

MODUL PRAKTIKUM  
PEMROGRAMAN BERBASIS WEB



Disusun Oleh:

Budi Setiawan

LABORATORIUM KOMPUTER  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS SINGAPERBANGSA KARAWANG  
2021

# DAFTAR ISI

|  |    |
|--|----|
| DAFTAR ISI .....   | 2  |
| PERTEMUAN 1 HTML & CSS .....                               | 3  |
| PERTEMUAN 2 HTML & CSS .....                               | 20 |
| PERTEMUAN 3 JAVASCRIPT DASAR .....                         | 32 |
| PERTEMUAN 4 BOOTSTRAP .....                                | 43 |
| PERTEMUAN 5 GITHUB.....                                    | 55 |
| PERTEMUAN 6 PHP DASAR.....                                 | 64 |
| PERTEMUAN 7 PHP MYSQL (CRUD).....                          | 76 |
| PERTEMUAN 8 PHP MYSQL AUTHENTICATION & AUTHORIZATION ..... | 84 |
| PERTEMUAN 9 PHP OOP .....                                  | 91 |
| PERTEMUAN 10 PHP Restful API .....                         | 96 |

# PERTEMUAN 1

## HTML & CSS

### 1.1 Pengertian HTML

HTML merupakan singkatan dari Hypertext Markup Language adalah bahasa markup yang digunakan untuk membuat halaman web. HTML berfungsi untuk membuat struktur dari sebuah website. Dengan HTML dapat membuat paragraf, judul, tautan, tabel dan lain sebagainya di halaman website.

### 1.2 Struktur Dasar HTML

Di dalam HTML terdapat Tag, Atribut, dan Element. Tag adalah sebuah penanda awalan dan akhiran dari sebuah element, sebagian besar tag HTML ditulis secara berpasangan. Atribut adalah informasi tambahan yang ditulis pada tag pembuka, fungsi dari atribut tergantung nilai dan tag mana ia ditempatkan. Element adalah gabungan dari tag, atribut, dan teks yang berada di antara tag pembuka dan tag penutup. Berikut merupakan struktur dasar dari HTML:

Contoh 1: contoh\_struktur\_dasar\_html.html

---

```
<!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>Pemrograman Berbasis Web</title>
</head>
<body>
  <h1>Mata Kuliah Pemrograman Berbasis Web</h1>
  <p>Belajar HTML Dasar</p>
</body>
</html>
```

---

### 1.3 Pemformatan Teks

#### a. Paragraf

Untuk membuat paragraph menggunakan tag <p>

Contoh 2: paragraph.html

---

```
<p>HTML merupakan singkatan dari Hypertext Markup Language
adalah bahasa markup yang digunakan untuk membuat halaman
web.</p>
```

---

## b. Heading

Di dalam HTML heading digunakan untuk membuat judul, terdapat 6 level heading yang bisa digunakan, yakni <h1> sampai dengan <h6>

Contoh 3: heading.html

---

```
<h1>Ini merupakan heading level 1</h1>
<h2>Ini merupakan heading level 2</h2>
<h3>Ini merupakan heading level 3</h3>
<h4>Ini merupakan heading level 4</h4>
<h5>Ini merupakan heading level 5</h5>
<h6>Ini merupakan heading level 6</h6>
```

---

## c. Tautan

Tautan digunakan untuk membuat link ke halaman lain dan juga bisa digunakan untuk membuat link ke gambar, file, dan lain sebagainya. Untuk membuat tautan dapat menggunakan tag <a>. Terdapat 2 syarat untuk membuat tautan di HTML yakni teks yang berfungsi sebagai teks link dan alamat tujuan link (URL).

Contoh 4: tautan.html

---

```
<a href="https://unsika.ac.id/">Universitas Singaperbangsa
Karawang</a>
```

---

Perlu diperhatikan terdapat 2 jenis alamat dalam tautan yakni alamat absolut dan alamat relatif. Alamat absolut merupakan alamat yang mencantumkan URL secara lengkap, yakni terdiri dari nama protocol, nama domain, dan nama file. Sedangkan alamat relatif merupakan alamat yang relatif ke file tujuan berdasarkan struktur folder.

Alamat absolut:

Contoh 5: tautan\_absolut.html

---

```
<a href="https://unsika.ac.id/">Universitas Singaperbangsa
Karawang</a>
```

---

Alamat relatif:

Contoh 6: tautan\_relatif.html

---

```
<a href="auth/login.html">Login</a>
<a href="auth/mahasiswa/lihat.html">Data Mahasiswa</a>
<a href="index.html">Beranda</a>
<a href=" ../logout.html">Logout</a>
```

---

Di dalam tautan terdapat atribut target yang bisa digunakan untuk menentukan pada jendela mana halaman web akan ditampilkan. Ada 4 nilai yang bisa digunakan yakni \_self, \_blank, \_parent, dan \_top

## \_self

Contoh 7: tautan\_self.html

---

```
<a href="https://unsika.ac.id/" target="_self">Universitas  
Singaperbangsa Karawang</a>
```

---

## \_blank

Contoh 8: tautan\_blank.html

---

```
<a href="https://unsika.ac.id/" target="_blank">Universitas  
Singaperbangsa Karawang</a>
```

---

### d. Bold, Italic, dan Underline

Untuk membuat huruf tebal ada dua tag yang bisa digunakan yakni `<b>` dan tag `<strong>`. Kemudian untuk membuat huruf miring ada dua tag yang bisa digunakan yakni `<i>` dan `<em>`. Terakhir untuk membuat garis bawah pada teks terdapat dua tag yang bisa digunakan yakni `<u>` dan tag `<ins>`. Tidak ada perbedaan diantara keduanya sama-sama bisa untuk menebalkan huruf, membuat huruf miring, dan membuat garis bawah, hanya saja untuk kata-kata yang penting maka gunakan tag `<strong>`, `<em>`, atau `<ins>`. Sedangkan untuk kata yang tidak penting maka gunakan tag `<b>`, `<i>`, atau `<u>`.

#### Huruf tebal

Contoh 9: bold.html

---

```
<p>  
<strong>Pemformatan Teks itu penting!</strong> Karena dapat  
membuat tulisan terlihat lebih menarik sehingga akan membuat  
<b>pengunjung senang</b> ketika membacanya.  
</p>
```

---

#### Huruf Miring

Contoh 10: italic.html

---

```
<p>  
Gunakan <i>teks miring</i> untuk memberikan penekanan pada teks,  
sehingga akan <em>menarik perhatian</em> pembaca. Biasanya  
digunakan pada <i>istilah asing</i> atau kata serapan dari  
<em>bahasa daerah</em>.  
</p>
```

---

## Underline

Contoh 11: underline.html

---

```
<p>
<u>Text formatting itu penting</u>, karena dapat membuat teks
terlihat lebih menarik dibandingkan <ins>plain text</ins>.
</p>
```

---

### e. Preformatted

Preformatted yang berarti tidak diformat. Untuk menggunakan preformatted menggunakan tag `<pre>`. Ketika menggunakan tag ini maka teks yang berada diantara tag `<pre>` akan ditampilkan apa adanya termasuk spasi, enter, dan lain sebagainya.

Contoh 12: pre.html

---

```
<pre>
Cinta ibu dan ayah
Dalam dinginnya malam
Aku terbagun dan menjerit
Membayangkan ketakutan
Semua ketakutan itu sirna

Saat ibu dan ayah datang mendekat
Pelukah hangat keluarga
Menjadi pahlawanku
Dengan penuh cinta dan kasih sayang
</pre>
```

---

### f. Address

Tag `<address>` digunakan untuk membuat alamat

Contoh 13: address.html

---

```
<address>
  Anda bisa menghubungi kami di
  <a href="https://unsika.ac.id/">unsika.ac.id</a>
  <br>
  Jika ada kritik/saran/pertanyaan, bisa email ke
  <a href="mailto:info@unsika.ac.id">Info UNSIKA</a>
  <br>
  Alamat kami berada di Jl. HS.Ronggo Waluyo, Puseurjaya,
  Kec. Telukjambe Tim., Kabupaten Karawang, Jawa Barat 41361
</address>
```

---

### g. Blockquote

Tag `<blockquote>` digunakan untuk membuat sebuah kutipan

```
<blockquote>
    Technology is just a tool. In terms of getting the kids working
    together and motivating them, the teacher is the most
    important.
    <cite>Bill Gates</cite>
</blockquote>
```

---

## 1.4 Daftar

HTML menyediakan elemen untuk membuat daftar. Ada tiga macam jenis daftar yang bisa dibuat di HTML:

### a. Ordered List

Ordered list merupakan daftar yang berurutan. Ordered list menggunakan angka atau huruf sebagai penentu urutan nya. Ordered list cocok digunakan untuk daftar yang harus tersusun seperti instruksi atau daftar kegiatan. Terdapat dua tag untuk membuat daftar menggunakan ordered list, yang pertama kita harus menggunakan tag <ol> dan untuk membuat isinya kita menggunakan tag <li>

Contoh 15: ol.html

---

```
<h1>Cara Membuat Teh Manis</h1>

<h2>Bahan- bahan:</h2>

<ol>
    <li>Siapkan 1 teh celup.</li>
    <li>Siapkan gula secukupnya.</li>
    <li>Air hangat.</li>
</ol>

<h2>Alat- alat:</h2>

<ol>
    <li>Siapkan sebuah gelas.</li>
    <li>Lalu Siapkan juga sendok. </li>
</ol>

<h2>Cara membuat: </h2>

<ol>
    <li>Langkah awal yakni dengan memasukkan air hangat dengan
    takaran yang sesuai dengan keinginan.</li>
    <li>Langkah yang kedua ialah dengan menyelupkan teh gantung
    tersebut ke dalam gelas yang sudah terisi air.</li>
    <li>Tidak lupa juga untuk memasukkan gula secukupnya aduk air
    hingga gula larut.</li>
    <li>Jika semua langkah telah dilakukan, maka teh sudah siap
    untuk di sajikan.</li>
</ol>
```

---

b. Unordered List

Unordered list merupakan daftar yang tidak memerlukan urutan. Unordered list cocok digunakan untuk membuat menu. Untuk membuat unordered list kita harus menggunakan tag <ul> dan untuk membuat isinya menggunakan tag <li>

Contoh 16: ul.html

---

```
<p>Rekomendasi text editor</p>
<ul>
  <li>Visual Studio Code</li>
  <li>Sublime Text</li>
  <li>Notepad++</li>
  <li>Bracket</li>
</ul>
```

---

c. Description List

Description list merupakan daftar yang memiliki judul beserta keterangannya. Description list cocok digunakan untuk membuat kamus. Terdapat 3 (tiga) tag untuk membuat description list, yang pertama kita harus menggunakan tag <dl>, kemudian untuk membuat judul nya menggunakan tag <dt>, dan terakhir untuk membuat keterangan nya menggunakan tag <dd>

Contoh 17: dl.html

---

```
<dl>
  <dt>HTML</dt>
  <dd>Hypertext Markup Language</dd>
  <dt>CSS</dt>
  <dd>Cascading Style Sheet</dd>
  <dt>SQL</dt>
  <dd>Structured Query Language</dd>
</dl>
```

---

## 1.5 Pengertian CSS

CSS merupakan singkatan dari Cascading Style Sheet. CSS biasanya digunakan untuk mengubah atau mengatur tampilan halaman web. Dengan menggunakan CSS kita dapat mengatur seluruh tampilan dari HTML, seperti ukuran font, warna teks, gambar background, layout dan lain sebagainya.

## 1.6 Cara Menginput Kode CSS

Terdapat 3 cara untuk menginput kode CSS ke dalam HTML yaitu inline, internal dan external style CSS:

a. Inline

Untuk menggunakan cara ini kita langsung menuliskan kode css ke dalam atribut tag

Contoh 18: inline\_css.html



---

```
<p style="color: blue;font-size: 14px;">Saya sedang belajar  
css di mata kuliah Pemrograman Berbasis Web</p>
```

---

## b. Internal

Untuk menggunakan cara ini kita menuliskan kode css ke dalam tag <style> di dalam tag <head>

Contoh 19: internal\_css.html

---

```
<!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>Document</title>  
  <style>  
    p{  
      color: red;  
      font-size: 12px;  
      text-decoration: underline;  
    }  
  </style>  
</head>  
<body>  
  <h2>Belajar CSS</h2>  
  <p>Saya sedang belajar css di mata kuliah Pemrograman  
Berbasis Web</p>  
</body>  
</html>
```

---

## c. External

Untuk menggunakan cara ini kita membuat file css terpisah kemudian memanggilnya menggunakan tag <link>

Contoh 20: external\_css.html

---

```
<!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>Document</title>  
  <link rel="stylesheet" href="style.css">  
</head>  
<body>  
  <h2>Belajar CSS</h2>
```

```
<p>Saya sedang belajar css di mata kuliah Pemrograman  
Berbasis Web</p>  
</body>  
</html>
```

Contoh 21: style.css

---

```
p {  
    color: red;  
    font-size: 12px;  
    text-decoration: underline;  
}
```

---

## 1.7 Selector

Selector merupakan kata kunci yang digunakan untuk mencari suatu tag/elemen di dalam HTML, ada 8 macam selector di CSS yaitu:

### a. Universal Selector

CSS *Universal Selector* (\*) digunakan untuk menyeleksi setiap single element dari tipe element apapun. Artinya, penyeleksian berlaku secara keseluruhan untuk semua single element. Penggunaan *Universal Selector* ditandai dengan karakter asterik \* yang dapat digunakan secara berdiri sendiri ataupun digabungkan dengan tipe selektor lainnya.

Contoh 22: style.css

---

```
*{  
    font-size: 12px;  
    color: blue;  
    text-decoration: underline;  
    text-align: center;  
}
```

### b. Element Selector

Element selector merupakan salah satu selector yang digunakan untuk mencari semua tag HTML

Contoh 23: style.css

---

```
a{  
    text-decoration: none;  
    color: black;  
    font-size: 14px;  
}  
p{  
    color: blue;  
    font-size: 14px;  
}
```

### c. Class Selector

Class selector merupakan salah satu selector yang digunakan untuk mencari semua tag HTML yang memiliki atribut class di dalam tag tersebut. Penggunaan class selector ditandai dengan karakter (.). Class selector dapat digunakan untuk lebih dari satu tag.

Contoh 24: index.html

```
<p class="alert-success">Selamat kamu diterima menjadi Front  
End Developer</p>
```

Contoh 25: style.css

```
.alert-success{  
    color: green;  
    font-weight: bold;  
}
```

#### d. ID Selector

Id selector merupakan salah satu selector yang digunakan untuk mencari semua tag HTML yang memiliki atribut id di dalam tag tersebut. Penggunaan id selector ditandai dengan karakter (#). Id selector hanya dapat digunakan untuk satu tag.

Contoh 26: index.html

```
<p id="alert-danger">Mohon maaf! kamu tidak diterima menjadi  
Front End Developer</p>
```

Contoh 27: style.css

```
#alert-danger{  
    color: red;  
    font-weight: bold;  
}
```

#### e. Attribute Selector

Atribut selector merupakan salah satu selector yang digunakan untuk mencari semua tag HTML berdasarkan atributnya.

Contoh 28: index.html

```
<h2>Daftar Fakultas Universitas Singaperbangsa Karawang</h2>  
<ul>  
    <li>Fakultas Ilmu Komputer</li>  
    <li title="Fakultas Hukum">Fakultas Hukum</li>  
    <li title="Fakultas Teknik">Fakultas Teknik</li>  
    <li>Fakultas Ilmu Sosial dan Ilmu Politik</li>  
    <li title="Fakultas Pertanian">Fakultas  
Pertanian</li>  
</ul>
```

Contoh 29: style.css

```
[title]{  
    color: blue;  
    font-weight: bold;  
}
```

#### f. Group Selector

Group selector berfungsi untuk menggabungkan beberapa selector.

Contoh 30: index.html

```
<h1>My Lorem Ipsum</h1>
  <h4>Date: 15 November 2021</h4>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing
elit. Non, porro sint nesciunt necessitatibus consequatur
odit ullam aliquam dolor. Veniam asperiores voluptatem dicta
harum quasi labore quo quidem repellendus expedita quos?</p>
```

Contoh 31: style.css

```
h1 .date #text{
  color: green;
}
```

#### g. Dynamic Pseudo Class Selector

Selector ini digunakan untuk mencari struktur HTML berdasarkan interaksi dari user, terdapat 4 jenis yakni:

##### a. :link

Contoh 32: index.html

```
<!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>CSS</title>
  <style>
    :link{
      color: red;
    }
  </style>
</head>
<body>
  <h2>Daftar Prodi di Fakultas Ilmu Komputer</h2>
  <ol type="A">
    <li><a href="ti.html">Teknik Informatika</a></li>
    <li><a href="si.html">Sistem Informasi</a></li>
  </ol>
</body>
</html>
```

##### b. :visited

Contoh 31: index.html

```
<!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>CSS</title>
  <style>
    :link{
      color: red;
    }
  </style>
</head>
<body>
  <h2>Daftar Prodi di Fakultas Ilmu Komputer</h2>
  <ol type="A">
    <li><a href="ti.html">Teknik Informatika</a></li>
    <li><a href="si.html">Sistem Informasi</a></li>
  </ol>
</body>
</html>
```

```

        }
        :visited{
            color: green;
        }
    }
</style>
</head>
<body>
    <h2>Daftar Prodi di Fakultas Ilmu Komputer</h2>
    <ol type="A">
        <li><a href="ti.html">Teknik Informatika</a></li>
        <li><a href="si.html">Sistem Informasi</a></li>
    </ol>
</body>
</html>

```

### c. :hover

Contoh 32: index.html

---

```

<!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>CSS</title>
    <style>
        :link{
            color: red;
        }
        :visited{
            color: green;
        }
        a:hover{
            color: yellow;
        }
    </style>
</head>
<body>
    <h2>Daftar Prodi di Fakultas Ilmu Komputer</h2>
    <ol type="A">
        <li><a href="ti.html">Teknik Informatika</a></li>
        <li><a href="si.html">Sistem Informasi</a></li>
    </ol>
</body>
</html>

```

---

### d. :active

Contoh 33: index.html

```

<!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>CSS</title>
    <style>
        :link{
            color: red;
        }
    </style>

```

```

        :visited{
            color: green;
        }
        a:hover{
            color: yellow;
        }
        :active{
            color: aqua;
        }
    }
</style>
</head>
<body>
    <h2>Daftar Prodi di Fakultas Ilmu Komputer</h2>
    <ol type="A">
        <li><a href="ti.html">Teknik Informatika</a></li>
        <li><a href="si.html">Sistem Informasi</a></li>
    </ol>
</body>
</html>

```

## 1.8 Tipografi dan Box Model

Tipografi merupakan suatu seni untuk menata huruf dengan pengaturan penyebarannya pada ruang yang tersedia, untuk menciptakan kesan tertentu, guna kenyamanan membaca semaksimal mungkin, ada berbagai macam property css yang bisa kita gunakan untuk melakukan itu semua yaitu:

### a. Color

Color berfungsi untuk mengatur warna font

Contoh 34: style.css

---

```

h1{
    /* Color Keyword */
    color: tomato;
}

h2{
    /* RRGGBB */
    color: #00ff00;
}

h3{
    /* RGB */
    color: rgb(255, 0, 255);
}

h4{
    /* RGBA */
    color: rgba(45, 45, 80, 1);
}

h5{
    /* HSL */
    color: hsl(40, 100%, 20%);
}

```

---

### b. Font-size

Font-size berfungsi untuk mengatur ukuran font

Contoh 35: style.css

---

```
/* pixel */
p.text{
    font-size: 18px;
}

/* persen */
h1.title{
    font-size: 80%;
}

/* em */
p.em{
    font-size: 1.2em;
}

/* root em */
p.root{
    font-size: 2rem;
}
```

---

### c. Font-family

Font-family berfungsi untuk mengatur jenis font

Contoh 36: style.css

---

```
body{
    font-family: 'Lucida Sans', 'Lucida Sans Regular',
    'Lucida Grande', 'Lucida Sans Unicode', Geneva, Verdana,
    sans-serif;
}
```

---

### d. Font-weight

Font-weight berfungsi untuk mengatur ketebalan font

Contoh 37: style.css

```
.bold100{
    font-weight: 100;
}
.bold200{
    font-weight: 200;
}
.bold300{
    font-weight: 300;
}
.bold400{
    font-weight: 400;
}
.bold500{
    font-weight: 500;
}
.bold600{
    font-weight: 600;
}
.bold700{
    font-weight: 700;
}
.bold800{
    font-weight: 800;
}
.bold900{
    font-weight: 900;
}
.bold{
```

```
font-weight: bold;
}
```

#### e. Text-decoration

Text decoration berfungsi untuk membuat dekorasi pada teks

Contoh 38: style.css

---

```
.text-satu{
    text-decoration: underline;
}
.text-dua{
    text-decoration: overline;
}
.text-tiga{
    text-decoration: line-through;
}
.text-empat{
    text-decoration: underline overline;
}
.text-lima{
    text-decoration: line-through underline overline;
}
.text-enam{
    text-decoration: none;
}
```

---

#### f. Text-align

Text align berfungsi untuk mengatur perataan teks

Contoh 39: style.css

---

```
.kiri{
    text-align: left;
}
.kanan{
    text-align: right;
}
.tengah{
    text-align: center;
}
.rata{
    text-align: justify;
}
```

---

#### g. Line-height

Line height berfungsi untuk mengatur jarak antar baris

Contoh 40: style.css

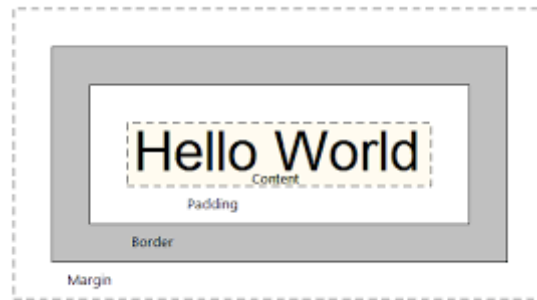
---

```
.text-one{
    line-height: 1.7;
}
.text-two{
    line-height: 1.1em;
}
.text-three{
    line-height: 110%;
}
```

---



Box model adalah sebuah konsep dimana setiap element yang terdapat pada halaman web di proses sebagai kotak (box). Mulai dari header, paragraph, gambar, dan lain sebagainya ini sebenarnya di tampilkan oleh web browser sebagai box. HTML terdiri dari 4 lapisan yakni: konten, padding, border dan margin.



a. Padding

Padding berfungsi agar konten tidak menempel langsung dengan garis tepi (border)

Contoh 41: index.html

```
<!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>CSS</title>
  <style>
    p{
      padding-top: 40px;
      padding-right: 25px;
      padding-bottom: 30px;
      padding-left: 25px;
      border: 1px solid black;
    }
  </style>
</head>
<body>
  <h1>Property Padding</h1>
  <p>Lorem ipsum dolor sit amet consectetur adipisicing
elit. Dolores quasi dolorem nihil quae animi sequi sed
accusamus, corporis optio voluptatibus incidunt
reprehenderit iure ipsa accusantium aperiam reiciendis
magnam. Exercitationem, itaque.</p>
</body>
</html>
```

b. Border

Border merupakan garis batas luar dari sebuah element

### c. Margin

Margin merupakan space atau ruang kosong yang berada di sisi paling luar sebuah element

Contoh 42: index.html

---

```
<!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>CSS</title>
  <style>
    *{
      margin: 0;
      padding: 0;
    }
    div{
      width: 100px;
      height: 100px;
      background-color: tomato;
      text-align: center;
    }
    .box-one{
      margin-top: 20px;
      margin-left: 40px;
      margin-bottom: 20px;
    }

    .box-two{
      margin:0 0 60px 80px
    }

    .box-three{
      margin:0 100px
    }
  </style>
</head>
<body>
  <div class="box-one">One</div>
  <div class="box-two">Two</div>
  <div class="box-three">Three</div>
</body>
</html>
```

---

## 1.9 Tugas

Buatlah website seperti berikut menggunakan HTML dan CSS yang sudah anda pelajari di pertemuan 1

Category

# H1 – Strawberries History Research Paper

By: [Nama Lengkap] | on 5 october 2020 | [NPM]

## Daftar Isi

- [1. Introduction](#)
- [2. Detailed History](#)
- [3. Various Uses Of Strawberries](#)

## Introduction

**S**trawberries are sweet supple red fruits with surfaces that are seed-studded. They come from a low-growing plant belonging to genus *Fragaria* of the rose (*Rosaceae*) family, enclosing white flowers, lobed foliage, and runners. They originate all through the north temperate zones.

They are the most famous and trendiest fruits in the planet (Roger). There are over 600 assortments of strawberries that vary in taste, size, and feel; all of which have similar trait; heart silhouette, red fleshy tissue with yellow seeds shooting the surface, petite, royal, leafy green lids, and stalk that embellish their crowns (WHFoods). Below is picture of the strawberry fruit.

## Detailed history

The word strawberry is coined from 'strewn' berries about the plants and thus the "strewn berry" finally turned into "Strawberry" (Roger). They are not actual fruits or berries but distended ends of the stamen of the plant. The seeds are on the external surface, rather than the internal berry. There are around 200 seeds for each berry. They are low in calories and are non-fat, loaded with vitamin c, folic acid, fiber, vitamin B6 and potassium (Roger).

Over the years they have been utilized as medications for ailments like, stained teeth, sunburns, gout and digestion (Hopley, 9). Strawberries were served up at medieval state occasions as a show of peace, wealth, and excellence. The most well-known communal intake of strawberries is at Wimbledon annually, when strawberries and cream are eaten by very well attired Englishmen in the middle of tennis games (Hopley, 9).

## Various uses of strawberries

Strawberries have numerous uses. One of the most famous uses of strawberries is being used as recipe or an ingredient in preparation of drinks, meals or snacks. Strawberries are used as ingredients in many types of foods like cakes, yoghurts, desserts, salads, juices, meals and chocolates.

— (Lyons, 92)

# PERTEMUAN 2

## HTML & CSS

### 2.1 Gambar

Gambar merupakan salah satu komponen penting yang ada di halaman web. Hampir setiap website pasti memiliki gambar. Untuk memasukkan gambar ke website dengan HTML menggunakan tag `<img>`. Ada beberapa atribut yang bisa digunakan pada tag `<img>` diantaranya sebagai berikut:

#### a. Source

Atribut source digunakan untuk memasukkan alamat gambar yang ingin ditampilkan

Contoh 1: image.html

---

```
<!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>Image</title>
</head>
<body>
  
</body>
</html>
```

---

#### b. Width

Atribut width digunakan untuk mengatur ukuran lebar gambar

Contoh 2: image.html

---

```
<!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>Image</title>
</head>
<body>
  
</body>
</html>
```

---

#### c. Height

Atribut height digunakan untuk mengatur ukuran tinggi gambar

Contoh 3: image.html

---

```
<!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>Image</title>
```

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

d. Alt

Atribut alt digunakan untuk memberikan keterangan terhadap gambar, ketika gambar gagal ditampilkan

Contoh 4: image.html

---

```
<!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>Image</title>
</head>
<body>
  
</body>
</html>
```

---

e. Title

Atribut title digunakan untuk memberikan keterangan terhadap gambar, namun berbeda dengan atribut alt, atribut ini akan muncul dalam bentuk tooltip

Contoh 5: image.html

---

```
<!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>Image</title>
</head>
<body>
  
</body>
</html>
```

---

## 2.2 Tabel

Untuk membuat tabel di HTML, kita membutuhkan 3 element diantaranya yakni, <table>, <tr>, dan <td>. Ada beberapa atribut yang bisa digunakan pada tabel diantaranya sebagai berikut:

---

```
<!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>Tabel</title>
</head>
<body>
  <table>
    <tr>
      <td>Baris ke-1, Kolom ke-1</td>
      <td>Baris ke-1, Kolom ke-2</td>
    </tr>
    <tr>
      <td>Baris ke-2, Kolom ke-1</td>
      <td>Baris ke-2, Kolom ke-2</td>
    </tr>
    <tr>
      <td>Baris ke-3, Kolom ke-1</td>
      <td>Baris ke-3, Kolom ke-2</td>
    </tr>
  </table>
</body>
</html>
```

---

#### a. Border

Atribut border berfungsi untuk menampilkan garis pada tabel

Contoh 7: tabel.html

---

```
<table border="1">
  <tr>
    <td>Baris ke-1, Kolom ke-1</td>
    <td>Baris ke-1, Kolom ke-2</td>
  </tr>
  <tr>
    <td>Baris ke-2, Kolom ke-1</td>
    <td>Baris ke-2, Kolom ke-2</td>
  </tr>
  <tr>
    <td>Baris ke-3, Kolom ke-1</td>
    <td>Baris ke-3, Kolom ke-2</td>
  </tr>
</table>
```

---

#### b. Width dan Height

Atribut ini berfungsi untuk mengatur ukuran lebar dan tinggi tabel

Contoh 8: tabel.html

```
<table border="1" width="400" height="200">
  <tr>
    <td>Baris ke-1, Kolom ke-1</td>
    <td>Baris ke-1, Kolom ke-2</td>
  </tr>
  <tr>
    <td>Lorem ipsum, dolor sit amet consectetur
adipisicing elit. Aut nobis fugiat inventore veniam
laudantium maxime minima nam, nihil sit voluptate asperiores

```

```

adipisci qui sunt, esse alias. Aliquid assumenda alias
reiciendis.</td>
    <td>Baris ke-2, Kolom ke-2</td>
</tr>
<tr>
    <td>Baris ke-3, Kolom ke-1</td>
    <td>Baris ke-3, Kolom ke-2</td>
</tr>
</table>

```

#### c. Rowspan dan Colspan

Atribut rowspan berfungsi untuk menggabungkan tabel dalam bentuk baris sedangkan atribut colspan berfungsi untuk menggabungkan tabel dalam bentuk kolom

Contoh 9: tabel.html

---

```

<table border="1">
    <tr>
        <td colspan="2">Nama Lengkap</td>
        <td rowspan="2">Umur</td>
    </tr>
    <tr>
        <td>Nama Depan</td>
        <td>Nama Belakang</td>
    </tr>
    <tr>
        <td>Jill</td>
        <td>Smith</td>
        <td>43</td>
    </tr>
    <tr>
        <td>Eve</td>
        <td>Jackson</td>
        <td>57</td>
    </tr>
</table>

```

---

#### d. Th element

Th merupakan singkatan dari *table header* yang berfungsi untuk membuat judul kolom

Contoh 10: tabel.html

---

```

<table border="1">
    <tr>
        <th>Judul 1</th>
        <th>Judul 2</th>
    </tr>
    <tr>
        <td>Baris ke-1, Kolom ke-1</td>
        <td>Baris ke-1, Kolom ke-2</td>
    </tr>
    <tr>
        <td>Baris ke-2, Kolom ke-1</td>
        <td>Baris ke-2, Kolom ke-2</td>
    </tr>
    <tr>
        <td>Baris ke-3, Kolom ke-1</td>
        <td>Baris ke-3, Kolom ke-2</td>
    </tr>
</table>

```

---

e. Thead, Tbody, dan Tfoot

Ini merupakan tag html yang berfungsi untuk membuat struktur tabel secara lengkap.

Contoh 11: tabel.html

---

```
<table border="1">
  <thead>
    <tr>
      <th>No</th>
      <th>Nama Lengkap</th>
      <th>Kelas</th>
      <th>Nilai</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>Rina Kumalasari</td>
      <td>1A</td>
      <td>90</td>
    </tr>
    <tr>
      <td>2</td>
      <td>Ujang Abdul</td>
      <td>1A</td>
      <td>95</td>
    </tr>
  </tbody>
  <tfoot>
    <tr>
      <td colspan="3">Total</td>
      <td>185</td>
    </tr>
  </tfoot>
</table>
```

---

## 2.3 Form

Form merupakan komponen yang sangat penting di dalam sebuah website. Hampir setiap web pasti memiliki form. Form biasanya digunakan untuk login, register, komentar, dan lain sebagainya. Ada banyak sekali tag maupun atribut yang ada pada form.

a. Form

Tag ini berfungsi untuk membuat form, dalam tag ini harus disertakan atribut action dan method. Atribut action berfungsi untuk menulis alamat file yang digunakan untuk memproses isi form. Sedangkan atribut method digunakan untuk memberitahu web browser metode apa yang digunakan pada form tersebut, terdapat 2 nilai yang bisa digunakan yakni GET dan POST.

Contoh 12: form.html

```
<!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>Form</title>
    </head>
    <body>
        <form action="#" method="GET">
            //masukkan input di dalam sini
        </form>
    </body>
</html>

```

b. Input type text

adalah inputan biasa yang menerima input berupa text, contohnya digunakan untuk inputan *nama*, *username*, dan inputan yang berupa text pendek.

Contoh 13: form.html

---

```

<form action="#" method="GET">
    <label for="username">Username</label>
    <input type="text" name="username" id="username">
</form>

```

---

c. Input type submit dan reset

Submit berfungsi untuk mengirimkan form ke server dan berbentuk tombol sedangkan reset berfungsi untuk menghapus semua isian user pada inputan.

Contoh 14: form.html

---

```

<form action="#" method="GET">
    <label for="username">Username</label>
    <input type="text" name="username" id="username">
    <br>
    <input type="submit" value="Tambah">
    <input type="reset" value="Reset">
</form>

```

---

d. Input type email

Tampilannya sama dengan type text, hanya saja type ini berfungsi untuk inputan dalam bentuk alamat email.

Contoh 15: form.html

---

```

<form action="#" method="GET">
    <label for="username">Username</label>
    <input type="text" name="username" id="username">
    <br>
    <label for="email">Email</label>
    <input type="email" name="email" id="email">
    <br>
    <br>
    <input type="submit" value="Tambah">
    <input type="reset" value="Reset">
</form>

```

---

e. Input type password

Tampilannya sama dengan type text, namun teks yang diinput tidak akan terlihat, akan berupa bintang atau bulatan. Biasanya hanya digunakan untuk inputan yang sensitif seperti *password*.

---

```

<form action="#" method="GET">
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <br>
  <label for="email">Email</label>
  <input type="email" name="email" id="email">
  <br>
  <label for="password">Password</label>
  <input type="password" name="password"
id="password">
  <br>
  <br>
  <input type="submit" value="Tambah">
  <input type="reset" value="Reset">
</form>

```

---

## f. Input type checkbox

Tampilan nya berbentuk kotak persegi dimana user bisa ceklis kotak isian tersebut untuk memilih satu atau lebih.

---

```

<form action="#" method="GET">
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <br>
  <label for="email">Email</label>
  <input type="email" name="email" id="email">
  <br>
  <label for="password">Password</label>
  <input type="password" name="password"
id="password">
  <br>
  <label for="hobi">Hobi</label>
  <input type="checkbox" name="hobi" id="hobi"
value="Membaca"> Membaca
  <input type="checkbox" name="hobi" id="hobi"
value="Menulis"> Menulis
  <input type="checkbox" name="hobi" id="hobi"
value="Memancing"> Memancing
  <br>
  <br>
  <input type="submit" value="Tambah">
  <input type="reset" value="Reset">
</form>

```

---

## g. Input type radio

Hampir sama seperti checkbox, namun ketika kita menggunakan radio pengguna hanya diizinkan memilih salah satu pilihan saja, biasanya radio digunakan seperti memilih jenis kelamin.

---

```

<form action="#" method="GET">
  <label for="username">Username</label>
  <input type="text" name="username" id="username">
  <br>
  <label for="email">Email</label>
  <input type="email" name="email" id="email">
  <br>
  <label for="password">Password</label>

```

---

```

        <input type="password" name="password"
id="password">
        <br>
        <label for="hobi">Hobi</label>
        <input type="checkbox" name="hobi" id="hobi"
value="Membaca"> Membaca
        <input type="checkbox" name="hobi" id="hobi"
value="Menulis"> Menulis
        <input type="checkbox" name="hobi" id="hobi"
value="Memancing"> Memancing
        <br>
        <label for="jk">Jenis Kelamin</label>
        <input type="radio" name="jk" id="jk" value="Laki-
Laki"> Laki-Laki
        <input type="radio" name="jk" id="jk"
value="Perempuan"> Perempuan
        <br>
        <br>
        <input type="submit" value="Tambah">
        <input type="reset" value="Reset">
    </form>

```

#### h. Textarea

Tag textarea pada dasarnya sama dengan input type text, namun lebih besar dan dapat berisi banyak baris. Panjang dan banyak baris untuk text area di atur melalui atribut rows dan cols, atau melalui CSS.

Contoh 19: form.html

---

```

<label for="alamat">Alamat</label>
<textarea name="alamat" id="alamat" cols="30"
rows="10"></textarea>
<br>

```

---

#### i. Select option

Tag select digunakan untuk inputan yang telah tersedia nilainya, dan user hanya dapat memilih dari nilai yang ada. Tag select digunakan bersama-sama dengan tag option untuk membuat box pilihan.

Contoh 20: form.html

---

```

<label for="jurusan">Jurusan</label>
<select name="jurusan" id="jurusan">
    <option value="Teknik Informatika">Teknik
Informatika</option>
    <option value="Sistem Informasi">Sistem
Informasi</option>
</select>
<br>

```

---

#### j. Button

Hampir sama fungsinya seperti input type submit. Button sangat cocok digunakan apabila dihubungkan dengan javascript

Contoh 21: form.html

```

<button>Kirim Data</button>

```

## 2.4 Background dan Positioning

Properti background digunakan untuk mengatur background sedangkan positioning untuk mengatur cara penempatan element di dalam halaman web.

a. Background-color

Properti ini berfungsi untuk mengubah warna latar belakang

Contoh 22: index.html

---

```
<!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>CSS</title>
  <style>
    div{
      width: 400px;
      margin: 10px;
      text-align: center;
      line-height: 80px;
      height: 80px;
      font-size: 20px;
      float: left;
    }

    .box-one{
      background-color: tomato;
    }

    .box-two{
      background-color: #3498db;
    }

    .box-three{
      background-color:rgb(153, 102, 153);
    }
  </style>
</head>
<body>
  <div class="box-one">
    Box 1
  </div>
  <div class="box-two">
    Box 2
  </div>
  <div class="box-three">
    Box 3
  </div>
</body>
</html>
```

---

b. Background-image

Properti ini berfungsi untuk mengubah latar belakang dengan menggunakan gambar.

Contoh 23: index.html

---

```
<!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>CSS</title>
```

---

```

<style>
    body{
        background-image: url('strawberry.png');
    }
</style>
</head>
<body>
    <p>Saya sedang belajar menggunakan background image</p>
</body>
</html>

```

---

#### c. Background-repeat

Properti ini berfungsi untuk mengatur latar belakang gambar yang kita gunakan apakah berulang atau tidak.

Contoh 24: index.html

```

<!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>CSS</title>
    <style>
        body{
            background-image: url('strawberry.png');
            background-repeat: no-repeat;
        }
    </style>
</head>
<body>
    <p>Saya sedang belajar menggunakan background repeat</p>
</body>
</html>

```

---

#### d. Background-position

Properti ini berfungsi untuk mengatur posisi dari latar belakang gambar yang kita gunakan, terdapat 5 kata kunci yang bisa digunakan yakni top, bottom, right, left, dan center.

Contoh 25: index.html

```

<!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>CSS</title>
    <style>
        body{
            background-image: url('strawberry.png');
            background-repeat: no-repeat;
            background-position:center;
        }
    </style>
</head>
<body>

```

```
<p>Saya sedang belajar menggunakan background
position</p>
</body>
</html>
```

---

e. Background-size

Properti ini berfungsi untuk mengatur ukuran dari latar belakang gambar yang kita gunakan.

Contoh 26: index.html

```
<!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>CSS</title>
  <style>
    body{
      background-image: url('1.jpg');
      background-repeat: no-repeat;
      background-position:center;
      background-size: cover;
    }
  </style>
</head>
<body>
  <p>Saya sedang belajar menggunakan background size</p>
</body>
</html>
```

---

## 2.5 Tugas

1. Buatlah form sederhana menggunakan HTML & CSS seperti berikut:

### Pendaftaran Mahasiswa Baru

NPM (Nomor Pokok Mahasiswa)

Nama Lengkap

Alamat

Jenis Kelamin

☐ Laki-Laki

☒ Perempuan

Program Studi

Teknik Informatika

Tambah

2. Buatlah tabel seperti berikut menggunakan HTML & CSS

| Tabel Nilai Mahasiswa |                  |       |   |   |   |   |   |   |   |   |    |       |    |    |    |    |    |    |    |    |    |     |     |
|-----------------------|------------------|-------|---|---|---|---|---|---|---|---|----|-------|----|----|----|----|----|----|----|----|----|-----|-----|
| NPM                   | Nama Lengkap     | Absen |   |   |   |   |   |   |   |   |    | Tugas |    |    |    |    |    |    |    |    |    | UTS | UAS |
|                       |                  | 1     | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 1     | 2  | 3  | 4  | 5  | 6  | 7  | 8  | 9  | 10 |     |     |
| 192061001             | Rina Kumala Sari | H     | H | H | H | H | H | H | H | H | H  | 90    | 85 | 70 | 95 | 88 | 76 | 95 | 97 | 80 | 88 | 85  | 90  |
| 192061002             | Kevin Mulyana    | H     | A | H | H | H | A | A | H | H | H  | 80    | 95 | 90 | 85 | 98 | 96 | 75 | 87 | 70 | 78 | 95  | 90  |
| 192061003             | Lastri Handayani | H     | H | H | H | H | H | H | H | H | H  | 90    | 85 | 70 | 95 | 88 | 76 | 95 | 97 | 80 | 88 | 85  | 90  |
| 192061004             | Doni Sanjaya     | H     | A | H | H | H | A | A | H | H | H  | 80    | 95 | 90 | 85 | 98 | 96 | 75 | 87 | 70 | 78 | 95  | 90  |
| 192061005             | Santi Amelia     | H     | H | H | H | H | H | H | H | H | H  | 90    | 85 | 70 | 95 | 88 | 76 | 95 | 97 | 80 | 88 | 85  | 90  |

3. Buat galeri dari beberapa gambar dengan menggunakan HTML & CSS seperti berikut



# PERTEMUAN 3

## JAVASCRIPT DASAR

### 3.1 Pengertian Javascript

Javascript merupakan bahasa pemrograman yang populer digunakan untuk membuat web dengan konten yang dinamis. Konten dinamis artinya konten yang dapat bergerak atau berubah di depan layar tanpa perlu mereload halaman. Javascript sendiri biasanya dikolaborasikan dengan HTML dan CSS, yang dimana HTML digunakan untuk membuat struktur website dan CSS untuk merancang style halaman website.

### 3.2 Cara Menginput Kode Javascript

Untuk bisa menginput kode Javascript ke dalam HTML terdapat tiga cara yang bisa digunakan yaitu:

#### a. Inline Javascript

Menuliskan kode javascript langsung ke dalam tag html

Contoh 1: index.html

---

```
<!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>Javascript</title>
</head>
<body>
  <button onclick="alert('Saya sedang belajar bahasa
pemrograman Javascript')">Klik Saya</button>
</body>
</html>
```

---

#### b. Internal Javascript

Menuliskan kode javascript ke dalam tag <script>

Contoh 2: index.html

---

```
<!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>Javascript</title>
</head>
<body>
  <script>
    alert("Halo ini merupakan alert yang saya buat
dengan javascript");
  </script>
</body>
</html>
```

---



c. External Javascript

Menulis kode javascript dengan cara membuat file khusus javascript kemudian di panggil oleh halaman HTML yang membutuhkan kode tersebut

Contoh 3: app.js

```
alert('Ini merupakan contoh alert javascript yang dibuat di external javascript');
```

Contoh 4: index.html

---

```
<!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>Javascript</title>
</head>
<body>

  <script src="1.js"></script>
</body>
</html>
```

---

### 3.3 Variabel, Konstanta, Tipe Data dan Operator

Dalam setiap bahasa pemrograman apapun sudah pasti terdapat variabel, konstanta, tipe data dan operator.

a. Variabel

Variabel adalah sebuah nama yang mewakili sebuah nilai. Variabel bisa diisi dengan berbagai macam nilai seperti string (teks), number (angka), objek, array, dan sebagainya. Kita bisa ibaratkan, variabel itu seperti wadah untuk menyimpan sesuatu.

#### Deklarasi variabel dengan perintah “let”

Dalam bahasa pemrograman javascript untuk membuat variabel menggunakan kata kunci “let”

---

```
let name;
```

---

#### Assignment nilai variabel

Dalam bahasa pemrograman javascript kita bisa mengisi variabel dengan nilai apapun seperti teks, angka, objek dan lain sebagainya. Sama seperti bahasa pemrograman pada umumnya untuk mengisi nilai pada variabel menggunakan operator assignment (=).

---

```
let name;
name = "Budi Setiawan";
console.log(name);
```

---

Kita juga bisa langsung untuk mendeklarasikan variabel dan assignment nilai nya dengan cara:

---

```
let name = "Budi Setiawan";  
console.log(name);
```

---

b. Konstanta

Konstanta artinya tetap, konstanta sama seperti variabel hanya saja ketika konstanta dibuat dan diberi nilai awal, maka isi konstanta tersebut tidak bisa diisi lagi oleh nilai yang lain. Untuk membuat konstanta menggunakan kata kunci **“const”**.

---

```
const phi = 3.14;  
const greet = "Good Morning";  
  
console.log("Nilai phi adalah "+ phi);  
console.log(greet+" Budi Setiawan");
```

---

c. Tipe Data

Dalam bahasa pemrograman Javascript tipe data terbagi ke dalam 2 kelompok yaitu tipe data primitif dan object

1. Tipe data primitive

- Number

Sesuai dengan namanya tipe data ini merupakan tipe data yang berisi angka. Tidak seperti bahasa pemrograman lain yang membedakan antara bilangan bulat dan decimal. Dalam bahasa pemrograman javascript bilangan bulat dan decimal tidak dibedakan.

---

```
let ipk = 3.90;  
let price = 15000;  
let celcius = -15;  
  
console.log(ipk);  
console.log(price);  
console.log(celcius);
```

---

- String

Tipe data ini berfungsi untuk data yang berisi teks ataupun huruf.

---

```
let activity;  
  
activity = 'Hello FASILKOM';  
console.log(activity);  
  
activity = "I'm learning javascript";  
console.log(activity);  
  
activity = "2021";  
console.log(activity);
```

---

- Boolean

Tipe data ini hanya mempunyai dua nilai yakni true atau false.

---

```
let isSuccess = true;  
let isFailed = false;  
  
console.log(isSuccess);  
console.log(isFailed);
```

---

## 2. Tipe data object

### - Array

Array dalam bahasa Indonesia yang berarti larik merupakan tipe data yang berisi kumpulan tipe data yang lain. Contohnya seperti berikut:

---

```
let faculties = [
  "Fakultas Ilmu Komputer",
  "Fakultas Hukum",
  "Fakultas Teknik",
  "Fakultas Ilmu Sosial dan Ilmu Politik",
  "Fakultas Agama Islam"
];

console.log(faculties[0]);
console.log(faculties[1]);
console.log(faculties[2]);
console.log(faculties[3]);
console.log(faculties[4]);
```

---

### Cara Mengubah Elemen Array

Untuk mengubah elemen array, kita cukup mengakses elemen array tersebut dengan menulis nama array kemudian diikuti oleh nomor indexnya (index dimulai dari 0)

---

```
let faculties = [
  "Fakultas Ilmu Komputer",
  "Fakultas Hukum",
  "Fakultas Teknik",
  "Fakultas Ilmu Sosial dan Ilmu Politik",
  "Fakultas Agama Islam"
];

faculties[1] = "Fakultas Ekonomi dan Bisnis";
console.log(faculties);

faculties[4] = 50;
console.log(faculties);
```

### Array 2 Dimensi

```
let coordinate = [
  [0,3],
  [6,2],
  [9,8]
];

console.log(coordinate[0][0]);
console.log(coordinate[0][1]);

console.log(coordinate[1][0]);
console.log(coordinate[1][1]);

console.log(coordinate[2][0]);
console.log(coordinate[2][1]);
```

---

## d. Operator

Operator digunakan untuk memproses data yang sudah diinput ke dalam Javascript. Terdapat 8 operator yang ada di Javascript:

### 1. Operator Aritmatika

Operator ini berfungsi untuk melakukan perhitungan seperti penjumlahan, pengurangan, perkalian, pembagian dan juga modulus.

---

```
let results = 9+10;
console.log(results);

results = 15-10;
console.log(results);

results = 9*5;
console.log(results);

results = 18/2;
console.log(results);

results = 28 % 3;
console.log(results);

results = 10 + 15 * 2 / 2;
console.log(results);
```

---

### 2. Operator Perbandingan

Operator ini berfungsi untuk membandingkan dua buah operand atau lebih. Operator ini mengembalikan nilai true atau false

---

```
let results = 18 < 2;
console.log(results);

results = 10 > 3;
console.log(results);

results = 5 >= 10;
console.log(results);

results = 16 <= 4;
console.log(results);

results = 19 == 10;
console.log(results);

results = 20 === 20;
console.log(results);
```

---

### 3. Operator Logika

Operator ini berfungsi untuk membandingkan 2 buah kondisi logika dan nilai yang akan dibandingkan harus bertipe Boolean.

---

```
let bool;

bool = true && false;
console.log(bool);

bool = false || true;
console.log(bool);
```

```
bool = !false;
console.log(bool);

bool = true || true || false;
console.log(bool);

bool = true && false || true;
console.log(bool);
```

---

#### 4. Operator String

Digunakan untuk melakukan penyambungan string atau biasa disebut dengan *string concatenation*. Operator ini menggunakan simbol (+)

---

```
let firstName, lastName, fullName;

firstName = "Budi";
lastName = "Setiawan";
fullName = firstName + " " + lastName;

console.log(fullName);

console.log("Halo, my name is "+fullName);

let students = ["Reni Mulyasari", "Kevin Alam", "Putri Nabila", "Rudi Widodo"];
console.log(students[0]+" dan "+students[2]+" merupakan teman dekat");

let exam = students[3] + " mendapatkan nilai " + 9*9;
console.log(exam);
```

---

#### 5. Operator Spread

Operator spread berfungsi untuk menggabungkan array. Operator ini menggunakan simbol (...)

---

```
let male = ["Ujang Sanjaya", "Kevin Alam", "Dedi Wijaya"];
console.log(male);

let female = ["Lilis Puspitasari", "Ririn Noviyanti", "Putri Wijaya"];
console.log(female);

let students = [...male, ...female];
console.log(students);
```

---

### 3.4 Percabangan dan Perulangan

#### a. If

Apabila sebuah kondisi terpenuhi (true) maka jalankan statement yang berada di dalam blok if.

---

```
let gender;
gender = 'L';

if(gender === 'L'){
  console.log("I'm a man");
}
```

---

#### b. If Else

Apabila sebuah kondisi terpenuhi (true) maka jalankan statement yang berada di dalam blok if. Sedangkan apabila kondisi tidak terpenuhi (false) maka jalankan statement yang berada di dalam blok else

---

```
let gender;
gender = 'P';

if(gender === 'L'){
  console.log("I'm a man");
}else{
  console.log("I'm a woman");
}
```

---

c. If Else If

Apabila ingin melakukan pengecekan kondisi lebih dari satu kali maka gunakan kondisi if else if

---

```
let username, password;

username = "Admin";
password = "qwerty";

const isSuccess = true;
const isFailed = false;

if(username === "Admin" && password === "qwerty"){
  console.log(isSuccess);
}else if(username === "Admin" && password !== "qwerty"){
  console.log(isFailed);
}else if(username !== "Admin" && password === "qwerty"){
  console.log(isFailed);
}else{
  console.log(isSuccess);
}
```

---

d. Nested If Else

Nested if else atau biasa disebut dengan if bersarang. Disini kita bisa membuat kondisi di dalam kondisi.

---

```
let username, password;

username = "Admin";
password = "qwerty";

const isSuccess = true;
const isFailed = false;

if(username === "Admin"){
  if(password === "qwerty"){
    console.log(isSuccess);
  }else{
    console.log(isFailed);
  }
}else if(password === "qwerty"){
  console.log(isFailed);
}else{
  console.log(isFailed);
}
```

---

e. Switch

Switch merupakan percabangan yang akan memeriksa satu variabel, lalu menjalankan perintah sesuai dengan kondisi.

---

```
let grade = 'A';

switch(grade){
  case 'A':
    console.log("Very good");
    break;
  case 'B':
    console.log("Good");
    break;
  case 'C':
    console.log("Enough");
    break;
  case 'D':
    console.log("Very Enough");
    break;
  default:
    console.log("No Grade");
    break;
}
```

---

f. Operator Conditional

Operator conditional mirip seperti if else. Berikut merupakan format dari operator conditional:

Results = (condition) ? true : false

---

```
let age = 18;

let result = (age >= 18) ? "Eligible for voting" : "Not
eligible for voting";
console.log(result);
```

---

g. For

---

```
for (let i = 0; i < 10; i++) {
  console.log("Halo Javascript");
}
```

---

h. While

---

```
let i = 50;
while(i > 0){
  console.log(i + " * 5 = " + i*5);
  i = i -5;
}
```

---

i. Do While

---

```
let i = 1;
do{
```

---

```
        console.log(i+" kuadrat = "+i*i);
        i = i + 1;
    }while(i <= 10);
```

---

j. For Of

```
let faculties = [
    "Fakultas Ilmu Komputer",
    "Fakultas Hukum",
    "Fakultas Pertanian",
    "Fakultas Ilmu Sosial dan Ilmu Politik",
    "Fakultas Teknik"
];

for(let faculty of faculties){
    console.log(faculty);
}
```

---

k. Break dan Continue

Perintah break digunakan untuk menghentikan sebuah perulangan sedangkan untuk perintah continue berfungsi untuk melewati satu perulangan.

```
for(let i = 1; i<=10; i++){
    if(i === 6){
        break;
    }
    console.log(i);
}

for(let i = 1; i<=10; i++){
    if(i === 6){
        continue;
    }
    console.log(i);
}
```

---

l. Nested Loop

Nested loop merupakan perulangan di dalam perulangan

```
let weeks = 3;
let days = 7;
for (let i = 1; i <= weeks; ++i) {
    console.log("Week: " + i);
    for (let j = 1; j <= days; ++j) {
        console.log("    Day: " + j);
    }
}
```

---

### 3.5 Function

Fungsi (atau *Function*) di bahasa pemrograman adalah kode program yang dirancang untuk menyelesaikan sebuah tugas tertentu, dan merupakan bagian dari program utama. Kita dapat membuat fungsi sendiri, atau menggunakan fungsi yang dibuat oleh programmer lain.

#### Cara Membuat Function



Untuk membuat function kita menggunakan keyword function dan diikuti oleh nama function seperti contoh berikut:

---

```
function greet() {  
  console.log("Good Morning");  
  console.log("Good Afternoon");  
  console.log("Good Night");  
}
```

---

### Cara Memanggil Function

Untuk memanggil function yang kita buat, cukup panggil nama function nya

---

```
function greet() {  
  console.log("Good Morning");  
  console.log("Good Afternoon");  
  console.log("Good Night");  
}
```

```
greet();
```

---

### Return Function

Keyword return berfungsi untuk mengembalikan nilai yang ada di dalam function dan bisa ditampilkan terlebih dahulu ke dalam variabel, seperti contoh berikut:

---

```
function greet() {  
  return "Good Morning";  
}
```

```
let regard = greet();  
console.log(regard);
```

```
console.log(greet());
```

---

### Return Function Array

---

```
function hobbies() {  
  const listHobbies =  
    ["Football", "Swimming", "Reading", "Writing", "Coding"];  
  return listHobbies;  
}
```

```
let hobby = hobbies();
```

```
for (const iterator of hobby) {  
  console.log(iterator);  
}
```

---

### Function Parameter

Sebuah fungsi juga dapat dideklarasikan dengan parameter. Parameter adalah nilai yang dilewatkan saat mendeklarasikan suatu fungsi.

---

```
function sum(number1, number2) {  
  let results = number1 + number2;  
  return results;  
}
```

```
console.log(sum(5, 5));
```

---

### Spread Operator

Sebelumnya sudah dipelajari di materi array untuk menggabungkan array, di dalam function pun terdapat spread operator yang berfungsi untuk menggabungkan seperti berikut:

---

```
function average(...index) {  
  let sum = index.length;  
  let result = 0;  
  for(const i of index) {  
    result += i;  
  }  
  return result/sum;  
}  
  
let x = average(4,8,3,7);  
console.log(x);
```

---

### Arrow Function

Ini merupakan alternative untuk menuliskan function, fitur ini tersedia dalam ECMAScript 6. Dalam menggunakan arrow function kita akan menggunakan tanda panah (arrow) => untuk membuat function. Berikut contohnya

#### Arrow Function Tanpa Parameter

---

```
const greet = () => {  
  console.log('Good Morning');  
}  
  
greet();
```

---

#### Arrow Function Dengan Parameter

---

```
const sum = (x, z) => {  
  let result = x + z;  
  return result;  
}  
  
let result1 = sum(5,10);  
console.log(result1);
```

---

### 3.6 Tugas

1. Buatlah program javascript untuk mencetak deret Fibonacci!
2. Buatlah kalkulator dengan menggunakan javascript serta buat lah menggunakan arrow function dan spread operator yang terdiri dari (+-/\*%)

# PERTEMUAN 4

## BOOTSTRAP

### 4.1 Pengertian Bootstrap

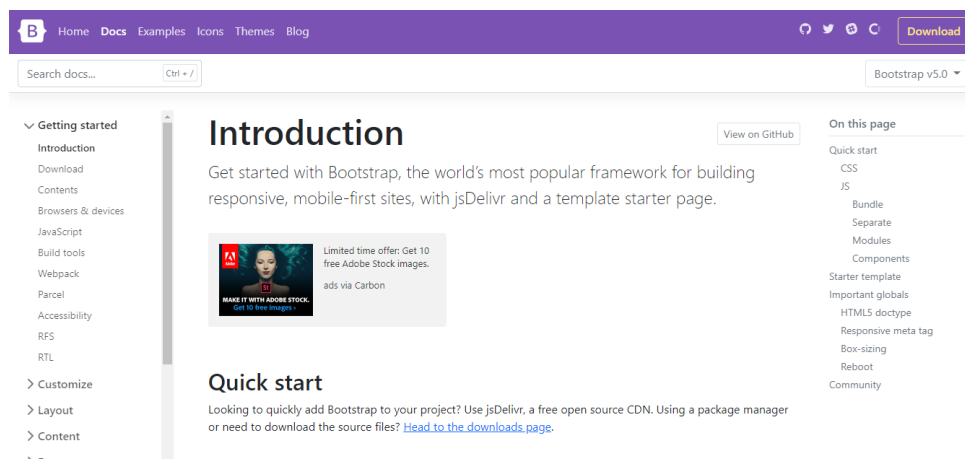
Bootstrap merupakan *framework* yang dirancang menggunakan HTML, CSS, dan Javascript yang dapat memudahkan kita untuk mendesain web. Dengan adanya Bootstrap kita dapat membuat desain web yang menarik dan juga responsive tanpa harus memahami kode-kode HTML, CSS, dan Javascript yang sangat rumit. Kita hanya perlu paham tentang dasar-dasarnya aja.

### 4.2 Cara Menginstall Bootstrap

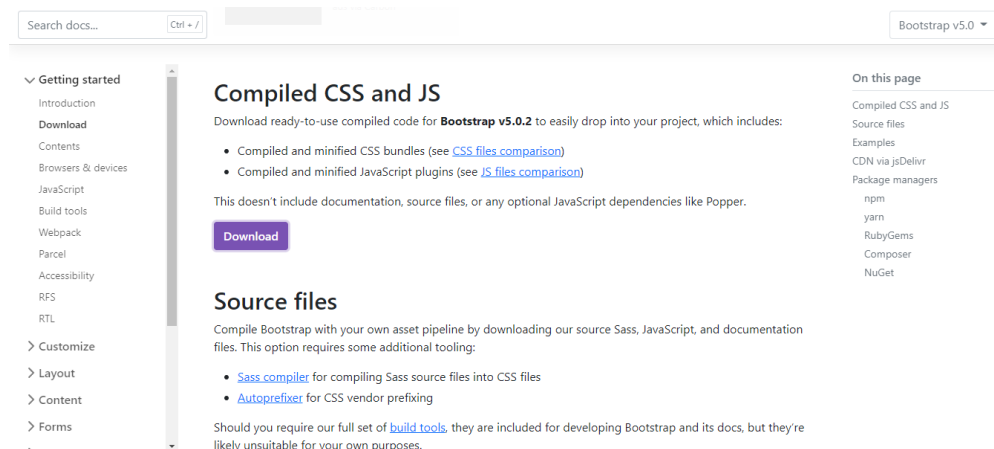
Terdapat dua cara untuk menggunakan bootstrap kita bisa menggunakan file bootstrap offline atau menggunakan file bootstrap cdn (*content delivery network*).

#### Cara download file bootstrap

1. Buka situs resmi bootstrap di halaman <https://getbootstrap.com/docs/5.0/getting-started/introduction/>



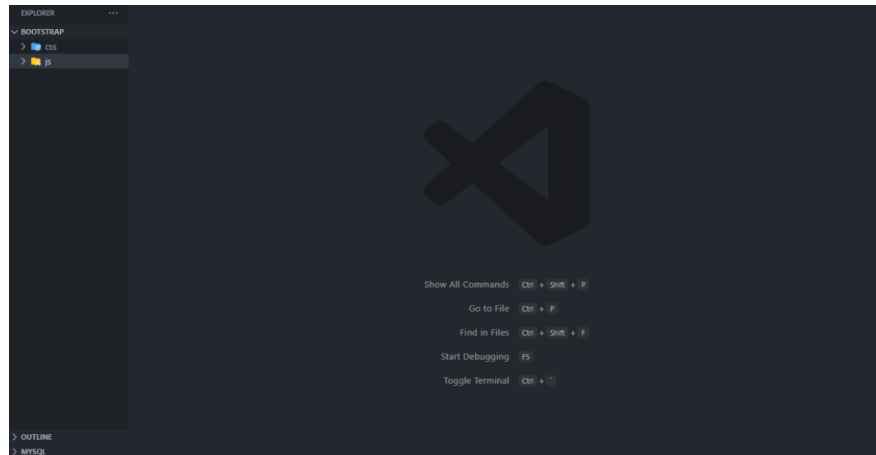
2. Klik menu download yang berada di pojok kanan atas



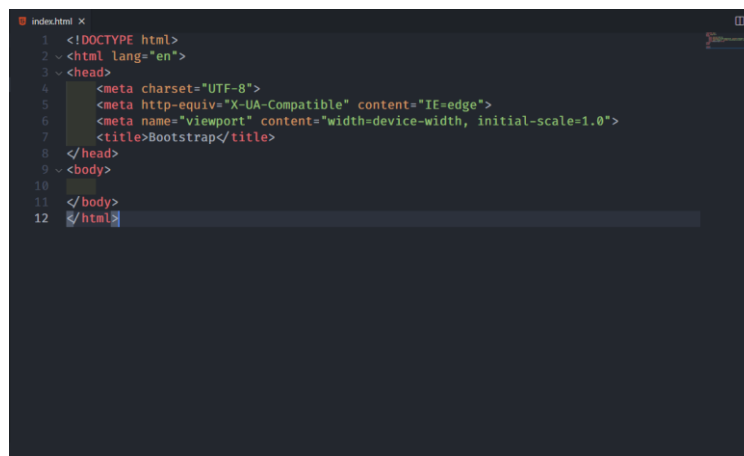
3. Scroll kebawah dan cara **Compiled CSS and JS**, jika memilih pilihan yang pertama maka akan mendownload file css nya saja, jika memilih pilihan yang kedua maka

akan mendownload file js nya saja. Jika ingin mendownload semuanya maka klik tombol **download**

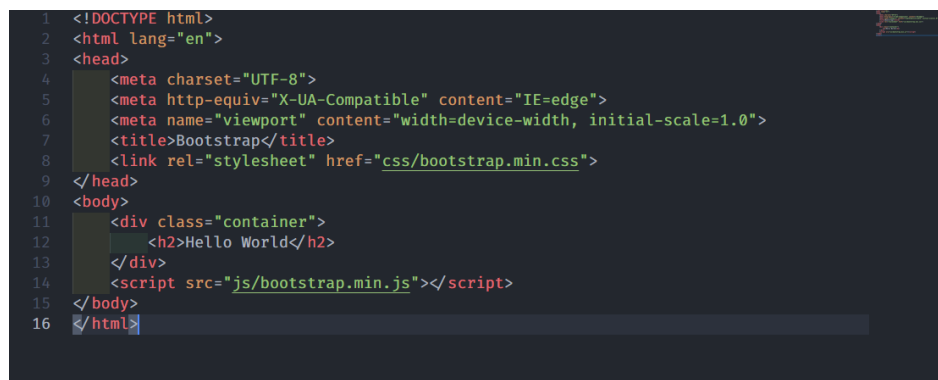
4. Buat folder baru misalnya (bootstrap) kemudian masukkan semua folder css dan js ke dalam folder bootstrap



5. Untuk cara menggunakannya buat file baru dan beri nama index.html, kemudian langsung masukkan struktur dasar html.



6. Kemudian tinggal kita panggil file css bootstrap dan file js bootstrap ke dalam file index.html



7. Selesai, silahkan jalankan file tersebut dan lihat perbedaan nya.

### 4.3 Layout

Dengan menggunakan bootstrap kita dapat dengan mudah untuk membuat atau mengatur layout web

#### Container dan Container Fluid

Container merupakan struktur dasar dari bootstrap yang berisi, tempat untuk menyelaraskan konten dalam perangkat atau resolusi tertentu. Di dalam container inilah seluruh komponen bootstrap berada.

---

```
<!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</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <style>
    div{
      background-color: rgba(86, 61, 124, 0.15);
      border: 1px solid rgba(86, 61, 124, 0.2);
      padding: 0.4em;
      margin-top: 0.5em;
    }
  </style>
</head>
<body>
  <div class="container">
    <h2>Hello Bootstrap</h2>
  </div>
  <div class="container-fluid">
    <h2>Hello Bootstrap</h2>
  </div>
  <script src="js/bootstrap.min.js"></script>
</body>
</html>
```

---

#### Row and Column

Sebelumnya telah membuat container, container merupakan struktur dasar dari Bootstrap. Di dalam container kita bisa menggunakan row(baris) dan col(kolom) untuk membuat layout dalam bentuk grid. Untuk membuat baris kita menggunakan class **row** sedangkan untuk membuat kolom kita menggunakan class **col**.

---

```
<!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</title>
  <link rel="stylesheet" href="css/bootstrap.min.css">
  <style>
    .col{
      background-color: rgba(86, 61, 124, 0.15);
```

---

```

        border: 1px solid rgba(86, 61, 124, 0.2);
        padding: 0.4em;
        margin-top: 0.5em;
    }
</style>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col">Kolom 1</div>
            <div class="col">Kolom 2</div>
            <div class="col">Kolom 3</div>
        </div>
    </div>
    <script src="js/bootstrap.min.js"></script>
</body>
</html>

```

---

## Col Number

Dalam contoh sebelumnya kolom dibagi menjadi sama rata. Bootstrap menyediakan cara untuk membagi ukuran kolom secara proporsional, yaitu tidak harus sama besar. Kita bisa mengaturnya secara manual. Namun perlu diketahui, dalam sistem grid bootstrap maksimal kolom dalam 1 baris yaitu sebanyak 12, jika lebih dari 12 maka kolom yang ke 13 akan secara otomatis turun kebawah begitupun seterusnya.

---

```

<!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</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
    <style>
        .row{
            margin-top: 1em;
        }
        [class^="col"]{
            background-color: rgba(86, 61, 124, 0.15);
            border: 1px solid rgba(86, 61, 124, 0.2);
            padding: 0.4em;
            margin-top: 0.5em;
        }
    </style>
</head>
<body>
    <div class="container">
        <div class="row">
            <div class="col-12">Kolom 12</div>
        </div>
        <div class="row">
            <div class="col-6">Kolom 6</div>
            <div class="col-6">Kolom 6</div>
        </div>
        <div class="row">
            <div class="col-4">Kolom 4</div>
            <div class="col-4">Kolom 4</div>

```

```

        <div class="col-4">Kolom 4</div>
    </div>
    <div class="row">
        <div class="col-3">Kolom 3</div>
        <div class="col-3">Kolom 3</div>
        <div class="col-3">Kolom 3</div>
        <div class="col-3">Kolom 3</div>
    </div>
    <div class="row">
        <div class="col-2">Kolom 2</div>
        <div class="col-2">Kolom 2</div>
        <div class="col-2">Kolom 2</div>
        <div class="col-2">Kolom 2</div>
        <div class="col-2">Kolom 2</div>
        <div class="col-2">Kolom 2</div>
    </div>
</div>
<script src="js/bootstrap.min.js"></script>
</body>
</html>

```

---

## Breakpoint

Dengan breakpoint kita dapat mengatur lebar kolom tergantung dari jenis breakpoint yang kita gunakan, dalam bootstrap terdapat 6 jenis breakpoint yang dapat kita gunakan sesuai dengan resolusi layar, berikut breakpoint yang ada di bootstrap:

|                     | xs<br><576px                      | sm<br>≥576px | md<br>≥768px | lg<br>≥992px | xl<br>≥1200px | xxl<br>≥1400px |
|---------------------|-----------------------------------|--------------|--------------|--------------|---------------|----------------|
| Container max-width | None (auto)                       | 540px        | 720px        | 960px        | 1140px        | 1320px         |
| Class prefix        | .col-                             | .col-sm-     | .col-md-     | .col-lg-     | .col-xl-      | .col-xxl-      |
| # of columns        | 12                                |              |              |              |               |                |
| Gutter width        | 1.5rem (.75rem on left and right) |              |              |              |               |                |
| Custom gutters      | <a href="#">Yes</a>               |              |              |              |               |                |
| Nestable            | <a href="#">Yes</a>               |              |              |              |               |                |
| Column ordering     | <a href="#">Yes</a>               |              |              |              |               |                |

## 4.4 Content

### Heading

Pada materi HTML ketika ingin membuat heading maka kita menggunakan tag <h1> sampai <h6>. Bootstrap juga sudah menyediakan class yang bisa kita gunakan untuk membuat heading yaitu class .h1, .h2, .h3, .h4, .h5, .h6

---

```

<!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</title>
    <link rel="stylesheet" href="css/bootstrap.min.css">
</head>
<body>
    <div class="container">

```

```
<p class="h1">Heading 1</p>
<p class="h2">Heading 2</p>
<p class="h3">Heading 3</p>
<p class="h4">Heading 4</p>
<p class="h5">Heading 5</p>
<p class="h6">Heading 6</p>
</div>
<script src="js/bootstrap.min.js"></script>
</body>
</html>
```

---

## Display

Jika kita menginginkan teks yang lebih besar, maka bootstrap sudah menyediakan class .display-1 hingga .display-4 yang bisa kita gunakan

```
<div class="container">
<h1 class="display-1">Display 1</h1>
<h1 class="display-2">Display 2</h1>
<h1 class="display-3">Display 3</h1>
<h1 class="display-4">Display 4</h1>
</div>
```

---

## Lead

Class lead berfungsi untuk menampilkan paragraph menjadi lebih besar untuk font nya.

```
<div class="container">
<p class="lead">Lorem ipsum dolor sit, amet consectetur adipisicing
elit. Eius, corporis ad blanditiis voluptates illo facilis harum
explicabo iure aut suscipit quos aperiam deserunt, a, assumenda aliquam
voluptatibus dolorem provident dolores.</p>
</div>
```

---

## Lists

Untuk membuat daftar bootstrap sudah menyediakan class list-unstyled untuk menghapus karakter list yang ada disebelah kiri bisa berupa bullet atau numbering

```
<ul class="list-unstyled">
<li><a href="home.html">Home</a></li>
<li><a href="article.html">Article</a></li>
<li><a href="about.html">About</a></li>
<li><a href="galery.html">Gallery</a></li>
<li><a href="contact.html">Contact Me</a></li>
</ul>
```

---

## Image Responsive

Untuk membuat gambar responsive bootstrap sudah menyediakan class .img-fluid

```

```

---

## Tabel

Untuk membuat tabel bootstrap sudah menyediakan class .table

```
<table class="table">
<thead>
<tr>
<th scope="col">#</th>
<th scope="col">First</th>
<th scope="col">Last</th>
```



```
<th scope="col">Handle</th>
</tr>
</thead>
<tbody>
<tr>
<th scope="row">1</th>
<td>Mark</td>
<td>Otto</td>
<td>@mdo</td>
</tr>
<tr>
<th scope="row">2</th>
<td>Jacob</td>
<td>Thornton</td>
<td>@fat</td>
</tr>
<tr>
<th scope="row">3</th>
<td colspan="2">Larry the Bird</td>
<td>@twitter</td>
</tr>
</tbody>
</table>
```

---

## 4.5 Utilities

### Color

Color berfungsi untuk merubah warna teks. Bootstrap sudah menyediakan 13 class yang bisa kita gunakan yaitu:

1. .text-primary
2. .text-secondary
3. .text-success
4. .text-danger
5. .text-warning
6. .text-info
7. .text-light
8. .text-dark
9. .text-body
10. .text-muted
11. .text-white
12. .text-black-50
13. .text-white-50

### Background Color

Background color berfungsi untuk merubah warna latar belakang. Bootstrap menyediakan 10 class yang bisa kita gunakan yaitu:

1. .bg-primary
2. .bg-secondary
3. .bg-success
4. .bg-danger
5. .bg-warning
6. .bg-info
7. .bg-light
8. .bg-dark
9. .bg-white
10. .bg-transparent

### **Border**

Border berfungsi untuk membuat garis tepi. Bootstrap menyediakan 5 class yang bisa kita gunakan yaitu:

1. .border
2. .border-top
3. .border-end
4. .border-bottom
5. .border-start

### **Text Alignment**

Text alignment berfungsi untuk mengatur perataan teks. Terdapat 3 class yang sudah disediakan oleh bootstrap yakni:

1. .text-start
2. .text-center
3. .text-end

### **Text Decoration**

Text decoration berfungsi untuk membuat efek pada teks. Terdapat 3 class yang bisa kita gunakan:

1. .text-decoration-underline
2. .text-decoration-line-through
3. .text-decoration-none

### **Font Weight**

Font weight berfungsi untuk mengubah tampilan teks menjadi huruf tebal. Bootstrap sudah menyediakan 5 class yaitu:

1. .fw-bold
2. .fw-bolder
3. .fw-normal
4. .fw-light
5. .fw-lighter

## 4.6 Form

### Form Control

Bootstrap telah menyediakan class .form-control untuk membuat tampilan form menjadi lebih menarik berikut contohnya:

---

```
<form>
<div class="mb-4">
<label for="username">Username</label>
<input type="text" class="form-control">
</div>
<div class="mb-4">
<label for="password">Email</label>
<input type="password" class="form-control">
</div>
<div class="mb-4">
<label for="tanggal_lahir">Tanggal Lahir</label>
<input type="date" class="form-control">
</div>
<div class="mb-4">
<label for="alamat">Alamat</label>
<textarea class="form-control"></textarea>
</div>
</form>
```

---

### Form Checkbox dan Radio

Untuk membuat tampilan checkbox dan radio lebih menarik maka gunakan class .form-check-input untuk tag dan class .form-check-label untuk label

---

```
<form>
<!-- checkbox -->
<div class="form-check">
<input type="checkbox" class="form-check-input">
<label class="form-check-label">Sepakbola</label>
</div>
<div class="form-check">
<input type="checkbox" class="form-check-input">
<label class="form-check-label">Membaca</label>
</div>
<!-- Radio -->
<div class="form-check">
<input type="radio" class="form-check-input">
<label class="form-check-label">Laki-laki</label>
</div>
<div class="form-check">
<input type="radio" class="form-check-input">
```

```
<label class="form-check-label">Perempuan</label>
</div>
</form>
```

---

### Form Select

Untuk merubah tampilan select bootstrap telah menyediakan class .form-select yang bisa kita gunakan sebagai berikut:

```
<div class="mb-4">
<label>Prodi</label>
<select class="form-select">
<option value="Teknik Informatika">Teknik Informatika</option>
<option value="Sistem Operasi">Sistem Operasi</option>
</select>
</div>
```

---

## 4.7 Components

### Button

Untuk membuat button dengan bootstrap menggunakan class .btn seperti berikut:

```
<button class="btn">Tambah</button>
```

---

### Button Color

Ketika kita hanya menggunakan class .btn maka button yang kita buat secara default akan berwarna putih. Untuk mengatasi permasalahan tersebut bootstrap sudah menyediakan class untuk mengatur warna button, terdapat 8 class yang bisa kita gunakan yaitu:

1. .btn-primary
2. .btn-secondary
3. .btn-success
4. .btn-danger
5. .btn-warning
6. .btn-info
7. .btn-light
8. .btn-dark

### Navbar

Bootstrap memudahkan kita untuk membuat menu navigasi. Berikut contoh navbar dengan bootstrap:

```
<nav class="navbar navbar-expand-lg navbar-light bg-light">
<div class="container-fluid">
<a class="navbar-brand" href="#">Navbar</a>
<button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav me-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page" href="#">Home</a>
</li>
<li class="nav-item">
```

```

<a class="nav-link" href="#">Link</a></li>
<li class="nav-item dropdown">
<a class="nav-link dropdown-toggle" href="#" id="navbarDropdown"
role="button" data-bs-toggle="dropdown" aria-expanded="false">
Dropdown </a>
<ul class="dropdown-menu" aria-labelledby="navbarDropdown">
<li><a class="dropdown-item" href="#">Action</a></li>
<li><a class="dropdown-item" href="#">Another action</a></li>
<li><hr class="dropdown-divider"> </li>
<li><a class="dropdown-item" href="#">Something else here</a></li>
</ul></li>
<li class="nav-item">
<a class="nav-link disabled" href="#" tabindex="-1" aria-
disabled="true">Disabled</a></li>
</ul>
</div>
</div>
</nav>

```

---

## Card

Card merupakan komponen bootstrap untuk membuat berbagai fitur tampilan. Berikut cara membuat card di bootstrap

```


---



## Alert



Alert merupakan komponen yang disediakan oleh bootstrap untuk menampilkan peringatan atau pemberitahuan. Untuk membuat alert kita bisa menggunakan class .alert. Selain itu bootstrap juga sudah menyediakan background color untuk alert.



Terdapat 8 class yang bisa kita gunakan yaitu:

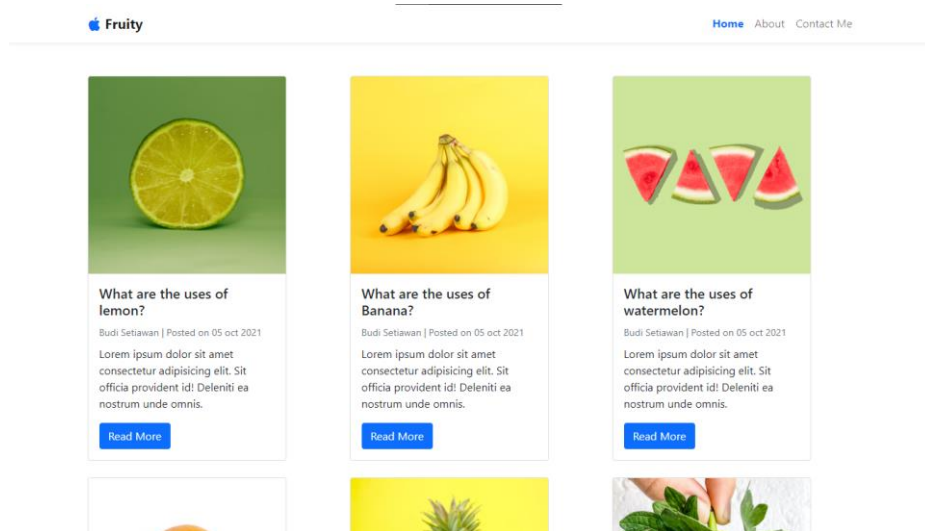


1. .alert-primary
2. .alert-secondary
3. .alert-success
4. .alert-danger
5. .alert-warning
6. .alert-info
7. .alert-light
8. .alert-dark


```

## 4.8 Tugas

1. Buatlah website dengan tata letak seperti berikut (untuk konten tidak harus sama) menggunakan bootstrap



2. Buatlah form seperti berikut dengan melanjutkan soal nomor 1

The screenshot shows a 'Contact Me' form on the 'Fruity' website. The header includes the 'Fruity' logo and navigation links 'Home', 'About', and 'Contact Me'. The form title is 'Contact Me' in blue. Below the title is a message: 'If you need help, please contact us via the form below'. The form contains the following fields: 'Email' (a single input field), 'First Name' (an input field), 'Last Name' (an input field), and 'Message' (a large text area). Below the 'Message' field is a checkbox labeled 'I agree to the terms and conditions'. At the bottom of the form is a blue 'Send' button.

# PERTEMUAN 5

## GITHUB

### 5.1 Pengertian Git

Git merupakan salah satu sistem pengontrol versi (*Version Control System*) pada proyek perangkat lunak yang diciptakan oleh Linus Torvalds. Pengontrol versi bertugas mencatat setiap perubahan pada file proyek yang dikerjakan oleh banyak orang maupun sendiri. Fungsi git adalah untuk mengatur versi dari source code program dengan memberikan tanda baris dan kode mana yang ditambah atau diganti.

### 5.2 Cara Install Git

Berikut merupakan cara instalasi git di windows

#### 1. Download file git

Untuk menginstall git, kita perlu mengunduh file nya terlebih dahulu di situs resmi git yaitu <https://git-scm.com/downloads>. Silahkan untuk mengunduh sesuai dengan system operasi yang digunakan oleh laptop masing-masing dan sesuaikan tipe nya apakah 64 bit atau 32 bit.

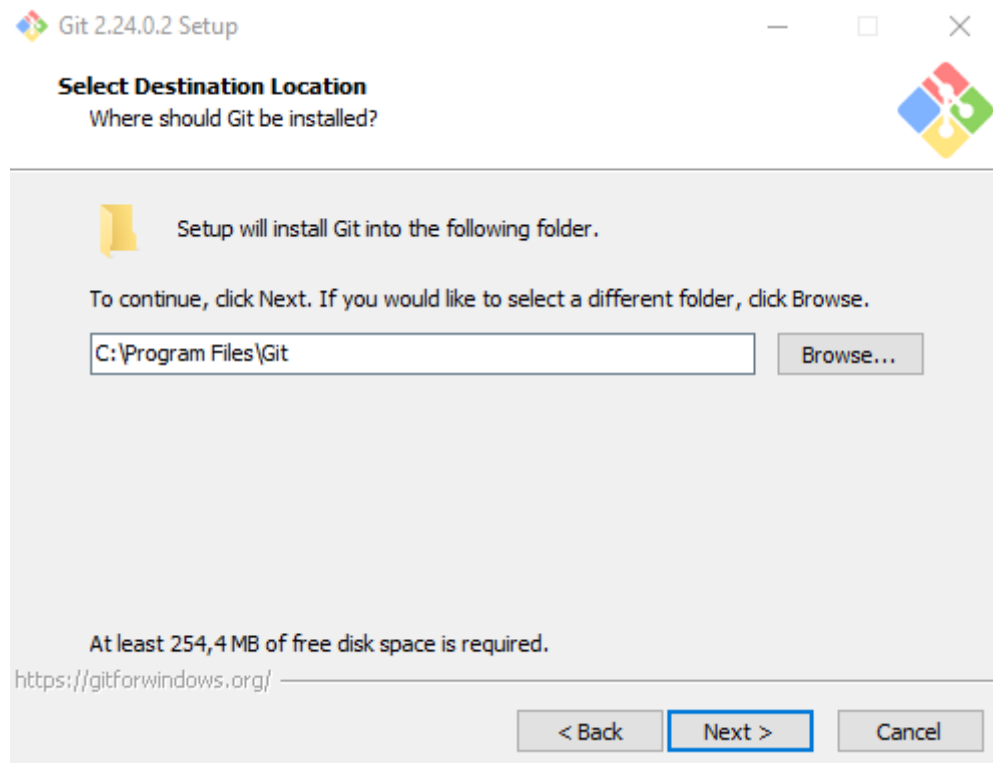
#### 2. Install git

Setelah selesai mengunduh, selanjutnya buka file instalasi git nya. Jika sudah terbuka maka akan tampil halaman berikut



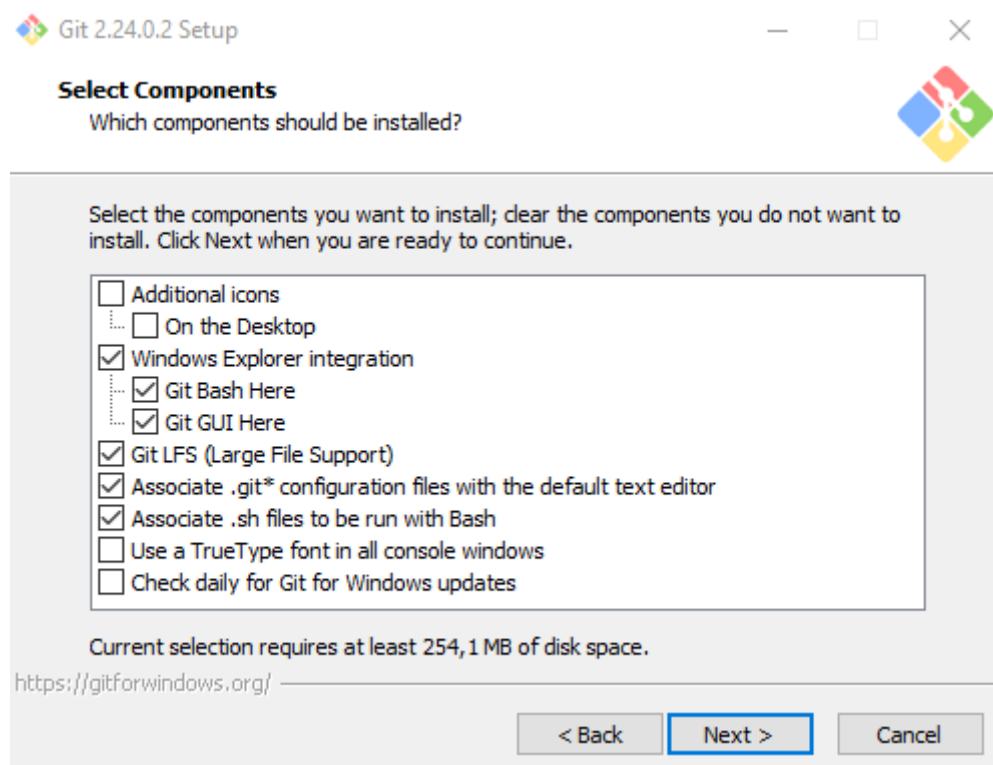
#### 3. Tentukan Lokasi Instalasi Git

Selanjutnya kita tentukan lokasi dimana git akan kita simpan di laptop kita, kalau sudah langsung klik next



#### 4. Pilih Komponen Tambahan

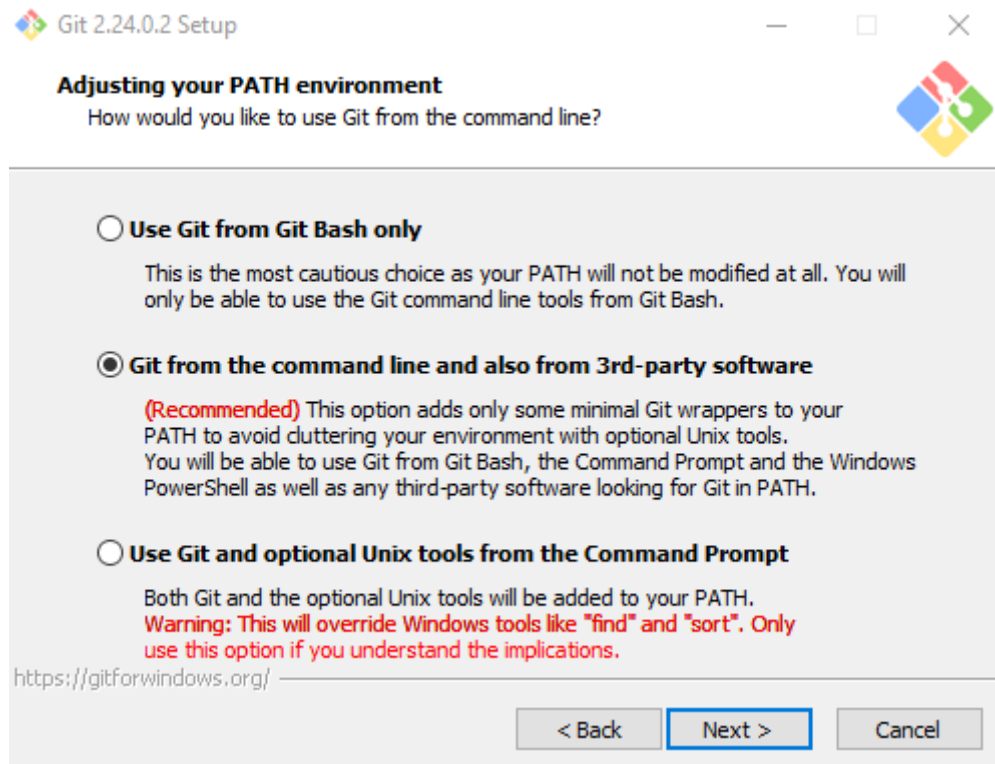
Setelah itu kita pilih komponen tambahan apa saja yang ingin kita gunakan dalam git. Komponen ini berfungsi untuk memperlancar penggunaan git dan mendukung file dengan ukuran besar. Kalau sudah langsung saja klik next





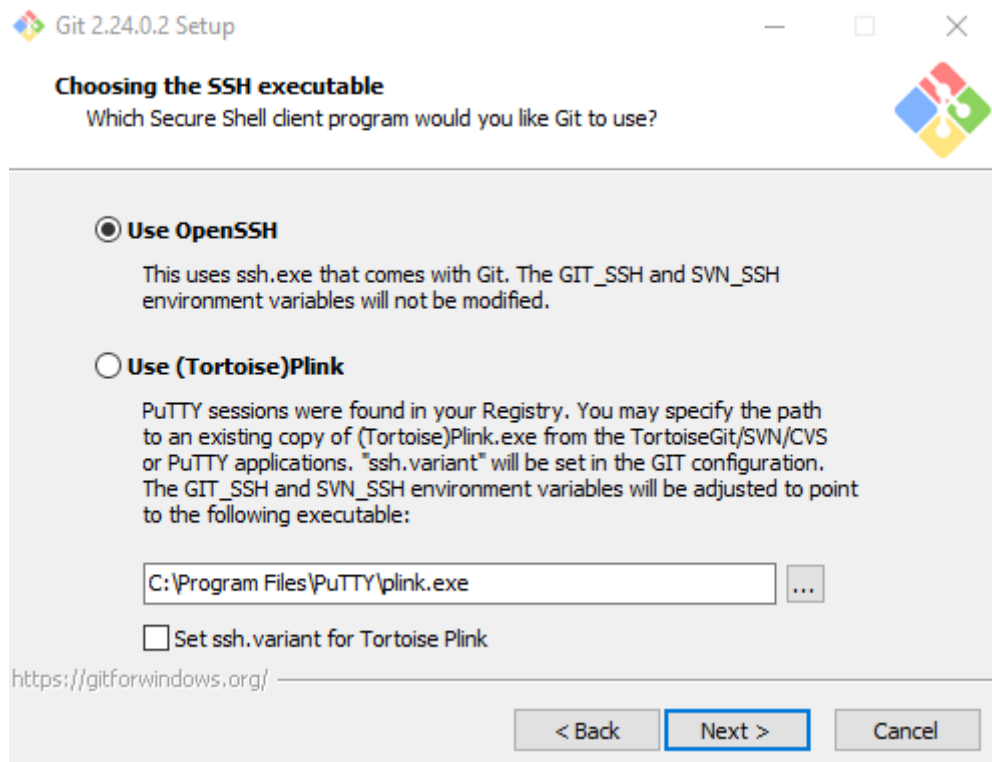
## 5. Atur Environment

Selanjutnya kita pilih path environment yang berfungsi untuk mengeksekusi perintah-perintah yang ada pada git. Silahkan untuk memilih **git from the command line and also 3<sup>rd</sup> party software** supaya ketika kita menjalankan perintah git dapat dikenali di **command prompt**



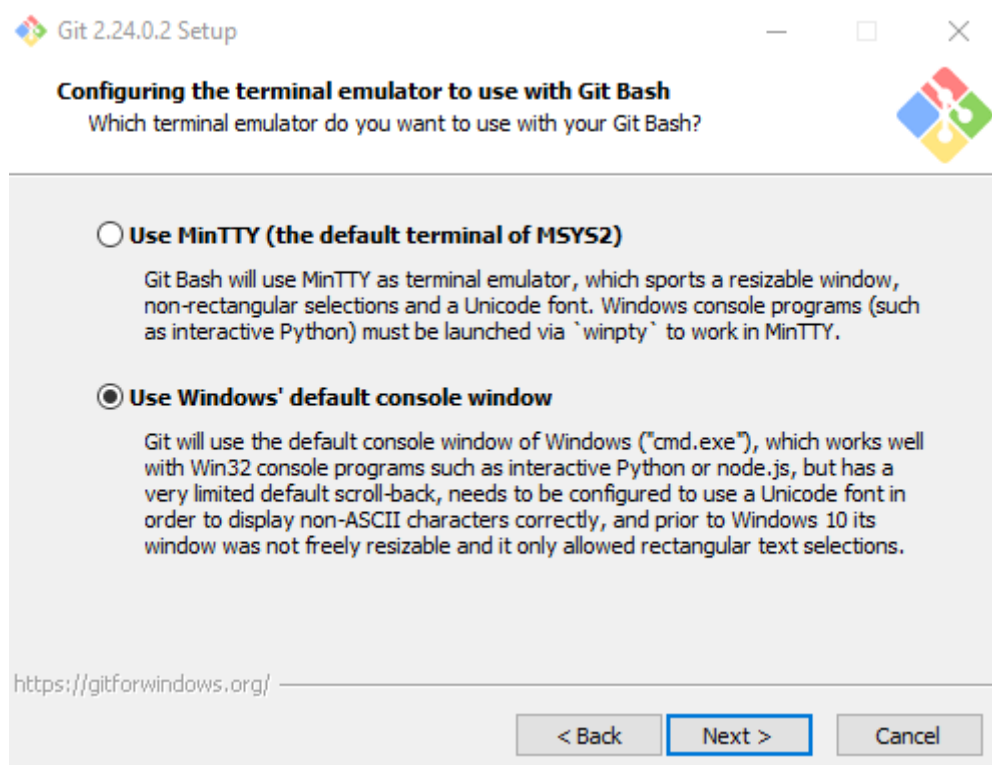
## 6. Pilih SSH

Untuk mengeksekusi SSH, kita bisa menggunakan aplikasi bawaan dari git atau menggunakan platform lain seperti PuTTY dan Bitvise.



## 7. Pilih Emulator Terminal

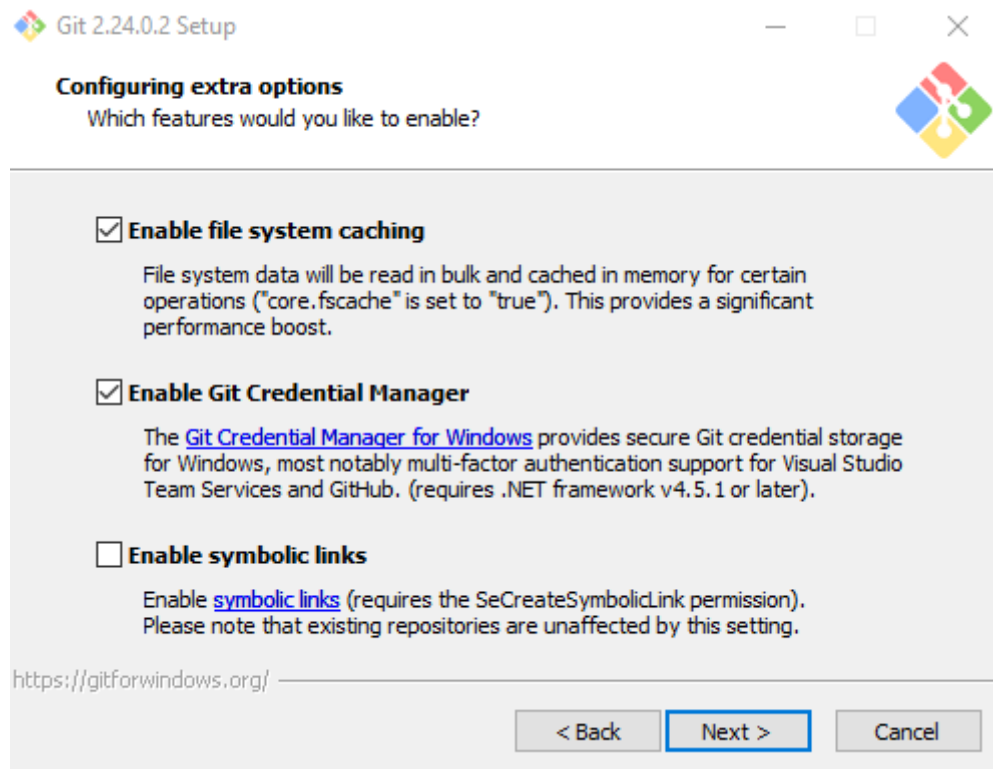
Selanjutnya kita perlu memilih emulator terminal yang akan kita gunakan. Kita bisa menggunakan cmd atau MinTTY.



## 8. Tentukan Opsi Ekstra

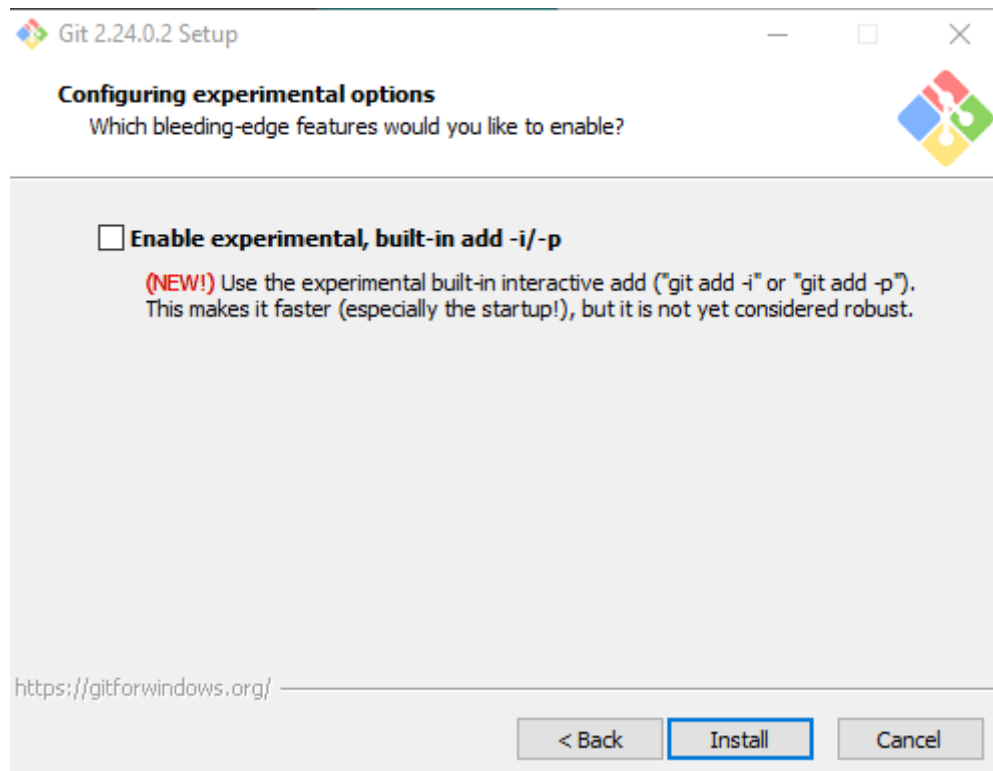
Terdapat beberapa pilihan ekstra yang dapat kita pilih. Diantaranya, pertama pilih Enable File System Caching ini berfungsi supaya git memiliki fungsi system caching. Kemudian pilih Enable Git Credential Manager agar Git bisa

dikombinasikan dengan aplikasi yang lain seperti Visual Studio Code, Android Studio, GitHub dan lain sebagainya.



## 9. Proses Instalasi

Selanjutnya kita langsung klik tombol install



## 10. Cek Version Git

Setelah melakukan instalasi selesai kita perlu mengecek apakah git berhasil di install atau tidak. Untuk mengeceknya kita bisa menggunakan Command Prompt. Kemudian masukkan perintah git --version seperti berikut

```
Microsoft Windows [Version 10.0.19042.1466]
(c) Microsoft Corporation. All rights reserved.

C:\Users\budis>git --version
git version 2.32.0.windows.2
```

### 5.3 Cara Menggunakan Git

Berikut di bawah ini merupakan cara menggunakan git

#### 1. Login Git

Ada berbagai cara yang bisa kita gunakan untuk login ke dalam git, kita bisa menggunakan akun github, gitlab, atau bitbucket. Jika kita tidak memiliki akun dari ketiga platform tersebut, maka kita diharuskan untuk daftar terlebih dahulu. Jika sudah memiliki akun maka kita bisa melakukan login awal pada git menggunakan cmd. Terdapat beberapa perintah yang akan kita gunakan untuk login ke git sebagai berikut:

Masukkan username GitHub menggunakan perintah di bawah ini lalu tekan **ENTER** jika sudah benar

**\$ git config --global user.name "UsernameAnda"**

Kemudian masukkan email yang terdaftar di GitHub menggunakan perintah di bawah ini lalu tekan **ENTER** jika sudah benar

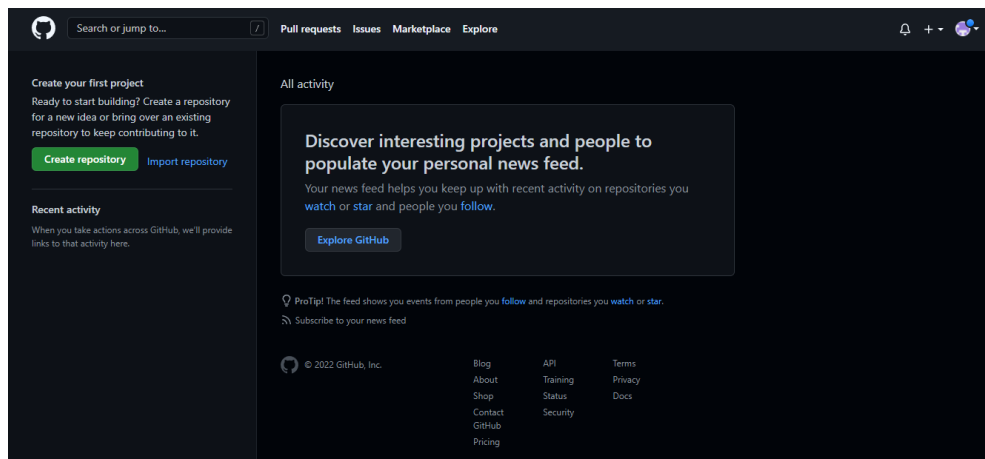
**\$ git config --global user.email isiDenganEmailAnda@gmail.com**

Selanjutnya untuk memastikan apakah proses login berhasil atau tidak, masukkan perintah berikut

**\$ git config --list**

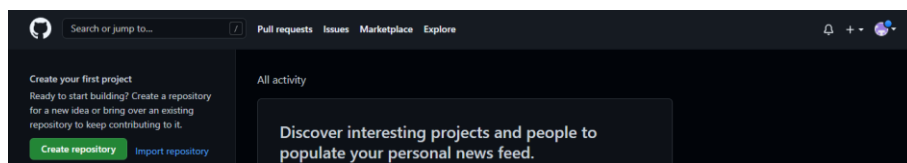
#### 2. Cara Login Github

Silahkan untuk login akun github di halaman berikut <https://github.com/>. Git dan github memiliki hubungan special, git berperan sebagai version control system sedangkan github menjadi hosting atau sebagai penyimpanan kode pemrograman. Setelah anda login, maka akan muncul halaman dashboard github seperti berikut:



### 3. Cara Membuat Repository

Setelah kita berhasil login ke GitHub, kita bisa langsung membuat repository dengan cara klik tombol **Create Repository** pada menu **repositories** untuk membuat repository baru.



Selanjutnya anda akan diarahkan ke halaman berikut

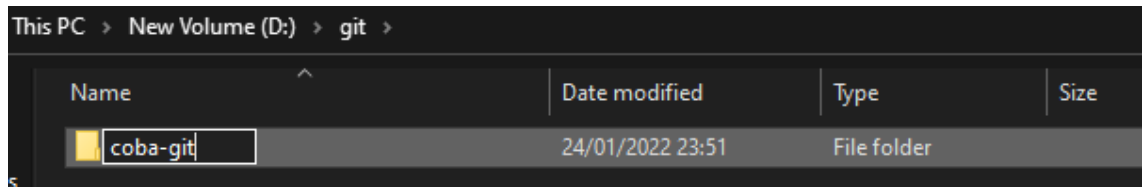
Ada beberapa form yang harus di isi diantaranya sebagai berikut:

- Nama Repository: digunakan untuk identitas repository yang dibuat
- Deskripsi: digunakan untuk deskripsi dari repository yang dibuat
- Jenis repository: jenis repository dibagi menjadi **public** dan **private**. Ketika mengatur repository menjadi **public**, orang lain dapat melihat repository yang

kita buat. Sebaliknya, jika kita mengaturnya sebagai **private**, repository tersebut hanya bisa diakses oleh kita.

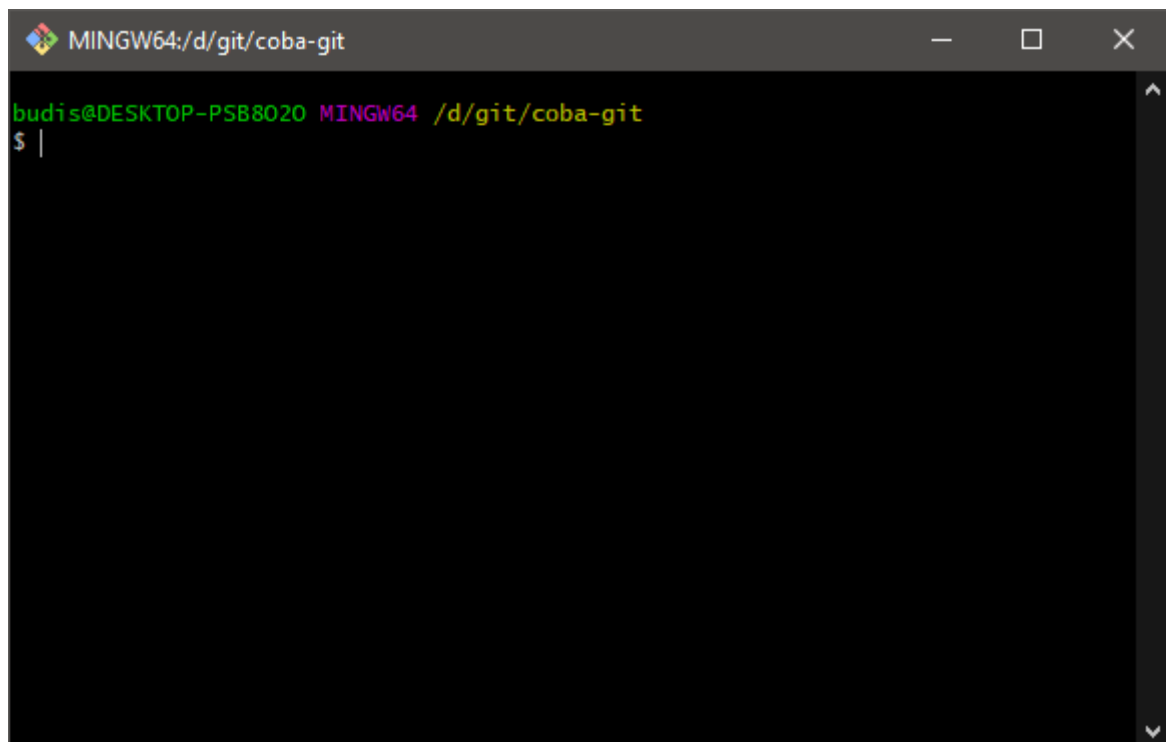
#### 4. Cara Membuat Folder pada Windows

Selanjutnya kita perlu membuat folder pada laptop/pc kita. Fungsinya adalah untuk menyimpan update file dari repository github yang telah dibuat.



#### 5. Buka Folder Menggunakan Git Bash

Setelah membuat folder, selanjutnya kita klik kanan pada folder tersebut kemudian pilih **git bash here**. Setelah itu akan muncul tampilan seperti berikut



#### 6. Mengubah folder menjadi repository

Untuk mengubah folder tersebut menjadi repository kita bisa menggunakan perintah berikut:

**\$ git ini**

#### 7. Menambahkan file ke dalam repository

Untuk bisa menambahkan file ke repository github, kita perlu menerapkan langkah-langkah di bawah ini:

- Buat file index.html di folder yang sudah dibuat sebelumnya
- Buka gitbash lalu masukkan perintah berikut:

```
$ git add index.html
```

## **8. Membuat Commit**

Setelah menambahkan file ke dalam repository, langkah selanjutnya adalah kita perlu untuk melakukan commit. Commit berfungsi untuk menambahkan update file serta komentar. Untuk melakukan commit masukkan perintah berikut:

```
$ git commit -m "first commit"
```

## **9. Remote Repository GitHub**

Remote repository berfungsi untuk mengupload file yang telah kita buat sebelumnya di laptop/komputer kita. Untuk cara remote masukkan perintah berikut:

```
$ git remote add origin git@github.com:UserNameGit>NamaRepository.git
```

## **10. Cara push ke Github**

Langkah terakhir adalah push ke Github yang berfungsi untuk mengupload hasil akhir dari langkah-langkah sebelumnya. Untuk melakukan push masukkan perintah berikut:

```
$ git push -u origin main
```

Perintah di atas akan menampilkan pop up sign in github. Anda perlu login untuk melanjutkan proses push ke Github.

# PERTEMUAN 6

## PHP DASAR

### 6.1 Pengertian PHP

PHP adalah singkatan dari Hypertext Preprocessor, yaitu bahasa pemrograman yang sebenarnya mirip dengan JavaScript. Perbedaannya adalah, PHP sering kali digunakan untuk komunikasi sisi server, sedangkan JavaScript bisa digunakan untuk frontend dan backend. Pengertian sederhananya PHP merupakan bahasa pemrograman yang digunakan untuk meng-generate atau menghasilkan kode HTML. Untuk menjalankan kode PHP kita membutuhkan web server. Web server yang bisa digunakan ada banyak salah satunya adalah apache server yang tersedia di dalam Xampp.

### 6.2 Variabel, Konstanta, Tipe Data, dan Operator

#### 1. Variabel

Variabel adalah suatu lokasi penyimpanan di dalam memori komputer yang berisi data atau informasi yang nilainya telah diketahui maupun belum diketahui. Variabel memiliki nama atau identifier yang digunakan untuk mengakses nilai.

#### Cara Membuat Variabel di PHP

Untuk membuat variabel di PHP menggunakan simbol dollar (\$)

---

```
<?php
$name_book;
$price;
$qty;
?>
```

---

#### Cara assignment nilai ke dalam variabel

Untuk memberikan nilai ke dalam variabel yang di deklarasikan kita menggunakan operator assignment (=)

---

```
<?php
$name_book = "PHP Programming For Beginner";
$price = 5.5;
$qty = 25;
?>
```

---

#### Cara mengakses/menampilkan nilai variabel

Untuk mengakses atau menampilkan nilai variabel bisa menggunakan perintah **echo** kemudian diikuti oleh nama variabelnya

```
<?php
$name_book = "PHP Programming For Beginner";
$price = 5.5;
$qty = 25;
echo "Name Book: $name_book <br>";
echo "Price: $price";
echo "<br>";
echo "QTY: $qty";
?>
```



## 2. Konstanta

konstanta (constant) adalah lokasi penyimpanan di dalam memori komputer yang berisi data atau informasi yang nilainya bersifat tetap dan tidak bisa diubah. Sama seperti variabel, konstanta juga menggunakan nama atau identifier yang digunakan untuk mengakses nilai dari konstanta tersebut.

### Cara Membuat Konstanta

Ada dua cara yang bisa kita gunakan untuk membuat konstanta yang pertama menggunakan keyword **const**, kemudian yang kedua menggunakan function **define**.

---

```
<?php
const PUBLISHER = "GRAMEDIA";
define("AUTHOR", "Rasmus Lerdorf");
?>
```

---

### Cara menampilkan konstanta

Untuk menampilkan konstanta sama seperti variabel menggunakan perintah **echo** akan tetapi ketika ingin menampilkan konstanta kita langsung menulis nama konstanta nya tidak usah menggunakan simbol (\$) seperti berikut:

---

```
<?php
const PUBLISHER = "GRAMEDIA";
define("AUTHOR", "Rasmus Lerdorf");

echo PUBLISHER;
echo "<br>";
echo "Author: ".AUTHOR;
?>
```

---

## 3. Tipe Data

Dalam bahasa pemrograman PHP terdapat 8 jenis tipe data. PHP sama seperti bahasa pemrograman Javascript sebelumnya ketika kita membuat sebuah variabel tidak usah mendeklarasikan tipe datanya terlebih dahulu. Variabel PHP bisa diisi dengan tipe data apapun dan diubah nilainya kapanpun. Berikut merupakan jenis tipe data yang ada di PHP:

### 1. Integer

Integer merupakan tipe data yang berfungsi untuk menampung nilai bilangan bulat. Tipe data ini cocok digunakan seperti jumlah hari, jumlah mahasiswa, jumlah barang dan lain sebagainya.

### 2. Float / Double

Float / Double merupakan tipe data yang berfungsi untuk menampung nilai bilangan pecahan. Tipe data ini cocok digunakan untuk nilai ipk, phi, dan lain sebagainya

### 3. Boolean

Tipe data boolean merupakan tipe data yang paling sederhana karena tipe data ini hanya memiliki 2 nilai saja yaitu true dan false. Tipe data ini cocok digunakan dalam percabangan ataupun perulangan.

### 4. String

Tipe data string merupakan tipe data yang berfungsi untuk menampung nilai karakter atau kumpulan karakter. Seperti a,b,saya, belajar bahasa pemrograman, dan lain sebagainya. Untuk membedakan antara karakter dan kumpulan karakter lebih baik ketika assignment variabel untuk karakter menggunakan tanda kutip satu (‘’) sedangkan untuk kumpulan karakter menggunakan tanda kutip 2 (“”).

### 5. Array

Array dalam bahasa Indonesia artinya larik yang berarti tipe data ini merupakan tipe data yang berisi kumpulan tipe data lain dan tersusun secara berurutan. Tipe data ini cocok digunakan untuk data yang banyak, karena dengan kita menggunakan array ini menjadi jauh lebih mudah dan juga efisien.

## Cara Membuat Array

Terdapat 2 cara yang bisa kita gunakan untuk membuat array di php. Cara yang pertama menggunakan keyword array sedangkan cara yang kedua menggunakan kurung siku.

---

```
<?php
$books = array("PHP For Beginner", "Javascript For
Beginner", "Bootstrap For Beginner");
$authors = ["Rasmus Lerdorf", "Brendan Eich", "Mark Otto"];
?>
```

---

## Cara Mengakses Elemen Array

Untuk mengakses elemen array kita cukup tuliskan nama array kemudian diikuti oleh nomor index yang ada di dalam kurung siku.

---

```
<?php
$books = array("PHP For Beginner", "Javascript For
Beginner", "Bootstrap For Beginner");
$authors = ["Rasmus Lerdorf", "Brendan Eich", "Mark Otto"];
echo $books[1];
echo "<br>";
echo $authors[2];
?>
```

---

## Cara Menambah Elemen Array

Walaupun array sudah di deklarasikan apabila data di dalam array dirasa masih kurang kita bisa menambah elemen baru ke dalam array tersebut. Seperti berikut:

---

```
<?php
$books = array("PHP For Beginner", "Javascript For
Beginner", "Bootstrap For Beginner");
$authors = ["Rasmus Lerdorf", "Brendan Eich", "Mark Otto"];

$books[3] = "HTML & CSS For Beginner";
$books[] = "PHP Restful API";

var_dump($books);
?>
```

---

## Cara Menghapus Elemen Array

Berbeda dengan cara menambah elemen array. Kalau kita ingin menghapus elemen array maka harus menggunakan function unset() seperti berikut:

---

```
<?php
$books = array("PHP For Beginner", "Javascript For
Beginner", "Bootstrap For Beginner");
$authors = ["Rasmus Lerdorf", "Brendan Eich", "Mark Otto"];
$books[3] = "HTML & CSS For Beginner";
$books[] = "PHP Restful API";
var_dump($books);
unset($books[4]);
var_dump($books);
?>
```

---

## Array Asosiatif

Array asosiatif merupakan array yang indexnya bisa kita ubah menjadi string bukan lagi menjadi nomor. Untuk membuat array asosiatif caranya sama seperti array biasa akan tetapi untuk array asosiatif menggunakan tanda =>

---

```
<?php
$books = array(
    "name" => "PHP For Beginner",
    "price" => 5.5,
    "qty" => 15,
    "author" => "Rasmus Lerdorf"
);

echo "Name: ".$books["name"];
echo "<br>";
echo "Price: ".$books["price"];
?>
```

---

## 6. Object

Object merupakan tipe data khusus yang terdiri dari berbagai macam tipe data dan di dalam object dapat memiliki method. Tipe data ini cocok kalau kita menggunakan oop.

## 7. Resource

## 8. Null

### 4. Operator

Operator pada PHP berfungsi untuk melakukan operasi pada suatu nilai dan variabel pada PHP. Terdapat 11 operator yang bisa kita gunakan di PHP

#### 1. Operator Aritmatika

Operator aritmatika merupakan operator untuk melakukan operasi aritmatika.

Operator aritmatika terdiri dari:

| Nama Operator | Simbol |
|---------------|--------|
| Penjumlahan   | +      |
| Pengurangan   | -      |
| Perkalian     | *      |
| Pembagian     | /      |
| Pangkat       | **     |
| Modulus       | %      |

#### 2. Operator Increment dan Decrement

Operator increment dan decrement merupakan operator yang digunakan untuk menambah +1 (tambah satu) dan mengurangi -1 (kurangi dengan satu). Operator increment menggunakan simbol ++, sedangkan decrement menggunakan simbol --

#### 3. Operator Perbandingan

Operator perbandingan digunakan untuk membandingkan 2 buah operand atau lebih. Operator perbandingan terdiri dari:

| Nama Operator           | Simbol     |
|-------------------------|------------|
| Sama Dengan             | ==         |
| Identik Dengan          | ===        |
| Tidak Sama Dengan       | != atau <> |
| Tidak Identik Dengan    | !==        |
| Kurang Dari             | <          |
| Lebih Dari              | >          |
| Kurang Dari Sama Dengan | <=         |
| Lebih Dari Sama Dengan  | >=         |

#### 4. Operator Logika

Operator logika digunakan untuk membandingkan 2 buah kondisi logika.

Operator logika terdiri dari:

| Nama Operator | Simbol |
|---------------|--------|
| AND           | &&     |
| OR            |        |
| XOR           | XOR    |
| NOT           | !      |

#### 5. Operator String

Operator string digunakan untuk melakukan penyambungan string atau biasa disebut dengan konkatenasi string. Operator ini menggunakan simbol titik (.)

#### 6. Operator Bitwise

Operator bitwise digunakan untuk menangani operasi logika bilangan biner.

#### 7. Operator Assignment

Operator assignment digunakan untuk memasukkan nilai ke dalam variabel.

#### 8. Operator Error Control

Operator error control digunakan untuk mengabaikan pesan kesalahan pada sebuah perintah. Operator ini menggunakan simbol (@)

#### 9. Operator Array

Operator array digunakan untuk melakukan operasi penggabungan dan perbandingan array

| Nama Operator | Simbol |
|---------------|--------|
| Union         | +      |
| Equality      | ==     |
| Identity      | ===    |
| Inequality    | !=     |
| Inequality    | <>     |
| Non Identity  | !==    |

## 6.3 Percabangan

### 1. If

Percabangan if jika kondisi yang di cek bernilai True, maka blok statement if akan dijalankan. Penulisan struktur dasar if membutuhkan 2 bagian, yakni kondisi dan statement.

---

```
<?php
$price = 95000;
if ($price > 70000) {
    echo "Very Expensive";
}
```

---

## 2. Else dan Else If

### - Else

Percabangan else jika kondisi if bernilai false maka statament else yang akan dijalankan.

---

```
<?php
$price = 55000;

if ($price > 70000) {
    echo "Very Expensive";
} else {
    echo "Normal";
}
```

---

### - Else If

Untuk kode program yang sangat kompleks, kita bisa menggunakan else if.

---

```
<?php
$price = 35000;

if ($price >= 70000):
    echo "Very Expensive";
elseif ($price > 40000):
    echo "Normal";
else:
    echo "Cheap";
endif;
```

---

## 3. Switch Case

Perintah switch digunakan untuk memeriksa kebenaran suatu nilai dengan memiliki banyak pemilihan atau nilai pengecekannya hampir sama dengan penggunaan if else.

---

```
<?php
$rank = 1;
switch ($rank) {
    case 1:
        echo "Gold Medal";
        break;
    case 2:
        echo "Silver Medal";
        break;
    case 3:
        echo "Bronze Medal";
        break;
    default:
        echo "No Ranking";
}
```

---

```
        break;
    }
}
```

---

## 6.4 Perulangan

Perulangan merupakan perintah untuk mengulang beberapa baris perintah. Dalam bahasa pemrograman PHP Terdapat 4 perulangan yang bisa kita gunakan:

### 1. For

Perulangan for merupakan perulangan yang termasuk ke dalam jenis *counted loop* karena kita bisa menentukan jumlah perulangannya.

```
<?php
for ($i = 1; $i <= 10; $i++) {
    $multiplication = 5 * $i;
    echo "10 x $i = $multiplication <br>";
}
```

---

### 2. While

Perulangan while merupakan perulangan yang termasuk ke dalam jenis *uncounted loop* karena biasanya digunakan untuk mengulang sesuatu yang belum jelas jumlah pengulangannya.

```
<?php
$i = 0;
while ($i < 5) {
    echo $i + 1 . "<br>";
    $i++;
}
```

---

### 3. Do While

Do While sama seperti while hanya saja do while akan selalu melakukan pengulangan sebanyak 1 kali kemudian baru melakukan pengecekan kondisi. Sedangkan while akan mengecek kondisi terlebih dahulu, baru melakukan perulangan.

```
<?php
$i = 9;
do {
    echo "$i <br>";
} while ($i < 9);
```

---

### 4. Foreach

Perulangan foreach sama seperti perulangan for, hanya saja foreach khusus digunakan untuk menampilkan array

```
<?php
$animals = ["Lion", "Tiger", "Crocodile", "Cat", "Dog"];
```

---

```
foreach ($animals as $animal) {  
    echo "$animal <br>";  
}
```

---

## 6.5 Function

Function adalah kode program yang dibuat khusus untuk mempersingkat sebuah aksi yang ingin dilakukan di dalam sebuah penulisan kode program. Function dapat dipanggil berkali-kali untuk melakukan tugas tertentu dalam penulisan kode program.

### Cara Membuat Function

Untuk membuat function di PHP diawali dengan menuliskan keyword **function** kemudian diikuti oleh nama function tersebut

```
<?php  
function show_name() {  
    echo "My name is Budi Setiawan";  
}
```

---

### Cara Memanggil Function

Untuk memanggil atau menggunakan function bisa kita lakukan dengan cara menulis nama function kemudian diikuti oleh argumen nya. Apabila function yang kita buat tidak memiliki argumen, maka tetap ditulis dengan tanda kurung kosong

```
<?php  
function show_name() {  
    echo "My name is Budi Setiawan";  
}  
  
show_name();
```

---

### Membuat Argumen Function

Dalam sebuah function argumen adalah variabel input untuk fungsi tersebut. Dengan adanya argumen ini dapat membuat function menjadi lebih fleksibel

```
<?php  
function show_name($name) {  
    echo "My name is $name";  
}  
  
show_name("Kevin Alam");
```

---

### Mengembalikan Nilai Function



Supaya function yang kita buat menjadi lebih fleksibel lebih baik function tersebut dapat mengembalikan nilai bukan langsung menampilkan nilai dengan perintah echo.

---

```
<?php
function show_name($name)
{
    return $name;
}

echo "My name is " . show_name("Rina Nurmallasari");
```

---

## 6.6 Session

Dalam bahasa pemrograman PHP, session merupakan data yang disimpan dalam suatu server yang dapat digunakan secara global di server tersebut, dimana data tersebut spesifik merujuk ke user tertentu. Contoh penggunaan session adalah ketika kita telah login di halaman tertentu, maka ketika membuka halaman lain, php akan mengingat bahwa kita telah login.

### Cara Pembuatan Session

Untuk membuat session, kita harus menggunakan function **session\_start()**. Function session\_start akan membuat sebuah cookie yang berisi referensi ke file session di web browser. Berikut cara membuat session:

---

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Session</title>
</head>
<body>
    <h2>Session</h2>
</body>
</html>
```

---

### Cara Input Data Session

---

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Session</title>
</head>
<body>
    <h2>Session</h2>
    <?php
        $_SESSION['name'] = "Budi Setiawan";
```

```
$_SESSION['roles'] = "Admin";
?>
</body>
</html>
```

---

## Cara Mengakses Data Session

```
<?php
session_start();
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <title>Session</title>
</head>
<body>
    <h2>Session</h2>
    <?php
    echo $_SESSION['name'];
    echo "<br>";
    echo $_SESSION['roles'];
    ?>
</body>
</html>
```

---

## Menghapus Session

Terdapat 2 cara yang bisa kita lakukan untuk menghapus session. Pertama dengan menggunakan function unset() untuk menghapus data session satu persatu dan yang kedua menggunakan function session\_unset() untuk menghapus semua data session.

### 1. unset()

```
<?php
session_start();
unset($_SESSION['name']);
?>
```

---

### 2. session\_unset()

```
<?php
session_start();
session_unset();
?>
```

---

## 6.7 Tugas

1. Buatlah sebuah website pendaftaran rute penerbangan dengan ketentuan sebagai berikut:

| Format Masukan                     | Format Keluaran                           |
|------------------------------------|---|
| User memasukkan data nama maskapai | Nomor                                     |
| User memilih bandara asal          | Tanggal (Ketika data penerbangan diinput) |
| User memilih bandara tujuan        | Nama Maskapai                             |

|                                  |                    |
|----------------------------------|--------------------|
| User memasukkan data harga tiket | Asal Penerbangan   |
|                                  | Tujuan Penerbangan |
|                                  | Harga Tiket        |
|                                  | Pajak              |
|                                  | Total Harga Tiket  |

### **Ketentuan Pajak:**

1. Sumber pajak berasal dari bandara asal dan bandara tujuan
2. Masing-masing bandara mempunyai tarif pajak sendiri
3. Total pajak diambil dari jumlah pajak bandara asal dan bandara tujuan
4. Berikut rincian pajak masing-masing bandara

| No | Bandara Asal        | Pajak |
|----|---------------------|-------|
| 1  | Soekarno Hatta      | 65000 |
| 2  | Husein Sastranegara | 50000 |
| 3  | Abdul Rachman Saleh | 40000 |
| 4  | Juanda              | 30000 |

| No | Bandara Tujuan       | Pajak |
|----|----------------------|-------|
| 1  | Ngurah Rai           | 85000 |
| 2  | Hasanuddin           | 70000 |
| 3  | Inanwatan            | 90000 |
| 4  | Sultan Iskandar Muda | 60000 |

### **Catatan:**

1. Bandara asal dan bandara tujuan menggunakan array kemudian tampilkan pada select option menggunakan for atau foreach
2. Urutkan nama bandara asal dan bandara tujuan menggunakan function array
3.  $\text{Pajak} = \text{pajak}(\text{bandara asal}) + \text{pajak}(\text{bandara tujuan})$
4.  $\text{Total harga tiket} = \text{pajak} + \text{total harga tiket}$
5. Seleksi kondisi boleh menggunakan if atau switch case

# PERTEMUAN 7

## PHP MYSQL (CRUD)

CRUD merupakan singkatan dari Create, Read, Update, dan Delete adalah untuk memanipulasi data dasar untuk database. Nantinya akan terdapat kode program untuk menampilkan data dari database (Read), membuat data (Create), mengedit data (Update), dan menghapus data (Delete)

### 7.1 Membuat Database

Pertama kita harus membuat database nya terlebih dahulu, kemudian setelah membuat database kita lanjut untuk membuat tabelnya.

---

```
-- Buat database dan tabel

CREATE DATABASE my_blog;

use my_blog;

CREATE TABLE articles(
    id int(11) PRIMARY KEY AUTO_INCREMENT,
    title varchar(100),
    content text,
    category varchar(50),
    thumbnail varchar(255)
);
```

---

### 7.2 Membuat file koneksi.php

Untuk menghubungkan PHP dengan MYSQL kita membutuhkan file koneksi seperti berikut:

---

```
<?php
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "my_blog";

$koneksi = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
```

---

### Penjelasan

- \$db\_host: berfungsi untuk menyimpan database host yang kita gunakan
- \$db\_user: berfungsi untuk menyimpan username user, secara default apabila kita tidak membuat user baru maka gunakan root

- \$db\_pass: berfungsi untuk menyimpan password user, apabila tidak menggunakan password maka dikosongkan
- \$db\_name: berfungsi untuk menyimpan nama database yang akan kita gunakan

Untuk menghubungkan antara PHP dan Mysql maka kita menggunakan fungsi **mysqli\_connect** dengan argument pertama diisi oleh \$db\_host, agument kedua diisi oleh \$db\_user, argument ketiga diisi oleh \$db\_pass, dan terakhir diisi oleh \$db\_name

### 7.3 Membuat file add.php

File ini berfungsi untuk menampilkan form tambah data

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Add</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap
.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>

<body>
<div class="container my-5">
<div class="row">
<div class="col-md-12">
<div class="card border-0">
<div class="card-header">
<div class="d-flex justify-content-between">
<h2>Add Article</h2>
<a href="#" class="btn btn-primary">List Articles</a>
</div>
</div>
<div class="card-body">
<form action="add-process.php" method="post"
enctype="multipart/form-data">
<div class="mb-4">
<label for="title" class="form-label">Title</label>
<input type="text" name="title" id="title" class="form-control">
</div>
<div class="mb-4">
<label for="content" class="form-label">Content</label>
<textarea name="content" id="content" class="form-
control"></textarea>
</div>
<div class="mb-4">
<label for="category" class="form-label">Category</label>
<select name="category" id="category" class="form-select">
<option value="Programming">Programming</option>
<option value="Business">Business</option>
<option value="News">News</option>
</select>
</div>
<div class="mb-4">
<label for="thumbnail" class="Thumbnail">Thumbnail</label>
<input type="file" name="thumbnail" id="thumbnail" class="form-
control">
</div>
```

```
<button type="submit" name="add" class="btn btn-primary">Add</button>
</form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

---

## 7.4 Membuat file add-process.php

File ini berfungsi untuk melakukan pemrosesan pada form add yang sebelumnya telah di buat, dengan adanya file ini data yang kita masukkan di form akan langsung ditambahkan ke dalam database.

---

```
<?php
include 'koneksi.php';
if (isset($_POST['add'])) {
    $title = $_POST['title'];
    $content = $_POST['content'];
    $category = $_POST['category'];
    $thumbnail = $_POST['thumbnail'];

    $extension_allowed = array('png', 'jpg');
    $name = $_FILES['thumbnail']['name'];
    $x = explode('.', $name);
    $extension = strtolower(end($x));
    $size = $_FILES['thumbnail']['size'];
    $file_tmp = $_FILES['thumbnail']['tmp_name'];

    if (in_array($extension, $extension_allowed) === true) {
        if ($ukuran < 1044070) {
            move_uploaded_file($file_tmp, 'images/' . $name);
            $query = mysqli_query($koneksi, "INSERT INTO articles
VALUES (NULL, '$title', '$content', '$category', '$name')");
            if ($query) {
                $message = "Data berhasil ditambahkan";
                $message = urlencode($message);
                header("Location:index.php?message={$message}");
            } else {
                $message = "Data gagal ditambahkan";
                $message = urlencode($message);
                header("Location:add.php?message={$message}");
            }
        } else {
            $message = "Ukuran File Terlalu Besar";
            $message = urlencode($message);
            header("Location:add.php?message={$message}");
        }
    } else {
        $message = "Extension tidak diperbolehkan";
        $message = urlencode($message);
        header("Location:add.php?message={$message}");
    }
}
```

---

## 7.5 Membuat File Index.php

File index merupakan file utama yang berfungsi untuk menampilkan semua artikel yang ada di database.

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Articles</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap
.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgppJLIIm9Nao0YzlztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
</head>
<body>
<div class="container my-5">
<div class="row">
<div class="col-md-12">
<?php
if (isset($_GET['message'])) {
$message = $_GET['message'];
?>
<div class="alert alert-success my-4"><?= $message ?></div>
<?php
}
?>
<div class="card border-0">
<div class="card-header">
<div class="d-flex justify-content-between">
<h2>List Articles</h2>
<a href="add.php" class="btn btn-primary">Add Article</a>
</div>
</div>
<div class="card-body">
<table class="table">
<thead>
<tr>
<th>#</th>
<th>Title</th>
<th>Content</th>
<th>Category</th>
<th>Thumbnail</th>
<th>Act</th>
</tr>
</thead>
<tbody>
<?php
include "koneksi.php";
$no = 1;
$query = mysqli_query($koneksi, "SELECT * FROM articles");
foreach ($query as $data) {
?>
<tr>
<td><?= $no++ ?></td>
<td><?= $data['title'] ?></td>
<td><?= $data['content'] ?></td>
<td><?= $data['category'] ?></td>
<td>

</td>
<td>
```

```
<div class="btn-group">
<a href="edit.php?id=<?= $data['id'] ?>" class="btn btn-
warning">Edit</a>
<a href="delete.php?id=<?= $data['id'] ?>" class="btn btn-
danger">Delete</a>
</div>
</td>
</tr>
<?php
}
?>
</tbody>
</table>
</div>
</div>
</div>
</div>
</body>
</html>
```

## 7.6 Membuat file edit.php

File ini berfungsi untuk menampilkan data mana yang di edit ke dalam form

```
<?php
if (isset($_GET['id'])) {
include "koneksi.php";
$id = $_GET['id'];
$query = mysqli_query($koneksi, "SELECT * FROM articles WHERE
id='$id'");
$data = mysqli_fetch_array($query);
?>
<!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>Edit</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap
.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgppJLIIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOMLASjC"
crossorigin="anonymous">
</head>
<body>
<div class="container my-5">
<div class="row">
<div class="col-md-12">
<?php
if (isset($_GET['message'])) {
$message = $_GET['message'];
?>
<div class="alert alert-danger my-4"><?= $message ?></div>
<?php
}
?>
<div class="card border-0">
<div class="card-header">
<div class="d-flex justify-content-between">
```



```
<h2>Edit Article</h2>
<a href="index.php" class="btn btn-primary">List Articles</a>
</div>
</div>
<div class="card-body">
<form action="edit-process.php" method="post"
enctype="multipart/form-data">
<input type="hidden" name="id" value="<?= $id ?>">
<div class="mb-4">
<label for="title" class="form-label">Title</label>
<input type="text" name="title" id="title" class="form-control"
value="<?= $data['title'] ?>">
</div>
<div class="mb-4">
<label for="content" class="form-label">Content</label>
<textarea name="content" id="content" class="form-control"><?=
$data['content'] ?></textarea>
</div>
<div class="mb-4">
<label for="category" class="form-label">Category</label>
<select name="category" id="category" class="form-select">
<option value="Programming" <?= $data['category'] == "Programming" ?
"Selected" : "" ?>>Programming</option>
<option value="Business" <?= $data['category'] == "Business" ?
"Selected" : "" ?>>Business</option>
<option value="News" <?= $data['category'] == "News" ? "Selected" :
"" ?>>News</option>
</select>
</div>
<div class="mb-4">
<label for="thumbnail" class="Thumbnail">Thumbnail</label>
<br>

<input type="file" name="thumbnail" id="thumbnail" class="form-
control">
</div>
<button type="submit" name="edit" class="btn btn-
primary">Edit</button>
</form>
</div>
</div>
</div>
</div>
</body>
</html>
<?php
}
?>
```

## 7.7 Membuat file edit-process.php

```
<?php
include 'koneksi.php';
```

```

if (isset($_POST['edit'])) {
    $title = $_POST['title'];
    $id = $_POST['id'];
    $content = $_POST['content'];
    $category = $_POST['category'];
    $thumbnail = $_POST['thumbnail'];

    $extension_allowed = array('png', 'jpg');
    $name = $_FILES['thumbnail']['name'];
    $x = explode('.', $name);
    $extension = strtolower(end($x));
    $size = $_FILES['thumbnail']['size'];
    $file_tmp = $_FILES['thumbnail']['tmp_name'];

    if (in_array($extension, $extension_allowed) === true) {
        if ($ukuran < 1044070) {
            move_uploaded_file($file_tmp, 'images/' . $name);
            $query = mysqli_query($koneksi, "UPDATE articles SET
title='$title', content='$content', category='$category',
thumbnail='$name' WHERE id='$id'");
            if ($query) {
                $message = "Data berhasil diubah";
                $message = urlencode($message);
                header("Location:index.php?message={$message}");
            } else {
                $message = "Data gagal diubah";
                $message = urlencode($message);
                header("Location:add.php?message={$message}");
            }
        } else {
            $message = "Ukuran File Terlalu Besar";
            $message = urlencode($message);
            header("Location:add.php?message={$message}");
        }
    } else {
        $query = mysqli_query($koneksi, "UPDATE articles SET
title='$title', content='$content', category='$category' WHERE
id='$id'");
        if ($query) {
            $message = "Data berhasil diubah";
            $message = urlencode($message);
            header("Location:index.php?message={$message}");
        } else {
            $message = "Data gagal diubah";
            $message = urlencode($message);
            header("Location:add.php?message={$message}");
        }
    }
}
}

```

---

## 7.8 Membuat file delete.php

File ini berfungsi untuk menghapus data yang ada di database

---

```

<?php
if (isset($_GET['id'])) {
    include "koneksi.php";
    $id = $_GET['id'];
    $query = mysqli_query($koneksi, "DELETE FROM articles WHERE
id='$id'");
    if ($query) {
        $message = "Data berhasil dihapus";
        $message = urlencode($message);
        header("Location:index.php?message={$message}");
    }
}

```

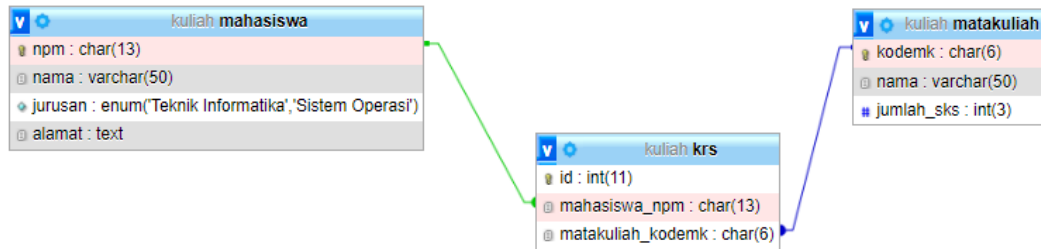
```

    } else {
        $message = "Data gagal dihapus";
        $message = urlencode($message);
        header("Location: add.php?message={$message}");
    }
}

```

## 7.9 Tugas

1. Buatlah database seperti berikut:



2. Buatlah crud untuk masing-masing tabel dan untuk tabel krs tampilkan datanya seperti berikut:

| No | Nama Lengkap        | Mata Kuliah                 | Keterangan  |
|----|---------------------|-----------------------------|---|
| 1  | Siska Putri         | Basis Data                  | Siska Putri Mengambil Mata Kuliah Basis Data (3 SKS)                          |
| 2  | Ujang Aziz          | Pemrograman Berbasis Web    | Ujang Aziz Mengambil Mata Kuliah Pemrograman Berbasis Web (3 SKS)             |
| 3  | Veronica Setyano    | Basis Data                  | Veronica Setyano Mengambil Mata Kuliah Basis Data (3 SKS)                     |
| 4  | Rizka Nurmala Putri | Algoritma dan Struktur Data | Rizka Nurmala Putri Mengambil Mata Kuliah Algoritma dan Struktur Data (3 SKS) |
| 5  | Eren Putra          | Kajian Jurnal Informatika   | Eren Putra Mengambil Mata Kuliah Kajian Jurnal Informatika (2 SKS)            |
| 6  | Putra Abdul Rachman | Basis Data                  | Putra Abdul Rachman Mengambil Mata Kuliah Basis Data (3 SKS)                  |
| 7  | Rahmat Andriyadi    | Basis Data                  | Rahmat Andriyadi Mengambil Mata Kuliah Basis Data (3 SKS)                     |
| 8  | Ayu Puspitasari     | Pemrograman Berbasis Web    | Ayu Puspitasari Mengambil Mata Kuliah Pemrograman Berbasis Web (3 SKS)        |
| 9  | Putri Ayuni         | Pemrograman Berbasis Web    | Putri Ayuni Mengambil Mata Kuliah Pemrograman Berbasis Web (3 SKS)            |
| 10 | Andri Muhammad      | Algoritma dan Struktur Data | Andri Muhammad Mengambil Mata Kuliah Algoritma dan Struktur Data (3 SKS)      |

# **PERTEMUAN 8**

## **PHP MYSQL AUTHENTICATION & AUTHORIZATION**

### **8.1 Pengertian Authentication**

Authentication merupakan proses dimana seorang user mendapatkan hak akses kepada suatu website/aplikasi. Seorang user melakukan login kedalam suatu infrastruktur jaringan dan sistem mengenali user ID ini dan menerimanya untuk kemudian diberikan akses terhadap resources jaringan sesuai dengan authorisasi yang dia terima.

### **8.2 Jenis Based Authentication**

Terdapat 2 jenis based authentication yang dapat kita gunakan yaitu:

#### **1. Session Based Authentication**

Pada session authentication, server akan membuat session untuk user setelah user berhasil login. Session id akan disimpan di dalam browser yang digunakan oleh user. Selama user tetap dalam keadaan login, session dapat dikirim pada setiap request yang dilakukan oleh user. Server lalu melakukan perbandingan dengan session id yang disimpan pada session dengan informasi session yang disimpan pada memory untuk mengverifikasi identitas user dan mengembalikan respon sesuai dengan status yang diberikan.

#### **2. Token Based Authentication**

Terdapat banyak aplikasi web menggunakan JSON Web Token (JWT) dibandingkan sessions untuk authentication. Pada aplikasi token based, server akan membuat JWT dengan rahasia dan mengirim JWT kepada client. Client kemudian menyimpan JWT (biasanya pada local storage) dan memasukan JWT kedalam headers untuk setiap request yang dilakukan oleh client. Server kemudian akan memvalidasi JWT pada setiap request yang dilakukan client dan mengembalikan respons yang sesuai.

Perbedaan yang sangat besar disini adalah user's state tidak disimpan di dalam server, akan tetapi disimpan didalam token pada sisi client. Sebagian besar aplikasi web modern menggunakan JWT untuk authentication dengan alasan scalability dan mobile device authentication.

### **8.3 Pengertian Authorization**

Authorization adalah proses penentuan apakah user tersebut diijinkan / ditolak untuk melakukan satu atau beberapa action atau akses terhadap resources tertentu dalam system. User login terhadap system dengan menggunakan user-ID dan password,

kemudian system mengenalinya dan user mendapatkan akses atau ditolak terhadap suatu resource system tertentu.

## 8.4 Hashing Password

Hasing password merupakan komponen sangat penting yang harus kita terapkan dalam setiap pembuatan website **“Jangan pernah menyimpan password apa adanya di dalam database”**. Hashing adalah metode enkripsi satu arah (*one way encryption*). Di sini pesan yang sudah di-enkripsi tidak bisa di kembalikan ke kondisi awal. Terdapat 6 jenis hashing yang bisa kita gunakan dalam bahasa pemrograman PHP:

### 1. sha1()

Fungsi ini akan mengembalikan nilai hash sebanyak 40 karakter

```
<?php
$password = "Private";
echo sha1($password);
```

### 2. md5()

Fungsi ini akan mengembalikan nilai hash sebanyak 32 karakter

```
<?php
$password = "Private";
echo md5($password);
```

### 3. crypt()

Fungsi ini menghasilkan kode hash dengan menggunakan algoritma DES, Blowfish, dan MD5

```
<?php
$password = "Private";
echo crypt($password, "garam");
```

### 4. password\_hash()

Fungsi ini akan menghasilkan sebuah kode hash baru dengan metode one-way hashing. one-way hashing artinya, hasil enkripsinya tidak bisa dikembalikan seperti semula.

```
<?php
$password = "Private";
echo password_hash($password, PASSWORD_DEFAULT);
```

### 5. hash()

Fungsi ini akan menciptakan sebuah kode *hash* dengan algoritma tertentu.

```
<?php
$password = "Private";
```

```
echo hash('md5', $password);
```

## 6. base64\_encode()

Fungsi ini akan menghasilkan kode hash dari teks yang diinputkan dan bisa dikembalikan ke bentuk semula dengan fungsi base64\_decode(). Metode ini disebut two-way hasing.

```
<?php
$password = "Private";
echo base64_encode($password);
echo "<br>";
echo base64_decode("UHJpdmF0ZQ==");
```

## 8.5 Membuat Register dan Login

### 1. Membuat Database

Pertama kita buat database beserta tabelnya seperti berikut:

---

```
CREATE DATABASE my_blog;
use my_blog;

CREATE TABLE user(
    id int(11) PRIMARY KEY AUTO_INCREMENT,
    name varchar(50),
    email varchar(100),
    password varchar(100)
);
```

---

### 2. Membuat Koneksi

Buat file koneksi.php untuk menghubungkan php dengan mysql

---

```
<?php
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "my_blog";

$koneksi = mysqli_connect($db_host, $db_user, $db_pass, $db_name);
```

---

### 3. Membuat Form Register

---

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Register</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOMLAsjC"
crossorigin="anonymous">
</head>
<body>
<?php
```

```

session_start();
?>
<div class="container my-5">
<div class="row">
<div class="col-md-6 mx-auto">
<?php
if (isset($_SESSION['danger'])) {
?>
<div class="alert alert-danger"><?= $_SESSION['danger'] ?></div>
<?php
unset($_SESSION['danger']);
}
?>
<div class="card">
<div class="card-header">
<h2>Register Form</h2>
<p>Already have an account? <a class="text-decoration-none"
href="login.php">Login</a></p>
</div>
<div class="card-body">
<form action="register-process.php" method="post">
<div class="mb-4">
<label for="name" class="form-label">Name</label>
<input type="text" name="name" id="name" class="form-control"
required>
</div>
<div class="mb-4">
<label for="email" class="form-label">Email</label>
<input type="email" name="email" id="email" class="form-control"
required>
</div>
<div class="mb-4">
<label for="password" class="form-label">Password</label>
<input type="password" name="password" id="password" class="form-
control" required>
</div>
<div class="mb-4">
<label for="confirm_password" class="form-label">Confirm
Password</label>
<input type="password" name="confirm_password"
id="confirm_password" class="form-control" required>
</div>
<button type="submit" class="btn btn-primary"
name="register">Register</button>
</form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>

```

---

#### 4. Membuat Register Process

```

<?php
if (isset($_POST['register'])) {
include "koneksi.php";
session_start();
$name = $_POST['name'];
$email = $_POST['email'];
$password = $_POST['password'];
$confirm_password = $_POST['confirm_password'];

```

```

// Hashing
$password = sha1($password);
$confirm_password = sha1($confirm_password);
$query = mysqli_query($koneksi, "SELECT email from user WHERE
email='$email'");
$data = mysqli_fetch_array($query);
// Validasi
if ($email == $data['email']) {
    $_SESSION['danger'] = "E-mail already used";
    header("Location:register.php");
} else {
    if ($password === $confirm_password) {
        $create = mysqli_query($koneksi, "INSERT INTO user
VALUES(null,'$name','$email','$password')");
        $_SESSION['name'] = $name;
        $_SESSION['success'] = "Congratulations " . $_SESSION['name'] . ",
your registration was successful. Please login to enter";
        header("Location:login.php");
    } else {
        $_SESSION['danger'] = "Password doesn't match";
        header("Location:register.php");
    }
}
}

```

---

## 5. Membuat Form Login

```

<!DOCTYPE html>
<html lang="en">
<head>
<title>Register</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootst
rap.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgppJLIIm9Nao0Yz1ztcQTWfSpd3yD65VohhpUuCOMLASjC"
crossorigin="anonymous">
</head>
<body>
<?php
session_start();
?>
<div class="container my-5">
<div class="row">
<div class="col-md-6 mx-auto">
<?php
if (isset($_SESSION['success'])) {
?>
<div class="alert alert-success"><?= $_SESSION['success'] ?></div>
<?php
unset($_SESSION['success']);
} else if (isset($_SESSION['danger'])) {
?>
<div class="alert alert-danger"><?= $_SESSION['danger'] ?></div>
<?php
unset($_SESSION['danger']);
}
?>
<div class="card">
<div class="card-header">
<h2>Login Form</h2>

```



```
<p>Don't have an account yet? <a class="text-decoration-none"
href="register.php">Register</a></p>
</div>
<div class="card-body">
<form action="login-process.php" method="post">
<div class="mb-4">
<label for="email" class="form-label">Email</label>
<input type="email" name="email" id="email" class="form-control"
required>
</div>
<div class="mb-4">
<label for="password" class="form-label">Password</label>
<input type="password" name="password" id="password" class="form-
control" required>
</div>
<button type="submit" class="btn btn-primary"
name="login">Login</button>
</form>
</div>
</div>
</div>
</div>
</div>
</body>
</html>
```

## 6. Membuat Login Proccess

```
<?php
if (isset($_POST['login'])) {
include "koneksi.php";
session_start();
$email = $_POST['email'];
$password = $_POST['password'];
// Hashing
$password = sha1($password);
$query = mysqli_query($koneksi, "SELECT * FROM user WHERE
email='$email' AND password='$password'");
$data = mysqli_fetch_array($query);
if (mysqli_num_rows($query) > 0) {
$_SESSION['name'] = $data['name'];
$_SESSION['success'] = "Welcome " . $_SESSION['name'] . " to the
Dashboard Page";
header("Location:index.php");
} else {
$_SESSION['danger'] = "Login failed, wrong password";
header("Location:login.php");
}
}
```

## 7. Membuat Halaman Dashboard

```
<!DOCTYPE html>
<html lang="en">
<head>
<title>Articles</title>
<link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootst
rap.min.css" rel="stylesheet" integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTWfspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">
```

```

</head>
<body>
<?php
session_start();
?>
<nav class="navbar navbar-expand-lg navbar-light bg-white">
<div class="container">
<a class="navbar-brand" href="#">My Blog</a>
<button class="navbar-toggler" type="button" data-bs-
toggle="collapse" data-bs-target="#navbarSupportedContent" aria-
controls="navbarSupportedContent" aria-expanded="false" aria-
label="Toggle navigation">
<span class="navbar-toggler-icon"></span>
</button>
<div class="collapse navbar-collapse" id="navbarSupportedContent">
<ul class="navbar-nav ms-auto mb-2 mb-lg-0">
<li class="nav-item">
<a class="nav-link active" aria-current="page"
href="#">Dashboard</a></li>
<li class="nav-item">
<a class="nav-link" href="#">Article</a>
</li>
<li class="nav-item">
<a class="nav-link" href="logout.php">Logout</a>
</li>
</ul>
</div>
</div>
</nav>
<div class="container my-5">
<div class="row">
<div class="col-md-12">
<?php
if (isset($_SESSION['success'])) {
?>
<div class="alert alert-success my-4"><?= $_SESSION['success']
?></div>
<?php
unset($_SESSION['success']);
}
?>
</div>
</div>
</div>
</body>
</html>

```

---

## 8. Membuat Logout

```

<?php
session_start();
unset($_SESSION['name']);
$_SESSION['danger'] = "Logout Successful";
header("Location:login.php");

```

---

# PERTEMUAN 9

## PHP OOP

### 9.1 Pengertian OOP

OOP merupakan singkatan dari *Object Oriented Programming*. OOP merupakan metode pemrograman yang lebih berorientasi pada objek. Objek dianggap sebagai bagian dari kode utama yang bisa berfungsi secara mandiri. Terdapat 3 konsep dasar yang melandasi pemrograman berbasis object, yakni *Encapsulation*, *Inheritance* dan *Polymorphism*

### 9.2 Class dan Object

#### 1. Class

Class dalam oop merupakan kerangka kerja atau blue print untuk object. Untuk membuat class dalam PHP caranya adalah dengan menulis keyword **class** kemudian diikuti oleh nama class nya. Seperti contoh berikut:

```
<?php
class Book
{
}
```

#### 2. Object

Object merupakan implementasi dari sebuah class. Untuk membuat object di PHP kita harus melakukan instansiasi object dengan menggunakan keyword **new**. Instansiasi object merupakan sebuah proses pembuatan object dari sebuah class.

```
<?php
class Book
{
}
$book_one = new Book();
```

### 9.3 Property dan Method

#### 1. Property

Property merupakan data-data yang terdapat di dalam class kurang lebih sama seperti variabel.

```
<?php
class Book
{
    public $name = "PHP For Beginner";
    public $price = 5.5;
    public $qty = 18;
    public $publisher = "Gramedia";
}
```

```

$book_one = new Book();
echo $book_one->$name;
echo "<br>";
echo $book_one->$price;
echo "<br>";
echo $book_one->$qty;
echo "<br>";
echo $book_one->$publisher;
echo "<br>";

```

## 2. Method

Method merupakan sebuah aksi yang terdapat dalam class

```

<?php
class Book
{
    public $name = "PHP For Beginner";
    public $price = 5.5;
    public $qty = 18;
    public $publisher = "Gramedia";

    public function orderBook()
    {
        return "Book Ordered...";
    }
}
$book_one = new Book();
$book_one->orderBook();

```

## 3. Access Modifier

Access modifier merupakan sebuah cara untuk membatasi akses ke property ataupun method dalam suatu class. Terdapat 3 access modifier dalam PHP yaitu:

### 1. Public

Apabila sebuah property ataupun method diatur menjadi public maka kode program yang berada di dalam atau di luar class dapat mengaksesnya.

### 2. Private

Apabila sebuah property ataupun method diatur menjadi private maka kode program yang bisa mengaksesnya hanya kode program yang satu class dengan nya.

### 3. Protected

Apabila sebuah property ataupun method diatur menjadi protected maka kode program yang bisa mengaksesnya hanya kode program yang merupakan turunan class nya.

## 9.4 Constructor dan \$this

### 1. \$this

\$this berfungsi untuk menyatakan object saat ini

```

<?php
class Book
{
    public $name;
    public $price;
    public $qty;
    public $publisher;

    public function orderBook()
    {
        return $this->name . " $" . $this->price;
    }
}

$book_one = new Book();
$book_one->name = "Javascript For Advanced";
$book_one->price = 10.2;
$book_one->orderBook();

```

## 2. Constructor

Constructor merupakan sebuah method yang akan dijalankan ketika sebuah object di instansiasi. Untuk membuat constructor di PHP menggunakan kata kunci **\_\_construct**.

```

<?php
class Book
{
    public $name;
    public $price;
    public $qty;
    public $publisher;

    public function __construct($name, $price, $qty, $publisher)
    {
        $this->name = $name;
        $this->price = $price;
        $this->qty = $qty;
        $this->publisher;
    }

    public function orderBook()
    {
        return "Book Ordered...";
    }
}

$book_one = new Book("OOP For Beginner", 15.5, 20, "Gramedia");
$book_one->orderBook();

```

## 9.5 Inheritance

Inheritance atau pewarisan merupakan salah satu konsep OOP untuk menurunkan property ataupun method dari sebuah class ke dalam class yang lain nya. Untuk menggunakan inheritance kita harus menggunakan kata kunci **extends**

```

<?php
class Book
{
    public $name = "PHP For Beginner";
}

```

```

        public $qty = 19;

        public function stockCheck()
        {
            return "Remaining Stok: " . $this->qty;
        }
    }

    class Category extends Book
    {
        public $name_category = "Programming";

        public function stockCheckCategory()
        {
            return $this->name_category . " " . $this->name . ", " .
$this->stockCheck();
        }
    }

    $book_one = new Category();
    echo $book_one->name;
    echo "<br>";
    echo $book_one->stockCheckCategory();

```

## 9.6 Getter dan Setter

Getter dan setter merupakan method untuk melakukan mengisi (set) dan mengambil nilai (get).

```

<?php
class Book
{
    private $name;
    private $qty;

    public function setName($name)
    {
        $this->name = $name;
    }

    public function getName()
    {
        return $this->name;
    }

    public function setQty($qty)
    {
        $this->qty = $qty;
    }

    public function getQty()
    {
        return $this->qty;
    }
}

$book_one = new Book();
$book_one->setName("PHP OOP For Advanced");
$book_one->setQty(19);

echo $book_one->getName();
echo "<br>";

```

```
echo $book_one->getQty();
```

## 9.7 Tugas

1. Buatlah class Book yang memiliki 3 property, yaitu \$code\_book, \$name, \$qty yang di set sebagai private. Kemudian buatlah method setter dan getter untuk kedua property \$code\_book dan \$qty dengan aturan berikut:
  - \$code\_book dalam format “BB00”, dimana 2 karakter awal berupa alphabet dalam huruf besar, kemudian diikuti oleh 2 karakter angka 0-9. Jika format yang diinput tidak sesuai, tampilkan pesan error
  - \$qty hanya bisa menerima angka integer dan harus positif. Jika diinput integer negatif atau 0 tampilkan pesan error
  - Setter di set sebagai private dan dijalankan dari constructor
  - Getter tidak perlu pemrosesan cukup kembalikan nilai saja

# PERTEMUAN 10

## PHP Restful API

### 10.1 Pengertian RestFul API

RESTful API / REST API merupakan implementasi dari API (Application Programming Interface). REST (Representational State Transfer) adalah suatu arsitektur metode komunikasi yang menggunakan protokol HTTP untuk pertukaran data dan metode ini sering diterapkan dalam pengembangan aplikasi. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan (scale) terutama dalam pertukaran dan komunikasi data.

Rest API atau sering juga disebut dengan istilah API saja sering digunakan sebagai penghubung antara front-end dan back-end aplikasi. Front-end sendiri aplikasi interface yang digunakan user atau pengguna aplikasi. Contoh dari front-end aplikasi yaitu aplikasi android dan iOS. Back-end aplikasi yang digunakan administrator untuk manajemen data. Sehingga dengan Rest API data dari front-end ke back-end ataupun sebaliknya dapat diolah seperti menampilkan, menambahkan, mengubah dan menghapus data.

### 10.2 URL Design

RESTful API dapat diakses dengan menggunakan protokol HTTP. Penamaan dan struktur URL yang konsisten pasti akan menghasilkan API yang baik dan mudah dimengerti oleh developer. URL API biasa disebut endpoint dalam pemanggilannya. Contoh penamaan URL / endpoint yang baik adalah seperti berikut:

/books

/books/1234

/books/1234/covers

/books/1234/covers/xyz

### 10.3 HTTP Verbs

Setiap request yang dilakukan terdapat method yang dipakai supaya server dapat mengerti apa yang sedang di request oleh client, diantaranya yang umum dipakai adalah:

|     |  |
|-----|--|
| GET | GET adalah metode HTTP Request yang paling simpel, metode ini digunakan untuk membaca atau mendapatkan data dari sumber. |
|-----|--|



|        |   |
|--------|---|
| POST   | POST adalah metode HTTP Request yang digunakan untuk membuat data baru dengan menyisipkan data dalam body saat request dilakukan. |
| PUT    | PUT adalah metode HTTP Request yang biasanya digunakan untuk melakukan update data resource.                                      |
| DELETE | DELETE adalah metode HTTP Request yang digunakan untuk menghapus suatu data pada resource.  |

#### 10.4 Response Code

HTTP response code adalah kode standarisasi dalam menginformasikan hasil request kepada client. Berikut merupakan response code yang bisa digunakan pada REST:

|                           |  |
|---------------------------|--|
| 200 OK                    | Response code ini menandakan bahwa request yang dilakukan berhasil.  |
| 201 Created               | Response code ini menandakan bahwa request yang dilakukan berhasil dan data telah dibuat. Kode ini digunakan untuk mengkonfirmasi berhasilnya request PUT atau POST. |
| 400 Bad Request           | Response code ini menandakan bahwa request yang dibuat salah atau data yang dikirim tidak ada.   |
| 401 Unauthorized          | Response code ini menandakan bahwa request yang dibuat membutuhkan authentication sebelum mengakses resource.  |
| 404 Not Found             | Response Code ini menandakan bahwa resource yang di dipanggil tidak ditemukan.   |
| 405 Method Not Allowed    | Response code ini menandakan bahwa request endpoint ada tetapi metode HTTP yang digunakan tidak diizinkan.   |
| 409 Conflict              | Response code ini menandakan bahwa request yang dibuat terdapat duplikasi, biasanya informasi yang dikirim sudah ada sebelumnya.                                     |
| 500 Internal Server Error | Response code ini menandakan bahwa request yang dilakukan terdapat kesalahan pada sisi server atau resource.   |

## 10.5 Format Response

Setiap request yang dilakukan client akan menerima data response dari server, response tersebut biasanya berupa data JSON ataupun XML. Setelah mendapatkan data response tersebut barulah client bisa menggunakannya dengan cara memarsing data tersebut dan diolah sesuai kebutuhan.

## 10.6 Studi Kasus Restful API

### 1. Buat database

Buat database dan struktur tabel seperti berikut:

---

```
CREATE DATABASE book_store;

USE book_store;

CREATE TABLE books (
    id int(11) PRIMARY KEY AUTO_INCREMENT,
    name varchar(50),
    price double(5,5) DEFAULT 0,
    qty int(3) DEFAULT 0,
    author varchar(50),
    publisher varchar(50)
);
```

---

### 2. Input Data

Tambahkan beberapa data ke dalam database yang sudah kita buat sebelumnya

---

```
INSERT INTO books VALUES
(NULL, "PHP For Beginner", 15.9, 25, "Rasmus Lerdorf", "Gramedia"),
(NULL, "Javascript For Beginner", 12.9, 40, "Brendan Eich", "Gramedia");
```

---

### 3. Buat koneksi

Buat folder untuk tempat project dengan nama book\_store kemudian membuat file koneksi.php untuk menghubungkan dengan database kita.

---

```
<?php
$db_host = "localhost";
$db_user = "root";
$db_pass = "";
$db_name = "book_store";

$koneksi = new mysqli($db_host, $db_user, $db_pass, $db_name);
```

---

### 4. Buat file .htaccess

File .htaccess berfungsi untuk mempercantik tampilan url kita ketika nanti diakses melalui postman

---

```
RewriteEngine On # Turn on the rewriting engine
RewriteRule ^book/?$ book.php [NC,L]
```

---

---

```
RewriteRule ^book/([0-9]+)/?$ book.php?id=$1 [NC,L]
```

---

## 5. Buat method

Buat sebuah file method dengan nama file method.php yang bertujuan untuk mengakses database dan return data dalam bentuk JSON.

---

```
<?php
require_once "koneksi.php";
class Book
{
    public function get_books()
    {
        global $koneksi;
        $query = "SELECT * FROM books";
        $data = array();
        $result = $koneksi->query($query);
        while ($row = mysqli_fetch_object($result)) {
            $data[] = $row;
        }
        $response = array(
            'status' => 1,
            'message' => 'Get List Book Successfully.',
            'data' => $data
        );
        header('Content-Type: application/json');
        echo json_encode($response);
    }

    public function get_book($id = 0){
        global $koneksi;
        $query = "SELECT * FROM books";
        if ($id != 0) {
            $query .= " WHERE id=" . $id . " LIMIT 1";
        }
        $data = array();
        $result = $koneksi->query($query);
        while ($row = mysqli_fetch_object($result)) {
            $data[] = $row;
        }
        $response = array(
            'status' => 1,
            'message' => 'Get Book Successfully.',
            'data' => $data
        );
        header('Content-Type: application/json');
        echo json_encode($response);
    }

    public function insert_book(){
        global $koneksi;
        $arrcheckpost = array('name' => '', 'price' => '', 'qty' => '',
            'author' => '', 'publisher' => '');
        $hitung = count(array_intersect_key($_POST, $arrcheckpost));
        if ($hitung == count($arrcheckpost)) {
            $result = mysqli_query($koneksi, "INSERT INTO books SET
            name = '$_POST[name]',
            price = '$_POST[price]',
            qty = '$_POST[qty]',
            author = '$_POST[author]',
            publisher = '$_POST[publisher]'");
            if ($result) {
                $response = array(
                    'status' => 1,
```

```

'message' => 'Book Added Successfully.'
);
} else {
$response = array(
'status' => 0,
'message' => 'Book Addition Failed.'
);
}
} else {
$response = array(
'status' => 0,
'message' => 'Parameter Do Not Match'
);
}
header('Content-Type: application/json');
echo json_encode($response);
}
function update_book($id)
{
global $koneksi;
$arrcheckpost = array('name' => '', 'price' => '', 'qty' => '',
'author' => '', 'publisher' => '');
$hitung = count(array_intersect_key($_POST, $arrcheckpost));
if ($hitung == count($arrcheckpost)) {
$result = mysqli_query($koneksi, "UPDATE books SET
name = '$_POST[name]',
price = '$_POST[price]',
qty = '$_POST[qty]',
author = '$_POST[author]',
publisher = '$_POST[publisher]'
WHERE id='$id'");
if ($result) {
$response = array(
'status' => 1,
'message' => 'Book Updated Successfully.'
);
} else {
$response = array(
'status' => 0,
'message' => 'Book Updation Failed.'
);
}
} else {
$response = array(
'status' => 0,
'message' => 'Parameter Do Not Match'
);
}
header('Content-Type: application/json');
echo json_encode($response);
}
function delete_book($id) {
global $koneksi;
$query = "DELETE FROM books WHERE id=" . $id;
if (mysqli_query($koneksi, $query)) {
$response = array(
'status' => 1,
'message' => 'Book Deleted Successfully.'
);
} else {
$response = array(
'status' => 0,
'message' => 'Book Deletion Failed.'
);
}
}

```

```
}  
header('Content-Type: application/json');  
echo json_encode($response);  
}  
}
```

---

## 6. Request method

Buat file book.php yang berfungsi untuk pengecekan request method yang dilakukan dan menjalankan nya pada file method.php

```
<?php  
require_once "method.php";  
$obj_book = new Book();  
$request_method = $_SERVER["REQUEST_METHOD"];  
switch ($request_method) {  
case 'GET':  
if (!empty($_GET["id"])) {  
$id = intval($_GET["id"]);  
$obj_book->get_book($id);  
} else {  
$obj_book->get_books();  
}  
break;  
case 'POST':  
if (!empty($_GET["id"])) {  
$id = intval($_GET["id"]);  
$obj_book->update_book($id);  
} else {  
$obj_book->insert_book();  
}  
break;  
case 'DELETE':  
$id = intval($_GET["id"]);  
$obj_book->delete_book($id);  
break;  
default:  
header("HTTP/1.0 405 Method Not Allowed");  
break;  
}
```

---

## 7. Test API dengan postman

Terakhir saatnya untuk menguji nya melalui aplikasi postman seperti berikut

Get All Books

localhost/book\_store/book

GET localhost/book\_st... Bookshelf API Test

localhost/book\_store/book

GET localhost/book\_store/book Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results Status: 200 OK Time: 34 ms Size: 538 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "status": 1,
3    "message": "Get List Book Successfully.",
4    "data": [
5      {
6        "id": "1",
7        "name": "PHP For Beginner",
8        "price": "15.90",
9        "qty": "25",
10       "author": "Rasmus Lerdorf",
11       "publisher": "Gamedia"
12     },
13     {
14       "id": "2",
15       "name": "Javascript For Beginner",
16       "price": "12.90",
17       "qty": "40",
```

## Get Book

localhost/book\_store/book/2

GET localhost/book\_st... Bookshelf API Test

localhost/book\_store/book/2

GET localhost/book\_store/book/2 Send

Params Authorization Headers (6) Body Pre-request Script Tests Settings Cookies

Body Cookies Headers (7) Test Results Status: 200 OK Time: 49 ms Size: 420 B Save Response

Pretty Raw Preview Visualize JSON

```
1  {
2    "status": 1,
3    "message": "Get Book Successfully.",
4    "data": [
5      {
6        "id": "2",
7        "name": "Javascript For Beginner",
8        "price": "12.90",
9        "qty": "40",
10       "author": "Brendan Eich",
11       "publisher": "Gamedia"
12     }
13   ]
14 }
```

## Insert Book

POST localhost/book\_s... Bookshelf API Test

localhost/book\_store/book/ Save

POST localhost/book\_store/book/ Send

Params Authorization Headers (8) Body Pre-request Script Tests Settings Cookies

none form-data x-www-form-urlencoded raw binary GraphQL

|                                     |           |           |             |
|-------------------------------------|-----------|-----------|-------------|
| <input checked="" type="checkbox"/> | qty       | 5         |             |
| <input checked="" type="checkbox"/> | author    | Mark Otto |             |
| <input checked="" type="checkbox"/> | publisher | Gramedia  |             |
|                                     | Key       | Value     | Description |

Body Cookies Headers (7) Test Results Status: 200 OK Time: 2.28 s Size: 294 B Save Response

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": 1,
3   "message": "Book Added Successfully."
4 }
```

## Update Book

localhost/book\_store/book/3

POST localhost/book\_s... Bookshelf API

localhost/book\_store/book/3 Save

POST localhost/book\_store/book/3

Params Authorization Headers (8) Body Pre-request Script Tests Settings

none form-data x-www-form-urlencoded raw binary GraphQL

|                                     | KEY   | VALUE                   | DESCRIPTION |
|-------------------------------------|-------|-------------------------|-------------|
| <input checked="" type="checkbox"/> | name  | Bootstrap From Advanced |             |
| <input checked="" type="checkbox"/> | price | 9.5                     |             |
| <input checked="" type="checkbox"/> | qty   | 5                       |             |

Body Cookies Headers (7) Test Results Status: 200 OK Time: 81 ms Size: 296 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": 1,
3   "message": "Book Updated Successfully."
4 }
```

## Delete Book

localhost/book\_store/book/2

DEL localhost/book\_st... + ... Bookshelf API

localhost/book\_store/book/2 Save

DELETE localhost/book\_store/book/2

Params Authorization Headers (8) **Body** Pre-request Script Tests Settings

none form-data **x-www-form-urlencoded** raw binary GraphQL

|                                     | KEY   | VALUE                   | DESCRIPTION |
|-------------------------------------|-------|-------------------------|-------------|
| <input checked="" type="checkbox"/> | name  | Bootstrap From Advanced |             |
| <input checked="" type="checkbox"/> | price | 9.5                     |             |
| <input checked="" type="checkbox"/> | qty   | 5                       |             |

Body Cookies Headers (7) Test Results Status: 200 OK Time: 124 ms Size: 296 B

Pretty Raw Preview Visualize JSON

```
1 {
2   "status": 1,
3   "message": "Book Deleted Successfully."
4 }
```

## 10.7 Tugas

1. Buatlah RestFul API dengan contoh studi kasus yang lain kemudian test dengan menggunakan postman!