

Trik Kolaborasi
REACT dengan PHP dan MySQL

I Nyoman Sweta



CV. LOKOMEDIA

Trik Kolaborasi REACT dengan PHP dan MySQL

Penulis : I Nyoman Sweta

Trik Kolaborasi REACT dengan PHP dan MySQL

- Cet. I. - Yogyakarta : Penerbit Lokomedia, 2021

176 halaman; 14 x 20 cm

ISBN : 978-602-6231-29-1

Penerbit Lokomedia,

Cetakan Pertama : Mei 2021

Editor : Lukmanul Hakim

Cover : Subkhan Anshori

Layout : Lukmanul Hakim

Diterbitkan pertama kali oleh :

CV. LOKOMEDIA

Jl. Jambon, Perum. Pesona Alam Hijau 2 Kav. B-4, Kricak
Yogyakarta 55242.

email : redaksi@bukulokomedia.com

website : www.bukulokomedia.com

Copyright © Lokomedia, 2021

Hak Cipta dilindungi oleh Undang-Undang

Dilarang memperbanyak, mencetak ataupun menerbitkan sebagian maupun seluruh isi buku ini tanpa izin tertulis dari penerbit.

KATA PENGANTAR

Puja dan puji penulis tujukan kepada Tuhan Yang Maha Esa, karena berkat Beliau buku ini telah selesai disusun. Buku ini ditujukan untuk semua kalangan yang ingin mempelajari atau sedang membangun aplikasi berbasis web dengan React PHP MySQL.

Penulis menyadari, jika penyusunan buku ini mempunyai kekurangan, namun keyakinan penulis bahwa sekecil apapun buku ini tetap akan memberikan manfaat bagi pengguna.

Kritik dan saran yang bersifat konstruktif dari pembaca sangat kami harapkan untuk perbaikan buku ini. Sehingga jangan segan-segan untuk mengontak penulis lewat media yang tersedia.

Akhir kata, kami ucapkan banyak terima kasih kepada semua pihak yang sudah membantu dalam proses penyusunan buku ini. Semoga kita semua selalu dalam keadaan sehat.

Denpasar, April 2021

I Nyoman Sweta

sweta@bps.go.id

Halaman ini Sengaja Dikosongkan

www.bukulokomedia.com

DAFTAR ISI

BAB 1. Mengenal dan Memulai React	1
1.1. Apa dan Mengapa React?	2
1.2. Prasyarat Belajar React.....	3
1.3. Kekurangan dan Kelebihan React.....	4
1.4. Struktur React.....	5
1.5. Software Wajib dan Tambahan.....	6
1.6. Install React.....	6
1.7. Prinsip Kerja React	11
BAB 2. Teknik Dasar Memahami React	15
2.1. Membuat "Hello World".....	16
2.2. Memahami JSX.....	17
2.3. Menggunakan Style	25
2.2. Mempercantik Tampilan dengan Bootstrap.....	27
BAB 3. React ES6	30
3.1. Apa itu ES6?.....	30
3.2. Pengenalan ES6.....	30
3.2.1. Class	30
3.2.2. Arrow Function	35
3.2.3. Variabel.....	39

3.2.4. forEach	45
3.2.5. Filter.....	47
3.2.6. Map.....	48
BAB 4. Komponen	51
4.1. Pengenalan Komponen	52
4.2. Functional Component.....	55
4.3. Class Component	57
4.4. Template UI dengan Component.....	58
BAB 5. Routing.....	63
5.1. Mengenal Routing.....	64
5.2. Menginstall React Router	65
5.3. Membuat Navigasi Antar Component.....	65
5.4. Menggunakan Parameter	71
BAB 6. Props, State, Hook	76
6.1. Props	76
6.2. State	80
6.3. Hook	83
BAB 7. Form.....	85
7.1. Pengenalan Form	86
7.2. Membuat Form di React.....	86
7.3. Menangani Form.....	88

7.4. Menggunakan React Hook Form	90
7.5. Validasi dan Submit Form dengan React Hook Form.....	91
BAB 8. REST API	95
8.1. Pengenalan API.....	96
8.2. Pengenalan Axios.....	96
8.3. Mengakses API	97
8.4. Membuat REST API dengan PHP.....	100
8.5. Submit Form API (JSON)	108
BAB 9. Interaksi React PHP MySQL	113
9.1. Konsep Interaksi React PHP MySQL.....	114
9.2. Pembuatan Database untuk REST API	115
9.3. Membuat PHP REST API dari Database MySQL.....	116
9.4. Menampilkan Data.....	117
9.5. Fitur Pencarian Data	120
9.6. Pagination.....	123
BAB 10. Mengolah Database (CRUD)	128
10.1. Pengenalan Operasi CRUD.....	128
10.2. Menyiapkan Kerangka File	129
10.3. Menyiapkan Database.....	131
10.4. Menyiapkan Template.....	133
10.5. Menampilkan Data Produk.....	138
10.6. Menambah Data Produk	144

10.7. Mengedit Data Produk.....	151
10.8. Menghapus Data Produk	159
BAB 11. Membuat Chart (Grafik).....	163
11.1. Mengenal Chart JS2.....	164
11.2. Install Chart JS2	164
11.3. Membuat Grafik dengan Data Statis	164
11.4. Membuat Grafik dengan Data Dinamis (Database/JSON).....	169

BAB I



MENGENAL DAN MEMULAI REACT

BAB 1

Mengenal dan Memulai React

1.1 Apa dan Mengapa React

Kita tentu sudah tidak asing lagi dengan *Facebook*, layanan jejaring sosial yang sangat populer ini. Di Indonesia sendiri pengguna *Facebook* per Desember 2020 mencapai 170 juta. Angka yang sangat luar biasa tentunya.

Menggunakan Facebook kita sudah akrab dengan tombol *Like*, *Comment*, dan *Share*. Pengguna umum ketika menggunakan tombol tersebut mungkin akan merasakan hal yang biasa saja. Tapi bagi pengguna yang berkecimpung dalam pemrograman khususnya yang berbasis *web*, akan merasakan kelebihan fitur tersebut. Ya, pada web yang tidak menerapkan teknologi seperti itu, setiap memilih tombol navigasi untuk berpindah halaman, maka kita kan menunggu beberapa saat sampai halaman yang kita tuju ditampilkan, dan itu memerlukan waktu.

Sementara di *Facebook*, kalau menulis komentar kemudian menekan tombol enter maka komentar yang ditulis langsung muncul saat itu juga, tidak menunggu proses loading. Begitu juga ketika berpindah halaman, proses transisi atau perpindahan berjalan dengan cepat. Itulah salah satu contoh penggunaan React yang dibuat langsung oleh *Facebook*. Dengan kemampuan tersebut, aplikasi berbasis *web* memiliki kemampuan seperti aplikasi berbasis desktop pada umumnya.

React merupakan sebuah *library Javascript* untuk membangun *user interfaces* [S02]. Dari pernyataan tersebut ditekankan bahwa React itu adalah *library* bukan *framework*. Berikutnya *React* lebih fokus membangun desain UI (*user interfaces*). Bagaimana dengan aspek fungsi lain seperti HTTP, Routing dan sebagainya? Jangan khawatir, *React* memiliki ekosistem yang luar biasa sehingga fungsionalitas dengan mudah kita dapatkan baik yang dibuat oleh *Facebook* maupun pihak ketiga.

Muncul pertanyaan mengapa kita perlu mempelajari React, kenapa tidak cukup bahasa yang sudah umum kita gunakan sebelumnya? Itu pertanyaan yang wajar

ketika ingin belajar sesuatu yang baru kita ketahui, sehingga tidak serta merta ikut arus sementara kita tidak memahami manfaatnya. Berikut alasan mengapa perlu belajar *React*:

1. Ingin membuat aplikasi yang interaktif, dinamis seperti fitur *like*, *comment* dan *share* yang dimiliki oleh *Facebook*.
2. *React* dibuat dan di *maintenance* oleh *Facebook* yang reputasinya tidak perlu kita ragukan lagi. Sehingga perkembangan dan keberlangsungan dari *React* itu sendiri sudah tidak usah kita ragukan lagi.
3. Memiliki komunitas yang sangat besar. Ketika kita mengalami permasalahan dalam pengembangan maka akan lebih mudah menanyakan atau mencarinya di *internet*. Contohnya di *stackoverflow*, komunitas di sosial media maupun mencari langsung lewat *Google*.
4. Reputasi dan ranking yang bagus membuat developer membutuhkan orang yang menguasai *React*. Tentu saja ini peluang bagus bagi yang ingin berkecimpung dalam dunia pengembangan sistem.

1.2 Prasyarat Belajar React

Darimana harus mulai belajar *React*? *React* sangat menarik untuk dipelajari namun tetap ada beberapa hal yang harus diketahui sebelumnya. Untuk bisa belajar *React* setidaknya anda harus menguasai dasar-dasar:

1. **HTML** berfungsi untuk membuat tampilan UI.
2. **CSS** berfungsi untuk mengatur tampilan UI, warna, jenis huruf dan sebagainya.
3. **Javascript** bahasa pemrograman yang bekerja di sisi *client*, membuat *website* menjadi lebih interaktif.

Bagaimana jika prasyaratnya tersebut belum kita kuasai? Kalau memang tidak bisa sama sekali ada baiknya anda meluangkan beberapa waktu untuk mempelajari hal tersebut. *HTML* dan *CSS* dasar bisa anda pelajari 2 hari, *Javascript* dan *ES6* sekitar 1-2 minggu. Waktu pastinya sendiri bisa berbeda tergantung antusias dan motivasi belajar, bisa jadi lebih cepat dari itu.

Untuk materi keseluruhan pada buku ini, karena kita ingin melakukan kolaborasi dengan *PHP* dan *MySQL*, maka perlu juga menguasai materi tersebut minimal cara koneksi dan melakukan operasi CRUD (*Create*, *Read*, *Update*, dan *Delete*).

1.3 Kekurangan dan Kelebihan React

Setiap teknologi pasti ada kelemahan dan kelebihan. Tapi kelebihan yang diperoleh membuat orang akan mengabaikan kekurangan.

Kekurangan React:

1. View Layer

Karena *React* merupakan sebuah *library*, maka fokus utamanya adalah di View saja pada konsep pemrograman berorientasi object (OOP) MVC, sementara bagian *Model* dan *Controller* tidak disertakan pada saat project *React* dibuat pertama kali. Misalnya kita membutuhkan fungsi untuk routing, HTTP, maka harus mencari dan menambahkan *library* untuk fungsi tersebut.

2. Permasalahan Lisensi

Isu lisensi masih muncul sampai saat ini, tapi *Facebook* memperbolehkan menggunakan *React* untuk mengembangkan produk dengan catatan untuk tidak berkompetisi dengan dengan produk *Facebook*, tapi tidak masalah.

3. Keterbatasan Browser

Biar dapat berjalan dengan baik, *React* membutuhkan browser versi terbaru.

Kelebihan React:

1. *Component Based Architecture*

Untuk membangun aplikasi besar, dengan konsep **komponen** yang dimilikinya, kita bisa membagi UI *project* menjadi beberapa komponen, misalnya: Header, Navigasi, Konten dan Footer.

2. *Reuseable Component*

Sebuah Component bisa digunakan atau dipanggil berulang kali. Ini tentu saja membuat pekerjaan menjadi lebih efisien.

3. *Virtual DOM*.

Javascript karena bekerja pada sisi *client* bekerja lebih cepat. Namun penanganan DOM lumayan memakan proses dan waktu. Dengan virtual DOM, Pembaruan DOM ditangani dengan halus di *React*. Dengan kemampuan ini, *React* hanya melakukan render ulang pada DOM yang mengalami perubahan, sehingga aplikasi menjadi lebih cepat.

4. JSX.

JSX merupakan singkatan dari *Javascript XML*. Dengan JSX kita bisa menulis *tag HTML* di *Javascript* sebagaimana anda membuat *tag HTML* seperti biasa.

5. Declarative

Dalam dunia pemrograman dikenal istilah *deklaratif* dan *imperatif*. Analogi dari *deklaratif* dan *imperatif* dapat dilihat pada tabel berikut:

No	Perbandingan	Declaratif	Imperatif
1	Memperbaiki Komputer	Teknisi tolong cek Komputer saya. Teknisi atas kebijakan dan SOP yang berlaku melakukan permintaan tsb untuk menangani Komputer tsb sampai selesai.	1. Tolong ke t4 servis komputer 2. Cek Komputer Saya 3. Kalau rusak, tlg diperbaiki 4. Teknisi memeriksa Komputer 5. Teknisi memperbaiki Komputer 6. Komputer diperbaiki.
2	Contoh Kode	<pre>angka=[10, 20, 30, 40] print(sum(angka))</pre>	<pre>angka = [10, 20, 30, 40] total = 0 for angka in angka: total += angka print(total)</pre>

Dari tabel tersebut bisa kita simpulkan bahwa dalam *declarative* kita cuma melakukan perintah, kita tidak perlu mengetahui cara dan apa yang digunakan dalam melakukan perintah tersebut. Sementara pada *imperative*, cara dan apa yang digunakan kita yang menentukan.

1.4 Struktur React

Ketika membuat *project React* pertama kali, secara default akan tersedia 3 buah *folder* utama, yaitu:

1. **node_modules** berisi modul utama dari *nodeJS* dan *React* itu sendiri. Jika kita akan menambah *library* baru seperti *Routing*, *HTTP*, *Bootstap* dan sebagainya, maka otomatis ditempatkan pada folder ini.

2. **public** berisi file tambahan untuk mengatur tampilan misalnya file css, gambar, icon dan sebagainya.
3. **src** di folder inilah tempat menuliskan dan menyimpan kode *Javascript* dan komponen yang kita buat.

1.5 Software Wajib dan Tambahan

Software wajib yang harus ada untuk materi pembelajaran buku ini tersedia di *internet* dan kabar baiknya lagi semua *open source* alias gratis. Berikut *software* wajib yang harus ada:

- **React: NodeJS** (<https://nodejs.org>). Pada saat buku ini ditulis versi terbarunya adalah 15.6.0.
- **AMP: Apache, MySQL, PHP**. Untuk memudahkan pengguna, tersedia paket yang berisi keseluruhan programnya tanpa harus *install* satu per satu, salah satunya XAMPP (<https://www.apachefriends.org/index.html>).
- **Browser** untuk menampilkan UI yang kita buat. *Browser* yang dipakai disarankan *Google Chrome*, untuk memudahkan dalam melakukan *debug*.

Untuk mempercepat mengerjakan *project*, ada beberapa *software* tambahan yang sangat membantu produktivitas anda dalam bekerja, diantaranya:

- **Visual Code**: salah satu code editor yang banyak digunakan oleh *developer* saat ini.
- **phpMyAdmin**: memudahkan pengelolaan database *MySQL*. *PhpMyAdmin* sendiri sudah otomatis ada dalam *XAMPP*. Jika anda meng-*install* *Apache*, *Php*, *MySQL* secara terpisah, maka anda harus mengunduhnya dari <https://www.phpmyadmin.net/>.

1.6 Install React

React berjalan lewat *NodeJS*, sehingga kita harus melakukan *install NodeJS* terlebih dahulu. *NodeJS* berjalan di berbagai sistem operasi. Pada materi ini kita akan membahas cara *install NodeJS* untuk sistem operasi *Windows* dan *Linux* distro *Ubuntu*. Sistem operasi lain, bisa anda ikuti petunjuknya langsung dari situs *NodeJS*. Lihat gambar 1.1.

Downloads

Latest Current Version: 15.6.0 (includes npm 7.4.0)

Download the Node.js source code or a pre-built installer for your platform, and start developing today.

LTS Recommended For Most Users	Current Latest Features	
 Windows Installer <small>node-v15.6.0-x64.msi</small>	 macOS Installer <small>node-v15.6.0.pkg</small>	 Source Code <small>node-v15.6.0.tar.gz</small>
Windows Installer (.msi)	32-bit	64-bit
Windows Binary (.zip)	32-bit	64-bit
macOS Installer (.pkg)	64-bit	
macOS Binary (.tar.gz)	64-bit	
Linux Binaries (x64)	64-bit	

Gambar 1.1 Website NodeJS

Install NodeJS di Windows

Download paket *installer* di situs <https://nodejs.org/>. Pada saat buku ini ditulis versi terbarunya adalah 15.6.0. Jalankan program *installer* sampai selesai.

Install NodeJs di Ubuntu

```
sudo apt update
sudo apt install nodejs
```

Untuk sistem operasi lain, silahkan lihat petunjuk di website *NodeJS*. Memastikan apakah *NodeJS* sudah ter-*install*, masuk ke *command line* dan ketikkan perintah:

```
node -v
npm -v
```

Perintah **node -v** adalah untuk mengecek versi *NodeJS*, sedangkan **npm -v** adalah perintah untuk mengecek veris *NPM*. *NPM* sesuai namanya *Node Package Manager* yang sudah otomatis terinstall ketika anda menginstall *NodeJS*. *NPM* nantinya akan berfungsi untuk menginstall modul-modul *NodeJS*. Setelah dua perintah tersebut, di layar akan muncul versi *NodeJS* dan *NPM*. Lihat gambar 1.2.

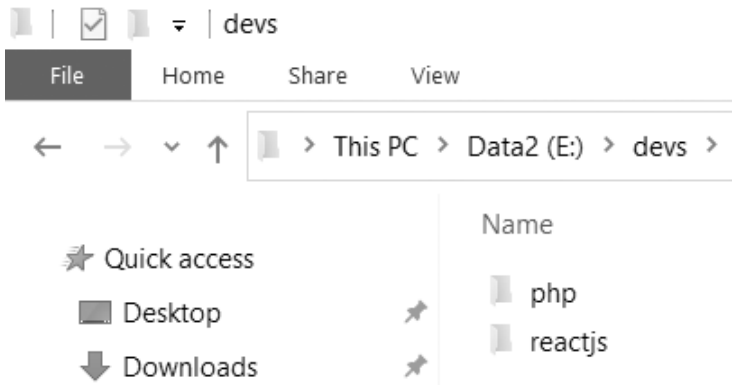
```
C:\ Command Prompt
D:\devs>node -v
v15.6.0
D:\devs>npm -v
7.4.0
```

Gambar 1.2 Cek versi NodeJS

Lokasi Folder React

Dimana saya harus menaruh file *project React*? Bebas, *project React* pada sembarang lokasi. Namun, untuk kemudahan dalam pengelolaan dan pengembangan kita buat folder khusus untuk menyimpan semua *project React* misalnya dalam *folder* dengan nama **reactjs** atau nama lain yang Anda inginkan.

Sebagai panduan saja, berikut ini contoh dalam manajemen folder dalam memuat kolaborasi *React* dan *PHP MySQL*. Untuk folder utama kita beri nama dengan **devs**. Dalam folder **devs** kita buat folder lagi dengan nama **php** untuk menaruh *project PHP*, dan **reactjs** untuk menaruh *project React*. Sehingga struktur folder akan terlihat seperti pada gambar 1.3.



Gambar 1.3 Manajemen folder aplikasi

Khusus untuk *PHP* agar file bisa langsung di eksekusi, jika kita menggunakan *XAMPP* maka file *PHP* harus ditaruh pada folder `C:\xampp\htdocs`. Agar kita bisa mengeksekusi file *PHP* pada lokasi folder tertentu, kita bisa melakukan konfigurasi pada file `C:\xampp\apache\conf\httpd.conf` dengan mengubah bagian pada *DocumentRoot* dan *Directory* menjadi:

```
DocumentRoot "D:/devs/php"  
<Directory "D:/devs/php">
```

Setelah manajemen folder sudah beres, sekarang kita akan meng-*install React*. Untuk membuat *project React*, pastikan kita sudah terhubung ke internet, karena modul-modul yang disertakan di ambil langsung dari *internet*. Masuk ke command line atau terminal dan ketikkan perintah:

```
npm install -g create-react-app
```

Perintah tersebut adalah meng-*install* modul *NodeJS* dengan nama **create-react-app**. Dengan **create-react-app** membantu kita untuk membuat *project React* dengan modul dasar yang diperlukan. Modul dasar pendukung tidak perlu kita buat secara manual satu per satu.

Perintah **create-react-app** ini cukup anda install sekali saja di awal pembuatan *project React*. Untuk *project* berikutnya perintah tersebut tidak perlu lagi kecuali ada pemberitahuan untuk updating module.

Sekarang anda masuk ke *folder* di mana anda mau menaruh *project React* yang akan dibuat. Sesuai gambar 1.4, kita akan menyimpan di `E:\devs\reactjs`. Pada *command line* ketikkan perintah:

```
npx create-react-app bab01_project_pertama
```

Untuk membuat *project React*, perintahnya adalah **npx create-react-app**. Kemudian perhatikan perintah yang ditebalkan, itu nama *project React* sekaligus akan menjadi nama *folder* baru tempat aplikasi *React*. Tunggu sampai proses selesai dan muncul tampilan seperti gambar 1.4.

Pesan di layar menyatakan *project* sudah selesai dibuat dan dalam folder `E:\devs\reactjs` sudah terdapat *folder* baru bernama **bab01_project_pertama**. Masih di command line, sesuai petunjuk pesan tersebut, kita akan pindah ke folder baru tersebut kemudian menjalankan aplikasi dengan perintah **npm start**.

```
cd bab01_project_pertama  
npm start
```

Success! Created bab01_project_pertama at D:\devs\reactjs\bab01_project_pertama
 Inside that directory, you can run several commands:

```
npm start
  Starts the development server.

npm run build
  Bundles the app into static files for production.

npm test
  Starts the test runner.

npm run eject
  Removes this tool and copies build dependencies, configuration files
  and scripts into the app directory. If you do this, you can't go back!
```

We suggest that you begin by typing:

```
cd bab01_project_pertama
npm start
```

Happy hacking!

```
PS D:\devs\reactjs> █
```

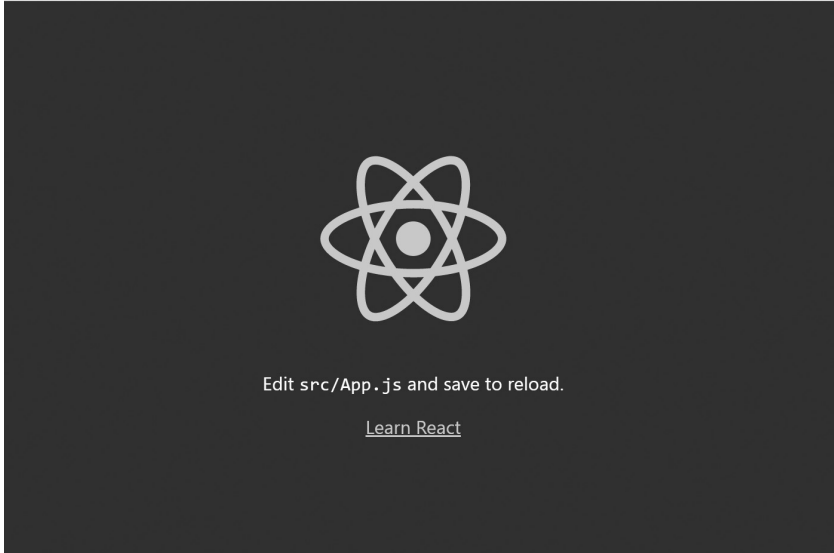
Gambar 1.4 Tampilan Project React sukses dibuat

NodeJS akan melakukan kompilasi sampai sukses, akan muncul pesan berikut:

```
Compiled successfully!
You can now view project-react-pertama in the browser.
Local:          http://localhost:3000
On Your Network: http://192.168.1.92:3000
Note that the development build is not optimized.
To create a production build, use npm run build.
```

Pesan tersebut menyampaikan informasi bahwa proses kompilasi berhasil. Untuk melihat hasilnya anda bisa melihat di browser dengan mengetikkan URL: <http://localhost:3000> atau <http://192.168.1.92:3000>. Lihat gambar 1.5.

Jika muncul seperti gambar 1.5, maka kita sudah berhasil membuat *project React* pertama kali.

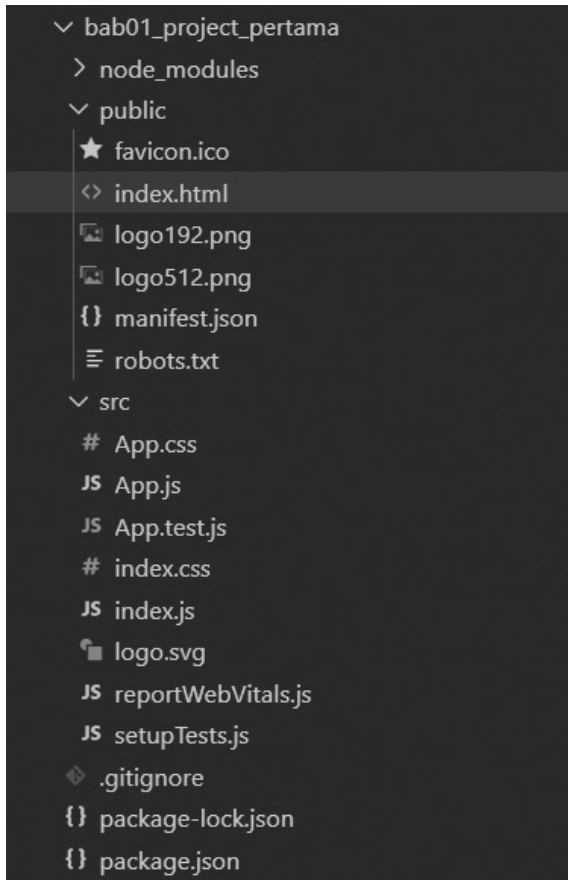


Gambar 1.5 Project React pertama

1.7 Prinsip Kerja React

Setelah project React pertama berhasil dibuat, pertanyaanya bagaimana tampilan tersebut dibuat? Oke, sekarang kita lihat struktur *project React* yang sudah kita buat. Lihat gambar 1.6.

Saat dijalankan pertama kali, *project React* akan mengakses *file src/index.js* dan *file HTML* yang menjadi dasar *template* UI adalah **public/index.html**. Dalam pengembangan kedepannya, kita tidak akan teralu banyak atau bahkan tanpa melakukan perubahan pada *file src/index.js*, karena kita akan bekerja dengan komponen. Sedangkan pada *file public/index.html* kita masih bisa melakukan modifikasi dengan menambahkan *content delivery network* (CDN) seperti *Bootstrap*, *icon*, *meta tags* dan sebagainya sepanjang tidak mengurangi *template* awal. Sekarang kita lihat code file **src/index.js** berikut:



Gambar 1.6 Struktur project React

```
1. import React from 'react';
2. import ReactDOM from 'react-dom';
3. import './index.css';
4. import App from './App';
5. import reportWebVitals from './reportWebVitals';
6.
7. ReactDOM.render(
8.   <React.StrictMode>
9.     <App />
```

```

10.     </React.StrictMode>,
11.     document.getElementById('root')
12.   );
13.
14.   // If you want to start measuring performance, pass a function
15.   // to log results (for example: reportWebVitals (console.log))
16.   // or send to an analytics. Learn at https://bit.ly/CRA-vitals
17.   reportWebVitals();

```

Penjelasan:

- **Baris 1-5.** Melakukan import terhadap library dan file css.
- **Baris 4.** Melakukan import komponen **App**. komponen **App** adalah komponen parent atau komponen utama yang akan kita gunakan untuk membangun project yang terdiri dari banyak komponen. Pembahasan tentang komponen akan kita bahas pada bab khusus.
- **Baris 7-12.** Melakukan perintah render. Konsep dasar dari perintah render adalah sebagai berikut:

```

ReactDOM.render(Apa yang mau di-render, Di mana mau me-render );
ReactDOM.render(Elemen HTML, Lokasi );

```

Apa yang mau di-render adalah elemen HTML misalnya: text, gambar, video atau bahkan komponen.

Dimana mau me-render: lokasi tempat me-render, misalnya pada elemen div, span dengan atribut id atau class tertentu.

Pada contoh file `src/index.js` diatas yang mau di-render adalah `<React.StrictMode><App /></React.StrictMode>`. Sedangkan lokasi tempat di-render adalah `document.getElementById('root')`, elemen HTML yang memiliki atribut id `root`.

Dimana elemen id `root`? Itu terletak di file template dasar `public/index.html`.

- **Baris 11-16.** Komentar pada React.
- **Baris 17.** Memanggil fungsi `reportWebVitals()`.

Sekarang kita lihat file `public/index.html`, khususnya pada bagian di antara tag `<body></body>`:

1. `<noscript>You need to enable JavaScript to run this app.</noscript>`
2. `<div id="root"></div>`

Penjelasan:

- **Baris 1.** Pesan yang akan ditampilkan jika Javascript tidak berfungsi.
- **Baris 2.** Sesuai perintah pada baris 7-12 di file `src/index.js`, di lokasi inilah kode HTML akan di-render.

Bagaimana, ada bayangan bukan? Ingat konsep dasarnya, **apa yang mau di-render dan dimana mau me-render**.

Kita lihat kembali code `src/index.js` pada Baris 9, ada perintah `<App />`. Perintah tersebut artinya memanggil komponen dengan nama `App`. Sebelum dipanggil komponen harus terlebih dulu di-import oleh komponen lain yang menggunakan seperti perintah pada code `src/index.js` di baris 4: `import App`. Komponen `App` isinya sebagai berikut:

```
1.   import logo from './logo.svg';
2.   import './App.css';
3.
4.   function App() {
5.     return (
6.       <div className="App">
7.         <header className="App-header">
8.           <img src={logo} className="App-logo" alt="logo" />
9.           <p>
10.            Edit <code>src/App.js</code> and save to reload.
11.          </p>
12.          <a
13.            className="App-link"
14.            href="https://reactjs.org"
15.            target="_blank"
16.            rel="noopener noreferrer"
17.          >
18.            Learn React
19.          </a>
20.        </header>
21.      </div>
22.    );
23.  }
24.
25.  export default App;
```

Tampilan pada gambar 1.6 adalah sesuatu yang dibuat dalam komponen `App`.

Untuk ujicoba, silahkan ganti pesan pada Baris 10, dan simpan. Tanpa refresh *browser*, maka tampilan otomatis berubah. Itulah salah satu kelebihan *React*.