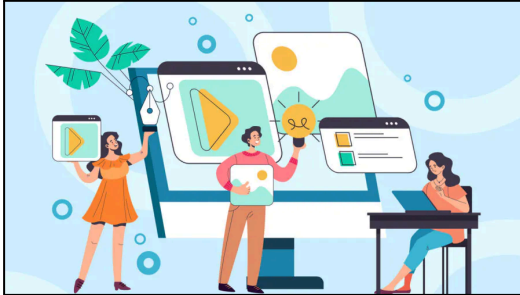


Modul Internship Exercise Software Frontend - 2024

Minggu pertama: HTML, CSS, basic JS, and Git

- Frontend



- Bagian dari website yang dapat dilihat dan berinteraksi dengan pengguna seperti antarmuka pengguna grafis (GUI / Graphical User Interface) termasuk desain, menu navigasi, teks, gambar, video, dll.

- Html



- Hypertext Markup Language atau HTML adalah sebuah markup language yang mendefinisikan komponen apa saja yang ada di dalam website untuk membangun bagian front-end dari website tersebut
- Common tag : h1, h2, h3, div, p, img

- CSS



- Cascading Style Sheets atau biasa disebut dengan CSS merupakan sebuah language sederhana yang didesain untuk mempermudah proses penghiasan suatu website sehingga tampilan dari website tersebut terlihat indah
 - Flexbox

Flexible Box atau Flexbox merupakan sebuah attribute pada CSS yang biasanya digunakan sebagai pengatur layout untuk komponen - komponen yang ada di dalam suatu website secara dinamis. Flexbox mengatur layout dengan cara mendistribusikan ruang dan mengorganisir *item - item* yang ada di dalam *container* yang memiliki attribute flexbox tersebut.

ikuti

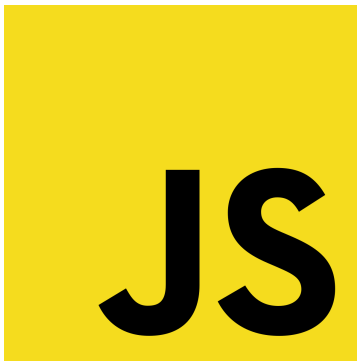
video

ini:

https://www.youtube.com/watch?v=K74l26pE4YA&ab_channel=Fireship

- Pengenalan JS

- Javascript



JavaScript adalah bahasa pemrograman yang digunakan dalam pengembangan website agar lebih dinamis dan interaktif. Kalau sebelumnya kamu hanya mengenal HTML dan CSS, nah sekarang kamu jadi tahu bahwa JavaScript dapat meningkatkan fungsionalitas pada halaman web. Bahkan dengan JavaScript ini kamu bisa membuat aplikasi, tools, atau bahkan game pada web.

- Keuntungan menggunakan JS

- Interaktivitas
 - Kecepatan
 - Fleksibilitas dan kemudahan integrasi
 - Kompatibilitas browser
 - Komunitas besar dan ekosistem
 - Pemrograman sisi server dengan Node.js
 - Pembaharuan dan pengembangan berkelanjutan

- Variable dan function

- Variable adalah tempat untuk menyimpan data. Variabel dapat menyimpan berbagai jenis data seperti angka, teks, objek, atau array
 - Function adalah blok kode yang dapat digunakan ulang untuk melakukan tugas tertentu. Fungsi dapat menerima parameter (input) dan mengembalikan hasil (output)

- const, var, let

- var

- Scope

var memiliki fungsi atau global scope, yang berarti variabel yang dideklarasikan dengan var akan tersedia di seluruh fungsi di mana ia dideklarasikan atau di seluruh skrip jika dideklarasikan di luar fungsi.

- Hoisting

Deklarasi var dihoist ke atas ruang lingkungannya, yang berarti variabel dapat digunakan sebelum dideklarasikan tetapi akan memiliki nilai undefined.

- Redeclaration

var memungkinkan redeklarasikan variabel di dalam ruang lingkup yang sama tanpa menghasilkan error.

- let

- Scope

let memiliki block scope, yang berarti variabel yang dideklarasikan dengan let hanya tersedia di dalam blok di mana ia dideklarasikan (misalnya, di dalam if, for, atau {}).

- Hoisting

let dihoist, tetapi tidak diinisialisasi. Menggunakan let sebelum deklarasinya dalam ruang lingkup yang sama akan menghasilkan error ReferenceError.

- Redeclaration

let tidak memungkinkan redeklarasikan variabel di dalam ruang lingkup yang sama.

- const

- Scope

Sama seperti let, const memiliki block scope.

- Hoisting

Sama seperti let, deklarasi const dihoist tetapi tidak diinisialisasi. Menggunakan const sebelum deklarasinya dalam ruang lingkup yang sama akan menghasilkan error ReferenceError.

- Immutable

Variabel yang dideklarasikan dengan const harus diinisialisasi saat deklarasi dan nilainya tidak dapat diubah setelahnya. Namun, jika const digunakan untuk objek atau array, properti internal dari objek atau elemen array masih dapat diubah.

- Data types in JS (primitive and non-primitive)

- Any

JavaScript adalah bahasa yang didalamnya ada class any. variabel yang bertipe ini bersifat fleksibel dimana tipe data yang disimpan didalamnya dapat menjadi apa saja. Dari integer (angka bulat), float (angka koma), string (teks), variable, void, dan objek. Any memiliki kelebihan serta kekurangannya

- Conditionals

- If

Pernyataan **if** digunakan untuk mengeksekusi blok kode jika suatu kondisi bernilai **true**. Jika kondisi tersebut bernilai **false**, maka blok kode tersebut dilewati.

- Else if

Pernyataan **else if** digunakan untuk mengevaluasi kondisi lain jika kondisi sebelumnya dalam **if** bernilai **false**. Ini memungkinkan pengecekan beberapa kondisi.

- Else

Pernyataan **else** digunakan sebagai blok terakhir, yang akan dieksekusi jika semua kondisi sebelumnya (baik **if** maupun **else if**) bernilai **false**.

- Objects

- Apa itu object

Objek dalam pemrograman adalah entitas yang menyimpan data (Field) dan perilaku (Method) yang terkait. Objek adalah salah satu konsep utama dalam pemrograman berorientasi objek (OOP) dan sering digunakan untuk memodelkan entitas dunia nyata dalam kode. Berikut adalah penjelasan lebih rinci.

- Field

Field adalah variabel yang terkait dengan objek dan digunakan untuk menyimpan data. Misalnya, sebuah objek "mobil" mungkin memiliki properti seperti "warna", "merek", dan "kecepatan".

- Method

Metode adalah fungsi yang terkait dengan objek dan digunakan untuk mendefinisikan perilaku objek. Misalnya, objek "mobil" mungkin memiliki metode seperti "jalan" dan "berhenti".

- Contoh:

```
Membuat objek menggunakan literal objek
```

```
mobil = {  
  warna: "merah",  
  merek: "Toyota",  
  kecepatan: 0,  
}
```

```
/ Metode untuk menjalankan mobil
```

```
jalan: function(tambahKecepatan) {  
  this.kecepatan += tambahKecepatan;  
  console.log(`Mobil berjalan dengan kecepatan ${this.kecepatan} km/jam`);  
},
```

```
/ Metode untuk menghentikan mobil
```

```
berhenti: function() {  
  this.kecepatan = 0;  
  console.log("Mobil berhenti");  
}
```

```
Mengakses properti objek
```

```
console.log(mobil.warna); // Output: merah
```

```
Menggunakan metode objek
```

```
il.jalan(50); // Output: Mobil berjalan dengan kecepatan 50 km/jam
```

```
il.berhenti(); // Output: Mobil berhenti
```

- Git dan github



- Git adalah sistem kontrol versi terdistribusi yang dirancang untuk menangani segala sesuatu mulai dari proyek kecil hingga proyek besar dengan kecepatan dan efisiensi.
- GitHub adalah platform pengembangan perangkat lunak yang menyediakan hosting untuk repositori Git. Ini adalah alat kolaborasi yang kuat yang memungkinkan pengembang untuk bekerja sama, mengelola proyek, dan berbagi kode.
- Command-command git
 - Git add : Menambahkan perubahan file yang baru dibuat atau diubah ke **staging area**. File yang ada di staging area siap untuk di-commit.
 - Git commit : Menyimpan snapshot dari perubahan yang telah ada di **staging area** ke dalam repositori lokal dengan memberikan pesan yang menjelaskan perubahan tersebut.
 - Git push : Mengirim perubahan yang telah di-commit ke **repositori remote** (misalnya, GitHub).

Minggu dua : React, Tailwind, and advanced javascript

- Javascript



- Dalam konteks front-end, Javascript atau biasa disingkat JS merupakan sebuah scripting language yang digunakan dalam suatu website agar komponen yang ada didalamnya dapat digunakan user untuk berinteraksi dengan website

tersebut. Sederhananya, Javascript bertugas untuk memberikan fungsionalitas terhadap komponen - komponen yang ada dalam website tersebut.

- Lambda

Lambda, atau lebih dikenal sebagai arrow function dalam JavaScript, adalah cara singkat untuk menulis fungsi. Arrow function diperkenalkan di ECMAScript 6 (ES6) dan memiliki sintaks yang lebih ringkas dibandingkan dengan fungsi tradisional. contoh:

- Arrow function dengan beberapa parameter dan terdapat body

```
let sum = (a, b) => {  
  return a + b;  
};  
console.log(sum(3, 4)); // Output: 7
```

- Arrow function dengan satu parameter dan body sederhana

```
let square = x => x * x;  
console.log(square(5)); // Output: 25
```

- arrow function tanpa parameter

```
let greet = () => console.log("Hello, World!");  
greet(); // Output: Hello, World!
```

- Async

Asynchronous programming adalah teknik penting dalam JavaScript untuk menangani operasi yang memakan waktu tanpa menghambat eksekusi kode lainnya. Dengan menggunakan callback, promise, dan async/await, pengembang dapat menulis kode yang lebih efisien dan responsif. Async/await, khususnya, menawarkan cara yang lebih mudah dan lebih bersih untuk menulis kode asinkron, membuatnya lebih mudah untuk dibaca dan dipelihara.

- Callback

Callback adalah fungsi yang diteruskan sebagai argumen ke fungsi lain dan dipanggil setelah operasi asinkron selesai. Contoh:

```
function fetchData(callback) {  
  setTimeout(() => {  
    callback("Data fetched");  
  }, 2000);  
}  
  
fetchData(data => {  
  console.log(data); // Output setelah 2 detik: Data fetched  
});
```

- Async/Await

Async/await adalah sintaks yang diperkenalkan di ECMAScript 2017 (ES8) untuk menulis kode asinkron yang terlihat dan berperilaku seperti kode sinkron. Fungsi yang menggunakan await harus dideklarasikan dengan async.

```

async function fetchData() {
  let promise = new Promise((resolve, reject) => {
    setTimeout(() => {
      resolve("Data fetched");
    }, 2000);
  });

  let data = await promise;
  console.log(data); // Output setelah 2 detik: Data fetched
}

fetchData();

```

Contoh :

```

let angka = 1;

if (angka > 5) {
  console.log("Angka lebih besar dari 5");
} else if (angka > 2) {
  console.log("Angka lebih besar dari 2 tetapi kurang dari atau sama dengan 5");
} else {
  console.log("Angka lebih kecil atau sama dengan 2");
}

```

- Looping
 - For, While

A for loop	A while loop
<pre> for (let i=1; i<=5; i++) { if (i == 4) break; document.write(i+"
"); } </pre>	<pre> let i=1; while (i <= 5) { if (i == 4) break; document.write(i+"
"); i++; } </pre>

For merupakan looping yang dipakai jika user sudah mengetahui berapa kali loop akan dijalankan. Sedangkan while merupakan looping yang dipakai jika user belum mengetahui berapa kali loop akan dijalankan.

- .forEach

```
let foods = ['bread', 'rice', 'meat', 'pizza'];

foods.forEach(function(food) {
  console.log(food);
})
```

forEach dalam JavaScript adalah sebuah metode yang diterapkan pada objek array dan digunakan untuk melakukan iterasi atau pengulangan pada setiap elemen dalam array tersebut.

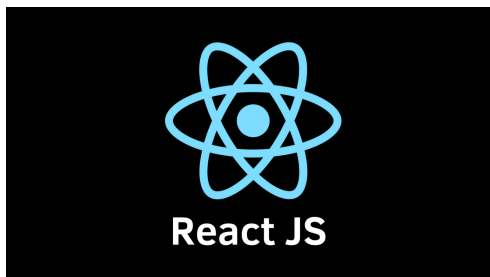
- .map

```
const arr = [1,2,3,4,5]

arr.map(function(item){
  console.log(item + 1);
});
```

map merupakan sebuah metode untuk memetakan setiap variabel yang ada di dalam array. Di dalam map kita dapat menentukan perlakuan apa yang ingin dilakukan pada setiap variabel data.

- Apa itu react



- React merupakan sebuah JavaScript library yang digunakan untuk membuat User Interfaces (UIs) pada suatu web. React merupakan sebuah deklaratif, component-based library yang memungkinkan developer untuk membangun komponen UI yang dapat digunakan kembali. React mengikuti pendekatan Virtual DOM (Document Object Model) yang mengoptimalkan performa rendering dengan meminimalkan DOM update. React kompatibel dengan banyak tools dan library lainnya.
- Sebuah file react disimpan dalam 2 jenis file, yaitu .jsx dan .tsx. Kedua jenis file tersebut memungkinkan developer untuk menulis HTML dalam kode

javascriptnya. Perbedaan utama antara .jsx dan .tsx adalah .jsx menggunakan plain javascript sedangkan .tsx menggunakan typescript

- Dalam reactjs, ada yang dinamakan dengan “Hook”. Hooks merupakan sebuah reusable function yang memberikan kemampuan untuk menggunakan state dan fitur react lainnya tanpa harus menulis sebuah Class. Terdapat dua react hook yang sering digunakan yaitu useState dan useEffect. React useState digunakan untuk mentrack state dalam sebuah component. State biasanya mengacu pada data atau properti yang perlu ditrack dalam aplikasi. React useEffect digunakan untuk melakukan side effect pada komponen dalam react application seperti fetching data, mengupdate DOM, dan timers.

- Apa itu vite



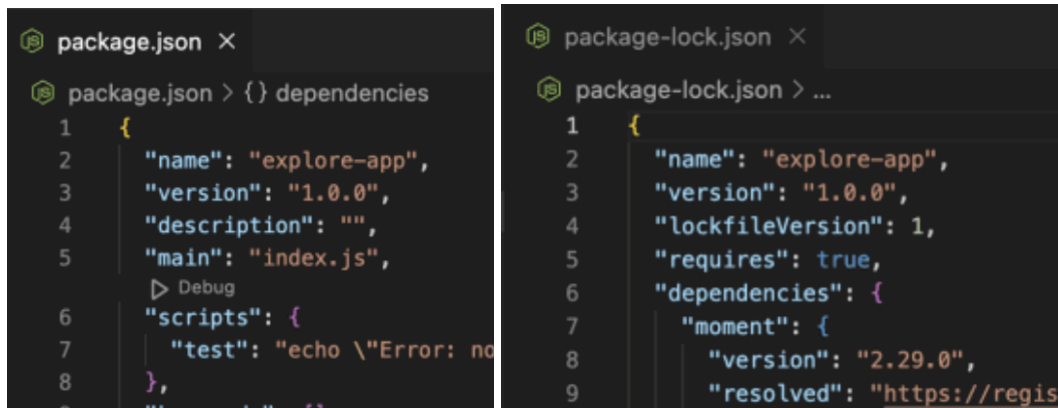
- Vite merupakan sebuah build tool yang bertujuan untuk memberikan pengalaman pengembangan yang lebih cepat dan ringan untuk proyek web - web modern. Vite menyediakan 2 hal utama, yaitu :
 - Sebuah dev server yang menyediakan peningkatan fitur dibandingkan native ES module, seperti Hot Module Replacement (HMR)
 - Build command yang telah dikonfigurasi untuk menghasilkan aset statis yang sangat dioptimalkan untuk production.

- Tailwind



- Tailwind CSS merupakan sebuah CSS framework yang didesain untuk mempercepat user dalam proses penghiasan web, seperti mengontrol tata letak, warna, jarak, tipografi, bayangan, dan lainnya untuk membuat desain komponen yang benar benar custom, dengan menyediakan sebuah set utility-class yang telah dirancang sebelumnya. Sehingga komponen dapat dikustomisasi menggunakan class - class yang telah disediakan, tanpa harus meninggalkan file HTML / JSX yang anda kerjakan sekarang.

- Files and folder system



- package.json dan package-lock.json

Dalam sebuah project yang menggunakan Node.JS terdapat dua buah file yang terbuat secara otomatis, yaitu package.json dan package-lock.json.

package.json merupakan sebuah file yang utamanya digunakan untuk mengelola dan mendokumentasikan metadata tentang projectnya, termasuk nama, versi, penulis, dependensi, skrip, dan detail konfigurasi lainnya. File ini bertindak sebagai manifest untuk proyek. file ini dapat diedit secara manual untuk menambah atau menghapus dependensi, memodifikasi skrip, memperbarui versi, atau membuat perubahan konfigurasi lainnya.

package-lock.json merupakan sebuah file yang dibuat dan diperbarui oleh npm ketika menginstal atau memperbarui packages. File ini digunakan untuk mengunci versi yang tepat dari dependensi yang diinstal dalam proyek, memastikan reproduktifitas dan instalasi yang konsisten di lingkungan yang berbeda. File ini umumnya tidak dimaksudkan untuk diedit secara manual, karena dikelola secara otomatis oleh npm. Perubahan manual pada file ini dapat menyebabkan ketidakkonsistenan atau konflik dalam penyelesaian dependencies.

- assets

Folder **assets** biasanya digunakan untuk menyimpan gambar, ikon, dan file lain yang diperlukan dalam sebuah proyek, seperti video, audio, atau font. File dalam folder ini sering digunakan untuk mendukung tampilan dan fungsi aplikasi atau situs web.

- src

- pages

berisi file-file .jsx dari masing-masing page

- components

berisi file-file .jsx yang merupakan component yang dapat dipakai di berbagai page

Minggu tiga : Advanced React and Axios

- useState

useState digunakan untuk mendeklarasikan state dalam komponen fungsi. Ketika kita menggunakan useState, kita mendapatkan sepasang nilai:

- Nilai saat ini dari state
- Fungsi untuk memperbarui nilai state

Contoh :

```
1 import React, { useState } from 'react';
2
3 function Counter() {
4   const [count, setCount] = useState(0);
5
6   return (
7     <div>
8       <p>Count: {count}</p>
9       <button onClick={() => setCount(count + 1)}>Increment</button>
10    </div>
11  );
12 }
13
```

- useEffect

useEffect digunakan untuk menangani efek samping, seperti melakukan pemanggilan API, mengatur atau membersihkan timer, atau memodifikasi DOM langsung. useEffect dipanggil setelah komponen dirender.

useEffect menerima dua parameter

- Fungsi efek: Fungsi yang akan dijalankan.
- Dependency array: Array yang menentukan kapan efek akan dijalankan ulang. Jika array kosong [], efek hanya akan dijalankan sekali setelah render pertama.

```

1 import React, { useState, useEffect } from 'react';
2 import axios from 'axios';
3
4 function DataFetcher() {
5   const [data, setData] = useState([]);
6   const [loading, setLoading] = useState(true);
7
8   useEffect(() => {
9     const fetchData = async () => {
10      try {
11        const response = await axios.get('https://jsonplaceholder.typicode.com/posts');
12        setData(response.data); // Menyimpan data hasil GET request ke state
13      } catch (error) {
14        console.error("Error fetching data:", error);
15      } finally {
16        setLoading(false); // Mengubah status loading menjadi false
17      }
18    };
19
20    fetchData();
21  }, []); // Array kosong berarti efek hanya dijalankan sekali
22
23  if (loading) return <p>Loading...</p>;
24
25  return (
26    <div>
27      <h1>Posts</h1>
28      <ul>
29        {data.map((post) => (
30          <li key={post.id}>{post.title}</li>
31        ))}
32      </ul>
33    </div>
34  );
35 }
36
37 export default DataFetcher;
38

```

- Request
 - Jenis - Jenis request
 - GET

Method untuk mengambil data dari server, sebaiknya menggunakan params untuk request
 - POST

Method untuk memberikan data kepada server, sebaiknya menggunakan request body untuk memuat data yang dikirimkan
 - PUT

Method untuk mengupdate data yang sudah ada di database server
 - DELETE

Method untuk menghapus data yang sudah ada di database server

Minggu empat : collab dengan BE