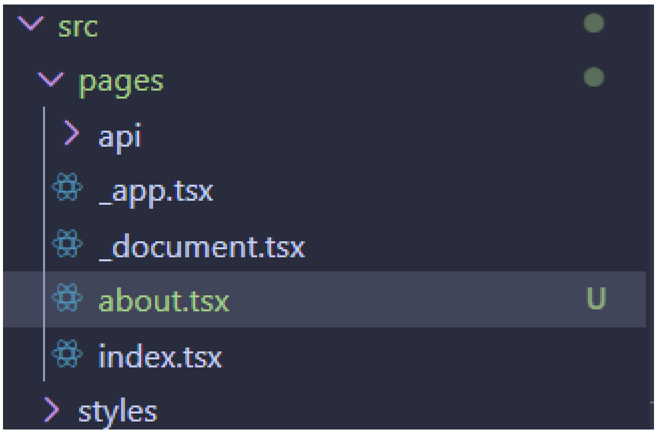
KEMENTERIAN PENDIDIKAN, KEBUDAYAAN, RISET, DAN TEKNOLOGI
POLITEKNIK NEGERI MALANG
JURUSAN TEKNOLOGI INFORMASI
Jl. Soekarno Hatta No. 9, Jatimulyo, Lowokwaru, Malang 65141
Telp. (0341) 404424 – 404425, Fax (0341) 404420
https://jti.polinema.ac.id

Mata Kuliah	:	Pemrograman Framework
Program Studi	:	D4 – Teknik Informatika
Semester	:	6

Kelas	:	TI-3E
NIM	:	2341720219
Nama	:	Satrio Wisnu Adi Pratama
Jobsheet Ke-	:	3
Link Github	:	**https://github.com/flywithsaturn/pemrograman-berbasis-framework/tree/main/praktikum-2/next-routing**

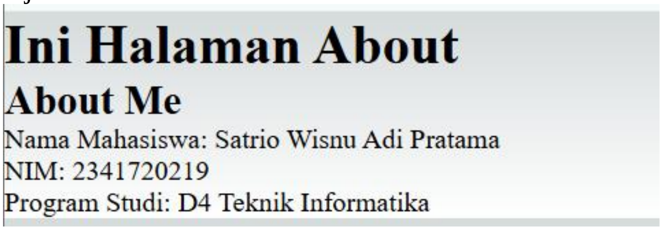
Laporan Jobsheet

1. **Routing Dasar (Static Routing)**
   Tambahkan Halaman About



```
src
  pages
    api
    _app.tsx
    _document.tsx
    about.tsx        U
    index.tsx
  styles
```

Uji di Browser

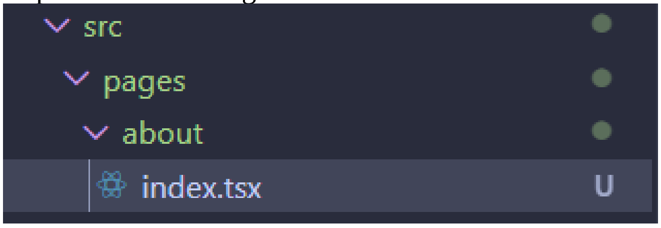


Ini Halaman About

About Me

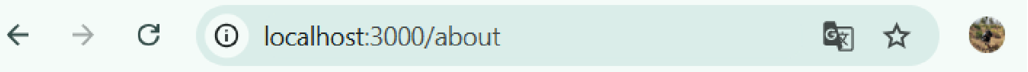
Nama Mahasiswa: Satrio Wisnu Adi Pratama
NIM: 2341720219
Program Studi: D4 Teknik Informatika

2. **Routing Menggunakan Folder**
   Rapikan Struktur Pages



```
src
  pages
    about
      index.tsx      U
```

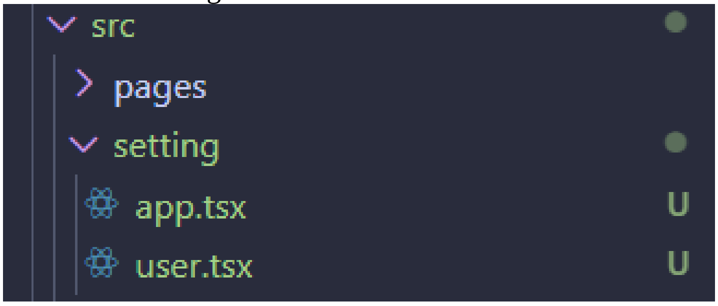
Akses dari halaman browser (tetap sama tetapi lebih rapi)



localhost:3000/about

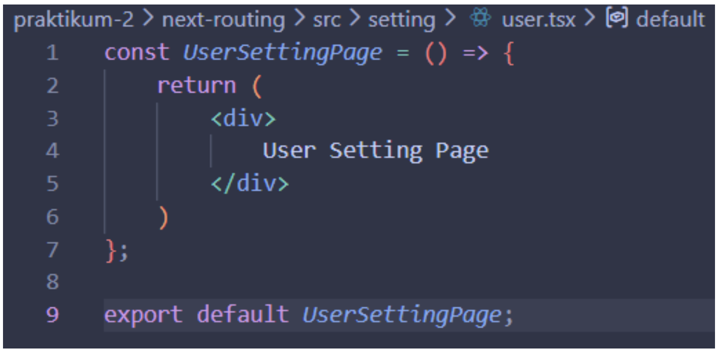
3. Nested Routing

Buat folder setting



```
src
  pages
  setting
    app.tsx     U
    user.tsx    U
```

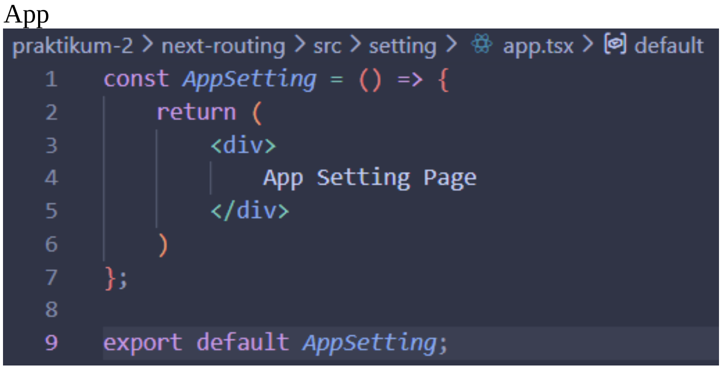
user



praktikum-2 > next-routing > src > setting > user.tsx > default

```
const UserSettingPage = () => {
    return (
        <div>
            User Setting Page
        </div>
    )
};

export default UserSettingPage;
```

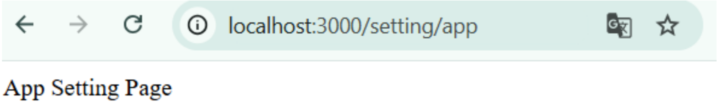
App



praktikum-2 > next-routing > src > setting > app.tsx > default

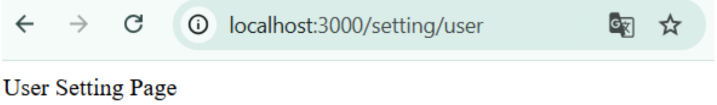
```
const AppSetting = () => {
    return (
        <div>
            App Setting Page
        </div>
    )
};

export default AppSetting;
```

Hasil



localhost:3000/setting/app

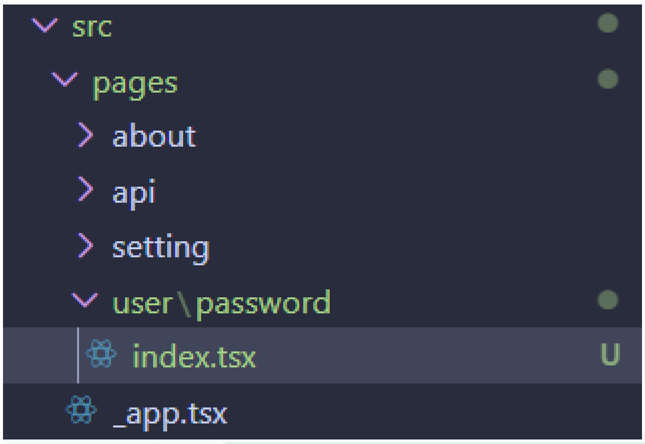
App Setting Page



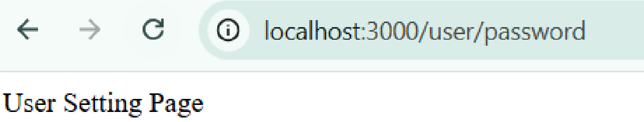
localhost:3000/setting/user

User Setting Page

Nested lebih dalam



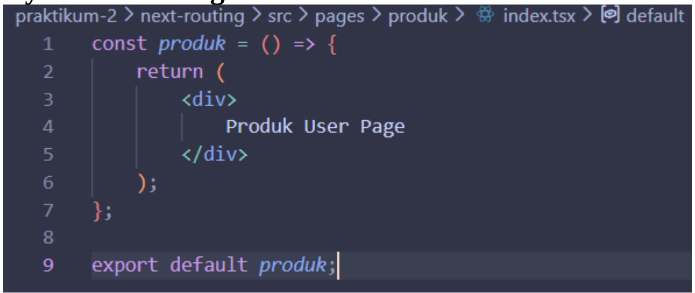
```
src
  pages
    about
    api
    setting
    user\password
      index.tsx        U
    _app.tsx
```



localhost:3000/user/password

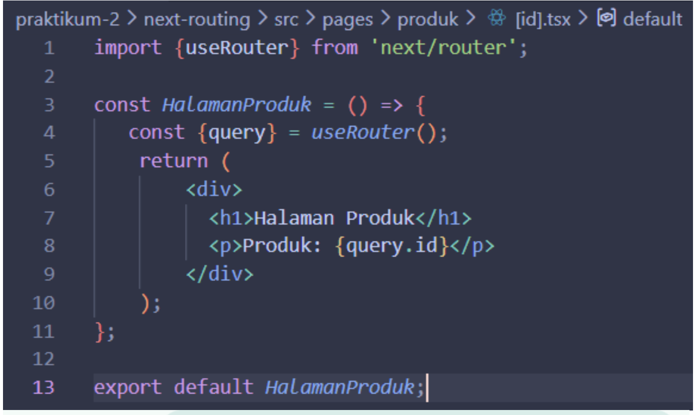
User Setting Page

4. **Dynamic Routing**



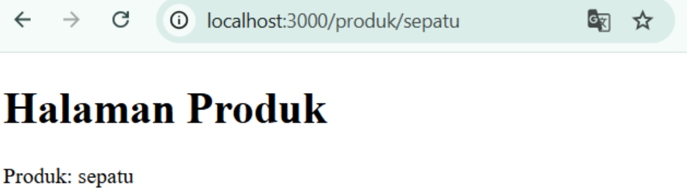
praktikum-2 > next-routing > src > pages > produk > index.tsx > default

```tsx
const produk = () => {
    return (
        <div>
            Produk User Page
        </div>
    );
};

export default produk;
```



praktikum-2 > next-routing > src > pages > produk > [id].tsx > default

```tsx
import {useRouter} from 'next/router';

const HalamanProduk = () => {
   const {query} = useRouter();
    return (
        <div>
          <h1>Halaman Produk</h1>
          <p>Produk: {query.id}</p>
        </div>
    );
};

export default HalamanProduk;
```

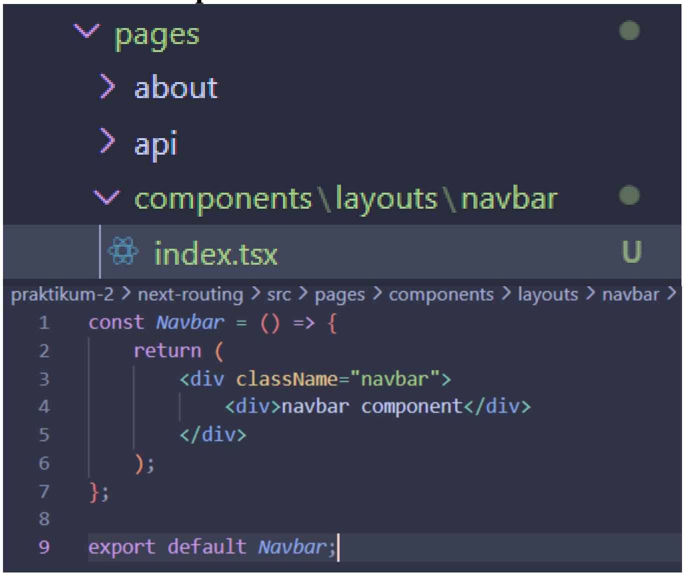


localhost:3000/produk/sepatu

Halaman Produk

Produk: sepatu

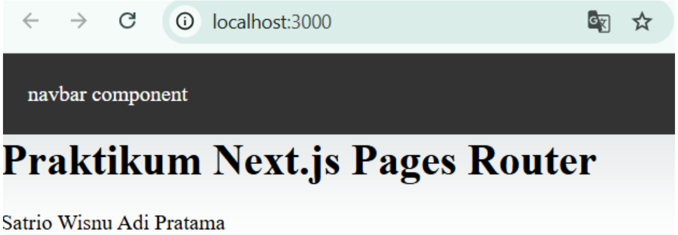
5. Membuat Komponen Navbar



```
pages
  about
  api
  components\layouts\navbar
    index.tsx                U
```

praktikum-2 > next-routing > src > pages > components > layouts > navbar >

```tsx
const Navbar = () => {
    return (
        <div className="navbar">
            <div>navbar component</div>
        </div>
    );
};

export default Navbar;
```

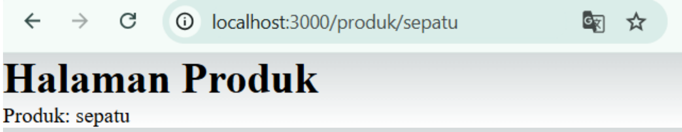


localhost:3000

navbar component

Praktikum Next.js Pages Router

Satrio Wisnu Adi Pratama

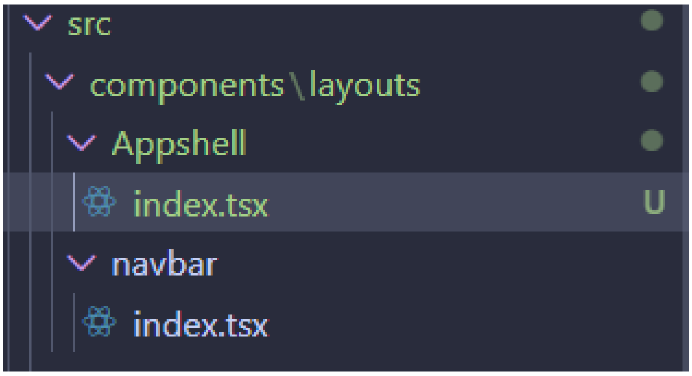


localhost:3000/produk/sepatu

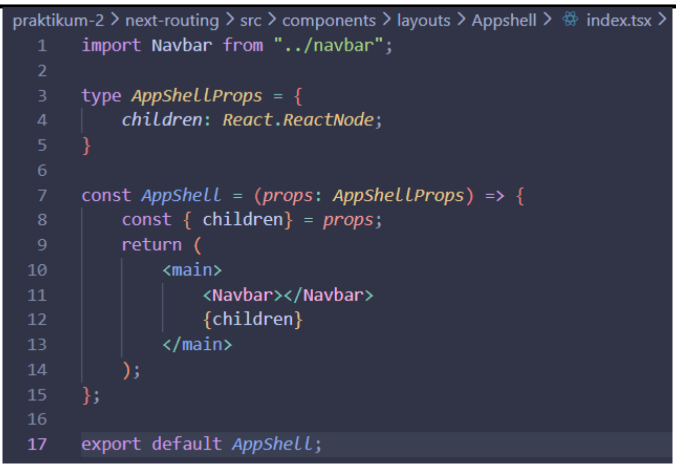
Halaman Produk

Produk: sepatu

6. Membuat Layout Global (App Shell)



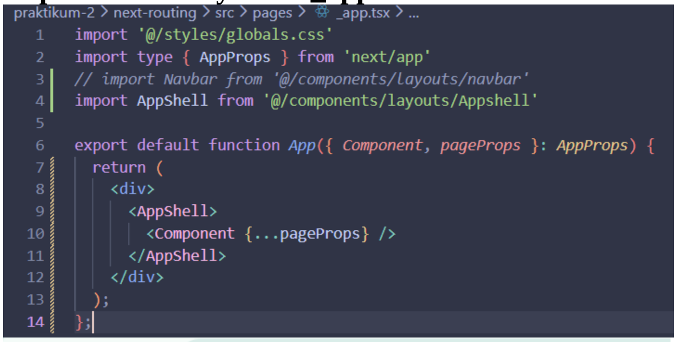
```
src
  components\layouts
    Appshell
      index.tsx              U
    navbar
      index.tsx
```



praktikum-2 > next-routing > src > components > layouts > Appshell > index.tsx >

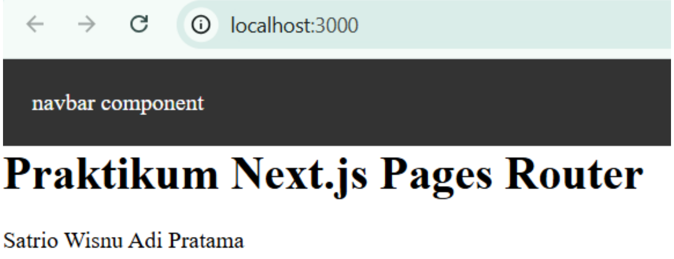
```
import Navbar from "../navbar";

type AppShellProps = {
    children: React.ReactNode;
}

const AppShell = (props: AppShellProps) => {
    const { children} = props;
    return (
        <main>
            <Navbar></Navbar>
            {children}
        </main>
    );
};

export default AppShell;
```

7. **Implementasi Layout di _app.tsx**



praktikum-2 > next-routing > src > pages > _app.tsx > ...

```
import '@/styles/globals.css'
import type { AppProps } from 'next/app'
// import Navbar from '@/components/layouts/navbar'
import AppShell from '@/components/layouts/Appshell'

export default function App({ Component, pageProps }: AppProps) {
  return (
    <div>
      <AppShell>
        <Component {...pageProps} />
      </AppShell>
    </div>
  );
};
```



localhost:3000

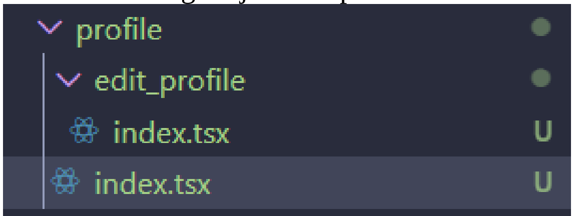
navbar component

Praktikum Next.js Pages Router

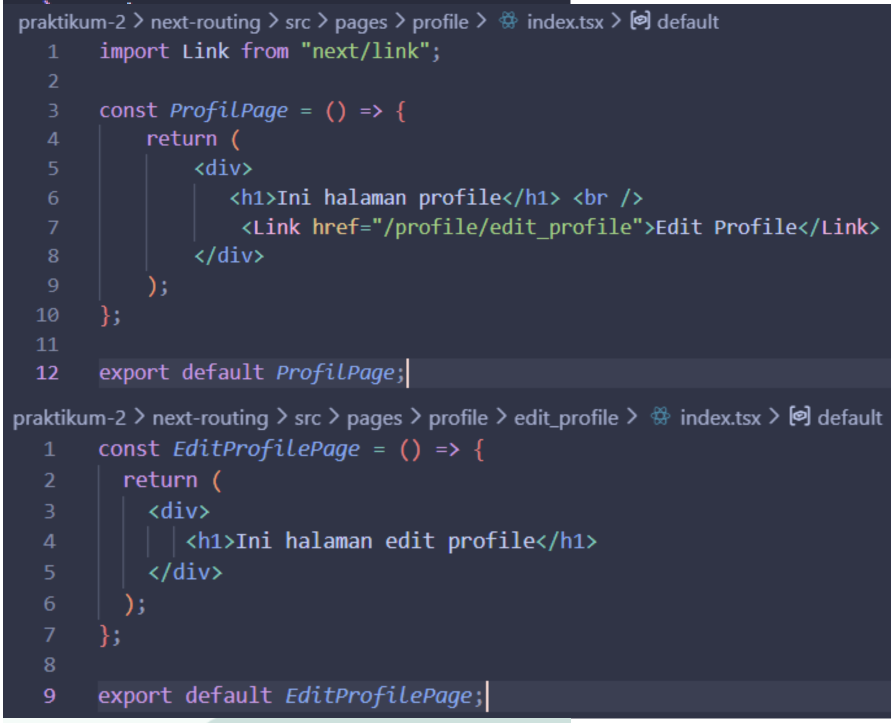
Satrio Wisnu Adi Pratama

Tugas Praktikum

1. **Tugas 1 – Routing**
    1. Buat halaman:
        - /profile
        - /profile/edit 2
    2. Pastikan routing berjalan tanpa error



```
profile
  edit_profile
    index.tsx    U
  index.tsx      U
```

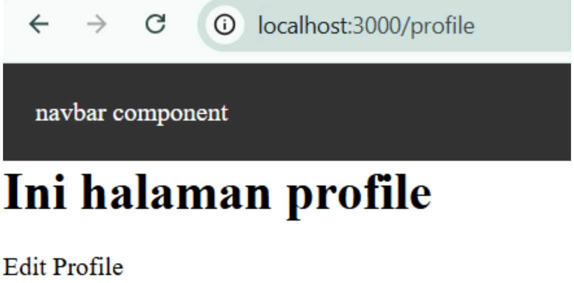


praktikum-2 > next-routing > src > pages > profile > index.tsx > default

```tsx
import Link from "next/link";

const ProfilPage = () => {
    return (
        <div>
            <h1>Ini halaman profile</h1> <br />
             <Link href="/profile/edit_profile">Edit Profile</Link>
        </div>
    );
};

export default ProfilPage;
```

praktikum-2 > next-routing > src > pages > profile > edit_profile > index.tsx > default

```tsx
const EditProfilePage = () => {
  return (
    <div>
      <h1>Ini halaman edit profile</h1>
    </div>
  );
};

export default EditProfilePage;
```

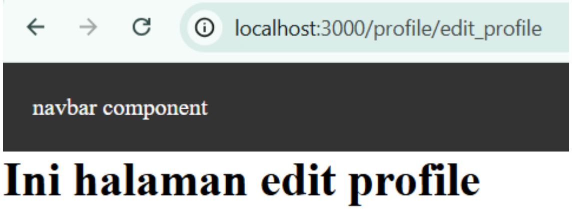


localhost:3000/profile

navbar component

Ini halaman profile

Edit Profile



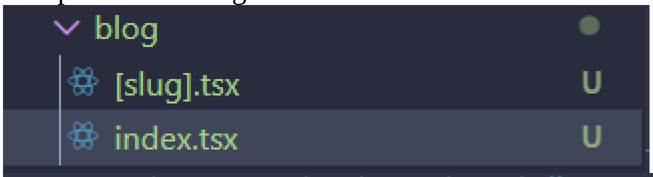
localhost:3000/profile/edit_profile

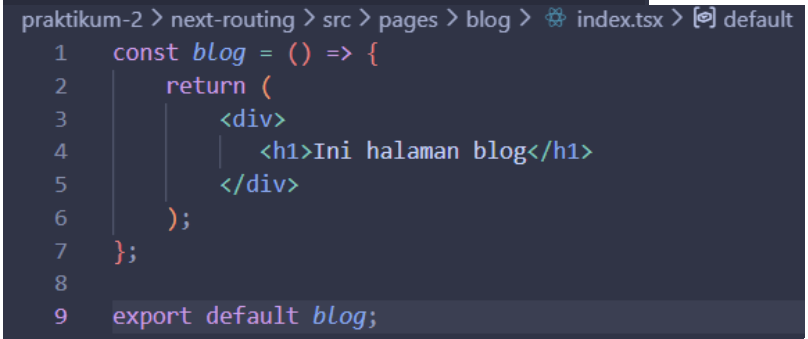
navbar component

Ini halaman edit profile

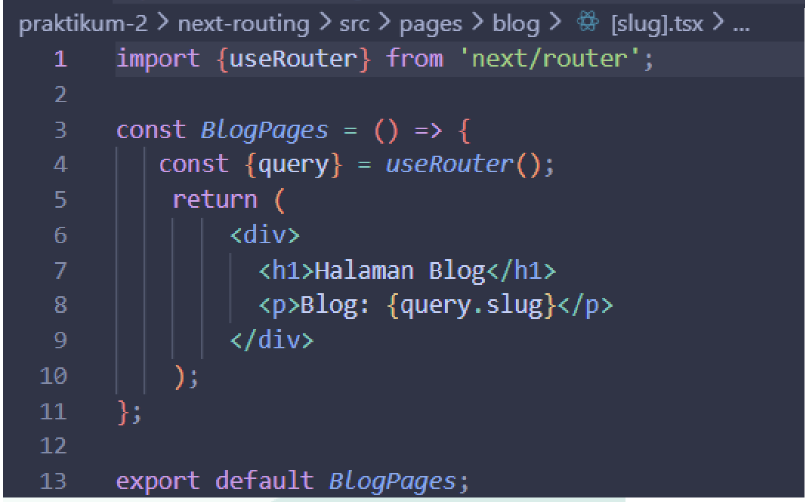
2. Tugas 2 – Dynamic Routing

1. Buat routing:
2. /blog/[slug]
3. Tampilkan nilai slug di halaman

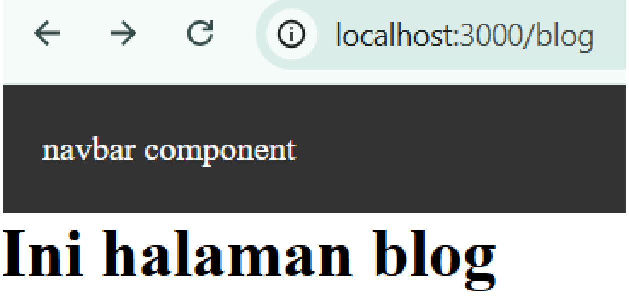



praktikum-2 > next-routing > src > pages > blog > index.tsx > default

```
const blog = () => {
    return (
        <div>
            <h1>Ini halaman blog</h1>
        </div>
    );
};

export default blog;
```

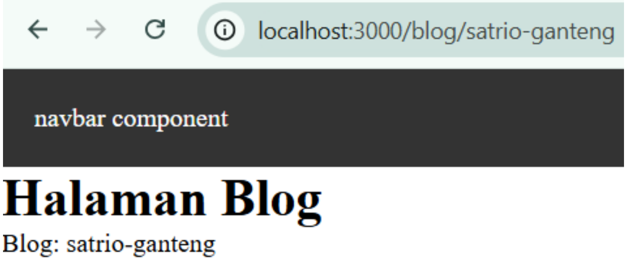

praktikum-2 > next-routing > src > pages > blog > [slug].tsx > ...

```
import {useRouter} from 'next/router';

const BlogPages = () => {
   const {query} = useRouter();
    return (
        <div>
            <h1>Halaman Blog</h1>
            <p>Blog: {query.slug}</p>
        </div>
    );
};

export default BlogPages;
```



navbar component

Ini halaman blog



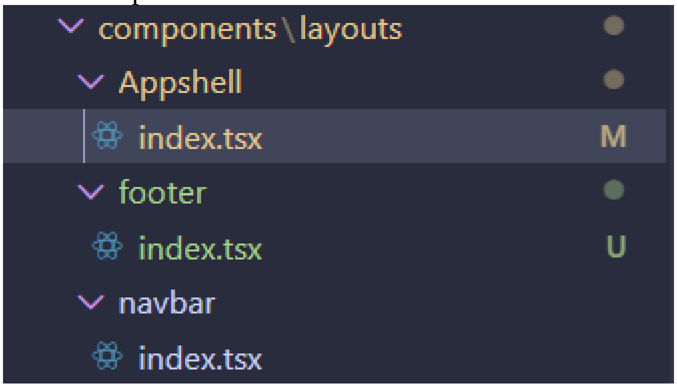
navbar component

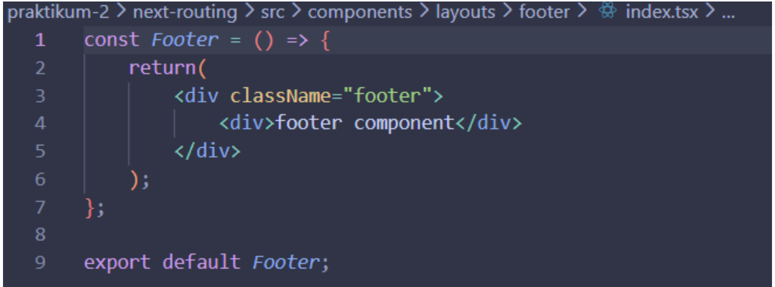
Halaman Blog

Blog: satrio-ganteng

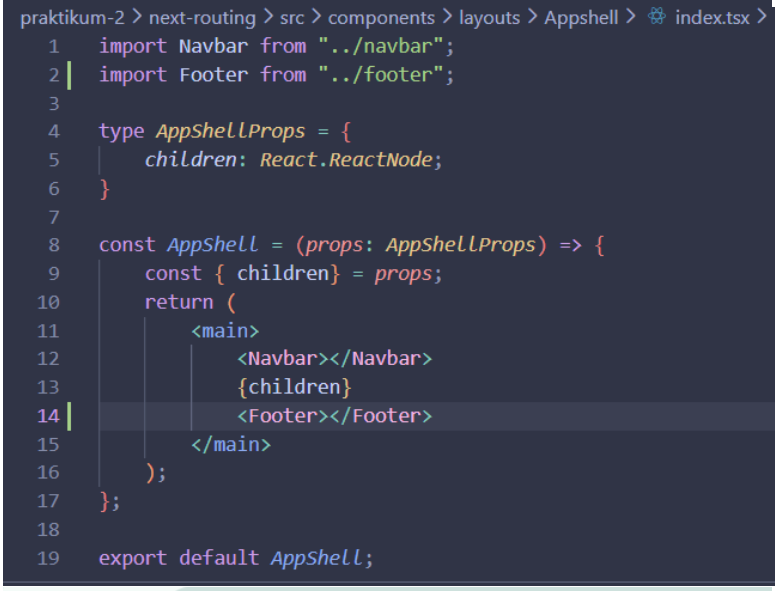
3. Tugas 3 – Layout

1. Tambahkan Footer pada AppShell
2. Footer tampil di semua halaman

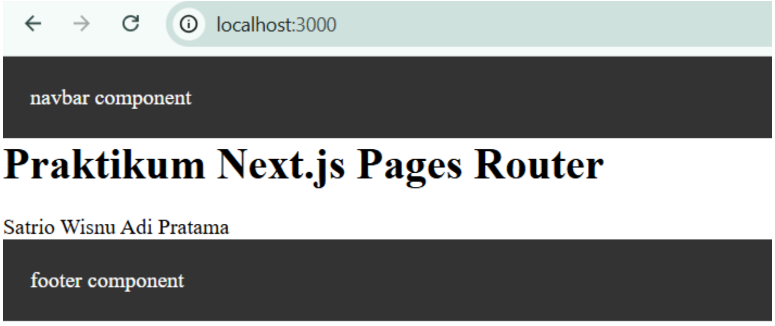



praktikum-2 > next-routing > src > components > layouts > footer > index.tsx > ...

```tsx
const Footer = () => {
    return(
        <div className="footer">
            <div>footer component</div>
        </div>
    );
};

export default Footer;
```


praktikum-2 > next-routing > src > components > layouts > Appshell > index.tsx >

```tsx
import Navbar from "../navbar";
import Footer from "../footer";

type AppShellProps = {
    children: React.ReactNode;
}

const AppShell = (props: AppShellProps) => {
    const { children} = props;
    return (
        <main>
            <Navbar></Navbar>
            {children}
            <Footer></Footer>
        </main>
    );
};

export default AppShell;
```



navbar component

Praktikum Next.js Pages Router

Satrio Wisnu Adi Pratama

footer component

Pertanyaan Refleksi

1. Apa perbedaan routing berbasis file dan routing manual?
   Jawab: Routing berbasis file adalah sistem routing yang otomatis mengikuti struktur folder dan nama file. Jadi cukup buat file, route langsung terbentuk sedangkan Routing manual adalah kita harus menuliskan sendiri satu per satu alamat (route) di dalam kode.
2. Mengapa dynamic routing penting dalam aplikasi web?
   Jawab: Dynamic routing penting karena bisa menampilkan halaman berbeda berdasarkan data atau parameter tertentu, misalnya halaman detail produk berdasarkan ID. Jadi tidak perlu membuat halaman satu per satu secara manual.
3. Apa keuntungan menggunakan layout global dibanding memanggil komponen satu per satu?
   Jawab: Layout global membuat tampilan seperti header dan footer cukup dibuat sekali dan otomatis dipakai di semua halaman. Lebih rapi, hemat kode, dan mudah dikelola.

Kesimpulan

Melalui praktikum ini, mahasiswa telah memahami bahwa Next.js Pages Router:

- Menghemat waktu konfigurasi routing
- Mendukung nested dan dynamic routing secara natural
- Memudahkan pengelolaan layout global menggunakan _app.tsx