

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/339947152>

Pemrograman Web Dasar I: Belajar HTML 5

Book · August 2020

CITATIONS

0

READS

5,247

1 author:



M Reza Faisal

Universitas Lambung Mangkurat

62 PUBLICATIONS 144 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



IT Asset Management [View project](#)



DNA Sequence Classification [View project](#)



Belajar
HTML 5

Pemrograman Web Dasar I

Panduan pembuatan halaman web dengan HTML 5 & CSS.
Pemrograman dasar client-side dengan JavaScript serta
pemanfaatan Bootstrap 4 untuk membuat halaman web
yang dinamis dan responsif.

M Reza Faisal, Friska Abadi

Pemrograman Web Dasar I

Belajar HTML 5

M. Reza Faisal

Friska Abadi



Pemrograman Web Dasar I: Belajar HTML 5

© 2020 M Reza Faisal, Friska Abadi

343 halaman, 16,5 x 23 cm

Katalog dalam Terbitan (KDT)

Perpustakaan Nasional Republik Indonesia

ISBN: 978-602-0950-72-3

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: Agustus 2020

Penerbit

Scripta Cendekia

Banjarbaru, Kalimantan Selatan, Indonesia

E: scriptacendekia@gmail.com

Daftar Isi

| | |
|--|-----------|
| <i>Daftar Isi</i> | <i>ii</i> |
| <i>Daftar Gambar</i> | <i>vi</i> |
| 1 Pendahuluan | 1 |
| Tentang Buku | 1 |
| Internet | 1 |
| Sejarah Internet..... | 1 |
| Hypertext Transfer Protocol (HTTP) | 3 |
| Web Server | 3 |
| Web Browser..... | 4 |
| Aplikasi Web | 4 |
| Aplikasi Desktop & Aplikasi Web | 4 |
| Teknologi..... | 5 |
| Code Editor | 6 |
| Source Code..... | 6 |
| 2 Visual Studio Code..... | 7 |
| Installasi | 7 |
| Antarmuka..... | 8 |
| Tool Tambahan | 9 |
| Extension | 9 |
| Integrated Terminal | 10 |
| 3 HyperText Markup Language | 12 |
| Pendahuluan | 12 |
| Sejarah..... | 12 |
| Sintaks Elemen..... | 13 |
| Atribut..... | 16 |
| Aturan Penamaan File & Folder..... | 16 |
| Halaman Web Sederhana | 17 |

| | |
|---------------------------------------|------------|
| Tag HTML Dasar | 21 |
| Heading..... | 21 |
| Paragraf..... | 26 |
| Italic, Bold & Underline | 31 |
| List | 34 |
| Gambar..... | 39 |
| Hyperlink..... | 43 |
| Table | 50 |
| Form Input Data | 60 |
| Project..... | 83 |
| Tag HTML Lajut..... | 96 |
| Teks..... | 96 |
| Form | 101 |
| IFrame | 104 |
| Media..... | 109 |
| Grafik..... | 113 |
| 4 Cascading Style Sheets | 116 |
| Pendahuluan..... | 116 |
| Sintaks | 116 |
| Tata Cara Penggunaan..... | 117 |
| Dasar-Dasar CSS | 128 |
| Selector | 128 |
| Komentar | 141 |
| Unit | 142 |
| Warna | 143 |
| Box Model..... | 148 |
| Teks..... | 153 |
| Background Image | 156 |
| CSS Layout..... | 160 |

| | |
|------------------------------------|------------|
| Display | 160 |
| Lebar Elemen | 164 |
| Posisi | 166 |
| Float..... | 175 |
| Project..... | 180 |
| Form Registrasi..... | 180 |
| Layout Website | 188 |
| Video Player | 195 |
| 5 JavaScript..... | 211 |
| Pendahuluan | 211 |
| Aturan Penulisan | 211 |
| Internal..... | 211 |
| External..... | 213 |
| Dasar-Dasar Pemrograman | 214 |
| Komentar | 214 |
| Variable..... | 214 |
| Operasi Aritmatika..... | 215 |
| Percabangan..... | 215 |
| Perulangan | 220 |
| Method..... | 222 |
| Event | 224 |
| Document Object Model | 226 |
| Akses Elemen HTML | 227 |
| Modifikasi Elemen HTML | 231 |
| Menambah Elemen | 235 |
| Menghapus Element | 236 |
| Akses Elemen dengan Selector | 237 |
| Project..... | 239 |
| Validasi Form..... | 239 |

| | |
|--|------------|
| Animasi | 245 |
| Carousel | 247 |
| 6 Bootstrap 4 | 252 |
| Pendahuluan..... | 252 |
| Tantangan..... | 252 |
| Bootstrap 4..... | 254 |
| Installasi | 254 |
| Perangkat Pengembang..... | 257 |
| Penulisan Nama Class Pada Teks Di Buku Ini | 260 |
| Dasar-Dasar Bootstrap | 261 |
| Cara Penggunaan Bootstrap..... | 261 |
| Container | 263 |
| Grid..... | 267 |
| Layout Website..... | 275 |
| Navigasi | 279 |
| Bekerja dengan Table | 284 |
| Membuat Form..... | 288 |
| Tipe Layout Form | 288 |
| Validasi Form..... | 291 |
| Elemen Input..... | 297 |
| Project | 301 |
| Form Registrasi | 301 |
| Video Player | 314 |
| 7 Penutup | 326 |
| Tentang Penulis | 327 |

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. Extension berhasil diinstall. | 10 |
| Gambar 6. Integrated terminal. | 11 |
| Gambar 7. Halaman web sederhana. | 18 |
| Gambar 8. Halaman Web Sederhana - Latihan 1. | 19 |
| Gambar 9. Halaman web sederhana - Latihan 2. | 20 |
| Gambar 10. Halaman web sederhana - Latihan 3..... | 20 |
| Gambar 11. Style heading pada MS Word. | 21 |
| Gambar 12. Memformat teks - Heading. | 23 |
| Gambar 13. Memformat teks - Heading - Latihan 1..... | 24 |
| Gambar 14. Memformat teks - Heading - Latihan 2..... | 25 |
| Gambar 15. Memformat teks - Paragraf..... | 27 |
| Gambar 16. Memformat teks - Paragraf - Latihan 1..... | 28 |
| Gambar 17. Memformat teks - Paragraf - Latihan 2..... | 30 |
| Gambar 18. Memformat teks - Italic, bold & underline - Latihan 1..... | 32 |
| Gambar 19. Memformat teks - Italic, bold & underline - Latihan 2 | 33 |
| Gambar 20. Memformat teks - List..... | 34 |
| Gambar 21. List di dalam list..... | 37 |
| Gambar 22. Paragraf di dalam list..... | 38 |
| Gambar 23. Lokasi folder dan file HTML..... | 39 |
| Gambar 24. Menambahkan gambar pada halaman web..... | 41 |
| Gambar 25. Halaman web untuk menampilkan gallery foto. | 42 |
| Gambar 26. Membuat hyperlink untuk menghubungkan dua halaman web. 44 | |
| Gambar 27. Hyperlink ke alamat web. | 46 |

| | |
|--|-----|
| Gambar 28. Hyperlink di dalam halaman web..... | 48 |
| Gambar 29. Tabel daftar buku..... | 54 |
| Gambar 30. Form input untuk halaman Contact Us | 60 |
| Gambar 31. Form input data - Form - Contoh form sederhana..... | 61 |
| Gambar 32. Form input data - form - isi nilai kolom input..... | 62 |
| Gambar 33. Form input data - Textbox - Contoh 1..... | 65 |
| Gambar 34. Form input data - Tombol - Contoh 1..... | 66 |
| Gambar 35. Form input data - Tombol - Latihan 1..... | 67 |
| Gambar 36. Form input data - Tombol - Registrasi form..... | 69 |
| Gambar 37. Form input data - Label - Contoh label pada form..... | 70 |
| Gambar 38. Form input data - Radio button - Latihan 1. | 74 |
| Gambar 39. Form input data - Checkbox - Contoh. | 75 |
| Gambar 40. Form input data - Checkbox - Form login. | 76 |
| Gambar 41. Form input data - Textarea - Form isian email. | 78 |
| Gambar 42. Form input data - Drop-down list - Contoh..... | 79 |
| Gambar 43. Form input data - Drop-down list - Latihan 1. | 80 |
| Gambar 44. Form input data- Upload file - antarmuka upload file pada Microsoft Edge..... | 82 |
| Gambar 45. Antarmuka project Web Profile Pribadi. | 91 |
| Gambar 46. Sebuah komentar pada buku tamu. | 95 |
| Gambar 47. Antarmuka halaman guestbook.html. | 96 |
| Gambar 48. Perbedaan tag pre dan p. | 97 |
| Gambar 49. Perbedaan tag q dan blockquote..... | 99 |
| Gambar 50. Contoh penggunaan tag strong, em dan ins. | 99 |
| Gambar 51. Contoh penggunaan tag del, mark, sup & sub. | 100 |
| Gambar 52. Contoh penggunaan tag code, kbd, samp, & var. | 100 |
| Gambar 53. Antarmuka pada MS Edge & Google Chrome. | 103 |
| Gambar 54. Tombol berbentuk gambar. | 104 |
| Gambar 55. Contoh iframe 1..... | 105 |
| Gambar 56. Web detik.com pada iframe..... | 106 |

| | |
|--|-----|
| Gambar 57. Iframe dan hyperlink. | 107 |
| Gambar 58. Video youtube pada iframe..... | 108 |
| Gambar 59. Video player dengan kontrol. | 110 |
| Gambar 60. Menampilkan audio player. | 111 |
| Gambar 61. File pdf pada halaman web. | 112 |
| Gambar 62. Tag embed untuk menampilkan halaman web. | 113 |
| Gambar 63. Canvas..... | 114 |
| Gambar 64. Phaser - HTML 5 game framework. | 114 |
| Gambar 65. SVG pada halaman web..... | 115 |
| Gambar 66. Contoh CSS untuk style teks. | 117 |
| Gambar 67. Contoh Inline CSS..... | 118 |
| Gambar 68. Efek hirarki CSS. | 119 |
| Gambar 69. Border..... | 121 |
| Gambar 70. Efek style pada elemen HTML..... | 122 |
| Gambar 71. Sintaks pada Internal CSS..... | 123 |
| Gambar 72. Internal CSS - Contoh..... | 124 |
| Gambar 73. Internal CSS - Layout. | 126 |
| Gambar 74. Selector dan deklarasi CSS. | 129 |
| Gambar 75. Contoh simple selector..... | 130 |
| Gambar 76. Contoh combinator selector. | 132 |
| Gambar 77. Contoh pseudo-element selector. | 137 |
| Gambar 78. Contoh selector ::selection. | 138 |
| Gambar 79. Contoh attribute selector. | 139 |
| Gambar 80. Contoh selector [attribute*=value]. | 141 |
| Gambar 81. Contoh satuan mutlak dan relatif..... | 143 |
| Gambar 82. Contoh CSS warna pada teks dan latarbelakarn..... | 145 |
| Gambar 83. Contoh CSS warna pada border. | 146 |
| Gambar 84. Contoh warna dengan RGL & HSL..... | 147 |
| Gambar 85. Box model CSS. | 148 |

| | |
|--|-----|
| Gambar 86. Contoh border 1 | 149 |
| Gambar 87. Contoh border 2 | 150 |
| Gambar 88. Contoh property padding..... | 152 |
| Gambar 89. Contoh property margin..... | 153 |
| Gambar 90. Contoh property CSS untuk memformat teks. | 156 |
| Gambar 91. Background image menggunakan gambar ukuran besar. | 157 |
| Gambar 92. Lokasi penyimpanan gambar dan file default.css..... | 158 |
| Gambar 93. Membuat background image dengan gambar ukuran kecil..... | 159 |
| Gambar 94. Background image yang diulang..... | 159 |
| Gambar 95. Propert display = block. | 161 |
| Gambar 96. Property display = inline. | 162 |
| Gambar 97. Modifikasi property display..... | 163 |
| Gambar 98. Contoh property width pada window browser kecil. | 165 |
| Gambar 99. Contoh property width pada window browser dengan lebar > 500px..... | 166 |
| Gambar 100. Contoh nilai static dan relative pada property position..... | 168 |
| Gambar 101. Contoh position dengan nilai fixed dan absolute..... | 170 |
| Gambar 102. Contoh penggunaan property z-index..... | 172 |
| Gambar 103. Contoh position dengan nilai sticky..... | 174 |
| Gambar 104. Contoh property float dengan nilai right. | 176 |
| Gambar 105. Contoh property float dengan nilai left. | 177 |
| Gambar 106. Contoh efek property float pada elemen setelahnya. | 179 |
| Gambar 107. Contoh penggunaan dan efek property clear. | 179 |
| Gambar 108. Form registrasi event..... | 180 |
| Gambar 109. Window form input..... | 182 |
| Gambar 110. Elemen input data tanpa CSS..... | 184 |
| Gambar 111. Form input data yang telah diberikan CSS. | 188 |
| Gambar 112. Contoh layout website sederhana 1. | 189 |
| Gambar 113. Layout website - tahap 1..... | 191 |
| Gambar 114. Elemen list di dalam elemen <div> menu..... | 191 |

| | |
|---|-----|
| Gambar 115. Elemen list yang telah diberi style..... | 192 |
| Gambar 116. Contoh layout website sederhana 2 | 193 |
| Gambar 117. Website video player..... | 195 |
| Gambar 118. Video player - latarbelakang | 196 |
| Gambar 119. Video player - menu dan kolom pencarian..... | 198 |
| Gambar 120. Video player - video player | 199 |
| Gambar 121. Video player - deskripsi film..... | 201 |
| Gambar 122. Video player - daftar film. | 203 |
| Gambar 123. Message box keluaran JavaScript. | 212 |
| Gambar 124. Message box dari fungsi JavaScript. | 213 |
| Gambar 125. Pohon obyek HTML DOM. | 226 |
| Gambar 126. Mengubah warna berdasarkan warna class..... | 230 |
| Gambar 127. Tampilan awal..... | 234 |
| Gambar 128. Tampilan setelah tombol set diklik. | 234 |
| Gambar 129. Tampilan setelah tombol get diklik..... | 235 |
| Gambar 130. Memodifikasi banyak elemen dengan querySelectorAll. | 239 |
| Gambar 131. Membuat validasi textbox. | 240 |
| Gambar 132. Pesan validasi. | 245 |
| Gambar 133. Animasi dengan JavaScript. | 247 |
| Gambar 134. Animasi carousel. | 248 |
| Gambar 135. Jenis perangkat..... | 253 |
| Gambar 136. favicon.ico..... | 256 |
| Gambar 137. Struktur file dan folder pada project Bootstrap..... | 256 |
| Gambar 138. Menu untuk mengakses alat Pengembang pada Google Chrome. | 258 |
| Gambar 139. Alat Pengembang pada Google Chrome | 258 |
| Gambar 140. Menu untuk mengakses Developer tools. | 259 |
| Gambar 141. Developer Tools pada Microsoft Edge. | 260 |
| Gambar 142. Penggunaan class .container. | 262 |
| Gambar 143. Penggunaan class .p-3, .my-3, .bg-primary, .text-white. | 263 |

| | |
|--|-----|
| Gambar 144. Perbandingan .container dan .container-fluid..... | 263 |
| Gambar 145. Ukuran lembar .container berdasarkan perangkat | 264 |
| Gambar 146. Penggunaan .container..... | 265 |
| Gambar 147. Contoh penggunaan .container-fluid..... | 267 |
| Gambar 148. Layout menggunakan grid..... | 267 |
| Gambar 149. Penggunaan .col..... | 270 |
| Gambar 150. Contoh penggunaan .col-..... | 272 |
| Gambar 151. Antarmuka elemen .col-sm pada layar lebar..... | 274 |
| Gambar 152. Antarmuka elemen .col-sm pada perangkat layar sempit..... | 274 |
| Gambar 153. Antarmuka elemen dengan .col-sm-3 dan .col-sm-9 pada perangkat layar lebar..... | 275 |
| Gambar 154. Antarmuka elemen dengan .col-sm-3 dan .col-sm-9 pada perangkat layar kecil..... | 275 |
| Gambar 155. Antarmuka layout pada perangkat dengan ukuran layar besar..... | 278 |
| Gambar 156. Antarmuka layout pada perangkat dengan ukuran layar kecil. | 279 |
| Gambar 157. Antarmuka navigasi pada perangkat dengan ukuran layar besar..... | 281 |
| Gambar 158. Antarmuka navigasi pada perangkat dengan ukuran kecil. | 282 |
| Gambar 159. Navigasi posisi tetap di atas..... | 283 |
| Gambar 160. Table dengan style .table..... | 286 |
| Gambar 161. Table dengan class .table-striped..... | 287 |
| Gambar 162. Table dengan class .table-bordered. | 287 |
| Gambar 163. Tipe layout form – stacked form..... | 289 |
| Gambar 164. Tipe layout form – inline form. | 290 |
| Gambar 165. Pesan kesalahan pada validasi dengan class .was-validated.... | 293 |
| Gambar 166. Pesan sukses pada validasi dengan class .was-validated..... | 293 |
| Gambar 167. Halaman form dengan validasi menggunakan class .need-validation saat dimuat..... | 296 |
| Gambar 168. Tampilan pesan kesalahan dan sukses. | 296 |
| Gambar 169. Perbedaan ukuran elemen input..... | 299 |

| | |
|---|-----|
| Gambar 170. Penggunaan class .form-control | 300 |
| Gambar 171. Antarmuka form registrasi dengan Bootstrap 4..... | 301 |
| Gambar 172. Antarmuka form registrasi pada layar kecil. | 302 |
| Gambar 173. Form Registrasi - Langkah 1..... | 304 |
| Gambar 174. Form Registrasi - Langkah 2..... | 305 |
| Gambar 175. Form Registrasi - Langkah 3..... | 306 |
| Gambar 176. Form Registrasi - Langkah 4..... | 307 |
| Gambar 177. Form Registrasi - Langkah 5..... | 307 |
| Gambar 178. Form Registrasi - Langkah 6..... | 309 |
| Gambar 179. Form Registrasi dengan validasi. | 310 |
| Gambar 180. Video player dengan Bootstrap 4. | 314 |
| Gambar 181. Video player Bootstrap 4 pada layar kecil. | 315 |
| Gambar 182. Video Player - Navigasi bar..... | 317 |
| Gambar 183. Video player - Elemen video. | 319 |
| Gambar 184. Video player - Poster rekomendasi film. | 320 |

1

Pendahuluan

Tentang Buku

Pembahasan Pemrograman Web Dasar ini terdiri atas dua buku. Pemrograman Web Dasar I ini adalah buku pertama yang membahas dasar-dasar HTML 5, CSS dan JavaScript untuk membuat halaman web. Dan sebagai tambahan adalah panduan penggunaan framework Bootstrap 4 untuk membuat halaman web responsif.

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 seperti pada Gambar 1.



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 desktop tidak dapat memaksa untuk melakukan upgrade.
7. Aplikasi desktop bergantung dengan sistem operasi. Aplikasi desktop yang dibangun pada sistem operasi Windows tidak bisa digunakan pada sistem operasi Linux atau MacOS.

Aplikasi 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. Bootstrap 4 sebagai framework untuk membuat halaman web yang responsif.

Code Editor

Perangkat code editor yang digunakan 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:

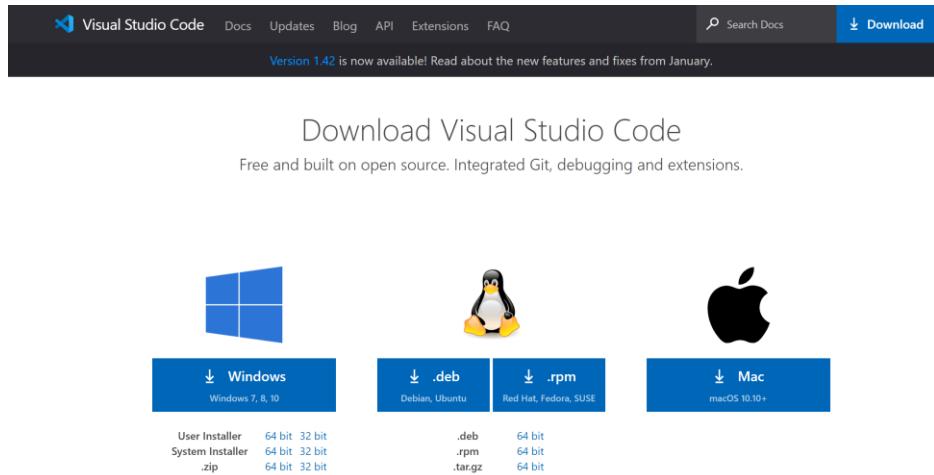
<https://github.com/rezafaisal/PemrogramanWebDasar>

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

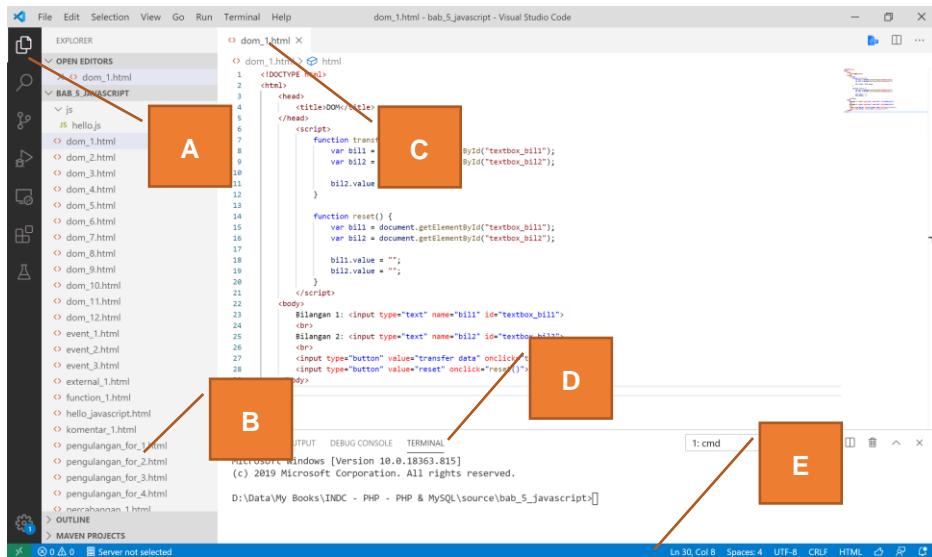
Installer Visual Studio Code dapat diunduh pada link berikut ini <https://code.visualstudio.com/download> dengan tampilan halaman seperti pada Gambar 2. Pilih installer yang sesuai dengan sistem operasi yang digunakan. Saat buku ini diketik versi yang ada adalah 1.45.1.



Gambar 2. Halaman download Visual Studio Code.

Antarmuka

Pada Gambar 3 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:

- View Bar, berisi tombol-tombol yaitu:
 - Explorer yang berfungsi untuk melihat daftar folder dan file-file didalamnya.
 - Search untuk melakukan pencarian.
 - Source Control untuk melihat daftar folder yang disimpan pada source control repository.
 - Extension untuk melihat daftar extension yang tersedia pada repository.
 - Manage untuk mendampulkan context menu seperti Command Pallate, Settings dan lain-lain.
 - Selain itu pada Gambar 3 fitur lain seperti Run, Remote Explorer, Test yang ada jika extension terkait diinstall.
- Side Bar, akan menampilkan view sesuai dengan tombol pada View Bar yang dipilih.
- 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.
- 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.

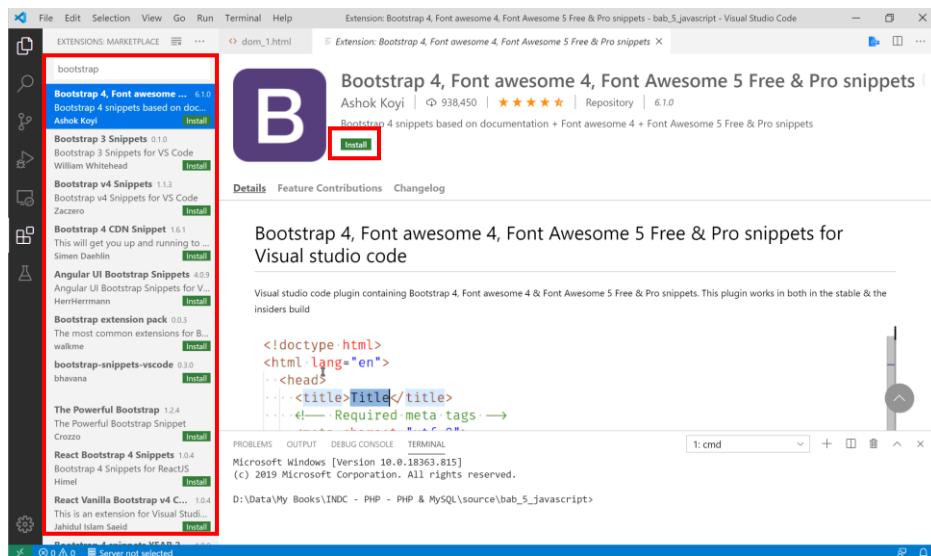
Tool Tambahan

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.

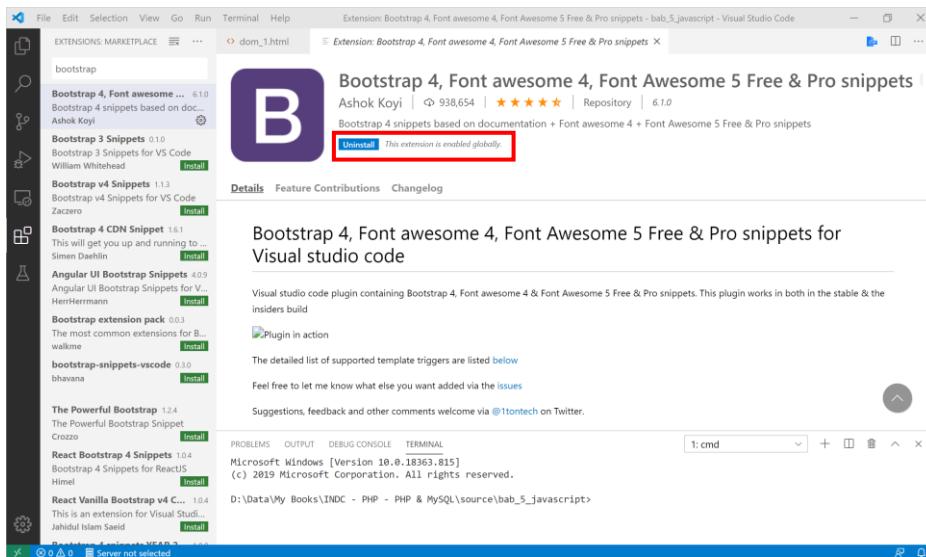
Extension

Untuk mengakses daftar extension yang tersedia dapat mengklik tombol Extensions pada View Bar. Untuk melihat daftar extension diperlukan koneksi internet. Daftar extension dapat dilihat pada kotak merah pada area sidebar di Gambar 4.



Gambar 4. Daftar extension.

User dapat mencari extension yang diinginkan dengan mengetik kata kunci pada kolom yang disediakan. Misalnya diketik kata kunci “bootstrap” maka dapat dilihat hasilnya seperti pada Gambar 4. Klik extension yang sesuai dengan kemudian hasilnya dapat dilihat pada gambar di atas. Kemudian klik tombol install maka secara otomatis file extension akan diunduh dan diinstall. Status berhasilnya proses installasi adalah tombol hijau dengan teks “Install” akan menjadi tombol biru dengan teks “Uninstall” seperti yang dapat dilihat pada Gambar 5. Tombol Uninstall dapat digunakan untuk menghapus extension jika tidak diperlukan.



Gambar 5. Extension berhasil diinstall.

Selain itu secara otomatis Visual Studio Code akan memeriksa versi terbaru dari extension yang telah diinstall. Jika ada versi terbaru dari extension maka pada detail extension akan ada tombol untuk update.

Integrated Terminal

Cara kedua adalah dengan Terminal. Untuk menampilkan terminal dapat dilakukan dengan cara memilih menu View > Integrated Terminal seperti pada kotak merah pada Gambar 6.

The screenshot shows the Visual Studio Code interface. On the left is the Explorer sidebar with a tree view of files and folders. In the center is the code editor with a file named 'dom_1.html' open. The code contains JavaScript for a simple application involving two text boxes and two buttons. At the bottom of the screen is the integrated terminal window, which is highlighted with a red rectangle. It displays the command prompt 'D:\Data\My Books\INDC - PHP - PHP & MySQL\source\bab_5_javascript:', followed by the output of the 'npm -v' command: '4.1.1'. The terminal also shows the Microsoft Windows version information.

```
dom_1.html > dom_1.html
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <title>DOM</title>
5   </head>
6   <script>
7     function transfer() {
8       var bill1 = document.getElementById("textbox_bill1");
9       var bill2 = document.getElementById("textbox_bill2");
10
11       bill2.value = bill1.value;
12     }
13
14     function reset() {
15       var bill1 = document.getElementById("textbox_bill1");
16       var bill2 = document.getElementById("textbox_bill2");
17
18       bill1.value = "";
19       bill2.value = "";
20     }
21   </script>
22 <body>
23   Bilangan 1: <input type="text" name="bill1" id="textbox_bill1">
24   <br>
25   Bilangan 2: <input type="text" name="bill2" id="textbox_bill2">
26   <br>
27   <input type="button" value="transfer" data onclick="transfer()">
28   <input type="button" value="reset" onclick="reset()">

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Microsoft Windows [Version 10.0.18363.815]
© 2019 Microsoft Corporation. All rights reserved.
D:\Data\My Books\INDC - PHP - PHP & MySQL\source\bab_5_javascript: 4.1.1

Gambar 6. 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 melihat versi dari npm maka dapat menggunakan perintah berikut ini.

```
npm -v
```

Maka hasilnya akan dapat dilihat pada kotak merah pada Gambar 6.

HyperText Markup Language

HyperText Markup Language (HTML) adalah bahasa yang digunakan untuk membuat antarmuka halaman web. Aplikasi web mengabungkan antarmuka yang dibuat dengan HTML dengan bahasa pemrograman yang berfungsi sebagai logika dan mengelola data. Oleh karena itu pengetahuan tentang HTML sangat penting dalam pembangunan aplikasi 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 Elemen

HTML elemen atau elemen terdiri atas tag pembuka, konten atau isi di dalam tag dan tag penutup. 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

```
<br>
```

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 penulisan dimana tag ditulis dengan menggunakan penutup (contoh:
) 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
. Sedangkan aturan pada HTML 5 lebih longgar sehingga tag tipe void elemen dapat ditulis tanpa penutup. Sehingga cukup ditulis sebagai

berikut
 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 5 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>
```

Kode ini menghasilkan tampilan seperti pada Gambar 7. Penjelasan kode di atas adalah sebagai berikut.

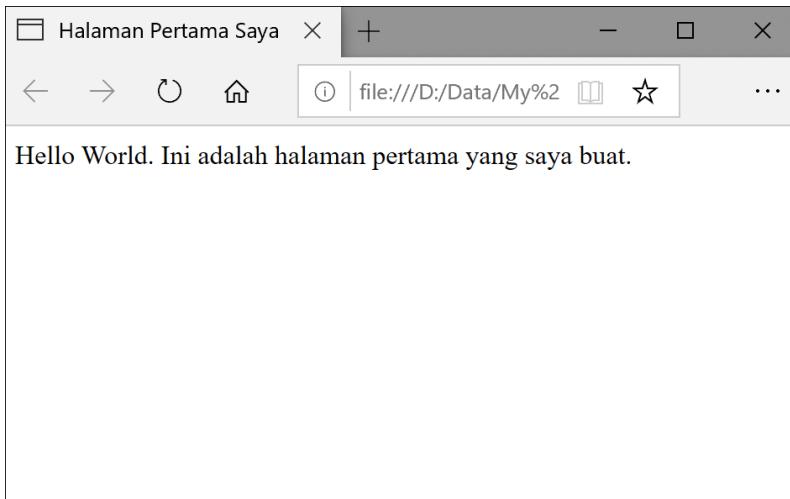
Baris pertama adalah tag spesial yang berfungsi untuk memberikan informasi kepada web browser bahwa tipe dokumen ini adalah HTML.

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.



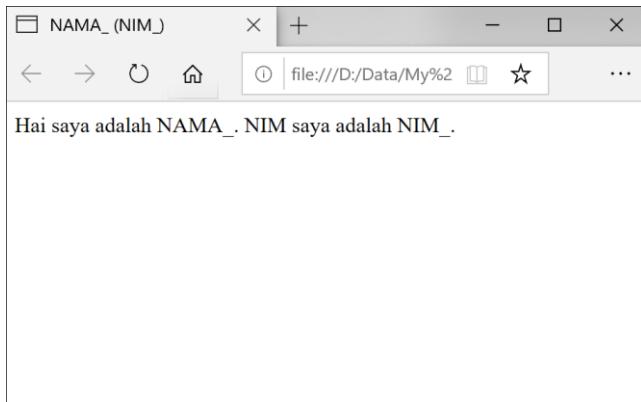
Gambar 7. Halaman web sederhana.

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 pada Gambar 8. ganti "NAMA_" dan "NIM_" dengan nama dan NIM pembaca inginkan.



Gambar 8. 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

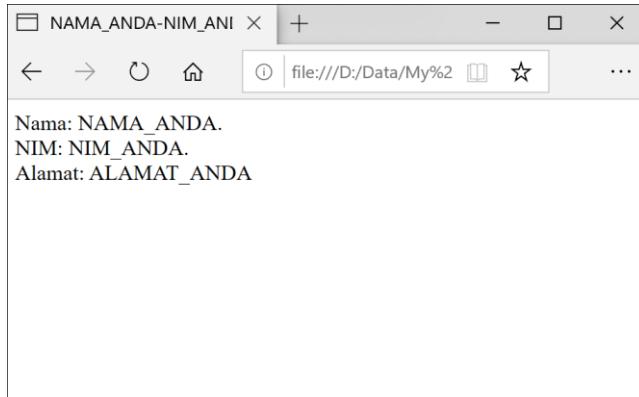
Ketik kode HTML berikut dengan dan simpan ke dalam file dengan nama web_sederhana_latihan_2.html.

```
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>
```

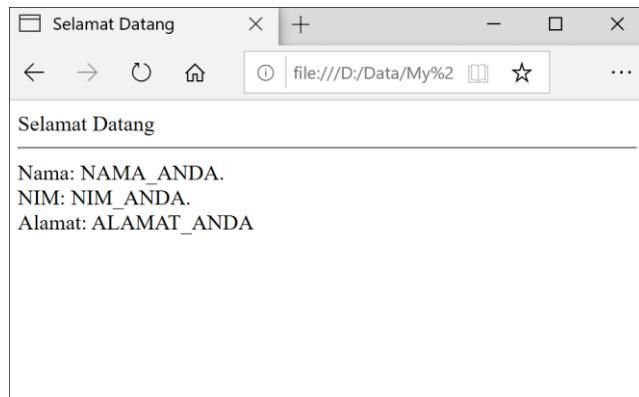
Hasilnya dapat dilihat pada Gambar 9. Pada kode di atas dapat dilihat tag baru yaitu
 yang berfungsi seperti menekan tombol enter. Tag
 membuat teks setelahnya ditulis pada baris baru.



Gambar 9. 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 10.



Gambar 10. 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>
```

Tag HTML Dasar

Pada sub bab ini diberikan tag-tag HTML yang minimal harus diketahui oleh pembaca yang ingin membuat aplikasi web. Tag-tag ini umum dan sering digunakan pada sebuah aplikasi web.

Heading

Pada aplikasi word processor seperti MS Word (Gambar 11), 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 11. 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>

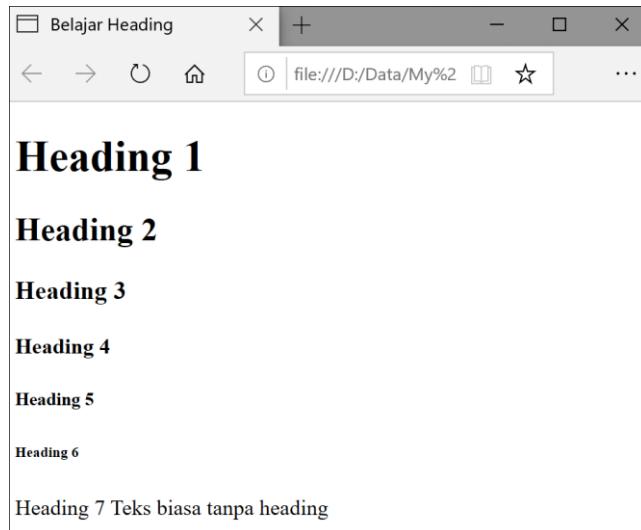
<body>
    <h1>Heading 1</h1>
    <h2>Heading 2</h2>
    <h3>Heading 3</h3>
    <h4>Heading 4</h4>
    <h5>Heading 5</h5>
    <h6>Heading 6</h6>
    <h7>Heading 7</h7>
    Teks biasa tanpa heading
</body>
</html>
```

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 12.

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 12, dimana teks “Heading 7” yang menggunakan tag `<h7>` dan teks “Teks biasa tanpa heading” mempunyai ukuran teks yang sama.



Gambar 12. Memformat teks - Heading.

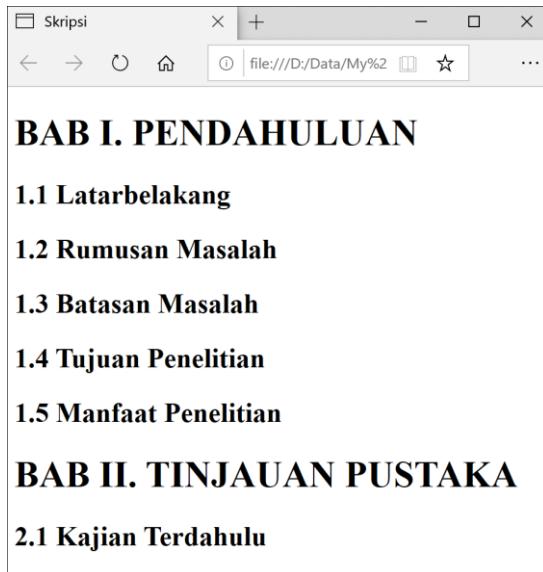
Latihan 1

Buat teks dengan hirarki memiliki tampilan seperti pada Gambar 13. Hirarki hanya terdiri atas 2 tingkatan, sehingga cukup menggunakan tag <h1> dan <h2> saja.

Berikut adalah kode HTML untuk membuat tampilan seperti Gambar 13.

```
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>
```

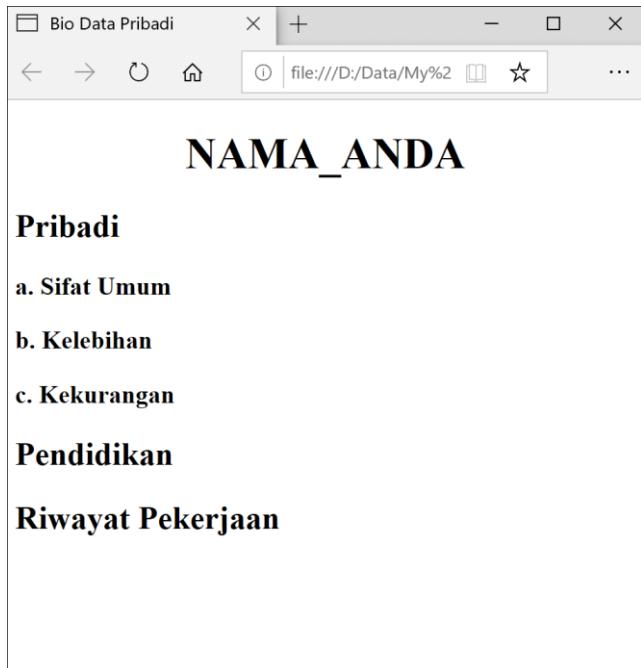


Gambar 13. Memformat teks - Heading - Latihan 1.

Latihan 2

Pada Gambar 14 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 14. 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?

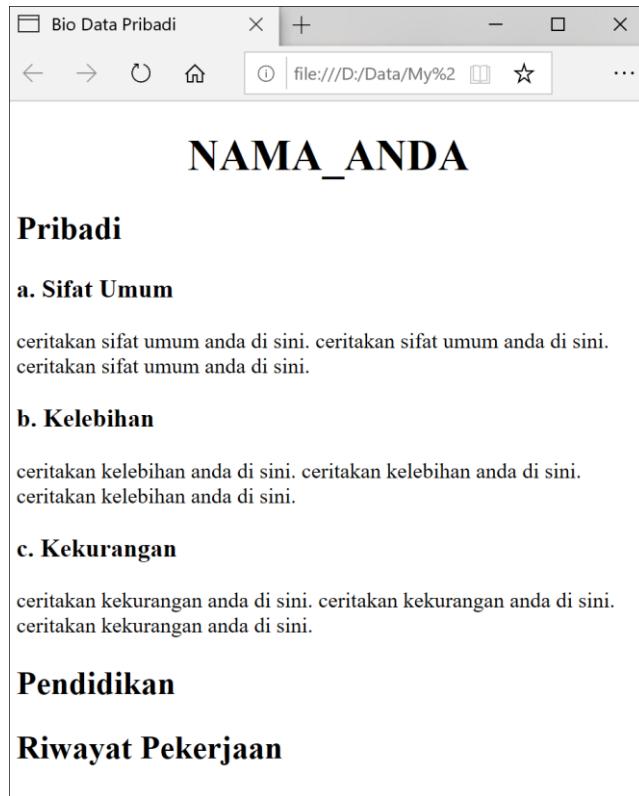
Paragraf

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 15. Memformat teks - Paragraf.

Berikut ini adalah kode HTML dari gambar di atas.

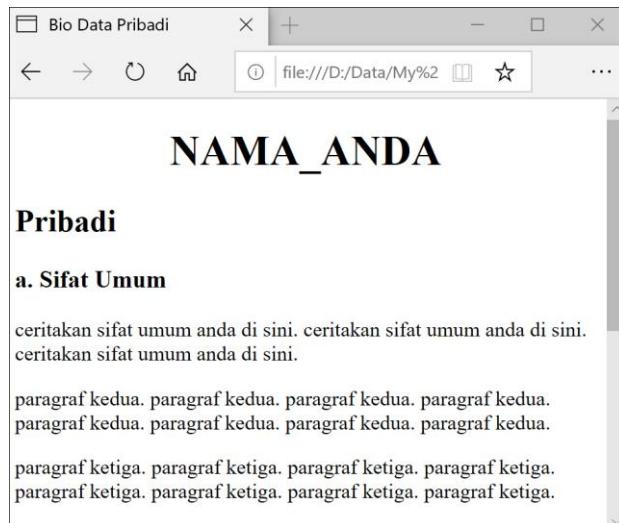
```
paragraf_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <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 16. 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 16. Memformat teks - Paragraf - Latihan 1.

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

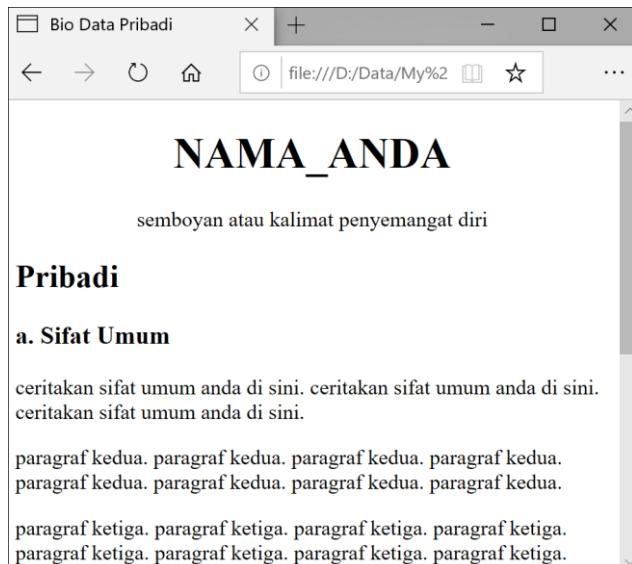
```
paragraf_02.html
```

```
<!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 17.



Gambar 17. 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 ``. 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.

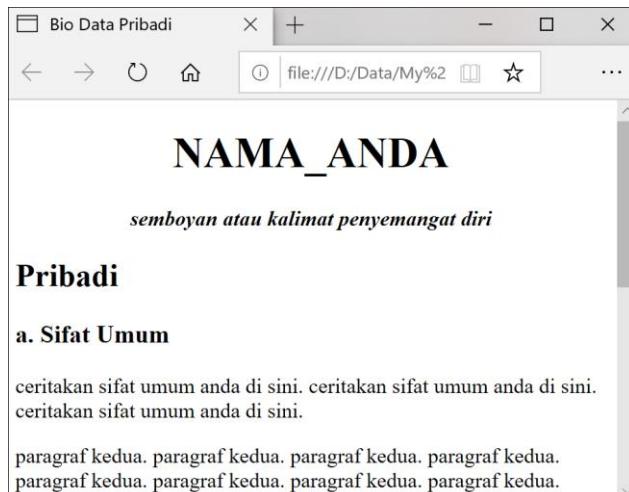
```

<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 18. Kemudian simpan hasilnya pada file baru dengan nama ibu_1.html.



Gambar 18. Memformat teks - *Italic*, **bold** & underline - Latihan 1.

Sebagian kode HTML dari antarmuka di atas adalah sebagai berikut.

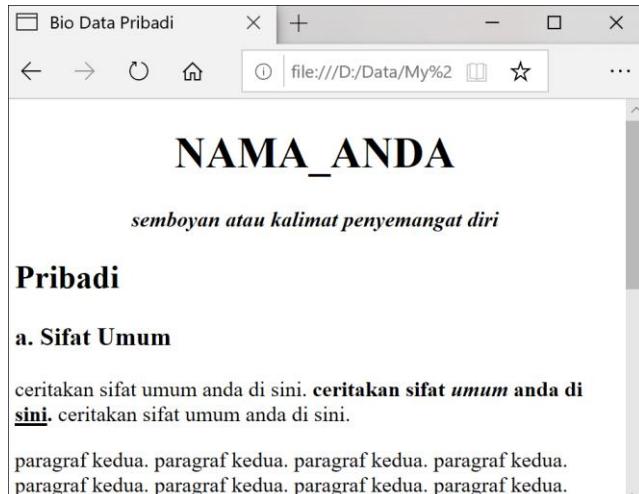
```
ibu_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Bio Data Pribadi</title>
  </head>

  <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 19.



Gambar 19. Memformat teks - *Italic*, **bold** & underline - Latihan 2.

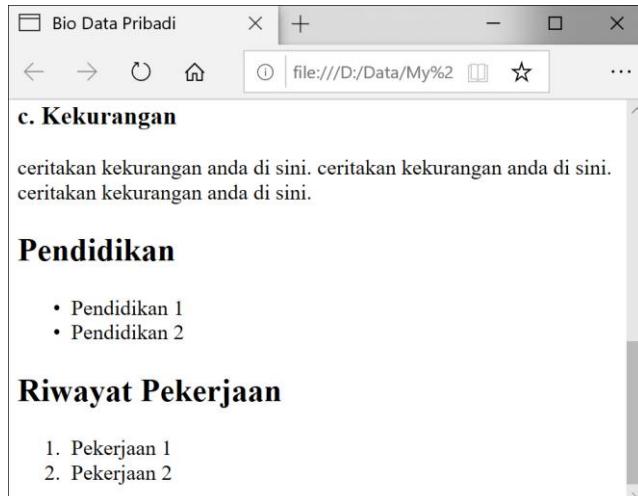
Berikut adalah sebagian kode HTML yang harus ditulis.

```
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*. Pada Gambar 20 adalah contoh tampilan sederhana list pada HTML.



Gambar 20. Memformat teks - List.

Pada Gambar 20 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 `` 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 `` 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 atau terdapat tag 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 20. 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 20. 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>

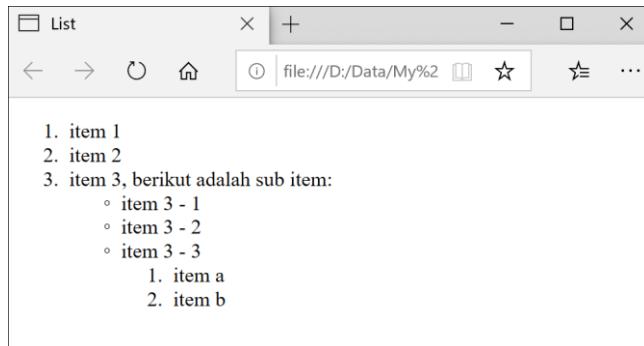
  <body>
    <center>
      <h1>NAMA_ANDA</h1>
      <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
    </center>
    <h2>Pribadi</h2>
    <h3>a. Sifat Umum</h3>
    <p>
      ceritakan sifat umum anda di sini.
      <b>ceritakan sifat <i>umum</i> anda di <u>sini</u>.</b>
      ceritakan sifat 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 21.



Gambar 21. List di dalam list.

Berikut adalah kode HTML dari gambar di atas.

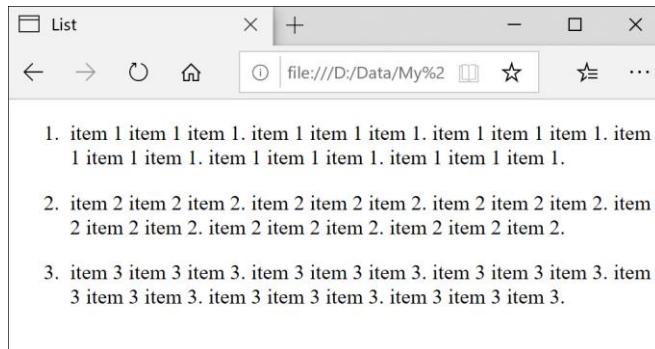
```
list_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>List</title>
    </head>

    <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 paragrapf di dalam list.



Gambar 22. Paragraf di dalam list.

Di bawah ini adalah kode sHTML dari tampilan Gambar 22.

```
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 23. 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 . 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:

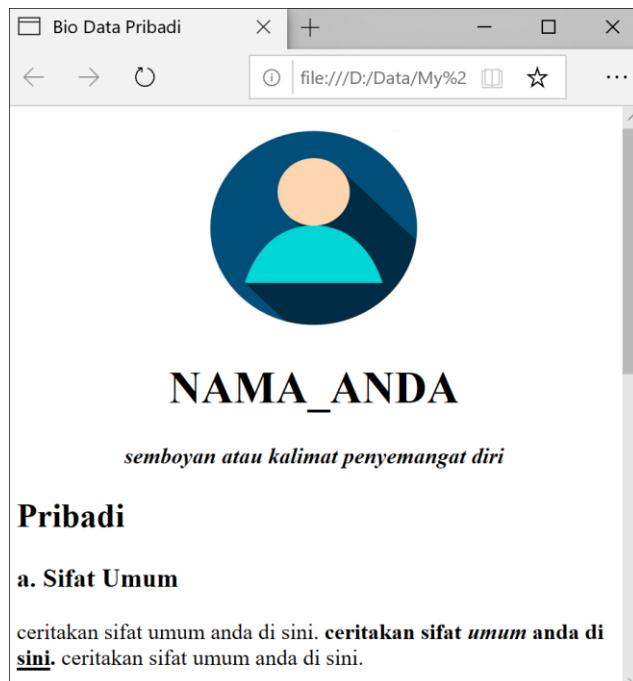
- scr, atribut ini untuk menentukan file yang akan ditampilkan.
- width, atribut ini untuk menentukan ukuran lebar gambar. Jika atribut ini tidak digunakan maka secara otomatis akan digunakan ukuran asli dari file gambar tersebut.
- height, atribut ini untuk menentukan ukuran tinggi gambar. Atribut ini 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. Diharapkan 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 24. Menambahkan gambar pada halaman web.

Buat kode HTML untuk menambahkan sebuah gambar pada halaman web seperti pada Gambar 24. 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>
      
      <h1>NAMA_ANDA</h1>
      <p><i><b>semboyan atau kalimat penyemangat diri</b></i></p>
    </center>
    . . .
  </body>
</html>
```

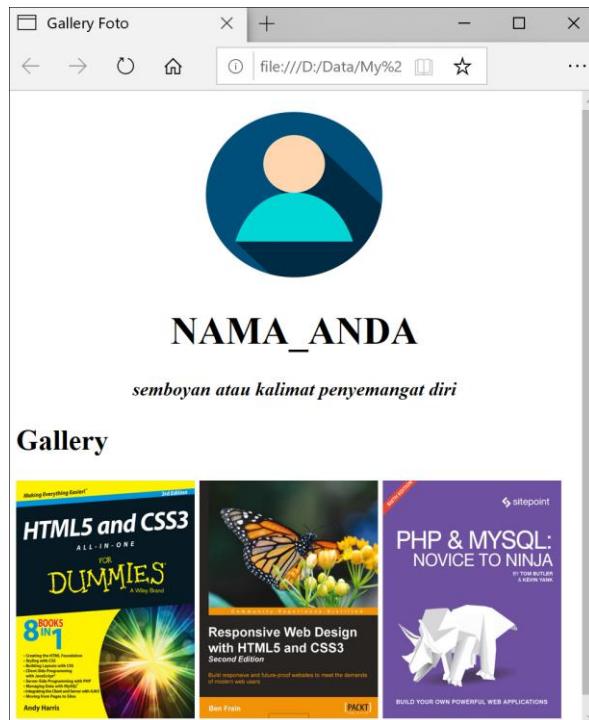
```
</body>  
</html>
```

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 25.

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



Gambar 25. 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>
```

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

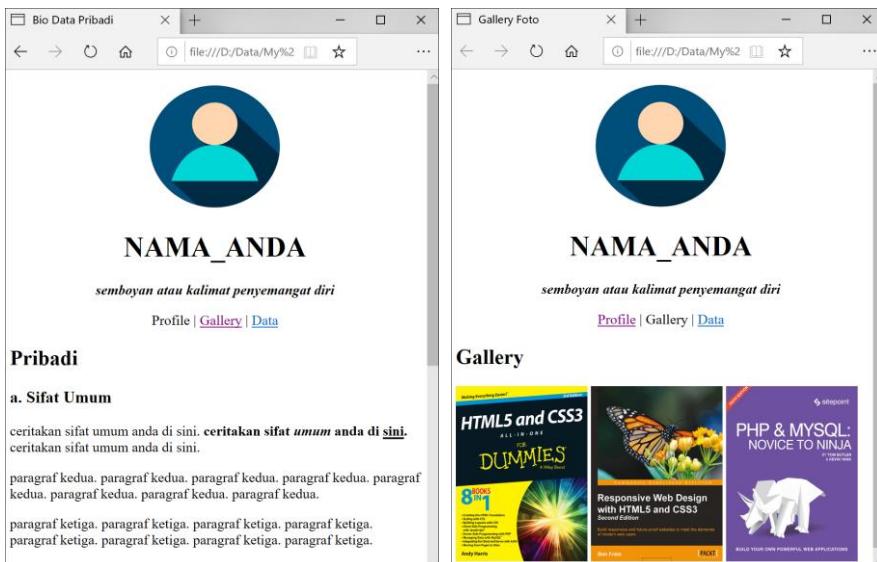
istilah hyperlink. Untuk membuat teks atau gambar menjadi hyperlink digunakan tag <a>.

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 26. Membuat hyperlink untuk menghubungkan dua halaman web.

Tambahkan tag <a> sehingga dimiliki tampilan seperti Gambar 26. 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 <a> 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>
  </body>
</html>
```

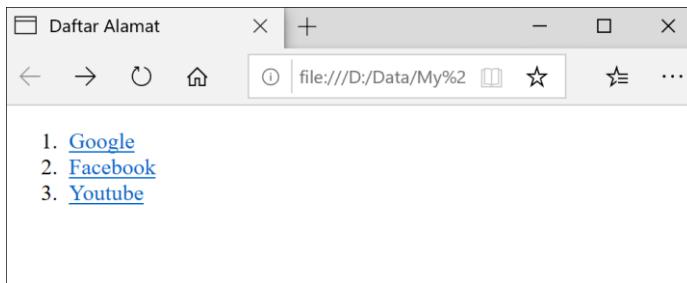
```

</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 atau www.facebook.com dan lain-lain.



Gambar 27. 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>
        </ol>
    </body>
</html>

```

```
<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:

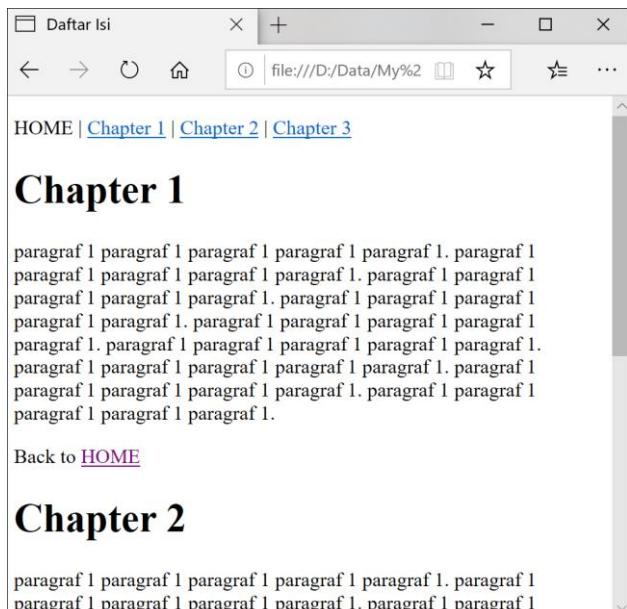
- `_self`, adalah nilai default. Jadi jika tag `<a>` tidak menggunakan atribut target maka secara default nilai atribut ini adalah `_self`. Nilai ini berarti dokumen akan dibuka pada window atau tab yang sama.
- `_parent`, nilai ini berfungsi untuk membuka dokumen pada parent frame.
- `_top`, nilai ini 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 28 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 28. 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 “ ” 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.

```
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>
```

Karena tabel terdiri atas dua baris maka di dalam tag <table></table> ditambahkan dua pasang tag <tr></tr>. Kemudian di dalam baris ditambahkan masing-masing tag <td></td> untuk membuat sebuah kolom.

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>
```

Karena tabel terdiri atas satu baris saja maka di dalam tag <table></table> ditambahkan sepsang tag <tr></tr>. Kemudian didalam tag tersebut tambahkan dua pasang tag <td></td> untuk membuat dua kolom.

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>

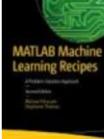
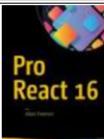
```

Pada table di atas terdiri atas 3 baris dan 4 kolom. Untuk membuatnya maka terlebih dahulu fokus untuk membuat sebuah baris terlebih dahulu, artinya setelah tag `<table></table>` isi dengan sepasang tag `<tr></tr>`. Kemudian didalamnya tambahkan tag `<td></td>` sebanyak jumlah kolom yang yaitu 4 pasang.

Selanjutnya tinggal copy dan paste baris yang sudah dibuat sebanyak dua kali sehingga akhirnya didapat table seperti pada contoh di atas. Karena ingin membuat baris yang paling atas sebagai header maka tag `<td></td>` dapat diubah menjadi tag `<th></th>`. Dengan menggunakan tag ini maka secara otomatis teks didalamnya dicetak tebal.

Latihan 2

Latihan berikutnya adalah membuat tabel seperti pada Gambar 29. 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 29. 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 |
|----------|-------------|
| | Curah Hujan |
| 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 Bulan | Data Curah Hujan | |
|-----------------|------------------|--|
| 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 `<th>` saja di dalam tag `<tr>`.

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>
<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 30.

Contact Us

First Name

Last Name

Company

Phone Number

Email address

Your Website Please include http://

When is the best time of day to reach you?

Morning

Evening

Would you like to receive our email newsletter?

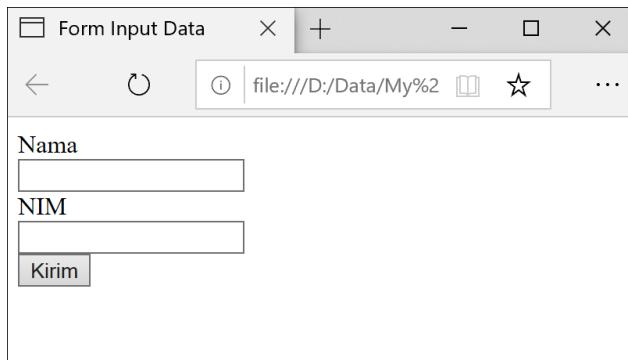
Sure!

Gambar 30. 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 30 dan pada Gambar 31. Namun sebenarnya sebelum tag elemen-elemen yang user lihat tersebut ditambahkan perlu ditambahkan atau digunakan tag <form>.



Gambar 31. 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 (walau 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

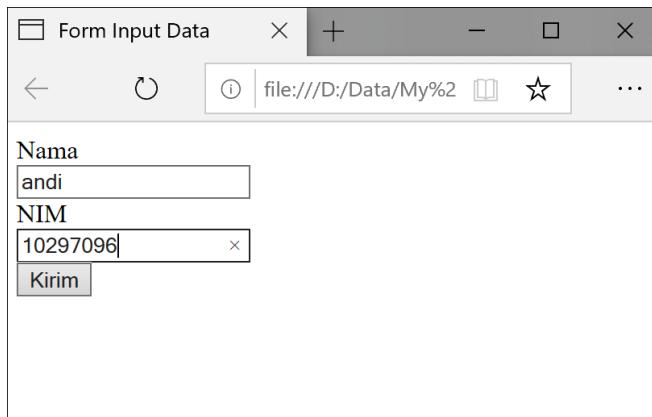
oleh form sebagai client ke server. Jika menggunakan POST, nilai-nilai parameter yang dikirimkan berada di dalam badan pesan sedangkan jika menggunakan metode GET maka parameter dan nilainya dapat dilihat pada URL pada alamat web browser.

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 31. 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".



Gambar 32. Form input data - form - isi nilai kolom input.

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 32.

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 32, 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 **Error! Reference source not found..**

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.

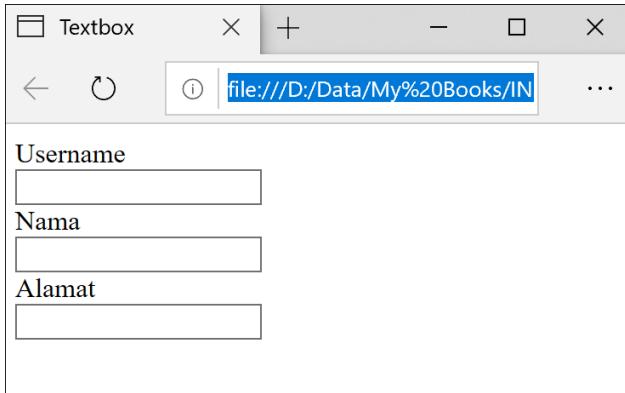
Atribut ketiga adalah id, atribut ini berfungsi untuk memberi identitas unik pada tag. Artinya tidak ada nilai atribut id yang sama digunakan pada sebuah halaman web. Umumnya nilai pada atribut id ini digunakan oleh pemrograman sisi client (client-side programming seperti Javascript) untuk mengkases 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 33. 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.

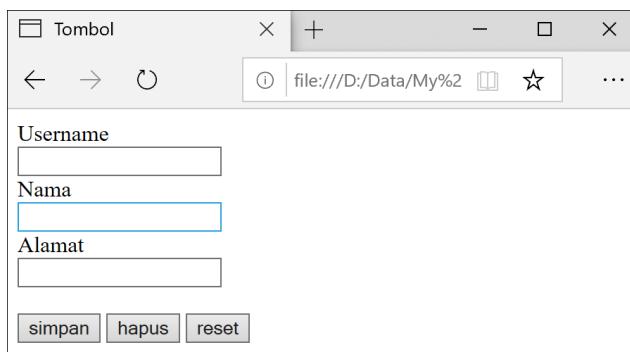
Atribut value berfungsi untuk memberikan teks yang dapat dilihat pada tombol.

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.



Gambar 34. 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.

The screenshot shows a browser window titled "Tombol". The address bar displays "file:///D:/Data/My%20.html". The page content contains a form with two text input fields labeled "Bilangan 1" and "Bilangan 2". Below the inputs are two buttons labeled "kirim" and "reset". The browser interface includes standard controls like back, forward, and search.

Gambar 35. 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 window titled "Registrasi Form". Inside, there's a heading "Registrasi User". Below it are three input fields: "Username" (empty), "Password" (empty), and "Email" (containing "test@gmail.com"). At the bottom are two buttons: "register" and "batal".

Gambar 36. 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>

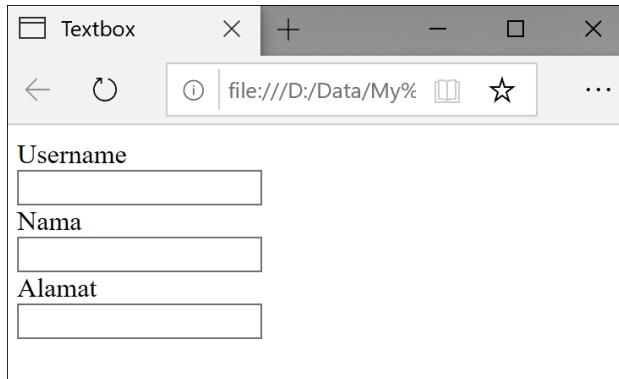
    <body>
        <form method="GET">
            <h3>Registrasi User</h3>
            Username<br>
            <input type="text" name="username" id="textbox_username">
            <br>
            Password<br>
            <input type="password" name="password"
                   id="textbox_password">
            <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 37. 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>
```

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

```

</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 38. Pada Kode di atas dapat dilihat untuk membuat group radio button dilakukan dengan cara memberikan nilai atribut name yang sama.

The screenshot shows a Windows-style file explorer window titled "Radio Button". The main content area contains a form with the following fields:

- Nama:** A text input field.
- Jenis Kelamin:**
 - Pria
 - Wanita
- Pekerjaan:**
 - Sekolah
 - Pegawai Swasta
 - Pegawai Negeri
- Action Buttons:** "kirim data" and "batal".

Gambar 38. 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 bernali berbeda.

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

The screenshot shows a simple web form. At the top, there's a header bar with icons for back, forward, search, and file operations. Below it is a text input field labeled "Nama". Underneath, there's a section titled "Bahasa pemrograman yang dikuasai:" containing two checkboxes: "Java" and "PHP". At the bottom of the form are two buttons: "kirim" and "batal".

Gambar 39. 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">
    </form>
  </body>
</html>
```

```

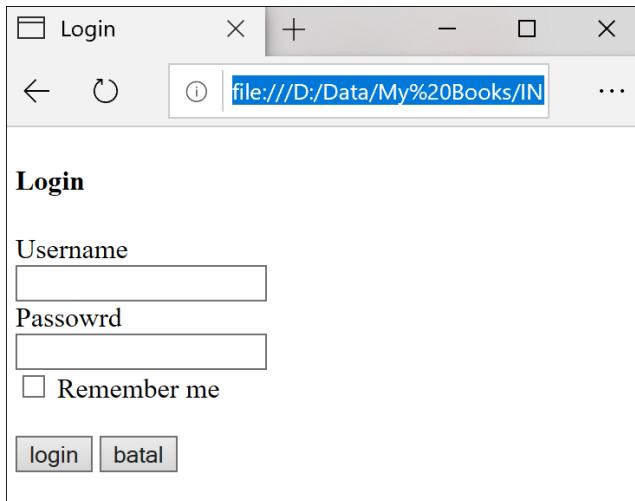
        id="checkbox_programming2">
<label for="checkbox_programming2">PHP</label>

<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.



Gambar 40. 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">
    </form>
  </body>
</html>

```

```
<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.

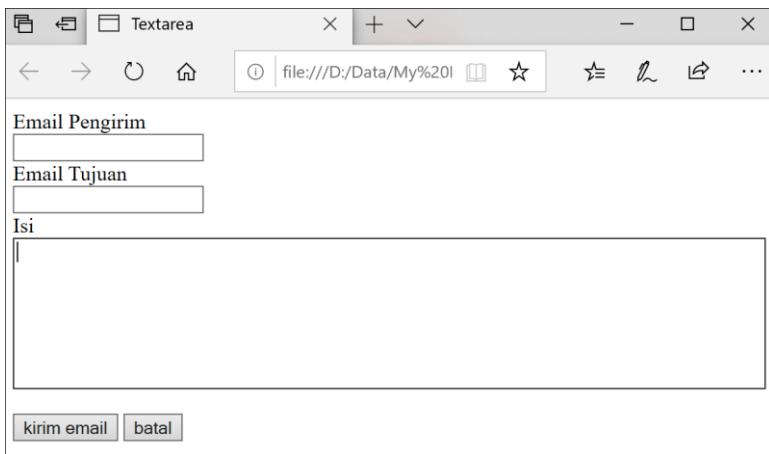
Textarea

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 41. 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

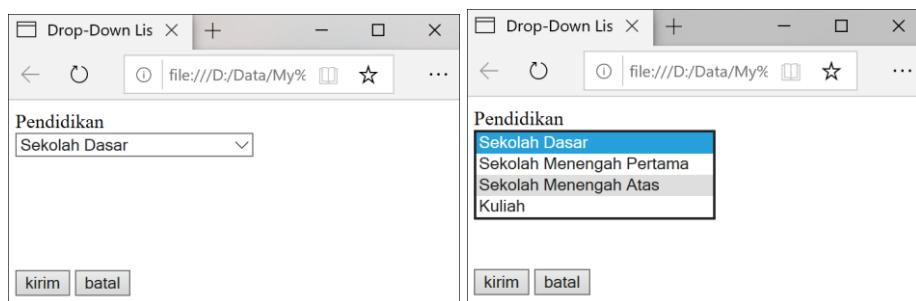
Untuk membuat drop-down list digunakan tag <select> dan <option>. Logika pembuatan drop-down list ini sama dengan logika membuat list dengan atau yang didalamnya ada tag .

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 42. 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.

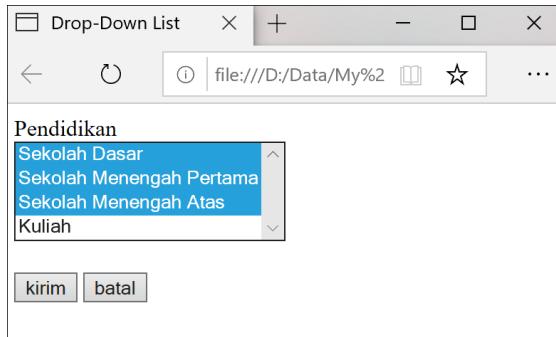
```
dropdownlist_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Drop-Down List</title>
  </head>

  <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 43. 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>
```

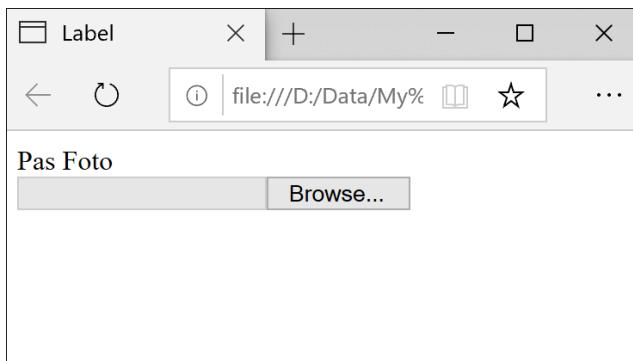
Latihan 1

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 44. Form input data- Upload file - antarmuka upload file pada Microsoft Edge.

Project

Berikut ini adalah project yang harus dikerjakan dan diselesaikan dengan menggunakan tag-tag dasar yang telah diberikan di atas.

Web Profile Pribadi

Deskripsi Project

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 600 x 750 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.

Langkah-Langkah Penggerjaan

Gunakan halaman profile.html yang sebelumnya telah dibuat dengan kode lengkap sebagai berikut.

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

```

Halaman ini menampilkan gambar dari file profile.jpg yang disimpan pada folder images. Pastikan buat folder images dengan lokasi yang sama dengan file profile.html berada. Kemudian salin foto ke folder ini dan ganti nama file foto tersebut menjadi profile.jpg. Kemudian tambahkan menu yang berisi hyperlink untuk mengakses halaman gallery.html dan data.html sebagai berikut.

```

<p>
    Profile | 
    <a href="gallery.html">Gallery</a> | 
    <a href="data.html">Data</a>
</p>

```

Selanjutnya adalah membuat file gallery.html untuk menampilkan gallery yang berisi 20 foto. Berikut adalah sebagian kode dari halaman ini.

```

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>
<!-- salin awal -->
<p>
    <a href="images/gallery/gallery-01.jpg" target="_blank">
        
    </a>
    <a href="images/gallery/gallery-02.jpg" target="_blank">
        
    </a>
    <a href="images/gallery/gallery-03.jpg" target="_blank">
        
    </a>
    <a href="images/gallery/gallery-04.jpg" target="_blank">
        
    </a>
    <a href="images/gallery/gallery-05.jpg" target="_blank">
        
    </a>
</p>
<!-- salin akhir -->
...
<!--
    copy bagian dari "salin awal" sampai "salin akhir"
    paste sebanyak 3 kali agar lengkap dibuat 20 foto gallery
    sesuaikan nama file dengan file yang dimiliki
-->
</body>
</html>

```

Bagian atas halaman ini sama dengan halaman profile.html, yaitu memiliki foto profile, nama dan menu. Ada sedikit perbedaan menu pada halaman ini dengan halaman sebelumnya. Berikut adalah kode yang digunakan pada halaman gallery.html. Pada menu ini terdapat hyperlink untuk menuju ke halaman profile.html dan data.html.

```
<p>
    <a href="profile.html">Profile</a> |
    Gallery |
    <a href="data.html">Data</a>
</p>
```

Selanjutnya membuat kode untuk menampilkan 20 foto dengan kode sebagai berikut.

```
<p>
    <a href="images/gallery/gallery-01.jpg" target="_blank">
        
    </a>
    . . .
</p>
. . .
```

Di dalam tag `<p>` ditambahkan 5 elemen yang berisi hyperlink dan image seperti potongan kode di atas. Pada hyperlink menggunakan atribut href yang merujuk ke file asli dari gambar. Atribut `target="_blank"` digunakan agar ketika gambar diklik akan ditampilkan pada tab baru.

Di tengah elemen hyperlink, tambahkan tag `` untuk menampilkan gambar. Gambar gallery-01.jpg sampai gallery-20.jpg disimpan pada folder gallery yang ada di dalam folder images.

Langkah berikutnya adalah membuat halaman data.html untuk menampilkan 10 buku yang pernah dibaca dalam bentuk tabel. Berikut adalah kode lengkap dari file data.html.

```
data.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> |
    <a href="gallery.html">Gallery</a> |
    Data
</p>
</center>
<h2><a name="TOP">Daftar Buku</a></h2>
<table width="100%" border="1">
    <tr>
        <th>Judul Buku</th>
        <th>Pengarang</th>
        <th>Foto</th>
        <th>Detail</th>
    </tr>
    <tr>
        <td>Judul 1</td>
        <td>Pengarang 1</td>
        <td>
            
        </td>
        <td align="center">
            <a href="#book1">Detail</a>
        </td>
    </tr>
    <!--
        salin baris tabel ini sebanyak 9 baris
        dan sesuaikan isinya dengan data yang
        ingin ditampilkan
    -->
</table>

<h2>Detail Buku</h2>
<table width="100%" border="1">
    <tr>
        <td></td>
        <td valign="top">
            Judul: <a name="book1">Judul 1</a><br>
            Pengarang: Pengarang 1<br>
            detail detail detail detail detail
            detail detail detail detail detail
            detail detail detail detail detail
            detail detail detail detail detail <br>
            <a href="#TOP">TOP</a>
        </td>
    </tr>

```

```

<!--
    salin baris tabel ini sebanyak 9 baris
    dan sesuaikan isinya dengan data yang
    ingin ditampilkan
-->
</table>
</body>
</html>

```

Pada halaman ini memiliki menu yang sedikit berbeda dengan halaman profile.html dan gallery.html. Hyperlink digunakan untuk menunjuk ke halaman profile dan gallery saja.

```

<p>
    <a href="profile.html">Profile</a> |
    <a href="gallery.html">Gallery</a> |
    Data
</p>

```

Pada bagian konten atau isi terbagi atas dua bagian yaitu:

1. Daftar Buku.
2. Detail Buku.

Keduanya ditandai oleh teks dengan menggunakan tag <h2>. Pada bagian pertama yaitu Daftar Buku ditampilkan data buku dalam bentuk tabel seperti berikut.

```

<table width="100%" border="1">
    <tr>
        <th>Judul Buku</th>
        <th>Pengarang</th>
        <th>Foto</th>
        <th>Detail</th>
    </tr>
    <tr>
        <td>Judul 1</td>
        <td>Pengarang 1</td>
        <td>
            
        </td>
        <td align="center">
            <a href="#book1">Detail</a>
        </td>
    </tr>
    <!--
        salin baris tabel ini sebanyak 9 baris
        dan sesuaikan isinya dengan data yang
    -->

```

```
    ingin ditampilkan  
-->  
</table>
```

Pada kolom foto ditampilkan gambar buku. File gambar disimpan pada folder books di dalam folder images. Setiap baris data memiliki hyperlink Detail yang ketika diklik akan menuju pada detail buku tersebut. Hyperlink menggunakan tag dengan atribut #book1 untuk baris pertama. Nilai #book1 ini dapat dilihat pada potongan kode di bawah ini, dimana Judul 1 diapit oleh tag dengan atribut name="book1". Yang artinya ketika hyperlink Detail diklik maka secara otomatis halaman akan bergeser pada bagian detail ini.

Sedangkan pada bagian ditampilkan detail buku dalam bentuk tabel sebagai berikut.

```
<table width="100%" border="1">  
  <tr>  
    <td></td>  
    <td valign="top">  
      Judul: <a name="book1">Judul 1</a><br>  
      Pengarang: Pengarang 1<br>  
      detail detail detail detail detail  
      detail detail detail detail detail  
      detail detail detail detail detail  
      detail detail detail detail detail <br>  
      <a href="#TOP">TOP</a>  
    </td>  
  </tr>  
<!--  
  salin baris tabel ini sebanyak 9 baris  
  dan sesuaikan isinya dengan data yang  
  ingin ditampilkan  
-->  
</table>
```

Pada setiap baris ini terdapat hyperlink dengan atribut href="#TOP" yang merujuk pada teks "Daftar Buku" berikut ini. Hyperlink di atas mempunyai fungsi jika diklik akan mengantarkan ke bagian di bawah ini.

```
<h2><a name="TOP">Daftar Buku</a></h2>
```

Pada Gambar 45 adalah antarmuka dari ketiga halaman tersebut.

NAMA_ANDA

semboyan atau kalimat penyemangat diri

[Profile](#) | [Gallery](#) | [Data](#)

Pribadi

a. Sifat Umum

ceritakan sifat umum anda di sini. **ceritakan sifat umum anda di sini.** ceritakan sifat umum anda di sini.

paragraf kedua. paragraf kedua. paragraf kedua. paragraf kedua. paragraf kedua. paragraf kedua.

paragraf ketiga. paragraf ketiga. paragraf ketiga. paragraf ketiga. paragraf ketiga. paragraf ketiga.

Daftar Buku

| Judul Buku | Pengarang | Foto | Detail |
|------------|-------------|------|------------------------|
| Judul 1 | Pengarang 1 | | Detail |
| | | | |
| | | | |

Detail Buku

| | |
|--|--|
| | Judul: Judul 1 Pengarang: Pengarang 1 detail TOP |
| | Judul: Judul 2 Pengarang: Pengarang 2 detail TOP |
| | Judul: Judul 3 Pengarang: Pengarang 3 detail TOP |

Gambar 45. Antarmuka project Web Profile Pribadi.

Form & Daftar Buku Tamu

Deskripsi Project

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 Tanggal . |  |
| Komentar | |
| Nama (Email) memberikan komentar pada Tanggal . |  |
| Komentar | |
| Nama (Email) memberikan komentar pada Tanggal . |  |
| Komentar | |
| Nama (Email) memberikan komentar pada Tanggal . |  |
| Komentar | |

Langkah-Langkah Penggerjaan

Berikut adalah kode lengkap dari project ini.

```
guestbook.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Buku Tamu</title>
  </head>
```

```

<body>
    <h2>Form Buku Tamu</h1>
    <form method="POST">
        <label>Nama</label>:
        <input type="text" name="nama">
        <br>
        <label>Email</label>:
        <input type="email" name="email">
        <br>
        <label>Tanggal</label>:
        <input type="date" name="tanggal">
        <br>
        <label>Apakah website ini bagus?</label>
        <a href="#">
            
        </a>
        <a href="#">
            
        </a>
        <a href="#">
            
        </a>
        <br>
        <label>Komentar</label><br>
        <textarea rows="4" cols="50"></textarea><br><br>
        <input type="submit" value="Kirim">
        <input type="reset" value="Batal">
    </form>

    <h2>Buku Tamu</h1>
    <table border="1" width="100%">
        <!-- salin awal-->
        <tr>
            <td width="85%">
                <strong>Nama (Email)</strong>
                memberikan komentar pada
                <b>31 Mei 2020</b>
            </td>
            <td width="15%">
                
                
                
            </td>
        </tr>
        <tr>
            <td colspan="2" style="text-align: center; vertical-align: middle;">
                komentar komentar komentar komentar
            </td>
        </tr>
    </table>

```

```

        komentar komentar komentar komentar
        komentar komentar komentar komentar
    </td>
</tr>
<tr>
    <td colspan="2" bgcolor="grey">&nbsp;</td>
</tr>
<!-- salin akhir-->
<!--
    copy dari "salin awal" sampai "salin akhir"
    kemudian paste sejumlah yang diinginkan
-->
</table>
</body>
</html>
```

Halaman ini memiliki dua bagian. Bagian pertama adalah form untuk mengisi buku tamu dengan penggalan kode sebagai berikut.

```

<form method="POST">
    <label>Nama</label>:<input type="text" name="nama"><br>
    <label>Email</label>:<input type="email" name="email"><br>
    <label>Tanggal</label>:<input type="date" name="tanggal"><br>
    <label>Apakah website ini bagus?</label>
    <a href="#"></a>
    <a href="#"></a>
    <a href="#"></a>
    <br>
    <label>Komentar</label><br>
    <textarea rows="4" cols="50"></textarea><br><br>
    <input type="submit" value="Kirim">
    <input type="reset" value="Batal">
</form>
```

Untuk mengisi nilai pertanyaan “Apakah website ini bagus?” digunakan gambar bintang perak sebanyak 3 buah. Untuk sementara gambar ketiga bintang perak itu menjadi hyperlink.

Bagian kedua adalah menampilkan daftar isian buku tamu yang ditampilkan dalam bentuk tabel dengan penggalan kode sebagai berikut.

```

<table border="1" width="100%">
    <!-- salin awal-->
    <tr>
        <td width="85%">
            <strong>Nama (Email)</strong> memberikan komentar pada
            <b>31 Mei 2020</b>
        </td>
```

```

<td width="15%">
    
    
    
</td>
</tr>
<tr>
    <td colspan="2">
        komentar komentar komentar komentar
        komentar komentar komentar komentar
        komentar komentar komentar komentar
    </td>
</tr>
<tr>
    <td colspan="2" bgcolor="grey">&ampnbsp</td>
</tr>
<!-- salin akhir-->
<!--
    copy dari "salin awal" sampai "salin akhir"
    kemudian paste sejumlah yang diinginkan
-->
</table>

```

Sebuah komentar pada buku tamu ini menggunakan tiga baris seperti terlihat pada Gambar 46.

| | |
|--|--|
| Nama (Email) memberikan komentar pada 31 Mei 2020 |  |
| komentar komentar komentar komentar komentar komentar komentar komentar komentar komentar | |

Gambar 46. Sebuah komentar pada buku tamu.

Baris pertama adalah menampilkan informasi nama dan email penulis komentar disertai dengan tanggal. Dan juga terdapat rating berupa bintang. Baris kedua adalah isi komentar. Sedangkan baris ketiga adalah baris berwarna abu-abu yang berfungsi untuk memisahkan satu komentar dengan komentar yang lain.

Pada Gambar 47 adalah antarmuka dari kode guestbook.html di atas.

The screenshot shows a web browser window titled "Buku Tamu". The page contains a form with fields for Name, Email, and Date, followed by a rating scale and a comment area. Below the form is a list of two comments, each with a timestamp, a star rating, and the text of the comment.

Form Buku Tamu

Nama:

Email:

Tanggal: mm/dd/yyyy

Apakah website ini bagus? ★ ★ ★

Komentar

Buku Tamu

| | |
|---|---|
| Nama (Email) memberikan komentar pada 31 Mei 2020 | ★ ★ ★ |
| komentar komentar komentar komentar komentar komentar komentar komentar | |
| <hr/> | |
| Nama (Email) memberikan komentar pada 31 Mei 2020 | ★ ★ ★ |
| komentar komentar komentar komentar komentar komentar komentar komentar | |
| <hr/> | |

Gambar 47. Antarmuka halaman guestbook.html.

Tag HTML Lajut

Pada sub bab ini diberikan tag-tag HTML selain yang telah dibahas pada sub bab sebelumnya. Pada aplikasi web sederhana, tag-tag HTML ini mungkin jarang digunakan namun tetap perlu diketahui jika suatu saat diperlukan.

Teks

Preformatted Text

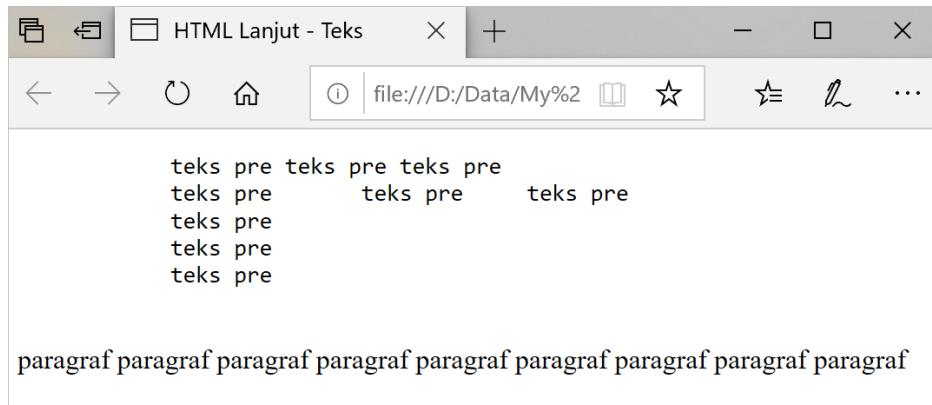
Pada sub bab sebelumnya diperkenalkan tag <p> untuk menulis teks dalam paragraf. Pada sub bab ini diperkenalkan tag <pre> yang berfungsi untuk menampilkan teks didalamnya dengan format yang memperhatikan spasi dan pindah baris teks pada kode HTML.

Untuk mengetahui maksud dari penjelasan di atas dapat melihat kode berikut ini.

```
teks lanjut 1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>HTML Lanjut - Teks</title>
  </head>

  <body>
    <pre>
      teks pre teks pre teks pre
      teks pre           teks pre       teks pre
      teks pre
      teks pre
      teks pre
    </pre>
    <p>
      paragraf paragraf paragraf
      paragraf paragraf paragraf
      paragraf       paragraf
      paragraf
    </p>
  </body>
</html>
```

Dan hasilnya dapat dilihat pada Gambar 48. Dapat dilihat dengan tag `<pre>` spasi dan pindah baris ada pengaruhnya namun tidak pada tag `<p>`.



Gambar 48. Perbedaan tag pre dan p.

Mengutip & Citasi

Untuk melakukan mengutip (quote) suatu teks dapat menggunakan tag <q>. Teks yang berada di tengah tag ini akan secara otomatis diberi tanda kutip. Sedangkan tag lain adalah <blockquote> yang berfungsi untuk melakukan rujukan atau citasi. Teks yang berada di antara tag ini akan ditulis tanpa tanda kutip namun terlihat menjorok ke dalam.

Berikut adalah contoh penggunaan kedua tag tersebut.

```
teks_lanjut_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Teks</title>
    </head>

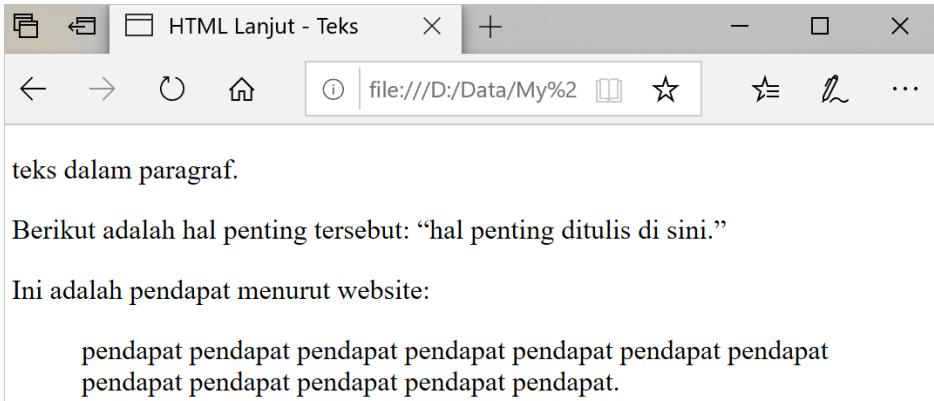
    <body>
        <p>teks dalam paragraf.</p>

        <p>
            Berikut adalah hal penting tersebut:
            <q>hal penting ditulis di sini.</q>
        </p>

        <p>Ini adalah pendapat menurut website:</p>
        <blockquote cite="http://www.rezafaisal.net">
            pendapat pendapat pendapat pendapat
            pendapat pendapat pendapat pendapat
            pendapat pendapat pendapat pendapat.
        </blockquote>
    </body>
</html>
```

Dan hasilnya dapat dilihat pada Gambar 49. Teks yang berada diantara tag <q> dapat ditampilkan mengikuti dan bersambung dengan teks sebelumnya namun diberikan tanda kutip.

Sedangkan teks yang berada pada tag <blockquote> akan ditampilkan pada paragraf sendiri tanpa tanda kutip dan menjorok ke dalam. Tag ini juga mempunyai atribut cite yang dapat diisi dengan sumber kalimat dari teks di dalam tag ini.



Gambar 49. Perbedaan tag `q` dan `blockquote`.

Format Teks

Pada sub bab sebelumnya telah diberikan dasar memformat teks seperti cetak miring, tebal atau garis bawah. Pada bagian ini diberikan tag-tag lain untuk melakukan hal tersebut dan format-format lainnya.

Untuk membuat cetak miring dapat digunakan alternatif tag ``, sedangkan untuk cetak tebal dapat menggunakan tag `` dan garis bawah dapat menggunakan tag `<ins>`. Sebenarnya fungsi tag `` ini mempertegas kata. Sedangkan tag `` untuk menyatakan bahwa teks tersebut penting. Dan `<ins>` menyatakan teks yang disisipkan. Seperti yang terlihat pada Gambar 50.

Bagian I

Penekanan teks dengan garis miring

Teks penting dengan cetak tebal

Favorit saya adalah warna merah.

Gambar 50. Contoh penggunaan tag `strong`, `em` dan `ins`.

Format teks yang lain adalah dengan menggunakan tag `` untuk mencoret suatu teks. Tag `<mark>` berfungsi untuk memberikan highlight pada teks. Kemudian tag `<sup>` untuk membuat teks menjadi agak kecil dan letaknya agak di atas tulisan normal. Sedangkan tag `<sub>` berfungsi membuat teks menjadi agak kecil dan letaknya agak di bawah tulisan normal. Seperti yang terlihat pada Gambar 51.

Bagian II

Warna favorit saya adalah **biru** merah.

Teks dengan **highlight**.

Contoh teks **subscripted**

Contoh teks **superscripted**

Gambar 51. Contoh penggunaan tag del, mark, sup & sub.

Tag `<code>` dapat digunakan untuk menulis kode program. Sedangkan untuk menyatakan teks adalah tombol pada input keyboard dapat digunakan tag `<kbd>`. Tag `<samp>` menyatakan teks adalah keluaran dari komputer. Dan tag `<var>` menyatakan teks adalah variable. Pada Gambar 52 dapat dilihat hasil dari tag-tag tersebut.

Bagian III

```
x = 5;  
y = 6;  
z = x + y;
```

Tombol untuk menyimpan dokumen **Ctrl + S**

Pesan Error!

Rumus: $E = mc^2$.

Gambar 52. Contoh penggunaan tag code, kbd, samp, & var.

Contoh penggunaan tag-tag di atas dapat dilihat pada kode di bawah ini.

```
teks_lanjut_3.html  
<!DOCTYPE HTML>  
<html>  
  <head>  
    <title>HTML Lanjut - Teks</title>  
  </head>  
  
  <body>  
    <h3>Bagian I</h3>  
    <em>Penekanan teks dengan garis miring</em>  
    <br>
```

```

<strong>Teks penting dengan cetak tebal</strong>
<br>
<p>Favorit saya adalah <ins>warna</ins> merah.</p>
<hr>
<h3>Bagian II</h3>
<p>Warna favorit saya adalah <del>biru</del> merah.</p>
Teks dengan <mark>highlight</mark>.
<br>
<p>Contoh teks <sub>subscripted</sub></p>
<p>Contoh teks <sup>superscripted</sup></p>
<hr>
<h3>Bagian III</h3>
<code>
    x = 5;<br>
    y = 6;<br>
    z = x + y;
</code>
<p>Tombol untuk menyimpan dokumen <kbd>Ctrl + S</kbd></p>
<p>Pesan <samp>Error!</samp></p>
Rumus: <var>E</var> = <var>mc</var><sup>2</sup>.
</body>
</html>

```

Form

Pada sub bab sebelumnya telah diberikan beberapa tipe dari tag `<input>`. Tipe yang telah dibahas adalah:

- Button.
- Checkbox.
- File.
- Submit.
- Reset.
- Text.
- Password.
- Radio.

Selain itu masih ada beberapa tipe yang dapat digunakan yaitu:

- Color, untuk memilih warna.
- Date, untuk memilih tanggal.
- Datetime dan Datetime Local untuk memilih tanggal dan jam.
- Range, untuk memilih nilai rentang.
- Number, input untuk menerima nilai angka.

- Tel, input untuk menerima nomor telepon.
- Time, input untuk menerima nilai jam.
- Url, input untuk menerima nilai alamat atau url website.
- Week, input untuk memilih minggu dan setahun.
- Month, input untuk memilih bulan dan tahun
- Search, input untuk menerima kata kunci pada suatu pencarian.

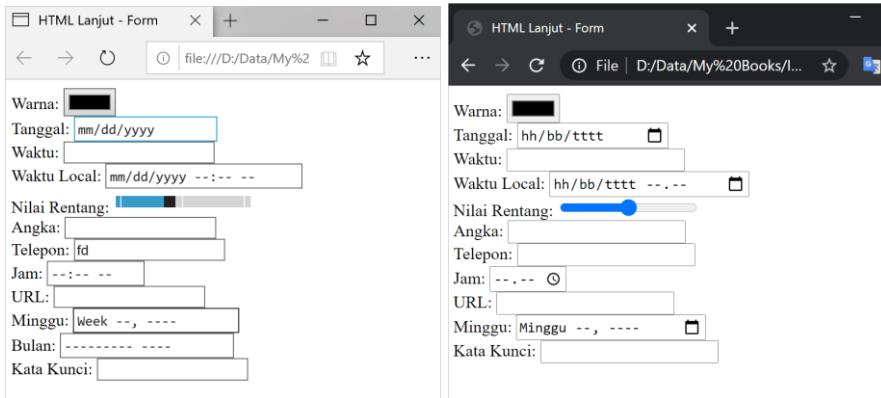
Berikut contoh kode dari dari tipe-tipe input yang telah disebutkan di atas.

```
form_lanjut_1.html

<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Form</title>
    </head>

    <body>
        <form method="POST">
            Warna: <input type="color" name="warna">
            <br>
            Tanggal: <input type="date" name="tanggal">
            <br>
            Waktu: <input type="datetime" name="waktu">
            <br>
            Waktu Local: <input type="datetime-local"
                           name="waktu-lokal">
            <br>
            Nilai Rentang: <input type="range" name="nilai-rentang">
            <br>
            Angka: <input type="number" name="angka">
            <br>
            Telepon: <input type="tel" name="telepon">
            <br>
            Jam: <input type="time" name="jam">
            <br>
            URL: <input type="url" name="url">
            <br>
            Minggu: <input type="week" name="minggu">
            <br>
            Bulan: <input type="month" name="bulan">
            <br>
            Kata Kunci: <input type="search" name="keyword">
        </form>
    </body>
</html>
```

Hasil dari kode di atas akan memiliki antarmuka yang berbeda jika dibuka pada beda web browser. Pada dapat dilihat perbedaan antarmukanya pada web browser MS Edge dan Google Chrome.



Gambar 53. Antarmuka pada MS Edge & Google Chrome.

Tipe input yang telah dijelaskan di atas dapat dilihat secara visual. User akan akan memilih atau memasukkan suatu nilai. Tetapi ada juga input yang tidak terlihat pada halaman web namun memiliki nilai. Tipe input yang digunakan adalah hidden dengan contoh sebagai berikut.

```
form_lanjut_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Form</title>
    </head>

    <body>
        <form method="GET">
            <input type="hidden" name="hidden_value_1" value="1">
            <input type="submit" value="kirim">
        </form>
    </body>
</html>
```

Jika halaman ini ditampilkan pada web browser maka yang terlihat hanyalah tombol saja. Ketika tombol diklik dan form disubmit maka nilai hidden_value_1 dapat dilihat pada alamat web browser.

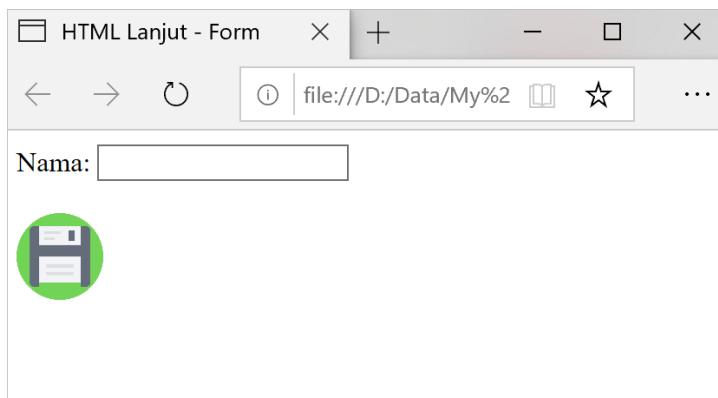
Tipe input tidak hanya untuk mengisi nilai atau memilih nilai saja, tapi juga berbentuk tombol. Pada sub bab sebelumnya telah dikenal tombol-tombol yang dibuat dari tag <input> dengan nilai atribut type adalah submit, reset dan

button. selain itu juga ada tombol berbentuk gambar. Untuk membuat tombol dengan menggunakan gambar dapat menggunakan tag <input> dengan tipe adalah image. Kode berikut adalah contoh penggunaan tipe ini.

```
form_lanjut_3.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Form</title>
    </head>

    <body>
        <form method="GET">
            Nama: <input type="text" name="nama" >
            <br><br>
            <input type="image" src="images/button/save.png"
                   width="50px">
        </form>
    </body>
</html>
```

Hasilnya dapat dilihat pada Gambar 54.



Gambar 54. Tombol berbentuk gambar.

Iframe

Tag <iframe> berfungsi untuk menampilkan halaman website pada halaman web yang dibuat. Sebagai contoh dibuat halaman dengan nama iframe_1.html. Halaman ini memiliki kotak yang didalamnya dapat melihat halaman profile.html.

Berikut adalah contoh kode HTML untuk hal tersebut di atas.

```
iframe 1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Form</title>
    </head>

    <body>
        <iframe src="profile.html" width="500px" height="500px">
        </iframe>
    </body>
</html>
```

Hasilnya dapat dilihat pada Gambar 55.



Gambar 55. Contoh iframe 1.

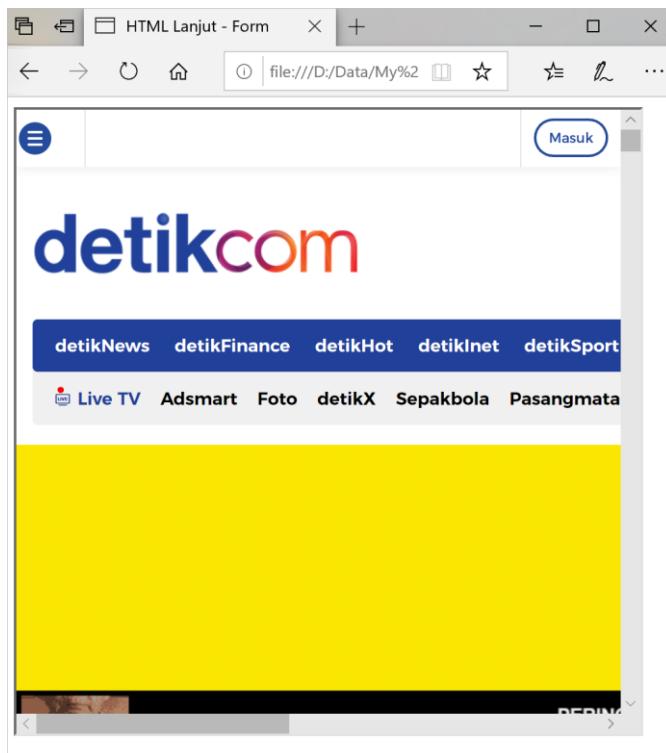
Contoh lain penggunaan tag ini adalah sebagai berikut.

```
iframe_2.html  
<!DOCTYPE HTML>  
<html>
```

```
<head>
    <title>HTML Lanjut - Form</title>
</head>

<body>
    <iframe src="http://www.detik.com" width="500px"
            height="500px">
    </iframe>
</body>
</html>
```

Pada contoh di atas atribut scr berisi alamat suatu web, maka iframe akan menampilkan web tersebut seperti pada Gambar 56.



Gambar 56. Web detik.com pada iframe.

Contoh lain adalah membuat iframe yang dapat berisi alamat yang bisa ditentukan user dengan cara mengklik hyperlink seperti berikut ini.

```
iframe 3.html
<!DOCTYPE HTML>
<html>
<head>
```

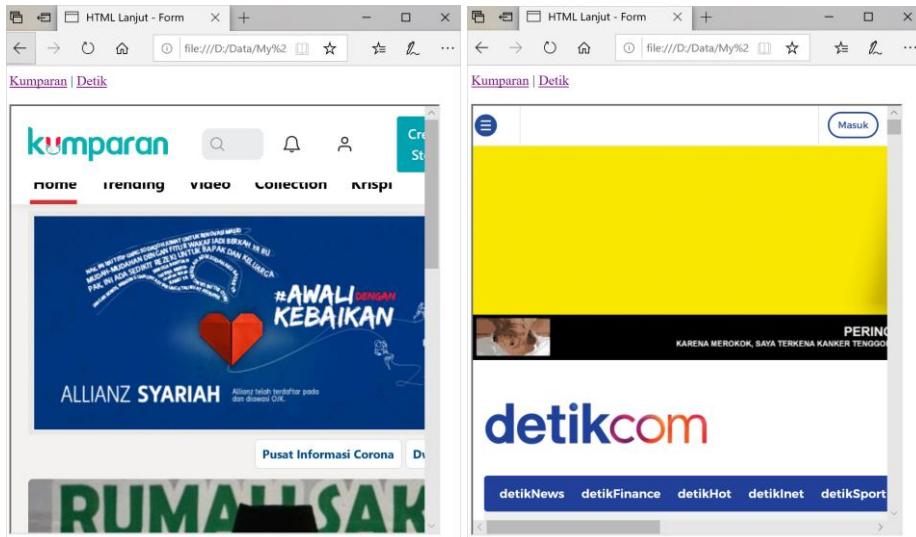
```

<title>HTML Lanjut - Form</title>
</head>

<body>
    <a href="https://www.kumparan.com" target="layar">Kumparan</a> |
    <a href="http://www.detik.com" target="layar">Detik</a>
    <br><br>
    <iframe src="profile.html" name="layar" width="500px"
            height="500px">
    </iframe>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 57. Ketika hyperlink Kumparan diklik maka kotak iframe akan menampilkan web tersebut. Dan ketika hyperlink Detik diklik maka iframe menampilkan web Detik.



Gambar 57. Iframe dan hyperlink.

Iframe juga dapat digunakan untuk menampilkan video Youtube pada halaman web. Berikut adalah contoh kode yang digunakan untuk hal tersebut.

```

iframe_4.html

<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - IFrame</title>
    </head>

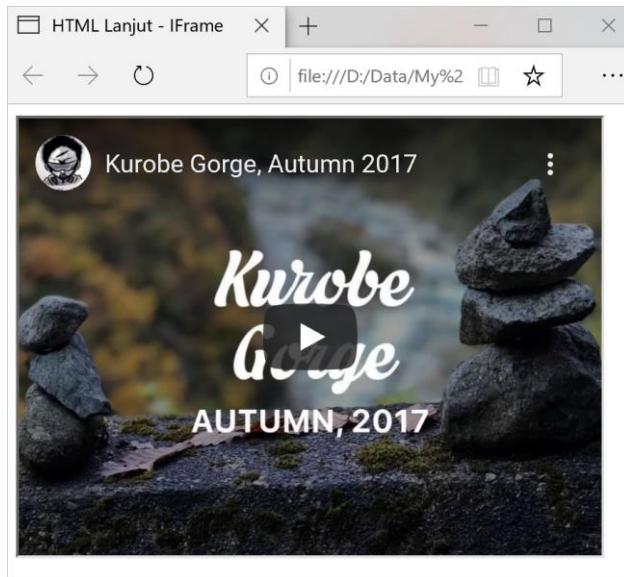
```

```
<body>
    <iframe width="420" height="315"
        src="https://www.youtube.com/embed/_4zGz14TxA?controls=1">
    </iframe>
</body>
</html>
```

Untuk menampilkan video Youtube pada iframe harus memperhatikan alamat yang digunakan. Ada perbedaan alamat/URL yang dipergunakan saat mengakses video Youtube pada umumnya. Alamat/URL yang digunakan terdapat tambahan “embed” diantara alamat dan id dari video. Selain itu dapat diberikan opsi untuk mengatur video player seperti:

- controls untuk menampilkan kontrol video player. Jika parameter ini diberi nilai 1 maka kontrol akan ditampilkan.
- loop, secara default video hanya dimainkan sekali. Jika ingin memainkan video berulang-ulang maka parameter loop diberi nilai 1.

Hasilnya dapat dilihat pada Gambar 58.



Gambar 58. Video youtube pada iframe.

Media

Multimedia adalah sesuatu yang dapat didengar dan dilihat seperti gambar, audio dan video. Format multimedia yang didukung oleh standar HTML adalah:

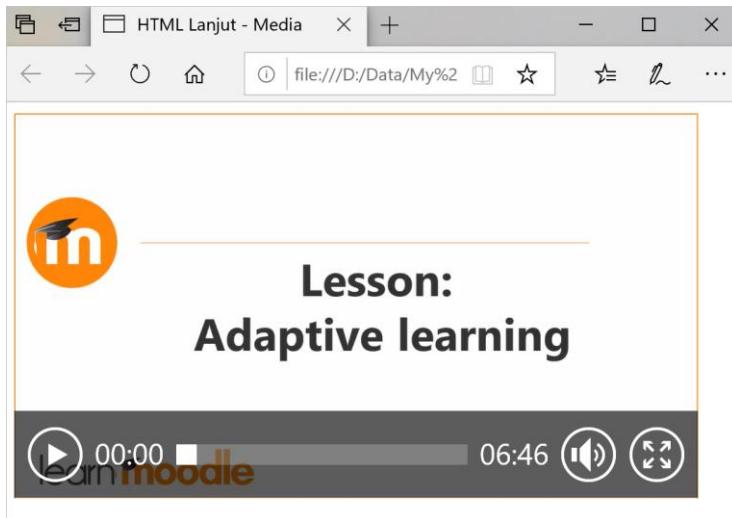
1. Video
 - a. MP4.
 - b. WebM.
 - c. Ogg video.
2. Audio
 - a. MP3.
 - b. WAV.
 - c. Ogg audio.

Untuk menampilkan video pada halaman HTML digunakan tag <video>. Berikut adalah contoh dari penggunaan tag ini.

```
media_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Media</title>
    </head>

    <body>
        <video width="500" controls>
            <source src="multimedia/movie.mp4" type="video/mp4">
            Your browser does not support the video tag.
        </video>
    </body>
</html>
```

Pada tag <video> digunakan atribut width untuk mengatur lebar video player yang ditampilkan pada halaman web. Sedangkan atribut controls untuk menampilkan kontrol seperti play, pause, volume suara dan lain-lain pada video player seperti yang terlihat pada Gambar 59. Kemudian di dalam tag <video> ditambahkan tag <source> untuk menentukan lokasi file video yang akan ditampilkan dengan mengisi atribut src. Sesuaikan tipe file dengan menulisnya pada atribut type.



Gambar 59. Video player dengan kontrol.

Atribut lain yang dapat diberikan pada tag <video> adalah:

- autoplay untuk memutar video secara langsung namun tidak ditampilkan kontrol.
- height untuk menentukan tinggi dari video player.
- loop untuk memerintahkan agar video akan dimainkan kembali setelah selesai.
- muted untuk mematikan audio dari video.
- poster untuk menentukan cover dari video player yang akan ditampilkan selagi video diunduh.

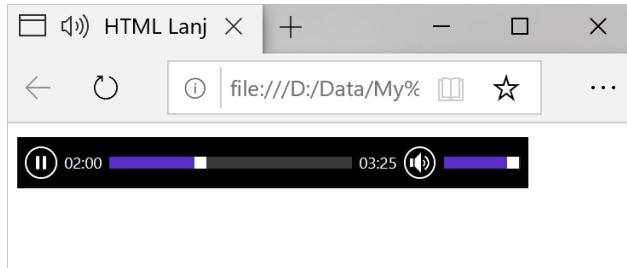
Untuk memperdengarkan audio pada halaman web dapat dilakukan dengan menggunakan tag <audio> dengan cara yang sama seperti cara menggunakan tag <video>.

Berikut adalah contoh kode penggunaan tag ini. Hasilnya dapat dilihat pada Gambar 60.

```
media 3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>HTML Lanjut - Media</title>
  </head>

  <body>
    <audio controls>
      <source src="multimedia/audio.mp3" type="audio/mpeg">
```

```
Your browser does not support the video tag.  
</audio>  
</body>  
</html>
```



Gambar 60. Menampilkan audio player.

Jika audio player tidak ingin ditampilkan pada halaman web namun secara otomatis audio diputar maka dapat menggunakan kode berikut ini.

```
media_4.html  
<!DOCTYPE HTML>  
<html>  
    <head>  
        <title>HTML Lanjut - Media</title>  
    </head>  
  
    <body>  
        <audio autoplay>  
            <source src="multimedia/audio.mp3" type="audio/mpeg">  
            Your browser does not support the video tag.  
        </audio>  
    </body>  
</html>
```

Tag lain untuk menampilkan media adalah <object>. Tag ini dapat digunakan untuk menampilkan gambar, flash player atau file PDF. Berikut adalah contoh penggunaan tag ini untuk menampilkan file PDF.

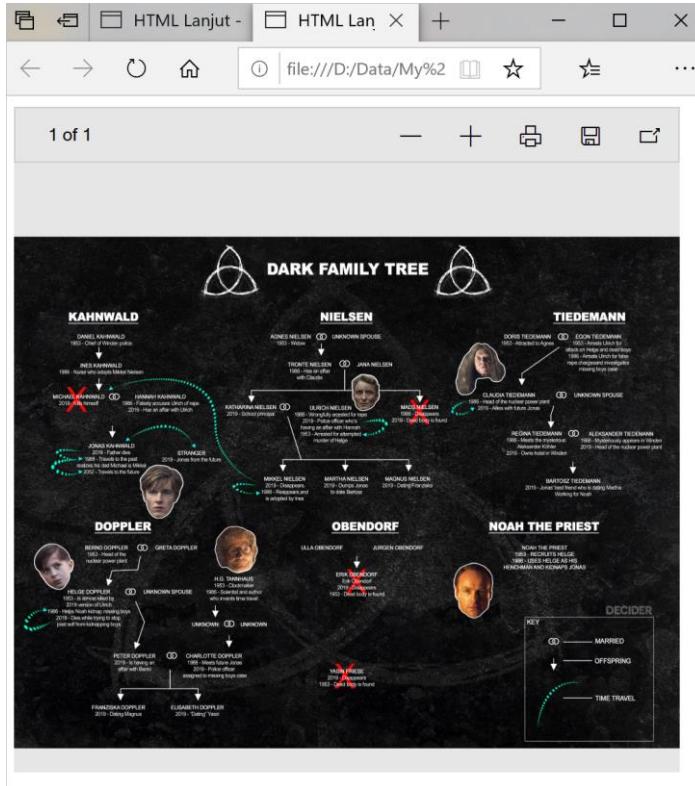
```
media_5.html  
<!DOCTYPE HTML>  
<html>  
    <head>  
        <title>HTML Lanjut - Media</title>  
    </head>  
  
    <body>
```

```

<object data="multimedia/object.pdf"
        width="500px" height="500px">
</body>
</html>

```

Pada contoh kode di atas, sumber file yang ditampilkan dapat ditentukan dengan mengisi atribut data. Hasilnya dapat dilihat pada Gambar 61.



Gambar 61. File pdf pada halaman web.

Tag yang terakhir adalah <embed>. Selain menampilkan file multimedia, tag ini juga dapat menampilkan halaman web. Berikut adalah contoh penggunaan tag ini.

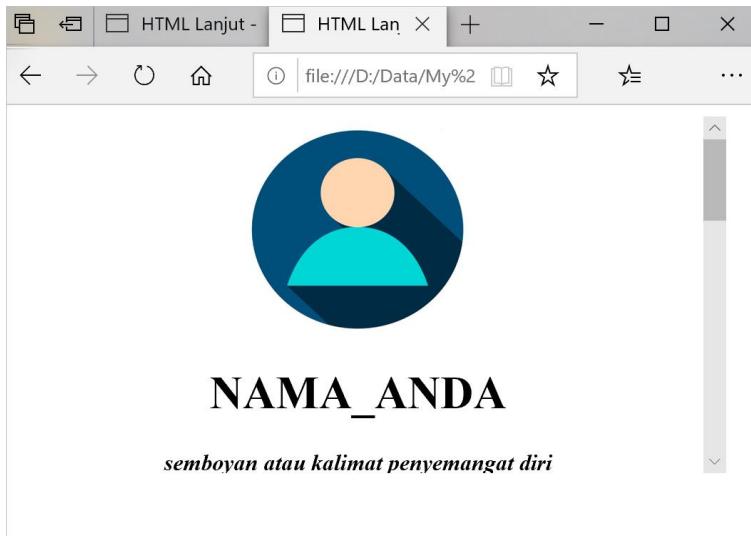
```

media_6.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>HTML Lanjut - Media</title>
  </head>

```

```
<body>
    <embed src="profile.html" width="500px" height="250px">
</body>
</html>
```

Hasilnya dapat dilihat pada Gambar 62.



Gambar 62. Tag embed untuk menampilkan halaman web.

Grafik

Tag `<canvas>` adalah tempat untuk menggambar secara on-the-fly. Namun untuk menggambar pada canvas diperlukan pengetahuan pemrograman client side dengan menggunakan JavaScript. Canvas umum digunakan untuk membuat game pada antarmuka yang menggunakan HTML. Berikut adalah contoh kode penggunaan tag `<canvas>`.

```
media_canvas_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Media</title>
    </head>

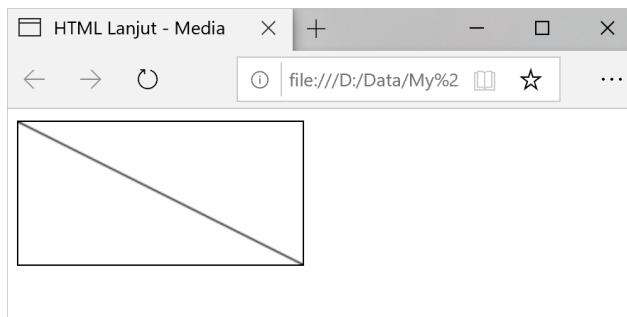
    <body>
        <canvas id="myCanvas" width="200" height="100"
                style="border:1px solid #000000;">
        </canvas>
    </body>
</html>
```

```

<script>
    var c = document.getElementById("myCanvas");
    var ctx = c.getContext("2d");
    ctx.moveTo(0, 0);
    ctx.lineTo(200, 100);
    ctx.stroke();
</script>
</body>
</html>

```

Hasilnya dapat dilihat pada gambar berikut ini.



Gambar 63. Canvas.

Contoh pembuatan game sederhana dengan cara ini dapat dilihat pada https://www.w3schools.com/graphics/game_intro.asp. Sedangkan salah satu game engine atau framework yang menggunakan HTML5 adalah <https://phaser.io/> (Gambar 64).



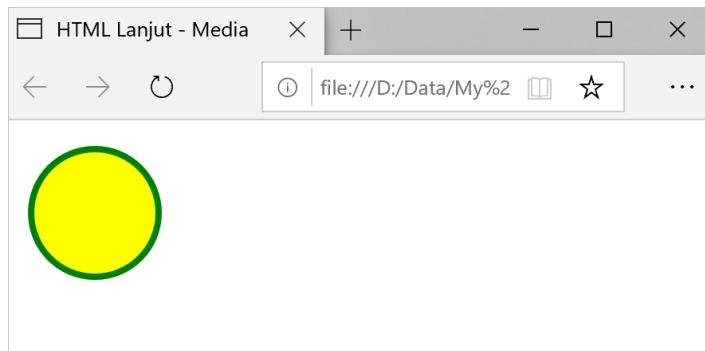
Gambar 64. Phaser - HTML 5 game framework.

Tag lain untuk menggambar adalah <svg>. SVG adalah singkatan dari Scalable Vector Graphics. Contoh penggunaan tag ini dapat dilihat pada kode di bawah ini.

```
media_svg_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>HTML Lanjut - Media</title>
    </head>

    <body>
        <svg width="100" height="100">
            <circle cx="50" cy="50" r="40"
                    stroke="green" stroke-width="4"
                    fill="yellow" />
        </svg>
    </body>
</html>
```

Hasil dari kode di atas dapat dilihat pada.



Gambar 65. SVG pada halaman web.

Untuk mengenal lebih jauh tentang penerapan SVG pada halaman web dapat melihat tutorial pada https://www.w3schools.com/graphics/svg_intro.asp.

Cascading Style Sheets

Cascading Style Sheets (CSS) adalah bahasa yang digunakan memberikan style pada 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. Pengetahuan CSS adalah pengetahuan wajib dimiliki oleh web designer atau developer.

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

Pendahuluan

Sub bab ini berisi sintaks dan tata cara penggunaan CSS 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 66. Contoh CSS untuk style teks.

Tata Cara Penggunaan

Penggunaan CSS dapat dilakukan dengan beberapa cara yaitu:

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

Inline CSS

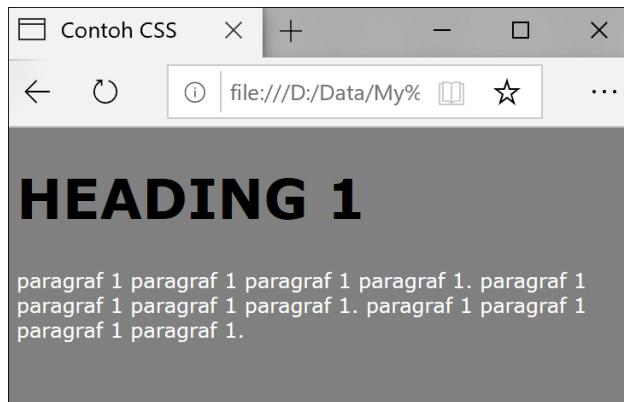
Cara ini dituliskan langsung pada sebuah tag HTML dengan memberikan property dan value CSS di dalam atribut style pada tag HTML tersebut. Sintaks dari cara ini dapat dilihat di bawah ini.

```
<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 67. Contoh Inline CSS.

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

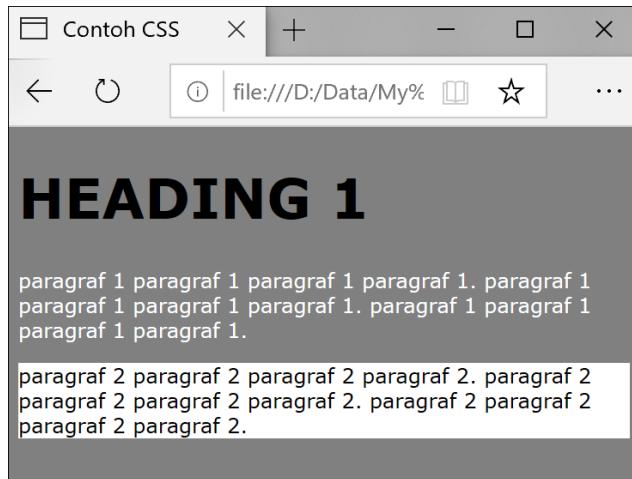
```
contoh_css_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Contoh CSS</title>
    </head>

    <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 68. 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>

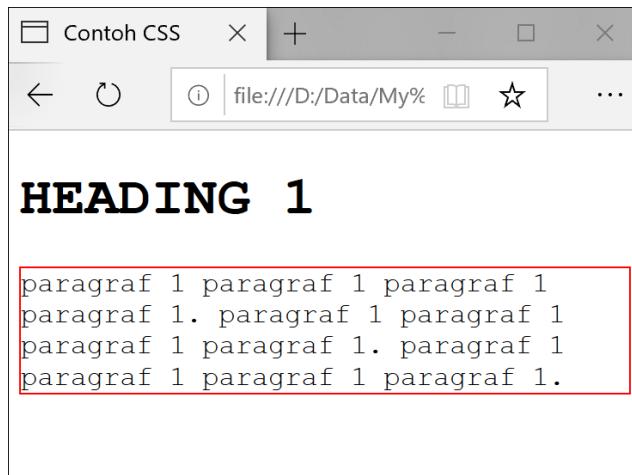
    <body style="font-family: 'Courier New', Courier, monospace;">
        <h1 style="text-transform: uppercase;">Heading 1</h1>
        <p style="font-size: 12px;">
            paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf 1 paragraf 1.
            paragraf 1 paragraf 1 paragraf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



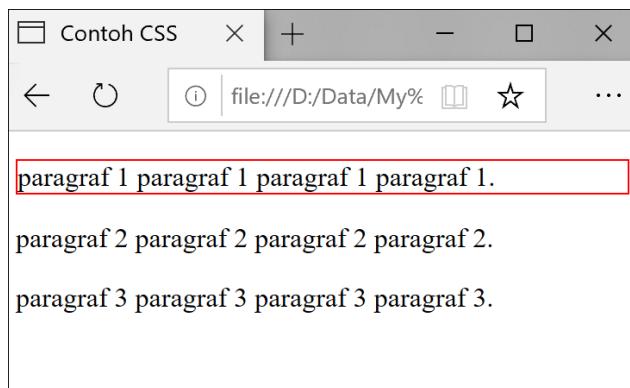
Gambar 69. Border.

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 70. 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.

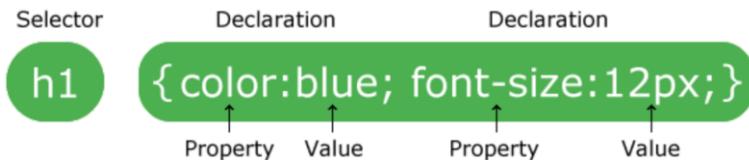
Penulisan CSS dengan cara ini dilakukan di dalam tag <head>, bukan di dalam tag <body> seperti yang dilakukan pada cara inline CSS. Untuk menulis CSS di dalam tag <head> diperlukan tag <style>. Berikut adalah sintaks dan posisi dari tag <style>.

```
<!DOCTYPE HTML>
<html>
  <head>
    <title>Internal CSS</title>
    <style>
      /* style CSS ditulis 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 71. Sintaks pada Internal CSS.

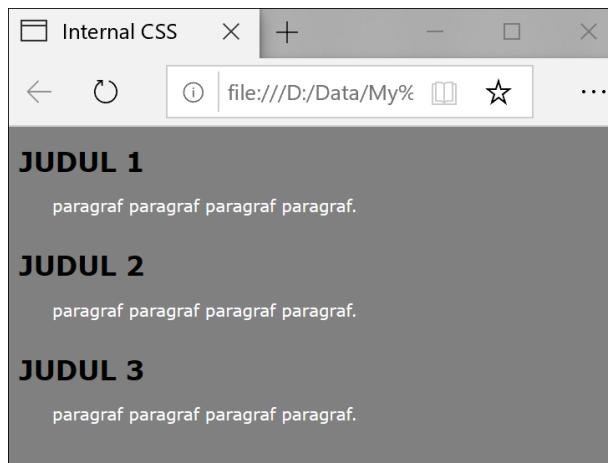
Selector adalah tag atau elemen HTML yang akan diberikan style. Kemudian style ditulis setelahnya di dalam tanda { dan }. Berikut adalah contoh penulisannya.

```
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 72. 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 73. 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.

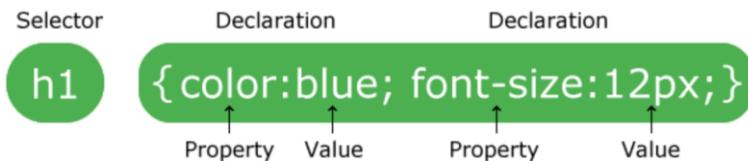
Dasar-Dasar CSS

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.

Selector

Pada sub bab Tata Cara Penggunaan telah dibahas sedikit tentang selector. Selector digunakan untuk “menemukan” atau memilih elemen atau tag HTML yang ingin diberikan style. Hal ini sangat penting untuk diketahui karena merupakan pengetahuan dasar dalam implementasi CSS pada halaman web.

Selector dipergunakan jika penulisan CSS menggunakan tata cara internal dan external CSS. Secara umum tata cara penulisan CSS dapat dilihat pada Gambar 74.



Gambar 74. Selector dan deklarasi CSS.

CSS selector dapat dibagi menjadi 5 kategori, yaitu:

1. Simple selector (selector sederhana) yang mana pemilihan elemen dengan menggunakan nama elemen atau tag, id atau class. Cara ini telah dijelaskan pada sub bab Tata Cara Penggunaan.
2. Combinator selector.
3. Pseudo-class selector.
4. Pseudo-elements selector.
5. Attribute selector.

Dari 5 kategori ini yang wajib diketahui oleh seorang web designer atau developer adalah simple dan combinator selector.

Simple Selector

Berikut adalah contoh selector sederhana yang umum digunakan memberikan style CSS pada halaman web.

```
selector_simple_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>CSS - Simple Selector</title>
    <style>
      p {
        border:solid 1px #000000;
        padding: 5px;
      }

      #p2 {
        background-color: red;
        color: white;
      }
    </style>
  </head>
  <body>
    <p>Hello World</p>
    <div id="p2">Hello World</div>
  </body>
</html>
```

```

.p3 {
    background-color: blue;
    color:white;
}
</style>
</head>

<body>
<p>Paragraf 1</p>
<p id="p2">Paragraf 2</p>
<p class="p3">Paragraf 3</p>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 75 berikut ini.



Gambar 75. Contoh simple selector.

Pada kode CSS yang berada pada tag <style> dapat dilihat selector pertama menggunakan nama elemen atau tag <p>, ini artinya style yang berada dalam tanda {} setelah selector itu akan diberikan pada seluruh tag <p> yang ada pada halaman web. Style yang diberikan adalah memberikan border warna hitam dan padding sehingga dapat dilihat seluruh tag <p> memiliki style ini.

Selector selanjutnya adalah pemilihan elemen yang akan diberikan style berdasarkan id. Jika selector dipilih berdasarkan id maka digunakan tanda # (pagar) kemudian diikuti nama id. Karena selector adalah #p2 maka yang dipilih adalah elemen yang memiliki atribut id="p2" saja. Sehingga dapat dilihat elemen p dengan id="p2" memiliki style dengan warna latar merah dan teks warna putih.

Selector terakhir adalah pemilihan elemen berdasarkan class dimana selector yang digunakan diawali dengan tanda . (titik) kemudian diikuti dengan nama class. Karena selector adalah .p3 maka yang dipilih adalah elemen yang

menggunakan atribut class="p3". Sehingga dapat dilihat elemen p dengan class="p3" memiliki style dengan warna latar merah dan teks warna putih.

Selector lain yang dapat digunakan adalah dengan menggunakan tanda *, selector ini akan memilih semua elemen yang ada pada halaman web.

Combinator Selector

Selector jenis ini merupakan kombinasi dari beberapa selector sederhana. Ada 4 cara melakukan kombinasi seperti yang dapat dilihat pada contoh kode berikut ini.

```
selector_combinator_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>CSS - Combinator Selector</title>
        <style>
            div p {
                border:solid 5px #000000;
                padding: 5px;
            }

            div > p {
                background-color: red;
                color:white;
            }

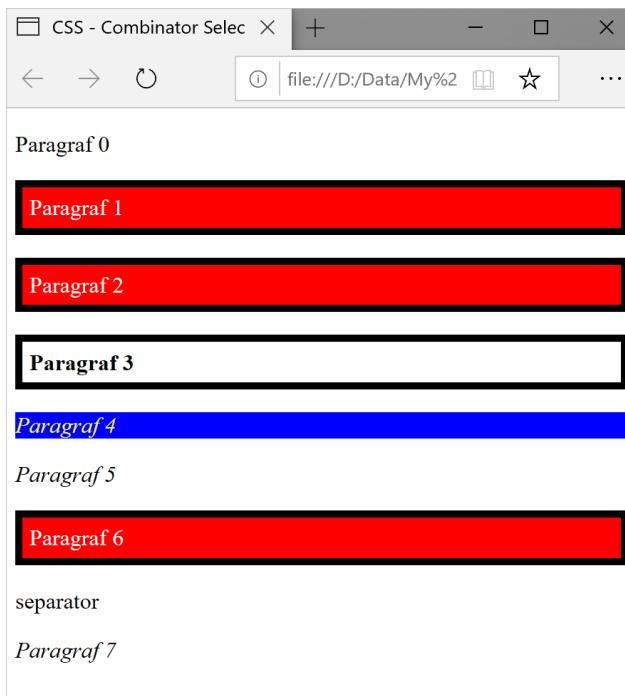
            div + p {
                background-color: blue;
                color:yellow;
            }

            div ~ p {
                font-style: italic;
            }
        </style>
    </head>

    <body>
        <p>Paragraf 0</p>
        <div>
            <p>Paragraf 1</p>
            <p id="p2">Paragraf 2</p>
            <b>
                <p class="p3">Paragraf 3</p>
            </b>
        </div>
    </body>
```

```
<p>Paragraf 4</p>
<p>Paragraf 5</p>
<div>
    <p>Paragraf 6</p>
</div>
<span>separator</span>
<p>Paragraf 7</p>
</body>
</html>
```

Hasilnya dapat dilihat pada berikut ini.



Gambar 76. Contoh combinator selector.

Berikut keterangan dari 4 tipe combinator selector:

Kombinasi element1 element2, kombinasi ini menggunakan spasi yang mempunyai arti bahwa element2 dipilih jika berada di dalam element1. Contohnya dapat dilihat pada potongan kode CSS berikut ini.

```
div p {
    border:solid 5px #000000;
    padding: 5px;
}
```

Kode CSS di atas mempunyai makna untuk memilih semua elemen <p> yang berada di dalam elemen <div>. Elemen yang dipilih akan diberi style dengan border tebal berwarna hitam. Element-element tersebut dapat dilihat pada:

- Paragraf 1.
- Paragraf 2.
- Paragraf 3, elemen ini berada didalam elemen namun berada di dalam elemen <div>.
- Paragraf 6.

Kombinasi element1 > element2, kombinasi ini menggunakan tanda > yang mempunyai arti bahwa element2 dipilih jika mempunyai induk element1, atau element2 yang dipilih adalah anak dari element1. Contohnya dapat dilihat pada potongan kode CSS berikut ini.

```
div > p {  
    background-color: red;  
    color:white;  
}
```

Kode CSS di atas mempunyai makna untuk memilih semua elemen <p> yang merupakan anak dari elemen <div>. Elemen yang dipilih akan diberi style dengan latarbelakang berwarna merah dengan teks warna putih, seperti yang dapat dilihat pada:

- Paragraf 1.
- Paragraf 2.
- Paragraf 6.

Ketiga element <p> di atas berada di dalam elemen <div>. Sedangkan elemen <p> dari paragraf 3 bukan merupakan anak dari elemen <div>, tetapi adalah anak dari elemen .

Kombinasi element1 + element2, kombinasi ini menggunakan tanda + yang mempunyai arti bahwa element2 yang dipilih adalah yang berada setelah element1. Contohnya dapat dilihat pada potongan kode CSS berikut ini.

```
div + p {  
    background-color: blue;  
    color:yellow;  
}
```

Kode CSS di atas mempunyai arti bahwa semua elemen <p> yang dipilih berada setelah elemen <div>. Elemen yang dipilih akan diberi style latarbelakang warna biru dengan teks warna kuning seperti yang dapat dilihat pada:

- Paragraf 4, dapat dilihat hanya elemen <p> ini yang berada disamping element <div>.

Kombinasi element1 ~ element2, kombinasi ini menggunakan tanda ~ yang mempunyai arti untuk memilih seluruh element2 yang berada setelah element1, namun element2 harus berada setara dengan element1. Contohnya dapat dilihat pada potongan kode CSS berikut ini.

```
div ~ p {  
    font-style: italic;  
}
```

Kode CSS di atas mempunyai arti bahwa semua elemen <p> yang dipilih adalah elemen yang ada setelah elemen <div> dan kedudukan atau posisinya setara dengan elemen <div>. Elemen yang dipilih akan diberi style cetak miring seperti yang dapat dilihat pada:

- Paragraf 4.
- Paragraf 5.
- Paragraf 7.

Elemen <p> dari Paragraf 6 tidak dipilih karena kedudukan elemen ini tidak setara dengan elemen <div> sebelumnya karena berada di dalam elemen <div> yang lain.

Pseudo-Class Selector

Pseudo-class adalah class yang tidak nyata, artinya class CSS yang tidak dibuat oleh pembuat halaman web. Pseudo-class merupakan state atau status spesial dari suatu elemen. Sebagai contoh elemen hyperlink <a> memiliki status visited ketika hyperlink tersebut telah diklik. Status yang lain adalah hover ketika cursor mouse mengarah ke hyperlink tersebut.

Berikut adalah contoh kode CSS untuk tipe selector ini.

```
selector_pseudo_class_1.html  
<!DOCTYPE HTML>  
<html>  
    <head>  
        <title>CSS - Pseudo-Class Selector</title>  
        <style>  
            /* unvisited link */  
            a:link {  
                color: #000000;  
            }
```

```

        /* visited link */
        a:visited {
            color: #00FF00;
        }

        /* mouse over link */
        a:hover {
            color: #FF0000;
        }

        /* selected link */
        a:active {
            color: #0000FF;
        }
    </style>
</head>

<body>
    <a href="#">Hyperlink</a>
</body>
</html>

```

Hasilnya dapat dilihat hyperlink berwarna hitam, jika cursor mouse diarahkan ke hyperlink tersebut maka akan berganti warna menjadi merah. Jika hyperlink diklik maka statusnya akan menjadi visited dan warna berganti menjadi hijau.

Pada prakteknya selector ini akan sering digunakan untuk memberikan style pada hyperlink atau menu. Berikut ini contoh yang lain penggunaan selector ini.

```

selector_pseudo_class_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>CSS - Pseudo-Class Selector</title>
        <style>
            p {
                display: none;
                background-color: yellow;
                padding: 20px;
            }

            div:hover p {
                display: block;
            }
        </style>
    </head>

```

```
<body>
    <div><b>Arahkan cursor mouse ke sini!</b>
        <p>Pesan tersembunyi</p>
    </div>
</body>
</html>
```

Kode di atas akan bekerja ketika cursor mouse diarahkan ke tulisan “Arahakan cursor mouse ke sini!” yang berada di dalam elemen `<div>`, hasilnya akan ditampilkan kotak berwarna kuning dengan tulisan “Pesan tersembunyi”.

Masih banyak lagi penggunaan selector ini yang dapat dilihat pada link berikut https://www.w3schools.com/css/css_pseudo_classes.asp.

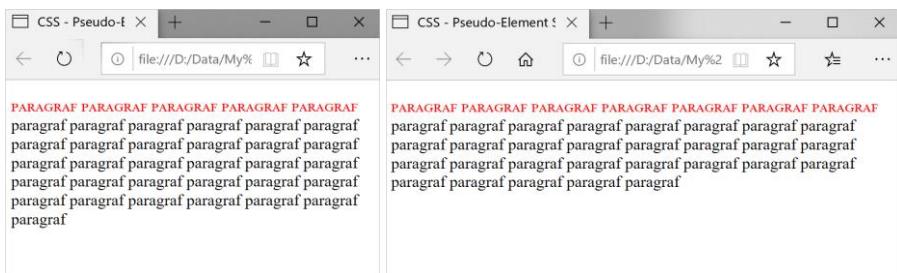
Pseudo-Elements Selector

Pseudo-element CSS digunakan untuk memberikan style pada bagian tertentu pada element. Untuk mengetahui definisi tersebut dapat melihat contoh berikut ini.

```
selector_pseudo_element_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>CSS - Pseudo-Element Selector</title>
        <style>
            p::first-line {
                color: #ff0000;
                font-variant: small-caps;
            }
        </style>
    </head>

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

Hasilnya dapat dilihat pada Gambar 77.



Gambar 77. Contoh pseudo-element selector.

Kode CSS yang digunakan memilih bagian tertentu dari elemen <p>, yaitu hanya baris pertama saja. Coba untuk mengubah-ubah ukuran window web browser, maka secara dinamis teks akan mengikuti ukuran window. Namun dengan menggunakan kode CSS berikut ini, sebagian elemen <p>, hanya baris pertama yang terlihat di window diberikan style.

```
p::first-line {  
    color: #ff0000;  
    font-variant: small-caps;  
}
```

Sintaks dari pseudo-element selector ini adalah sebagai berikut.

```
selector::pseudo-element {  
    property: value;  
}
```

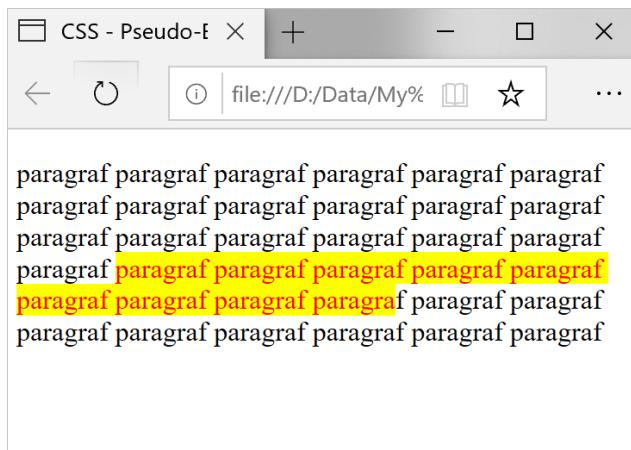
selector adalah seperti yang telah dijelaskan pada bagian simple selector dan combinator selector. Sedangkan pseudo-element yang dapat digunakan adalah sebagai berikut.

| Selector | Contoh | Keterangan |
|----------------|-----------------|--|
| ::after | p::after | Menyisipkan sesuai setelah elemen <p> |
| ::before | p::before | Menyisipkan sesuatu sebelum elemen <p> |
| ::first-letter | p::first-letter | Memilih huruf pertama pada setiap elemen <p> |
| ::first-line | p::first-line | Memilih baris pertama setiap elemen <p> |
| ::selection | p::selection | Memilih bagian dari elemen yang dipilih oleh user. |

Berikut adalah contoh penggunaan salah satu selector yaitu ::selection.

```
selector_pseudo_element_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>CSS - Pseudo-Element Selector</title>
        <style>
            p::selection {
                color: #ff0000;
                background-color: yellow;
            }
        </style>
    </head>

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



Gambar 78. Contoh selector ::selection.

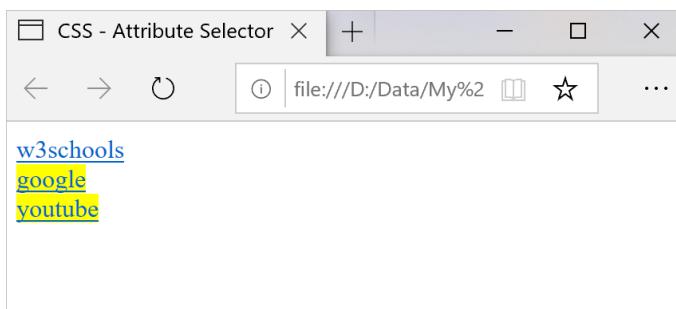
Attribute Selector

Penggunaan selector ini memungkinkan memilih elemen berdasarkan atribut atau nilai atribut tertentu. Untuk memilih elemen berdasarkan atribut tertentu dapat dilihat pada contoh berikut.

```
selector_attribute_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>CSS - Attribute Selector</title>
        <style>
            a[target] {
                background-color: yellow;
            }
        </style>
    </head>

    <body>
        <a href="https://www.w3schools.com">w3schools</a>
        <br>
        <a href="http://www.google.com" target="_blank">google</a>
        <br>
        <a href="http://www.youtube.org" target="_top">youtube</a>
    </body>
</html>
```

Kode CSS di atas memberi contoh untuk memilih elemen `<a>` yang memiliki atribut “target” saja. Sehingga dapat dilihat hanya hyperlink google dan youtube saja yang diberi style dengan latarbelakang warna kuning seperti terlihat pada.



Gambar 79. Contoh attribute selector.

Sedangkan untuk memilih elemen berdasarkan nilai suatu atribut dapat dilihat pada tabel berikut.

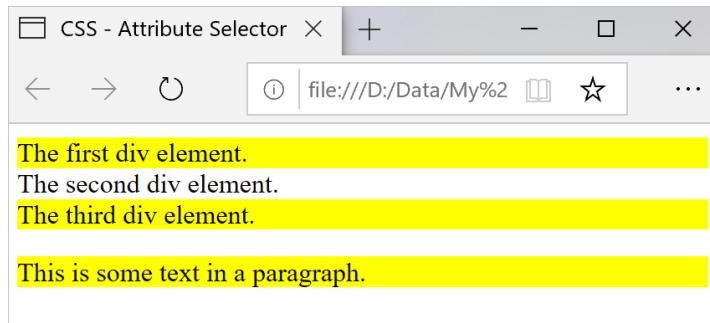
| Selector | Contoh | Keterangan |
|---------------------|---------------------|---|
| [attribute=value] | [target=_blank] | Memilih semua elemen yang memiliki atribut target="_blank" |
| [attribute~=value] | [title~=bunga] | Memilih semua elemen dengan atribut title yang ada kata "bunga". |
| [attribute =value] | [lang =id] | Memilih semua elemen dengan atribut lang yang nilainya dimulai dengan kata "id". |
| [attribute^=value] | a[href^="https"] | Memilih semua elemen <a> yang memiliki atribut href dengan nilai dimulai kata "https" |
| [attribute\$=value] | a[href\$=".pdf"] | Memilih semua elemen <a> yang memiliki atribut href dengan nilai diakhiri kata ".pdf" |
| [attribute*=value] | a[href*="ilkomulm"] | Memilih elemen <a> yang memiliki atribut href yang mengandung substring "ilkomulm" |

Berikut adalah contoh salah satu implementasi tadi selector yang telah disebutkan di atas.

```
selector attribute_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>CSS - Attribute Selector</title>
    <style>
      [class*="te"] {
        background: yellow;
      }
    </style>
  </head>

  <body>
    <div class="first_test">The first div element.</div>
    <div class="second">The second div element.</div>
    <div class="my-test">The third div element.</div>
    <p class="mytest">This is some text in a paragraph.</p>
  </body>
</html>
```

Kode di atas menggunakan selector [class*=te] yang berfungsi untuk memilih elemen yang memiliki atribut class yang nilainya mengandung substring "te". Hasilnya dapat dilihat pada gambar berikut ini.



Gambar 80. Contoh selector [attribute*=value].

Penjelasan detail dan contoh-contoh lain tentang attribute selector dapat dilihat pada link berikut ini https://www.w3schools.com/css/css_attribute_selectors.asp.

Komentar

Untuk membuat komentar atau menonaktifkan kode CSS dapat dilakukan dengan cara berikut ini.

```
komentar_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Komentar</title>
    <style>
      /*
        kode CSS di bawah ini membuat
        latar belakang abu-abu &
        teks warna putih
      */
      body {
        background-color: grey;
        color: #FFFFFF;
        font-family: Verdana, Geneva, Tahoma, sans-serif;
      }

      /* kode CSS di bawah ini tidak akan dieksekusi */
      /* p {
```

```

        font-weight: bold;
    } */

```

</head>

<body>

Hello World

<p>Saya sedang belajar CSS.</p>

</body>

</html>

Dari contoh di atas dapat dilihat cara untuk membuat komentar pada kode CSS.

Unit

Unit atau satuan digunanya di belakang angka untuk memberi nilai pada property seperti height, width, margin, padding, font-size dan lain-lain. Unit dapat dibagi menjadi 2 jenis yaitu:

- Mutlak (absolute), ukuran dengan satuan ini bersifat tetap. Satuan yang digunakan adalah:
 - o cm, centimeter.
 - o mm, milimeter.
 - o in, inci dimana 1in = 96px = 2.54cm.
 - o px, pixel dimana 1px = 1/96th dari 1in.
 - o pt, point dimana 1pt = 1/72 dari 1in.
 - o pc, picas dimana 1pc = 12pt.
- Relatif (relative), ukuran menghasilkan panjang atau tinggi yang berubah-ubah tergantung dari ukuran layar web browser. Elemen HTML dengan ukuran yang menggunakan unit ini akan menyesuaikan ukuran dengan ukuran window web browser. Ukuran saat window web browser penuh dengan ukuran saat window diresize menjadi setengahnya akan berbeda, tapi proporsi atau prosentasenya akan sama. Satuan yang digunakan untuk tujuan ini adalah:
 - o em, ukuran relatif dengan ukuran font-size elemen. Sebagai contoh 2em artinya ukuran 2 kali dari ukuran font semula).
 - o ex, ukuran relatif dengan ukuran x-height dari ukuran font semula.
 - o %, ukuran relatif dengan ukuran elemen induk.

Berikut adalah contoh sebagian satuan yang disebutkan di atas.

| |
|-------------|
| unit_1.thml |
|-------------|

```

<!DOCTYPE HTML>
<html>

```

```

<head>
    <title>Text</title>
    <style>
        body {
            font-size:16px;
        }

        div {
            font-size: 150%;
            border: 1px solid black;
        }
    </style>
</head>

<body>
    <p>Ukuran font-size halaman ini adalah 16px.</p>
    <div>Ukuran font-size elemen div adalah 150%.</div>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 81.



Gambar 81. Contoh satuan mutlak dan relatif.

Warna

Hampir semua elemen HTML dapat diberi warna. Warna dapat diberikan pada:

- Teks.
- Latarbelakang (background).
- Border atau sisi dari elemen. Umumnya elemen HTML mempunyai bentuk persegi empat, seperti elemen `<p>`, `<div>`, `<td>` dan lain-lain.

Sehingga warna border dapat diberikan untuk seluruh sisi atau sebagian sisi yang diinginkan saja.

Warna yang dapat digunakan dengan cara berikut ini:

- Nama warna, menggunakan nama warna seperti blue, black, grey dan lain-lain. CSS/HTML mendukung nama 140 warna standar yang didukung oleh semua web browser. Daftarnya dapat dilihat pada link berikut ini https://www.w3schools.com/colors/colors_names.asp.
- Hexadesimal, menentukan warna dengan menggunakan angka hexadesimal. Nilai terdiri atas 6 bilangan dimana 2 nilai pertama menyatakan nilai merah (Red), 2 nilai kedua adalah hijau (Green) dan 2 nilai terakhir adalah nilai biru (Blue). Untuk menggunakan angka hexadesimal harus diawali tanda #. Sebagai contoh untuk warna hitam digunakan hex code #000000 dan untuk warna putih digunakan hex code #FFFFFF.
- RGB, singkatan dari Red, Green dan Blue. Warna ditentukan dengan menentukan nilai untuk masing-masing komponen tersebut. Setiap komponen bernilai dalam rentang 0 - 255.
- HSL, singkatan dari Hue, Saturation, dan Lightness. Hue adalah derajat warna dari roda warna yang memiliki rentang nilai antara 0 - 360. Sebagai contoh 0 adalah merah, 120 adalah hijau. Saturation adalah kepekatan warna yang mempunyai nilai dalam persen Lightness memiliki nilai dalam persen.

Dari keempat cara di atas, yang umum digunakan adalah nama warna dan hexadesimal. Pembaca dianjurkan untuk mempelajari, mengingat dan implementasikan kedua cara tersebut saat membuat halaman web.

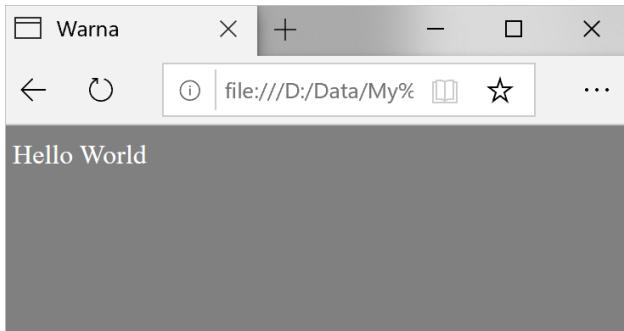
Teks umumnya berada di dalam suatu elemen HTML seperti <p>, <div> dan lain-lain. Sehingga pemberian warna untuk teks dilakukan dengan memberikan style pada elemen HTML tempat teks berada. Property yang digunakan untuk memberi warna pada teks adalah color. Sedangkan untuk memberikan warna latar pada elemen tersebut digunakan property background-color.

```
warna 1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Warna</title>
    <style>
      body {
        background-color: grey;
        color: #FFFFFF;
      }
    </style>
  </head>
  <body>
    <h1>Selamat Datang di Webku</h1>
    <p>Ini adalah contoh penggunaan warna pada teks dan latar.</p>
  </body>
</html>
```

```
</style>
</head>

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

Pada kode di atas dicontohkan pemberikan warna dengan cara memberikan nama warna dan hexadesimal. Hasilnya dapat dilihat pada Gambar 82.



Gambar 82. Contoh CSS warna pada teks dan latarbelakang.

Telah disebutkan sebelumnya bahwa hampir setiap elemen HTML berbentuk persegi empat sehingga memiliki border. Untuk memberikan warna pada elemen HTML dapat digunakan property:

- border-color.
- border-top.
- border-left.
- border-right.
- border-bottom.
- border, untuk membuat border pada keempat sisi.
- Berikut contoh penggunaan beberapa property dari yang telah disebutkan di atas. Hasilnya dapat dilihat pada Gambar 83.

```
warna_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Warna</title>
        <style>
            body {
                background-color: grey;
                color:#FFFFFF;
            }
        </style>
    </head>
    <body>
        Hello World
    </body>
</html>
```

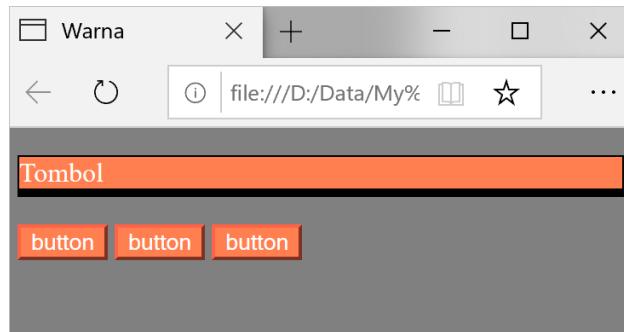
```

p {
    background-color: coral;
    border: solid 1px #000000;
    border-bottom: solid 5px #000000;
}

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

<body>
<p>Tombol</p>
<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>

```



Gambar 83. Contoh CSS warna pada border.

Contoh penggunaan warna dengan cara RGB dan HSL dapat dilihat pada contoh di bawah ini.

```

warna_3.html
<!DOCTYPE HTML>
<html>
<head>
    <title>Warna</title>
    <style>
        p {
            margin: 5px;
            padding: 5px;
            color: white;

```

```

        }

    .warna {
        background-color: tomato;
    }

    .hexadesimal {
        background-color: #ff6347;
    }

    .rgb {
        background-color: rgb(255, 99, 71);
    }

    .hsl {
        background-color: hsl(9, 100%, 64%);
    }

```

</style>

</head>

<body>

<p class="warna">Nama Warna</p>

<p class="hexadesimal">Hexadesimal</p>

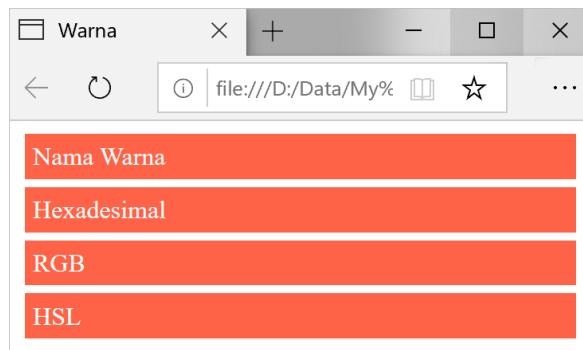
<p class="rgb">RGB</p>

<p class="hsl">HSL</p>

</body>

</html>

Pada contoh di atas nilai warna RGB ditentukan dengan menggunakan fungsi `rgb(nilai_red, nilai_green, nilai_blue)`. Sedangkan untuk nilai warna HSL ditentukan dengan fungsi `hsl(nilai_warna, prosentase_saturation, prosentase_lightness)`. Hasilnya dapat dilihat pada gambar di bawah ini.



Gambar 84. Contoh warna dengan RGL & HSL.

Box Model

Hampir semua elemen HTML adalah kotak. Hal ini dapat dilihat dari penjelasan dan contoh dari pembahasan pada sub bab sebelumnya yaitu Display. Box model CSS secara visual dapat dijelaskan pada Gambar 85.



Gambar 85. Box model CSS.

Sebuah elemen HTML berbentuk kotak seperti terlihat pada gambar di atas. sebuah kotak memiliki border, dan di dalam kotak terdapat isi (content) baik berupa teks, gambar, elemen HTML atau yang lain.

Secara default border pada kotak tidak terlihat tetapi ada. Untuk melihat border tersebut dapat dilakukan dengan dengan kode CSS berikut ini.

```
box model 1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Box Model</title>
    <style>
      div, span {
        border: solid 1px #000000;
      }
    </style>
  </head>

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

```

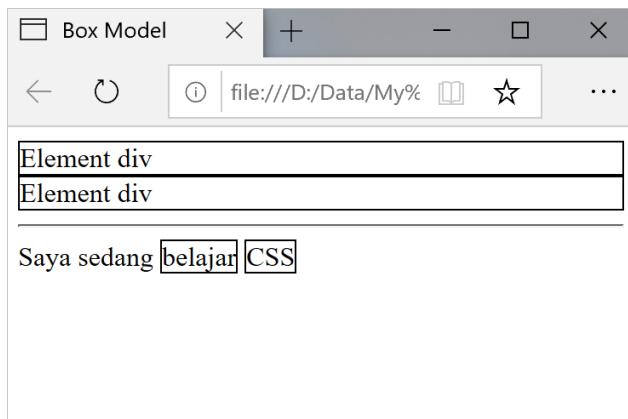
<div>
    Element div
</div>
<hr>
Saya sedang <span>belajar</span> <span>CSS</span>
</body>
</html>

```

Pada contoh di atas digunakan property border dengan nilai adalah:

- Style border yang dapat berisi nilai solid, dotted, dashed, none dan lain-lain.
- Ukuran tebal dari border.
- Warna border.

Hasilnya dapat dilihat pada Gambar 86.



Gambar 86. Contoh border 1.

Contoh di atas adalah membuat border untuk seluruh sisi kotak. Sedangkan untuk membuat border per sisi dapat digunakan cara seperti pada contoh kode berikut ini.

```

box_model_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Box Model</title>
        <style>
            div, span {
                border-top: solid 2px red;
                border-left: dashed 1px red;
                border-bottom: solid 2px black;
                border-right: dashed 1px red;
            }
        </style>
    </head>
    <body>
        <div>
            Saya sedang <span>belajar</span> <span>CSS</span>
        </div>
    </body>
</html>

```

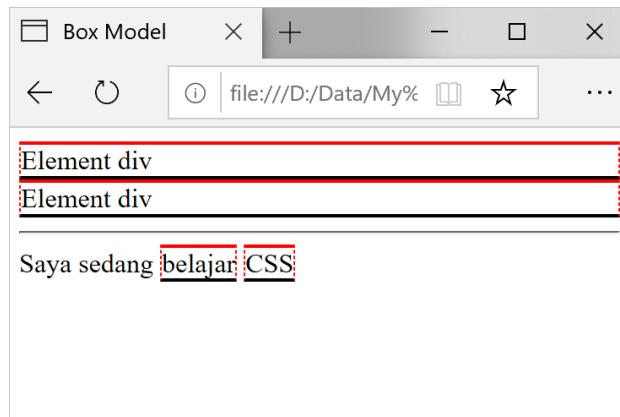
```

        }
    </style>
</head>

<body>
    <div>
        Element div
    </div>
    <div>
        Element div
    </div>
    <hr>
    Saya sedang <span>belajar</span> <span>CSS</span>
</body>
</html>

```

Dan hasilnya dapat dilihat pada Gambar 87.



Gambar 87. Contoh border 2.

Pada Gambar 85 terlihat diantara border dan isi (content) terdapat padding yang merupakan jarak antara keduanya. Untuk mengatur jarak ini dapat dilakukan dengan cara yang sama dengan yang telah dilakukan untuk mengatur border. Berikut adalah contoh kode yang dapat digunakan.

```

box_model_3.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Box Model</title>
        <style>
            .div1 {
                border: solid 1px #000000;
                padding: 5px;

```

```
}

.div2 {
    border: solid 1px #000000;
    padding: 10px 30px 5px 30px;
}
</style>
</head>

<body>
    <div class="div1">
        Element div
    </div>
    <div class="div2">
        Element div
    </div>
</body>
</html>
```

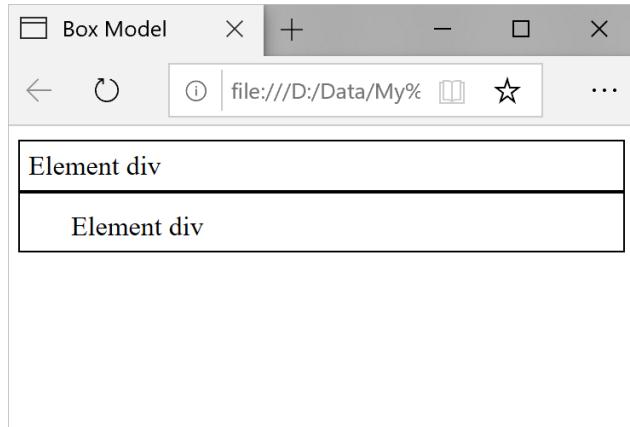
Pada kode di atas terdapat dua cara menentukan nilai property padding. Cara pertama adalah dengan memberikan sebuah nilai untuk keempat sisi area padding seperti terlihat pada potongan kode CSS berikut.

```
padding: 5px;
```

Sedangkan cara kedua adalah dengan memberikan nilai empat nilai yang setiap nilainya untuk setiap sisi padding seperti dapat dilihat pada potongan kode CSS berikut.

```
padding: 10px 30px 5px 30px;
```

Pada potongan kode ini dapat dilihat terdapat empat nilai, nilai yang pertama adalah nilai untuk sisi atas, nilai kedua untuk sisi kanan, nilai ketiga untuk sisi bawah dan nilai keempat untuk sisi kiri. Hasilnya dapat dilihat pada Gambar 88, bahwa nilai padding ini memberikan jarak antara teks dengan border.



Gambar 88. Contoh property padding.

Kembali ke Gambar 85, dapat dilihat di luar border terdapat property margin yang dapat digunakan untuk memberi jarak antara elemen ini dengan elemen lainnya. Berikut adalah contoh kode untuk memberi nilai property margin.

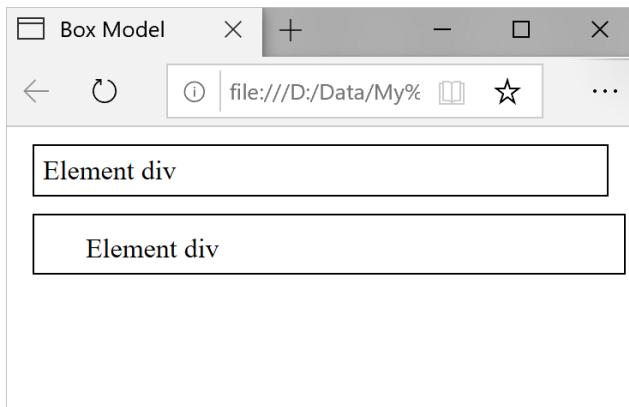
```
box model 4.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Box Model</title>
    <style>
      .div1 {
        border: solid 1px #000000;
        padding: 5px;
        margin: 10px;
      }

      .div2 {
        border: solid 1px #000000;
        padding: 10px 30px 5px 30px;
        margin: 0px 0px 0px 10px;
      }
    </style>
  </head>

  <body>
    <div class="div1">
      Element div
    </div>
    <div class="div2">
```

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

Cara memberi nilai pada property margin memiliki kesamaan dengan cara memberi nilai property pada padding. Pada Gambar 89 dapat dilihat dengan property margin dapat memberi jarak antara elemen <div>.



Gambar 89. Contoh property margin.

Teks

Teks adalah salah satu nilai yang sering diberikan di dalam elemen HTML. Teks juga merupakan konten utama pada halaman web seperti media social atau web berita online. Sehingga perlu dilakukan pengaturan format teks agar informasi mudah dibaca dan terlihat bagus.

Berikut ini adalah property-property CSS yang umum digunakan untuk memformat teks:

- text-align, digunakan untuk menentukan alignment secara horizontal dari teks. Umumnya nilai yang digunakan adalah left agar teks rata ke sisi kiri, right untuk kanan, center untuk tengah, justify agar teks rata pada sisi kiri dan kanan. Selain itu juga terdapat nilai lain yang dapat digunakan untuk mengisi property ini.
- vertical-align, digunakan untuk menentukan alighment secara vrtial dengan nilai yang dapat digunakan adalah top, bottom, middle dan lain-lain.
- text-decoration, digunakan untuk memberikan dan menghilangkan dekorasi pada teks. Jika nilai property ini adalah none maka dekorasi

akan dihilangkan. Nilai yang dapat digunakan untuk memberikan dekorasi adalah overline untuk garis atas teks, underline untuk garis bawah teks, dan lain-lain.

- text-transform, digunakan untuk mengubah teks yang ada menjadi huruf kecil semua dengan nilai lowercase, huruf besar semua dengan nilai uppercase, huruf besar pada awal kalimat dengan nilai capitalize dan lain-lain.
- text-indent, digunakan untuk menentukan berapa dalam teks akan masuk pada awal alinea. Nilai yang digunakan untuk property ini dapat berupa angka dengan satuan px (pixel), cm (centimeter), % (persen) dan lain-lain. Sebagai contoh 1cm yang artinya adalah teks akan di awal alinea akan masuk sedalam 1 cm.
- letter-spacing, digunakan untuk menentukan ukuran antar karakter.
- word-spacing, digunakan untuk menentukan ukuran antar kata.
- font-family, digunakan untuk menentukan jenis font yang digunakan pada teks. Untuk mengetahui jenis font apasaja yang dapat digunakan akan lebih mudah jika menggunakan text editor yang mempunyai fitur auto complete sehingga dapat dengan mudah memilih jenis font yang tersedia.
- font-style, digunakan untuk membuat teks menjadi cetak miring. Nilai yang dapat digunakan untuk mengisi property ini adalah normal, italic dan oblique.
- font-weight, digunakan untuk memberi bobot pada teks. Umumnya property ini digunakan untuk membuat teks terlihat tebal. Contoh nilai untuk property ini adalah normal dan bold.
- font-size, digunakan untuk mengubah ukuran teks. Nilai yang dapat diberikan untuk property ini dapat berupa nilai mutlak atau relatif seperti yang dijelaskan pada sub bab Unit.
- Dan masih banyak lagi property yang dapat digunakan untuk memberi style pada teks.

Berikut adalah contoh penggunaan beberapa property CSS di atas.

```
text_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Text</title>
    <style>
      div {
        border: solid 1px #000000;
        min-height: 30px;
        margin: 3px;
      }
    </style>
  </head>
  <body>
    <div>Text</div>
  </body>
</html>
```

```

        .text-left {
            text-align: left;
            text-decoration: underline;
        }

        .text-right {
            text-align: right;
            text-transform: capitalize;
        }

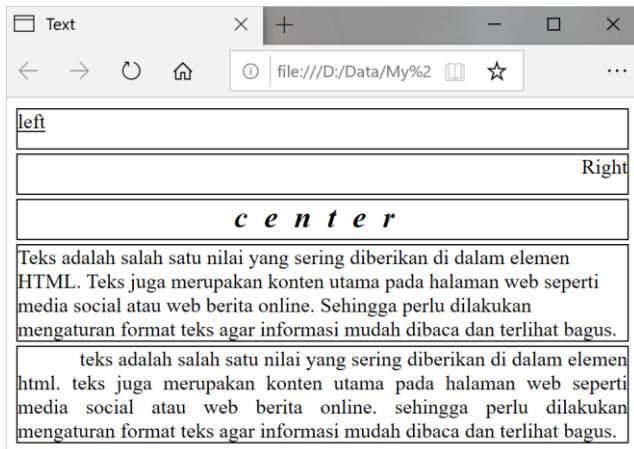
        .text-center {
            text-align: center;
            letter-spacing: 13px;
            font-size: 23px;
            font-weight: bold;
            font-style: italic;
        }

        .text-justify {
            text-align: justify;
            text-transform: lowercase;
            text-indent: 10%;
        }
    </style>
</head>

<body>
    <div class="text-left">left</div>
    <div class="text-right">right</div>
    <div class="text-center">center</div>
    <div>
        Teks adalah salah satu nilai yang sering diberikan di dalam elemen HTML. Teks juga merupakan konten utama pada halaman web seperti media social atau web berita online. Sehingga perlu dilakukan pengaturan format teks agar informasi mudah dibaca dan terlihat bagus.
    </div>
    <div class="text-justify">
        Teks adalah salah satu nilai yang sering diberikan di dalam elemen HTML. Teks juga merupakan konten utama pada halaman web seperti media social atau web berita online. Sehingga perlu dilakukan pengaturan format teks agar informasi mudah dibaca dan terlihat bagus.
    </div>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 90.



Gambar 90. Contoh property CSS untuk memformat 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 91. 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 92. 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 |

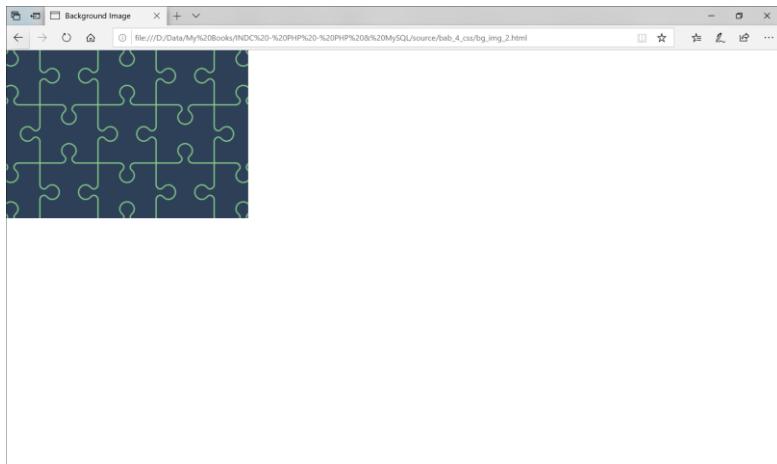
Gambar 92. Lokasi penyimpanan gambar dan file default.css.

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

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

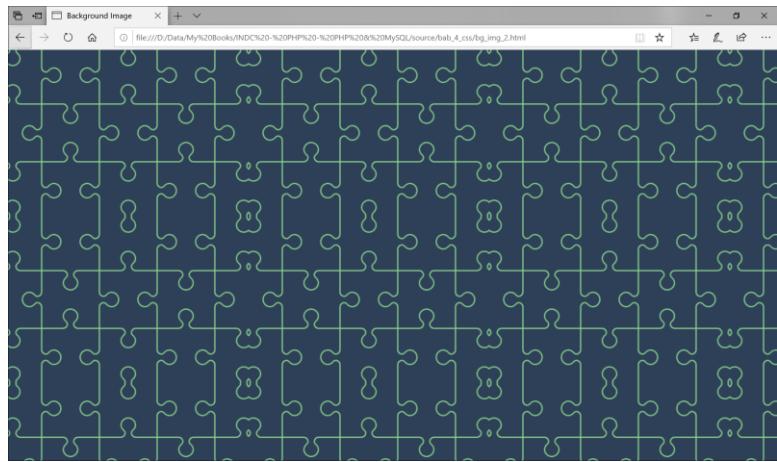
Kode CSS tersebut mempunyai fungsi yang sama dengan dua baris property CSS yang digunakan sebelumnya.

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 93. Membuat background image dengan gambar ukuran kecil.

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



Gambar 94. 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.

CSS Layout

Pada sub bab ini diberikan property-property yang berguna sebagai dasar untuk membuat layout atau antarmuka pada halaman HTML.

Display

Display adalah property CSS yang memiliki peran penting untuk mengontrol antarmuka atau layout. Setiap elemen HTML memiliki nilai property display default yang tergantung dari tipe elemen tersebut. Nilai default tersebut adalah:

- block.
- inline.

Berikut ini akan diperlihatkan contoh kode untuk memperlihatkan perbedaan antara elemen dengan nilai property display block dan inline.

Elemen HTML yang memiliki nilai property display adalah block diantaranya adalah:

- <div>
- <h1> - <h6>
- <p>
- <form>
- Dan lain-lain.

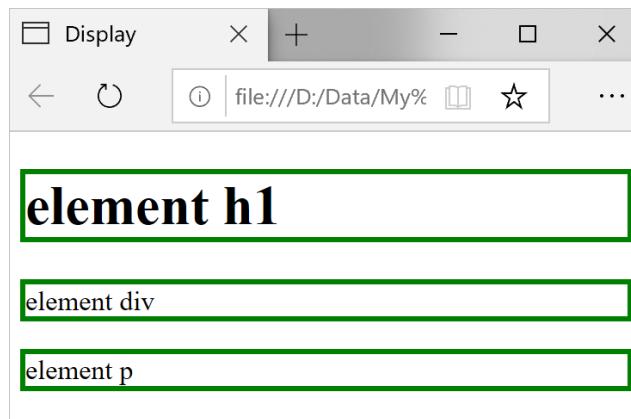
Berikut adalah contoh tampilan dari elemen dengan property display adalah block.

```
display_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Display</title>
    <style>
      h1, div, p {
        border: solid 3px green;
      }
    </style>
  </head>
  <body>
    <h1>Hello World!</h1>
    <div>This is a div element.</div>
    <p>This is a paragraph element.</p>
  </body>
</html>
```

```
</style>
</head>

<body>
    <h1>element h1</h1>
    <div>element div</div>
    <p>element p</p>
</body>
</html>
```

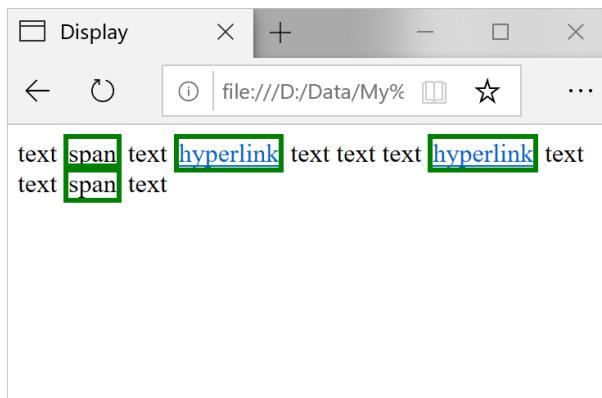
Hasilnya dapat dilihat pada Gambar 95.



```
</head>

<body>
    text <span>span</span>
    text <a href="#">hyperlink</a> text text
    text <a href="#">hyperlink</a> text
    text <span>span</span> text
</body>
</html>
```

Hasilnya dapat dilihat pada Gambar 96.



Gambar 96. Property display = inline.

Dari kedua contoh di atas dapat dilihat perbedaannya keduanya, yaitu:

- block, elemen ini terlihat lebar kotak menghabiskan 100% dari lebar halaman web.
- inline, elemen ini terlihat lebar kotak hanya sesuai ukuran isi elemen tersebut saja.

Dengan CSS dapat property display dari suatu elemen dapat dimodifikasi dengan contoh sebagai berikut.

```
display_3.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Display</title>
        <style>
            h1, div, p {
                border: solid 3px green;
                display: inline;
            }
        </style>
    </head>
    <body>
        text <span>span</span>
        text <a href="#">hyperlink</a> text text
        text <a href="#">hyperlink</a> text
        text <span>span</span> text
    </body>
</html>
```

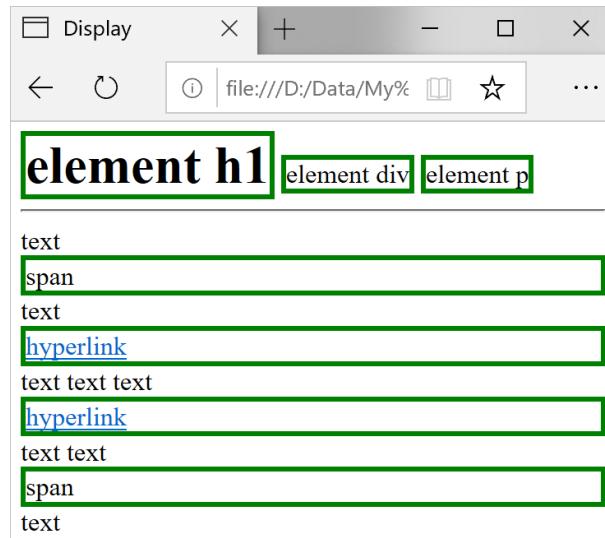
```

        span, a {
            border: solid 3px green;
            display: block;
        }
    </style>
</head>

<body>
    <h1>element h1</h1>
    <div>element div</div>
    <p>element p</p>
    <hr>
    text <span>span</span>
    text <a href="#">hyperlink</a> text text
    text <a href="#">hyperlink</a> text
    text <span>span</span> text
</body>
</html>

```

Pada contoh di atas elemen `<h1>`, `<div>` dan `<p>` yang secara default memiliki property display adalah block diganti nilainya menjadi inline. Begitu juga sebaliknya untuk elemen `` dan `<a>`. Hasilnya dapat dilihat pada Gambar 97.



Gambar 97. Modifikasi property display.

Cara yang dicontohkan pada kode di atas dapat digunakan untuk membuat menu vertikal atau horisontal dengan memanfaatkan elemen `` dan ``. Hal

ini akan diberikan contohnya pada sub bab selanjutnya mengenai pengaturan tata letak.

Lebar Elemen

Elemen HTML memiliki property untuk menentukan lebar yaitu property width. Nilai property ini dapat diberikan dengan nilai mutlak (absolute) atau relatif seperti yang telah dijelaskan pada sub bab Dasar Dasar CSS > Unit.

Property lain yang berhubungan dengan penentuan lebar dan sering digunakan adalah max-width. Property ini menentukan lebar maksimal dari suatu elemen. Jika ditentukan nilai untuk property ini adalah 500px, maka lebar suatu elemen tidak akan melebihi nilai itu. Jika lebar elemen induk atau window browser kurang dari 500px maka elemen yang menggunakan property ini akan mengikuti lebar elemen induk atau window browser tersebut. Selain itu juga terdapat property min-width yang mempunyai fungsi sebaliknya.

Berikut adalah contoh penggunaan property-property tersebut.

```
width_1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Width</title>
    <style>
      div {
        border: solid 1px #000000;
        padding: 5px;
        margin: 0 0 13px 0;
      }

      .div1 {
        width: 100px;
      }

      .div2 {
        width: 90%;
      }

      .div3 {
        max-width: 500px;
      }

      .div4 {
        width: 90%;
      }
    </style>
  </head>
  <body>
    <div>
      <div class="div1">Div 1</div>
      <div class="div2">Div 2</div>
      <div class="div3">Div 3</div>
      <div class="div4">Div 4</div>
    </div>
  </body>
</html>
```

```

        margin: auto;
    }

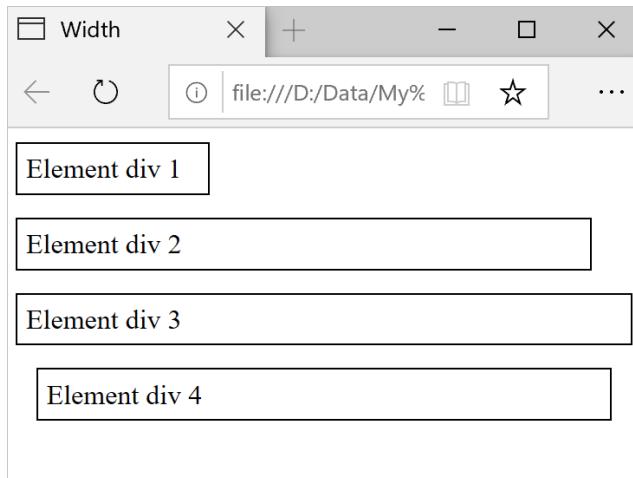
```

```

</style>
</head>

<body>
    <div class="div1">
        Element div 1
    </div>
    <div class="div2">
        Element div 2
    </div>
    <div class="div3">
        Element div 3
    </div>
    <div class="div4">
        Element div 4
    </div>
</body>
</html>
```

Hasilnya dapat dilihat pada Gambar 98, lebar window browser pada gambar ini dibuat kecil.



Gambar 98. Contoh property width pada window browser kecil.

Pada Gambar 99 adalah ketika lebar window browser diperlebar dengan ukuran lebih dari 500px. Pada elemen `<div>` 1 tidak ada perubahan ukuran karena menggunakan property width. Untuk elemen `<div>` 2 menggunakan ukuran dengan nilai relatif yaitu 90% sehingga ukurannya akan menyesuaikan

ukuran elemen induk atau window web browser. Elemen <div> 3 menggunakan property max-width, terlihat pada Gambar 98 lebar elemen ini mengikuti lebar window web browser yang lebarnya lebih kecil daripada lebar elemen <div> 3 ini. Namun ketika lebar window web browser diubah menjadi lebih besar dari 500px (Gambar 99) maka lebar elemen <div> 3 menjadi 500px.



Gambar 99. Contoh property width pada window browser dengan lebar > 500px.

Pada elemen <div> 4 menggunakan nilai lebar persen seperti element <div> 2, namun pada elemen <div> 4 digunakan property margin=auto yang membuat posisi elemen ini selalu berada di tengah (center). Cara ini umum digunakan untuk membuat tampilan isi (content) pada halaman web selalu berada di tengah walaupun ukuran windows web browser diubah-ubah.

Posisi

Property untuk mengatur posisi suatu elemen pada halaman adalah position. Nilai dari property ini adalah:

- static.
- relative.
- fixed.
- absolute.
- sticky.

Setiap elemen HTML memiliki property dengan nilai default adalah static. selain property position juga digunakan beberapa property pendukung untuk menentukan posisi yaitu:

- top, untuk menentukan jarak dari atas.
- bottom, untuk menentukan jarak dari bawah.

- left, untuk menentukan jarak dari kiri.
- right, untuk menentukan jarak dari kanan.

Property-property tersebut harus diberikan setelah property position digunakan. Untuk mengetahui bagaimana cara kerja dari setiap nilai tersebut dapat melihat contoh berikut ini.

```
position 1.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Position</title>
    <style>
      div {
        border: solid 1px #000000;
        padding: 5px;
        margin: 0 0 13px 0;
      }

      .div1 {
        width: 100px;
        position: static;
        left: 50px;
        top: 50px;
      }

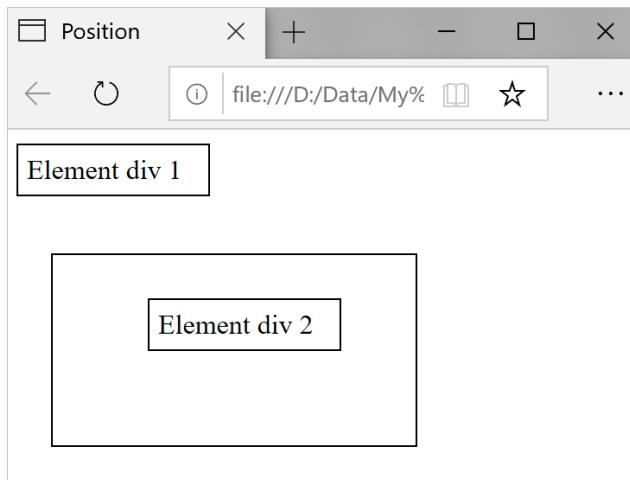
      .div2 {
        width: 200px;
        height: 100px;
        position: relative;
        left: 20px;
        top: 20px;
      }

      .div3 {
        width: 100px;
        position: relative;
        left: 50px;
        top: 20px;
      }
    </style>
  </head>

  <body>
    <div class="div1">
      Element div 1
    </div>
    <div class="div2">
```

```
<div class="div3">
    Element div 2
</div>
</body>
</html>
```

Kode di atas memperlihatkan perbedaan antara nilai property position adalah static dan relative. Perbedaan keduanya dapat dilihat pada Gambar 100. Untuk property position dengan nilai static posisinya tidak berubah walaupun diberikan nilai untuk property top dan left. Hal ini dapat diartikan posisi sesuai dengan sesuai keadaan normal dari halaman web.



Gambar 100. Contoh nilai static dan relative pada property position.

Property position dengan nilai relative mengijinkan elemen untuk diatur posisinya dengan titik acuan berdasarkan induk dari elemen tersebut. Pada element <div> dengan nilai property class="div2" dapat dilihat posisinya berjarak 20px dari elemen sebelumnya dan berjarak 20px dari sisi kiri windows web browser. Tetapi perhatikan elemen <div> dengan nilai property class="div3" yang berada di dalam elemen <div> dengan nilai property class="div2", titik acurannya bukan window web browser tetapi titik acuannya adalah elemen induknya yaitu elemen <div> dengan property class="div2". Oleh karena itu jika diberi property top="50px" maka berarti dihitung dari sisi atas elemen <div> dengan property class="div2". Sedangkan jika diberi property left="20px" maka jarak dihitung dari sisi kiri elemen <div> dengan property class="div2".

Selanjutnya adalah contoh perbedaan property position dengan nilai fixed dan absolute.

```
position_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Position</title>
        <style>
            div {
                border: solid 1px #000000;
                padding: 5px;
                margin: 5px
            }

            .parent {
                width: 90%;
                margin: auto;
                height: 100px;
                position: relative;
            }

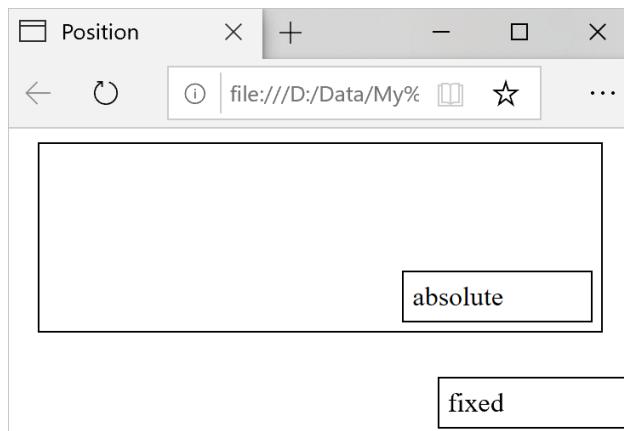
            .fixed {
                width: 100px;
                position: fixed;
                bottom: 0;
                right: 0;
            }

            .absolute {
                width: 100px;
                position: absolute;
                bottom: 0;
                right: 0;
            }
        </style>
    </head>

    <body>
        <div class="parent">
            <div class="fixed">
                fixed
            </div>
            <div class="absolute">
                absolute
            </div>
        </div>
    </body>
```

```
</html>
```

Hasil dari kode di atas dapat dilihat pada Gambar 101. Persamaan keduanya adalah posisi dapat ditentukan secara lebih bebas dengan menentukan nilai property top, bottom, left atau right. Perbedaannya adalah titik acuannya, jika menggunakan nilai absolute maka titik acuannya adalah elemen induknya. Sedangkan jika menggunakan nilai fixed maka titik acuannya adalah windows web broser.



Gambar 101. Contoh position dengan nilai fixed dan absolute.

Penggunaan property position dengan nilai fixed ini sering ditemui pada halaman web yang mempunyai fitur chat berupa tombol yang berada pada posisi kanan bawah halaman tersebut.

Selain property top, left, right, bottom juga dapat digunakan property z-index untuk mengatur posisi pada sumbu z yang dapat membuat elemen dapat saling tumpang tindih seperti lapisan atau layer. Nilai property z-index adalah bilangan bulat dimulai dari 0. Semakin besar nilai property yang dimiliki oleh sebuah elemen ini membuat elemen tersebut berada di atas elemen lain yang nilai z-index lebih kecil. Property ini dapat digunakan pada elemen yang menggunakan property position dengan nilai fixed atau absolute. Kode berikut adalah contoh dari implementasi property z-index.

```
position_3.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Position</title>
    <style>
```

```
div {  
    border: solid 1px #000000;  
    padding: 5px;  
    margin: 5px  
}  
  
.parent {  
    width: 90%;  
    margin: auto;  
    height: 100px;  
    position: relative;  
}  
  
.fixed1 {  
    width: 100px;  
    position: fixed;  
    bottom: 0;  
    right: 0;  
    background-color: blue;  
    color: white;  
}  
  
.fixed2 {  
    width: 100px;  
    position: fixed;  
    bottom: 13px;  
    right: 13px;  
    z-index: 1;  
    background-color: red;  
    color: white;  
}  
  
.absolute1 {  
    width: 100px;  
    position: absolute;  
    bottom: 0;  
    right: 0;  
    background-color: blue;  
    color: white;  
}  
  
.absolute2 {  
    width: 100px;  
    position: absolute;  
    bottom: 13px;  
    right: 13px;  
    z-index: 1;  
    background-color: red;
```

```

        color:white;
    }

```

```
</style>
```

```
</head>
```

```
<body>
```

```
    <div class="parent">
```

```
        <div class="fixed1">
```

```
            fixed 1
```

```
        </div>
```

```
        <div class="fixed2">
```

```
            fixed 2
```

```
        </div>
```

```
        <div class="absolute1">
```

```
            absolute 1
```

```
        </div>
```

```
        <div class="absolute2">
```

```
            absolute 2
```

```
        </div>
```

```
    </div>
```

```
</body>
```

```
</html>
```

Hasilnya dapat dilihat pada Gambar 102 dimana elemen terlihat saling tumpang tindih.



Gambar 102. Contoh penggunaan property z-index.

Nilai property position yang terakhir adalah sticky. Nilai ini membuat posisi berdasarkan posisi scroll dari user. Nilai ini membuat elemen dapat berprilaku seperti elemen yang menggunakan nilai position adalah fixed dan relative tergantung dari posisi scroll. Untuk mengetahui cara kerja nilai property ini dapat dilihat pada contoh berikut ini.

```
position_4.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Position</title>
        <style>
            #header {
                width: 90%;
                margin: auto;
                margin-bottom: 13px;
                padding: 5px;
                height: 50px;
                background-color: grey;
                color: white;
            }

            #menu {
                width: 90%;
                margin: auto;
                position: -webkit-sticky;
                position: sticky;
                top: 0;
                padding: 5px;
                background-color: #cae8ca;
                border: 2px solid #4CAF50;
            }

            #content {
                width: 90%;
                margin: auto;
            }
        </style>
    </head>

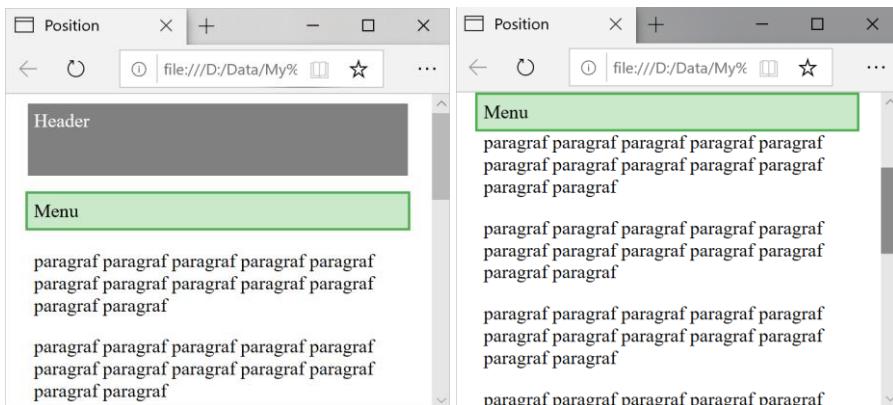
    <body>
        <div id="header">
            Header
        </div>
        <div id="menu">
            Menu
        </div>
    </body>
```

```

<div id="content">
    <!-- paragraf - start -->
    <p>
        paragraf paragraf paragraf
        paragraf paragraf paragraf
        paragraf paragraf paragraf
        paragraf paragraf paragraf
    </p>
    <!-- paragraf - end -->
    <!--
        copy paragraf start sampai end
        paste sebanyak sepuluh kali
        ke bawah.
    -->
</div>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 103. Elemen `<div>` yang berbentuk kotak warna hijau adalah elemen yang menggunakan property position dengan nilai sticky.



Gambar 103. Contoh position dengan nilai sticky.

Pada gambar sebelah kanan dapat dilihat posisi menu tersebut yang berada di bawah kotak header. Kemudian pada gambar di sebelah kanan, ketika halaman digulung ke bawah, kotak menu akan menempel di posisi yang telah ditentukan pada kasus ini adalah `top:0`. Dengan karakteristik seperti ini maka hal ini dapat digunakan untuk membuat menu yang banyak digunakan pada beberapa website.

Float

Property float berfungsi untuk menentukan bagaimana sebuah elemen melayang. Property ini umum digunakan untuk menata layout. Untuk mengetahui cara kerja property ini akan diberikan beberapa contoh penggunaan nilai-nilai yang berbeda untuk property ini.

Berikut ini adalah contoh property float dengan nilai left.

```
float_1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Float</title>
        <style>
            div {
                border: solid 1px black;
                width: 100px;
                height: 32px;
                margin: 5px;
            }

            .right {
                float: right;
            }
        </style>
    </head>

    <body>
        <div class="right ">
            div 1
        </div>
        <div class="right ">
            div 2
        </div>
        <div class="right ">
            div 3
        </div>
    </body>
</html>
```

Hasilnya dapat dilihat pada Gambar 104. Pada kode di atas dibuat tiga elemen <div> yang diberikan property float dengan nilai adalah right. Terlihat ketiga elemen melayang ke arah kanan dengan dimulai oleh elemen <div> 1, 2 kemudian 3.



Gambar 104. Contoh property float dengan nilai right.

Contoh selanjutnya adakan memberi nilai left untuk property float dengan kode sebagai berikut.

```
float_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Float</title>
    <style>
      div {
        border: solid 1px black;
        width: 100px;
        height: 32px;
        margin: 5px;
      }

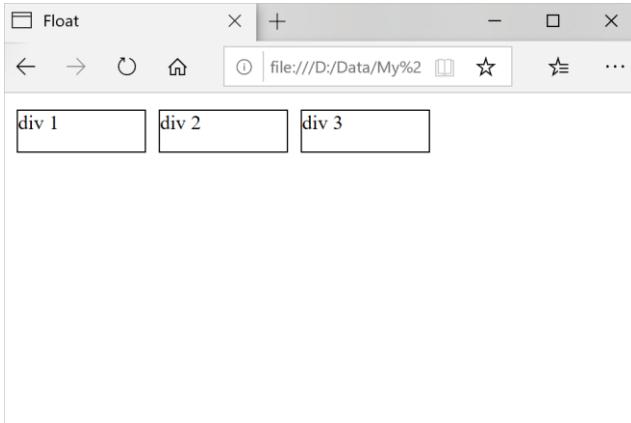
      .left {
        float: left;
      }
    </style>
  </head>

  <body>
    <div class="left">
      div 1
    </div>
    <div class="left">
      div 2
    </div>
    <div class="left">
      div 3
    </div>
  </body>

```

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

Hasilnya dapat dilihat seluruh elemen <div> melayang ke arah kiri seperti yang terlihat pada.



Gambar 105. Contoh property float dengan nilai left.

Walau pemberian nilai right atau float hanya diberikan pada sebuah elemen saja, namun hal itu dapat berdampak pada elemen lain yang ada setelah elemen tersebut. Berikut adalah contoh dari kasus tersebut.

```
float_3.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Float</title>
        <style>
            div {
                border: solid 1px black;
                margin-bottom: 5px;
                text-align: center;
            }

            #header {
                width: 100%;
                height: 32px;
                background-color: gainsboro;
            }

            #menu-right {
                width: 132px;
            }
        </style>
    </head>
    <body>
        <div>div 1</div>
        <div>div 2</div>
        <div>div 3</div>
    </body>
</html>
```

```

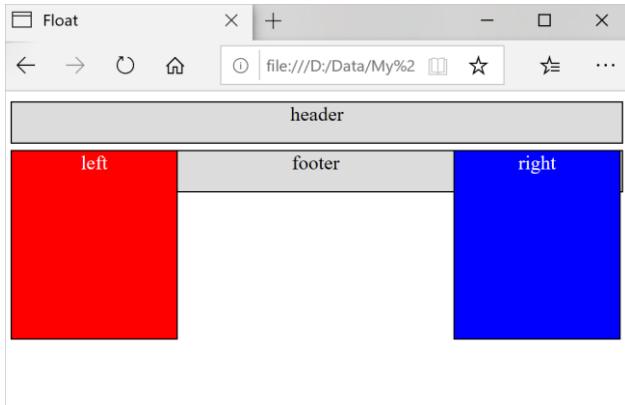
        height: 150px;
        background-color: blue;
        float: right;
        color: white;
    }

    #menu-left {
        width: 132px;
        height: 150px;
        background-color: red;
        float: left;
        color: white;
    }

    #footer {
        width: 100%;
        height: 32px;
        background-color: gainsboro;
        clear: both;
    }
</style>
</head>

<body>
    <div id="header">
        header
    </div>
    <div id="menu-right">
        right
    </div>
    <div id="menu-left">
        left
    </div>
    <div id="footer">
        footer
    </div>
</body>
</html>
```

Jika halaman web dari kode di atas dilihat pada web browser maka akan dapat dilihat hasil seperti pada Gambar 106. Dapat dilihat elemen <div> footer berada di sisi kiri elemen <div> left (warna merah). Hal ini terjadi karena efek penggunaan property float dengan nilai left pada elemen <div> left.



Gambar 106. Contoh efek property float pada elemen setelahnya.

Untuk menghilangkan pengaruh efek tersebut dapat dilakukan dengan memberi property clear pada elemen setelah elemen <div> left tersebut. Untuk kasus ini adalah elemen <div> footer. Berikut adalah contoh penggunaan property footer untuk elemen <div> footer. Dan hasilnya dapat dilihat pada Gambar 107.

```
#footer {  
    width: 100%;  
    height: 32px;  
    background-color: gainsboro;  
    clear: both;  
}
```



Gambar 107. Contoh penggunaan dan efek property clear.

Project

Pada sub bab ini akan diberikan project yang akan diterangkan langkah-langkah untuk membuat tata letak dengan CSS.

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 screenshot displays a registration form titled "EVENT REGISTRATION FORM". The form fields include Name (First Name and Last Name), Company, Email (example@email.com), Phone (Area Code and Phone Number), Subject (Choose Option dropdown), and a question about existing customer status (Yes or No). A large red "REGISTER" button is at the bottom.

| Name | |
|--------------------------------------|-----------------------------|
| First Name | Last Name |
| Company | |
| Email | example@email.com |
| Phone | Area Code Phone Number |
| Subject | Choose Option |
| Are you an existing customer? | |
| <input checked="" type="radio"/> Yes | <input type="radio"/> No |
| REGISTER | |

Gambar 108. Form registrasi event.

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

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;
            }

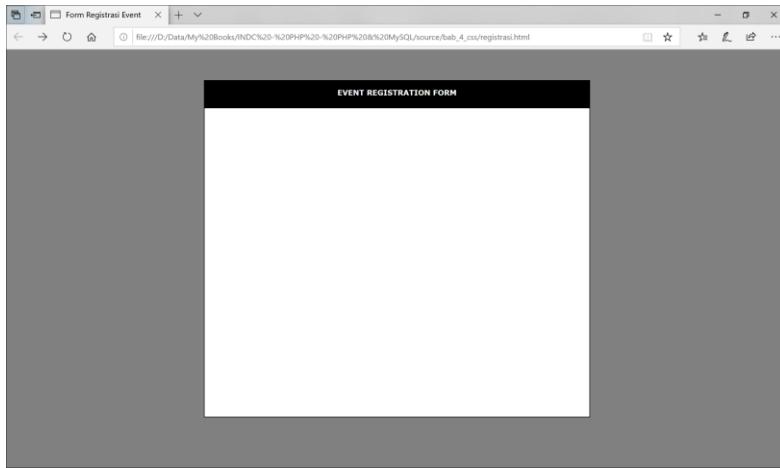
            .form_title {
                border: solid 1px #000000;
                background-color: #000000;
                color: #FFFFFF;
                height: 32px;
                font-weight: bold;
                text-align: center;
                padding-top: 11px;
            }

            .form_content {
                width: 90%;
                margin: auto;
            }
        </style>
    </head>
    <body>
        <div id="main_content">
            <div class="form_box">
                <div class="form_title">
                    <h2>Registrasi Event</h2>
                </div>
                <div class="form_content">
                    <form>
                        <div>
                            <label>Nama <input type="text" name="name"></label>
                        </div>
                        <div>
                            <label>Email <input type="text" name="email"></label>
                        </div>
                        <div>
                            <label>Telepon <input type="text" name="phone"></label>
                        </div>
                        <div>
                            <label>Pesan <input type="text" name="pesan"></label>
                        </div>
                    </form>
                </div>
            </div>
        </div>
    </body>
</html>
```

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

Hasilnya dapat dilihat pada gambar berikut ini.



Gambar 109. 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".

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>
```

```

</ul>

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

```

Hasilnya dapat dilihat pada gambar berikut ini.

The screenshot shows a registration form titled "EVENT REGISTRATION FORM". It contains fields for Name, Company, Email, and Phone. A dropdown menu for Subject is shown with the option "Choose Option". A question "Are you an existing customer?" is followed by four radio button options: "Yes", "No", and two empty radio buttons. A "REGISTER" button is at the bottom.

Gambar 110. 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;
      }
    </style>
  </head>
  <body>
    <h1>Event Registration</h1>
    <form>
      <ul>
        <li>Name: <input type="text" /></li>
        <li>Company: <input type="text" /></li>
        <li>Email: <input type="text" /></li>
        <li>Phone: <input type="text" /></li>
      </ul>
      <label>Subject: <input type="button" value="Choose Option" /></label>
      <p>Are you an existing customer?</p>
      <ul>
        <li><input type="radio" /> Yes</li>
        <li><input type="radio" /> No</li>
        <li><input type="radio" /></li>
        <li><input type="radio" /></li>
      </ul>
      <input type="button" value="REGISTER" />
    </form>
  </body>
</html>

```

```
        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;
    }

    .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="form_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>

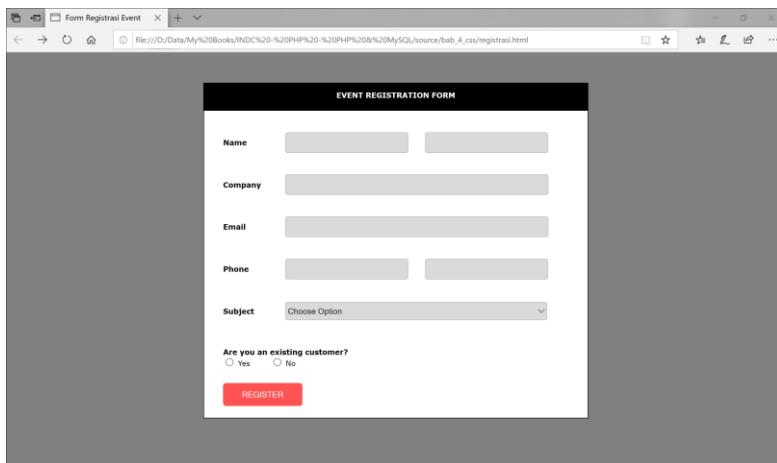
<label class="form_label_horizontal">
    Are you an existing customer?
</label>
<ul class="form_radio_horizontal">
    <li>
        <input type="radio"
               name="is_existing"
               value="yes" checked>
    </li>
    <li class="item_text">Yes</li>
    <li>
        <input type="radio"
               name="is_existing"
               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 111. Form input data yang telah diberikan CSS.

Layout Website

Pada sub bab ini akan diperlihatkan langkah-langkah untuk membuat layout website yang umum digunakan. Umumnya website terbagi atas tiga bagian yaitu:

- Header, kepala website ini umumnya berisi nama website dan gambar banner atau logo.
- Menu, area ini berisi hyperlink untuk menuju halaman lain.
- Content, area isi dari website.
- Footer, area paling bawah pada halaman web. Biasanya berisi keterangan pembuat atau copyright.



Gambar 112. Contoh layout website sederhana 1.

Berikut pada Gambar 112 adalah antarmuka layout website yang sederhana. Untuk membuat halaman tersebut dapat dilakukan dengan tahapan-tahapan sebagai berikut. Tahap pertama digunakan kode sebagai berikut.

```
website layout 1.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Website Layout</title>
        <style>
            #main-container {
                width: 90%;
                margin: auto;
            }

            #header {
                width: 100%;
                height: 75px;
                background-color: lightgray;
            }

            #menu {
                margin-top: 7px;
                margin-bottom: 7px;
                width: 100%;
                height: 32px;
                background-color: rgb(150, 150, 150);
            }
        </style>
    </head>
    <body>
        <div id="header"></div>
        <div id="menu"></div>
        <div id="content"></div>
        <div id="footer"></div>
    </body>
</html>
```

```

#main-content {
    margin-top: 7px;
    margin-bottom: 7px;
    width: 100%;
    height: 450px;
    background-color: lightgray;
}

#footer {
    width: 100%;
    height: 50px;
    background-color: lightgray;
}

.separator {
    clear: both;
}

```

</style>

</head>

<body>

<div id="main-container">

<div id="header">

 header

</div>

<div id="menu">

 menu

</div>

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

<div id="main-content">

 content

</div>

<div id="footer">

 footer

</div>

</div>

</body>

</html>

Kode di atas akan membuat antarmuka seperti pada Gambar 113. Kode di atas menggunakan property-property CSS yang telah dijelaskan sebelumnya, sehingga tidak perlu ada penjelasan mendalam. Pada kode HTML terdapat elemen <div> dengan id adalah main-container yang berfungsi sebagai wadah utama untuk elemen-elemen didalamnya. Lebar elemen ini menggunakan nilai relatif yaitu 90%. Nilai multlak juga dapat digunakan untuk nilai lebar elemen

ini. Cara ini umum digunakan agar elemen-elemen didalamnya akan mengikuti lebar dan posisi dari wadah utama ini.

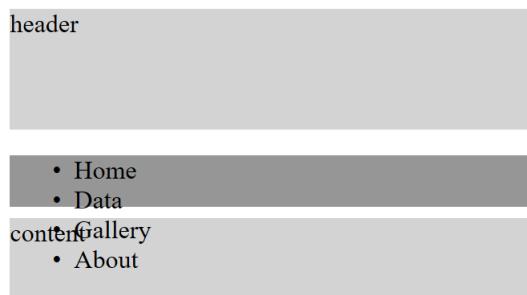


Gambar 113. Layout website - tahap 1.

Selanjutnya adalah menambahkan elemen di dalam elemen <div> dengan id adalah menu. Berikut adalah elemen yang ditambahkan tersebut.

```
<div id="menu">
  <ul>
    <li>Home</li>
    <li>Data</li>
    <li>Gallery</li>
    <li>About</li>
  </ul>
</div>
```

Hasilnya dapat dilihat seperti pada Gambar 114.

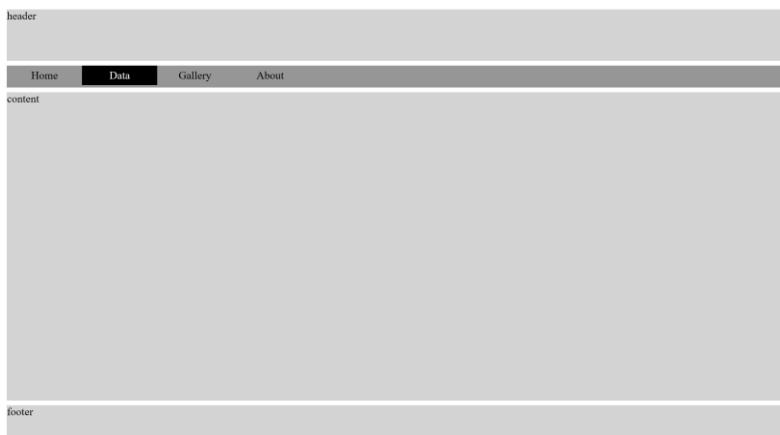


Gambar 114. Elemen list di dalam elemen <div> menu.

Untuk membuat menu umum digunakan elemen dan . Namun tampilan dari elemen tersebut adalah seperti pada Gambar 114. Sehingga perlu diberikan style untuk membuat menu horisontal. Berikut adalah kode CSS yang dapat digunakan.

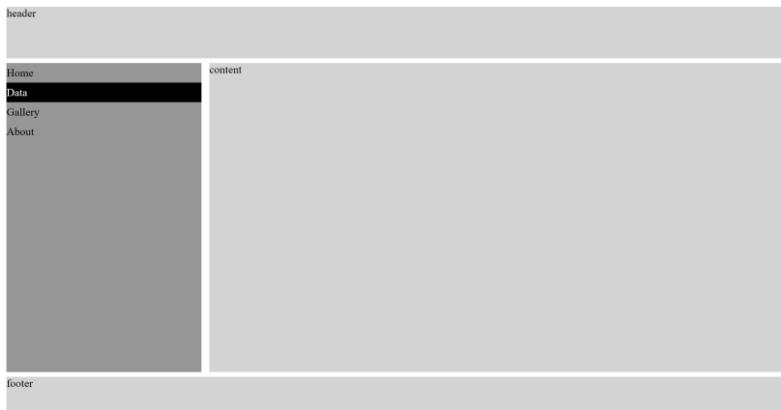
```
#menu ul {  
    margin: 0;  
    padding: 0;  
    list-style-type: none;  
}  
  
#menu ul li {  
    float: left;  
    width: 100px;  
    padding: 5px;  
    text-align: center;  
}  
  
#menu ul li:hover {  
    background-color: black;  
    color: white;  
}
```

Dan hasilnya dapat dilihat pada Gambar 115 dimana menu ditampilkan secara horisontal. Dan ketika cursor diarahkan ke salah satu item menu terlihat warna item menu berubah warna.



Gambar 115. Elemen list yang telah diberi style.

Contoh kedua adalah membuat layout website seperti pada Gambar 116, dimana menu berada di sisi kiri dan content berada di sisi kanan.



Gambar 116. Contoh layout website sederhana 2.

Untuk membuat antarmuka di atas digunakan kode berikut ini.

```
website_layout_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Website Layout</title>
        <style>
            #main-container {
                width: 90%;
                margin: auto;
            }

            #header {
                width: 100%;
                height: 75px;
                background-color: lightgray;
            }

            #menu {
                margin-top: 7px;
                margin-bottom: 7px;
                width: 100%;
                height: 450px;
                background-color: rgb(150, 150, 150);
                width: 25%;
                float: left;
            }

            #menu ul {
                margin: 0;
                padding: 0;
            }
        </style>
    </head>
    <body>
        <div id="header"></div>
        <div id="menu">
            <ul>
                <li>Home</li>
                <li>Data</li>
                <li>Gallery</li>
                <li>About</li>
            </ul>
        </div>
    </body>
</html>
```

```
        list-style-type: none;
    }

    #menu ul li {
        width: 100%;
        padding-top: 5px;
        padding-bottom: 5px;
    }

    #menu ul li:hover {
        background-color: black;
        color: white;
    }

    #main-content {
        margin-top: 7px;
        margin-bottom: 7px;
        width: 74%;
        height: 450px;
        background-color: lightgray;
        float: right;
    }

    #footer {
        width: 100%;
        height: 50px;
        background-color: lightgray;
    }

    .separator {
        clear: both;
    }

```

</style>

</head>

<body>

```
    <div id="main-container">
        <div id="header">
            header
        </div>
        <div id="menu">
            <ul>
                <li>Home</li>
                <li>Data</li>
                <li>Gallery</li>
                <li>About</li>
            </ul>
        </div>
    </div>
```

```

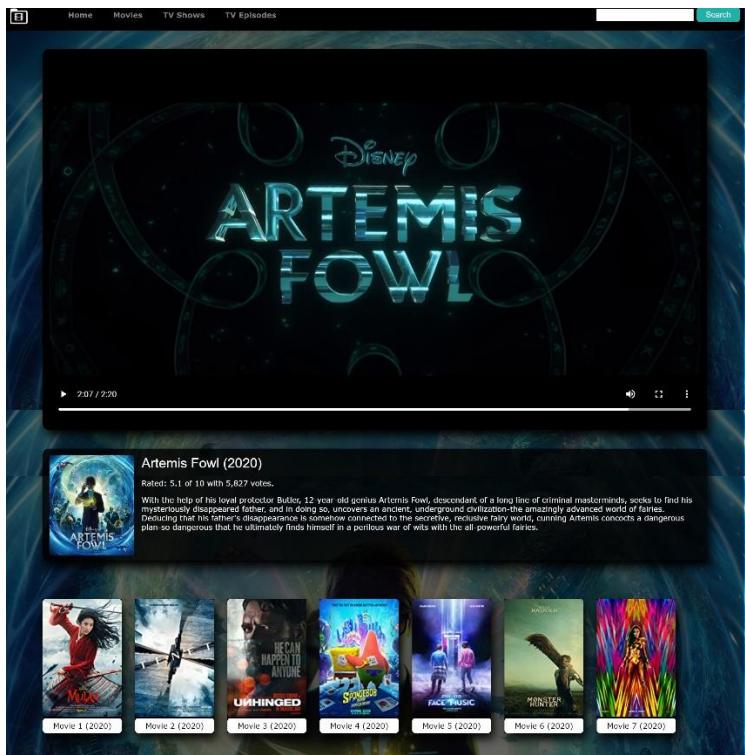
<div id="main-content">
    content
</div>
<div class="separator"></div>
<div id="footer">
    footer
</div>
</div>

</body>
</html>

```

Video Player

Pada project ini bertujuan untuk membuat halaman web untuk memutar video dengan layout yang umum dipakai oleh website streaming film. Pada Gambar 117 dapat dilihat antarmuka yang akan dibuat.



Gambar 117. Website video player.

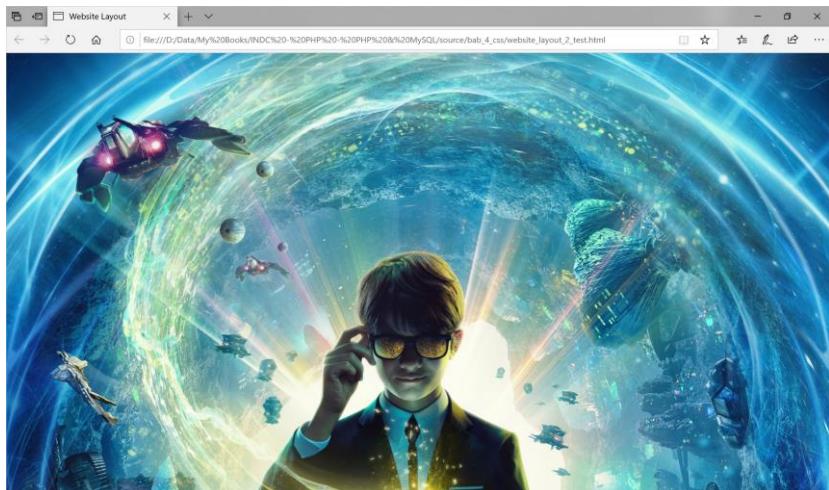
Langkah pertama adalah menyiapkan style untuk elemen <body> dengan kode sebagai berikut. Kode CSS ini bertujuan untuk menentukan jenis font dan ukuran. Selain itu juga diberikan gambar sebagai latarbelakang halaman.

```
website_layout_2.html
<!DOCTYPE HTML>
<html>
  <head>
    <title>Website Layout</title>
    <style>
      body {
        background-image: url(images/movie/movie-bg.jpg);
        background-repeat: no-repeat;
        background-size: cover;
        background-attachment: fixed;
        font-family: Verdana, Geneva, Tahoma, sans-serif;
        font-size: small;
      }
    </style>
  </head>

  <body>

  </body>
</html>
```

Hasilnya dapat dilihat pada Gambar 118.



Gambar 118. Video player - latarbelakang.

Langkah kedua adalah membuat elemen <div> untuk menutupi latarbelakang dengan lapisan berwarna hitam transparansi sehingga background di belakangnya masih terlihat. Di dalam elemen <body> ditambahkan kode berikut.

```
<div id="main-container">  
    </div>
```

Kemudian tambahkan kode CSS ini pada blok style.

```
#main-container {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    background-color: rgba(0, 0, 0, 0.6);  
    color: white;  
    margin: 0;  
    padding: 0;  
}
```

Jika halaman direfresh, maka lapisan hitam ini belum terlihat. Elemen <div> dari lapisan ini akan terlihat jika didalamnya sudah dipenuhi oleh elemen-elemen HTML.

Langkah ketiga adalah membuat menu dengan menambahkan kode HTML berikut ini di dalam elemen <div> yang sebelumnya telah dibuat.

```
<!-- header - start -->  
<div id="header">  
    <!-- logo - start -->  
    <div id="logo">  
          
    </div>  
    <!-- logo - end -->  
  
    <!-- menu - start -->  
    <div id="menu">  
        <ul>  
            <li>Home</li>  
            <li>Movies</li>  
            <li>TV Shows</li>  
            <li>TV Episodes</li>  
        </ul>  
    </div>  
    <!-- menu - end -->
```

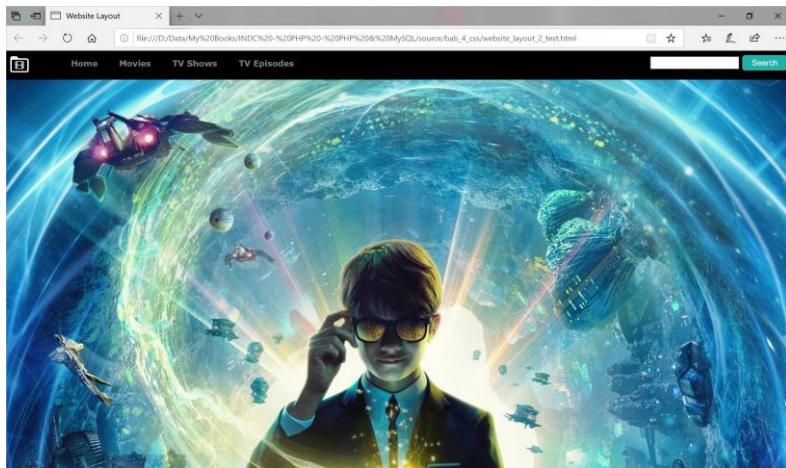
```

<!-- search - start -->
<div id="search">
    <form method="POST">
        <input type="text" name="keyword">
        <input type="button" value="Search">
    </form>
</div>
<!-- search - end -->

<div class="separator"></div>
</div>
<!-- header - end -->

```

Hasilnya dapat dilihat pada Gambar 119. Pada header ini dapat dilihat logo, menu dan kolom pencarian.



Gambar 119. Video player - menu dan kolom pencarian.

Langkah keempat adalah membuat content dengan kode sebagai berikut ini ditambahkan yang di bawah kode header.

```

<!-- content - start -->
<div id="content-container">

    <div class="separator" style="margin-bottom: 50px;"></div>
</div>
<!-- content - end -->

```

Dengan tambahan kode CSS sebagai berikut.

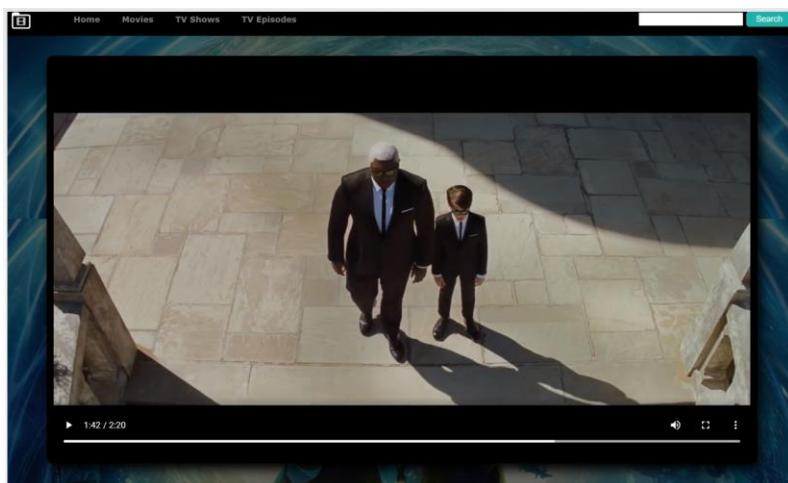
```
#content-container {  
    width: 90%;  
    margin: auto  
}
```

Langkah kelima adalah menambahkan video player di dalam elemen <div> content tersebut dengan kode HTML sebagai berikut.

```
<!-- video player - start -->  
<div id="video-player">  
    <video width="100%" controls>  
        <source src="images/movie/movie.mp4" type="video/mp4">  
        Your browser does not support the video tag.  
    </video>  
</div>  
<!-- video player - end -->
```

Dan tambahkan kode CSS berikut ini.

```
#video-player {  
    margin-top: 32px;  
    background-color: black;  
    border: solid 1px black;  
    border-radius: 10px;  
    padding: 10px;  
    box-shadow: 5px 10px 18px black;  
}
```



Gambar 120. Video player - video player

Hasil dari tahap kelima ini dapat dilihat tampilan seperti pada Gambar 120. Sekarang sudah dapat dilihat latarbelakang transparansi warna hitam yang sebelumnya dibuat. Kemudian dapat dilihat video player ini juga berwarna hitam. Jika diperhatikan player ini seperti berada di atas halaman dan mempunyai bayangan. Hal ini bisa terjadi karena menggunakan property box-shadow.

Langkah keenam adalah membuat keterangan dari film yang ditampilkan pada video player tersebut dengan menggunakan kode HTML sebagai berikut.

```
<!-- deskripsi film - start -->
<div id="description-container">
    
    <div class="description">
        <h4 class="movie-title">Artemis Fowl (2020)</h4>
        <p>
            Rated: 5.1 of 10 with 5,827 votes.
        </p>
        <p>
            With the help of his loyal protector Butler, 12-year-old genius Artemis Fowl, descendant of a long line of criminal masterminds, seeks to find his mysteriously disappeared father, and in doing so, uncovers an ancient, underground civilization—the amazingly advanced world of fairies. Deducing that his father's disappearance is somehow connected to the secretive, reclusive fairy world, cunning Artemis concocts a dangerous plan—so dangerous that he ultimately finds himself in a perilous war of wits with the all-powerful fairies.
        </p>
    </div>
    <div class="separator"></div>
</div>
<!-- deskripsi film - end -->
```

Dan kode CSS sebagai berikut.

```
#description-container {
    margin-top: 32px;
    background-color: rgba(0, 0, 0, 0.8);
    border: solid 1px black;
    border-radius: 10px;
    padding: 10px;
    box-shadow: 5px 10px 18px black;
}

.movie-poster {
```

```

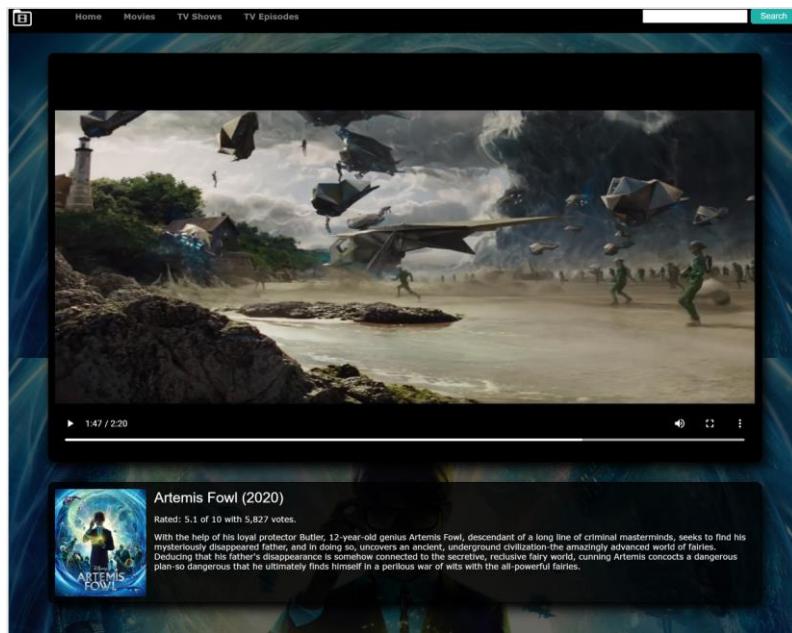
width: 13%;
float: left;
}

.description {
width: 85%;
float: left;
padding-left: 13px;
}

.movie-title {
margin: 0;
padding: 0;
font-family: Arial, Helvetica, sans-serif;
font-size: 23px;
font-weight: lighter;
}

```

Hasilnya dapat dilihat pada Gambar 121. Pada gambar ini dapat dilihat keterangan dari film ini di bawah video player.



Gambar 121. Video player - deskripsi film.

Langkah yang terakhir adalah membuat daftar film yang bisa ditonton. Berikut adalah kode HTML yang ditambahkan.

```
<!-- daftar film - start -->
<div id="movie-list-container">
    <div class="movie-item">
        
        <div class="title">
            <a href="#">Movie 1 (2020)</a>
        </div>
    </div>
    . . .
    <div class="movie-item">
        
        <div class="title">
            <a href="#">Movie 7 (2020)</a>
        </div>
    </div>
</div>
<!-- daftar film - end -->
```

Dan berikut adalah kode CSS yang digunakan.

```
#movie-list-container {
    width: 100%;
    margin-top: 32px;
}

.movie-item {
    width: 135px;
    float: left;
    margin-top: 32px;
    margin-right: 23px;
    background-color: black;
    box-shadow: 5px 10px 18px black;
    position: relative;
}

.movie-item img {
    width: 100%;
    border-radius: 5px;
}

.movie-item .title {
    background-color: white;
    color: black;
    text-align: center;
    width: 100%;
```

```

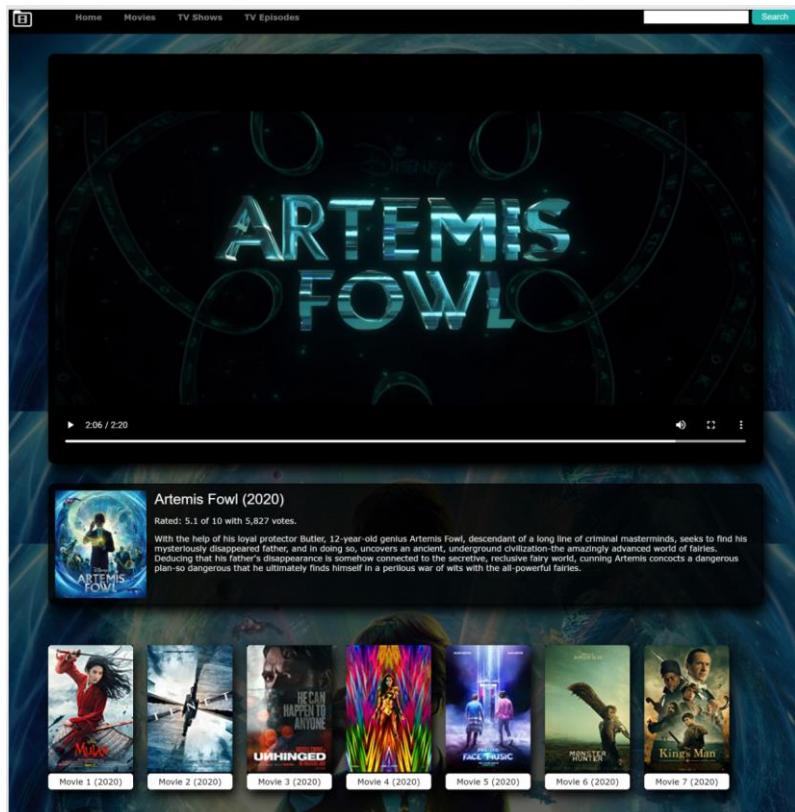
border-radius: 5px;
padding-top: 5px;
padding-bottom: 5px;
}

.movie-item .title a {
    text-decoration: none;
    color: black;
}

.movie-item .title a:hover {
    text-decoration: underline;
    color: black;
}

```

Hasilnya dapat dilihat pada Gambar 122.



Gambar 122. Video player - daftar film.

Kode lengkap dari halaman web di atas dapat dilihat pada kode di bawah ini.

```
website_layout_2.html
<!DOCTYPE HTML>
<html>
    <head>
        <title>Website Layout</title>
        <style>
            body {
                background-image: url(images/movie/movie-bg.jpg);
                background-repeat: no-repeat;
                background-size: cover;
                background-attachment: fixed;
                font-family: Verdana, Geneva, Tahoma, sans-serif;
                font-size: small;
            }

            #main-container {
                position: absolute;
                top: 0;
                left: 0;
                width: 100%;
                background-color: rgba(0, 0, 0, 0.6);
                color: white;
                margin: 0;
                padding: 0;
            }

            #header {
                background-color: black;
                width: 100%;
            }

            #logo {
                float: left;
                padding: 7px;
            }

            #menu {
                float: left;
                margin-left: 50px;
                padding: 7px;
            }

            #menu ul {
                margin: 0;
                padding: 0;
                list-style-type: none;
            }
        </style>
    </head>
    <body>
        <div id="main-container">
            <div id="header"></div>
            <div id="logo"></div>
            <div id="menu">
                <ul>
                    <li>Home</li>
                    <li>Movie</li>
                    <li>About</li>
                    <li>Contact</li>
                </ul>
            </div>
        </div>
    </body>
</html>
```

```
}

#menu ul li {
    float: left;
    padding: 5px;
    margin-right: 25px;
    text-align: center;
    color: grey;
    font-weight: bolder;
}

#menu ul li:hover {
    color: white;
    font-weight: bolder;
}

#search {
    float: right;
    padding: 7px;
}

#search input[type=button] {
    background-color: lightseagreen;
    border: solid 1px lightseagreen;
    border-radius: 5px;
    height: 25px;
    padding-left: 15px;
    padding-right: 15px;
    color: white;
}

.separator {
    clear: both;
}

#content-container {
    width: 90%;
    margin: auto
}

#video-player {
    margin-top: 32px;
    background-color: black;
    border: solid 1px black;
    border-radius: 10px;
    padding: 10px;
    box-shadow: 5px 10px 18px black;
}
```

```
#description-container {
    margin-top: 32px;
    background-color: rgba(0, 0, 0, 0.8);
    border: solid 1px black;
    border-radius: 10px;
    padding: 10px;
    box-shadow: 5px 10px 18px black;
}

.movie-poster {
    width: 13%;
    float: left;
}

.description {
    width: 85%;
    float: left;
    padding-left: 13px;
}

.movie-title {
    margin: 0;
    padding: 0;
    font-family: Arial, Helvetica, sans-serif;
    font-size: 23px;
    font-weight: lighter;
}

#movie-list-container {
    width: 100%;
    margin-top: 32px;
}

.movie-item {
    width: 135px;
    float: left;
    margin-top: 32px;
    margin-right: 23px;
    background-color: black;
    box-shadow: 5px 10px 18px black;
    position: relative;
}

.movie-item img {
    width: 100%;
    border-radius: 5px;
}
```

```

.movie-item .title {
    background-color: white;
    color: black;
    text-align: center;
    width: 100%;
    border-radius: 5px;
    padding-top: 5px;
    padding-bottom: 5px;
}

.movie-item .title a {
    text-decoration: none;
    color: black;
}

.movie-item .title a:hover {
    text-decoration: underline;
    color: black;
}
</style>
</head>

<body>
    <div id="main-container">
        <!-- header - start -->
        <div id="header">
            <!-- logo - start -->
            <div id="logo">
                
            </div>
            <!-- logo - end -->

            <!-- menu - start -->
            <div id="menu">
                <ul>
                    <li>Home</li>
                    <li>Movies</li>
                    <li>TV Shows</li>
                    <li>TV Episodes</li>
                </ul>
            </div>
            <!-- menu - end -->

            <!-- search - start -->
            <div id="search">
                <form method="POST">

```

```

        <input type="text" name="keyword">
        <input type="button" value="Search">
    </form>
</div>
<!-- search - end -->

<div class="separator"></div>
</div>
<!-- header - end -->

<!-- content - start -->
<div id="content-container">

    <!-- video player - start -->
    <div id="video-player">
        <video width="100%" controls>
            <source src="images/movie/movie.mp4"
                    type="video/mp4">
                Your browser does not support the video tag.
        </video>
    </div>
    <!-- video player - end -->

    <!-- deskripsi film - start -->
    <div id="description-container">
        
        <div class="description">
            <h4 class="movie-title">
                Artemis Fowl (2020)
            </h4>
            <p>
                Rated: 5.1 of 10 with 5,827 votes.
            </p>
            <p>
                With the help of his loyal protector Butler, 12-year-old genius Artemis Fowl, descendant of a long line of criminal masterminds, seeks to find his mysteriously disappeared father, and in doing so, uncovers an ancient, underground civilization—the amazingly advanced world of fairies. Deducing that his father's disappearance is somehow connected to the secretive, reclusive fairy world, cunning Artemis concocts a dangerous plan—so dangerous that he ultimately finds himself in a perilous war of wits with the all-powerful fairies.
            </p>
        </div>
        <div class="separator"></div>

```

```
</div>
<!-- deskripsi film - end --&gt;

&lt;!-- daftar film - start --&gt;
&lt;div id="movie-list-container"&gt;
    &lt;div class="movie-item"&gt;
        &lt;img src="images/movie/m-1.jpg"&gt;
        &lt;div class="title"&gt;
            &lt;a href="#"&gt;Movie 1 (2020)&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="movie-item"&gt;
        &lt;img src="images/movie/m-2.jpg"&gt;
        &lt;div class="title"&gt;
            &lt;a href="#"&gt;Movie 2 (2020)&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="movie-item"&gt;
        &lt;img src="images/movie/m-3.jpg"&gt;
        &lt;div class="title"&gt;
            &lt;a href="#"&gt;Movie 3 (2020)&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="movie-item"&gt;
        &lt;img src="images/movie/m-8.jpg"&gt;
        &lt;div class="title"&gt;
            &lt;a href="#"&gt;Movie 4 (2020)&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="movie-item"&gt;
        &lt;img src="images/movie/m-5.jpg"&gt;
        &lt;div class="title"&gt;
            &lt;a href="#"&gt;Movie 5 (2020)&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="movie-item"&gt;
        &lt;img src="images/movie/m-6.jpg"&gt;
        &lt;div class="title"&gt;
            &lt;a href="#"&gt;Movie 6 (2020)&lt;/a&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="movie-item"&gt;</pre>
```

```
        
        <div class="title">
            <a href="#">Movie 7 (2020)</a>
        </div>
    </div>
    <!-- daftar film - end -->

    <div class="separator" style="margin-bottom: 50px;">
        </div>
    </div>
    <!-- content - end -->

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

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 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>

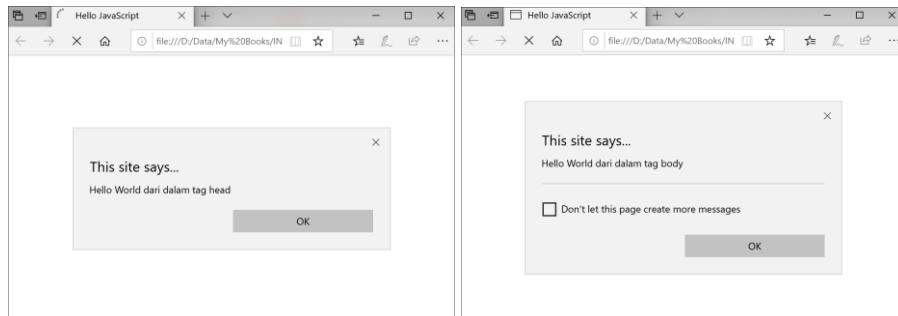
```

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 123.



Gambar 123. Message box keluaran JavaScript.

External

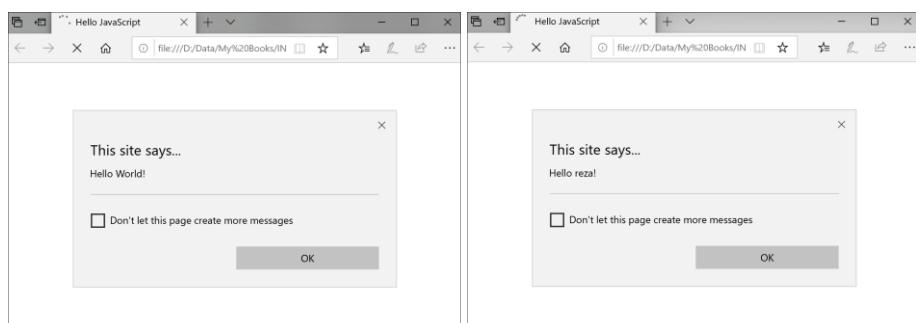
Aturan penulisan ini memisahkan kode dalam bentuk fungsi-fungsi ke suatu file terpisah. Script JavaScript disimpan pada file dengan ekstensi .js. Berikut adalah cara untuk menyematkan file ini pada suatu halaman web.

```
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 `src` 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 + "!");
}
```



Gambar 124. Message box dari fungsi JavaScript.

Keluaran program di atas mengeluarkan message box sebanyak dua kali seperti pada Gambar 124.

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

Berikut ini adalah contoh penggalan kode pengulangan dengan while.

```
// inisialisasi nilai
var i = 1;

// kondisi pengulangan
while (i <= 3) {
    // statement yang akan diulang
```

```
// meningkatkan pencacah  
i = i + 1;  
}
```

Berikut ini adalah contoh kode untuk membuat window baru sebanyak 3 buah dengan menggunakan pengulangan while.

```
pengulangan_while_1.html  
<!DOCTYPE html>  
<html>  
    <head>  
        <title>Pengulangan</title>  
        <script>  
            // inisialisasi nilai  
            var i = 1;  
  
            // kondisi pengulangan  
            while (i <= 3) {  
                window.open("http://www.google.com");  
  
                // meningkatkan pencacah  
                i = i + 1;  
            }  
        </script>  
    </head>  
  
    <body>  
  
    </body>  
</html>
```

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  
        }  
    </script>
```

```

        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.

Berikut ini adalah contoh penulisan method dan cara untuk menggunakan atau memanggil fungsi tersebut. Contoh kode berikut ini menampilkan dialog window yang bertuliskan pesan “hello world”.

```

function 2.html
<!DOCTYPE html>
<html>
<head>
    <title>Fungsi</title>
</head>
<script>
    function hello() {
        alert("hello world")
    }

    hello();
</script>
<body>
</body>
</html>

```

Contoh berikutnya adalah method yang memiliki parameter input. Contoh di bawah ini akan menampilkan hasil penjumlahan bilangan yang dimasukkan sebagai nilai parameter input bil1 dan bil2.

```

function 3.html
<!DOCTYPE html>
<html>
<head>
    <title>Fungsi</title>
</head>

```

```

<script>
    function tambah(bil1, bil2) {
        var hasil = bil1 + bil2;
        alert("hasil penjumlahan "+ bil1 +" dan "+bil2 +
              " adalah "+hasil);
    }

    tambah(3,4);
</script>
<body>
</body>
</html>

```

Contoh yang ketiga adalah method yang mengembalikan nilai. Untuk menggunakan method jenis ini perlu digunakan variable penampung nilai. pada contoh di bawah ini, yang menjadi penampung nilai adalah nilai.

```

function 4.html
<!DOCTYPE html>
<html>
<head>
    <title>Fungsi</title>
</head>
<script>
    function tambah(bil1, bil2) {
        var hasil = bil1 + bil2;
        return hasil;
    }

    var nilai = tambah(3, 4);
    alert(nilai);
</script>
<body>
</body>
</html>

```

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>
```

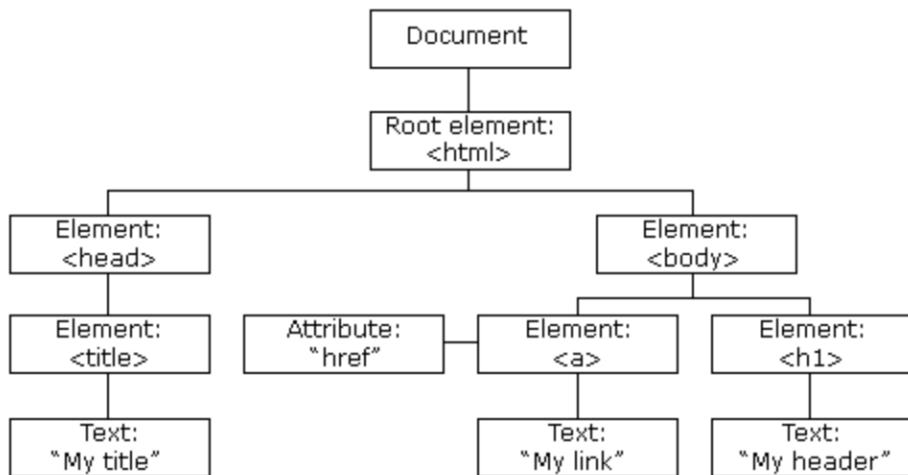
```

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 hierarki obyek-obyek pada halaman dengan struktur pohon seperti yang terlihat pada Gambar 125.



Gambar 125. 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 = "";
    }
  </script>
</html>
```

```

        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 = "";

```

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 getElementsByClassName 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.

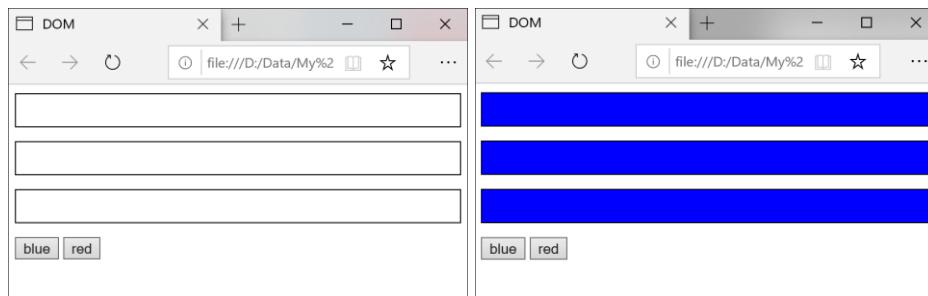
Dari paparan di atas dapat disimpulkan bahwa:

- getElementById cocok digunakan untuk mendapatkan elemen yang unik.
- Sedangkan jika ingin mendapatkan sekumpulan elemen digunakan method getElementsByName dan getElementsByClassName.

Berikut ini contoh penggunaan method getElementsByClassName.

```
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.getElementsByClassName("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 126.



Gambar 126. 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.attribute 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 atribut 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>
</html>
```

```
</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

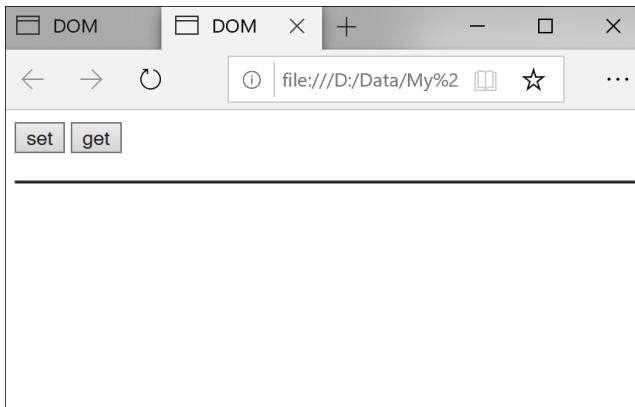
Sedangkan fungsi dan cara penggunaan innerHTML dapat dilihat pada kode berikut ini.

```
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>
```

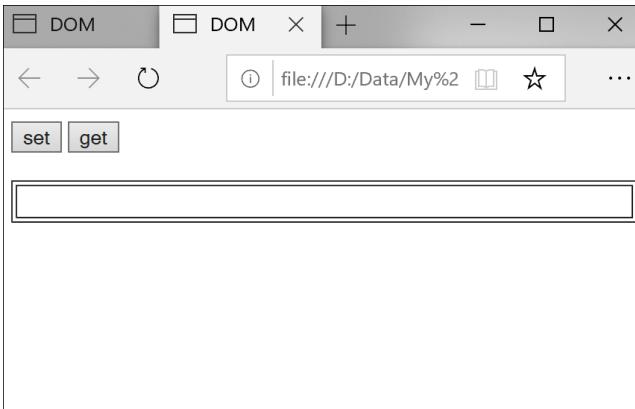
innerHTML dapat digunakan untuk menyisipkan kode HTML ke dalam suatu elemen. Selain itu juga dapat digunakan untuk mengetahui kode HTML pada suatu elemen.

Pada gambar Gambar 127 dapat dilihat tampilan awal saat halaman baru dimuat.



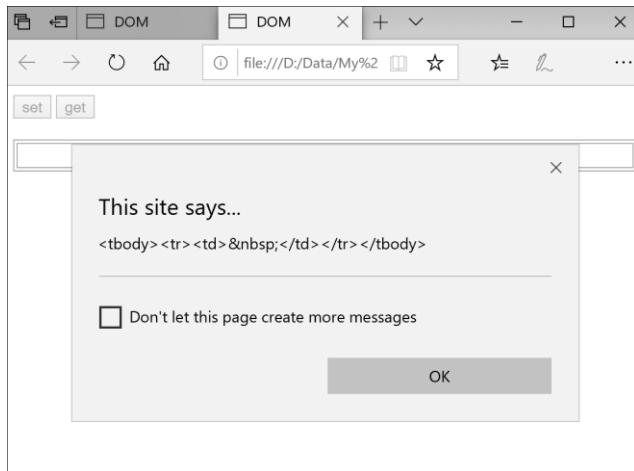
Gambar 127. Tampilan awal.

Selanjutnya setelah tombol set diklik maka akan dipanggil method set yang menggunakan innerHTML untuk menyisipkan kode HTML sehingga dapat dilihat hasilnya seperti pada Gambar 128.



Gambar 128. Tampilan setelah tombol set diklik.

Kemudian setelah tombol get diklik akan dapat dilihat hasilnya seperti pada Gambar 129.



Gambar 129. 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 setMessage bertujuan untuk menambah elemen div baru sebelum div1. Elemen div baru berisi teks yang nilainya diisi oleh value pada kolumn 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.

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";

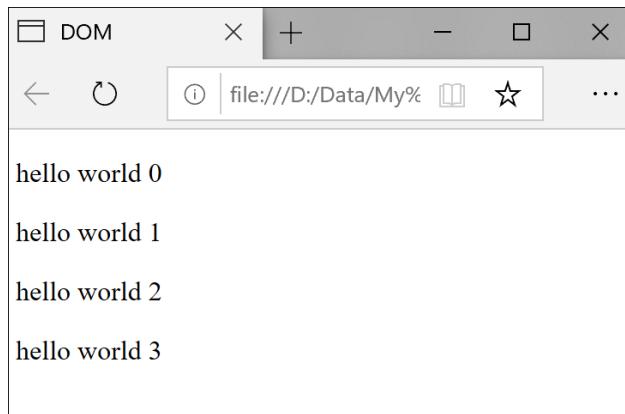
      var obj2 = document.querySelector("#p2");
      obj2.innerHTML = "hello world 2";
    }
  </script>
<body onload="init()">
  <p class="p1"></p>
  <p id="p2"></p>
</body>
</html>
```

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 130.

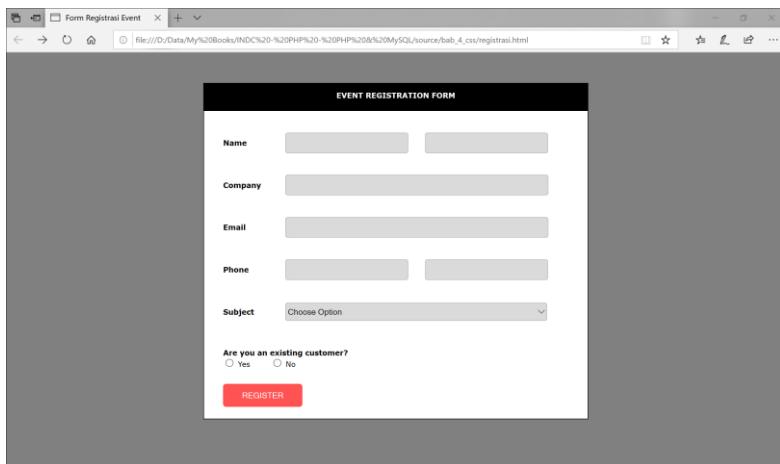


Gambar 130. Memodifikasi banyak elemen dengan querySelectorAll.

Project

Validasi Form

Pada bab sebelumnya telah dibuat form registrasi seperti pada Gambar 131. Proyek kali ini bertujuan untuk memvalidasi pengisian textbox dan dropdown list oleh user. Kode HTML dan CSS yang digunakan adalah seperti yang telah diterangkan pada sub bab Cascading Style Sheets > Project > Form Registrasi.



Gambar 131. Membuat validasi textbox.

Langkah pertama adalah menambahkan atribut id pada setiap elemen pengisian textbox dan dropdown list seperti kode berikut ini.

```
validasi_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;
}

.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 method="GET" id="form1">

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

<label class="form_label">Name</label>

<input type="text" id="first_name" name="first_name" class="form_textbox_short">

<input type="text" id="last_name" name="last_name" class="form_textbox_short">

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

<label class="form_label">Company</label>

<input type="text" id="company" name="company" class="form_textbox">

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

```

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

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

        <label class="form_label">Subject</label>
        <select id="subject" 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_existing"
                       value="yes" checked>
            </li>
            <li class="item_text">Yes</li>
            <li>
                <input type="radio"
                       name="is_existing"
                       value="no">
            </li>
            <li class="item_text">No</li>
        </ul>
        <div class="separator"></div>

        <input type="button" value="REGISTER"
               onclick="register()"
               class="form_button_red">
    </form>
</div>

```

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

Pada kode di atas dapat dilihat pada elemen `<input type="button">` diberikan tambah antribut `onclick="register()"`. Berikut adalah kode JavaScript untuk fungsi `register()`. Pada kode di bawah dapat dilihat deklarasi variable dengan nama sesuai dengan atribut id dari elemen yang akan divalidasi nilainnya. Selanjutnya validasi yang dilakukan hanya validasi sederhana yaitu untuk memeriksa apakah elemen telah diisi atau belum. Jika elemen belum diisi maka akan dipanggil fungsi `alert()` untuk menampilkan pesan. Kemudian cursor diarahkan ke elemen tersebut dengan fungsi `focus()`.

```
<script>
function register() {
    var first_name = document.getElementById("first_name");
    var last_name = document.getElementById("last_name");
    var company = document.getElementById("company");
    var email = document.getElementById("email");
    var area_code = document.getElementById("area_code");
    var phone_number = document.getElementById("phone_number");
    var subject = document.getElementById("subject");

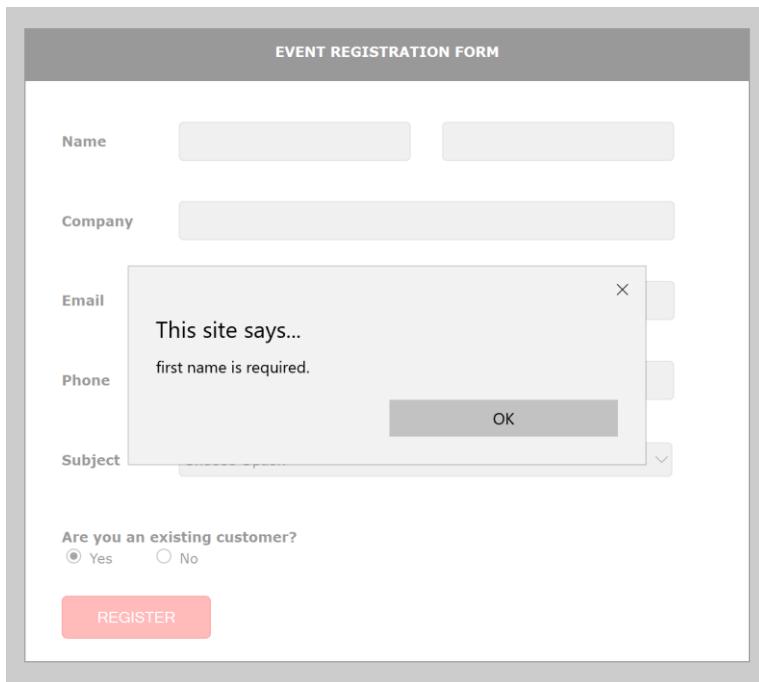
    if(first_name.value == ""){
        alert("first name is required.")
        first_name.focus();
    } else if(last_name.value == ""){
        alert("last name is required.")
        last_name.focus();
    } else if(company.value == ""){
        alert("company is required.")
        company.focus();
    } else if(email.value == ""){
        alert("email is required.")
        email.focus();
    } else if(area_code.value == ""){
        alert("area code is required.")
        area_code.focus();
    } else if(phone_number.value == ""){
        alert("phone number is required.")
        phone_number.focus();
    } else if(subject.value == ""){
        alert("subject is required.")
        subject.focus();
    }
}
```

```
        document.getElementById("form1").submit();
    }
</script>
```

Dan yang terakhir adalah baris berikut ini, yang berfungsi untuk mengirimkan nilai pada setiap elemen. Hal ini dilakukan karena elemen <input type="button"> tidak bisa mengirim nilai secara otomatis seperti elemen <input type="submit">.

```
document.getElementById("form1").submit();
```

Hasilnya dapat dilihat seperti pada Gambar 132.



Gambar 132. Pesan validasi.

Animasi

Project berikutnya membuat animasi sederhana yaitu mengerakkan sebuah elemen <div> dengan id adalah animate yang berada di dalam elemen <div> container.

```
animation_1.html
<!DOCTYPE html>
<html>
    <head>
        <title>Animasi</title>
    </head>
    <style>
        #container {
            width: 400px;
            height: 400px;
            position: relative;
            background: yellow;
        }
        #animate {
            width: 50px;
            height: 50px;
            position: absolute;
            background-color: red;
        }
    </style>

    <script>
        function play() {
            var elem = document.getElementById("animate");
            var pos = 0;
            var id = setInterval(frame, 5);
            function frame() {
                if (pos == 350) {
                    clearInterval(id);
                } else {
                    pos++;
                    elem.style.top = pos + "px";
                    elem.style.left = pos + "px";
                }
            }
        }
    </script>
<body>
    <p><input type="button" onclick="play()" value="Play"></p>

    <div id ="container">
        <div id ="animate"></div>
    </div>
</body>
</html>
```

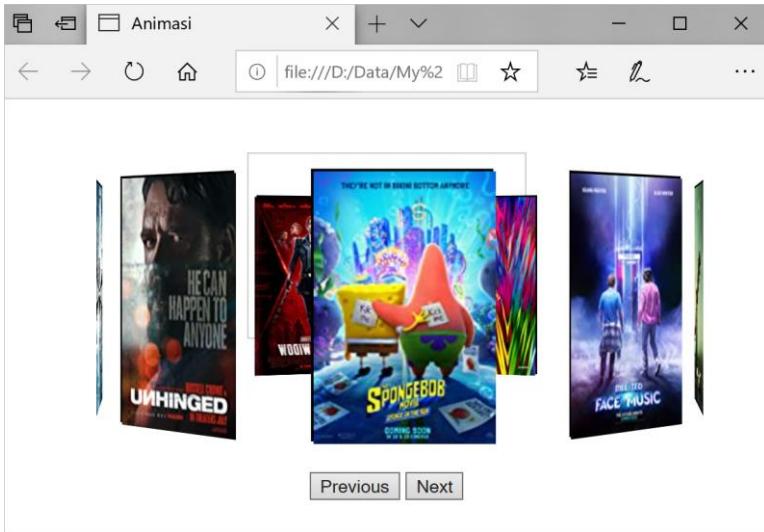
Hasilnya dapat dilihat pada Gambar 133, setelah tombol diklik maka kotak berwarna merah (elemen <div> animate) akan berpindah dari ujung kiri atas ke ujung kanan bawah. Hal ini terjadi karena pada kode dapat dilihat bahwa elemen <div> animate memiliki posisi absolute. Sehingga dengan mengubah nilai top dan left secara terus menerus maka terlihat elemen ini bergerak.



Gambar 133. Animasi dengan JavaScript.

Carousel

Project selanjutnya adalah membuat animasi berupa gambar yang tersusun dan dapat bergerak memutar seperti kemidi putar. Antarmukanya dapat dilihat pada Gambar 134. Jika tombol Next ditekan maka gambar akan bergerak memutar ke arah kanan. Sedangkan jika tombol Previous diklik maka gambar bergerak ke arah kiri.



Gambar 134. Animasi carousel.

Kode yang digunakan untuk project ini adalah sebagai berikut.

```
animation 2.html
<!DOCTYPE html>
<html>
  <head>
    <title>Animasi</title>
  </head>
  <style>
    * { box-sizing: border-box; }

    body { font-family: sans-serif; }

    .scene {
      border: 1px solid #CCC;
      margin: 40px 0;
      position: relative;
      width: 210px;
      height: 140px;
      margin: 40px auto;
      perspective: 1000px;
    }

    .carousel {
      width: 100%;
      height: 100%;
      position: absolute;
      transform: translateZ(-288px);
      transform-style: preserve-3d;
      transition: transform 1s;
    }
```

```
}

.carousel__cell {
    position: absolute;
    width: 140px;
    height: 209px;
    left: 10px;
    top: 10px;
    border: 2px solid black;
    line-height: 116px;
    font-size: 80px;
    font-weight: bold;
    color: white;
    text-align: center;
}

.carousel__cell:nth-child(1) {
    transform: rotateY( 0deg) translateZ(288px);
}

.carousel__cell:nth-child(2) {
    transform: rotateY( 40deg) translateZ(288px);
}

.carousel__cell:nth-child(3) {
    transform: rotateY( 80deg) translateZ(288px);
}

.carousel__cell:nth-child(4) {
    transform: rotateY(120deg) translateZ(288px);
}

.carousel__cell:nth-child(5) {
    transform: rotateY(160deg) translateZ(288px);
}

.carousel__cell:nth-child(6) {
    transform: rotateY(200deg) translateZ(288px);
}

.carousel__cell:nth-child(7) {
    transform: rotateY(240deg) translateZ(288px);
}

.carousel__cell:nth-child(8) {
    transform: rotateY(280deg) translateZ(288px);
}

.carousel__cell:nth-child(9) {
    transform: rotateY(320deg) translateZ(288px);
```

```

        }
    </style>

<body>
    <div class="scene">
        <div class="carousel">
            <div class="carousel_cell">
                
            </div>
            <div class="carousel_cell">
                
            </div>
        </div>
    </div>
    <p style="text-align: center; margin-top: 100px;">
        <button class="previous-button">Previous</button>
        <button class="next-button">Next</button>
    </p>

    <script>
        var carousel = document.querySelector('.carousel');
        var cellCount = 9;
        var selectedIndex = 0;

        function rotateCarousel() {
            var angle = selectedIndex / cellCount * -360;
            carousel.style.transform =
                'translateZ(-288px) rotateY(' + angle + 'deg)';
        }
    </script>

```

```
var prevButton =  
document.querySelector('.previous-button');  
prevButton.addEventListener( 'click', function() {  
    selectedIndex--;  
    rotateCarousel();  
});  
  
var nextButton = document.querySelector('.next-button');  
nextButton.addEventListener( 'click', function() {  
    selectedIndex++;  
    rotateCarousel();  
});  
</script>  
</body>  
</html>
```

Pada kode di atas dapat dilihat digunakan 9 gambar poster film yang ditampilkan di dalam elemen <div> dengan nilai property class adalah scene dan carousel. Setiap poster film berada di dalam elemen <div> dengan nilai property class adalah carousel_class. Elemen-elemen ini diberikan style seperti terlihat pada kode CSS. untuk memberikan efek dan posisi melingkar, setiap poster diberikan style dengan property transform.

Selanjutnya untuk memberikan gerakan pada setiap poster film ketika tombol diklik dengan cara kode JavaScript yang ada di bawah tombol Next dan Previous. Efek animasi dilakukan dengan cara mengubah nilai property transform untuk setiap poster yang dapat dilihat pada method rotateCarousel.

6

Bootstrap 4

Pada sub bab ini diberikan langkah-langkah membuat layout atau antarmuka halaman website dengan memanfaatkan framework CSS yaitu Bootstrap 4. Framework ini telah menyediakan class style CSS yang dapat langsung digunakan untuk membuat antarmuka yang umum digunakan pada suatu website. Hal ini membuat web designer atau developer dapat mempercepat dan mempermudah pekerjaannya.

Pendahuluan

Pada sub bab ini dijelaskan tantangan design antarmuka halaman web yang menjadi latarbelakang diperlukannya penggunaan Bootstrap. Kemudian dilanjutkan tujuan utama yang harus dicapai oleh web designer atau developer.

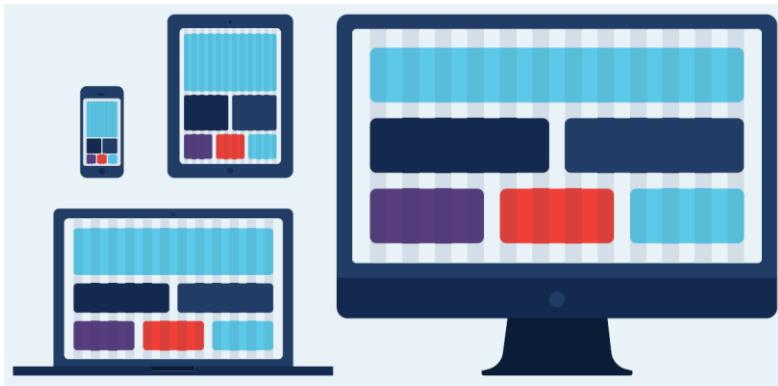
Tantangan

Seperti kita ketahui bahwa aplikasi web dapat diakses dengan menggunakan web browser. Saat ini telah tersedia bermacam web browser dari Internet Explorer, Edge, Firefox, Chrome, Safari dan lain-lain. Halaman web yang telah dibuat oleh seorang web designer atau developer dapat saja terlihat seperti yang diinginkan pada Edge namun kemungkinan ada sedikit perbedaan jika dilihat pada web browser lain.

Agar halaman web yang dibuat dapat terlihat sama pada setiap web browser yang ada maka seorang web designer atau developer harus mengujinya pada setiap web browser tersebut. Jika terjadi perbedaan maka perlu dilakukan perubahan pada kode HTML dan CSS agar dapat diterjemahkan oleh setiap web browser untuk menghasilkan antarmuka yang sama.

Hal lain yang menjadi tantangan adalah jenis perangkat yang digunakan untuk mengakses halaman web seperti terlihat pada Gambar 135, yaitu:

- Komputer seperti PC atau laptop.
- Tablet dengan berbagai ukuran layar.
- Smartphone dengan berbagai ukuran layar.



Gambar 135. Jenis perangkat.

Pada umumnya halaman web didesain dalam format landscape karena dulu halaman web di akses dari PC atau laptop. Pada halaman ini umumnya menu ditampilkan dalam bentuk horisontal di atas atau vertikal di sisi sebelah kiri atau kanan. Pada bagian atas umumnya ditemukan logo dan banner yang dipasang secara horisontal. Dan isi (content) halaman dapat berisi teks dan gambar.

Namun untuk perangkat smartphone, halaman web ditampilkan secara potrait dengan ukuran layar yang jauh lebih kecil jika dibandingkan ukuran layar PC atau laptop. Hal ini mempengaruhi desain halaman web untuk di akses pada perangkat smartphone. Menu tidak bisa selalu ditampilkan secara horisontal atau vertikal karena akan menghabiskan ruang pada halaman. Umumnya menu ditampilkan secara vertikal dan hanya ditampilkan jika user memerlukannya saja. Jumlah dan ukuran gambar yang ditampilkan juga dikurangi dan disesuaikan agar sisa ruang diisi dengan lebih banyak teks dan mengurangi ukuran data yang diunduh oleh perangkat smartphone.

Agar halaman web dapat menyesuaikan dengan berbagai ukuran layar, maka web designer atau developer harus menambahkan kode agar halaman web mengetahui perangkat apa yang sedang mengaksesnya kemudian secara otomatis halaman web tersebut menyesuaikan antarmuka berdasarkan ukuran layar perangkat tersebut.

Dengan menguasai HTML, CSS dan JavaScript yang sebagian telah diberikan di atas maka masalah-masalah yang telah disebutkan di atas tentu dapat diselesaikan. Namun menyelesaikan hal tentu tidak mudah dan akan memakan waktu terlebih lagi bagi yang baru saja belajar membuat halaman web. Namun karena hal-hal tersebut adalah tantangan dan permasalahan yang umum ditemui dalam mendesain halaman web maka tentu telah banyak solusi dan

panduan standar yang dibuat oleh para web developer yang berpengalaman yang umumnya dikenal dengan istilah framework. Salah satu framework yang banyak digunakan untuk membangun front-end aplikasi web atau halaman web adalah Bootstrap.

Bootstrap 4

Bootstrap adalah framework populer untuk membangun front-end atau antarmuka website. Saat buku ini ditulis telah tersedia Bootstrap 4.

Bootstrap mempunyai sejarah yang dimulai dari:

- Sebelum 2011 sebagai tool internal Twitter.
- Agustus 2011 dirilis sebagai open source.
- Januari 2012 Bootstrap 2 dirilis.
- Agustus 2013 Bootstrap 3 dirilis.
- Agustus 2015 Bootstrap 4 mulai dirilis dimulai versi Bootstrap 4 Alpha kemudian dilanjutkan Bootstrap 4 Beta pada Agustus 2017 dan finalisasi Bootstrap 4 pada Januari 2018.

Bootstrap 4 sebagai framework untuk membangun front-end atau antarmuka web memiliki template CSS dan JavaScript seperti untuk membuat form, typography, dropdown, popover, modal, carousels dan lain-lain. Framework ini memiliki class-class CSS yang dapat dipergunakan atau dimodifikasi sesuai keperluan, sehingga tidak perlu membuat dari awal. Sehingga waktu pengembangan website menjadi lebih singkat.

Framework ini juga dibuat berdasarkan standar pengembangan antarmuka sehingga antarmuka yang dihasilkan akan dapat diterjemahkan dengan baik oleh berbagai web browser. Selain itu juga halaman web yang dihasilkan bersifat responsif mengikuti ukuran layar perangkat yang mengaksesnya.

Installasi

Pada sub bab ini dijelaskan proses installasi Bootstrap 4 pada halaman website yang akan kita bangun. Ada beberapa cara penggunaan Bootstrap 4 untuk membangun halaman web. Salah satunya adalah dengan cara mengunduh terlebih dahulu file CSS dan JavaScript Bootstrap 4.

Kunjungi <https://getbootstrap.com/docs/4.5/getting-started/download/> untuk mengunduh Bootstrap 4. Ada dua pilihan yang dapat diunduh, yaitu:

- Compiled CSS & JS, jenis ini hanya berisi file-file CSS dan JavaScript saja.
- Source files, jenis ini selain berisi file-file CSS dan JavaScript yang diperlukan untuk membangun halaman web juga berisi dokumentasi dan contoh-contoh penggunaan Bootstrap 4. Sehingga dapat digunakan untuk melihat bagaimana antarmuka dibangun berdasarkan contoh-contoh yang ada.

Membuat Project

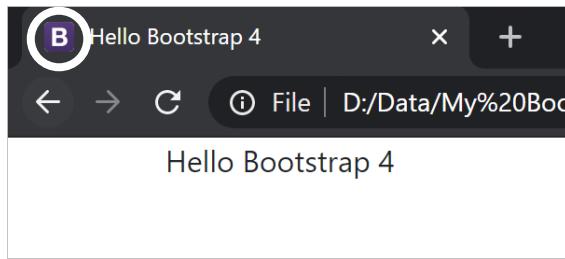
Untuk memulai menggunakan Bootstrap 4, terlebih dahulu dibuat project atau folder pada file explorer pada komputer. Kemudian ekstrak file Bootstrap jenis Compiled CSS & JSS dengan nama file bootstrap-4.5.0-dist.zip. Setelah diekstrak dapat dilihat dua folder yaitu:

1. css, berisi file-file CSS. Jika diperhatikan pada folder ini terdapat sepasang file CSS dengan nama yang mirip sebagai contoh bootstrap.css dan bootstrap.min.css. Keduanya berisi class-class CSS yang sama, perbedaannya adalah jika file bootstrap.css dibuka maka dapat dilihat setiap nama class dan property-property didalamnya ditulis per baris, sedangkan pada file bootstrap.min.css seluruh teks ditulis menjadi 1 baris saja. Hal ini bertujuan untuk meminimalkan ukuran file sesuai dengan kepanjangan dari min adalah minified.
2. js, berisi file JavaScript. Hal yang sama juga dapat dilihat folder ini terdapat file bootstrap.js dan bootstrap.min.js. Fungsi-fungsi yang terdapat pada file ini bertujuan agar komponen Bootstrap dapat bekerja. Namun fungsi-fungsi pada file-file JavaScript tersebut memerlukan fungsi dari jQuery.

Salin kedua folder tersebut ke dalam folder yang akan berisi halaman web yang akan dibangun.

Kemudian ekstrak file Bootstrap jenis Source files. Dan salin file favicon.ico ke dalam folder project yang sebelumnya telah berisi folder css dan js. File favicon.ico adalah file gambar yang biasa terlihat di samping title seperti terlihat pada lingkaran putih di Gambar 136. Icon ini dapat diganti dengan logo organisasi atau perusahaan.

Langkah selanjutnya adalah mengunduh jQuery dari alamat berikut ini <https://jquery.com/download/>. Pada saat buku ini ditulis diunduh jQuery versi 3.5.1. Simpan file jquery-3.5.1.min.js ke dalam folder js.



Gambar 136. favicon.ico.

Membuat Halaman Web

Selanjutnya adalah membuat halaman web, misalnya ditambahkan file hello.html. Maka dapat dilihat struktur file dan folder project ini seperti pada Gambar 137.

| Name | Date modified | Type |
|---------------|-------------------|-------------|
| css | 6/21/2020 3:44 PM | File folder |
| js | 6/21/2020 3:44 PM | File folder |
| B favicon.ico | 1/19/2018 2:29 AM | Icon |
| hello.html | 6/21/2020 6:08 PM | HTML File |

Gambar 137. Struktur file dan folder pada project Bootstrap.

Berikut adalah isi dari file hello.html.

```
hello.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
                                initial-scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Hello Bootstrap 4</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>

  <body>
    <div class="container">
      <h1>Greeting</h1>
```

```
    Hello Bootstrap 4
  </div>

  <!--
  Bootstrap core JavaScript
  Placed at the end of the document so the pages load faster
  -->
  <script src="js/jquery-3.5.1.min.js"></script>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Pada kode di atas dapat dilihat elemen <meta> dengan atribut name dengan nilai viewport dan content dengan nilai adalah width=device-width, initial-scale=1 yang menyatakan bahwa halaman web untuk responsif pada perangkat mobile. Karena framework Bootstrap ini mengutamakan style untuk perangkat mobile. Nilai width=device-width menyatakan halaman akan mengikuti lebar dari layar perangkat. Sedangkan nilai initial-scake=1 adalah nilai zoom level saat halaman dimuat oleh web browser.

Pada kode di atas dapat dilihat dipanggil file bootstrap.min.css di dalam elemen <head>. Selanjutnya di dalam elemen <body> dapat dilihat elemen <div> dengan atribut class bernilai container sebagai elemen pertama yang ada pada halaman web yang dibangun dengan framework Bootstrap. Selain nilai container, juga dapat digunakan nilai container-fluid.

Dan penggunaan file jquery-3.5.1.min.js diikuti dengan file bootstrap.min.js. Hal ini dikarenakan file bootstrap.min.js akan menggunakan fungsi-fungsi yang ada pada file jquery-3.5.1.min.js. Menurut panduan Bootstrap, seluruh file JavaScript diletakkan diakhir dokumen (sebelum tag penutup elemen <body>) dengan tujuan agar halaman web dapat dimuat lebih cepat.

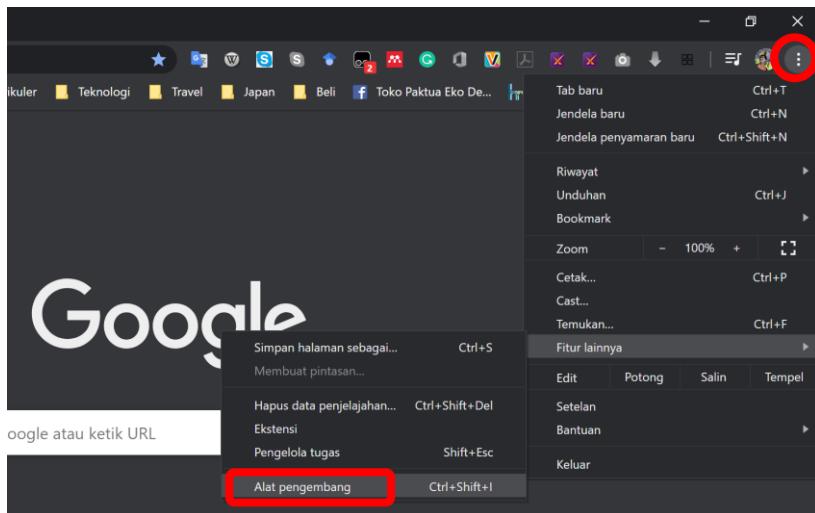
Perangkat Pengembang

Jika kita sebagai web designer atau developer disarankan untuk memeriksa bagaimana antarmuka halaman web yang telah dibangun pada berbagai web browser atau perangkat. Cara yang paling mudah dan murah untuk melakukan hal ini adalah dengan memanfaatkan tool pengembang pada web browser.

Pada sub bab ini diperlihatkan tool pengembang pada web browser Google Chrome dan Microsoft Edge.

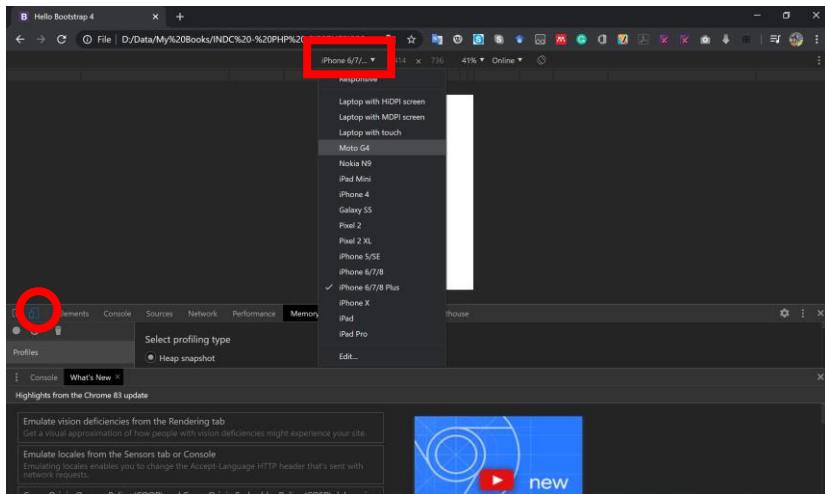
Pada Google Chrome dikenal dengan istilah Alat pengembang. Untuk menampilkan tool ini pilih tombol titik tidak di ujung kanan atas (lingkaran

merah pada Gambar 138) kemudian pada menu pilih Fitur lainnya > Alat pengembang.



Gambar 138. Menu untuk mengakses alat Pengembang pada Google Chrome.

Setelah Alat pengembang dipilih maka dapat dilihat tampilan seperti pada Gambar 139.

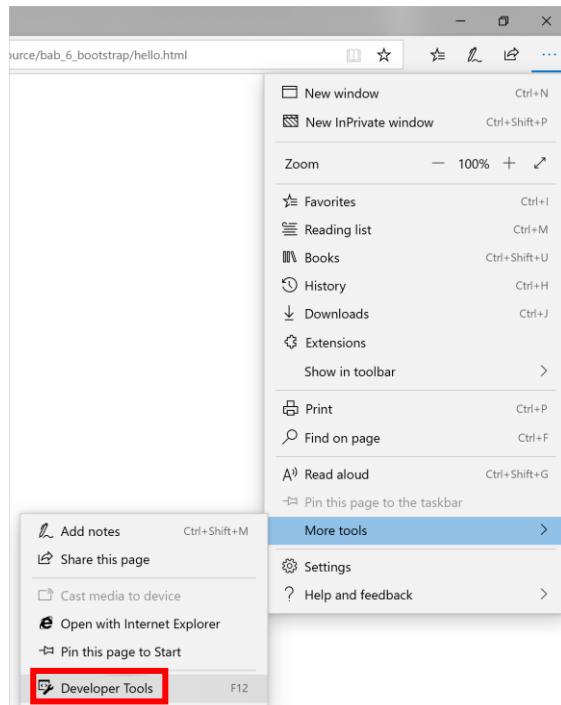


Gambar 139. Alat Pengembang pada Google Chrome

Untuk mengubah perangkat atau device yang ingin digunakan klik tombol "Toggle device toolbar" pada lingkaran merah. Kemudian dapat dipilih

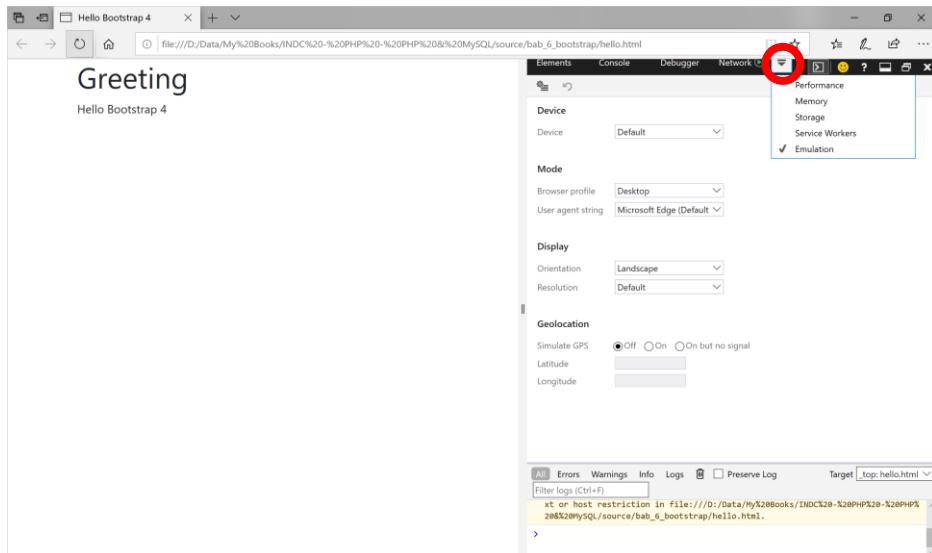
perangkat apa yg ingin digunakan pada kotak warna merah. Dengan memilih perangkat, maka ukuran layar akan menyesuaikan dengan ukuran perangkat tersebut.

Pada Microsoft Edge terdapat fitur Developer Tools yang dapat diakses dengan memilih tombol “...” kemudian pilih More tools > Developer Tools seperti pada Gambar 140 atau dengan menekan tombol F12.



Gambar 140. Menu untuk mengakses Developer tools.

Setelah menu Developer Tools dipilih maka dapat dilihat tampilan sebagai mana pada Gambar 141. Pilih tombol pada lingkarna merah, kemudian pilih Emulation dan akan ditampilkan antarmuka seperti pada sisi kanan pada gambar. Pada dropdown Device dapat dipilih perangkat yang ingin digunakan. Kemudian pada dropdown Mode dapat dipilih jenis web browser yang ingin digunakan.



Gambar 141. Developer Tools pada Microsoft Edge.

Pada web browser lain juga kemungkinan dilengkapi perangkat pengembang seperti kedua web browser di atas. Dengan adanya perangkat atau tool ini maka pengujian halaman web yang telah dibuat pada berbagai web browser dan perangkat dapat mudah dilakukan.

Penulisan Nama Class Pada Teks Di Buku Ini

Pada referensi tentang Bootstrap umumnya untuk menyebut nama class ditulis dengan cara berikut dibawah, yaitu tanda titik kemudian diikuti dengan nama class.

```
.nama_class
```

Penulisan ini hanya untuk teks di buku saja. Jadi jika pada teks buku dilihat .container, itu artinya class CSS dengan nama container. Sedangkan untuk penggunaannya pada kode HTML tetap ditulis tanpa tanda titik seperti contoh berikut.

```
<div class="container">
```

Dasar-Dasar Bootstrap

Pada sub bab ini diberikan dasar-dasar Bootstrap yang wajib diketahui. Pada bagian ini akan diberikan contoh-contoh seperti pada bab sebelumnya. Untuk membuat file-file pada contoh pada bab ini digunakan template atau struktur halaman web sebagai berikut.

```
template.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
                                initial-scale=1">
    <link rel="icon" href="favicon.ico">
    <title>Bootstrap 4: Basic</title>
    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>
  <body>
    <!-- content -->

    <!--
        Bootstrap core JavaScript
        Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery-3.5.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </body>
</html>
```

Cara Penggunaan Bootstrap

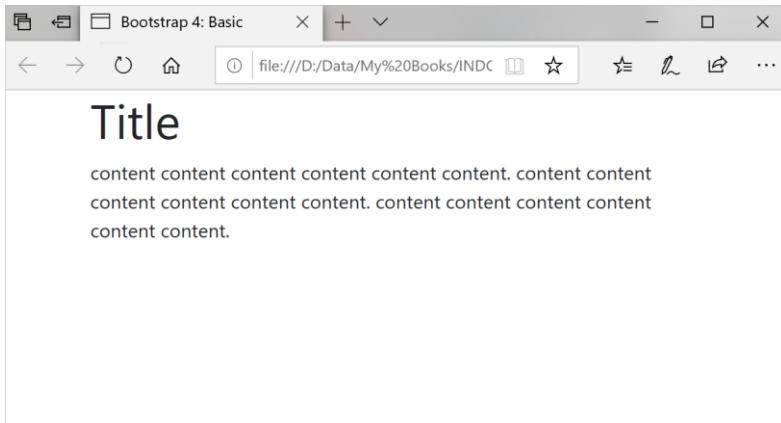
Secara singkat dan sederhana, Bootstrap berisi class-class CSS yang dapat langsung digunakan pada halaman web.

Sebagai contoh dapat dilihat pada potongan kode di bawah ini. .container adalah nama class yang sudah ada pada file bootstrap.css. Fungsi class ini bertujuan untuk membuat rangka utama aplikasi.

```
<div class="container">
  <h1>Title</h1>
  <p>
    content content content content content content.
    content content content content content content.
    content content content content content content.
  </p>
```

</div>

Dan hasilnya seperti pada Gambar 142, dapat dilihat bagaimana telah ditentukan jenis dan ukuran font yang digunakan pada teks. Selain itu juga margin sehingga area content berada di tengah-tengah layar.



Gambar 142. Penggunaan class .container.

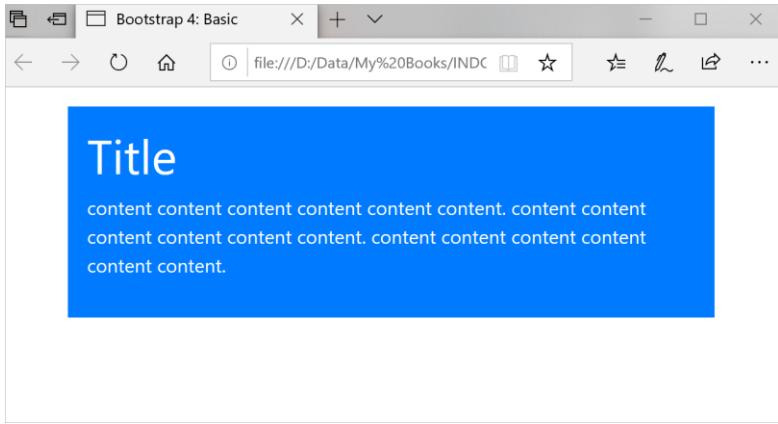
Class-class Bootstrap dapat digunakan lebih dari satu pada suatu elemen HTML seperti contoh kode di bawah ini. Pada contoh, elemen <div> ditambahkan beberapa class yang ada pada Bootstrap sebagai nilai atribut class dengan dipisahkan tanda spasi.

```
<div class="container p-3 my-3 bg-primary text-white">
    ...
</div>
```

Hasilnya dapat dilihat pada Gambar 143. Pada kode di atas selain berisi .container, atribut class juga berisi:

- .p-3, membuat padding selebar 1rem untuk semua sisi. Jika ingin membuat padding selebar 1rem untuk sisi atas saja telah tersedia class .pt-3 (yang diartikan sebagai padding top).
 - .my-3, membuat margin selebar 1rem untuk sisi atas dan bawah.
 - .bg-primary, memberi warna utama yaitu biru untuk latarbelakang.
 - .text-white, memberi style warna putih untuk teks.

Untuk melihat class-class apa saja dan kode CSS dari setiap class tersebut maka bisa melihat file bootstrap.css yang terdapat pada folder css.



Gambar 143. Penggunaan class .p-3, .my-3, .bg-primary, .text-white.

Container

Pada sub bab sebelumnya telah disinggung sedikit tentang container. Class yang digunakan untuk container adalah:

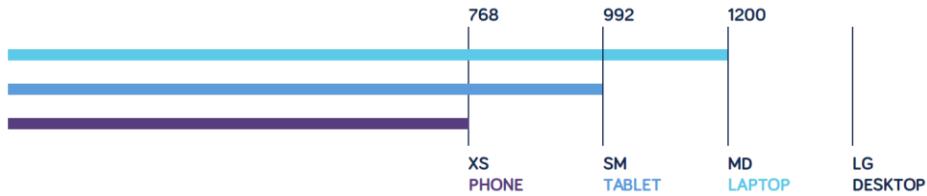
- .container, membuat elemen <div> menjadi container dengan lebar tetap.
 - .container-fluid, membuat elemen <div> menjadi container dengan lebar penuh atau width=100%.

Secara umum perbedaan kedua nilai class tersebut dapat dilihat pada Gambar 144.



Gambar 144. Perbandingan .container dan .container-fluid.

Untuk .container, ukurannya tetap namun mengikuti ukuran perangkat yang mengaksesnya seperti terlihat pada Gambar 145.



Gambar 145. Ukuran lembar .container berdasarkan perangkat.

Atau dapat juga melihat tabel berikut ini.

| | Extra Small $<567\text{px}$ | Small $\geq 576\text{px}$ | Medium $\geq 720\text{px}$ | Large $\geq 992\text{px}$ | Extra Large $\geq 1200\text{px}$ |
|------------------|--------------------------------|------------------------------|-------------------------------|------------------------------|-------------------------------------|
| max-width | 100% | 540px | 720px | 960px | 1140px |

Berikut contoh dari penggunaan .container.

```
container_1.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width,
                                initial-scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Bootstrap 4: Basic</title>

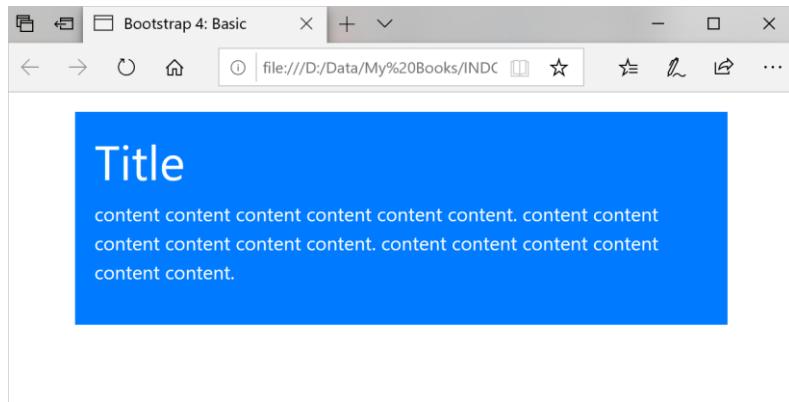
    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
  </head>

  <body>
    <div class="container p-3 my-3 bg-primary text-white">
      <h1>Title</h1>
      <p>
        content content content content content content.
        content content content content content.
        content content content content content.
      </p>
    </div>

    <!--
    Bootstrap core JavaScript
    Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery-3.5.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
  </body>
```

```
</html>
```

Hasilnya dapat dilihat pada Gambar 146.



Gambar 146. Penggunaan .container.

Selain .container juga dapat digunakan dapat digunakan class lain untuk membuat container yang responsif yaitu:

- .container-sm.
- .container-md.
- .container-lg
- .container-xl.

Perbedaan keempat class tersebut dapat dilihat pada tabel berikut.

| class | Extra Small <567px | Small ≥576px | Medium ≥720px | Large ≥992px | Extra Large ≥120px |
|---------------|-----------------------|-----------------|------------------|-----------------|-----------------------|
| .container-sm | 100% | 540px | 720px | 960px | 1140px |
| .container-md | 100% | 100% | 720px | 960px | 1140px |
| .container-lg | 100% | 100% | 100% | 960px | 1140px |
| .container-xl | 100% | 100% | 100% | 100% | 1140px |

Untuk penggunaan kode ini dapat dilihat pada contoh berikut.

```
container_2.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">
```

```
<title>Bootstrap 4: Basic</title>

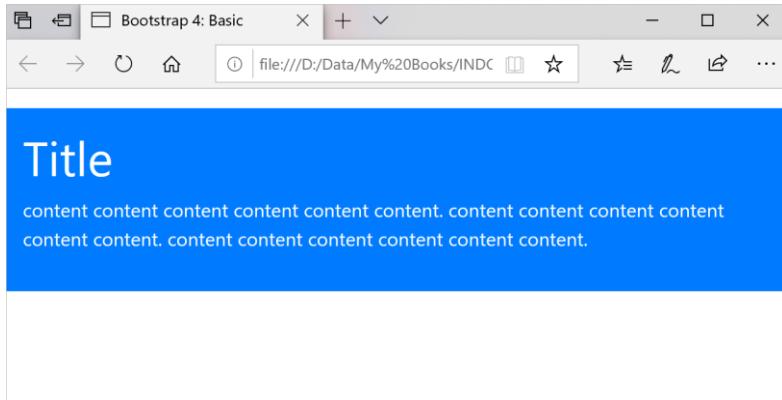
<!-- Bootstrap core CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet">
</head>

<body>
    <!-- content -->
    <div class="container-sm p-3 my-3 bg-primary text-white">
        .container-sm
    </div>
    <div class="container-md p-3 my-3 bg-primary text-white">
        .container-md
    </div>
    <div class="container-lg p-3 my-3 bg-primary text-white">
        .container-lg
    </div>
    <div class="container-xl p-3 my-3 bg-primary text-white">
        .container-xl
    </div>

    <!--
    Bootstrap core JavaScript
    Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery-3.5.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

Untuk melihat perbedaan efek dari keempat class tersebut adalah dengan meresize lebar web browser yang mengakses halaman tersebut, dari lebar layar web browser maksimal sampai minimal.

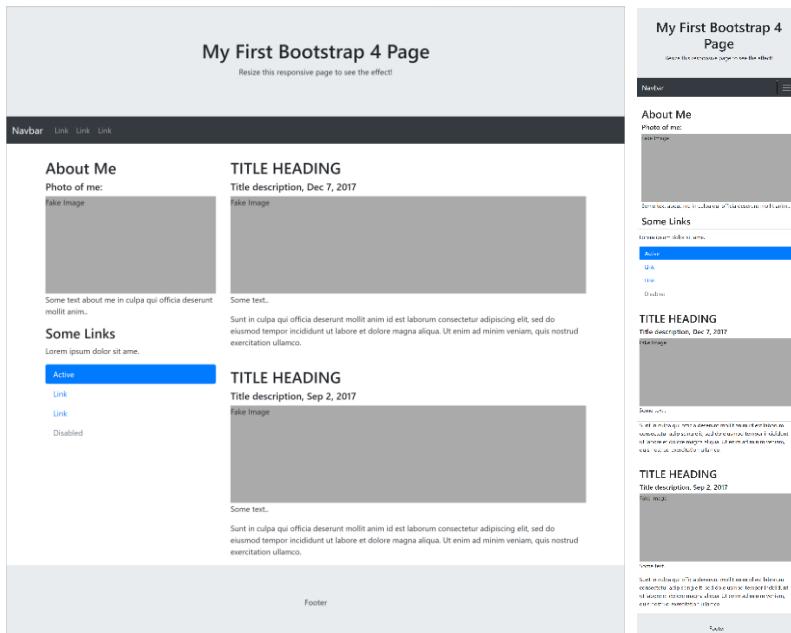
Jika .container diganti menjadi .container-fluid maka dapat dilihat antarmukanya sebagai berikut.



Gambar 147. Contoh penggunaan .container-fluid.

Grid

Setelah wadah utama dibuat dengan menggunakan .container atau .container-fluid, maka wadah tersebut siap diisi dengan gambar, teks dan lain-lain.



Gambar 148. Layout menggunakan grid.

Jika ingin membagi suatu area menjadi beberapa kolom seperti pada contoh sebelah kiri di Gambar 148, area content dibagi menjadi dua yaitu sisi kiri yang berisi About Me dan Some Links. Kemudian kolom sebelah kanan berisi Title Heading.

Agar pembagian area ini tetap membuat halaman web responsif dan menyesuaikan dengan ukuran layar perangkat maka pembagian suatu area menjadi beberapa kolom maka digunakan sistem grid Bootstrap. Sehingga jika ukuran layar yang mengakses kecil maka halaman web akan menyesuaikan dengan lebar layar seperti yang terlihat pada sebelah kanan di Gambar 148.

Sistem grid Boostrap membagi suatu area pada layar menjadi 12 bagian atau kurang, tidak lebih dari 12. Sistem grid ini memiliki 5 jenis prefix class yaitu:

- .col-, class ini ditujukan untuk perangkat dengan ukuran layar sangat kecil yaitu $< 576\text{px}$.
- .col-sm-, class ini ditujukan untuk perangkat dengan ukuran layar kecil yaitu $\leq 576\text{px}$.
- .col-md-, class ini ditujukan untuk perangkat dengan ukuran layar sedang yaitu $\leq 768\text{px}$.
- .col-lg-, class ini ditujukan untuk perangkat dengan ukuran layar sedang yaitu $\leq 992\text{px}$.
- .col-xl-, class ini ditujukan untuk perangkat dengan ukuran layar sedang yaitu $\leq 1200\text{px}$.

Cara penggunaan prefix class di atas adalah sebagai berikut. Class harus ditulis di dalam elemen `<div>` dengan class `.row`. Prefix class di atas dapat ditulis secara lengkap sebagai berikut (contoh prefix `.col-`):

- .col
- .col-1
- .col-2
- .col-3
- .col-4
- .col-5
- .col-6
- .col-7
- .col-8
- .col-9
- .col-10
- .col-11
- .col-12

Untuk class .col, ukuran lebar akan menyesuaikan dengan jumlah elemen <div> yang menggunakan .col di dalam elemen <div> yang menggunakan .row. Sebagai contoh dapat dilihat pada contoh berikut ini.

```
grid_1.html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="icon" href="favicon.ico">

    <title>Bootstrap 4: Basic - Grid</title>

    <!-- Bootstrap core CSS -->
    <link href="css/bootstrap.min.css" rel="stylesheet">
</head>

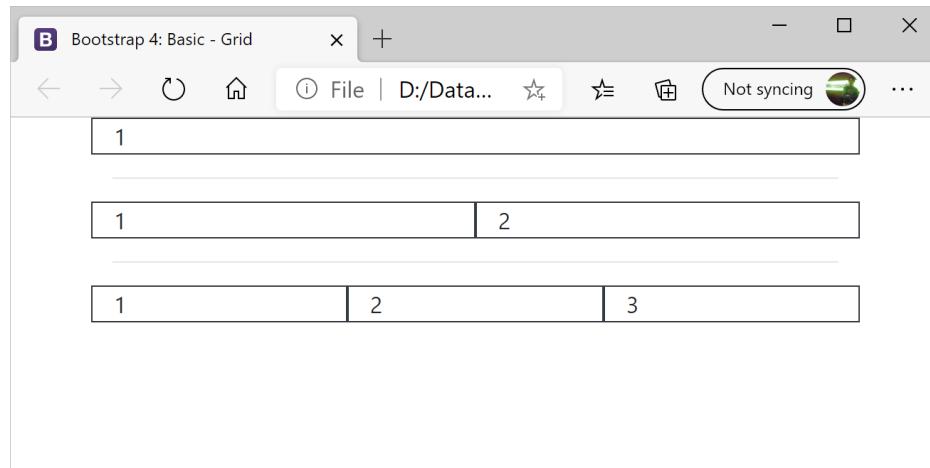
<body>
    <!-- content -->
    <div class="container">
        <div class="row">
            <div class="col border border-dark">
                1
            </div>
        </div>
        <hr>
        <div class="row">
            <div class="col border border-dark">
                1
            </div>
            <div class="col border border-dark">
                2
            </div>
        </div>
        <hr>
        <div class="row">
            <div class="col border border-dark">
                1
            </div>
            <div class="col border border-dark">
                2
            </div>
            <div class="col border border-dark">
                3
            </div>
        </div>
    </div>
</body>
```

```

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery-3.5.1.min.js"></script>
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 149. Jika elemen <div> dengan class .col hanya ada 1 di dalam elemen <div> dengan class .row, maka lebarnya akan penuh. Sedangkan jika ada dua eleman <div> dengan class .col pada elemen <div> class .row maka lebarnya terbagi dua dengan ukuran yang sama. Begitu juga jika terdapat tiga elemen <div> dengan class .col, lebarnya akan terbagi tiga dengan ukuran yang sama.



Gambar 149. Penggunaan .col.

Jumlah elemen <div> yang menggunakan class .col ini dapat berjumlah lebih dari 12 di dalam elemen <div> dengan class .row. Lebar setiap elemen <div> dengan class .col akan berukuran sama dengan lebar elemen induk dibagi jumlah elemen <div> tersebut.

Contoh prefix .col- yang lain adalah sebagai contoh kode berikut ini.

```

grid_2.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">

```

```

<link rel="icon" href="favicon.ico">

<title>Bootstrap 4: Basic - Grid</title>

<!-- Bootstrap core CSS -->
<link href="css/bootstrap.min.css" rel="stylesheet">
</head>

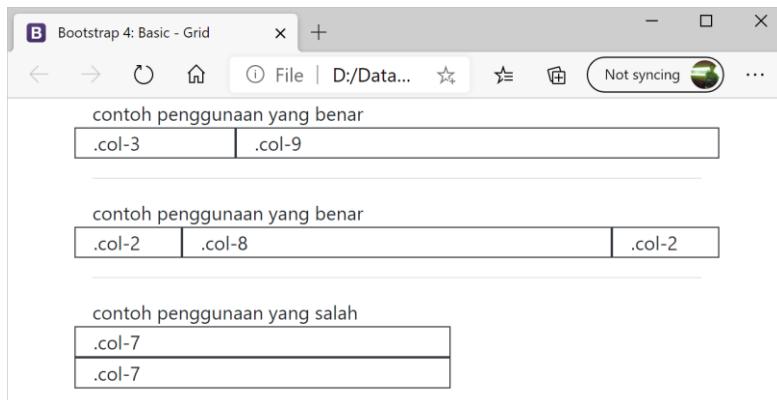
<body>
    <!-- content -->
    <div class="container">
        contoh penggunaan yang benar
        <div class="row">
            <div class="col-3 border border-dark">
                .col-3
            </div>
            <div class="col-9 border border-dark">
                .col-9
            </div>
        </div>
        <hr>
        contoh penggunaan yang benar
        <div class="row">
            <div class="col-2 border border-dark">
                .col-2
            </div>
            <div class="col-8 border border-dark">
                .col-8
            </div>
            <div class="col-2 border border-dark">
                .col-2
            </div>
        </div>
        <hr>
        contoh penggunaan yang salah
        <div class="row">
            <div class="col-7 border border-dark">
                .col-7
            </div>
            <div class="col-7 border border-dark">
                .col-7
            </div>
        </div>
    </div>

    <!--
    Bootstrap core JavaScript
    Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery-3.5.1.min.js"></script>
    <script src="js/bootstrap.min.js"></script>

```

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

Hasilnya dapat dilihat pada Gambar 150. Seperti yang telah dijelaskan di atas bawah sistem grid pada Bootstrap membagi layar menjadi 12 bagian. namun dengan prefix .col-, yang mesti diperhatikan adalah jumlah angka setelah .col- yang digunakan harus berjumlah 12.



Gambar 150. Contoh penggunaan .col-

Sebagai contoh, .col-3 dan .col-9 dimana penjumlahan 3 dan 9 adalah 12. Contoh berikutnya adalah .col-2, col-8 dan .col-2 dimana penjumlahan angka 2, 8 dan 2 adalah 12.

Sedangkan contoh yang salah adalah penggunaan .col-7 dan .col-7 dimana jumlah angkanya adalah 14 yang mana lebih besar daripada 12. Karena kesalahan ini dapat dilihat hasilnya kotak tidak tersusun secara horisontal.

Dari contoh-contoh di atas, dapat dilihat bahwa prefix .col- ini membuat elemen-elemennya tersusun secara horisontal untuk berbagai ukuran layar perangkat. Hal ini akan berbeda jika dibandingkan dengan prefix yang lain seperti:

- .col-sm-
- .col-md-
- .col-lg-
- .col-xl-

Keempat prefix di atas tetap membagi layar menjadi 12 namun memiliki lebar minimal sesuai dengan penjelasan sebelumnya. Sebagai contoh prefix .col-sm- adalah seperti kode berikut.

```
grid_3.html
```

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="icon" href="favicon.ico">

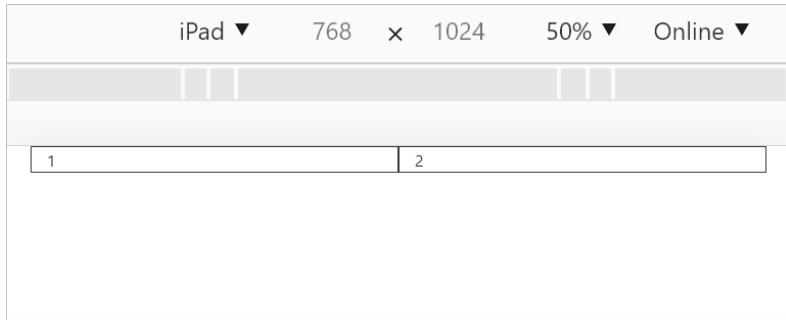
        <title>Bootstrap 4: Basic - Grid</title>

        <!-- Bootstrap core CSS -->
        <link href="css/bootstrap.min.css" rel="stylesheet">
    </head>

    <body>
        <!-- content -->
        <div class="container">
            <div class="row">
                <div class="col-sm border border-dark">
                    1
                </div>
                <div class="col-sm border border-dark">
                    2
                </div>
            </div>
        </div>

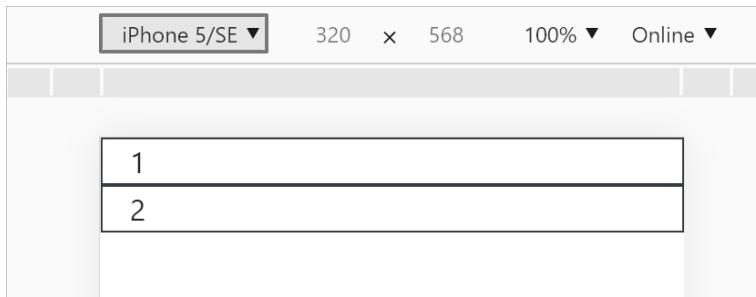
        <!--
        Bootstrap core JavaScript
        Placed at the end of the document so the pages load faster
        -->
        <script src="js/jquery-3.5.1.min.js"></script>
        <script src="js/bootstrap.min.js"></script>
    </body>
</html>
```

Kode pada bagian content membagi layar menjadi lebar sama besar seperti pada Gambar 151 jika lebar layar masih di atas dari lebar minimal yang diset pada .col-sm.



Gambar 151. Antarmuka elemen .col-sm pada layar lebar.

Sedangkan jika halaman web di atas diakses oleh perangkat dengan lebar layar lebih kecil dari lebar minimal yang diset pada .col-sm maka akan dapat dilihat hasilnya seperti pada Gambar 152.



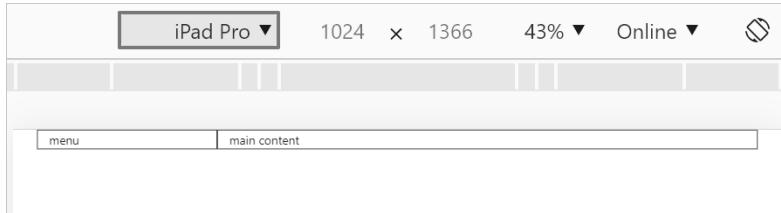
Gambar 152. Antarmuka elemen .col-sm pada perangkat layar sempit.

Seperti halnya prefix .col, prefix .col-sm juga dapat ditulis dengan menambahkan angka 1 sampai dengan 12 dibelakangnya. Berikut adalah penggalan kode penggunaan .col-sm-3 dan .col-sm-9. Seperti yang telah dijelaskan sebelumnya jumlah angka di belakang class jangan melebihi nilai 12.

```
grid_4.html
<!-- content -->
<div class="container">
  <div class="row">
    <div class="col-sm-3 border border-dark">
      menu
    </div>
    <div class="col-sm-9 border border-dark">
      main content
    </div>
  </div>
</div>
...

```

Jika halaman web diakses oleh perangkat berlayar lebar maka hasilnya dapat dilihat pada Gambar 153.



Gambar 153. Antarmuka elemen dengan .col-sm-3 dan .col-sm-9 pada perangkat layar lebar.

Sedangkan jika halaman web diakses pada perangkat berlayar kecil maka hasilnya dapat dilihat pada Gambar 154.



Gambar 154. Antarmuka elemen dengan .col-sm-3 dan .col-sm-9 pada perangkat layar kecil.

Layout Website

Berdasarkan dasar-dasar Bootstrap di atas, maka telah dapat dibuat layout website seperti pada Gambar 148 namun tanpa menu. Berikut adalah kode yang digunakan untuk membuat tampilan seperti gambar tersebut.

```
layout_1.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">
  <title>Bootstrap 4: Basic - Grid</title>
```

```

<!-- Bootstrap core CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
<style>
    .fakeimg {
        height: 200px;
        background: #aaa;
    }
</style>
</head>

<body>
    <!-- main content - start -->
    <!-- header -->
    <div class="jumbotron text-center" style="margin-bottom:0">
        <h1>Header - Bootstrap 4</h1>
        <p>Resize this responsive page to see the effect!</p>
    </div>

    <!-- content -->
    <div class="container" style="margin-top:30px">
        <div class="row">
            <div class="col-sm-4">
                <h2>About Me</h2>
                <h5>Photo of me:</h5>
                <div class="fakeimg">Fake Image</div>
                <p>Some text about me.</p>
                <h3>Some Links</h3>
                <ul class="nav nav-pills flex-column">
                    <li class="nav-item">
                        <a class="nav-link" href="#">Link 1</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Link 2</a>
                    </li>
                    <li class="nav-item">
                        <a class="nav-link" href="#">Link 3</a>
                    </li>
                </ul>
                <hr class="d-sm-none">
            </div>

            <div class="col-sm-8">
                <h2>TITLE HEADING 1</h2>
                <h5>Title description</h5>
                <div class="fakeimg">Fake Image</div>
                <p>Some text...</p>
                <br>
                <h2>TITLE HEADING 2</h2>
                <h5>Title description</h5>
                <div class="fakeimg">Fake Image</div>
            </div>
        </div>
    </div>

```

```
        <p>Some text..</p>
    </div>
</div>
</div>

<!-- footer -->
<div class="jumbotron text-center" style="margin-bottom:0">
    <p>Footer</p>
</div>

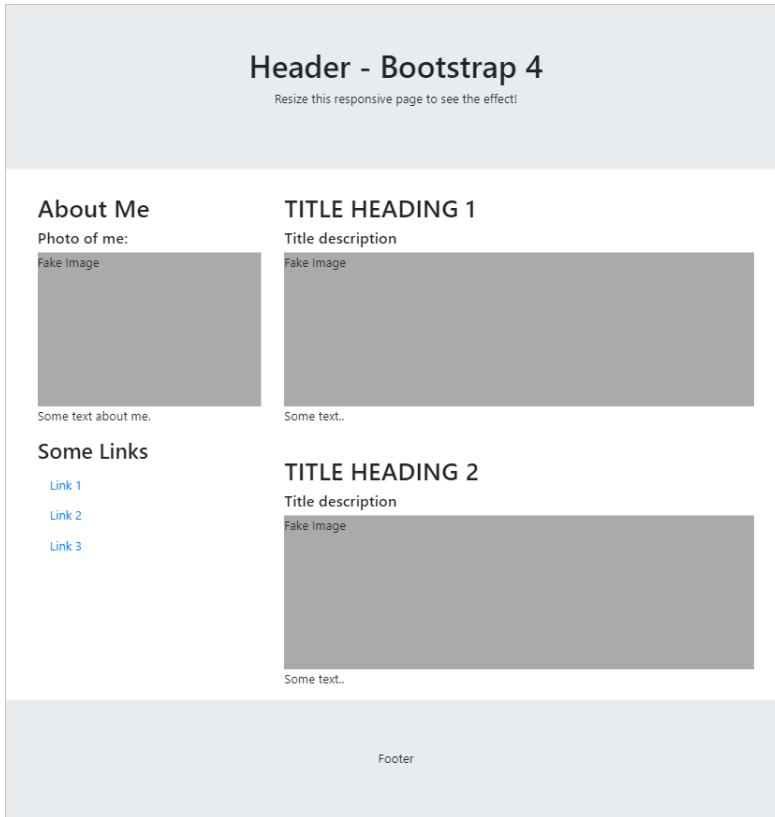
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
</body>
</html>
```

Dari kode di atas dapat dilihat pada blok “main content” terdapat tiga komponen yaitu:

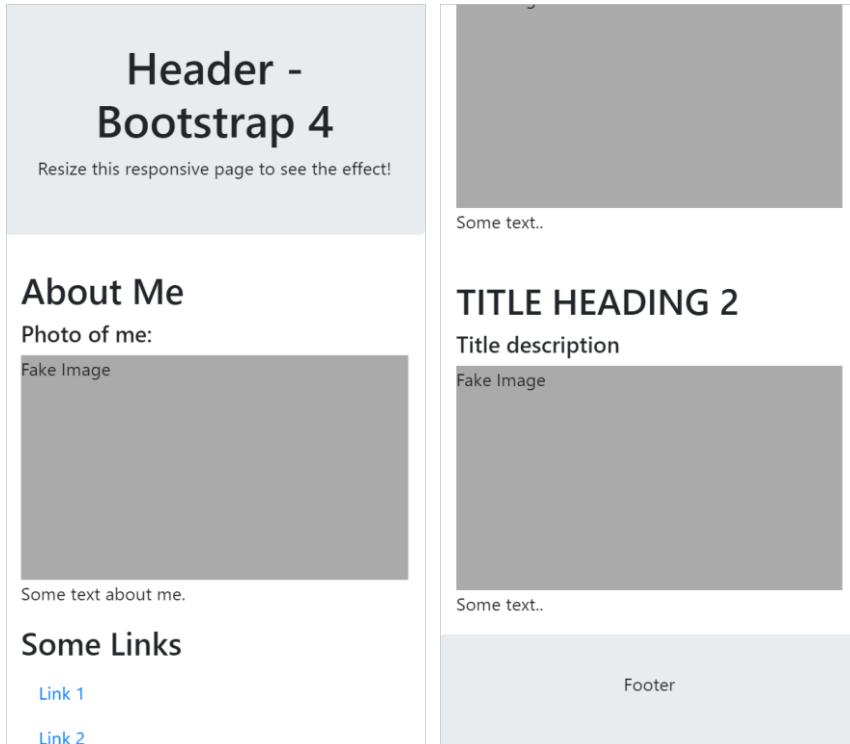
- Header.
- Content yang menggunakan grid dengan dua kolom.
- Footer.

Antarmuka kode di atas jika dilihat pada perangkat dengan ukuran layar besar dapat dilihat seperti pada Gambar 155.



Gambar 155. Antarmuka layout pada perangkat dengan ukuran layar besar.

Sedangkan jika halaman web tersebut diakses pada perangkat dengan ukuran layar kecil maka antarmukanya adalah seperti pada Gambar 156.



Gambar 156. Antarmuka layout pada perangkat dengan ukuran layar kecil.

Dari contoh kode dan gambar di atas dapat dengan mudah untuk membuat antarmuka yang responsif mengikuti dengan ukuran layar dari perangkat yang digunakan.

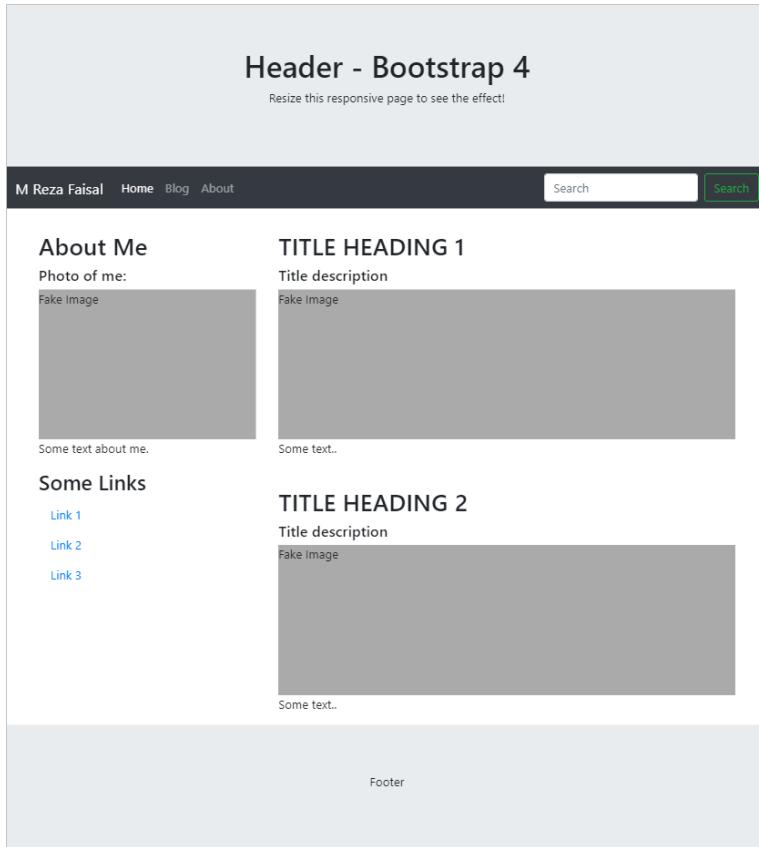
Navigasi

Komponen utama yang umum ada pada halaman website adalah menu atau navigasi. Bootstrap juga telah menyiapkan style atau css yang siap digunakan untuk membuat navigasi.

Berikut adalah template Bootstrap untuk kode navigasi yang dapat digunakan.

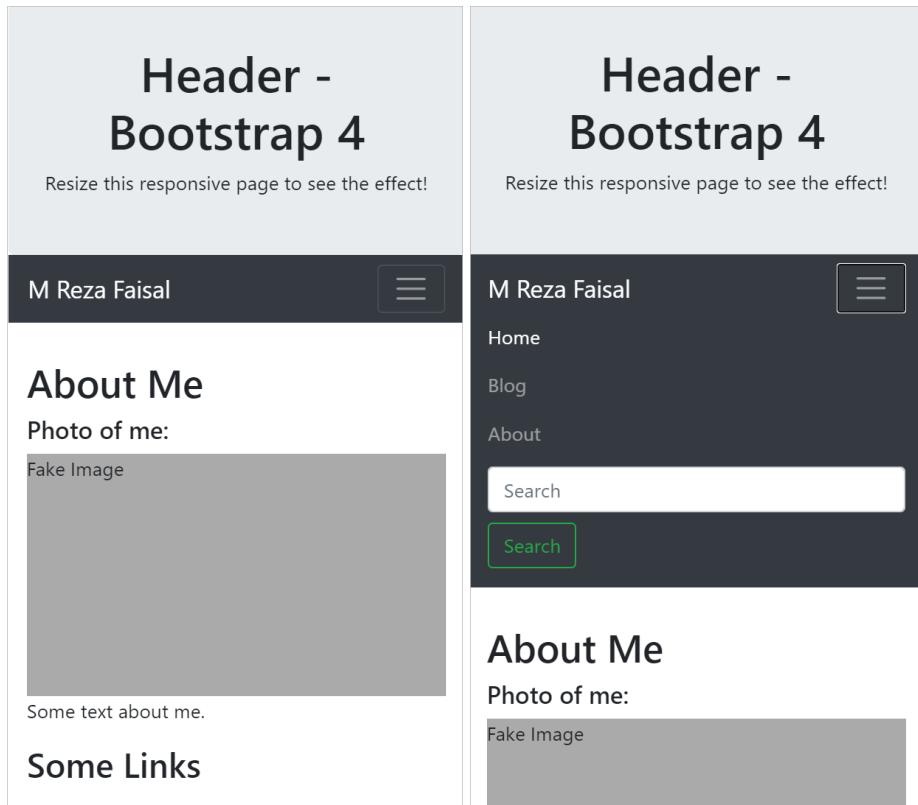
```
...
<!-- nav -->
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
    <a class="navbar-brand" href="#">M Reza Faisal</a>
    <button class="navbar-toggler" type="button"
        data-toggle="collapse"
        data-target="#navbarCollapse"
        aria-controls="navbarCollapse"
        aria-expanded="false" aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Blog</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">About</a>
            </li>
            </ul>
            <form class="form-inline mt-2 mt-md-0">
                <input class="form-control mr-sm-2"
                    type="text" placeholder="Search" aria-label="Search">
                <button class="btn btn-outline-success my-2 my-sm-0"
                    type="submit">Search</button>
            </form>
        </div>
    </nav>
...
```

Kode ini dapat diletakkan di atas area header jika ingin memiliki navigasi di bagian paling atas halaman web. Atau dapat juga diletakkan di antara header dan content sehingga hasilnya dapat dilihat seperti pada Gambar 157.



Gambar 157. Antarmuka navigasi pada perangkat dengan ukuran layar besar.

Navigasi ini juga bersifat responsif, jika halaman web ini diakses oleh perangkat dengan ukuran layar kecil maka navigasi akan terlihat seperti pada Gambar 158. Pada gambar tersebut dapat dilihat perubahan navigasi dimana yang terlihat hanya title dari navigasi dan tombol dengan garis tiga horisontal. Jika tombol tersebut diklik maka akan terbuka menu seperti pada gambar di sebelah kanan.



Gambar 158. Antarmuka navigasi pada perangkat dengan ukuran kecil.

Jika diperhatikan pada contoh kode dan gambar navigasi di atas maka dapat dilihat navigasi ini dimulai dengan menggunakan tag `<nav>` seperti pada potongan kode berikut.

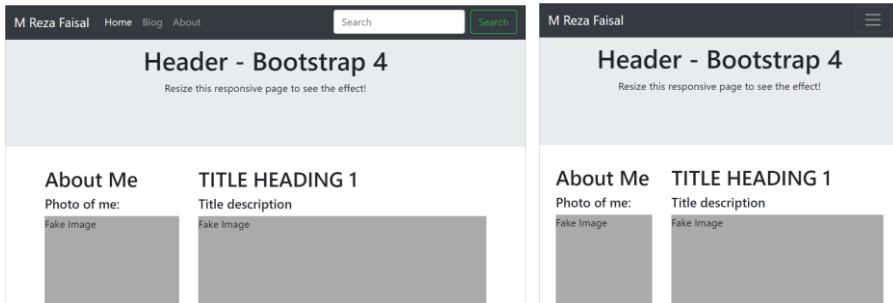
```
<nav class="navbar navbar-expand-md navbar-dark bg-dark">  
...  
</nav>
```

Pada tag `<nav>` di atas dapat dilihat class-class Bootstrap yang digunakan. Class `navbar`, `navbar-expand-md` digunakan untuk membuat tag `<nav>` menjadi navigasi. Class `navbar-dark` dan `bg-dark` digunakan untuk mengatur warna tulisan dan latarbelakang dari navigasi.

Pada tag `<nav>` juga bisa ditambahkan class lain seperti potongan kode berikut.

```
<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">  
...  
</nav>
```

Pada kode ini dapat dilihat penggunaan class fixed-top yang membuat navigasi selalu berada di atas. Walaupun halaman digeser ke bawah, navigasi akan tetap terlihat seperti pada Gambar 159.



Gambar 159. Navigasi posisi tetap di atas.

Di dalam tag <nav> ini terdapat beberapa komponen utama. Yang pertama adalah title yang berada ditampilkan pada bagian paling kiri. Pada contoh di atas title adalah hyperlink dengan teks adalah “M Reza Faisal” seperti terlihat pada potongan kode di bawah ini.

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
  <a class="navbar-brand" href="#">M Reza Faisal</a>
  . . .
</nav>
```

Bagian ini dapat berupa teks atau gambar. Komponen utama berikutnya adalah tombol yang terlihat pada potongan kode berikut.

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
  . . .
  <button class="navbar-toggler" type="button"
    data-toggle="collapse" data-target="#navbarCollapse"
    aria-controls="navbarCollapse" aria-expanded="false"
    aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  . . .
</nav>
```

Tombol ini akan terlihat ketika diakses oleh perangkat dengan layar kecil. Tombol berada pada bagian paling kanan dari navigasi dengan bentuk tiga buah garis horizontal seperti yang terlihat pada Gambar 158. Ketika tombol diklik maka akan dapat dilihat hyperlink dari navigasi. Tombol ini dapat diklik dikarenakan penggunaan fungsi JavaScript dari file-file berikut yang dipanggil pada halaman website:

- jquery.min.js
- bootstrap.min.js
- popper.min.js

komponen berikutnya adalah hyperlink pada navigasi dan input pencarian yang dapat dilihat pada potongan kode berikut.

```
<nav class="navbar navbar-expand-md navbar-dark bg-dark">
    . . .
    <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home</a>
            </li>
            . . .
        </ul>
        <form class="form-inline mt-2 mt-md-0">
            <input class="form-control mr-sm-2" type="text"
                placeholder="Search" aria-label="Search">
            <button class="btn btn-outline-success my-2 my-sm-0"
                type="submit">Search</button>
        </form>
    </div>
</nav>
```

Bekerja dengan Table

Sub bab ini memberikan beberapa contoh penerapan style Bootstrap 4 untuk table dengan cara memberikan class-class yang telah disediakan Bootstrap 4 untuk mengisi nilai atribut class pada tag <table>.

Untuk memberi style pada table adalah dengan cara memberi .table pada table seperti pada contoh kode berikut ini.

```
table_1.html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="icon" href="favicon.ico">

        <title>Bootstrap 4: Basic - Table</title>
```

```

<!-- Bootstrap core CSS -->
<link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>
    <!-- main content - start -->
    <table class="table">
        <tr>
            <th>No</th>
            <th>Nama</th>
            <th>Alamat</th>
        </tr>
        <tr>
            <td>1</td>
            <td>Nama 1</td>
            <td>Alamat 1</td>
        </tr>
        <tr>
            <td>2</td>
            <td>Nama 2</td>
            <td>Alamat 2</td>
        </tr>
        <tr>
            <td>3</td>
            <td>Nama 3</td>
            <td>Alamat 3</td>
        </tr>
    </table>
    <!-- main content - end -->
    <!--
        Bootstrap core JavaScript
        Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/popper.min.js"></script>
</body>
</html>

```

Hasilnya adalah seperti pada Gambar 160.

| No | Nama | Alamat |
|----|--------|----------|
| 1 | Nama 1 | Alamat 1 |
| 2 | Nama 2 | Alamat 2 |
| 3 | Nama 3 | Alamat 3 |

Gambar 160. Table dengan style .table.

Dari contoh di atas dapat dilihat dengan hanya memberikan .table pada tag <table> seperti pada potongan kode di bawah ini.

```
<table class="table">
. . .
</table>
```

Selain itu juga dapat ditambahkan class .table-striped untuk membuat perbedaan warna setiap baris pada data seperti potongan kode berikut.

```
<table class="table table-striped">
. . .
</table>
```

Hasilnya akan membuat table seperti pada Gambar 161.

A screenshot of a web browser window titled "Bootstrap 4: Basic - Table". The table has three columns: "No", "Nama", and "Alamat". The rows are styled with alternating background colors: white for the first row, light gray for the second, and white for the third. The data in the table is as follows:

| No | Nama | Alamat |
|----|--------|----------|
| 1 | Nama 1 | Alamat 1 |
| 2 | Nama 2 | Alamat 2 |
| 3 | Nama 3 | Alamat 3 |

Gambar 161. Table dengan class .table-striped.

Sedangkan untuk membuat table di atas memiliki border dapat menggunakan class .table-bordered.

```
<table class="table table-striped table-bordered">  
  ...  
</table>
```

Hasilnya dapat dilihat pada Gambar 162.

A screenshot of a web browser window titled "Bootstrap 4: Basic - Table". The table has three columns: "No", "Nama", and "Alamat". Each cell in the table is enclosed in a thin black border. The data in the table is the same as in Gambar 161:

| No | Nama | Alamat |
|----|--------|----------|
| 1 | Nama 1 | Alamat 1 |
| 2 | Nama 2 | Alamat 2 |
| 3 | Nama 3 | Alamat 3 |

Gambar 162. Table dengan class .table-bordered.

Class lain yang dapat ditambahkan adalah .table-hover. Class ini membuat baris yang ditunjuk oleh cursor mouse berubah warna. Dari contoh-contoh di atas dapat dilihat beberapa class yang dapat digunakan untuk memberikan style pada table. Selain itu masih banyak class lain yang dapat digunakan.

Membuat Form

Sub bab ini memberikan penjelasan dan contoh-contoh implementasi class-class Bootstrap 4 untuk membuat antarmuka form pada halaman web.

Tipe Layout Form

Bootstrap 4 menyediakan dua tipe layout form yaitu:

- Stacked form.
- Inline form.

Untuk mengetahui perbedaan keduanya dapat dilihat pada contoh kode dan hasil antarmuka di bawah ini.

Berikut ini adalah contoh tipe layout form stacked.

```
form 1.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Bootstrap 4: Basic - Form</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>

  <body>
    <!-- main content - start -->
    <div class="container">
      <form>
        <label for="email">Email address:</label>
        <input type="email" class="form-control"
               placeholder="Enter email" id="email">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control"
               placeholder="Enter password" id="pwd">
        <div class="form-check">
          <label class="form-check-label">
            <input class="form-check-input"
                  type="checkbox"> Remember me
          </label>
        </div>
      </form>
    </div>
  </body>
</html>
```

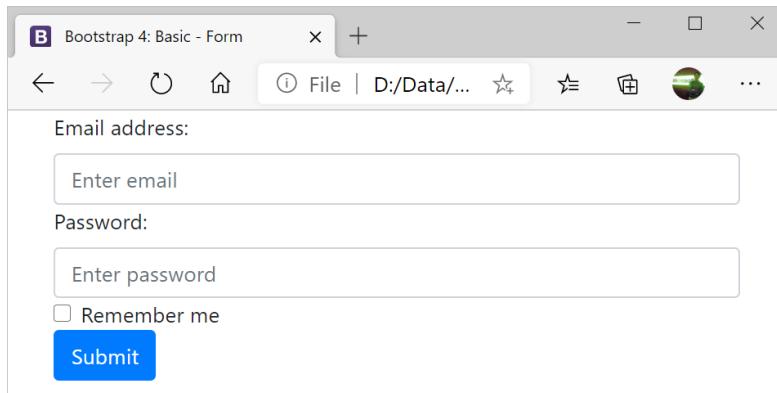
```

        </label>
    </div>
    <button type="submit" class="btn btn-primary">
        Submit
    </button>
</form>
</div>
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
</body>
</html>

```

Hasilnya dapat dilihat pada Gambar 163.



Gambar 163. Tipe layout form – stacked form.

Dan berikut ini adalah contoh kode dan tampilan dari tipe layout inline form.

```

form_2.html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="icon" href="favicon.ico">
    <title>Bootstrap 4: Basic - Form</title>
    <!-- Bootstrap core CSS -->

```

```

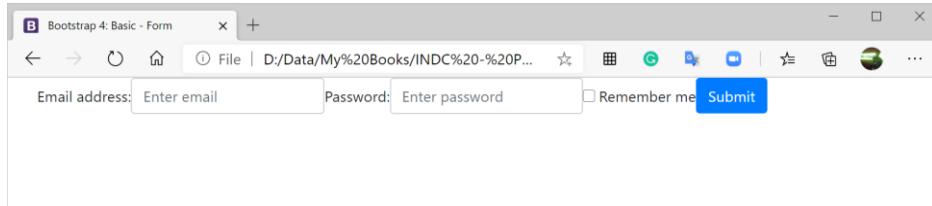
<link rel="stylesheet" href="css/bootstrap.min.css">
</head>

<body>
    <!-- main content - start -->
    <div class="container">
        <form class="form-inline">
            <label for="email">Email address:</label>
            <input type="email" class="form-control"
                   placeholder="Enter email" id="email">
            <label for="pwd">Password:</label>
            <input type="password" class="form-control"
                   placeholder="Enter password" id="pwd">
            <div class="form-check">
                <label class="form-check-label">
                    <input class="form-check-input"
                           type="checkbox"> Remember me
                </label>
            </div>
            <button type="submit" class="btn btn-primary">
                Submit
            </button>
        </form>
    </div>
    <!-- main content - end -->

    <!--
        Bootstrap core JavaScript
        Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/popper.min.js"></script>
</body>
</html>

```

Hasilnya adalah seperti pada Gambar 164.



Gambar 164. Tipe layout form – inline form.

Dari kedua gambar di atas dapat dilihat jika layout stacked membuat susunan elemen-elemen form seperti input text, checkbox dan tombol tersusun ke

bawah. Sedangkan layout inline membuat elemen-elemen tersebut tersusun ke samping.

Dari kedua contoh di atas diketahui bahwa secara default Bootstrap menggunakan layout stacked, hal ini terlihat dari elemen <form> yang tidak menggunakan class dari Bootstrap 4. Namun untuk membuat form dengan layout inline diperlukan pemberikan class .form-inline pada elemen <form>.

Validasi Form

Untuk membantu validasi pengisian form, Bootstrap 4 menyediakan beberapa class yang dapat digunakan diantaranya adalah:

- .was-validated, class ini akan menampilkan pesan kesalahan pada elemen input yang harus diisi. Jika input telah diisi maka pesan kesalahan akan digantikan dengan pesan valid.
- .need-validation, input tidak akan menampilkan pesan kesalahan saat awal ditampilkan. Setelah tombol submit diklik dan jika ditemukan ada input yang belum valid maka pesan kesalahan akan ditampilkan. Untuk menggunakan class ini diperlukan fungsi JavaScript yang dieksekusi ketika tombol diklik.

Kedua class ini ditambahkan pada nilai atribut class di elemen <form>.

Berikut adalah contoh kode dan antarmuka penggunaan class .was-validated.

```
form_3.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Bootstrap 4: Basic - Form</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>

  <body>
    <!-- main content - start -->
    <div class="container">
      <form class="was-validated">
        <div class="form-group">
```

```

        <label for="uname">Username:</label>
        <input type="text" class="form-control"
               id="uname" placeholder="Enter username"
               name="uname" required>
        <div class="valid-feedback">Valid.</div>
        <div class="invalid-feedback">
            Please fill out this field.
        </div>
    </div>
    <div class="form-group">
        <label for="pwd">Password:</label>
        <input type="password" class="form-control"
               id="pwd" placeholder="Enter password"
               name="pswd" required>
        <div class="valid-feedback">Valid.</div>
        <div class="invalid-feedback">
            Please fill out this field.
        </div>
    </div>
    <div class="form-group form-check">
        <label class="form-check-label">
            <input class="form-check-input"
                   type="checkbox" name="remember" required>
            Remember me.
        <div class="valid-feedback">Valid.</div>
        <div class="invalid-feedback">
            Check this checkbox to continue.
        </div>
        </label>
    </div>
    <button type="submit" class="btn btn-primary">
        Submit
    </button>
</form>
</div>
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
</body>
</html>
```

Dari kode di atas kita melihat potongan kode sebagai berikut dimana atribut class pada elemen <form> menggunakan class .was-validated.

```
<form class="was-validated">
```

```
...  
</form>
```

Dan pada Gambar 165 dapat dilihat antarmuka validasi form dari penerapan class ini.

The screenshot shows a web browser window titled "Bootstrap 4: Basic - Form". The form contains two input fields: "Username" and "Password", both of which have red borders and error messages: "Please fill out this field." A checkbox labeled "Remember me." also has an error message: "Check this checkbox to continue.". A blue "Submit" button is at the bottom. The browser's address bar shows "D:/Data/...".

Gambar 165. Pesan kesalahan pada validasi dengan class .was-invalid.

Saat form ditampilkan pada web browser, telah dapat dilihat pesan-pesan kesalahan pada elemen-elemen yang perlu diverifikasi. Jika kedua input text dan checkbox diisi maka warna dan pesan yang ditampilkan menjadi seperti pada Gambar 166.

The screenshot shows a web browser window titled "Bootstrap 4: Basic - Form". The form fields now have green borders and green checkmarks. Below each field, there is a green "Valid." message. The "Remember me." checkbox is checked, and its message "Valid." is also green. The "Submit" button is blue. The browser's address bar shows "D:/Data/...".

Gambar 166. Pesan sukses pada validasi dengan class .was-valid.

Kode berikut ini adalah contoh penerapan class .need-validation untuk validasi form.

```
form_4.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Bootstrap 4: Basic - Form</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>

  <body>
    <!-- main content - start -->
    <div class="container">
      <form class="needs-validation" novalidate>
        <div class="form-group">
          <label for="uname">Username:</label>
          <input type="text" class="form-control"
                 id="uname" placeholder="Enter username"
                 name="uname" required>
          <div class="valid-feedback">Valid.</div>
          <div class="invalid-feedback">
            Please fill out this field.
          </div>
        </div>
        <div class="form-group">
          <label for="pwd">Password:</label>
          <input type="password" class="form-control"
                 id="pwd" placeholder="Enter password"
                 name="pswd" required>
          <div class="valid-feedback">Valid.</div>
          <div class="invalid-feedback">
            Please fill out this field.
          </div>
        </div>
        <div class="form-group form-check">
          <label class="form-check-label">
            <input class="form-check-input"
                  type="checkbox" name="remember" required>
            Remember me.
          <div class="valid-feedback">Valid.</div>
          <div class="invalid-feedback">
            Check this checkbox to continue.
          </div>
        </div>
      </form>
    </div>
  </body>

```

```

        </label>
    </div>
    <button type="submit" class="btn btn-primary">
        Submit
    </button>
</form>
</div>
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
<script>
    (function() {
        'use strict';
        window.addEventListener('load', function() {
            var forms =
                document.getElementsByClassName('needs-validation');
            var validation =
                Array.prototype.filter.call(forms, function(form)
            {
                form.addEventListener('submit', function(event) {
                    if (form.checkValidity() === false) {
                        event.preventDefault();
                        event.stopPropagation();
                    }
                    form.classList.add('was-validated');
                }, false);
            });
        }, false);
    })();
</script>
</body>
</html>

```

Saat halaman ini dimuat di web browser maka dapat dilihat antarmuka seperti pada Gambar 167.

The screenshot shows a web browser window with a title bar 'Bootstrap 4: Basic - Form'. The main content area contains a form with two input fields: 'Username' and 'Password', both with placeholder text 'Enter username' and 'Enter password' respectively. Below the inputs is a checkbox labeled 'Remember me.' followed by a blue 'Submit' button. A validation message 'Please fill out this field.' is displayed next to the empty password input field.

Gambar 167. Halaman form dengan validasi menggunakan class `.needs-validation` saat dimuat.

Pada kode di atas dapat dilihat terdapat tambahan kode JavaScript untuk memeriksa valid atau tidaknya pengisian form setelah tombol Submit diklik. Jika ada elemen input yang tidak diisi maka akan dapat dilihat pesan seperti pada Gambar 168.

The screenshot shows a web browser window with a title bar 'Bootstrap 4: Basic - Form'. The main content area contains a form with two input fields: 'Username' and 'Password'. The 'Username' field contains the text 'mrezafaisal' and has a green checkmark icon to its right. Below the 'Username' field is a green success message 'Valid.'. The 'Password' field is empty and has a red exclamation mark icon to its right. Below the 'Password' field is a red error message 'Please fill out this field.' A checkbox labeled 'Remember me.' is present, with a note below it saying 'Check this checkbox to continue.' A blue 'Submit' button is at the bottom of the form.

Gambar 168. Tampilan pesan kesalahan dan sukses.

Elemen Input

Dari contoh-contoh di atas, form telah berisi elemen input dan tombol. Class Bootstrap 4 yang digunakan untuk memberikan style pada elemen input adalah .form-control.

Berikut adalah contoh kode penggunaan class .form-control pada beberapa elemen input yang umum digunakan.

```
form_5.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Bootstrap 4: Basic - Form</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>

  <body>
    <!-- main content - start -->
    <div class="container">
      <form>
        <div class="form-group">
          <label for="usr">Name:</label>
          <input type="text" class="form-control"
                 id="usr">
        </div>
        <div class="form-group">
          <label for="pwd">Password:</label>
          <input type="password" class="form-control"
                 id="pwd">
        </div>
        <div class="form-group">
          <label for="comment">Comment:</label>
          <textarea class="form-control" rows="5"
                    id="comment"></textarea>
        </div>
        <div class="form-check-inline">
          <label class="form-check-label">
            <input type="checkbox"
                   class="form-check-input"
                   value="">Option 1
          </label>
        </div>
      </form>
    </div>
  </body>
</html>
```

```

        <div class="form-check-inline">
            <label class="form-check-label">
                <input type="checkbox"
                    class="form-check-input"
                    value="">Option 2
            </label>
        </div>
        <br>
        <div class="form-check-inline">
            <label class="form-check-label">
                <input type="radio"
                    class="form-check-input"
                    name="optradio">Option 1
            </label>
        </div>
        <div class="form-check-inline">
            <label class="form-check-label">
                <input type="radio"
                    class="form-check-input"
                    name="optradio">Option 2
            </label>
        </div>
        <br>
        <div class="form-group">
            <label for="sel1">Select list:</label>
            <select class="form-control" id="sel1">
                <option>1</option>
                <option>2</option>
                <option>3</option>
                <option>4</option>
            </select>
        </div>
        <div class="form-group">
            <label for="sel1">File</label>
            <input type="file"
                class="form-control-file border">
        </div>
        <button type="submit"
            class="btn btn-primary">Submit</button>
    </form>
</div>
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
</body>

```

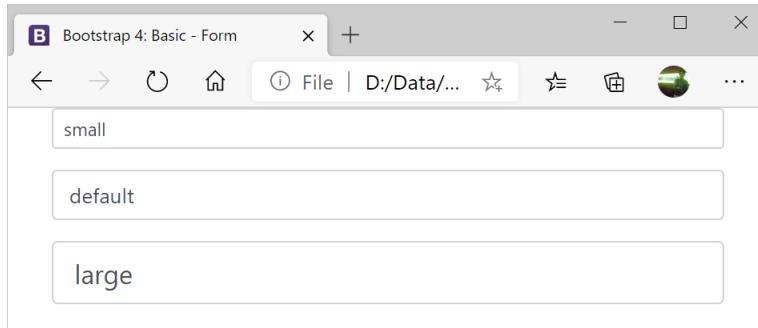
```
</html>
```

Antarmuka yang dihasilkan dari kode di atas adalah seperti pada Gambar 170.

Ukuran elemen input ini dapat diatur dengan menambahkan class berikut setelah .form-control, yaitu:

- .form-control-sm.
- .form-control-lg.

Pada dapat dilihat perbedaan ukuran dari implementasi class-class tersebut pada elemen input.



Gambar 169. Perbedaan ukuran elemen input.

The screenshot shows a web browser window titled "Bootstrap 4: Basic - Form". Inside the window, there is a form with the following elements:

- A text input field labeled "Name:"
- A password input field labeled "Password:"
- A text area labeled "Comment:"
- Two checkboxes: "Option 1" (checked) and "Option 2" (unchecked)
- Two radio buttons: "Option 1" (checked) and "Option 2" (unchecked)
- A select list containing the number "1"
- A file input field with the placeholder "Choose File" and "No file chosen"
- A blue "Submit" button at the bottom

Gambar 170. Penggunaan class .form-control.

Class Bootstrap lain yang erat hubungannya dengan pengaturan tata letak elemen input adalah .form-group yang berfungsi untuk menggabungkan atau mengelompokkan elemen input dengan elemen label atau elemen lainnya seperti potongan kode berikut.

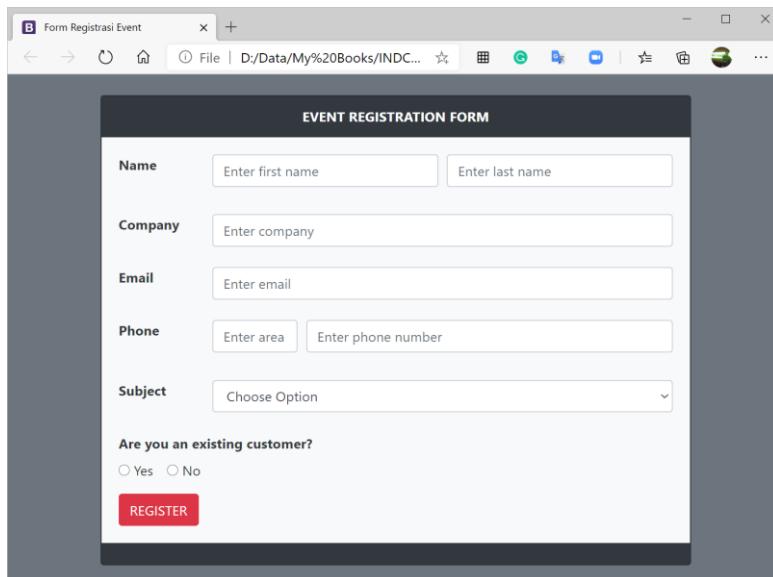
```
...
<div class="form-group">
    <label for="usr">Name:</label>
    <input type="text" class="form-control" id="usr">
</div>
...
<div class="form-group">
    <label for="pwd">Password:</label>
    <input type="password" class="form-control" required>
    <div class="valid-feedback">Valid.</div>
    <div class="invalid-feedback">Not Valid.</div>
</div>
...
```

Project

Pada sub bab CSS telah dibuat beberapa studi kasus pembuatan layout dan antarmuka web untuk form registrasi, layout website dan video player. Pada sub bab ini dibuat antarmuka yang sama namun menggunakan Bootstrap 4.

Form Registrasi

Pada sub bab ini membuat antarmuka untuk form registrasi seperti pada Gambar 171.



Gambar 171. Antarmuka form registrasi dengan Bootstrap 4.

Karena menggunakan Bootstrap 4 maka antarmuka ini bersifat resposif. Pada gambar di atas antarmuka dilihat pada perangkat dengan layar lebar. Sedangkan jika halaman web diakses pada perangkat dengan layar kecil hasilnya seperti pada Gambar 172.

The screenshot shows a mobile-optimized event registration form titled "EVENT REGISTRATION FORM". The form consists of several input fields:

- Name**: Two stacked input fields for "Enter first name" and "Enter last name".
- Company**: A single input field for "Enter company".
- Email**: A single input field for "Enter email".
- Phone**: Two stacked input fields for "Enter area code" and "Enter phone number".
- Subject**: A dropdown menu labeled "Choose Option".
- Are you an existing customer?**: Radio buttons for "Yes" and "No".

A large red "REGISTER" button is located at the bottom of the form.

Gambar 172. Antarmuka form registrasi pada layar kecil.

Berikut ini adalah langkah-langkah untuk membuat antarmuka di atas. Langkah pertama adalah menyiapkan file untuk halaman ini. Berikut adalah file tersebut.

```
project_bootstrap_1.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Form Registrasi Event</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
```

```
</head>

<body class="bg-secondary text-white">
    <!-- main content - start -->
    <div class="container">

        </div>
    <!-- main content - end -->

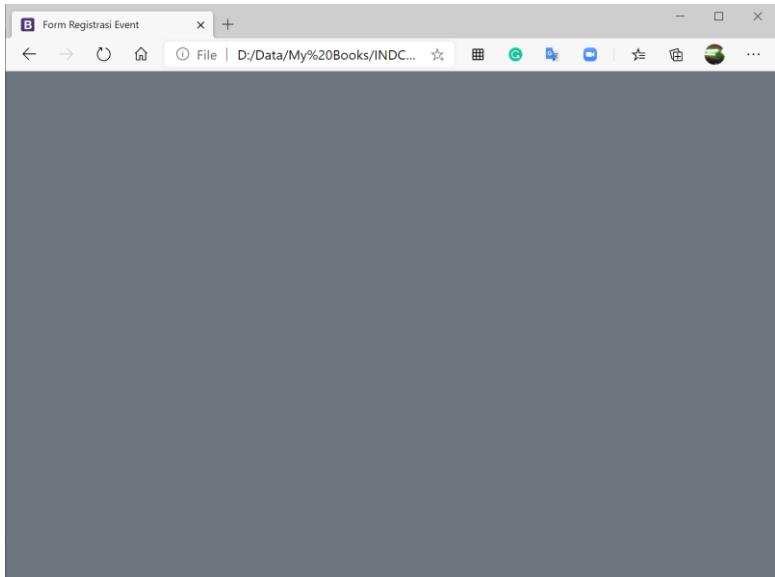
    <!--
        Bootstrap core JavaScript
        Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/popper.min.js"></script>
</body>
</html>
```

Pada kode di atas dapat dilihat potongan kode berikut. Pada tag `<body>` menggunakan class yang telah disediakan oleh Bootstrap yaitu `.bg-secondary` dan `.text-white`.

```
...
<body class="bg-secondary text-white">
...
</body>
...
```

Hasilnya membuat warna latar halaman menjadi seperti pada Gambar 173. Seperti yang telah dibahas pada dasar-dasar Bootstrap bahwa antarmuka ditulis di dalam tag `<div>` dengan menggunakan class `.container`.

```
...
<div class="container">
...
</div>
...
```



Gambar 173. Form Registrasi - Langkah 1.

Selanjutnya adalah menambahkan antarmuka card di dalam container. Card adalah kontainer isi (content) yang fleksibel. Card memiliki tiga area yaitu header, isi dan footer. Sebuah card adalah elemen <div> yang menggunakan class .card. Kemudian didalamnya terdapat tiga elemen <div>. Area head menggunakan elemen <div> dengan nilai atribut class adalah .card-header. Area isi menggunakan elemen <div> dengan nilai atribut class adalah .card-body. Dan area footer adalah elemen <div> dengan nilai atribut class adalah .card-footer. Berikut adalah kode untuk membuat sebuah card.

```
<div class="card">
  <div class="card-header">
    EVENT REGISTRATION FORM
  </div>
  <div class="card-body">

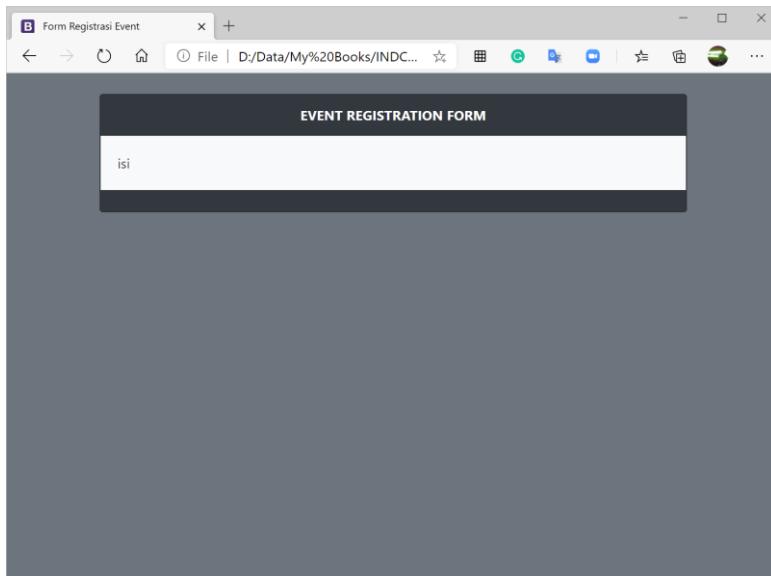
  </div>
  <div class="card-footer"></div>
</div>
```

Untuk membuat card dengan warna area header hitam, area isi putih dan area footer hitam maka ditambahkan class-class lain sehingga kode di atas menjadi sebagai berikut.

```
<div class="card bg-dark text-white mt-4">
  <div class="card-header text-center font-weight-bold">
    EVENT REGISTRATION FORM
  </div>
```

```
</div>
<div class="card-body bg-light text-dark">
    isi
</div>
<div class="card-footer"></div>
</div>
```

Kode di atas menghasilkan antarmuka seperti pada Gambar 174.



Gambar 174. Form Registrasi – Langkah 2.

Langkah selanjutnya adalah menambahkan elemen-elemen form pada area isi. Berikut adalah sebagian isi kode untuk layout form di dalam area isi di card.

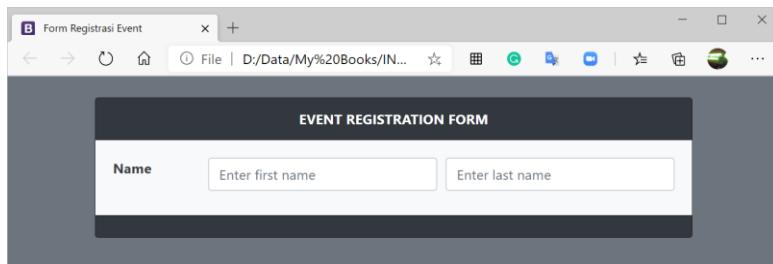
```
...
<form method="POST">
    <div class="form-row">
        <div class="col-md-2">
            <label class="font-weight-bold">Name</label>
        </div>
        <div class="col-md-5 mb-2">
            <input type="text" class="form-control"
                id="first_name"
                placeholder="Enter first name"
                name="first_name">
        </div>
        <div class="col-md-5">
            <input type="password" class="form-control"
                id="last_name"
                placeholder="Enter last name"
                name="last_name">
        </div>
    </div>
</form>
```

```

        name="last_name">
    </div>
</div>
...
</form>
...

```

Hasil kode di atas terlihat seperti pada Gambar 175. Pada Tag pertama di dalam area isi ini adalah <form> kemudian elemen input pertama adalah Name. Pada Gambar 175 terlihat label dan elemen input tersusun dalam 1 baris dan terbagi atas 3 kolom.



Gambar 175. Form Registrasi – Langkah 3.

Untuk membuat agar tersusun rapi dalam 1 baris digunakan tag <div> dengan nilai atribut class adalah .form-row. Kemudian untuk membagi baris ini menjadi 3 kolom digunakan tag <div> didalamnya. Setiap tag menggunakan nilai atribut class adalah .col-md-2, .col-md-5 dan .col-md-5 sehingga jumlah angkanya adalah 12. Hal ini sesuai dengan konsep grid yang telah dibahas pada sub bab sebelumnya. Kemudian di dalam masing-masing elemen <div> tersebut diisi sebuah elemen <label> dan dua elemen <input>.

Di bawah kode di atas ditambah kode berikut ini. Dan hasilnya dapat dilihat pada Gambar 176.

```

<div class="form-row mt-4">
    <div class="col-md-2">
        <label class="font-weight-bold">Company</label>
    </div>
    <div class="col-md-10">
        <input type="text" class="form-control"
            id="company" placeholder="Enter company"
            name="company">
    </div>
</div>

```

Pada kode di atas 1 baris dibagi menjadi 2 kolom yang berisi elemen <div> dengan class adalah .col-md-2 dan .col-md-10 dengan jumlah angka adalah 12.

The screenshot shows a web browser window titled 'Form Registrasi Event'. The main content is a form titled 'EVENT REGISTRATION FORM'. It has two input fields for 'Name': one for 'Enter first name' and one for 'Enter last name'. Below that is a field for 'Company' labeled 'Enter company'.

Gambar 176. Form Registrasi – Langkah 4.

Kemudian langkah selanjutnya adalah menambahkan input untuk email, phone dan subject sehingga akan ada penambahan elemen seperti pada Gambar 177.

The screenshot shows a web browser window titled 'Form Registrasi Event'. The main content is a form titled 'EVENT REGISTRATION FORM'. It includes the fields from Gambar 176, plus new fields: 'Email' (labeled 'Enter email'), 'Phone' (with sub-fields 'Enter area' and 'Enter phone number'), and 'Subject' (a dropdown menu labeled 'Choose Option').

Gambar 177. Form Registrasi – Langkah 5.

Untuk mengambahkan ketiga input tersebut digunakan cara yang mirip dengan input name dan company yang telah ditambahkan sebelumnya. Berikut adalah kode yang digunakan untuk menambahkan ketiga input tersebut.

```
<div class="form-row mt-4">
  <div class="col-md-2">
    <label class="font-weight-bold">Email</label>
  </div>
  <div class="col-md-10">
```

```

        <input type="text" class="form-control" id="email"
               placeholder="Enter email" name="email">
    </div>
</div>
<div class="form-row mt-4">
    <div class="col-md-2">
        <label class="font-weight-bold">Phone</label>
    </div>
    <div class="col-md-2 mb-2">
        <input type="text" class="form-control" id="phone_code"
               placeholder="Enter area code" name="phone_code">
    </div>
    <div class="col-md-8">
        <input type="password" class="form-control"
               id="phone_number"
               placeholder="Enter phone number"
               name="phone_number">
    </div>
</div>
<div class="form-row mt-4">
    <div class="col-md-2">
        <label class="font-weight-bold">Subject</label>
    </div>
    <div class="col-md-10">
        <select class="form-control" id="exampleFormControlSelect1">
            <option>Choose Option</option>
            <option>2</option>
            <option>3</option>
            <option>4</option>
            <option>5</option>
        </select>
    </div>
</div>

```

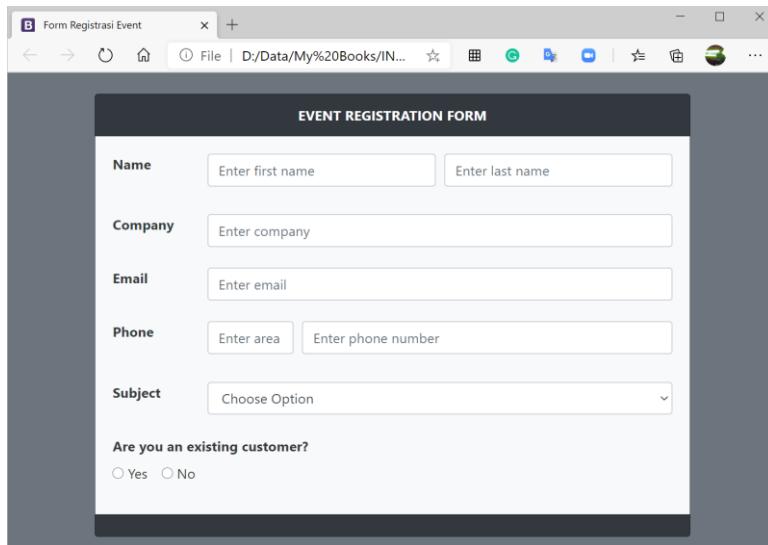
Langkah selanjutnya adalah menambahkan elemen input pemilihan dengan radiobutton seperti pada Gambar 178. Berikut potongan kode yang ditulis untuk input jenis tersebut.

```

<div class="form-group mt-4">
    <label class="font-weight-bold">
        Are you an existing customer?
    </label>
    <br>
    <div class="form-check-inline">
        <label class="form-check-label">
            <input type="radio" class="form-check-input"
                   name="optradio">Yes
        </label>
    </div>
    <div class="form-check-inline">
        <label class="form-check-label">

```

```
<input type="radio" class="form-check-input"  
       name="optradio">No  
</label>  
</div>  
</div>
```



Gambar 178. Form Registrasi – Langkah 6.

Potongan kode di atas agak berbeda dengan penataan tata letak pada kode-kode sebelumnya. Pada kode ini tidak digunakan grid dan kontrol disusun secara inline.

Langkah terakhir untuk membuat antarmuka form registrasi ini adalah menambahkan tombol dengan kode singkat berikut ini sehingga akhirnya didapat antarmuka akhir seperti yang dapat dilihat pada Gambar 171.

```
<button type="submit" class="btn btn-danger">REGISTER</button>
```

Pada sub bab Project di bab JavaScript diberikan contoh melakukan validasi form. Kemudian pada bab ini juga telah diberikan cara validasi form dengan menggunakan Bootstrap 4. Pada project ini dibuat validasi form hanya untuk memastikan nama dan email telah diisi oleh user. Hasilnya adalah pada Gambar 179.

The screenshot shows a web browser window titled 'Form Registrasi Event'. The form has the following fields and validation messages:

- Name**: Two input fields for 'Enter first name' and 'Enter last name'. Below them are error messages: 'First name is required.' and 'Last name is required.'
- Company**: An input field for 'Enter company' with a green checkmark icon.
- Email**: An input field for 'Enter email' with a red error message: 'Email is required.'
- Phone**: Two input fields for 'Enter a' and 'Enter phone number', both with green checkmark icons.
- Subject**: An input field for 'Choose Option' with a green checkmark icon.

Below the form is a question: 'Are you an existing customer?' with 'Yes' and 'No' radio buttons. At the bottom is a red 'REGISTER' button.

Gambar 179. Form Registrasi dengan validasi.

Dan berikut adalah kode lengkap dari halaman web ini.

```
project_bootstrap_1.html
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Form Registrasi Event</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
  </head>

  <body class="bg-secondary text-white">
    <!-- main content - start -->
    <div class="container">
      <div class="card bg-dark text-white mt-4">
        <div class="card-header text-center font-weight-bold">
          EVENT REGISTRATION FORM
        </div>
        <div class="card-body bg-light text-dark">
```

```
<form method="POST" class="needs-validation"
      novalidate>
  <div class="form-row">
    <div class="col-md-2">
      <label class="font-weight-bold">
        Name
      </label>
    </div>
    <div class="col-md-5 mb-2">
      <input type="text"
            class="form-control"
            id="first_name"
            placeholder="Enter first name"
            name="first_name" required>
      <div class="valid-feedback">
        Valid.
      </div>
      <div class="invalid-feedback">
        First name is required.
      </div>
    </div>
    <div class="col-md-5">
      <input type="password"
            class="form-control"
            id="last_name"
            placeholder="Enter last name"
            name="last_name" required>
      <div class="valid-feedback">
        Valid.
      </div>
      <div class="invalid-feedback">
        Last name is required.
      </div>
    </div>
  </div>
  <div class="form-row mt-4">
    <div class="col-md-2">
      <label class="font-weight-bold">
        Company
      </label>
    </div>
    <div class="col-md-10">
      <input type="text"
            class="form-control"
            id="company"
            placeholder="Enter company"
            name="company">
    </div>
  </div>
  <div class="form-row mt-4">
    <div class="col-md-2">
```

```
<label class="font-weight-bold">
    Email
</label>
</div>
<div class="col-md-10">
    <input type="text"
        class="form-control"
        id="email"
        placeholder="Enter email"
        name="email" required>
    <div class="valid-feedback">
        Valid.
    </div>
    <div class="invalid-feedback">
        Email is required.
    </div>
</div>
<div class="form-row mt-4">
    <div class="col-md-2">
        <label class="font-weight-bold">
            Phone
        </label>
    </div>
    <div class="col-md-2 mb-2">
        <input type="text"
            class="form-control"
            id="phone_code"
            placeholder="Enter area code"
            name="phone_code">
    </div>
    <div class="col-md-8">
        <input type="password"
            class="form-control"
            id="phone_number"
            placeholder="Enter phone number"
            name="phone_number">
    </div>
</div>
<div class="form-row mt-4">
    <div class="col-md-2">
        <label class="font-weight-bold">
            Subject
        </label>
    </div>
    <div class="col-md-10">
        <select class="form-control"
            id="exampleFormControlSelect1">
            <option>Choose Option</option>
            <option>2</option>
            <option>3</option>
        </select>
    </div>
</div>
```

```

        <option>4</option>
        <option>5</option>
    </select>
</div>
</div>
<div class="form-group mt-4">
    <label class="font-weight-bold">
        Are you an existing customer?
    </label>
    <br>
    <div class="form-check-inline">
        <label class="form-check-label">
            <input type="radio"
                class="form-check-input"
                name="optradio">Yes
        </label>
    </div>
    <div class="form-check-inline">
        <label class="form-check-label">
            <input type="radio"
                class="form-check-input"
                name="optradio">No
        </label>
    </div>
    <button type="submit"
        class="btn btn-danger">
        REGISTER
    </button>
</form>
</div>
<div class="card-footer"></div>
</div>
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
<script>
    (function() {
        'use strict';
        window.addEventListener('load', function() {
            var forms =
                document.getElementsByClassName('needs-validation');
            var validation =
                Array.prototype.filter.call(forms, function(form) {

```

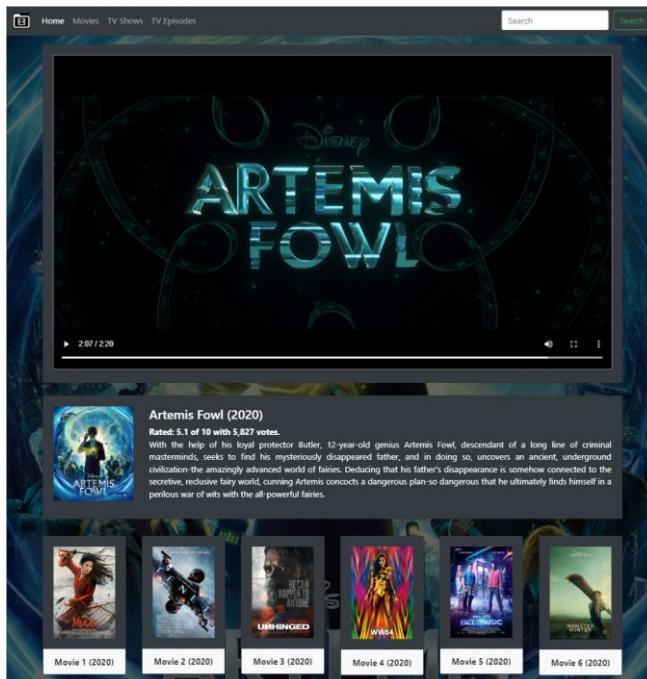
```

        form.addEventListener('submit', function(event) {
            if (form.checkValidity() === false) {
                event.preventDefault();
                event.stopPropagation();
            }
            form.classList.add('was-validated');
        }, false);
    });
}, false);
})();
</script>
</body>
</html>

```

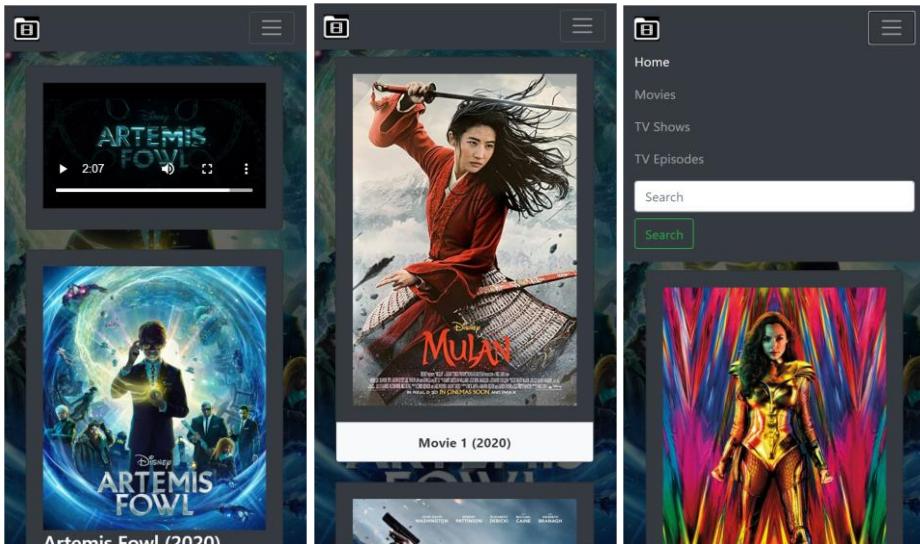
Video Player

Pada project ini dibuat antarmuka Video Player seperti yang telah dibuat pada sub bab Project di pembahasan CSS. Pada project ini akan membuat ulang antarmuka Video Player dengan menggunakan class-class yang telah disediakan pada Bootstrap 4.



Gambar 180. Video player dengan Bootstrap 4.

Pada Gambar 180 adalah antarmuka halaman web Video Player ketika ditampilkan pada perangkat dengan layar lebar. Sedangkan pada Gambar 181 adalah antarmuka halaman ketika diakses pada perangkat dengan layar kecil.



Gambar 181. Video player Bootstrap 4 pada layar kecil.

Untuk membuat halaman ini maka langkah-langkah yang mesti dilakukan adalah sebagai berikut.

Langkah pertama adalah menulis kode berikut ini.

```
<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1">
    <link rel="icon" href="favicon.ico">

    <title>Movie Website</title>

    <!-- Bootstrap core CSS -->
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <style>
      body {
        background: url(images/movie/movie-bg.jpg) no-repeat
                    top center fixed;
        -webkit-background-size: cover;
        -moz-background-size: cover;
```

```

        -o-background-size: cover;
        background-size: cover;
    }
</style>
</head>

<body class="bg-secondary text-white">
    <!-- main content - start -->

    <!-- main content - end -->

    <!--
        Bootstrap core JavaScript
        Placed at the end of the document so the pages load faster
    -->
    <script src="js/jquery.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/popper.min.js"></script>
</body>
</html>

```

Pada kode di atas dapat dilihat style CSS untuk elemen body yang bertujuan untuk menampilkan gambar sebagai latar belakang. Selanjutnya ditambahkan kode di bawah ini untuk navigasi bar di antara area “main content - start” dan “main content - end”.

```

<nav class="navbar navbar-expand-md navbar-dark fixed-top bg-dark">
    <a class="navbar-brand" href="#">
        
    </a>
    <button class="navbar-toggler" type="button"
            data-toggle="collapse" data-target="#navbarCollapse"
            aria-controls="navbarCollapse" aria-expanded="false"
            aria-label="Toggle navigation">
        <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarCollapse">
        <ul class="navbar-nav mr-auto">
            <li class="nav-item active">
                <a class="nav-link" href="#">Home</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">Movies</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">TV Shows</a>
            </li>
            <li class="nav-item">
                <a class="nav-link" href="#">TV Episodes</a>
            </li>
        </ul>
    </div>
</nav>

```

```

<form class="form-inline mt-2 mt-md-0">
    <input class="form-control mr-sm-2" type="text"
        placeholder="Search" aria-label="Search">
    <button class="btn btn-outline-success my-2 my-sm-0"
        type="submit">
        Search
    </button>
</form>
</div>
</nav>

```

Hasilnya dapat dilihat seperti pada Gambar 182.



Gambar 182. Video Player – Navigasi bar.

Selanjutnya adalah menambahkan container di bawah kode di atas. Berikut adalah potongan kode yang harus ditambahkan.

```

<div class="container-fluid mt-5 pt-1">
    <div class="container">
        ...
    </div>

```

```
</div>
```

Kemudian memodifikasi class .container-fluid dengan menambahkan kode berikut pada area <style> di bawah style body. Tujuan memberikan style ini adalah untuk membuat elemen ini berwarna hitam transparan

```
.container-fluid {  
    background-color: rgba(0, 0, 0, 0.6);  
}
```

Elemen <div> dengan nilai atribut class .container-fluid dibuat untuk membuat container dengan lebar 100%. Kemudian untuk mendapatkan kontainer yang ukurannya tidak 100% sehingga terdapat ruang antara sisi kiri dan sisi kanan maka ditambahkan elemen <div> dengan nilai atribut class .container di dalam elemen di atas.

Kemudian ditambahkan elemen untuk membuat video player di dalam elemen container di atas dengan kode sebagai berikut.

```
<div class="card bg-dark text-white mt-4">  
    <div class="card-body bg-dark">  
        <video width="100%" controls>  
            <source src="images/movie/movie.mp4" type="video/mp4">  
            Your browser does not support the video tag.  
        </video>  
    </div>  
</div>
```

Kode di atas memanfaatkan card namun tidak menggunakan .card-header dan .card-footer, hanya menggunakan .card-body saja. Kemudian didalamnya ditambahkan elemen <video>.

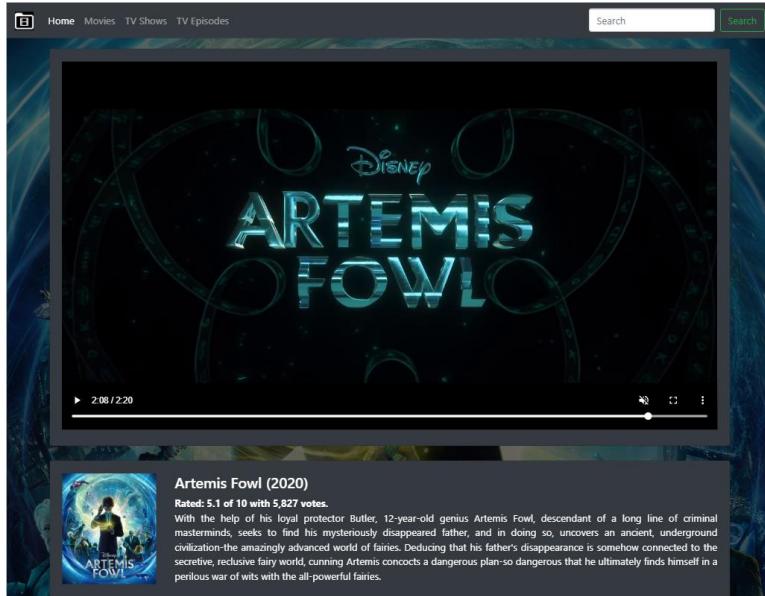
Langkah selanjutnya adalah menambahkan deskripsi dari film berupa poster film, judul, rating dan keterangan dari film dengan menambahkan kode berikut ini. Hasilnya dapat dilihat pada Gambar 183.

```
<div class="card bg-dark text-white mt-4">  
    <div class="card-body bg-dark">  
        <div class="row">  
            <div class="col-sm-2">  
                  
            </div>  
            <div class="col-sm-10">  
                <h4>Artemis Fowl (2020)</h4>  
                <p class="mt-0 pt-0 mb-0 pb-0 font-weight-bold">  
                    Rated: 5.1 of 10 with 5,827 votes.  
                </p>  
            </div>  
        </div>  
    </div>
```

```

<p class="text-justify">
    Text (movie description)
</p>
</div>
</div>
</div>
</div>

```



Gambar 183. Video player – Elemen video.

Selanjutnya adalah menambahkan rekomendasi film dengan menampilkan 6 poster film. Untuk itu perlu dibuat grid dengan yang berisi 1 row dan 6 col (column) dengan kode sebagai berikut.

```

<div class="row mt-2 pb-4">
    <div class="col-sm-2">
        ...
    </div>

    <div class="col-sm-2">
        ...
    </div>

    <div class="col-sm-2">
        ...
    </div>

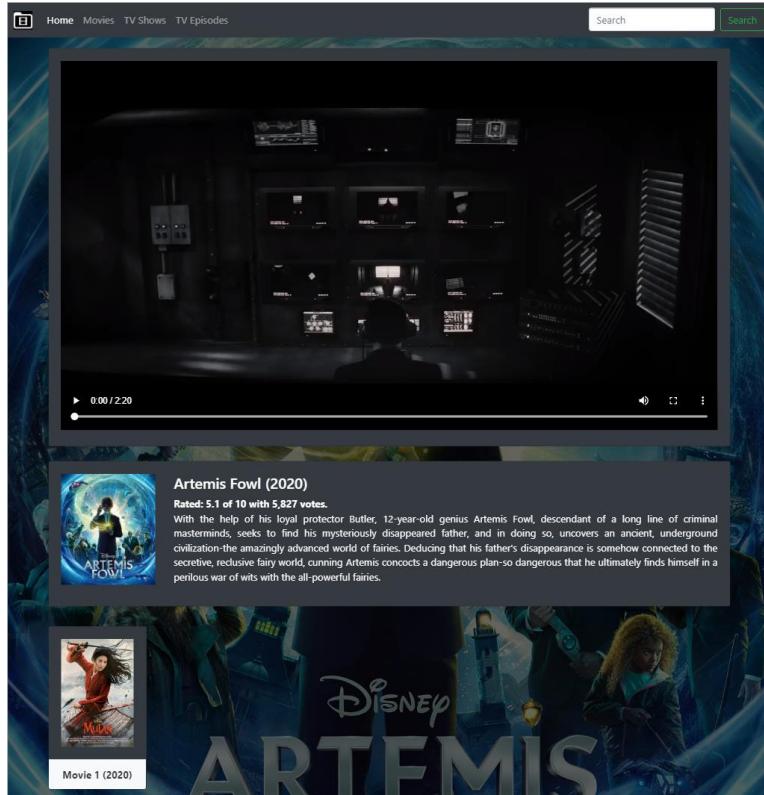
    <div class="col-sm-2">
        ...
    </div>

```

```
...
</div>

<div class="col-sm-2">
...
</div>

<div class="col-sm-2">
...
</div>
</div>
```



Gambar 184. Video player – Poster rekomendasi film.

Langkah selanjutnya adalah menambahkan informasi rekomendasi film ke masing-masing kolom. Informasi rekomendasi film ini ditampilkan dengan menggunakan card dengan kode sebagai berikut. Hasilnya dapat dilihat pada Gambar 184

```
<div class="card bg-dark text-white mt-4">
  <div class="card-body bg-dark">
    
```

```

</div>
<div class="card-footer bg-light text-dark text-center
            font-weight-bold">
            Movie 1 (2020)
        </div>
    </div>

```

Dengan kode di atas, tambahkan kode serupa ke dalam kolam-kolom lainnya sehingga akhirnya ditulis kode lengkap seperti berikut ini.

```

project bootstrap 2.html
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-
scale=1">
        <link rel="icon" href="favicon.ico">

        <title>Movie Website</title>

        <!-- Bootstrap core CSS -->
        <link rel="stylesheet" href="css/bootstrap.min.css">
        <style>
            body {
                background: url(images/movie/movie-bg.jpg) no-repeat
                            top center fixed;
                -webkit-background-size: cover;
                -moz-background-size: cover;
                -o-background-size: cover;
                background-size: cover;
            }

            .container-fluid {
                background-color: rgba(0, 0, 0, 0.6);
            }
        </style>
    </head>

    <body class="bg-secondary text-white">
        <!-- main content - start -->
        <nav class="navbar navbar-expand-md navbar-dark fixed-top
                    bg-dark">
            <a class="navbar-brand" href="#">
                
            </a>
            <button class="navbar-toggler" type="button"
                    data-toggle="collapse"
                    data-target="#navbarCollapse"
                    aria-controls="navbarCollapse"
                    aria-expanded="false"

```

```

        aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse"
    id="navbarCollapse">
    <ul class="navbar-nav mr-auto">
        <li class="nav-item active">
            <a class="nav-link" href="#">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">Movies</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">TV Shows</a>
        </li>
        <li class="nav-item">
            <a class="nav-link" href="#">TV Episodes</a>
        </li>
    </ul>
    <form class="form-inline mt-2 mt-md-0">
        <input class="form-control mr-sm-2"
            type="text" placeholder="Search"
            aria-label="Search">
        <button class="btn btn-outline-success my-2
            my-sm-0" type="submit">
            Search
        </button>
    </form>
</div>
</nav>

<div class="container-fluid mt-5 pt-1">
    <div class="container">
        <div class="card bg-dark text-white mt-4">
            <div class="card-body bg-dark">
                <video width="100%" controls>
                    <source src="images/movie/movie.mp4"
                        type="video/mp4">
                    Your browser does not support
                    the video tag.
                </video>
            </div>
        </div>
        <div class="card bg-dark text-white mt-4">
            <div class="card-body bg-dark">
                <div class="row">
                    <div class="col-sm-2">
                        
                    </div>

```

```

<div class="col-sm-10">
    <h4>Artemis Fowl (2020)</h4>
    <p class="mt-0 pt-0 mb-0 pb-0
        font-weight-bold">
        Rated: 5.1 of 10 with 5,827
        votes.
    </p>
    <p class="text-justify">
        With the help of his loyal prote
        ctor Butler, 12-year-
        old genius Artemis Fowl, descendant of a long line of criminal maste
        rminds, seeks to find his mysteriously disappeared father, and in do
        ing so, uncovers an ancient, underground civilization-
        the amazingly advanced world of fairies. Deducing that his father's
        disappearance is somehow connected to the secretive, reclusive fairy
        world, cunning Artemis concocts a dangerous plan-
        so dangerous that he ultimately finds himself in a perilous war of w
        its with the all-powerful fairies.
    </p>
</div>
</div>
</div>
</div>

<div class="row mt-2 pb-4">
    <div class="col-sm-2">
        <div class="card bg-dark text-white mt-4">
            <div class="card-body bg-dark">
                
            </div>
            <div class="card-footer bg-light
                text-dark text-center
                font-weight-bold">
                Movie 1 (2020)
            </div>
        </div>
    </div>
</div>

<div class="col-sm-2">
    <div class="card bg-dark text-white mt-4">
        <div class="card-body bg-dark">
            
        </div>
        <div class="card-footer bg-light
            text-dark text-center
            font-weight-bold">
            Movie 2 (2020)
        </div>
    </div>
</div>

```

```
</div>

<div class="col-sm-2">
    <div class="card bg-dark text-white mt-4">
        <div class="card-body bg-dark">
            
        </div>
        <div class="card-footer bg-light
            text-dark text-center
            font-weight-bold">
            Movie 3 (2020)
        </div>
    </div>
</div>

<div class="col-sm-2">
    <div class="card bg-dark text-white mt-4">
        <div class="card-body bg-dark">
            
        </div>
        <div class="card-footer bg-light
            text-dark text-center
            font-weight-bold">
            Movie 4 (2020)
        </div>
    </div>
</div>

<div class="col-sm-2">
    <div class="card bg-dark text-white mt-4">
        <div class="card-body bg-dark">
            
        </div>
        <div class="card-footer bg-light
            text-dark text-center
            font-weight-bold">
            Movie 5 (2020)
        </div>
    </div>
</div>

<div class="col-sm-2">
    <div class="card bg-dark text-white mt-4">
        <div class="card-body bg-dark">
            
```

```
</div>
<div class="card-footer bg-light
text-dark text-center
font-weight-bold">
    Movie 1 (2020)
</div>
</div>
</div>
</div>
</div>
<!-- main content - end -->

<!--
Bootstrap core JavaScript
Placed at the end of the document so the pages load faster
-->
<script src="js/jquery.min.js"></script>
<script src="js/bootstrap.min.js"></script>
<script src="js/popper.min.js"></script>
</body>
</html>
```

Penutup

Dengan penjelasan dasar-dasar di atas maka diharapkan dapat menjadi dasar untuk membuat antarmuka halaman web dengan menggunakan HTML 5, CSS dan JavaScript. Selain itu juga buku ini juga dapat memberikan dasar-dasar penggunaan Bootstrap 4 untuk membuat antarmuka halaman web.

Masih materi yang perlu dibahas tentang Bootstrap 4 yang belum diberikan pada buku ini. Pembahasan detail tentang Bootstrap 4 akan dibahas secara khusus pada buku terpisah.

Sedangkan untuk pemrograman web di sisi server dengan menggunakan PHP akan diberikan pada buku kedua yaitu Pemrograman Web Dasar II: Belajar PHP.

Jika ada pertanyaan atau saran perbaikan tentang buku ini maka bisa disampaikan lewat email ke alamat reza.faisal@ulm.ac.id atau friska.abadi@ulm.ac.id.

Tentang Penulis



M Reza Faisal, lahir di Banjarmasin, 20 Desember 1976. Lulus SMA tahun 1995 kemudian melanjutkan kuliah di Informatika Universitas Pasundan (Unpas) dan Fisika Institut Teknologi Bandung (ITB). Sejak kuliah telah aktif menulis 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 di 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, Banjarbaru dan pendidikan S2 jurusan Magister Teknik Informatika di STMIK Amikom, Yogyakarta.