



Download

Mobile cloud

Appz Theme

Easy To Use

Dare to HOPE

of Apps to.day!

Get your free Download

Clean & Minimalistic Design.

Simple, So Easy To Use.

Desain Web Responsif dengan HTML5 dan CSS3



Edy Winarno ST, M.Eng
Ali Zaki
SmitDev Community

Desain Web Responsif dengan HTML5 dan CSS3

Sanksi Pelanggaran Pasal 113
Undang-Undang Nomor 28 Tahun 2014
tentang Hak Cipta

1. Setiap Orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf i untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan/atau pidana denda paling banyak Rp100.000.000 (seratus juta rupiah).
2. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf c, huruf d, huruf f, dan/atau huruf h untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan/atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah).
3. Setiap Orang yang dengan tanpa hak dan/atau tanpa izin Pencipta atau pemegang Hak Cipta melakukan pelanggaran hak ekonomi Pencipta sebagaimana dimaksud dalam Pasal 9 ayat (1) huruf a, huruf b, huruf e, dan/atau huruf g untuk Penggunaan Secara Komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan/atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah).
4. Setiap Orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan/atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah).

Desain Web Responsif dengan HTML5 dan CSS3

**Edy Winarno ST, M.Eng
Ali Zaki
SmitDev Community**

PENERBIT PT ELEX MEDIA KOMPUTINDO



KOMPAS GRAMEDIA

Desain Web Responsif dengan HTML5 dan CSS3

Edy Winarno ST, M.Eng

Ali Zaki

SmitDev Community

©2015, PT Elex Media Komputindo, Jakarta

Hak cipta dilindungi undang-undang

Diterbitkan pertama kali oleh

Penerbit PT Elex Media Komputindo

Kelompok Gramedia, Anggota IKAPI, Jakarta 2015

nkfadli@elexmedia.co.id

121150271

ISBN: 9786020258256

Dilarang keras menerjemahkan, memfotokopi, atau memperbanyak sebagian atau seluruh isi buku ini tanpa izin tertulis dari penerbit.

Dicetak oleh Percetakan PT Gramedia, Jakarta

Isi di luar tanggung jawab percetakan

Kata Pengantar

Sekarang ini, website lazim diakses dari berbagai platform, mulai dari komputer desktop sampai piranti mobile. Desainer web dan para pengelola website seharusnya dapat melihat tren ini. Website modern harus mengakomodasi kebutuhan berinternet para pengguna beragam piranti smartphone.

Buku ini membahas tren baru dalam pengembangan web, yaitu Responsive Web Design (RWD) dengan menggunakan tool HTML5 dan CSS3. Melalui teknik desain web responsif, secara cepat dan praktis Anda dapat menyulap website Anda untuk dapat diakses dari berbagai platform piranti, tanpa harus mengembangkan aplikasi-aplikasi tersendiri untuk tiap jenis platform piranti.

Penulis menyadari bahwa buku ini tidak luput dari kesalahan dan masih jauh dari sempurna. Untuk itu, penulis membuka diri untuk segala bentuk tanggapan dan pertanyaan pembaca berkaitan dengan buku ini. Untuk ber-konsultasi secara langsung dengan penulis, silakan layangkan email ke winarno@smitdev.com atau ali@smitdev.com.

Semarang, Desember 2014

**Edy Winarno ST, M.Eng
Ali Zaki
SmitDev Community**

Daftar Isi

| | |
|----------------------------|-----------|
| Kata Pengantar..... | v |
| Daftar Isi..... | vi |

BAB 1 Pendahuluan1

| | |
|--|----|
| 1.1 Sejarah Pengembangan Website | 1 |
| 1.1.1 Apa Itu Desain Responsif..... | 2 |
| 1.1.2 Kenapa Piranti Mobile Sangat Penting | 3 |
| 1.1.3 Desain Responsif vs Desain Gadget Khusus | 4 |
| 1.1.4 Desain Web Responsif Tidak Terbatas Hanya di Piranti Mobile..... | 7 |
| 1.1.5 Kapan Tidak Menggunakan Desain Responsif..... | 8 |
| 1.1.6 Memahami Viewport..... | 9 |
| 1.1.7 Breakpoint..... | 10 |
| 1.2 Fitur Baru HTML 5 dan CSS 3 | 10 |
| 1.2.1 Mengeset Browser untuk Simulasi..... | 12 |
| 1.2.2 Simulator Android dan Firefox Emulator..... | 15 |
| 1.3 Media Query dan Keunggulannya | 18 |
| 1.3.1 Pengenalan Media Query..... | 18 |
| 1.3.2 Pengetesan Media Query..... | 19 |
| 1.4 Media Query..... | 22 |
| 1.4.1 Merangkai Media Query | 24 |
| 1.4.2 Menggunakan Media Query di CSS..... | 25 |

BAB 2 HTML 527

| | |
|--|----|
| 2.1 Pengenalan | 28 |
| 2.1.1 Struktur Dasar HTML..... | 29 |
| 2.1.2 Mengetahui Kode HTML Sebuah Web Page | 31 |
| 2.1.3 Mengenal XHTML | 32 |
| 2.1.4 Menampilkan Teks | 34 |
| 2.1.5 Menampilkan Link..... | 37 |
| 2.1.6 Membuat List..... | 38 |
| 2.1.7 Menampilkan Gambar | 39 |
| 2.1.8 Tabel di HTML | 40 |
| 2.1.9 Mengenal Frame di HTML..... | 41 |
| 2.1.10 Mengenal Scripting | 43 |

| | |
|---|----|
| 2.2 Apa Yang Baru di HTML 5 | 44 |
| 2.2.1 Paradigma Bebas Plugin | 45 |
| 2.2.2 Doctype dan Character Set Baru..... | 47 |
| 2.2.3 Elemen-elemen Baru dan Lama..... | 47 |
| 2.2.4 Markup Semantik..... | 48 |
| 2.2.5 Tag-tag Baru | 51 |
| 2.3 Bekerja dengan Audio dan Video | 54 |
| 2.3.1 Codec Audio dan Video | 55 |
| 2.3.2 Batasan Audio dan Video | 56 |
| 2.3.3 Mendeklarasikan Elemen Media..... | 57 |
| 2.4 Menggunakan Web Storage | 62 |
| 2.4.1 Local Storage..... | 63 |
| 2.4.2 Session Storage | 64 |
| 2.5 Form untuk Input Type..... | 65 |
| 2.5.1 Email | 66 |
| 2.5.2 URL | 66 |
| 2.5.3 Number | 67 |
| 2.5.4 Range | 67 |
| 2.5.5 Date Picker..... | 67 |

BAB 3 CSS 369

| | |
|----------------------------|-----|
| 3.1 Apa Itu CSS?..... | 69 |
| 3.2 Sintaks CSS..... | 70 |
| 3.2.1 Komentar CSS..... | 72 |
| 3.2.2 Selektor CSS | 73 |
| 3.2.3 Memasukkan CSS | 75 |
| 3.3 Modul CSS 3 | 76 |
| 3.4 Border | 77 |
| 3.5 Background..... | 79 |
| 3.6 Gradients | 82 |
| 3.7 Efek Teks..... | 87 |
| 3.8 Web Font..... | 90 |
| 3.9 Transformasi 2D | 94 |
| 3.10 Transformasi 3D..... | 100 |
| 3.11 Transisi | 103 |
| 3.12 Multi Kolom..... | 107 |
| 3.13 User Interface | 109 |

BAB 4 Menggunakan Layout Fluid 115

| | |
|---|-----|
| 4.1 Tipe Layout | 115 |
| 4.1.1 Fixed Width | 116 |
| 4.1.2 Elastis | 118 |
| 4.1.3 Fluid | 119 |
| 4.2 Desain Responsif dengan Layout Fluid..... | 122 |
| 4.2.1 Prinsip Desain Fluid | 122 |
| 4.2.2 Contoh Pembuatan Layout..... | 123 |
| 4.2.3 Usahakan Tidak Menggunakan Scroll Bar Horizontal | 129 |
| 4.2.4 Mempertimbangkan Tampilan Gambar dalam Berbagai Ukuran..... | 129 |
| 4.2.5 Membuat Wrapping Konten | 132 |
| 4.3 Memberikan Spasi..... | 132 |
| 4.3.1 Content-box | 134 |
| 4.3.2 Padding-box | 134 |
| 4.3.3 Border-box..... | 135 |
| 4.4 Pengaturan Panjang Teks..... | 136 |
| 4.5 Membangun Desain Fluid Menggunakan Grid CSS..... | 137 |
| 4.6 Membangun Grid | 138 |
| 4.6.1 Kolom | 139 |
| 4.6.2 Gutter | 140 |
| 4.6.3 Row | 141 |
| 4.6.4 Wrapper | 142 |
| 4.7 Membuat Layout Responsif..... | 142 |
| 4.8 Framework di Desain Responsif | 148 |
| 4.8.1 Sistem Grid | 149 |
| 4.9 Framework CSS..... | 154 |

Tentang Penulis 156

BAB 1

Pendahuluan

Cara membuat website sudah berkembang sangat pesat sejak pertama kali WWW menjadi massal sekitar 20 tahun ke belakang. Pada tahun 1990, website umumnya dikembangkan menggunakan tabel sebagai alat bantu untuk menata konten di dalamnya. Di bab ini akan dijelaskan bagaimana perkembangan website sampai era pengembangan website responsif.

1.1 Sejarah Pengembangan Website

Walaupun spesifikasi CSS sudah dirilis sejak tahun 1996, tapi CSS baru banyak diimplementasikan untuk membuat website di tahun 2003-an. Ini membuat banyaknya kode CSS yang dipakai untuk membuat website, di mana CSS ini memungkinkan tampilan website berubah drastis walaupun tanpa mengutak atik kode HTML utamanya.

Dengan perkembangan komputer lebih lanjut, maka muncul layar dengan resolusi lebih besar, sehingga pembuat website pun mulai menyesar layar dengan resolusi 1024 x 800, di mana makin banyak ruang kosong di layar. Sehingga kalau dibuat standar, maka kalau dibuka di layar besar akan banyak ruang kosong, sementara kalau dibuka di layar kecil akan perlu scrolling untuk mengecilkannya.

Lalu smartphone pun makin populer, dimulai dengan adanya iPhone di tahun 1997, dan web juga menjadi sesuatu yang lazim diakses di smartphone, sehingga sebuah website harus bisa responsif, yaitu bisa merespon alat pengaksesnya, dari mulai layar lebar, hingga layar kecil smartphone.

Pada tahun 2010, CSS3 mulai diimplementasikan di browser piranti mobile. Dari mulai Android 2.1, dan diikuti dengan iOS 3.2. Adanya query media memungkinkan pembuatan website untuk berbagai target di berbagai resolusi yang ada.

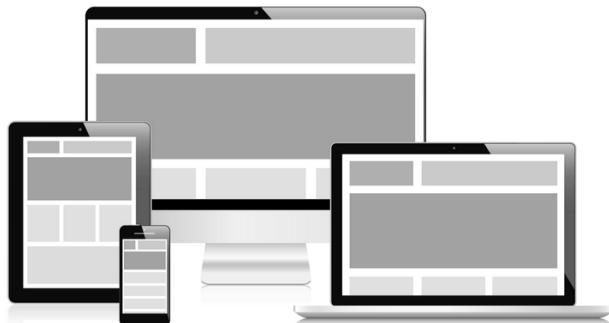
Selain perubahan ukuran layar, ada juga peningkatan kualitas layar yang digunakan, misalnya resolusi DPI menjadi lebih tinggi, sehingga lebih baik menggunakan tampilan DPI tinggi agar website terlihat bagus di piranti mobile. Untuk menangani hal ini, digunakanlah metodologi pengembangan website responsif atau lazim disebut *Responsive Web Design*.

Website responsif ini menjadi tren sekarang ini. Di bab ini akan dijelaskan pendahuluan mengenai desain responsif, apa, dan bagaimana desain responsif.

1.1.1 Apa Itu Desain Responsif

Istilah responsif berkaitan dengan bagaimana browser merespon lingkungan sekitarnya. Jadi, desain responsif berkaitan bagaimana sebuah website dikembangkan, sehingga bisa menyesuaikan dengan lingkungan pengaksesnya, apakah browser, piranti, atau ukuran layar yang digunakannya.

Jadi pendekatan desain responsif akan membuat layout website mudah diadaptasi menggunakan grid yang fleksibel, konten yang fleksibel (seperti image, video, dan teks) serta query media CSS3.



Gambar 1.1 Desain responsif, memberikan tampilan yang responsif terhadap semua jenis browser di berbagai piranti

Desain responsif akan menghindari unit yang fix seperti piksel, dan menggunakan unit yang relatif seperti prosentase. Ini artinya, lebar dari beberapa bagian dari website merupakan prosentase dari viewport.

Yang dimaksud dari desain responsif adalah bahwa semua display harus menerima konten yang sama, namun dibangun dengan fleksibel, sehingga mengoptimalkan tampilan untuk tiap piranti.

1.1.2 Kenapa Piranti Mobile Sangat Penting

Dengan tren smartphone, kini semua orang bisa mengakses internet dengan jauh lebih mudah karena tinggal mengambilnya dari kantong atau tas, tidak perlu repot menyalakan, dan menunggu loading sistem operasi yang sangat lama, apabila mengakses website dari komputer.

Pasar smartphone kini membesar, riset menunjukkan bahwa di tahun 2013 saja, penjualan smartphone hampir 60% dari total penjualan ponsel, tren ini terus meningkat di mana nantinya hampir semua ponsel yang dijual adalah yang berjenis smartphone.



Gambar 1.2 Beragam piranti mobile

Konsekuensinya, nanti akan lebih banyak orang yang mengakses website dari browser di smartphone daripada di komputer. Sehingga, tren ini tidak boleh disepelekan. Kedepannya, mengakses internet via ponsel akan lumrah, dan lebih lazim daripada mengakses internet via piranti lainnya.

Piranti mobile memiliki potensi untuk hadir di mana-mana, bahkan menggantikan komputer desktop. Sekarang ini, Anda bisa membeli tablet dengan harga kurang dari 500 ribu rupiah dan juga smartphone dengan harga kurang dari 300 ribu rupiah. Sehingga biaya untuk memiliki atau cost entry-nya jauh lebih murah.

Terlebih, antarmukanya yang touch screen membuat lebih intuitif untuk dipakai dibandingkan piranti desktop atau notebook. Piranti komputer desktop juga kurang optimal karena kalau kelamaan dipakai, akan membuat tangan dan bahu menjadi pegal, kurang ergonomis.

Indikator lain dari makin populernya piranti mobile adalah penjualan untuk komputer sudah makin jenuh, sementara penjualan untuk piranti mobile masih terus berkembang.

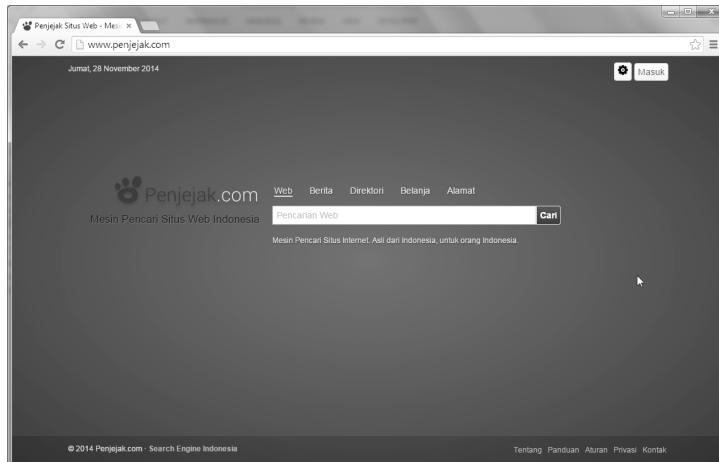
1.1.3 Desain Responsif vs Desain Gadget Khusus

Piranti mobile tumbuh luar biasa pesat, mungkin Anda bertanya kenapa tidak membangun website yang khusus mobile saja, khusus platform yang menjadi target. Misalnya, membuat website khusus mobile seperti memakai ekstensi `m.domain.com`?

Ketika membandingkan desain responsif dan desain yang khusus platform tertentu, Anda perlu mengetahui keunggulan desain responsif. Yang pertama lebih murah dan mudah dalam menerapkan konten, dan desain kalau kode dasarnya sama. Jadi, Anda bisa menampilkan konten yang seragam bahkan di versi cetak, online, desktop, atau piranti mobile. Jadinya UX (*user experience*) yang bagus akan membutuhkan integrasi platform yang baik.

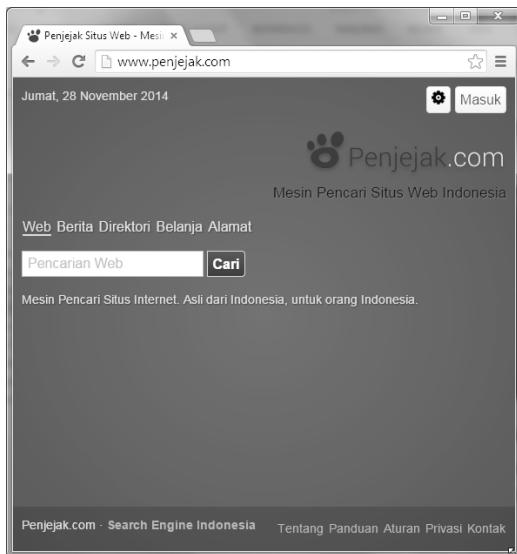
Salah satu keuntungan utama desain web responsif adalah simpel. Anda tidak perlu memisahkan versi mobile. Karena desain responsif memungkinkan Anda menggunakan url yang sama dan kode dasar yang sama. Dengan satu kode dasar, pengetesan menjadi lebih sederhana, terutama kalau Anda bekerja di tempat kerja yang menggunakan pengembangan test-driven. Di mana, dua kode bisa menyebabkan pengetesan menjadi lebih repot.

Misalnya situs <http://www.penjejak.com>, sebuah situs pencarian yang dikembangkan oleh smitdev.com yang ketika ditampilkan di tampilan full screen terlihat seperti berikut.



Gambar 1.3 Tampilan awal situs pencari Indonesia, Penjejak.com

Ketika ukuran browser dikecilkan setengah, maka tampilan berubah, yang secara fisik disebabkan konfigurasi CSS yang menargetkan width memang berubah. Ini ditujukan untuk pengguna piranti layar sedang seperti tablet.



Gambar 1.4 Tampilan browser kondisi setengah

Ketika diubah ke ukuran kecil, maka tampilan situs mengelil seperti di tampilan piranti ponsel.



Gambar 1.5 Browser menjadi ukuran kecil

Kesederhanaan dari desain kode responsif adalah Anda tidak perlu mengelola lebih dari satu konten, sehingga lebih menghemat uang, dan waktu pemrograman Anda. Selain itu, website dengan desain responsif juga lebih optimal ditinjau dari sisi SEO oleh mesin pencari dibandingkan desain website yang terpisah. Google merekomendasikan desain web responsif dengan beberapa catatan seperti berikut:

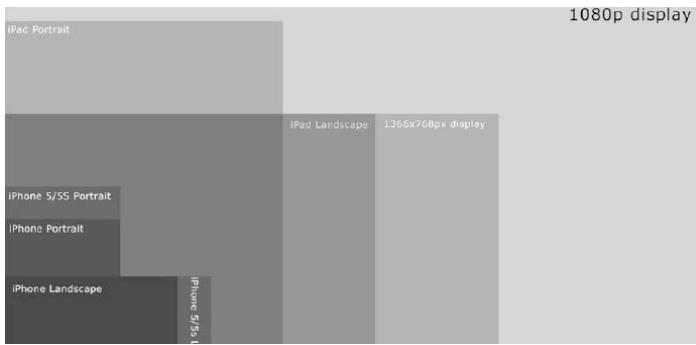
- URL tunggal untuk konten yang membuat user lebih mudah berinteraksi dan berbagi konten.
- URL tunggal untuk konten membuat algoritma Google lebih mudah mengindeks situs.
- Tidak ada redirection atau pendekripsi piranti di server yang membuat proses loading menjadi lebih cepat.
- GoogleBot akan meng-crawl situs sekali dan tidak berulang kali seperti halnya situs dengan multi versi.

1.1.4 Desain Web Responsif Tidak Terbatas Hanya di Piranti Mobile

Desain web responsif tidak hanya terbatas untuk piranti mobile saja, hampir semua platform bisa mengambil manfaat dari desain web responsif. Apalagi sekarang ini yang digunakan untuk melihat internet tidak cuma piranti mobile saja, tapi ada televisi, konsol game, dan lain sebagainya. Sekarang ini, hampir semua konsol game memiliki web browser untuk mengakses website.

Dengan menggunakan teknik responsif, Anda bisa memanfaatkan desain responsif untuk diakses melalui berbagai platform. Di negara-negara maju, umumnya televisi dan piranti game sudah terhubung ke internet. Ini menyebabkan banyak sekali platform yang berbeda-beda, bahkan di satu piranti dari satu pabrikan sekalipun (seperti iPad dan iPhone) memiliki berbagai variasi resolusi layar.

Berikut ini contoh variasi untuk resolusi layar.



Gambar 1.6 Contoh resolusi di piranti apple

Contoh di atas, satu pabrikan saja bisa memiliki berbagai resolusi yang perlu didukung, apalagi pabrikan piranti ada lebih dari satu, sehingga memakai desain responsif sangat menghemat.

1.1.5 Kapan Tidak Menggunakan Desain Responsif

Walaupun desain responsif sangat berguna, tapi tidak semuanya bisa diterapkan desain responsif. Misalnya, pada kondisi ketika aplikasi web dibuat untuk memberikan pengalaman pemakaian seperti aplikasi desktop. Misalnya Google Docs, jika menggunakan ukuran layar yang kecil, Google Docs tidak optimal.

A screenshot of a Google Docs spreadsheet titled "Data Perusahaan". The spreadsheet has columns A through F. Column A is labeled "Timestamp", column B is "Status hukum perusahaan?", column C is "Omzet per bulan?", and column D is "Nama direktur?". The data shows four rows of information:

| | Timestamp | Status hukum perusahaan? | Omzet per bulan? | Nama direktur? | |
|----|-----------------------|--------------------------|------------------|------------------|--|
| 2 | 3/26/2011 19:33:50 PT | | | 3 Keith McDonald | |
| 3 | 3/26/2011 19:34:19 UD | | | 1 Ramdhani | |
| 4 | 3/26/2011 19:34:47 PT | | | 4 Luke Joe | |
| 5 | | | | | |
| 6 | | | | | |
| 7 | | | | | |
| 8 | | | | | |
| 9 | | | | | |
| 10 | | | | | |
| 11 | | | | | |
| 12 | | | | | |
| 13 | | | | | |
| 14 | | | | | |
| 15 | | | | | |
| 16 | | | | | |
| 17 | | | | | |
| 18 | | | | | |

Gambar 1.7 Aplikasi di web seperti Google Docs tidak cocok untuk Desain Responsif

Apalagi kalau ukuran web sangat besar, maka basis kodennya perlu dicek ulang apakah bisa diubah ke desain responsif atau tidak. Jika basis kode terlalu bloatware, maka perlu dirampingkan terlebih dahulu baru kemudian diset menggunakan desain responsif.

1.1.6 Memahami Viewport

Salah satu konsep penting di desain responsif adalah viewport. Viewport adalah tampilan di mana Anda melihat website Anda. Sebelum HTML5 dan CSS3, umumnya sebuah website dilihat dalam ukuran browser yang digunakan.

Pada umumnya ukuran browser pada saat itu adalah 1024 x 800 pada saat full screen, dan ketika itu website umumnya dibuat menggunakan panjang sekitar 960 px hingga 980 px.

Ketika awal munculnya piranti mobile, para pabrikan agak bingung menghadapi hal ini, di mana ukuran website pada umumnya fix dan lebih besar dari ukuran layar piranti mobile.

Solusinya adalah dengan mengeset viewport lebih besar dari lebar piranti, sehingga website akan diskala menjadi lebih kecil. Jadi pengunjung akan men-zoom konten untuk melihat lebih detail.

Agar developer bisa mengendalikan width viewport, ada meta tag baru yang diperkenalkan yang memungkinkan lebar, dan skala awal viewport bisa diset. Jadi Anda bisa mengeset browser mobile untuk me-render situs pada berbagai lebar viewport.

Berikut ini contoh viewport width, dan height relatif pada jendela browser.



Gambar 1.8 Width, dan Height dari Viewport

1.1.7 Breakpoint

Selain viewport, Anda perlu memahami apa itu breakpoint. Breakpoint di desain responsif adalah nilai lebar, di mana website akan mengubah layout berdasarkan deklarasi media query.

Umumnya, situs responsif memiliki dua atau lebih breakpoint, tergantung kepada breakpoint yang ditujukan untuk piranti tertentu. Beberapa breakpoint yang digunakan adalah:

- Piranti sangat kecil, misalnya telepon (<768px).
- Piranti kecil, misalnya tablet (>768px, dan <992px).
- Piranti medium, misalnya untuk netbook dan komputer desktop (>992px, dan <1200px).
- Piranti besar, seperti komputer desktop (>1200px).

Selain breakpoint, terminologi yang perlu Anda ketahui adalah State, yaitu versi website di antara tiap breakpoint. Jadi piranti mobile, tablet, dan desktop adalah state Anda, di antaranya ada 2 break point. Perlu memahami bahwa query media merespon width dari viewport dan bukannya width dari layar, sehingga Anda bisa dengan mudah mengatur browser untuk mengetes breakpoint.

1.2 Fitur Baru HTML 5 dan CSS 3

Desain web responsif dibangun di atas HTML5 dan CSS 3. Sebelum membahas ke tahapan desain responsif, Anda perlu melihat beberapa fitur baru dari HTML 5, dan CSS 3 ini.

HTML 5 adalah spesifikasi terbaru dari bahasa HTML yang disetujui oleh W3C (*World Wide Web Consortium*). Spesifikasi HTML 5 adalah bagian dari framework lebih besar yang disebut “Open Web Platform” oleh W3C. Framework ini memungkinkan semua orang membuat website, dan aplikasi web yang bisa dijalankan di platform web manapun.

Ada banyak keunggulan HTML 5 dibandingkan versi sebelumnya, antara lain:

- Aksesibilitas: HTML 5 memudahkan Anda membuat situs lebih aksesibel. Semantik tag baru dari HTML 5 memung-

kinkan screen reader untuk lebih mudah mengidentifikasi tipe konten. Ini memudahkan untuk menyediakan user experience yang lebih baik. HTML 5 juga mendukung ARIA (*Accessible Rich Internet Application*) data role, yang memungkinkan Anda mengalokasikan role ke bagian konten dari section. Ini sangat berguna terutama kalau Anda meng-update konten halaman menggunakan Java-Script, karena Anda bisa mendefinisikan role ARIA yang memperlihatkan bagian dari halaman untuk pengubahan dan menotifikasi user.

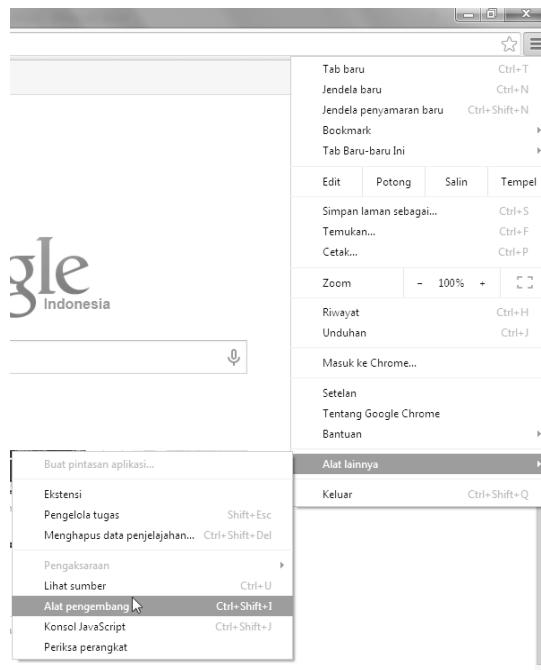
- Dukungan Video, dan Audio: HTML 5 memiliki dukungan natif untuk codec audio, dan video. Salah satu keuntungannya, ini lebih mendukung piranti mobile karena bekerja di browser tanpa plugin seperti Adobe Flash.
- Storage Lebih Baik: Sebelum HTML 5, satu-satunya cara untuk menyimpan data di client side adalah dengan menggunakan cookies, cara ini memiliki kekurangan karena objek ini juga dikirim ke server walaupun tidak digunakan di server. HTML 5 memperkenalkan penyimpanan DOM (*Document Object Model*). Keuntungan DOM adalah data tetap ada di client side, sehingga tidak memengaruhi ukuran HTTP request dan memungkinkan Anda menyimpan data lebih banyak. Saat ini, DOM storage memungkinkan menyimpan data hingga 5 MB per domain.
- Interaksi baru: API JavaScript baru dari HTML 5 memungkinkan Anda menambahkan interaksi baru, misalnya ada API baru seperti drag and drop, geolocation, and history. Tujuan dari interaksi baru ini untuk memungkinkan Anda membangun aplikasi yang lebih kaya dan lebih mudah pengaksesan antarmukanya.
- Canvas: HTML 5 memperkenalkan elemen canvas, ini adalah tag baru HTML 5 yang bisa digunakan untuk drawing, ini memungkinkan Anda membangun aplikasi internet yang kaya, mirip dengan Adobe Flash.
- Mobile: HTML 5 megenalkan beberapa perkembangan baru untuk piranti mobile, seperti tag viewport yang memungkinkan Anda menentukan width viewport serta setting zoom-nya. Ada juga tag yang spesifik untuk piranti tertentu

yang memungkinkan developer berinteraksi dengan fitur browser spesifik, misalnya menggunakan tag meta, developer bisa memberitahukan iOS bahwa jika di-bookmark di homepage, harus ditampilkan sebagai aplikasi full screen.

1.2.1 Mengeset Browser untuk Simulasi

Untuk mencoba beberapa kode HTML dan CSS saat berlatih di buku ini, idealnya Anda mencoba menggunakan beberapa piranti. Tapi Anda tidak perlu membeli piranti satu demi satu, karena browser Chrome menyediakan fasilitas simulasi browser piranti mobile. Cara mengaktifkannya seperti berikut:

1. Klik pada tombol Options, kemudian klik pada menu Atur lainnya > Alat pengembang.



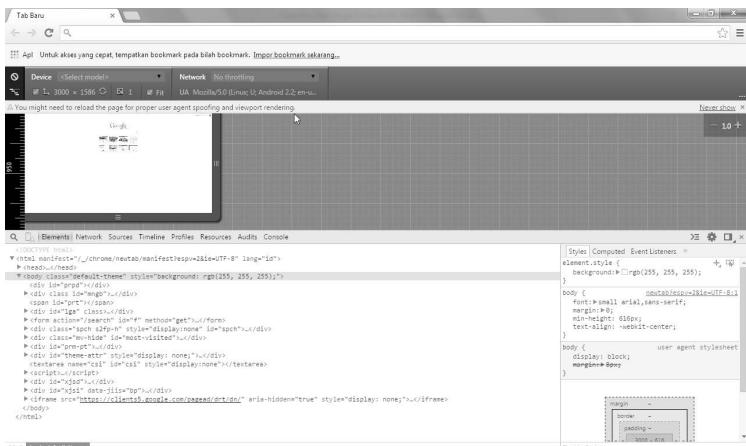
Gambar 1.9 Klik menu Alat pengembang

2. Di bagian bawah langsung muncul Element Inspector, Anda bisa klik pada tombol ikon **Toggle device mode**.



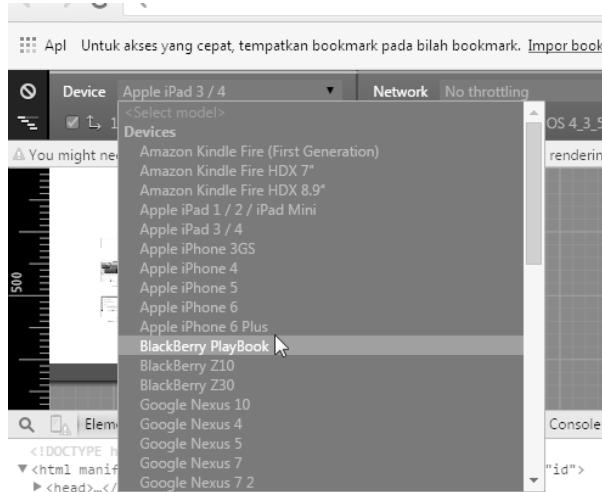
Gambar 1.10 Toggle device mode

3. Di bagian atas langsung muncul simulasi piranti mobile.



Gambar 1.11 Simulasi piranti mobile

4. Anda bisa memilih piranti yang disimulasikan dengan memilih pada combobox **Device**.



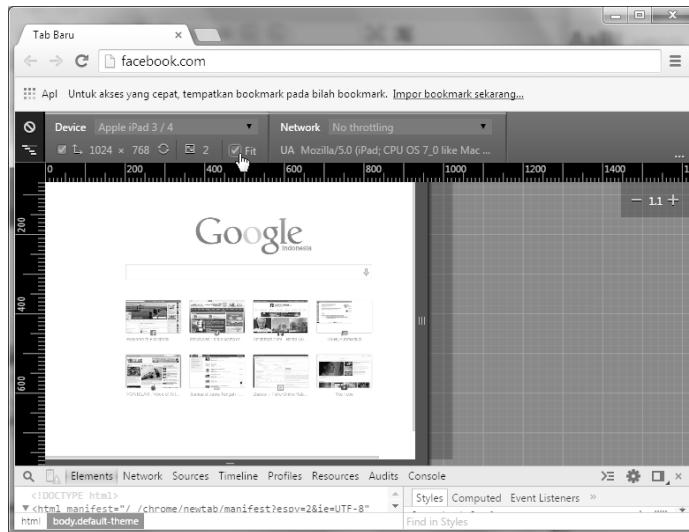
Gambar 1.12 Memilih piranti dengan memilih combobox Device

5. Maka muncul browser sesuai piranti device yang diinginkan.



Gambar 1.13 Browser berubah sesuai piranti yang dipilih

6. Anda bisa mengubah tampilan fit atau tidak, kalau fit akan sesuai dengan jendela yang dibuka, kalau tidak maka akan sesuai ukuran piksel sebenarnya.



Gambar 1.14 tampilan berubah ketika ukuran browser diperbesar, tapi diset ke Fit

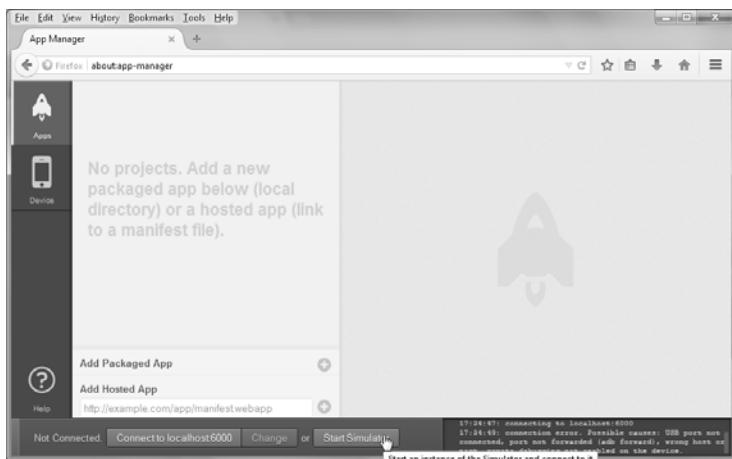
1.2.2 Simulator Android dan Firefox Emulator

Anda juga bisa menggunakan Android Emulator untuk mencoba kode-kode di buku ini. Caranya seperti berikut:

1. Download Android SDK dari <http://developer.android.com/sdk>.
2. Ekstrak file zip tersebut ke C:\Program Files\Android.
3. Jalankan **SDK Manager.exe**.

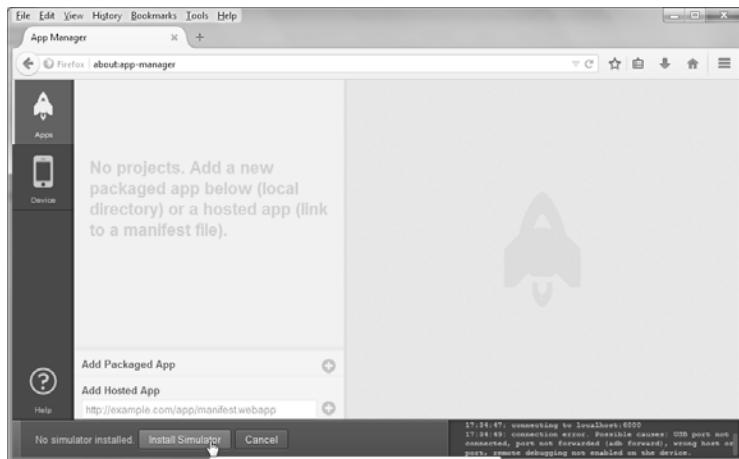
Adapun untuk menjalankan simulator di Firefox, Anda perlu melakukan langkah-langkah berikut ini:

1. Jalankan firefox.
2. ketikkan “about:app-manager” di kotak url.



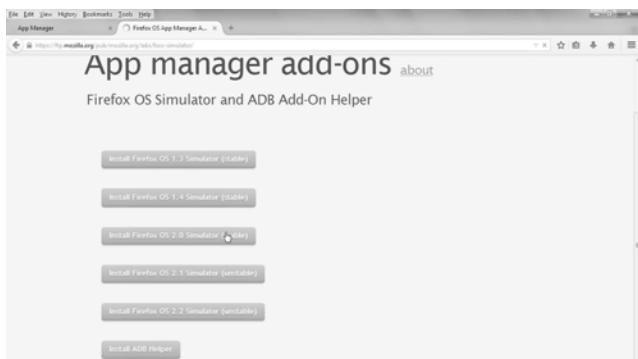
Gambar 1.15 Halaman about:app-manager

3. Di bagian bawah, klik **Start Simulator**. Karena belum ada, akan muncul tombol **Install Simulator**, klik tombol tersebut.



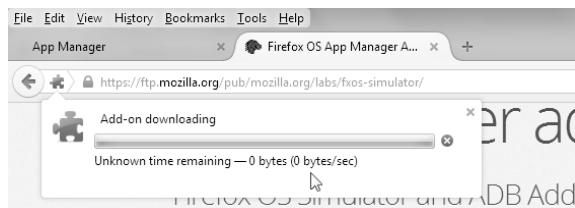
Gambar 1.16 Klik pada Install Simulator

4. Muncul beberapa versi simulator, pilih versi terakhir yang stabil.



Gambar 1.17 Pemilihan versi terakhir Firefox simulator

5. Tunggu hingga file di-download dan diinstal.



Gambar 1.18 File di-download dan diinstal

6. Jalankan lagi program, kemudian jalankan Firefox OS Simulator. Akan muncul tampilan seperti contoh berikut.



Gambar 1.19 Tampilan OS Simulator di Firefox

1.3 Media Query dan Keunggulannya

Media query di CSS 3 adalah salah satu komponen utama dari desain responsif karena memungkinkan developer untuk menarikkan style CSS spesifik berdasarkan kemampuan piranti. Salah satu keunggulannya adalah bisa menentukan width viewport, tapi sebenarnya masih banyak lainnya. Akan dijelaskan fitur, dan keunggulan media query di subbab ini.

1.3.1 Pengenalan Media Query

Media query adalah dasar dari desain responsif, yang memungkinkan Anda mengubah tampilan dari website berdasarkan query yang telah ditentukan. Sebagai perkenalan, berikut ini tipe media yang awalnya diperkenalkan di spesifikasi CSS 2.

```
<link rel="stylesheet" type="text/css" href="all.css"
media="all" />
<link rel="stylesheet" type="text/css" href="screen.css"
media="screen" />
<link rel="stylesheet" type="text/css" href="print.css"
media="print" />
```

Di contoh di atas, ada media type screen, dan print. Sebenarnya jenis media ada banyak yang mendukung beberapa tipe piranti, antara lain:

- **all:** semua piranti.
- **aural:** speech synthesizer.
- **braille:** piranti feedback braille.
- **embossed:** printer braille.
- **handheld:** piranti genggam kecil atau handheld.
- **print:** printer.
- **projection:** presentase yang diproyeksikan, seperti proyektor, dan slide proyektor.
- **screen:** layar komputer.
- **tty:** media yang menggunakan grid karakter fixed-pitch, seperti teletype, dan terminal.
- **tv:** piranti tipe televisi.

Selain bisa menentukan berbagai style untuk tiap tipe media, Anda juga bisa memasukkan style ini ke stylesheet menggunakan beberapa sintaks berikut:

```
@media print{  
    /*style print di sini*/  
}
```

Ada beberapa keuntungan menggunakan berbagai tipe media query di style sheet utama, yaitu:

- Mengurangi jumlah request HTTP, pentingkatan jumlah request HTTP bisa mengurangi performa loading dari site.
- Memisahkan style sheet bisa mengurangi waktu bloking dari pemuatan halaman. Karena mayoritas browser akan menunggu stylesheet individual termuati dulu semuanya, baru me-render tampilan halaman web.

Query media ditambahkan di CSS 3 sebagai ekstensi dari tipe media dengan tujuan agar developer memiliki kontrol lebih dalam terhadap website-nya ketika ditampilkan di browser, dan piranti yang berbeda. Ide dasarnya, daripada Anda membuat berbagai versi website untuk berbagai platform, Anda bisa menggunakan satu website saja untuk berbagai platform dan menuliskan CSS yang menentukan atribut, properti atau karakteristik berbeda dari piranti. Artinya bukannya menargetkan piranti berdasarkan tipe, Anda bisa menargetkan karakteristik khusus dari piranti.

1.3.2 Pengetesan Media Query

Ada berbagai tipe query yang berbeda yang bisa dites oleh media query, berikut ini beberapa di antaranya:

width | min-width | max-width

Query width memungkinkan Anda mengetes width dari viewport browser. Ini memungkinkan Anda menargetkan style pada width browser tertentu. Anda tidak hanya bisa mengetes di set width, tapi juga menargetkan width minimum dan maksimum dari viewport browser. Ini artinya, Anda bisa meng-query width dari berbagai piranti. Ekspresi media width adalah salah satu media expression untuk mengadaptasi site agar menjadi responsif.

height | min-height | max-height

Query height memungkinkan Anda mengetes height dari viewport browser. Mirip dengan query width, Anda bisa menargetkan pada height tertentu, height minimum atau height maksimum. Walaupun query height jarang dipakai dibandingan dengan width, query height sangat bermanfaat ketika Anda ingin memastikan konten bisa dilihat atau tidak bisa dilihat ketika loading pertama kali. Karena Anda bisa menggunakannya untuk menyesuaikan height dari konten, sehingga mencocoki height dari viewport.

device-width | min-device-width | max-device-width

Query device-width memungkinkan Anda mengetes width dari piranti, Anda bisa menargetkan pada width tertentu, width minimum atau width maksimum. Perbedaan antara width dan device-width adalah width berdasarkan width-nya browser, sementara device-width berhubungan dengan width dari screen layar. Walaupun ada beberapa kasus menggunakan device-width, masalahnya jika user me-resize browser di desktop, situs tidak akan me-resize ke fit. Jika Anda menggunakan meta tag viewport, dan mengatur width sama dengan device-width, Anda bisa menggunakan query width.

device-height | min-device-height | max-device-height

Query device-height memungkinkan Anda mengetes height dari piranti, Anda bisa menargetkan height tertentu, height minimum atau height maksimum. Perbedaan antara device-height, dan height adalah device-height berhubungan dengan height dari layar sementara height berhubungan dengan height dari viewport. Perbedaan ini sangat penting di piranti apabila Anda bisa mengatur ulang ukuran browser.

aspect-ratio | min-aspect-ratio | max-aspect ratio

Query aspect-ratio memungkinkan Anda mengetes aspect ratio dari viewport piranti yang digunakan. Aspect ratio dari piranti adalah perbandingan antara panjang dari bagian yang panjang dari piranti dibandingkan dengan lebar dari bagian piranti yang lebih pendek. Query aspect-ratio bisa sangat berguna, terutama kalau Anda ingin menargetkan asset untuk mencocoki aspect ratio dari piranti, termasuk menampilkan video, yang dioptimalkan ke piranti user.

device-aspect-ratio | min-device-aspect-ratio | max-device-aspect-ratio

Query device-aspect-ratio memungkinkan Anda mengetes aspect-ratio dari piranti. Perbedaannya adalah, device-aspect-ratio berhubungan dengan aspect-ratio dari piranti, sementara aspect-ratio berhubungan dengan viewport saja. Perbedaan ini penting pada piranti di mana Anda bisa me-resize jendela browser, karena aspect-ratio bersifat fluid sama seperti ukuran viewport. Padahal ukuran device-aspect-ratio tidak akan berubah.

color | min-color | max-color

Query color memungkinkan Anda mengetes kemampuan warna dari piranti berdasarkan jumlah bit per komponen warna.

color-index | min-color | max-color

Query color-index memungkinkan Anda mengetes berapa jumlah warna yang didukung oleh piranti dan nilainya harus berupa integer dan tidak bisa negatif.

monochrome | min-monochrome | max-monochrome

Query monochrome memungkinkan Anda mengetes bits per piksel dari piranti monokrom.

resolution | min-resolution | max-resolution

Query resolution memungkinkan Anda mengetes densitas piksel dari piranti. Query resolution menerima tiga tipe nilai, yaitu: dpi (dot per css inch), dpcm (dot per css centimeter), dan dppx (dots per pixel). Anda disarankan menggunakan dppx yang merupakan tambahan terakhir ke spesifikasi dibanding dengan dpcm dan dpi. Keuntungan dppx adalah berkaitan langsung dengan kerapatan piksel dari layar, sehingga lebih mudah dipahami oleh developer.

scan

Query scan memungkinkan Anda mengetes proses scanning dari piranti. Ini spesifik untuk televisi yang memiliki tipe scanning progresif atau interlace. Perbedaannya, tampilan progresif menampilkan semua baris di display sekaligus, dan display interlace menampilkan semua baris ganjil kemudian menampilkan garis genap, ini untuk “menipu” mata Anda bahwa Anda melihat semua baris.

grid

Query grid memungkinkan Anda mengetes apakah piranti yang Anda gunakan adalah piranti tipe grid atau tipe bitmap. Ada dua kemungkinan nilai, jika nilai diset ke 1, maka query akan mengaktifkan CSS jika tampilan display dari piranti berbasis grid, misalnya sebuah ponsel dengan satu font fixed. Alternatifnya, Anda bisa mengecek semua piranti dengan menyetting nilainya ke 0.

orientation

Query orientation memungkinkan Anda mengetes apakah piranti Anda sedang dalam posisi landscape atau portrait, dan menerapkan CSS Anda sesuai dengan hasil pengetesannya. Contoh penggunaannya ketika Anda ingin memindahkan antara satu kolom di orientasi portrait menjadi dua kolom di orientasi landscape. Tipe-tipe query yang bisa digunakan tergantung kepada tipe piranti, misalnya jika user menggunakan piranti aural, maka query yang mengacu ke viewport layar, seperti width dan device-width tentu tidak mungkin digunakan.

1.4 Media Query

Ketika Anda sudah familiar dengan tipe media dan media query, saatnya Anda memahami bagaimana sintaks dari media query. Sebuah media query minimalnya terdiri dari satu tipe media dan bisa ditambahkan lebih dari satu ekspresi media dengan nilai antara true atau false. Media query bisa spesifik atau ambigu sesuai keinginan Anda.

Berikut ini contoh pertama bagaimana menuliskan media query di CSS di mana Anda bisa menambahkan rule max-width mengeset nilainya ke 767px. Contoh sederhananya seperti berikut ini:

```
all and (max-width: 767px)
```

Pada contoh di atas, tipe media diset ke all, artinya diterapkan ke semua tipe media. Ada ekspresi media max-width yang diset ke 767px. Ini contohnya kalau targetnya piranti kecil (misalnya ponsel), yaitu yang kurang dari 767px. Jadi kalau width dari viewport kurang dari 767px, maka CSS akan ditampilkan.

Jika Anda ingin menerapkan CSS ke piranti layar, Anda bisa mengeset screen, contohnya seperti berikut:

```
screen and (max-width: 767 px)
```

Logika dari media query ini mirip dengan contoh sebelumnya, hanya saja perbedannya Anda mengubah tipe media ke screen. Logikanya, semua perangkat dari tipe screen, kalau memiliki viewport kurang dari atau sama dengan 767 px, maka CSS akan diaktifkan.

Ada juga operator logik Not yang digunakan untuk menegasi. Misalnya jika Anda ingin menerapkan CSS ke semua tipe media selain layar, Anda bisa membalik kode dengan menggunakan Not seperti contoh berikut ini:

```
not screen and (max-width: 767px)
```

jadi untuk contoh di atas, apabila browser bukan perangkat screen dengan width kurang dari atau sama dengan 767px, maka CSS akan diaktifkan.

Ada juga operator logika Only yang digunakan untuk menghindari browser yang tidak mendukung ekspresi media agar tidak memproses media query. Tanpa operator only, browser lama akan membaca media type, tapi tidak memahami dan tidak akan dianggap.

Contohnya seperti berikut:

```
only screen and (max-width: 767px)
```

Anda bisa membuat media query lebih dari satu. Untuk merangkaikan ekspresi media query, Anda perlu menggunakan kata kunci di antara ekspresi, misalnya ingin CSS diaktifkan untuk min-width = 768px, dan max-width 1023px. Ini untuk memastikan apakah piranti tersebut adalah piranti kecil.

```
screen and (min-width:768px) and (max-width:1023px)
```

Kode di atas mengatakan bahwa jika tipe media screen, dan viewport sama atau lebih besar dari 768 px atau kurang dari atau sama dengan 1023 px, maka style sheet akan diaktifkan.

1.4.1 Merangkai Media Query

Sejauh ini, contoh media query hanya tunggal. Tapi Anda bisa merangkaikan media query sebenarnya, sehingga CSS bisa diterapkan di berbagai kemungkinan. Misalnya, Anda bisa membuat CSS yang diterapkan jika ada salah satu query yang mengembalikan nilai True.

Tiap media query bisa memiliki tipe media tersendiri, ekspresi media tersendiri, dan tiap-tiap media query menargetkan pada fitur, tipe, dan state piranti.

Cara untuk merangkai dan menggabungkan media query cukup mudah, Anda tinggal menambahkan tanda koma antara tiap query, maka tiap query akan diakses secara individual untuk melihat apakah nilainya true. Contohnya seperti ini:

```
screen and (min-width:768px) and (max-width:1023px), screen  
and (orientation:portrait) {...}
```

Di contoh di atas, ada dua media query, media query pertama bernilai true jika width dari browser sama atau lebih dengan 768px, dan kurang atau sama dengan 1023px. Media query yang kedua bernilai true jika orientasi dari piranti adalah portrait.

Salah satu skenario di mana Anda perlu menggunakan lebih dari satu media query misalnya ketika Anda ingin membuat fitur yang diimplementasikan secara berbeda di beberapa browser yang berbeda. Misalnya jika Anda ingin menggunakan min-resolution.

```
@media only screen and ( -webkit-min-device-pixel-ratio: 2 ),  
only screen and ( -o-min-device-pixel-ratio: 2/1 ),  
only screen and ( min--moz-device-pixel-ratio: 2 ),  
only screen and ( min-device-pixel-ratio: 2 ),  
only screen and ( min-resolution: 192dpi ),  
only screen and ( min-resolution: 2dppx ) {  
/* style resolusi tinggi di sini */  
}
```

Di awal-awal perkenalan media query CSS3, dibutuhkan teknik untuk menargetkan densitas piksel besar, sehingga WebKit mengimplementasikan device-pixel-ratio di prefix vendor. Karena spesifikasi ini sudah matang, maka implementasi yang benar menggunakan ekspresi resolution saja, namun browser-browser lama masih membutuhkan media query lebih dari satu.

Berikut ini beberapa ukuran yang disarankan digunakan untuk media-query:

- 320px
- 480px
- 600px
- 900px
- 1200px

1.4.2 Menggunakan Media Query di CSS

Setelah Anda memahami sintaks penulisan query media, Anda perlu mengetahui bagaimana menerapkannya ke page. Tiga metode yang ada antara lain:

- Memisahkan style sheet untuk tiap media query.
- Menggunakan @import di file CSS utama untuk me-load file CSS jika diinginkan.
- Menggunakan media query di dalam file CSS.

Cara pertama adalah menggunakan style sheet terpisah. Pada umumnya, ketika menggunakan style sheet screen atau print dan style sheet dengan aturan-aturan di mana akan menerapkan tipe media. Anda juga bisa melakukan ini ke media query, sehingga akan mengaktifkan atau tidak mengaktifkan style sheet.

Untuk menambahkan rule di CSS terpisah, Anda bisa menambahkan ke atribut media, seperti terlihat di contoh berikut:

```
<link rel="stylesheet" media ="screen and (min-width:768px)  
and (max-width:1023px)" href="galaxytab.css">
```

Pada contoh di atas, media query mengecek apakah user menggunakan perangkat kecil dengan width antara 768 px, dan 1023 px. Keuntungan menggunakan style sheet terpisah adalah Anda bisa mengelola kode dengan mudah. Kalau Anda ingin mengutak-atik style sheet saja, Anda bisa mengutak-atik di file CSS, tidak perlu membuka file HTML utamanya.

Jika Anda ingin menggunakan style sheet terpisah untuk setiap media query, tapi tidak ingin mendefinisikan di HTML, Anda bisa menggunakan sintaks @import dari CSS dengan media query yang diterapkan. Sintaks @import ini bisa dipecah ke dalam tiga bagian:

- Deklarasi @import.
- Url ke file CSS yang akan dimasukkan.
- Satu media query atau lebih yang dipakai.

Berikut ini contoh sintaks bagaimana menggunakan media query:

```
@import url("galaxytab.css") screen and (min-width:768px) and  
(max-width: 1023px);
```

Di contoh ini, Anda menggunakan media query untuk mengecek apakah pengunjung menggunakan piranti yang width-nya antara 768 px dan 1023px sebelum kemudian memuati file stylesheet galaxytab.css.

Anda juga bisa mendefinisikan media query di dalam stylesheet-nya. Ini memungkinkan Anda mendefinisikan style baru, dan kemudian menimpa pada style jika kondisinya True. Untuk menuliskan media query di dokumen CSS, Anda bisa menggunakan sintaks @media.

Berikut ini pembagian ketika menggunakan sintaks ini:

- Deklarasi @media.
- Satu ekspresi media atau lebih.

Contoh sintaksnya seperti berikut ini:

```
@media only screen and (min-width:768px) and (max-  
width:1023px) {  
/*style yang diisikan*/  
}
```

Keuntungan menggunakan CSS dalam satu file adalah mengurangi jumlah HTTP request dan menurunkan kemungkinan blocking.

Di bab ini, Anda bisa melihat bahwa banyak sekali faktor yang perlu diketahui untuk membuat website responsif. Sebenarnya, semua berputar dari konten utama HTML, kemudian dimodifikasi menggunakan CSS yang menggunakan media query.

Media query adalah pondasi dari desain web responsif, sehingga developer seperti Anda harus memahami bagaimana menggunakan dan mengoptimalkan website untuk bisa bekerja di banyak piranti. Dengan banyaknya piranti mobile yang terus bermunculan, developer juga tidak boleh gaptek untuk mengetahui apakah ada perangkat-perangkat baru lagi.

BAB 2

HTML 5

Hypertext Markup Language (HTML) adalah sebuah bahasa untuk menampilkan konten di Web. HTML sendiri adalah bahasa pemrograman yang bebas yang artinya tidak dimiliki oleh siapa pun, pengembangannya dilakukan oleh banyak orang di banyak negara dan bisa dikatakan sebagai sebuah bahasa yang dikembangkan bersama-sama secara global.

Sebuah dokumen HTML sendiri adalah dokumen teks yang dapat diedit oleh editor teks apa pun. Dokumen HTML punya beberapa elemen yang dikelilingi oleh tag-teks yang dimulai dengan < dan berakhir dengan sebuah >.

Contoh dari tag adalah . Tag ini fungsinya menampilkan gambar dari file gambar bernama "gambar.gif". Nantinya gambar akan ditampilkan jika file HTML ini dibuka di browser.

2.1 Pengenalan

Dasar HTML dapat mencakup petunjuk untuk memformat dalam bahasa yang disebut *Cascading Style Sheets* (CSS) dan program untuk interaksi dalam bahasa yang disebut JavaScript. Halaman web yang biasa Anda lihat ketika browsing di internet selalu ditulis menggunakan bahasa HTML. Tak peduli halaman web tersebut dibuat menggunakan berbagai bahasa seperti PHP, JSP, ASPX atau cold fusion, hasil akhirnya yang ditampilkan oleh browser tetap adalah HTML murni.

HTML awalnya hanya didesain sebagai bahasa untuk pertukaran dokumen teknis dan saintifik. Karena itu, HTML didesain secara sederhana, HTML menggunakan tag-tag yang dapat dipahami dengan mudah untuk membuat dokumen sederhana. Dalam waktu yang singkat, HTML kini telah populer dan dikuasai banyak orang untuk membuat halaman web.

Browser, seperti Firefox dan Chrome, menafsirkan HTML bersama dengan CSS dan JavaScript untuk menghasilkan apa yang kita lihat ketika kita mengunjungi sebuah situs web.

HTML adalah bahasa yang mengatur bagaimana tampilan isi dari situs web. Di dalam HTML ada tag-tag di mana tag berfungsi menyediakan informasi berkaitan dengan sifat dan struktur konten serta referensi untuk gambar dan media lainnya.

HTML juga akan berkaitan erat dengan CSS. CSS menentukan format dari konten. Dengan adanya CSS, konten website bisa lebih seragam hingga hasilnya bagus dan format ini dapat diatur dengan cara yang berbeda.

Bericara mengenai HTML tidak terlepas dari JavaScript, yang merupakan bahasa pemrograman client-side untuk memberikan efek dinamis. JavaScript mampu membuat web terlihat lebih dinamis dan interaktif. Jadi HTML biasanya tidak berdiri sendiri, sebuah web page biasanya hasil kolaborasi antara HTML, CSS, dan JavaScript.

Tapi dasar yang harus Anda kuasai adalah HTML, nantinya Anda bisa menggunakan CSS dan JavaScript untuk mempercantik website Anda.

Versi terbaru dari HTML (yang juga membahas CSS dan JavaScript) adalah HTML5. Banyak orang yang menyambut versi baru ini dengan kegembiraan karena fitur-fitur baru seperti seperti kanvas untuk menampilkan gambar dan animasi, dukungan untuk video dan audio, serta tag baru untuk mendefinisikan elemen-elemen dokumen seperti header, konten, dan footer.

Jadi dengan dengan HTML5, Anda dapat membuat situs, web canggih yang sangat interaktif. Memang hingga saat ini tidak semua browser bisa mem-parsing dan menampilkan semua fitur, tetapi browser yang berbobot sudah bisa menampilkannya. Dan Anda pun mulai bisa belajar HTML5, CSS, dan JavaScript dengan spesifikasi baru.

2.1.1 Struktur Dasar HTML

Elemen HTML dimulai dengan tag awal, yang diikuti dengan isi elemen dan tag akhir. Tag berakhir termasuk simbol / diikuti oleh tipe elemen, misalnya </HEAD>. Sebuah elemen HTML dapat bersarang di dalam elemen lainnya. Sebuah dokumen HTML standar terlihat seperti ini:

```
<html>
  <head>
    <title>Ini judul dokumen HTML </title>
  </head>
  <body>
    Teks ini adalah teks yang muncul di Body dari dokumen
  </body>
</html>
```

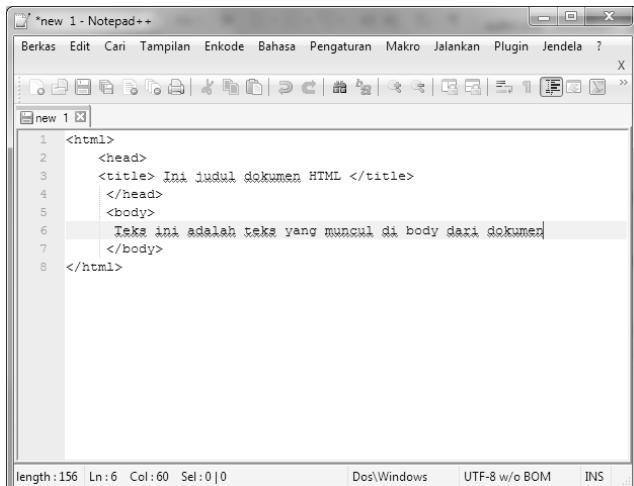
Kode di atas penulis beri indentasi untuk membuat program bisa terbaca lebih jelas, tapi HTML sebenarnya mengabaikan indentasi ini (sering juga disebut spasi atau whitespace). Jadi tidak wajib bagi Anda untuk membuat indentasi di kode Anda sendiri.

Dokumen HTML terdiri dari unsur HTML, ditandai dengan tag pembuka <html>, dan diakhiri dengan tag penutup berupa </ html>. Dokumen HTML biasanya memiliki HEAD, dan elemen BODY. HEAD ini di dalamnya berisi judul atau tag <title>. Judul nantinya muncul di bagian title bar dari browser.

Anda bisa mengetikkan kode HTML di atas menggunakan sembarang editor teks. Yang paling mudah adalah menggunakan Notepad yang merupakan aplikasi bawaan dari MS Windows.

Setelah mengetikkan, Anda bisa langsung menyimpan teks menjadi kode HTML menggunakan cara berikut:

1. Buka Notepad, kemudian tulis contoh dan pastikan kode yang ada semuanya sudah benar.



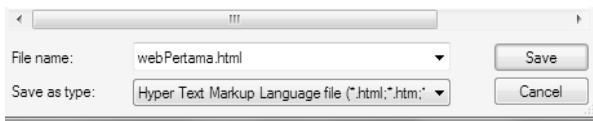
A screenshot of the Notepad++ application window. The title bar says "new 1 - Notepad++". The menu bar includes "Berkas", "Edit", "Cari", "Tampilan", "Enkode", "Bahasa", "Pengaturan", "Makro", "Jalankan", "Plugin", "Jendela", and "?". Below the menu is a toolbar with various icons. The main editor area contains the following HTML code:

```
1 <html>
2   <head>
3     <title> Ini judul dokumen HTML </title>
4   </head>
5   <body>
6     Teks ini adalah teks yang muncul di body dari dokumen
7   </body>
8 </html>
```

The status bar at the bottom shows "length: 156 Ln: 6 Col: 60 Sel: 0 | 0 Dos\Windows UTF-8 w/o BOM INS".

Gambar 2.1 Pengetikan kode HTML di Notepad

2. Tekan **CTRL + S** atau klik **File > Save** untuk menyimpan.
3. Muncul jendela **Save As**, pilih **All Files** di combobox **Save as type**.



Gambar 2.2 Save as type

4. Isikan nama file HTML dengan ekstensi **.html** di **File name**.
5. Untuk melihat hasilnya di browser, Anda bisa klik dua kali pada file HTML yang sudah tersimpan. Hasilnya, tampilan akan menjadi seperti berikut.

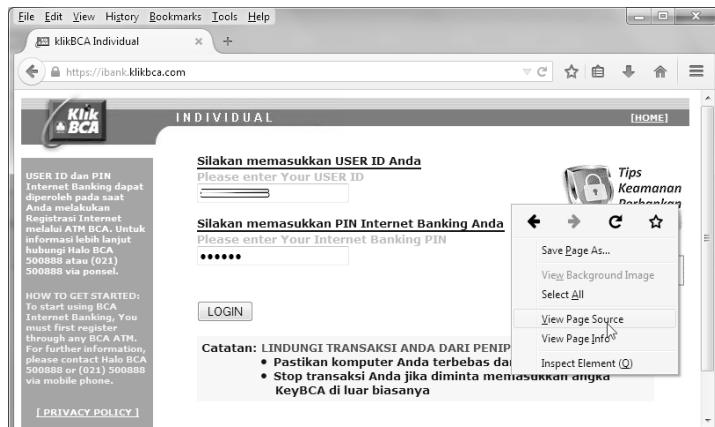
Teks ini adalah teks yang muncul di body dari dokumen

Gambar 2.3 Tampilan terlihat di Notepad

2.1.2 Mengetahui Kode HTML Sebuah Web Page

Untuk mengetahui kode HTML sebuah website, coba lakukan langkah berikut ini:

1. Buka sebuah website menggunakan browser.
2. Klik **View > Page Source** atau klik kanan, dan pilih menu **View Page Source**.



Gambar 2.4 View > Page Source untuk melihat source halaman

3. Maka kode-kode HTML yang digunakan akan terlihat seperti halnya di gambar berikut.

```
Source of: https://ibank.klikbca.com/ - Mozilla Firefox
File Edit View Help
29
30 (function() {
31   var bt="text/java",z=document,fh=z.getElementsByTagName('head')[0],k
32   var y=z.createElement(k);y.async=true;y.type=bt+k;y.src=[j,"static/re
33   fh.appendChild(y);
34 });
35 </script>
36
37 </head>
38
39 <body topmargin="0" marginheight="0" bgcolor="white" link="white" al
40   <div id="gotohome" style="position: absolute; left:701px; top:25px;
41   <div id="Layer1" style="position: absolute; left:657px; top:170px;
42   <p align="center">
43     <!-- <font face="Verdana, Arial, Helvetica, sans-serif" size="1">R
44     BCA is secured with<br>
45     SSL 2048 bit encryption.<br><br>
46   </font>-->
47   <a href="http://secure.omniroot.com/en/find/sealct.cfm?cn=ibank.kl
48   <IMG SRC="https://ibank.klikbca.com/images/exseal2.gif;bcaa7d+a38b
49   </div>
50   <div id="Layer2" style="position: absolute; left:620px; top:75px; w
51     <a href="http://www.bca.co.id/id/tips-keamanan/tips-keamanan.jsp
52     <IMG SRC="https://ibank.klikbca.com/images/keamananIB.cif;bcaa180"
```

Gambar 2.5 Kode HTML dari sebuah webpage ditampilkan di source

4. Kode HTML adalah kode scripting, artinya hanya berupa teks murni seperti gambar di atas. Sehingga, walaupun Anda menyisipkan kode PHP di script halaman web, kode HTML yang dihasilkan tetap berupa kode HTML murni.
 5. File HTML murni yang berekstensi .htm atau .html akan langsung diperoleh dari server.
 6. Sementara file HTML yang memiliki skrip PHP umumnya berekstensi .php. Karenanya skrip .php akan diolah/diparsing terlebih dahulu di PHP parser yang ada di server untuk menghasilkan tampilan kode HTML murni.
 7. Begitu pula yang lainnya, seperti JSP dan ASPX.

2.1.3 Mengenal XHTML

XHTML merupakan singkatan dari *Extensible Hypertext Markup Language*. XHTML merupakan bentuk reformulasi dari HTML menggunakan paradigma XML.

Kenapa perlu XHTML? Karena XHTML adalah bentuk HTML yang lebih bagus dan rapih, karenanya dapat di-parsing di banyak platform browser, seperti di web browser biasa, lalu di browser ponsel, dan gadget lain.

XHTML memanfaatkan kelebihan XML untuk memperkaya HTML. Keuntungan menuliskan HTML dalam bentuk XML antara lain:

- Di XML, Anda bisa memperkenalkan elemen-elemen baru atau atribut elemen baru. Sehingga Anda bisa membuat spesifikasi modul-modul baru.
- XHTML lebih mudah mengakomodasi interoperabilitas dari cara mengakses halaman web. Sehingga halaman web yang dibuat menggunakan XHTML bisa menghasilkan tampilan yang lebih seragam.

Dalam prakteknya, menulis dokumen XHTML hampir sama seperti menulis dokumen HTML, hanya saja ada beberapa ciri khas dokumen XHTML, seperti berikut:

- Ditulis dengan huruf kecil, baik nama tag maupun atributnya.
- Nilai atribut haruslah ada di dalam tanda petik dua.
- Anda tidak bisa menggunakan atribut yang berdiri sendiri (*stand alone*) di XHTML. Jika menemukan atribut yang berdiri sendiri, Anda bisa mengalokasikan ke variabel dummy. Misalnya noborder = "noborder".
- Elemen yang kosong perlu diakhiri dengan `>`. Contohnya `
` di atas yang bisa saja dilakukan di HTML, namun tidak bisa dilakukan di XHTML.
- Elemen `<a>` tidak bisa mengandung elemen-elemen `<a>` lainnya.
- Elemen `<button>` tidak boleh mengandung elemen `<input>`, `<select>`, `<textarea>`, `<label>`, `<button>`, `<form>`, `<fieldset>`, `<iframe>`, atau `<isindex>`.
- Elemen `<form>` tidak boleh mengandung elemen-elemen `<form>` lainnya.
- Elemen `<label>` tidak boleh mengandung elemen label lainnya.
- Elemen `<pre>` tidak boleh mengandung elemen ``, `<object>`, `<big>`, `<small>`, `<sub>`, atau `<sup>`.

- Tidak boleh menggunakan karakter sensitif. Misalnya, ketika nilai atribut mengandung tanda ampersand (&), maka ampserand-nya harus ditulis sebagai &;

Untuk validasi, Anda bisa menggunakan <http://validator.w3.org> untuk memvalidasi apakah file yang sudah Anda buat sudah benar atau tidak.



Gambar 2.6 Validator w3.org berfungsi menentukan apakah file HTML yang sudah dibuat benar atau tidak

2.1.4 Menampilkan Teks

HTML adalah dokumen sederhana, berfungsi untuk menampilkan teks, dan juga halaman gambar. Berikut ini contoh meletakkan teks di dokumen HTML, dan pengaturan alignment-nya. Lihat detail kodennya untuk mengetahui cara membuat dokumen HTML seperti ini:

```
<html>
<head>
<title>Profil PT Dirgantara Indonesia</title>
</head>
<body>
<!--yang di dalam ini adalah komentar tidak akan dieksekusi -->
<hr /> <!--ini adalah horisontal line-->
<h1> Teks di Tengah</h1>
<p style="text-align:center">
```

Di era millenium baru ini perubahan-perubahan mendasar di lingkungan global , regional, maupun nasional bergerak begitu cepat. Kenichi Ohmae dalam bukunya "The Borderless world" mengungkapkan tentang kecenderungan-kecenderungan, dan logika baru pasar global yang menunjukkan semakin kecilnya peranan negara bangsa dalam kaitannya dengan ekonomi, dan bisnis.

Fakta inti bisnis dewasa ini adalah kedaulatan konsumen. Standar pelaksanaan produk sekarang ini ditetapkan di dalam pasar global oleh mereka yang membeli produk, bukan mereka yang membuatnya. Ini artinya bahwa perusahaan perlu memikir ulang bagaimana mereka mengembangkan, dan mengkomersilkan produknya.

</p>

<hr /> <!--horizontal line-->

<h1> Teks di Kanan</h1>

<p style="text-align:right">

Restrukturisasi Bisnis dilakukan dengan mengukuhkan visi, dan misi, menetapkan fokus bisnis Dirgantara Indonesia ke dalam bisnis inti (core) pesawat terbang, dan bisnis plasma (non-core). Bisnis inti adalah pesawat terbang, dan helikopter meliputi lini usaha: CN-235, N250, NC-212, helikopter.

Bisnis Plasma meliputi lini usaha: Manufacturing Services, Component Manufacturing, Tools Manufacturing, Aircraft Services, Technology & Engineering Services, Interior, Special Mission Aircraft, sistem antariksa, teknologi informasi, serta sistem hankam , dan pusat pendidikan, dan pelatihan teknologi maju (ATEC) .

</p>

<hr /> <!--horizontal line-->

<h1> Teks di Kiri</h1>

<p style="text-align:left">

Program Efisiensi Biaya difokuskan pada tiga hal: pertama, penurunan lead time; kedua, efisiensi sumber daya manusia (SDM); ketiga evaluasi struktur biaya terutama biaya beban usaha.

Dalam upaya penurunan lead time, ditargetkan pencapaian customer lead time 24 bulan untuk produk pesawat terbang. Hal ini dicapai melalui perbaikan production lead time dari rata-rata 50 bulan menjadi 30 bulan; kebijakan buffer stock untuk detail part; , dan standardisasi konfigurasi basic, dan optional.

</p>

<hr /> <!--horizontal line-->

</body>

</html>

Jika disimpan, dan dibuka di browser, tampilannya akan terlihat seperti berikut, Anda bisa melihat tiap paragraf memiliki alignment yang berbeda-beda, di tengah, kanan, dan kiri.

Teks di Tengah

Di era milenium baru ini perubahan-perubahan mendasar di lingkungan global, regional, maupun nasional bergerak begitu cepat. Kenichi Ohmae dalam bukunya "The Borderless world" mengungkapkan tentang kecenderungan-kecenderungan dan logika baru pasar global yang menunjukkan semakin kecilnya peranan negara bangsa dalam kaitannya dengan ekonomi dan bisnis. Fakta inti bisnis dewasa ini adalah kedaulatan konsumen. Standar pelaksanaan produk sekarang ini ditetapkan di dalam pasar global oleh mereka yang membeli produk, bukan mereka yang membuatnya. Ini artinya bahwa perusahaan perlu memikir ulang bagaimana mereka mengembangkan dan mengkomersilkan produknya.

Teks di Kanan

Restrukturisasi Bisnis dilakukan dengan mengukuhkan visi dan misi, menetapkan fokus bisnis Dirgantara Indonesia ke dalam bisnis inti (core) pesawat terbang dan bisnis plasma (non-core). Bisnis inti adalah pesawat terbang dan helikopter meliputi lini usaha : CN-235, N250, NC-212, helikopter.

Bisnis Plasma meliputi lini usaha : Manufacturing Services, Component Manufacturing, Tools Manufacturing, Aircraft Services, Technology & Engineering Services, Interior, Special Mission Aircraft, sistem antariksa, teknologi informasi, serta sistem hukum, dan pusat pendidikan dan pelatihan teknologi maju (ATEC).

Teks di Kiri

Program Efisiensi Biaya difokuskan pada tiga hal : pertama, penurunan lead time; kedua, efisiensi sumber daya manusia (SDM); ketiga evaluasi struktur biaya terutama biaya beban usaha. Dalam upaya pemuraman lead time, ditargetkan pencapaian customer lead time 24 bulan untuk produk pesawat terbang. Hal ini dicapai melalui perbaikan production lead time dari rata-rata 50 bulan menjadi 30 bulan; kebijakan buffer stock untuk detail part; dan standarisasi konfigurasi basic dan optional.

Gambar 2.7 Contoh tampilan kode halaman web yang alignment-nya diatur

Pada kode di atas, ada tag `<!--isi-->` yang merupakan kode untuk memberikan komentar, jadi kode di situ tidak dieksekusi. Ada juga kode `<h1>` yang menunjukkan bahwa teks itu memiliki tingkat heading1.

Anda juga bisa memberikan pemformatan pada font berupa tebal, miring, garis bawah atau coret menggunakan teknik berikut:

- Tebal dibuat menggunakan tag `` ditutup dengan ``. B di dalam tag tersebut singkatan dari bold.
- Miring dibuat menggunakan tag `<i>`, dan ditutup dengan `</i>`. Huruf I singkatan dari italic.
- Garis bawah dibuat menggunakan tag `<u>` ditutup dengan `</u>`. Huruf U singkatan dari underline.
- Coret dibuat menggunakan `<strike>`, dan ditutup dengan `</strike>`.

2.1.5 Menampilkan Link

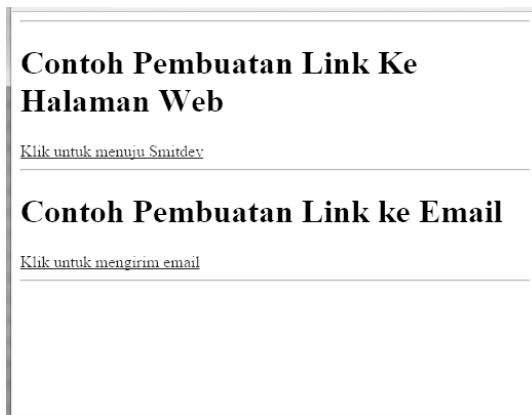
Sebuah halaman web bisa menavigasi ke halaman web lain dengan cara memberikan link. Di HTML, untuk membuat link digunakan tag `` Teks ``. Link dapat dibedakan menjadi 2 ditinjau dari URL-nya, yaitu ke URL web atau ke email.

- Untuk web, URL = `www.namasitus.com`.
- Untuk email, URL = `mailto:nama@tujuan.com`.

Berikut ini contoh kode yang bisa digunakan untuk menerapkan alamat email:

```
<html>
<head>
<title>Membuat Link</title>
</head>
<body>
<hr />
<h1> Contoh Pembuatan Link Ke Halaman Web </h1>
    <a href="http://www.smitdev.com"> Klik untuk menuju
Smitdev </a>
    <hr />
    <h1> Contoh Pembuatan Link ke Email </h1>
    <a href="mailto:user_tertentu@yahoo.com "> Klik untuk
mengirim email </a>
    <hr />
</body>
</html>
```

Simpan dan buka di browser, hasilnya terlihat tampilan seperti berikut.



Gambar 2.8 Pembuatan link

2.1.6 Membuat List

List di HTML memiliki beberapa versi, 2 yang sering digunakan adalah list bernomor, dan list tidak bernomor:

- List bernomor (ordered list) dimulai dengan tag , dan ditutup dengan tag . Tiap item list diapit dengan tag .
- List bullet (unordered list) sama seperti sebelumnya, hanya saja diawali dengan tag , dan ditutup dengan tag .

```
<html>
<head>
<title>Membuat List</title>
</head>

<body>
<hr />
<h1> Cara membuat program</h1>

<ol>
<li> Tulis script kode</li>
<li> Kompilasi</li>
<li> Buat file executable</li>
</ol>

<hr />
<h1> Komponen Mobil</h1>

<ul>
<li> Mesin</li>
<li> Kelistrikan</li>
<li> Casis</li>
</ul>

<hr />
</body>
</html>
```

Jika disimpan ke dalam format HTML, dan dibuka di browser, hasilnya seperti berikut.

Cara membuat program

1. Tulis script kode
2. Kompilasi
3. Buat file executable

Komponen Mobil

- Mesin
- Kelistrikan
- Casis

Gambar 2.9 Pembuatan list

2.1.7 Menampilkan Gambar

Untuk menampilkan gambar, Anda bisa menggunakan kode:

```

```

- Lokasi gambar bisa berupa lokasi di WWW (seperti www.lokasigambar.com/gambar.jpg) ataupun lokasi yang relatif dari halaman yang terbuka (seperti [/gambar/gambarlogo.jpg](#)).
- Teks alternatif ini berguna untuk alat penjelas gambar dan bisa juga meningkatkan aspek SEO dari website yang bersangkutan.

Buat kode seperti berikut ini:

```
<html>
<head>
<title>Membuat List</title>
</head>
<body>
<hr />
<h1> Google, mesin pencari paling top</h1>

<hr />
</body>
</html>
```

Jika disimpan, dan dibuka di browser, tampilan seperti berikut.



Gambar 2.10 Pemasukan gambar ke dokumen

2.1.8 Tabel di HTML

Tabel penting peranannya dalam halaman web, selain untuk menampilkan teks atau gambar dalam format lajur dan kolom, Anda bisa juga menggunakan tabel untuk membantu me-layout tampilan halaman.

Tabel merupakan sebuah kotak yang terdiri dari baris/row dan kolom/column. Untuk membuat tabel, Anda menggunakan tag <table> dan menutupnya dengan tag </table>. Anda bisa juga menambahkan atribut lain di tag <table> pembuka. Misalnya menentukan border, warna, dan sebagainya. Di dalam tag <table> ada beberapa tag lain yang perlu Anda pahami, yaitu:

- Tag <tr>, artinya adalah tag untuk menuliskan baris biasa di tabel. TR singkatan dari table row.
- Tag <td>, artinya adalah tag untuk menuliskan kotak di dalam baris, makanya tag <td> ada di dalam tag <tr>. TD singkatan dari table data.
- Tag <th> artinya adalah tag untuk menuliskan kotak biasa seperti <td>, namun untuk header tabel. TH singkatan dari table header.

Contoh kode pembuatan tabel, seperti berikut ini:

```
<html>
<head>
<title>Pembuatan Tabel</title>
</head>
<body>
<h1> Tabel Nilai Siswa</h1>
<table border="1" bgcolor="pink">
<tr>
<th>Nama</th>
<th>Tempat Tanggal Lahir</th>
<th>Nama Wali</th>
<th>Jurusan</th>
<th>Angkatan</th>
</tr>
<tr>
<td>Irfan Bachdim</td>
<td>Den Haag, 24 Januari 1986</td>
<td>Agus Salim</td>
<td>Pelayaran Niaga</td>
<td>2003</td>
</tr>
<tr>
<td>M Schumacer</td>
<td>Bandung, 10 September 1983</td>
<td>Asep Suhendar</td>
<td>Agronomi</td>
```

```

<td>2008</td>
</tr>
<tr>
<td>Gonzales</td>
<td>Kudus, 30 Agustus 1978</td>
<td>Toha</td>
<td>Teknik Industri</td>
<td>2000</td>
</tr>
</table>
</body>
</html>

```

Jika dilihat di browser, terlihat seperti berikut ini.

| Tabel Nilai Siswa | | | | |
|--------------------------|----------------------------|---------------|-----------------|----------|
| Nama | Tempat Tanggal Lahir | Nama Wali | Jurusan | Angkatan |
| Irfan Bachdim | Den Haag, 24 Januari 1986 | Agus Salim | Pelayaran Niaga | 2003 |
| M Schumacer | Bandung, 10 September 1983 | Asep Suhendar | Agronomi | 2008 |
| Gonzales | Kudus, 30 Agustus 1978 | Toha | Teknik Industri | 2000 |

Gambar 2.11 Pembuatan tabel HTML

Jumlah kotak di tiap baris tidak wajib sama, Anda bisa mengisikan jumlah kotak di tiap baris secara berbeda. Kemudian coba lihat hasilnya.

2.1.9 Mengenal Frame di HTML

Frame adalah tempat untuk menampilkan halaman website di dalam halaman web lainnya. Dengan menggunakan frame, Anda bisa menampilkan sebuah halaman di dalam halaman lainnya.

Bentuk frame adalah kotak segi empat yang letaknya ada di dalam jendela browser. Tiap frame bisa dimanfaatkan untuk membuka halaman web dengan URL tersendiri.

Walaupun frame sangat berguna, namun Anda juga harus membatasi penggunaannya karena jika jumlah frame terlalu banyak akan membuat user dan programmer menjadi bingung. Para praktisi mengatakan bahwa jumlah frame yang disarankan dalam sebuah halaman maksimal adalah tiga buah. Lebih dari itu, sepertinya akan membuat pengakses halaman menjadi bingung, dan programmer pun tambah pusing dalam membuat.

Untuk mengatur ukuran frame, Anda bisa menggunakan contoh berikut. Pertama adalah untuk frame yang horisontal (frame atas, dan frame bawah), sementara di bawahnya adalah frame yang vertikal (frame kiri, dan frame kanan).

```
<frameset rows="20%,80%">
<frameset cols="20%,80%">
```

Berikut ini contoh kode yang mendemokan frame:

```
<html>
<head>
<title>Penggunaan Frame</title>
</head>
<frameset rows="50%,50%">
    <!--ukuran frame atas, dan frame
bawah sama-->
<frame src="http://www.smitdev.com" name="atas" />
<frame src="http://www.smitdev.blogspot.com" name="bawah" />
<noframes>
<body>
Teks ini akan ditampilkan jika browser tidak mendukung Frame
</body>
</noframes>
</frameset>
</html>
```

Jika dibuka, maka akan diciptakan 2 frame, di mana tiap frame akan berisi halaman lainnya dari smitdev.com, dan smitdev.blogspot.com.



Gambar 2.12 Tampilan frame di HTML

2.1.10 Mengenal Scripting

HTML 5 akan sangat membutuhkan bahasa JavaScript untuk mengakses beberapa objek, seperti canvas, audio, dan video. Tapi bahkan di versi HTML sebelumnya, HTML bisa diperkaya tampilannya menggunakan JavaScript.

JavaScript adalah skrip program berbasis client yang dieksekusi oleh browser, sehingga membuat halaman web bisa melakukan tugas-tugas tambahan yang tidak bisa dilakukan oleh skrip HTML biasa. Kode JavaScript biasanya dituliskan dalam bentuk fungsi yang ditaruh di tag `<head>` yang dibuka dengan tag `<script type="text/javascript">` atau `<script>` saja.

Contoh kode JavaScript untuk menampilkan message box adalah:

```
<script type="text/javascript">
  alert("Halo Dunia!");
</script>
```

Kode JavaScript juga bisa diletakkan di file tersendiri yang berekstensi .js (singkatan dari JavaScript). Untuk memanggil kode JavaScript yang terdapat di file sendiri, di bagian awal `<head>` harus ditentukan dahulu nama file .js yang dimaksud dengan menggunakan contoh kode seperti berikut:

```
<script type="text/javascript"
src="alamat.js">
</script>
```

2.2 Apa Yang Baru di HTML 5

HTML5 adalah versi terbaru dari HTML dan ini akan menjadi standar baru untuk HTML, XHTML, dan HTML DOM. Versi terakhir dari HTML ada di tahun 1999.

Dan web sudah berkembang dengan pesat hingga jaman sekarang. HTML 5 sendiri sampai saat ini masih terus dikembangkan, tapi mayoritas browser modern sudah bisa mendukung HTML 5.

Ada beberapa aturan yang diterapkan untuk HTML 5, seperti:

- Fitur-fitur baru harus berbasis HTML, CSS, DOM, dan JavaScript.
- Mengurangi kebutuhan plugin eksternal (contohnya Flash).
- Error handling yang lebih baik.
- Markup tambahan untuk menggantikan scripting.
- HTML5 harus bisa diakses dari peranti manapun/device independent.
- Proses pengembangan harus bisa dilihat oleh publik.

HTML 5 punya beberapa spesifikasi baru yang membedakan dengan HTML 4. Dan semua spesifikasi ini dibangun atas dasar kemudahan, jadi semuanya didesain serba praktis. Contohnya pada penulisan kode berikut, semuanya boleh:

```
id="belajarhtml5"
id= belajarhtml5
ID= " belajarhtml5"
```

Jadi, sintaks untuk HTML 5 sangat fleksibel, namun bukan berarti Anda bisa menulis dengan sembarangan. HTML5 mendukung pemisahan yang tegas antara tahapan presentasi dan konten.

HTML5 memungkinkan ini menggunakan CSS. Bahkan beberapa fitur tampilan di versi HTML sebelumnya tidak lagi didukung (namun masih bisa bekerja).

Jika antara konten dan style tidak dipisahkan, maka akan banyak masalah. Antara lain adalah:

- Aksesibilitas jelek.
- Akan lebih kompleks, kode lebih sulit untuk dibaca dibandingkan jika konten dan style dipisahkan.
- Ukuran dokumen akan lebih besar, karena konten style akan diulang berulang kali. Sehingga proses loading dokumen akan menjadi lebih lama.

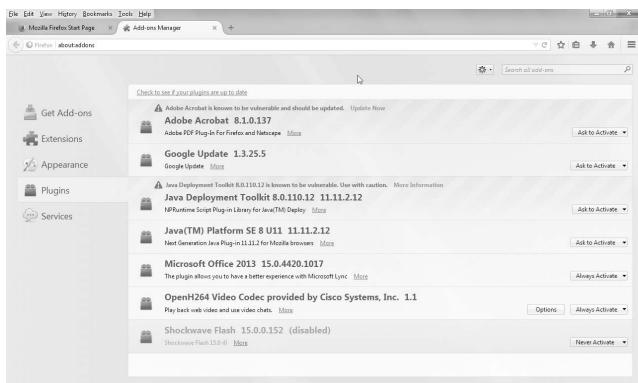
Beberapa fitur baru di HTML5 antara lain:

- Elemen canvas untuk menggambar.
- Elemen audio, dan video untuk memutar audio dan video.
- Dukungan untuk penyimpanan offline lokal.
- Elemen yang spesifik terhadap konten, seperti article, footer, header, nav, dan section.
- Kontrol form baru, seperti calendar, date, time, email, url, search.

2.2.1 Paradigma Bebas Plugin

HTML5 berusaha untuk memberikan dukungan standar untuk beberapa fitur yang sebelumnya tidak mungkin bisa digunakan tanpa adanya plugin, misalnya ada beberapa API untuk drawing, soket, dan lain sebagainya. Ini karena plugin bisa memiliki beberapa masalah, seperti:

- Tidak semua browser mendukung plugin.
- Plugin bisa di-disable dan diblok oleh komputer tertentu, misalnya iPad buatan Apple sampai saat ini tidak mendukung plugin Flash.
- Plugin bisa menjadi sasaran hacker.
- Plugin sulit diintegrasikan dengan dokumen lainnya, karena ada masalah seputar clipboard, dan transparansi.



Gambar 2.13 Daftar plugin dari Firefox, yang se bisa mungkin hendak dikikis fungsinya dengan HTML 5

Untuk mewujudkan paradigma ini, berikut fitur-fitur penting dari HTML 5:

- Canvas (2D, dan 3D)
- Channel messaging
- Cross-document messaging
- Geolocation
- MathML
- Microdata
- Event Server-Sent
- Scalable Vector Graphics (SVG)
- API WebSocket dan protocol
- Konsep Web origin
- Web storage
- Web SQL database
- Web Worker
- XMLHttpRequest Level 2

2.2.2 Doctype dan Character Set Baru

Termasuk di antara fasilitas baru dari HTML 5 adalah lebih simpel, yang perlu disederhanakan pertama kali adalah Doctype dan character set-nya. Lihat doctype standar dari HTML 4 seperti berikut:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
 "http://www.w3.org/TR/html4/loose.dtd">
```

Tentu sulit untuk menghafal karakter-karakter di atas, kebanyakan programmer akan langsung meng-copy paste kode di atas ke dalam dokumen. Karena itu ada sintaks doctype baru di HTML 5 ini, yaitu:

```
<!DOCTYPE html>
```

Kode doctype di atas akan sangat mudah dihafal. Selain itu character set juga diganti agar lebih mudah dihafal, berikut ini contoh character set yang lama:

```
<meta http-equiv="Content-Type" content="text/html;
 charset=utf-8">
```

Di HTML 5, akan diganti menjadi seperti berikut ini:

```
<meta charset="utf-8">
```

2.2.3 Elemen-elemen Baru dan Lama

HTML 5 memiliki beberapa elemen markup baru, yang bisa digolongkan menjadi 7 jenis tipe konten. Berikut ini beberapa deskripsi dari tipe konten di HTML 5.

| Tipe | Deskripsi |
|-------------|--|
| Embedded | Konten yang mengimpor sumber daya lain ke dokumen, contohnya audio, video, canvas, dan iframe. |
| Flow | Digunakan di bagian body dari dokumen, dan aplikasi, contohnya form, h1, dan small. |
| Heading | Bagian section, misalnya h1, h2, dan hgroup. |
| Interactive | Konten yang memungkinkan user berinteraksi, misalnya kontrol audio video, button, dan text area. |

| Tipe | Deskripsi |
|------------|---|
| Metadata | Elemen yang biasanya ditemukan di head yang menentukan perilaku atau tampilan dari semua dokumen. Contohnya script, style, dan title. |
| Phrasing | Teks, dan elemen markup, contohnya mark, kbd, sub, dan sup. |
| Sectioning | Elemen yang mendefinisikan section di dokumen, misalnya article, aside, dan title. |

Semua elemen ini bisa diberi style dengan CSS. Bahkan, beberapa di antaranya seperti canvas, audio, dan video, bisa digunakan oleh elemen itu sendiri. Beberapa elemen seperti elemen canvas, audio, dan video merupakan hal yang baru di HTML5.

Tapi ada juga elemen yang deprecated yang tidak lagi dipakai di HTML 5, kebanyakan adalah elemen yang melakukan styling secara inline di CSS, seperti: big, center, font, dan basefont.

2.2.4 Markup Semantik

Salah satu tipe konten yang baru di HTML 5 adalah tipe **Sectioning**. HTML5 mendefinisikan beberapa markup semantik baru untuk mendeskripsikan konten dari elemen. Menggunakan markup semantik memang tidak berpengaruh terhadap tampilan Anda secara langsung, tapi akan mempermudah proses desain dari halaman HTML., dan di masa depan akan lebih memudahkan pembuatan SEO friendly karena search engine akan memanfaatkan semantik ini saat crawling dan mengindeks halaman.

Jadi, HTML 5 ini adalah sesuatu yang baru yang akan menentukan bagaimana sejarah HTML di masa depan. Berikut ini beberapa sectioning untuk elemen HTML.

| Sectioning | Deskripsi |
|------------|--|
| header | Konten header (untuk halaman atau section dari halaman). |
| footer | Konten footer (untuk halaman atau section dari |

| | |
|---------|-----------------------------------|
| | halaman). |
| section | Sebuah section di halaman web. |
| article | Sebuah konten artikel independen. |
| aside | Konten yang berkaitan atau quote. |
| nav | Bantuan navigasi. |

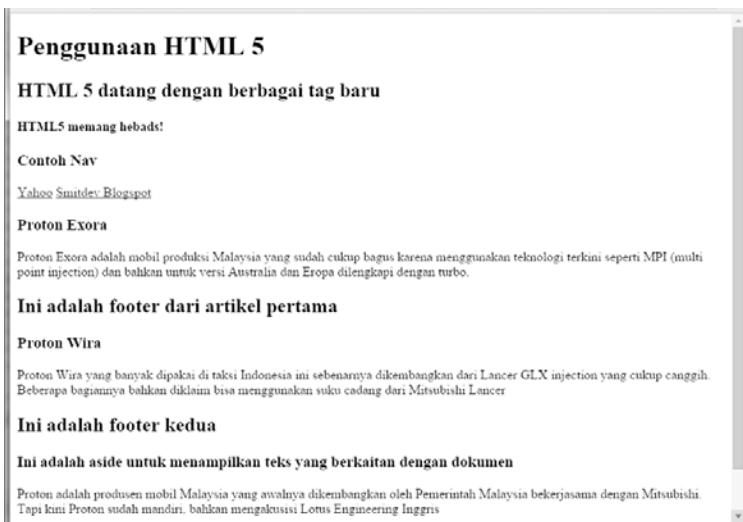
Semua elemen ini bisa diatur style-nya menggunakan CSS. Nantinya ini akan memudahkan pengaturan dokumen.

Berikut ini contoh penggunaan teks HTML.

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" >
    <title>Contoh konten HTML5</title>
  <!-- asumsinya nama file CSS adalah html5.css -->
  <link rel="stylesheet" href="html5.css">
  </head>
  <body>
    <header>
      <h1>Penggunaan HTML 5</h1>
      <h2>HTML 5 datang dengan berbagai tag baru </h2>
      <h4>HTML5 memang hebads!</h4>
    </header>
    <div id="container">
      <nav>
        <h3>Contoh Nav</h3>
        <a href="http://www.yahoo.com">Yahoo</a>
        <a href="http://www.smitdev.com">Smitdev </a>
        <a href="http://www.blogspot.com">Blogspot</a>
      </nav>
      <section>
        <article>
          <header>
            <h1>Proton Exora</h1>
          </header>
          <p>Proton Exora adalah mobil produksi Malaysia yang sudah cukup bagus karena menggunakan teknologi terkini seperti MPI (multi point injection), dan bahkan untuk versi Australia, dan Eropa dilengkapi dengan turbo.</p>
          <footer>
            <h2>Ini adalah footer dari artikel pertama</h2>
          </footer>
        </article>
        <article>
          <header>
            <h1>Proton Wira</h1>
          </header>
          <p>Proton Wira yang banyak dipakai di taksi Indonesia ini sebenarnya dikembangkan dari Lancer GLX injection yang cukup canggih. Beberapa bagiannya bahkan diklaim bisa menggunakan suku cadang dari Mitsubishi Lancer </p>
```

```
<footer>
<h2>Ini adalah footer kedua</h2>
</footer>
</article>
</section>
<aside>
<h3>Ini adalah aside untuk menampilkan teks yang berkaitan
dengan dokumen</h3>
<p>Proton adalah produsen mobil Malaysia yang awalnya
dikembangkan oleh Pemerintah Malaysia bekerjasama dengan
Mitsubishi. Tapi kini Proton sudah mandiri, bahkan mengakuisisi
Lotus Engineering Inggris</p>
</aside>
<footer>
<h2>Ini footer</h2>
</footer>
</div>
</body>
</html>
```

Jika dijalankan di browser, kode di atas menjadi seperti berikut.



Gambar 2.14 Kode HTML 5 yang pertama

Jika Anda lihat, tampilannya masih standar karena memang ini adalah konten, style-nya nanti disediakan menggunakan CSS sendiri. Anda bisa membuat style sheet CSS secara manual atau menggunakan alat bantu seperti Dreamweaver untuk mempermudah pembuatan CSS.

2.2.5 Tag-tag Baru

Ada beberapa tag baru di HTML 5 ini, banyak malahan. Berikut ini beberapa di antaranya yang perlu Anda pahami.

| Tag | Deskripsi |
|--------------|--|
| <article> | Untuk konten eksternal, seperti teks dari artikel, blog, forum atau sumber eksternal lainnya. |
| <aside> | Untuk konten sampingan, konten ini harus berkaitan dengan konten di sekitarnya. |
| <command> | Sebuah button, radiobutton, atau checkbox. |
| <details> | Untuk menjelaskan detail dari dokumen, atau bagian dari dokumen. |
| <summary> | Sebuah caption, atau summary, di dalam elemen detail. |
| <figure> | Untuk mengelompokkan konten stand alone, misalnya video. |
| <figcaption> | Caption dari section figure. |
| <footer> | Untuk bagian footer dari dokumen atau section, termasuk di dalamnya nama pengarang, tanggal dokumen, informasi kontak, atau informasi copyright. |
| <header> | Untuk pengenalan dokumen atau section, bisa termasuk navigasi di dalamnya. |
| <hgroup> | Untuk section dari heading, menggunakan <h1> sampai <h6>, di mana bagian heading terbesar adalah heading utama dari section, dan lainnya adalah sub heading. |
| <mark> | Untuk teks yang harus di-highlight. |

| Tag | Deskripsi |
|------------|--|
| <meter> | Untuk pengukuran, hanya digunakan jika nilai maksimum, dan minimum diketahui. |
| <nav> | Untuk bagian dari navigasi. |
| <progress> | Untuk menampilkan proses. |
| <ruby> | Untuk anotasi ruby (karakter atau notasi dalam bahasa Cina). |
| <rt> | Untuk penjelasan dari anotasi ruby. |
| <rp> | Apa yang harus ditampilkan oleh browser yang tidak mendukung elemen ruby. |
| <section> | Untuk bagian dokumen, seperti chapter, header, footer, atau section lain dari dokumen. |
| <time> | Untuk mendefinisikan waktu, tanggal atau keduanya. |
| <wbr> | Word break. Untuk mendefinisikan kemungkinan line break. |

Untuk elemen media baru, ada beberapa tag yang diperkenalkan di HTML 5, antara lain:

| Tag | Deskripsi |
|----------|---|
| <audio> | Untuk konten multimedia, suara, audio stream atau lainnya. |
| <video> | Untuk konten video, seperti klip video, atau stream video lainnya. |
| <source> | Untuk resource media dari elemen media. Didefinisikan di dalam elemen video atau audio. |
| <embed> | Untuk konten embedded, seperti plug in. |

Untuk elemen canvas, cuma ada satu tag, yaitu:

| Tag | Deskripsi |
|----------|-------------------------------------|
| <canvas> | Untuk membuat grafik dengan script. |

Untuk elemen form, ada beberapa seperti:

| Tag | Deskripsi |
|------------|---|
| <datalist> | Daftar opsi untuk nilai input. |
| <keygen> | Generate key untuk user yang sudah terotentifikasi. |
| <output> | Untuk beberapa tipe output seperti output yang ditulis oleh script. |

Untuk tipe atribut input lainnya, seperti:

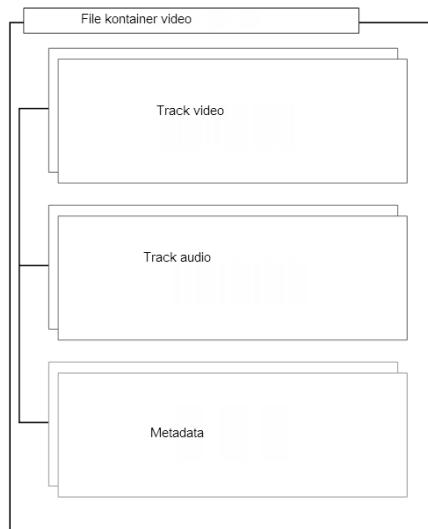
| Tipe | Deskripsi |
|----------|---|
| tel | Tipe nilai input untuk nomor telepon. |
| search | Tipe input untuk field pencarian. |
| url | Tipe input untuk URL. |
| email | Tipe input dari satu alamat email atau lebih. |
| datetime | Tipe input untuk tanggal, dan/atau waktu. |
| date | Tipe input dari tanggal. |
| month | Tipe input dari bulan. |
| week | Tipe input dari minggu. |
| time | Tipe input dari waktu. |

| | |
|----------------|--|
| datetime-local | Tipe input dari waktu/tanggal lokal. |
| number | Tipe input dari angka. |
| range | Tipe input dari range tertentu. |
| color | Tipe input dari warna heksadesimal, seperti #FF8800. |

2.3 Bekerja dengan Audio dan Video

Sebuah file audio atau video sebenarnya adalah file kontainer, artinya ada file video dan audio di dalamnya, mirip dengan file arsip .zip atau .rar yang di dalamnya bisa ada beberapa file.

Contohnya sebuah file video, akan mengandung track video, track audio, dan metadata tambahan. Track audio, dan video ini dikombinasikan saat run time untuk memainkan video plus suaranya. Adapun metadata menampilkan informasi seputar video, seperti cover art, title dan subtitle, captioning, dan lain sebagainya.



Gambar 2.15 Contoh isi kontainer video

Beberapa format kontainer video yang populer antara lain:

- Audio Video Interleave (.avi)
- Flash Video (.flv)
- MPEG 4 (.mp4)
- Matroska (.mkv)
- Ogg (.ogg)

2.3.1 Codec Audio dan Video

Codec audio dan video adalah algoritma yang digunakan untuk encoding dan decoding video atau stream video tertentu sehingga bisa diputar. Media raw banyak sekali, sehingga tanpa encoding, klip video dan audio akan memiliki ukuran yang sangat besar. Ini menjadi tidak mungkin ditransfer via internet karena akan memakan waktu sangat banyak.

Tanpa decoder, penerima file tidak akan bisa menyusun kembali sumber media asli dari bentuk yang sudah ter-encoding. Codec memungkinkan memahami berbagai format kontainer, dan kemudian men-decode track audio dan video yang dikandungnya.

Beberapa jenis codec audio antara lain:

- AAC
- MPEG-3
- Ogg Vorbis

Adapun contoh codec video antara lain:

- H.264
- VP8
- Ogg Theora

Beberapa jenis codec dipatenkan adapun yang lainnya adalah bebas. Yang bebas misalnya codec audio Vorbis dan Theora. Adapun MPEG-4, dan H.264 meminta Anda membayar sejumlah uang untuk biaya lisensi.

Tidak semua browser mendukung semua codec di atas., dan kemungkinan ke depan, dukungan untuk codec akan semakin meluas, dan bisa jadi ada salah satu codec yang akan menjadi standar de facto untuk web.

2.3.2 Batasan Audio dan Video

Ada beberapa hal yang tidak didukung oleh HTML seputar spek audio dan video:

- Streaming file audio dan video: Ini karena tidak adanya standar untuk switching bitrate di video HTML 5. Hanya satu file media yang didukung oleh implementasi saat ini. Tapi kemungkinan di masa depan akan bisa didukung.
- Full screen: ini tidak bisa karena bisa melanggar alasan security. Tapi browser bisa mengijinkan user untuk melihat video full screen melalui kontrol tambahan.
- Aksesibilitas untuk audio dan video belum ditentukan secara penuh.

Tidak semua browser bisa mendukung tiap jenis video, dan audio. Berikut ini beberapa di antaranya (spek ini kemungkinan bisa berubah di masa depan):

- Chrome: mendukung Theora dan Vorbis, Ogg Container, H.264 dan AAC, MPEG 4.
- Firefox: mendukung Theora, dan Vorbis, Ogg Container.
- Internet Explorer: tidak mendukung.
- Opera: Theora and Vorbis, Ogg Container, VP8, dan Vorbis, WebM Format.
- Safari: mendukung H.264, dan AAC, MPEG 4 Container.

Untuk menampilkan video, Anda bisa menggunakan tag <video> ditutup dengan tag </video>. Di tengahnya, Anda bisa menuliskan teks atau kode yang akan ditampilkan jika ternyata browser tidak mendukung video atau audio.

```
<video src="contohvideo.ogg" controls>
Tolong update browser Anda, karena tidak mendukung Audio
video di HTML 5
</video>
```

2.3.3 Mendeklrasikan Elemen Media

Untuk memainkan file audio, dan video sama saja, hanya saja tag untuk audio adalah <audio>, dan ditutup dengan </audio>, di tag pembukanya, Anda bisa memasukkan atribut. Misalnya kode berikut:

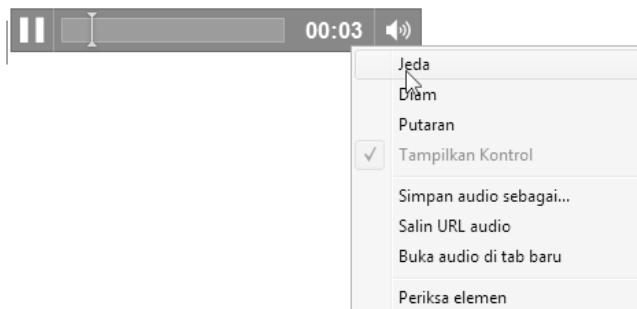
```
<html>
<head>
</head>
<body>
<audio controls src="110.mp3">
Memainkan klik audio.
</audio>
</body>
</html>
```

Asumsinya, di atas ada file audio dengan nama 110.mp3 dengan lokasi satu folder dengan file HTML tempat penyimpanan kode di atas. Jika dijalankan di browser yang mendukung, maka tampilan muncul seperti berikut (catatan: mungkin browser Anda tidak mendukung, Anda harus mengganti menggunakan browser yang mendukung).



Gambar 2.16 Tampilan HTML 5 saat memiliki audio untuk memainkannya

Jika diklik **Play**, barulah file audio akan dimainkan. Anda bisa klik kanan di player tersebut untuk mengontrol pemutaran file audio.



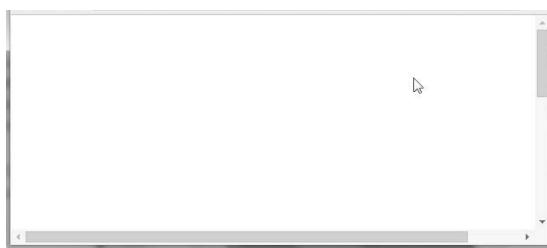
Gambar 2.17 Klik pada kontrol untuk menjalankan program

Anda juga bisa memberikan source file audio atau video lebih dari satu, nantinya browser akan melihat mana file yang paling feasibel untuk diakses.

```
<audio controls>
<source src="audioPertama.ogg">
<source src="audioKedua.mp3">
Silakan update browser Anda untuk menampilkan audio dengan
HTML 5.
</audio>
```

Ini juga membuat browser bisa memainkan lebih dari satu file. Adanya atribut controls di atas memerintahkan browser untuk menampilkan kontrol user untuk memulai (start), menghentikan (stop) dan mengarahkan ke titik tertentu di file. Selain itu ada volume control untuk mengatur volume.

Jika atribut controls ini tidak ada, maka clip akan terlihat tanpa control dan user tidak bisa memainkan file.



Gambar 2.18 Ketika atribut controls tidak ada, maka tidak bisa dimainkan, karena tidak ada atribut auto

Tulisan yang ada di antara tag <audio> akan ditampilkan di browser. Tulisan ini bisa diisikan kode yang menandakan bahwa browser tidak kompatibel dengan tag <audio> dari HTML 5.

Anda bisa juga menambahkan atribut type untuk menjelaskan tipe atribut, misalnya:

| Tipe | Nilai Atribut |
|---|---|
| Theora video, dan vorbis audio di kontainer Ogg | type='video/ogg; codecs="theora,vorbis"' |
| Vorbis audio di kontainer Ogg | type ='audio/ogg; codecs=vorbis' |

| | |
|--|---|
| Video H264, dan audio AAC yang kompleksitasnya rendah di kontainer MP4 | type='video/mp4'; codecs="mp4v.20.8, mp4a.40.2" |
|--|---|

Dari kode di atas, kontrol untuk menjalankan bisa ditampilkan dengan menggunakan kontrol attribute. Jika control tidak ingin ditampilkan, tapi diberi atribut autoplay, maka file multimedia akan langsung bisa dijalankan.

```
<html>
<head>
<title>
Memainkan file audio di HTML 5
</title>
</head>
<body>
<audio autoplay src="110.mp3">
Memainkan Klik audio.
</audio>
</body>
</html>
```

Dengan menggunakan atribut autoplay, media akan dimainkan begitu program di-load tanpa ada interaksi dari user. Namun, umumnya user akan merasa terganggu jika file dijalankan secara langsung tanpa interaksi dari user.

Anda juga bisa menjalankan secara manual menggunakan method, seperti load(), play(), dan pause () .

Beberapa atribut untuk tag <video> ataupun <audio> adalah:

| Atribut | Value | Deskripsi |
|----------|----------|---|
| autoplay | autoplay | Menentukan apakah video akan dimainkan secara langsung setelah ready. |
| controls | controls | Menentukan apakah kontrol akan ditampilkan, seperti tombol Play . |
| height | pixels | Menentukan tinggi dari player video. |
| loop | loop | Menentukan file media yang akan di-loop tiap kali sudah selesai dimainkan. |
| preload | preload | Menentukan bagaimana video akan di-load tiap kali halaman di-load, ini akan tidak dianggap jika ada autoplay. |

| Atribut | Value | Deskripsi |
|---------|--------|--|
| src | url | Menentukan url dari video yang akan dimainkan. |
| width | pixels | Menentukan lebar dari player video. |

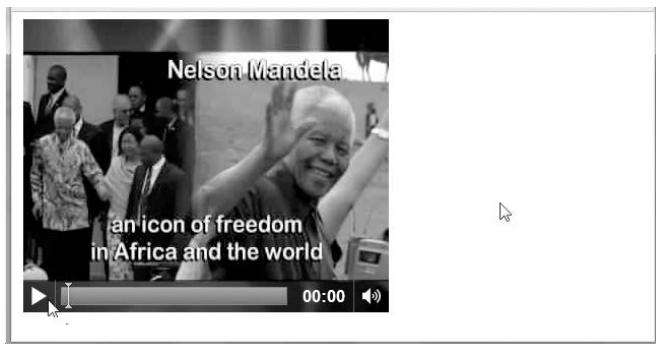
Adapun untuk video, berikut ini contoh kode untuk memainkannya:

```
<html>
<head>
<title> Memainkan Video </title>
</head>
<body>

<video src="ubuntu.ogg" controls="controls">
Browser anda tidak mendukung Video euy
</video>

</body>
</html>
```

Jika dijalankan di browser, maka muncul tampilan awal video, tapi video belum dimainkan.



Gambar 2.19 Tampilan awal video, tapi belum dimainkan

Untuk memainkan, klik tombol **Play**. Maka Anda bisa melihat video langsung dimainkan.



Gambar 2.20 Video dimainkan

Anda juga bisa mengontrol memainkan, dan menjalankan ini menggunakan kode secara otomatis. Misalnya kode berikut ini yang akan memainkan video jika pointer mouse hover (ada di atas video), dan kemudian mem-pause video ketika pointer mouse belum dilepaskan dari atas video.

```
<html>
<head>
<title>menjalankan Video Dengan Pointer Mouse </title>
</head>
<body>
Letakkan pointer mouse di atas video untuk memainkan <br>
Letakkan di luar video untuk pause <br>
<video id="movies" onmouseover="this.play()">
onmouseout="this.pause()"
autobuffer="true"
width="400px" height="300px">
<source src="ubuntu.ogg">
</video>
</body>
</html>
```

Jika dijalankan, maka akan terlihat seperti berikut.



Gambar 2.21 Jika pointer mouse tidak hover, video di-pause

Adapun jika pointer mouse diletakkan di atasnya, maka video akan langsung dimainkan. Video ini nanti bisa diintegrasikan di canvas kalau perlu.



Gambar 2.22 Video dimainkan ketika pointer mouse ada di atasnya

2.4 Menggunakan Web Storage

HTML5 menawarkan beberapa metode untuk menyimpan data di client, yaitu:

- `localStorage`: menyimpan data tanpa batas waktu.
- `sessionStorage`: menyimpan data untuk satu session.

Sebelumnya, ini dilakukan dengan cookies, tapi cookies tidak cocok dipakai jika ukuran data terlalu besar karena cookies akan dikirim tiap kali ada request ke server. Ini membuat transfer data menjadi pelan dan tidak efektif.

Di HTML 5, data tidak dikirim tiap kali ada request server. Tapi digunakan ketika diminta saja. Jadi dengan HTML 5, user bisa menyimpan banyak data tanpa mempengaruhi performa website.

Data ini disimpan di berbagai bagian dari website, dan Anda bisa memakai Javascript untuk menyimpan dan mengaksesnya. Data ini kemudian disimpan di berbagai area untuk berbagai website, sebuah website hanya bisa mengakses data yang disimpan oleh HTML 5 untuk menyimpan dan mengakses data.

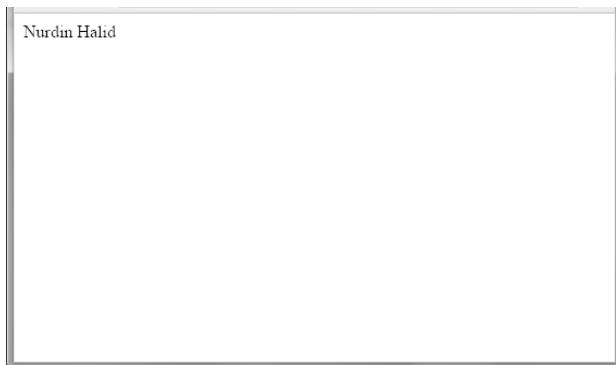
2.4.1 Local Storage

Method Local Storage ini digunakan untuk menyimpan data tanpa batasan waktu. Data akan tetap tersimpan kapanpun, bahkan sampai tahunan.

Berikut ini contoh kode untuk menyimpan data di localstorage:

```
<script>
localStorage.dataNama="Nurdin Halid";
document.write(localStorage.dataNama);
</script>
```

Berikut ini tampilannya di browser.



Gambar 2.23 Tampilan penyimpanan data di Local storage

Adapun contoh kode berikut fungsinya untuk mengetahui jumlah kunjungan user yang mengunjungi halaman tertentu tidak terbatas session, Anda bisa menambahkannya ke dalam kode sebelumnya.

```
<script>
//menghitung jumlah kunjungan
if (localStorage.jmlKunjungan)
{
    localStorage.jmlKunjungan=Number(localStorage.jmlKunjungan)
+1;
}
else
{
    localStorage.jmlKunjungan=1;
}
document.write("Jumlah kunjungan " + localStorage.jmlKunjungan
+ " kali.");
</script>
```

Tampilan halaman akan menjadi berubah.



Gambar 2.24 Kode yang menjelaskan jumlah kunjungan

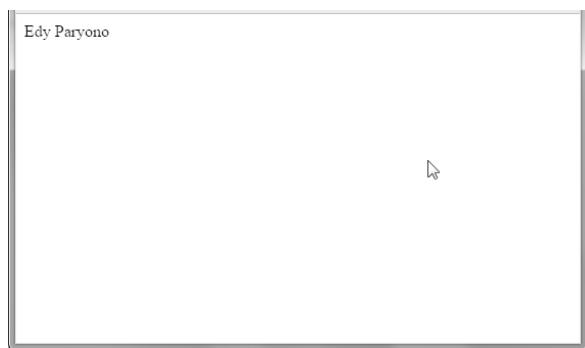
2.4.2 Session Storage

Session storage digunakan untuk menyimpan data dalam satu sesi. Data ini akan dihapus jika user menutup jendela browser.

Berikut ini contoh kode untuk menyimpan data dalam satu sesi:

```
<script>
  sessionStorage.simpananData="Edy Paryono";
  document.write(sessionStorage.simpananData);
</script>
```

Jika ditampilkan di browser, tampilan seperti berikut.

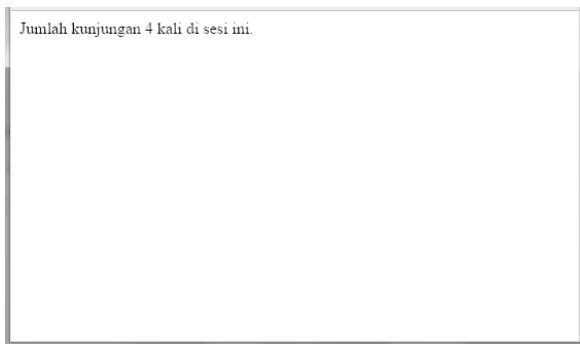


Gambar 2.25 Penyimpanan data, dan pengambilan dengan Session storage

Adapun kode berikut akan mencantumkan jumlah kunjungan seorang user dalam satu session:

```
<script>
if (sessionStorage.jumlahKunjunganPerSession)
{
    sessionStorage.jumlahKunjunganPerSession=Number(sessionStorage
    .jumlahKunjunganPerSession) +1
}
else
{
    sessionStorage.jumlahKunjunganPerSession=1;
}
document.write("Jumlah kunjungan
"+sessionStorage.jumlahKunjunganPerSession+" kali di sesi
ini.");
</script>
```

Jika dilihat di browser, tampilan akan seperti berikut.



Gambar 2.26 Hasil pemasukan dengan session storage

2.5 Form untuk Input Type

HTML5 punya beberapa tipe input untuk form. Fitur-fitur baru ini memungkinkan kontrol input dan validasi yang lebih baik. Di antara input form baru di HTML 5 adalah:

- email
- url
- number
- range

- date pickers (date, month, week, time, datetime, datetime-local)
- search
- color

Form merupakan faktor utama yang menyebabkan web bisa sangat produktif. Tanpa kontrol form, maka web tidak bisa digunakan untuk membuat aplikasi yang canggih, seperti untuk transaksi bisnis, social network, dan sebagainya.

Form di HTML 5 sendiri masih terus berkembang, tapi aturannya kurang lebih seperti berikut ini:

- Form masih harus dibuka di dalam tag <form>, dan </form>.
- Form masih mengirim nilai control ke server ketika user atau programmer men-submit halaman.
- Semua kontrol konvensional seperti text, radio button, check box, dan lain sebagainya masih tetap ada, dan tetap seperti sebelumnya.
- Kontrol form ini masih bisa dikoding jika ingin membuat handler dan modifier sendiri.

2.5.1 Email

Tipe input type email ini digunakan untuk memasukkan teks yang berupa email. Nilai dari email ini akan otomatis divalidasi ketika form di-submit.

Untuk memasukkan input type email ini, caranya adalah:

```
<input type="email" name="email_pengunjung" />
```

2.5.2 URL

Tipe URL ini digunakan untuk field input yang mengandung alamat URL dari website. Nilai dari field url ini akan divalidasi ketika form di-submit.

```
<input type="url" name="url_pengunjung" />
```

2.5.3 Number

Tipe number digunakan sebagai input field yang mengandung angka. Nilai dari field number ini akan divalidasi ketika form di-submit. Anda juga dapat mengatur batasan dari angka yang diperbolehkan.

```
<input type="number" name="angka" min="1" max="10" />
```

Ada beberapa atribut untuk tipe number ini, antara lain:

| Atribut | Nilai | Deskripsi |
|---------|-------|---|
| max | Angka | Menentukan nilai maksimal yang diperbolehkan. |
| min | Angka | Menentukan nilai minimum yang diperbolehkan. |
| step | Angka | Menentukan interval legal yang dibolehkan. |
| value | Angka | Menentukan nilai default. |

2.5.4 Range

Tipe range digunakan untuk field input yang harus memiliki nilai dari jangkauan angka. Di browser, range ini akan ditampilkan sebagai slider. Anda juga bisa membatasi angka apa saja yang dapat dipilih.

```
<input type="range" name="jangkauan" min="1" max="10" />
```

2.5.5 Date Picker

Ini gunanya untuk mengambil data atau waktu, ada beberapa atribut, seperti:

- Date: memilih tanggal, bulan, dan tahun.
- Month: memilih bulan, dan tahun.
- Week: memilih minggu, dan tahun.

- Time: memilih waktu (jam, dan menit).
- Datetime: memilih waktu, tanggal, bulan, dan tahun (waktu UTC).
- datetime-local: memilih waktu, tanggal, bulan, dan tahun (waktu local).

Contoh kodennya adalah:

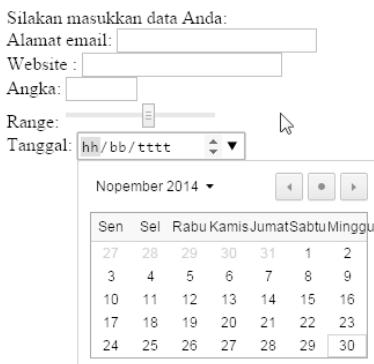
```
Date: <input type="date" name="tanggal" />
```

Berikut ini contoh beberapa kode untuk input type:

```
<html>
<head>
<title>Penggunaan Input Type</title>
</head>
<body>
<form>
Silakan masukkan data Anda: <br>
Alamat email: <input type="email" name="email_pengunjung" />
<br>
Website: <input type="url" name="url_pengunjung" /> <br>
Angka: <input type="number" name="angka" min="1" max="10" />
<br>
Range: <input type="range" name="jangkauan" min="1" max="10" /> <br>
Tanggal: <input type="date" name="tanggal" />

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

Jika dilihat di browser (contohnya Opera), tampilan tiap field akan sesuai dengan jenisnya.



Gambar 2.27 Berbagai jenis input type dibuka di browser

BAB 3

CSS 3

CSS3 atau CSS versi 3 merupakan standar terakhir dari CSS. CSS3 memiliki kompatibilitas ke belakang dengan versi CSS sebelumnya. Subbab ini akan menjelaskan beberapa fitur baru di CSS 3 yang mendukung desain web responsif.

3.1 Apa Itu CSS?

CSS merupakan singkatan dari *Cascading Style Sheets*. CSS berfungsi mendefinisikan bagaimana elemen HTML ditampilkan. Style sendiri mulai diperkenalkan sejak versi HTML 4.0 untuk menentukan style dokumen.

Style sheet eksternal bisa menghemat banyak waktu. Style sheet eksternal diletakkan di file tersendiri yang memiliki ekstensi .css. Kenapa CSS sangat penting? Ini karena pada awalnya HTML tidak pernah diniatkan untuk berisi tag untuk pemformatan dokumen. HTML hanya diniatkan untuk mendefinisikan konten dokumen saja, seperti:

```
<h1>Ini heading</h1>
<p>Ini paragraph.</p>
```

Ketika tag seperti , dan atribut warna mulai diperkenalkan sejak versi HTML 3.2, banyak developer yang kerja sangat keras karena mengubah font dan informasi global untuk situs ukuran besar menjadi pekerjaan yang panjang dan mahal.



Gambar 3.1 Logo CSS

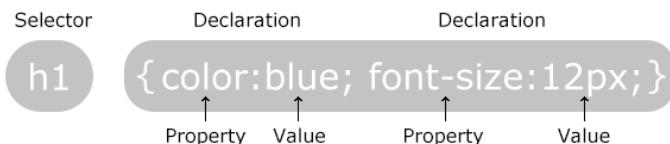
Untuk memecahkan masalah ini, World Wide Web Consortium (W3C) membuat CSS. Di HTML 4.0, semua pemformatan bisa dihilangkan dari dokumen HTML dan disimpan di file CSS terpisah. Semua browser di zaman sekarang mendukung CSS.

Dengan CSS, pekerjaan pengaturan style banyak dokumen bisa sangat mudah karena tinggal memilih menggunakan selektor, kemudian memformat style-nya. CSS mendefinisikan bagaimana elemen HTML ditampilkan.

Style CSS umumnya disimpan di file .css eksternal. Style sheet eksternal memungkinkan Anda mengubah tampilan dan layout semua halaman web di situs hanya dengan satu file saja.

3.2 Sintaks CSS

Untuk menuliskan CSS, Anda perlu memahami sintaks standar CSS, ada 2 bagian utama, yaitu blok selektor dan blok deklarasi. Anda bisa melihatnya di contoh gambar berikut.



Gambar 3.2 Sintaks CSS

Selektor mengacu kepada elemen HTML yang ingin diberi style, sementara blok deklarasi berisi satu atau lebih deklarasi style, dipisahkan dengan tanda titik koma.

Setiap deklarasi terdiri dari dua nilai, yaitu nama properti dan nilainya dipisahkan dengan tanda “:”.

Sebuah deklarasi CSS selalu diawali dengan tanda kurung kurawal buka dan ditutup dengan tanda kurung kurawal tutup.

Contohnya seperti berikut ini:

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

Untuk membuat kode CSS lebih mudah dibaca, Anda bisa membuat satu deklarasi di tiap baris, contohnya seperti berikut ini:

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

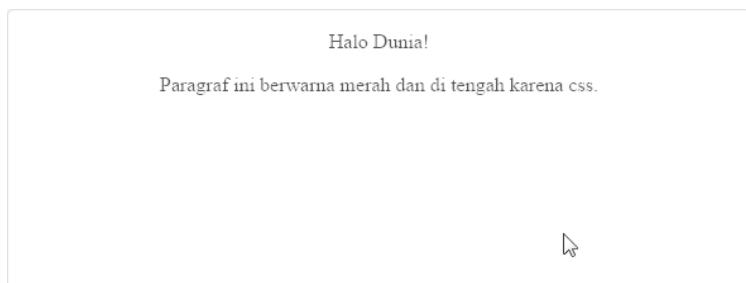
<style>
p {
  color: red;
  text-align: center;
}
</style>

</head>
<body>

<p>Halo Dunia!</p>
<p>Paragraf ini berwarna merah dan di tengah karena css.</p>

</body>
</html>
```

Apabila dilihat di browser, terlihat teks menjadi memiliki style karena CSS.



Gambar 3.3 Teks memiliki style karena CSS

3.2.1 Komentar CSS

Komentar digunakan untuk menjelaskan bagian dari kode CSS agar lebih mudah dipahami. Teks-teks yang ada di komentar tidak akan diproses oleh browser.

Komentar di CSS dimulai dengan /* dan diakhiri dengan */. Komentar bisa terdiri dari satu baris atau multi baris.

Contoh kodennya seperti berikut:

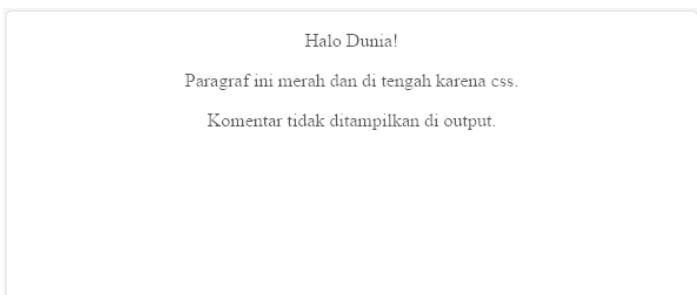
```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    color: red;
    /* ini komentar satu baris */
    text-align: center;
}

/* ini
komentar
multi-line
jadi tidak akan diperhatikan */
</style>
</head>
<body>

<p>Halo Dunia!</p>
<p>Paragraf ini merah dan di tengah karena css.</p>
<p>Komentar tidak ditampilkan di output.</p>

</body>
</html>
```

Apabila dilihat, kode CSS yang diberi komentar tidak akan diproses dan tidak mempengaruhi style apa pun.



Gambar 3.4 Kode komentar tidak diproses

3.2.2 Selektor CSS

Selektor CSS memungkinkan Anda memilih dan memanipulasi elemen-elemen HTML. Selektor CSS digunakan untuk mencari elemen HTML berdasarkan id, class, tipe, atribut, nilai atribut, dan lain sebagainya.

Misalnya, selektor elemen memilih elemen berdasarkan nama elemen. Contoh, Anda bisa memilih semua elemen `<p>` dan mengubah warnanya dengan text warna merah.

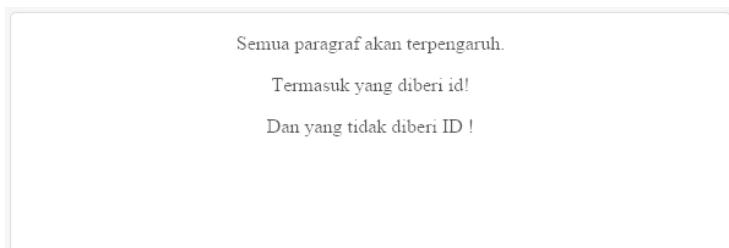
Contoh kodennya seperti ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
p {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Semua paragraf akan terpengaruh.</p>
<p id="paral">Termasuk yang diberi id!</p>
<p>Dan yang tidak diberi ID !</p>

</body>
</html>
```

Apabila ditampilkan, terlihat tampilan seperti berikut.



Gambar 3.5 Semua tag `<p>` baik id atau yang tidak diberi id akan diterapkan semua

Tipe selektor kedua adalah selektor **id**. Ini memilih berdasarkan id untuk memilih komponen tertentu dari HTML. Sebuah id biasanya unik di halaman, sehingga selektor id hanya digunakan untuk memilih elemen tunggal yang unik.

Untuk mencari elemen dengan id spesifik, Anda bisa menggunakan karakter hash diikuti dengan id elemen. Misalnya, kode berikut akan memilih elemen dengan id “para1” saja.

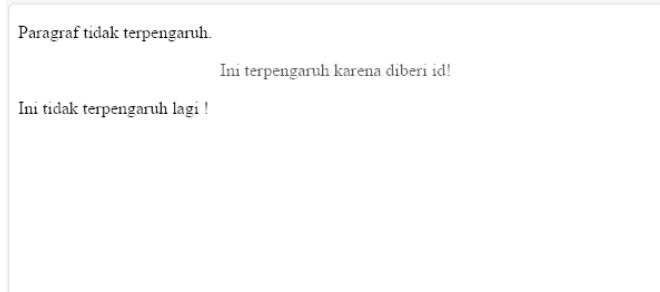
```
<!DOCTYPE html>
<html>
<head>
<style>

#para1 {
    text-align: center;
    color: red;
}
</style>
</head>
<body>

<p>Paragraf tidak terpengaruh.</p>
<p id="para1">Ini terpengaruh karena diberi id!</p>
<p>Ini tidak terpengaruh lagi !</p>

</body>
</html>
```

Apabila ditampilkan, terlihat hanya yang ber-id yang dicari saja yang terpengaruh.



Gambar 3.6 Tampilan yang id-nya ditentukan yang terpengaruh

Selektor berikutnya adalah selektor **class** yang berguna mencari elemen dengan class tertentu. Selektor class menggunakan atribut **class** dari HTML. Untuk mencari elemen di class tertentu, Anda bisa menuliskan karakter titik, diikuti dengan nama class tersebut.

Di contoh berikut, semua elemen HTML dengan class = ="center" akan diset alignment ke tengah:

```
<!DOCTYPE html>
<html>
<head>
<style>
```

```
.center {  
    text-align: center;  
    color: red;  
}  
</style>  
</head>  
<body>  
  
<h1 class="center">Heading merah di tengah</h1>  
<p class="center">Paragraf merah dan di tengah.</p>  
  
</body>  
</html>
```

Apabila dilihat, class yang diberi “center” akan di tengah, walaupun berupa heading ataupun paragraf.



*Gambar 3.7 Heading dan paragraf yang memiliki
class = “center” akan ada di tengah*

3.2.3 Memasukkan CSS

Ada beberapa cara memasukkan CSS ke dalam file HTML, yaitu:

- Style sheet eksternal
- Style sheet internal
- Inline

Untuk style sheet eksternal, ini idealnya dipakai ketika style diaplikasikan ke banyak halaman sekaligus. Dengan style sheet eksternal, Anda bisa mengubah tampilan keseluruhan website hanya dengan mengubah satu file saja.

Tiap webpage harus memasukkan link ke style sheet dengan tag <link>, tag <link> ini ditulis di bagian <head>.

Contohnya seperti ini:

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

Sebuah style sheet eksternal bisa ditulis atau dibuat dengan menggunakan editor teks apa pun, file ini tidak boleh memiliki tag HTML. File style sheet ini harus disimpan dengan ekstensi .css.

Contoh isi di dalam file .css seperti ini:

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

3.3 Modul CSS 3

CSS3 bisa dibagi ke beberapa modul. CSS3 terdiri dari spek CSS lama yang sudah pecah ke beberapa pecahan yang lebih kecil. Ada juga beberapa modul tambahan.

Beberapa modul CSS3 yang penting antara lain:

- Selector
- Box Model
- Background dan Border
- Image Value dan Replaced Content
- Text Effect
- 2D/3D Transformation
- Animation
- Layout Multi Kolom
- User Interface

3.4 Border

Dengan CSS3, Anda bisa membuat beragam jenis border, seperti menambahkan border lengkung, menambahkan shadow dan menggunakan gambar sebagai border. Anda bisa melakukan ini tanpa harus menggunakan software desain seperti Photoshop.

Di bagian ini, Anda akan belajar bagaimana menggunakan:

- border-radius
- box-shadow
- border-image

Di CSS3, membuat border sangat mudah, Anda tinggal menggunakan prroperti border-radius. Ini jauh lebih mudah dibandingkan membuat border lengkung di CSS 2m, di mana Anda harus menambahkan berbagai gambar untuk tiap ujung.

Berikut ini contoh kode di CSS3 yang menggunakan properti border-radius untuk membuat border bertipe lengkung:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 2px solid #ala2a2;
    padding: 10px 40px;
    background: #dddddd;
    width: 300px;
    border-radius: 25px;
}
</style>
</head>
<body>

<div>Teks ini akan diberi border. Semua teks di dalam div akan diberi border </div>

</body>
</html>
```

Apabila ditampilkan, terlihat teks `<div>` dikelilingi oleh border seperti berikut ini:

Teks ini akan diberi border. Semua teks di dalam div akan diberi border

Gambar 3.8 Teks `<div>` menjadi dikelilingi border

Anda juga bisa menggunakan box-shadow untuk membuat background kotak. Di CSS3, properti box-shadow digunakan untuk menambahkan bayangan ke kotak.

Berikut ini contoh bagaimana menambahkan bayangan di kotak:

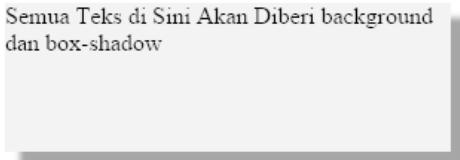
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 300px;
    height: 100px;
    background-color: yellow;
    box-shadow: 10px 10px 5px #888888;
}
</style>
</head>
<body>

<div>Semua Teks di Sini Akan Diberi background dan box-
shadow</div>

</body>
</html>
```

Apabila ditampilkan, terlihat di <div> ada kotak dan shadow/bayangan box.

Semua Teks di Sini Akan Diberi background
dan box-shadow



Gambar 3.9 <div> dan box shadow

Berikut ini contoh gambar yang akan dijadikan border, di mana salah satunya akan dipotong-potong.



Gambar 3.10 Contoh gambar yang akan dijadikan border

Untuk membuat gambar border, Anda bisa menggunakan properti border-image. Berikut ini contohnya:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 15px solid transparent;
    width: 250px;
    padding: 10px 20px;
}

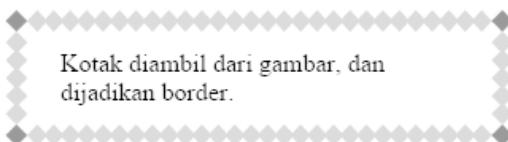
#round {
    -webkit-border-image: url(border.png) 30 30 round; /* Safari
3.1-5 */
    -o-border-image: url(border.png) 30 30 round; /* Opera 11-
12.1 */
    border-image: url(border.png) 30 30 round;
}

</style>
</head>
<body>

<div id="round">Kotak diambil dari gambar, dan dijadikan
border.</div>
<br>

</body>
</html>
```

Apabila ditampilkan seperti berikut kelihatannya.



Gambar 3.11 Tampilan border dengan CSS 3

3.5 Background

CSS3 memiliki beberapa properti background yang memungkinkan Anda mengontrol beberapa elemen background. Ada 2 properti background yang bisa dipakai untuk mengedit gambar:

- background-size
- background-origin

Properti background-size menentukan ukuran dari gambar background. Sebelum CSS 3, ukuran background hanya bisa sesuai dengan ukuran asli gambar. Di CSS 3, Anda bisa menentukan ukuran dari background, sehingga gambar background bisa dipakai berulang kali di konteks yang berbeda.

Anda bisa menentukan ukuran di piksel atau persentase. Kalau memakai persentase, maka ukurannya relatif terhadap width dan height dari elemen induk.

Contoh kodennya seperti berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
body {
    background:
url('https://www.ibm.com/developerworks/mydeveloperworks/blogs/
bobleah/resource/sky.gif');
    background-size: 100% 100%;
    background-repeat: no-repeat;
}
</style>
</head>
<body>

<p>Ini adalah paragraf pertama.</p>
<p>Ini paragraf kedua.</p>

</body>
</html>
```

Apabila dilihat, maka paragraf akan memiliki background seperti berikut.



Gambar 3.12 Paragraf memiliki background

Properti background-origin menentukan peletakan dari gambar background. Sebuah gambar background bisa diletakkan di dalam content-box, padding-box atau area border-box.

Contoh kodennya seperti berikut ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 1px solid black;
    padding: 35px;
    background-image: url('smiley.gif');
    background-repeat: no-repeat;
    background-position: left;
}

#div1 {
    background-origin: border-box;
}

#div2 {
    background-origin: content-box;
}
</style>
</head>
<body>

<p>background-origin:border-box:</p>
<div id="div1">
[I have seen you at the worst and I still think you are the
best, you are still the one that I love].
</div>

<p>background-origin:content-box:</p>
<div id="div2">
[I have seen you at the worst and I still think you are the
best, you are still the one that I love]
.
</div>

</body>
</html>
```

Apabila diperlihatkan di browser, kode di atas menghasilkan tampilan seperti berikut.

background-origin:border-box:



[I have seen you at the worst and I still think you are the best, you are still the one that I love].

background-origin:content-box:



I
[I have seen you at the worst and I still think you are the best, you are still the one that I love].

Gambar 3.13 Browser menghasilkan tampilan gambar

3.6 Gradients

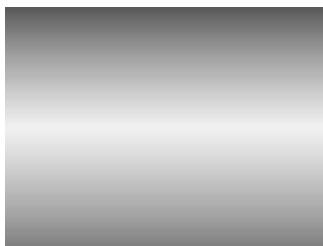
Fasilitas gradients di CSS3 memungkinkan Anda untuk membuat gradien transisi warna antara dua atau lebih warna. Sebelum CSS3, Anda harus menggunakan gambar untuk membuat gradien.

Tapi di CSS 3, Anda bisa menggunakan kode saja, sehingga menghemat bandwidth dan waktu download. Elemen gradien yang dibuat dengan CSS3 juga tampilannya lebih baik ketika di-zoom, ini karena gradient di-generate oleh browser.

CSS3 mendefinisikan dua tipe gradien:

- Gradien linear (down/up/left/right/diagonal).
- Radial Gradients (didefinisikan oleh pusatnya).

Untuk membuat gradien linear, Anda harus mendefinisikan minimal dua warna. Color stop adalah warna yang ingin di-render saat transisi. Anda bisa mengeset titik awal dan arah (atau sudut) bersamaan dengan efek gradien. Berikut ini contoh gradien linear.



Gambar 3.14 Contoh gradien linear

Untuk gradien linear, sintaksnya seperti berikut:

```
background: linear-gradient(arah, color-stop1, color-stop2,  
...);
```

Jenis gradien linear default adalah dari top ke bottom, berikut ini contoh gradien linear dari atas ke bawah, dari warna merah (red) ke biru (blue).

```
<!DOCTYPE html>  
<html>  
<head>  
<style>  
#grad1 {  
height: 200px;
```

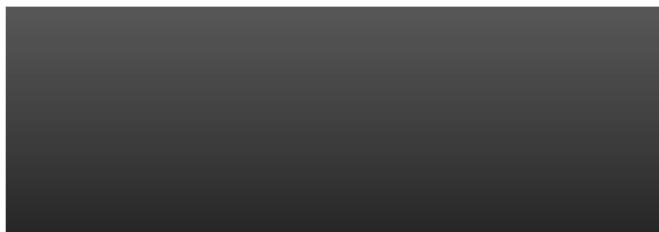
```

background: -webkit-linear-gradient(red, blue); /* untuk
Safari 5.1 to 6.0 */
background: -o-linear-gradient(red, blue); /* untuk Opera
11.1 to 12.0 */
background: -moz-linear-gradient(red, blue); /* untuk Firefox
3.6 to 15 */
background: linear-gradient(red, blue); /* sintaks standar,
harus diakhirkkan */
}
</style>
</head>
<body>
<h3>Contoh gradien linear dari merah ke biru</h3>
<div id="grad1"></div>
</body>
</html>

```

Apabila dilihat di browser, terlihat tampilan seperti berikut.

Contoh gradien linear dari merah ke biru



Gambar 3.15 Tampilan terlihat di browser

Untuk mengganti gradien dari kiri ke kanan, Anda bisa mengedit kode menjadi seperti berikut ini:

```

<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
height: 200px;
background: -webkit-linear-gradient(left, red , blue); /*
untuk Safari 5.1 to 6.0 */
background: -o-linear-gradient(right, red, blue); /* untuk
Opera 11.1 to 12.0 */
background: -moz-linear-gradient(right, red, blue); /* untuk
Firefox 3.6 to 15 */
background: linear-gradient(to right, red , blue); /* sintaks
standar (harus diakhirkan */
}
</style>
</head>

```

```
<body>
<h3>Gradien Linear dari Kiri Ke Kanan</h3>
<div id="grad1"></div>
</body>
</html>
```

Kalau kode di atas dibuka di browser, terlihat ada gradien dari sebelah kiri merah menuju ke kanan berwarna biru.

Gradien Linear dari Kiri Ke Kanan



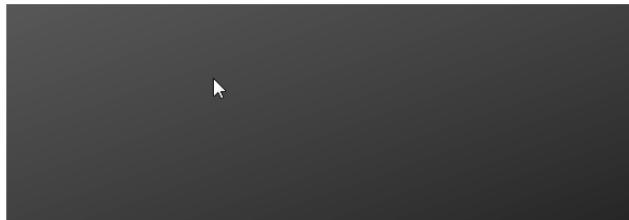
Gambar 3.16 Gradien linear dari kiri ke kanan

Jenis gradien linear berikutnya adalah diagonal. Anda bisa membuat gradien diagonal dengan menentukan posisi horizontal dan vertikal awalnya. Berikut ini contoh bagaimana menampilkan gradien linear yang bermula di kiri atas, ke arah kanan bawah:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 200px;
  background: -webkit-linear-gradient(left top, red , blue); /* untuk Safari 5.1 to 6.0 */
  background: -o-linear-gradient(bottom right, red, blue); /* untuk Opera 11.1 to 12.0 */
  background: -moz-linear-gradient(bottom right, red, blue); /* untuk Firefox 3.6 to 15 */
  background: linear-gradient(to bottom right, red , blue); /* sintaks standar, harus diakhirkkan */
}
</style>
</head>
<body>
<h3>Gradien Linear diagonal</h3>
<div id="grad1"></div>
</body>
</html>
```

Apabila ditampilkan di browser, terlihat tampilan seperti berikut.

Gradien Linear diagonal



Gambar 3.17 Tampilan gradien diagonal

Gradien tipe berikutnya adalah gradien tipe radial. Cirinya adalah pusatnya di tengah. Untuk membuat gradien radial, Anda perlu paling tidak memiliki dua color stop.

Sintaksnya seperti berikut:

```
background: radial-gradient(ukuran shape di posisi, warna-awal, ..., warna-akhir);
```

Secara default, bentuk shape standar adalah ellipse, ukurannya adalah ujung paling jauh, dan posisinya di tengah. Berikut ini contoh kode bagaimana membuat gradien radial dengan CSS3:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
  height: 150px;
  width: 200px;
  background: -webkit-radial-gradient(red, green, blue); /* Safari 5.1 sampai 6.0 */
  background: -o-radial-gradient(red, green, blue); /* Untuk Opera 11.6 sd 12.0 */
  background: -moz-radial-gradient(red, green, blue); /* Untuk Firefox 3.6 sd 15 */
  background: radial-gradient(red, green, blue); /* sintaks standar, harus diakhirkan */
}
</style>
</head>
<body>

<h3>Gradien radial</h3>
<div id="grad1"></div>
</body>
</html>
```

Apabila ditampilkan di browse, terlihat tampilan gradien ellipse dengan pusat berwarna merah, dan ujung berwarna biru.

Gradien radial



Gambar 3.18 Gradien ellipse radial

Parameter shape menentukan bentuk dari gradien. Anda bisa memilih antara circle atau ellipse, di mana bentuk default-nya adalah ellipse. Berikut ini contoh kodennya:

```
<!DOCTYPE html>
<html>
<head>
<style>
#grad1 {
height: 150px;
width: 200px;
background: -webkit-radial-gradient(red, yellow, green); /* For Safari 5.1 to 6.0 */
background: -o-radial-gradient(red, yellow, green); /* For Opera 11.6 to 12.0 */
background: -moz-radial-gradient(red, yellow, green); /* For Fx 3.6 to 15 */
background: radial-gradient(red, yellow, green); /* Standard syntax (must be last) */
}

#grad2 {
height: 150px;
width: 200px;
background: -webkit-radial-gradient(circle, red, yellow, green); /* For Safari 5.1 to 6.0 */
background: -o-radial-gradient(circle, red, yellow, green); /* For Opera 11.6 to 12.0 */
background: -moz-radial-gradient(circle, red, yellow, green); /* For Fx 3.6 to 15 */
background: radial-gradient(circle, red, yellow, green); /* sintaks standar, harus terakhir */
}
```

```
</style>
</head>
<body>

<h3>Gradien Radial Shape</h3>
<p><strong>Ellipse (ini bentuk default-nya):</strong></p>
<div id="grad1"></div>

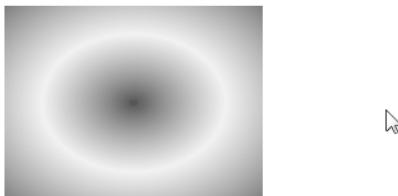
<p><strong>Lingkaran:</strong></p>
<div id="grad2"></div>

</body>
</html>
```

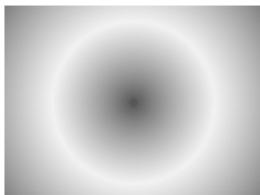
Kode di atas jika ditampilkan di browser, terlihat seperti berikut.

Gradien Radial Shape

Ellipse (ini bentuk default-nya):



Lingkaran:



Gambar 3.19 Tampilan shape saat berbentuk ellipse dan lingkaran

3.7 Efek Teks

CSS3 memiliki beberapa fitur teks baru. Di sini Anda akan belajar menggunakan text-shadow dan word-wrap.

Properti text-shadow berguna untuk memberikan bayangan dari teks. Berikut ini contoh bagaimana menggunakan text-shadow:

```
<!DOCTYPE html>
<html>
<head>
<style>
h1 {
    text-shadow: 5px 5px 5px #FF0000;
}
</style>
</head>
<body>

<h1>Efek bayangan!</h1>

</body>
</html>
```

Apabila ditampilkan di browser, teks tersebut akan memiliki efek bayangan.

Efek bayangan!

Gambar 3.20 Teks memiliki efek bayangan

Efek teks kedua adalah word-wrap. Ini akan meringkas teks untuk mencocoki area tertentu. Anda bisa menggunakan properti word-wrap untuk memaksakan teks agar pas terhadap tempatnya, walaupun harus memecah teks di tengah kata.

Berikut ini contoh kode yang akan digunakan untuk mengetes word-wrap:

```
<!DOCTYPE html>
<html>
<head>
<style>
p.test {
    width: 11em;
    border: 1px solid #000000;
    word-wrap: break-word;
}
</style>
</head>
<body>

<p class="test"> Teks ini sangat
panjaaaaaaaaaaaaaaaaaaaaaaaaaaaaang sekali. Tapi
tidak akan melewati kotak border karena menggunakan properti
word-wrap
</body>
</html>
```

Apabila ditampilkan, terlihat tampilan seperti berikut.

Teks ini sangat
panjaaaaaaaang
sekali. Tapi tidak akan
melewati kotak border
karena menggunakan
properti word-wrap

Gambar 3.21 Teks setelah diaktifkan word-wrap

Berikut ini beberapa properti lain dari CSS 3 untuk efek teks.

| Properti | Deskripsi |
|---------------------|---|
| hanging-punctuation | Menentukan apakah karakter tanda baca bisa diletakkan di luar box. |
| punctuation-trim | Menentukan apakah karakter tanda baca harus dipotong. |
| text-align-last | Menentukan bagaimana baris terakhir dari blok sebelum line break di-alignment ketika text-alignment dalam posisi justify. |
| text-emphasis | Menerapkan tanda penekanan, dan warna foreground ke elemen teks. |
| text-justify | Menentukan justifikasi yang digunakan ketika text-align diset ke justify. |
| text-outline | Menentukan outline teks. |
| text-overflow | Menentukan apa yang terjadi ketika teks melebihi elemen kontainer. |
| text-shadow | Menambahkan bayangan ke teks. |
| text-wrap | Menentukan aturan line breaking untuk teks. |
| word-break | Menentukan aturan line breaking untuk script non-CJK |
| word-wrap | Memungkinkan kata yang panjang, dan tidak bisa dipisahkan menjadi terpisah dan mengalami wrap ke baris berikutnya. |

3.8 Web Font

Web Font memungkinkan desainer web untuk menggunakan font yang tidak diinstal di komputer user. Kalau Anda ingin membuat tampilan font yang istimewa, Anda bisa memasukkan file font di web server. Otomatis akan di-load ke user ketika sudah dibutuhkan. Jadi pengaturan font ini di server-side dan bukan di client side.

Anda perlu mendefinisikan font terlebih dahulu dengan menggunakan rule @font-face. Anda perlu mengenal beberapa tipe format font yang berbeda:

- TrueType Fonts (TTF).

TrueType adalah jenis font standar yang dikembangkan di akhir tahun 1980 oleh Apple dan Microsoft. TrueType adalah jenis font paling lazim untuk sistem operasi MAC ataupun Windows.

- OpenType Fonts (OTF).

OpenType adalah format untuk font scalable, dibangun di atas TrueType dan merupakan merek dagang dari Microsoft. OpenType adalah tipe font yang lazim dipakai di semua platform komputer.

- Web Open Font Format (WOFF).

WOFF adalah tipe format font yang lazim dipakai di halaman web. Ini dikembangkan sejak tahun 2009 dan merupakan rekomendasi W3C. WOFF pada dasarnya adalah tipe OpenType atau TrueType dengan kompresi dan tambahan metadata. Tujuannya mendukung distribusi font dari server ke client melalui jaringan dengan bandwidth terbatas.

- SVG Fonts/Shapes.

Font SVG memungkinkan fileSVG digunakan sebagai glyph ketika menampilkan teks. Spek SVG 1.1 mendefinisikan modul font yang memungkinkan pembuatan font di dalam dokumen SVG. Anda juga bisa menerapkan CSS ke dokumen SVG dan rule @font-face bisa diterapkan ke dokumen SVG.

- Embedded OpenType Fonts (EOT).

Ini adalah tipe compact dari OpenType yang didesain oleh Microsoft dan digunakan sebagai font embedded pada webpage.

Untuk menggunakan font yang Anda inginkan, Anda perlu terlebih dahulu mendefinisikan nama untuk font dan kemudian mengacu pada file font-nya. Berikut ini contoh kode yang akan dipakai untuk mengakses file font sensation_light.woff:

```
<!DOCTYPE html>
<html>
<head>
<style>
@font-face {
  font-family: fontPertama;
  src: url(sansation_light.woff);
}

div {
  font-family: fontPertama;
}
</style>
</head>
<body>
Ini teks standar
<div>
Ini teks yang diubah font-nya menggunakan CSS3.
</div>

</body>
</html>
```

Apabila dijalankan di browser, Anda bisa melihat perbedaan antara teks standar dan teks yang sudah diberi efek font, di mana file font-nya ada di server dan tidak ada di komputer client.

Ini teks standar

Ini teks yang diubah font-nya menggunakan CSS3.

Gambar 3.22 Perbedaan teks standar dan teks di server

Anda bisa menggunakan teks bold untuk memasukkan rule @font-face deskriptor. Artinya, Anda bisa mendefinisikan font untuk bold sendiri.

Berikut ini contoh kode yang mendefinisikan bold:

```
<!DOCTYPE html>
<html>
<head>
<style>
@font-face {
    font-family: fontpertama;
    src: url(sansation_light.woff);
}

@font-face {
    font-family: fontpertama;
    src: url(sansation_bold.woff);
    font-weight: bold;
}

div {
    font-family: fontpertama;
}
</style>
</head>
<body>

<div>
Ini teks dengan font baru, Anda bisa membuat teks <b>tebal</b>
dengan font baru.
</div>

</body>
</html>
```

Apabila dijalankan, terlihat style tebal dengan font baru seperti berikut.

Ini teks dengan font baru, Anda bisa membuat teks **tebal** dengan font baru.

Gambar 3.23 Teks dengan font baru

Berikut ini adalah beberapa font descriptor yang bisa didefinisikan di dalam aturan @font-face.

| Deskriptor | Nilai | Deskripsi |
|-------------|-------|---|
| font-family | nama | Diperlukan. Mendefinisikan nama untuk font-nya. |
| src | URL | Dibutuhkan. Menentukan url dari file font. |

| Deskriptor | Nilai | Deskripsi |
|---------------|--|--|
| font-stretch | normal condensed ultra-condensed extra-condensed semi-condensed expanded semi-expanded extra-expanded ultra-expanded | Opsional, mendefinisikan bagaimana font harus di-stretching. Nilai default adalah "normal" |
| font-style | normal italic oblique | Opsional. Mendefinisikan bagaimana font harus diberi style, nilai default-nya adalah normal. |
| font-weight | normal bold 100 200 300 400 500 600 700 800 900 | Opsional. Mendefinisikan ketebalan font. Nilai default adalah normal. |
| unicode-range | unicode-range | Opsional. Mendefinisikan range dari karakter unicode dari dukung font, nilainya "U+0-10FFFF" |

3.9 Transformasi 2D

CSS3 mendukung transformasi objek. Dengan transform, Anda bisa memindahkan, mengatur ukuran, membelokkan, memutar, dan merentangkan elemen.

Sebuah transformasi adalah efek yang memungkinkan elemen mengubah bentuk, ukuran, dan posisinya. Anda bisa mentransformasikan elemen menggunakan transformasi 2D atau 3D. Tapi di bagian ini hanya akan dijelaskan transformasi 2D. Ada beberapa method yang bisa digunakan:

- translate()
- rotate()
- scale()
- skew()
- matrix()

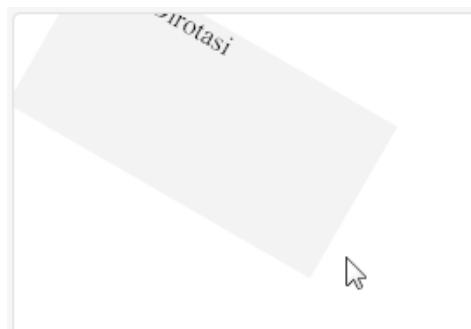
Untuk merotasi, Anda tinggal menentukan derajat yang akan dipakai untuk rotasi:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 200px;
  height: 100px;
  background-color: yellow;
  /* Rotate div */
  -ms-transform: rotate(30deg); /* IE 9 */
  -webkit-transform: rotate(30deg); /* Chrome, Safari, Opera */
  transform: rotate(30deg); /* Sintaks standar */
}
</style>
</head>
<body>

<div>Kotak Dirotasi</div>

</body>
</html>
```

Apabila dilihat di browser, terlihat gambar kotak dirotasi.



Gambar 3.24 Kotak dirotasi 30 degree

Method `translate()` digunakan untuk memindahkan elemen dari posisi sekarang, tergantung kepada parameter yang diberikan untuk posisi kiri (sumbu X) dan atas (sumbu Y).

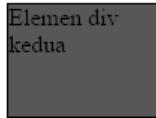
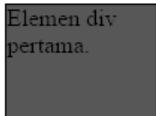
Berikut ini contoh kode untuk melakukan translasi:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 75px;
    background-color: red;
    border: 1px solid black;
}

div#div2 {
    -ms-transform: translate(50px,100px); /* IE 9 */
    -webkit-transform: translate(50px,100px); /* Chrome, Safari, Opera */
    transform: translate(50px,100px); /* Sintaks standar */
}
</style>
</head>
<body>

<div>Elemen div pertama.</div>
<div id="div2">Elemen div kedua</div>
</body>
</html>
```

Ketika dijalankan di browser, terlihat ada dua elemen `<div>`.



Gambar 3.25 Elemen <div>

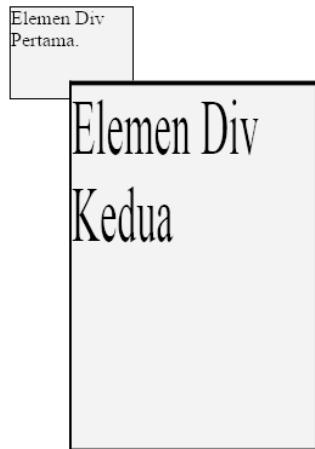
Method transformasi 2D berikutnya adalah scale(). Dengan method scale(), Anda bisa menambahkan atau mengurangi ukuran tergantung parameter yang diberikan di lebar (sumbu X) dan tinggi (sumbu Y). Contoh kodennya seperti ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 75px;
    background-color: yellow;
    border: 1px solid black;
}

div#div2 {
    margin: 100px;
    -ms-transform: scale(2,4); /* IE 9 */
    -webkit-transform: scale(2,4); /* Chrome, Safari, Opera */
    transform: scale(2,4); /* Sintaks standar */
}
</style>
</head>
<body>

<div>Elemen Div Pertama.</div>
<div id="div2">Elemen Div Kedua </div>
</body>
</html>
```

Apabila dijalankan, Anda bisa melihat div kedua ukurannya sudah berubah terbentang karena mengalami scale dengan method scale().



Gambar 3.26 Method scale()

Anda juga bisa memiringkan objek dengan menggunakan skew. Dengan method skew(), elemen akan dimiringkan pada sudut tertentu, tergantung kepada parameter yang diberikan untuk horizontal (sumbu X) dan vertikal (sumbu Y).

Contoh kodennya seperti berikut:

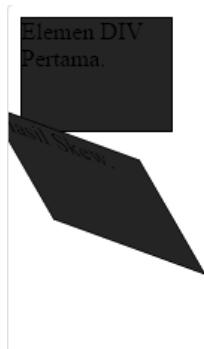
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 75px;
  background-color: blue;
  border: 1px solid black;
}

div#div2 {
  -ms-transform: skew(30deg,20deg); /* IE 9 */
  -webkit-transform: skew(30deg,20deg); /* Chrome, Safari, Opera */
  transform: skew(30deg,20deg); /* sintaks standar */
}
</style>
</head>
<body>

<div>Elemen DIV Pertama.</div>
<div id="div2">Hasil Skew.</div>

</body>
</html>
```

Apabila dilihat di browser, terlihat tampilan div kedua menjadi termiringkan karena skew().



Gambar 3.27 Tampilan div kedua menjadi skew

Anda bisa menggunakan method matrix untuk mengombinasikan semua metode transformasi 2D menjadi satu. Method Matrix memungkinkan Anda memasukkan enam parameter, berisi fungsi matematik untuk melakukan rotate,s cale, move (translate) dan skew. Contoh kodennya seperti ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 75px;
    background-color: yellow;
    border: 1px solid black;
}

div#div2 {
    -ms-transform: matrix(0.866,0.5,-0.5,0.866,0,0); /* IE 9 */
    -webkit-transform: matrix(0.866,0.5,-0.5,0.866,0,0); /* Chrome, Safari, Opera */
    transform: matrix(0.866,0.5,-0.5,0.866,0,0); /* Sintaks standar */
}
</style>
</head>
<body>

<div>Elemen DIV pertama.</div>
<div id="div2">Elemen setelah transformasi matrix.</div>

</body>
</html>
```

Contoh tampilan setelah elemen di transformasi matrix seperti berikut.



Gambar 3.28 Elemen setelah ditransformasi dengan matrix

Berikut ini rangkuman beberapa fungsi yang digunakan untuk transformasi 2 dimensi.

| Fungsi | Deskripsi |
|---------------------|---|
| matrix(n,n,n,n,n,n) | Mendefinisikan transformasi 2D menggunakan matrix dengan enam nilai. |
| translate(x,y) | Mendefinisikan translasi 2D, menggerakkan elemen sepanjang sumbu X dan sumbu Y. |
| translateX(n) | Mendefinisikan translasi 2D, menggerakkan elemen sepanjang sumbu X. |
| translateY(n) | Mendefinisikan translasi 2D, menggerakkan elemen sepanjang sumbu Y. |
| scale(x,y) | Mendefinisikan transformasi 2D scale, mengubah ukuran width dan height dari elemen. |
| scaleX(n) | Mendefinisikan transformasi 2D scale, mengubah width dari elemen. |

| Fungsi | Deskripsi |
|--------------------------------|--|
| scaleY(<i>n</i>) | Mendefinisikan transformasi 2D scale, mengubah height dari elemen. |
| rotate(<i>angle</i>) | Mendefinisikan rotasi 2D, sudutnya ditentukan di parameter. |
| skew(<i>x-angle,y-angle</i>) | Mendefinisikan transformasi 2D skew sepanjang sumbu X dan sumbu Y. |
| skewX(<i>angle</i>) | Mendefinisikan transformasi 2D skew sepanjang sumbu X. |
| skewY(<i>angle</i>) | Mendefinisikan transformasi 2D skew sepanjang sumbu Y. |

3.10 Transformasi 3D

CSS3 memungkinkan Anda memformat elemen menggunakan transformasi 3 dimensi. Dua di antara method untuk transformasi 3D yang paling lazim dipakai adalah:

- rotateX()
- rotateY()

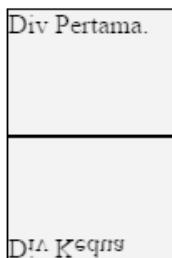
Method rotateX() berguna untuk merotasi elemen di sumbu X-nya pada sudut tertentu. Contoh penggunaan kodennya seperti berikut:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 75px;
    background-color: yellow;
    border: 1px solid black;
}

div#div2 {
    -webkit-transform: rotateX(180deg); /* Chrome, Safari, Opera */
    transform: rotateX(180deg); /* Sintaks standar */
}
</style>
</head>
<body>
```

```
<div>Div Pertama.</div>
<div id="div2">Div Kedua</div>
</body>
</html>
```

Hasilnya saat dilihat di browser seperti berikut.



Gambar 3.29 Tampilan saat hasil transformasi 3D dengan rotateX browser

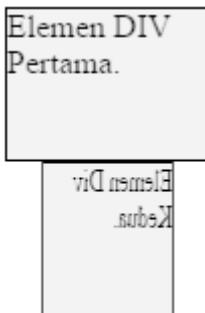
Method rotateY() digunakan untuk memutar elemen berdasarkan sumbu Y-nya pada sudut tertentu.

Contoh kodennya seperti berikut:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 75px;
    background-color: yellow;
    border: 1px solid black;
}

div#div2 {
    -webkit-transform: rotateY(130deg); /* Chrome, Safari, Opera */
    transform: rotateY(130deg); /* Sintaks standar */
}
</style>
</head>
<body>

<div>Elemen DIV Pertama.</div>
<div id="div2">Elemen Div Kedua.</div>
</body>
</html>
```



Gambar 3.30 Contoh transformsai rotateY

Berikut ini beberapa method transform 3D yang ada di CSS3.

| Fungsi | Deskripsi |
|---|---|
| matrix3d (n,n,n,n,n,n,n,n,n,n,n,n,n,n,n) | Mendefinisikan transformasi 3D menggunakan matrix 4x4 yang terdiri dari 16 nilai. |
| translate3d(x,y,z) | Mendefinisikan translasi 3D. |
| translateX(x) | Mendefinisikan translasi 3D menggunakan nilai untuk sumbu X. |
| translateY(y) | Mendefinisikan translasi 3D, menggunakan nilai dari sumbu Y. |
| translateZ(z) | Mendefinisikan translasi 3D, menggunakan hanya nilai sumbu Z. |
| scale3d(x,y,z) | Mendefinisikan transformasi 3D scale. |
| scaleX(x) | Mendefinisikan transformasi 3D scale menggunakan nilai sumbu X. |

| Fungsi | Deskripsi |
|------------------------------------|--|
| <code>scaleY(y)</code> | Mendefinisikan transformasi 3D scale menggunakan nilai untuk sumbu Y. |
| <code>scaleZ(z)</code> | Mendefinisikan transformasi 3D scale menggunakan nilai dari sumbu Z. |
| <code>rotate3d(x,y,z,angle)</code> | Mendefinisikan rotasi 3D. |
| <code>rotateX(angle)</code> | Mendefinisikan rotasi 3D sepanjang sumbu X. |
| <code>rotateY(angle)</code> | Mendefinisikan rotasi 3D sepanjang sumbu Y. |
| <code>rotateZ(angle)</code> | Mendefinisikan rotasi 3D sepanjang sumbu Z. |
| <code>perspective(n)</code> | Mendefinisikan tampilan perspektif untuk elemen yang ditransformasikan 3D. |

3.11 Transisi

Transisi adalah efek baru di CSS 3 yang memungkinkan sebuah elemen berubah dari satu kondisi ke kondisi style lainnya. Untuk melakukan ini, Anda harus memiliki dua hal:

- Properti CSS yang ingin ditambahkan.
- Durasi dari efek.

Efek transisi akan dimulai ketika properti CSS tertentu berubah nilainya, misalnya ketika mouse ada di atas komponen atau sedang hover. Berikut ini contoh kodennya:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  width: 100px;
  height: 100px;
  background: blue;
  -webkit-transition: width 2s; /* For Safari 3.1 to 6.0 */
  transition: width 2s;
}
```

```
    transition: width 2s;
}

div:hover {
    width: 300px;
}
</style>
</head>
<body>
```

Letakkan pointer mouse di atas kotak

```
<div></div>
```

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

Awalnya ada kotak dengan ukuran 100x100, kemudian Anda meletakkan pointer di atas kotak tersebut.

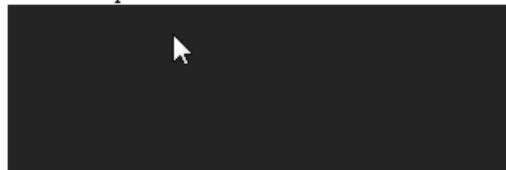
Letakkan pointer mouse di atas kotak



Gambar 3.31 Meletakkan pointer di atas kotak

Maka akan ada animasi kotak memanjang sampai ukurannya 300px, di mana waktu animasi tersebut selama 2 detik.

Letakkan pointer mouse di atas kotak



Gambar 3.32 Animasi memanjang

Perubahan dalam transisi bisa dibuat lebih dari satu properti CSS, Anda bisa memisahkan properti dengan tanda koma.

Berikut ini contoh kodennya:

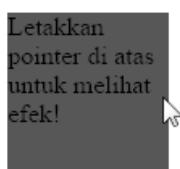
```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    width: 100px;
    height: 100px;
    background: red;
    -webkit-transition: width 2s, height 2s, -webkit-transform
2s; /* untuk Safari 3.1 sd 6.0 */
    transition: width 2s, height 2s, transform 2s;
}

div:hover {
    width: 200px;
    height: 200px;
    -webkit-transform: rotate(180deg); /* Chrome, Safari, Opera
*/
    transform: rotate(180deg);
}
</style>
</head>
<body>

<div>Letakkan pointer di atas untuk melihat efek!</div>

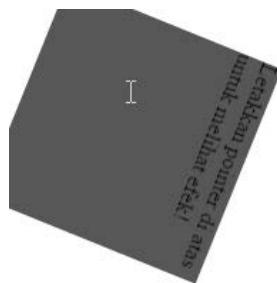
</body>
</html>
```

Ketika awal, belum ada efek dan Anda bisa meletakkan pointer mouse ke atas <div>.



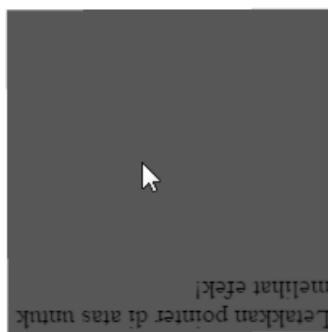
Gambar 3.33 Tampilan awal

Setelah di-hover, maka muncul animasi.



Gambar 3.34 Animasi dalam posisi pertengahan

Setelah selesai, maka transformasi berhenti, ada 2 jenis perubahan, yaitu perubahan ukuran dan perubahan rotasi.



Gambar 3.35 Perubahan rotasi

Berikut ini beberapa properti transisi CSS3 yang bisa Anda gunakan untuk melakukan transisi pada objek.

| Properti | Deskripsi |
|------------|---|
| transition | Properti untuk menyeting empat properti transisi menjadi satu properti. |

| Properti | Deskripsi |
|----------------------------|--|
| transition-delay | Menentukan kapan transisi akan dimulai. |
| transition-duration | Menentukan berapa detik atau milidetik transisi akan berhenti. |
| transition-property | Menentukan nama dari properti CSS. |
| transition-timing-function | Menentukan kecepatan kurva dari efek transisi. |

3.12 Multi Kolom

Dengan CSS3, Anda bisa membuat lebih dari satu kolom untuk menampilkan teks dalam objek seperti `<div>` atau paragraf. Ini sangat memudahkan me-layout, terutama untuk me-layout situs-situs berita. Anda tidak perlu membuat tabel secara manual, sangat cocok dipakai untuk mengatur layout.

Untuk menentukan jumlah kolom, Anda bisa menggunakan properti `column-count`. Berikut ini contoh multi kolom:

```

<!DOCTYPE html>
<html>
<head>
<style>
.tampilanKoran {
  -webkit-column-count: 3; /* Chrome, Safari, Opera */
  -moz-column-count: 3; /* Firefox */
  column-count: 3;
}
</style>
</head>
<body>

<div class="tampilanKoran">
Samanhudi atau sering disebut Kyai Haji Samanhudi (lahir di Laweyan, Surakarta, Jawa Tengah, 1868; meninggal di Klaten, Jawa Tengah, 28 Desember 1956) adalah pendiri Sarekat Dagang Islam, sebuah organisasi massa di Indonesia yang awalnya merupakan wadah bagi para pengusaha batik di Surakarta. Nama kecilnya ialah Sudarno Nadi.

```

Dalam dunia perdagangan, Samanhudi merasakan perbedaan perlakuan oleh penguasa Hindia Belanda antara pedagang pribumi yang mayoritas beragama Islam dengan pedagang Tionghoa pada tahun 1905. Oleh sebab itu Samanhudi merasa pedagang pribumi

harus mempunyai organisasi sendiri untuk membela kepentingan mereka. Pada tahun 1905, ia mendirikan Sarekat Dagang Islam untuk mewujudkan cita-citanya.

Ia dimakamkan di Banaran, Grogol, Sukoharjo. Sesudah itu, Serikat Islam dipimpin oleh Oemar Said Tjokroaminoto.
Sumber:id.wikipedia.org

</div>

</body>
</html>

Maka kode di atas akan menghasilkan tampilan <div> yang otomatis terlihat menjadi tiga kolom tanpa Anda harus membuat tabel. Kalau ukuran layar dikecilkan dan dibesarkan, maka tampilan teks ketiga kolom ini otomatis ter-update. Sesuai dengan prinsip kerja web responsif.

Samanhudi atau sering disebut Kyai Haji Samanhudi (lahir di Laweyan, Surakarta, Jawa Tengah, 1868; meninggal di Klaten, Jawa Tengah, 28 Desember 1956) adalah pendiri Sarekat Dagang Islam, sebuah organisasi massa di Indonesia yang awalnya merupakan wadah bagi para pengusaha batik di Surakarta. Nama

kecilnya ialah Sudarno Nadi. Dalam dunia perdagangan, Samanhudi merasakan perbedaan perlakuan oleh pengusa Hindia Belanda antara pedagang pribumi yang mayoritas beragama Islam dengan pedagang Tionghoa pada tahun 1905. Oleh sebab itu Samanhudi merasa pedagang pribumi harus mempunyai

organisasi sendiri untuk membela kepentingan mereka. Pada tahun 1905, ia mendirikan Sarekat Dagang Islam untuk mewujudkan cita-citanya. Ia dimakamkan di Banaran, Grogol, Sukoharjo. Sesudah itu, Serikat Islam dipimpin oleh Oemar Said Tjokroaminoto.
Sumber:id.wikipedia.org



Gambar 3.36 Hasil pembuatan kode responsif

Berikut ini beberapa properti yang berkaitan dengan efek multikolom di CSS3.

| Properti | Deskripsi |
|--------------|---|
| column-count | Menentukan jumlah kolom yang akan dibuat untuk membagi elemen tersebut. |
| column-fill | Menentukan bagaimana mem-fill kolom. |

| Properti | Deskripsi |
|-------------------|--|
| column-gap | Menentukan jarak atau gap antar kolom. |
| column-rule | Properti ringkas untuk menyetting semua aturan properties column-rule-*. |
| column-rule-color | Menentukan aturan warna antar kolom. |
| column-rule-style | Menentukan aturan style antar kolom. |
| column-rule-width | Menentukan lebar antar kolom. |
| column-span | Menentukan jumlah kolom sebuah elemen harus dibentangkan. |
| column-width | Menentukan lebar kolom. |
| columns | Properti ringkas untuk menyetting column-width dan column-count. |

3.13 User Interface

Di CSS3, ada beberapa antarmuka baru yang diperkenalkan, seperti resize box, outline dan resize.

Di chapter ini, Anda akan belajar bagaimana mengatur properties antarmuka seperti berikut:

- resize
- box-sizing
- outline-offset

Di CSS3, properti resize menentukan apakah elemen bisa di-resize ukurannya oleh user atau tidak.

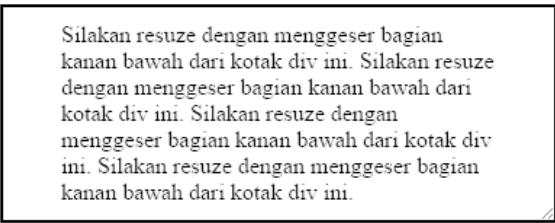
Berikut ini contoh kode yang membuat kotak bisa di-resize:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
    border: 2px solid;
    padding: 10px 40px;
    width: 300px;
    resize: both;
    overflow: auto;
}
</style>
</head>
<body>

<div>Silakan resuze dengan menggeser bagian kanan bawah dari
kotak div ini.
Silakan resuze dengan menggeser bagian kanan bawah dari kotak
div ini.
Silakan resuze dengan menggeser bagian kanan bawah dari kotak
div ini.
Silakan resuze dengan menggeser bagian kanan bawah dari kotak
div ini.
</div>

</body>
</html>
```

Kalau awal dijalankan, Anda bisa melihat kotak memiliki ukuran standar.

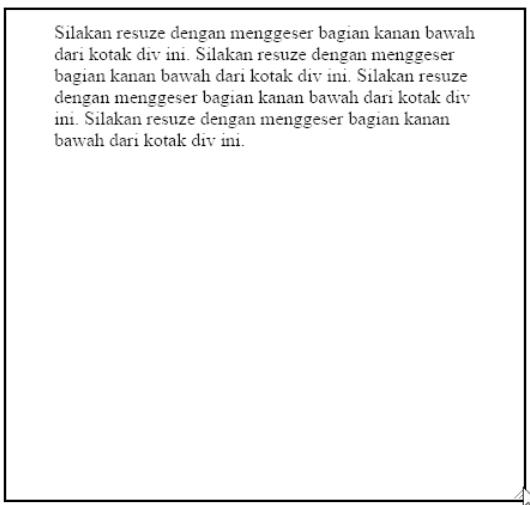


Silakan resuze dengan menggeser bagian
kanan bawah dari kotak div ini. Silakan resuze
dengan menggeser bagian kanan bawah dari
kotak div ini. Silakan resuze dengan
menggeser bagian kanan bawah dari kotak div
ini. Silakan resuze dengan menggeser bagian
kanan bawah dari kotak div ini.

Gambar 3.37 Tampilan saat awal dijalankan

Anda bisa me-resize dengan meletakkan pointer di bagian kanan bawah, kemudian me-resize sesuai kebutuhan.

Silakan resuze dengan menggeser bagian kanan bawah dari kotak div ini. Silakan resuze dengan menggeser bagian kanan bawah dari kotak div ini. Silakan resuze dengan menggeser bagian kanan bawah dari kotak div ini. Silakan resuze dengan menggeser bagian kanan bawah dari kotak div ini.



Gambar 3.38 Tampilan kotak setelah di-resize

Properti berikutnya adalah box sizing yang digunakan untuk memberitahukan browser berapa properti sizing, seperti width dan height yang harus dimasukkan.

Misalnya, apakah harus memasukkan border-box atau content-box yang menjadi ukuran standar properti width dan height. Misalnya apabila Anda ingin membuat dua kotak bersisian, maka Anda harus mengeset box-sizing ke border-box. Ini memaksa browser untuk me-render box dengan width dan height tertentu dan meletakkan border dan padding di dalam kotak. Contoh kodennya seperti ini:

```
<!DOCTYPE html>
<html>
<head>
<style>
div.container {
    width: 30em;
    border: 1em solid;
}

div.box
{
    -moz-box-sizing: border-box; /* Firefox */
    box-sizing: border-box;
    width: 50%;
    border: 1em solid red;
    float: left;
}
```

```
</style>
</head>
<body>

<div class="container">
<div class="box">Div pertama di sebelah kiri.</div>
<div class="box">Div kedua di sebelah kanan.</div>
</div>

</body>
</html>
```

Apabila ditampilkan di browser, terlihat tampilan seperti berikut ini, dua `<div>` diletakkan beriringan.



Gambar 3.39 Dua `<div>` terlihat bersisian

Properti `outline-offset` mengeset properti `offset` dari `outline` dan Anda bisa menggambarkannya di luar border.

Outline berbeda dengan border dalam dua hal:

- Outline tidak memakan tempat.
- Outline bisa berbentuk tidak kotak.

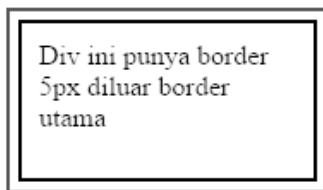
Berikut ini contoh bagaimana menggunakan `outline offset`:

```
<!DOCTYPE html>
<html>
<head>
<style>
div {
  margin: 20px;
  width: 150px;
  padding: 10px;
  height: 70px;
  border: 2px solid black;
  outline: 2px solid red;
  outline-offset: 5px;
}
</style>
</head>
<body>

<div>Div ini punya border 5px diluar border utama</div>

</body>
</html>
```

Apabila ditampilkan di browser, terlihat tampilan seperti berikut.



Gambar 3.40 Hasil pembuatan border

Berikut ini beberapa properti user-interface dari CSS3.

| Properti | Deskripsi |
|----------------|--|
| appearance | Memungkinkan Anda membuat tampilan elemen seperti standar antarmuka. |
| box-sizing | Mendefinisikan elemen tertentu agar fit area dengan cara tertentu. |
| icon | Memberi style pada elemen dengan ikon ekivalen. |
| nav-down | Menggunakan kunci navigasi panah bawah. |
| nav-index | Menentukan urutan tab untuk elemen. |
| nav-left | Menavigasi kapan menggunakan kunci navigasi panah kiri. |
| nav-right | Menentukan navigasi menggunakan panah kanan. |
| nav-up | Navigasi menggunakan panah ke atas. |
| outline-offset | Melakukan offset pada outline dan menggambar di luar batas border. |
| resize | Menentukan apakah elemen bisa di-resize oleh user atau tidak. |

BAB 4

Menggunakan Layout Fluid

Setelah belajar tentang media query di atas, Anda harus memahami layout fluid agar bisa membuat desain web responsif. Di bab ini Anda akan belajar beberapa hal berikut:

- Berbagai tipe layout.
- Prinsip bekerja dengan desain layout.
- Membangun desain fluid dengan grid CSS.
- Contoh pembuatan web page responsif.

4.1 Tipe Layout

Ketika Anda membicarakan struktur layout dari halaman web, ada beberapa tipe layout yang bisa dipilih. Tiga tipe layout yang paling populer adalah:

- Layout fixed-width.
- Layout fluid.
- Layout elastis.

Ketika ingin memutuskan layout mana yang ingin diambil, Anda perlu terlebih dahulu memahami user experience yang diinginkan. Untuk membuat situs responsif, pikirkan bahwa layout tersebut akan dilihat di berbagai perangkat.

Dengan beragamnya perangkat yang ada, maka pertimbangkan bagaimana layout yang baik, dari mulai dibuka dengan perangkat genggam terkecil, hingga televisi 80 inchi yang lazim dimiliki oleh orang-orang di luar negeri (kalau situs Anda memang menasaskan pengguna luar negeri).

Berikut ini beberapa tipe layout dan kelebihan serta kekurangan tiap-tiap situs tersebut.

4.1.1 Fixed Width

Layout fixed-width dibangun menggunakan wrapper yang memiliki lebar yang tetap. Ini diposisikan di layar, biasanya diletakkan di bagian tengah viewport browser. Jadi berapa pun ukuran layar yang dipakai, layout ini akan menampilkan width yang tetap.

Dulunya, ketika developer membangun website, umumnya dibangun dari tahap desain yang memiliki lebar tetap. Umumnya desainer akan menggunakan ukuran 960px, karena ini merupakan lebar ideal menggunakan grid, mengingat angka ini bisa dibagi 3,4,5,6,8,10,12, dan 15. Ketika developer ingin membuat website dengan tiga kolom, dua sidebar, dan konten area.

Salah satu masalah di sini adalah bahwasanya Anda mengasumsikan pengunjung memiliki ukuran layar standar di komputer desktop. Ketika Apple me-launching iPhone di tahun 2007, kebanyakan website menggunakan prinsip fixed-width, sehingga untuk membuat website yang besar akan sangat membingungkan.

Tapi menggunakan fixed width belum tentu jelek, terutama kalau Anda ingin tampilan yang seragam di semua viewport. Situs detiknews.com adalah contoh situs yang menggunakan layout fixed width.



Gambar 4.1 Tampilan detiknews saat viewport penuh

Ketika viewport dikecilkan, tampilan masih tetap memiliki layout yang sama atau fixed.



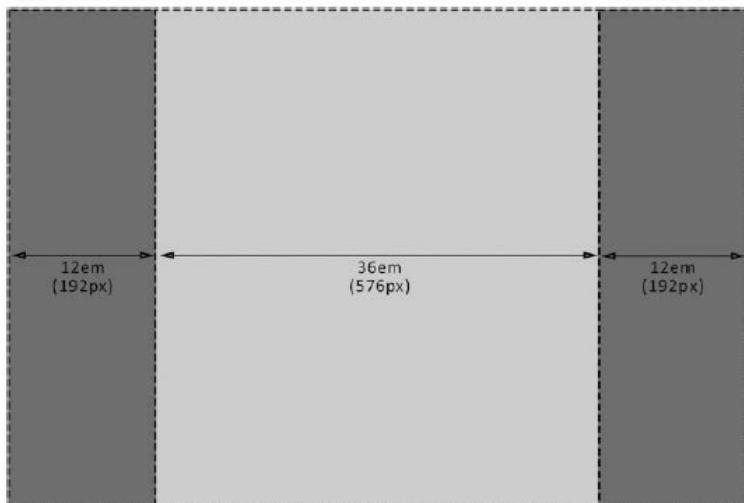
Gambar 4.2 Tampilan saat viewport dikecilkan

4.1.2 Elastis

Layout elastis tidak mendefinisikan width di piksel, tapi dalam satuan em. Unit em adalah kelipatan dari ukuran font. Jadi kalau ukuran font adalah 16px, maka 2em artinya 32px. Sehingga jika user mengubah ukuran font ketika melihat site, layout dari site akan berubah proporsional sesuai dengan peningkatan atau penurunan ukuran font.

Layout elastis membuat user memiliki kontrol lebih jauh terhadap website. User experience yang dialami oleh pengunjung juga lebih baik. Karena elemen dari situs akan me-resize sesuai dengan ukuran font. Sehingga developer bisa memastikan situs akan bisa diakses oleh semua user.

Jika Anda menggunakan desain standar dengan ukuran font 16px, Anda bisa mengedit ukuran menjadi em seperti contoh berikut.



Gambar 4.3 Pengubahan ukuran px menjadi em, untuk membuat konversi layout menjadi elastis

Cara untuk menghitung nilai em dari piksel sangat mudah, rumusnya seperti berikut:

Width dalam em = width dalam piksel / ukuran font dasar

Dengan menggunakan formula ini, Anda bisa menghitung nilai em pada gambar di atas:

$$\begin{aligned} 576/16 &= 36 \text{ em} \\ 192/16 &= 12 \text{ em} \end{aligned}$$

Perlu dipahami bahwa em bisa memiliki nilai desimal, sehingga jika perhitungan Anda tidak menghasilkan bilangan genap, Anda tidak perlu khawatir.

Salah satu masalah awal dari layout elastis adalah karena width dari site dihitung berdasarkan ukuran font, maka jika user menambahkan ukuran font melewati batas tertentu, tampilan webpage akan sangat terlihat besar di viewport, sehingga memunculkan scrollbar vertikal.

Selain itu, layout elastik pada prinsipnya masih berupa layout fixed width, hanya saja elastis yang dimaksud adalah berdasarkan ukuran font-nya. Jadi kalau ukuran font tidak diubah, tampilan situs ini masih tetap standar fixed-width.

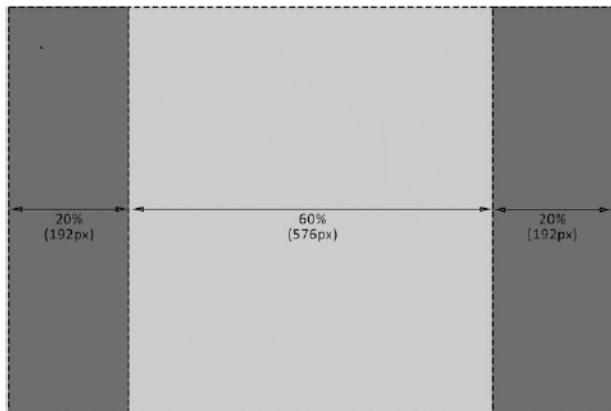
Jadi walaupun elastis berbeda dengan fixed-width yang berbasis piksel, tapi tidak fluid atau cair.

Salah satu sebab kenapa layout elastis kurang populer adalah karena Anda perlu menghitung terlebih dahulu nilai em-nya. Sebagian developer malas untuk melakukan perhitungan ini.

Selain itu em dikalkulasi relatif terhadap parent, sehingga jika parent mendefinisikan ukuran size yang berbeda dari body, maka width dari elemen akan berbeda dari yang Anda inginkan. Untuk menanggulangi hal ini, digunakan rem dan bukan em untuk ukuran font, artinya semua ukuran font akan relatif terhadap ukuran font base, tergantung ukuran elemen.

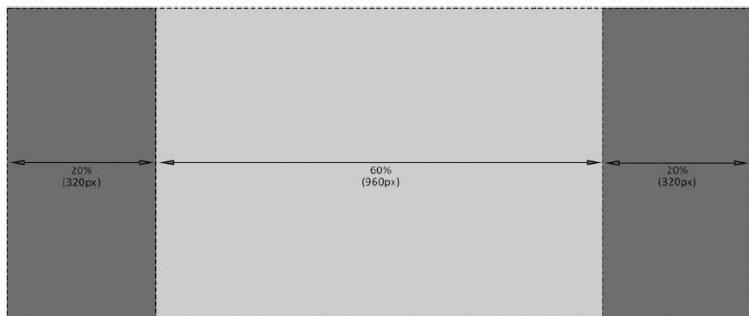
4.1.3 Fluid

Layout fluid atau layout relatif, width-nya akan berubah tergantung viewport. Tidak seperti layout fixed, di mana width didefinisikan di piksel, Anda mendefinisikan width dalam bentuk persentase yang menunjukkan persentase dari viewport.



Gambar 4.4 Contoh tampilan layout dengan persentase

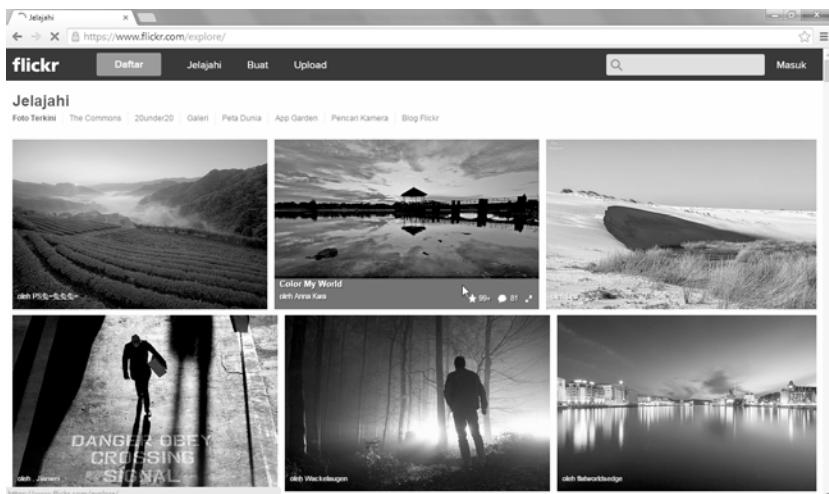
Misalnya pada contoh di atas, apabila viewport browser ditingkatkan menjadi 960px misalnya, maka nilai pikselnya akan turut bertambah, namun kolomnya proporsinya tetap sama.



Gambar 4.5 Contoh apabila viewport ditambahkan

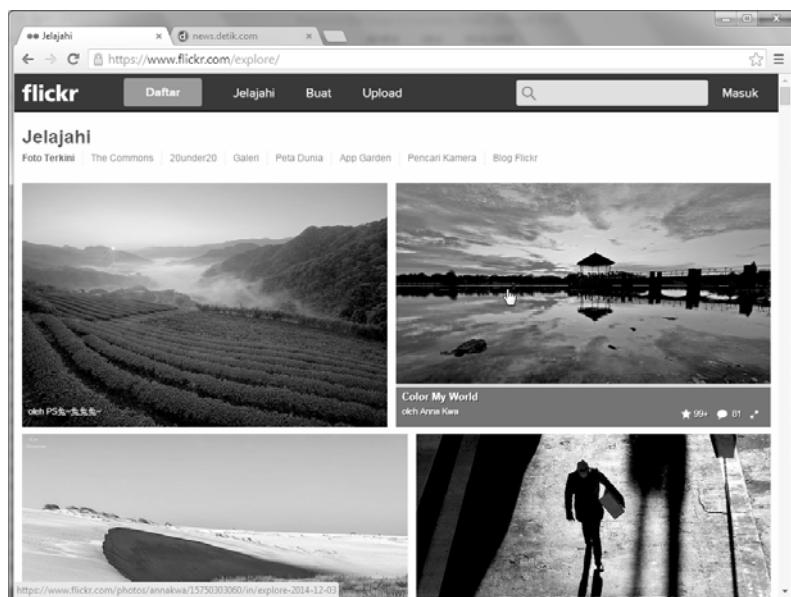
Jika Anda me-resize width viewport menjad 1600px, maka website otomatis akan terskala lagi. Begitu pula ketika lebar diperkecil seperti 320px, maka website akan terskala lagi, tapi proporsinya tetap sama.

Contoh website yang menggunakan layout fluid misalnya Flickr yang ketika tampilan besar menampilkan gambar full. Ini membuat user bisa melihat lebih banyak gambar.



Gambar 4.6 Tampilan Flickr dalam viewport lebar

Apabila diperkecil, area gambar akan mengecil, tapi tetap gambar memenuhi viewport.



Gambar 4.7 Tampilan ketika diperkecil

Salah satu keuntungan menggunakan Flickr adalah image bisa memenuhi area yang tersedia, sehingga pengunjung lebih mudah mengapresiasi foto. Berdasarkan pengalaman Flickr, 6 hari setelah pengubahan layout menjadi fluid, ada 80 juta foto baru yang di-upload, di mana 6 hari sebelumnya, user hanya meng-upload 47 juta foto saja. Layout fluid tidak hanya berguna untuk situs seperti Flickr yang sepenuhnya terdiri dari gambar. Tapi juga yang menggunakan teks, misalnya Pinterest.

4.2 Desain Responsif dengan Layout Fluid

Dari ketiga layout di atas, Anda bisa melihat bahwa tiap layout memiliki kelebihan dan kekurangan. Anda tinggal memadukan dengan media query yang akan melihat berapa ukuran viewport yang melihat halaman web.

Perangkat yang di-query oleh media query tidak hanya perangkat tertentu saja, tapi bisa sangat beragam, dari mulai televisi 80 inchi sampai ponsel 3,5 inchi. Anda harus mempertimbangkan bagaimana resize antar ukuran itu berjalan dengan smooth dan mengalir, atau dengan kata lain layout optimal terus menerus.

Ketika mempertimbangkan layout pilihan, Anda harus familiar dengan opsi lainnya. Jadi desain web responsif tidak berarti mengharamkan breakpoint, yang penting Anda mengoptimalkan user experience.

4.2.1 Prinsip Desain Fluid

Ketika bekerja dengan desain fluid untuk membuat desain responsif. Anda perlu memperhatikan beberapa prinsip utama berikut:

- Jangan menggunakan height yang fix.
- Jangan menggunakan scroll bar horizontal.
- Pikirkan bagaimana gambar Anda terlihat di berbagai ukuran.
- Pikirkan untuk melakukan wrapping pada konten.
- Pikirkan panjang teks Anda.

4.2.2 Contoh Pembuatan Layout

Anda bisa menggunakan CSS untuk membuat efek kolom dengan menggunakan CSS saja. Untuk layout dua kolom, di mana satu kolom lebih tinggi dari lainnya, Anda bisa mengeset elemen background dari parent ke kolom yang lebih pendek. Kemudian menyeting warna background dari elemen kolom yang lebih panjang.

Buat dulu kode HTML di <body> seperti berikut:

```
<body>
<div class="col-container">
<aside class="col">
Sidebar Samping
</aside>
<div class="col main">
Konten utama
</div>
</div>

</body>
```

Kemudian beri style seperti berikut:

```
<head>
<meta name="viewport" content="width=device-width, minimum-
scale=1.0, maximum-scale=1.0" />
<title>Demo responsif </title>

<style type="text/css">

.col-container{
background: #000;
color: #fff;
}
.col-container:after{
content: ' ';
clear: both;
display: block;
}
.col{
float: left;
width: 50%;
}
.col.main{
background: #999;
}
</style>

</head>
```

Ketika sudah dibuat, terlihat tampilan bersisian seperti berikut.



Gambar 4.8 Tampilan bersisian

Di contoh di atas, ada elemen tambahan yang semu yang ditambahkan ke kontainer kolom untuk mengosongkan kolom yang mengambang. Ini perlu agar browser bisa mengkalkulasi panjang container. Jika situs Anda sudah memakai class clearfix, Anda bisa menambahkan ini ke container kolom dan bukannya menambahkan elemen semu ke elemen tersebut.

Walaupun sederhana, tapi teknik CSS ini cukup mudah. Anda bisa juga menambahkan satu kolom lagi. Update kode body seperti berikut:

```
<body>
<div class="col-container">
<aside class="col">
Sidebar Samping
</aside>
<div class="col main">
Konten Utama Sebagai Body
</div>
<div class="col">
Konten Lainnya
</aside>
</div>
</body>
```

Kemudian CSS di-update seperti berikut:

```
<style type="text/css">
.col-container{
background: linear-gradient(to right, #000000 0%,#000000
33%,#a0a0a0 33%,#a0a0a0 66%,#a0a0a0
66%,#707070 66%);
color: #fff;
}
.col-container:after{
content: ' ';
clear: both;
display: block;
}
.col{
float: left;
width: 33.3%;
}
</style>
```

Apabila dilihat di browser, terlihat tampilan menjadi tiga kolom, dimana height sama, walaupun jumlah baris berbeda.

| | | |
|-----------------|------------------------------|----------------|
| Sidebar Samping | Konten Utama Sebagai Body | Konten Lainnya |
|-----------------|------------------------------|----------------|

Gambar 4.9 Tampilan tiga kolom dengan height yang sama

Selain CSS, Anda juga bisa me-layout memakai JavaScript. Pertama membuat beberapa kolom, kemudian me-loop kolom-kolom tersebut, dan mengeset height sesuai kolom yang paling besar height-nya. Buat dulu HTML seperti berikut:

```
<body>
<div class="col-container">
<aside class="col nav">
Sidebar samping
</aside>
<div class="col main">
Kolom Utama
</div>
<div class="col tambahan">
Konten lainnya
</aside>
</div>
```

Kemudian berikan CSS seperti berikut:

```
<style type="text/css">

.col-container{
color: #fff;
}
.col-container:after{
content: ' ';
clear: both;
display: block;
}
.col{
float: left;
width: 33.3%;
}
.col.nav{
background: #aaa;
}
.col.main{
background: #000;
}
.col.tambahan{
background: #999;
}
</style>
```

Ketika ini sudah dilakukan, kolom terbuat dan terlihat bagus, dengan warna background yang bagus. Hanya saja height dari kolom belum seragam.



Gambar 4.10 Height dari kolom tidak seragam

Untuk membuat seragam, Anda perlu menambahkan kode <script> di body yang fungsinya mengenumerasi semua col dan kemudian membuat tingginya sesuai dengan tinggi yang paling besar.

```
<script>
var samakanKolom = function(){
var kolomnya = document.getElementsByClassName('col'), height
= 0;
//reset semua height
for (var i = 0; i < kolomnya.length; i++) {
kolomnya[i].style.height = "auto";
}
//Loop melalui kolom dan mencari height maksimal
for (var i = 0; i < kolomnya.length; i++) {
if(height < kolomnya[i].clientHeight){
height = kolomnya[i].clientHeight;
}
}
//Menerapkan max height ke semua kolom
for (var i = 0; i < kolomnya.length; i++) {
kolomnya[i].style.height = height + "px";
}
}
//mengeset height kolom
samakanKolom();
//Meng-update height kolom di saat browser resize
window.addEventListener("resize", samakanKolom, true);
</script>
```

Kode lengkapnya menggunakan style plus JavaScript seperti berikut:

```
<!DOCTYPE HTML>
<html lang="en-US">
<head>
<meta name="viewport" content="width=device-width, minimum-scale=1.0, maximum-scale=1.0" />
<title>Demo responsif </title>

<style type="text/css">
.col-container{
color: #fff;
}
.col-container:after{
content: ' ';
clear: both;
display: block;
}
.col{
float: left;
width: 33.3%;
}
.col.nav{
background: #aaa;
}
.col.main{
background: #000;
}
.col.tambahan{
background: #999;
}
</style>

</head>
<body>

<script>
var samakanKolom = function(){
var kolomnya = document.getElementsByClassName('col'), height = 0;
//reset semua height
for (var i = 0; i < kolomnya.length; i++) {
kolomnya[i].style.height = "auto";
}
//Loop melalui kolom dan mencari height maksimal
for (var i = 0; i < kolomnya.length; i++) {
if(height < kolomnya[i].clientHeight){
height = kolomnya[i].clientHeight;
}
}
//Menerapkan max height ke semua kolom
for (var i = 0; i < kolomnya.length; i++) {
kolomnya[i].style.height = height + "px";
}
}
//mengeset height kolom
samakanKolom();
```

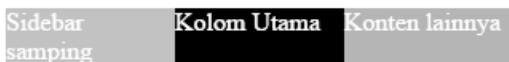
```
//Meng-update height kolom di saat browser resize  
window.addEventListener("resize", samakanKolom, true);  
</script>  
  
<div class="col-container">  
<aside class="col nav">  
Sidebar samping  
</aside>  
<div class="col main">  
Kolom Utama  
</div>  
<div class="col tambahan">  
Konten lainnya  
</aside>  
</div>  
</body>  
</html>
```

Apabila digerakkan, maka tampilan height akan menjadi seragam.



Gambar 4.11 Tampilan height dari tiap kolom menjadi seragam

Kalau misalnya diperbesar lagi, tetapi ukuran height akan seperti ukuran height tertinggi.



Gambar 4.12 Ukuran height menjadi tertinggi

4.2.3 Usahakan Tidak Menggunakan Scroll Bar Horizontal

Ketika membuat site, usahakan tidak menggunakan scroll bar horizontal karena merusakkan user experience. Kalau memang diperlukan, buat scroll bar vertikal saja, perangkat mobile lebih mudah men-scroll ke bawah daripada men-scroll ke samping.

4.2.4 Mempertimbangkan Tampilan Gambar dalam Berbagai Ukuran

Ketika mengimplementasikan gambar menggunakan desain fluid, Anda perlu mempertimbangkan terlebih dahulu bagaimana image akan terlihat di berbagai tampilan yang berbeda. Anda perlu berpikir kalau gambar diatur ulang skalanya, bagaimana tampilannya.

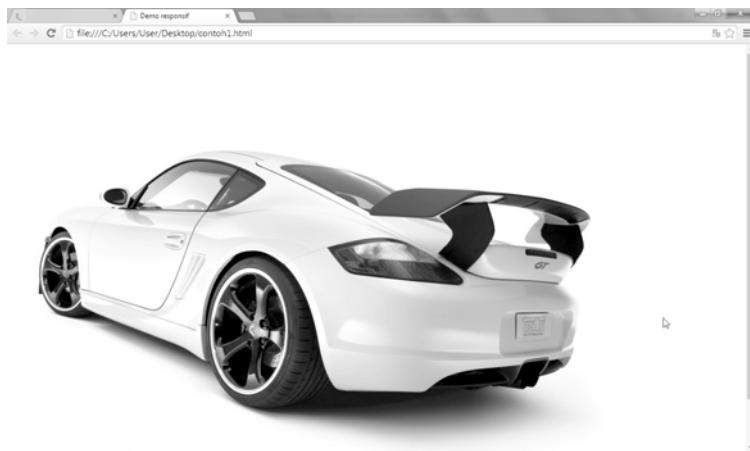
Untuk mengatur ukuran gambar inline, Anda tinggal mengubah width dari gambar. Jika situs Anda responsif, Anda tidak perlu tahu ukuran yang diterapkan ke gambar. Jadi bukannya mendefinisikan width absolut, Anda bisa mengeset properti max-width dengan nilai 100% seperti contoh kode berikut:

```
img {  
    max-width:100%;  
}
```

Dengan kode ini, jika image lebih besar dari kontainer, maka akan bekurang ukurannya, namun jika image lebih kecil dari container ukurannya akan sesuai ukuran.

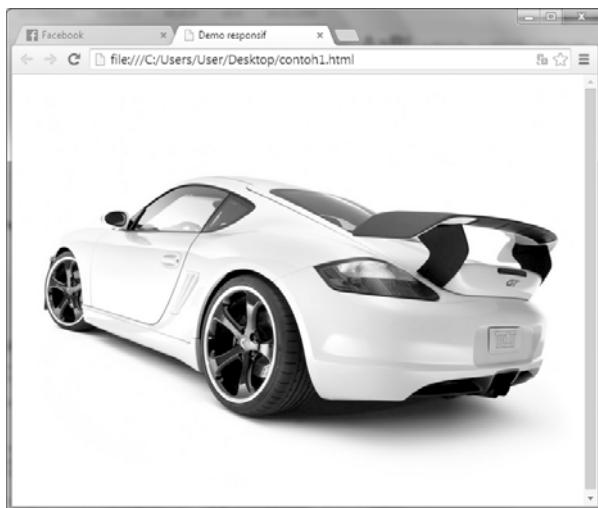
```
<!DOCTYPE HTML>  
<html lang="en-US">  
<head>  
    <title>Demo responsif </title>  
    <style type="text/css">  
  
        img {  
            max-width:100%;  
        }  
  
    </style>  
  
</head>  
  
<body>  
      
</body>  
</html>
```

Apabila viewport lebar, di bagian kanan gambar masih ada ruang kosong.



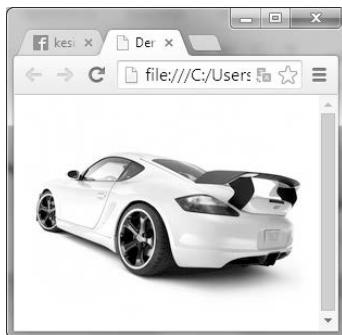
Gambar 4.13 Viewport lebar

Kalau viewport diperkecil, maka tampilan gambar menjadi lebih kecil.



Gambar 4.14 Viewport menjadi lebih kecil, tampilan gambar menjadi 100% width

Kalau viewport diperkecil lagi, tampilan teks menjadi lebih kecil lagi.



Gambar 4.15 Tampilan viewport ketika menjadi lebih kecil lagi

Jika Anda ingin mengatur ukuran gambar background, Anda bisa menggunakan properti background-size. Ini memerlukan tiga nilai yang berbeda: cover, contain, dan nilai width/height.

Kode CSS-nya seperti berikut:

```
.image{  
    background-size:80px 60px;  
}
```

Masalahnya dengan background-size adalah walaupun Anda bisa menggunakan width relatif untuk mengeset width elemen agar fluid, tidak mudah dengan height-nya. Karena itu Anda perlu mendefinisikan <div> yang menjadi kontainer untuk gambar background.

Setelah itu Anda perlu mendefinisikan CSS, pertama definisikan dulu container image. Karena Anda ingin width menjadi relatif, Anda bisa menggunakan persentase di contoh ini adalah width:100%. Height dari gambar perlu proporsional dengan width, ini tidak bisa dilakukan dengan properti height karena height tidak bisa relatif, tapi Anda mempergunakan padding.

Karena persentase padding berdasarkan width dari elemen, Anda bisa menggunakan ini untuk mendefinisikan height-nya. Ini agak kompleks, dengan width dan height yang cocok, Anda tinggal menjadikan gambar sebagai background dan mengatur ukuran background-sie ke width dan height penuh dari elemen.

Contoh kode CSS-nya seperti ini:

```
.image{  
width: 100%;  
position: relative;  
padding-bottom: 125%;  
background: url(gambar.jpg) 0 0 no-repeat;  
background-size: 100% 100%;  
}
```

Dengan contoh ini, Anda bisa melihat gambar di browser dan melihat background akan berubah width-nya sesuai proporsi halaman.

4.2.5 Membuat Wrapping Konten

Ketika membuat site fluid, tidak selamanya harus mengecilkan elemen. Karena pengecilan ada batasnya di mana kalau dikecilkan terus konten elemen bisa tidak terbaca atau terlihat. Begitupula kalau viewport diperbesar, mungkin tidak ideal jika semuanya menjadi lebih besar, karena itu Anda bisa menggunakan content-wrap pada beberapa viewport tertentu.

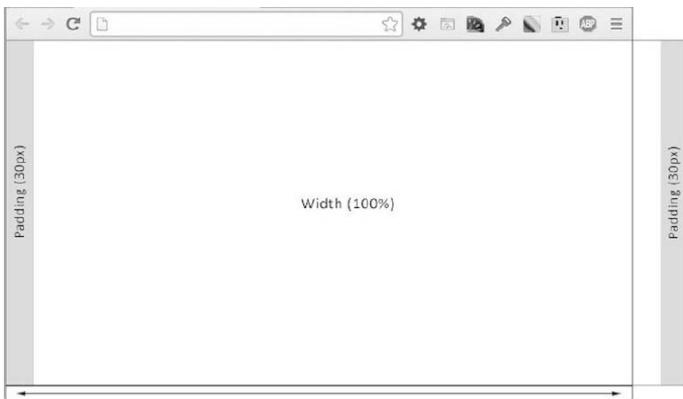
Contohnya sidebar dengan konten di samping website. Jika sidebar mengambil 25% ruang, dan viewport 320px, maka kolom tersebut width-nya hanya 25%.

Jadi nanti Anda bisa meng-query media dan melihat kapan harus mengubah CSS yang ada.

4.3 Memberikan Spasi

Ketika bekerja dengan desain fluid, Anda menggunakan persentase untuk width dari elemen. Keuntungannya adalah browser bisa mengatur width berdasarkan width-nya viewport. Anda juga perlu mempertimbangkan spacing antara elemen.

Idealnya, Anda mendefinisikan width elemen di persentase dan padding elemen di piksel atau ems. Secara default, cara browser menampilkan model box meletakkan padding di luar width, sehingga ketika mendefinisikan piksel untuk padding, fluid dari desain akan pecah ketika scroll bar horizontal muncul.



Gambar 4.16 Padding dan width

Jika Anda sudah memprogram web cukup lama, Anda mungkin mengenal istilah quirks mode. Di browser lama, standar belum diimplementasikan dengan benar, dan mulai IE 6, diputuskan untuk mengimplementasikan standar HTML dan CSS.

Tapi diperlukan fallback yang akan mengakomodasi situs lama yang belum di-update untuk mendukung standar ini, fallback ini disebut quirks mode dan tidak menampilkan doctype di baris pertama HTML.

Di browser lama, model box akan diimplementasikan dengan tidak tepat dibandingkan standar w3C di mana menurut standar, width dan height dari elemen harus didefinisikan, di mana tidak termasuk padding, border, dan marjin. Yang akan diaplikasikan di luar dari width atau height yang ditentukan. Di browser lama, yang diimplementasikan sudah termasuk content, padding, dan border, semua di dalam width atau height.

Implementasi quirk mode ideal yang memungkinkan Anda menambahkan spasi di desain fluid. Tapi Anda tidak perlu memaksakan situs ke quirk mode apabila Anda ingin situs sesuai standar. Dengan adanya CSS3, Anda bisa mengontrol bagaimana model box bekerja agar sesuai keinginan developer.

Properti box-sizing memberikan tiga nilai yang bisa dipilih, dan tiap nilai tersebut menampilkan box dengan berbagai versi model box.

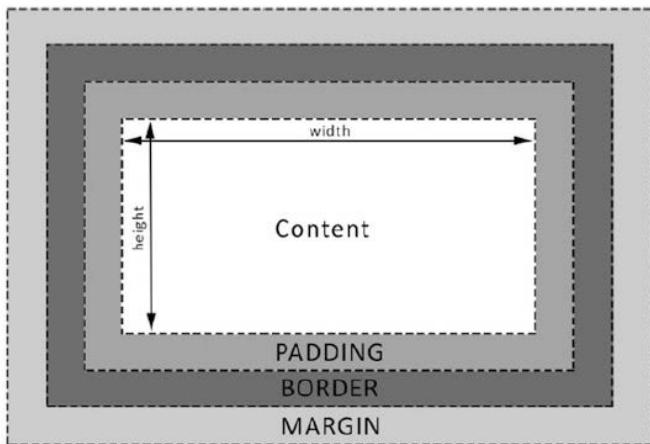
Nilai yang ada adalah:

- content-box
- padding-box
- border-box

4.3.1 Content-box

Style aslinya ditentukan oleh standar CSS. Properti height dan width hanya berupa area konten saja, di mana border, margin, dan padding ditambahkan di luarnya.

Contohnya seperti gambar berikut.

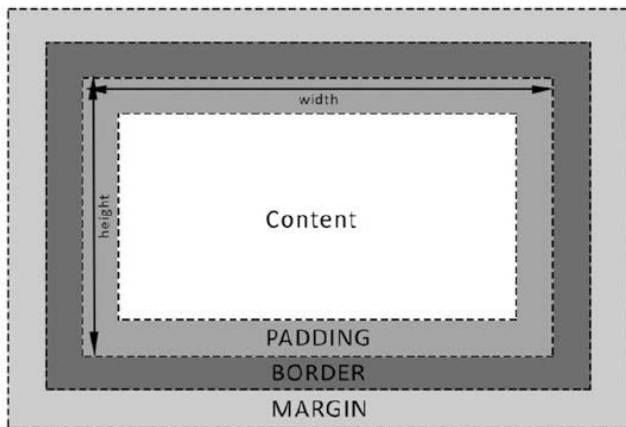


Gambar 4.17 Model box yang diterapkan standar menggunakan properti CSS box-sizing ke content-box

4.3.2 Padding-box

Dengan padding-box, properti width dan height termasuk ukuran padding dan tidak termasuk border atau margin.

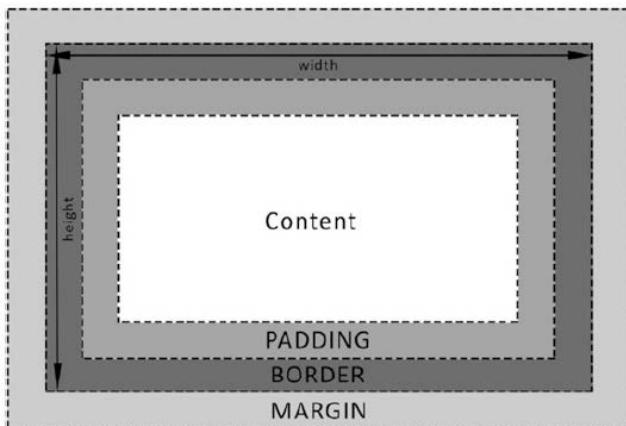
Contohnya seperti berikut.



Gambar 4.18 Pengaturan model box menggunakan properti CSS box-sizing dengan nilai padding-box

4.3.3 Border-box

Properti width dan height termasuk padding dan border, hanya saja tidak termasuk margin. Ini merupakan border-box yang dipakai oleh IE dalam quirks mode.



Gambar 4.19 Pengaturan properti box-sizing ke border-box

Untuk pengembangan situs responsif, idealnya menggunakan border-box untuk menambahkan padding konsisten ke elemen ketika menggunakan width berbasis persentase. Sangat ideal bagi situs responsif untuk menggunakan border-box untuk semua elemen. Tidak hanya untuk konsistensi, tapi Anda juga bisa menggunakan width prosentase untuk semua dari div ke field input. Sama dengan ketika Anda menerapkan border-box ke semua elemen untuk menggunakan selektor universal (*). Berikut ini contoh kode-nya:

```
* {  
-webkit-box-sizing: border-box; /* Safari/Chrome, other WebKit */  
/*  
-moz-box-sizing: border-box; /* Firefox, other Gecko */  
box-sizing: border-box; /* Opera/IE 8+ */  
}
```

4.4 Pengaturan Panjang Teks

Ketika membangun site fluid, sering kali ketika lebar dijadikan full, maka teks menjadi sangat panjang. Ini kurang efisien sebenarnya dan kurang ergonomis. Karena itulah kenapa koran-koran memakai kolom agar lebih mudah dibaca.

Anda bisa menggunakan fasilitas multi kolom dari CSS3 yang sudah dijelaskan sebelumnya untuk memperpendek ukuran teks, sehingga teks tidak terlalu penuh.

Anda juga bisa menggunakan properti max-width dari CSS untuk membatasi ukuran elemen. Anda bisa mengontrol panjang dari baris teks dengan mendefinisikan width maksimum dari elemen kontainer atau paragraf. Ketika mendefinisikan width maksimum, Anda bisa mengatur agar width maksimum tersebut relatif terhadap ukuran font menggunakan em.

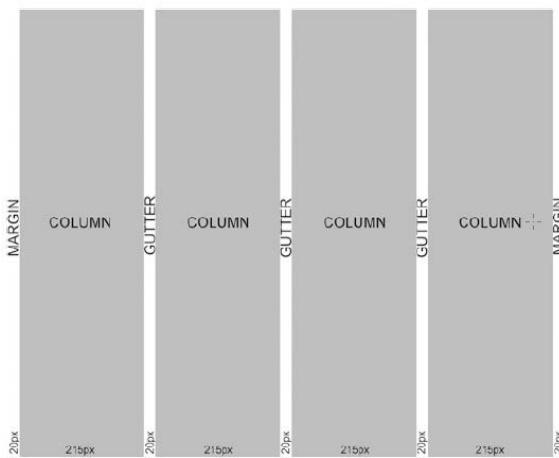
Anda juga harus memastikan bahwa baris teks tidak terlalu pendek. Anda bisa menggunakan properti min-width yang sudah ada sejak CSS2. Properti min-width memungkinkan Anda menentukan width minimum yang bisa dimiliki oleh elemen. Ini akan menimpa nilai dari width dan max-width jika width dari elemen ada di bawah min-width.

Properti min-width tidak perlu digunakan jika elemen tidak terlalu lebar, tapi jika kontainer memiliki width kurang dari elemen, elemen akan melebar di luar batas kontainer.

4.5 Membangun Desain Fluid Menggunakan Grid CSS

Grid awalnya berasal dari dunia percetakan, di mana buku atau majalah memiliki grid-grid untuk meletakkan konten. Sehingga tampilannya simetris sepanjang halaman. Tujuan dari grid ini adalah agar user lebih mudah membaca konten. Ini sama dengan tujuan website responsif.

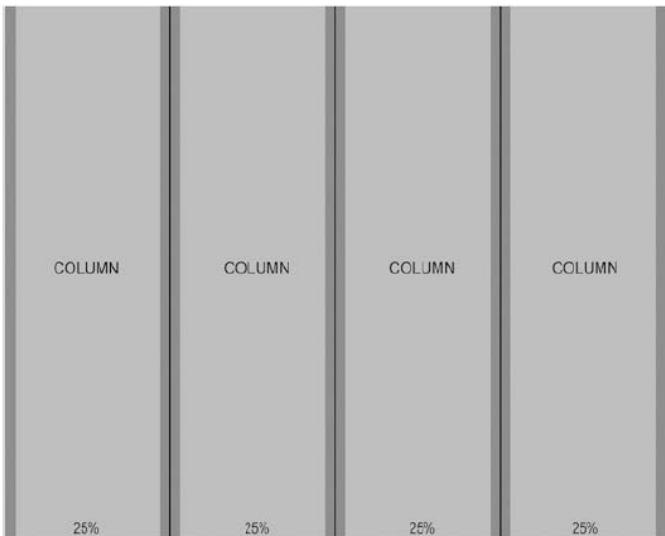
Grid CSS umumnya dibangun dalam bentuk kolom-kolom. Kolom adalah ukuran grid terkecil dan blok bisa terdiri dari satu kolom atau lebih. Sebuah grid bisa memiliki 12 sampai 16 kolom. Di antara kolom-kolom tersebut ada gutter. Gutter adalah spasi antara tiap kolom. Contohnya, di gambar berikut ada grid dengan layout fixed yang memiliki kolom 215px dan gutter 20px di tiap sisi kolom.



Gambar 4.20 Kolom fixed-width dengan gutter di antaranya

Layout fixed-width memungkinkan Anda melihat bagaimana tampilan grid. Tapi kalau mau grid yang fluid, Anda harus mendefinisikan kolom 25% dan untuk mengaktifkan gutter, Anda bisa memakai properti box-sizing untuk memindahkan padding di dalam kolom.

Misalnya untuk total 960px, tiap kolom adalah 240px lebarnya. Tapi jika Anda ingin dikecilkan ukuran viewport sampai 320px, kolom akan otomatis dinamis dan responsif di mana lebarnya menjadi 80px. Contoh grid yang fluid seperti berikut.



Gambar 4.21 Contoh grid yang fluid

4.6 Membangun Grid

Walaupun Anda bisa menggunakan grid dengan framework CSS, tapi Anda perlu terlebih dahulu mengetahui cara kerja pembuatan grid sendiri, sehingga lebih memahami prinsip grid responsif. Dalam bentuk yang sederhana, grid CSS terdiri dari beberapa hal berikut:

- Kolom
- Gutter
- Row
- Wrapper

Akan dijelaskan dengan lebih detail masing-masing teks komponen di atas.

4.6.1 Kolom

Ketika membangun grid CSS, Anda perlu memutuskan jumlah kolom atau sistem grid yang akan digunakan. Lazim bagi sistem grid untuk memiliki kolom sejumlah 12 sampai 16 biji.

Kolom di grid CSS, umumnya dibuat dengan width yang sama, dan diletakkan bersisian satu dengan lainnya. Tipe ini biasanya adalah tipe kolom float.

Berikut ini contoh CSS yang bisa digunakan:

```
.col {  
    float: left;  
    width: 25%;  
}
```

Untuk contoh grid CSS yang akan ditulis, misalnya akan menggunakan empat kolom. Agar bisa menggunakan satu selektor untuk memilih semua kolom, Anda bisa menggunakan kata awal yang sama, dengan nama col-span-x di mana x adalah nomor kolom. Ini nanti mudah untuk memilihnya menggunakan selektor, seperti telah dicontohkan di bab sebelumnya.

Berikut ini contoh CSS yang bisa dipakai:

```
[class*="col-span"]{  
    float: left;  
}  
.col-span-1{  
    width: 25%;  
}  
.col-span-2{  
    width: 50%;  
}  
.col-span-3{  
    width: 75%;  
}  
.col-span-4{  
    width: 100%;  
}
```

Anda bisa memperhatikan bahwa ada karakter selektor wildcard yaitu [class*="col-span"]. Ini memungkinkan Anda menargetkan semua kolom tanpa harus menambahkan class tambahan. Di contoh di atas, selektor tersebut dipakai sebagai pengganti untuk menuliskan:

```
.col-span-1, .col-span-2, .col-span-3, .col-span-4{  
    float: left;  
}
```

Sampai di sini, kita sudah membuat empat kolom, tapi Anda bisa menyesuaikan situs terhadap viewport. Ketika membahas prinsip membangun desain responsif yang fluid, Anda hendaknya mengoptimalkan dengan mencocokkannya dengan viewport. Kalau viewport mengecil, maka jumlah kolom diperkecil. Anda bisa menambahkan dengan query media.

Contohnya seperti ini:

```
[class*="col-span"]{  
float: left;  
}  
.sm-col-span-1{  
width: 25%;  
}  
.sm-col-span-2{  
width: 50%;  
}  
.sm-col-span-3{  
width: 75%;  
}  
.sm-col-span-4{  
width: 100%;  
}  
 @media screen and (min-width: 768px){  
.lg-col-span-1{  
width: 25%;  
}  
.lg-col-span-2{  
width: 50%;  
}  
.lg-col-span-3{  
width: 75%;  
}  
.lg-col-span-4{  
width: 100%;  
}  
}
```

Dari contoh kode di atas, Anda bisa melihat bahwa membangun grid CSS di desain web fluid yang responsif sangat mudah. Akan jauh lebih mudah lagi kalau menggunakan framework yang banyak tersedia di internet. Tapi paling tidak Anda memahami prinsip dasar cara kerjanya.

4.6.2 Gutter

Gutter dari grid CSS adalah spasi dari tiap kolom. Awalnya ini dibuat menggunakan margin. Namun kalau bekerja dengan desain fluid, Anda menggunakan persentase untuk width kolom. Anda bisa menggunakan padding untuk gutter antara kolom.

Kerugian menggunakan padding untuk gutter adalah warna background tidak bisa diberikan untuk kolom, karena kalau warna background ada di kolom, background tersebut muncul juga di gutter. Pencegahannya adalah menambahkan warna background ke elemen di dalam kolom, dan bukan ke kolom itu sendiri.

Sebelumnya, sudah dijelaskan tentang properti box-sizing yang memungkinkan Anda melihat bagaimana model box bisa dipakai untuk menentukan padding di dalam atau di luar width. Untuk grid ini, disarankan menggunakan box-sizing diset ke borderbox. Jika Anda sudah menerapkan ini ke selektor universal, maka Anda tidak perlu mengeset box-sizing lagi.

Contoh kodennya seperti ini:

```
[class*="col-span"] {  
  box-sizing: border-box;  
  padding-left: 15px;  
  padding-right: 15px;  
}
```

Seperti Anda lihat, Anda juga bisa menambahkan padding ke kolom, Anda sudah menambahkan padding menggunakan properti padding-left, dan padding-right karena ini akan mencegah Anda untuk melakukan penimpaan setting terhadap semua padding di atas atau bawah elemen yang mungkin sudah ditentukan di CSS.

4.6.3 Row

Row digunakan untuk menampung kolom, karena kolom float ke kiri, maka akan lepas dari halaman.

Berikut ini contoh kode CSS untuk row:

```
.row:after{  
  content: ' ';  
  clear: both;  
  display: block;  
}
```

Di contoh di atas, ada elemen semu yang merupakan elemen block-level dengan clear:both.

4.6.4 Wrapper

Berikutnya, Anda menambahkan wrapper. Ini digunakan untuk menambahkan padding tambahan ke sebelah kiri dan kanan situs.

Contoh kode CSS-nya seperti ini:

```
.wrapper{  
    padding-left:15px  
    padding-right:15px  
}
```

4.7 Membuat Layout Responsif

Berikut ini tutorial bagaimana membuat layout responsif.

1. Pertama Anda membuat dulu halaman HTML seperti berikut:

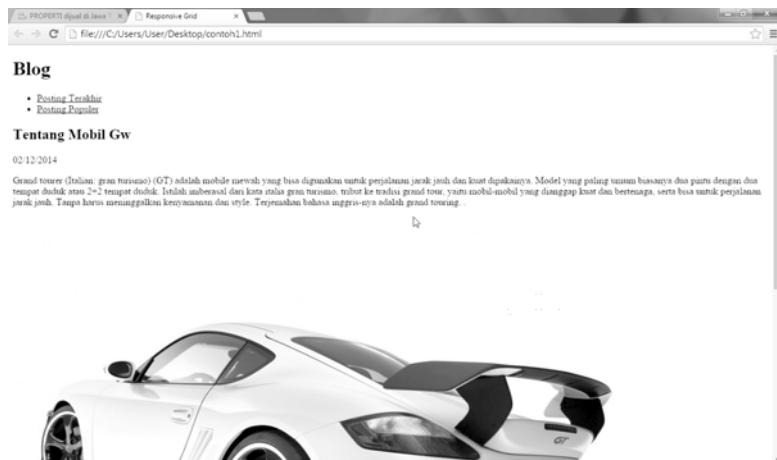
```
<!DOCTYPE html>  
<html>  
<head>  
<title>Responsive Grid</title>  
<meta name="viewport" content="width=device-width">  
<link rel="stylesheet" type="text/css" href="grid-  
responsif.css">  
</head>  
<body>  
<header>  
<div class="wrapper">  
<div class="row">  
<div class="sm-col-span-2 lg-col-span-4">  
<h1>Blog</h1>  
</div>  
<nav class="sm-col-span-2 lg-col-span-4">  
<ul>  
<li><a href="#">Posting Terakhir</a></li>  
<li><a href="#">Posting Populer</a></li>  
</ul>  
</nav>  
</div>  
</div>  
</header>  
<div class="content" role="main">  
<div class="wrapper">  
<div class="row">  
<article class="sm-col-span-4 lg-col-span-3">  
<h2>Tentang Mobil Gw</h2>  
<p>02/12/2014</p>  
<p>Grand tourer (Italian: gran turismo) (GT) adalah mobil  
mewah yang bisa  
digunakan untuk perjalanan jarak jauh dan kuat dipakainya.  
Model yang paling umum biasanya dua pintu dengan dua tempat  
duduk atau 2+2 tempat duduk.  
Istilah ini berasal dari kata italia gran turismo, tribut ke  
tradisi grand tour, yaitu  
mobil-mobil yang dianggap kuat dan bertenaga, serta bisa  
untuk perjalanan jarak jauh.
```

Tanpa harus meninggalkan kenyamanan dan style. Terjemahan bahasa inggris-nya adalah grand touring.

```
.</p>
</article>

<aside class="sm-col-span-4 lg-col-span-1">
<h2>Artikel terkait</h2>
<nav>
<ul>
<li><a href="#">Link ke-1 </a></li>
<li><a href="#">Link ke-2</a></li>
<li><a href="#">Link ke-3</a></li>
</ul>
</nav>
</aside>
</div>
</div>
</div>
</body>
</html>
```

- Apabila ditampilkan masih terlihat standar, belum ada efek responsif.



Gambar 4.22 Tampilan awal setelah kode HTML dibuat

- Setelah diperkecil, tetap gambar tidak berubah, ini karena belum responsif.



Gambar 4.23 Tampilan gambar mobil tidak mengecil karena belum ada CSS untuk efek responsif

4. Buat file CSS bernama grid-responsif.css dan isikan kode berikut:

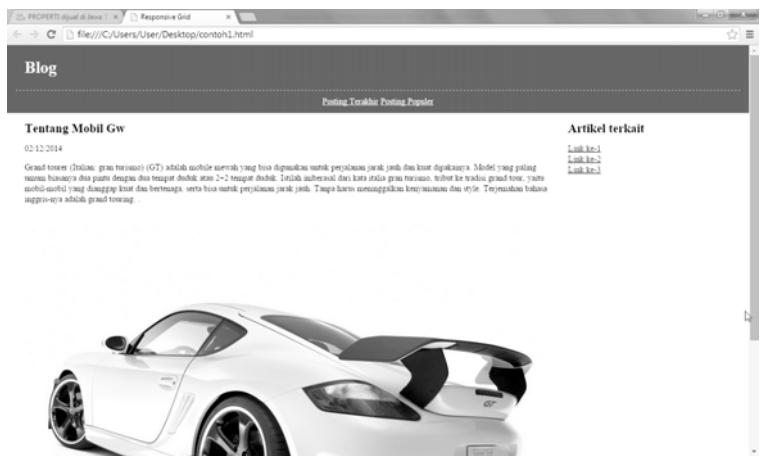
```
/*Style grid*/
img {
    max-width:100%;
}

.wrapper{
padding-left: 15px;
padding-right: 15px;
}
.row:after{
content: '';
clear: both;
display: block;
}
[class*="col-span"] {
float: left;
box-sizing: border-box;
padding-left: 15px;
padding-right: 15px;
}
.sm-col-span-1{
width: 25%;
}
```

```
.sm-col-span-2{
width: 50%;
}
.sm-col-span-3{
width: 75%;

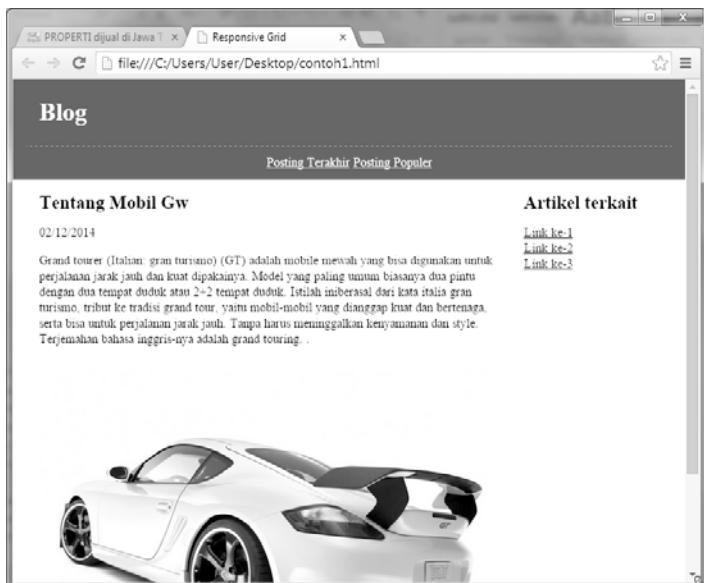
}
.sm-col-span-4{
width: 100%;
}
@media screen and (min-width: 768px){
.lg-col-span-1{
width: 25%;
}
.lg-col-span-2{
width: 50%;
}
.lg-col-span-3{
width: 75%;
}
.lg-col-span-4{
width: 100%;
}
}
/*Spesifik ke situs*/
body{
margin: 0;
padding: 0;
font-size: 14px;
line-height: 18px;
}
header{
background: #304480;
padding-top: 10px;
padding-bottom: 10px;
color: #fff;
}
ul{
padding: 0px;
margin: 0px;
}
ul li{
list-style: none;
}
header nav{
text-align: right;
}
header nav a{
color: #fff;
}
@media screen and (min-width: 768px){
header nav{
text-align: center;
padding-top: 10px;
border-top: 1px dashed #ccc;
margin-top: 10px;
}
header nav li{
display: inline-block;
}
}
```

5. Apabila ditampilkan, terlihat tampilan seperti berikut.



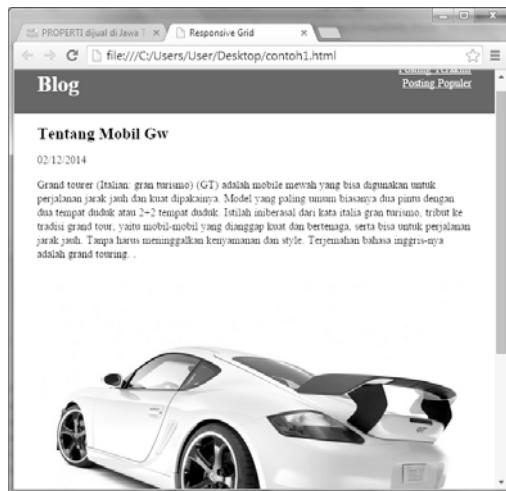
Gambar 4.24 Tampilan website sudah responsif

6. Ketika diperkecil, terlihat posisi akan me-respon viewport yang mengecil.



Gambar 4.25 Tampilan merespon viewport yang mengecil

7. Ketika viewport mengecil lagi, menu di sebelah kanan akan berpindah ke bawah.



Gambar 4.26 Menu di viewport kanan berpindah ke bawah

8. Apabila di-scroll ke bawah, di bawah gambar mobil yang mengecil terlihat ada menu.



Gambar 4.27 Tampilan menu kanan beralih ke bawah

9. Ini menunjukkan desain responsif sudah berjalan dengan sempurna.



Gambar 4.28 Desain responsif berjalan dengan baik

10. Di contoh di atas, jumlah kolom hanya empat. Tapi umumnya grid bisa berjumlah antara 12 sampai 16 kolom. Anda bisa menentukan width dari kolom dengan rumus:

$$\text{Width_Kolom} = (\text{100}/\text{Jumlah_Kolom}) * \text{span_kolom}$$

4.8 Framework di Desain Responsif

Di contoh di atas, Anda membuat style responsif secara manual. Tapi untuk mempercepat proses development web, kadang lebih mudah dan cepat menggunakan framwork untuk menyeting situs Anda agar responsif.

Di internet, saat ini banyak sekali framework untuk membantu proses desain responsif.

4.8.1 Sistem Grid

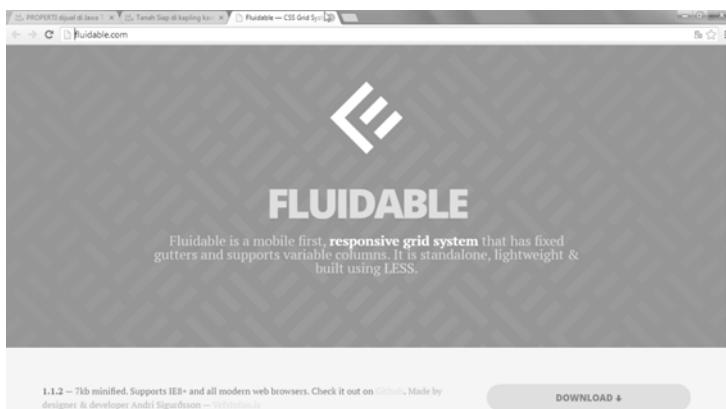
Sistem grid adalah struktur di mana Anda bisa membangun layout dari website. Ini terdiri dari baris horizontal dan kolom vertikal. Ini akan menampung konten dari website yang bisa menjangkau satu kolom atau lebih. Sebuah sistem grid bisa memberikan kemudahan ketika membuat website. Karena website Anda nantinya lebih konsisten spasinya maupun jaraknya.

Saat melihat sistem grid open source, umumnya fitur sederhana dan hanya berisi style yang diperlukan untuk menampilkan grid. Sehingga semua aspek pembangunan situs lainnya bisa bebas sesuai keinginan developer. Sebagian juga memiliki fasilitas media query untuk merespon width dari viewport browser.

Berikut ini tiga contoh grid CSS open source untuk membantu Anda membangun website responsif untuk mobile ataupun tidak.

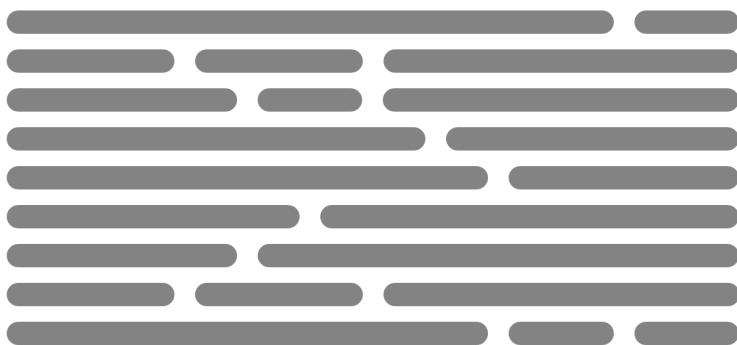
- Fluidable (<http://fluidable.com/>).
- CSS Smart Grid (<http://dryan.github.io/css-smart-grid/>).
- Cswizardry-grids (<http://csswizardry.com/csswizardry-grids/>).

Yang pertama, Fluidable adalah sistem grid mobile-first yang dikembangkan oleh Andri Sigursson yang memiliki gutter fix dan width kolom bisa berubah. Tampilan home-page situs fluidable.com seperti berikut.



Gambar 4.29 Tampilan situs fluidable.com

Tampilan grid-nya terlihat seperti contoh berikut.



Gambar 4.30 Tampilan grid fluidable

Anda bisa melihat berapa banyak kolom sebuah elemen menjangkau untuk tiap perangkat di Fluidable. Artinya dengan teknologi ini, Anda bisa mengontrol bagaimana situs ditampilkan ke tipe perangkat tertentu. Ini berkat penamaan dari nama class elemen.

- .col-mb-x: menentukan berapa jumlah kolom yang dijangkau di mobile.
- .col-x: menentukan berapa kolom yang dijangkau di tablet.
- .col.dt.x: menentukan berapa kolom yang harus dijangkau di desktop.

Untuk tiap class elemen di atas, nilai x adalah nilai kolom yang harus dijangkau. Ditinjau dari sisi teknologi, Fluidable dapat dikonfigurasi melalui lima variabel, yaitu @columns, @maxWidth, @gutterWidth, @screenTablet, dan @screenDesktop. Anda bisa mengonfigurasi kelima itu untuk lebih fleksibel.

Yang kedua adalah CSS Smart Grid. Ini dikembangkan oleh Daniel Ryan dan dapat diakses dari dryan.github.io/css-smart-grid. Awalnya dirilis pada tahun 2011.



About CSS Smart Grid

Performance Driven

Built to Scale

Mobile Friendly & Mobile First

SASS Support

Usage

As a Grid System

As Mixins

[View the Demo](#)

What does it do?

CSS Smart Grid builds on the popular 960 grid concept, mixing in responsive grids that are "mobile first". It is built on four concepts:

1. Performance Driven

CSS Smart Grid was engineered to be lightweight. Minified and zipped, it weighs in at only 0.65K.

It does not require the use of polyfills to work in IE 7-8. Instead, CSS Smart Grid can use a class of "oldie" or "IE-9" set via Paul Irish's conditional <html> syntax (as seen in HTML5 Boilerplate) to give IE 8 the default 960 pixel version of the grid.

IE 7 and lower are not officially supported. In version 3+, IE support is not included by default. If you wish to use it, uncomment the final section of smart-grid.scss and recompile.

2. Built to Scale

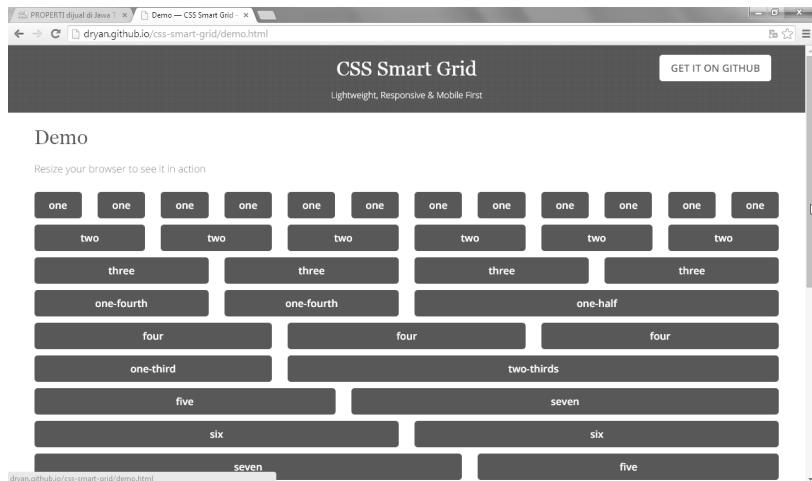
CSS Smart Grid version 3 was completely rebuilt from the ground up to use percentage-based flexible columns and gutters. When the browser is at least 960 pixels wide, columns will be 60 pixels wide with 20 pixel gutters. Browsers from 481-959 pixels wide will scale to fit while smaller screens get full width columns. If you want to support larger layouts, its as simple as changing one variable.

This page uses CSS Smart Grid, resize your browser to see it in action.

3. Not Just Mobile Friendly, But Mobile First

Gambar 4.31 Halaman awal CSS Smart Grid

Tampilan grid CSS bisa dilihat di bagian Demo.



Gambar 4.32 Tampilan grid CSS

Kolom di CSS Smart grid didefinisikan dengan menambahkan class "columns" dengan class tambahan untuk menentukan jumlah kolom yang harus dijangkau, misalnya satu, dua, atau tiga.

Anda bisa mengisikan maksimal 12 kolom, Anda juga bisa menggunakan kata kunci untuk menentukan selebar apa elemennya, yaitu dengan mengisikan:

- one-fourth: sama dengan 25% atau lebar 3 kolom.
- one-half: sama dengan 50% atau lebar 3 kolom.
- one-third: sama dengan 33,33% atau lebar 4 kolom.
- two-thirds: sama dengan 66,66% atau sama dengan lebar 8 kolom.

Contoh kode berikut menggunakan grid. Anda bisa melihat bahwa sebuah row didefinisikan bersamaan dengan dua kolom. Tiap-tiap row tersebut dengan width yang didefinisikan menggunakan kata-kata di atas:

```
<div class="row">
<div class="columns two-thirds">
<p>Lorem ipsum</p>
</div>
<div class="columns one-third">
<p>Lorem ipsum</p>
</div>
</div>
```

Dengan menggunakan keyword ini, Anda bisa mengatur label kolom dengan fleksibel. Di CSS, caranya adalah dengan menggunakan nama class kolom numerik. Tapi ada kekurangannya juga, jika Anda ingin mengaitkan elemen agar bekerja untuk width tertentu, Anda harus mengaturnya di CSS.

Yang ketiga adalah `csswizardry-grids`, ini dikembangkan oleh Harry Roberts dengan alamat di csswizardry.com/csswizardry-grids. Ini merupakan grid yang mobile-first yang dapat di-download dari github.



Gambar 4.33 Halaman utama situs csswizardry-grids

Grid di sini dibangun sangat fleksibel, lebih baik dibandingkan dua framework sebelumnya. Karena Anda bisa menentukan apakah kolom harus center atau tanpa gutter. Artinya, Anda tidak perlu meng-offset kolom untuk membuatnya center. Anda bisa membuat kolom menjadi center dengan menggunakan class. Grid juga memiliki fleksibilitas tambahan yang memungkinkan Anda menyusurkan kolom di dalam kolom lainnya.

Sama seperti CSS Smart Grid, sistem grid di sini menggunakan konvensi penamaan class yang mengacu ke width kolom.

Contohnya:

- one-half
- one-third
- one-quarter, two-quarter, three-quarter

Penamaan class ini memudahkan dibandingkan teknik span-x yang digunakan oleh beberapa sistem grid. CSS Smart Grid dan CSSWlazdry-grids memungkinkan Anda mengonfigurasikan bersama menggunakan SASS.

Salah satu kekurangan menggunakan csswizardry-grids adalah menggunakan block inline dan bukan float untuk meletakkan kolom satu dengan lainnya. Sehingga, jika ada spasi antara kolom di HTML, maka akan muncul spacing yang tidak diinginkan di halaman. Ada dua cara untuk mencegah hal ini, yaitu dengan meletakkan komentar HTML di antara elemen.

Contohnya seperti berikut:

```
<div class="one-half">  
lorem ipsum  
</div><!--  
--><div class="one-half">  
lorem ipsum  
</div>
```

Atau bisa juga dengan menghapus spasi antara kolom, contohnya seperti berikut:

```
<div class="one-half">  
lorem ipsum  
</div><div class="one-half">  
lorem ipsum  
</div>
```

Walaupun menggunakan blok inline, ini juga memiliki kekurangan, tapi ada juga kelebihan menggunakan blok inline di samping float. Ini memungkinkan Anda membalik urutan order dari konten menggunakan atribut `dir` HTML untuk menargetkan area tertentu dari situs, atau ditambahkan di elemen HTML untuk menargetkan halaman penuh.

Atribut `dir` HTML memiliki dua alternatif nilai, yaitu `ltr` untuk *left to right*, dan `rtl` untuk *right to left*.

4.9 Framework CSS

Selain framework grid CSS open source, ada juga banyak framework open source. Framework CSS adalah ekstensi dari sistem grid CSS standar, tapi tidak hanya menyediakan layout grid saja, tapi juga menyediakan set browser standar, tipografi, dan elemen user interface yang bisa digunakan untuk membangun situs responsif.

Ada beberapa framework CSS yang tersedia, namun ada dua yang terkenal di komunitas web developer, yaitu Twitter Bootstrap dan Zurb Foundation.

Beberapa elemen UI yang didukung di framework CSS antara lain:

- Dropdown
- Button dan button group
- Form
- Navbar
- Breadcrumb
- Pagination
- Label
- Badge
- Page header
- Thumbnail
- Alert
- Progress bar
- Media object
- List group

Walaupun menyediakan banyak komponen, framework CSS jika diseriusi akan memudahkan proses pengembangan karena sudah ada library bawaan yang tidak perlu membuat Anda mengetikkan dari nol lagi. Anda bisa melihat detil penggunaan tiap framework di situs masing-masing penyedia framework yang bersangkutan.

Tentang Penulis

Edy Winarno ST, M.Eng

Penulis adalah staf pengajar di Universitas Stikubank Semarang. Pendidikan terakhirnya adalah Master of Engineering (M.Eng.) Computer & Informatics System di Teknik Elektro Universitas Gadjah Mada (UGM). Selain mengajar, penulis juga berkecimpung di dunia blogging dan sedikit melakukan entrepreneurship.

Ali Zaki

Penulis adalah seorang pekerja TI di bidang pemrograman, multimedia, dan web. Saat ini menjadi outsourcee untuk sebuah perusahaan konsultan TI di Jakarta. Penulis bekerja secara remote melalui SOHO di rumahnya di Semarang. Di sela-sela aktivitasnya, penulis juga aktif melakukan penulisan buku di komunitas SmitDev. Beberapa karya buku penulis telah diterbitkan oleh Elex Media Komputindo.

SmitDev Community

SmitDev merupakan lembaga komunitas yang menjadi wadah bagi para praktisi teknologi informasi dan komputer. Salah satu aktivitas SmitDev saat ini adalah melakukan penulisan buku komputer dan mempersembahkan karya buku bagi kalangan pengguna dan peminat bidang teknologi informasi dan komunikasi di tanah air. Pembaca dapat menyampaikan saran, tanggapan, dan pertanyaan melalui layanan forum interaksi pembaca yang tersedia di <http://www.smitdev.com>, atau melalui email info@smitdev.com.

Catatan:

- Untuk melakukan pemesanan buku, hubungi Layanan Langsung PT Elex Media Komputindo:

Gramedia Direct

Jl. Palmerah Barat No. 33, Jakarta 10270

Telemarketing/CS: 021-53650110/111 ext: 3901/3902

Email: endang@gramediapublishers.com

Desain Web Responsif dengan HTML5 dan CSS3



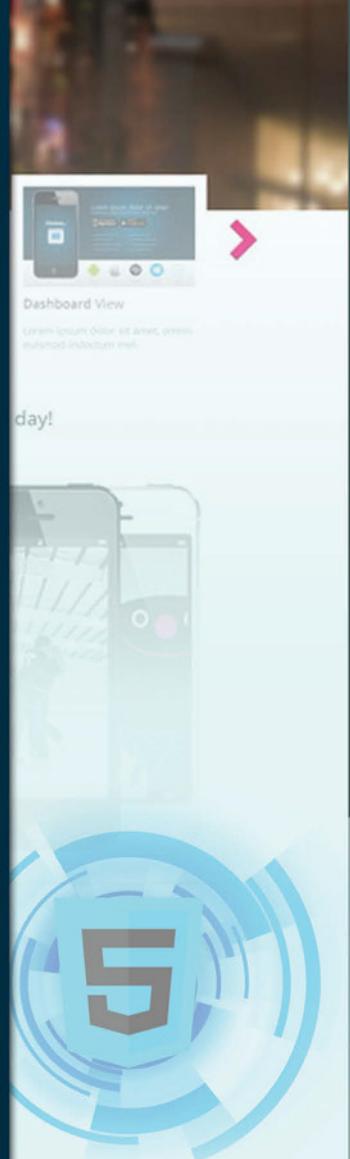
Sekarang ini, sebuah website lazim diakses dari berbagai platform, mulai dari komputer desktop sampai peranti mobile. Desainer web dan para pengelola website seharusnya dapat melihat tren ini. Website modern harus dapat mengakomodasi kebutuhan internet untuk pengguna smartphone.

Buku ini membahas tren baru pengembangan web, yaitu Responsive Web Design (RWD) dengan menggunakan tool HTML5 dan CSS3. Melalui teknik desain web responsif, secara cepat dan praktis Anda dapat menyulap website untuk dapat diakses dari berbagai platform peranti, tanpa harus mengembangkan aplikasi-aplikasi tersendiri untuk tiap jenis platform peranti.

Pembahasan selengkapnya meliputi:

- Sekilas tentang desain responsif.
- Fitur terbaru dalam HTML 5 dan CSS 3.
- Pengenalan media query.
- Desain responsif dengan layout fluid.

PT ELEX MEDIA KOMPUTINDO
Kompas Gramedia Building
Jl. Palmerah Barat 29-37, Jakarta 10270
Telp. (021) 53650110-53650111, Ext 3214
Webpage: <http://elexmedia.co.id>



| |
|--|
| Kelompok |
| INTERNET |
| Keterampilan |
| <input checked="" type="checkbox"/> Tingkat Pemula |
| <input checked="" type="checkbox"/> Tingkat Menengah |
| <input type="checkbox"/> Tingkat Mahir |
| Jenis Buku |
| <input checked="" type="checkbox"/> Referensi |
| <input checked="" type="checkbox"/> Tutorial |
| <input type="checkbox"/> Latihan http://pustaka-indo.blogspot.com |

gramedia
ISBN 978-602-02-5825-6



9 78602 0258256

121150271

<http://pustaka-indo.blogspot.com>