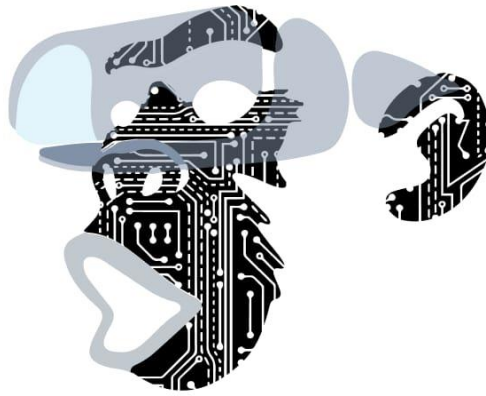


**MODUL**  
**WEB DEVELOPMENT 1**



**IQ PRIMATECH**

## Informasi Umum

Materi Pokok	:	a. HTML b. CSS c. JAVASCRIPT
Capaian Pembelajaran	:	Mampu memahami konsep dan menerapkan perintah HTML, CSS, Pemrograman Javascript dalam pembuatan web statis dan dinamis.
Kompetensi Awal	:	Memahami Pemrograman dasar pengembangan perangkat lunak.
Sarana dan Prasarana a. Sarana • Alat  b. Prasarana • Media ajar • lingkungan	:	<ul style="list-style-type: none"><li>• Laptop</li> <li>• Text Editor Visual Studio Code</li><li>• Aula Fakultas Teknik Universitas Islam Riau</li></ul>

## DAFTAR ISI

### Bab 1 Persiapan

1.1 Web Browser.....	1
1.2 Text Editor .....	1

### Bab 2 Pengenalan HTML

2.1. Apa itu HTML? .....	2
2.2. Struktur file HTML.....	2
2.3. Pengenalan Tag HTML.....	3
2.4. Mengenal Atribut HTML .....	3
2.5. membuat link.....	4
2.6. heading.....	4
2.7. List .....	5
2.8. Division.....	6
2.9. Form.....	7
2.10. Tabel .....	8

### Bab 3 Cascading Style Sheet (CSS)

3.1. Pengenalan CSS.....	10
3.2. Penulisan CSS .....	10
3.3. Styling CSS .....	10
3.4. Position CSS .....	14
3.5. Box Model .....	17

### Bab 4 Javascript

4.1. Pengenalan Javascript .....	19
4.2. Tipe Data .....	20
4.3. Variable.....	22
4.4. Operator .....	23
4.5. Conditional Statement .....	28
4.6. Array.....	29
4.7. Looping.....	30
4.8. Dom (Document Object Model) .....	31

## **BAB 1 PERSIAPAN**

### **1.1 Web Browser**

web browser adalah perangkat lunak yang memungkinkan kamu untuk mencari, mengakses, dan menampilkan halaman website di internet. Pada dasarnya, website berisi kode seperti JavaScript dan HTML yang tidak dapat langsung dibaca oleh manusia. Web browser akan menerjemahkan kode tersebut ke dalam tulisan, gambar, audio, maupun elemen lainnya. Web browser bukanlah satu-satunya perangkat lunak yang bisa membantumu mengakses website. Namun, kebanyakan orang lebih memilih menggunakan perangkat lunak ini karena mudah digunakan. Jenis – jenis Web Browser

- a. Firefox
- b. Google Chrome
- c. Maxthon
- d. Safari
- e. Internet Explorer

### **1.2 Text Editor**

Text editor adalah sebuah aplikasi yang biasa dipakai untuk menyusun kode program sehingga menjadi aplikasi baru. Text editor biasanya digunakan oleh berbagai kalangan, yang memerlukan alat untuk menulis atau mengedit teks. Text editor sangat berguna dalam pengembangan perangkat lunak, karena memungkinkan programmer untuk menulis dan menguji kode dengan efisien. Beberapa contoh teks editor yang populer adalah:

- a. Notepad (Windows)
- b. TextEdit (MacOS)
- c. Visual Studio Code (bisa digunakan di berbagai sistem operasi)
- d. Sublime Text

## BAB 2 HTML

### 2.1. Apa itu HTML?

Bahasa utama dalam membuat website adalah HTML atau Hyper-Text Markup Language. Lazimnya semua halaman web ditulis menggunakan variasi HTML. Dengan HTML, pengembang website dapat memastikan bahwa text, gambar dan multimedia dapat menyatu dengan elemen lainnya ketika dijalankan di browser. Saat ini perkembangan HTML sudah mencapai versi terbaru yaitu HTML5. HTML5 ini memiliki pembaharuan fitur-fitur dari HTML lama. Fitur baru selalu update dan ditambahkan ke HTML5 di setiap rilisnya.

### 2.2. Struktur file HTML

Setiap dokumen HTML memiliki aturan penulisan, terdapat tag yang harus diketikkan dan sudah menjadi keharusan. Struktur kodenya menjadi sebagai berikut:

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>PRIMATECH</title>
7  </head>
8  <body>
9
10 </body>
11 </html>
```

<!DOCTYPE html>

Tag awal dari tiap dokumen HTML, tag ini berguna untuk memberi informasi ke browser bahwa dokumen yang sedang dibuka merupakan dokumen HTML.

<html> ... </html>

Pemuka isi dokumen HTML. Semua kode HTML yang dibuat harus ditulis didalam tag ini.

Setiap tag <html> wajib memiliki tag penutup </html>

```
<head> ... </head>
```

Berfungsi untuk menampung beragam informasi terkait dokumen HTML. Tulisan yang terdapat didalam tag ini, tidak akan ditampilkan secara langsung pada browser.

```
<title> ... </title>
```

Merupakan salah satu contoh informasi yang terdapat didalam tag head. Title menampilkan judul browser atau pada tab browser.

```
<body> ... </body>
```

Tag body merupakan menampung apa yang ingin kita tampilan pada browser. Tag body merupakan tag pembuka dari badan dokumen HTML.

### 2.3. Pengenalan Tag HTML

HTML merupakan Bahasa Markup, artinya bahasa HTML semata mata digunakan untuk me-markup suatu dokumen. Ketika membuat dokumen word. Biasanya akan mengerjakan formatting teks seperti menebalkan, memiringkan, memberi garis bawah. Pada HTML, formatting ini dapat dikutip sebagai proses Markup dengan memakai tag HTML. Misalnya untuk mem-format suatu teks menjadi miring, tambahkan tag `<em>` atau `<i>`. Disebut dengan tag HTML, yang berfungsi untuk memformat dokumen HTML. Setiap tag pasti memiliki pasangan yaitu tag penutup yang ditandai dengan slash (/). Untuk menebalkan tulisan atau teks, bisa memakai tag `<strong>` atau `<b>`.

### 2.4. Mengenal Atribut HTML

Setiap tag HTML boleh mempunyai satu atau lebih atribut, atribut ini berguna untuk menampung informasi yang berkaitan dengan tag tersebut. Pada tag `<img>` kita akan menggunakan atribut `src` untuk menunjukkan lokasi.

```
<img src="">
```

Selanjutnya guna memasukkan nilai pada atribut, kita gunakan tanda operator sama dengan (=) diikuti dengan lokasi gambar yang diapit oleh tanda kutip.

```

```

Tag <img> termasuk tag spesial karena tidak memiliki tag penutup (Self Closed Tag). Jika kita menyimpan gambar tersebut pada sebuah folder seperti berikut. Maka kita perlu menambahkan nama folder tersebut diikuti dengan tanda slash (/) dan nama file gambar yang akan dimuat.

```

```

Apabila dalam folder ada folder. Maka penulisan.

```

```

## 2.5. Membuat Link

Link atau anchor (pengait) bisa dibuat dengan membubuhkan tag pada teks yang akan dijadikan link.

```
<a> Informasi lebih lanjut </a>
```

Dengan penulisan tag diatas, kita belum melihat perubahan pada teks “disini” karena kita belum “manautkannya” ke halaman web lain. Untuk itu kita perlu menggunakan atribut href untuk menunjukkan alamat web yang akan dituju ketika link di klik.

```
< a href=https://www.iqprima.tech> Disini </a>
```

## 2.6. Heading

Heading pada HTML adalah Title atau subtitle yang ingin ditampilkan pada website. Heading terdiri dari tag <h1> hingga <h6>. Tag <h1> dipergunakan untuk heading yang tingkat hierarkinya paling tinggi sedangkan tag <h6> untuk yang paling rendah. Heading merupakan bagian penting pada website karna search engine (google, Yahoo, dll) menggunakan heading sebagai indeks strukru dan konten pada website.

Para pengguna juga seringkali menelusuri website berdasarkan heading pada website tersebut.

```
<h1>Heading 1</h1>
<h2>Heading 2</h2>
<h3>Heading 3</h3>
<h4>Heading 4</h4>
<h5>Heading 5</h5>
<h6>Heading 6</h6>
```

# Heading 1

## Heading 2

### Heading 3

#### Heading 4

##### Heading 5

###### Heading 6

## 2.7. List

Sebuah list dapat diartikan sebagai sebuah daftar yang mana menggunakan penomoran angka atau abjad ataupun sebuah simbol. List terbagi menjadi dua yaitu ordered(berurutan) dan unordered(acak). Ordered list dibuat dengan menggunakan tag `<ol>` dan setiap listnya menggunakan tag `<li>` berikut contohnya:

```
<ol>
  <li>Aku</li>
  <li>Kamu</li>
  <li>Kita</li>
</ol>
```

1. aku
2. kamu
3. kita



Unordered list dibuat dengan menggunakan tag <ul> dan setiap listnya menggunakan tag <li> berikut contohnya:

```
<ul>
  <li>Aku</li>
  <li>Kamu</li>
  <li>Kita</li>
</ul>
```

- aku
- kamu
- kita

## 2.8. Division

Dalam konteks HTML, "division" atau sering disebut sebagai <div>, adalah elemen yang digunakan untuk membuat kelompok atau bagian dalam halaman web. Elemen <div> tidak memberikan arti khusus secara langsung, tetapi digunakan untuk mengelompokkan dan memformat bagian-bagian tertentu dari halaman web.

Secara sederhana, <div> adalah kotak kosong di mana kita dapat menempatkan berbagai elemen HTML lainnya, seperti teks, gambar, atau elemen-elemen lainnya. Ini memudahkan kita dalam mengatur tata letak halaman web dan memungkinkan kita untuk menerapkan gaya CSS ke kelompok elemen tertentu secara banyak atau tunggal.

div berikut contohnya :

```
<div>
  <p>ini link youtube</p>
  <a href="youtube.com">klik</a>
</div>
```

## 2.9. Form

Form digunakan untuk mengumpulkan data dari pengguna website. Untuk membuat form menggunakan tag `<form>`. Tag `<form>` merupakan sebuah container yang berisi tag `<input>`.

Tipe	Deskripsi
<code>&lt;input type="text"&gt;</code>	Menampilkan kolom input satu baris
<code>&lt;input type="radio"&gt;</code>	Menampilkan tombol radio(pilih salah satu)
<code>&lt;input type="checkbox"&gt;</code>	Menampilkan CheckBox(pilih lebih dari satu)
<code>&lt;input type="submit"&gt;</code>	Menampilkan tombol submit
<code>&lt;input type="button"&gt;</code>	Menampilkan button

```
<form>
  <label for="fname">First name:</label><br>
  <input type="text" id="fname" name="fname"><br>
  <label for="lname">Last name:</label><br>
  <input type="text" id="lname" name="lname">
</form>
```

First name:

Last name:

Perhatikan penggunaan tag `<label>` pada contoh di atas. Tag `<label>` mendefinisikan label untuk banyak elemen formulir.

```
<form>
  <input type="radio" id="html" name="fav_language" value="HTML">
  <label for="html">HTML</label><br>
  <input type="radio" id="css" name="fav_language" value="CSS">
  <label for="css">CSS</label><br>
  <input type="radio" id="javascript" name="fav_language"
value="JavaScript">
  <label for="javascript">JavaScript</label>
</form>
```

- ☐ HTML
- ☐ CSS
- ☐ JavaScript

```

<form>
  <input type="checkbox" id="vehicle1" name="vehicle1"
value="Bike">
  <label for="vehicle1"> I have a bike</label><br>
  <input type="checkbox" id="vehicle2" name="vehicle2" value="Car">
  <label for="vehicle2"> I have a car</label><br>
  <input type="checkbox" id="vehicle3" name="vehicle3"
value="Boat">
  <label for="vehicle3"> I have a boat</label>
</form>

```

☐ I have a bike  
☐ I have a car  
☐ I have a boat

```

<form action="/action_page.php">
  <label for="fname">First name:</label><br>

  <input type="text" id="lname" name="lname" value="Doe"><br><br>
  <input type="submit" value="Submit">
</form>

```

First name:

Doe

Submit

## 2.10. Tabel

Tabel merupakan gabungan kolom dan baris, membuat tabel pada HTML menggunakan tag <table>.

```

<table>
  <tr>
    <th>Company</th>
    <th>Contact</th>
    <th>Country</th>
  </tr>
  <tr>
    <td>Alfreds Futterkiste</td>
    <td>Maria Anders</td>
    <td>Germany</td>
  </tr>
  <tr>
    <td>Centro comercial Moctezuma</td>
    <td>Francisco Chang</td>
    <td>Mexico</td>
  </tr>
</table>

```

<tr> : table row

<th> : table header

<td> : table data

Company	Contact	Country
Alfreds Futterkiste	Maria Anders	Germany
Centro comercial Moctezuma	Francisco Chang	Mexico

## Bab 3 Cascading Style Sheet (CSS)

### 3.1. Pengenalan CSS

CSS adalah singkatan dari *cascading style sheets*, yaitu bahasa yang digunakan untuk menentukan tampilan dan format halaman website. Dengan CSS, Anda bisa mengatur jenis font, warna tulisan, dan latar belakang halaman.

### 3.2. Penulisan CSS

CSS dapat ditulis di dalam Tag HTML `<style> ... </style>` atau menggunakan file baru dengan **format** CSS `"Style.css"`.

Jika menambahkan file baru, maka kita harus menambahkan link file tersebut di dalam tag head. contoh :

```
1  <!DOCTYPE html>
2  <html lang="en">
3
4  <head>
5    <meta charset="UTF-8">
6    <title>Document</title>
7    <dumb></dumb>
8    <link rel="stylesheet" href="style.css">
9  </head>
10
11 <body>
12   <h1>
13 </body>
```

Berikut tahapan yang sudah kami rangkum menggunakan CSS untuk pemula.

### 3.3. Styling CSS

CSS (Cascading Style Sheets) adalah bahasa pemrograman yang digunakan untuk mengatur tata letak dan tampilan elemen HTML pada suatu halaman web. CSS memungkinkan pengembang web untuk mengontrol gaya, warna, ukuran, jarak, dan berbagai properti lainnya dari elemen HTML.

Penjelasan beberapa konsep dasar dan contoh-contoh styling CSS:

### 1. Font Style

font-style pada CSS adalah properti yang digunakan untuk mengatur atau menentukan gaya teks pada suatu elemen. Properti ini memungkinkan kita untuk mengubah penampilan teks menjadi bentuk yang lebih khusus, seperti membuatnya menjadi italic, normal, atau oblique.

Contoh penggunaan:

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

Dalam contoh ini, semua elemen `<p>` pada halaman HTML akan memiliki gaya teks yang diatur sebagai italic.

### 2. Font Size

Font-size adalah properti CSS yang digunakan untuk menentukan ukuran teks pada suatu elemen. Properti ini memungkinkan kita mengatur seberapa besar atau kecil teks akan ditampilkan pada halaman web.

Contoh penggunaan:

```
h1 {  
  font-size: 36px;  
}
```

Dalam contoh ini, teks dalam elemen `<h1>` akan memiliki ukuran 36 piksel. Properti font-size dapat diatur dengan berbagai satuan pengukuran, seperti piksel (px), em, rem, persen (%), dan lainnya.

### 3. Font Color

Color pada CSS adalah properti yang digunakan untuk menentukan warna teks pada suatu elemen. Properti ini memungkinkan kita mengatur warna teks sesuai dengan preferensi desain halaman web.

Contoh penggunaan:

```
body {  
  color: #333; /* Warna teks abu-abu tua */  
}
```

Pemilihan warna yang tepat dapat meningkatkan estetika dan membantu menyampaikan pesan visual dengan lebih efektif pada halaman web.

#### 4. Border

adalah properti CSS yang digunakan untuk menentukan garis batas atau tepi suatu elemen. Properti ini memungkinkan kita mengatur ketebalan, jenis, dan warna garis batas elemen tersebut. contoh:

```
div {  
  
border: 2px solid #ccc;  
  
}
```

Dalam contoh ini, elemen <div> akan memiliki garis batas dengan ketebalan 2 piksel, jenis solid, dan warna abu-abu muda (#ccc). Properti border dapat diatur dengan tiga nilai utama:

##### a) ketebalan (Border-width) menentukan ketebalan border

contoh:

```
p {  
  
border-width: 1px;}  

```

##### b) Jenis (border-style): Menentukan jenis garis batas, seperti solid, dashed, dotted, dan lainnya.

Contoh:

```
img {  
  
border-style: dashed;  
  
}
```

c) Warna (border-color) digunakan untuk menentukan garis batas. contoh:

```
a {  
  
border-color: #ff0000;  
  
}
```

Kita juga dapat menggabungkan ketiga properti diatas menjadi satu dengan menggunakan tag border seperti pada contoh pertama agar menjadi lebih ringkas.

## 5. Background

Background merupakan properti CSS yang digunakan untuk mengatur latar belakang suatu elemen HTML. Properti ini mencakup beberapa sub-properti yang memungkinkan kita menentukan warna latar belakang, gambar latar belakang, pengaturan posisi latar belakang, dan lainnya.

Contoh penggunaan:

```
section {  
  
background-color: #f0f0f0;  
  
}
```

Dalam contoh ini, elemen<section> akan memiliki latar belakang dengan warna abu-abu pucat (#f0f0f0;). Beberapa sub-properti background yang umum digunakan:

- Background color
- Background image
- Background repeat
- Background position
- Background size

## 6. Height

Height adalah properti CSS yang digunakan untuk menentukan tinggi suatu elemen. Properti ini memungkinkan kita mengatur dimensi vertikal elemen HTML.

contoh penggunaan :

```
img {  
  
height: 200px;}
```



Dalam contoh ini, elemen `<img>` akan memiliki tinggi sebesar 200 piksel. Properti `height` dapat diatur menggunakan berbagai satuan pengukuran, seperti piksel (px), persen (%), viewport height (vh), dan lainnya.

## 7. Width

Width adalah properti CSS yang digunakan untuk menentukan lebar suatu elemen. Properti ini memungkinkan kita mengatur dimensi horizontal elemen HTML.

contoh penggunaan:

```
.container {  
width: 80%;  
}
```

Dalam contoh ini, elemen dengan kelas `.container` akan memiliki lebar setara dengan 80% lebar elemen induknya. Properti `width` dapat diatur menggunakan berbagai satuan pengukuran, seperti piksel (px), persen (%), viewport width (vw), dan lainnya

## 3.4. Position CSS

`position` adalah properti CSS yang digunakan untuk mengatur metode penempatan suatu elemen di dalam halaman web. Properti ini memungkinkan Anda mengontrol posisi elemen relatif terhadap elemen lain atau terhadap viewport (area tampilan seluruh halaman).

### 1. Padding

`padding` adalah properti CSS yang digunakan untuk menentukan ruang tambahan di sekitar isi suatu elemen, di antara konten elemen dan batas (border) elemen tersebut. Properti ini memberikan ruang internal di dalam elemen, sehingga mempengaruhi tata letak dan penataan elemen.

Contoh penggunaan:

```
div {  
padding: 20px;  
}
```

Dalam contoh ini, elemen `<div>` akan memiliki ruang `padding` sebesar 20 piksel di sekitar kontennya. Properti `padding` dapat diatur untuk keempat sisi elemen (atas, kanan, bawah, kiri) atau untuk sisi-sisi tertentu.

## 2. Margin

margin adalah properti CSS yang digunakan untuk menentukan ruang di sekitar suatu elemen, di antara batas elemen tersebut dengan elemen-elemen lainnya. Ruang ini memberikan jarak antara elemen dan elemen-elemen lain dalam tata letak halaman web.

Contoh penggunaan:

```
p {  
margin: 10px;  
}
```

Dalam contoh ini, elemen <p> akan memiliki ruang margin sebesar 10 piksel di sekitarnya. Properti margin dapat diatur untuk keempat sisi elemen (atas, kanan, bawah, kiri) atau untuk sisi-sisi tertentu.

## 3. Float

float adalah properti CSS yang digunakan untuk mengatur apakah suatu elemen akan "mengapung" (float) ke sisi kiri atau kanan dari elemen induknya. Elemen yang diapit oleh elemen yang mengapung akan berusaha mengelilingi elemen yang mengapung tersebut.

Contoh penggunaan:

```
img {  
float: left;  
}
```

Dalam contoh ini, elemen <img> akan mengapung ke sisi kiri dan elemen-elemen berikutnya akan mencoba mengelilingi elemen tersebut di sebelah kanan.

Nilai properti float:

- left: Elemen akan mengapung ke sisi kiri.
- right: Elemen akan mengapung ke sisi kanan.
- none: Ini adalah nilai default. Elemen tidak akan mengapung.
- inherit: Elemen akan mengambil nilai float dari elemen induknya.

Contoh penggunaan nilai right:

```
aside {  
float: right;  
}
```

Penting untuk dicatat bahwa penggunaan properti float memiliki beberapa dampak pada tata letak halaman dan dapat menyebabkan elemen-elemen lain "mengikuti" elemen yang mengapung. Oleh karena itu, teknik lain seperti menggunakan flexbox atau grid seringkali lebih direkomendasikan untuk tata letak modern. Properti float lebih sering digunakan untuk pengaturan kolom-kolom pada desain lama.

#### 4. Align

text-align adalah properti CSS yang digunakan untuk mengatur penataan teks dalam suatu elemen atau kontainer. Properti ini bisa digunakan untuk mengatur posisi horizontal teks dalam elemen tersebut.

Contoh penggunaan:

```
div {  
text-align: center;  
}
```

#### 5. Z-Index

Z-index adalah properti CSS yang digunakan untuk mengontrol tumpukan atau kedalaman tampilan elemen. Properti ini menentukan urutan tampilan suatu elemen dalam tumpukan elemen-elemen pada halaman web. Elemen dengan z-index yang lebih tinggi akan muncul di depan elemen-elemen dengan z-index yang lebih rendah.

Contoh penggunaan:

```
.overlay {  
z-index: 2;  
}
```

Dalam contoh ini, elemen dengan kelas `.overlay` akan memiliki urutan tampilan lebih tinggi (z-index 2) daripada elemen-elemen sekitarnya. Properti ini sering digunakan untuk mengontrol tampilan elemen-elemen seperti overlay, tooltip, atau pop-up agar dapat muncul di depan elemen-elemen lainnya. Z-index hanya memiliki efek pada elemen-elemen yang memiliki posisi (position) selain static. Oleh karena itu, seringkali diperlukan kombinasi dengan properti position. Nilai z-index yang lebih tinggi tidak selalu membuat elemen tersebut lebih besar atau lebih besar secara fisik, tetapi hanya mengatur tumpukan atau tata letak tampilannya.

```
.container {
```

```
position: relative;
```

```
}
```

```
.popup {
```

```
position: absolute;
```

```
z-index: 3;
```

```
}
```

Dalam contoh ini, elemen dengan kelas `.popup` yang berada di dalam elemen dengan kelas `.container` akan memiliki urutan tampilan yang lebih tinggi.

### 3.5. Box Model CSS

Box Model pada CSS adalah konsep fundamental yang digunakan untuk merinci struktur elemen-elemen HTML sebagai sebuah kotak yang terdiri dari beberapa komponen utama. Setiap elemen dianggap sebagai kotak dengan empat bagian utama: content, padding, border, dan margin.

Komponen Box Model:

a. Content (Isi):

Ini adalah area tempat konten atau isi sebenarnya dari elemen ditempatkan (teks, gambar, dll.).

b. Padding (Ruangan Dalam):

Ini adalah ruang tambahan di sekitar isi elemen. Padding terletak di antara konten dan border.

c. Border (Batas):

Ini adalah garis batas yang mengelilingi padding. Border memiliki ketebalan, jenis (solid, dashed, dll.), dan warna.

d. Margin (Ruangan Luar):

Ini adalah ruang tambahan di sekitar elemen, di antara border elemen dan elemen-elemen tetangganya. Margin memberikan jarak antara elemen ini dengan elemen-elemen lain pada halaman.

Contoh Penggunaan:

```
.box {  
width: 200px;  
height: 100px;  
padding: 10px;  
border: 2px solid #333;  
margin: 20px;  
}
```

Dalam contoh ini, elemen dengan kelas `.box` akan memiliki lebar 200 piksel, tinggi 100 piksel, padding 10 piksel, border dengan ketebalan 2 piksel dan jenis solid berwarna hitam (#333), serta margin 20 piksel di sekitarnya.

Penting untuk memahami Model Kotak karena pengaturan ukuran, margin, padding, dan border memengaruhi tata letak elemen dan keseluruhan desain halaman web. Model Kotak menyediakan kontrol yang kuat dalam merancang tata letak dan penataan elemen.

## BAB 4 JAVASCRIPT

### 4.1. Pengenalan Javascript

#### a. Sejarah singkat javascript

JavaScript atau sering disingkat JS, pertama kali dibuat tahun 1995. Programmer di Netscape bernama Brandan Eich membuat sebuah bahasa pemrograman scripting untuk berjalan di Netscape Navigator Browser. Awalnya namanya adalah Mocha, lalu berubah menjadi LiveScript, dan akhirnya menjadi JavaScript.

JavaScript dibuat agar halaman website yang sudah kita buat menggunakan HTML dan CSS bisa lebih interaktif JavaScript sekarang sudah mengalahkan Java Applet dan Flash sebagai bahasa pemrograman untuk membuat halaman web menjadi lebih interaktif, hal ini dikarenakan kemudahan bahasa nya dan juga secara default sekarang semua browser sudah bisa menjalankan JavaScript tanpa harus menginstall aplikasi tambahan seperti Java Applet dan Adobe Flash Player

#### b. Peralatan Development

Saat kita belajar JavaScript, kita perlu menyiapkan beberapa perangkat lunak untuk membantu development. Browser, ini sudah pasti, karena kita perlu menjalankan kode program JavaScript menggunakan Browser. Text Editor atau Integrated Development Environment, ini digunakan untuk membuat kode program JavaScript.

#### c. Browser

- Google Chrome
- Firefox
- Microsoft Edge
- Safari
- Opera

#### e. Text Editor

- Visual Studio Code : <http://code.visualstudio.com/>
- Atom : <https://atom.io/>
- Sublime : <https://www.sublimetext.com/>
- WebStorm : <https://www.jetbrains.com/webstorm/>
- Notepad++ : <https://notepad-plus-plus.org/>
- Textmate : <https://macromates.com/>
- GNU Emacs : <https://www.gnu.org/software/emacs/>
- GEdit : <https://wiki.gnome.org/Apps/Gedit>

## Program Hello World

Ada beberapa cara untuk membuat kode JavaScript. Bisa langsung di file HTML. Atau bisa menggunakan file .js (ekstensi untuk JavaScript), lalu di include di dalam file HTML. Pada praktikum ini kita akan menggunakan HTML langsung agar mudah membuat kode program nya.

index.html

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>JAVASCRIPT TUTORIAL</title>
7 </head>
8 <body>
9   <script src="script/hello-world.js"></script>
10 </body>
11 </html>
```

script.js

```
script > JS script.js
1 console.log("Hello World");
2
```

## 4.2. Tipe Data

### a. Tipe Data Number

JavaScript hanya mendukung satu tipe data number, dimana tipe data number di JavaScript bisa berupa bilangan bulat atau bilangan desimal. Tipe seperti di bahasa pemrograman lain yang biasanya membedakan tipe data number bilangan bulat dan bilangan desimal, di JavaScript semuanya disatukan

```
<body>
  <!-- Tipe data Number dalam javascript -->
  <script>
    console.log(100)
    console.log(100.200)
  </script>
</body>
```

#### b. Number Notation

JavaScript mendukung number notation, defaultnya ada basis 10, JavaScript juga mendukung binary, hexadecimal dan octal

- Hexadecimal : 0xFF
- Binary : 0b10101
- Octal : 0o10

#### c. Tipe Data Boolean

Tipe data boolean adalah tipe data yang berisikan data kebenaran, artinya hanya ada dua data, benar dan salah (yes atau no). Benar di representasikan dengan kata kunci true, dan salah direpresentasikan dengan kata kunci false. Walaupun tipe data boolean merupakan tipe data yang sangat sederhana, tapi tipe data boolean akan banyak digunakan di berbagai fitur pemrograman

```
<body>
  <script>
    document.writeln(true)
    document.writeln("<br>")
    document.writeln(false)
  </script>
</body>
```

```
<body>
  <script>
    console.log(true)
    console.log(false)
  </script>
</body>
```

#### d. Tipe Data String

Tipe data string atau text adalah tipe data yang berisikan kumpulan kosong atau lebih karakter. Sebelumnya kita sudah pernah menggunakan string ketika membuat program hello world. Di JavaScript, untuk membuat data dengan tipe string, kita perlu menggunakan “ (petik dua) atau ‘ (petik satu) sebelum dan setelah isi text nya



```

<body>
  <script>
    document.writeln("Tegar Raihan Akmali");
    document.writeln("<br>");
    document.writeln("Pemrograman Web");
  </script>
</body>

```

e. Menambah String

Data string juga bisa digabungkan menggunakan operator + (plus)

```

<body>
  <script>
    //Menggabungkan String menggunakan Tipe Data +
    document.writeln("<br>");
    document.writeln("Tegar " + "Raihan " + "Akmali");
  </script>
</body>

```

#### 4.3. Variable

Variable adalah tempat untuk menyimpan data. Dengan menyimpan data di variable, kita bisa menggunakannya lagi dengan menyebutkan nama variable nya. Untuk membuat variable di JavaScript, kita bisa menggunakan kata kunci var diikuti dengan nama variable nya. JavaScript adalah dynamic language, artinya variable di JavaScript tidak terpaku harus menggunakan satu tipe data, kita bisa mengubah-ubah tipe data di variable yang sama

```

//Deklarasi Variable
var namaLengkap;

```

a. Membuat Variable Langsung Dengan Value

Di JavaScript juga kita bisa mendeklarasikan sebuah variable, langsung dengan isi value nya. Caranya kita bisa menggunakan kata kunci var, diikuti nama variable, lalu diikuti dengan tanda = (sama dengan), dan di ikuti dengan value atau nilai nya

```
//Membuat Variable Dengan Value  
var namaLengkap = "Tegar Raihan";  
var namaLengkap = "Akmali";  
var namaLengkap = "Tegar" + " " + "Raihan" + " " + "Akmali";
```

b. Mengubah Value di Variable

Setelah variable di deklarasikan, kita bisa mengubah value atau nilai di dalam variable tersebut. Untuk mengubahnya, kita bisa menggunakan perintah nama variable diikuti dengan tanda = (sama dengan) lalu diikuti dengan value atau nilai nya

```
//Mengubah Value Di Variable  
var namaLengkap = "Tegar Raihan Akmali";  
var namaLengkap = "Tegar Raihan";
```

c. Mengakses Variable

Salah satu keuntungan menggunakan variable adalah variable bisa digunakan kembali. Hal ini akan mempermudah ketika membutuhkan data yang sama berkali-kali. Untuk mengakses variable, kita cukup menyebutkan nama variable nya

```
//Mengubah Value Di Variable  
var namaLengkap = "Tegar Raihan Akmali";  
var namaLengkap = "Tegar Raihan";
```

```
//Mengakses Variable  
console.log(namaLengkap);
```



#### d. Variable Const

Kata kunci let itu seperti kata kunci var, dimana data di variable tersebut bisa diubah-ubah sesuka kita. Sedangkan kata kunci const berbeda, ketika sebuah variable sudah diisi di variable const, maka variable tersebut tidak bisa diubah lagi value nya. Variable sejenis ini kadang dibilang juga constant

```
//Variable const  
const npm = "213510040";  
console.log(npm);
```

### 4.4. Operator

#### a. Operator Aritmatika

JavaScript mendukung banyak sekali operator Matematika untuk tipe data Number, seperti:

- Operator Aritmatika
- Operator Augmented Assignments
- Operator Unary
- Dan lain-lain

Operator	Keterangan
+	Pertambahan
-	Pengurangan
*	Perkalian
**	Exponensial
/	Pembagian
%	Sisa Bagi

```
var penjumlahan = 1 + 2;  
console.log("Hasil dari 1 + 2 = " + penjumlahan);
```

## b. Operator Assignments

Assignment Operator (operator penugasan) adalah operator yang menggunakan tanda sama dengan (=) untuk mengisi sebuah nilai dalam suatu variabel.

Operator Matematika	Operator Augmented Assignments
<code>result = result + 10</code>	<code>result += 10</code>
<code>result = result - 10</code>	<code>result -= 10</code>
<code>result = result * 10</code>	<code>result *= 10</code>
<code>result = result ** 10</code>	<code>result **= 10</code>
<code>result = result / 10</code>	<code>result /= 10</code>
<code>result = result % 10</code>	<code>result %= 10</code>

//Menggunakan Operator Augmented Assignments

**let hasil = 10;**

**hasil += 10;**

**console.log("Hasil dari 10 + 10 = " + hasil);**

## c. Operator Unary

Operator unary adalah operator yang digunakan untuk melakukan operasi-operasi matematik yang hanya melibatkan satu buah operand.

Operator	Keterangan
<code>+</code>	Menandakan nilai positif
<code>-</code>	Menandakan nilai negatif
<code>++</code>	Increment, menaikkan 1 angka
<code>--</code>	Decrement, menurunkan 1 angka

// Operator Unary

```
let hasil = +1;
console.log("Hasil :" + hasil)

hasil--;
console.log("Hasil :" + hasil)

hasil++;
console.log("Hasil :" + hasil);

hasil++;
console.log("Hasil :  " + hasil)
```

d. Operator Perbandingan

Operasi perbandingan adalah operasi untuk membandingkan dua buah data. Operasi perbandingan adalah operasi yang menghasilkan nilai boolean (benar atau salah). Jika hasil operasinya adalah benar, maka nilainya adalah true. Jika hasil operasinya adalah salah, maka nilainya adalah false

```
<script>
let hasil = 5 == "t";
console.log("Hasil :" + hasil);

result = 5 === "5";
console.log("Hasil :" + result);

result = 5 > 10;
console.log("Hasil :" + result);

result = 10 > 5;
console.log("Hasil :" + result);
</script>
```

Operator	Keterangan
>	Lebih Dari
<	Kurang Dari
>=	Lebih Dari Sama Dengan
<=	Kurang Dari Sama Dengan
==	Sama Dengan
===	Sama Dengan dan Sama Tipe
!=	Tidak Sama Dengan
!==	Tidak Sama Dengan atau Tidak Sama Tipe

e. Operator Logika

Operator	Keterangan
&&	Dan
	Atau
!	Kebalikan

f. Operator and

Nilai 1	Operator	Nilai 2	Hasil
true	&&	true	true
true	&&	false	false
false	&&	true	false
false	&&	false	false

g. Operator or ||

Nilai 1	Operator	Nilai 2	Hasil
true		true	true
true		false	true
false		true	true
false		false	false

h. Operator Unary !

Operator	Nilai 2	Hasil
!	true	false
!	false	true



```

<script>
  const nilaiUjian = 70;
  const nilaiAbsensi = 70;

  const lulusUjian = nilaiUjian > 75;
  const lulusAbsensi = nilaiAbsensi > 75;

  const lulus = lulusUjian && lulusAbsensi;
  console.log('Hasil : ' + lulus);
</script>

```

#### 4.5. Conditional Statement

Sering kali ketika menulis kode, Kita ingin melakukan tindakan berbeda untuk keputusan berbeda. Anda dapat menggunakan pernyataan kondisional dalam kode Anda untuk melakukan ini. Dalam JavaScript kami memiliki pernyataan kondisional berikut :

- Gunakan if untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang ditentukan benar
- Gunakan else untuk menentukan blok kode yang akan dieksekusi, jika kondisi yang sama salah
- Gunakan else if untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama salah
- Gunakan switch untuk menentukan banyak blok kode alternatif yang akan dieksekusi

```

<script>
  let nilai = 70;

  if(nilai > 75){
    console.log('Selamat Anda Lulus');
  } else{
    console.log('Tidak Lulus');
  }
</script>

```

## JavaScript Switch

Gunakan switch pernyataan tersebut untuk memilih salah satu dari banyak blok kode yang akan dieksekusi.

- Ekspresi switch dievaluasi satu kali.
- Nilai ekspresi dibandingkan dengan nilai setiap kasus.
- Jika ada kecocokan, blok kode terkait akan dieksekusi.
- Jika tidak ada kecocokan, blok kode default dijalankan.

```
<script>
  const marks = 85;

  let Branch;

  switch (true) {
    case marks >= 90:
      Branch = "Computer science engineering";
      break;
    case marks >= 80:
      Branch = "Mechanical engineering";
      break;
    case marks >= 70:
      Branch = "Chemical engineering";
      break;
    case marks >= 60:
      Branch = "Electronics and communication";
      break;
    case marks >= 50:
      Branch = "Civil engineering";
      break;
    default:
      Branch = "Bio technology";
      break;
  }
  console.log(Branch)
</script>
```

### 4.6. Array

Array adalah variabel khusus yang dapat menampung lebih dari satu nilai:

```
const cars = ["Saab", "Volvo", "BMW"];
```

## Mengapa Menggunakan Array?

Jika Anda memiliki daftar item (misalnya daftar nama mobil), menyimpan mobil dalam variabel tunggal akan terlihat seperti ini:

```
let car1 = "Saab";  
let car2 = "Volvo";  
let car3 = "BMW";
```

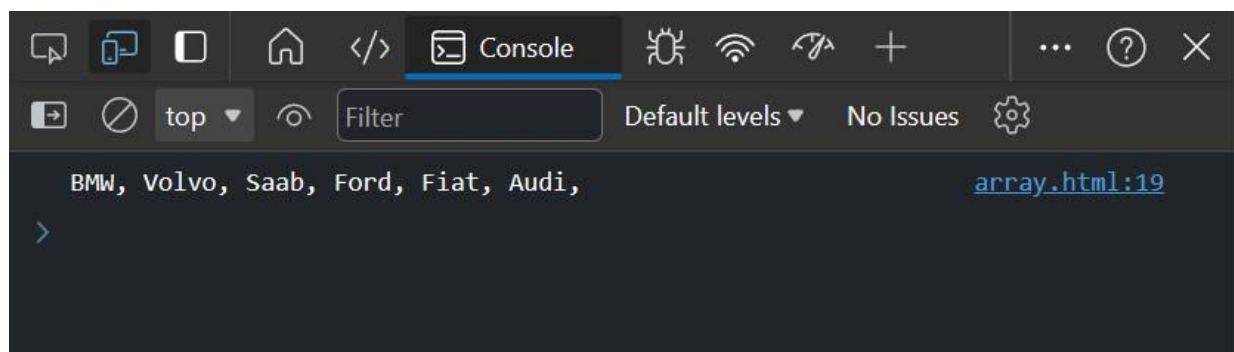
Namun, bagaimana jika Anda ingin menelusuri mobil-mobil tersebut dan menemukan mobil tertentu? Dan bagaimana jika Anda tidak punya 3 mobil, tapi 300?

Solusinya adalah sebuah array!

### 4.7. Looping

Perulangan berguna jika Anda ingin menjalankan kode yang sama berulang kali, setiap kali dengan nilai berbeda.

```
<script>  
const cars = ["BMW", "Volvo", "Saab", "Ford", "Fiat", "Audi"];  
let text = "";  
  
for (let i = 0; i < cars.length; i++) {  
  text += cars[i] + ", ";  
}  
  
console.log(text)  
</script>
```



## 4.8. DOM

### Apa itu DOM HTML?

HTML DOM adalah model objek standar dan antarmuka pemrograman untuk HTML. Ini mendefinisikan:

- Elemen HTML sebagai objek
- Properti semua elemen HTML
- Metode untuk mengakses semua elemen HTML
- Peristiwa untuk semua elemen HTML

Dengan kata lain: HTML DOM adalah standar cara mendapatkan, mengubah, menambah, atau menghapus elemen HTML.

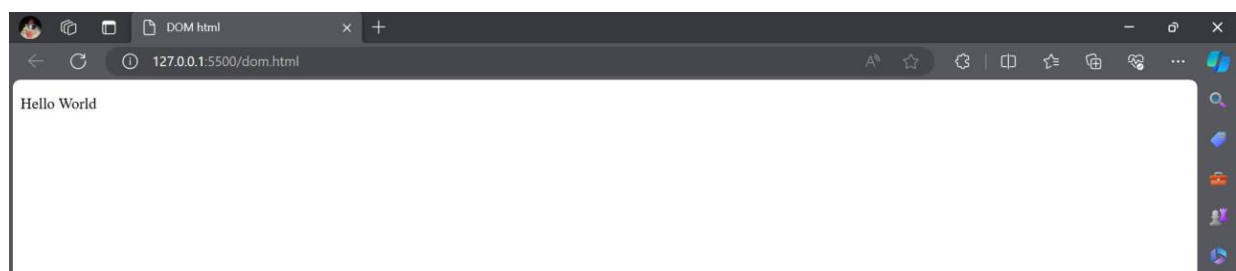
### Antarmuka Pemrograman DOM

- HTML DOM dapat diakses dengan JavaScript (dan dengan bahasa pemrograman lainnya).
- Di DOM, semua elemen HTML didefinisikan sebagai objek.
- Antarmuka pemrograman adalah properti dan metode setiap objek.
- Properti adalah nilai yang bisa Anda dapatkan atau tetapkan (seperti mengubah konten elemen HTML).
- Metode adalah tindakan yang dapat Anda lakukan (seperti menambah atau menghapus elemen HTML) .

Contoh:

```
<p id="demo"> </p>
```

```
<script>  
  document.getElementById('demo').innerHTML = "Hello World"  
</script>
```



Agar lebih menarik disini ada beberapa contoh code javascript untuk membuat efek text seperti mengetik

index.html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta http-equiv="X-UA-Compatible" content="IE=edge">
6    <meta name="viewport" content="width=device-width, initial-scale=1.0">
7    <title>Personal Portofolio Website</title>
8    <link rel="stylesheet" href="style.css">
9  </head>
10 <body>
11   <div class="hero">
12     <nav>
13       <img src="" class="logo">
14       <ul>
15         <li><a href="#">HOME</a></li>
16         <li><a href="#">ABOUT</a></li>
17         <li><a href="#">PORTOFOLIO</a></li>
18         <li><a href="#">SERVICE</a></li>
19         <li><a href="#">HIRE ME</a></li>
20       </ul>
21     </nav>
22     <div class="detel">
23       <h1>I,m Nurfauzan <span>Hanif</span></h1>
24       <p>This is my official portofolio website to showes all<br> Detail and work experience</p>
25       <a href="#">DOWNLOAD CV</a>
26     </div>
27     <div class="images">
28       
29       
30     </div>
31   </div>
32   <script src="script.js"></script>
33 </body>
34 </html>
```

script.js

```
1  document.addEventListener('DOMContentLoaded', function () {
2    var p = "This is my official portofolio website to showes all Detail and work experience";
3    var element = document.querySelector('.detel p');
4    var speedWrite = 50;
5
6    function typeWriter(element, p, speedWrite) {
7      var i = 0;
8
9      function type() {
10        if (i < p.length) {
11          element.innerHTML += p.charAt(i);
12          i++;
13          setTimeout(type, speedWrite);
14        }
15      }
16      type();
17    }
18
19    element.innerHTML = ''; // Clear existing content before typing
20    setTimeout(function () {
21      typeWriter(element, p, speedWrite);
22    }, element.innerHTML.length * speedWrite + 500);
23  });
24
```

### 1. Document Ready Event Listener:

```
document.addEventListener('DOMContentLoaded', function () {
```

- Ini adalah event listener yang akan menunggu sampai seluruh dokumen HTML telah dimuat sebelum menjalankan koding di dalamnya.

### 2. Variable

```
var p = "This is my official portofolio website to showes all Detail and work experience";  
var element = document.querySelector('.detel p');  
var speedWrite = 50;
```

- p: Variabel yang berisi teks yang akan ditampilkan dengan efek mengetik.
- element: Variabel yang menyimpan referensi ke elemen <p> di dalam elemen dengan kelas .detel, di mana teks akan ditampilkan.
- speedWrite: Variabel yang menyimpan kecepatan penulisan karakter, dalam mili detik.

### 3. Fungsi typeWriter:

```
function typeWriter(element, p, speedWrite) {  
  var i = 0;  
  function type() {  
    if (i < p.length) {  
      element.innerHTML += p.charAt(i);  
      i++;  
      setTimeout(type, speedWrite);  
    }  
  }  
  type();  
}
```

- typeWriter: Fungsi yang akan menampilkan teks secara bertahap ke dalam elemen HTML dengan efek mengetik.
- i: Variabel yang digunakan untuk melacak posisi saat ini dalam teks p.
- type: Fungsi internal yang bertanggung jawab untuk menambahkan karakter satu per satu ke dalam elemen HTML, menggunakan setTimeout untuk mengatur kecepatan penulisan.

#### 4. Memanggil Fungsi typeWriter:

```
element.innerHTML = ""; // Menghapus konten sebelumnya sebelum mengetik
setTimeout(function () {
    typeWriter(element, p, speedWrite);
}, element.innerHTML.length * speedWrite + 500);
```

- Pertama, konten sebelumnya di dalam elemen element dihapus.
- Kemudian, menggunakan setTimeout untuk memanggil fungsi typeWriter. Timeout ini ditetapkan berdasarkan panjang konten sebelumnya dalam elemen element, sehingga efek mengetik dimulai setelah konten sebelumnya dihapus.