BINAR ACADEMY

Next.Js

Gold - Chapter 8 - Topic 2

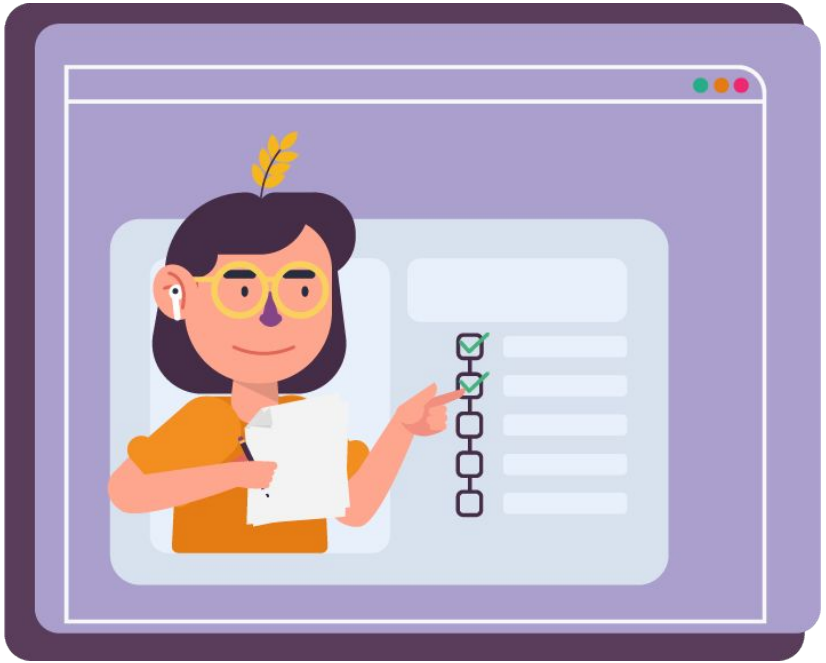
Selamat datang di Chapter 8 Topik 2 online course Fullstack Web dari Binar Academy!

Selamat datang di topik 2🥳

Nah, pada topik ini, kita bakal bahas tuntas tentang konsep fungsi Server-Side Rendering (SSR) serta implementasinya di dalam Next.js. Berangkat bestie 🚀🚀🚀

Detailnya, kita bakal bahas hal-hal berikut ini:

- Pengenalan konsep dan fungsi SSR
- Pengenalan Next.Js
- Fitur-fitur yang ada pada Next.Js
- Penerapan Next.Js



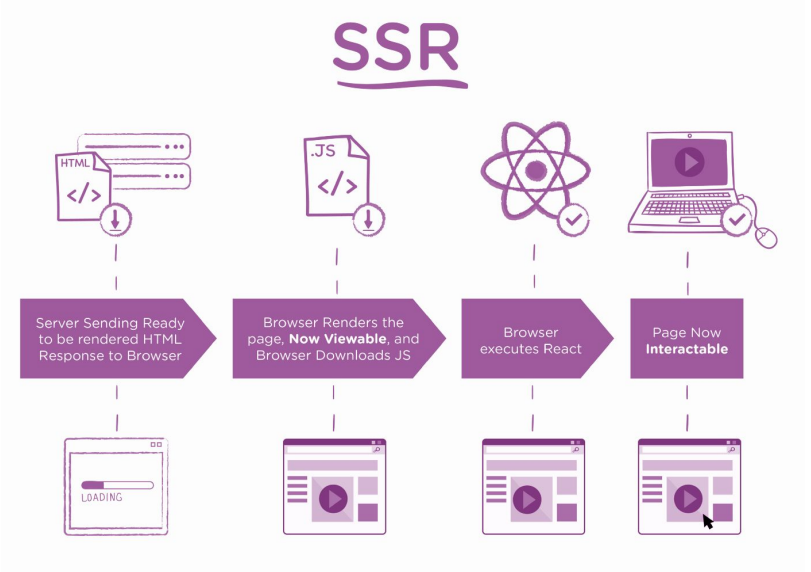
SSR (Server Side Rendering)

Server itu menyediakan data atau layanan sedangkan client yang melakukan permintaan data. Kalau server side rendering (SSR) dan client side rendering (CSR) itu apa ya kira–kira ?

Letsgoo kita kupas tuntas satu–satu perbedaannya🧐

Server-Side Rendering

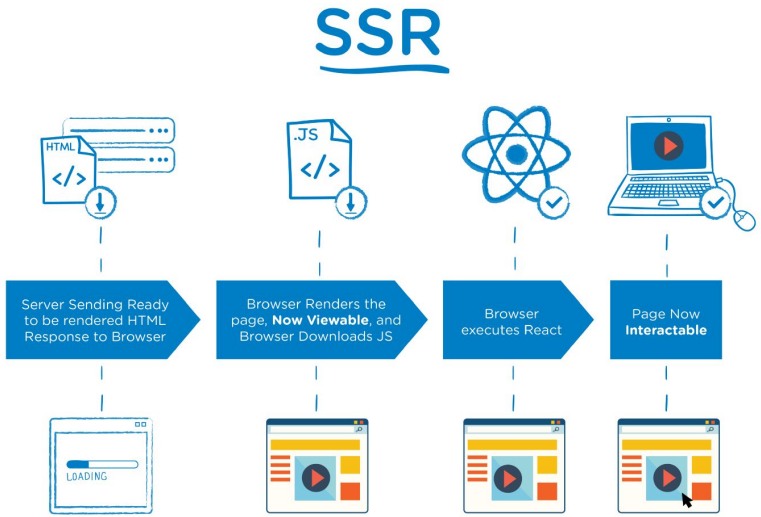
SSR adalah **metode rendering dimana proses pembuatan HTML terjadi di dalam server, lalu server akan mengirimkannya ke browser (client) untuk ditampilkan ke user**. Dengan kata lain, browser akan meminta informasi dari server, dan server merespon dengan mengirimkan halaman yang diminta dalam bentuk full-render.



SSR (Server Side Rendering)

SSR itu mirip seperti Single Page Architecture/ Client Side Rendering, yang membedakan adalah **proses render-nya dilakukan di server, jadi semua kode JavaScript yang kita tulis itu akan direnderi di server lalu setelah itu baru dikirim ke browser**.

Contoh framework dari SSR ini adalah Next.js, Nuxt.js, dan GatsbyJS.



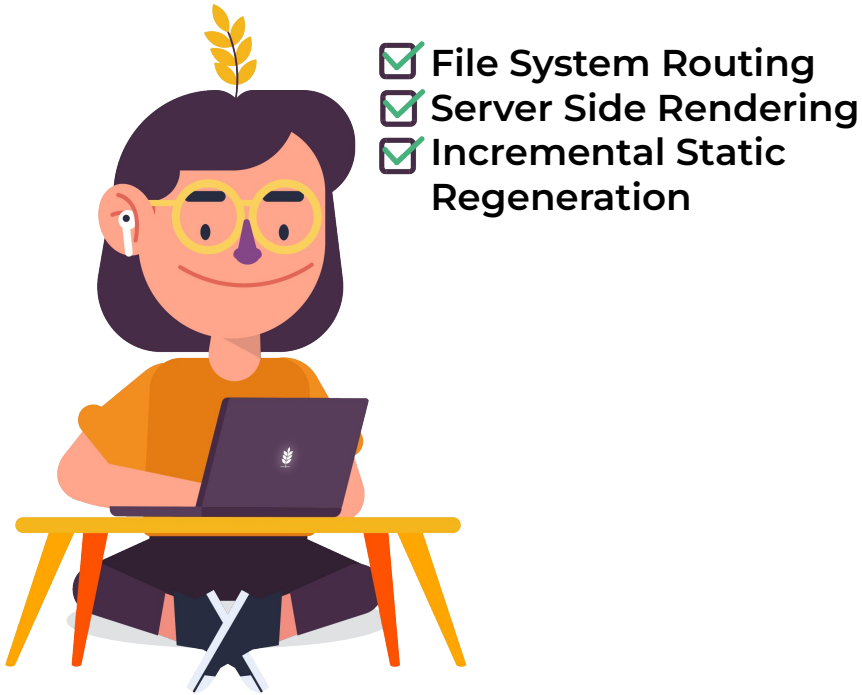
Kelebihan	Kekurangan
Search engine dapat menelusuri website dengan baik (SEO)	Server akan memakan biaya yang lebih.
Tidak akan ada tag <body> yang kosong sehingga konten yang akan ditelusuri search engine tersedia.	SSR membutuhkan server atau cloud untuk render, sehingga proses deployment lebih kompleks.
Baik untuk halaman website yang statis.	SSR bisa tidak kompatibel dengan kode JavaScript pihak ketiga.
Proses loading halaman website akan lebih cepat.	Efektif untuk website yang statis, sehingga jika website sangat kompleks maka prosesnya akan terhambat karena adanya perbedaan kapasitas.
Membantu user dengan internet yang lambat atau perangkat jadul, karena semuanya di render di server.	

Nah, kalian udah paham kan apa itu SSR? Sekarang saatnya kita cari tahu implementasinya seperti apa di dalam **Next.Js**~

Next.Js

Next.Js adalah sebuah **framework yang bisa digunakan untuk membuat user interface dari sebuah website**. Nah next.js ini jauh lebih canggih di banding CRA, karena ia memiliki beberapa fitur killer yang ga boleh kamu lewatin, beberapa diantaranya adalah:

- File System Routing
- Server Side Rendering (SSR)
- Incremental Static Regeneration (ISR)



Mari kita mulai dengan membuat project Next.Js

Sebelum kita bahas lebih dalam tentang Next.Js-nya, lebih baik kita coba buat project Next.Js dulu agar lebih tergambar. Prosesnya mudah kamu hanya perlu menjalankan perintah disamping.

Setelah kamu berhasil ngebikin next.js project, langsung aja kuy kita bahas tentang **File System Routing**.

```
npx create-next-app nama-project
```

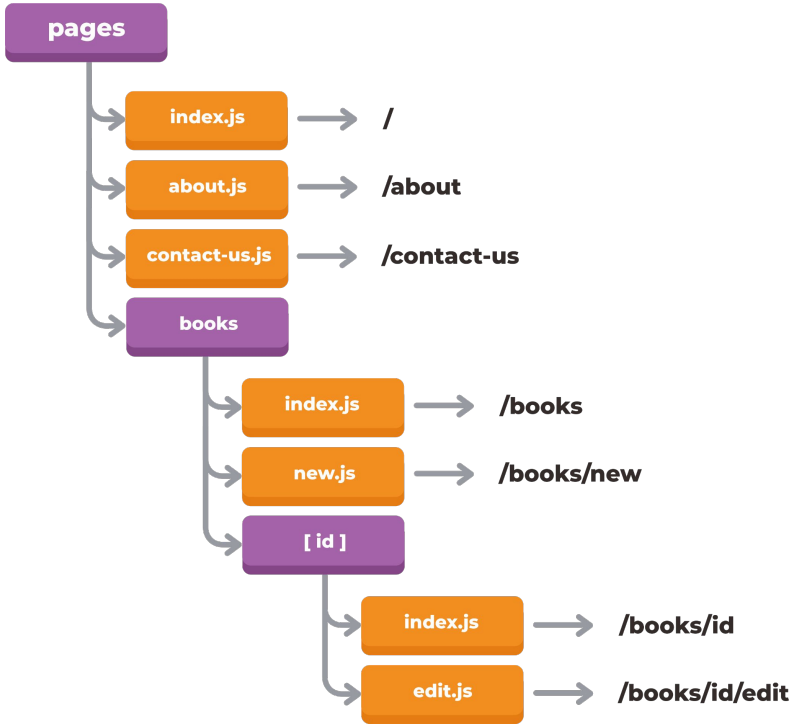
```
yarn create next-app nama-project
```

File System Routing

Beda nih sob sama si React yang mana kalo kita mau melakukan routing dia harus pake library **react-router-dom**, di Next.js untuk masalah routing tuh mudaaah banget-nget-nget!

Kita cuma perlu **membuat file di dalam folder pages di dalam project next.js**. File tersebut akan meng-eskpor component sebagai export default-nya, dan component tersebut akan menjadi page di dalam aplikasi kita.

Detail terkait routing bisa kamu lihat disini.



Data Fetching

Nah **data fetching inilah yang menentukan page dari next.js ini perlu di-render dengan metode apa.**

Ada 3 metode rendering di dalam next.js

1. Server Side Rendering
2. Incremental Static Regeneration
3. Static Site Generation

Masing-masing metode memiliki implementasi data fetching yang berbeda. Mari kita mulai dengan **Server Side Rendering**.

Server Side Rendering

Untuk mulai merender halaman dengan metode SSR, kamu perlu membuat fungsi yang bernama **getServerSideProps** di dalam komponen Halamanmu, seperti contoh di samping.

Pake metode ini kalo misal halamanmu sangat dinamis, datanya selalu berubah-rubah.

Code ini bisa kalian lihat disini.

```
// { posts } ini ngambil dari hasil fungsi getServerSideProps
export default function PakeServerSideRendering({ posts }) {
  return ( ... ComponentMu)
}

export async function getServerSideProps() {
  const response = await fetch("https://jsonplaceholder.typicode.com/posts");
  const posts = await response.json();

  return {
    props: {
      posts,
    }
  }
}
```

Incremental Static Regeneration

Nah metode ini biasa digunakan ketika data yang kamu tampilkan jarang mengalami perubahan. Sangat disarankan untuk menggunakan metode ini misalnya jika kamu ingin membuat 1 halaman static. Karena dengan menggunakan metode ini, halaman static-mu bisa jadi dinamis di background, jadi secara performa lebih kenceng.

Buat pakai metode ini, kamu hanya perlu membuat fungsi yang namanya **getStaticProps**, kalo misal paths dari halaman yang ingin dibuat itu di-generate secara terprogram, kamu bisa nambahin metode **getStaticPaths**

```
import Head from 'next/head'
import Image from 'next/image'
import styles from '../../styles/Contoh.module.css'

export default function Contoh({ title, body }) {
  return ( ... ComponentMu)
}

export async function getStaticPaths() {
  const response = await fetch("https://jsonplaceholder.typicode.com/posts");
  const posts = await response.json();

  const paths = posts.map(({ id }) => ({
    params: {
      id: id.toString(),
    },
  }))

  return {
    paths,
    fallback: true
  }
}
```

Nah kode di samping ini harus ditaruh di file yang sama yak!

Code ini bisa kalian lihat disini.

```
export async function getStaticProps({ params }) {
  const response = await
fetch("https://jsonplaceholder.typicode.com/posts/" +
params.id);
  const post = await response.json();

  return {
    props: {
      ...post,
    }
  }
}
```

Referensi

- https://nextjs.org/docs/basic-features/data-fetching/incremental-static-regeneration
- https://nextjs.org/docs/basic-features/pages
- https://www.youtube.com/watch?v=Sklc_fQBmcs

Terima Kasih!

Next Topic

loading...

BINAR