

[FE] Study Jam 1 - Eksplorasi Next JS

1. Persiapan

Untuk mengikuti pembelajaran ini kamu perlu menyiapkan:

- Install **Node JS** (<https://nodejs.org/en>)
- **VS Code** untuk text editor, atau editor kesukaanmu (<https://code.visualstudio.com/>). Pastikan menginstall Plugin:
 - **ES7+ React/Redux/React-Native snippets**
 - **Tailwind CSS IntelliSense**
 - **Prettier - Code formatter**

Semua contoh code dapat dilihat juga di:

https://github.com/anjarmath/belajar_nextjs

2. Pengenalan Next.js

Next.js (<https://nextjs.org/>) adalah framework React yang powerful untuk membangun aplikasi web modern. Framework ini menyediakan solusi lengkap untuk pengembangan aplikasi dengan fitur-fitur seperti server-side rendering, routing otomatis, dan optimasi performa. Dengan Next.js, developer dapat fokus pada pengembangan fitur tanpa perlu mengkhawatirkan konfigurasi yang kompleks.

Beberapa perbedaan utama antara Next.js dan React.js biasa terletak pada cara rendering dan routing. React.js adalah library UI yang membutuhkan konfigurasi tambahan untuk fitur-fitur seperti routing dan server-side rendering (SSR),

sementara Next.js menyediakan solusi all-in-one dengan fitur-fitur tersebut built-in. Next.js juga menawarkan performa yang lebih baik melalui automatic code splitting dan optimisasi gambar.

Penting dipelajari: Perbedaan Server Side Rendering (SSR) dan Client Side Rendering (CSR)

3. Membuat Proyek Next.js Pertama

Untuk membuat proyek Next.js baru, jalankan perintah berikut di terminal:

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

Masukkan konfigurasi seperti berikut:

```
✓ 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 your code inside a `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to use Turbopack for `next dev`? ... No / Yes
✓ Would you like to customize the import alias (`@/*` by default)? ... No / Yes
```

Untuk run proyek masukkan perintah:

```
npm run dev
```

Berikut adalah struktur folder dasar dalam proyek Next.js:

```
coba-next-js/
├── .next/
└── node_modules/
```

```
├── public/
│   ├── next.svg
│   └── vercel.svg
├── src/
│   ├── app/
│   │   ├── favicon.ico
│   │   ├── globals.css
│   │   ├── layout.jsx
│   │   └── page.jsx
│   └── components/
├── .eslintrc.json
├── .gitignore
├── jsconfig.json
├── next.config.js
├── package-lock.json
├── package.json
├── postcss.config.js
└── tailwind.config.js
```

Beberapa folder dan file penting:

- **app/** - Folder utama yang menggunakan App Router, tempat semua halaman dan komponen routing berada
- **components/** - Folder untuk menyimpan komponen React yang dapat digunakan ulang di berbagai halaman
- **public/** - Untuk menyimpan file statis seperti gambar yang dapat diakses langsung dari browser

Selengkapnya: <https://nextjs.org/docs/app/getting-started/project-structure>

4. Praktik 1: Routing, Layout, Komponen, dan Asset

A. Routing

Pertama, hapus konten dari `page.jsx`, dan ubah menjadi seperti berikut:

```
export default function Home() {
  return (
    <div className=" w-full h-screen flex flex-col gap-3 justify-center items-center">
      <h1 className=" font-bold text-xl">Selamat Datang!</h1>
    </div>
  );
}
```

Di dalam folder `app/`, kamu bisa membuat route dengan membuat folder dan file `page.jsx`. Contoh:

```
app
├── (auth)          <--- Group Routing
│   ├── signup
│   └── page.jsx
├── blog            <--- Basic Routing
│   ├── page.jsx
│   └── [slug]      <--- Dynamic Routing
│       └── page.jsx
├── ...
└── page.jsx
```

- Basic Routing

Sederhananya, jika kamu ingin membuat halaman misalnya

`https://mydomain.com/blog/`, maka kamu perlu membuat folder `/blog` lalu di dalam folder buat file `page.tsx` sehingga menjadi `/blog/page.jsx`.

- Dynamic Routing

Dinamic routing dipakai untuk menampilkan halaman yang isinya dinamis, misalnya halaman blog post. Sebagai contoh di atas terdapat

`/blog/[slug]/page.jsx` (perhatikan bahwa slug diapit kurung siku) maka akan

menghasilkan halaman `/blog/[judul_blog]` yang isinya akan menyesuaikan apapun yang diisikan di `judul_blog` pada address bar.

- Group Routing

Group routing dipakai untuk mengelompokkan route yang masih satu konteks, misalnya halaman login, daftar (seperti contoh di atas), lupa password, dsb masih satu konteks sehingga bisa dikelompokkan di folder auth supaya lebih terorganisir dengan nama folder diapit kurung menjadi contohnya

`(auth)/signup/page.jsx`. Dalam browser, folder (auth) akan diabaikan/dilewati, sehingga hanya akan menjadi halaman `/signup`.

Berikut code lengkap folder blog:

```
//app/blog/page
import Link from "next/link";
import React from "react";

const BlogPage = () => {
  const blogs = ["artikel_1", "artikel_2", "artikel_3", "artikel_4", "artikel_5"];
  return (
    <div className=" px-6 py-4">
      <ol>
        {blogs.map((post, index) => (
          <li key={index}>
            <Link href={` /blog/${post}`} className=" underline"
              {post}
            </Link>
          </li>
        ))}
      </ol>
    </div>
  );
};
```

```
export default BlogPage;
```

```
//app/blog/[slug]/page.jsx
import React, { use } from "react";

const BlogPost = ({ params }) => {
  const { slug } = use(params);
  return (
    <div className=" flex w-full h-screen justify-center items-center">
      Blog Post: {slug}
    </div>
  );
};

export default BlogPost;
```

B. Component

buat folder `component` di dalam src lalu buat file `Button.jsx`

```
import React from "react";

const MyButton = ({ children, onClick }) => {
  return (
    <button
      className=" bg-blue-500 text-white rounded-md px-5 py-3
      hover:bg-blue-700"
      onClick={onClick}
    >
      {children}
    </button>
  );
};
```

```
export default MyButton;
```

Lalu tambahkan ke dalam home (`/app/page.tsx`):

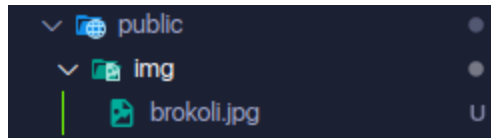
```
"use client";
// Tambahkan use client sebab kita perlu interaksi dengan browser
import MyButton from "@/components/Button";
import { useState } from "react";

export default function Home() {
  const [counter, setCounter] = useState(0);
  return (
    <div className="w-full h-screen flex flex-col gap-3 justify-center items-center">
      <h1 className="font-bold text-xl">Selamat Datang!</h1>

      {counter}
      <MyButton
        onClick={() => {
          setCounter((prev) => prev + 1);
        }}
      >
        Coba Tombol
      </MyButton>
    </div>
  );
}
```

C. Asset

Tambahkan sebuah gambar di folder `public/img`. Berikut contohnya:



Lalu tambahkan komponen `Image` bawaan next js, seperti berikut di `/app/page.jsx`

```
import Image from "next/image";
...

<Image
  src={"/img/brokoli.jpg"}
  width={200}
  height={200}
  alt="Gambar Brokoli"
/>
...
```

D. Layout

Layout dalam Next.js memungkinkan kita untuk membuat tampilan yang konsisten di seluruh halaman. Layout didefinisikan dalam file `layout.jsx` dan akan membungkus semua halaman yang ada di bawahnya. Secara default, Next.js sudah menyediakan layout di `app/layout.jsx`.

Kita bisa membuat layout tambahan dengan membuat file `layout.jsx` di folder yang diinginkan. Layout tersebut hanya akan mempengaruhi halaman-halaman yang berada di dalam folder tersebut.

Mari buat komponen lagi bernama `NavBar.jsx`

```
import Link from "next/link";
import React from "react";

const NavBar = () => {
  return (
    <div className="bg-blue-500 w-full flex gap-4 justify-between items-center px-6 py-3 sticky top-0">
```



```

    <Link className=" font-bold text-lg text-white" href=
    {"/"}>
      HOME
    </Link>

    <div className=" flex gap-4 items-center">
      <Link href={"/blog"} className=" underline">
        Blog
      </Link>
      <Link href={"/login"} className=" font-bold underlin
e">
        Login
      </Link>
      <Link href={"/signup"} className=" font-bold underlin
e">
        Daftar
      </Link>
    </div>
  </div>
);
};

export default NavBar;

```

Lalu Tambahkan ke `layout.jsx`

```

// app/layout.jsx
export default function RootLayout({ children }) {
  return (
    <html lang="en">
      <body
        className={` ${geistSans.variable} ${geistMono.variable}
      >
        <NavBar /> <---- tambahkan, jangan lupa import
        {children}
      </body>

```

```

    </html>
  );
}

```

5. Praktik 2: Membuat Form Sederhana

Pada `/app/(auth)/signup/page.jsx`

```

import MyButton from "@components/Button";
import React from "react";

const SignupPage = () => {
  const onSubmit = async (formData) => {
    "use server";
    console.log(formData);
  };
  return (
    <div className=" px-6 py-4">
      <div className=" space-y-3 max-w-2xl mx-auto">
        <h1 className=" font-bold text-xl">Daftar Akun</h1>
        <form action={onSubmit} className=" flex flex-col gap
-3">
          <input
            placeholder="Username"
            id="username"
            name="username"
            type="text"
            required
            className=" border-2"
          />
          <input
            placeholder="Nama Lengkap"
            id="name"
            name="name"
            type="text"

```

```
        required
        className=" border-2"
      />

      <MyButton>Daftar</MyButton>
    </form>
  </div>
</div>
);
};

export default SignupPage;
```

6. Apa Selanjutnya?

- Menggunakan Library Untuk Memudahkan Development
- Integrasi dengan Backend API