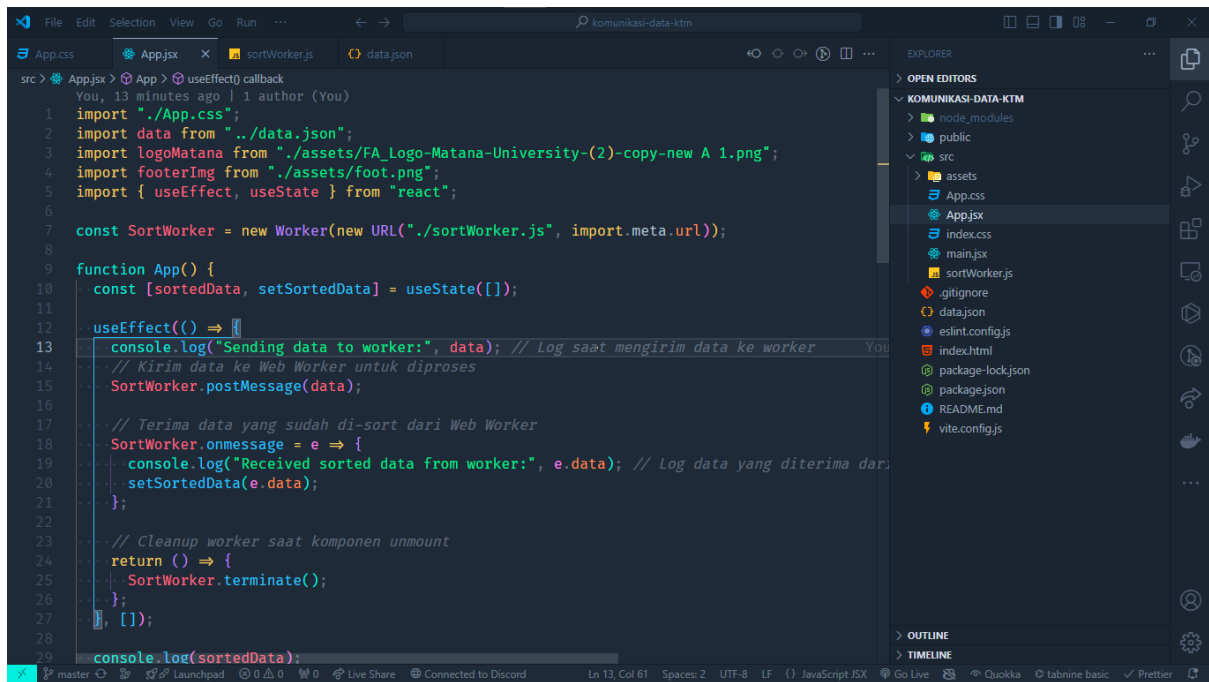


UAS Komunikasi Data

Penilaian Individu, Jeriko Ichtus Seo(20235520002)



The screenshot shows a VS Code editor with a project named 'KOMUNIKASI-DATA-KTM'. The main file being edited is 'App.jsx'. The code in 'App.jsx' uses 'useState' to manage a 'sortedData' state and 'useEffect' to interact with a 'SortWorker' Web Worker. The worker is initialized with a URL to 'sortWorker.js'. The 'App' function sets up the state and uses 'useEffect' to send data to the worker and receive sorted data back. Comments in the code explain the flow: sending data to the worker for processing, receiving sorted data, and cleaning up the worker when the component unmounts. The file explorer on the right shows the project structure, including 'assets', 'App.css', 'App.jsx', 'index.css', 'main.jsx', 'sortWorker.js', 'data.json', and various configuration files like 'package.json', 'vite.config.js', and 'eslint.config.js'.

```
src > App.jsx > App > useEffect() callback
You, 13 minutes ago | 1 author (You)
1 import "../App.css";
2 import data from "../data.json";
3 import logoMatana from "../assets/FA Logo-Matana-University-(2)-copy-new A 1.png";
4 import footerImg from "../assets/foot.png";
5 import { useEffect, useState } from "react";
6
7 const SortWorker = new Worker(new URL("./sortWorker.js", import.meta.url));
8
9 function App() {
10   const [sortedData, setSortedData] = useState([]);
11
12   useEffect(() => {
13     console.log("Sending data to worker:", data); // Log saat mengirim data ke worker
14     // Kirim data ke Web Worker untuk diproses
15     SortWorker.postMessage(data);
16
17     // Terima data yang sudah di-sort dari Web Worker
18     SortWorker.onmessage = e => {
19       console.log("Received sorted data from worker:", e.data); // Log data yang diterima dari worker
20       setSortedData(e.data);
21     };
22
23     // Cleanup worker saat komponen unmount
24     return () => {
25       SortWorker.terminate();
26     };
27   }, []);
28
29   console.log(sortedData);
30 }
```

1. State Management:

- *useState* digunakan untuk menyimpan data mahasiswa yang sudah diurutkan.
- *useEffect* dijalankan saat komponen pertama kali dirender. Fungsi ini mengirim data mentah ke Web Worker untuk diurutkan, kemudian menerima data yang sudah diurutkan dan menyimpannya ke dalam state *sortedData*.

2. Web Worker:

- Web Worker dibuat menggunakan *new Worker*, dan fungsinya adalah mengurutkan data tanpa membebani thread utama.

- Saat worker selesai mengurutkan data, ia mengirimkan data yang sudah diurutkan kembali ke komponen React, yang kemudian menyimpannya dalam state.

3. Render Data:

- Jika `sortedData` sudah terisi, data akan ditampilkan dalam elemen-elemen HTML (gambar, nama mahasiswa, program studi, dan NIM).
- Jika `sortedData` masih kosong (data belum tersedia), pesan "Loading data..." akan ditampilkan.

4. Clean-Up:

- Worker di-terminate saat komponen di-unmount untuk menghindari kebocoran memori.

Tujuan:

Menggunakan Web Worker untuk mengurutkan data secara asinkron sehingga antarmuka tetap responsif.