React and React Native Basics

Server-Side Rendering Concept

Sesi 12

PERHATIAN

Jangan lupa untuk isi form absensi. Bagi instruktur mohon mengisi form absensi yang telah diinfokan dan konfirmasi nomor urut peserta yang hadir.

Untuk student harap mengisi form absensi di sini sebelum kelas dimulai. Untuk kode peserta dapat ditanyakan kepada instruktur dan jangan lupa mencantumkan pertemuan ke 1

Mengenal Server-Side Rendering

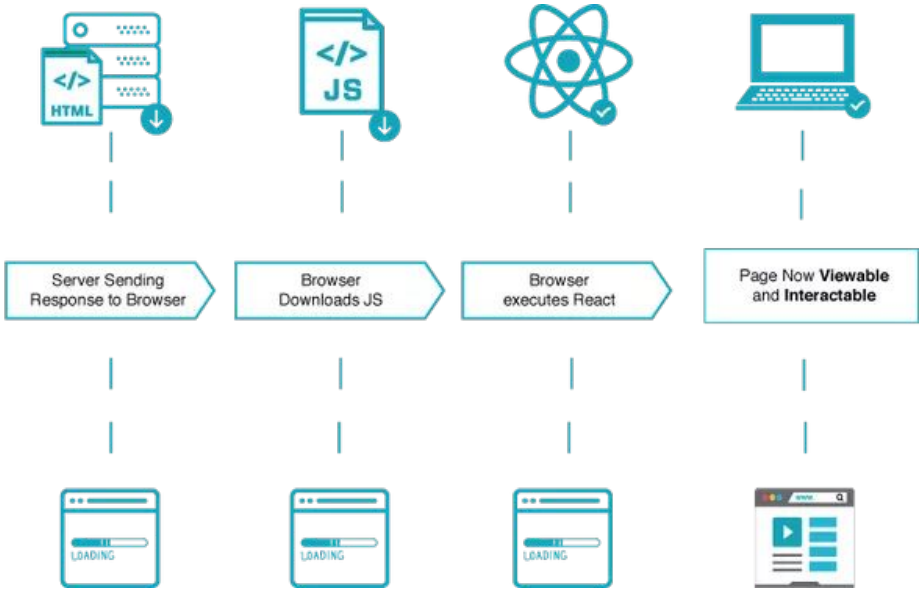


Image credit: https://www.telerik.com/blogs/understanding-server-side-rendering

Server-Side Rendering (SSR) adalah teknik untuk menampilkan **Single Page Application** (SPA) yang akan di-render oleh **server**. Lalu, hasilnya akan diteruskan kepada **client**.

Apa itu Server-Side Rendering?

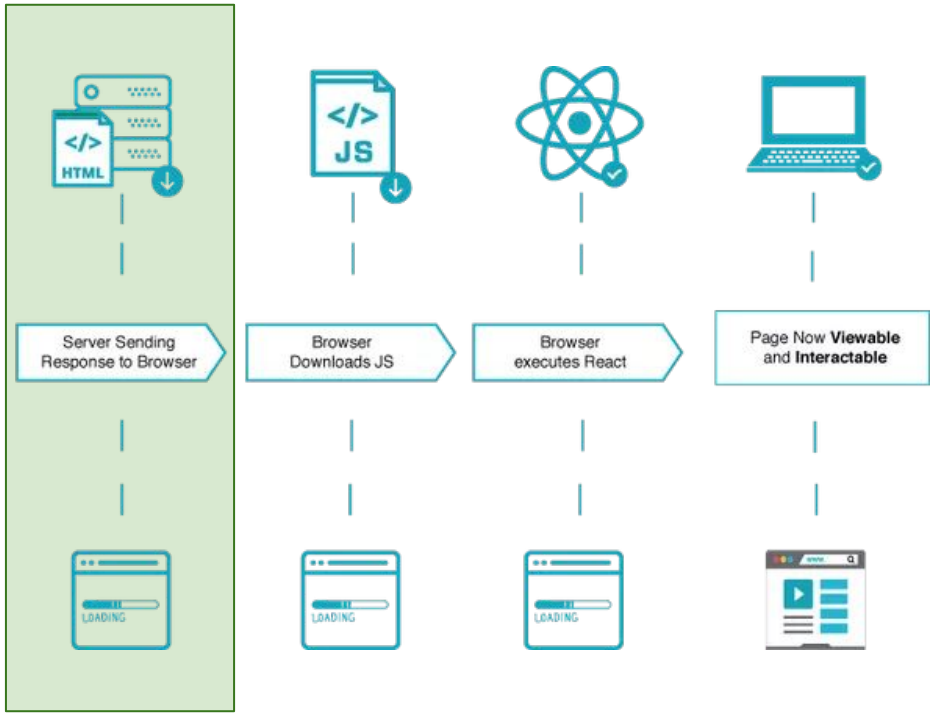


Image credit: https://www.telerik.com/blogs/understanding-server-side-rendering

Ketika sebuah **client** melakukan request, **server** akan memproses request tersebut, kemudian data dari hasil proses ini akan dirangkai dalam format HTML.

HTML ini kemudian dikirimkan oleh **server** kepada **client** tersebut.

Cara Kerja Server-Side Rendering

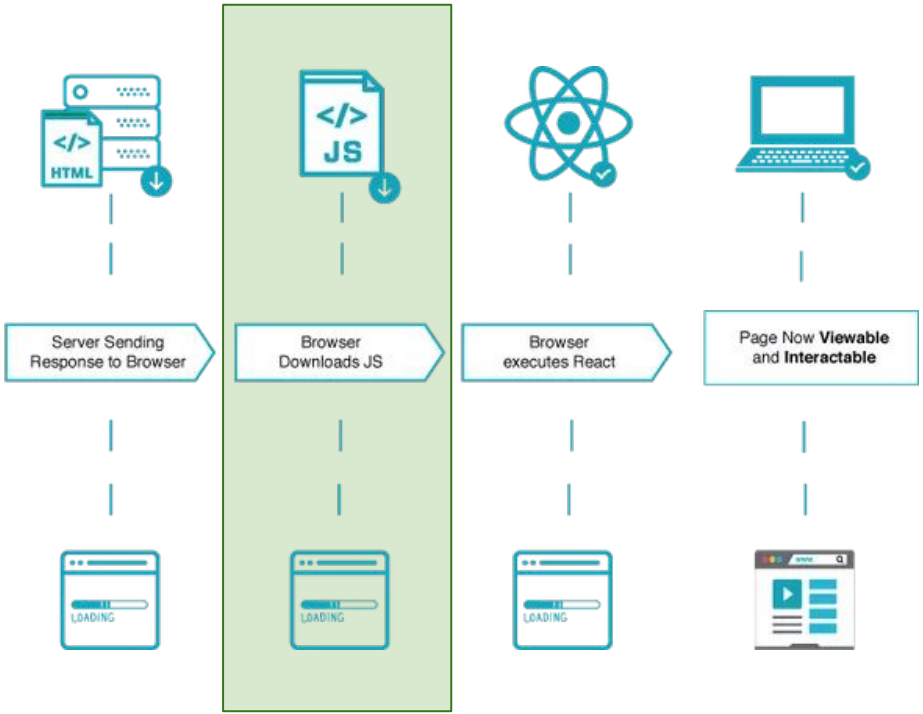


Image credit: https://www.telerik.com/blogs/understanding-server-side-rendering

Pada tahap ini, client berhasil mendapatkan HTML dari server.

Kemudian, client akan mengambil kode JavaScript yang diperlukan.

Cara Kerja Server-Side Rendering

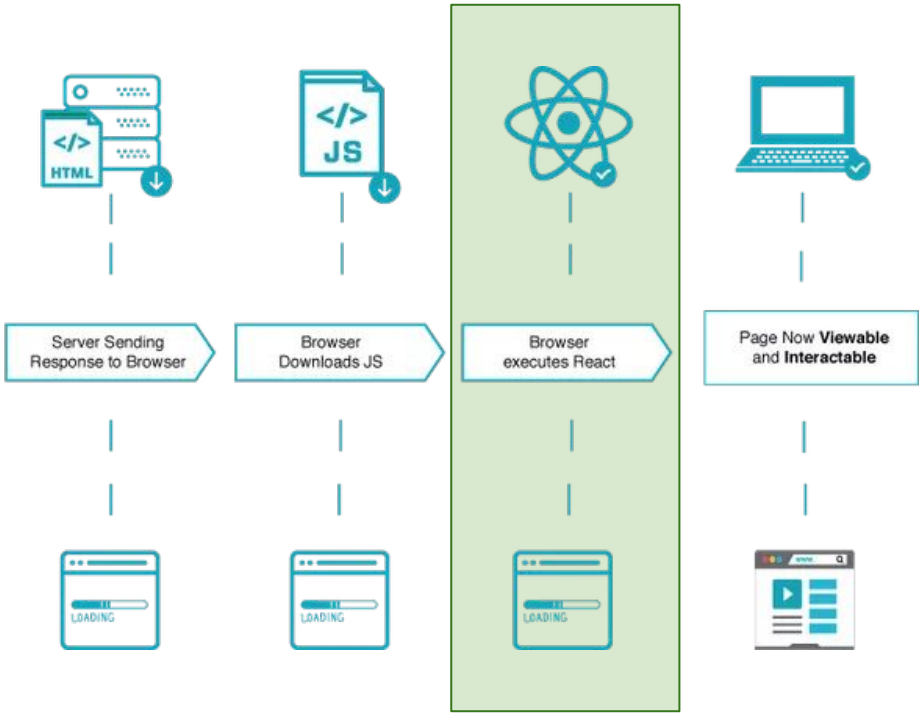


Image credit: https://www.telerik.com/blogs/understanding-server-side-rendering

Setelah kode JavaScript berhasil didapat, client selanjutnya akan mengeksekusi kode React.

Cara Kerja Server-Side Rendering

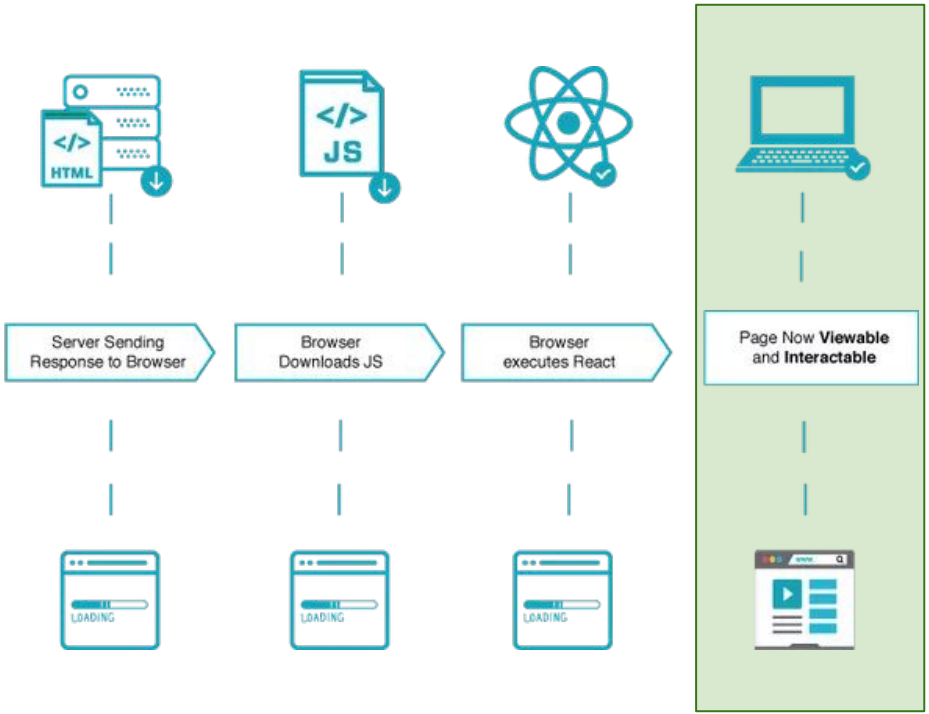


Image credit: https://www.telerik.com/blogs/understanding-server-side-rendering

Nah, pada langkah ini, kode React berhasil dijalankan pada client. Setelah itu, user dapat berinteraksi dengan halaman tersebut.

Cara Kerja Server-Side Rendering

Keuntungan dan Kekurangan SSR

Search Engine Optimization

SSR akan **membantu search engine melakukan crawling konten aplikasi/website kita**, karena data akan diambil terlebih dahulu sebelum di-serve. Mengapa? Karena pada dasarnya, **crawling dari search engine umumnya tidak mendukung JavaScript**, sehingga **ketika di-crawl hanya akan menampilkan halaman kosong saja**.

Di samping itu, hanya beberapa search engine tertentu (seperti Google), yang mampu melakukan crawling konten tanpa SSR.

Efisien dan Cepat

Karena prosesnya dilakukan pada server, **maka server-lah yang akan menanggung beban ketika mengambil data**. Oleh karena itu, user dengan kecepatan internet yang lambat, akan mendapatkan manfaatnya.

Di samping itu, **konten akan ditampilkan terlebih dahulu**, berkat bantuan SSR ini (masih ingat dengan cara kerja SSR sebelumnya?). Dengan begitu, user dapat melihat konten tanpa harus menunggu kode JavaScript selesai di-download.

Kelebihan Server-Side Rendering

Keuntungan & Kekurangan SSR

Load Time Meningkat

Karena semua pengambilan data dan rendering dilakukan pada server, maka waktu yang dibutuhkan untuk mengirimkan response akan meningkat juga.

Lama waktu yang dibutuhkan juga tergantung pada workload dan spesifikasi yang dimiliki server tersebut. Jika aplikasi semakin kompleks, waktu yang dibutuhkan juga semakin lama.

Potensi Masalah dengan Kode Pihak Ketiga

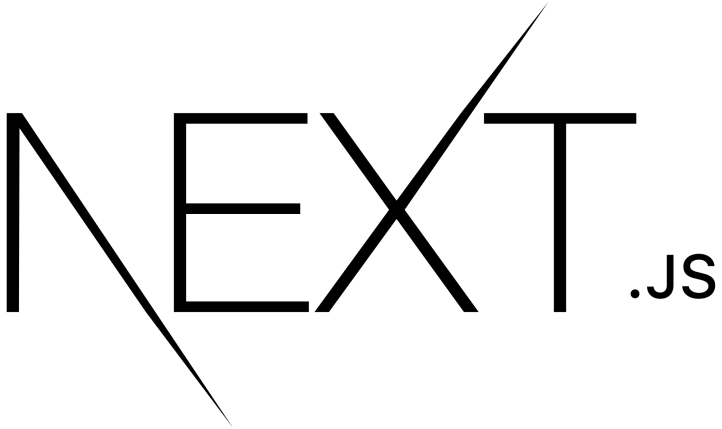
Ada juga kemungkinan atau potensi masalah dengan kode JavaScript oleh pihak ketiga. Namun, hal ini bukan berarti kamu tidak diperbolehkan menggunakan kode JavaScript pihak ketiga, ya.

Jadi, sebelum kamu coba terapkan SSR pada aplikasi kamu, ada baiknya kamu uji coba apakah kode JavaScript pihak ketiga dapat berjalan ketika menggunakan SSR atau tidak.

Kekurangan Server-Side Rendering

Keuntungan & Kekurangan SSR

Yuk, Praktik SSR!



Pernah mendengar sebutan framework sebelumnya? Kita bisa analogikan framework sebagai "bumbu siap pakai" ketika memasak.

Nah, **Next.js** adalah salah satunya. Next.js merupakan framework berbasis React yang menyediakan semua yang kita butuhkan. Salah satunya adalah, fitur SSR yang akan kita praktikkan selanjutnya.

Next.js: Framework React dengan SSR

Yuk, Praktik SSR!

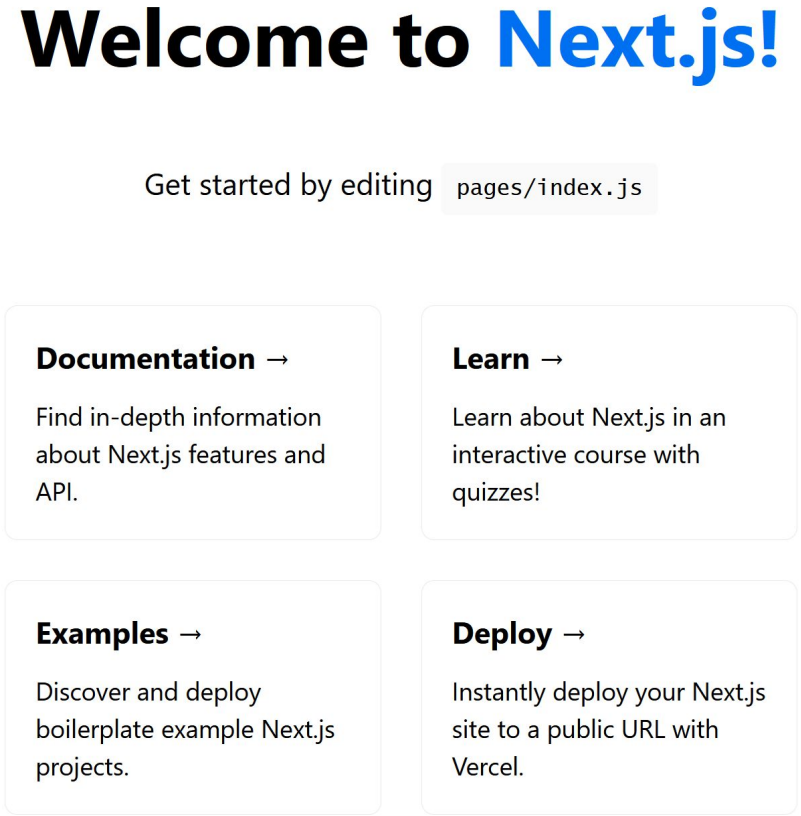
HACKTIV8

```
npx create-next-app@latest
```

Untuk **memulai project baru dengan framework Next.js**, kita bisa melakukan dengan 2 cara, yaitu **setup otomatis** dan **setup manual**.

Kita akan menjalankan **setup otomatis** saja, yaitu dengan menjalankan perintah seperti pada contoh di samping. Lalu, kita ikuti instruksi yang diminta.

Menyiapkan Project dengan Next.js



Setelah selesai, kamu bisa langsung masuk ke dalam project yang dibuat dengan `create-next-app`, lalu jalankan `npm run dev`.

Setelah selesai di-compile, buka `localhost:3000` seperti biasanya.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

HACKTIV8

Di sini, kamu akan dihadapkan dengan 4 link yang akan membantu kamu dalam mempelajari framework Next.js, masing-masing adalah **Dokumentasi**, **Pelajari**, **Contoh**, dan **Deploy** ke Vercel.

Kita akan berfokus pada penerapan SSR untuk kali ini, ya.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

HACKTIV8

```jsx
import { useEffect, useState } from 'react'

export default function Home() {
  const [users, setUsers] = useState([]);

  useEffect(
    () => {
      fetch('https://jsonplaceholder.typicode.com/users')
      .then(res => res.json())
      .then(data => setUsers(data));
    },
    []
  );

  return (
    <table>
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        {
          users.map(user => (
            <tr key={user.id}>
              <td>{user.id}</td>
              <td>{user.name}</td>
              <td>{user.email}</td>
            </tr>
          ))
        }
      </tbody>
    </table>
  );
}
```

pages/**index.js**

Pada file **pages/index.js**, kita timpa seluruh baris kode dengan kode seperti contoh pada gambar di samping.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

TANPA SSR

HACKTIV8

ID	Name	Email
1	Leanne Graham	Sincere@april.biz
2	Ervin Howell	Shanna@melissa.tv
3	Clementine Bauch	Nathan@yesenia.net
4	Patricia Lebsack	Julianne.OConner@kory.org
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info
7	Kurtis Weissnat	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me
9	Glenna Reichert	Chaim_McDermott@dana.io
10	Clementina DuBuque	Rey.Padberg@karina.biz

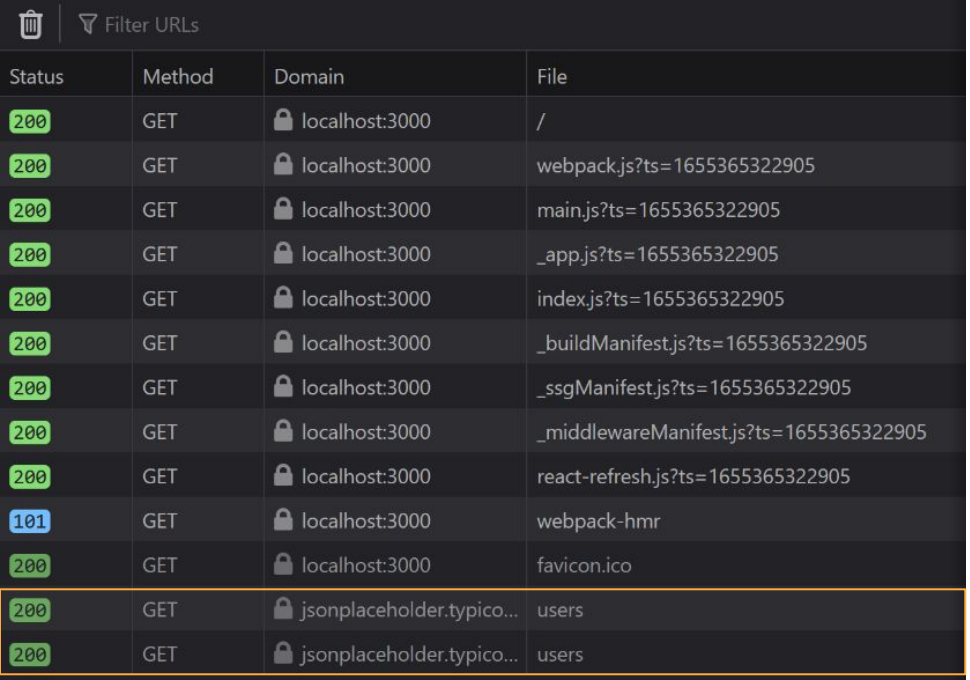
Jika implementasi kamu benar, seharusnya akan tertampil data user seperti pada contoh di samping.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

TANPA SSR

HACKTIV8



Nah, sekarang kamu coba buka console di browser dengan menekan **[F12]** pada keyboard, lalu tuju tab **Network**. Setelah itu, refresh halamannya.

Perhatikan daftar yang mengarah ke domain `jsonplaceholder.typicode.com` ya!

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

TANPA SSR

HACKTIV8

```jsx
export default function Users(props) {
  return (
    <table>
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        {
          props.users?.map(user => (
            <tr key={user.id}>
              <td>{user.id}</td>
              <td>{user.name}</td>
              <td>{user.email}</td>
            </tr>
          ))
        }
      </tbody>
    </table>
  );
}

export async function getServerSideProps(context) {
  const res = await fetch('https://jsonplaceholder.typicode.com/users');
  const users = await res.json();

  return {
    props: { users }
  }
}
```

pages/**users-ssr.js**

Sekarang kita akan menambahkan 1 component baru dengan nama file **`users-ssr.js`** pada folder **`pages`**. Perhatikan nama file-nya, teman-teman, karena akan kita gunakan ketika membuka halaman pada browser.

Lalu, isi file tersebut dengan contoh kode di samping.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

DENGAN SSR

HACKTIV8

```jsx
export default function Users(props) {
  return (
    <table>
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        {
          props.users?.map(user => (
            <tr key={user.id}>
              <td>{user.id}</td>
              <td>{user.name}</td>
              <td>{user.email}</td>
            </tr>
          ))
        }
      </tbody>
    </table>
  );
}
```

```jsx
export async function getServerSideProps(context) {
  const res = await fetch('https://jsonplaceholder.typicode.com/users');
  const users = await res.json();

  return {
    props: { users }
  }
}
```

pages/**users-ssr.js**

Pada kode ini, kamu akan melihat ada satu function baru bernama **`getServerSideProps()`**. Apa fungsi dari function tersebut?

Merujuk pada dokumentasi, function **`getServerSideProps()`** akan dijalankan pada level request. Jadi, ketika kita membuka **`localhost:3000/users-ssr`**, maka kita akan langsung mendapatkan data-nya, tanpa harus melakukan request ulang.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

DENGAN SSR

HACKTIV8

```jsx
export default function Users(props) {
  return (
    <table>
      <thead>
        <tr>
          <th>ID</th>
          <th>Name</th>
          <th>Email</th>
        </tr>
      </thead>
      <tbody>
        {
          props.users?.map(user => (
            <tr key={user.id}>
              <td>{user.id}</td>
              <td>{user.name}</td>
              <td>{user.email}</td>
            </tr>
          ))
        }
      </tbody>
    </table>
  );
}

export async function getServerSideProps(context) {
  const res = await fetch('https://jsonplaceholder.typicode.com/users');
  const users = await res.json();

  return {
    props: { users }
  }
}
```

pages/**users-ssr.js**

Function **`getServerSideProps()`** ini akan dijalankan secara asynchronous, sehingga request dapat dilakukan berkali-kali tanpa harus menunggu request yang lain selesai.

Selain itu, hasil dari function ini akan dikembalikan dalam bentuk **`props`**.

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

DENGAN SSR

ID	Name	Email
1	Leanne Graham	Sincere@april.biz
2	Ervin Howell	Shanna@melissa.tv
3	Clementine Bauch	Nathan@yesenia.net
4	Patricia Lebsack	Julianne.OConner@kory.org
5	Chelsey Dietrich	Lucio_Hettinger@annie.ca
6	Mrs. Dennis Schulist	Karley_Dach@jasper.info
7	Kurtis Weissnat	Telly.Hoeger@billy.biz
8	Nicholas Runolfsdottir V	Sherwood@rosamond.me
9	Glenna Reichert	Chaim_McDermott@dana.io
10	Clementina DuBuque	Rey.Padberg@karina.biz

Kita kembali lagi ke browser, namun kali ini tuju ke `localhost:3000/users-ssr`.

Jika implementasi kamu sudah benar, seharusnya tampilannya akan tetap sama seperti pada halaman tanpa menggunakan SSR.

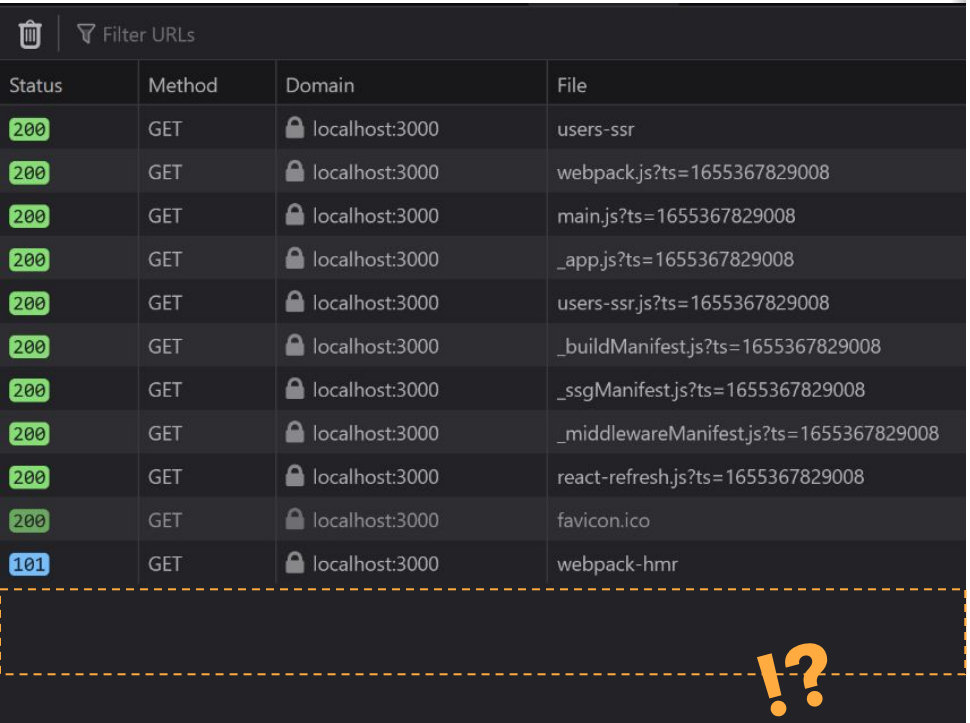
Lalu bedanya apa?

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

DENGAN SSR

HACKTIV8



Nah, bedanya terletak pada **bagaimana request tersebut dipanggil**. Masih ingat ketika kita membuka halaman tanpa penerapan SSR sebelumnya?

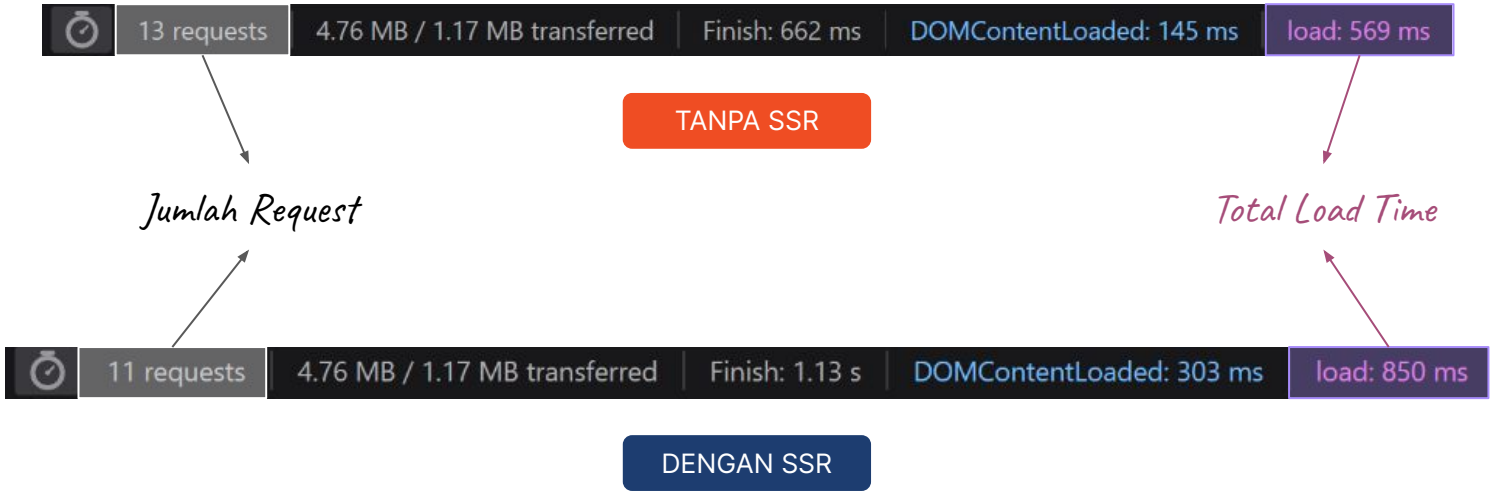
Nah, coba perhatikan pada tab **Network**. Apakah kamu menemukan domain yang sama pada waktu request?

Menyiapkan Project dengan Next.js

Yuk, Praktik SSR!

DENGAN SSR

HACKTIV8



Kemudian perhatikan juga load time pada waktu membuka **halaman tanpa SSR** dan **halaman dengan SSR**.

Di sini terlihat perbedaannya pada **jumlah request** yang dipanggil, dan **load time** yang dibutuhkan.

Hasil tanpa SSR dan dengan SSR

Jadi, kapan menggunakan SSR?

Image Credit: https://www.freepik.com/free-vector/flat-helpful-tip-colorful-background_4396540.htm

Semua kembali kepada tujuan kamu membuat aplikasi web dengan React.

Apakah aplikasi kamu akan menjadi kompleks, atau sederhana? Atau, apakah kamu mau aplikasi web kamu dapat di-crawl oleh semua search engine?

Waktu yang Tepat Adalah...

Jadi, kapan menggunakan SSR?

Image Credit: https://www.freepik.com/free-vector/flat-helpful-tip-colorful-background_4396540.htm

Nah, ketika kamu membuat aplikasi sederhana dan tidak kompleks -- semisal untuk menampilkan artikel -- maka kamu bisa mengimplementasikan SSR.

Dan, jika kamu ingin konten aplikasi kamu dicari oleh search engine, maka kamu disarankan untuk mengimplementasikan SSR, agar lebih mudah dicari.

Waktu yang Tepat Adalah...

Jadi, kapan menggunakan SSR?

HACKTIV8

Thank You

PT Hacktivate Teknologi Indonesia
Gedung Aquarius Pondok Indah
Jalan Sultan Iskandar Muda No.7
Kebayoran Lama, Jakarta Selatan

www.hacktiv8.com