

Pengenalan tentang React.js

Pengertian:

React.js adalah sebuah pustaka JavaScript yang dikembangkan oleh Facebook untuk membangun antarmuka pengguna (user interfaces) yang interaktif dan efisien. React.js memungkinkan pengembang untuk membuat komponen-komponen UI yang dapat digunakan kembali, serta memfasilitasi pengelolaan state dan rendering yang efisien. Dengan pendekatan deklaratifnya, React membuat kode lebih mudah untuk diprediksi dan debug. Selain itu, React menggunakan Virtual DOM, yang memungkinkan pembaruan UI dilakukan dengan cepat tanpa harus memuat ulang seluruh halaman.

Persyaratan Laptop untuk Pengembangan dengan React.js:

Untuk mengembangkan aplikasi menggunakan React.js dengan lancar, laptop yang digunakan sebaiknya memenuhi beberapa persyaratan teknis.

- Sistem operasi yang disarankan adalah Windows 10 atau lebih baru, macOS Mojave (10.14) atau lebih baru, atau Linux dengan distribusi terbaru.
- Prosesor minimal Intel Core i3 atau setara, namun disarankan Intel Core i5 atau lebih tinggi.
- RAM minimal 4GB, tetapi disarankan 8GB atau lebih untuk performa yang lebih baik.
- Penyimpanan minimal 20GB ruang kosong, dengan SSD sangat disarankan.
- Resolusi layar minimal 1366 x 768 piksel, dengan 1920 x 1080 piksel atau lebih tinggi sebagai pilihan terbaik.

Selain spesifikasi perangkat keras, beberapa perangkat lunak pendukung juga diperlukan.

- Node.js versi terbaru (disarankan LTS) beserta npm (Node Package Manager) harus diinstal, yang dapat diunduh dari nodejs.org.
- Editor kode seperti Visual Studio Code sangat disarankan, meskipun editor teks lain yang mendukung JavaScript dan JSX juga bisa digunakan, seperti Atom, Sublime Text, atau Notepad++.
- Browser modern seperti Google Chrome atau Firefox diperlukan untuk mendukung alat pengembang web.

Untuk persiapan lingkungan pengembangan, pastikan Node.js dan editor kode terinstal, dan sistem operasi serta perangkat lunak diperbarui ke versi terbaru. Koneksi internet diperlukan untuk mengunduh paket dan dependensi yang dibutuhkan. Dengan memenuhi persyaratan ini, mahasiswa akan siap untuk mengikuti praktikum React.js dengan lancar dan efektif.

Kesimpulan:

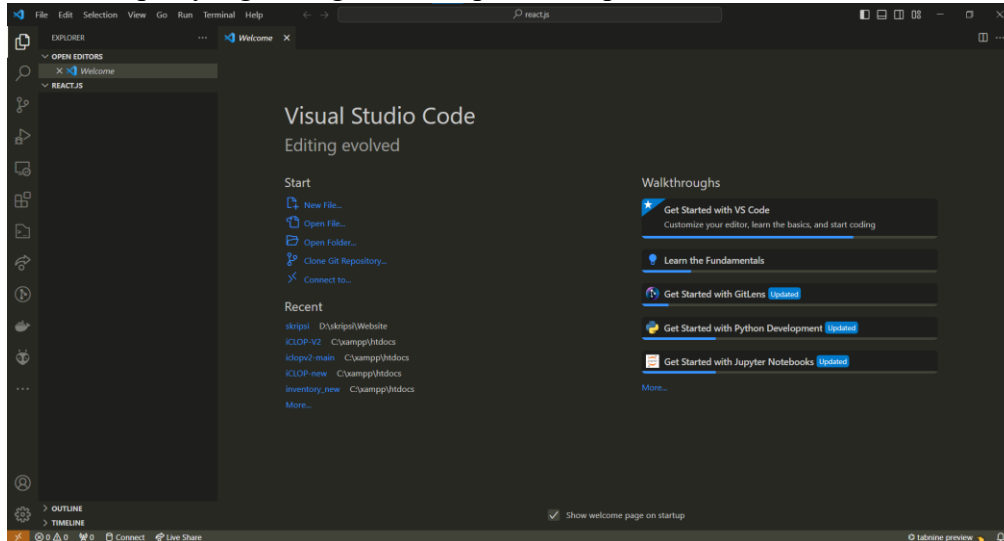
Dalam praktikum ini, mahasiswa akan belajar membuat aplikasi React.js dasar, mulai dari setup project hingga membuat komponen interaktif sederhana. Praktikum ini memberikan pemahaman tentang bagaimana membuat dan mengelola komponen dalam React, serta bagaimana memanipulasi state dan event handling untuk membuat aplikasi yang responsif dan interaktif. Mahasiswa juga diharapkan mampu melakukan modifikasi dan pengembangan lebih lanjut berdasarkan pemahaman yang telah didapatkan.

Langkah-langkah Pembuatan Project React.js:

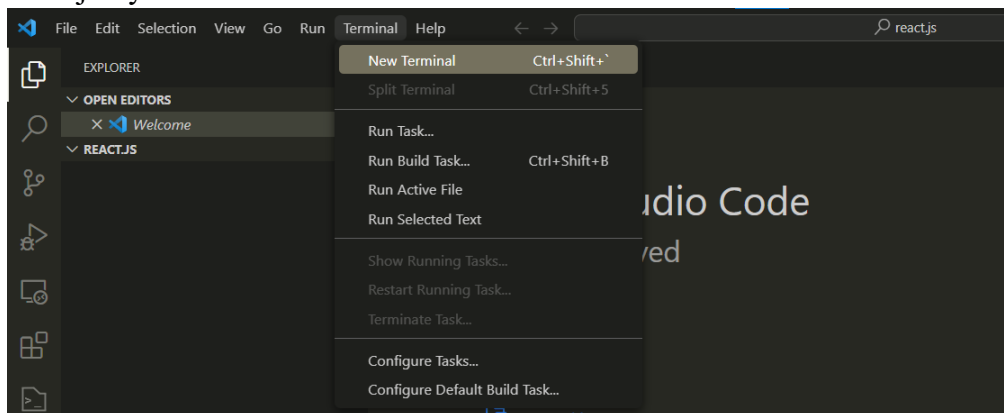
1. Buatlah folder baru dengan nama react.js.



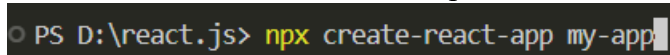
2. Buka notepad yang anda gunakan seperti: notepad++, Visual Studio Code, dll.



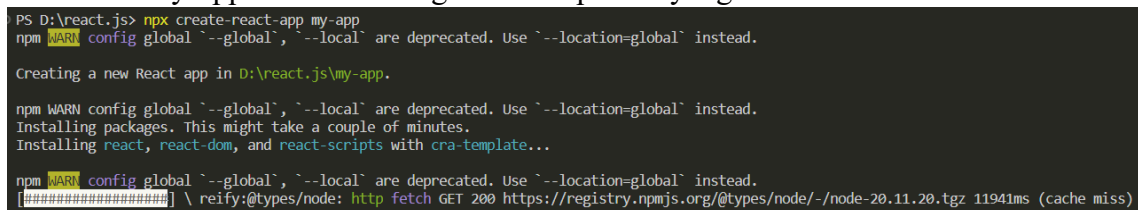
3. Selanjutnya buka terminal.



4. Setelah terminal terbuka masukan perintah berikut untuk membuat project React.js.

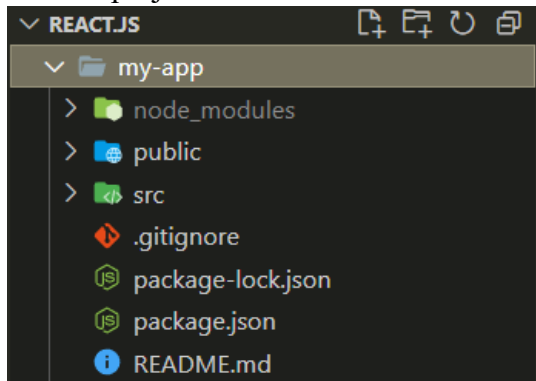


Penamaan my-app sesuaikan dengan nama aplikasi yang anda buat.



Tunggu proses pembuatan project React.js diatas hingga selesai.

- Setelah project React.js berhasil dibuat maka akan terdapat folder *my-app* yang berisi file-file project react.



- Kemudian masukan perintah berikut untuk masuk ke dalam folder project React.js yang bernama *my-app*.

```
PS D:\react.js> cd my-app
```

- Selanjutnya jika ingin menjalankan aplikasi React.js maka lakukan perintah berikut.

```
PS D:\react.js\my-app> npm start
```

Jika berhasil maka akan tampil pada terminal seperti berikut ini

```
o Compiled successfully!

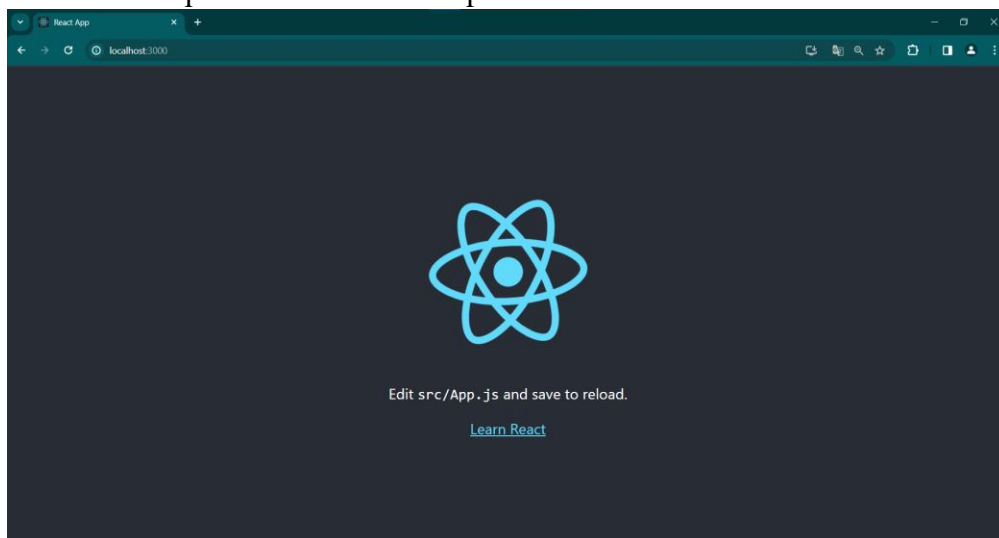
You can now view my-app in the browser.

Local:      http://localhost:3000
On Your Network: http://192.168.1.2:3000

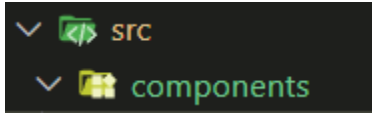
Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

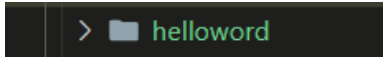
Pada gambar diatas terdapat IP local host dan IP local network yang anda gunakan. Dan akan tampil halaman welcome pada browser anda.



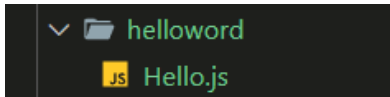
8. Selanjutnya membuat interaksi berupa button dan alert sederhana. Buatlah folder componets pada folder src.



9. Selanjutnya buatlah folder baru Bernama helloword di dalam folder components.



10. Setelah membuat folder buatlah file bernama Hello.js di dalam folder helloword.



11. Kemudian isi file Hello.js dengan code berikut ini.

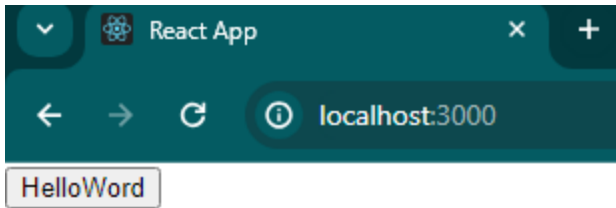
```
1  export default function App() {
2    const handleHelloWorld = () => {
3      alert('Hello!');
4    };
5  };
6  return (
7    <Toolbar
8      onHelloWord={handleHelloWorld}
9    />
10 );
11 }
12
13 function Toolbar({onHelloWord}){
14   return (
15     <div>
16       <Button onClick={onHelloWord}>HelloWord</Button>
17     </div>
18   );
19 }
20
21
22 function Button({ onClick, children }) {
23   return (
24     <button onClick={onClick}>
25       {children}
26     </button>
27   );
28 }
```

12. Selanjutnya import File Hello.js pada App.js serta deklarasikan Hello.js agar return pada file Hello.js tampil pada halaman React App.

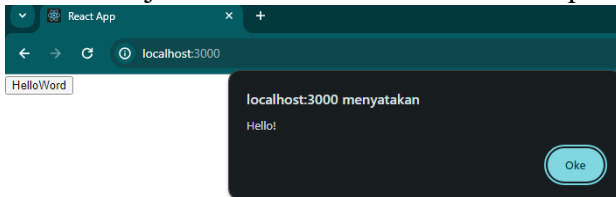
```
import HelloWord from "./components/helloword/Hello";
< HelloWord />
```

Setiap ada penambahan components baru anda wajib membuat import dan mendeklarasikan components yang anda seperti yang ada pada gambar diatas.

13. Browser otomatis akan terbuka kemudian akan tampil halaman seperti berikut ini.



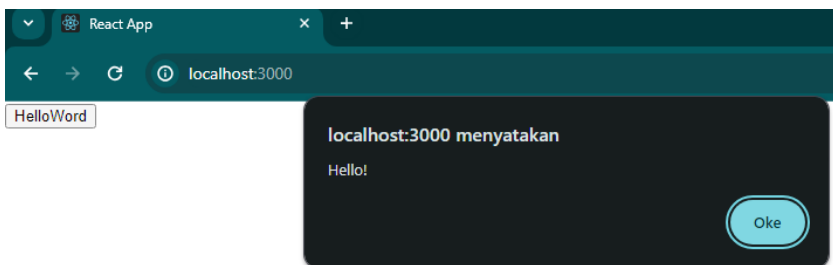
14. Kemudian jika button diklik maka akan tampil alert seperti berikut ini.



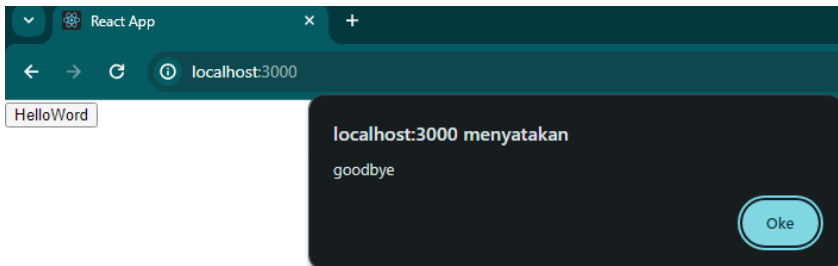
Test Case

Modifikasilah button diatas ketika alert muncul dan diklik “Oke” maka akan muncul alert “GoodBye!” Seperti berikut ini:

Sebelum diklik “Oke”

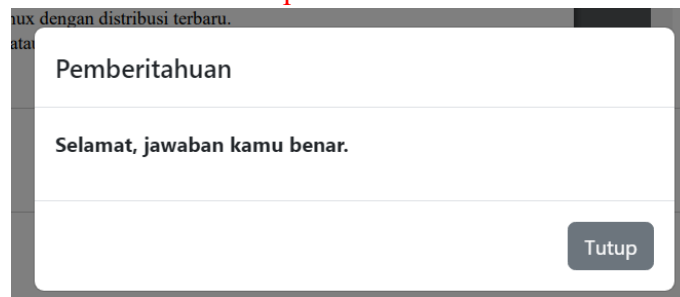


Setelah diklik “Oke”



Jika sudah upload file Hello.js ke dalam iCLOP:

Jika file yang di Upload sesuai dengan perintah praktikum diatas maka akan muncul seperti berikut:



Jika file tidak sesuai dengan perintah diatas praktikum maka akan muncul seperti berikut:

