



MODUL

AMSP
2024

PEMROGRAMAN WEBSITE DASAR

MODUL FRONT-END PEMROGRAMAN WEB



Disusun Oleh :

Wahyuni Edsa Safira	(220209502074)
Azzah Ulima Rahma	(220209500016)
Athiyyah Anandira	(220209500030)
Aan Alamsyah	(220209501108)
Muhammad Fadhil Hani	(220209502059)
Andi Asmul Akzam	(220209502033)
Muhammad Adrian	(220209502060)
Nur Aisyah Bahtiar	(220209502051)
Muhammad Syafaad Jufri	(220209502079)
Muhammad Zulfahrizi Zubair	(220209502080)

PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER

JURUSAN TEKNIK INFORMATIKA DAN KOMPUTER

FAKULTAS TEKNIK

UNIVERSITAS NEGERI MAKASSAR

2024

BAB I

PENGENALAN PEMROGRAMAN WEB DAN FRONT-END

A. Tujuan

1. Siswa mampu menjelaskan konsep dasar pemrograman web, termasuk definisi, tujuan, dan jenis-jenis situs web (statis dan dinamis).
2. Siswa mampu mengidentifikasi dan memahami fungsi dari komponen utama dalam pengembangan web, seperti Web Editor, Web Browser, Web Server, dan Database Server.
3. Siswa dapat membedakan antara client-side scripting dan server-side scripting, serta memahami contoh-contoh dari masing-masing jenis scripting.
4. Siswa mampu menjelaskan peran dan tanggung jawab seorang Front-End Developer, serta menguasai dasar-dasar HTML, CSS, dan JavaScript, termasuk penggunaan framework seperti jQuery atau Bootstrap.

B. Teori

1. Pengenalan Pemrograman Web

Pemrograman web diambil dari 2 suku kata yaitu pemrograman dan web. Pemrograman yang dalam bahasa English adalah programming dan diartikan proses, cara, perbuatan program (secara bahasa Indonesia). Definisi Web: jaringan komputer yang terdiri dari kumpulan situs internet yang menawarkan teks dan grafik dan suara dan sumber daya animasi melalui protokol transfer hypertext.

Web programming adalah proses pembuatan program berupa aplikasi yang berisikan dokumen-dokumen multimedia (text, gambar, suara, animasi, video) di dalamnya yang menggunakan protokol http (hypertext transfer protocol) dan untuk mengaksesnya menggunakan perangkat lunak yang disebut browser.

World Wide Web adalah layanan internet yang paling populer saat ini internet dan dikenal juga sebagai halaman-halaman website yang dapat saling terkoneksi satu dengan lainnya (hyperlink) yang membentuk samudra belantara informasi. WWW berjalan dengan protokol HyperText Transfer Protokol (HTTP).

Situs/web dapat dikategorikan menjadi dua yaitu web statis dan web dinamis.

a) Web Statis

Web statis adalah web yang berisi/menampilkan informasi-informasi yang sifatnya statis (tetap). Pada sebuah web statis, pengguna hanya dapat melihat isi dokumen pada halaman web dan apabila diklik akan berpindah ke halaman web yang lain. Interaksi pengguna hanya terbatas dapat melihat informasi yang ditampilkan, tetapi tidak dapat mengolah informasi yang dihasilkan. Web statis biasanya merupakan HTML yang ditulis pada editor teks dan disimpan dalam bentuk .html atau .htm.

b) Web Dinamis

Web dinamis adalah sebuah web yang menampilkan informasi serta dapat berinteraksi dengan pengguna. Web yang dinamis memungkinkan pengguna untuk berinteraksi menggunakan form sehingga dapat mengolah informasi yang ditampilkan web dinamis bersifat interaktif, tidak kaku, dan terlihat lebih indah.

Secara umum pemrograman web dapat dibagi 2, yaitu: clientside scripting dan server side scripting.

- a) Client side scripting (CSS), merupakan jenis script yang pengolahannya dilakukan di sisi client. pengolahan disini berarti “di terjemahkan/ interpreted”. yang memiliki tugas untuk menterjemahkan script jenis ini disisi client adalah web browser. agar semua script yang masuk kategori ini dapat diterjemahkan oleh web browser maka didalam web browser terdapat sebuah komponen/ modul/ “engine” yang memiliki daftar pustaka (library) yang mampu mengenali semua perintah-perintah yang terdapat pada kategori client side scripting. berikut contoh web browser yang populer digunakan: internet explorer, mozilla firefox, opera., safari. berikut adalah contoh client side scripting: HTML (hypertext markup language), java script, XML (extensible markup language), CSS (cascading style sheet). client side scripting merupakan script yang digunakan untuk membuat halaman web statis.
- b) Server side scripting (SSS), merupakan script yang pengolahannya (baca: diterjemahkan) di sisi server. server yang dimaksud disini adalah sebuah komponen yang biasa disebut web server yang didalamnya terintegrasi dengan sebuah mesin (engine)/ modul yang didalamnya terdapat daftar pustaka (library) yang mampu menterjemahkan script-script tersebut yang kemudian setelah di terjemahkan di server kemudian dikirim (ditampilkan) ke client (web browser) dalam format HTML (hypertext markup language). mesin (engine) tersebut harus di install terlebih dahulu karena mesin-mesin tersebut memiliki karakteristik yang berbeda untuk tiap script (bahasa pemrograman) yang digunakan. berikut contoh server side scripting: PHP (Hypertext Pre Processor), ASP (active server pages), JSP (java server pages). server side scripting digunakan untuk membuat halaman web dinamis. hal ini disebabkan karena server side scripting memiliki kelebihan, diantaranya: mampu berinteraksi dengan banyak perangkat lunak basis data (database), mampu mengelola sumber daya yang terdapat di sistem operasi dan perangkat keras komputer, mampu dijalankan di semua sistem operasi (multi platform), aman karena scriptnya di proses di server yang tidak dapat dilihat dari sisi client.

Adapun komponen penyusun web, yaitu:

- a) Web Editor merupakan program aplikasi yang berfungsi untuk mengetikkan perintah-perintah dokumen web baik client side scripting maupun server side scripting.
- b) Web Browser Merupakan program yang berfungsi untuk menampilkan dokumen-dokumen web dalam format HTML.
- c) Web Server adalah program aplikasi yang memiliki fungsi sebagai tempat menyimpan dokumen-dokumen web.
- d) Database Server adalah program yang digunakan untuk menyimpan data yang akan diolah di halaman web. Database biasa disebut Database Management System (DBMS) adalah sebuah aplikasi yang mengembangkan user dan data-data di dalam database.

2. Pengenalan Front-End

Front-End merupakan istilah yang mengacu pada developer yang bekerja dibagian desain layout suatu website atau aplikasi, dimana seorang developer akan mengerjakan user interface agar desain terlihat menarik dan tidak terlihat membosankan untuk dilihat. Seorang Front-End developer wajib menguasai program HTML, CSS, Javascript dan juga framework seperti jQuery, Bootstrap dan lain sebagainya.

BAB II

HTML

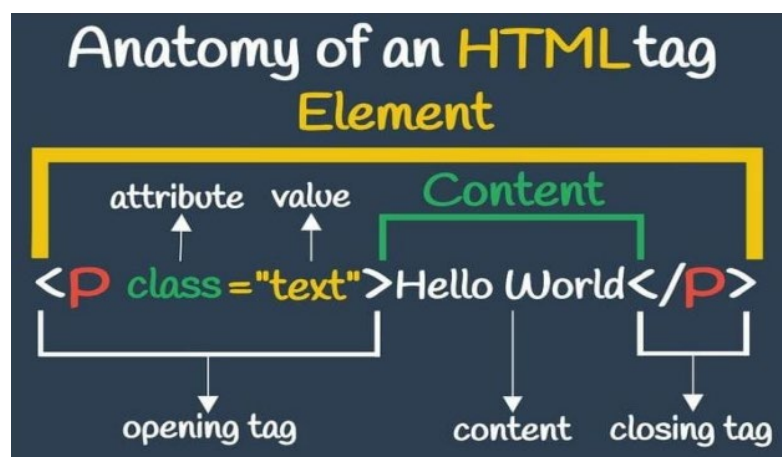
A. Tujuan

1. Siswa mampu menjelaskan pengertian HTML, fungsinya dalam pengembangan web, serta memahami struktur dasar dokumen HTML beserta komponen utamanya, seperti tag, elemen, dan atribut.
2. Siswa dapat mengidentifikasi dan menuliskan elemen-elemen dasar HTML, serta memahami fungsinya masing-masing dalam membangun halaman web.
3. Siswa mampu mengenali dan menggunakan berbagai tipe elemen HTML umum.
4. Siswa mampu menjelaskan dan menerapkan atribut-atribut pada elemen HTML.
5. Siswa mampu mengaplikasikan pengetahuan tentang elemen-elemen HTML dan atributnya dalam membuat halaman web yang terstruktur dan fungsional.

B. Teori

1. Pengenalan HTML (HyperText Markup Language)

HTML (HyperText Markup Language) adalah bahasa standar yang digunakan untuk membuat dan merancang halaman web. Dalam dunia web development, HTML berfungsi sebagai kerangka dasar yang mendefinisikan struktur dan isi dari sebuah halaman web. HTML berbasis pada teks mode, sehingga penulisannya menggunakan program-program yang berbasis pada teks, seperti Notepad, Textpad, Wordpad, dan lain-lain. Pada HTML, tag merupakan instruksi atau perintah dalam HTML, yaitu teks khusus berupa dua karakter “<” dan “>”. Adapun elemen yaitu komponen yang menyusun dokumen HTML, dibentuk dari tag pembuka, isi, dan tag penutup. Atribut merupakan kata khusus yang berada di dalam tag pembuka. Atribut juga disebut sebagai modifier yang akan menentukan perilaku dari elemen. Suatu atribut dapat ditambahkan ke elemen-elemen HTML. Suatu atribut memiliki nilai yang dapat diisi kesetiap elemen yang menggunakannya.



2. Elemen Dasar HTML

Sebelum membuat kode bahasa HTML sebaiknya dipersiapkan juga aplikasi web browser yang seperti internet explorer (browser) bisa juga Chrome, Mozilla,

Netscape, Opera dan yang lainnya untuk menjalankan HTML. Adapun elemen dasar bahasa HTML, yaitu sebagai berikut.

Doctype	Doctype adalah perintah yang menentukan versi HTML yang digunakan. Untuk HTML 5, doctype ditulis sebagai berikut: <code><!DOCTYPE html></code>
HTML	Tag <code><html></html></code> menandai bahwa konten yang ada didalamnya adalah konten HTML.
Head	Tag <code><head></head></code> menyimpan informasi tentang halaman web, seperti judul, deskripsi, dan referensi file lain yang digunakan.
Title	Tag <code><title></title></code> menentukan judul halaman web. Judul ini akan ditampilkan di tab browser atau jendela browser.
Body	Tag <code><body></body></code> menyimpan konten utama dari halaman web.

Penulisan elemen dasar HTML adalah sebagai berikut :

```
<html>
<head><title> ..... </title> </head>
  <body>
    .....
  </body>
</html>
```

Catatan :

Setiap elemen dasar diawali dengan tanda pembuka < > dan diakhiri dengan tanda penutup sesuai jenis elemen.

3. Tipe-tipe Elemen HTML

Berikut adalah beberapa tipe elemen HTML yang umum digunakan:

a. Heading (<h1> sampai <h6>)

Elemen heading digunakan untuk membuat judul atau subjudul. Tingkatan heading lebih rendah memiliki ukuran teks yang lebih kecil.

Contoh:

```
<h1>Judul Utama</h1>
<h2>Judul Sub</h2>
<h3>Judul Sub Sub</h3>
```

Output:

Judul Utama

Judul Sub

Judul Sub Sub

b. Paragraf (<p>)

Elemen paragraf digunakan untuk membuat paragraf teks.

Contoh:

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

Output:

Hello World!

c. Garis Horizontal <hr>

Tag <hr> mendefinisikan pembuatan garis horizontal, dan merupakan elemen kosong tanpa tag penutup.

Contoh:

```
<h1>Judul Utama</h1>
<hr>
<h2>Judul Sub</h2>
```

Output:

Judul Utama

Judul Sub

d. Baris Baru

Tag
 mendefinisikan jeda baris, dan merupakan elemen kosong tanpa tag penutup.

Contoh:

```
<p>Hello World! <br> Today is the day!</p>
```

Output:

Hello World!

Today is the day!

e. Tautan (<a>)

Elemen tautan digunakan untuk membuat hyperlink. Atribut yang biasa dipakai ialah href yaitu atribut yang menentukan alamat atau lokasi dari elemen HTML.

Contoh:

```
<a href="https://www.example.com/">Kunjungi Situs Web Kami</a>
```

Output:

[Kunjungi Situs Web Kami](https://www.example.com/)

f. Gambar ()


Elemen gambar digunakan untuk menampilkan gambar. Atribut yang sering dipakai di adalah src yaitu input gambar dari lokasi penyimpanan gambar, alt (alternative text) yaitu deskripsi dari gambar untuk membantu pengguna yang tidak dapat melihat gambar, width merupakan lebar dari gambar dan height merupakan panjang dari gambar. Jika ukuran width dan height tidak dituliskan maka ukuran gambar yang ditampilkan adalah ukuran sebenarnya dari file gambar tersebut.

Contoh:

```

```

Output:

Deskripsi Gambar

g. Form (<form>)

Elemen form digunakan untuk membuat formulir input, seperti formulir kontak atau login. Formulir memungkinkan pengguna untuk mengirimkan data ke server web, seperti mengisi formulir pendaftaran atau pencarian. Pada form terdapat elemen <label> yang memberikan teks, "Nama:" sebagai label untuk input, dan terdapat atribut for yang menghubungkan label dengan input yang memiliki id="nama". Elemen selanjutnya ialah <input type="text">, yaitu kotak teks untuk memasukkan nama, id berupa identitas unik untuk input, dan name berupa nama yang digunakan saat data dikirim. Adapun elemen terakhir yaitu <input type="submit"> sebagai tombol untuk mengirimkan form, dan value="Kirim" untuk menampilkan teks "Kirim" pada tombol.

Contoh:

```
<form>
  <label for="nama">Nama:</label>
  <input type="text" id="nama" name="nama">
  <input type="submit" value="Kirim">
</form>
```

Output:

Nama:

h. Tabel (<table>)

Elemen tabel digunakan untuk membuat tabel data di halaman web. Tabel terdiri dari baris dan kolom, dengan elemen-elemen tambahan seperti <tr>(baris tabel), <th>(sel kepala tabel), dan <td>(sel data) yang digunakan untuk mengatur konten tabel. Tabel berguna untuk menampilkan data dalam format terstruktur dan rapi.

Contoh:

```
<table border="1">
  <tr>
    <th>Nama</th>
    <th>Umur</th>
  </tr>
  <tr>
    <td>Kim Namjoon</td>
    <td>30</td>
  </tr>
  <tr>
    <td>Jeon Jungkook</td>
    <td>27</td>
  </tr>
</table>
```

Output:

Nama	Umur
Kim Namjoon	30
Jeon Jungkook	27

Adapun beberapa atribut dari tabel yaitu sebagai berikut.

Atribut	Fungsi
<i>Border</i>	Menentukan ukuran <i>border</i> /garis tabel
<i>Width</i>	Menentukan lebar tabel
<i>Height</i>	Menentukan tinggi tabel
<i>Bgcolor</i>	Menentukan <i>background</i> tabel
<i>Background</i>	Menentukan gambar yang digunakan untuk <i>background</i> tabel
<i>Color</i>	Untuk mengatur warna suatu sel dalam tabel
<i>Align</i>	Mengatur bentuk perataan horisontal
<i>Valign</i>	Mengatur bentuk perataan vertikal
<i>Rowspan</i>	Menggabungkan beberapa baris
<i>Colspan</i>	Menggabungkan beberapa kolom
<i>Cellspacing</i> dan <i>cellpadding</i>	Mengatur spasi antar sel dan spasi dalam sel

- i. Iframe (<iframe>)

Elemen digunakan untuk menanamkan dokumen HTML atau sumber eksternal ke dalam halaman web saat ini. Iframe dapat menampilkan konten dari situs web lain atau dokumen HTML terkait di dalam halaman web. Iframe sering digunakan untuk menanamkan peta, video, atau konten lainnya dari sumber eksternal.

Contoh:

```
<iframe  
  src=https://www.smkmuhda.sch.id/  
  name="smkmuhda"  
  title="smkmuhda">  
</iframe>
```

Output:



j. List

List atau daftar dalam HTML digunakan untuk mengorganisasi dan menyajikan informasi dalam format yang terstruktur:

- Unordered List (Daftar Tidak Berurutan): Digunakan untuk menyajikan elemen dalam urutan yang tidak teratur, seperti dengan simbol bullet points atau tanda centang.

Contoh:

```
<ul>  
  <li>Mobil</li>  
  <li>Sepeda</li>  
  <li>Truk</li>  
</ul>
```

Output:

- Mobil
 - Sepeda
 - Truk
- Ordered List (Daftar Berurutan): Menyajikan elemen dalam urutan tertentu, seperti menggunakan nomor atau huruf.

Contoh:

```
<ol>  
  <li>Mobil</li>  
  <li>Sepeda</li>
```

```
<li>Truk</li>
</ol>
```

Output:

1. Mobil
2. Sepeda
3. Truk

- Definition List (Daftar Definisi): Digunakan untuk menyajikan pasangan kata kunci (term) dan definisi yang sesuai. Tag <dl> digunakan untuk mendefinisikan daftar deskripsi, tag <dt> digunakan untuk mendefinisikan istilah (nama), dan tag <dd> digunakan untuk mendeskripsikan setiap istilah.

Contoh:

```
<dl>
  <dt>Mobil</dt>
  <dd>- Kendaraan roda empat</dd>
  <dt>Sepeda</dt>
  <dd>- Kendaraan roda dua</dd>
</dl>
```

Output:

Mobil
- Kendaraan roda empat
Sepeda
- Kendaraan roda dua

k. Komentar

Dalam sebuah pengembangan halaman web, komentar dapat diberikan atau digunakan untuk memberikan penjelasan atau keterangan mengenai script yang dibuat programmer. Untuk membuat komentar, menggunakan tag, diawali dengan <!-- dan diakhiri dengan -->. Komentar tidak ditampilkan oleh browser, tetapi dapat membantu mendokumentasikan kode sumber HTML.

Contoh:

```
<!-- tuliskan komentar disini -->
```

BAB III

CSS

A. Tujuan

B. Teori

BAB IV

JAVASCRIPT

A. Tujuan

1. Siswa dapat mengenal dasar JavaScript HTML
2. Siswa mampu memahami dan membuat dasar javascript dalam dokumen html menggunakan text editor dan menampilkannya lewat browser
3. Siswa mampu membuat file JavaScript extensi HTML sederhana menggunakan text editor dan menampilkan pada browser

B. Teori

1. Dasar Javascript

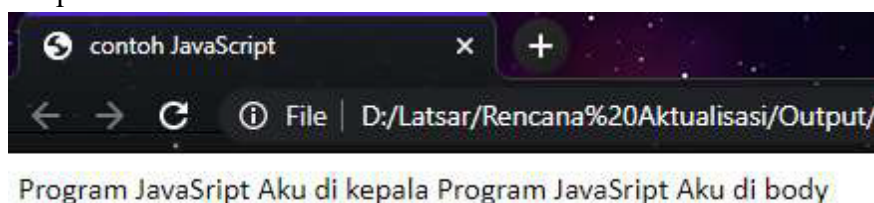
JavaScript adalah bahasa pemrograman yang bisa dimasukkan ke dalam halaman web dan digunakan untuk membuat situs lebih interaktif. Contohnya, JavaScript memungkinkan kita membuat form atau formulir di situs web yang bisa langsung diproses tanpa harus memuat ulang halaman.

Bahasa ini tidak membutuhkan program khusus untuk menjalankannya, cukup dengan menggunakan browser.

JavaScript juga disebut "case sensitive", yang berarti huruf besar dan huruf kecil pada kode memiliki arti berbeda. Setiap perintah dalam JavaScript diakhiri dengan tanda titik koma (;). Untuk menambahkan catatan atau komentar dalam kode, kita bisa menggunakan dua garis miring (//) untuk komentar singkat atau tanda /* dan */ untuk komentar yang lebih panjang. Kode JavaScript biasanya ditulis di dalam tag <script>. Tag ini dapat ditempatkan di bagian kepala (antara <head> dan </head>) atau di badan dokumen (antara <body> dan </body>) dari halaman web. Cara yang umum yang dipakai adalah menuliskannya dalam tag <script>. Tag <script> bisa dibuat di dalam tag <head>, maupun di dalam tag <body>. Contoh:

```
<HTML>
  <HEAD>
    <TITLE> contoh JavaScript</TITLE>
    <script language="JavaScript"> document.write("Program JavaSript Aku di kepala");
  </script>
  </HEAD>
  <BODY>
    <script language="JavaScript"> document.write("Program JavaSript Aku di body");
  </script>
  </BODY>
</HTML>
```

Output:



Menuliskan kode program JavaScript dalam suatu file teks dan kemudian file teks yang berisi kode JavaScript di panggil dari dalam dokumen HTML

```
<SCRIPT LANGUAGE= Javascript SRC= url/file.js > </SCRIPT>
```

Dimana url/file.js adalah lokasi dan nama file yang berisi kode JavaScript, jika perintah tambahan SRC tidak disertakan maka tag Script akan mencari kode yang terletak di dalam tag Script.

JavaScript memiliki peran yang sangat penting dalam pengembangan web modern. Beberapa fungsi utamanya meliputi:

- a. JavaScript memungkinkan pembuatan elemen-elemen halaman web yang dapat berinteraksi dengan pengguna. Ini termasuk tombol yang merespons klik, formulir yang dapat divalidasi secara real-time, dan animasi yang membuat situs web lebih menarik.
- b. JavaScript memungkinkan pengembang untuk mengakses dan memanipulasi elemen-elemen DOM. Dengan demikian, Kita dapat mengubah teks, atribut, atau bahkan menambahkan atau menghapus elemen dengan JavaScript.
- c. JavaScript memungkinkan pengembang mengirim permintaan ke server dan menerima data secara asinkron melalui AJAX (Asynchronous JavaScript and XML). Ini memungkinkan pengembangan aplikasi web yang responsif tanpa perlu me-refresh seluruh halaman.
- d. Saat pengguna mengisi formulir, JavaScript dapat digunakan untuk memeriksa apakah data yang dimasukkan sesuai dengan format yang diharapkan sebelum data dikirim ke server.

2. Variabel

Dalam JavaScript, variabel adalah sebuah wadah atau tempat yang digunakan untuk menyimpan data. Variabel digunakan untuk menyimpan informasi yang dapat diubah atau diakses dalam skrip JavaScript. Data yang disimpan dalam variabel bisa berupa angka, teks, objek, atau tipe data lainnya. Dengan menggunakan variabel, Kita dapat menyimpan, mengolah, dan memanipulasi data dalam program JavaScript. Cara mendefinisikan variabel dalam JavaScript adalah dengan menggunakan kata kunci var, let, atau const, diikuti dengan nama variabel.

- a. var digunakan untuk mendeklarasikan variabel dengan cakupan (scope) fungsi atau global. Namun, variabel var memiliki cakupan yang lebih luas, yang dapat menyebabkan masalah tertentu dalam kode.
- b. let digunakan untuk variabel dengan cakupan blok (block scope). Ini adalah pilihan yang lebih modern dan aman dibandingkan dengan var.
- c. const digunakan untuk variabel yang nilainya tidak akan berubah setelah diberikan nilai awal. Variabel ini bersifat konstan.

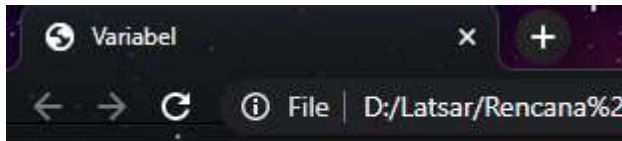
Contoh:

```
var namaVariabel; // Variabel dengan kata kunci var
let usia; // Variabel dengan kata kunci let
const pi = 3.14; // Variabel konstan dengan kata kunci const
```

Jika dideklarasikan dibagian awal dari skrip program, yang artinya sebelum pendeklarasian semua fungsi, maka semua fungsi di dalam program bisa mengakses variabel ini, dan variabel ini menjadi variabel global. Contoh:

```
<SCRIPT language= Javascript >
var a = 12;
var b = 4;
function Perkalian_Dengan2(b) {
a = b * 2;
return a;
}
document.write( Dua kali dari ,b, adalah ,Perkalian_Dengan2(b));
document.write( Nilai dari a adalah ,a);
</SCRIPT>
```

Output:



Dua kali dari 4 adalah 8Nilai dari a adalah8

Jika dia deklarasikan dengan menggunakan kata kunci var di dalam suatu fungsi tertentu, maka variabel itu hanya bisa di akses dari dalam fungsi tersebut, dan artinya variabel ini tidak berguna bagi fungsi fungsi yang lain, dan kita sebut variabel ini menjadi variabel lokal. Contoh:

```
<SCRIPT language="Javascript">
var a = 12;
var b = 4;
function PerkalianDengan2(b) {
var a = b * 2;
return a;
}
document.write)("Dua kali dari “,b,” adalah “, Perkalian dengan2(b));
document.write("Nilai dari a adalah",a);
</SCRIPT>
```

Output:



Dua kali dari 4 adalah 8Nilai dari a adalah12

3. Tipe Data

Tipe data adalah cara komputer mengorganisasi, menyimpan, dan memanipulasi informasi. Hampir setiap bahasa pemrograman memiliki tipe data yang berbeda-beda. Ini adalah beberapa tipe data umum :

a. Tipe Data Dasar

- Number (Angka) digunakan untuk merepresentasikan angka, baik bilangan bulat maupun pecahan. Contoh:

```
let bilBulat = 42;  
let bilPecahan = 3.14;
```

- String (Teks) digunakan untuk merepresentasikan teks. Teks harus diapit oleh tanda kutip tunggal (' ') atau ganda (" "). Contoh:

```
let teks1 = 'Halo, dunia!';  
let teks2 = "Ini adalah contoh string.";
```

- Boolean (Boolean) yang hanya memiliki dua nilai: true atau false. Tipe data ini digunakan untuk ekspresi logika. Contoh:

```
let benar = true;  
let salah = false;
```

b. Tipe Data Khusus

- Null digunakan untuk menyatakan bahwa sebuah variabel tidak memiliki nilai atau referensi. Contoh :

```
let tidakAdaNilai = null;
```

- Undefined digunakan untuk variabel yang telah dideklarasikan tetapi belum diberi nilai. Contoh :

```
let variabelBelumDiisi;  
console.log(variabelBelumDiisi);
```

- Symbol (Simbol) digunakan untuk membuat nilai yang unik dan tidak dapat diubah. Ini berguna dalam konteks pengoptimalan dan penggunaan objek. Contoh :

```
const simbolUnik = Symbol('deskripsi-simbol');
```

c. Tipe Data Referensi

- Object (Objek) digunakan untuk menyimpan banyak nilai dalam satu variabel dalam bentuk pasangan kunci-nilai. Contoh:

```
let mahasiswa = {  
  nama: 'John';  
  usia: 25,  
  jurusan: 'Teknik Informatika'  
};
```

- Array (Larik) digunakan untuk menyimpan sejumlah nilai dalam urutan. Contoh:

```
let angka = [1, 2, 3, 4, 5];
```

- Function (Fungsi): Fungsi adalah tipe data khusus yang digunakan untuk menjalankan blok kode tertentu. Contoh:

```
function tambah(a, b) {  
  return a + b;
```

}

4. Operator

Operator adalah simbol atau karakter khusus yang digunakan untuk melakukan operasi pada tipe data. Mereka memungkinkan kita untuk melakukan perhitungan matematika, perbandingan, manipulasi teks, dan banyak lagi.

a. Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi aritmatika antara variabel dan atau nilai. Misal diberikan $y = 5$, tabel berikut menjelaskan tentang operator aritmatika.

Operator	Deskripsi	Contoh	Nilai x	Nilai y
+	Penjumlahan	$x = y + 2$	7	5
-	Pengurangan	$x = y - 2$	3	5
*	Perkalian	$x = y * 2$	10	5
/	Pembagian	$x = y / 2$	2.5	5
%	Modulus (Sisa bagi)	$x = y \% 2$	1	5
++	Increment	$x = ++y$	6	6
		$x = y++$	5	6
--	Decrement	$x = --y$	4	4
		$x = y--$	5	4

Contoh:

```
<SCRIPT LANGUAGE = "JavaScript">
  <!--
    document.write("2 + 3 = " + (5 + 2) ); document.write("<BR>");
    document.write("20 - 3 = " + (15 - 5) ); document.write("<BR>");
    document.write("20* 3 = " + (5 * 3) ); document.write("<BR>");
    document.write("40 / 3 = " + (15 / 3) ); document.write("<BR>");
  //-->
</SCRIPT>
```

Output:

```
2 + 3 = 7
20 - 3 = 10
20* 3 = 15
40 / 3 = 5
```

b. Operator Logika dan Pembandingan

Operator perbandingan biasanya berfungsi untuk menentukan kode mana dalam program yang harus dijalankan, operasi mana yang perlu dijalankan jika diberikan kondisi yang spesifik. Operator ini biasanya digunakan bersamaan dengan statement if..else, for loop, while loop, switch, dll.

Operator	Keterangan	Kategori
==	Kesamaan	Pembandingan
!=	Ketidaksamaan	Pembandingan
<	Kurang dari	Pembandingan
<=	Kurang dari atau sama dengan	Pembandingan
>	Lebih dari	Pembandingan
>=	Lebih dari atau sama dengan	Pembandingan
!	Bukan	Logika
&&	Dan	Logika
	Atau	Logika
?	Kondisi ? Nilai Benar : Nilai Salah	Pembandingan

Contoh:

```
<HTML>
  <HEAD>
    <TITLE>Operator ?</TITLE>
  </HEAD>
  <BODY>
    <SCRIPT LANGUAGE = "JavaScript">
      <!--
      var nilai = prompt("Nilai (0-100): ", 0);
      var hasil = (nilai >= 60) ? "Lulus" : "Tidak Lulus"; document.write("Hasil: " + hasil);
      //-->
    </SCRIPT>
  </BODY>
</HTML>
```

Output:

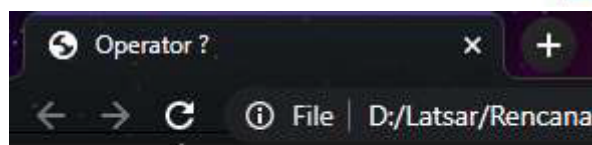
This page says

Nilai (0-100):

85

OK

Cancel



Hasil: Lulus

5. Fungsi Javascript (Function)

JavaScript adalah bahasa pemrograman yang bisa dimasukkan ke dalam halaman web dan digunakan untuk membuat situs lebih interaktif. Contohnya, JavaScript memungkinkan kita membuat form atau formulir di situs web yang bisa langsung diproses tanpa harus memuat ulang halaman. Bahasa ini tidak membutuhkan program khusus untuk menjalankannya, cukup dengan menggunakan browser.

Sebuah fungsi JavaScript didefinisikan dengan keyword `function`, diikuti dengan sebuah nama, diikuti oleh tanda kurung `()`. Nama fungsi dapat berisi huruf, angka, garis bawah, dan tanda dolar (aturan yang sama dengan variabel). Di dalam tanda kurung, dapat menyertakan nama parameter yang dipisahkan oleh koma: (parameter1, parameter2, ...) Kode yang akan dieksekusi oleh fungsi tersebut ditempatkan di dalam kurung kurawal: `{}`. Contoh:

```
function name(parameter1, parameter2, parameter3) {  
  // code to be executed  
}
```

Untuk membuat fungsi di JavaScript, terdapat tiga cara utama yang dapat digunakan cara biasa (function declaration), ekspresi fungsi (function expression) dan fungsi panah (arrow function).

a. Function Declaration

Function declaration adalah cara mendefinisikan function di JavaScript yang menggunakan kata kunci `function` diikuti oleh nama function dan parameter (jika ada) yang diletakkan dalam tanda kurung. Function declaration mendeklarasikan function di scope global, sehingga dapat diakses dari mana saja dalam program. Contoh:

```
function sapa(nama) {  
  console.log('Halo, $(nama)!');  
}
```

b. Function Expression

Dalam metode ini, fungsi didefinisikan sebagai nilai yang disimpan dalam variabel. Fungsi ini sering kali merupakan fungsi anonim (tanpa nama). Untuk memanggilnya, gunakan nama variabel diikuti tanda kurung. Contoh:

```
var namaFungsi = function() {  
  console.log("Hello World!");  
};
```

c. Arrow Function

Fungsi panah diperkenalkan dalam ES6 dan menawarkan sintaks yang lebih ringkas. Fungsi ini juga sering kali lebih mudah dibaca, terutama untuk fungsi satu baris. Seperti sebelumnya, untuk memanggil fungsi ini, cukup gunakan `namaFungsi()`. Contoh:

```
var namaFungsi = () => {  
  console.log("Hello World!");  
}
```

```
};
```

// Atau jika hanya memiliki satu pernyataan:

```
var namaFungsi = () => console.log("Hello World!");
```

d. Return Statement

Function dalam JavaScript juga dapat mengembalikan nilai dengan menggunakan pernyataan return. Ini berguna ketika Kita ingin mengembalikan hasil perhitungan atau nilai tertentu dari function. Contoh:

```
function tambah(angka1, angka2) {  
    return angka1 - angka2;  
}  
let hasil = tambah(5, 3);  
console.log(hasil); //output 2
```

6. Looping

Dalam JavaScript, variabel adalah sebuah wadah atau tempat yang digunakan untuk menyimpan data. Variabel digunakan untuk menyimpan informasi yang dapat diubah atau diakses dalam skrip JavaScript. Data yang disimpan dalam variabel bisa berupa angka, teks, objek, atau tipe data lainnya. Dengan menggunakan variabel, Kita dapat menyimpan, mengolah, dan memanipulasi

BAB V BOOTSTRAP

A. Tujuan

1. Siswa dapat menjelaskan apa itu Bootstrap dan mengapa digunakan untuk membuat situs web yang responsif
2. Siswa dapat menginstal Bootstrap melalui beberapa cara
3. Siswa dapat memahami dan menggunakan komponen atau class Bootstrap
4. Siswa mampu membuat desain web yang responsif dan dapat menyesuaikan tampilan di berbagai perangkat

B. Teori

1. Pengenalan Bootstrap

Bootstrap adalah framework frontend open-source yang digunakan untuk merancang, mengembangkan, dan menyesuaikan situs web yang responsif serta mobile-first dengan cepat. Framework ini menawarkan sistem grid yang fleksibel, berbagai komponen interaktif dan aksesibel yang sudah siap digunakan, serta beragam utilitas dan alat bantu yang sangat berguna.

Bootstrap menyediakan gaya CSS bawaan untuk semua komponennya, mencakup elemen-elemen seperti tipografi, warna, ukuran, spasi, dan breakpoint, yang mendukung sistem grid. Bootstrap 5 bisa digunakan dengan pengaturan gaya default, namun pengguna juga bisa menyesuaikannya sesuai kebutuhan.

Sebagai kerangka kerja front-end gratis, Bootstrap mempermudah dan mempercepat pengembangan web dengan menyediakan template desain berbasis HTML dan CSS untuk berbagai elemen seperti tipografi, tombol, formulir, tabel, navigasi, modals, carousel, serta tambahan plugin JavaScript opsional. Dengan Bootstrap, pembuatan desain web responsif menjadi lebih sederhana, memungkinkan situs web menyesuaikan tampilannya dengan baik di berbagai perangkat, mulai dari ponsel hingga desktop.

Bootstrap sendiri memiliki 5 versi dan bootstrap 5 adalah versi terbaru dari Bootstrap yang dirilis pada tahun 2021. Versi ini menawarkan komponen baru, stylesheet yang lebih cepat, dan peningkatan responsivitas. Bootstrap 5 mendukung versi stabil terbaru dari semua browser dan platform utama. Namun, Internet Explorer 11 dan versi sebelumnya tidak didukung. Perbedaan utama antara Bootstrap 5 dan versi sebelumnya (Bootstrap 3 & 4) adalah bahwa Bootstrap 5 telah beralih menggunakan JavaScript murni, bukan jQuery.

2. Instalasi Bootstrap

Ada beberapa cara untuk mulai menggunakan Bootstrap 5 di situs web yaitu sebagai berikut.

a. Menggunakan CDN

CDN (Content Delivery Network) adalah jaringan server yang terdistribusi secara geografis di berbagai lokasi di seluruh dunia. Untuk instalasi bootstrap cukup copy link CDN Bootstrap yang bisa di dapat pada website resmi bootstrap, yaitu <https://getbootstrap.com/docs/5.2/getting-started/introduction/> lalu di taruh pada kode HTML. Kelebihan dari cara ini adalah kita bisa langsung menggunakan Bootstrap tanpa harus download. Tapi tetap butuh koneksi internet. Berikut langkah-langkah secara rinci.

- Buat file baru *index.html* di root proyek Anda. Sertakan `<meta name="viewport">` juga tag untuk perilaku responsif yang tepat di perangkat seluler.
- Sertakan CSS dan JS Bootstrap. Tempatkan `<link>` tag di dalam `<head>` untuk CSS kita, dan `<script>` tag untuk bundel JavaScript kita (termasuk Popper untuk memposisikan dropdown, popper, dan tooltip) sebelum penutup `</body>` contoh:

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css
"
rel="stylesheet"
integrity="sha384-
rbsA2VBKQhggwzxH7pPCaAqO46MgnOM80zW1RWuH61DGLwZJEdK2
Kadq2F9CUG65" crossorigin="anonymous">
  </head>
  <body>
    <h1>Hello, world!</h1>
    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.mi
n.js"
integrity="sha384-
kenU1KFdBIe4zVF0s0G1M5b4hcxpD9F7jL+jjXkk+Q2h455rYXK/7HAuo
Jl+0I4" crossorigin="anonymous"></script>
  </body>
</html>

```

Adapun link CDN bootstrap 5 versi terbaru, yaitu sebagai berikut:

CSS	https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/css/bootstrap.min.css
JavaScript	https://cdn.jsdelivr.net/npm/bootstrap@5.2.3/dist/js/bootstrap.bundle.min.js

b. Download secara manual

Bootstrap dapat diunduh dari website resminya, tetapi tidak termasuk dokumentasi, file sumber, atau dependensi JavaScript opsional seperti Popper.

Compiled CSS and JS

Download ready-to-use compiled code for **Bootstrap v5.2.3** to easily drop into your project, which includes:

- Compiled and minified CSS bundles (see [CSS files comparison](#))
- Compiled and minified JavaScript plugins (see [JS files comparison](#))

This doesn't include documentation, source files, or any optional JavaScript dependencies like Popper.

[Download](#)

Setelah diunduh, maka akan mendapatkan file zip dengan isi folder CSS dan JavaScript. Setelah itu ekstrak folder CSS dan JavaScript ke dalam folder HTML

masing-masing. Siapkan tag link disamping pada tag head halaman HTML, dan tag script di dalam tag body.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bootstrap dari Lokal</title>

  <link rel="stylesheet" href="css/bootstrap.min.css" />
</head>
<body>

<header class="py-5 bg-success">
  <div class="container">
    <h1 class="display-4 text-white">Hello Bootstrap Lokal!</h1>
  </div>
</header>

<script src="js/bootstrap.min.js"></script>

</body>
</html>
```

c. Download dengan package manager (npm)

Masukkan berkas sumber Bootstrap ke hampir semua proyek dengan beberapa pengelola paket yang paling populer. Apa pun pengelola pakatnya, Bootstrap akan memerlukan kompiler Sass dan Autoprefixer untuk pengaturan yang sesuai dengan versi resmi yang kami kompilasi. Berikut ini adalah langkah-langkah dalam mendownload bootstrap dengan package manager

- Install Node.js dari website resminya, yaitu: <https://nodejs.org/en/download/package-manager>, karena npm (node package manager) adalah bagian dari instalasi Node.js
- Buat direktori proyek baru atau navigasikan ke direktori proyek yang sudah ada di terminal
- Jika belum menginisialisasi proyek npm, jalankan perintah berikut untuk membuat file package.json: *npm init -y*
- Jalankan perintah berikut untuk mengunduh Bootstrap menggunakan npm: *npm install bootstrap@5.2.3*
- Setelah Bootstrap diunduh, maka dapat digunakan file CSS Bootstrap yang ada di dalam proyek.


```
<link rel="stylesheet" href="node_modules/bootstrap/dist/css/bootstrap.min.css">
```

- Pastikan telah menyesuaikan path (alamat file) sesuai dengan struktur direktori proyek
- Jika perlu menggunakan komponen interaktif Bootstrap yang memerlukan JavaScript, maka dapat disertakan JavaScript Bootstrap.

```
<script src="node_modules/bootstrap/dist/js/bootstrap.min.js"></script>
```

- Class-class Bootstrap dalam elemen HTML sudah dapat digunakan

3. Class Bootstrap

Class dalam konteks Bootstrap mengacu pada atribut CSS class yang dapat ditambahkan ke elemen HTML untuk mengaplikasikan gaya dan perilaku tertentu sesuai dengan aturan yang telah ditentukan oleh Bootstrap framework. Bootstrap menggunakan sistem class untuk mengontrol tampilan dan perilaku elemen-elemen web secara konsisten. Untuk referensi terdapat di website: https://www.w3schools.com/bootstrap/bootstrap_ref_all_classes.asp

a. Container

Class ini digunakan untuk membungkus konten dengan lebar tetap yang responsif. Ini membuat konten Anda tetap dalam lebar tertentu pada berbagai perangkat dan ukuran layar. Contoh:

```
<body>
  <div class="container">
    <h1>Judul</h1>
    <p>Paragraf</p>
  </div>
</body>
```

Output:



Judul

Paragraf

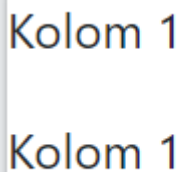
b. Grid

Sistem grid Bootstrap dibangun dengan flexbox dan memungkinkan hingga 12 kolom di seluruh halaman. Jika tidak ingin menggunakan semua 12 kolom secara individual, maka dapat mengelompokkan kolom-kolom tersebut bersama-sama untuk membuat kolom yang lebih lebar. Sistem grid responsif, dan kolom akan tersusun ulang secara otomatis tergantung pada ukuran layar. Pastikan jumlahnya mencapai 12 atau kurang. Pada grid terdapat class row dan col-{breakpoint}-

{number}. Class row digunakan untuk membuat baris dalam grid Bootstrap. Ini membantu dalam mengatur elemen-elemen dalam baris-baris horizontal untuk membuat layout responsif. Sedangkan col-{breakpoint}-{number} digunakan untuk membuat kolom dalam grid. {breakpoint} dapat diganti dengan xs, sm, md, atau lg untuk mengatur perilaku responsif pada berbagai ukuran layar. {number} menunjukkan jumlah kolom yang ingin digunakan dalam baris tersebut. Contoh:

```
<div class="row">
  <div class="col-md-6">
    <p>Kolom 1</p>
  </div>
  <div class="col-md-6">
    <p>Kolom 2</p>
  </div>
</div>
```

Output:



Kolom 1

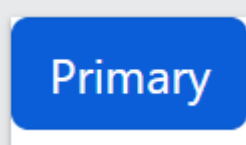
Kolom 1

c. Button (btn)

Class btn digunakan untuk mengubah tampilan tombol. Class ini memberikan gaya dasar pada tombol dan membuatnya terlihat lebih menarik. Contoh:

```
<body>
  <div class="container">
    <button class="btn btn-primary">Primary</button>
  </div>
</body>
```

Output:



Primary

d. Form-control

Class form-control digunakan untuk mengubah tampilan input dan textarea dalam formulir. Ini memberikan tampilan yang konsisten dan bersih untuk elemen-elemen form. Contoh:

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

```

<div class="form-group">
  <label for="nama">Nama: </label>
  <input type="text" class="form-control" id="nama">
</div>
<div class="form-group">
  <label for="email">Email:</label>
  <input type="email" class="form-control" id="email">
</div>
<button type="submit" class="btn btn-primary">Submit</button>
</form>
</div>
</body>

```

Output:

Nama:

Email:

e. Navbar

Navbar adalah komponen antarmuka yang digunakan untuk membuat navigasi di bagian atas sebuah halaman web. Navbar menyediakan struktur dan gaya yang responsif, sehingga bisa beradaptasi dengan berbagai ukuran layar, dari desktop hingga perangkat mobile. Navbar sangat fleksibel dan dapat disesuaikan dengan elemen-elemen lain seperti logo, link, tombol, dropdown, form pencarian, dan lain-lain.

Navbars dilengkapi dengan dukungan bawaan untuk beberapa sub-komponen. Pilih dari yang berikut ini sesuai kebutuhan:

- `` .navbar-brand`` untuk nama perusahaan, produk, atau proyek Anda.
- `` .navbar-nav`` untuk navigasi dengan tinggi penuh dan ringan (termasuk dukungan untuk dropdown).
- `` .navbar-toggler`` untuk digunakan dengan plugin collapse kami dan perilaku toggling navigasi lainnya.
- Utilitas flex dan spasi untuk kontrol formulir dan aksi apa pun.
- `` .navbar-text`` untuk menambahkan string teks yang diposisikan secara vertikal di tengah.
- `` .collapse.navbar-collapse`` untuk mengelompokkan dan menyembunyikan isi navbar berdasarkan breakpoint induk.
- Tambahkan `` .navbar-scroll`` opsional untuk menetapkan tinggi maksimal dan menggulir konten navbar yang diperluas.

Contoh:

```
<nav class="navbar navbar-expand-sm bg-light">

  <div class="container-fluid">
    <!-- Links -->
    <ul class="navbar-nav">
      <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>
  </div>

</nav>
```

Output:

Link 1 Link 2 Link 3