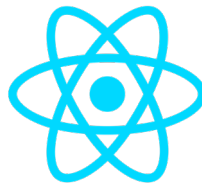


# CLIENT SIDE FRAMEWORK

# ~~NEXT~~.JS





Next.js adalah framework pengembangan web open-source yang dibangun di atas React dan dikembangkan oleh Vercel. Framework ini sangat populer karena fitur-fiturnya yang luar biasa, termasuk server-side rendering (SSR) dan static site generation (SSG).

Berikut adalah beberapa fitur utama Next.js:

1. **Server-Side Rendering (SSR):** Memungkinkan halaman web untuk dirender di server sebelum dikirim ke klien, yang meningkatkan performa dan SEO.
2. **Static Site Generation (SSG):** Membuat halaman statis pada waktu build, yang sangat cepat dan efisien untuk situs dengan konten yang jarang berubah.
3. **Routing Berbasis File Sistem:** Membuat rute dengan mudah menggunakan struktur direktori dan file.

4. **Data Fetching:** Mendukung pengambilan data baik di sisi server maupun klien dengan menggunakan `async/await`.
5. **Optimasi Built-in:** Termasuk optimasi gambar, font, dan skrip untuk meningkatkan pengalaman pengguna dan Core Web Vitals.
6. **Dukungan CSS:** Mendukung berbagai metode styling seperti CSS Modules, Tailwind CSS, dan CSS-in-JS

Next.js memungkinkan pengembang untuk membangun aplikasi web yang interaktif, dinamis, dan cepat dengan menggunakan komponen React

## 1. Instalasi

untuk melakukan instalasi gunakan command `create-next-app` :

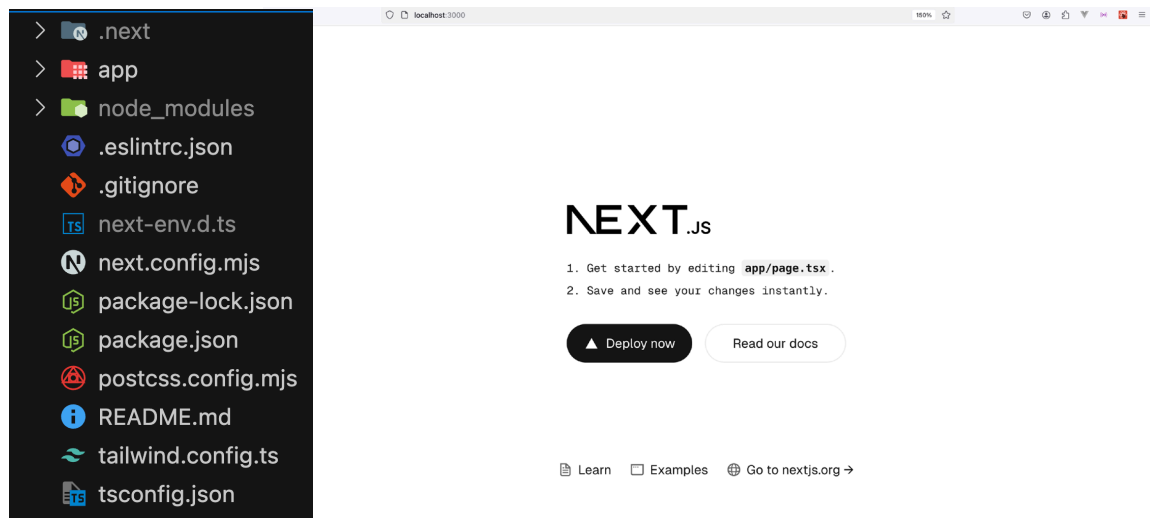
`npx create-next-app@latest`

pada saat proses instalasi akan menampilkan beberapa pertanyaan terkait dengan proses instalasi:

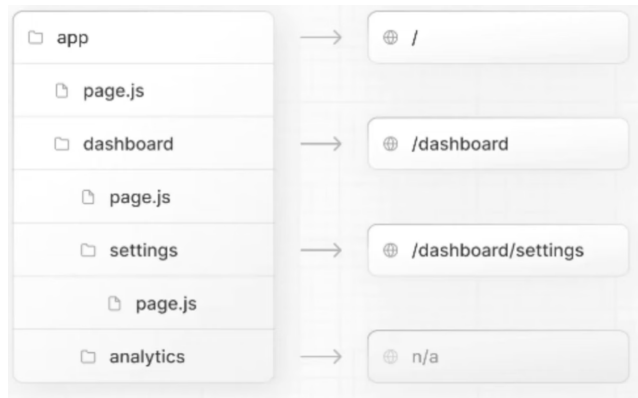
```
✓ What is your project named? ... latihan_next
✓ Would you like to use TypeScript? ... No / Yes
✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like to use `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to customize the default import alias (@/*)? ... No / Yes
Creating a new Next.js app in /Users/aliakbar/project/latihan guru/latihan_next.
```

untuk menjalankan next.js ketikkan perintah:

`npm run dev`



**2. Routing** next.js menggunakan router berbasis sistem folder dimana folder digunakan untuk menentukan route. setiap folder mewakili segmen route yang dipetakan ke segmen URL page.js, page.jsx, page.ts, page.tsx akan menjadi spesial file yang akan ditampilkan.



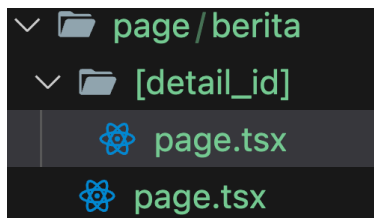
**3. Konfigurasi Awal** untuk konfigurasi kita akan menghapus isi file page.tsx dengan mengganti baris kode sebagai berikut:

```
export default function Home(){
  return(
    <h1>Home</h1>
  )
}
```

mengatur file global.css:

```
@tailwind base;
@tailwind components;
@tailwind utilities;
```

## 4. Dynamic Route



Dynamic routing di Next.js memungkinkan Anda untuk membuat rute yang dapat berubah berdasarkan parameter yang diberikan dalam URL. Ini sangat berguna untuk membuat halaman yang perlu menampilkan konten berbeda berdasarkan input, seperti profil pengguna, detail produk, atau posting blog. untuk membuat folder dynami route nama folder dibungkus dengan `[ nama_folder ]`. dsegmen dynamic router dapat diakses melalui props

```
export default function page({ params }: {
  params: {
    detail_id: string
  }
}) {
  return (
    <h1>detail halaman {params.detail_id} </h1>
  )
}
```

localhost:3000/page/berita/1

detail halaman 1

## 5. Nested Routes

nested routes adalah cara untuk membuat rute yang bersarang di dalam rute lain menggunakan struktur direktori. Next.js menggunakan sistem routing berbasis file, di mana setiap file di dalam direktori pages mewakili sebuah rute dalam aplikasi. Untuk membuat nested route, Anda cukup membuat sub-direktori di dalam direktori pages. Misalnya, jika Anda memiliki struktur seperti ini:

```
pages/
├── index.js
├── about.js
├── blog/
│   ├── index.js
│   └── [id].js
```



pages/index.js akan merujuk ke rute /.  
pages/about.js akan merujuk ke rute /about.  
pages/blog/index.js akan merujuk ke rute /blog.  
pages/blog/[id].js akan merujuk ke rute dinamis seperti /blog/1, /blog/2, dll.

```
export default function page({params}:{
  params:{
    id:string,
    detail_id:string
  }
}) {
  return (
    <div>halaman /{params.detail_id}/{params.id}</div>
  )
}
```

localhost:3000/page/berita/12/section/12  
halaman /12/12

## 6. Catch-All Segment

catch-all segment adalah fitur yang memungkinkan Anda menangani berbagai pola URL dengan satu rute. Ini sangat berguna ketika Anda tidak tahu berapa banyak segmen yang akan ada dalam URL. Untuk membuat catch-all segment, Anda perlu menambahkan tiga titik (...) di dalam tanda kurung siku dalam nama file. Misalnya:

```
pages/
├── shop/
│   └── [...slug].js
```

## 7. PageNotFound

untuk membuat file notfound cukup dengan membuat file not-found.tsx dalam folder utama. secara default next.js sudah memiliki halaman notfound default akan tetapi jika ingin melakukan costum file dapat membuat file sendiri

```
export default function NotFound() {
  return (
    <div>
      <h1>ups halaman tidak ditemukan</h1>
    </div>
  )
}
```

ups halaman tidak ditemukan

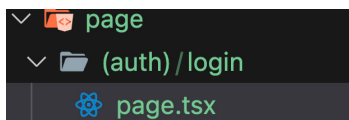
## 8. Privat folder

merupakan optional folder yang memisahkan diri dari routing system cara membuatnya adalah dengan menambahkan underscore didepan nama folder sebagai contoh `_namaFolder`. semua child segmen dari privat folder tidak dapat diakses oleh system routing.

## 9. Route Group

route groups adalah cara untuk mengorganisir rute dalam direktori aplikasi tanpa mengubah struktur URL. Ini sangat berguna untuk menjaga kode tetap teratur dan mudah dikelola. Untuk membuat route group, Anda cukup membungkus nama folder dengan tanda kurung. Misalnya:

```
app/
├── (marketing)/
│   ├── page.js
│   └── about.js
└── (shop)/
    ├── page.js
    └── product.js
```



```
export default function Login() {
  return (
    <div>Login</div>
  )
}
```



localhost:3000/page/login

Login