

**LAPORAN JOBSHEET 2**  
**Pemrograman Berbasis Framework**  
**Page dan Layout**



**NAMA :**

**LUTHFI TRIASWANGGA**

**2341720208**

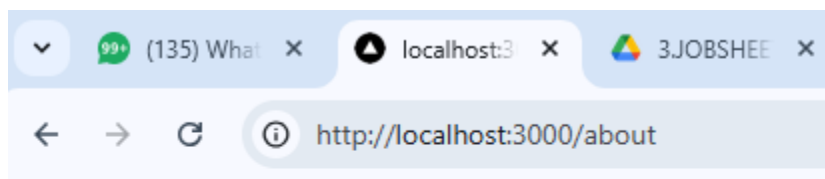
**Jurusan Teknologi Informasi**  
**D4 Teknik Informatika**  
**Politeknik Negeri Malang**  
**2025**

## 1. Routing Dasar (Static Routing)

Kode program untuk about.tsx

```
export default function About() {  
  return (  
    <div>  
      <h1>Ini Halaman About</h1>  
      <p>Ini adalah halaman tentang saya.</p>  
    </div>  
  );  
}
```

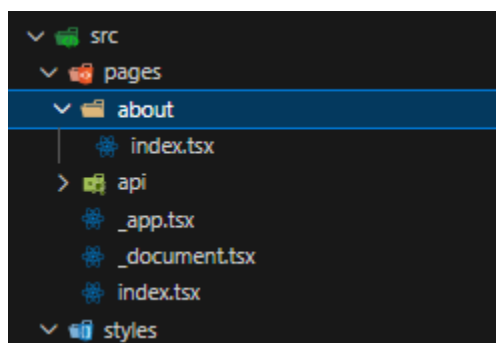
Dan hasil nya seperti ini



## Ini Halaman About

Ini adalah halaman tentang saya.

## 2. Routing Menggunakan Folder

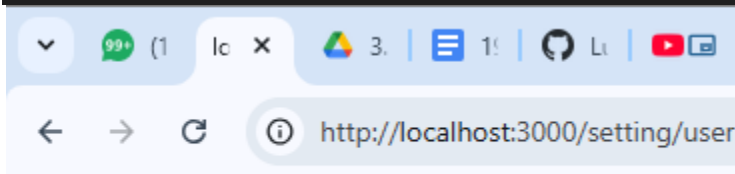


### 3. Nested Routing

#### a. Buat Folder Setting

##### User.tsx

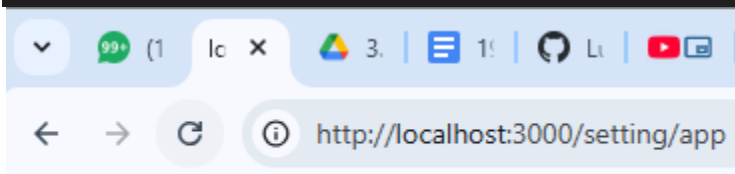
```
const UserSettingPage = () => {  
  return <div>User Setting Page</div>;  
}  
  
export default UserSettingPage;
```



User Setting Page

##### App.tsx

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

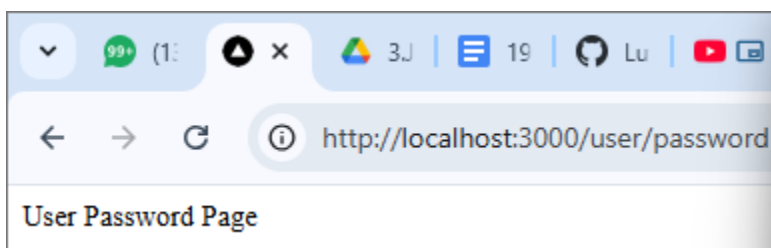


App Setting Page

#### b. Nested Lebih Dalam

##### Password.tsx

```
const UserPasswordPage = () => {  
  return <div>User Password Page</div>;  
}  
  
export default UserPasswordPage;
```



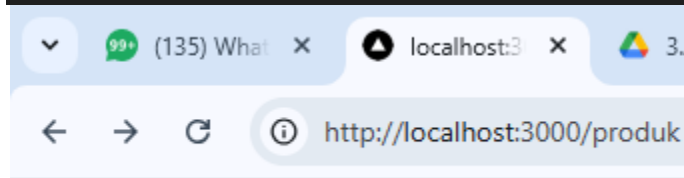
## 4. Dynamic Routing

### a. Buat Halaman Produk

Produk

Index.tsx

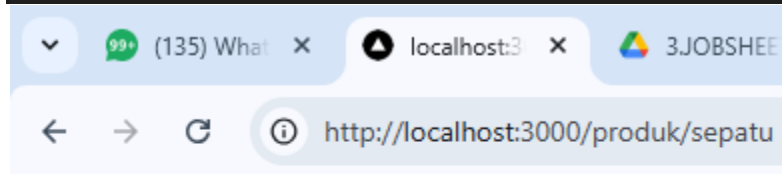
```
const produk = () => {  
  return (  
    <div>Produk User Page</div>  
  )  
}  
  
export default produk;
```



Produk User Page

[id].tsx

```
import {Router, useRouter} from 'next/router';  
  
const HalamanProduk = () => {  
  const Router = useRouter();  
  console.log(Router);  
  return (  
    <div>  
      <h1>Halaman Produk</h1>  
      <p>Produk :</p>  
    </div>  
  )  
}  
  
export default HalamanProduk;
```



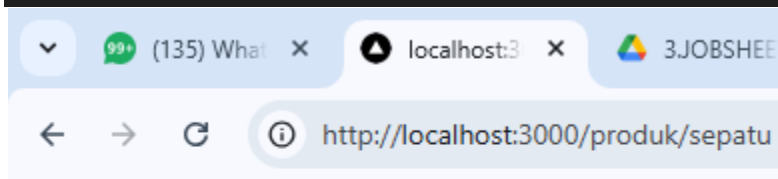
# Halaman Produk

Produk :

## B. Modifikasi [id].tsx agar dapat mengambil nilai dari id

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

```
const HalamanProduk = () => {  
  // const Router = useRouter();  
  // console.log(Router);  
  const { query } = useRouter();  
  return (  
    <div>  
      <h1>Halaman Produk</h1>  
      <p>Produk : {query.id}</p>  
    </div>  
  )  
}  
  
export default HalamanProduk;
```

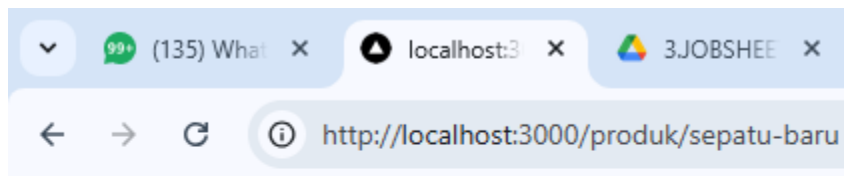


# Halaman Produk

Produk : sepatu

## C. Uji di Browser

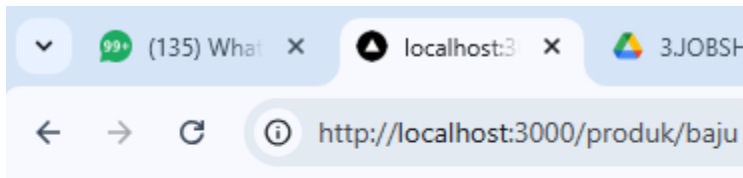
/produk/sepatu-baru



# Halaman Produk

Produk : sepatu-baru

/produk/baju



## Halaman Produk

Produk : baju

### 5. Membuat Komponen Navbar

\_app.tsx

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

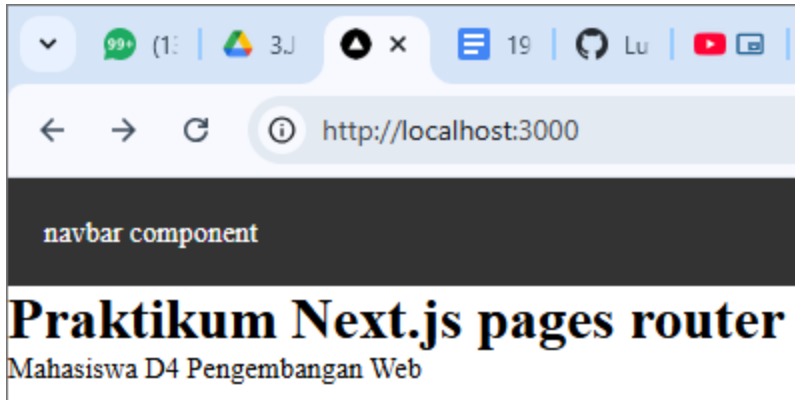
export default function App({ Component, pageProps }: AppProps) {
  return (
    <>
      <Navbar />
      <Component {...pageProps} />
    </>
  )
}
```

Index.tsx

```
import Head from 'next/head'
import Image from 'next/image'
import { Inter } from 'next/font/google'
import styles from '@styles/Home.module.css'

const inter = Inter({ subsets: ['latin'] })

export default function Home() {
  return (
    <div>
      <h1>Praktikum Next.js pages router</h1>
      <p>Mahasiswa D4 Pengembangan Web</p>
    </div>
  );
}
```



## 6. Membuat Layout Global (App Shell)

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

type AppShellProps = {
  children: React.ReactNode;
}

const AppShell = (props: AppShellProps) => {
  const { children } = props;

  return (
    <main>
      <Navbar />
      {children}
    </main>
  );
};

export default AppShell;
```

## 7. Implementasi Layout di \_app.tsx

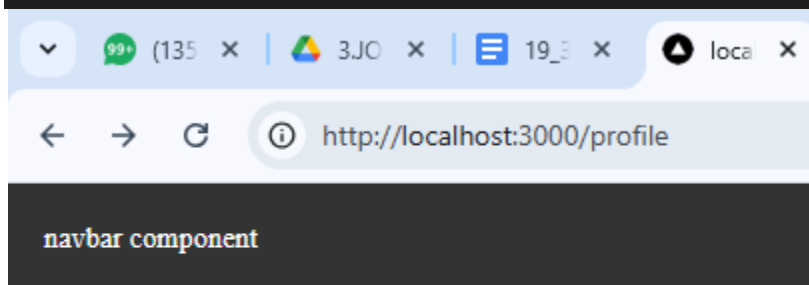
```
import '@styles/globals.css'
import type { AppProps } from 'next/app'
import AppShell from '@components/layouts/AppShell'
import Navbar from '@components/layouts/navbar'

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

## Tugas 1 – Routing

### Profile-index.tsx

```
export default function Profile() {
  return (
    <div style={{ padding: "20px" }}>
      <h1>Ini halaman profil</h1>
      <p>Halaman ini digunakan untuk menampilkan profil pengguna.</p>
    </div>
  );
}
```



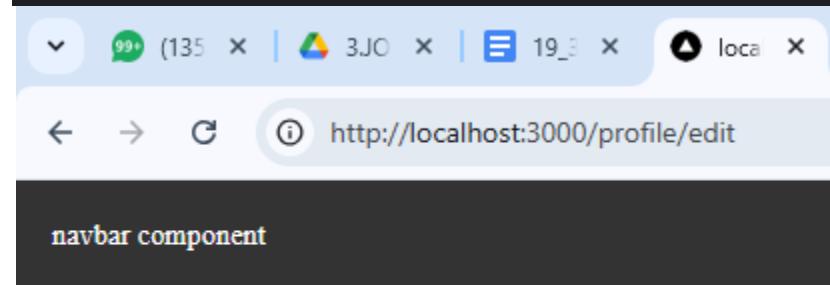
## Ini halaman profile

Halaman ini digunakan untuk menampilkan profil pengguna.



profile/edit-index.tsx

```
export default function EditProfile() {  
  return (  
    <div style={{ padding: "20px" }}>  
      <h1>Ini halaman edit profile</h1>  
      <p>Halaman ini digunakan untuk mengedit profil pengguna.</p>  
    </div>  
  );  
}
```



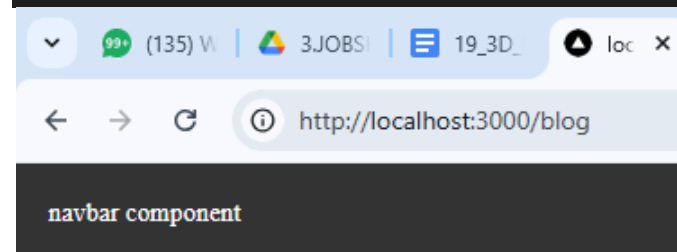
## Ini halaman edit profile

Halaman ini digunakan untuk mengedit profile pengguna.

## Tugas 2 – Dynamic Routing

Index.tsx

```
const Blog = () => {  
  return (  
    <div>  
      <h1>Blog Page</h1>  
      <p>Ini adalah halaman blog.</p>  
    </div>  
  );  
};  
  
export default Blog;
```



## Blog Page

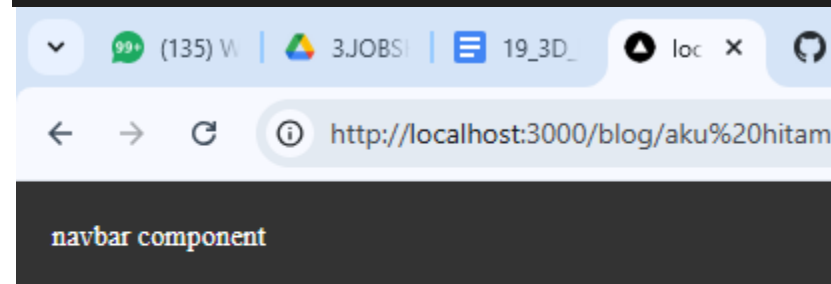
Ini adalah halaman blog.

[slug].tsx

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

export default function BlogSlugPage() {
  const router = useRouter();
  const { slug } = router.query;

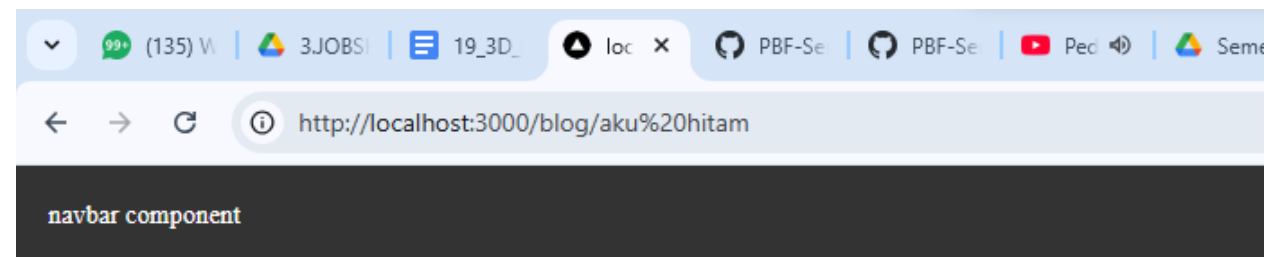
  return (
    <div>
      <h1>Detail Blog</h1>
      <p>Slug: {slug}</p>
    </div>
  );
}
```



## Detail Blog

Slug: aku hitam

### Tugas 3 – Layout



## Detail Blog

Slug: aku hitam



## **Pertanyaan Refleksi**

1. Apa perbedaan routing berbasis file dan routing manual?

Routing berbasis file otomatis dibuat dari struktur folder (seperti di Next.js), jadi lebih praktis. Routing manual harus ditulis konfigurasi path-nya secara kode, tapi lebih fleksibel.

2. Mengapa dynamic routing penting dalam aplikasi web?

Dynamic routing memungkinkan satu halaman menampilkan banyak data berbeda berdasarkan parameter URL (misalnya slug). Ini membuat aplikasi lebih efisien, hemat kode, dan cocok untuk data dari database.

3. Apa keuntungan menggunakan layout global dibanding memanggil komponen satu per satu?

Dynamic routing memungkinkan satu halaman menampilkan banyak data berbeda berdasarkan parameter URL (misalnya slug). Ini membuat aplikasi lebih efisien, hemat kode, dan cocok untuk data dari database.