

USE EFFECT

```
// pakai useEffect  
// kepanggil ketika ?  
// mounted + ketika setiap ada perubahan di statenya  
useEffect(() => console.count('ini hanya 1 parameter'));  
  
// pakai useEffect dengan param 2 nya arr kosong  
// cuman ketika mounted  
useEffect(() => console.count('ini 2 parameter, array kosong []'), []);  
  
// pakai useEffect dengan param 2, array counter  
// ketika mounted + ketika ada perubahan di counter  
useEffect(  
  () => console.count('ini 2 parameter, array [counter]'),  
  [counter]
```

```
9 import logo from "./logo.svg";  
8 import "./App.css";  
7 import { useState, useEffect } from "react"  
6  
5 function App() {  
4   const [users, setUsers] = useState([])  
3  
2   useEffect(() => {  
1  
10    }, [1])  
1  
2   return (  
3     <div className="App">  
4       <h1>Home Page</h1>  
5     </div>  
6   );  
7 }  
8  
9 export default App;
```

Kalo array kosong, dijalankan sekali saja Ketika component di render.
Kalo array ada isi, setiap variable berubah UseEffect kepanggil.

```
9 import logo from "./logo.svg";
8 import "./App.css";
7 import { useState, useEffect } from "react"
6
5 function App() {
4   const [users, setUsers] = useState([])
3
2   useEffect(() => {
1
10  }, [users, products])
1
2   return (
3     <div className="App">
4       <h1>Home Page</h1>
5     </div>
6   );
7 }
8
9 export default App;
```

Kalau ada salah 1 dependency berubah, maka useEffect berjalan

```
25 import logo from "./logo.svg";
24 import "./App.css";
23 import { useState, useEffect } from "react";
22
21 function App() {
20   const [users, setUsers] = useState([]);
19
18   useEffect(() => {
17     fetch("http://localhost:3001/users")
16       .then((res) => {
15         if (!res.ok) throw new Error("ada error nih");
14         return res.json();
13       })
12       .then((data) => {
11         setUsers(data);
10       })
9       .catch(console.log);
8   }, []);
7
6   return (
5     <div className="App">
4       <h1>Home Page</h1>
3       <h2>Users</h2>
2       <ul>
1         {users.map((user) => (
26           <li>{user.name}</li>
1         )));
2       </ul>
3     </div>
4   );
5 }
6
7 export default App;
~
```

Loading, Error handling

```
buffers

23 import logo from "./logo.svg";
22 import "./App.css";
21 import { useState, useEffect } from "react";
20
19 function App() {
18   const [users, setUsers] = useState([]);
17   const [loading, setLoading] = useState(true);
16   const [error, setError] = useState(false);
15
14   useEffect(() => {
13     fetch("http://localhost:3001/users")
12       .then((res) => {
11         if (!res.ok) throw new Error("ada error nih");
10         return res.json();
9       })
8       .then((data) => {
7         setUsers(data);
6       })
5       .finally(() => {
4         setLoading(false)
3       })
2       .catch(() => {
1         setError(true)
24     });
1   }, []);
2
3   return (
4     <div className="App">
5       <h1>Home Page</h1>
6       <h2>Users</h2>
7       <ul>
8         {loading && <h3>Loading</h3>}
9         {error && <h3>Ada error nih ... </h3>}
10        {!loading && !error && users.map((user) => <li>{user.name}</li>)}
11      </ul>
12    </div>
13  );
14}

INSERT ↵ master  client/src/App.js[+]
jav... 60% In :24/40≡%:8
```

CUSTOM HOOK

```
Fetch.js | c/s/h/useFetch.js | buffers
  9 import logo from "./logo.svg";
  8 import "./App.css";
  7 import { useState, useEffect } from "react";
  6 import useFetch from "./hooks/useFetch";
  5
  4 function App() {
  3   const [posts, setPosts] = useState([]);
  2   const [loadingPosts, setLoadingPosts] = useState(true);
  1   const [errorPosts, setErrorPosts] = useState(false);
10   const { data: users, loading, error } = useFetch(
  1    "http://localhost:3001/users"
  2  );
  3
  4   useEffect(() => {
  5     fetch("http://localhost:3002/posts")
  6       .then((res) => {
  7         if (!res.ok) throw new Error("ada error nih");
  8         return res.json();
  9       })
10       .then((data) => {
11         setPosts(data);
12       })
13       .finally(() => {
14         setLoadingPosts(false);
15       })
16       .catch(() => {
17         setErrorPosts(true);
18       });
19   }, []);
20
21   return (
22     <div className="App">
23       <h1>Home Page</h1>
24       <h2>Users</h2>
25       <ul>
26         {loading && <h3>Loading</h3>}
27         {error && <h3>Ada error nih ...</h3>}
28         {!loading && !error && users.map((user) => <li>{user.name}</li>)}

NORMAL ⌂ master client/src/App.js                jav... 19% ln :10/52≡%:1
```

```
1 import { useState, useEffect } from "react";
0
9 const useFetch = (url) => {
8   const [data, setData] = useState([]);
7   const [loading, setLoading] = useState(true);
6   const [error, setError] = useState(false);
5
4   useEffect(() => {
3     fetch(url)
2       .then((res) => {
1       if (!res.ok) throw new Error("ada error n");
0       return res.json();
9     })
8       .then((resData) => {
7       setData(resData);
6     })
5       .finally(() => {
4       setLoading(false);
3     })
2       .catch(() => {
1       setError(true);
0     });
9   }, []);
8
7   return {
6     data,
5     loading,
4     error,
3   };
2 };
1
  export default useFetch;
```

2 useFetch

```
Fetch.js | c/s/h/useFetch.js | buffers
13 import logo from "./logo.svg";
12 import "./App.css";
11 import { useState, useEffect } from "react";
10 import useFetch from "./hooks/useFetch";
9
8 function App() {
7 const { data: users, loading, error } = useFetch(
6 "http://localhost:3001/users"
5 );
4 const { data: posts, loading: loadingPosts, error: errorPosts } = useFetch(
3 "http://localhost:3001/posts"
2 );
1
14 return (
15   <div className="App">
16     <h1>Home Page</h1>
17     <h2>Users</h2>
18     <ul>
19       {loading && <h3>Loading</h3>}
20       {error && <h3>Ada error nih ...</h3>}
21       {!loading && !error && users.map((user) => <li>{user.name}</li>)}
22     </ul>
23     <h2>Posts</h2>
24     <ul>
25       {loadingPosts && <h3>Loading</h3>}
26       {errorPosts && <h3>Ada error nih ...</h3>}
27       {!loadingPosts &&
28         !errorPosts &&
29         posts.map((post) => <li>{post.title}</li>)}
30     </ul>
31   </div>
32 );
33
34 export default App;
```

NORMAL ↵ master client/src/App.js

jav... 40% ln :14/35≡%:2

RE-RENDER DATA KARENA SETDATA DI REACT HARUS RENDER COMPONEN LAGI

```
useFetch.js  c/s/h/useFetch.js                                buffers
21  const [data, setData] = useState([]);
20  const [loading, setLoading] = useState(true);
19  const [error, setError] = useState(false);
18  const [counter, setCounter] = useState(0);
17  useEffect(() => {
16    fetch(url)
15      .then((res) => {
14        if (!res.ok) throw new Error("ada error nih");
13        return res.json();
12      })
11      .then((resData) => {
10        setData(resData);
9      })
8        .finally(() => {
7          setLoading(false);
6        })
5        .catch(() => {
4          setError(true);
3        });
2      }, []);
1
25  console.log(data, "<<<<<");
1 // myFn
2 //
3  const inc = () => {
4    setCounter(counter + 1);
5  };
6
7  return {
8    data,
9    loading,
10   error,
11   //myFn
12   inc,
13   counter,
14 };
15 };
16
NORMAL  ↵ master  client/src/hooks/useFetch.js          jav...  59%  ln :25/42≡%:2
```



strict mode sebenarnya ketika development mereka akan mencoba render 2 kali untuk debug component kalian, ketika di production dia otomatis sekali aja rendernya. Biar temen2 ga bingung boleh di matiin dulu strict modenya

HUKUM HARAM HOOKS

```
| 20 import logo from "./logo.svg";
| 19 import "./App.css";
| 18 import { useState, useEffect } from "react";
| 17 import useFetch from "./hooks/useFetch";
| 16
| 15 function App() {
| 14   const { counter, inc, data: users, loading, error } = useFetch(
| 13     "http://localhost:3001/users"
| 12   );
| 11   const { data: posts, loading: loadingPosts, error: errorPosts } = useFetch(
| 10     "http://localhost:3001/posts"
| 9   );
| 8   [
| 7   if() {
| 6     useFetch()
| 5   }
| 4
| 3   const x = () => {
| 2     useFetch()
| 1   };
| 21
| 1
| 2   useFetch()
| 3
| 4   return (
| 5     <div className="App">
| 6       <h1>Home Page</h1>
| 7       <h2>Users</h2>
| 8       <ul>
| 9         <p>counter: {counter}</p>
| 10        <button onClick={inc}>inc</button>
| 11        {loading && <h3>Loading</h3>}
| 12        {error && <h3>Ada error nih ... </h3>}
| 13        {!loading && !error && users.map((user) => <li>{user.name}</li>)}
| 14      </ul>
| 15      <h2>Posts</h2>
| 16      <ul>
| 17        {loadingPosts && <h3>Loading</h3>}
```

INSERT ↵ master client/src/App.js[+]

jav... 43% ln :21/48≡%:1

TARO HOOKS DI TOP LEVEL / PALING AWAL BIAR KONSISTEN SAAT RENDERING