

# **PENGENALAN REACTJS**

---



# APA ITU REACTJS

---

- React Js adalah sebuah library JavaScript yang di buat oleh facebook. React bukanlah sebuah framework MVC.
- React adalah library yang bersifat composable user interface, yang artinya kita dapat membuat berbagai UI yang bisa kita bagi menjadi beberapa komponen.

# KEUNGGULAN REACTJS

---

- **1. Cepat dan Efisien**

Karena berbasis komponen maka react hanya perlu me-render resource yang berhubungan dengan data yang berganti, tidak perlu me-render seluruh resource .

- **2. Reusable (dapat digunakan berulang kali)**

Komponen yang telah kita buat dapat kita gunakan berkali-kali pada saat dibutuhkan. Ini sangat berguna bagi kita untuk mempersingkat waktu dan mengurangi resource yang ada.

- **3. Library JavaScript**

**JSX** (JavaScript Extension) singkatnya kita dapat menyematkan syntax HTML kedalam Javascript. Ini sangat membantu kita dalam proses development, apalagi dengan adanya fungsi dari **ES6** (Ecma Script).

- **4. Immutable State**

Kita dapat manajemen state yang ada dengan menggunakan **Redux**. Kita dapat mengatasi permasalahan mutable state dengan **RamdaJs**. Untuk state yang berinteraksi dengan API kita dapat menggunakan **Redux-Saga**.

# COMPONENT PADA REACT

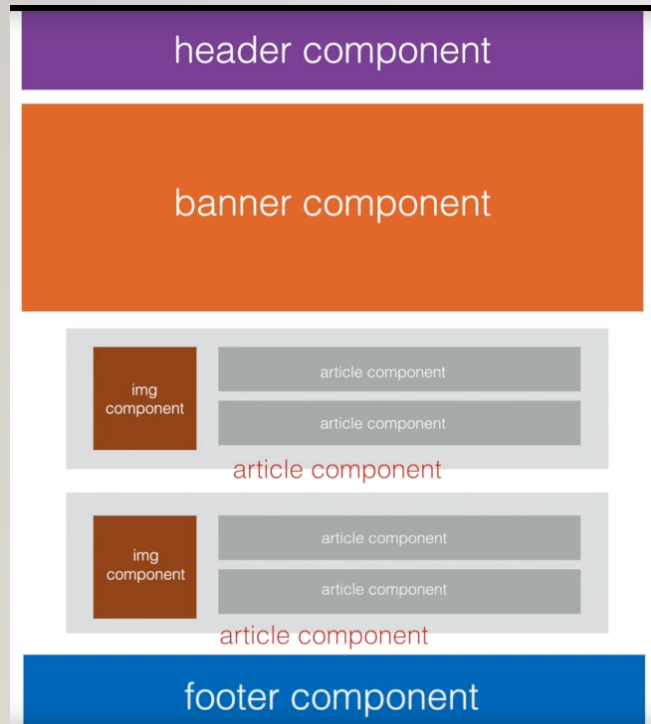
---

- React menganggap setiap elemen UI sebagai komponen jadi bisa di analogikan sama dengan membuat lego yang artinya untuk membuat sebuah UI kita akan memasang component yang ada dan menggabungkannya agar menjadi sebuah website.
- Dan setiap komponen diharapkan hanya mempunyai satu tanggung jawab. Jika komponen tersebut berupa tombol maka tugasnya hanya bisa ditekan dan berteriak, “aku ditekan”. Oke. Boleh saja dia diam — itu hak asasi tombol.
- Jika komponen hanya mempunyai satu tanggung jawab, kita bisa memanfaatkan komponen berulang kali (*reusable*) dan bisa juga menggabungkan satu komponen dengan lainnya (*composable*).



# CONTOH COMPONENT PADA REACT

---



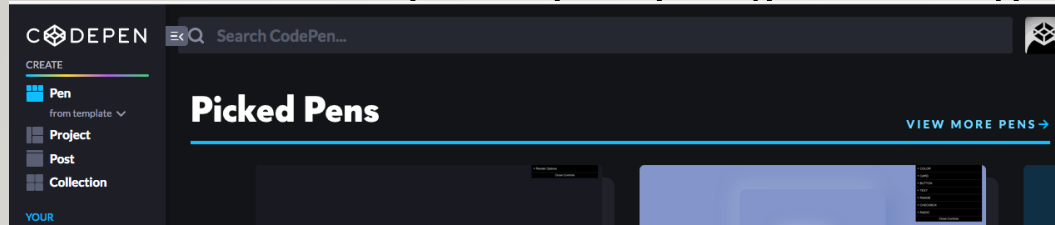
Ada beberapa component yg berbeda-beda peran dan tanggung jawabnya  
Antara lain

- Header component
- Banner component
- Article component
- Footer component

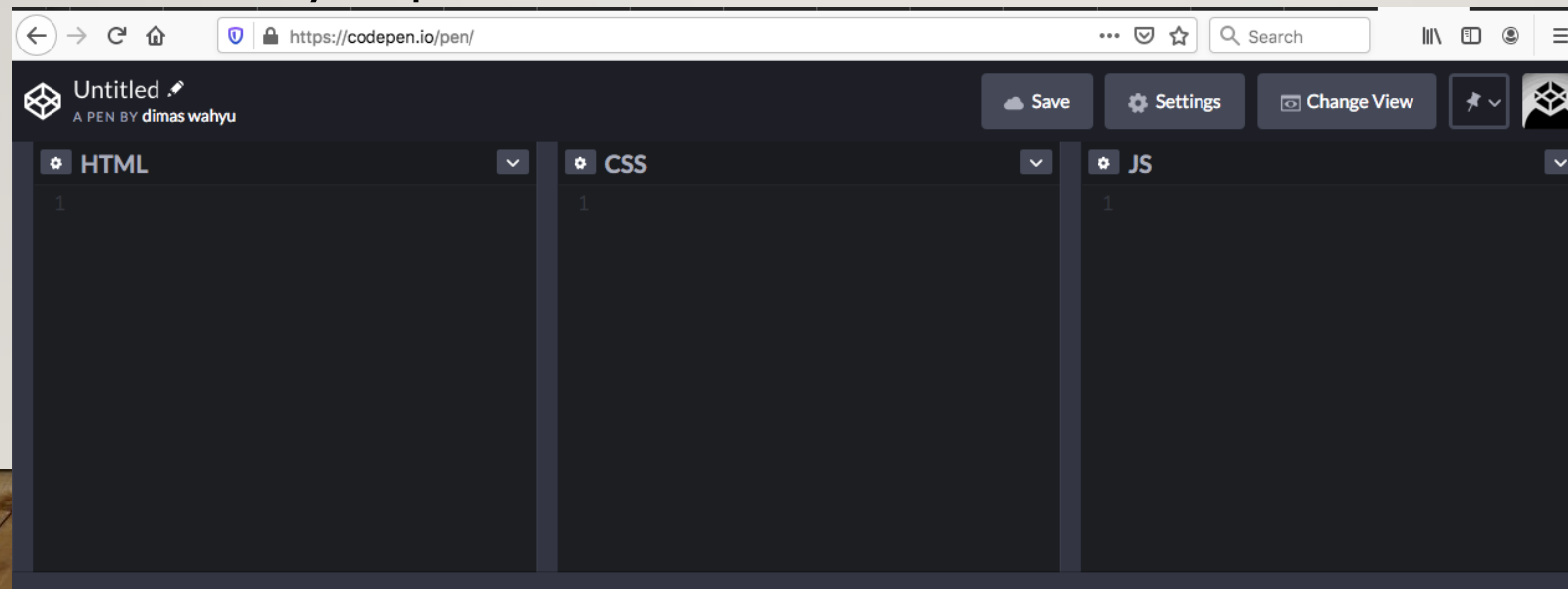
# PEMAHAMAN TENTANG COMPONENT

---

- Buka halaman codepen.io pilih pen (pastikan login terlebih dahulu)

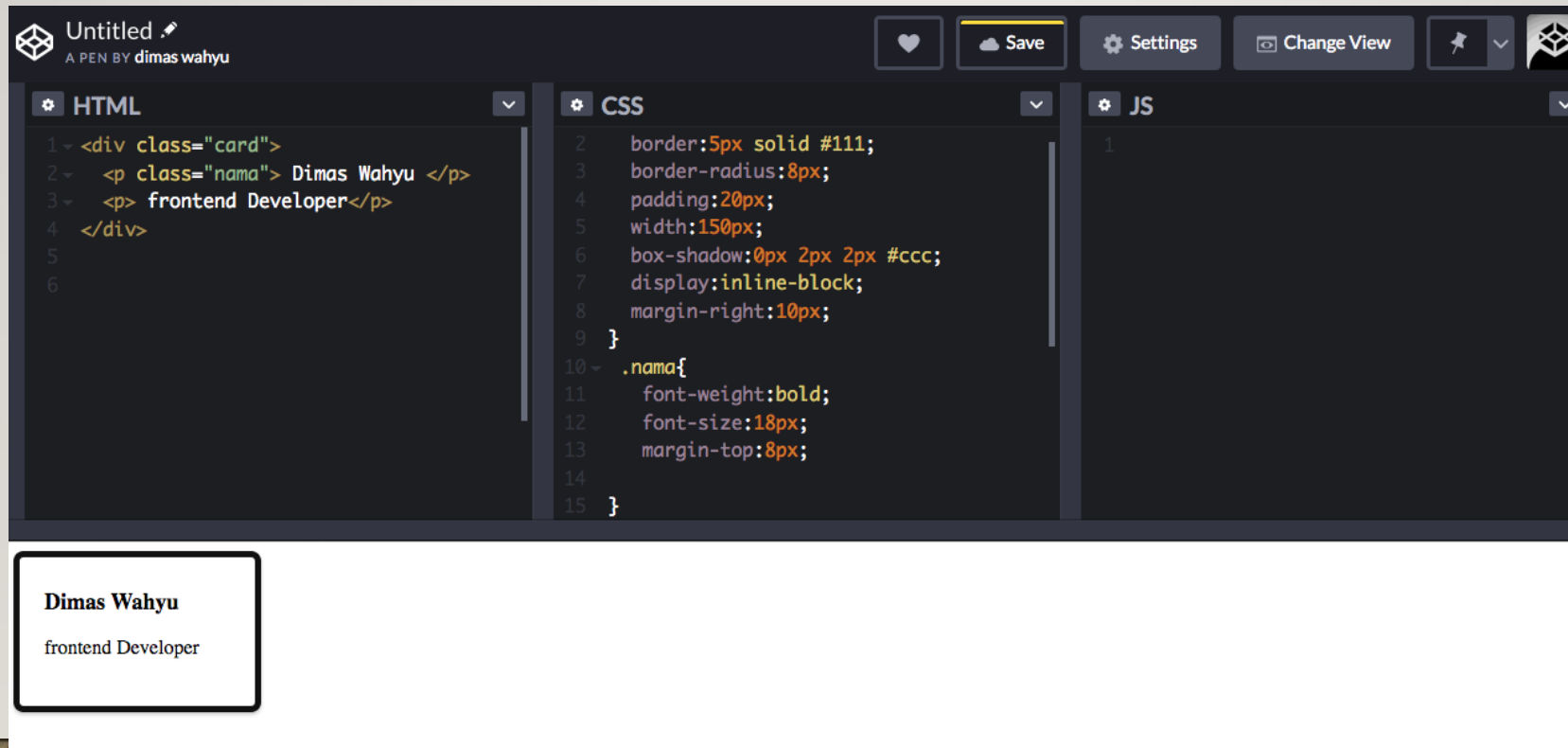


- Maka hasilnya seperti berikut



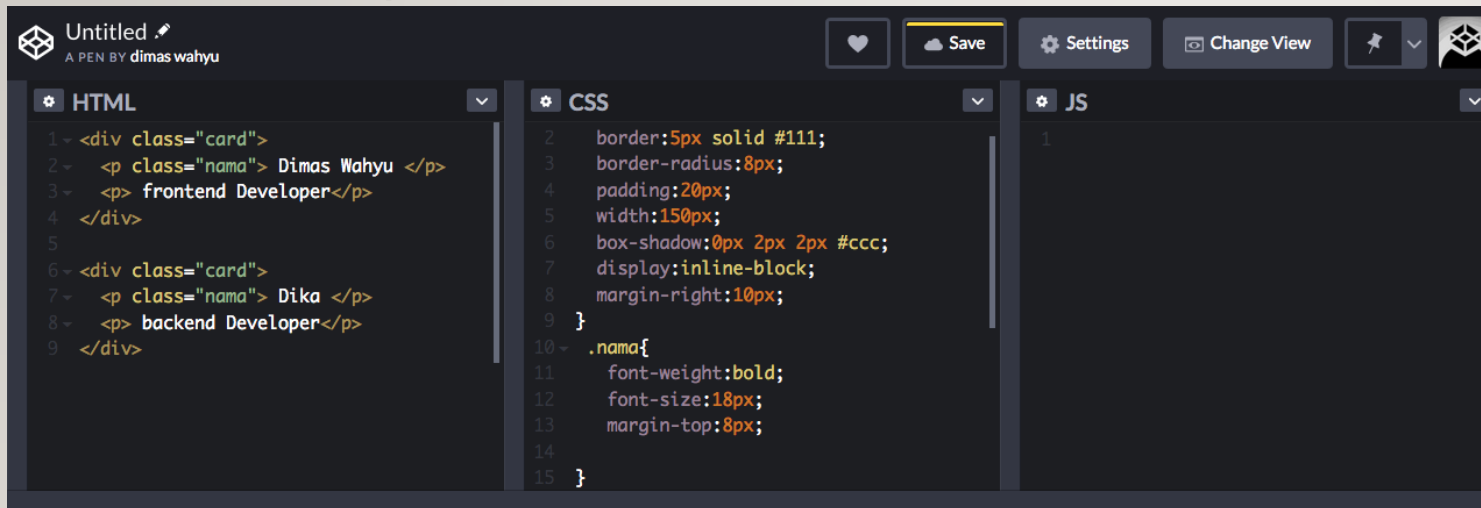
# PEMAHAMAN TENTANG COMPONENT

- Buatlah seperti berikut



# PEMAHAMAN TENTANG COMPONENT

- Dari code diatas jika kita ingin menambahkan card lagi maka kita harus copy paste code seperti berikut



The screenshot shows a code editor with three panels: HTML, CSS, and JS. The HTML panel contains two `<div class="card">` blocks. The first block contains a `<p class="nama"> Dimas Wahyu </p>` and `<p> frontend Developer</p>`. The second block contains a `<p class="nama"> Dika </p>` and `<p> backend Developer</p>`. The CSS panel shows styles for the `.card` and `.nama` classes. The `.card` class has styles for border, border-radius, padding, width, box-shadow, display, and margin-right. The `.nama` class has styles for font-weight, font-size, and margin-top. The JS panel is empty.

```
HTML
1 <div class="card">
2   <p class="nama"> Dimas Wahyu </p>
3   <p> frontend Developer</p>
4 </div>
5
6 <div class="card">
7   <p class="nama"> Dika </p>
8   <p> backend Developer</p>
9 </div>

CSS
2 border:5px solid #111;
3 border-radius:8px;
4 padding:20px;
5 width:150px;
6 box-shadow:0px 2px 2px #ccc;
7 display:inline-block;
8 margin-right:10px;
9 }
10 .nama{
11   font-weight:bold;
12   font-size:18px;
13   margin-top:8px;
14 }
15 }
```

- pada akhirnya jika kita harus membuat card sejumlah 100 maka kita harus copy dan paste 1000 code yang sama

Dimas Wahyu

frontend Developer

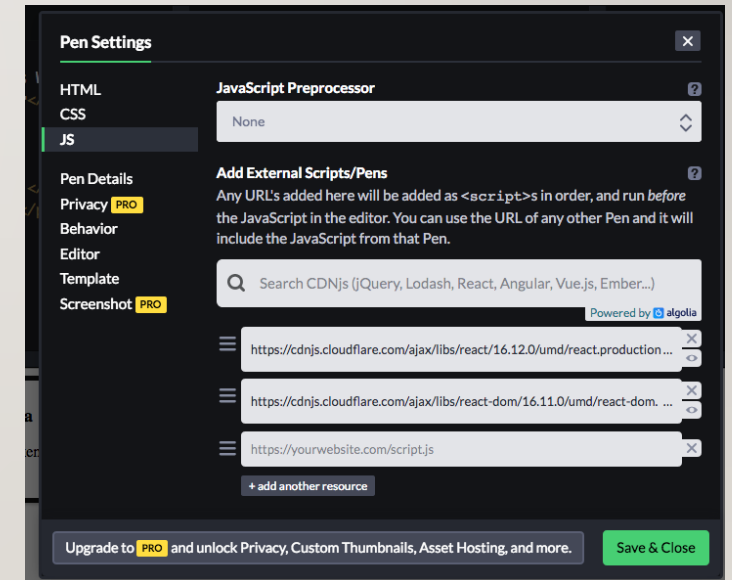
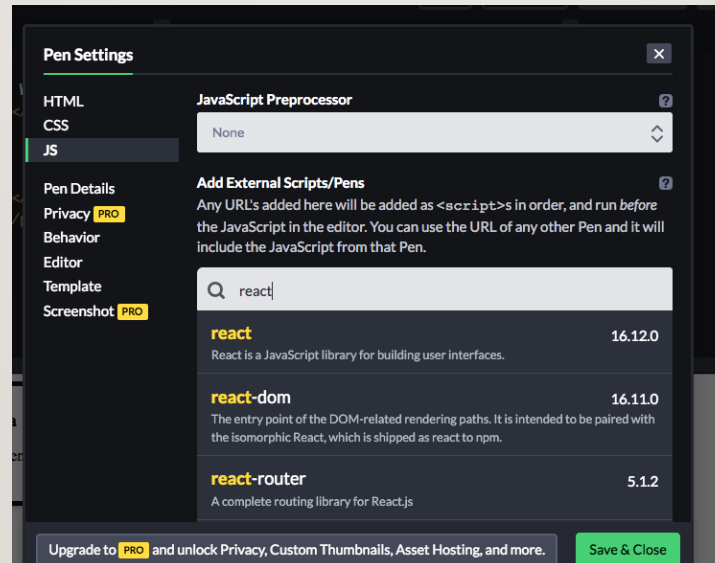
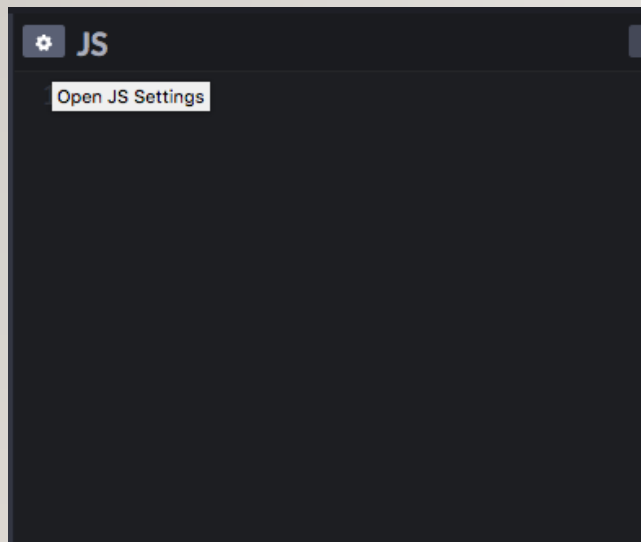
Dika

backend Developer



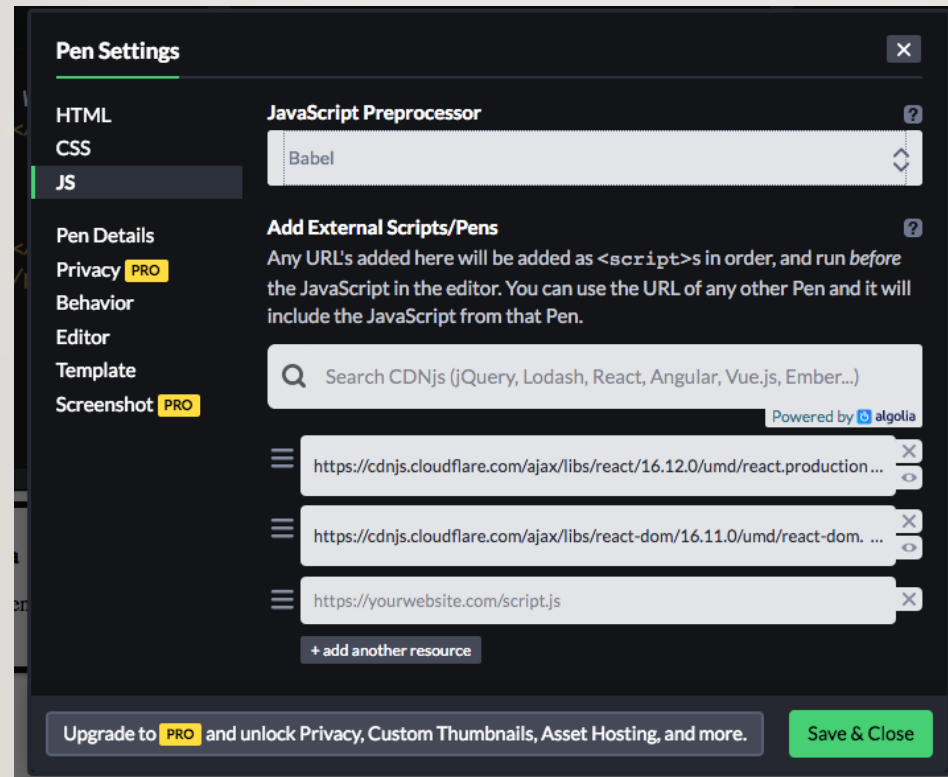
# PEMAHAMAN TENTANG COMPONENT

- Untuk mengatasi seperti hal diatas maka kita dapat menggunakan library reactjs
- Tambahkan library react dan react-dom



# PEMAHAMAN TENTANG COMPONENT

- Dikarenakan tidak semua browser support dengan react maka kita tambahkan babel pada preprosecor



# PEMAHAMAN TENTANG COMPONENT

---

- Selain cara menggunakan cara diatas kita juga dapat juga menampilkan card menggunakan javascript
- Tambahkan code pada javascript dan html



The screenshot shows a code editor with two panels. The left panel is labeled 'HTML' and contains the following code:

```
1 <div id="root"></div>
2 |
3 <div class="card">
4   <p class="nama"> Dimas Wahyu </p>
5   <p> frontend Developer</p>
6 </div>
7
8 <div class="card">
9   <p class="nama"> Dika </p>
10  <p> backend Developer</p>
11 </div>
```

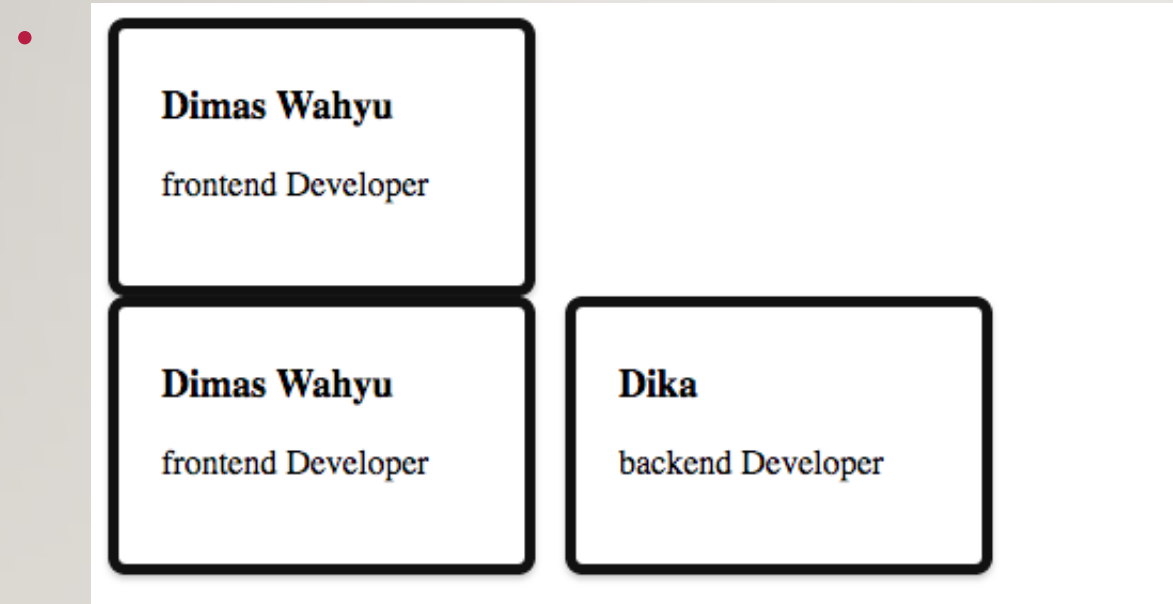
The right panel is labeled 'JS (Babel)' and contains the following code:

```
1 function Card(){
2   return(
3     <div class="card">
4       <p class="nama"> Dimas Wahyu </p>
5       <p> frontend Developer</p>
6     </div>
7   )
8 }
9
10 ReactDOM.render(<Card />,
11   document.querySelector('#root'))
```

# PEMAHAMAN TENTANG COMPONENT

---

- Hasilnya



# PEMAHAMAN TENTANG COMPONENT

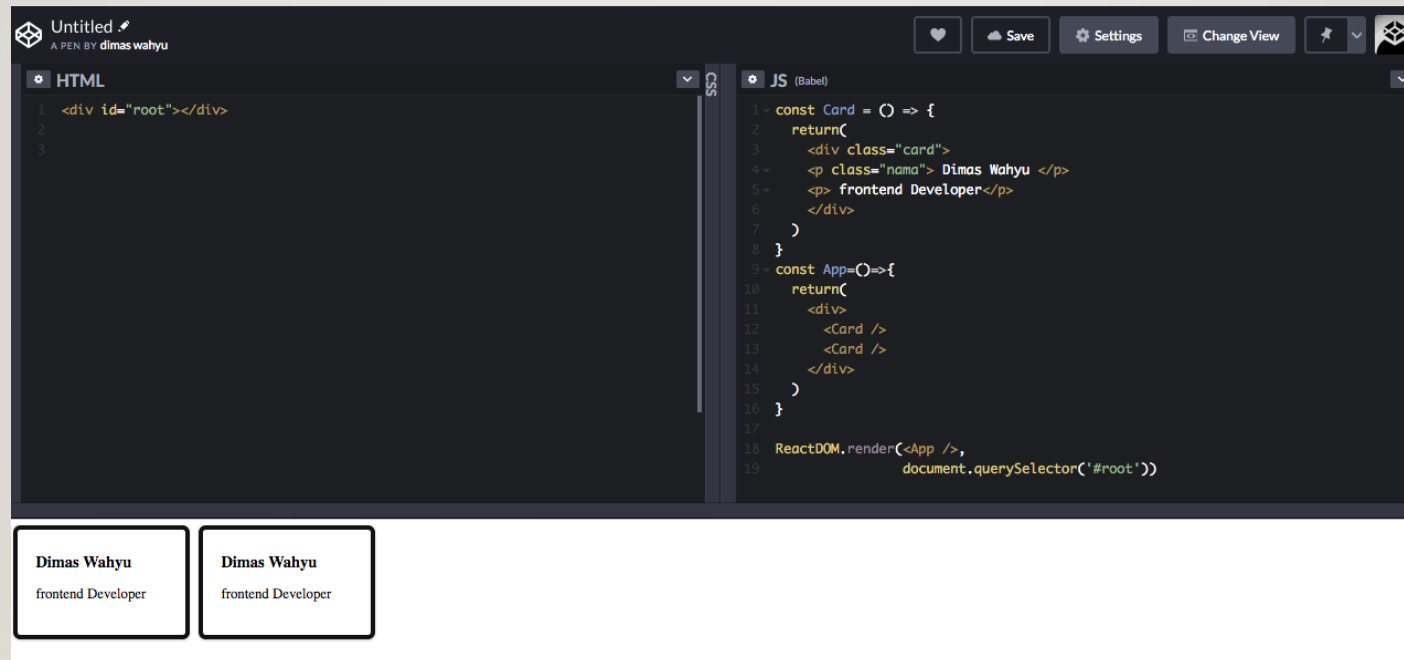
---

- Keterangan
  - Kegunaan ReactDOM di sini cukup jelas, yaitu *me-render* sesuatu ke sebuah HTML element yang memiliki `id=root`. Coba lihat code html, pada baris ke-1 akan Anda temukan `<div id="root"></div>`



# PEMAHAMAN TENTANG COMPONENT

- Untuk mempersingkat code pada HTML maka kita bisa menggunakan javascript seperti berikut



The screenshot shows a code editor with two panels. The left panel is labeled 'HTML' and contains the following code:

```
1 <div id="root"></div>
2
3
```

The right panel is labeled 'JS (Babel)' and contains the following code:

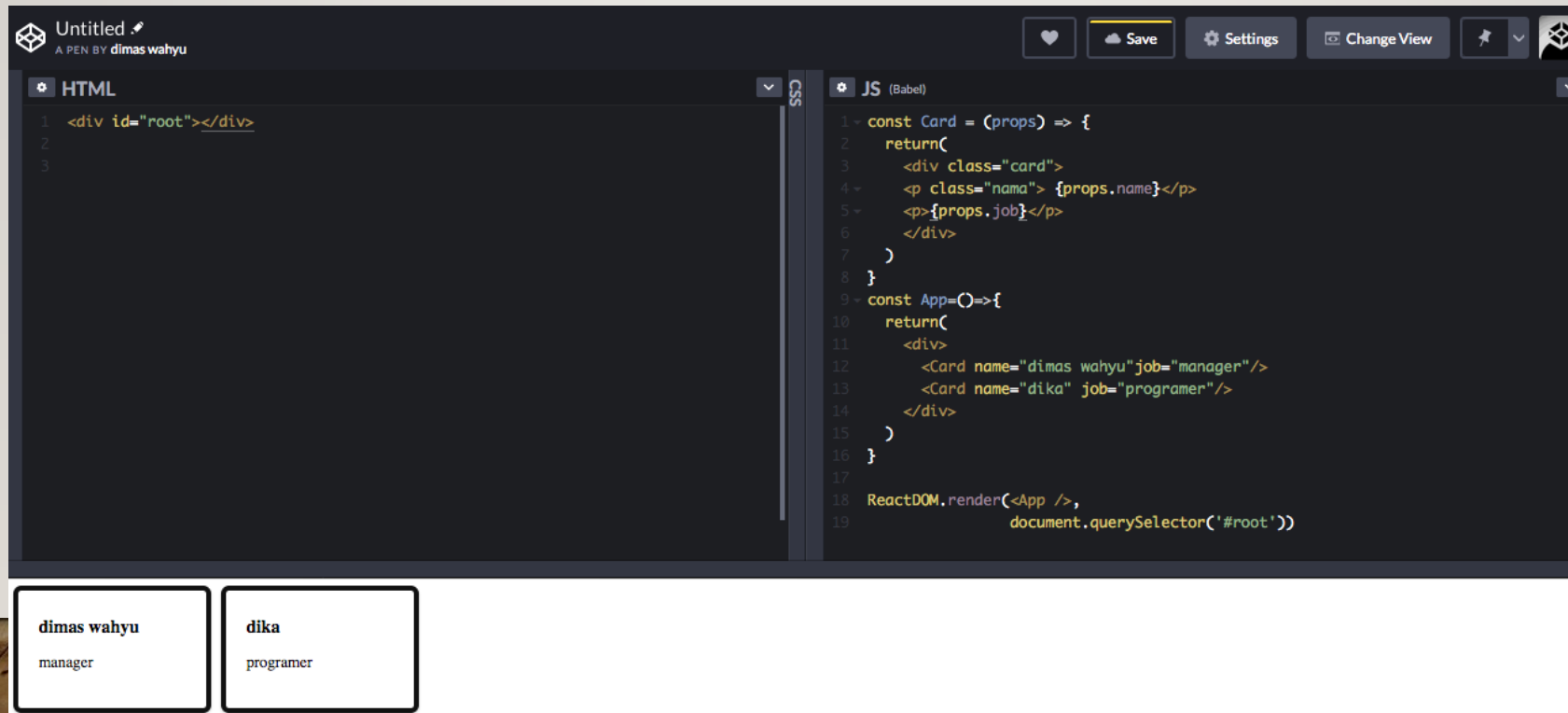
```
1 const Card = () => {
2   return(
3     <div class="card">
4       <p class="nama"> Dimas Wahyu </p>
5       <p> frontend Developer</p>
6     </div>
7   )
8 }
9
10 const App = () => {
11   return(
12     <div>
13       <Card />
14     </div>
15   )
16 }
17
18 ReactDOM.render(<App />,
19   document.querySelector('#root'))
```

Below the code editor, there are two identical boxes, each containing the text:

Dimas Wahyu  
frontend Developer

# PEMAHAMAN TENTANG COMPONENT

- Jika kita lihat hasilnya masih tidak dinamis, nama masih sama
- Untuk itu kita tambahkan props seperti pada code berikut



```
Untitled
A PEN BY dimas wahyu

HTML
1 <div id="root"></div>
2
3

CSS

JS (Babel)
1 const Card = (props) => {
2   return(
3     <div class="card">
4       <p class="nama"> {props.name}</p>
5       <p>{props.job}</p>
6     </div>
7   )
8 }
9 const App=()=>{
10  return(
11    <div>
12      <Card name="dimas wahyu"job="manager"/>
13      <Card name="dika" job="programer"/>
14    </div>
15  )
16 }
17
18 ReactDOM.render(<App />,
19   document.querySelector('#root'))
```

dimas wahyu  
manager

dika  
programer

# PEMAHAMAN TENTANG COMPONENT

---

- Jadi yang dimaksud dari component pada uji coba ini adalah card

```
const App=()=>{  
  return(  
    <div>  
      <Card name="dimas wahyu" job="manager"/>  
      <Card name="dika" job="programer"/>  
    </div>  
  )  
}
```

# MENAMBAHKAN REACT PADA WEBSITE

---

- Selain menggunakan CLI kita juga dapat menambahkan react pada website
- Langkahnya sebagai berikut
  - Buatlah folder belajar-react/pertemuan-kedua/web-react
  - Buat file index.html
  - Setelah itu buat file js dengan nama like\_button.js (tulisan nya sesuaikan dengan index.html line 19) seperti pada gambar setelah ini

# MENAMBAHKAN REACT PADA WEBSITE

<> index.html X JS like\_button.js

pertemuan-kedua > web-react > <> index.html > html

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4    <meta charset="UTF-8">
5    <meta name="viewport" content="width=device-width, initial-scale=1.0">
6    <!-- ... other HTML ... -->
7    <title>Belajar react</title>
8  </head>
9  <body>
10   <p>Hello word</p>
11   <div id="like_button_container"></div>
12
13   <!-- Load React. -->
14   <!-- Note: when deploying, replace "development.js" with "production.min.js". -->
15   <script src="https://unpkg.com/react@16/umd/react.development.js" crossorigin></script>
16   <script src="https://unpkg.com/react-dom@16/umd/react-dom.development.js" crossorigin></script>
17
18   <!-- Load our React component. -->
19   <script src="like_button.js"></script>
20 </body>
21 </html>
```



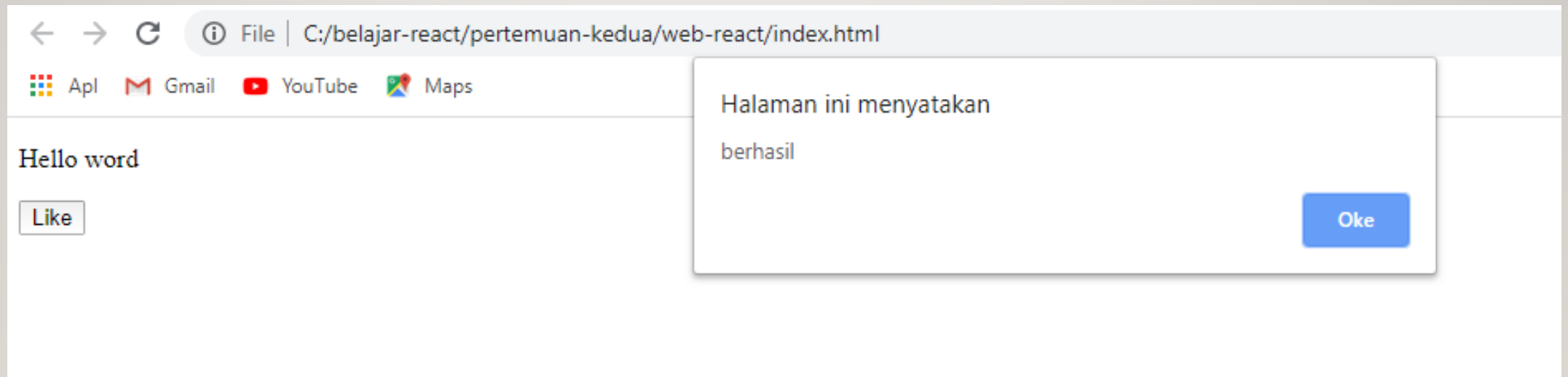
# MENAMBAHKAN REACT PADA WEBSITE

```
<> index.html JS like_button.js X
pertemuan-kedua > web-react > JS like_button.js > LikeButton
1 // ... the starter code you pasted ...
2 const e = React.createElement;
3
4 function LikeButton(){
5 // Display a "Like" <button>
6 return e(
7   'button',
8   {
9     onClick: () => alert('berhasil')
10  },
11  'Like'
12 );
13
14
15 }
16
17 // const button={()=>{
18 //   return <button>Like</button>
19 // }
20
21 const domContainer = document.querySelector('#like_button_container');
22 ReactDOM.render(e(LikeButton), domContainer);
```

# MENAMBAHKAN REACT PADA WEBSITE

---

Hasilnya



# MEMBUAT PROJECT MENGGUNAKAN REACT-APP

- <https://reactjs.org/docs/create-a-new-react-app.html>
- Buat project dengan nama belajar-react-app

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  1: node  +  [ ]  [ ]  ^  x

(base) Dimass-MacBook-Pro:belajar-react-app dimaswahu$ npx create-react-app my-app
npx: installed 99 in 7.753s

Creating a new React app in /Users/dimaswahu/belajar-react/Pemograman_framework/pertemuan-kedua/belajar-react-app/my-app.

warning ../../../../../../package.json: No license field
Installing packages. This might take a couple of minutes.
Installing react, react-dom, and react-scripts with cra-template...

yarn add v1.22.0
warning ../../../../../../package.json: No license field
[1/4]  Resolving packages...
[2/4]  Fetching packages...
[#####-----] 1081/1356
```

# MEMBUAT PROJECT MENGGUNAKAN REACT-APP

---

- Hasil install react-app

```
yarn remove v1.22.0
warning ../../../../package.json: No license field
[1/2] Removing module cra-template...
[2/2] Regenerating lockfile and installing missing dependencies...
warning " > @testing-library/user-event@7.2.1" has unmet peer dependency "@testing-library/dom@>=5".
warning "react-scripts > eslint-config-react-app@5.2.0" has incorrect peer dependency "eslint-plugin-flowtype@3.x".
warning "react-scripts > @typescript-eslint/eslint-plugin > tsutils@3.17.1" has unmet peer dependency "typescript@>=2.8.0 || >= 3.2.0-dev || >= 3.3.0-dev || >= 3.4.0-dev || >= 3.5.0-dev || >= 3.6.0-dev || >= 3.6.0-beta || >= 3.7.0-dev || >= 3.7.0-beta".
success Uninstalled packages.
✨ Done in 12.24s.

Created git commit.

Success! Created my-app at /Users/dimaswahyu/belajar-react/Pemograman_framework/pertemuan-kedua/belajar-react-app/my-app
Inside that directory, you can run several commands:

  yarn start
    Starts the development server.

  yarn build
    Bundles the app into static files for production.

  yarn test
    Starts the test runner.

  yarn 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 my-app
  yarn start

Happy hacking!
(base) Dimass-MacBook-Pro:belajar-react-app dimaswahyu$
```



# MEMBUAT PROJECT MENGGUNAKAN REACT-APP

- Untuk menjalankan app nya terlebih dulu masuk folder my-app
- Setelah itu jalankan menggunakan npm start atau menggunakan react-scripts start

```
package.json X
Pemograman_framework > pertemuan-kedua > belajar-react-app > my-app > {}
12 },
13 "scripts": {
14   "start": "react-scripts start",
15   "build": "react-scripts build",
16   "test": "react-scripts test",
17   "eject": "react-scripts eject"
18 },
19 "eslintConfig": {
20   "extends": "react-app"
21 },
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
(base) Dimass-MacBook-Pro:belajar-react-app dimaswahu$ cd my-app/
(base) Dimass-MacBook-Pro:my-app dimaswahu$ react-scripts start
```

```
package.json X
Pemograman_framework > pertemuan-kedua > belajar-react-app > my-app
12 },
13 "scripts": {
14   "start": "react-scripts start",
15   "build": "react-scripts build",
16   "test": "react-scripts test",
17   "eject": "react-scripts eject"
18 },
19 "eslintConfig": {
20   "extends": "react-app"
21 },
```

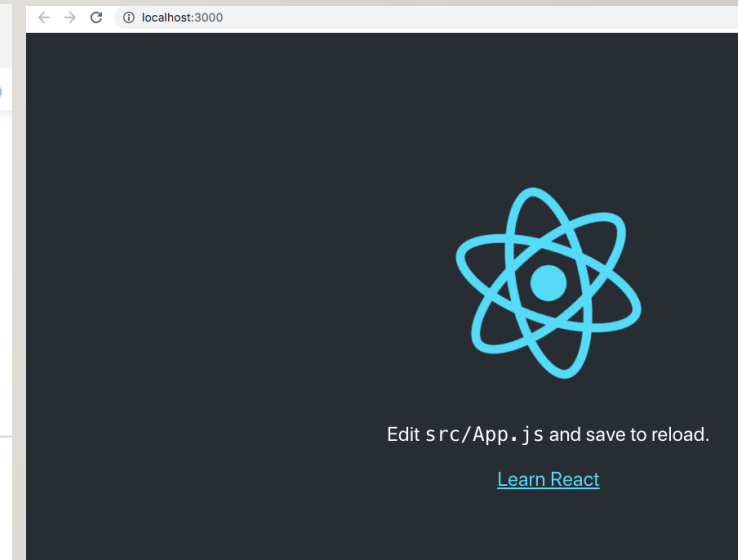
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
Compiled successfully!

You can now view my-app in the browser.

Local:      http://localhost:3000/
On Your Network:  http://192.168.1.8:3000/

Note that the development build is not optimized.
To create a production build, use yarn build.
```





# STRUKTUR PROJECT REACT

---

- Pertama kali react akan menjalankan pada folder src index.js
- Pada line 7
  - Terdapat code proses render di DOM
  - Pada parameter render ada dua
  - ReactDOM.render(element, container[, callback])
  - <https://reactjs.org/docs/react-dom.html#render>
- Jadi pada line 7 terdapat 2 yang akan dirender
  - Component App (pada src/ App.js)
  - Id=root pada index.html (pada public/index.html)

JS index.js X

Pemograman\_framework > pertemuan-kedua > belajar-react-app > my-app > src > JS index.js

```
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  ReactDOM.render(<App />, document.getElementById('root'));
8
9  // If you want your app to work offline and load faster, you can change
10 // unregister() to register() below. Note this comes with some pitfalls.
11 // Learn more about service workers: https://bit.ly/CRA-PWA
12 serviceWorker.unregister();
13
```

# STRUKTUR PROJECT REACT

---

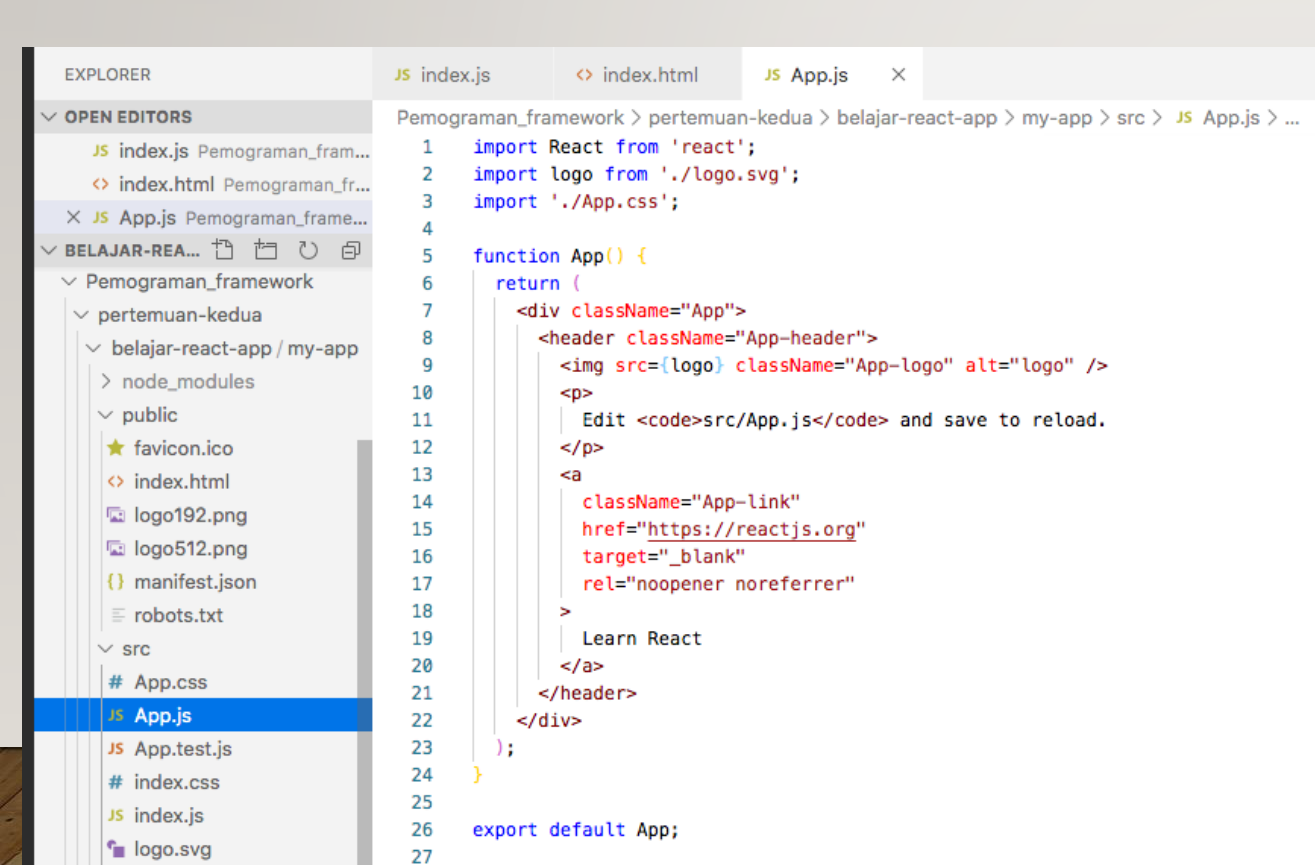
- id = root ada pada index.html (src/index.html), line 31



```
JS index.js  <> index.html X
pertemuan-kedua > my-app > public > <> index.html > ...
10  <!-- Content= web site created using create-react-app
11  />
12  <link rel="apple-touch-icon" href="%PUBLIC_URL%/logo192.png" />
13  <!--
14  manifest.json provides metadata used when your web app is installed on a
15  user's mobile device or desktop. See 
```

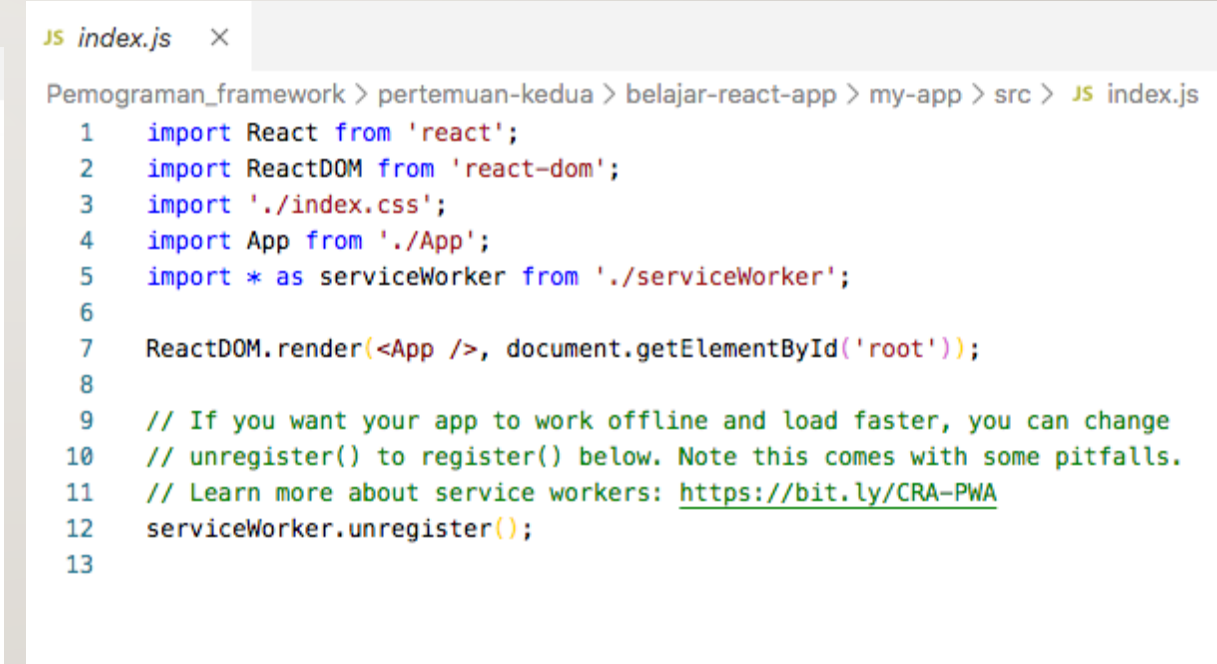
# STRUKTUR PROJECT REACT

- Component/functions App pada src/App.js , maka pada index.js pada line 4 diimportkan App nya



The screenshot shows the VS Code interface. On the left, the Explorer sidebar displays the project structure: 'Pemograman\_framework' > 'pertemuan-kedua' > 'belajar-react-app / my-app' > 'src'. The 'App.js' file is selected and highlighted in blue. The main editor area shows the content of 'App.js' with line numbers 1 through 27. The code defines an 'App' function that returns a JSX element with a header, a logo, and a link to 'Learn React'. The file is exported as default.

```
1 import React from 'react';
2 import logo from './logo.svg';
3 import './App.css';
4
5 function App() {
6   return (
7     <div className="App">
8       <header className="App-header">
9         <img src={logo} className="App-logo" alt="logo" />
10        <p>
11          Edit <code>src/App.js</code> and save to reload.
12        </p>
13        <a
14          className="App-link"
15          href="https://reactjs.org"
16          target="_blank"
17          rel="noopener noreferrer"
18        >
19          Learn React
20        </a>
21      </header>
22    </div>
23  );
24 }
25
26 export default App;
27
```



The screenshot shows the VS Code editor with the 'index.js' file open. The code imports 'React' from 'react', 'ReactDOM' from 'react-dom', './index.css', and the 'App' component from './App'. It also imports 'serviceWorker' from './serviceWorker'. The code then renders the 'App' component into the 'root' element of the document. Comments explain that the service worker is unregistered to allow the app to work offline and load faster, and provide a link to learn more about service workers.

```
1 import React from 'react';
2 import ReactDOM from 'react-dom';
3 import './index.css';
4 import App from './App';
5 import * as serviceWorker from './serviceWorker';
6
7 ReactDOM.render(<App />, document.getElementById('root'));
8
9 // If you want your app to work offline and load faster, you can change
10 // unregister() to register() below. Note this comes with some pitfalls.
11 // Learn more about service workers: https://bit.ly/CRA-PWA
12 serviceWorker.unregister();
13
```

# MEMBUAT COMPONENT HELLO MENGGUNAKAN ARROW FUNCTION

---

- Buka file index.js pada src dan tambahkan arrow function dengan nama Hello

```
JS index.js  x  <> index.html  JS App.js

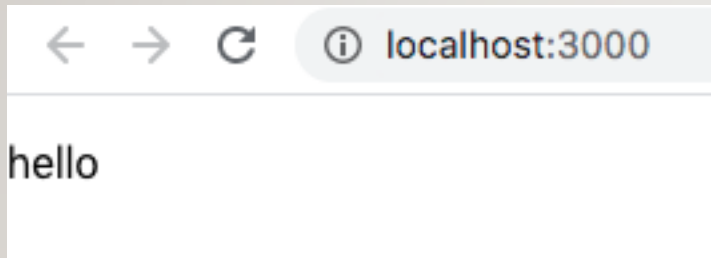
Pemograman_framework > pertemuan-kedua > belajar-react-app > my-app > src > JS index.js > [🔗] Hello
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  // import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  const Hello = () => {
8    return <p>hello</p>
9  }
10
11 // ReactDOM.render(<App />, document.getElementById('root'));
12
13 ReactDOM.render(<Hello />, document.getElementById('root'));
14
15 // ReactDOM.render(<p> hello word</p>, document.getElementById('root'));
16
17 // If you want your app to work offline and load faster, you can change
18 // unregister() to register() below. Note this comes with some pitfalls.
19 // Learn more about service workers: https://bit.ly/CRA-PWA
20 serviceWorker.unregister();
21
```



# MEMBUAT COMPONENT HELLO MENGGUNAKAN ARROW FUNCTION

---

- Hasilnya akan seperti berikut



- Hal ini disebabkan karena `ReactDOM.render` mengrender function `hello` yang akan ditampilkan pada `id root`



# MENAMPILKAN TEXT KE HALAMAN REACT MENGGUNAKAN REACTDOM.RENDER

---

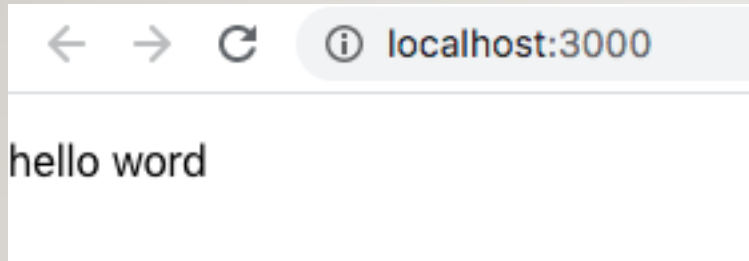
- Selain cara diatas terdapat juga cara yg lain untuk menampilkan text pada halaman react

```
JS index.js  x  <> index.html  JS App.js
Pemograman_framework > pertemuan-kedua > belajar-react-app > my-app > src > JS index.js > ...
1  import React from 'react';
2  import ReactDOM from 'react-dom';
3  import './index.css';
4  // import App from './App';
5  import * as serviceWorker from './serviceWorker';
6
7  const Hello = () =>{
8    |   return <p>hello</p>
9  }
10
11  // ReactDOM.render(<App />, document.getElementById('root'));
12
13  // ReactDOM.render(<Hello />, document.getElementById('root'));
14
15  ReactDOM.render(<p> hello word</p>, document.getElementById('root'));
16
17  // If you want your app to work offline and load faster, you can change
18  // unregister() to register() below. Note this comes with some pitfalls.
19  // Learn more about service workers: https://bit.ly/CRA-PWA
20  serviceWorker.unregister();
```

# MENAMPILKAN TEXT KE HALAMAN REACT MENGUNAKAN REACTDOM.RENDER

---

- Hasilnya



- ReactDOM.render akan mengrender `<p> hello word <p>` pada `id= root` pada `index.html`

# TUGAS

---

- Buatlah sebuah halaman biodata kalian dan tambahkan foto
- Menggunakan bootstrap