

VERSION 1.2
SEPTEMBER 16, 2024



PEMROGRAMAN WEBSITE

HTML, CSS, DESAIN WEB – MODUL 1

DISUSUN OLEH:

OGYA ADYATMA PUTRA
GERALDI NATHAN TOMMY SAPUTRA

DIAUDIT OLEH:

AMINUDIN, S.KOM., M.CS

LAB. INFORMATIKA

UNIVERSITAS MUHAMMADIYAH MALANG

PEMROGRAMAN WEBSITE

PERSIAPAN MATERI

- <https://www.w3schools.com/html/>
- <https://www.w3schools.com/w3css/>

TUJUAN

Jelaskan materi yang harus dipelajari oleh praktikan sebelum mengerjakan tugas

TARGET MODUL

1. Mahasiswa mampu membuat desain website
2. Mahasiswa mampu menjalankan kode HTML, dan CSS
3. Mahasiswa mampu mengimplementasi desain menjadi kode

PERSIAPAN SOFTWARE/APLIKASI

Hardware

- Laptop / PC
- Koneksi Internet

Software

- Text editor (Visual Studio Code, Sublime, Atom, Notepad++, dll)
- Figma

Tambahan (Extension Visual Studio Code)

- Live Server by Ritwick dey
- HTML CSS Support by Ecmel
- Prettier – Code Formatter by Prettier

MATERI

1. HTML (Hypertext Mark Up Language)

HTML adalah singkatan dari *Hypertext Markup Language*, HTML merupakan salah satu format pengkodean atau pemrograman yang digunakan untuk membuat halaman website yang ditampilkan pada web browser. Sebagian besar halaman yang kita temukan pada internet kebanyakan menggunakan Bahasa HTML.

Kelebihan dari HTML diantara lain sebagai Berikut:

1. Memiliki banyak sumber dan komunitas yang sangat besar.
2. Bersifat Open Source sehingga gratis untuk digunakan
3. Memiliki Bahasa Markup yang konsisten dan rapi
4. Mudah diintegrasikan dengan Bahasa backend seperti PHP , Node.js dan javascript
5. Bisa dijalankan oleh semua Browser.
6. Dapat dipelajari dengan mudah oleh web developer pemula.

Sedangkan kekurangan dari HTML adalah sebagai Berikut:

1. HTML hanya dapat diimplementasikan untuk halaman web static . jika ingin menambah fitur dinamis, pengguna dapat menggunakan bahasa pemrograman back-end seperti javascript atau Bahasa pemrograman back-end lainnya.
2. Bahasa pemrograman HTML tidak mendukung perintah logic sehingga semua halaman website harus dibuat secara terpisah. Dengan alasan ini, HTML biasanya dikategorikan sebagai bukan Bahasa pemrograman.

Seperti kebanyakan Bahasa lainnya, HTML juga memiliki *syntax* khusus dalam penulisannya. Berikut ini merupakan contoh *syntax* yang biasanya sering digunakan beserta fungsinya.

Syntax HTML	Penjelasan Fungsi
<html>	Menandai awal dan akhir dari sebuah dokumen HTML.
<head>	Berisi informasi meta tentang halaman, seperti judul, deskripsi, dan tautan ke file CSS.
<title>	Menentukan judul halaman yang akan ditampilkan pada tab browser.
<body>	Mengandung konten utama yang akan ditampilkan di halaman web.
<h1> hingga <h6>	Mendefinisikan judul dengan berbagai tingkat variasi ukuran.
<p>	Mendefinisikan sebuah paragraf.
<a>	Membuat tautan (link) ke halaman lain atau url lain.
	Menambahkan gambar ke dalam halaman.
 dan 	Membuat daftar berurutan. untuk daftar, untuk item dalam daftar.
<table>	Membuat tabel.
<div>	Digunakan untuk menandai bagian (division) atau wadah dari sebuah struktur html.

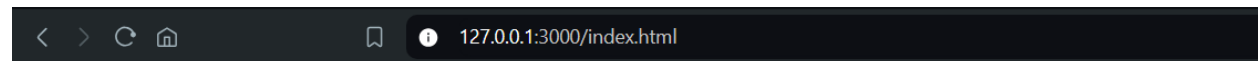
Jika di dalam Bahasa pemrograman java kita sering menemui file utama berupa main.java seperti dibawah ini:

```
class Program {  
    public static void main(String args[]){  
        System.out.println("Hello World");  
    }  
}
```

Maka untuk HTML kita juga memiliki file main, berupa **index.html** dan biasanya memiliki basic syntax seperti dibawah ini:

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Document</title>  
</head>  
<body>  
    <h1>Hello World</h1>  
</body>  
</html>
```

Berikut merupakan hasil output jika program html di atas dijalankan.



Hello World

Berikut ini merupakan contoh penerapan dari tabel syntax-syntax html diatas. Anda bisa copy dan coba pada IDE yang Anda gunakan, kemudian gunakan extension live server (khusus Visual Studio Code) untuk menampilkan kode dibawah ini.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Document</title>
</head>
<body>
  <h1>Tentang Saya</h1>
  
  <p>Halo, nama saya lorem ipsum. Saya seorang dolor sit amet.</p>
  <h2>Keahlian</h2>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>JavaScript</li>
  </ul>
  <h2>Kontak</h2>
  <p>Anda bisa menghubungi saya melalui:</p>
  <ul>
    <li><a href="mailto:alamatemail@contoh.com">Email</a></li>
    <li><a href="https://www.linkedin.com/in/nama-profil-
anda">LinkedIn</a></li>
  </ul>
</body>
</html>
```

2. CSS (Cascading Style Sheets)

CSS merupakan singkatan dari *Cascading Style Sheets*, CSS digunakan untuk menentukan gaya dari tampilan website yang akan kita buat seperti tata letak halaman, warna , dan font. Semuanya dapat ditentukan oleh CSS. CSS bertujuan agar website terlihat lebih menarik.

Pada dasarnya tanpa menggunakan CSS kita tetap dapat membuat website dengan hanya mengandalkan HTML. Namun, hal tersebut kurang efektif jika digunakan untuk website yang lebih mengedepankan tampilan. Oleh karena itu, biasanya CSS dan HTML tidak dapat dipisahkan satu sama lain.

Berikut ini merupakan contoh syntax yang ada pada CSS serta pengertian fungsinya:

Type Syntax	Penjelasan	Contoh Penggunaan
Declaration	Menentukan properti dan nilainya untuk mengubah tampilan elemen.	color: blue; (Mengatur warna teks menjadi biru)
Property	Atribut visual dari suatu elemen, seperti warna, ukuran, font, dll.	font-size, background-color, margin, padding
Value	Nilai yang diberikan pada suatu properti.	16px, #f00, auto, inherit
Font	Menentukan jenis, ukuran, warna, dan gaya font.	font-family: Arial, sans-serif; font-size: 16px;
Color	Menentukan warna elemen.	color: blue; background-color: #f0f0f0;
Display	Mengatur tipe tampilan elemen (block, inline, inline-block, none, flex, grid, dll).	display: none; (Menyembunyikan elemen)
Margin	Ruang di luar border elemen.	margin: 10px;
Padding	Ruang di dalam border elemen.	padding: 15px;
Border	Garis di sekitar elemen.	border: 1px solid black;
Height	Mengatur tinggi elemen.	height: 200px;
Width	Mengatur lebar elemen.	width: 300px;
Text-align	Menentukan posisi teks di dalam elemen.	text-align: center;

Agar sebuah halaman website dapat memiliki tampilan yang menarik, kita perlu menghubungkan CSS dengan struktur HTML. Terdapat berbagai cara untuk memanggil CSS ke dalam HTML. Beberapa metode ini tentunya memiliki kelebihan dan kekurangannya masing masing.

1. CSS Inline

CSS ditulis langsung di dalam elemen HTML menggunakan atribut style. Metode ini sangat cocok untuk penyesuaian tampilan yang sangat spesifik pada elemen tertentu. Namun, kurang efisien jika ingin menerapkan tampilan yang sama pada banyak elemen atau jika tampilan tersebut sering berubah.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Inline</title>
</head>
<body>
  <h1 style="color: red;">Hello World!</h1>
</body>
</html>
```

2. CSS Internal

CSS ditulis di dalam tag `<style>` yang berada di dalam bagian tag `<html>`. Metode ini memungkinkan kita untuk mengatur tampilan untuk seluruh halaman dalam satu tempat. Meskipun lebih baik daripada CSS inline, metode ini masih kurang efisien untuk proyek yang besar dan kompleks.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>CSS Internal</title>

  <style>
    h1{
      color: red;
    }
  </style>

</head>
<body>
  <h1>Hello World!</h1>
</body>
</html>
```

3. CSS Eksternal

CSS ditulis dalam file terpisah (biasanya dengan ekstensi `style.css`) dan kemudian dihubungkan dengan dokumen HTML menggunakan tag `<link>`. Metode inilah yang paling umum digunakan dan disarankan, karena:

- Memudahkan dalam pengelolaan kode dan kolaborasi tim.
- Gaya hanya perlu didefinisikan sekali dan dapat digunakan di banyak halaman.
- Perubahan gaya dapat dilakukan dengan mudah hanya dengan mengubah file CSS.

File: **index.html**

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <link rel="stylesheet" href="style.css">
  <title>CSS Eksternal</title>
</head>
<body>
  <h1>Hello World!</h1>
</body>
```

```
</html>
```

File: **style.css**

```
h1{
  color: red;
}
```

Ketiga cara diatas akan menghasilkan tampilan yang sama, berupa Heading 1 dengan bertuliskan "Hello World" berwarna merah. Metode yang tepat untuk memanggil CSS tergantung pada kebutuhan proyek kita. Untuk proyek yang sederhana dan tidak terlalu banyak perubahan gaya tampilan, CSS internal mungkin sudah cukup. Namun, untuk proyek yang lebih besar dan kompleks, CSS eksternal adalah pilihan yang paling baik.

Dibawah ini merupakan contoh HTML pada materi 1 namun, kita menambahkan CSS untuk mempercantik tampilan. Anda bisa copy kode dibawah ini, kemudian jalankan seperti HTML sebelumnya.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Modul 1</title>

  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 0;
      padding: 20px;
    }
    h1 {
      text-align: center;
      color: #333;
    }
    img {
      display: block;
      margin: 0 auto;
      width: 200px;
      border-radius: 50%;
    }
    ul {
      list-style: none;
      padding: 0;
    }
    a {
```



```

        color: blue;
        text-decoration: none;
    }
</style>
</head>
<body>
    <h1>Tentang Saya</h1>
    
    <p>Halo, nama saya lorem ipsum. Saya seorang dolor sit amet.</p>
    <h2>Keahlian</h2>
    <ul>
        <li>HTML</li>
        <li>CSS</li>
        <li>JavaScript</li>
    </ul>
    <h2>Kontak</h2>
    <p>Anda bisa menghubungi saya melalui:</p>
    <ul>
        <li><a href="mailto:alamatemail@contoh.com">Email</a></li>
        <li><a href="https://www.linkedin.com/in/nama-profil-
anda">LinkedIn</a></li>
    </ul>
</body>
</html>

```

3. HTML Framework (Bootstrap)

Bootstrap adalah sebuah kerangka kerja (framework) CSS dan javascript. Fungsinya adalah untuk mempercepat proses pengembangan website atau aplikasi web. Dengan menggunakan Bootstrap, kita bisa membuat tampilan website yang responsif, modern, dan konsisten tanpa perlu menulis kode CSS dari awal.

Seperti hanya HTML sebelumnya, bootstrap juga memiliki starter code untuk menuliskan kode CSS menggunakan bootstrap. Berikut merupakan contoh awal penggunaan Bootstrap.

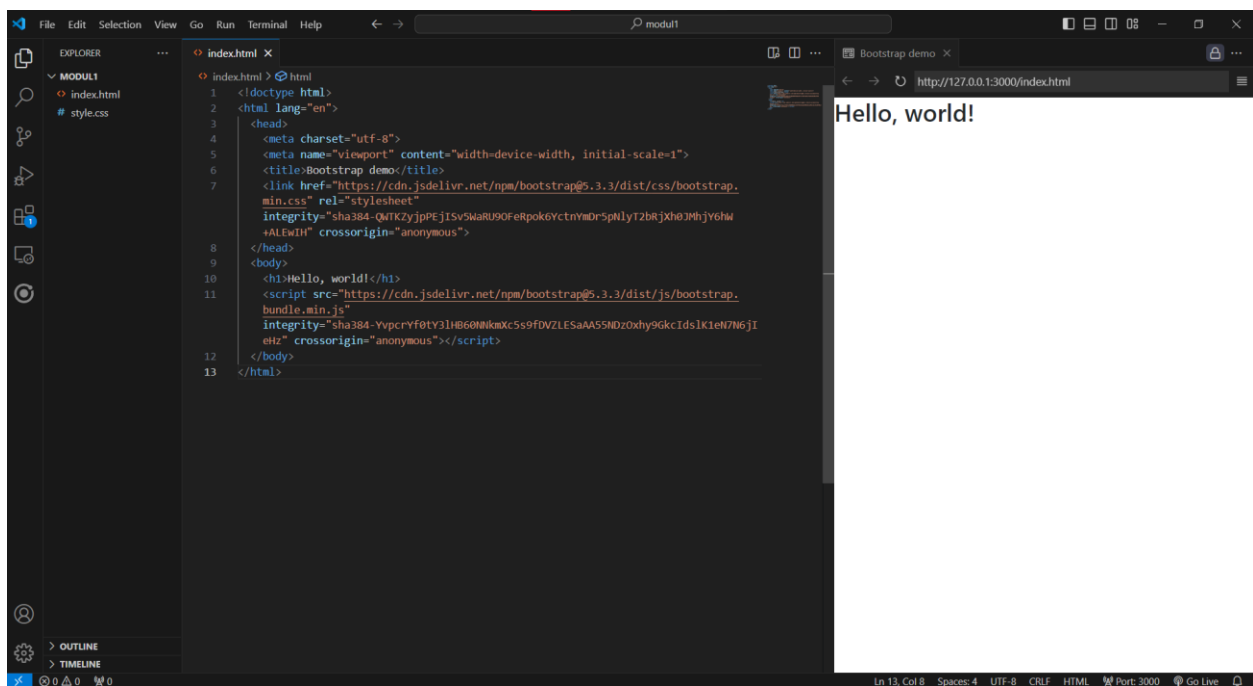
```

<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta name="viewport" content="width=device-width, initial-scale=1">
        <title>Bootstrap demo</title>
        <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-

```

```
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhjY6hW+ALEwIH"
crossorigin="anonymous">
</head>
<body>
  <h1>Hello, world!</h1>
  <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js
" integrity="sha384-
YvpcrYf0tY3lHB60NNkMc5s9fDVZLEsAA55NDz0xhy9GkcIdslK1eN7N6jIeHz"
crossorigin="anonymous"></script>
</body>
</html>
```

Dengan kode diatas, maka akan menampilkan output seperti dibawah ini.



Coba bandingkan html biasa pada materi 1 dan Bootstrap apakah memiliki perbedaan.

Sekarang, mari kita coba untuk melakukan import salah satu Components yang ada pada Bootstrap.

```
<nav class="navbar navbar-expand-lg bg-body-tertiary">
  <div class="container-fluid">
    <a class="navbar-brand" href="#">Navbar</a>
    <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-
expanded="false" aria-label="Toggle navigation">
      <span class="navbar-toggler-icon"></span>
    </button>
    <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
```

```

    <div class="navbar-nav">
      <a class="nav-link active" aria-current="page" href="#">Home</a>
      <a class="nav-link" href="#">Features</a>
      <a class="nav-link" href="#">Pricing</a>
      <a class="nav-link disabled" aria-disabled="true">Disabled</a>
    </div>
  </div>
</div>
</nav>

```

Kode diatas berasal dari website Bootstrap dan sudah disediakan untuk membuat sebuah Navbar.

Selanjutnya, mari kita coba untuk gabungan dengan starter template Bootstrap tadi.

```

<!doctype html>
<html lang="en">
  <head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Bootstrap demo</title>
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/css/bootstrap.min.css"
rel="stylesheet" integrity="sha384-
QWTKZyjpPEjISv5WaRU90FeRpok6YctnYmDr5pNlyT2bRjXh0JMhY6hW+ALEwIH"
crossorigin="anonymous">
  </head>
  <body>
    <nav class="navbar navbar-expand-lg bg-body-tertiary">
      <div class="container-fluid">
        <a class="navbar-brand" href="#">Navbar</a>
        <button class="navbar-toggler" type="button" data-bs-toggle="collapse"
data-bs-target="#navbarNavAltMarkup" aria-controls="navbarNavAltMarkup" aria-
expanded="false" aria-label="Toggle navigation">
          <span class="navbar-toggler-icon"></span>
        </button>
        <div class="collapse navbar-collapse" id="navbarNavAltMarkup">
          <div class="navbar-nav">
            <a class="nav-link active" aria-current="page" href="#">Home</a>
            <a class="nav-link" href="#">Features</a>
            <a class="nav-link" href="#">Pricing</a>
            <a class="nav-link disabled" aria-disabled="true">Disabled</a>
          </div>
        </div>
      </div>
    </nav>

    <h1>Hello, world!</h1>

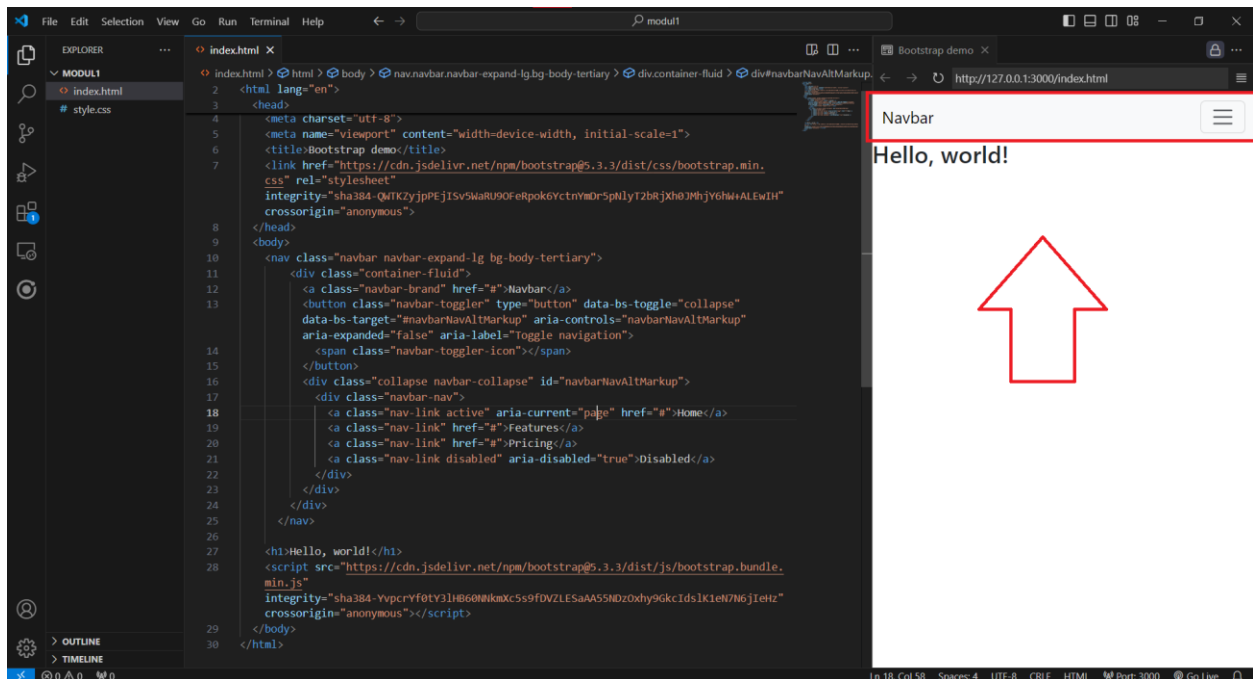
```

```

<script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.3.3/dist/js/bootstrap.bundle.min.js"
integrity="sha384-YvpcrYf0tY3lHB60NNkmXc5s9fDVZLESaAA55NDz0Xhy9GkcIdslK1eN7N6jIeHhZ"
crossorigin="anonymous"></script>
</body>
</html>

```

Dari kode diatas, maka akan menjadi seperti dibawah ini.



Navbar tersebut berasal dari kode Bootstrap yang telah kita implementasi pada starter template sebelumnya.

Diatas merupakan contoh penggunaan sederhana untuk Bootstrap. Dikarenakan materi tentang Bootstrap bisa sangat panjang, maka praktikan diharapkan dapat belajar melalui referensi dibawah ini secara mandiri.

- <https://getbootstrap.com/>
- <https://getbootstrap.com/docs/5.3/getting-started/introduction/>

4. Desain UI/UX

Desain User Interface (UI) adalah proses yang digunakan desainer untuk membuat tampilan dalam perangkat lunak atau perangkat komputer, dengan fokus pada tampilan atau gaya. Sedangkan desain User Experience (UX) adalah proses untuk menciptakan pengalaman yang lancar dan intuitif pada saat user menggunakan website maupun aplikasi. Tujuan UX design adalah menciptakan produk atau layanan yang mampu memenuhi kebutuhan dan mengatasi masalah pengguna, serta menciptakan pengalaman yang menyenangkan secara keseluruhan.

Berikut ini merupakan contoh penerapan before – after UI dan UX Design:

Before

Totals:	
Products	\$493.69
Accessories	\$698.69
Sales Tax	\$69.36
Local Tax	\$854.35
Shipping fee	\$98.36
Handling Fees	\$68.36
Shipping Tax	\$968.34
Fork Delivery	\$589.57
Extra Labor	\$54.58
Total	\$4,256.98

After

Product:	
Products	\$493.69
Accessories	\$698.58
Sales Tax	\$69.36
Local Tax	\$854.35
Product Total	\$2,256.68
Shipping:	
Shipping fee	\$98.36
Handling Fees	\$68.59
Shipping Tax	\$968.34
Fork Delivery	\$189.57
Extra Labor	\$54.58
Shipping Total	\$1,698.66
Grand Total:	
\$4,256.98	

Before**System Settings:**

- ☒ Auto pay enabled
- ☒ Lock account
- ☒ Allow checkout
- ☒ Subscribe to emails

After**System Settings:**

☒ **Auto Pay**

We will automatically charge the card on file.

☒ **Account Lock**

Prevent users from doing any activity.

☒ **Order Checkout**

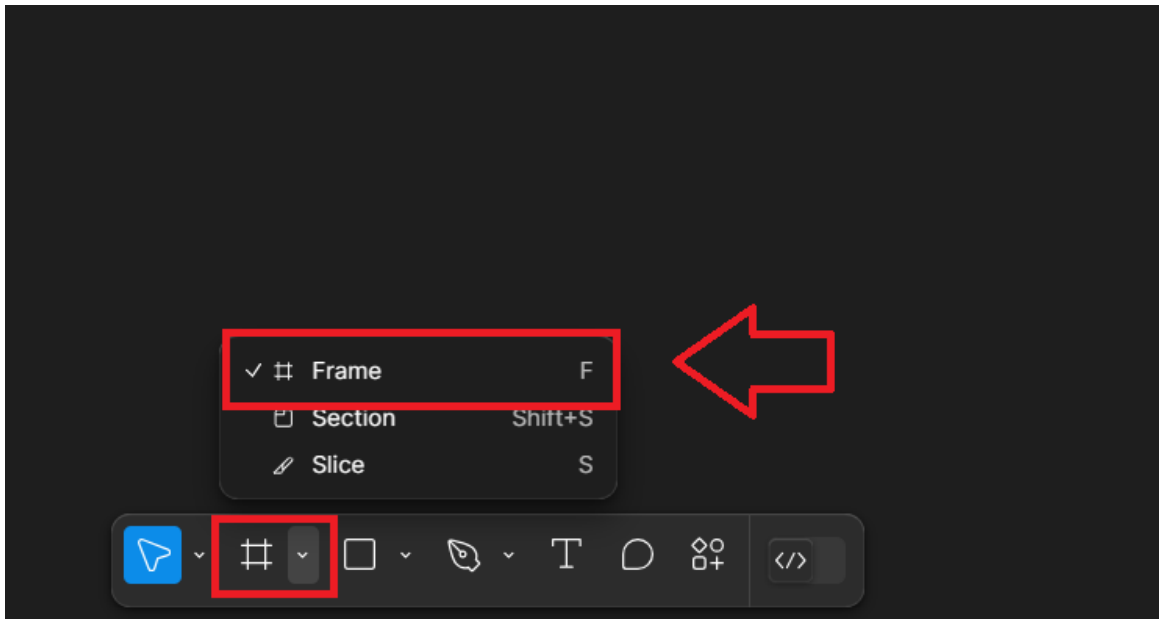
The user will be able to checkout their cart.

☒ **Subscribe to emails**

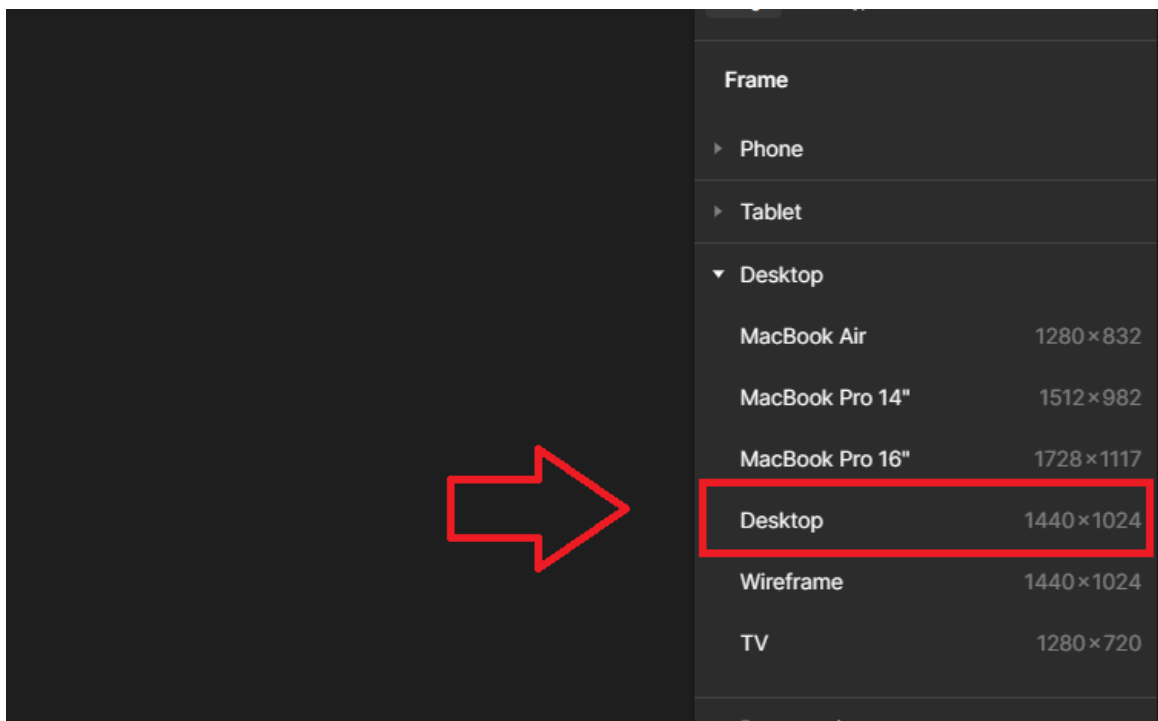
Perbedaan UI dan UX	
UI	UX
Tampilan dan fungsi produk itu sendiri	Kesan yang dirasakan pengguna terhadap suatu produk
Berfokus pada elemen visual produk, seperti warna, layout, animasi, tombol, dan tipografi	Berfokus pada aspek emosional pengguna
Bertujuan untuk menciptakan produk yang menarik secara estetika	Bertujuan untuk menciptakan pengalaman pengguna yang lancar dan tanpa masalah

5. Figma Dasar

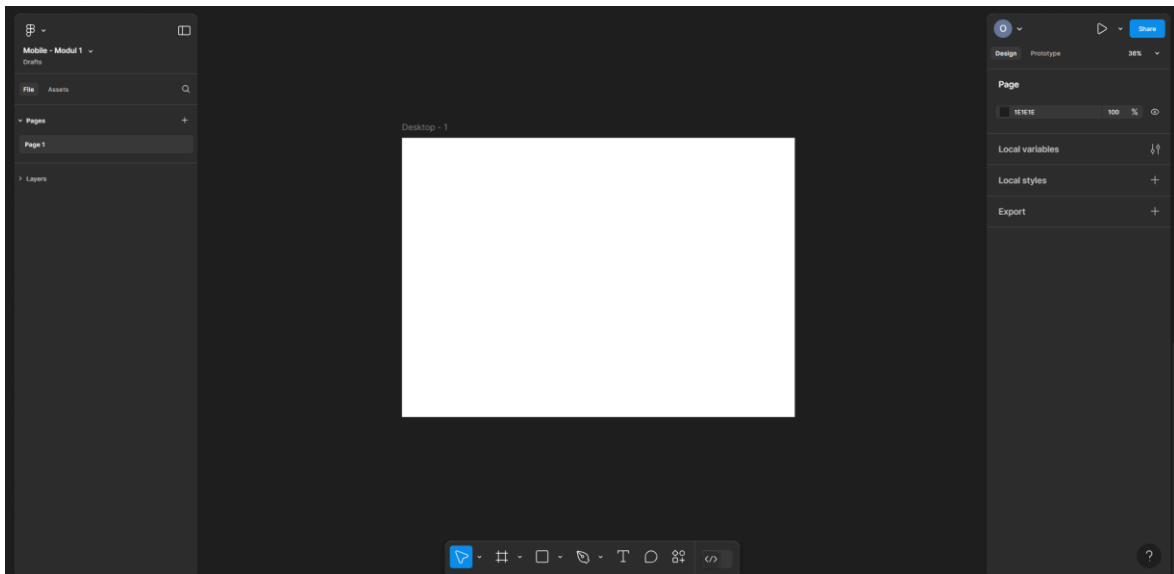
- Membuat Frame Dalam Figma



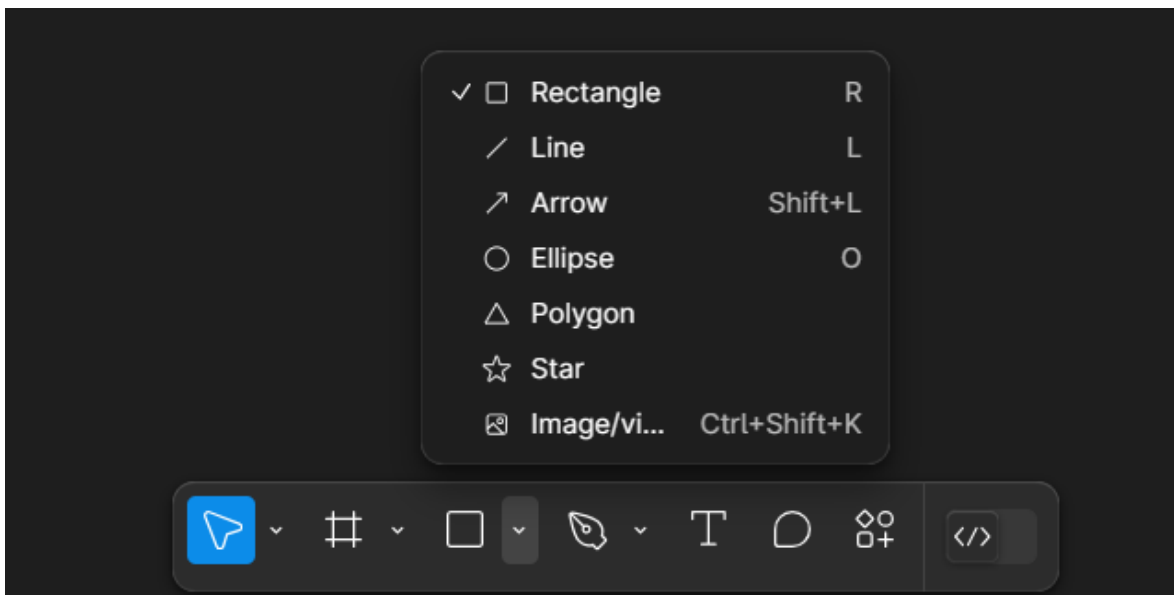
- Menentukan Frame sesuai dengan kebutuhan
Dalam studi kasus kali ini, kita akan membuat Frame untuk kebutuhan desain website. Maka dari itu, kita bisa memilih opsi "Desktop".



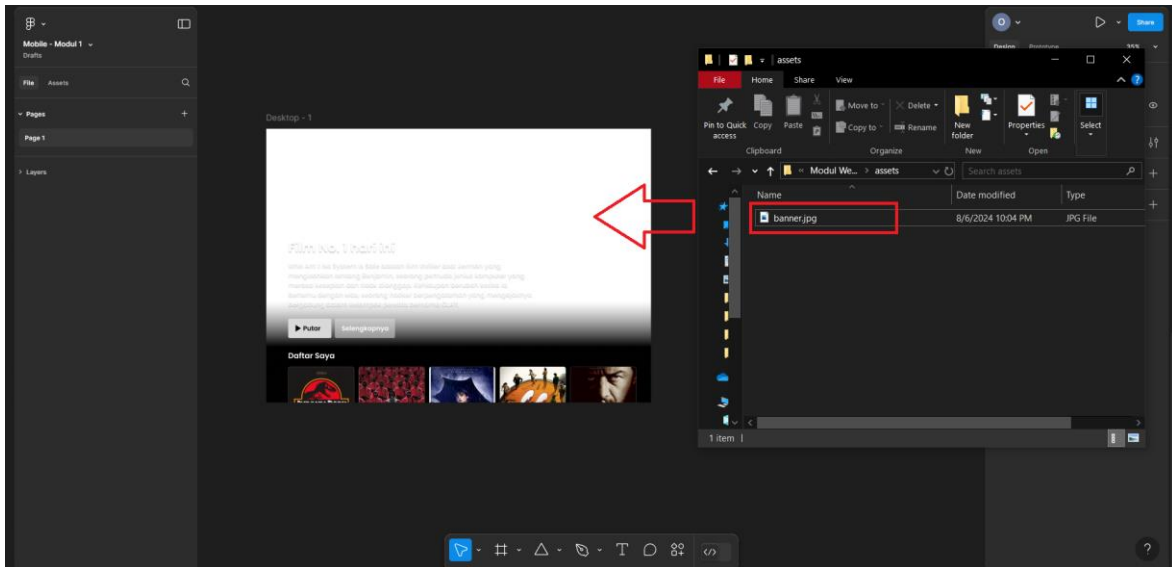
- Jika frame sudah tampil, maka kita dapat mengisi Frame dengan text, gambar, ataupun bentuk yang kita inginkan. Dalam contoh kasus kali ini, kita akan menambahkan text dan gambar pada Frame yang telah kita buat.



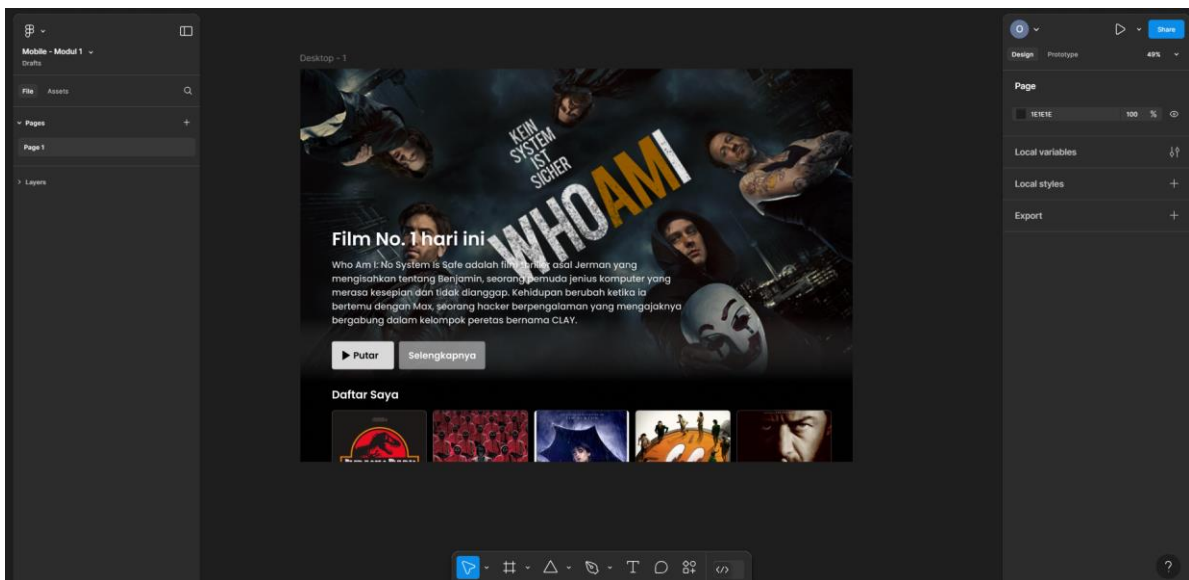
- Kita dapat menambahkan bentuk seperti persegi, lingkaran, garis, maupun text menggunakan menu Figma dibawah ini.



- Untuk dapat menambah gambar, kita bisa menggunakan 2 cara. Cara pertama adalah dengan mengcopy gambar kemudian langsung paste pada halaman Figma, atau bisa Drag and Drop dari File Manager menuju Figma



- Berikut ini merupakan contoh hasil desain kita menggunakan Figma. Desain dibuat untuk mempermudah developer dalam melakukan proses pengembangan aplikasi. Oleh karena itu, desain menjadi peran penting dalam tahap pengembangan.



CODELAB

Membuat tabel sederhana untuk jadwal kuliah

CODELAB 1**JADWAL KULIAH**

Senin	Selasa	Rabu
Pemrograman Web	Jaringan Komputer	Piranti Cerdas
Pemrograman Fungsional	Sistem Operasi	Komunikasi Data
Penulisan Ilmiah	Interaksi Manusia Komputer	Kecerdasan Buatan

Clue: gunakan heading, table, tr, td, th.

Kemudian, buat menjadi lebih variatif dengan mengimplementasikan CSS kedalam tabel hingga menjadi seperti contoh dibawah ini. Kalian diperbolehkan mencoba ketiga penerapan pemanggilan CSS kedalam program.

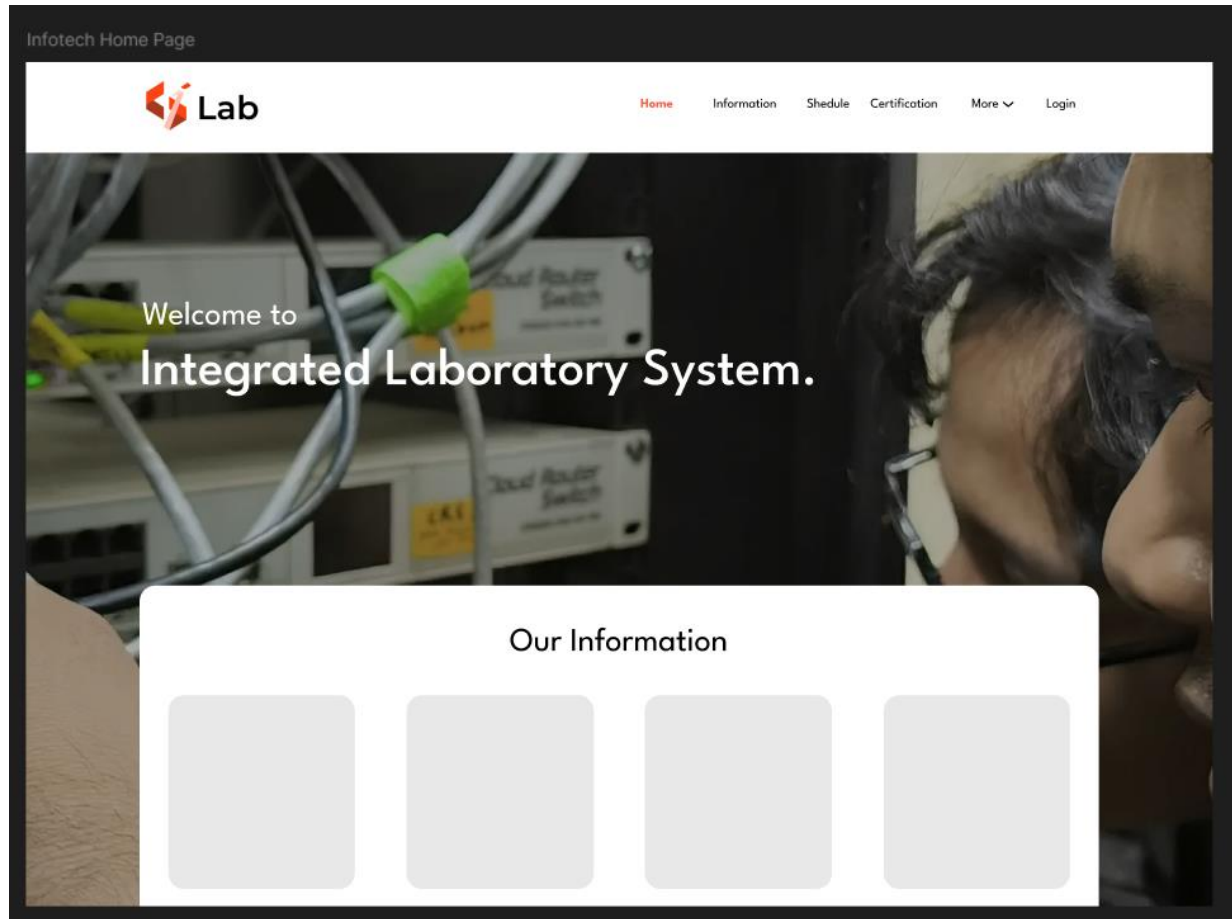
JADWAL KULIAH

Senin	Selasa	Rabu
Pemrograman Web	Jaringan Komputer	Piranti Cerdas
Pemrograman Fungsional	Sistem Operasi	Komunikasi Data
Penulisan Ilmiah	Interaksi Manusia Komputer	Kecerdasan Buatan

CODELAB 2

Lakukan *slicing* pada desain dibawah ini menggunakan Figma. Bisa juga lakukan tambahan desain untuk latihan. Berikut merupakan url untuk asset yang dapat digunakan sebagai referensi:

<https://drive.google.com/drive/folders/1f7hMEYwdH9ITg4J4ESdS9rF5Nw9hUIZI?usp=sharing>



TUGAS PRAKTIKUM

TUGAS 1

Membuat desain website sederhana yang menarik melalui Figma (tidak ada ketentuan tema), namun harus menerapkan header, main content, dan footer kedalam desain.

TUGAS 2

Dari desain yang telah kalian buat, implementasikan kedalam kode **HTML dan CSS** kemudian demokan kepada asisten.

Diperbolehkan jika ingin menggunakan Bootstrap atau HTML framework lain selama dapat menjelaskan dengan baik.

Note: karena tidak ada ketentuan tema, maka dipastikan desain web mahasiswa 1 dengan yang lainnya pasti akan berbeda. Jika terdapat kemiripan yang signifikan dari desain dan kode, maka akan dilakukan pengurangan nilai.

KRITERIA & DETAIL PENILAIAN

Kriteria Penilaian	Persentase Penilaian
Dapat membuat desain yang menarik (Tugas 1)	30%
Dapat mengimplementasikan CSS dan HTML (Tugas 2)	50%
Dapat menjawab pertanyaan Asisten	20%