa itu MathML mari simak contoh berikut

```
<math>
<mrow>
  <mi>r</mi>
  <mo>=</mo>
  <msqrt>
    <mfrac>
      <mrow>
        <mi>A</mi>
      </mrow>
      <mrow>
        <mn>&Pi;</mn>
      </mrow>
    </mfrac>
  </msqrt>
</mrow>
</math>
```

Pada contoh kode diatas terdapat tiga mrow. Bila kode diatas dijalankan maka akan memberikan tampilan seperti dibawah ini:

$$r = A \Pi$$

Setiap blok MathML selalu diawali dengan $dan diakhiri dengan$.

Elemen	Nama	Tujuan
Mrow	row	Elemen untuk membuat group ekspresi
Mi	I untuk identifier	Variable, nama fungsi, atau konstanta
Mo	O untuk operator	Operator seperti sama dengan, plus, minus, atau tanda multiplikasi
Msqrt	Sqrt untuk square root	Ekspresi akar pangkat
Mfrac	Fract untuk fraction	Fraction atau division
Mn	N untuk nomer	Untuk nomer

Berikut ini adalah beberapa simbol yang sering digunakan dalam penulisan rumus matematika:

Simbol	Kode HTML	Penjelasan
\approx	≈	Hamper sama dengan
\neq	≠	Tidak sama dengan
\leq	≤	Lebih kecil sama dengan
\geq	≥	Lebih besar sama dengan
\equiv	≡	Equivalent
\subset	⊂	Subset dari
\supset	⊃	Superset dari
$\not\subset$	⊄	bukan subset dari
\subseteq	⊆	Subset atau equal
\supseteq	⊇	Superset atau equal
Σ	Σ	Sigma
ϵ	Ε	Epsilon
\emptyset	∅	Himpunan kosong
\sum	∑	Total penjumlahan
$-$	−	Minus
\forall	∀	Untuk semua
∂	∂	Bagian
∇	∇	Nabla
\in	∈	Himpunan bagian dari
\notin	∉	Bukan hompunan bagian dari
∞	∞	Tak terhingga
\int	∫	Integral
Ω	Ω	Omega
α	α	Alpha
β	β	Beta
γ	γ	Gamma
δ	δ	Delta
θ	&theta	Theta
λ	&lambd	Lambda
μ	&mu	Mu

11.2. SVG

Apa itu SVG? SVG adalah singkatan dari Scalable Vector Graphics. SVG digunakan untuk mendefinisikan grafik dalam web. Seperti namanya, SVG berbasis pada vector, sehingga memiliki keunggulan ketika gambar dizoom tidak akan pecah. Dibalik keunggulannya, SVG memiliki kelemahan yaitu kita tidak dapat menyimpan gambar foto pemandangan ke dalam SVG, karena terlalu kompleks untuk dibuat persamaan matematikanya. Oleh karena itu gambar SVG biasanya hanya berupa icon, teks, sketsa atau gambar 2 dimensi yang berbentuk sederhana.

Tidak semua browser telah mendukung SVG. Berikut ini adalah table beberapa browser beserta versinya yang telah support SVG:

Broswer	chrome	IE	Firefox	Safari	Opera
Versi	4.0	9.0	3.0	3.2	10.1

Perbedaan SVG dan Canvas

- SVG adalah bahasa untuk mendeskripsikan grafik 2 dimensi dalam XML
- Canvas membuat gambar grafik 2 dimensi dengan cepat dengan JavaScript
- SVG berbasis pada XML, yang berarti bahwa setiap elemen tersedia dalam SVG DOM. Anda dapat melampirkan penanganan kejadian JavaScript untuk elemen.
- Dalam SVG, setiap bentuk yang ditarik diingat sebagai objek. Jika atribut objek SVG diubah, peramban dapat secara otomatis merender ulang bentuknya.
- Kanvas dirender piksel demi piksel. Di kanvas, begitu gambar ditarik, itu dilupakan oleh browser. Jika posisinya harus diubah, seluruh adegan perlu digambar ulang, termasuk objek apa pun yang mungkin telah ditutupi oleh grafik.

Perbandingan Kanvas dan SVG

Tabel di bawah ini menunjukkan beberapa perbedaan penting antara Canvas dan SVG:

Canvas	SVG
Bergantung pada resolusi	Tidak bergantung pada resolusi
Tidak mendukung untuk event handlers	Memberi dukungan pada event handler
Buruk dalam kapabilitas rendering teks	Baik untuk aplikasi dengan rendering area besar (google maps)
Dapat di save dalam .png dan .jpg	Rendering lambat jika kompleks (segala sesuatu yang menggunakan DOM akan lambat)
Baik untuk grafis dalam game	Tidak cocok untuk aplikasi game

Berikut ini adalah tabel yang berisi beberapa elemen dalam SVG beserta tujuan dan atributnya:

Elemen	Tujuan	Atribut
<circle>	Membuat lingkaran	cx, cy, r
<rect>	Membuat segi empat	width, height, rx, ry
<ellipse>	Membuat ellipse	Cx, cy, rx, ry
<polyline>	Membuat polyline	points
<polygon>	Membuat polygon	points

Blok SVG selalu diawali dengan tag <svg> dan diakhiri tag </svg>. Berbeda dengan MathML, tag awal juga menentukan lebar dan tinggi dari grafik SVG. Jangan lupa untuk menyisakan sejumlah ruang yang sesuai pada halaman HTML. Berikut ini adalah contoh SVG:

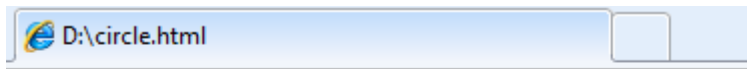
```
<svg width="200" height="200">
  <circle cx="100" cy="100" r="80"/>
```

Maaf, browser anda tidak support SVG.
</svg>

Kode di atas membuat sebuah lingkaran dengan pusat cx/cy dan jari-jari r. Bila dijalankan dalam browser yang support SVG, maka akan menampilkan lingkaran berwarna hitam dengan jari-jari 80 pixel. Namun jika dibuka dalam browser yang tidak support SVG maka akan muncul tulisan, "Maaf, browser anda tidak support SVG". Berikut ini adalah tampilannya:



Berikut ini adalah tampilan jika dibuka dengan browser yang tidak support SVG:



Maaf, browser anda tidak support SVG.

Kita juga dapat menambahkan style CSS dalam SVG. Beberapa property CSS yang dapat digunakan dalam SVG adalah sebagai berikut:

- Fill: digunakan untuk mengatur warna elemen.
- Stroke-width: digunakan untuk mengatur lebar garis tepi suatu elemen.
- Stroke: digunakan untuk menentukan warna garis tepi elemen.
- Opacity: Digunakan untuk mengatur transparansi suatu objek.

Berikut ini adalah contoh kode SVG yang telah ditambahkan kode inline CSS:

```
<svg width="800" height="200">  
  <rect x="70" y="60" rx="20" ry="20" width="80" height="80"  
  style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />  
  Maaf browser anda tidak support SVG.  
</svg>
```

Bila kode diatas dijalankan dalam browser maka akan memberikan tampilan seperti berikut ini:



Dari hasil tampilan diatas dapat dilihat bahwa penambahan kode CSS dapat mempercantik tampilan SVG.

11.2.1. Membuat Lingkaran

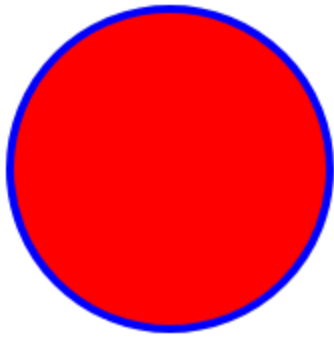
Untuk membuat lingkaran, kita dapat menggunakan elemen `<circle>`. Terdapat tiga atribut utama dalam circle, yaitu:

- Cx menyatakan koordinat sumbu X pusat lingkaran
- Cy menyatakan koordinat sumbu Y pusat lingkaran
- R menyatakan radius lingkaran yang dinyatakan dalam pixel.

Berikut ini adalah contohnya:

```
<!DOCTYPE html>
<html>
<head>
</head>
<body>
<svg width="200" height="200">
  <circle cx="100" cy="100" r="80"
  stroke="blue" stroke-width="4" fill="red" />
  Maaf, browser anda tidak support SVG
</svg>
</body>
</html>
```

Atribut lain seperti stroke, stroke-width, dan fill adalah atribut umum yang dapat digunakan dalam elemen SVG mana saja. Atribut stroke digunakan untuk menyatakan warna garis tepi objek yang digambar. Atribut stroke-width menyatakan ketebalan garis tepi. Atribut fill menyatakan warna dalam gambar. Jika kode di atas dijalankan dalam browser, maka akan menjadi seperti ini:



11.2.2. Membuat Persegi

Untuk membuat persegi, dalam SVG disediakan tag `<rect>`, yaitu singkatan dari rectangular. Berikut ini contoh penggunaannya:

```
<svg width="200" height="100">
  <rect width="200" height="100" />
</svg>
<svg width="200" height="100">
  <rect width="200" height="100" rx="30" ry="30" />
</svg>
```

Atribut `width` dan `height` menyatakan lebar dan tinggi persegi. Bila kode diatas dijalankan dalam browser, maka akan menampilkan hasil sebagai berikut ini:



Dapat dilihat bahwa persegi yang menggunakan atribut `rx` dan `ry`, memiliki sudut yang melengkung.

Selain bentuk persegi sederhana yang secara otomatis berwarna hitam seperti diatas, kita juga dapat menambahkan inline style CSS dalam SVG. Berikut ini contohnya:

```
<!DOCTYPE html>
<html>
<head>
</head>
```

```

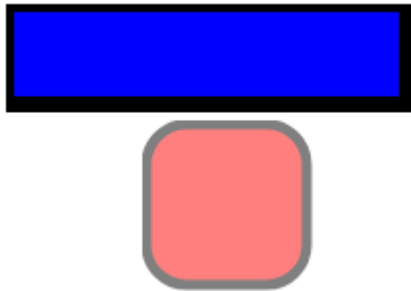
<body>
  <svg width="800" height="200">
    <rect width="200" height="50" style="fill:rgb(0,0,255);stroke-
      width:8;stroke:rgb(0,0,0)" />
    <rect x="70" y="60" rx="20" ry="20" width="80" height="80"
      style="fill:red;stroke:black;stroke-width:5;opacity:0.5" />
    Maaf browser anda tidak support SVG.
  </svg>
</body>
</html>

```

Beberapa property CSS dalam SVG adalah sebagai berikut:

- Fill: digunakan untuk mengatur warna elemen.
- Stroke-width: digunakan untuk mengatur lebar garis tepi suatu elemen.
- Stroke: digunakan untuk menentukan warna garis tepi elemen.
- Opacity: Digunakan untuk mengatur transparansi suatu objek.

Bila dijalankan dalam browser, kode diatas akan memberikan tampilan seperti berikut ini:



11.2.3. Membuat Ellipse

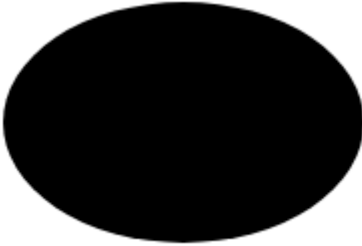
Untuk membuat ellipse kita dapat menggunakan tag <ellipse>. Berbeda dengan <rect>, untuk <ellipse> terdapat 2 radius, yaitu radius untuk sumbu x (rx) dan radius untuk sumbu y (ry). Kedua radius ini harus berbeda agar terbentuk ellipse. Jika keduanya sama, maka akan membentuk sebuah lingkaran. Terdapat beberapa property dasar dalam tag <ellipse> yaitu cx, cy, rx, ry. Cx digunakan untuk menentukan titik koordinat X awal, dan Cy digunakan untuk menentukan titik koordinat Y awal. Rx digunakan untuk menentukan jari-jari ellipse pada sumbu x, dan Ry digunakan untuk menentukan jari-jari ellipse pada sumbu y.

```

<svg width="200" height="200">
  <ellipse cx="100" cy="60" rx="90" ry="60" />
</svg>

```

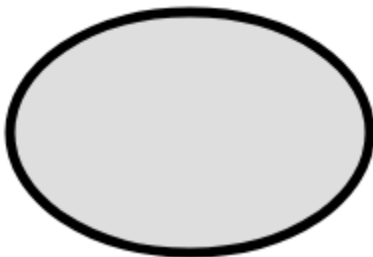
Bila kode diatas ditampilkan dalam browser, maka akan memberikan tampilan seperti berikut ini:



Bila ingin menambahkan beberapa style CSS, maka berikut ini adalah contoh kode lengkapnya:

```
<!DOCTYPE html>
<html>
<body>
<svg width="300" height="300">
  <ellipse cx="100" cy="70" rx="90" ry="60" style="fill:rgb(222,222,222);stroke-
width:5;stroke:rgb(0,0,0)" />
Maaf browser anda tidak support SVG.
</svg>
</body>
</html>
```

Bila kode diatas dijalankan dalam browser yang support SVG dan HTML5, maka hasilnya adalah sebagai berikut ini:

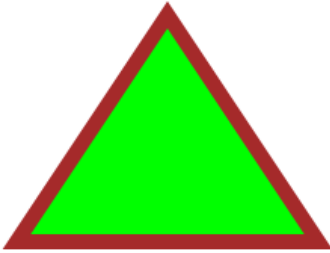


11.2.4. Membuat Polygon

Polygon adalah istilah dalam SVG untuk membuat gambar dengan berbagai titik koordinat. Ini dibuat menggunakan tag `<polygon>`. Sebagai contoh, segitiga bisa disebut sebagai polygon, karena memiliki 3 titik. Berikut contoh penggunaannya:

```
<svg width="500" height="300" >
<polygon points="250,50 350,200 150,200"
stroke="brown" stroke-width="10" fill="lime"/>
</svg>
```

Atribut `points` di dalam tag `<polygon>` berfungsi untuk membuat titik koordinat. Titik koordinat ini bergantian dari titik $x_1, y_1, x_2, y_2, x_3, y_3$, dst. Atribut `points="250,50 350,200 150,200"` artinya, saya membuat poligon dengan 3 titik, titik pertama pada koordinat 250,50, titik kedua pada 350,200 dan titik ketiga pada 150,200. Saat titik ketiga dibuat, poligon otomatis menghubungkannya kembali dengan titik pertama. Bila kode diatas dijalankan, maka akan memberikan hasil tampilan seperti berikut ini:



11.2.5. Membuat Garis Terhubung

Untuk membuat sebuah garis yang saling bersambung, SVG menyediakan tag `<polyline>`. Dalam tag `<polyline>` juga menggunakan atribut `points`. Nilai dari atribut ini berupa pasangan koordinat x dan y . Berikut contoh penggunaannya:

```
<svg>
<polyline points="0,100 50,25 50,75 100,0" />
<polyline points="100,100 150,25 150,75 200,0"
fill="none" stroke="black" />
</svg>
```

Bila kode diatas ditampilkan dalam browser, maka akan memberikan tampilan seperti berikut ini:



12. Multimedia

Pada awalnya, HTML hanya digunakan sebagai sarana berbagi hasil penelitian antar ilmuwan, dimana sebagian besar konten terdiri dari teks, tabel, dan gambar. Saat ini berbagai situs web saling berlomba menampilkan konten multimedia untuk menarik perhatian pengunjung. Salah satunya dengan menambahkan file multimedia seperti audio dan video.

Sebelum kehadiran HTML5, untuk menjalankan file multimedia, kita harus bergantung kepada aplikasi pihak ketiga seperti adobe flash. Oleh karenanya, setiap komputer harus terinstall adobe flash player. Jika pengunjung tidak memiliki aplikasi adobe flash player, video tidak bisa dijalankan.

Keterbatasan inilah yang membuat berbagai pihak mendesak agar HTML dapat mendukung audio dan video tanpa bantuan aplikasi lain. W3C dan WHATWG menjawab masalah ini dengan menyediakan tag <audio> dan <video>. Tag <audio> dan <video> adalah fitur HTML5 yang relatif baru. Dengan demikian, tidak semua web browser mendukung tag ini. Secara umum versi terbaru web browser Google Chrome, Mozilla Firefox, dan Opera mendukung penuh tag <audio> dan <video>. Yang sering menjadi masalah adalah web browser Internet Explorer 8 (bawaan Windows 7) dan versi IE di bawahnya.

12.1. Elemen Video

Salah satu fitur yang populer dari HTML5 adalah dukungannya untuk multimedia. Dulu, anda perlu memuat plug-in untuk memutar video atau audio. Mungkin plug-in multimedia yang paling populer adalah Flash. Bab ini membahas dukungan HTML5 untuk memutar video.

Pada bab ini kita akan belajar menampilkan video dalam HTML5. Berikut ini adalah format penulisan dalam HTML5 untuk elemen video.

```
<video src="filevideo.mp4" controls ></video>
```

12.1.1. Format Video

World Wide Web Consortium (W3C) memperkenalkan elemen video untuk tujuan memutar video dan film. Tujuannya adalah menawarkan cara standar untuk memutar video di web tanpa memerlukan plug-in. Tujuan ini telah dielakkan oleh kurangnya kesepakatan tentang format video mana yang harus didukung di browser web.

Saat ini, W3C tidak menentukan format video mana yang harus didukung oleh browser. W3C sedang mencoba untuk datang dengan format video yang dapat ditentukan untuk semua browser untuk mendukung sebagai minimum. Memiliki minimal satu format yang umum di semua browser akan menyederhanakan pekerjaan pengembang selama format memberikan kompresi dan kualitas yang baik dan bebas royalti. Ada banyak format; daftar berikut ini menjelaskan format yang paling populer.

- Oog
Pada satu titik, W3C menentukan format Ogg/Theora dikarenakan format yang bebas royalti dan tanpa paten. Kemudian W3C menghapus Ogg/Theora dari spesifikasi karena kemungkinan litigasi. Format ini didukung oleh browser Firefox, Chrome, dan Opera.
- Webm
Google mengakuisisi format WebM/VP8 ketika membeli Teknologi On2. Menurut pejabat Google, WebM berfungsi dengan baik pada perangkat yang berdaya lebih rendah, termasuk netbook dan handheld. WebM akan tersedia di bawah lisensi *open source* Berkeley Software Distribution (BSD) yang bebas royalti. Format video WebM/VP8 ini didukung oleh browser Firefox, Chrome, Opera, dan Android. Selain itu, Internet Explorer 9+ akan mendukung format ini ketika codec VP8 diinstal.
- Mp4
Format ini merupakan evolusi format sebelumnya yang dimaksudkan untuk membuat standar yang mampu memberikan kualitas video yang bagus pada tingkat bit yang jauh lebih rendah daripada standar sebelumnya. Format MPEG-4/H.264 memiliki teknologi yang dipatenkan, tetapi MPEG LA, yang merupakan organisasi swasta yang bertanggung jawab atas pengadministrasian paten, mengumumkan bahwa video Internet yang dikodekan H.264 yang gratis untuk pengguna akhir tidak akan dikenakan biaya royalti. MPEG-4/H.264 didukung oleh browser Internet Explorer, Chrome, dan Safari, tetapi Chrome telah mengumumkan niatnya untuk menghapus dukungan untuk MPEG-4 / H.264 dalam waktu dekat.

12.1.2. Width dan Controls

Pada subbab ini kita akan belajar tentang width dan controls. Width adalah atribut dalam video yang digunakan untuk mengatur ukuran tampilan video dalam browser. Controls adalah atribut dalam elemen video yang digunakan untuk memunculkan fungsional kontrol video dalam browser. Simak contoh kode dibawah ini:

```
<video width="400" controls>
  <source src="filevideo.mp4" type="video/mp4">
  <source src="filevideo.ogg" type="video/ogg">
  Maaf, browser anda tidak support video HTML5.
</video>
```

Pada contoh kode diatas, atribut width diberikan nilai sebesar 400, artinya ukuran lebar video yang akan tampil dalam browser adalah sebesar 400 pixel. Control digunakan untuk menampilkan kontrol seperti play, pause, dan volume. Source digunakan untuk memanggil sumber video yang ingin ditampilkan. Jika lebar video perlu diatur, apakah juga harus mengatur ukuran tinggi video?

Merupakan ide bagus untuk selalu menyertakan atribut lebar dan tinggi. Jika tinggi dan lebar tidak diatur, halaman mungkin akan berkedip saat video dimuat.



12.1.3. Elemen Source

Elemen `<source>` memungkinkan Anda menentukan file video alternatif yang dapat dipilih oleh browser. Browser akan menggunakan format yang dikenal pertama. Teks peringatan antara tag `<video>` dan `</ video>` hanya akan ditampilkan di browser yang tidak mendukung elemen `<video>`.

Untuk membuat video agar berjalan secara otomatis tanpa perlu kontrol dari pengunjung, dapat dilakukan dengan cara sebagai berikut:

```
<video width="320" height="240" autoplay>  
  <source src="filevideo.ogg" type="video/ogg">  
  <source src="filevideo.mp4" type="video/mp4">  
Maaf, browser anda tidak support video HTML5.  
</video>
```

Ketika pembuat website menyediakan banyak format video, browser dapat memilih format yang dapat digunakan untuk menampilkan video. Posisi elemen `<source>` penting karena browser mulai mencari di bagian atas dan berhenti ketika menemukan file yang dapat ditampilkan. Urutan yang disarankan adalah mulai dengan file `.webm` karena ini bebas royalti dan sumber terbuka dan lebih populer. Selanjutnya, gunakan file `.ogv` karena juga bebas royalti, tetapi kualitasnya tidak sebagus file `.webm`. Terakhir, gunakan format `.mp4` untuk browser yang tidak mendukung file `.webm` atau `.ogv`.

12.1.4. Atribut Autoplay

Atribut autoplay bisa ditambahkan ke dalam tag <video> agar video langsung berjalan ketika halaman ditampilkan. Berikut contoh penggunaannya:

```
<video src="filevideo.mp4" autoplay></video>
```

12.1.5. Atribut Muted

Secara default, video akan dijalankan dengan volume penuh. Kita bisa menambahkan atribut muted untuk mematikan volume. Walaupun begitu, pengguna bisa memperbesar volume melalui tombol yang tersedia. Berikut contohnya:

```
<video src="filevideo.mp4" muted controls></video>
```

Menjalankan video tanpa suara bisa digunakan untuk memutar iklan atau promosi produk agar tidak terlalu mengganggu. Bila kode diatas dijalankan maka akan menampilkan video dengan simbol volume dicoret seperti berikut ini:



12.1.6. Atribut Loop

Atribut loop akan membuat video berulang otomatis ketika sampai di akhir durasi. Tanpa menggunakan atribut ini, video akan berhenti apabila telah selesai diputar. Berikut contoh penulisannya:

```
<video src="filevideo.mp4" loop controls></video>
```

12.1.7. Atribut Preload

Atribut preload digunakan untuk mengatur cara buffering oleh web browser. Nilai yang digunakan pada atribut ini dapat berupa none, metadata, dan auto. Jika nilai atribut preload="auto", maka web browser akan mencoba mendownload atau men-buffer file video pada saat web ditampilkan pertama kali. Jika nilai atribut preload="none", maka web browser tidak menbuffer file video tersebut hingga tombol play di-klik. Saat video mulai diklik play, ketika itulah web browser baru memulai proses buffering.

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Video Element</title>
</head>
<body>
  <h3>Tag video dengan atribut preload = auto</h3>
    <video src="filevideo.mp4" controls preload="auto"></video>
<br>
  <h3>Tag video dengan atribut preload = none</h3>
    <video src="filevideo.mp4" controls preload="none"></video>
<br>
  <h3>Tag video dengan atribut preload = meta</h3>
    <video src="filevideo.mp4" controls preload="meta"></video>
</body>
</html>
```

Bila kode diatas dijalankan, maka akan memberikan hasil seperti berikut ini:

Tag video dengan atribut preload = auto



Tag video dengan atribut preload = none



Tag video dengan atribut preload = meta



Dapat dilihat bahwa pada video dengan atribut preload=none tidak memunculkan gambar sebelum pengunjung mengklik tombol play pada video.

12.1.8. Atribut Poster

Tag <video> memiliki atribut poster yang bisa digunakan untuk menampilkan gambar placeholder sebelum video dijalankan. Atribut ini berisi nilai alamat kepada file gambar.

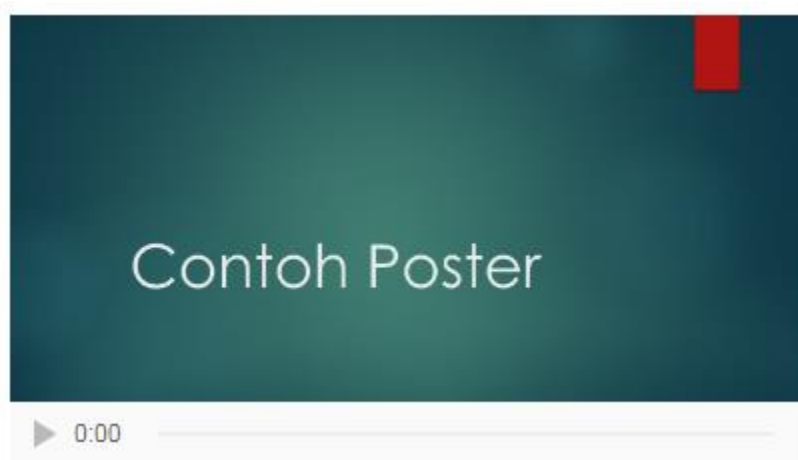
Contoh:

```
<!DOCTYPE html>
```

```
<html>
```

```
<head>
  <title>Belajar Video Element</title>
</head>
<body>
  <video src="filevideo.mp4" controls width="400" poster="contoh-poster.jpg">
  </video>
</body>
</html>
```

Jika kode diatas dijalankan, maka akan menampilkan tampilan seperti berikut ini:



12.2. Elemen Audio

Sebelum HTML5, file audio dapat diputar di web browser dengan bantuan plugin seperti flash. Dalam HTML5 elemen `<audio>` digunakan untuk menanamkan file audio dalam halaman web. Berikut adalah cara penggunaan tag `<audio>`:

```
<audio controls>
  <source src="fileaudio.ogg" type="audio/ogg">
  <source src="fileaudio.mp3" type="audio/mpeg">
  Browser anda tidak mendukung elemen audio.
</audio>
```

Terdapat beberapa format audio:

- MP3
Format MP3 (MPEG-2 Audio Layer III) merupakan format paling populer untuk audio. MP3 menggunakan teknik lossy compression, dimana kita bisa men-kompres ukuran file audio menjadi lebih kecil dengan mengorbankan sedikit kualitas. Walaupun populer, format MP3 bersifat proprietary (berlisensi). Jika anda ingin mengembangkan aplikasi pemutar MP3 (termasuk membuat web browser yang mendukung MP3) maka harus membayar sejumlah biaya. Karena alasan inilah web

browser dari perusahaan ‘kecil’ seperti Opera dan Mozilla Firefox pada awalnya tidak mendukung format MP3. Namun saat ini kedua web browser sudah mendukung penuh MP3 pada versi terbarunya.

- WAV

Format WAV (Waveform Audio File Format) juga bersifat proprietary (dikembangkan oleh IBM dan Microsoft), namun berbeda dengan MP3 maupun Ogg, WAV menerapkan lossless compression, dimana file audio tidak dikompres untuk mempertahankan kualitas. Karena itu file yang dihasilkan berukuran besar sehingga tidak cocok digunakan untuk web.

- Ogg

Format Ogg (atau lengkapnya: Ogg Vorbis) adalah format audio open source, sehingga penggunaannya tidak membutuhkan biaya lisensi. Ogg menggunakan prinsip lossy compression seperti MP3 dimana ukuran file bisa diperkecil dengan mengorbankan sedikit kualitas audio. Format yang dikembangkan oleh Xiph.Org Foundation ini pada awalnya dirancang untuk menggantikan MP3. Namun karena relatif baru, penggunaan format ogg masih kalah populer jika dibandingkan mp3. Karena bersifat open source, format Ogg pada awalnya ditetapkan sebagai format audio standar untuk HTML5. Namun hal ini akhirnya dibatalkan dan web browser diberi kebebasan untuk menggunakan jenis format audio. Web browser Mozilla Firefox dan Opera mendukung penuh format Ogg sejak kemunculan tag <audio>.

- ACC

Format ACC (Advanced Audio Coding) dikembangkan oleh beberapa perusahaan teknologi seperti Bell Labs, Fraunhofer Institute, Dolby Labs, Sony dan Nokia. Format proprietary ini juga menggunakan lossy compression seperti MP3, namun dirancang menghasilkan kualitas suara yang lebih baik dari pada MP3 (dengan ukuran file yang sama). Karena hal inilah format ACC mulai populer digunakan terutama untuk music store seperti iTunes.

12.2.1. Atribut Source

Atribut src (source) digunakan sebagai penunjuk lokasi file audio. Anda dapat meletakkan file audio dimana saja selama bisa diakses menggunakan alamat relatif maupun alamat absolut. Dibawah ini adalah contoh penggunaan atribut src dimana file audio berada pada satu folder yang sama dengan file HTMLnya.

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar Elemen Audio</title>
</head>
<body>
```

```
<audio src="fileaudio.mp3"></audio>
</body>
</html>
```

12.2.2. Atribut Control

Agar kita bisa memutar file audio, perlu ditambahkan atribut controls kepada tag <audio>, seperti berikut ini:

```
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar Elemen Audio</title>
</head>
<body>
  <audio src="fileaudio.mp3" controls></audio>
</body>
</html>
```

Atribut controls berfungsi untuk menampilkan tombol control pada audio player, yakni tombol play/stop, slider, volume, timer, dll. Jika tag <audio> ditulis tanpa menggunakan atribut controls, tidak akan terlihat tampilan apa-apa.

12.2.3. Atribut Autoplay

Kita juga bisa menambahkan atribut autoplay pada tag <audio>. Atribut ini akan membuat audio langsung berjalan ketika halaman web ditampilkan. Tanpa atribut ini, file audio hanya akan berjalan ketika tombol player di tekan. Berikut contoh penggunaannya:

```
<audio src="fileaudio.mp3" controls autoplay></audio>
```

Silahkan jalankan kode di atas, dan audio akan langsung terdengar ketika halaman web ditampilkan. Jika atribut autoplay digunakan tanpa atribut controls, maka anda dapat membuat halaman web dengan audio yang langsung berjalan, namun tidak bisa dihentikan. Sehingga satusatunya cara untuk menghentikan musik adalah dengan menutup halaman web.

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Belajar Audio Element</title>
</head>
<body>
  <audio src="fileaudio.mp3" autoplay></audio>
</body>
</html>
```


12.2.4. Atribut Loop

Secara default, audio akan berhenti setelah diputar. Apabila anda ingin audio tersebut otomatis diulang kembali dari awal, bisa menambahkan atribut loop. Berikut contohnya:

```
<audio src="fileaudio.mp3" loop controls></audio>
```

Untuk mencoba efek atribut loop, silahkan lompati musik hingga di akhir durasi, dan file audio akan otomatis diputar kembali dari awal. Efek seperti ini cocok untuk membuat music background yang diputar terus menerus.

12.2.5. Atribut Muted

Agar efek audio tidak mengganggu, kita bisa menjalankan audio dengan volume=0. Jika pengguna ingin mendengarkan musik, ia tinggal memperbesar volume dari tombol audio player. Untuk hal ini, HTML menyediakan atribut **muted**. Berikut contoh penggunaannya:

```
<audio src="fileaudio.mp3" controls autoplay muted></audio>
```

Kode di atas akan menjalankan file audio ketika halaman web di jalankan, namun tanpa suara. Jika anda melakukan hal ini, paling tidak informasikan kepada pengguna bahwa sebenarnya terdapat file audio yang di-mute-kan.



12.2.6. Atribut Preload

Tag <audio> menyediakan atribut preload yang digunakan untuk mengatur cara web browser mendownload file audio ketika halaman web dijalankan. Atau bisa juga dipahami dengan cara web browser men-buffer file audio tersebut. Atribut ini bisa diset dengan salah satu nilai berikut: none, metadata, dan auto. Berikut ini adalah contoh implementasi dari atribut <audio>:

```
<!DOCTYPE html>
<html>
<head>
  <title>Belajar Elemen Audio</title>
</head>
<body>
<h4>Ini contoh elemen audio dengan atribut preload=auto</h4>
  <audio src="fileaudio.mp3" controls preload="auto"></audio>
<br>
```

```
<h4>Ini contoh elemen audio dengan atribut preload=none</h4>
  <audio src="fileaudio.mp3" controls preload="none"></audio>
<br>
<h4>Ini contoh elemen audio dengan atribut preload=meta</h4>
  <audio src="fileaudio.mp3" controls preload="meta"></audio>
</body>
</html>
```

Bila kode diatas dijalankan dalam browser, maka akan memberikan tampilan seperti berikut ini:

Ini contoh elemen audio dengan atribut preload=auto



Ini contoh elemen audio dengan atribut preload=none



Ini contoh elemen audio dengan atribut preload=meta



Referensi

Brian P. Hogan. 2010. *HTML5 and CSS3 Develop with Tomorrow's Standards Today*. Pragmatic Programmers, LLC.

<https://www.ariona.net/ebook-belajar-html-dan-css/>

<https://www.duniailkom.com>

<https://developer.mozilla.org>

<https://www.w3schools.com/>

<https://www.youtube.com/>

Klaus Förster, Bernd Öggl. 2011. *HTML5 Guidelines for Web Developers*. Pearson Education, Inc.

Widodo, Triyoga Budi. 2013. *Pedoman Dasar Membuat Website*. Pustaka Media