

LAPORAN JOBSHEET PEMROGRAMAN BERBASIS FRAMEWORK



Nama : Yan Daffa Putra Liandhie

NIM: 2341720142

Kelas: TI 3E

**POLITEKNIK NEGERI MALANG
TAHUN 2025**

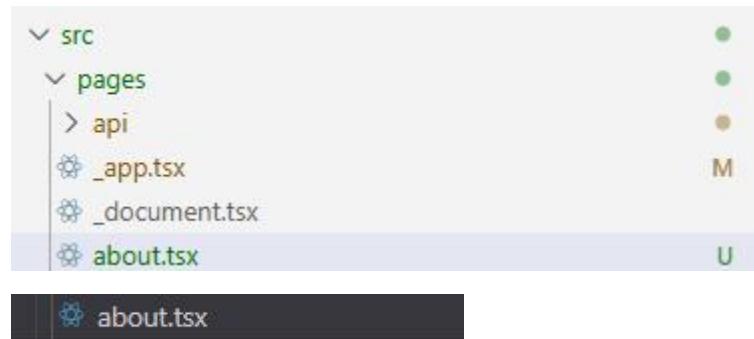
A. Langkah Praktikum

1. Routing Dasar (Static Routing)

a. Struktur Awal

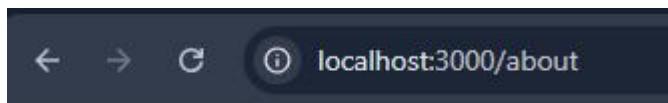
```
pages/  
└── index.tsx
```

b. Tambahkan Halaman About

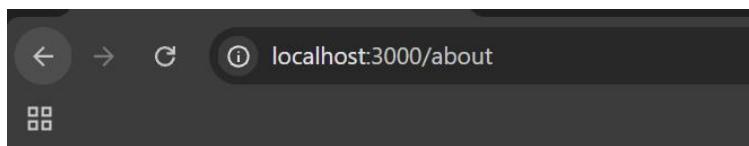


c. Uji di Browser

- <http://localhost:3000/about>



ini halaman about



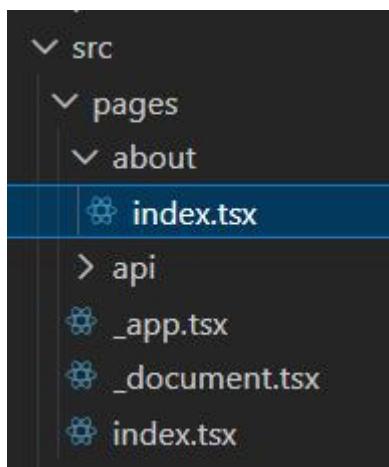
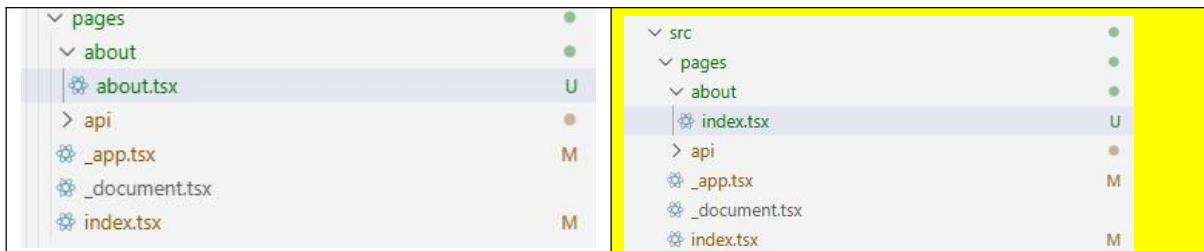
ini halaman about

Catatan:

Next.js otomatis membuat routing berdasarkan nama file tanpa konfigurasi tambahan.

2. Routing Menggunakan Folder

a. Rapikan Struktur Pages



Ubah struktur menjadi:

```
pages/
  └── about/
    └── index.tsx ( yang sebelumnya about.tsx menjadi index.tsx )
```

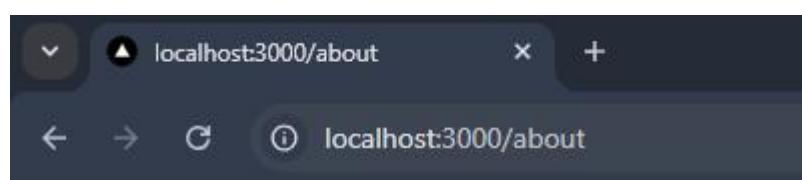
Akses:

/about

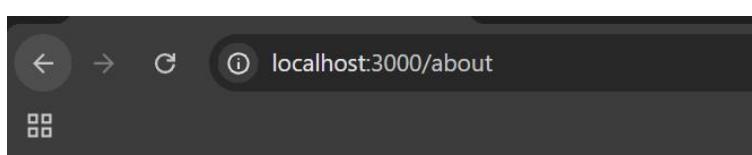
⚡ Insight:

index.tsx di dalam folder mewakili root folder tersebut.

b. Akses dari halaman browser (tetap sama tetapi lebih rapi)



ini halaman about



ini halaman about

3. Nested Routing

a. Buat Folder Setting

pages/

 └─ setting/

 ├─ user.tsx

 └─ app.tsx

 └─ src

 └─ pages

 > about

 > api

 └─ setting

 └─ app.tsx

 └─ user.tsx

 •

 •

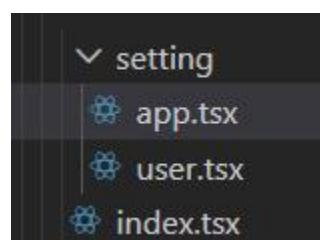
 •

 •

 •

 •

 •



Modifikasi kodennya

- **user.tsx**

```
praktikum-PagesLayout > my-app > src > pages > setting > user.tsx > ...
Windsurf: Refactor | Explain | Generate JSDoc | X
  1 const UserSettingPage = () => {
  2   return (
  3     <div>
  4       User Setting Page
  5     </div>
  6   );
  7 }
 8
 9 export default UserSettingPage;
pages > about > setting > user.tsx > UserSettingPage.tsx
const UserSetting = () => [
  return (
    <div>
      User Setting page
    </div>
  )
]
```

- **app.tsx**

```

praktikum-PagesLayout > my-app > src > pages > setting > app.tsx > ...
Windsurf: Refactor | Explain | Generate JSDoc | X
1 const Appsetting = () => {
2   return (
3     <div>
4       App Setting Page
5     </div>
6   );
7 }
8
9 export default Appsetting;

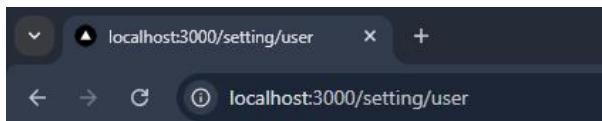
src > pages > about > setting > app.tsx > [U] UsserSetting
1 const UsserSetting = () => {
2   return (
3     <div>
4       User Setting page
5     </div>
6   )
7 }

```

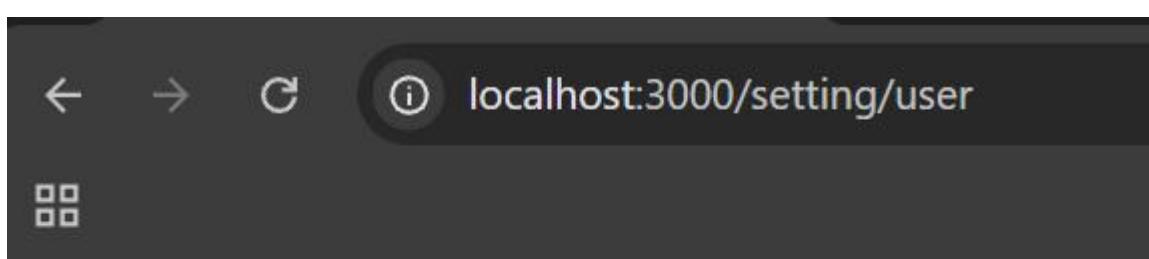
Tips: Copy paste dari user.tsx , block Usersettingpage + Control+D maka saat merubah usersettingpage menjadi Appsetting akan berubah semua tanpa harus rename satu-satu

Akses:

- /setting/user

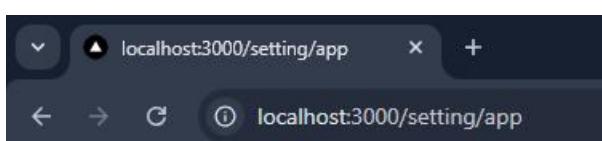


User Setting Page

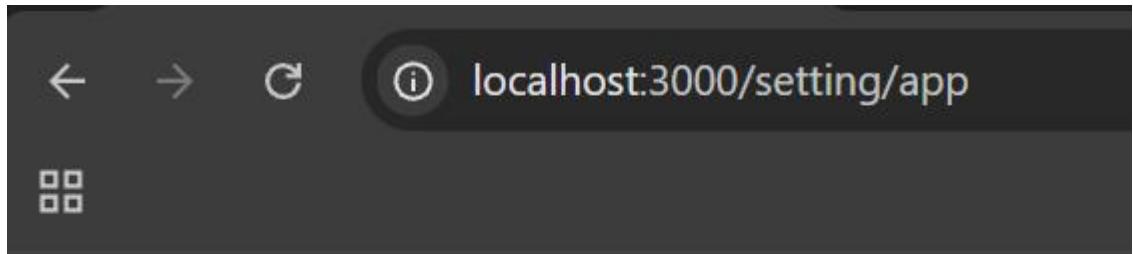


User Setting page

- /setting/app

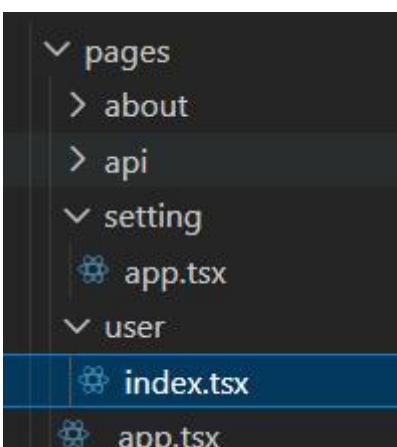


App Setting Page

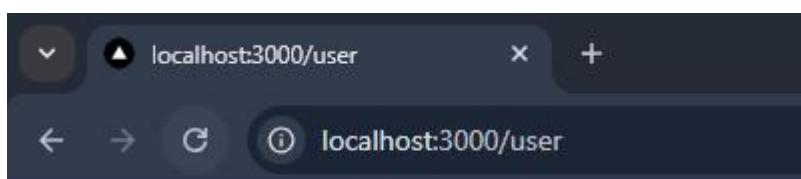


App Setting Page

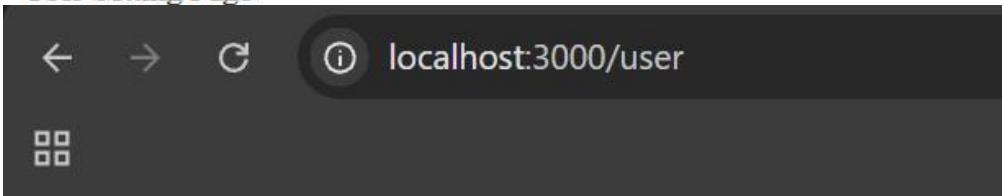
Modifikasi struktur folder pages dengan menambahkan folder user dan user.tsx pada setting dipindah ke folder user dan rubah file user.tsx menjadi index.tsx



Jalankan pada browser

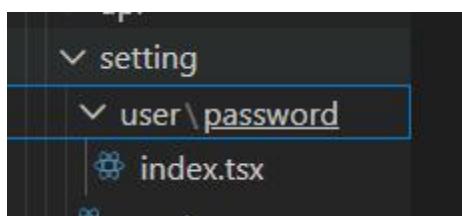
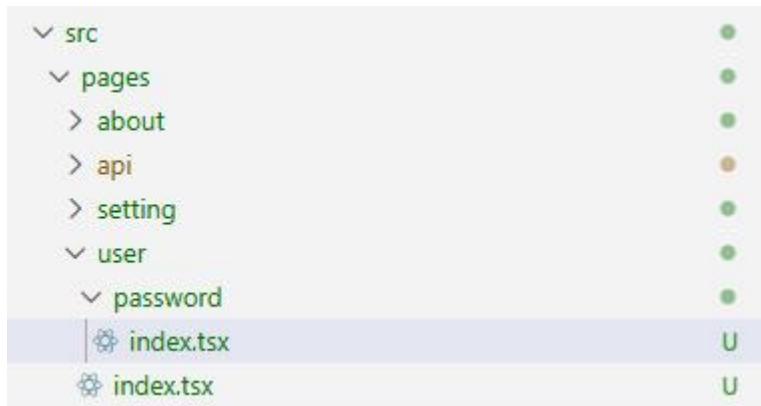


User Setting Page



User Setting page

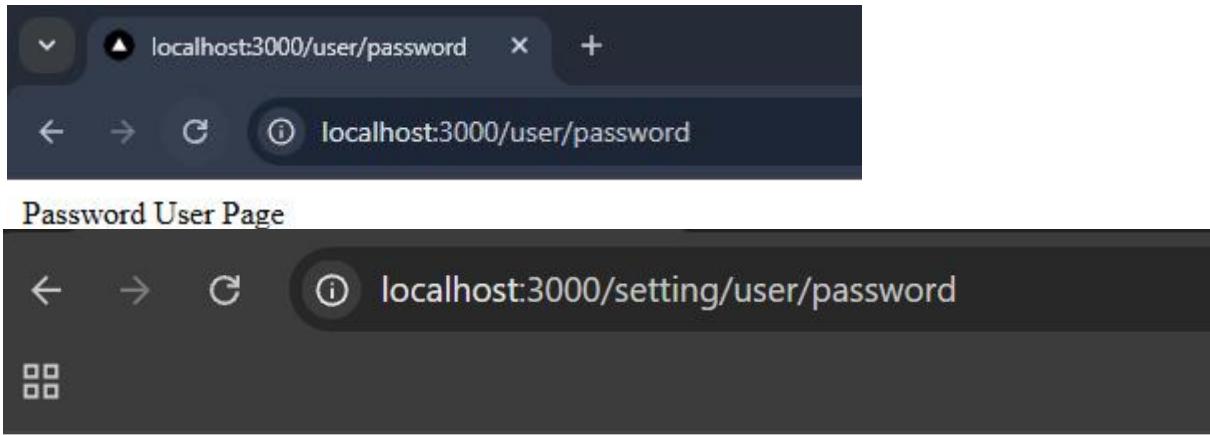
b. Nested Lebih Dalam



```
pages/
└── setting/
    └── user/
        └── password/
            └── index.tsx
```

Akses:

/setting/user/password



Password User page

Keunggulan:

Tidak perlu konfigurasi manual seperti React Router atau Laravel Route.

4. Dynamic Routing

a. Buat Halaman Produk

pages/

```
└── produk/  
    ├── index.tsx  
    └── [id].tsx
```



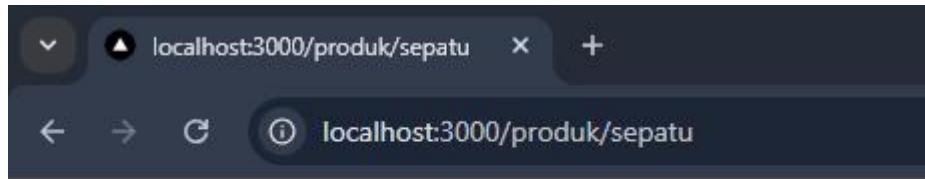
- Modifikasi index.tsx

```
praktikum-PagesLayout > my-app > src > pages > produk > index.tsx > ...
Windsurf: Refactor | Explain | Generate JSDoc | X
1 const produk = () => {
2   return (
3     <div>
4       Produk User Page
5     </div>
6   );
7 };
8
9 export default produk;
10
```

```
src > pages > produk > index.tsx > t > default
Windsurf: Refactor | Explain | Generate JSDoc | X
1 const produk=() => {
2   return(
3     <div>
4       Produk User page
5     </div>
6   );
7 };
8
9 export default produk;      Pin selection
```

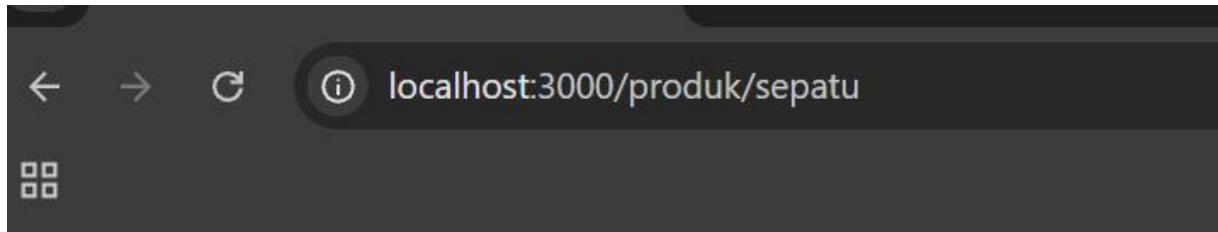
- **Modifikasi [id].tsx**

Buka browser <http://localhost:3000/produk/sepatu> tambahkan segment sepatu



Halaman Produk

Produk:



Halaman Produk

Produk

- Cek menggunakan console.log

```
praktikum-PagesLayout > my-app > src > pages > produk > [id].tsx > HalamanProduk
  1 import { useRouter } from "next/router";
  2
  3
  4 const HalamanProduk = () => {
  5   const Router = useRouter();
  6   console.log(Router); //x4 { pathname: '/produk/[id]', route: '/produk/[id]', query: { id: 'sepatu' }, asPath: '/produk/sepatu', components: { '/produk/...
  7   return (
  8     <div>
  9       <h1>Halaman Produk</h1>
10       <p>Produk:</p>
11     </div>
12   );
13 }
14
15 export default HalamanProduk;
16
17
import { useRouter } from "next/router";

Windsurf: Refactor | Explain | Generate JSDoc | X
const HalamanProduk=() => [
  const Router = useRouter();
  console.log(Router);] x5 { pathname: '/produk/[id]', route: '/produk/[id]', query: { id: 'sepatu' }, asPath: '/produk/sepatu', components: {
return(
  <div>
    <h1>Halaman Produk</h1>
    <p>Produk:</p>
  </div>
);
];
export default HalamanProduk;
```

jika berhasil maka pada console.log dapat terlihat pada id terdapat nilai sepatu.

⚡ Catatan Penting:

Pada console.log data langsung terlihat dikarenakan terdapat ext vscode console ninja

- Modifikasi [id].tsx agar dapat mengambil nilai dari id

```
praktikum-PagesLayout > my-app > src > pages > produk > [id].tsx > ...
1 import { useRouter } from "next/router";
2
3
4 const HalamanProduk = () => {
5     // const Router = useRouter();
6     // console.log(Router);
7     const { query } = useRouter();
8     return (
9         <div>
10            <h1>Halaman Produk</h1>
11            <p>Produk: {query.id}</p>
12        </div>
13    );
14}
15
16 export default HalamanProduk;
17
```

src > pages > produk > [id].tsx > HalamanProduk

```
1 D:\tugas\semester 6\framework\js 2\mastwer\praktikum\my-app\src\pages\produk\[id].tsx
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const HalamanProduk=() => {
5     // const Router = useRouter();
6     // console.log(Router);
7     const {query} = useRouter();
8     return(
9         <div>
10            <h1>Halaman Produk</h1>
11            <p>Produk: {query.id}</p>
12        </div>
13    );
14
15 export default HalamanProduk;
```

- Buka browser



Halaman Produk

Produk: sepatu



Halaman Produk

Produk: sepatu

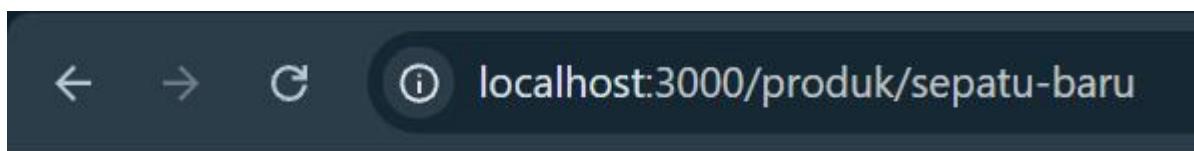
c. Uji di Browser

- /produk/sepatu-baru



Halaman Produk

Produk: sepatu-baru



Halaman Produk

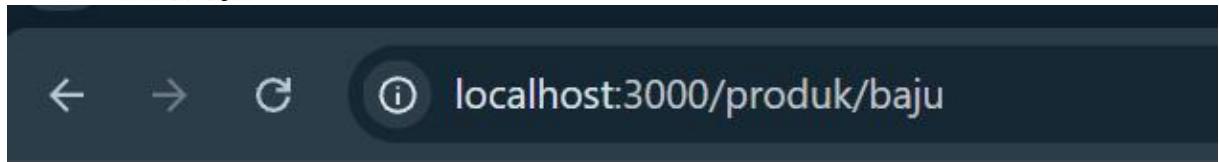
Produk: sepatu-baru

- /produk/baju



Halaman Produk

Produk: baju



Halaman Produk

Produk: baju

⚡ Catatan Penting:

Nama file di dalam [] akan menjadi **parameter URL**. Contoh

```
EXPLORER          ...          [no].tsx U X
praktikum-PagesLayout > my-app > src > pages > produk > [no].tsx > HalamanProduk
1 import { useRouter } from "next/router";
2
3
4 const HalamanProduk = () => {
5   // const Router = useRouter();
6   // console.log(Router);
7   const { query } = useRouter();
8   return (
9     <div>
10       <h1>Halaman Produk</h1>
11       <p>Produk: {query.no}</p>
12     </div>
13   );
14 }
15
16 export default HalamanProduk;
```

```
rc > pages > produk > [no].tsx > default
1 import { useRouter } from "next/router";
2
3 const HalamanProduk = () => {
4     const {query} = useRouter();
5     return(
6         <div>
7             <h1>Halaman Produk</h1>
8             <p>Produk: {query.no}</p>
9         </div>
10    );
11 }
12 export default HalamanProduk;
```

5. Membuat Komponen Navbar

a. Struktur Komponen

```
src/
└── components/
    └── layouts/
        └── Navbar/
            └── index.tsx
```



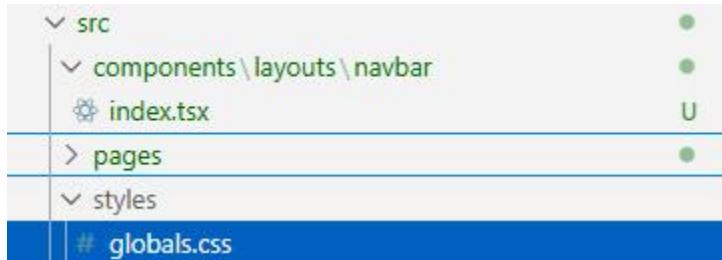
```
└── components\layouts\navbar
    └── index.tsx
```

- Modifikasi index.tsx

```
index.tsx U X
praktikum-PagesLayout > my-app > src > components > layouts > navbar > index.tsx > ...
Windsurf: Refactor | Explain | Generate JSDoc | X
1 const Navbar = () => {
2     return (
3         <div className="">
4             <div>navbar Component </div>
5         </div>
6     );
7 };
8
9 export default Navbar;
10
```

```
src / pages / components / layouts / navbar / index.tsx
      Windsurf: Refactor | Explain | Generate JSDoc | X
1  ↴ const Navbar = () => {
2    ↴   return(
3      ↴     <div className="">
4       |       <div>Navbar Component</div>
5       |     </div>
6    );
7  };
8  export default Navbar;
```

- Buka globals.css untuk nantinya digunakan pada style navbar



- globals.css

```
praktikum-PagesLayout > my-app > src > styles > # globals.css > :root
  1 > :root { ...
  41   }
  42
  43 > @media (prefers-color-scheme: dark) { ...
  74   }
  75
  76 > * { ...
  80   }
  81
  82   html,
  83 > body { ...
  86   }
  87
  88 > body { ...
  96   }
  97
  98 > a { ...
 101   }
 102
 103 > @media (prefers-color-scheme: dark) { ...
 107   }
 108
```

- Modifikasi global.css

```
praktikum-PagesLayout > my-app > src > styles > # globals.css > ...
  1
  2
  3   * {
  4     box-sizing: border-box;
  5     padding: 0;
  6     margin: 0;
  7   }
  8
  9   html,
 10  body {
 11    max-width: 100vw;
 12    overflow-x: hidden;
 13  }
 14
 15
 16
 17  a {
 18    color: inherit;
 19    text-decoration: none;
 20  }
 21
```

```
styles > # globals.css > * a
*{
  box-sizing: border-box;
  padding: 0;
  margin: 0;
}

html, body {
  max-width: 100vw;
  overflow-x: hidden;
}

a[
  color: inherit;
  text-decoration: none;
]
```

- Modifikasi index.tsx dengan menambahkan classname untuk style navbar

```
praktikum-PagesLayout > my-app > src > components > layouts > navbar > index.tsx > ...
Windsurf: Refactor | Explain | Generate JSDoc | X
1 const Navbar = () => {
2   return (
3     <div className="navbar">
4       <div>navbar Component </div>
5     </div>
6   );
7 };
8
9 export default Navbar;
10

src > pages > components > layouts > navbar > index.tsx >
Windsurf: Refactor | Explain | Generate JSDoc | X
1 const Navbar = () => {
2   return(
3     <div className="navbar">
4       <div>Navbar Component</div>
5     </div>
6   );
7 };
8 export default Navbar;
```

- Modifikasi globals.css

```
praktikum-PagesLayout > my-app > src > styles > # globals.css > .navbar
1
2
3 > * { ...
7 }
8
9 html,
10 > body { ...
13 }
14
15
16
17 > a { ...
20 }
21
22
23 .navbar {
24   width: 100%;
25   height: 60px;
26   background-color: #333;
27   color: white;
28   display: flex;
29   align-items: center;
30   padding: 0 20px;
31 }
```

```
src > styles > # globals.css > ⚙ navbar
1 > *{ ...
5   }
6
7 > html, body { ...
10  }
11
12 > a{ ...
15  }
16
17 navbar{
18   width: 100%;
19   height: 60px;
20   background-color: □#333;
21   color: ■white;
22   display: flex;
23   align-items: center;
24   padding: 0 20px;|
25 }
```

- Modifikasi index.tsx pada folder pages

```
praktikum-PagesLayout > my-app > src > pages > ❁ index.tsx > ...
1 import Head from 'next/head'
2 import Image from 'next/image'
3 import { Inter } from 'next/font/google'
4 import styles from '@/styles/Home.module.css'
5 import Navbar from '@/components/layouts/navbar'
6
7 const inter = Inter({ subsets: ['latin'] })
8
9 Windsurf: Refactor | Explain | Generate JSDoc | X
10 export default function Home() {
11   return (
12     <div>
13       <Navbar />
14       <h1>Praktikum Next.js Pages Router</h1> <br />
15       <p>Mahasiswa D4 Pengembangan Web</p>
16     </div>
17   )
18 }
19
```

```

src > pages > index.tsx > ...
1 import Head from 'next/head'
2 import Image from 'next/image'
3 import { Inter } from 'next/font/google'
4 import styles from '@/styles/Home.module.css'
5 import Link from 'next/link'
6 import Navbar from './components/layouts/navbar'
7
8 const inter = Inter({ subsets: ['latin'] })
9

Windsurf: Refactor | Explain | Generate JSDoc | X
10 export default function Home() {
11   return (
12     <div>
13       <Navbar />
14       <h1>Praktikum Next.js Pages Router</h1> <br />
15       <p> Mahasiswa D4 Pengembangan Web</p>
16
17       <Link href="/about">
18         <About>
19       </Link>
20
21     </div>
22   )
23 }
24

```

- Modifikasi _app.tsx (pastikan import styles dalam keadaan aktif)

```

praktikum-PagesLayout > my-app > src > pages > _app.tsx > ...
1 import '@/styles/globals.css'
2 import type { AppProps } from 'next/app'
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 export default function App({ Component, pageProps }: AppProps) {
6   return <Component {...pageProps} />
7 }

```

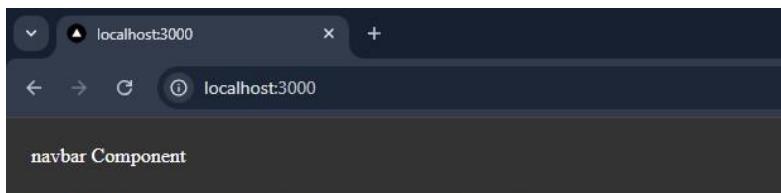
```

pages > _app.tsx > ...
import '@/styles/globals.css'
import type { AppProps } from 'next/app'

Windsurf: Refactor | Explain | Generate JSDoc | X
export default function App({ Component, pageProps }: AppProps) {
  return <Component {...pageProps} />
}

```

- Jalankan di browser (Navbar akan tampil)



Praktikum Next.js Pages Router

Mahasiswa D4 Pengembangan Web

Catatan Penting:

navbar hanya akan muncul pada index page, pada page produk navbar tidak akan muncul. Contoh

The screenshot displays two browser windows side-by-side. The left window shows the URL `localhost:3000` with the title "Praktikum Next.js Pages Router". It features a dark header bar with the text "navbar Component" and a main content area with the text "Mahasiswa D4 Pengembangan Web". The right window shows the URL `localhost:3000/produk/sepatu` with the title "Halaman Produk". It has a dark header bar with the text "Produk: sepatu" and a main content area with the same text.

This screenshot shows the product page from the previous image, but with the navbar component removed from the code. The page now displays the text "Produk: sepatu" directly under the title "Halaman Produk".

- Modifikasi navbar agar tampil di semua page

- Modifikasi index.tsx pada folder page (hapus navbar)

```
praktikum-PagesLayout > my-app > src > pages > index.tsx > ...
  1 import Head from 'next/head'
  2 import Image from 'next/image'
  3 import { Inter } from 'next/font/google'
  4 import styles from '@/styles/Home.module.css'
  5
  6
  7 const inter = Inter({ subsets: ['latin'] })
  8
  9 Windsurf: Refactor | Explain | Generate JSDoc | X
10 export default function Home() {
11   return (
12     <div>
13       <h1>Praktikum Next.js Pages Router</h1> <br />
14       <p>Mahasiswa D4 Pengembangan Web</p>
15     </div>
16   )
17 }
18 }
```

```

rc > pages > index.tsx > Home
  1 import Head from 'next/head'
  2 import Image from 'next/image'
  3 import { Inter } from 'next/font/google'
  4 import styles from '@/styles/Home.module.css'
  5 import Link from 'next/link'
  6 // import Navbar from './components/layouts/navbar'
  7
  8 const inter = Inter({ subsets: ['latin'] })
  9

Windsurf: Refactor | Explain | Generate JSDoc | X
10 export default function Home() {
11   return [
12     <div>
13       {/* <Navbar /> */}
14       <h1>Praktikum Next.js Pages Router</h1> <br />
15       <p> Mahasiswa D4 Pengembangan Web</p>
16
17       <Link href="/about">
18         About
19       </Link>
20
21     </div>
22   ]
23 }
24

```

- Modifikasi _app.tsx (Menambahkan navbar)

```

praktikum-PagesLayout > my-app > src > pages > _app.tsx > ...
  1 import '@/styles/globals.css'
  2 import type { AppProps } from 'next/app'
  3 import Navbar from '@/components/layouts/navbar'
  4

Windsurf: Refactor | Explain | Generate JSDoc | X
5 export default function App({ Component, pageProps }: AppProps) {
6   return (
7     <div>
8       <Navbar />
9       <Component {...pageProps} />
10      </div>
11    );
12  };
13

```

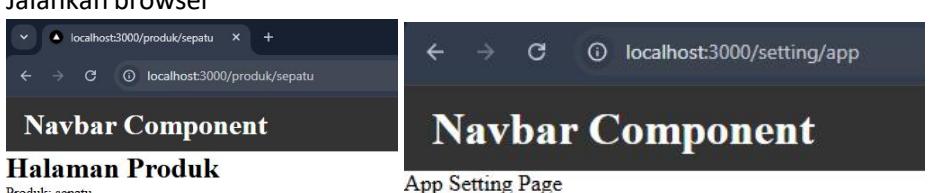
```

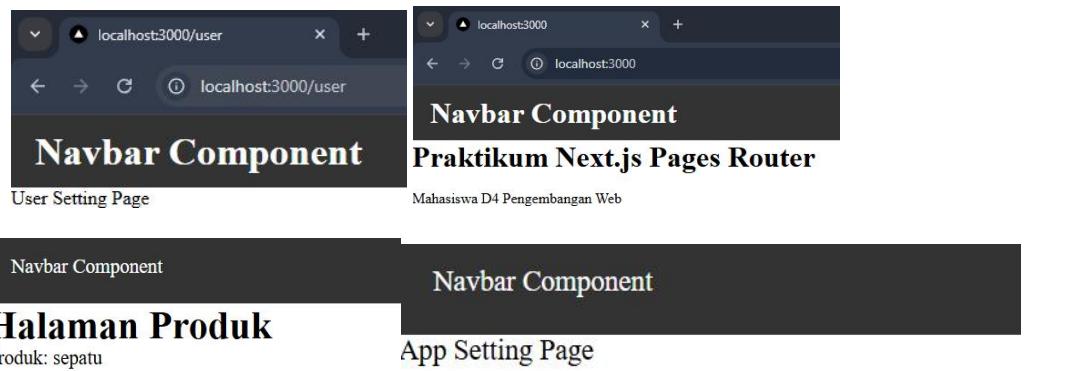
src > pages > _app.tsx > ...
  1 import '@/styles/globals.css'
  2 import type { AppProps } from 'next/app'
  3 import Navbar from './components/layouts/navbar'
  4

Windsurf: Refactor | Explain | Generate JSDoc | X
5 export default function App({ Component, pageProps }: AppProps) {
6   return (
7     <div>
8       <Navbar />
9       <Component {...pageProps} />
10      </div>
11    );
12  };
13

```

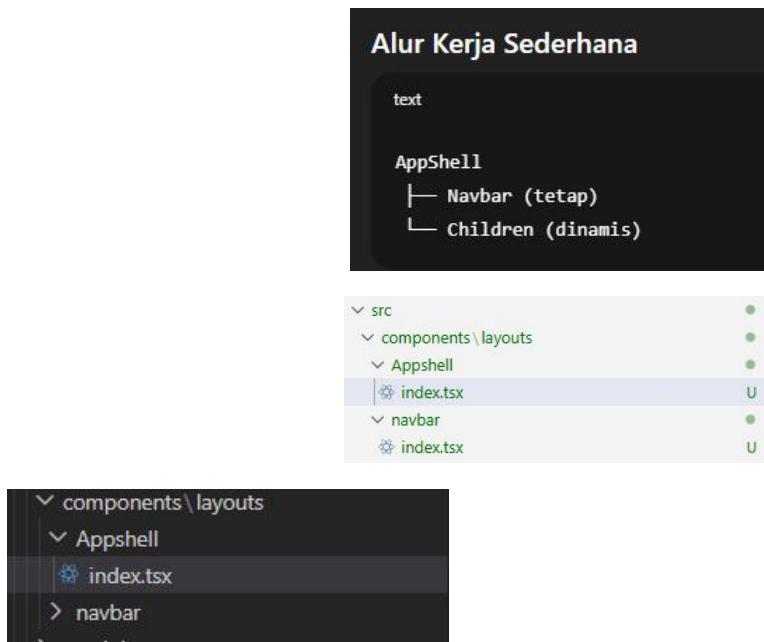
- Jalankan browser





6. Membuat Layout Global (App Shell)

a. Buat AppShell



- Modifikasi index.tsx pad AppShell

```
praktikum-PagesLayout > my-app > src > components > layouts > Appshell > index.tsx > [AppShell]
  1 import Navbar from "../navbar";
  2
  3 Windsurf: Refactor | Explain
  4 type AppShellProps = {
  5   children: React.ReactNode;
  6 }
  7 Windsurf: Refactor | Explain | Generate JSDoc | X
  8 const AppShell = (props:AppShellProps) => {
  9   const { children } = props;
 10   return (
 11     <main>
 12       <Navbar />
 13       {children}
 14     </main>
 15   );
 16 };
 17
 18 export default AppShell;
 19
src > pages > components > layouts > Appshell > index.tsx > [AppShell]
  1 import Navbar from "../navbar";
  2
  3 Windsurf: Refactor | Explain
  4 type AppShellProps = {
  5   children: React.ReactNode;
  6 }
  7 Windsurf: Refactor | Explain | Generate JSDoc | X
  8 const AppShell = (props:AppShellProps) => {
  9   const { children } = props;
 10   return(
 11     <main>
 12       <Navbar />
 13       {children}
 14     </main>
 15   );
 16 };
 17
 18 export default AppShell;
```

7. Implementasi Layout di _app.tsx

```
praktikum-PagesLayout > my-app > src > pages > _app.tsx > [App]
  1 import '@/styles/globals.css'
  2 import type { AppProps } from 'next/app'
  3 import AppShell from '@/components/layouts/Appshell'
  4 import Navbar from '@/components/layouts/navbar'
  5
  6 Windsurf: Refactor | Explain | Generate JSDoc | X
  7 export default function App({ Component, pageProps }: AppProps) {
  8   return (
  9     <AppShell>
 10       <Component {...pageProps} />
 11     </AppShell>
 12   );
 13
 14
 15};
```

```
1 import '@/styles/globals.css'
2 import type { AppProps } from 'next/app'
3 import Navbar from './components/layouts/navbar'
4 import AppShell from './components/layouts/Appshell';
5
6 Windsurf: Refactor | Explain | Generate JSDoc | X
7 export default function App({ Component, pageProps }: AppProps) {
8   return [
9     // <div>
10    <AppShell>
11      <Component {...pageProps} />
12    </AppShell>
13  // </div>
14 ];
15 }
```

✓ Hasil:

Navbar dan layout muncul di **semua halaman** tanpa perlu dipanggil satu per satu contoh

Modifikasi pada `_app.tsx` tambahkan footer seperti pada gambar dan amati hasilnya

```
27 */
28 const { children } = props;
29 return (
30   <main>
31     <Navbar />
32     {children}
33   <div>
34     footer
35   </div>
36 </main>
```

```
src > pages > components > layouts > Appshell > index.tsx > [edit]
1 import Navbar from "../navbar";
2
3 Windsurf: Refactor | Explain
4 type AppShellProps = {
5   children: React.ReactNode;
6 };
7 Windsurf: Refactor | Explain | Generate JSDoc | X
8 const AppShell = (props: AppShellProps) => {
9   const { children } = props;
10  return [
11    <main>
12      <Navbar />
13      {children}
14    <div>Footer</div>
15  </main>
16];
17};
18
19 export default AppShell;
20
```

localhost:3000/about

Navbar Component

ini halaman about

footer

localhost:3000

Navbar Component

Praktikum Next.js Pages Router

Mahasiswa D4 Pengembangan Web

footer

Navbar Component

Praktikum Next.js Pages Router

Mahasiswa D4 Pengembangan Web

About

footer

B. Tugas Praktikum

Tugas 1 – Routing

1. Buat halaman:

- o /profile

```
src > pages > profile > index.tsx > profile
1 import Link from 'next/link'
Windsurf: Refactor | Explain | Generate JSDoc | X
2 export default function profile() {
3   return (
4     <div>
5       <h1>Halaman Profil</h1> <br />
6       <Link href="/profile/edit">
7         <p>edit profil</p>
8       </Link>
9
10    </div>
11  )
12}
13
```

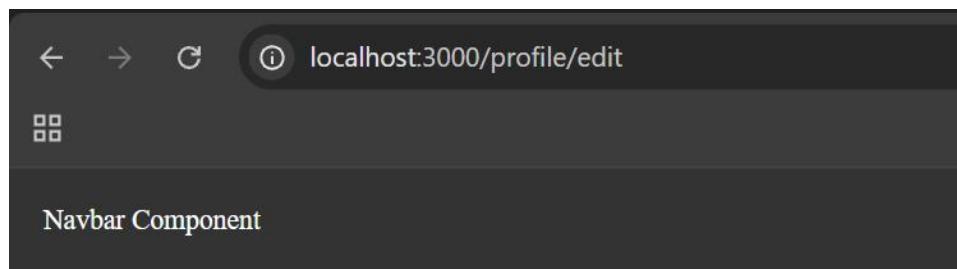
Navbar Component

Halaman Profil

edit profil
footer

- o /profile/edit

```
src > pages > profile > edit > index.tsx > profile
Windsurf: Refactor | Explain | Generate JSDoc | X
1 export default function profile() {
2   return (
3     <div>
4       <h1>Halaman edit Profil</h1> <br />
5     </div>
6   )
7 }
8
```



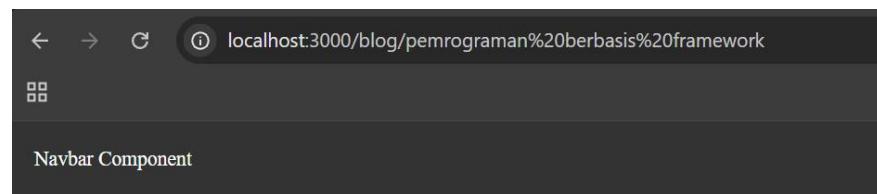
2. Pastikan routing berjalan tanpa error

Tugas 2 – Dynamic Routing

1. Buat routing:
2. /blog/[slug]

```
src/pages/blog/[slug].tsx
1 import { useRouter } from 'next/router';
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 export default function BlogDetail() {
5   const router = useRouter();
6   const { slug } = router.query;
7
8   return (
9     <div>
10    <h1>Halaman Blog</h1>
11    <p>Slug: {slug}</p>
12  );
13}
```

3. Tampilkan nilai slug di halaman



Tugas 3 – Layout

1. Tambahkan Footer pada AppShell

```
src > pages > components > layouts > Appshell > index.tsx > AppShell
  1 import Navbar from "../navbar";
  2
  3 type AppShellProps = {
  4   children: React.ReactNode;
  5 };
  6
  7
  8 const AppShell = (props: AppShellProps) => {
  9   const { children } = props;
 10   return [
 11     <main>
 12       <Navbar />
 13       {children}
 14
 15       <div className="footer">
 16         footer
 17       </div>
 18     </main>
 19   ];
 20 };
 21
 22 export default AppShell;
 23
```

2. Footer tampil di semua halaman



C. Pertanyaan Refleksi

1. Apa perbedaan routing berbasis file dan routing manual?

Jawab : perbedaan antara routing berbasis file dan routing manual terletak di konfigurasinya dimana routing berbasis file tidak memerlukan konfigurasi secara manual dan bisa langsung dipanggil, sementara routing manual perlu adanya konfigurasi secara manual terlebih dahulu sebelum bisa dipanggil

2. Mengapa dynamic routing penting dalam aplikasi web?

Jawab : dynamic routing penting dikarenakan memungkinkan aplikasi menampilkan konten berbeda menggunakan satu struktur halaman yang sama berdasarkan parameter tertentu

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

Jawab : dapat mengurangi duplikasi kode, mempermudah maintenance, menjaga konsistensi tampilan, dan membuat struktur lebih terorganisir

D. 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`