



Belajar  
**PHP**

# Pemrograman Web Dasar

Mempelajari membuat halaman web dengan HTML, memberikan style dan menata layout dengan CSS serta pemrograman client-side dengan JavaScript dan server-side dengan PHP.

M Reza Faisal, Friska Abadi



**Belajar**  
**PHP**  
**Pemrograman Web Dasar**

**M. Reza Faisal**  
**Friska Abadi**



*Belajar PHP: Pemrograman Web Dasar*

© 2020 M Reza Faisal, Friska Abadi

207 halaman, 16,5 x 23 cm

Katalog dalam Terbitan (KDT)

Perpustakaan Nasional Republik Indonesia

ISBN: -

Hak cipta dilindungi undang-undang. Dilarang memperbanyak sebagian atau seluruh isi buku ini dalam bentuk apa pun, baik secara elektronik, termasuk memfotokopi, merekam, atau dengan menggunakan sistem penyimpanan lainnya, tanpa izin tertulis dari penulis atau penerbit.

Cetakan I: Februari 2020

Penerbit

Scripta Cendekia

Banjarbaru, Kalimantan Selatan, Indonesia

E: [scriptacendekia@gmail.com](mailto:scriptacendekia@gmail.com)

# ***Daftar Isi***

<i>Daftar Isi .....</i>	<i>ii</i>
<i>Daftar Gambar .....</i>	<i>vi</i>
<b>1 Pendahuluan .....</b>	<b>1</b>
Internet .....	1
Sejarah Internet .....	1
Hypertext Transfer Protocol (HTTP) .....	2
Web Server .....	3
Web Browser .....	4
Aplikasi Web .....	4
Aplikasi Desktop & Aplikasi Web .....	4
Teknologi .....	5
Code Editor .....	5
Source Code .....	6
<b>2 Visual Studio Code .....</b>	<b>7</b>
Installasi .....	7
Antarmuka .....	8
Tool Tambahan .....	9
Extension .....	9
Integrated Terminal .....	11
<b>3 HyperText Markup Language .....</b>	<b>12</b>
Pendahuluan .....	12
Sejarah .....	12
Sintaks Tag .....	13
Atribut .....	16
Aturan Penamaan File & Folder .....	16
Halaman Web Sederhana .....	17
Memformat Teks .....	21

Heading.....	21
Paragraf.....	25
Italic, Bold & Underline .....	30
List .....	33
Gambar.....	38
Hyperlink .....	43
Table .. .	50
Form Input Data.....	60
Form .....	61
Textbox.....	64
Tombol .....	65
Label .....	70
Radio Button.....	72
Checkbox .....	75
Textarea.....	78
Drop-down List.....	79
Upload File .....	82
Project .....	83
Web Profile Pribadi .....	83
Form & Daftar Buku Tamu .....	84
<b>4 Cascading Style Sheets .....</b>	<b>86</b>
Pendahuluan.....	86
Sintaks .....	86
Tata Cara Penggunaan.....	87
Dasar-Dasar CSS .....	98
Warna .....	98
Teks.....	101
Background Image .....	103
Kotak .....	107

Tata Letak dengan CSS .....	108
Template Utama .....	108
Form Registrasi.....	129
CSS Library & Framework .....	138
Project.....	138
Web Profile Pribadi dengan CSS.....	138
Web Personal dengan Bootstrap .....	139
<b>5 JavaScript.....</b>	<b>140</b>
Pendahuluan .....	140
Aturan Penulisan .....	140
Internal.....	141
External.....	142
Dasar-Dasar Pemrograman.....	143
Komentar .....	143
Variable.....	144
Operasi Aritmatika.....	144
Percabangan .....	145
Perulangan .....	149
Method.....	151
Event .....	151
Document Object Model .....	153
Akses Elemen HTML .....	154
Modifikasi Elemen HTML .....	159
Menambah Elemen .....	162
Menghapus Element .....	164
Akses Elemen dengan Selector .....	164
Project .....	166
Validasi Form.....	166
<b>6 PHP: Hypertext Preprocessor.....</b>	<b>168</b>

Dasar-Dasar PHP .....	168
Form & File .....	168
Operasi Database .....	168
Otentikasi Pengguna .....	168
Object Oriented Programming.....	168
Reporting & Printing .....	168
<i>Tentang Penulis .....</i>	<i>169</i>

## ***Daftar Gambar***

Gambar 1. Aturan model client-server. ....	3
Gambar 2. Halaman download Visual Studio Code.....	7
Gambar 3. Layout dasar antarmuka Visual Studio Code.....	8
Gambar 4. Daftar extension. ....	9
Gambar 5. Pencarian extension. ....	10
Gambar 6. Extension berhasil diinstall. ....	10
Gambar 7. Integrated terminal. ....	11
Gambar 8. Halaman web sederhana. ....	18
Gambar 9. Halaman Web Sederhana - Latihan 1. ....	18
Gambar 10. Halaman web sederhana - Latihan 2.....	20
Gambar 11. Halaman web sederhana - Latihan 3.....	20
Gambar 12. Style heading pada MS Word. ....	21
Gambar 13. Memformat teks - Heading. ....	22
Gambar 14. Memformat teks - Heading - Latihan 1.....	23
Gambar 15. Memformat teks - Heading - Latihan 2.....	24
Gambar 16. Memformat teks - Paragraf.....	26
Gambar 17. Memformat teks - Paragraf - Latihan 1.....	27
Gambar 18. Memformat teks - Paragraf - Latihan 2.....	29
Gambar 19. Memformat teks - Italic, bold & underline - Latihan 1.....	31
Gambar 20. Memformat teks - Italic, bold & underline - Latihan 2.....	32
Gambar 21. Memformat teks - List. ....	33
Gambar 22. List di dalam list.....	36
Gambar 23. Paragraf di dalam list. ....	37
Gambar 24. Lokasi folder dan file HTML.....	39
Gambar 25. Menambahkan gambar pada halaman web.....	40
Gambar 26. Halaman web untuk menampilkan gallery foto. ....	42
Gambar 27. Membuat hyperlink untuk menghubungkan dua halaman web.	44

Gambar 28. Hyperlink ke alamat web.....	46
Gambar 29. Hyperlink di dalam halaman web.....	48
Gambar 30. Tabel daftar buku.....	54
Gambar 31. Form input untuk halaman Contact Us. ....	61
Gambar 32. Form input data - Form - Contoh form sederhana.....	61
Gambar 33. Form input data - form - isi nilai kolom input .....	63
Gambar 34. Form input data - Textbox - Contoh 1.....	65
Gambar 35. Form input data - Tombol - Contoh 1. ....	67
Gambar 36. Form input data - Tombol - Latihan 1.....	68
Gambar 37. Form input data - Tombol - Registrasi form.....	69
Gambar 38. Form input data - Label - Contoh label pada form.....	70
Gambar 39. Form input data - Radio button - Latihan 1. ....	75
Gambar 40. Form input data - Checkbox - Contoh. ....	76
Gambar 41. Form input data - Checkbox - Form login. ....	77
Gambar 42. Form input data - Textarea - Form isian email. ....	78
Gambar 43. Form input data - Drop-down list - Contoh. ....	80
Gambar 44. Form input data - Drop-down list - Latihan 1. ....	81
Gambar 45. Form input data- Upload file - antarmuka upload file pada Microsoft Edge.....	83
Gambar 46. Contoh CSS untuk style teks. ....	87
Gambar 47. Contoh Inline CSS. ....	88
Gambar 48. Efek hirarki CSS. ....	89
Gambar 49. Border. ....	91
Gambar 50. Efek style pada elemen HTML. ....	92
Gambar 51. Sintaks pada Internal CSS. ....	93
Gambar 52. Internal CSS - Contoh. ....	94
Gambar 53. Internal CSS - Layout.....	96
Gambar 54. Dasar-dasar CSS - Warna.....	99
Gambar 55. Style warna pada tombol. ....	101
Gambar 56. Style teks. ....	103

Gambar 57. Background image menggunakan gambar ukuran besar.....	105
Gambar 58. Lokasi penyimpanan gambar dan file default.css.....	105
Gambar 59. Membuat background image dengan gambar ukuran kecil. ....	106
Gambar 60. Background image yang diulang .....	106
Gambar 61. Border pada element HTML. ....	107
Gambar 62. Antarmuka aplikasi web Facebook. ....	109
Gambar 63. Antarmuka aplikasi web pencarian Google.....	109
Gambar 64. Antarmuka template utama. ....	110
Gambar 65. Menambahkan background. ....	111
Gambar 66. Menambahkan rangka utama. ....	113
Gambar 67. Menambahkan logo.....	115
Gambar 68. Menambahkan menu di samping logo.....	115
Gambar 69. Elemen list tanpa style. ....	117
Gambar 70. Hyperlink pada menu tanpa style.....	118
Gambar 71. Content kiri dan kanan. ....	122
Gambar 72. Menambahkan teks pada content sebelah kiri. ....	123
Gambar 73. Menambahkan hyperlink berupa icon media sosial.....	125
Gambar 74. Tombol SEE MY WORK. ....	126
Gambar 75. Antarmuka utuh web profile. ....	129
Gambar 76. Form registrasi event.....	130
Gambar 77. Window form input.....	132
Gambar 78. Elemen input data tanpa CSS.....	134
Gambar 79. Form input data yang telah diberikan CSS. ....	138
Gambar 80. Contoh-contoh antarmuka personal web.....	139
Gambar 81. Message box keluaran JavaScript .....	142
Gambar 82. Message box dari fungsi JavaScript. ....	143
Gambar 83. Pohon obyek HTML DOM. ....	153
Gambar 84. Mengubah warna berdasarkan warna class.....	158
Gambar 85. Tampilan awal.....	161

Gambar 86. Tampilan setelah tombol set diklik. ....	162
Gambar 87. Tampilan setelah tombol get diklik. ....	162
Gambar 88. Memodifikasi banyak elemen dengan querySelectorAll.....	166
Gambar 89. Membuat validasi textbox. ....	167



# 1

## Pendahuluan

---

### Internet

Internet atau World Wide Web (WWW) dikhkususkan digunakan untuk publik sejak tahun 1989.

---

### Sejarah Internet

---

#### *1969: Internet dicetuskan*

Pada Oktober 1969, peneliti-peneliti di Universitas California di Los Angeles (UCLA) berupaya untuk mengirimkan data dari satu komputer ke komputer lain. Mereka berupaya untuk mengirimkan tiga huruf, yaitu 'LOG', ke komputer kedua dalam bentuk kode biner, dimana komputer kedua nantinya akan menambahkan dua huruf lagi, sehingga menjadi 'LOGIN'.

Suksesnya penelitian tersebut melahirkan proyek ARPANET, yang dikembangkan dan ditujukan untuk keperluan militer Amerika Serikat. Sistem yang dinyatakan sebagai pendahulu internet tersebut semakin bertumbuh, dari yang sebelumnya hanya menghubungkan 4 komputer, menjadi 13 pada 1970. Pada 1981, telah ada 231 komputer yang terhubung dalam jaringan ARPANET.

#### *1971: Surat elektronik pertama*

Ray Tomlinson, seorang Amerika, pada tahun 1971, mengirimkan surat elektronik pertama menggunakan ARPANET, dan mencetuskan penggunaan simbol '@' dalam alamat surel kita saat ini. Simbol '@' digunakan untuk memisahkan nama pengguna, dengan jaringan yang sedang digunakan.

#### *1983: Komunikasi antar jaringan*

Sebelumnya, agar dua komputer dapat berkomunikasi satu sama lain dalam satu jaringan, diperlukan semacam protokol berupa rangkaian tahapan, yang ditentukan oleh pengatur komunikasi. Pada 1970, Robert Kahn, dan Vinton Cerf, mengembangkan TCP/IP (Transmission Control Protocol/Internet

Protocol) yang memungkinkan pertukaran data dilakukan bukan hanya dalam komputer dengan jaringan yang sama, melainkan juga dalam jaringan berbeda. Penggunaan TCP/IP masih digunakan hingga hari ini. ARPANET mengadopsi teknologi ini pada 1 Januari 1983, memungkinkan hubungan dengan jaringan komputer lain seperti Universitas. Pada poin inilah, internet ada dalam wujudnya yang kita kenal sekarang.

### ***1990: Lahirnya WWW***

WWW dicetuskan oleh Tim Berners-Lee pada 12 Maret 1989, yang saat itu bekerja untuk lab fisika CERN (Organisasi Eropa untuk Riset Nuklir). Ia mengusulkan sistem manajemen informasi yang terdesentralisasi.

Pada saat itu, CERN memiliki ribuan karyawan, dan semakin banyak dengan masuknya karyawan baru. Sangat sulit untuk mendapatkan informasi yang mungkin berkaitan, tetapi tidak diketahui, karena tempat penyimpanan yang berbeda tempat.

Usulnya adalah sistem koneksi hypertext, serta pencarian menggunakan kata kunci, yang dihubungkan dengan informasi-informasi terkait.

Pada tahun 1990, Robert Cailliau dari Belgia membantu mengembangkan usul Berners-Lee. Penemuan ini didasarkan pada dua pilar, yaitu: markup language HTML, yang memungkinkan keberadaan website, serta protokol pertukaran hypertext HTTP, yang memungkinkan pengguna meminta, dan menerima halaman yang diinginkan.

Sistem ini kemudian disebarluaskan pada April tahun 1993, dan semakin populer seiring rilisnya Mosaic pada bulan November. Mosaic merupakan situs pencari (search engine) pertama. Jumlah situs yang dapat diakses, kemudian meledak dari beberapa juta di awal 1990, menjadi 400 juta pada tahun 2000.

(sumber: <https://www.cnnindonesia.com/teknologi/20190312125646-185-376484/mengenal-sejarah-internet>)

---

### **Hypertext Transfer Protocol (HTTP)**

HTTP adalah application-layer protocol untuk transmisi dokumen web seperti HTML. HTTP didesain untuk komunikasi antara web browser (web client) dengan web browser, dan juga dapat digunakan untuk keperluan lain. HTTP mengikuti aturan model client-server pada umumnya. Dimana client memulai

koneksi dengan membuat sebuah request, kemudian menunggu server memberikan response.



*Gambar 1. Aturan model client-server.*

(sumber: <https://developer.mozilla.org/en-US/docs/Web/HTTP>)

## Web Server

---

Web server dapat merujuk sebagai hardware atau software, atau keduanya bekerja sama.

Sebagai software, web server adalah perangkat lunak yang berfungsi sebagai penerima permintaan yang dikirimkan melalui browser kemudian memberikan tanggapan permintaan dalam bentuk halaman situs web atau lebih umumnya dalam dokumen HTML.

Jika merujuk pada hardware, web server digunakan untuk menyimpan semua data seperti HTML dokumen, gambar, file CSS stylesheets, dan file JavaScript. Sedangkan pada sisi software, fungsi web server adalah sebagai pusat kontrol untuk memproses permintaan yang diterima dari browser.

Jadi sebenarnya semua yang berhubungan dengan website biasanya juga berhubungan dengan web server, karena tugas web server adalah mengatur semua komunikasi yang terjadi antara web browser dengan server untuk memproses sebuah website.

Web server yang umum digunakan sekarang ini diantaranya adalah:

1. Apache (<https://httpd.apache.org>)
2. Internet Information Server (IIS) (<https://www.iis.net>)

## Web Browser

---

Web browser adalah aplikasi client untuk mengakses informasi pada World Wide Web. Sejak tahun 1991 sampai sekarang telah banyak web browser yang digunakan. Namun yang sering kita dengar dan gunakan adalah:

1. Internet Explorer.
2. Microsoft Edge.
3. Firefox.
4. Safari.
5. Chrome.

---

## Aplikasi Web

### Aplikasi Desktop & Aplikasi Web

---

Berikut ini adalah beberapa perbandingan antara aplikasi desktop dan aplikasi web.

#### *Aplikasi Desktop*

Berikut adalah kelebihan dan kekurangan dari aplikasi desktop:

1. Aplikasi desktop harus diinstall ke seluruh komputer yang ingin menggunakan. Jika aplikasi dekstop memiliki update, maka update harus dilakukan pada seluruh komputer tersebut.
2. Dapat dijalankan atau digunakan secara offline (tanpa koneksi internet).
3. Menawarkan keamanan lebih bagus, hal ini disebabkan karena tidak terkoneksi dengan internet. Sehingga tidak perlu khawatir serangan dari pihak lain.
4. Bergantung pada kecepatan komputer milik user.
5. User dapat menjalankan versi lama dari aplikasi desktop.
6. Aplikasi dekstop tidak dapat memaksa untuk melakukan upgrade.
7. Aplikasi desktop bergantung dengan sistem operasi. Aplikasi dekstop yang dibangun pada sistem operasi Windows tidak bisa digunakan pada sistem operasi Linux atau MacOS.

## *Applikasi Web*

1. Hanya diinstall pada sebuah web server saja. Jika ada update, maka cukup melakukan update pada server saja.
2. Update yang dilakukan pada server akan segera digunakan oleh user yang menggunakan aplikasi web tersebut.
3. Aplikasi web mudah digunakan untuk digunakan oleh user yang banyak.
4. Aplikasi web tidak bergantung pada sistem operasi, karena hanya memerlukan web browser.

---

## **Teknologi**

Teknologi yang digunakan untuk membangun aplikasi web terdiri sebagai berikut:

1. Teknologi untuk membuat antarmuka yaitu HTML dan CSS.
2. Teknologi untuk programming yang dapat dibagi menjadi dua jenis yaitu client side programming dan server side programming.

Pada buku ini akan dibahas secara mendalam tentang:

1. HTML 5.
2. CSS 3.
3. Javascript sebagai client side programming.
4. PHP sebagai server side programming.

---

## **Code Editor**

Perangkat code editor yang digunakan untuk untuk menulis kode HTML 5, CSS, Javascript dan PHP yang tersedia saat ini sangat beragam, diantaranya adalah:

1. Visual Studio Code (<https://code.visualstudio.com/download>).
2. Notepad++ (<https://notepad-plus-plus.org/downloads/>).
3. Sublime Text (<https://www.sublimetext.com/>).

---

## Source Code

Contoh-contoh kode program yang ditulis pada buku ini dapat diunduh pada alamat web berikut ini:

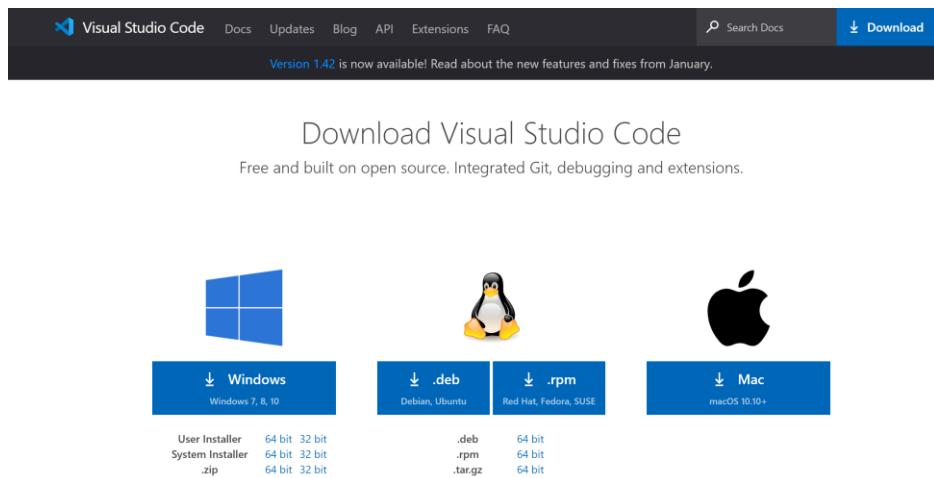
<https://github.com/rezafaisal/PemrogramanWebDasar>

## **Visual Studio Code**

Pada buku ini digunakan Visual Studio Code sebagai code editor untuk menulis kode HTML, CSS dan PHP. Selain itu code editor ini dapat digunakan untuk melakukan koneksi dan operasi ke database MySQL sehingga dapat digunakan untuk menyiapkan database untuk aplikasi web yang sedang dibangun.

### **Installasi**

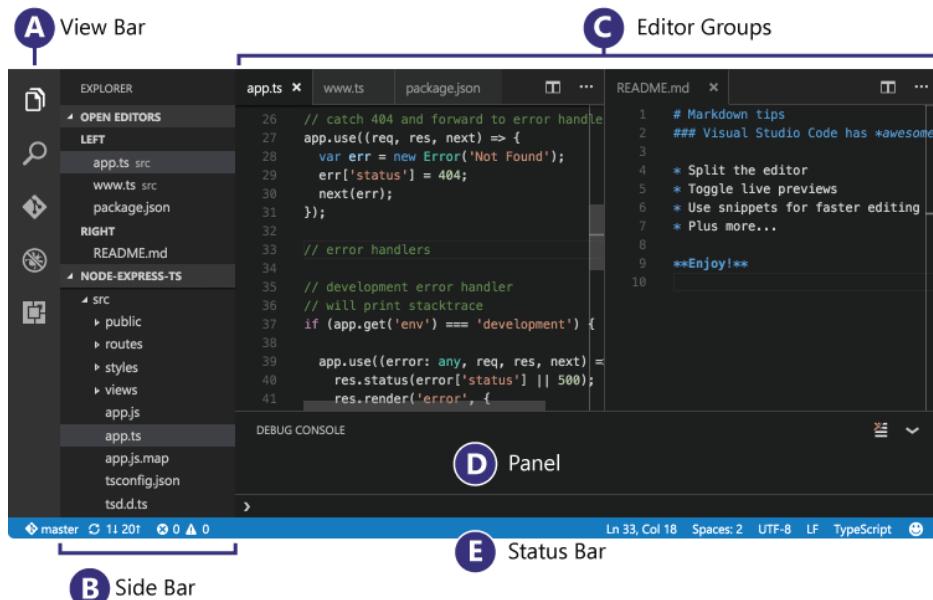
Installer Visual Studio Code dapat diunduh pada link berikut ini <https://code.visualstudio.com/download>. Pilih installer yang sesuai dengan sistem operasi yang digunakan. Saat buku ini diketik versi yang ada adalah 1.42.1.



**Gambar 2. Halaman download Visual Studio Code.**

## Antarmuka

Pada gambar di bawah ini dapat dilihat layout dasar antarmuka Visual Studio Code.



Gambar 3. Layout dasar antarmuka Visual Studio Code.

Dari gambar di atas dapat dilihat layout terbagi atas 5 bagian yaitu:

- a. View Bar, berisi tombol-tombol yaitu:
  - Explorer yang berfungsi untuk melihat daftar folder dan file didalamnya.
  - Search untuk melakukan pencarian.
  - Git untuk melihat daftar folder yang disimpan pada Git repository.
  - Debug.
  - Extension untuk melihat daftar extension yang tersedia pada repository.
- b. Side Bar, akan menampilkan view sesuai dengan tombol pada View Bar yang dipilih.
- c. Editor Group, akan menampilkan window-window editor dari file yang dibuka. Artinya jika dibuka 2 file maka akan dapat terlihat 2 windows seperti pada gambar di atas.
- d. Panel akan memberikan informasi output dari hasil proses debug, error, warning dan dapat juga berfungsi sebagai integrated terminal.

- e. Status Bar berfungsi untuk menampilkan informasi dari project dan file yang sedang diedit.

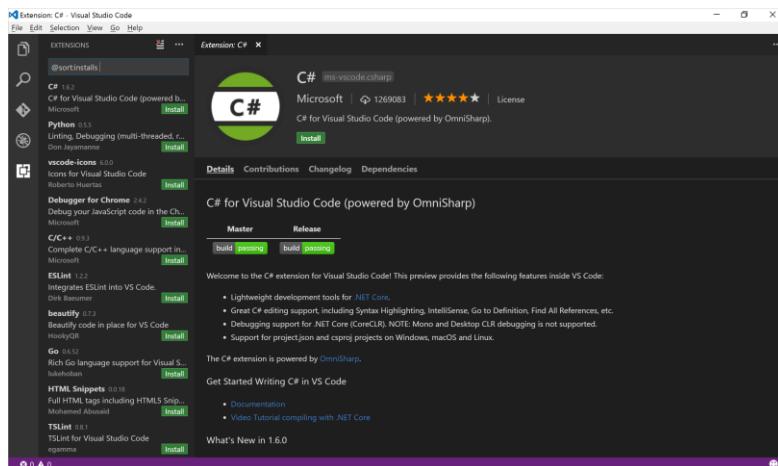
## Tool Tambahan

Pada sub bab ini akan diterangkan cara untuk menambah tool atau extension yang tersedia pada repository. Akses ke repository memerlukan akses internet. Cara menginstall extension dapat dilakukan dengan dua cara, yaitu dengan menggunakan:

1. Extensions.
2. Integrated Terminal.

## Extension

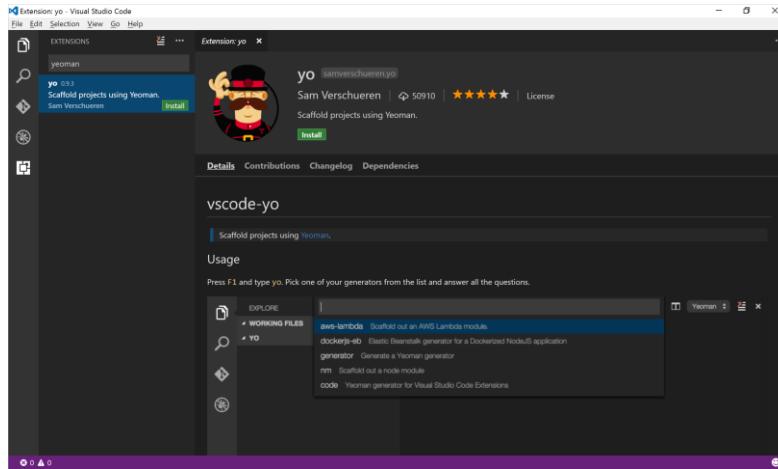
Untuk mengakses daftar extension yang tersedia dapat mengklik tombol Extensions pada View Bar. Berikut adalah tampilan daftar extension.



Gambar 4. Daftar extension.

Klik tombol Install kemudian setelah installasi selesai, klik tombol Reload.

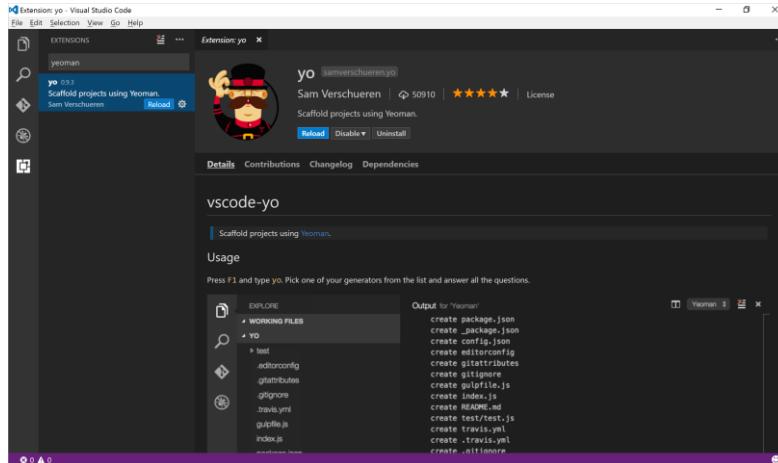
Untuk mencari extension yang diinginkan dapat mengetikkan keyword pada kolom pencarian.



*Gambar 5. Pencarian extension.*

Sebagai contoh, keyword yang digunakan pada kolom pencarian adalah yeoman. Maka akan ditampilkan extension yang sesuai dengan keyword. Jika ingin melihat detail dari extension maka dapat diklik. Maka detail extension yo dapat dilihat seperti pada gambar. Klik tombol install yang berwarna hijau untuk menginstall extension ini.

Setelah proses installasi selesai maka akan dapat dilihat perubahan status pada tombol install menjadi seperti pada gambar di bawah ini.

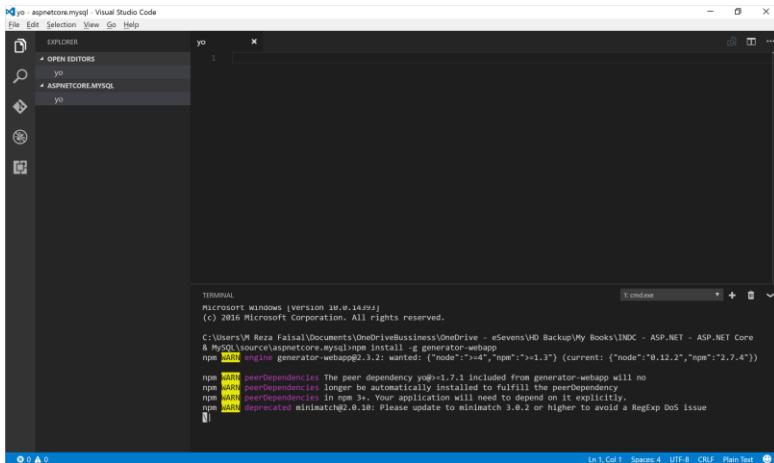


*Gambar 6. Extension berhasil diinstall.*

Klik tombol Reload untuk mengaktifkan extension ini.

## Integrated Terminal

Cara kedua adalah dengan Terminal. Untuk menampilkan terminal dapat dilakukan dengan cara memilih menu View > Integrated Terminal.



*Gambar 7. Integrated terminal.*

Integrated Terminal seperti command prompt yang memungkinkan pengguna menuliskan perintah. Perintah yang dapat digunakan untuk menginstall extension adalah npm (nuget package manager).

Sebagai contoh, jika ingin menginstall extension generator-web maka dapat diketik perintah berikut ini pada terminal.

```
npm install -g generator-webapp
```

Maka hasilnya akan dapat dilihat pada gambar di atas.

## ***HyperText Markup Language***

HyperText Markup Language atau lebih dikenal dengan HTML adalah bahasa yang digunakan untuk membuat antarmuka halaman web.

---

### **Pendahuluan**

#### **Sejarah**

---

HTML dibuat oleh Tim Berners-Lee diakhir tahun 1991 tetapi belum secara resmi. Kemudian tahun 1995 dirilis HTML 2.0. Sedangkan versi utama dirilis pada tahun 1999 yaitu HTML 4.01.

#### ***HTML 1.0***

Versi ini dirilis tahun 1993 dengan tujuan berbagi informasi yang dapat dibaca dan diakses melalui web browser. Tetapi karena belum banyak developer yang membuat website maka bahasa ini tidak berkembang.

#### ***HTML 2.0***

Versi ini dirilis 1995. Versi ini berisi seluruh fitur pada versi 1.0 ditambah beberapa tambahan fitur. Versi ini bertahan sampai tahun 1997.

#### ***HTML 3.0***

Versi ini memiliki banyak penambahan fitur baru HTML dan memberikan banyak kemudahan kepada web master untuk mendesain halaman web. Tetapi fitur-fitur hebat dari HTML baru ini memperlambat web browser.

## **HTML 4.01**

Versi ini banyak digunakan dan versi yang sukses sebelum HTML 5.0 dirilis. Versi 4.01 digunakan diseluruh dunia. Kemudian selanjutnya adalah HTML 5.0 yang merupakan extended version dari HTML 4.01.

## **Sintaks Tag**

---

HTML elemen atau elemen atau HTML tag atau tag adalah kata kunci dalam halaman web yang menentukan bagaimana web browser harus memformat dan menampilkan konten.

Hampir seluruh tag harus memiliki dua bagian yaitu tag pembuka dan penutup. Aturan penulisan tag pembuka adalah sebagai berikut:

```
<nama_tag>
```

Dari sintaks di atas dapat dilihat, tag pembuat diawali dengan tanda "<" kemudian diikuti dengan kata kunci "nama\_tag". Kemudian diakhiri dengan tanda ">".

Sedangkan sintaks tag penutup adalah sebagai berikut:

```
</nama_tag>
```

Jika dibandingkan dengan tag pembuka, sintaks tag penutup memiliki perbedaan yaitu adanya penambahan tanda "/" setelah tanda "<".

Tag ini ditulis berpasangan dengan sintaks sebagai berikut:

```
<nama_tag> ... </nama_tag>
```

Di antara tag pembuka dan penutup dari berisi tag-tag lain. Berikut ini adalah contoh-contoh tag HTML.

Contoh 1:

```
<p>Hello World</p>
```

Contoh 2:

```
<head>
    <title>Selamat Datang di Web Ilmu Komputer</title>
</head>
```

Contoh 3:

```
<b>Tulisan ini dicetak tebal</b>
```

Contoh 4:

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
</ul>
```

Tetapi tidak semua tag HTML terdiri dari dua bagian (pembuka dan penutup). Ada tag yang hanya memiliki satu bagian saja. Tag jenis ini disebut juga void elemen. Tag jenis ini memiliki sintaks sebagai berikut:

```
<nama_tag />
```

Atau

```
<nama_tag>
```

Contoh tag jenis ini adalah sebagai berikut:

Contoh 1:

```
<br/>
```

Atau

```
<br>
```

Contoh 2:

```
<hr/>
```

Atau

```
<hr>
```

Contoh 3:

```

```

Atau

```

```

Selain contoh di atas tag yang tergolong dengan void element adalah:

- area
- base
- br
- col
- embed
- hr
- img
- input
- link
- meta
- param
- source
- track
- wbr

Pada penjelasan tentang void elemen di atas terdapat dua cara penulisan. Cara pentulisan dimana tag ditulis dengan menggunakan penutup (contoh: <br/>) adalah cara penulisan mengikuti aturan XHTML (Extensible HyperText Markup Language). XHTML merupakan gabungan XML dan HTML. Aturan ini telah menjadi standar sejak tahun 2000. Salah satu aturannya adalah, setiap tag yang dibuka harus ditutup. Sehingga kita melihat penulisan tag seperti <hr/> atau <br/>. Sedangkan aturan pada HTML lebih longgar sehingga tag tipe void elemen dapat ditulis tanpa penutup. Sehingga cukup ditulis sebagai berikut <br> atau <hr>. Kedua aturan di atas didukung dan dikembangkan oleh World Wide Web Consortium (W3C)

Pembaca dapat memilih akan mengikuti aturan yang mana. Namun jika pada <https://www.w3schools.com> dapat dilihat penggunaan aturan HTML pada contoh-contohnya. Sehingga pada buku ini dapat dilihat penerapan aturan tersebut.

## Atribut

---

Tag dapat memiliki atribut-atribut. Artinya setiap tag dapat memiliki 1 atau lebih atribut. Atribut berfungsi untuk menyesuaikan sebuah tag. Atribut hanya diberikan pada tag pembuka atau tag yang hanya memiliki satu bagian saja. berikut adalah sintaks penggunaan atribut.

```
<nama_tag atribut1="nilai" atribut2="nilai"></nama_tag>
```

Atau sintaks berikut ini untuk tag yang hanya memiliki satu bagian saja.

```
<nama_tag atribut1="nilai" atribut2="nilai">
```

Contoh penulisan atribut pada tag HTML adalah sebagai berikut.

Contoh 1:

```
<body bgcolor="pink">
```

Contoh 2:

```

```

---

## Aturan Penamaan File & Folder

---

Nama file untuk halaman web sebaiknya jangan ada spasi, jadi jika halaman terdiri atas lebih 2 kata maka dapat disambung dengan tanda “\_” atau garis bawah. Sebagai contoh, misal dimiliki 2 kata yaitu “hello world”, maka nama filenya adalah hello\_world diikuti dengan ekstensi file (.html, .jpg, .png dan lain-lain).

Cara yang lain adalah menggabung kata-kata tersebut namun huruf awal kata dibuat menjadi huruf besar. Sebagai contoh kata “hello world”, maka nama filenya menjadi HelloWorld diikuti dengan ekstensi file.

Aturan ini juga bisa digunakan untuk penamaan folder. Sehingga nama folder atau file tidak ada spasi.

## Halaman Web Sederhana

Setelah mengetahui sintaks tag dan atribut. Maka selanjutnya adalah membuat sebuah halaman web sederhana. Halaman ini disimpan ke dalam file dengan nama hello\_world.html.

```
hello_world.html
<!DOCTYPE HTML>
<html>
    <!-- komentar -->
    <head>
        <title>Halaman Pertama Saya</title>
    </head>
    <body>
        Hello World.
        Ini adalah halaman pertama yang saya buat.
        <!-- komentar -->
    </body>
</html>
```

Penjelasan kode di atas adalah sebagai berikut.

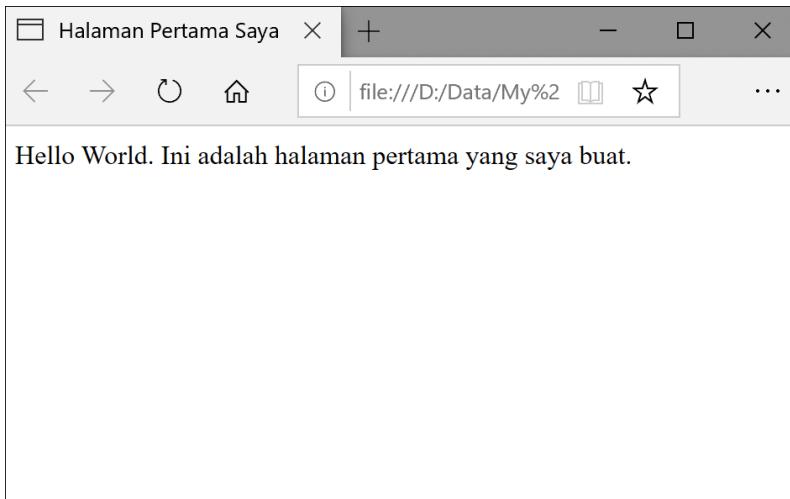
Baris pertama adalah tag spesial yang berfungsi untuk memberikan informasi kepada web browser bahwa tipe dokumen ini adalah HTML.

Baris kedua adalah tag `<html>` sebagai tag pembuka dan ditutup oleh `</html>` pada baris terakhir. Tag ini harus ada pada sebuah halaman web. Di antara tag `<html>` dan `</html>` terdapat tag-tag lain.

Pada baris ketiga dapat dilihat tag `<head>` sebagai pembuka dan kembali ditutup dengan tag `</head>`. Di antara tag ini disebut area header. Area header ini tidak akan ditampilkan secara visual pada web browser. Pada area ini juga sering digunakan untuk menulis kode program, informasi style dan lain-lain yang akan dijelaskan pada sub bab berikutnya.

Di dalam tag `<head></head>`, dapat dilihat tag `<title></title>` yang berfungsi untuk memberikan judul pada halaman. Nilai yang ingin ditampilkan sebagai judul halaman ditulis di antara tag pembuka `<title>` dan tag penutup `</title>`. Secara visual nilai ini akan dapat dilihat pada tab pada web browser.

Tag selanjutnya yang wajib ada adalah tag `<body></body>`. Tag atau nilai yang berada diantara tag ini akan ditampilkan pada web browser sebagai isi.



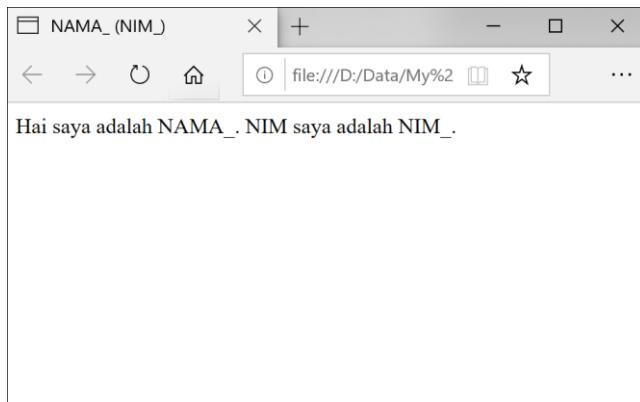
*Gambar 8. Halaman web sederhana.*

Selain tag, ada juga cara untuk memberikan komentar pada kode HTML. Contoh kode komentar dapat dilihat pada kode HTML di atas. Sintaks untuk membuat komentar secara umum adalah sebagai berikut.

```
<!-- ketik komentar di sini -->
```

### **Latihan 1**

Buat sebuah halaman web sederhana seperti gambar berikut ini. ganti "NAMA\_" dan "NIM\_" dengan nama dan NIM pembaca inginkan.



*Gambar 9. Halaman Web Sederhana - Latihan 1.*

Kode HTML dari tampilan di atas adalah sebagai berikut ini.

```
web_sederhana_latihan_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>NAMA_ (NIM_)</title>
  </head>

  <body>
    Hai saya adalah NAMA_.
    NIM saya adalah NIM_.
  </body>
</html>
```

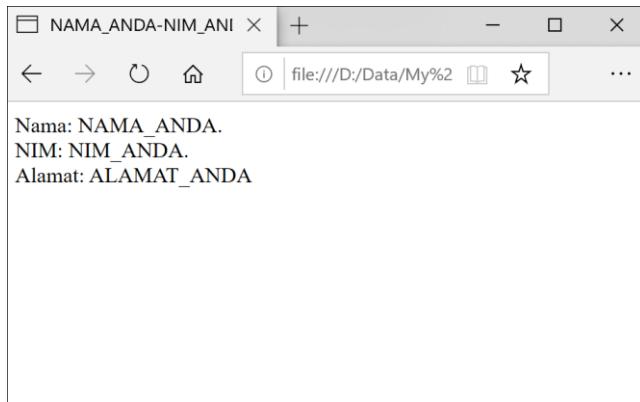
## Latihan 2

Ketik kode HTML berikut dengan dan simpan ke dalam file dengan nama web\_sederhana\_latihan\_2.html.

```
web_sederhana_latihan_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>NAMA_ANDA-NIM_ANDA</title>
  </head>

  <body>
    Nama: NAMA_ANDA. <br>
    NIM: NIM_ANDA. <br>
    Alamat: ALAMAT_ANDA
  </body>
</html>
```

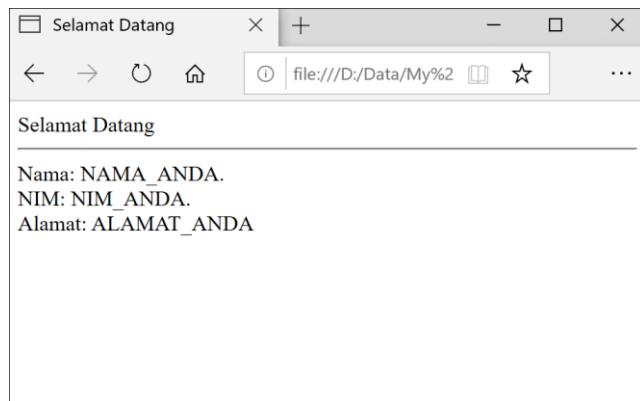
Hasilnya dapat dilihat pada Gambar 10. Pada kode di atas dapat dilihat tag baru yaitu <br/> yang berfungsi seperti menekan tombol enter. Tag <br/> membuat teks setelahnya ditulis pada baris baru.



**Gambar 10. Halaman web sederhana - Latihan 2.**

### Latihan 3

Pada latihan ini diperkenalkan tag <hr/> yang berfungsi untuk membuat garis horizontal. Buat halaman web dengan tampilan seperti pada Gambar 11.



**Gambar 11. Halaman web sederhana - Latihan 3.**

Kode HTML yang harus diketik untuk menghasilkan halaman seperti gambar di atas adalah sebagai berikut.

```
web_sederhana_latihan_3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Selamat Datang</title>
  </head>

  <body>
    Selamat Datang <hr>
```

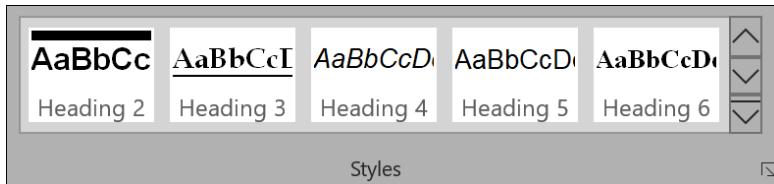
```
Nama: NAMA_ANDA. <br>
NIM: NIM_ANDA. <br>
Alamat: ALAMAT_ANDA
</body>
</html>
```

## Memformat Teks

Pada sub bab ini akan diberikan tag-tag untuk memformat teks.

### Heading

Pada aplikasi word processor seperti MS Word (Gambar 12), kita dapat melihat opsi untuk mengubah style teks yang ditulis untuk menjadi judul dan sub judul atau bab dan sub bab. Style yang digunakan untuk tujuan ini disebut sebagai heading.



Gambar 12. Style heading pada MS Word.

Pada HTML, untuk membuat heading digunakan tag sebagai berikut.

```
heading_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Belajar Heading</title>
  </head>

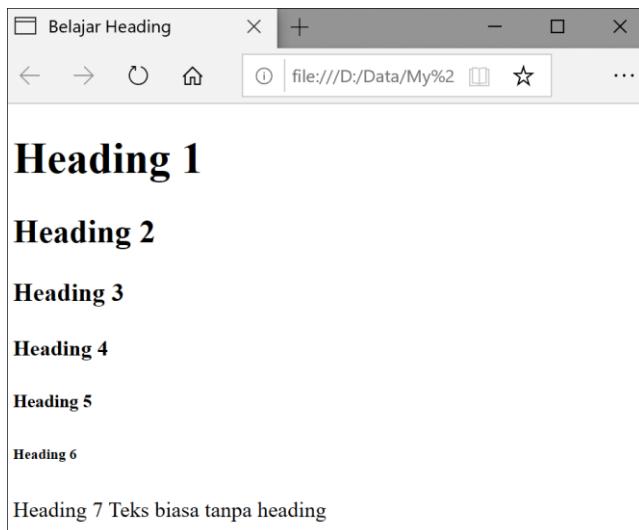
  <body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>
    <h7>Heading 7</h7>
    Teks biasa tanpa heading
  </body>
</html>
```

Untuk heading 1 maka digunakan tag <h1></h1>. Teks yang ingin menggunakan heading 1 harus diletakkan diantara tag pembuka dan penutup tersebut. Untuk heading dapat menggunakan heading 1 sampai heading 6 saja, artinya tag yang dapat digunakan adalah:

- <h1></h1>
- <h2></h2>
- <h3></h3>
- <h4></h4>
- <h5></h5>
- <h6></h6>

Perbedaan antara tag-tag di atas adalah ukuran font, dimana font dari tag <h1> adalah yang paling besar dan yang paling kecil adalah tag <h6>. Perbedaan ukuran font ini dapat dilihat pada Gambar 13.

Pada kode di atas juga dapat dilihat penggunaan tag <h7>, namun telah dijelaskan sebelumnya bahwa tag ini tidak ada hubungannya dengan heading 7. Sebagai buktinya dapat dilihat pada Gambar 13, dimana teks “Heading 7” yang menggunakan tag <h7> dan teks “Teks biasa tanpa heading” mempunyai ukuran teks yang sama.



*Gambar 13. Memformat teks - Heading.*

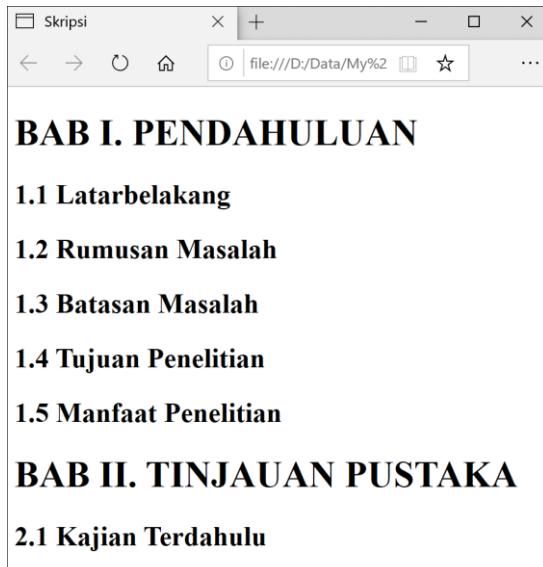
## Latihan 1

Buat teks dengan hierarki memiliki tampilan seperti pada Gambar 14. Hierarki hanya terdiri atas 2 tingkatan, sehingga cukup menggunakan tag <h1> dan <h2> saja.

Berikut adalah kode HTML untuk membuat tampilan seperti Gambar 14.

```
heading_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Skripsi</title>
  </head>

  <body>
    <h1>BAB I. PENDAHULUAN</h1>
    <h2>1.1 Latarbelakang</h2>
    <h2>1.2 Rumusan Masalah</h2>
    <h2>1.3 Batasan Masalah</h2>
    <h2>1.4 Tujuan Penelitian</h2>
    <h2>1.5 Manfaat Penelitian</h2>
    <h1>BAB II. TINJAUAN PUSTAKA</h1>
    <h2>2.1 Kajian Terdahulu</h2>
  </body>
</html>
```

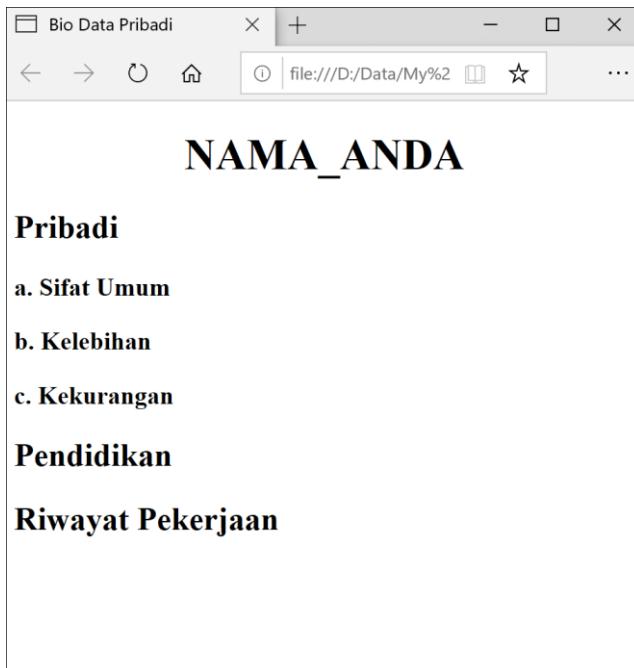


Gambar 14. Memformat teks - Heading - Latihan 1.

## Latihan 2

Pada Gambar 15 adalah contoh lain antarmuka yang dibuat sebagai latihan. Terlihat perbedaan pada teks “NAMA\_ANDA” yang berada pada posisi di tengah (center) dari halaman web. Untuk membuat teks berada di tengah maka gunakan tag <center> dengan sintaks sebagai berikut ini.

```
<center>Teks ditulis di antara tag ini</center>
```



Gambar 15. Memformat teks - Heading - Latihan 2.

Berikut ini adalah kode HTML untuk membuat tampilan seperti gambar di atas.

```
heading_3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <body>
    <center><h1>NAMA_ANDA</h1></center>
```

```
<h2>Pribadi</h2>
<h3>a. Sifat Umum</h3>
<h3>b. Kelebihan</h3>
<h3>c. Kekurangan</h3>
<h2>Pendidikan</h2>
<h2>Riwayat Pekerjaan</h2>
</body>
</html>
```

### Pertanyaan

Pada kode di atas dapat dilihat tag `<center>` dan `</center>` mengapit `<h1>NAMA_ANDA</h1>`. Hal ini membuat tulisan dengan heading 1 ini dibuat posisinya menjadi di tengah.

Jika kode tersebut diubah menjadi sebaliknya sebagai berikut:

```
<h1><center>NAMA_ANDA</center></h1>
```

Apakah teks NAMA\_ANDA masih terlihat teks dengan ukuran besar dan posisi di tengah?

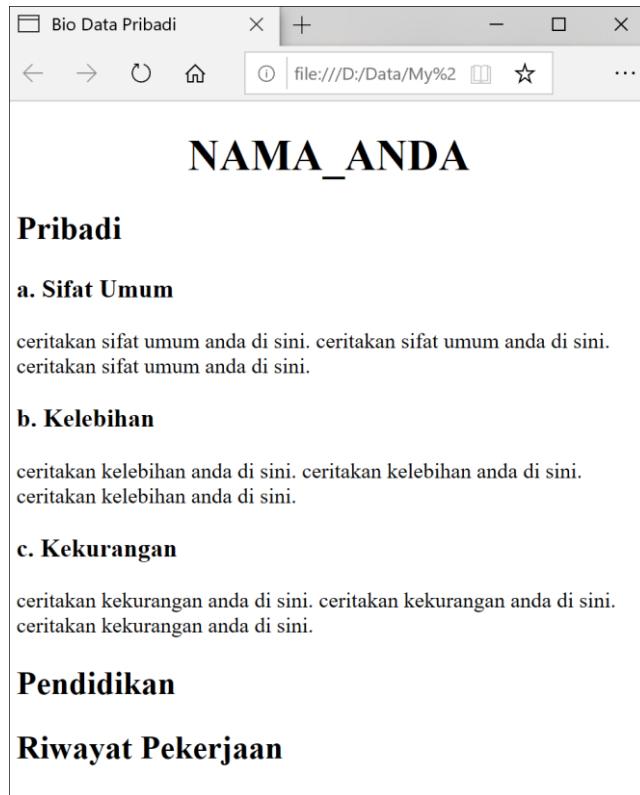
### Paragraf

Untuk mengumpulkan kumpulan teks kedalam sebuah paragraf digunakan tag `<p>` dengan sintaks sebagai berikut.

```
<p>
    teks ditulis di sini
</p>

<p>teks ditulis di sini</p>
```

Berikut ini adalah contoh penulisan paragraf maka gunakan file heading\_3.html yang telah ditulis sebelumnya. kemudian tambahkan tag `<p>` pada bagian “Pribadi” seperti pada gambar berikut ini.



*Gambar 16. Memformat teks - Paragraf.*

Berikut ini adalah kode HTML dari gambar di atas.

```
paragraf_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <body>
    <center><h1>NAMA_ANDA</h1></center>
    <h2>Pribadi</h2>
    <h3>a. Sifat Umum</h3>
    <p>
      ceritakan sifat umum anda di sini.
      ceritakan sifat umum anda di sini.
      ceritakan sifat umum anda di sini.
    </p>
    <h3>b. Kelebihan</h3>
    <p>
```

```

ceritakan kelebihan anda di sini.
ceritakan kelebihan anda di sini.
ceritakan kelebihan anda di sini.
</p>
<h3>c. Kekurangan</h3>
<p>
    ceritakan kekurangan anda di sini.
    ceritakan kekurangan anda di sini.
    ceritakan kekurangan anda di sini.
</p>
<h2>Pendidikan</h2>
<h2>Riwayat Pekerjaan</h2>
</body>
</html>

```

## Latihan 1

Untuk latihan, tambahkan dua paragraf pada bagian “a. Sifat Umum”, sehingga bagian itu memiliki tiga paragraf seperti pada Gambar 17. Pada gambar tersebut dapat dilihat teks yang ditulis di dalam sebuah paragraf memiliki jarak antar baris yang lebih rapat jika dibandingkan dengan jarak baris antar dua paragraf.



*Gambar 17. Memformat teks - Paragraf - Latihan 1.*

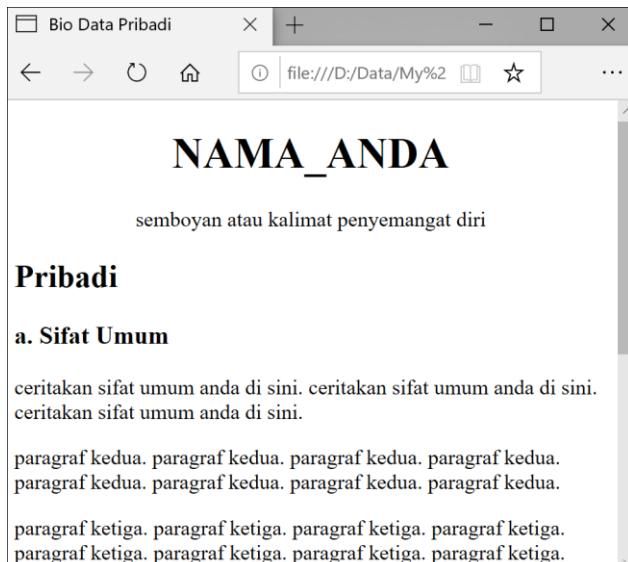
Berikut ini adalah kode HTML untuk membuat antarmuka seperti gambar tersebut.

```
paragraf_02.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Bio Data Pribadi</title>
    </head>

    <body>
        <center><h1>NAMA_ANDA</h1></center>
        <h2>Pribadi</h2>
        <h3>a. Sifat Umum</h3>
        <p>
            ceritakan sifat umum anda di sini.
            ceritakan sifat umum anda di sini.
            ceritakan sifat umum anda di sini.
        </p>
        <p>
            paragraf kedua. paragraf kedua.
            paragraf kedua. paragraf kedua.
            paragraf kedua. paragraf kedua.
            paragraf kedua. paragraf kedua.
        </p>
        <p>
            paragraf ketiga. paragraf ketiga.
            paragraf ketiga. paragraf ketiga.
            paragraf ketiga. paragraf ketiga.
            paragraf ketiga. paragraf ketiga.
        </p>
        <h3>b. Kelebihan</h3>
        <p>
            ceritakan kelebihan anda di sini.
            ceritakan kelebihan anda di sini.
            ceritakan kelebihan anda di sini.
        </p>
        <h3>c. Kekurangan</h3>
        <p>
            ceritakan kekurangan anda di sini.
            ceritakan kekurangan anda di sini.
            ceritakan kekurangan anda di sini.
        </p>
        <h2>Pendidikan</h2>
        <h2>Riwayat Pekerjaan</h2>
    </body>
</html>
```

## Latihan 2

Untuk latihan berikutnya adalah menambahkan kalimat di bawah teks “NAMA\_ANDA” dengan menggunakan tag `<p>`. Teks tersebut juga harus berada di posisi tengah (center) seperti pada Gambar 18.



*Gambar 18. Memformat teks - Paragraf - Latihan 2.*

Kode HTML dari antarmuka di atas adalah sebagai berikut.

```
paragraf_3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <body>
    <center>
      <h1>NAMA_ANDA</h1>
      <p>semboyan atau kalimat penyemangat diri</p>
    </center>
    <h2>Pribadi</h2>
    <h3>a. Sifat Umum</h3>
    <p>
      ceritakan sifat umum anda di sini.
      ceritakan sifat umum anda di sini.
      ceritakan sifat umum anda di sini.
    </p>
    <p>
```

```
paragraf kedua. paragraf kedua.  
paragraf kedua. paragraf kedua.  
paragraf kedua. paragraf kedua.  
paragraf kedua. paragraf kedua.  
</p>  
<p>  
paragraf ketiga. paragraf ketiga.  
paragraf ketiga. paragraf ketiga.  
paragraf ketiga. paragraf ketiga.  
paragraf ketiga. paragraf ketiga.  
</p>  
<h3>b. Kelebihan</h3>  
<p>  
ceritakan kelebihan anda di sini.  
ceritakan kelebihan anda di sini.  
ceritakan kelebihan anda di sini.  
</p>  
<h3>c. Kekurangan</h3>  
<p>  
ceritakan kekurangan anda di sini.  
ceritakan kekurangan anda di sini.  
ceritakan kekurangan anda di sini.  
</p>  
<h2>Pendidikan</h2>  
<h2>Riwayat Pekerjaan</h2>  
</body>  
</html>
```

Pada kode di atas dapat dilihat bahwa sebuah tag `<center>` dapat berisi beberapa jenis tag, pada contoh di atas yaitu tag `<h1>` dan `<p>`.

## **Italic, Bold & Underline**

---

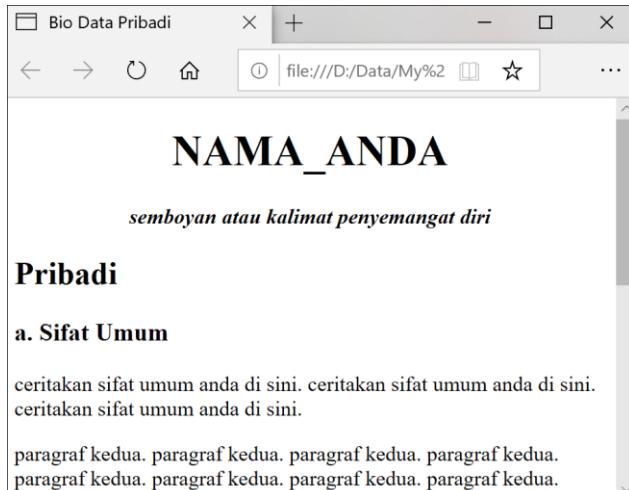
Untuk membuat teks pada suatu kata atau kalimat menjadi miring (*italic*) maka digunakan tag `<i>`. Sedangkan untuk membuat teks menjadi tebal (**bold**) digunakan tag `<b>`. Dan untuk membuat teks dengan garis bawah (underline) digunakan tag `<u>`.

Cara penggunaan ketiga tag ini seperti cara penggunaan tag heading dan paragraf, yaitu teks harus berada diantara tag pembuka dan penutup tag tersebut seperti sintaks di bawah ini.

```
<i>teks miring</i>  
<b>teks tebal</b>  
<u>teks dengan garis bawah</u>
```

## Latihan 1

Buat kalimat “semboyan atau kalimat penyemangat diri” pada file paragraf\_3.html menjadi garis miring dan cetak tebal seperti pada Gambar 19. Kemudian simpan hasilnya pada file baru dengan nama ibu\_1.html.



Gambar 19. Memformat teks - *Italic*, **bold** & underline - Latihan 1.

Sebagian kode HTML dari antarmuka di atas adalah sebagai berikut.

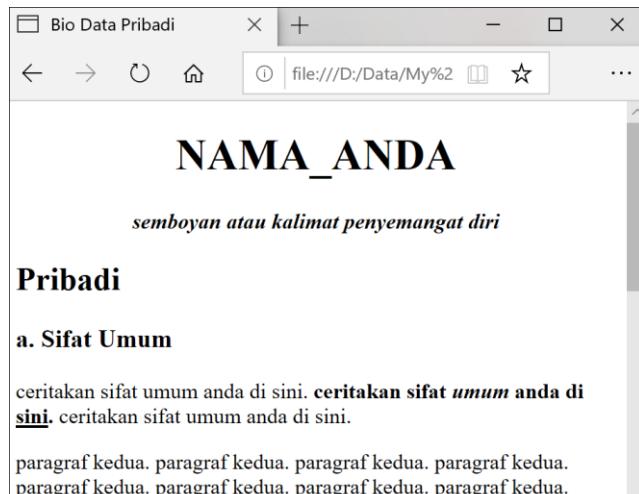
```
ibu_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <body>
    <center>
      <h1>NAMA_ANDA</h1>
      <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
    </center>
    .
    .
    .

  </body>
</html>
```

## Latihan 2

Latihan berikutnya adalah membuat sebuah kalimat dicetak tebal. Kemudian pilih satu kata untuk dicetak miring dan satu kata untuk bergaris bawah dari sebuah kalimat tersebut. Hasil yang dihasilkan adalah seperti pada Gambar 20.



*Gambar 20. Memformat teks - Italic, bold & underline - Latihan 2.*

Berikut adalah sebagian kode HTML yang harus ditulis.

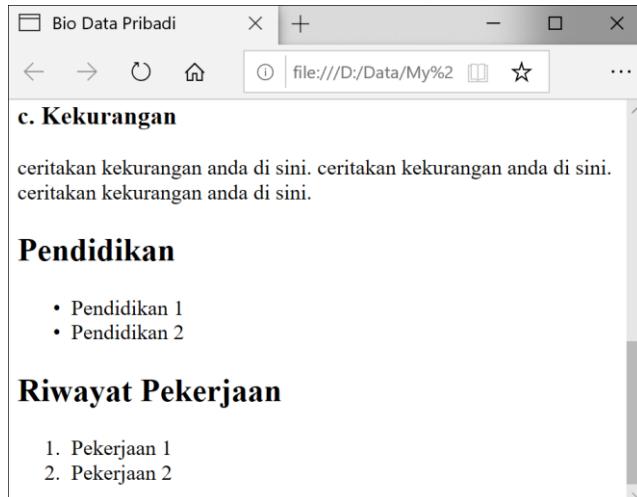
```
ibu_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <body>
    <center>
      <h1>NAMA_ANDA</h1>
      <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
    </center>
    <h2>Pribadi</h2>
    <h3>a. Sifat Umum</h3>
    <p>
      ceritakan sifat umum anda di sini.
      <b>ceritakan sifat <i>umum</i> anda di <u>sini</u>.</b>
      ceritakan sifat umum anda di sini.
    </p>
    . . .
  </body>
</html>
```

## List

---

Seperti halnya di word processor seperti MS Word, user juga bisa membuat list dalam bentuk *bullet* atau *numbering*. Berikut adalah contoh tampilan sederhana list pada HTML.



*Gambar 21. Memformat teks - List.*

Pada Gambar 21 dapat dilihat contoh list tipe *bullet* pada bagian "Pendidikan", sedangkan tipe *numbering* pada bagian "Riwayat Pekerjaan".

Untuk membuat list tipe bullet digunakan sintaks berikut ini. Tag yang digunakan adalah `<ul>` yang merupakan singkatan dari unordered list. Hal ini karena item list tidak ditampilkan dengan urutan angka.

```
<ul>
    <li>Item 1</li>
    <li>Item 2</li>
</ul>
```

Sedangkan untuk membuat list tipe numbering digunakan sintaks sebagai berikut. Tag yang digunakan adalah `<ol>` yang merupakan singkatan ordered list. Hal ini karena item list ditampilkan dengan angka yang terurut.

```
<ol>
    <li>Item 1</li>
    <li>Item 2</li>
</ol>
```

Diantara tag <ul></ul> atau <ol></ol> terdapat tag <li> yang berfungi untuk item yang akan ditampilkan. Item yang ditampilkan dapat berupa:

- Kata.
- Kalimat.
- Gambar.
- Dan lain-lain.

### Latihan 1

Dengan menggunakan file ibu\_2.html yang telah dibuat sebelumnya, tambahkan list tipe bullet atau unordered list pada bagian “Pendidikan” seperti pada Gambar 21. Tulis jenjang pendidikan dan nama sekolah yang pernah anda lakukan. Simpan file sebagai list\_1.html.

Berikut adalah sebagian kode HTML agar didapat tampilan seperti gambar tersebut.

```
list_1.html

<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <body>
    . . .

    <h2>Pendidikan</h2>
    <ul>
      <li>Pendidikan 1</li>
      <li>Pendidikan 2</li>
    </ul>
    . . .

  </body>
</html>
```

### Latihan 2

Latihan berikutnya adalah membuat list tipe numbering atau ordered list pada bagian “Pekerjaan” seperti pada Gambar 21. Tuliskan pekerjaan atau pengalaman berorganisasi pada bagian ini.

Berikut adalah contoh lengkap kode HTML agar didapat tampilan seperti gambar tersebut.

```
list_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Bio Data Pribadi</title>
    </head>

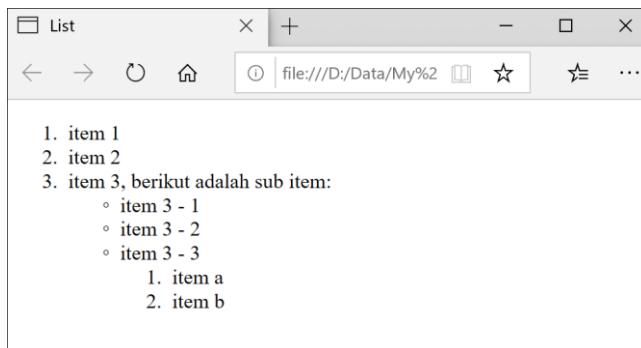
    <body>
        <center>
            <h1>NAMA_ANDA</h1>
            <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
        </center>
        <h2>Pribadi</h2>
        <h3>a. Sifat Umum</h3>
        <p>
            ceritakan sifat umum anda di sini.
            <b>ceritakan sifat <i>umum</i> anda di <u>sini</u>.</b>
            ceritakan sifat umum anda di sini.
        </p>
        <p>
            paragraf kedua. paragraf kedua.
            paragraf kedua. paragraf kedua.
            paragraf kedua. paragraf kedua.
            paragraf kedua. paragraf kedua.
        </p>
        <p>
            paragraf ketiga. paragraf ketiga.
            paragraf ketiga. paragraf ketiga.
            paragraf ketiga. paragraf ketiga.
            paragraf ketiga. paragraf ketiga.
        </p>
        <h3>b. Kelebihan</h3>
        <p>
            ceritakan kelebihan anda di sini.
            ceritakan kelebihan anda di sini.
            ceritakan kelebihan anda di sini.
        </p>
        <h3>c. Kekurangan</h3>
        <p>
            ceritakan kekurangan anda di sini.
            ceritakan kekurangan anda di sini.
            ceritakan kekurangan anda di sini.
        </p>
        <h2>Pendidikan</h2>
        <ul>
            <li>Pendidikan 1</li>
            <li>Pendidikan 2</li>
        </ul>
    </body>

```

```
<h2>Riwayat Pekerjaan</h2>
<ol>
    <li>Pekerjaan 1</li>
    <li>Pekerjaan 2</li>
</ol>
</body>
</html>
```

### Latihan 3

Dalam sebuah item pada suatu list, selain dapat berisi teks dan gambar juga dapat berisi list lain seperti pada Gambar 22.



*Gambar 22. List di dalam list.*

Berikut adalah kode HTML dari gambar di atas.

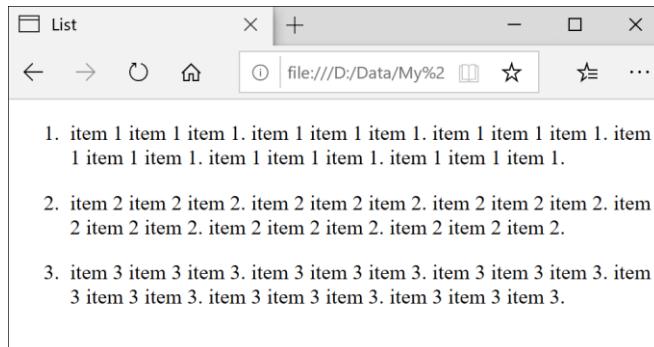
```
list_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>List</title>
    </head>

    <body>
        <ol>
            <li>item 1</li>
            <li>item 2</li>
            <li>
                item 3, berikut adalah sub item:
                <ul>
                    <li>item 3 - 1</li>
                    <li>item 3 - 2</li>
                    <li>
                        item 3 - 3
                        <ol>
```

```
        <li>item a</li>
        <li>item b</li>
    </ol>
</li>
</ol>
</body>
</html>
```

#### Latihan 4

Selain itu di dalam list dapat berisi paragraf atau gambar (contoh gambar akan diberikan pada sub bab selanjutnya). Berikut adalah contoh tampilan paragraf di dalam list.



*Gambar 23. Paragraf di dalam list.*

Di bawah ini adalah kode sHTML dari tampilan Gambar 23.

```
list_4.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>List</title>
    </head>

    <body>
        <ol>
            <li>
                <p>
                    item 1 item 1 item 1.
                    item 1 item 1 item 1.
                </p>
            </li>
        </ol>
    </body>
</html>
```

```
        item 1 item 1 item 1.  
    </p>  
    </li>  
    <li>  
        <p>  
            item 2 item 2 item 2.  
            item 2 item 2 item 2.  
        </p>  
    </li>  
    <li>  
        <p>  
            item 3 item 3 item 3.  
            item 3 item 3 item 3.  
        </p>  
    </li>  
</ol>  
</body>  
</html>
```

---

## Gambar

Selain teks, halaman web juga berisi gambar. Untuk menambahkan gambar pada halaman web ada beberapa hal yang harus dipersiapkan. Hal yang perlu dipersiapkan adalah folder tempat menyimpan file-file gambar. Lokasi folder dan file HTML harus berada di folder yang sama, seperti pada gambar di bawah ini. Folder dengan nama “images” akan digunakan untuk menyimpan file-file gambar.

Name	Date modified	Type
images	2/8/2020 7:30 PM	File folder
heading_1.html	2/6/2020 9:36 AM	HTML File
heading_2.html	2/6/2020 6:37 PM	HTML File
heading_3.html	2/7/2020 6:40 PM	HTML File
hello_world.html	2/7/2020 9:06 PM	HTML File

*Gambar 24. Lokasi folder dan file HTML.*

Di dalam folder images dapat dibuat lagi folder-folder lain jika diperlukan, misal jika ingin menyimpan gambar-gambar icon maka dapat dibuat folder "icons".

Setelah folder dibuat, simpan file gambar ke dalam folder tersebut, misal nama file yang simpan adalah profile.jpg.

Untuk menampilkan gambar pada halaman web digunakan tag <img />. Tag ini umumnya ditulis dengan dengan satu bagian saja, bukan dua bagian (tang pembuka dan penutup) seperti tag-tag sebelumnya. Sintaks tag adalah sebagai berikut.

```
<img scr="nama_file" width="ukuran_pixel" height="ukuran_pixel" ... >
```

Atribut-atribut yang dimiliki tag ini adalah:

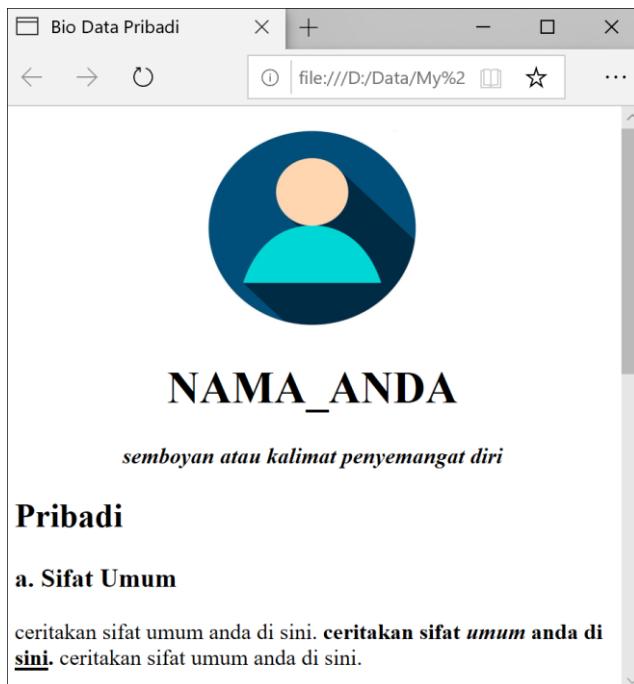
- scr, atribut ini untuk menentukan file yang akan ditampilkan.
- width, atribut ini untuk menentukan ukuran lebar gambar. Jika atribut ini tidak digunakan maka secara otomatis akan digunakan ukuran asli dari file gambar tersebut.
- height, atribut ini untuk menentukan ukuran tinggi gambar. Atribut ini dapat tidak digunakan seperti atribut width.

Selain ketiga atribut tersebut masih ada atribut-atribut lain.

### Perhatian

Dalam membuat halaman web harus diperhatikan ukuran file gambar. Diharapakan tidak menggunakan ukuran file gambar yang besar untuk membuat antarmuka web. Misal jika antarmuka menggunakan 20 file gambar dan 1 file gambar berukuran 1MB atau lebih maka ketika halaman web diakses memerlukan waktu yang lama. Karena ukuran total sebuah halaman web beserta gambarnya menjadi 20MB.

## Latihan 1



Gambar 25. Menambahkan gambar pada halaman web.

Buat kode HTML untuk menambahkan sebuah gambar pada halaman web seperti pada gambar di atas. Gunakan file HTML sebelumnya dan simpan kembali file tersebut dengan nama profile.html. Posisi gambar berada di atas NAMA\_ANDA dan di tengah halaman web.

Berikut adalah kode HTML untuk membuat antarmuka di atas.

```
profile.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

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

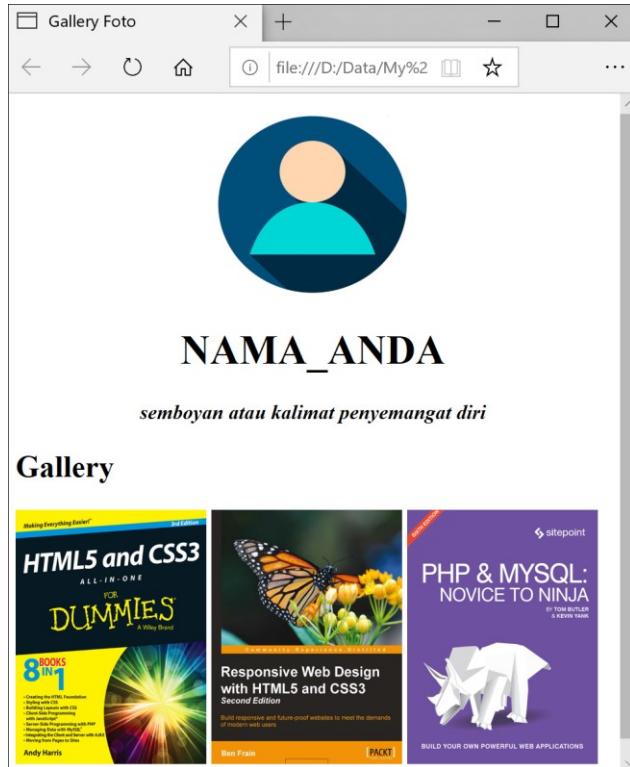
```
<h1>NAMA_ANDA</h1>
<p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
</center>
. . .
</body>
</html>
```

Dari kode di atas dapat dilihat bahwa file berada di dalam folder “images” dan nama filenya adalah profile.jpg. Ukuran dari file yang ditampilkan adalah 150 x 150 pixel. Walaupun ukuran file sebenarnya adalah 1000 x 1000 pixel.

## Latihan 2

Latihan berikutnya adalah membuat halaman web untuk gallery foto. Nama file dari halaman web ini adalah gallery.html. Antarmuka halaman ini dapat dilihat pada Gambar 26.

Setiap gambar ditampilkan dengan ukuran 150 x 200 pixel. Jumlah gambar yang ditampilkan adalah tiga gambar.



Gambar 26. Halaman web untuk menampilkan gallery foto.

Berikut ada kode HTML untuk membuat antarmuka di atas.

```
gallery.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Gallery Foto</title>
  </head>

  <body>
    <center>
      
      <h1>NAMA_ANDA</h1>
      <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
    </center>
    <h2>Gallery</h2>
    <p>
      
      
      
    </p>
  </body>
</html>
```

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

### Latihan 3

Atribut lain yang perlu diketahui adalah alt. Atribut ini berfungsi untuk memberikan keterangan pada gambar. Pemberian atribut ini juga berguna pada mesin pencarian gambar yang dimiliki oleh Google. Berikut adalah contoh penggunaan atribut ini. Gunakan file gallery.html kemudian tambahkan atribut alt dan berikan nilainya seperti contoh di bawah ini.

```
  
  
  
  

```

## Hyperlink

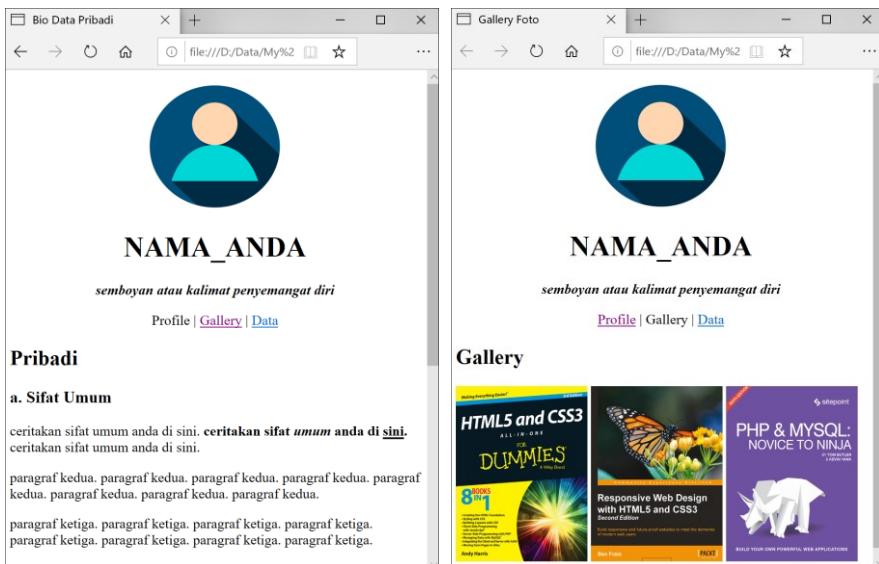
Pada halaman web, kita sering melihat ada teks atau gambar yang dapat diklik dan kemudian menghantarkan ke halaman lain. Hal tersebut dikenal dengan istilah hyperlink. Untuk membuat teks atau gambar menjadi hyperlink digunakan tag <a>.

Sintaks dari tag ini adalah sebagai berikut.

```
<a href="tujuan">teks atau gambar</a>
```

### Latihan 1

Gunakan halaman profile.html dan gallery.html untuk membuat hyperlink.



**Gambar 27. Membuat hyperlink untuk menghubungkan dua halaman web.**

Tambahkan tag <a> sehingga dimiliki tampilan seperti Gambar 27. Halaman web di sebelah kiri adalah profile.html. Pada halaman ini dapat dilihat menu sederhana yaitu:

Profile | Gallery | Data

Pada halaman ini Profile tidak bisa diklik. Sedangkan Gallery dan Data adalah hyperlink yang dapat diklik. Tujuan hyperlink Gallery ketika diklik adalah file gallery.html sedangkan tujuan hyperlink Data adalah data.html.

Sedangkan halaman web di sebelah kanan adalah gallery.html. Pada halaman ini Gallery tidak bisa diklik sedangkan Profile dan Data adalah hyperlink yang dapat diklik.

Berikut adalah kode HTML yang mesti ditambahkan pada halaman profile.html. Pada kode di bawah ini dapat dilihat yang menggunakan tag <a> hanya teks Gallery dan Data saja.

```
profile.html
<!DOCTYPE HTML>
<html>
  <head>
```

```

<title>Bio Data Pribadi</title>
</head>

<body>
    <center>
        
        <h1>NAMA_ANDA</h1>
        <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
        <p>
            Profile | 
            <a href="gallery.html">Gallery</a> | 
            <a href="data.html">Data</a>
        </p>
    </center>
    . . .
</body>
</html>

```

Berikut adalah kode HTML yang mesti ditambahkan pada halaman gallery.html. Pada kode di bawah ini dapat dilihat yang menggunakan tag [hanya teks Profile dan Data saja.](#)

```

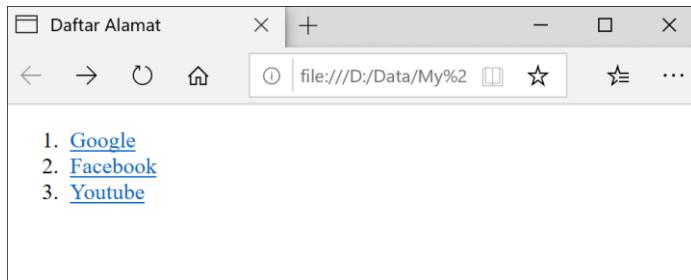
gallery.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Gallery Foto</title>
    </head>

    <body>
        <center>
            
            <h1>NAMA_ANDA</h1>
            <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
            <p>
                <a href="profile.html">Profile</a> | 
                Gallery | 
                <a href="data.html">Data</a>
            </p>
        </center>
        <h2>Gallery</h2>
        <p>
            
            
            
        </p>
    </body>
</html>

```

## Latihan 2

Pada tag <a>, telah disebutkan bahwa atribut href berfungsi untuk menentukan tujuan. Pada contoh di atas tujuannya adalah sebuah file HTML. Tujuan juga bisa berupa file lain seperti gambar dan lain-lain. Selain itu tujuan juga bisa berupa alamat web seperti [www.google.com](http://www.google.com) atau [www.facebook.com](http://www.facebook.com) dan lain-lain.



Gambar 28. Hyperlink ke alamat web.

Berikut ini contoh kode membuat hyperlink untuk mengakses alamat web.

```
link_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Daftar Alamat</title>
  </head>

  <body>
    <ol>
      <li><a href="http://www.google.com">Google</a></li>
      <li>
        <a href="http://www.facebook.com" target="_blank">
          Facebook
        </a>
      </li>
      <li><a href="http://www.youtube.com">Youtube</a></li>
    </ol>
  </body>
</html>
```

Pada kode di atas, diperkenalkan atribut target pada tag <a>. Atribut target berfungsi untuk menentukan dimana dokumen/alamat web dibuka. Pada

contoh di atas digunakan target="\_blank" yang artinya alamat web akan dibuka pada window atau tab baru.

Nilai atribut target yang lain adalah:

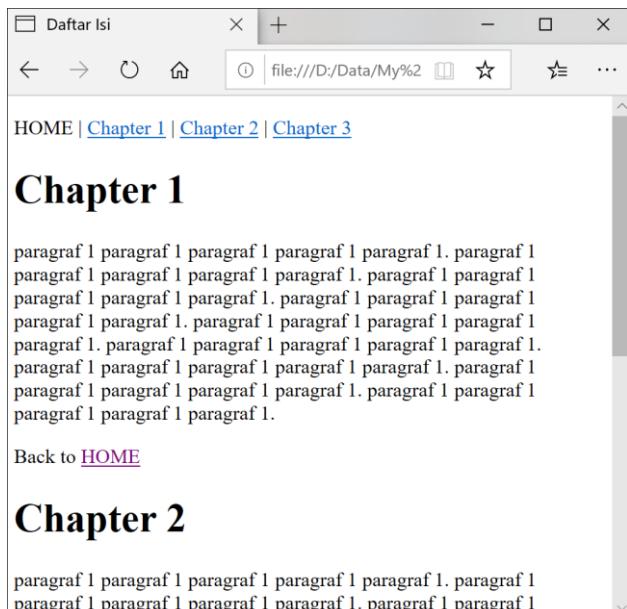
- \_self, adalah nilai default. Jadi jika tag <a> tidak menggunakan atribut target maka secara default nilai atribut ini adalah \_self. Nilai ini berarti dokumen akan dibuka pada window atau tab yang sama.
- \_parent, nilai ini berfungsi untuk membuka dokumen pada parent frame.
- \_top, nilai ini bertujuan agar dokumen dibuka pada seluruh bagian dari window.
- framename, berfungsi untuk membuka dokumen pada frame dengan nama ini.

Untuk atribut dengan nilai \_parent, \_top dan framename dapat dicoba jika halaman web menggunakan frame. Saat ini sudah jarang implementasi frame pada halaman web karena dianggap tertinggal.

### **Latihan 3**

Selain contoh-contoh di atas, hyperlink dengan tag <a> dapat dipergunakan untuk mengantarkan user pada area tertentu yang sudah ditandai pada sebuah halaman. Contohnya seperti daftar isi pada dokumen MS Word, dimana saat user mengklik salah satu link pada daftar isi (misal: Chapter 2) maka secara otomatis user tersebut diantarkan ke halaman yang berisi Chapter 2 tersebut.

Pada Gambar 29 dapat dilihat menu pada bagian atas kiri. Jika hyperlink Chapter 1 diklik maka layar akan bergeser ke bawah menuju bagian Chapter 1 yang sudah ditandai. Begitu juga jika hyperlink Chapter 2 dan Chapter 3 diklik. Pada halaman ini juga dapat dilihat hyperlink HOME disetiap akhir paragraf. Jika hyperlink ini diklik maka layar akan dikembalikan pada posisi teratas.



*Gambar 29. Hyperlink di dalam halaman web.*

Untuk menandai tujuan juga digunakan tag `<a>` namun tidak menggunakan atribut href. Untuk menandai tujuan digunakan atribut name seperti contoh berikut ini.

```
<h1><a name="chapter1">Chapter 1</a></h1>
```

Pada kode di atas dapat dilihat Chapter 1 ditandai dan diberi nilai name="chapter1". Kemudian pada menu, untuk menuju bagian yang telah ditandai cukup dengan memberikan nilai atribut name tersebut pada atribut href seperti contoh berikut ini. Bedanya nilai atribut name yang telah diberikan sebelumnya mesti ditambahkan tanda "#".

```
<a href="#chapter1">Chapter 1</a>
```

Kode HTML lengkap dari halaman di atas adalah sebagai berikut.

```
link_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Daftar Isi</title>
  </head>
```

```
<body>
  <p>
    <a name="home">HOME</a> |
    <a href="#chapter1">Chapter 1</a> |
    <a href="#chapter2">Chapter 2</a> |
    <a href="#chapter3">Chapter 3</a>
  </p>

  <h1><a name="chapter1">Chapter 1</a></h1>
  <p>
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
  </p>
  <p>
    Back to <a href="#home">HOME</a>
  </p>

  <h1><a name="chapter2">Chapter 2</a></h1>
  <p>
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
  </p>
  <p>
    Back to <a href="#home">HOME</a>
  </p>

  <h1><a name="chapter3">Chapter 3</a></h1>
  <p>
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
    paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.
```

```
paragraf 1 paragraf 1 paragraf 1 paragraf 1 paragraf 1.  
</p>  
<p> Back to <a href="#home">HOME</a>  
</p>  
</body>  
</html>
```

## Table

Tabel sering kita lihat ketika menggunakan aplikasi spreadsheet seperti MS Excel. Umumnya tabel digunakan untuk menampilkan data. Sama halnya seperti pada halaman web, tabel juga digunakan untuk menampilkan data baik data teks atau gambar.

Untuk membuat tabel digunakan beberapa tag HTML yang saling berhubungan. Untuk membuat sebuah tabel yang lengkap diperlukan tag-tag berikut ini:

- <table>
- <tr> digunakan untuk membuat baris.
- <th> digunakan untuk membuat kolom header.
- <td> digunakan untuk membuat kolom data.

Berikut adalah contoh tabel yang umum ditemui.

No	Nama	Email	Kota
1	Budi	budi@ulm.ac.id	Banjarmasin
2	Wati	wati@ulm.ac.id	Banjarbaru

Dari tabel di atas dapat dilihat bahwa tabel memiliki 3 kolom dan 3 baris. Dimana baris pertama dikenal dengan istilah header.

Untuk mengetahui bagaimana cara menulis kode HTML, maka akan diberikan contoh tabel yang sangat sederhana yaitu tabel yang terdiri atas 1 baris dan 1 kolom saja seperti tabel di bawah ini.

### Tabel Sangat Sederhana

Kode HTML untuk membuat tabel tersebut adalah sebagai berikut.

```
table_1.html  
<!DOCTYPE HTML>  
<html>
```

```

<head>
    <title>Table</title>
</head>

<body>
    <table border="1" width="100%">
        <tr>
            <td>Tabel Sangat Sederhana</td>
        </tr>
    </table>
</body>
</html>

```

Dari contoh kode HTML di atas dapat dilihat penggunaan tag `<table></table>` dimana didalamnya ditambahkan tag `<tr></tr>` untuk membuat row (baris). Namun sebuah baris yang dibuat dengan tag `<tr></tr>` belum terlihat hasilnya jika didalamnya belum ada minimal 1 kolom yang dibuat dengan tag `<td></td>`. Dan setiap kolom sebaiknya memiliki nilai bisa berupa teks atau gambar. Namun jika tidak ingin menampilkan teks atau gambar, hanya kolom kosong saja maka dapat menggunakan “&nbsp;” yang sama nilainya dengan spasi.

Contoh selanjutnya adalah membuat tabel yang terdiri atas dua baris dengan satu kolom seperti tabel di bawah ini.

Baris 1
Baris 2

Kode HTML untuk membuat tabel tersebut adalah sebagai berikut.

Baris 1
Baris 2

```

table_2.html

<!DOCTYPE HTML>
<html>
    <head>
        <title>Table</title>
    </head>

    <body>
        <table border="1" width="100%">
            <tr>
                <td>Baris 1</td>
            </tr>
            <tr>
                <td>Baris 2</td>
            </tr>
        </table>
    </body>

```

```
</html>
```

Contoh selanjutnya adalah membuat tabel dengan dua kolom dan 1 baris seperti tabel berikut ini.

Kolom 1	Kolom 2
---------	---------

Kode HTML yang ditulis untuk membuat tabel tersebut adalah sebagai berikut.

```
table_3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Table</title>
  </head>

  <body>
    <table border="1" width="100%">
      <tr>
        <td>Kolom 1</td>
        <td>Kolom 2</td>
      </tr>
    </table>
  </body>
</html>
```

### Latihan 1

Buat tabel yang memiliki header seperti berikut ini. Pada header, teks dibuat cetak tabel.

No	Nama	Email	Kota
1	Budi	budi@ulm.ac.id	Banjarmasin
2	Wati	wati@ulm.ac.id	Banjarbaru

Kode HTML untuk membuat tabel di atas adalah sebagai berikut ini.

```
table_4.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Daftar Pegawai</title>
  </head>

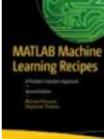
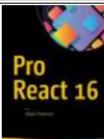
  <body>
    <table border="1" width="100%">
```

```
<tr>
    <th>No</th>
    <th>Nama</th>
    <th>Email</th>
    <th>Kota</th>
</tr>
<tr>
    <td>1</td>
    <td>Budi</td>
    <td>budi@ulm.ac.id</td>
    <td>Banjarmasin</td>
</tr>
<tr>
    <td>2</td>
    <td>Wati</td>
    <td>wati@ulm.ac.id</td>
    <td>Banjarbaru</td>
</tr>
</table>
</body>
</html>
```

Dengan menggunakan tag `<th>` maka secara otomatis teks berada pada tag tersebut akan dicetak tebal.

## Latihan 2

Latihan berikutnya adalah membuat tabel seperti pada Gambar 30. Perhatikan bahwa gambar pada kolom harus pada posisi di tengah kolom. Sedangkan angka pada kolom jumlah harus pada posisi rata kanan.

No	Gambar	Judul	Jumlah
1		Illustrated C#	5
2		MATLAB Machine Learning Recipes	10
3		Pro React 16	1

*Gambar 30. Tabel daftar buku.*

Berikut adalah kode HTML untuk membuat antarmuka dari tampilan di atas.

```
table_5.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Daftar Buku</title>
  </head>

  <body>
    <table border="1" width="100%">
      <tr>
        <th>No</th>
        <th>Gambar</th>
        <th>Judul</th>
        <th>Jumlah</th>
      </tr>
      <tr>
        <td>1</td>
        <td align="center">
          
        </td>
        <td>Illustrated C#</td>
        <td align="right">5</td>
      </tr>
      <tr>
```

```

        <td>2</td>
        <td align="center">
            
        </td>
        <td>MATLAB Machine Learning Recipes</td>
        <td align="right">10</td>
    </tr>
    <tr>
        <td>3</td>
        <td align="center">
            
        </td>
        <td>Pro React 16</td>
        <td align="right">1</td>
    </tr>
</table>
</body>
</html>
```

Pada kode di atas dapat dilihat penggunaan atribut align pada tag `<td>`. Atribut tersebut berfungsi untuk menentukan posisi teks atau gambar pada kolom. Jika nilai atribut `align="center"` artinya teks atau gambar adalah rata tengah. Sedangkan jika atribut `align="right"` artinya teks atau gambar adalah rata kanan.

### Latihan 3

Latihan berikutnya adalah membuat tabel seperti berikut ini.

Data	
Januari	Februari
12	13
13	14

Tabel ini terdiri atas dua kolom dan memiliki dua header. Header yang pertama adalah "Data" (baris pertama). Sedangkan header kedua adalah "Januari" dan "Februari" (baris kedua). Header pertama merupakan gabungan antara dua kolom.

Untuk membuat tabel seperti ini digunakan kode HTML sebagai berikut ini.

```

table_6.html
<!DOCTYPE HTML>
<html>
<head>
```

```
<title>Data</title>
</head>

<body>
    <table border="1" width="100%">
        <tr>
            <th colspan="2">Data</th>
        </tr>
        <tr>
            <th>Januari</th>
            <th>Februari</th>
        </tr>
        <tr>
            <td>12</td>
            <td>13</td>
        </tr>
        <tr>
            <td>13</td>
            <td>12</td>
        </tr>
    </table>
</body>
</html>
```

Pada kode di atas dapat diihat ada dua baris (yaitu baris pertama dan kedua) yang menggunakan tag `<th>` di dalam tag `<tr>`. Ini berarti ada dua baris yang menjadi header.

```
<tr>
    <th colspan="2">Data</th>
</tr>
<tr>
    <th>Januari</th>
    <th>Februari</th>
</tr>
```

Kemudian pada tag `<tr>` pada baris pertama hanya menggunakan satu tag `<th>` saja. Namun karena tabel ini adalah terdiri atas dua kolom, jika pada baris pertama ingin menjadi satu kolom saja maka kedua kolom dapat digabung dengan cara menggunakan atribut `colspan` dengan nilai adalah 2. Artinya untuk menggabungkan kolom dapat digunakan atribut `colspan`.

Contoh yang lain adalah seperti tabel di bawah ini.

Data		
Januari	Februari	Maret
1	2	3
4	5	6

Maka kode HTML tabelnya adalah sebagai berikut.

```
table_7.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Data</title>
  </head>

  <body>
    <table border="1" width="100%">
      <tr>
        <th colspan="3">Data</th>
      </tr>
      <tr>
        <th>Januari</th>
        <th>Februari</th>
        <th>Maret</th>
      </tr>
      <tr>
        <td>1</td>
        <td>2</td>
        <td>3</td>
      </tr>
      <tr>
        <td>4</td>
        <td>5</td>
        <td>6</td>
      </tr>
    </table>
  </body>
</html>
```

#### Latihan 4

Pada latihan ini akan diberikan cara untuk membuat tabel seperti berikut ini. tabel di bawah ini adalah tabel dengan 4 baris dan 2 kolom. Namun baris pertama dan kedua pada kolom pertama digabung dan didalamnya diberi nilai "Bulan". Sedangkan pada kolom kedua masih tetap memiliki dua baris yang keduanya adalah header.

Bulan	Data
<b>Curah Hujan</b>	
Januari	1
Februari	2

Dengan penjelasan dan gambar di atas maka kode HTML yang mesti ditulis adalah sebagai berikut.

```
table_8.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Data</title>
  </head>

  <body>
    <table border="1" width="100%">
      <tr>
        <th rowspan="2">Bulan</th>
        <th>Data</th>
      </tr>
      <tr>
        <th>Curah Hujan</th>
      </tr>
      <tr>
        <td>Januari</td>
        <td>1</td>
      </tr>
      <tr>
        <td>Februari</td>
        <td>2</td>
      </tr>
    </table>
  </body>
</html>
```

Mirip dengan tabel di atas, namun baris yang digabung adalah baris pada kolom kedua seperti pada tabel di bawah ini.

Daftar	Data Curah Hujan
Bulan	
Januari	1
Februari	2

Berikut adalah kode HTML untuk membuat kode tersebut.

```
table 9.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Data</title>
  </head>

  <body>
    <table border="1" width="100%">
      <tr>
        <th>Daftar</th>
        <th rowspan="2">Data Curah Hujan</th>
      </tr>
      <tr>
        <th>Bulan</th>
      </tr>
      <tr>
        <td>Januari</td>
        <td>1</td>
      </tr>
      <tr>
        <td>Februari</td>
        <td>2</td>
      </tr>
    </table>
  </body>
</html>
```

Pada kode di atas dapat dilihat atribut rowspan digunakan pada kolom kedua pada baris pertama. Sedangkan pada baris kedua tetap digunakan sebuah tagl  saja di dalam tag |. |  | |

## Latihan 5

Untuk lebih mengerti penggunaan atribut rowspan dapat melihat gambar dan contoh kode HTML berikut ini.

Header 1	Header 3	Header 4
Header 2		Header 5
1	2	3
4	5	6

Berikut adalah kode HTML untuk membuat tabel di atas.

```
table 10.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Data</title>
  </head>

  <body>
    <table border="1" width="100%">
      <tr>
        <th>Header 1</th>
        <th rowspan="2">Header 3</th>
        <th>Header 4</th>
      </tr>
      <tr>
        <th>Header 2</th>
        <th>Header 5</th>
      </tr>
      <tr>
        <td>1</td>
        <td>2</td>
        <td>3</td>
      </tr>
      <tr>
        <td>4</td>
        <td>5</td>
        <td>6</td>
      </tr>
    </table>
  </body>
</html>
```

---

## Form Input Data

Pada aplikasi web, salah satu hal yang penting adalah form input data. HTML memiliki tag-tag yang lengkap untuk membuat antarmuka untuk input data seperti pada Gambar 31.

## Contact Us

First Name

Last Name

Company

Phone Number  phone

Email address  email

Your Website Please include http://  url

When is the best time of day to reach you?

Morning

Evening

Would you like to receive our email newsletter?

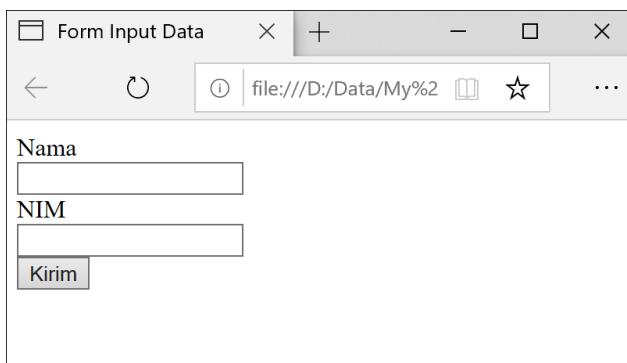
Sure!

*Gambar 31. Form input untuk halaman Contact Us.*

Dari gambar di atas dapat dilihat antarmuka untuk input data tidak berbeda dengan yang dimiliki oleh aplikasi desktop. Pada sub bab ini akan dijelaskan beberapa tag yang umum digunakan untuk membuat antarmuka form input.

## Form

Secara visual user hanya akan melihat elemen-elemen form seperti pada Gambar 31 dan pada Gambar 32. Namun sebenarnya sebelum tag elemen-elemen yang user lihat tersebut ditambahkan perlu ditambahkan atau digunakan tag <form>.



*Gambar 32. Form input data - Form - Contoh form sederhana.*

Tag <form> ini tidak terlihat secara visual pada halaman web namun sangat penting. Sintaks sederhana tag ini adalah sebagai berikut.

```
<form>
<!-- elemen form visual diketik di sini -->
```

```
</form>
```

Seperti dilihat di atas, tag ini termasuk tag yang memiliki tag pembuka dan penutup. Tag `<form>` juga memiliki atribut-atribut penting yang umum digunakan yaitu yang dapat dilihat pada kode di bawah ini.

```
<form action="tujuan" method="POST/GET">
    <!-- elemen form visual diketik di sini -->
</form>
```

Atribut `action` berfungsi untuk menentukan tujuan yang umumnya adalah nama file. Jika atribut `action` tidak digunakan atau nilai atribut `action` tidak diisi maka tujuannya adalah file halaman web itu sendiri. Misal kode HTML disimpan pada file `form_1.html`, jika atribut `action` tidak ditulis pada tag `form` atau nilai atribut `action` tidak diisi maka artinya tujuannya ke file `form_1.html`.

Atribut `method` dapat berisi sebuah nilai yaitu `POST` atau `GET` (walaupun masih ada metode yang lain). Hanya 1 nilai saja yang menjadi nilai dari atribut `method`. `POST` dan `GET` adalah cara bagaimana data dikirimkan atau ditrasfer oleh form sebagai client ke server. Jika menggunakan `POST`, nilai-nilai parameter yang dikirimkan berada di dalam badan pesan sedangkan jika menggunakan metode `GET` maka parameter dan nilainya dapat dilihat pada URL pada alamat web browser.

Sebagai contoh maka dapat dibuat kode HTML berikut ini.

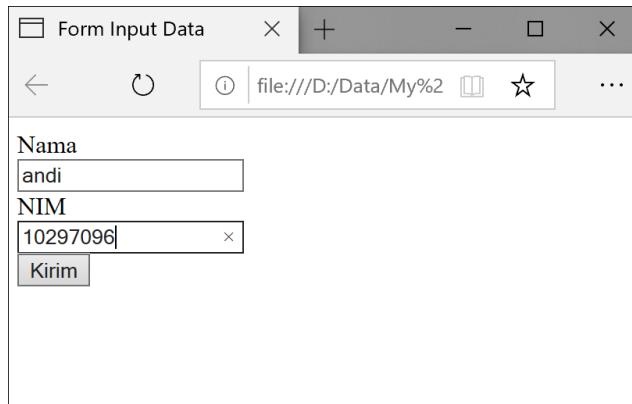
```
form_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Form Input Data</title>
    </head>

    <body>
        <form action="form_1.html" method="POST">
            Nama<br>
            <input type="text" name="nama"><br>
            NIM<br>
            <input type="text" name="nim"><br>
            <input type="submit" value="Kirim">
        </form>
    </body>
</html>
```

Tampilan kode HTML ini dapat dilihat pada Gambar 32. Halaman web ini menggunakan `method="POST"`, setelah kolom input diisi dan tombol Kirim

diklik maka dapat dilihat data akan dikirim ke alamat sesuai dengan nilai atribut action="form\_1.html".

Selanjutnya ganti nilai pada atribut method menjadi method="GET", kemudian buka kembali halaman form\_1.html. Kemudian isi kolom input Nama dan NIM seperti pada Gambar 33.



*Gambar 33. Form input data - form - isi nilai kolom input.*

Setelah tombol Kirim diklik, kemudian coba perhatikan sebagian URL pada alamat web browser akan terlihat seperti link berikut ini. Ini adalah cara method GET mengirim data yaitu dengan memanfaatkan URL.

```
.../source/bab_3_html/form_1.html?nama=andi&nim=10297096
```

Nama parameter di URL ini sesuai dengan atribut name yang ada pada tag <input> untuk kolom isian nama dan NIM, dimana untuk kolom isian nama digunakan atribut name="nama". Dan untuk kolom isian NIM digunakan atribut name="nim". Kedua kode tersebut dapat dilihat kembali di bawah ini.

```
...
<input type="text" name="nama">
...
<input type="text" name="nim">
...
```

Pada Gambar 33, diisi nilai "andi" pada kolom isian Nama dan nilai "10297096" pada kolom isian NIM. Sehingga dapat dilihat pada URL, parameter name=andi dan parameter nim=10297096.

Contoh-contoh penggunaan tag ini akan lebih banyak dibahas pada bab 6 PHP: Hypertext Preprocessor.

## Textbox

---

Textbox adalah nama yang umum digunakan untuk membuat kolom input untuk isian yang teks. Textbox umumnya digunakan untuk mengisi nilai seperti username, nama, alamat dan lain-lain.

Tag yang digunakan untuk membuat textbox adalah <input>. Sebenarnya tag ini dapat digunakan untuk membuat berbagai bentuk seperti radiobutton dan tombol. Untuk menggunakan tag ini sebagai textbox digunakan sintaks sebagai berikut.

```
<input type="text" name=" nilai_" id="nilai_">
```

Pada sintaks di atas terdapat tiga atribut penting. Yang pertama adalah atribut type. Untuk membuat tag <input> menjadi textbox harus digunakan nilai "text" sebagai nilai atribut type.

Atribut kedua adalah name, atribut ini berfungsi untuk memberikan nama kepada tag ini sekaligus menjadi nama parameter pada pengiriman data. dengan mengetahui nama parameter maka dapat diketahui nilai yang diisikan oleh user untuk parameter ini.

Atribut ketiga adalah id, atribut ini berfungsi untuk memberi identitas unik pada tag. Artinya tidak ada nilai atribut id yang sama digunakan pada sebuah halaman web. Umumnya nilai pada atribut id ini digunakan oleh pemrograman sisi client (client-side programming seperti Javascript) untuk mengkases tag. Penggunaan atribut ini akan diterangkan lebih lanjut pada bab 5 JavaScript.

Berikut adalah contoh dari penggunaan tag <input> untuk membuat textbox.

```
textbox_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Textbox</title>
  </head>

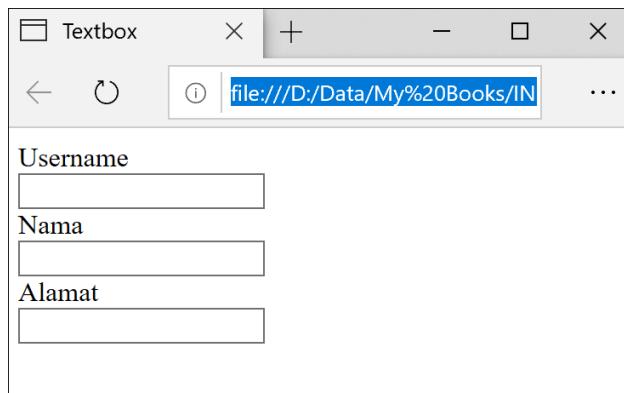
  <body>
    <form>
```

```

Username<br>
<input type="text" name="username" id="textbox_username">
<br>
Nama<br>
<input type="text" name="nama" id="textbox_nama"><br>
Alamat<br>
<input type="text" name="alamat" id="textbox_alamat">
</form>
</body>
</html>

```

Hasilnya dapat dilihat pada gambar di bawah ini. Textbox ini memiliki ukuran yang sama dan ditujukan untuk isian singkat saja. Untuk menentukan ukuran, warna dan lain-lain akan diberikan pada bahasan tentang Cascading Style Sheets (CSS).



*Gambar 34. Form input data - Textbox - Contoh 1.*

Pada kode di atas dapat dilihat untuk textbox Username menggunakan tag `<input>` dengan atribut `name="username"` yang artinya parameter data yang dikirimkan ke server adalah `username`. Sedangkan atribut `id="textbox_username"`. Pemberian nilai pada atribut `id` ini adalah bebas, namun harapannya nilainya dapat mencerminkan tipe antarmuka form (textbox, button, radiobutton dan lain-lain).

## Tombol

---

Button atau tombol adalah hal penting pada sebuah form. Umumnya tombol digunakan untuk antarmuka yang diklik untuk mengirimkan data. Untuk membuat tombol juga digunakan tag `<input>` namun atribut yang digunakan adalah "submit", "reset" atau "button".

Sintaks yang digunakan untuk membuat tombol adalah sebagai berikut.

```
<input type="tipe_tombol" value="text_pada_tombol">
```

Pada sintaks di atas dapat dilihat atribut yang umum digunakan yaitu atribut type. Nilai untuk atribut type yang dapat menghasilkan bentuk tombol adalah:

- submit.
- reset.
- button.

Ketiganya menghasilkan tombol dengan bentuk yang sama, tetapi mempunyai cara kerja yang berbeda.

Atribut value berfungsi untuk memberikan teks yang dapat dilihat pada tombol.

Berikut adalah contoh penggunaan ketiga atribut type tersebut.

```
button_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Tombol</title>
    </head>

    <body>
        <form method="POST">
            Username<br>
            <input type="text" name="username" id="textbox_username">
            <br>
            Nama<br>
            <input type="text" name="nama" id="textbox_nama"><br>
            Alamat<br>
            <input type="text" name="alamat" id="textbox_alamat">
            <br><br>
            <input type="submit" value="simpan">
            <input type="button" value="hapus">
            <input type="reset" value="reset">
        </form>
    </body>
</html>
```

Hasilnya dapat dilihat pada gambar di bawah ini.

The screenshot shows a window titled "Tombol". Inside, there are three text input fields with labels: "Username", "Nama", and "Alamat". Each label has a corresponding empty text box below it. At the bottom of the window are three buttons: "simpan" (highlighted with a blue border), "hapus", and "reset".

*Gambar 35. Form input data - Tombol - Contoh 1.*

Dari gambar di atas dapat dilihat fungsi atribut value. Pada tombol pertama digunakan value="simpan" sehingga teks pada tombol pertama adalah "simpan". Sedangkan pada tombol kedua digunakan value="hapus" dan "reset".

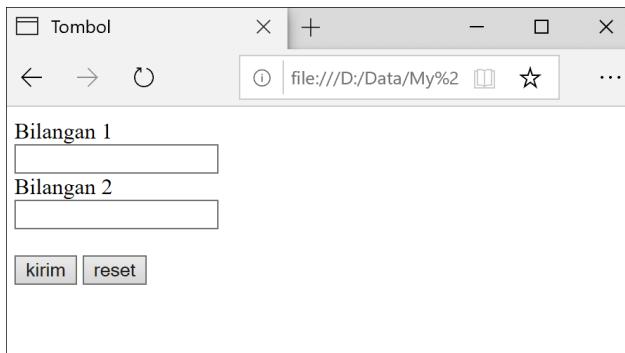
Penggunaan tag input dengan nilai atribut adalah submit di dalam form akan mempengaruhi kerja form tersebut secara keseluruhan. Tombol tipe ini ketika diklik akan secara otomatis melakukan proses pengiriman data. Selain itu jika user mengisi salah satu elemen input seperti textbox kemudian menekan tombol enter maka secara otomatis pula proses pengiriman data dilakukan. Ini yang dimaksud dengan tombol yang menggunakan tag input tipe submit ini mempengaruhi kerja form secara keseluruhan.

Berbeda jika dibandingkan dengan tag input yang menggunakan atribut adalah button. Tombol tipe ini tidak memberikan aksi apa-apa ketika diklik. Jika pada form hanya memiliki tombol tipe ini maka tidak akan ada proses pengiriman data ketika user menekan tombol enter saat telah mengisi kolom inputan. Tombol tipe ini hanya akan memberikan reaksi ketika tombol ini diberikan penanganan event (event handling). Penanganan event pada tombol tipe ini akan dibahas pada pembahasan tentang client-side programming di bab 5 JavaScript.

Tipe tombol selanjutnya adalah yang menggunakan atribut dengan nilai adalah reset. Sesuai dengan tipenya, tombol ini berfungsi untuk mereset form menjadi semula. Artinya setelah user mengisi nilai-nilai pada kolom inputan dan kemudian tombol tipe ini diklik maka setiap nilai-nilai yang tersebut akan dihapus.

## Latihan 1

Pada latihan ini akan diberikan contoh penggunaan input tipe submit dan reset. Dengan antarmuka sebagai berikut.

A screenshot of a web browser window titled "Tombol". The address bar shows the URL "file:///D:/Data/My%2". The main content area contains a form with two text input fields labeled "Bilangan 1" and "Bilangan 2". Below the inputs are two buttons: "kirim" and "reset". The browser interface includes standard controls like back, forward, and search.

Gambar 36. Form input data - Tombol - Latihan 1.

Dengan kode HTML seperti berikut ini.

```
button_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Tombol</title>
    </head>

    <body>
        <form method="GET">
            Bilangan 1<br>
            <input type="text" name="bilangan1" inputmode="numeric"
                   id="textbox_bilangan1">
            <br>
            Bilangan 2<br>
            <input type="text" name="bilangan2" inputmode="numeric"
                   id="textbox_bilangan2">
            <br><br>
            <input type="submit" value="simpan">
            <input type="reset" value="reset">
        </form>
    </body>
</html>
```

Jalankan halaman web di atas pada web browser, kemudian isi kolom input Bilangan 1 dan Bilangan 2. Ketika cursor masih di kolom input Bilangan 2, tekan tombol enter. Maka dapat dilihat ada proses pengiriman data, yang buktinya dapat dilihat dengan cara melihat perubahan pada URL (lihat lagi pembahasan pada sub bab Form).

Pada kode di atas juga dapat dilihat penggunaan atribut inputmode pada tag input untuk membuat textbox. Nilai pada atribut tersebut adalah numeric, dimaksudnya agar ketika user mengetik nilai pada kolom input ini akan ditampilkan keyboard numerik saja. Tapi hal ini hanya akan terlihat ketika diakses pada perangkat mobile.

## Latihan 2

Pada latihan ini diberikan contoh penggunaan input tipe button. Berikut adalah antarmukanya.

The screenshot shows a browser window titled "Registrasi Form". The address bar shows "file:///D:/Data/My%2". The main content is a form titled "Registrasi User". It contains three input fields: "Username", "Password", and "Email", each with a placeholder value. Below the fields are two buttons: "register" and "batal".

Gambar 37. Form input data - Tombol - Registrasi form.

Dan berikut ini adalah kode HTML dari antarmuka di atas.

```
button_3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Tombol</title>
  </head>

  <body>
    <form method="GET">
      <h3>Registrasi User</h3>
      Username<br>
      <input type="text" name="username" id="textbox_username">
      <br>
      Password<br>
      <input type="password" name="password"
             id="textbox_password">
      <br>
```

```

Email<br>
<input type="email" name="email" id="textbox_email">
<br><br>
<input type="button" value="register">
<input type="reset" value="batal">
</form>
</body>
</html>

```

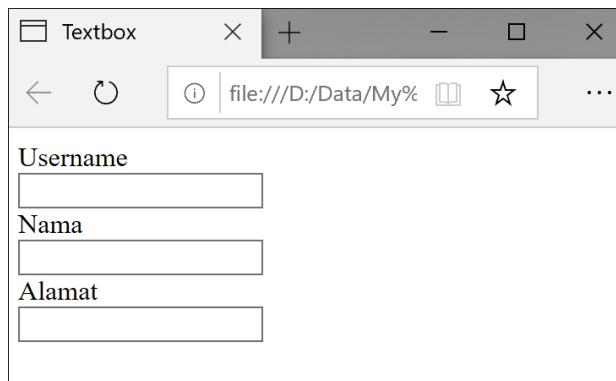
Buka halaman web ini, dan lakukan hal yang sama seperti pada Latihan 1. Dan perhatikan apakah ada perubahan pada URL ketika tombol enter ditekan atau tombol register ditekan.

Pada kode di atas juga diperkenalkan input dengan type="password" dan type="email". Input tipe password umum digunakan sejak HTML versi awal dan dipastikan dapat dikenali oleh web browser baru dan lama. Sedangkan input tipe email belum tentu dikenali oleh web browser lama.

## Label

---

Label adalah teks yang biasanya digunakan untuk menjelaskan elemen form. Sebagai contoh, sebelum kolom input seperti username, nama dan alamat pada gambar di bawah ini.



*Gambar 38. Form input data - Label - Contoh label pada form.*

Pada contoh sebelumnya untuk membuat teks tersebut hanya ditulis teks biasa tanpa menggunakan tag <label>. Tag ini memiliki beberapa manfaat jika digunakan, yang pertama adalah kemudahan untuk memberikan style pada teks. Yang kedua adalah memudahkan user mengakses elemen form hanya dengan mengklik labelnya saja.

Sintaks untuk membuat label adalah sebagai berikut.

```
<label for="id_elelen_tujuan">tulis Label di sini</label>
<input type="text" name="username" id="id_elelen_tujuan">
```

Pada kode di atas dapat dilihat bahwa tag <label> adalah jenis tag yang memiliki tag pembuka dan penutup. Teks yang menjadi label keterangan elemen form diletakkan diantara kedua tag pembuka dan penutup tersebut. Tag ini memiliki atribut for yang berfungsi untuk menunjuk elemen form mana yang diterangkan oleh label. Isi dari atribut ini adalah nilai atribut id dari elemen form tersebut.

Berikut adalah contoh kode HTML yang menggunakan tag ini.

```
label_1.html

<!DOCTYPE HTML>
<html>
  <head>
    <title>Label</title>
  </head>

  <body>
    <form>
      <label for="textbox_username">Username</label><br>
      <input type="text" name="username" id="textbox_username">
      <br>
      <label for="textbox_nama">Nama</label><br>
      <input type="text" name="nama" id="textbox_nama"><br>
      <label for="textbox_alamat">Alamat</label><br>
      <input type="text" name="alamat" id="textbox_alamat">
    </form>
  </body>
</html>
```

Simpan kode HTML di atas ke dalam file label\_1.html, kemudian buka pada web browser. Kemudian klik label Username, Nama atau Alamat. Maka dapat dilihat bahwa cursor langsung ditempat pada elemen form yang sesuai dengan tujuan label-label tersebut.

## Radio Button

---

Tag ini digunakan untuk membuat antarmuka yang memungkinkan user memilih hanya satu opsi dari beberapa opsi yang diberikan. Sebagai contoh jenis kelamin, pekerjaan dan lain-lain.

Untuk membuat radio button digunakan tag <input> dengan atribut type="radio". Berikut adalah sintaks dari tag ini.

```
<input type="radio" name="name_" value="nilai_1" id="id_>
<input type="radio" name="name_" value="nilai_2" id="id_>
```

Untuk membuat radio button, diperlukan tag <input type="radio"/> lebih dari satu seperti sintaks di atas. Minimal diperlukan dua elemen ini. Walaupun terdiri atas dua elemen atau lebih, semua elemen ini merupakan satu kesatuan, sehingga nilai atributnya harus sama. Kemudian atribut value berisi nilai yang akan dikirimkan. Nilai atribut id juga harus berisi nilai yang berbeda seperti yang dijelaskan sebelumnya.

Berikut adalah contoh untuk membuat radio button.

```
radiobutton 1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Radio Button</title>
    </head>

    <body>
        <form method="GET">
            <label for="textbox_nama">Nama</label> <br>
            <input type="text" name="nama" id="textbox_nama">
            <br><br>

            Jenis Kelamin<br>
            <input type="radio" name="jenis_kelamin" value="pria"
                   id="radio_jk_pria">
            <label for="radio_jk_pria">Pria</label>
            <br>

            <input type="radio" name="jenis_kelamin" value="wanita"
                   id="radio_jk_wanita">
            <label for="radio_jk_wanita">Wanita</label>

            <br><br>
            <input type="submit" value="kirim data">
            <input type="reset" value="batal">
        </form>
    </body>
</html>
```

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

Pada contoh di atas atribut name kedua radio button memiliki nilai yang sama yaitu “jenis\_kelamin”. Ini menandakan keduanya merupakan satu kesatuan, artinya user hanya akan bisa memilih salah satu radio button ini.

Untuk percobaan, buka file ini pada web browser kemudian isi nama dan klik radio button Pria. Kemudian klik tombol “kirim data”. Dan perhatikan URL pada kolom address akan dapat dilihat sebagai berikut.

```
...bab_3_html/radiobutton_1.html?nama=adi&jenis_kelamin=pria
```

Kemudian isi kembali form dengan nilai yang berbeda, dan klik radio button Wanita. Kemudian kembali perhatikan URL pada kolom address.

```
...bab_3_html/radiobutton_1.html?nama=adi&jenis_kelamin=wanita
```

Hasil hasil ini dapat dilihat bahwa dengan menggunakan nilai atribut name yang sama untuk radio button maka nama parameter data yang dikirim juga saya yaitu “jenis\_kelamin”. Yang berbeda hanyalah nilai atribut value yang sesuai dengan pilihan user.

## Latihan 1

Pada latihan ini diperlihatkan penggunaan dua group radio button yaitu group jenis kelamin dan pekerjaan. Pada group jenis kelamin digunakan dua tag radio button. Sedangkan pada group pekerjaan digunakan tiga tag radio button. Berikut adalah kode HTML halaman web ini.

```
radiobutton_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Radio Button</title>
  </head>

  <body>
    <form method="GET">
      <label for="textbox_nama">Nama</label> <br>
      <input type="text" name="nama" id="textbox_nama">
      <br><br>
```

```

Jenis Kelamin<br>
<input type="radio" name="jenis_kelamin" value="pria"
       id="radio_jk_pria">
<label for="textbox_jk_pria">Pria</label>
<br>
<input type="radio" name="jenis_kelamin" value="wanita"
       id="radio_jk_wanita">
<label for="textbox_jk_wanita">Wanita</label>

<br><br>

Pekerjaan<br>
<input type="radio" name="pekerjaan" value="sekolah"
       id="radio_pekerjaan_sekolah">
<label for="radio_pekerjaan_sekolah">Sekolah</label>
<br>
<input type="radio" name="pekerjaan" value="swasta"
       id="radio_pekerjaan_swasta">
<label for="radio_pekerjaan_swasta">
    Pegawai Swasta
</label>
<br>
<input type="radio" name="pekerjaan" value="pns"
       id="radio_pekerjaan_pns">
<label for="radio_pekerjaan_pns">Pegawai Negeri</label>
<br><br>
<input type="submit" value="kirim data">
<input type="reset" value="batal">
</form>
</body>
</html>

```

Sedangkan antarmukanya dapat dilihat pada Gambar 39. Pada Kode di atas dapat dilihat untuk membuat group radio button dilakukan dengan cara memberikan nilai atribut name yang sama.

The screenshot shows a window titled "Radio Button". At the top, there are standard window controls (minimize, maximize, close) and a toolbar with icons for back, forward, search, and file operations. Below the toolbar, the title bar displays the window name and a file path: "file:///D:/Data/My%". The main content area contains the following fields:

- A text input field labeled "Nama" with a placeholder text box below it.
- A label "Jenis Kelamin" followed by two radio button options: "Pria" and "Wanita".
- A label "Pekerjaan" followed by four radio button options: "Sekolah", "Pegawai Swasta", and "Pegawai Negeri".
- At the bottom, there are two buttons: "kirim data" and "batal".

*Gambar 39. Form input data - Radio button - Latihan 1.*

## Checkbox

Elemen checkbox juga diperuntukkan agar user dapat memilih opsi yang sudah disediakan. Opsi dari terdiri atas satu atau lebih. Berbeda dengan radio button yang hanya dapat memilih satu opsi saja, checkbox memungkinkan user untuk memilih satu atau lebih opsi yang tersedia.

Sintaks tag checkbox adalah sebagai berikut.

```
<input type="checkbox" name="name_" value="value_" id="id_">
```

Atribut type bernilai checkbox untuk membuat elemen checkbox. Jika terdapat lebih dari 1 opsi maka nilai atribut name masing-masing tag ini harus bernilai berbeda.

Berikut adalah contoh penggunaan tag <input> untuk membuat checkbox.

The screenshot shows a simple web form titled "Checkbox". It contains a text input field labeled "Nama", a section for programming languages with checkboxes for "Java" and "PHP", and two buttons at the bottom: "kirim" and "batal".

*Gambar 40. Form input data - Checkbox - Contoh.*

Berikut adalah kode HTML untuk membuat halaman web seperti pada gambar di atas.

```
checkbox_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Checkbox</title>
  </head>

  <body>
    <form method="GET">
      <label for="textbox_nama">Nama</label> <br>
      <input type="text" name="nama" id="textbox_nama">
      <br><br>

      Bahasa pemrograman yang dikuasai:<br>
      <input type="checkbox" name="programming1" value="Java"
             id="checkbox_programming1">
      <label for="checkbox_programming1">Java</label>
      <br>
      <input type="checkbox" name="programming2" value="PHP"
             id="checkbox_programming2">
      <label for="checkbox_programming2">PHP</label>

      <br><br>
      <input type="submit" value="kirim">
      <input type="reset" value="batal">
    </form>
  </body>
</html>
```

## Latihan 1

Pada latihan ini dibuat antarmuka untuk login yang menggunakan sebuah elemen checkbox.

The screenshot shows a window titled "Login". The address bar displays "file:///D:/Data/My%20Books/IN". The main content area is titled "Login". It contains two text input fields labeled "Username" and "Passowrd". Below these is a checkbox labeled "Remember me". At the bottom are two buttons: "login" and "batal".

Gambar 41. Form input data - Checkbox - Form login.

Kode HTML untuk form login di atas adalah sebagai berikut.

```
checkbox_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Login</title>
    </head>

    <body>
        <form method="GET">
            <h4>Login</h4>
            <label for="textbox_username">Username</label> <br>
            <input type="text" name="username" id="textbox_username">
            <br>
            <label for="textbox_password">Passowrd</label> <br>
            <input type="password" name="password"
                   id="textbox_password">

            <br>
            <input type="checkbox" name="remember_me" value="true"
                   id="checkbox_remember_me" checked="checked">
            <label for="checkbox_remember_me">Remember me</label>
            <br><br>
```

```
<input type="submit" value="login">
<input type="reset" value="batal">
</form>
</body>
</html>
```

Pada tag <input type="checkbox"> di atas diperkenalkan atribut baru yaitu checked yang berfungsi untuk melakukan inisialisasi status cek pada checkbox.

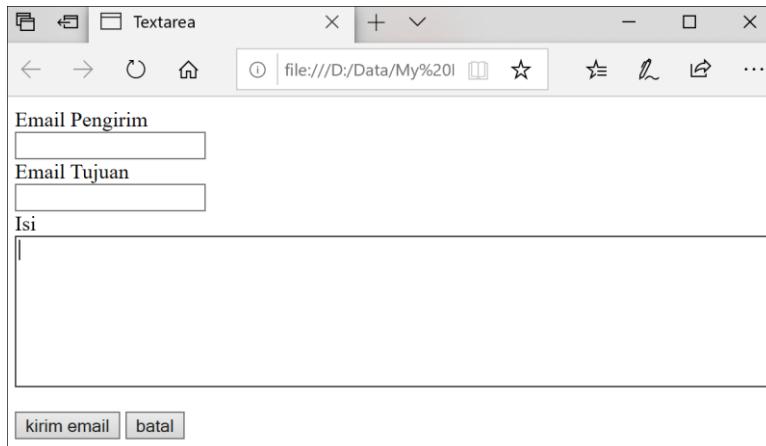
## Textarea

Pada sub bab sebelumnya telah dijelaskan tentang elemen textbox. TextBox umumnya digunakan jika isian hanya sebaris saja. Jika isian bisa lebih dari satu baris, bisa menggunakan elemen textarea. Contoh implementasi textarea adalah kolom isian untuk isi email atau isi blog.

Sintaks penggunaan tag textarea adalah sebagai berikut.

```
<textarea name="name_" id="id_" cols="75" rows="7"></textarea>
```

Atribut penting selain name dan id pada textarea adalah cols dan rows. Atribut cols untuk menentukan panjang textarea. Sedangkan atribut rows untuk menentukan jumlah baris yang ditampilkan. Walaupun jumlah baris telah ditentukan dengan atribut ini, user tetap bisa menulis teks dengan jumlah baris di atas nilai atribut rows ini.



Gambar 42. Form input data - Textarea - Form isian email.

Berikut adalah kode HTML untuk membuat antarmuka di atas.

```
textarea_1.html

<!DOCTYPE HTML>
<html>
    <head>
        <title>Textarea</title>
    </head>

    <body>
        <form method="POST">
            <label for="textbox_email_pengirim">Email Pengirim</label>
            <br>
            <input type="text" name="email_pengirim"
                   id="textbox_email_pengirim">

            <br>
            <label for="textbox_email_tujuan">Email Tujuan</label> <br>
            <input type="text" name="email_tujuan"
                   id="textbox_email_tujuan">

            <br>
            <label for="textarea_isi_email">Isi</label> <br>
            <textarea name="isi_email" id="textarea_isi_email"
                      cols="75" rows="7">
            </textarea>
            <br><br>
            <input type="submit" value="kirim email">
            <input type="reset" value="batal">
        </form>
    </body>
</html>
```

Untuk mengirimkan data dari halaman form yang memiliki elemen textarea, sebaiknya atribut method pada tag <form> adalah POST, bukan GET. Hal ini untuk menghindari teks yang sangat panjang pada URL.

## Drop-down List

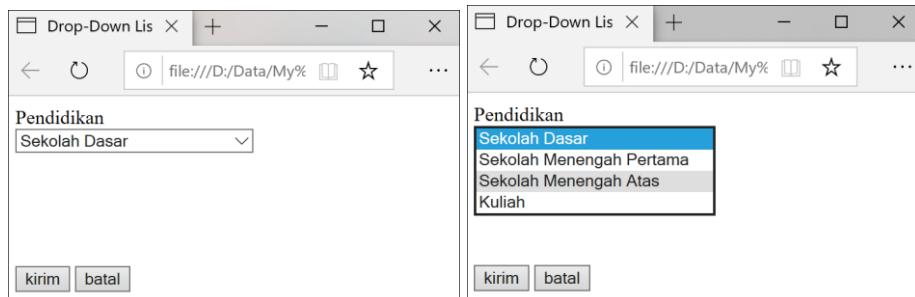
Untuk membuat drop-down list digunakan tag <select> dan <option>. Logika pembuatan drop-down list ini sama dengan logika pembuat list dengan <ul> atau <ol> yang didalamnya ada tag <li>.

Berikut adalah sintaks untuk membuat drop-down list.

```
<select name="name_" id="id_>
    <option value="value_1">opsi 1</option>
    <option value="value_2">opsi 2</option>
    . . .
</select>
```

Pada sintaks di atas dapat dilihat bahwa tag `<select>` merupakan tipe yang memiliki tag pembuka dan penutup. Di antara kedua tag pembuka dan penutup tersebut disi dengan opsi-opsi yang dapat dipilih user. Sebuah opsi diwakili oleh tag `<option>`. Tag ini juga merupakan tipe tag dengan pembuka dan penutup. Nilai dari opsi ini disimpan pada atribut `value`. Sedangkan teks dari sebuah opsi diletakkan di antara tag `<option>` pembuka dan tag `</option>` penutup.

Berikut adalah contoh drop-down list pada halaman web.



*Gambar 43. Form input data - Drop-down list - Contoh.*

Pada gambar di atas sebelah kiri dapat dilihat drop-down list yang belum diklik oleh user. Sedangkan pada gambar di sebelah kanan adalah ketika drop-down list diklik user.

Berikut adalah kode HTML untuk membuat tampilan seperti pada gambar di atas.

```
dropdownlist_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Drop-Down List</title>
    </head>

    <body>
        <form method="GET">
            <label for="select_pendidikan">Pendidikan</label> <br>
            <select name="pendidikan" id="select_pendidikan">
```

```

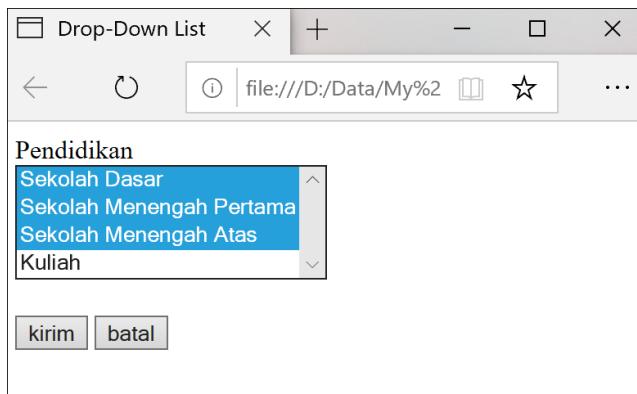
<option value="sd">Sekolah Dasar</option>
<option value="smp">Sekolah Menengah Pertama</option>
<option value="sma">Sekolah Menengah Atas</option>
    <option value="kuliah">Kuliah</option>
</select>
<br><br><br><br><br>
<input type="submit" value="kirim">
<input type="reset" value="batal">
</form>
</body>
</html>

```

Kode ini memungkinkan user hanya memilih satu opsi saja. Agar user dapat memilih lebih dari 1 opsi maka perlu penambahan atribut pada tag <select> seperti yang dijelaskan pada bagian Latihan 1.

## Latihan 1

Tag ini juga dapat digunakan untuk membuat tampilan dimana user dapat memilih lebih dari satu opsi seperti pada gambar di bawah ini.



*Gambar 44. Form input data - Drop-down list - Latihan 1.*

Berikut adalah kode HTML untuk membuat tampilan seperti gambar di atas.

```

dropdownlist_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Drop-Down List</title>
    </head>
    <body>
        <form method="GET">

```

```
<label for="select_pendidikan">Pendidikan</label> <br>
<select name="pendidikan" id="select_pendidikan"
        multiple="multiple">
    <option value="sd">Sekolah Dasar</option>
    <option value="smp">Sekolah Menengah Pertama</option>
    <option value="sma">Sekolah Menengah Atas</option>
    <option value="kuliah">Kuliah</option>
</select>
<br><br>
<input type="submit" value="kirim">
<input type="reset" value="batal">
</form>
</body>
</html>
```

Untuk membuat tag <select> dapat digunakan oleh user untuk memilih lebih dari satu opsi, cukup dengan menambahkan atribut multiple. Seperti pada kode di atas.

## Upload File

---

Elemen upload file digunakan untuk memilih file yang akan diupload. Ada beberapa syarat bagi halaman untuk mengupload file. Syarat pertama adalah tag <form> harus menggunakan atribut method="POST". Gunakan atribut enctype="multipart/form-data". Selanjutnya adalah gunakan tag <input> dengan atribut type="file" di dalam tag form tersebut. Berikut adalah contoh sintaks dari tag untuk membuat elemen upload file.

```
<form method="POST" enctype="multipart/form-data">
    <input type="file" name="name_" id="id_">
</form>
```

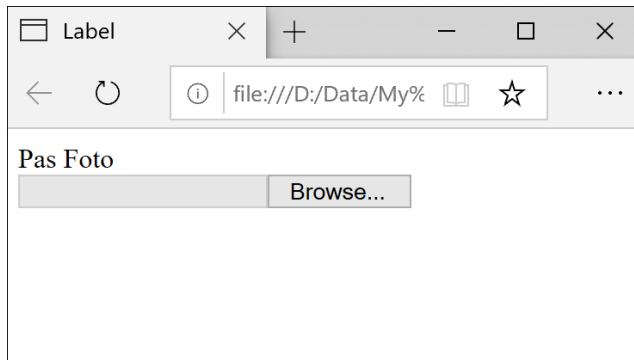
### Latihan 1

Berikut adalah contoh penggunaan elemen upload file untuk upload foto.

```
file_upload_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>File Upload</title>
    </head>
    <body>
```

```
<form method="POST" enctype="multipart/form-data">
    <label for="file_upload_foto">Pas Foto</label><br>
    <input type="file" name="file_foto"
           id="file_upload_foto">
</form>
</body>
</html>
```

Hasilnya dapat dilihat pada gambar di bawah ini.



*Gambar 45. Form input data- Upload file - antarmuka upload file pada Microsoft Edge.*

---

## Project

Berikut ini adalah project yang harus dikerjakan dan diselesaikan.

---

### **Web Profile Pribadi**

---

Pada tugas ini pembaca harus membuat beberapa halaman web yang saling terhubung. Beberapa halaman pada project ini telah dibuat pada latihan-latihan sebelumnya hanya saja perlu dilengkapi.

Halaman utama dari project ini adalah profile.html, lengkapi halaman ini untuk menceritakan profile diri anda.

Halaman kedua adalah gallery.html yang merupakan gallery foto anda. Isi halaman gallery dengan 20 foto. Ukuran asli foto usahakan jangan lebih kecil dari 800 x 600 pixel. Namun ketika ditampilkan pada halaman gallery.html, sesuaikan lebarnya menjadi 100 pixel saja. Setiap foto dapat diklik, ketika foto diklik maka user akan diantarkan ke file asli dari setiap foto tersebut. Gambar-gambar yang ditampilkan pada halaman ini disimpan dalam sebuah folder dengan nama gallery.

Halaman ketiga adalah data.html yang berisi data 10 buku yang pernah anda baca. Data ditampilkan dalam bentuk tabel yang berisi:

- Judul buku.
- Pengarang.
- Foto buku.
- Detail yang berisi hyperlink teks “Detail” yang dapat diklik. Ketika hyperlink ini diklik user akan diantarkan pada bagian detail dari buku tersebut.

Gambar-gambar buku yang ditampilkan pada halaman ini disimpan ke dalam sebuah folder yaitu books. Informasi detail dari setiap buku diletakkan di bawah tabel data buku tersebut. Di bawah informasi detail dari setiap buku berikan hyperlink “TOP” yang ketika diklik akan mengantarkan user kembali ke tabel data daftar buku (seperti yang telah dijelaskan pada Latihan 3 di sub bab Hyperlink).

Ketiga halaman ini harus memiliki header yang berisi foto anda dan semboyan yang berada di atas tengah (center) halaman web. kemudian dibawahnya tambahkan menu untuk mengakses halaman lainnya seperti yang telah dibuat pada Latihan 1 di sub bab Hyperlink.

## Form & Daftar Buku Tamu

---

Buat form untuk mengisi buku tamu yang terdiri atas beberapa isian, yaitu:

- Nama.
- Email.
- Tanggal.
- Pertanyaan: Apakah website ini bagus? Dengan jawaban dalam bentuk radio button yang terdiri atas 3 buah pilihan yaitu bagus, biasa dan jelek. Tapi pilihan tersebut tidak berupa teks. Pilihan dibuat berupa gambar bintang atau jempol. Jika pilihan bagus berikan 3 bintang atau 3 jempol. Jika pilihan biasa gunakan 2 bintang atau 2 jempol. Jika pilihan jelek gunakan 1 bintang atau jempol.
- Komentar yang dapat diisi oleh user berupa teks yang bisa terdiri atas lebih dari 1 baris.

Di bawah form buat tabel daftar isian buku tamu dengan tampilan seperti berikut.

Nama (Email) memberikan komentar pada <b>Tanggal</b> .	
<b>Komentar</b>	
Nama (Email) memberikan komentar pada <b>Tanggal</b> .	
<b>Komentar</b>	
Nama (Email) memberikan komentar pada <b>Tanggal</b> .	
<b>Komentar</b>	
Nama (Email) memberikan komentar pada <b>Tanggal</b> .	
<b>Komentar</b>	

## **Cascading Style Sheets**

Cascading Style Sheets (CSS) adalah bahasa yang digunakan memberikan style pada dokumen HTML atau halaman web yang ditulis dengan kode HTML. CSS menentukan bagaimana elemen HTML ditampilkan yang meliputi bentuk, warna dan posisi suatu tag atau elemen HTML. Selain itu CSS digunakan untuk mengatur tampilan beberapa halaman web sekaligus dalam sebuah file.

Pada bab ini akan dijelaskan konsep, contoh dan latihan untuk menggunakan CSS.

---

### **Pendahuluan**

Sub bab ini berisi sintaks dan tata cara penggunaannya pada halaman HTML.

---

### **Sintaks**

Secara umum CSS terdiri atas property dan value dengan sintaks sebagai berikut:

```
property: value;
```

Sebagai contoh adalah sebagai berikut:

```
font-size: 14px;
```

Contoh di atas mempunyai arti bahwa property adalah font-size dengan value adalah 14px. Jika style dengan property ini diberikan pada sebuah tag HTML maka teks yang berada di dalam tag tersebut akan berukuran 14px.

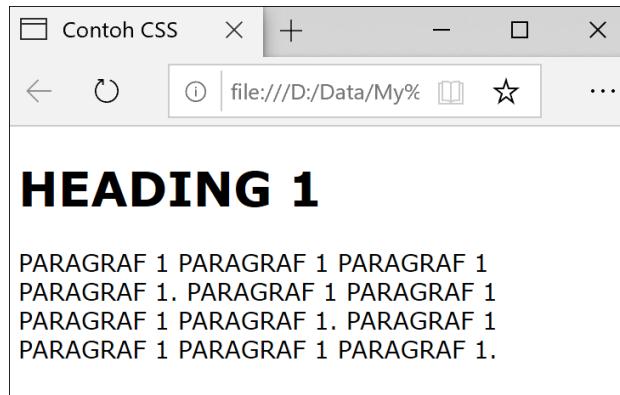
Style CSS tidak hanya berisi sebuah property saja, namun dapat berisi banyak property dan value. Sintaksnya dapat dilihat pada kode di bawah ini.

```
property1: value1; property2: value2; property3: value3; . . .
```

Sebagai contoh adalah sebagai berikut:

```
font-size: 14px; font-family: Verdana; text-transform: uppercase;
```

Contoh di atas memiliki arti bahwa teks yang berada di dalam tag yang menggunakan style ini berukuran 14px dengan nama font adalah Verdana dan huruf besar. Atau dapat dilihat pada gambar di bawah ini.



*Gambar 46. Contoh CSS untuk style teks.*

## Tata Cara Penggunaan

Penggunaan CSS dapat dilakukan dengan beberapa cara yaitu:

- Inline CSS.
- Internal CSS.
- External CSS.

### *Inline CSS*

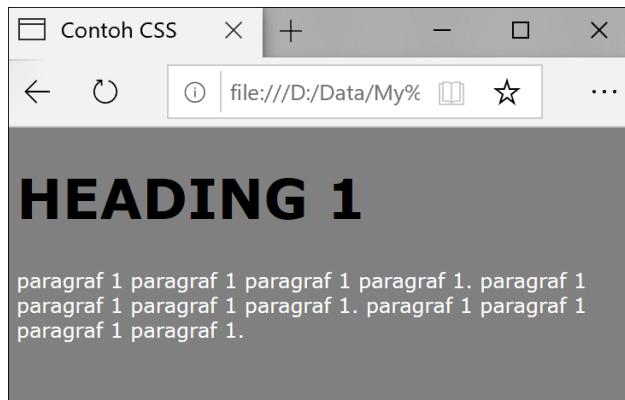
Cara ini dituliskan langsung pada sebuah tag HTML dengan memberikan property dan value CSS di dalam atribut style pada tag HTML tersebut. Sintaks dari cara ini dapat dilihat di bawah ini.

```
<tag_html style="property1: value1; property2: value2;">  
. . .  
</tag_html>
```

Untuk penulisan pada tag dengan tipe void elemen maka sintaksnya adalah sebagai berikut.

```
<tag_html style="property1: value1; property2: value2;">
```

Berikut adalah contoh penulisan dengan cara ini. Hasilnya dapat dilihat pada gambar di bawah ini.



*Gambar 47. Contoh Inline CSS.*

Kode HTML dan CSS untuk membuat hasil seperti gambar di atas adalah sebagai berikut.

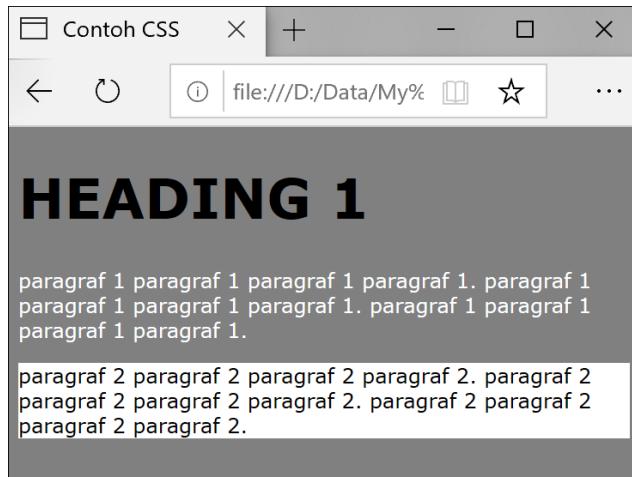
```
contoh_css_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Contoh CSS</title>
    </head>

    <body style="background-color: gainsboro; font-family: Verdana;">
        <h1 style="text-transform: uppercase;">Heading 1</h1>
        <p style="font-size: 12px; color: white;">
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
        </p>
        <p>paragraf 2 paragraf 2</p>
    </body>
</html>
```

Pada contoh di atas dapat dilihat teks heading 1 dan paragraf menggunakan font Verdana. Hal ini disebabkan penggunaan CSS dengan property font-

family:Verdana pada tag body. Karena heading 1 <h1> dan paragraf <p> berada di dalam tag <body> maka teks didalamnya juga menggunakan tipe font yang sama. Ini menunjukkan bahwa style CSS berefek secara hirarki.

Contoh bagaimana style CSS berefek secara hirarki dapat juga dilihat pada contoh berikut ini.



Gambar 48. Efek hirarki CSS.

Berikut adalah kode HTML dan CSS untuk membuat antarmuka di atas.

```
Contoh_css_3.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Contoh CSS</title>
    </head>

    <body style="background-color: gray; font-family: Verdana;">
        <h1 style="text-transform: uppercase;">Heading 1</h1>
        <p style="font-size: 12px; color: white;">
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.  

            paragraf 1 paragraf 1 paragraf 1 paragraf 1.  

            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
        </p>
        <p style="font-size: 12px; color: black; background-color: white;">
            paragraf 2 paragraf 2 paragraf 2 paragraf 2.  

            paragraf 2 paragraf 2 paragraf 2 paragraf 2.  

            paragraf 2 paragraf 2 paragraf 2 paragraf 2.
        </p>
    </body>
</html>
```

Pada contoh di atas, dapat dilihat efek background-color yang digunakan pada tag <body> ber efek pada tag <p> yang pertama. Namun tidak berlaku pada tag <p> yang kedua karena pada tag ini menggunakan nilai property background-color yang berbeda.

Pada sub bab Sintaks disebutkan bahwa value tidak boleh menggunakan spasi. Jika value berisi nilai yang terdiri atas dua kata maka perlu digabungkan dengan tanda petik satu ('). Berikut adalah contoh penggunaannya.

```
contoh_css_4.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Contoh CSS</title>
    </head>

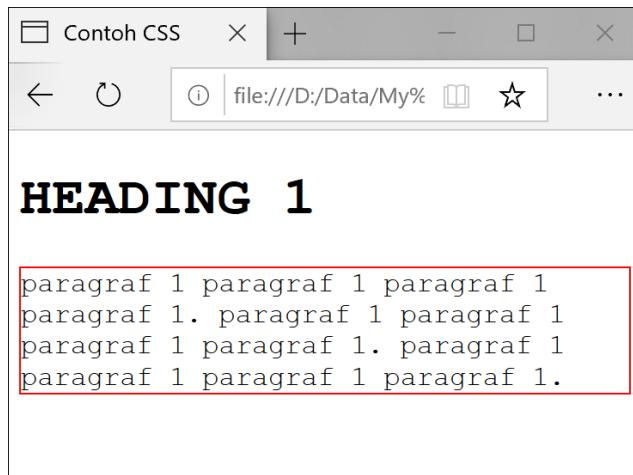
    <body style="font-family: 'Courier New', Courier, monospace;">
        <h1 style="text-transform: uppercase;">Heading 1</h1>
        <p style="font-size: 12px;">
            paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
        </p>
    </body>
</html>
```

Pada contoh di atas dapat dilihat bahwa sebuah property dapat memiliki lebih dari 1 nilai. Sebagai contoh property font-family memiliki nilai 'Courier New', Courier dan monospace. Namun untuk nilai yang terdiri atas dua kata disatukan dengan tanda petik satu. Contoh lain yang menunjukkan property dapat memiliki lebih satu nilai adalah sebagai berikut.

```
contoh_css_5.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Contoh CSS</title>
    </head>

    <body style="font-family: 'Courier New', Courier, monospace;">
        <h1 style="text-transform: uppercase;">Heading 1</h1>
        <p style="border: 1px solid red;">
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
        </p>
    </body>
</html>
```

Pada tag <p> dapat dilihat property border memiliki nilai 1px yang artinya ukuran border adalah 1 pixel. Nilai solid berarti border akan digambar dengan solid (bukan titik-titik atau dash). Sedangkan nilai red menyatakan warna dari border. Hasilnya dapat dilihat pada gambar di bawah ini.



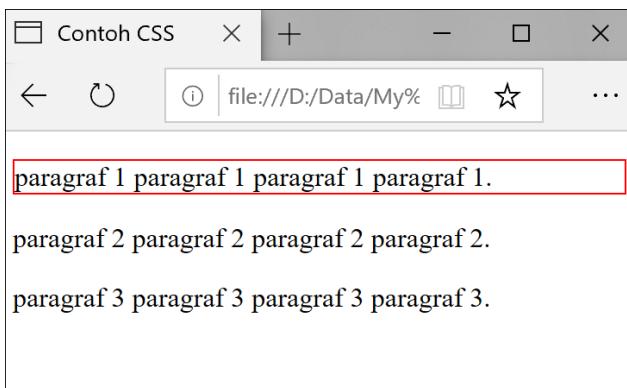
*Gambar 49. Border.*

Cara inline CSS ini mempunyai kekurangan yaitu efek style hanya berlaku pada tag yang menggunakan tersebut saja. Sebagai contoh, jika pada sebuah halaman web terdapat beberapa tag <h1> dan <p> yang berada diposisi yang setara, maka jika salah satu tag saya yang diberi style CSS maka tag lain tidak akan mendapatkan efeknya. Seperti contoh berikut ini.

```
contoh_css_6.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Contoh CSS</title>
    </head>

    <body>
        <p style="border: 1px solid red;">
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
        </p>
        <p>
            paragraf 2 paragraf 2 paragraf 2 paragraf 2.
        </p>
        <p>
            paragraf 3 paragraf 3 paragraf 3 paragraf 3.
        </p>
    </body>
</html>
```

Hasilnya dapat dilihat pada gambar di bawah ini.



*Gambar 50. Efek style pada elemen HTML.*

Jika ingin semua tag <p> mempunyai style yang sama maka seluruh tag ini harus diberikan nilai yang sama pada atribut style. Hal ini memiliki dua kekurangan. Yang pertama, ukuran file akan menjadi lebih besar karena banyak tambahan teks yang berisi nilai yang sama. Yang kedua, jika ternyata perubahan style pada tag tertentu maka perubahan harus dilakukan pada seluruh tag-tag yang lainnya.

### *Internal CSS*

Untuk menghindari penulisan style CSS yang berulang-ulang seperti yang telah disebutkan pada sub bab sebelumnya maka penulisan style CSS dapat dilakukan dengan cara internal CSS.

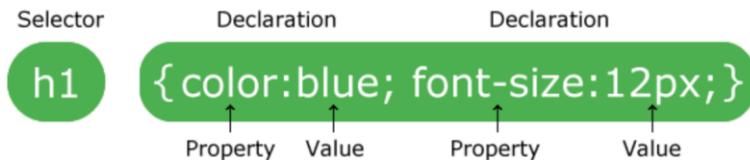
Penulisan CSS dengan cara ini dilakukan di dalam tag <head>, bukan di dalam tag <body> seperti yang dilakukan pada cara inline CSS. Untuk menulis CSS di dalam tag <head> diperlukan tag <style>. Berikut adalah sintaks dan posisi dari tag <style>.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Internal CSS</title>
    <style>
      /* style CSS dituliskan di sini */
    </style>
  </head>

  <body>
```

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

Cara penulisan CSS dengan cara ini sedikit berbeda dengan cara sebelumnya. Pada cara ini perlu ditunjuk tag yang ingin diberikan style kemudian style ditulis di antara tanda { dan }. Berikut adalah sintaks internal CSS.



*Gambar 51. Sintaks pada Internal CSS.*

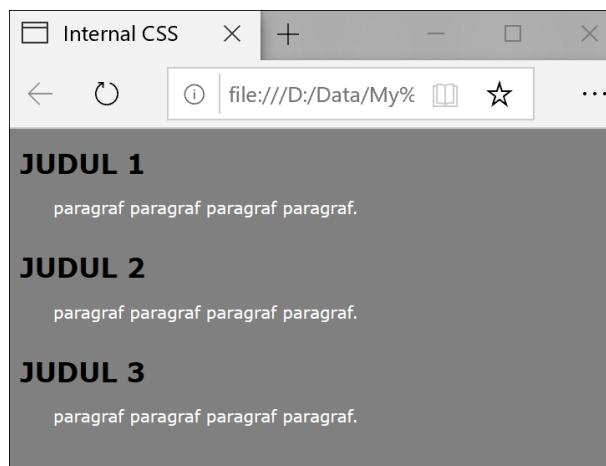
Selector adalah tag atau elemen HTML yang akan diberikan style. Kemudian style ditulis setelahnya di dalam tanda { dan }. Berikut adalah contoh penulisannya.

```
internal_css_1.html  
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Internal CSS</title>  
    <style>  
      body {  
        font-family: Verdana, Geneva, Tahoma, sans-serif;  
        background-color: grey;  
      }  
  
      h1 {  
        text-transform: uppercase;  
        font-size: 16px;  
      }  
  
      p {  
        color: white;  
        font-size: 10px;  
        margin: 0px 0px 20px 20px;  
      }  
    </style>  
  </head>  
  
<body>  
  <h1>Judul 1</h1>  
  <p>  
    paragraf paragraf paragraf paragraf.  
  </p>
```

```
<h1>Judul 2</h1>
<p>
    paragraf paragraf paragraf paragraf.
</p>

<h1>Judul 3</h1>
<p>
    paragraf paragraf paragraf paragraf.
</p>
</body>
</html>
```

Pada kode di atas, style diberikan pada elemen body, h1 dan p. Hasilnya dapat dilihat pada gambar di bawah ini.



*Gambar 52. Internal CSS - Contoh.*

Pada kode di atas, elemen p memiliki property margin dengan 4 nilai yang menyatakan margin untuk bagian atas, kanan, bawah dan kiri.

Pada contoh di atas selector menggunakan tag HTML. Selain itu selector juga dapat menggunakan nilai atribut class yang dimiliki oleh tag HTML. Berikut contoh implementasi cara ini.

```
internal_css_3.html
<!DOCTYPE HTML>
<html>
<head>
    <title>Internal CSS</title>
```

```

<style>
    body {
        font-family: Verdana, Geneva, Tahoma, sans-serif;
        background-color: grey;
    }

    h1 {
        text-transform: uppercase;
        font-size: 16px;
    }

    .paragraf_normal {
        color: white;
        font-size: 10px;
        margin: 0px 0px 20px 20px;
    }
</style>
</head>

<body>
    <h1>Judul 1</h1>
    <p>
        paragraf paragraf paragraf paragraf.
    </p>

    <h1>Judul 2</h1>
    <p class="paragraf_normal">
        paragraf paragraf paragraf paragraf.
    </p>
</body>
</html>

```

Pada kode di atas dapat dilihat terdapat dua tag `<p>`. Pada tag yang pertama tidak memiliki atribut. Sedangkan pada tag yang kedua memiliki atribut `class="paragraf_normal"`. Untuk memberikan style pada tag yang memiliki atribut `class` dengan nilai adalah `"paragraf_normal"`, cara penulisan selector diawali dengan tanda titik (.) kemudian diikuti dengan nilai atribut `class`.

Selain itu juga dapat digunakan nilai atribut `id` sebagai selector. Berikut adalah contoh jika ingin menggunakan nilai atribut `id` sebagai selector. Berikut adalah antarmuka yang akan dibuat.



*Gambar 53. Internal CSS - Layout.*

Dan berikut ini adalah kode HTML dan CSS dari antarmuka di atas.

```
internal_css_5.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Internal CSS</title>
    <style>
      #header {
        width: auto;
        height: 100px;
        border: solid 1px;
        background-color: lightblue;
      }

      #main_menu {
        width: auto;
        height: 30px;
        border: solid 1px gray;
        background-color: gray;
        color: white;
      }

      #side_menu {
        float: left;
        width: 10%;
        height: 500px;
        border: solid 1px;
        background-color: aqua;
      }
    </style>
  </head>
  <body>
    <div id="header"></div>
    <div id="main_menu"></div>
    <div id="side_menu"></div>
    <div id="content"></div>
  </body>
</html>
```

```

        #content {
            float: left;
            width: 89%;
            height: 500px;
            border: solid 1px;
            margin-left: 3px;
            background-color: aquamarine;
        }
    </style>
</head>

<body>
    <div id="header">header</div>
    <div id="main_menu">main menu</div>
    <div id="side_menu">side menu</div>
    <div id="content">content</div>
</body>
</html>

```

Pada kode di atas digunakan tag `<div>` untuk membuat layout. Tag ini mempunyai karakteristik seperti kotak sehingga dapat diatur ukurannya seperti tinggi dan lebar. Kemudian dapat diatur juga posisinya. Pada kode di atas terdapat 4 tag `<div>` untuk membuat area untuk header, menu utama (main menu), menu samping (side menu) dan isi (content). Setiap tag tersebut diberi atribut id dengan nilai yang berbeda. Kemudian untuk menggunakan nilai atribut id sebagai selector digunakan tanda `#` kemudian diikuti dengan nilai atribut id yang ingin diberi style.

Internal CSS mempunyai kelebihan jika dibandingkan Inline CSS. Namun tetap memiliki kekurangan karena style yang ditulis hanya berlaku pada sebuah halaman saja. Padahal aplikasi web dapat terdiri atas beberapa halaman web yang mempunyai antarmuka yang sama. Sehingga jika terjadi perubahan style pada semua halaman, maka modifikasi juga harus dilakukan pada kode CSS di halaman web lainnya.

### *External CSS*

External CSS adalah cara untuk menyimpan CSS pada sebuah file, kemudian file tersebut dapat digunakan oleh banyak halaman web. Sehingga jika terjadi perubahan CSS cukup dengan memodifikasi file CSS tersebut, maka seluruh halaman web yang menggunakan file tersebut akan mendapat efek perubahannya.

Untuk memanggil file CSS pada halaman web digunakan tag `<link>` dengan sintaks sebagai berikut ini.

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

Berikut adalah contoh implementasi cara ini.

```
external_css_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>External CSS</title>
    <link rel="stylesheet" type="text/css" href="css/default.css">
  </head>

  <body>

  </body>
</html>
```

Kemudian buat folder css yang lokasinya berada di folder yang sama dengan file external\_css\_1.html ini. Selanjutnya buat file default.css dan simpan di dalam folder css tersebut. Berikut adalah isi dari file default.css.

```
default.css
body {
  background-color: gainsboro;
  font-family: Verdana, Geneva, Tahoma, sans-serif;
}
```

Sebuah halaman web dapat menggunakan lebih dari satu file CSS.

---

## Dasar-Dasar CSS

Berikut ini dasar-dasar memberikan style pada elemen-elemen HTML. Contoh-contoh penulisan kode CSS yang diberikan pada sub bab ini menggunakan cara internal CSS.

---

### Warna

Untuk memberikan warna pada elemen HTML digunakan property CSS berikut ini:

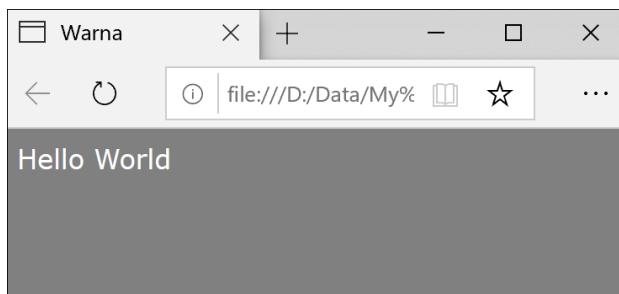
- color.
- background-color.
- border-color.

- outline-color.
- Dan lain-lain.

Sedangkan nilai property ini dapat menggunakan beberapa tipe yaitu:

- String, yaitu memberikan nilai warna dalam bentuk nama warna seperti white, black, grey dan lain-lain.
- Hex code, yaitu memberi nilai warna dengan angka hexadesimal. Nilai terdiri atas 6 bilangan hexadesimal dimana 2 nilai pertama menyatakan nilai merah (Red), 2 nilai kedua adalah hijau (Green) dan 2 nilai terakhir adalah nilai biru (Blue). Untuk menggunakan hex code angka harus diawali tanda #. Sebagai contoh untuk warna hitam digunakan hex code #000000 dan untuk warna putih digunakan hex code #FFFFFF.

Berikut adalah contoh penggunaan property color dan background-color.



*Gambar 54. Dasar-dasar CSS - Warna.*

Dan berikut adalah kode CSS dan HTML.

```
warna_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Warna</title>
    <link rel="stylesheet" type="text/css" href="css/default.css">
  </head>

  <body>
    Hello World
  </body>
</html>
```

```
warna_1.html
<!DOCTYPE HTML>
<html>
  <head>
```

```

<title>Warna</title>
<style>
    body {
        background-color: grey;
        color: #FFFFFF;
        font-family: Verdana, Geneva, Tahoma, sans-serif;
    }
</style>
</head>

<body>
    Hello World
</body>
</html>

```

Di dalam tag <style> dapat dilihat contoh implementasi property ini pada tombol. Berikut adalah kode HTML dan CSS yang digunakan. Pada kode HTML yang berisi elemen HTML untuk membuat tombol. Pada kode ini digunakan atribut class="tombol". Nilai atribut class ini digunakan pada file CSS di atas sebagai selector.

```

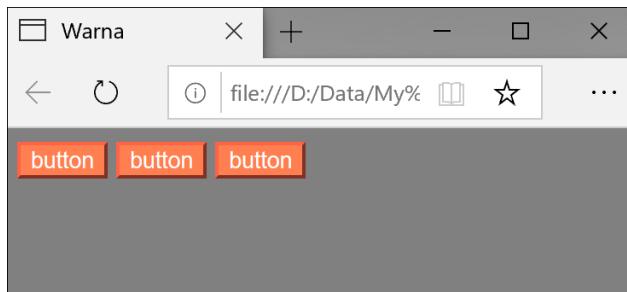
warna_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Warna</title>
        <style>
            body {
                background-color: grey;
                color:#FFFFFF;
                font-family: Verdana, Geneva, Tahoma, sans-serif;
            }

            .tombol {
                background-color: coral;
                color: #FFFFFF;
                border-color: tomato;
            }
        </style>
    </head>

    <body>
        <input type="button" value="button" id="button1" class="tombol">
        <input type="submit" value="button" id="button2" class="tombol">
        <input type="reset" value="button" id="button3" class="tombol">
    </body>
</html>

```

Hasilnya dari kode-kode di atas dapat dilihat pada gambar di bawah ini.



*Gambar 55. Style warna pada tombol.*

## Teks

---

Untuk mengatur style teks pada elemen HTML digunakan property CSS berikut ini:

- font-family.
- font-size.
- font-style digunakan untuk membuat teks menjadi cetak miring.
- font-weight digunakan untuk membuat teks cetak tebal.
- text-decoration digunakan untuk menghias teks seperti memberikan garis bawah, garis atas, coret dan lain-lain.
- Dan lain-lain.

Property font-family berisi nilai nama atau jenis font yang digunakan. Berikut ini beberapa nilai property ini:

- Verdana, Geneva, Tahoma, sans-serif.
- 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif.
- Dan lain-lain.

Isi nilai property ini dapat berisi banyak nama dan tipe font. Tapi juga dapat berisi sebuah nama saja. Berikut ini contoh kode CSS yang menunjukkan cara penggunaan property font-family.

```
body {  
background-color: grey;  
color:#FFFFFF;  
font-family: Verdana, Geneva, Tahoma, sans-serif;  
font-  
family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Luci  
da Sans Unicode', Geneva, Verdana, sans-serif;
```

```
}

.tombol{
    background-color: coral;
    color: #FFFFFF;
    border-color: tomato;
    font-family: Arial;
}
```

Sedangkan property font-size bertujuan untuk menentukan ukuran teks yang digunakan. Ada beberapa tipe ukuran yang dapat digunakan sebagai nilai dari property ini, yaitu:

- Unit pengukuran absolut, tipe ini berisi angka yang menyatakan ukuran font seperti 12px, 10pt dan lain-lain. Nilai tipe ini adalah angka diikuti dengan satuanya. Satuan yang bisa digunakan ada pixel (px), point (pt), centimenter (cm) dan lain-lain. Namun yang umum digunakan adalah satuan px dan pt. Penulisan angka dan satuan harus disatukan.
- Unit pengukuran relatif, tipe ini berisi string seperti xx-small, x-small, small, medium, large, x-large dan xx-large. Selain itu juga dapat berisi nilai persen seperti 150% yang artinya ukuran font akan 150% dari ukuran default. Satuan lain yang dapat digunakan adalah em, misal nilai property adalah 0.5em artinya ukuran font adalah 0.5 dari ukuran ukuran normal.

Berikut adalah contoh kode CSS dan HTML yang untuk contoh paparan di atas.

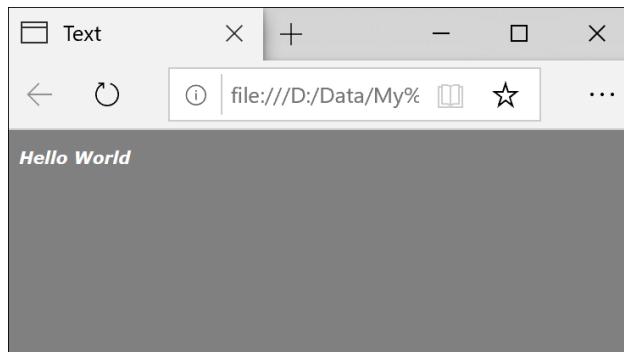
```
text_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Text</title>
        <style>
            body {
                background-color: grey;
                color:#FFFFFF;
                font-family: Verdana, Geneva, Tahoma, sans-serif;
                font-
family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Luci
da Sans Unicode', Geneva, Verdana, sans-serif;
            }

            p {
                font-family: Verdana;
                font-size: 10px;
                font-style: italic;
                font-weight: bolder;
            }
        </style>
    </head>
    <body>
        <p>Text</p>
    </body>
</html>
```

```
</style>
</head>

<body>
  <p>
    Hello World
  </p>
</body>
</html>
```

Berikut adalah tampilan dari halaman web yang menggunakan kode di atas.



*Gambar 56. Style teks.*

## Background Image

Pada halaman web, kita kadang melihat halaman tersebut memiliki background berupa gambar kemudian didepannya terdapat tulisan atau elemen HTML yang lain. Untuk membuat sebuah gambar menjadi latar belakang atau background pada sebuah elemen HTML maka dapat digunakan property seperti:

- background-image.
- background-repeat.
- background.
- Dan lain-lain.

Pada sub bab ini akan digunakan cara penulisan kode CSS dengan External CSS. Tujuannya adalah untuk memperlihatkan bagaimana cara menentukan lokasi file yang akan digunakan sebagai background.

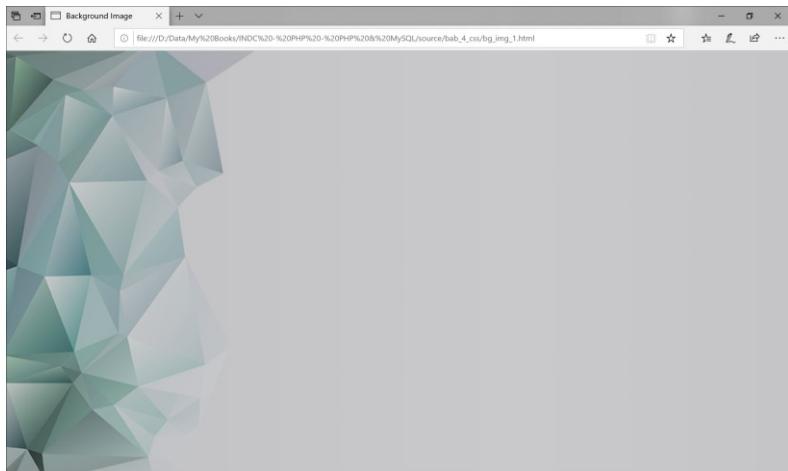
Misal dimiliki gambar dengan ukuran 1700 x 780 pixel dengan nama bg1.png yang disimpan pada folder images/bg. Jika ingin menjadikan gambar tersebut

sebagai background halaman web maka dapat dibuat kode CSS dan HTML sebagai berikut.

```
css/default.css
body {
    font-family: Verdana, Geneva, Tahoma, sans-serif;
    font-
family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Luci
da Sans Unicode', Geneva, Verdana, sans-serif;
    background-image: url(..../images/bg/bg1.png);
    background-repeat: no-repeat;
    width: 99%;
}
...
```

```
bg_img_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Background Image</title>
        <link rel="stylesheet" type="text/css" href="css/default.css">
    </head>
    <body>
    </body>
</html>
```

Dan hasilnya dapat dilihat pada gambar di bawah ini.



**Gambar 57. Background image menggunakan gambar ukuran besar.**

Pada kode CSS di atas dapat dilihat penggunaan property:

- background-image dengan nilai yaitu url(..../images/bg/bg1.png). Property ini berfungsi untuk menunjuk lokasi dan nama file dari gambar yang digunakan sebagai background. Seperti yang telah disebutkan sebelumnya bahwa file default.css ini disimpan pada folder css seperti pada Gambar 58. Sedangkan gambar disimpan pada folder images/bg. Sehingga cara menentukan lokasi gambar harus mengacu dari di mana lokasi file default.css berada. Sehingga posisinya harus keluar terlebih dahulu dari folder css dengan menggunakan tanda "../" kemudian masuk ke folder images/bg. Sehingga cara penentuan lokasinya adalah "../images/bg/bg1.png".
- background-repeat berfungsi untuk menentukan apakah gambar akan diulang atau tidak. Jika nilai property ini adalah no-repeat maka gambar tidak akan diulang.

Name	Date modified	Type
css	2/24/2020 7:12 PM	File folder
images	2/24/2020 7:12 PM	File folder
bg_img_1.html	2/24/2020 7:26 PM	HTML File
bg_img_2.html	2/24/2020 7:34 PM	HTML File

**Gambar 58. Lokasi penyimpanan gambar dan file default.css.**

Property lain yang dapat digunakan dengan fungsi yang sama adalah sebagai berikut:

```
background: url(..../images/bg/bg1.png) no-repeat;
```

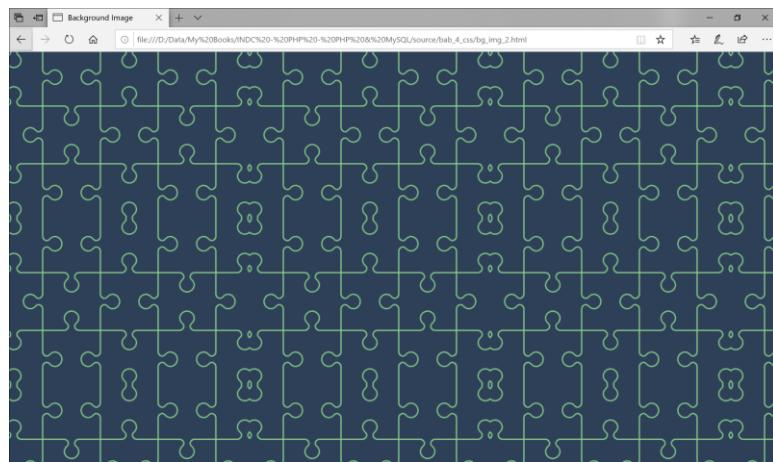
Kode CSS tersebut mempunyai fungsi yang sama dengan dua baris property CSS yang digunakan sebelumnya.

Contoh yang lain jika digunakan gambar dengan ukuran 400 x 277 pixel, dengan nama file adalah bg2.png. Jika file ini dijadikan background dengan cara di atas maka akan didapat hasil seperti berikut.



*Gambar 59. Membuat background image dengan gambar ukuran kecil.*

Sedangkan jika digunakan property background-repeat: repeat; maka akan didapat tampilan background seperti gambar ini.



*Gambar 60. Background image yang diulang.*

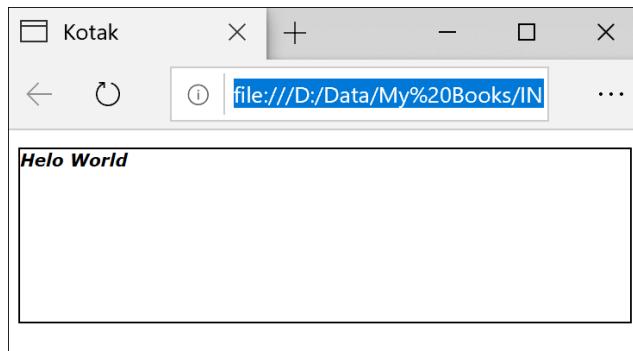
Berikut adalah kode CSS yang digunakan untuk membuat background gambar terulang pada tag <body>.

```
body {  
    font-family: Verdana, Geneva, Tahoma, sans-serif;  
    font-family: 'Lucida Sans', 'Lucida Sans Regular', 'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana, sans-serif;  
    background-image: url(..../images/bg/bg2.png);  
    background-repeat: repeat;  
}
```

Background berupa gambar ini juga dapat diberikan pada elemen HTML lain seperti <div> atau elemen yang lain.

## Kotak

Hampir seluruh elemen HTML sebenarnya mempunyai bingkai berupa kotak. Sehingga elemen tersebut mempunyai sisi kiri, kanan, atas dan bawah. Oleh karena itu hampir semua elemen tersebut dapat diberi border. Sebagai contoh tag <p> akan diberikan style seperti berikut ini.



*Gambar 61. Border pada element HTML.*

Berikut adalah kode CSS dan HTML yang digunakan.

```
border_1.html  
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>Kotak</title>  
    <style>  
      p.kotak {  
        border: 1px solid;  
        min-height: 100px;  
      }  
    </style>  
  </head>
```

```
<body>
  <p class="kotak">
    Hello World
  </p>
</body>
</html>
```

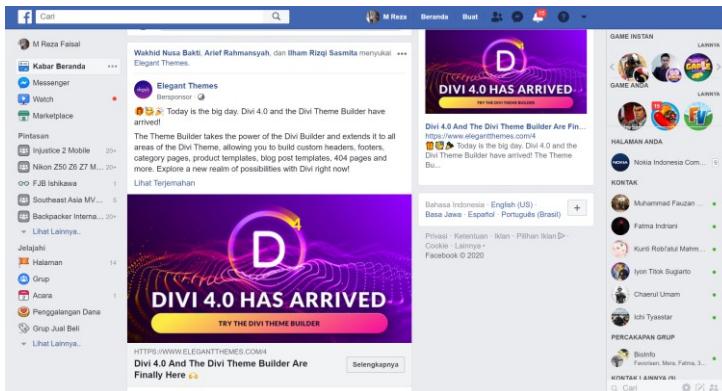
## Tata Letak dengan CSS

Pada sub bab ini akan diberikan konsep dan langkah-langkah untuk membuat tata letak dengan CSS.

### Template Utama

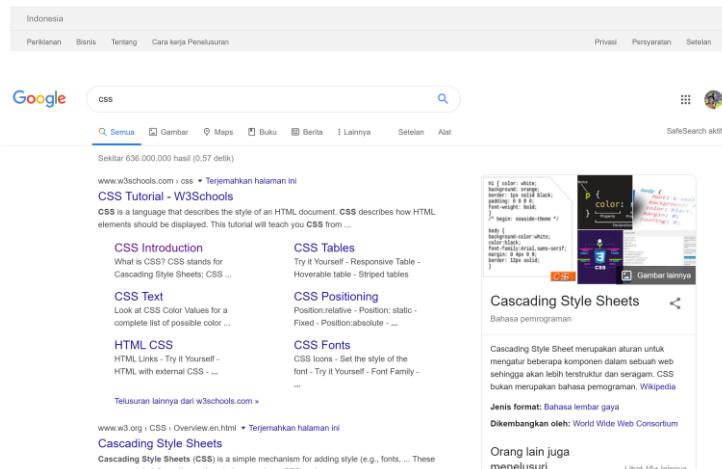
Umumnya suatu aplikasi web terdiri atas beberapa halaman web yang dapat berjumlah di bawah sepuluh, puluhan bahkan lebih. antarmuka pada suatu aplikasi web paling tidak memiliki dua tampilan yang berbeda. Contoh pertama adalah aplikasi web Facebook di bawah ini.





*Gambar 62. Antarmuka aplikasi web Facebook.*

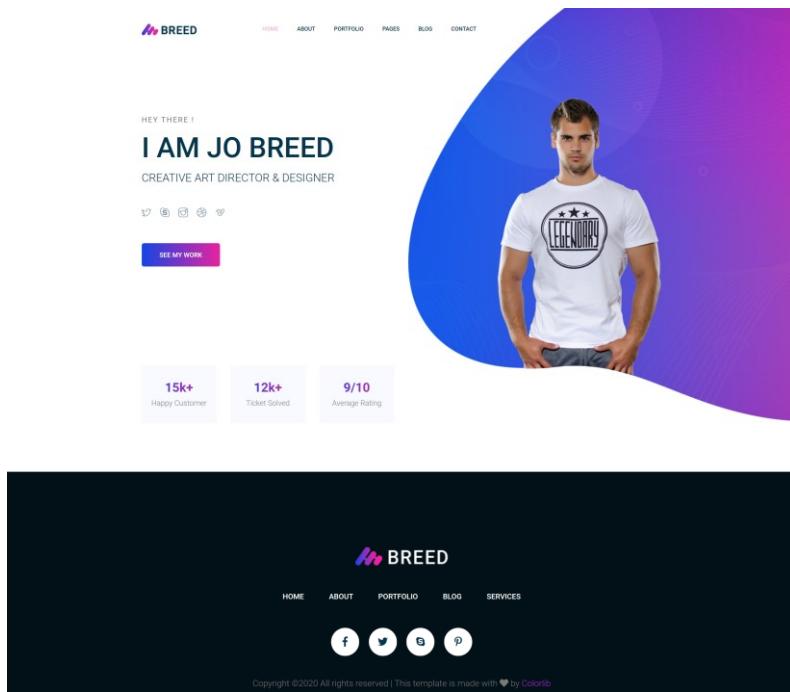
Contoh kedua adalah halaman web Google berikut ini.



*Gambar 63. Antarmuka aplikasi web pencarian Google.*

Halaman pertama tersebut adalah **landing page**, yaitu halaman yang pertama kali oleh pengunjung. Biasanya landing page ini hanya satu file saja. Setelah itu halaman berikutnya akan menggunakan antarmuka berbeda. Antarmuka ini umumnya digunakan oleh banyak file atau halaman dan dapat disebut sebagai **common page**.

Pada sub bab ini akan diberikan langkah-langkah yang umum dilakukan untuk membuat template common page agar bisa digunakan pada sebagai antarmuka template utama. Berikut ini adalah antarmuka yang akan dibuat sebagai latihan melakukan tata letak dengan CSS.



*Gambar 64. Antarmuka template utama.*

File-file gambar untuk background, banner dan icon tersedia pada folder `bab_4_css_layout`. Penulisan CSS pada sub bab ini menggunakan tata cara Internal CSS.

## *Background*

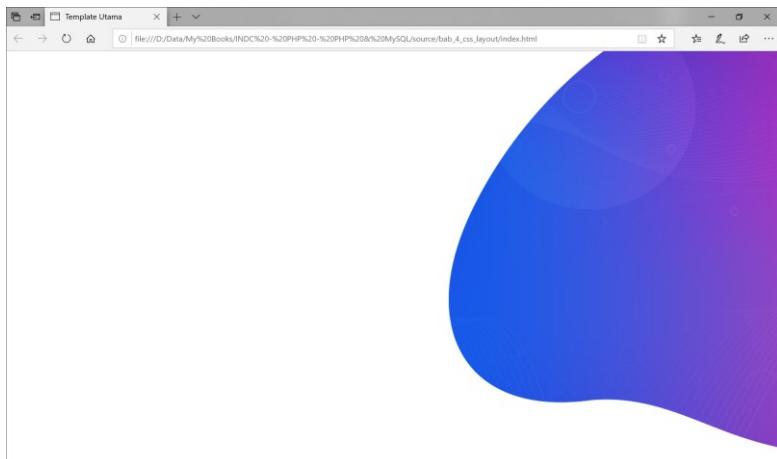
Terlihat pada gambar terdapat background di belakang gambar orang. Untuk menambahkan digunakan file home-banner.png pada folder images/banner. Berikut adalah kode HTML dan CSS yang digunakan.

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Template Utama</title>
    <style>
      body {
        font-family: Roboto;
        background-image: url(images/banner/home-
banner.png);
        background-repeat: no-repeat;
        background-position: top center;
        margin: 0;
        padding: 0;
      }
    </style>
  </head>

  <body>

  </body>
</html>
```

Hasilnya dapat dilihat pada gambar berikut ini.



*Gambar 65. Menambahkan background.*

## Rangka Utama

Jika diperhatikan antara logo pada kiri atas dan gambar orang, terlihat keduanya tidak rapat ke sisi kiri dan kanan. Namun terletak di tengah. Artinya ada kotak dengan ukuran tertentu yang berada di tengah.

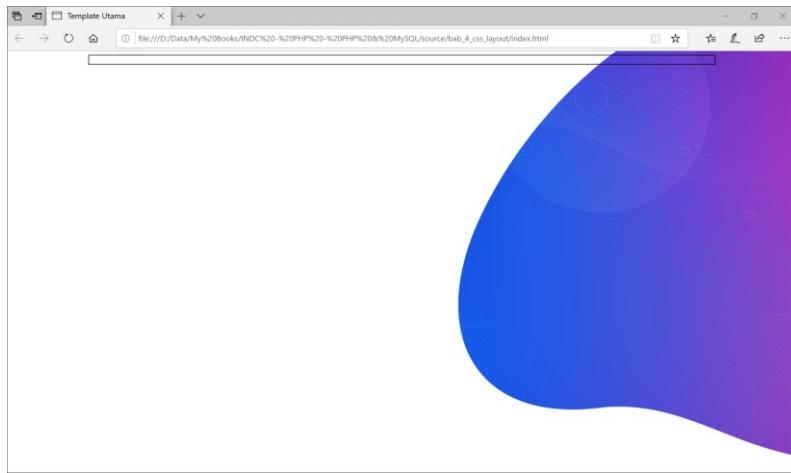
Untuk membuat kotak ini digunakan tag <div> dengan kode CSS sebagai berikut.

```
index.html
<!DOCTYPE html>
<html>
    <head>
        <title>Template Utama</title>
        <style>
            body {
                font-family: Roboto;
                background-image: url(images/banner/home-
banner.png);
                background-repeat: no-repeat;
                background-position: top left;
                margin: 0;
                padding: 0;
            }

            #main_box {
                width: 90%;
                margin: auto;
                border: solid 1px #000000;
            }
        </style>
    </head>

    <body>
        <div id="main_box">
            &nbsp;
        </div>
    </body>
</html>
```

Pada kode di atas ditambahkan tag <div> dengan atribut id="main\_box", selanjutnya style elemen ini diatur dengan menambahkan style dengan selector #main\_box. Maka hasilnya dapat dilihat pada gambar di bawah ini.



*Gambar 66. Menambahkan rangka utama.*

Untuk sementara elemen ini menggunakan border. Setelah antarmuka selesai maka property untuk menambahkan border akan dihapus.

### *Logo*

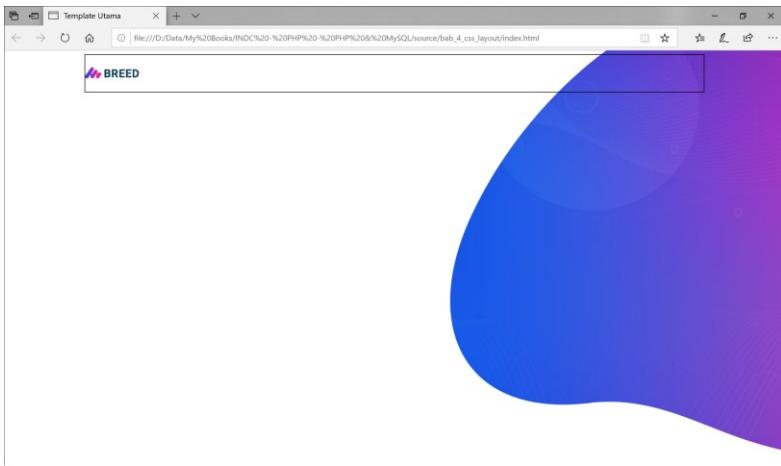
Langkah selanjutnya adalah menambah logo. Logo yang digunakan adalah file dengan nama logo.png yang disimpan pada folder images. Umumnya logo pada suatu halaman web dapat diklik dengan tujuan kembali ke halaman awal atau home page.

Berikut adalah kode yang digunakan untuk membuat logo.

```
index.html
<!DOCTYPE html>
<html>
  <head>
    <title>Template Utama</title>
    <style>
      body {
        font-family: Roboto;
        background-image: url(images/banner/home-
banner.png);
        background-repeat: no-repeat;
        background-position: top center;
        margin: 0;
        padding: 0;
      }
      #main_box {
```

```
        width: 90%;  
        margin: auto;  
        border: solid 1px #000000;  
    }  
  
    #main_logo {  
        float: left;  
    }  
  
    .separator {  
        clear:both;  
    }  
</style>  
</head>  
  
<body>  
    <div id="main_box">  
        <div id="main_logo">  
            <a href="#"><h1></h1></a>  
        </div>  
        <div class="separator"></div>  
    </div>  
</body>  
</html>
```

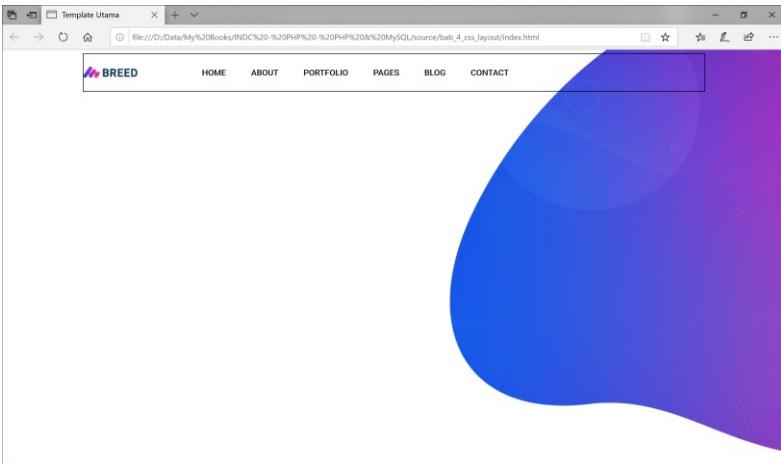
Pada kode di atas ditambahkan tag `<div>` dengan atribut `id="main_logo"`. Selanjutnya ditambahkan style untuk mengatur posisi dari elemen ini. selain itu juga ditambahkan elemen `<div>` dengan atribut `class="separator"`. Elemen ini berfungsi untuk menetralisir efek dari penggunaan property `float` pada style `#main_logo`. Berikut adalah hasil dari kode di atas.



*Gambar 67. Menambahkan logo.*

### *Menu*

Selanjutnya akan ditambahkan menu di samping logo. Untuk membuat menu umumnya digunakan elemen list dengan menggunakan tag <ul> dan <li>. Berikut ini adalah menu yang dihasilkan dengan menggunakan elemen list.



*Gambar 68. Menambahkan menu di samping logo.*

Berikut adalah kode HTML dan CSS untuk membuat menu di atas.

```
index.html
```

```
<!DOCTYPE html>
<html>
    <head>
        <title>Template Utama</title>
        <style>
            body {
                font-family: Roboto;
                background-image: url(images/banner/home-
banner.png);
                background-repeat: no-repeat;
                background-position: top left;
                margin: 0;
                padding: 0;
            }

            #main_box {
                width: 90%;
                margin: auto;
                border: solid 1px #000000;
            }

            #main_logo {
                float: left;
            }

            ul#main_menu {
                float: left;
                margin-left: 100px;
                margin-top: 30px;
                font-size: 11px;
                font-weight: bolder;
            }

            ul#main_menu li {
                list-style: none;
                display: inline;
                margin-left: auto;
                margin-right: 50px;
            }

            .separator {
                clear:both;
            }
        </style>
    </head>

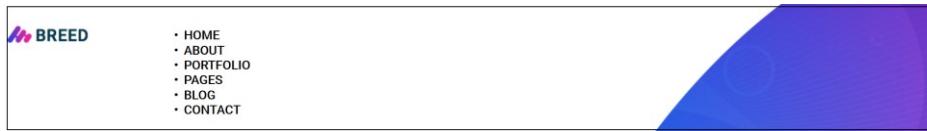
    <body>
        <div id="main_box">
            <div id="main_logo">
                <a href="#"><h1></h1></a>
            </div>
        </div>
```

```

<ul id="main_menu">
    <li>HOME</li>
    <li>ABOUT</li>
    <li>PORTFOLIO</li>
    <li>PAGES</li>
    <li>BLOG</li>
    <li>CONTACT</li>
</ul>
<div class="separator"></div>
</div>
</body>
</html>

```

Yang pertama diperhatikan adalah elemen `<ul>` dengan atribut `id="main_menu"`. Kemudian didalamnya dapat dilihat item dari list yang dibuat dengan menggunakan elemen `<li>`. Jika tambah menambahkan CSS maka kode di atas terlihat sebagai berikut ini.



*Gambar 69. Elemen list tanpa style.*

Oleh karena itu perlu ditambahkan CSS untuk mengatur tata letak elemen ini agar terlihat seperti pada Gambar 68. Kode CSS pertama yang digunakan adalah `ul#main_menu` yang berisi property-property CSS untuk mengatur tata letak elemen `<ul>`.

Selanjutnya adalah mengatur tata letak elemen `<li>` dengan menggunakan style pada `ul#main_menu li`. Pada style tersebut dapat dilihat property `list-style` untuk menghilangkan nomor atau bullet pada item list. Kemudian digunakan property `display: inline` untuk membuat item list ini tersusun ke samping (bukan ke bawah). Yang terakhir digunakan property `margin` untuk mengatur jarak antar item list.

## *Hyperlink*

Menu berfungsi untuk mengantarkan user ke halaman lain ketika salah satu item menu diklik. Oleh karena itu item-item list di atas akan diubah menjadi hyperlink dengan menambahkan elemen `<a>`.

Berikut adalah kode HTML pada elemen list setelah ditambahkan elemen hyperlink.

```
<ul id="main_menu">
<li><a href="#" class="main_menu">HOME</a></li>
<li><a href="#" class="main_menu">ABOUT</a></li>
<li><a href="#" class="main_menu">PORTFOLIO</a></li>
<li><a href="#" class="main_menu">PAGES</a></li>
<li><a href="#" class="main_menu">BLOG</a></li>
<li><a href="#" class="main_menu">CONTACT</a></li>
</ul>
```

Hasilnya dapat dilihat pada gambar berikut.



*Gambar 70. Hyperlink pada menu tanpa style.*

Style dari hyperlink di atas adalah standar dimana teks berwarna biru dengan dekorasi garis bawah. Untuk mendapatkan tampilan menu seperti pada Gambar 64 maka elemen <a> perlu diberikan style sebagai berikut.

```
a.main_menu {
    text-decoration: none;
    color: #000000;
}

a.main_menu:hover {
    text-decoration: none;
    color: #dd33ff;
}
```

Pada kode di atas digunakan selector a.main\_menu. Untuk menghilangkan dekorasi underline pada teks digunakan property text-decoration: none. Sedangkan style dengan selector a.main\_menu:hover dimaksudnya style hyperlink ketika mouse diarahkan ke hyperlink. Saat mouse diarahkan ke hyperlink maka warna teks akan menjadi warna ungu muda.

### *Content*

Sub-bab di atas adalah langkah-langkah untuk membuat header. Setelah seluruh langkah di atas dikerjakan, maka kotak header dapat dihilangkan dengan cara menghilangkan baris property border pada style dengan selector #main\_box. Sehingga akhirnya menjadi berikut ini.

```
#main_box {
    width: 80%;
    margin: auto;
```

```
}
```

Langkah selanjutnya adalah membuat blok untuk content yang terdiri atas blok kiri yang berisi nama dan informasi profile. Sedangkan blok kanan berisi gambar atau foto.

Berikut adalah kode HTML dan CSS yang ditulis untuk membuat antarmuka seperti paparan di atas.

```
index.html
<!DOCTYPE html>
<html>
    <head>
        <title>Template Utama</title>
        <style>
            body {
                font-family: Roboto;
                background-image: url(images/banner/home-
banner.png);
                background-repeat: no-repeat;
                background-position: top center;
                margin: 0;
                padding: 0;
            }

            #main_box {
                width: 80%;
                margin: auto;
                /* border: solid 1px #000000; */
            }

            #main_logo {
                float: left;
            }

            ul#main_menu {
                float: left;
                margin-left: 50px;
                margin-top: 30px;
                font-size: 11px;
                font-weight: bolder;
            }

            ul#main_menu li {
                list-style: none;
                display: inline;
                margin-left: auto;
                margin-right: 50px;
            }

            a.main_menu {
```

```

        text-decoration: none;
        color: #000000;
    }

    a.main_menu:hover {
        text-decoration: none;
        color: #dd33ff;
    }

    #content_left {
        float: left;
        width: 45%;
        border: solid 1px #000000;
        margin-top: 100px;
    }

    #content_right {
        float: right;
        width: 374px;
        height: 635px;
        border: solid 1px #000000;
        margin-top: 100px;
        background-image: url(images/banner/home-right.png);
        background-repeat: no-repeat;
        background-position: top right;
        background-size: 92%;
    }

    .separator {
        clear:both;
    }

```

</style>

```

</head>

<body>
    <div id="main_box">
        <div id="main_logo">
            <a href="#"><h1></h1></a>
        </div>
        <ul id="main_menu">
            <li><a href="#" class="main_menu">HOME</a></li>
            <li><a href="#" class="main_menu">ABOUT</a></li>
            <li><a href="#" class="main_menu">PORTFOLIO</a></li>
            <li><a href="#" class="main_menu">PAGES</a></li>
            <li><a href="#" class="main_menu">BLOG</a></li>
            <li><a href="#" class="main_menu">CONTACT</a></li>
        </ul>
        <div class="separator"></div>

        <div id="content_left">
            &nbsp;

```

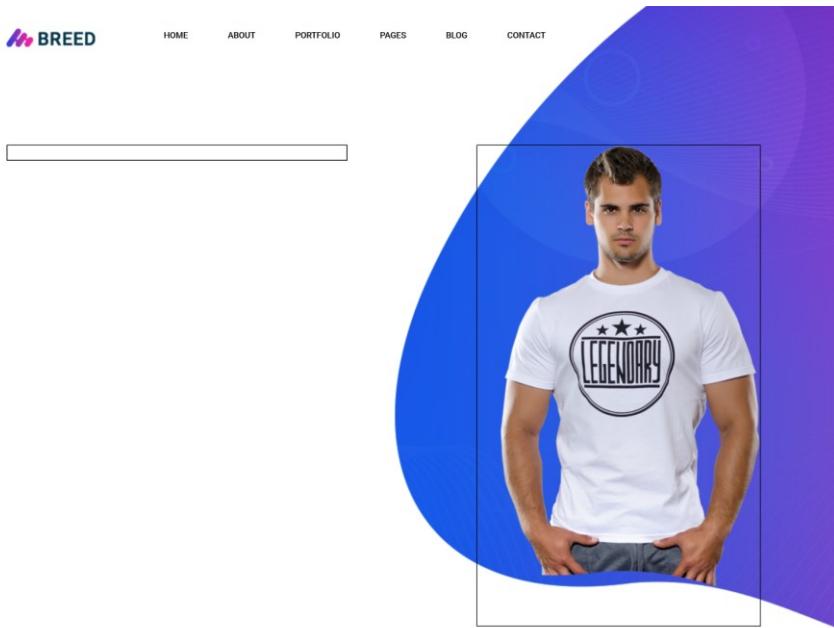
```
</div>

<div id="content_right">
    &nbsp;
</div>
<div class="separator"></div>
</div>
</body>
</html>
```

Pada kode HTML di atas ditambahkan dua elemen `<div>` dengan nilai atribut `id` yang berbeda yaitu `content_left` dan `content_right`. Kemudian pada diberikan style pada blok CSS untuk selector `#content_left` dan `#content_right`. Lebar masing-masing elemen ini adalah 45% dari lebar rangka utama yaitu elemen `<div>` dengan `id="main_box"`.

Untuk elemen `<div>` dengan `id="content_left"` akan rapat kiri, sedangkan elemen `<div>` dengan `id="content_right"` akan rapat kanan. Hal ini karena menggunakan property `float` dengan nilai sesuai arah yang diinginkan.

Untuk sementara kedua elemen ini diberikan border, sehingga dapat dilihat tampilan seperti pada .



*Gambar 71. Content kiri dan kanan.*

Selanjutnya ditambahkan foto orang pada content sebelah kanan. Dan hasilnya dapat dilihat pada gambar di atas.

Langkah berikutnya adalah mengisi content sebelah kiri. Pertama ditambahkan teks dengan mengisi elemen `<div>` dengan atribut `id="content_left"` dengan elemen-elemen berikut.

```
<div id="content_left">
  <p>HEY THERE!</p>
  <h2>I AM JO BREED</h2>
  <h3>CREATIVE ART DIRECTOR & DESIGNER</h3>
</div>
```

Selanjutnya adalah memberikan style pada elemen `<p>`, `<h2>` dan `<h3>` yang berada di dalam elemen content bagian kiri ini dengan kode CSS berikut ini.

```
#content_left p {
  margin: 0 0 0 0;
  color: #797979;
  font-family: Arial;
  font-size: 16px;
}
```

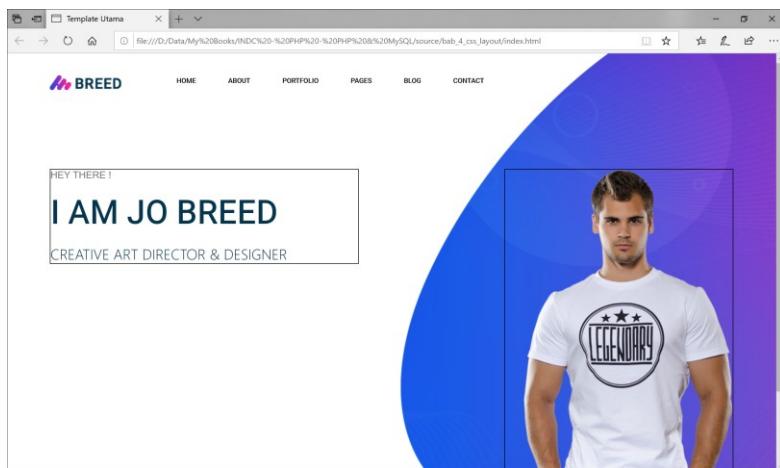
```

#content_left h2 {
    margin: 20px 0 0 0;
    color: #05364d;
    font-size: 40pt;
}

#content_left h3 {
    margin: 20px 0 0 0;
    color: #05364d;
    font-size: 18pt;
    font-family: 'Segoe UI';
    font-weight: 100;
}

```

Hasilnya dapat dilihat pada gambar berikut.



*Gambar 72. Menambahkan teks pada content sebelah kiri.*

Setelah teks ditambahkan maka akan ditambahkan hyperlink berupa gambar ke web social media. Daftar hyperlink dibuat dengan memanfaatkan elemen list. Berikut adalah kode HTML untuk membuat hyperlink ini.

```

<ul>
    <li>
        <a href="#">
            
        </a>
    </li>
    <li>
        <a href="#">
            
        </a>
    </li>
    <li>

```

```
<a href="#">

</a>
</li>
<li>
<a href="#">

</a>
</li>
<li>
<a href="#">

</a>
</li>
</ul>
```

Selanjutnya adalah memberikan style untuk elemen-elemen tersebut dengan kode CSS berikut ini.

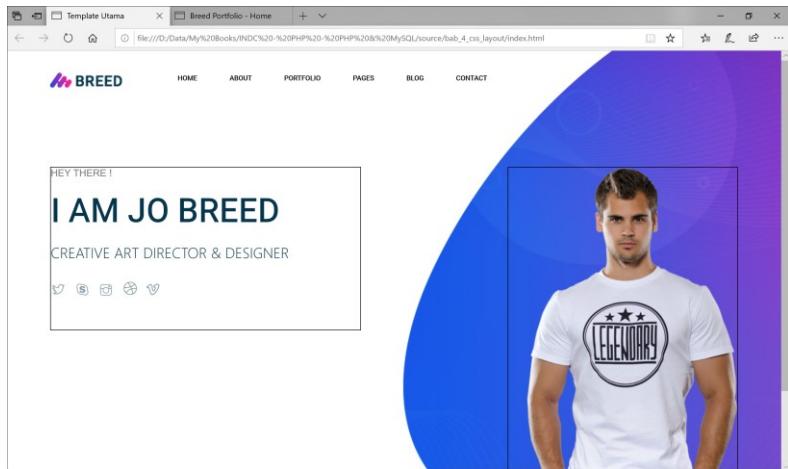
```
#content_left ul {
    margin: 30px 0 50px 0;
    padding: 0 0 0 0;
}

#content_left ul li {
    list-style: none;
    display: inline;
    margin-left: auto;
    margin-right: 10px;
}

#content_left ul li a {
    text-decoration: none;
}

#content_left ul li a img {
    width: 25px;
}
```

Hasilnya dapat dilihat pada gambar berikut ini.



*Gambar 73. Menambahkan hyperlink berupa icon media sosial.*

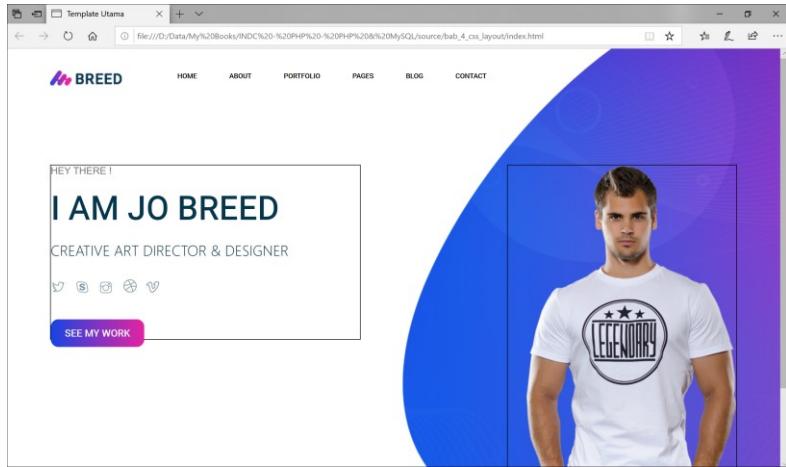
Yang terakhir adalah menambah tombol dengan gradient warna biru ke ungu. Tombol ini akan memperlihatkan animasi sederhana berupa perubahan gradient warna ungu ke biru ketika mouse diarahkan ke tombol. Berikut ini kode HTML untuk membuat tombol dengan menggunakan elemen `<a>`.

```
<a href="#" class="button_grad">SEE MY WORK</a>
```

Selanjutnya diberikan style dengan kode CSS berikut ini.

```
.button_grad {  
    border-radius: 13px;  
    color: #FFFFFF;  
    text-decoration: none;  
    font-weight: bold;  
    padding: 13px 23px 13px 23px;  
    background-image: linear-  
gradient(to right, #1744e5 0%, #8334c2 51%, #e324a2 100%)  
}  
  
.button_grad:hover {  
    padding: 13px 23px 13px 23px;  
    background-image: linear-  
gradient(to left, #1744e5 0%, #8334c2 51%, #e324a2 100%)  
}
```

Hasilnya dapat dilihat pada gambar berikut ini.



*Gambar 74. Tombol SEE MY WORK.*

Selanjutnya adalah adalah menghilangkan border warna hitam.

## Footer

Footer adalah area berwarna hitam yang terletak pada bagian paling bawah halaman web. Berikut adalah kode HTML untuk membuat blok footer. Kemudian elemen logo, menu dan hyperlink berupa menu di dalam blok footer tersebut.

```
<div id="footer_box">
    <div id="footer_content">
        <div id="logo_footer">
            
        </div>

        <ul id="menu_footer">
            <li><a href="#">HOME</a></li>
            <li><a href="#">ABOUT</a></li>
            <li><a href="#">PORFOLIO</a></li>
            <li><a href="#">BLOG</a></li>
            <li><a href="#">SERVICES</a></li>
        </ul>

        <ul id="icon_footer">
            <li>
                <a href="#">
                    
                </a>
            </li>
        </ul>
    </div>
</div>
```

```

<li>
    <a href="#">
        
    </a>
</li>
<li>
    <a href="#">
        
    </a>
</li>
<li>
    <a href="#">
        
    </a>
</li>
</ul>
</div>
</div>

```

Selanjutnya dibuat kode CSS untuk mengatur tata letak elemen-elemen di atas.

```

#footer_box {
    margin: 100px 0 0 0;
    padding: 50px 0 50px 0;
    width: 100%;
    height: 200px;
    background-color: #000000;
}

#footer_content {
    width: 100%;
    text-align: center;
    display: inline-block;
    margin: auto;
}

ul#menu_footer {
    margin: 30px 0 30px 0;
    padding: 0 0 0 0;
}

ul#menu_footer li {
    list-style: none;
    display: inline;
    margin-left: auto;
    margin-right: 30px;
}

ul#menu_footer li a {
    color: #FFFFFF;
}

```

```
text-decoration: none;
font-size: 12px;
font-weight: bold;
}

ul#menu_footer li a:hover {
    color: #FFFFFF;
    text-decoration: underline;
    font-size: 12px;
    font-weight: bold;
}

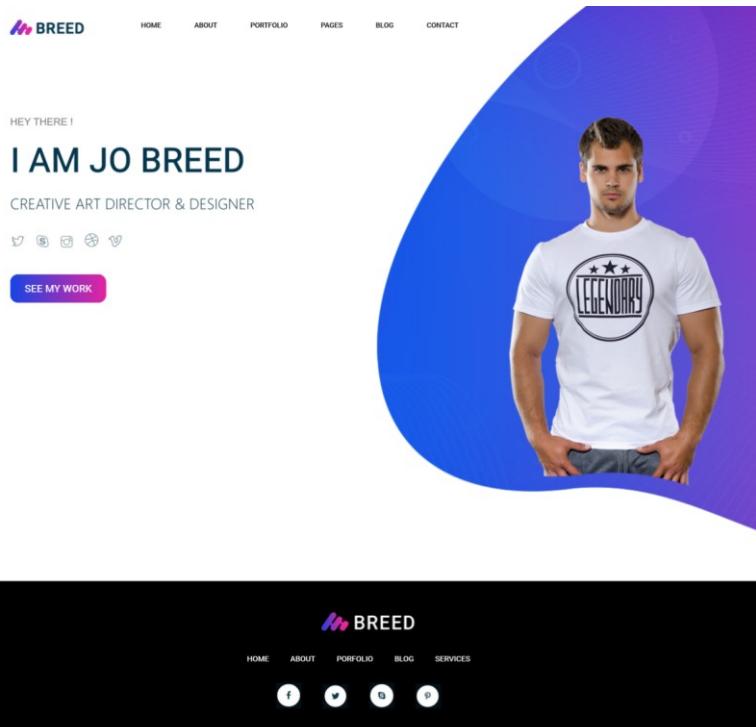
ul#icon_footer {
    margin: 0;
    padding: 0;
}

ul#icon_footer li {
    list-style: none;
    display: inline;
    margin-left: auto;
    margin-right: 30px;
}

ul#icon_footer li a {
    text-decoration: none;
}

ul#icon_footer li a img {
    width: 40px;
}
```

Hasilnya dapat dilihat pada gambar berikut ini.

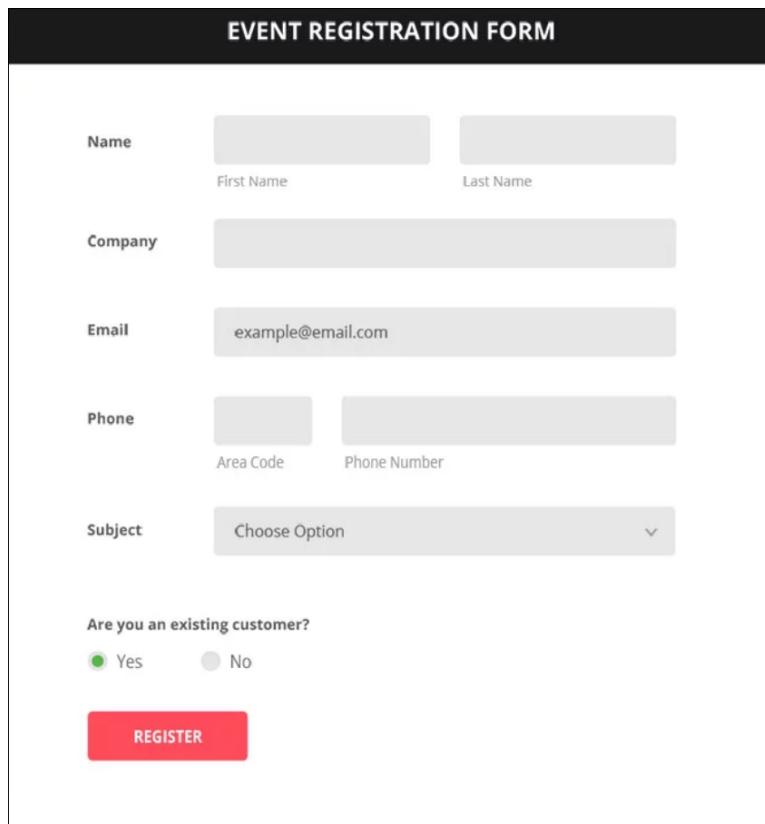


*Gambar 75. Antarmuka utuh web profile.*

## Form Registrasi

Contoh selanjutnya adalah melakukan pengaturan tata letak untuk form input data. Berikut adalah tampilan antarmuka form input data yang dibuat pada sub bab ini.

Penulisan CSS pada sub bab ini menggunakan cara Internal CSS untuk memudahkan saat melakukan menambah dan edit kode HTML dan CSS. Namun jika antarmuka sudah selesai dibuat sebaiknya kode CSS dipisahkan untuk disimpan pada file CSS.



The image shows a clean, modern event registration form. At the top, it has a dark header bar with the text "EVENT REGISTRATION FORM" in white capital letters. Below this, the form fields are organized into rows. The first row contains "Name" with two adjacent input fields for "First Name" and "Last Name". The second row contains "Company" with a single input field. The third row contains "Email" with a single input field containing the placeholder "example@email.com". The fourth row contains "Phone" with two adjacent input fields for "Area Code" and "Phone Number". The fifth row contains "Subject" followed by a dropdown menu with the placeholder "Choose Option". Below these fields is a question "Are you an existing customer?" with two radio button options: "Yes" (selected) and "No". At the bottom is a large red "REGISTER" button.

*Gambar 76. Form registrasi event.*

Untuk membuat antarmuka seperti di atas maka dilakukan langkah-langkah sebagai berikut.

#### *Window Form*

Langkah pertama membuat window atau rangka form dengan kode HTML dan CSS sebagai berikut.

```
registrasi.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Form Registrasi Event</title>
    <style>
      body {
        background-color: grey;
```

```
        color: #000000;
        font-family: Verdana;
        font-size: 12px;
    }

    #main_content {
        width: 50%;
        margin: auto;
    }

    .form_box {
        border: solid 1px #000000;
        background-color: #FFFFFF;
        height: 550px;
        margin: 50px 0 0 0;
    }

    .form_title {
        border: solid 1px #000000;
        background-color: #000000;
        color: #FFFFFF;
        height: 32px;
        font-weight: bold;
        text-align: center;
        padding-top: 11px;
    }

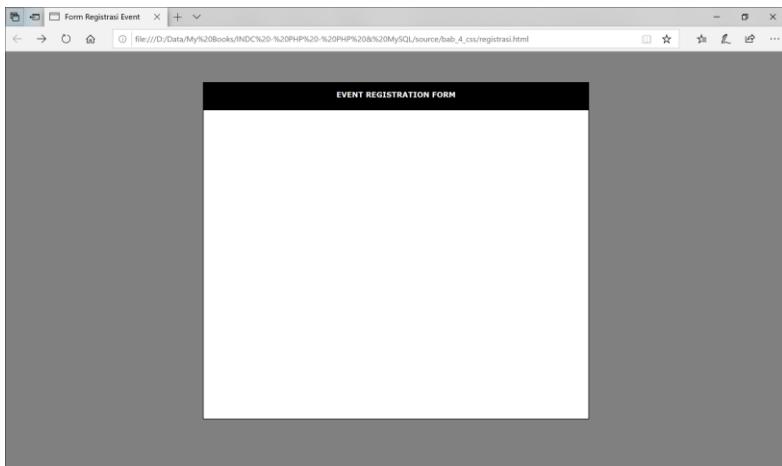
    .form_content {
        width: 90%;
        margin: auto;
    }
</style>
</head>

<body>
    <div id="main_content">
        <div class="form_box">
            <div class="form_title">
                EVENT REGISTRATION FORM
            </div>
            <div class="form_content">

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

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

Hasilnya dapat dilihat pada gambar berikut ini.



*Gambar 77. Window form input.*

Untuk membuat window seperti gambar di atas digunakan div dengan atribut id="main\_content" untuk membuat rangka utama dengan lebar 50% dari ukuran layar.

Selanjutnya di dalam elemen ini ditambahkan elemen <div> dengan class="form\_box" yang menjadi rangka window. Elemen ini diberikan style dengan form\_box sebagai selector. Style yang diberikan untuk elemen ini adalah memberikan border warna hitam dan lebar sesuai ukuran elemen induknya dan tinggi adalah 550px.

Berikutnya adalah membuat header untuk window dengan menggunakan elemen <div> dengan atribut class="form\_title". Pada kode CSS untuk class ini dapat dilihat diberikan border dan background dengan warna hitam dan tulisan warna putih.

Di bawah elemen untuk header ditambahkan elemen <div> dengan atribut class="form\_content".

## Elemen Form Input

Langkah selanjutnya adalah menambahkan elemen input data di dalam elemen <div> dengan class="form\_content". Berikut ini adalah kode HTML yang berisi elemen untuk input data.

```
<form style="display:none;">
    <label>Name</label>
    <input type="text" name="first_name">
    <input type="text" name="last_name">
    <div class="separator"></div>

    <label>Company</label>
    <input type="text" name="company">
    <div class="separator"></div>

    <label>Email</label>
    <input type="text" name="email">
    <div class="separator"></div>

    <label>Phone</label>
    <input type="text" name="area_code">
    <input type="text" name="phone_number">
    <div class="separator"></div>

    <label>Subject</label>
    <select name="subject">
        <option value="">Choose Option</option>
        <option value="1">Subject 1</option>
        <option value="2">Subject 2</option>
    </select>
    <div class="separator"></div>

    <label>Are you an existing customer?</label>
    <ul>
        <li>
            <input type="radio" name="is_existing" value="yes">
        </li>
        <li>Yes</li>
        <li>
            <input type="radio" name="is_existing" value="no">
        </li>
        <li>No</li>
    </ul>

    <input type="button" value="REGISTER">
</form>
```

Hasilnya dapat dilihat pada gambar berikut ini.

**EVENT REGISTRATION FORM**

Name

Company

Email

Phone

Subject

Are you an existing customer?

- 
- Yes
- 
- No

*Gambar 78. Elemen input data tanpa CSS.*

Kode HTML di atas akan dimodifikasi dengan pemberian atribut class pada elemen-elemennya. Kemudian class akan digunakan sebagai selector dan diberikan style untuk masing-masing selector.

Berikut ini adalah kode HTML dan CSS lengkap untuk membuat antarmuka seperti yang direncanakan.

```
index.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Form Registrasi Event</title>
    <style>
      body {
        background-color: grey;
        color: #000000;
        font-family: Verdana;
        font-size: 12px;
      }

      #main_content {
        width: 50%;
        margin: auto;
      }
    </style>
  </head>
  <body>
    <div id="main_content">
      <form>
        Name  
        Company 
        Email 
        Phone  
        Subject 
        Are you an existing customer?
        <ul style="list-style-type: none;">
          <li><input type="radio"/></li>
          <li><input checked="" type="radio"/> Yes</li>
          <li><input type="radio"/></li>
          <li><input type="radio"/> No</li>
        </ul>
        <input type="button" value="REGISTER"/>
      </form>
    </div>
  </body>
</html>
```

```
.form_box {  
    border: solid 1px #000000;  
    background-color: #FFFFFF;  
    height: 550px;  
    margin: 50px 0 0 0;  
}  
  
.form_title {  
    border: solid 1px #000000;  
    background-color: #000000;  
    color: #FFFFFF;  
    height: 32px;  
    font-weight: bold;  
    text-align: center;  
    padding-top: 11px;  
}  
  
.form_content {  
    width: 90%;  
    margin: auto;  
}  
  
.form_label {  
    width: 75px;  
    margin: 20px 0 0 0;  
    display: inline-block;  
    font-weight: bold;  
}  
  
.form_label_horizontal {  
    width: 300px;  
    margin: 20px 0 0 0;  
    display: inline-block;  
    font-weight: bold;  
}  
  
.form_textbox_short {  
    width: 200px;  
    height: 30px;  
    background-color: #dadada;  
    border: solid 1px #bebebe;  
    border-radius: 4px;  
    margin-left: 23px;  
}  
  
.form_textbox {  
    width: 430px;  
    height: 30px;  
    background-color: #dadada;  
    border: solid 1px #bebebe;
```

```

        border-radius: 4px;
        margin-left: 23px;
    }

    .form_radio_horizontal {
        margin: 0;
        padding: 0;
    }

    .form_radio_horizontal li {
        list-style: none;
        display: inline;
    }

    .form_radio_horizontal li.item_text {
        margin-right: 30px;
    }

    .form_button_red {
        color:#FFFFFF;
        background-color: #ff5353;
        border: solid 1px #ff3434;
        padding: 10px 30px 10px 30px;
        border-radius: 5px;
    }

    .separator {
        clear:both;
        margin: 25px 0 25px 0;
    }

```

</style>

</head>

<body>

<div id="main\_content">

<div class="form\_box">

<div class="form\_title">

EVENT REGISTRATION FORM

</div>

<div class="form\_content">

<form>

<div class="separator"></div>

<label class="form\_label">Name</label>

<input type="text" name="first\_name" class="form\_textbox\_short">

<input type="text" name="last\_name" class="form\_textbox\_short">

<div class="separator"></div>

<label class="form\_label">Company</label>

```

        <input type="text" name="company" class="for
m_textbox">
        <div class="separator"></div>

        <label class="form_label">Email</label>
        <input type="text" name="email" class="form_
textbox">
        <div class="separator"></div>

        <label class="form_label">Phone</label>
        <input type="text" name="area_code" class="form_
textbox_short">
        <input type="text" name="phone_number" class
="form_textbox_short">
        <div class="separator"></div>

        <label class="form_label">Subject</label>
        <select name="subject" class="form_textbox">
            <option value="">Choose Option</option>
            <option value="1">Subject 1</option>
            <option value="2">Subject 2</option>
        </select>
        <div class="separator"></div>

        <label class="form_label_horizontal">Are you
an existing customer?</label>
        <ul class="form_radio_horizontal">
            <li>
                <input type="radio" name="is_existin
g" value="yes" checked>
            </li>
            <li class="item_text">Yes</li>
            <li>
                <input type="radio" name="is_existin
g" value="no">
            </li>
            <li class="item_text">No</li>
        </ul>
        <div class="separator"></div>

        <input type="button" value="REGISTER" class=
"form_button_red">
    </form>
</div>
</div>
</body>
</html>

```

Hasilnya dapat dilihat pada gambar berikut.

*Gambar 79. Form input data yang telah diberikan CSS.*

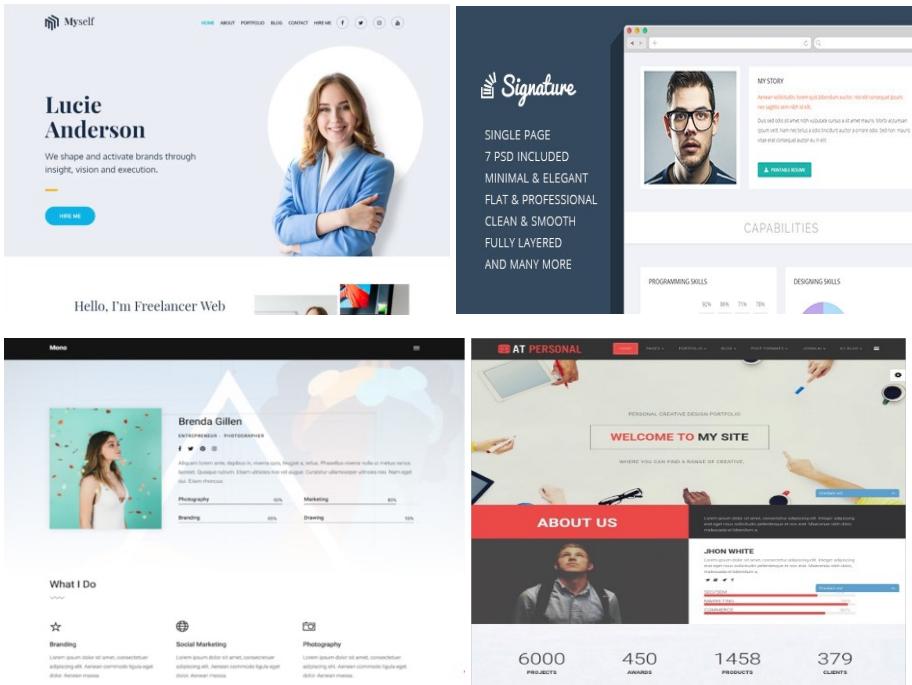
## CSS Library & Framework

Under development

## Project

### Web Profile Pribadi dengan CSS

Proyek ini membuat sebuah halaman web (1 file HTML) yang lengkap dengan style CSS yang ditulis dengan cara internal CSS. Desain antarmuka dapat diambil pada internet seperti contoh-contoh berikut ini.



*Gambar 80. Contoh-contoh antarmuka personal web.*

Pilih sebuah gambar seperti contoh di atas atau contoh-contoh lainnya, kemudian buat sebuah (1 halaman saja) halaman web untuk membuat antarmuka yang mirip. Usahakan jangan mengubah warna dan tata letak. Yang dapat diubah adalah gambar dan teks yang disesuaikan dengan informasi pribadi kalian.

Selain 1 file HTML juga folder images yang berisi gambar-gambar yang digunakan pada file HTML tersebut.

## Web Personal dengan Bootstrap

Project ini bertujuan agar pembaca terbiasa menggunakan CSS Library seperti Bootstrap. Berikut adalah contoh web personal yang menggunakan Bootstrap:

- <https://colorlib.com/wp/free-personal-website-templates/>
- <https://uicookies.com/free-bootstrap-personal-website-template/>

dan masih banyak lagi template web personal yang menggunakan Bootstrap. Pilih salah satu template tersebut dan ubah gambar dan teks serta hal-hal yang dianggap perlu agar isinya sesuai dengan informasi personal kalian.

# 5

## **JavaScript**

Halaman web dibuat dengan kode HTML kemudian dipercantik dengan CSS. Namun untuk membuatnya dapat bergerak, bernafas, bernyanyi dan menari, kita perlu menambahkan sebuah atau dua kode bahasa pemrograman didalamnya. Salah satu bahasa pemrograman yang dapat melakukan itu adalah JavaScript.

---

### **Pendahuluan**

JavaScript adalah bahasa pemrograman yang awalnya dikembangkan oleh Netscape Communication. Dan sekarang menjadi standar pada hampir seluruh web browser. Berikut ada beberapa hal yang perlu diketahui tentang JavaScript, yaitu:

- JavaScript adalah benar-benar bahasa pemrograman seperti halnya Python, C# dan Visual Basic.
- JavaScript bukan Java. Java awalnya dikembangkan oleh Sun Microsystems.
- JavaScript adalah bahasa pemrograman scripting, hal ini membuat tidak perlu proses kompilasi. Jenis bahasa ini memungkinkan program hanya berisi sebaris kode yang terdiri atas sebuah fungsi saja seperti pada contoh di sub bab Aturan Penulisan. Tidak seperti bahasa pemrograman Java yang sebuah file program harus memiliki bagian untuk menambahkan namespace, deklarasi nama class dan lain-lain.

---

### **Aturan Penulisan**

Untuk menulis kode program JavaScript pada halaman web perlu mengikuti aturan tertentu.

## Internal

---

Aturan penulisan dengan cara ini adalah dengan menulis kode pemrograman JavaScript di dalam halaman web bercampur dengan kode HTML. Penulisan kode programnya berada di dalam tag <script> seperti contoh di bawah ini.

```
hello_javascript.html
<!DOCTYPE html>
<html>
    <head>
        <title>Hello JavaScript</title>
        <script type="text/javascript">
            alert("Hello World dari dalam tag head");
        </script>
    </head>

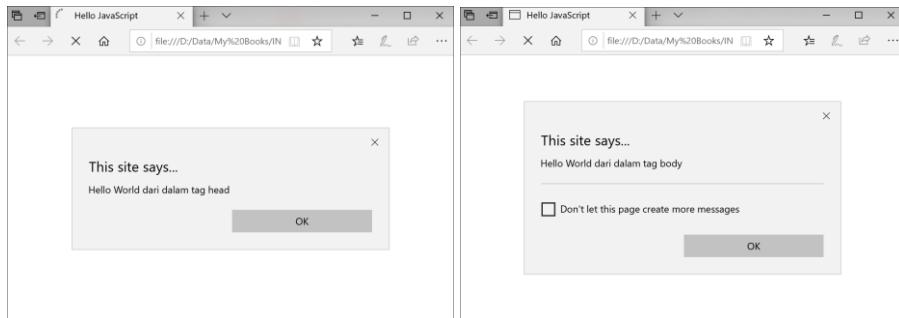
    <body>
        <script>
            alert("Hello World dari dalam tag body");
        </script>
    </body>
</html>
```

Dari kode di atas dapat dilihat bahwa tag <script> dapat berada di mana saja, baik itu di dalam tag <head> atau <body>. Bahkan dimungkin juga untuk berada diluar tag <body>.

Pada contoh di atas penulisan tag <script> ada yang menggunakan atribut type="text/javascript" dan tidak. Pada HTML 5, penulisan atribut type adalah pilihan. Sehingga untuk menulis kode JavaScript dapat menggunakan salah satu cara di atas.

Selanjutnya dapat dilihat kode JavaScript ditulis diantara tag tersebut. Pada contoh di atas, digunakan fungsi alert. Akhir sebaris kode diakhiri dengan tanda titik koma atau semicolon (;).

Hasil di atas hasilnya akan ditampilkan message box sebanyak dua kali. Seperti pada gambar di bawah ini.



Gambar 81. Message box keluaran JavaScript.

## External

Aturan penulisan ini memisahkan kode dalam bentuk fungsi-fungsi ke suatu file terpisah. Script JavaScript disimpan pada file dengan ekstensi .js. Berikut adalah cara untuk menyematkan file ini pada suatu halaman web.

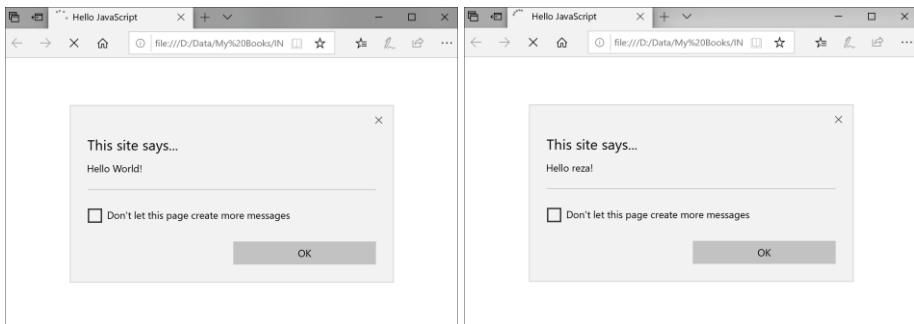
```
external 1.html
<!DOCTYPE html>
<html>
  <head>
    <title>Hello JavaScript</title>
    <script src="js/hello.js"></script>
  </head>
  <script>
    hello_world();
    say_hello("reza");
  </script>
  <body>
  </body>
</html>
```

Pada contoh di atas dapat dilihat tetap digunakan tag <script> namun digunakan atribut scr untuk menentukan nama file yang ingin disematkan. Dan berikut adalah isi dari file hello.js. Pada file js ini terdapat dua fungsi. Kemudian pada kode di atas dapat dilihat cara untuk memanggil kedua fungsi tersebut.

```
hello.js
function hello_world() {
  alert("Hello World!");
}

function say_hello(name) {
  alert("Hello " + name + "!");
}
```

Keluaran program di atas mengeluarkan message box sebanyak dua kali seperti pada gambar di bawah ini.



*Gambar 82. Message box dari fungsi JavaScript.*

## Dasar-Dasar Pemrograman

### Komentar

Untuk membuat komentar pada JavaScript digunakan dua cara seperti dilihat pada contoh kode berikut ini.

```
komentar_1.html
<!DOCTYPE html>
<html>
  <head>
    <title>Komentar</title>
  </head>
  <script>
    // komentar sebaris

    /*
    komentar multi baris
    */

    alert("Contoh pembuatan komentar");
  </script>
  <body>
  </body>
</html>
```

## Variable

---

JavaScript adalah tipe bahasa pemrograman yang tidak perlu mendeklarasikan tipe data pada suatu variable. Artinya sebuah variable dapat diisi dengan nilai dengan tipe data yang berbeda.

```
variable_1.html
<!DOCTYPE html>
<html>
  <head>
    <title>Variable</title>
  </head>
  <script>
    a = "hello";
    alert(a);
    a = 1;
    alert(a);
  </script>
  <body>
  </body>
</html>
```

Pada kode di atas dapat dilihat variable a awalnya diisi nilai tipe data string. kemudian variable a diisi kembali dengan nilai namun tipe datanya adalah integer.

Selain itu kadang untuk mendeklarasikan variable digunakan keyword var seperti contoh berikut ini.

```
variable_2.html
<!DOCTYPE html>
<html>
  <head>
    <title>Variable</title>
  </head>
  <script>
    var a = "hello";
    alert(a);
    var b = 1;
    alert(b);
  </script>
  <body>
  </body>
</html>
```

## Operasi Aritmatika

---

Tata cara melakukan operasi aritmatika di JavaScript sama dengan tata cara di bahasa pemrograman Java.

## Percabangan

---

Kata kunci yang digunakan untuk percabangan adalah if-else dan switch-case.

### **if-else**

Berikut adalah kata kunci dan sintaks yang digunakan untuk melakukan percabangan.

```
if(kondisi) {  
}
```

Kondisi di atas adalah operasi logika. Jika hasil operasi logika itu bernilai benar (true) maka kondisi memenuhi syarat. Dan kode di dalam kurung kurawal akan dieksekusi.

Untuk melakukan operasi logika digunakan operator logika seperti tabel di bawah ini. Misal diketahui  $x=5$ .

operator	keterangan	contoh	hasil
==	sama dengan	$x == 5$	false
		$x == 5$	true
		$x == "5"$	true
====	nilai dan tipe data sama	$x === 5$	true
		$x === "5"$	false
!=	tidak sama dengan	$x != 8$	true
!==	tidak sama dengan atau tidak sama tipe data	$x !== 5$	false
		$x !== "5"$	true
		$x !== 8$	true
>	lebih besar dari	$x > 8$	false
<	lebih kecil dari	$x < 8$	true
>=	lebih besar atau sama dengan	$x >= 8$	false
<=	lebih kecil atau sama dengan	$x <= 8$	true

Berikut adalah contoh percabangan jika diimplementasikan pada halaman web.

```
percabangan_1.html  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>Percabangan</title>  
    <script>  
      var a = 5;  
      var b = 5;  
      if(a == b) {
```

```
        alert("a dan b sama");
    }
</script>
</head>

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

Sintaks percabangan yang lain adalah sebagai berikut.

```
if(kondisi) {
    statement1;
} else {
    statement2;
}
```

Pada sintaks di atas terdapat sebuah kondisi. Jika kondisi benar maka statement1 akan dijalankan. Namun jika kondisi benar maka statement2 akan dijalankan. Berikut ini adalah contoh penggunaan sintaks ini.

```
percabangan_2.html
<!DOCTYPE html>
<html>
    <head>
        <title>Percabangan</title>
        <script>
            var a = 5;
            var b = 4;
            if(a == b) {
                alert("a dan b sama");
            } else {
                alert("a dan b tidak sama");
            }
        </script>
    </head>

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

Pada contoh di atas akan ditampilkan pesan "a dan b tidak sama". Hal ini dikarenakan kondisi "a == b" tidak benar (false) sehingga statement didalamnya tidak dijalankan. Oleh karena itu statement di dalam tanda kurung kurawal setelah kata kunci else yang dijalankan.

Sintaks yang lain adalah sebagai berikut.

```
if(kondisi1) {
    statement1;
```

```
    } else if(kondisi2) {
        statement2;
    }
} else {
    statement3;
}
```

Pada sintaks di atas dimiliki lebih dari 1 kondisi yaitu kondisi1 dan kondisi2. Pada prakteknya kondisi dapat berjumlah tiga, empat dan seterusnya. Cara kerja percabangan dengan sintaks di atas seperti aliran dari atas ke bawah selama kondisi belum terpenuhi. Namun jika kondisi sudah terpenuhi maka aliran akan berhenti. Artinya jika kondisi1 sudah terpenuhi maka tidak perlu lagi melakukan pemeriksaan ke bawah. Namun jika kondisi1 belum terpenuhi maka dilakukan pemeriksaan kondisi2. Dan jika kondisi2 tidak memenuhi juga maka yang akan dijalankan adalah statement3.

```
percabangan_3.html
<!DOCTYPE html>
<html>
    <head>
        <title>Percabangan</title>
        <script>
            var time = 9;
            var greeting = "";
            if(time < 10) {
                greeting = "good morning";
            } else if(time < 20) {
                greeting = "good day";
            }
            else {
                greeting = "good evening";
            }

            alert(greeting);
        </script>
    </head>

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

Pada contoh di atas nilai time adalah 9. Keluaran dari kode di atas adalah pesan “good morning”. Karena yang memenuhi syarat pertama kali adalah kondisi yang bertama dimana “time < 10”. Walaupun pada kondisi kedua juga memenuhi syarat yaitu “time < 20”.

### **switch-case**

Selain menggunakan sintaks di atas, percabangan juga dapat dilakukan dengan kata kunci switch-case dengan sintaks berikut ini.

```
switch(ekspresi) {  
    case nilai1:  
        // statement1  
        break;  
    case nilai2:  
        // statement2  
        break;  
    default:  
        // statement3  
}
```

Ekspresi berisi suatu nilai yang akan dibandingkan dengan nilai yang berada setelah kata kunci case. Misal nilai hasil ekspresi sama dengan nilai1 maka statement1 akan dijalankan. Jika tidak maka nilai ekspresi akan dibandingkan dengan nilai2 dan jika sesuai maka statement2 dijalankan. Jika nilai ekspresi tidak ada yang sesuai dengan nilai-nilai setelah kata kunci case maka akan dijalankan statement dari blok setelah kata kunci default.

Berikut adalah contoh penggunaan percabangan dengan switch-case.

```
percabangan_4.html  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Percabangan</title>  
        <script>  
            switch (new Date().getDay()) {  
                case 0:  
                    day = "Sunday";  
                    break;  
                case 1:  
                    day = "Monday";  
                    break;  
                case 2:  
                    day = "Tuesday";  
                    break;  
                case 3:  
                    day = "Wednesday";  
                    break;  
                case 4:  
                    day = "Thursday";  
                    break;  
                case 5:  
                    day = "Friday";  
                    break;  
                case 6:  
                    day = "Saturday";  
            }
```

```
        }
        alert(day);
    </script>
</head>

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

## Perulangan

---

Kata kunci yang digunakan untuk melakukan pengulangan adalah for dan while.

*for*

Berikut ini adalah sintaks pengulangan dengan for.

```
for (statement1; statement2; statement3) {
    // code block to be executed
}
```

Statement1 akan dieksekusi sekali saja. umumnya statement1 berfungsi untuk inisiasi nilai. Statement2 berisi kondisi, jika kondisi masih terpenuhi maka kode di dalam tanda kurung akan dikerjakan. Statement3 dieksekusi setiap kali setelah kode di dalam tanda kurung dikerjakan.

Berikut contoh program yang menggunakan pengulangan dengan for.

```
pengulangan_for_1.html
<!DOCTYPE html>
<html>
    <head>
        <title>Pengulangan</title>
        <script>
            for (var i=1; i<=3; i++) {
                window.open("http://www.google.com");
            }
        </script>
    </head>

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

Pada kode di atas dapat dilihat statement pertama berisi inisialisasi nilai variable i=1. Kemudian pengulangan akan dilakukan jika nilai i kurang atau sama dengan 3. Pada statement ketiga dapat dilihat bagaimana nilai variable i

diubah nilainya. Statement ini dijalankan setelah kode pada tanda kurung selesai dijalankan.

Contoh yang lain adalah sebagai berikut ini.

```
pengulangan_for_2.html
<!DOCTYPE html>
<html>
  <head>
    <title>Pengulangan</title>
    <script>
      for (url = "http://www.google.com", i=1; i<=3; i++) {
        window.open(url);
      }
    </script>
  </head>

  <body>

  </body>
</html>
```

Pada contoh di atas dapat dilihat bahwa statement1 dapat berisi beberapa statement. Ini adalah salah satu perbedaan sintaks JavaScript yang berbeda dengan bahasa pemrograman lain. Selain itu statement1, statement2 dan statement3 pada JavaScript adalah bersifat pilihan, artinya bisa digunakan atau tidak.

Berikut beberapa contoh kode yang membuktikan pertanyaan di atas. kode di bawah ini adalah contoh ketika statement pertama tidak diisi.

```
pesan = "hello";
i=1;
for (; i<=3; i++) {
  alert(pesan + " " + i);
}
```

Dan berikut ini adalah contoh ketika statement ketiga tidak diisi.

```
pesan = "hello";
i=1;
for (; i<=3; ) {
  alert(pesan + " " + i);
  i++;
}
```

*while*

## Method

---

Untuk membuat method di JavaScript digunakan dengan tata cara berikut ini.

```
fungsi_1.html
<!DOCTYPE html>
<html>
  <head>
    <title>Fungsi</title>
  </head>
  <script>
    function nama_fungsi1() {
      // isi fungsi
    }

    function nama_fungsi2(var1, var2){
      var3 = var1 + var2;
    }

    function nama_fungsi3(var1) {
      return var1;
    }
  </script>
  <body>
  </body>
</html>
```

Pada contoh di atas dapat dilihat tiga contoh penulisan method namun di JavaScript umumnya disebut fungsi (function). Fungsi pertama dan kedua adalah contoh fungsi yang tidak mengembalikan hasil. Sedangkan fungsi ketiga adalah contoh fungsi yang mengembalikan hasil.

## Event

---

Hampir seluruh elemen atau tag HTML memiliki event. Contoh event adalah sebuah elemen <input> dapat diklik dan selanjutnya akan memberikan reaksi. Contoh lain adalah ketika halaman web dimuat kemudian menjalankan suatu fungsi. Cara pemberian event pada suatu element adalah sama dengan memberikan atribut pada element tersebut.

Berikut ini adalah sebagian kecil contoh-contoh event yang dimiliki element HTML. Sebagai contoh element <body> dapat memiliki event onload. Artinya ketika halaman web diakses maka event ini akan dipanggil.

```
event_1.html
<!DOCTYPE html>
<html>
  <head>
    <title>Event</title>
  </head>
  <script>
    function pesan(){
      alert("pesan saat dokumen dimuat")
    }
  </script>
  <body onload="pesan()">
  </body>
</html>
```

Contoh berikutnya adalah pada element <input> diberikan event onclick yang berfungsi akan memberikan reaksi ketika elemen ini diklik.

```
event_2.html
<!DOCTYPE html>
<html>
  <head>
    <title>Event</title>
  </head>
  <script>
    function pesan(){
      alert("tombol diklik")
    }
  </script>
  <body>
    <input type="button" value="silakan diklik" onclick="pesan()">
  </body>
</html>
```

Contoh event yang lain adalah onmouseover yang membuat element akan memberikan reaksi ketika cursor diarahkan ke element. berikut adalah kode dari kasus ini.

```
event_3.html
<!DOCTYPE html>
<html>
  <head>
    <title>Event</title>
  </head>
  <script>
    function pesan(){
```

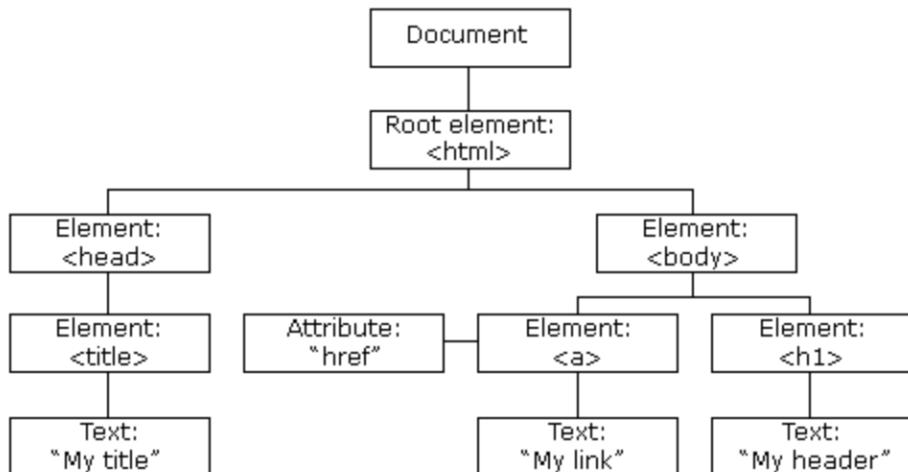
```

        alert("cursor sedang diarahkan ke tombol")
    }
</script>
<body>
    <input type="button" value="silakan diklik" onmouseover="pesan()">
</body>
</html>

```

## Document Object Model

Ketika halaman web telah dimuat, web browser membuat sebuah Document Object Model (DOM) dari halaman tersebut. HTML DOM membangun hirarki obyek-obyek pada halaman dengan struktur pohon seperti yang terlihat pada Gambar 83.



*Gambar 83. Pohon obyek HTML DOM.*

Dengan adanya DOM, JavaScript dapat mengakses obyek yang dimiliki suatu halaman. Akibatnya JavaScript memiliki kemampuan berikut ini:

- JavaScript dapat memodifikasi seluruh elemen HTML di dalam halaman web.
- JavaScript dapat memodifikasi seluruh atribut elemen HTML di dalam halaman web.
- JavaScript dapat memodifikasi seluruh style element HTML di dalam halaman web.
- JavaScript dapat menghapus elemen dan atribut HTML yang ada pada halaman web.

- JavaScript dapat menambahkan elemen dan atribut HTML baru pada halaman.
- JavaScript dapat bereaksi terhadap seluruh event HTML yang ada pada halaman.
- JavaScript dapat membuat event HTML baru pada halaman.

Untuk melakukan hal tersebut di atas maka diperlukan method-method yang disediakan oleh JavaScript. Pada sub bab ini diberikan method-method yang dapat digunakan untuk melakukan hal-hal tersebut di atas.

## Akses Elemen HTML

---

Untuk mengakses element HTML dapat dilakukan dengan beberapa cara, yaitu:

- Berdasarkan id element.
- Berdasarkan name element.
- Berdasarkan class element.

### Akses Elemen Unik

Untuk mengakses element berdasarkan id digunakan method getElementById seperti contoh berikut ini.

```
dom_1.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
  <script>
    function transfer(){
      var bil1 = document.getElementById("textbox_bill1");
      var bil2 = document.getElementById("textbox_bill2");

      bil2.value = bil1.value;
    }

    function reset() {
      var bil1 = document.getElementById("textbox_bill1");
      var bil2 = document.getElementById("textbox_bill2");

      bil1.value = "";
      bil2.value = "";
    }
  </script>
<body>
```

```

Bilangan 1: <input type="text" name="bil1" id="textbox_bil1">
<br>
Bilangan 2: <input type="text" name="bil2" id="textbox_bil2">
<br>
<input type="button" value="transfer data" onclick="transfer()">
<input type="button" value="reset" onclick="reset()">
</body>
</html>

```

Pada contoh di atas dapat dilihat dua element <input> dengan id masing-masing adalah textbox\_bil1 dan textbox\_bil2. Dengan mengetahui ini maka kedua element tersebut dapat diakses dengan method document.getElementById("id") dimana id adalah nilai atribut id dari kedua element.

```

var bil1 = document.getElementById("textbox_bil1");
var bil2 = document.getElementById("textbox_bil2");

```

Setelah obyek elemen telah didapatkan, maka dapat diakses nilai-nilai yang dimiliki oleh obyek tersebut. Pada kode di atas diakses nilai dari property value atau dapat dilihat potongan kode di bawah ini.

```

bil2.value = bil1.value;
. . .
bil1.value = "";
bil2.value = "";

```

```

dom 1.html
<!DOCTYPE html>
<html>
<head>
    <title>DOM</title>
</head>
<script>
    function transfer(){
        bil1 = document.getElementById("textbox_bil1").value;
        document.getElementById("textbox_bil2").value = bil1;
    }

    function reset() {
        document.getElementById("textbox_bil1").value = "";
        document.getElementById("textbox_bil2").value = "";
    }
</script>
<body>
    Bilangan 1: <input type="text" name="bil1" id="textbox_bil1">
    <br>
    Bilangan 2: <input type="text" name="bil2" id="textbox_bil2">

```

```
<br>
<input type="button" value="transfer data" onclick="transfer()">
<input type="button" value="reset" onclick="reset()">
</body>
</html>
```

```
bil1 = document.getElementById("textbox_bil1").value
```

Sedangkan untuk memberikan nilai pada suatu element yang telah diakses dapat dilakukan seperti potongan kode berikut.

```
document.getElementById("textbox_bil2").value = bil1
...
document.getElementById("textbox_bil1").value = "";
document.getElementById("textbox_bil2").value = "";
```

### Akses Koleksi Elemen

Seperti diketahui bahwa atribut id memberikan identitas unik pada suatu elemen. Sehingga method getElementById dipastikan hanya akan menemukan satu element saja. Namun berbeda jika ingin ditemukan elemen berdasarkan atribut name atau class, ada kemungkinan lebih dari satu elemen memiliki nilai atribut name atau class yang sama. Hal ini membuat method getElementsByName dan getElementsByTagName akan mengembalikan sekumpulan elemen.

Sebagai contoh sebagai berikut.

```
dom 3.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
  <script>
    function transfer(){
      bil1 = document.getElementsByName("bil1")[0].value;
      document.getElementsByName("bil2")[0].value = bil1;
    }

    function reset() {
      document.getElementsByName("bil1")[0].value = "";
      document.getElementsByName("bil2")[0].value = "";
    }
  </script>
<body>
```

```

Bilangan 1: <input type="text" name="bil1" id="textbox_bil1">
<br>
Bilangan 2: <input type="text" name="bil2" id="textbox_bil2">
<br>
<input type="button" value="transfer data" onclick="transfer()">
<input type="button" value="reset" onclick="reset()">
</body>
</html>

```

Kode di atas menggunakan method getElementsByName. Kode ini memiliki fungsi yang sama seperti kode sebelumnya yang menggunakan method getElementById. Namun dapat dilihat perbedaan terdapat index [0] seperti yang terlihat pada potongan kode berikut.

```

bil1 = document.getElementsByName("bil1")[0].value;
document.getElementsByName("bil2")[0].value = bil1;

```

Jika tidak digunakan penentuan index seperti contoh di atas maka kode di atas tidak akan berfungsi sama dengan kode sebelumnya.

Dari paparan di atas dapat disimpulkan bahwa:

- getElementById cocok digunakan untuk mendapatkan elemen yang unik.
- Sedangkan jika ingin mendapatkan sekumpulan elemen digunakan method getElementsByName dan getElementsByTagName.

Berikut ini contoh penggunaan method getElementsByTagName.

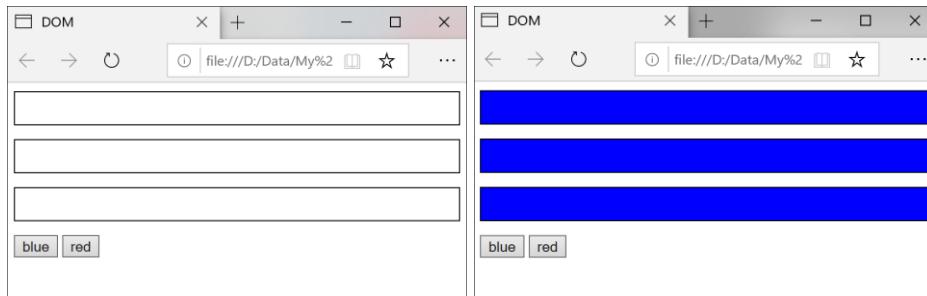
```

dom_4.html
<!DOCTYPE html>
<html>
<head>
    <title>DOM</title>
</head>
<style>
    .menu {
        height: 30px;
        border: solid 1px #000000;
        margin: 0 0 13px 0;
    }
</style>
<script>
    function set_color(color){
        col = document.getElementsByTagName("menu");
        count_col = col.length;
        for(i=0; i<count_col; i++){
            col[i].style.backgroundColor=color;
        }
    }
</script>
<body>

```

```
<div class="menu">&nbsp;</div>
<div class="menu">&nbsp;</div>
<div class="menu">&nbsp;</div>
<input type="button" value="blue" onclick="set_color('blue')">
<input type="button" value="red" onclick="set_color('red')">
</body>
</html>
```

Hasilnya dapat dilihat pada gambar berikut ini.



*Gambar 84. Mengubah warna berdasarkan warna class.*

Pada contoh di atas untuk mendapatkan seluruh dengan potongan kode berikut ini.

```
col = document.getElementsByClassName("menu");
```

Sedangkan kode berikut untuk mengetahui ada berapa jumlah elemen yang menggunakan atribut class="menu".

```
count_col = col.length;
```

Selanjutnya adalah mengubah elemen dengan menggunakan index. Karena terdapat banyak elemen yang harus diubah maka perlu dilakukan dengan cara pengulangan seperti potongan kode berikut.

```
for(i=0; i<count_col; i++){
    col[i].style.backgroundColor=color;
}
```

Pada kode di atas dapat dilihat JavaScript dapat mengakses atribut style dari elemen dan memberikan nilai.

Hasilnya dari kode di atas adalah ketika user mengklik tombol "blue" maka ketiga elemen <div> akan berwarna biru dan ketika diklik tombol "red" maka elemen-elemen tersebut menjadi merah.

## Modifikasi Elemen HTML

---

Pada sub bab di atas diberikan sedikit contoh untuk melakukan modifikasi elemen HTML. Untuk melakukan modifikasi dapat digunakan dua cara yaitu:

- Mengakses property.
- Menggunakan method setAttribute.

Berikut adalah beberapa sintaks modifikasi elemen dengan cara mengakses property.

```
element.innerHTML = new html content  
element.attribute = new value  
element.style.property = new style
```

Sedangkan jika menggunakan method setAttribute maka digunakan sintaks sebagai berikut.

```
element.setAttribute(attribute, value)
```

### Modifikasi Atribut Elemen

Berikut adalah contoh penggunaan cara element.getAttribute untuk memodifikasi elemen.

```
dom_5.html  
<!DOCTYPE html>  
<html>  
  <head>  
    <title>DOM</title>  
  </head>  
  <script>  
    function set_attribute_input(){  
      tombol1 = document.getElementById("tombol1");  
      tombol1.type = "button";  
      tombol1.value = "hello world";  
    }  
  </script>  
  <body onload="set_attribute_input()">  
    <input id="tombol1">  
  </body>  
</html>
```

Dari kode di atas dapat dilihat elemen `<input>` dengan `id="tombol1"`. Dari elemen tersebut belum diketahui apakah elemen tersebut menjadi tombol, textbox atau radiobutton.

Kemudian pada method set\_attribute\_input dapat dilihat bagaimana cara untuk memberikan atribut sekaligus nilainya.

```
tombol1.type = "button";
tombol1.value = "hello world";
```

Untuk memberikan antribut dan nilainya juga dapat menggunakan method seperti contoh berikut ini.

```
dom_6.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
  <script>
    function set_attribute_input(){
      tombol1 = document.getElementById("tombol1");
      tombol1.setAttribute("type", "button");
      tombol1.setAttribute("value", "hello world");
    }
  </script>
  <body onload="set_attribute_input()">
    <input id="tombol1">
  </body>
</html>
```

### Modifikasi Style Element

Untuk modifikasi elemen dengan cara element.style.property telah diberikan pada sub bab sebelumnya (sub bab Akses Elemen HTML) yaitu seperti berikut ini.

```
for(i=0; i<count_col; i++){
  col[i].style.backgroundColor=color;
}
```

### Modifikasi Content Element

Sedangkan fungsi dan cara penggunaan innerHTML dapat dilihat pada kode berikut ini.

```
dom_7.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
```

```

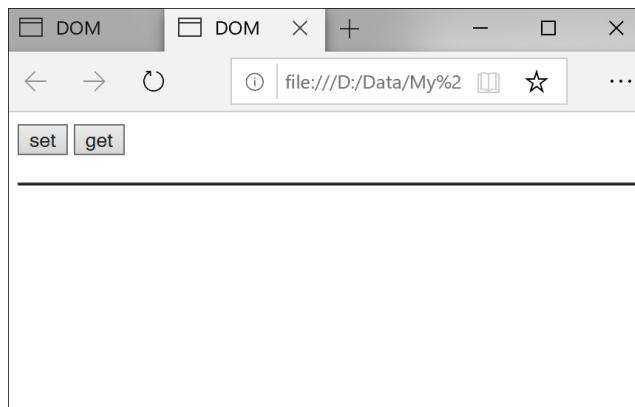
<script>
    function set(){
        table1 = document.getElementById("table1");
        table1.innerHTML = "<tr><td>&nbsp;</td></tr>";
    }

    function get(){
        table1 = document.getElementById("table1");
        alert(table1.innerHTML);
    }
</script>
<body>
    <input type="button" value="set" onclick="set()">
    <input type="button" value="get" onclick="get()">
    <br><br>
    <table width="100%" border="1" id="table1"></table>
</body>
</html>

```

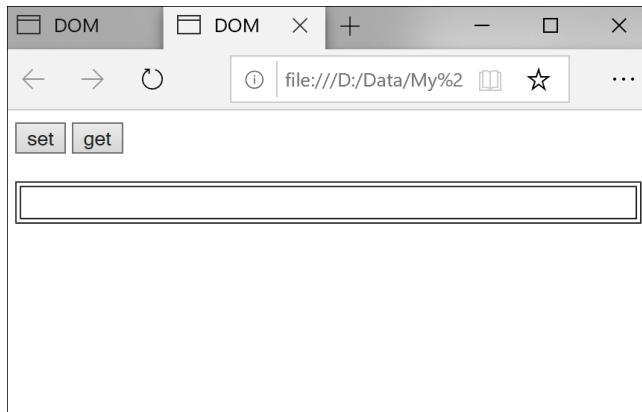
innerHTML dapat digunakan untuk menyisipkan kode HTML ke dalam suatu elemen. Selain itu juga dapat digunakan untuk mengetahui kode HTML pada suatu elemen.

Pada gambar Gambar 85 dapat dilihat tampilan awal saat halaman baru dimuat.



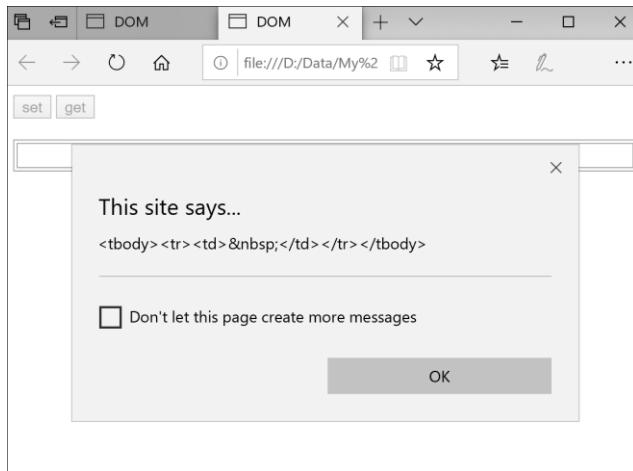
*Gambar 85. Tampilan awal.*

Selanjutnya setelah tombol set diklik maka akan dipanggil method set yang menggunakan innerHTML untuk menyisipkan kode HTML sehingga dapat dilihat hasilnya seperti pada Gambar 86.



*Gambar 86. Tampilan setelah tombol set diklik.*

Kemudian setelah tombol get diklik akan dapat dilihat hasilnya seperti pada Gambar 87.



*Gambar 87. Tampilan setelah tombol get diklik.*

## Menambah Elemen

---

Untuk menambah elemen dapat digunakan method-method berikut ini:

- `document.createElement`.
- `document.appendChild`.

Berikut adalah contoh kode penggunaannya.

```
dom_8.html
<!DOCTYPE html>
<html>
    <head>
        <title>DOM</title>
    </head>
    <script>
        function sendMessage(){
            var pesan = document.getElementById("textbox_pesan").value;
            var newDiv = document.createElement("div");
            var newContent = document.createTextNode(pesan);
            newDiv.appendChild(newContent);

            var currentDiv = document.getElementById("div1");
            document.body.insertBefore(newDiv, currentDiv);
        }
    </script>
<body>
    <div>
        <input type="text" name="pesan" id="textbox_pesan">
    </div>
    <input type="button" value="kirim" onclick="sendMessage()">
    <div id="div1">Hello World</div>
</body>
</html>
```

Pada kode HTML di atas dapat dilihat elemen <div> dengan id="div1". Kode JavaScript pada method sendMessage bertujuan untuk menambah elemen div baru sebelum div1. Elemen div baru berisi teks yang nilainya diisi oleh value pada kolom input pesan.

Untuk membuat elemen div baru digunakan cara berikut ini yaitu menggunakan method createElement. Sedangkan method.createTextNode bertujuan untuk menambahkan teks di dalam elemen tersebut.

```
var newDiv = document.createElement("div");
var newContent = document.createTextNode(pesan);
newDiv.appendChild(newContent);
```

Setelah elemen selesai dibuat selanjutnya elemen baru disisipkan dengan cara berikut ini. Pertama ditentukan dulu obyek yang telah ada di halaman. Kemudian sisipkan elemen baru (newDiv) ke obyek yang telah ada (currentDiv).

```
var currentDiv = document.getElementById("div1");
document.body.insertBefore(newDiv, currentDiv);
```

## Menghapus Element

---

Untuk menghapus elemen dapat menggunakan method removeChild. Sehingga perlu diketahui elemen induk dari elemen yang ingin dihapus.

```
dom 9.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
  <script>
    function removeItem(){
      var item = document.getElementById("item1");
      item.parentNode.removeChild(item);
    }
  </script>
  <body>
    <ul id="list1">
      <li id="item1">Item 1</li>
      <li id="item2">Item 2</li>
      <li id="item3">Item 3</li>
    </ul>
    <input type="button" value="remove" onclick="removeItem()">
  </body>
</html>
```

Cara di atas ini yang diketahui adalah id dari elemen yang ingin dihapus. Kemudian untuk mendapatkan menggunakan method removeChild perlu dilakukan dari elemen induk, sehingga digunakan cara seperti berikut ini.

```
item.parentNode.removeChild(item);
```

Selain cara di atas ada beberapa cara lain penggunaan method removeChild yang dapat dilihat pada link berikut ini [https://www.w3schools.com/jsref/met\\_node\\_removechild.asp](https://www.w3schools.com/jsref/met_node_removechild.asp).

## Akses Elemen dengan Selector

---

Selain dengan method yang dijelaskan pada dua sub bab di atas, dapat juga menggunakan method querySelector. Method ini mengembalikan elemen pertama yang sesuai dengan format selector CSS yang diberikan. Sedangkan jika ingin mendapatkan seluruh elemen yang cocok dengan selector yang diberikan dapat menggunakan method querySelectorAll.

Input untuk kedua method ini memiliki format yang sama dengan selector pada CSS. Sebagai contoh dapat dilihat pada contoh berikut ini.

```
dom_10.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
  <script>
    function bodyColor(){
      var body_obj = document.querySelector("body");
      body_obj.style.backgroundColor = "grey";
    }
  </script>
  <body onload="bodyColor()">

  </body>
</html>
```

Pada contoh di atas dapat dilihat input pada method querySelector adalah "body", sehingga variable body\_obj adalah elemen body.

Contoh berikutnya adalah mengakses elemen berdasarkan atribut class dan id.

```
dom_11.html
<!DOCTYPE html>
<html>
  <head>
    <title>DOM</title>
  </head>
  <script>
    function init(){
      var obj1 = document.querySelector(".p1");
      obj1.innerHTML = "hello world 1";

      var obj2 = document.querySelector("#p2");
      obj2.innerHTML = "hello world 2";
    }
  </script>
  <body onload="init()">
    <p class="p1"></p>
    <p id="p2"></p>
  </body>
</html>
```

Sedangkan contoh penggunaan method querySelectorAll dapat dilihat pada kode di bawah ini.

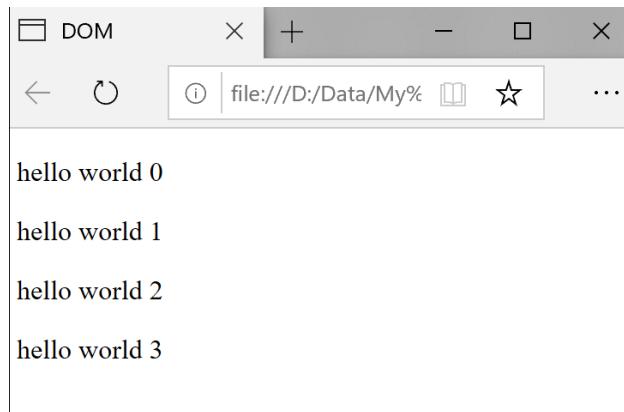
```
dom_12.html
<!DOCTYPE html>
<html>
  <head>
```

```

<title>DOM</title>
</head>
<script>
    function init(){
        var obj = document.querySelectorAll(".p1");
        var obj_count = obj.length;
        for(i=0; i<obj_count; i++){
            obj[i].innerHTML = "hello world " + i;
        }
    }
</script>
<body onload="init()">
    <p class="p1"></p>
    <p class="p1"></p>
    <p class="p1"></p>
    <p class="p1"></p>
    <p></p>
</body>
</html>

```

Pada kode di atas terdapat beberapa elemen <p> dengan atribut class="p1". Dengan method querySelectorAll seluruh elemen <p> dengan class="p1" saja yang akan dipilih dan dimodifikasi. Sehingga hasilnya dapat dilihat pada Gambar 88.



*Gambar 88. Memodifikasi banyak elemen dengan querySelectorAll.*

---

## Project

### Validasi Form

---

Pada bab sebelumnya telah dibuat form registrasi seperti pada gambar di bawah ini.

The screenshot shows a web browser window titled "Form Registrasi Event". The URL in the address bar is "file:///D:/Data/My%20Books/INDC%20-%20PHP%20-%20PHP%20&%20MySQL/source/bab\_4\_css/registrasi.html". The form itself is titled "EVENT REGISTRATION FORM" and contains the following fields:

- Name: Two adjacent input fields.
- Company: A single input field.
- Email: A single input field.
- Phone: Two adjacent input fields.
- Subject: A dropdown menu labeled "Choose Option".

Below these fields is a question: "Are you an existing customer?". It includes two radio buttons: "Yes" and "No". At the bottom of the form is a red "REGISTER" button.

*Gambar 89. Membuat validasi textbox.*

Proyek kali ini bertujuan untuk memvalidasi pengisian textbox dan dropdown list oleh user. Tulis kode program JavaScript untuk memeriksa apakah textbox dan dropdown list sudah diisi. Jika ada textbox atau dropdown list belum diisi maka tampilkan pesan dalam bentuk message box.

## ***PHP: Hypertext Preprocessor***

---

**Dasar-Dasar PHP**

---

**Form & File**

---

**Operasi Database**

---

**Otentikasi Pengguna**

---

**Object Oriented Programming**

---

**Reporting & Printing**

## ***Tentang Penulis***



**M Reza Faisal**, lahir di Banjarmasin, 20 Desember 1976. Pendidikannya, lulus SMA tahun 1995. Pernah Kuliah di Informatika Universitas Pasundan (Unpas) dan Fisika Institut Teknologi Bandung (ITB). Sejak kuliah telah aktif menulis sebagai di beberapa majalah komputer seperti Mikrodata, CHIP, dan InfoKomputer. Setelah menamatkan kuliah kemudian bekerja sebagai pengembang software di perusahaan swasta dan trainer training center di Bandung. Penulis juga aktif komunitas teknologi komputer dengan berbagi pengetahuan dalam bentuk blog atau ebook gratis yang fokus pada pembahasan teknologi Microsoft. Sehingga tahun 2008 mendapat penghargaan sebagai Microsoft Most Valuable Professional sampai sekarang. Kemudian pada akhir tahun 2008 kembali ke Banjarmasin dan menjadi dosen di Ilmu Komputer Universitas Lambung Mangkurat (ULM). Pada tahun 2010 melanjutkan studi di jenjang S2 di Informatika ITB konsentrasi Rekayasa Perangkat Lunak. Dan tahun 2015 melanjutkan ke pendidikan jenjang S3 di bidang Bioinformatika di Kanazawa University, Jepang.



**Friska Abadi** adalah dosen Program Studi S1 Ilmu Komputer Fakultas MIPA Universitas Lambung Mangkurat (ULM). Lahir di Karang Bintang tanggal 13 September 1988. Pendidikan S1 Ilmu Komputer di Universitas Lambung Mangkurat, Banjarmasin dan pendidikan S2 jurusan Magister Teknik Informatika di STMIK Amikom, Yogyakarta.