

## **JOBSCHEET PRAKTIKUM**

**Routing, Nested Routing, Dynamic Routing, dan Layouting pada Next.js (Pages Router)**

### **Mata Kuliah**

Pemrograman Framework

### **Topik Praktikum**

Routing & Layouting pada Next.js (Pages Router)

### **Waktu Praktikum**

2 × 50 menit

---

### **A. Tujuan Praktikum**

Setelah menyelesaikan praktikum ini, mahasiswa mampu:

1. Memahami konsep **Pages Router** pada Next.js
  2. Membuat **routing statis** berbasis file dan folder
  3. Mengimplementasikan **nested routing**
  4. Mengimplementasikan **dynamic routing** menggunakan parameter URL
  5. Membuat **layout global** menggunakan komponen layout (App Shell)
- 

### **B. Tools & Persiapan**

- Node.js (minimal v16)
- NPM / Yarn / PNPM
- Code Editor (VS Code disarankan)
- Browser (Chrome / Firefox)
- Project Next.js (TypeScript)

`npx create-next-app@latest next-routing`

`cd next-routing`

`npm run dev`

---

### **C. Dasar Konsep (Ringkas)**

- Folder **pages/** → otomatis menjadi routing
- `index.tsx` → root route (/)
- Folder di dalam **pages/** → nested route

- File [param].tsx → dynamic routing
  - pages/\_app.tsx → entry point global aplikasi
- 

## D. Langkah Praktikum

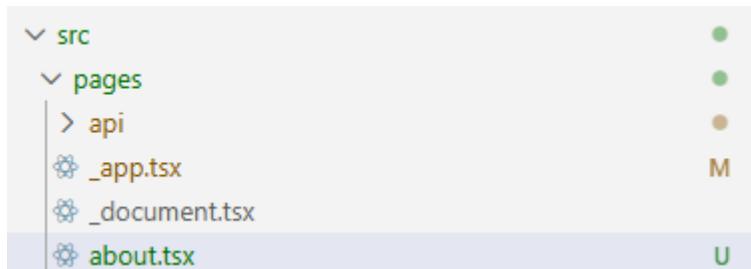
---

### 1. Routing Dasar (Static Routing)

#### a. Struktur Awal

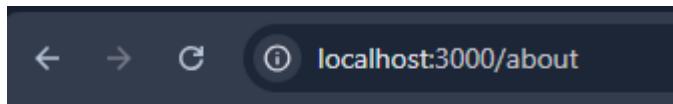
```
pages/  
└── index.tsx
```

#### b. Tambahkan Halaman About



#### c. Uji di Browser

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



# ini halaman about

#### ⭐ Catatan:

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

---

### 2. Routing Menggunakan Folder

#### a. Rapikan Struktur Pages

<ul style="list-style-type: none"> <li>pages</li> <li>  about</li> <li>    about.tsx</li> <li>  api</li> <li>  _app.tsx</li> <li>  _document.tsx</li> <li>  index.tsx</li> </ul>	<ul style="list-style-type: none"> <li>src</li> <li>  pages</li> <li>    about</li> <li>      index.tsx</li> <li>    api</li> <li>    _app.tsx</li> <li>    _document.tsx</li> <li>    index.tsx</li> </ul>
--	---

Ubah struktur menjadi:

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

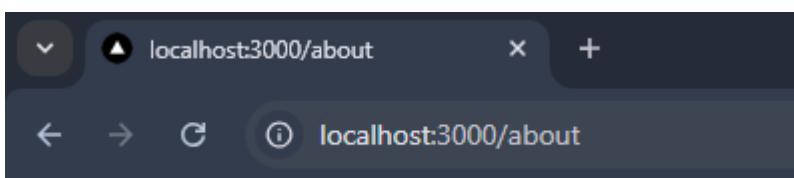
Akses:

```
/about
```



**Insight:** index.tsx di dalam folder mewakili root folder tersebut.

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



## ini halaman about

---

### 3. Nested Routing

#### a. Buat Folder Setting

```
pages/
  └── setting/
    └── user.tsx
  └── app.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;
```

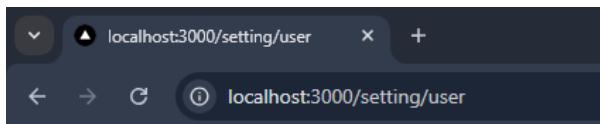
- **app.tsx**

```
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;
```

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

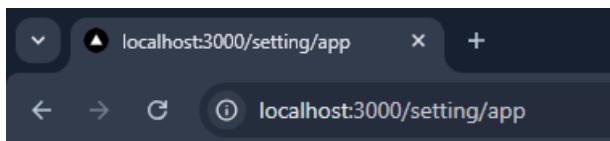
Akses:

- **/setting/user**



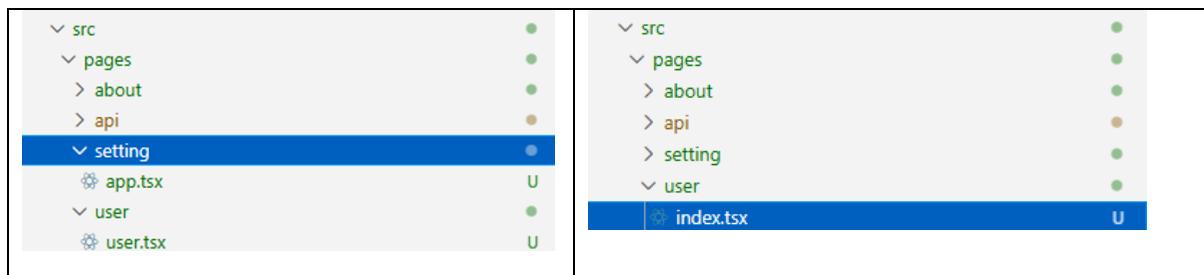
User Setting Page

- **/setting/app**

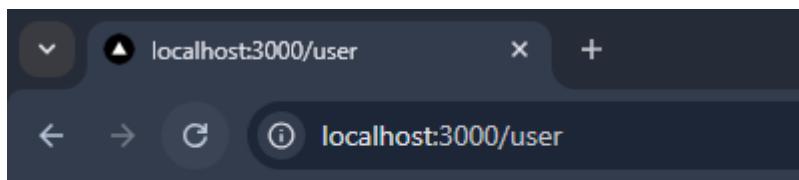


App Setting Page

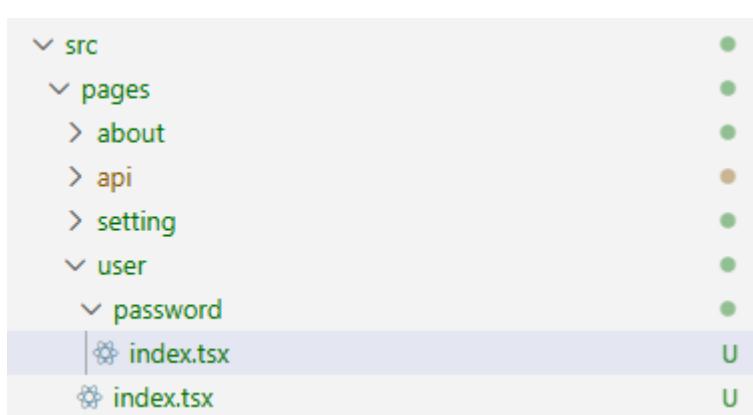
Modifikasi struktur folder pages dengan menambahkan folder user dan user.tsx pada setting dipindah ke folder user dan rubah file user.tsx menjadi index.tsx



Jalankan pada browser



#### b. Nested Lebih Dalam

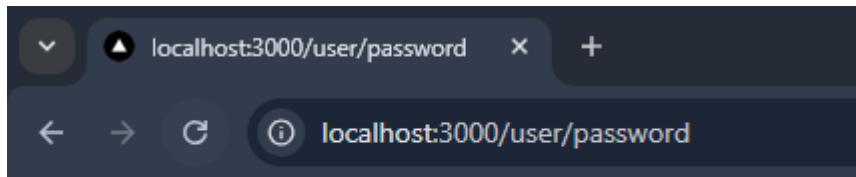


pages/

```
└── 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
```

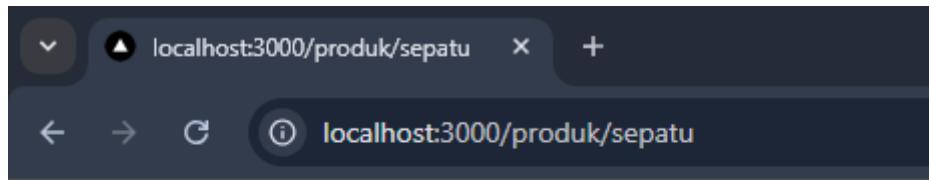


- Modifikasi index.tsx

```
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
```

- Modifikasi [id].tsx

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



# 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 Windsurf: Refactor | Explain | Generate JSDoc | X
  5 const HalamanProduk = () => {
  6   const Router = useRouter();
  7   console.log(Router); //<-- pathname: '/produk/[id]', route: '/produk/[id]', query: { id: 'sepatu' }, asPath: '/produk/sepatu', components: { '/produk/[...'
  8   return (
  9     <div>
 10       <h1>Halaman Produk</h1>
 11       <p>Produk:</p>
 12     </div>
 13   );
 14
 15 export default HalamanProduk;
 16
 17
```

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

## 📌 Catatan Penting:

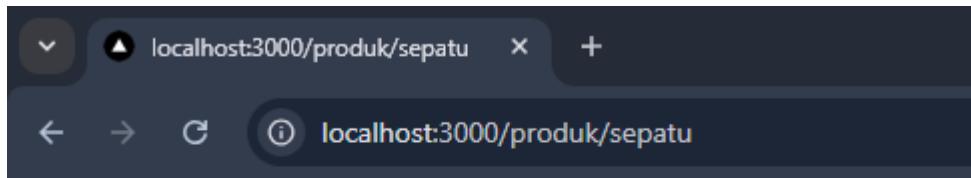
Pada console.log data langsung terlihat dikarenakan terdapat ext vscode console ninja

- 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 Windsurf: Refactor | Explain | Generate JSDoc | X
  5 const HalamanProduk = () => {
  6   // const Router = useRouter();
  7   // console.log(Router);
  8   const { query } = useRouter();
  9   return (
 10     <div>
 11       <h1>Halaman Produk</h1>
 12       <p>Produk: {query.id}</p>
 13     </div>
 14   );
 15
 16 export default HalamanProduk;
 17
```

- Buka browser

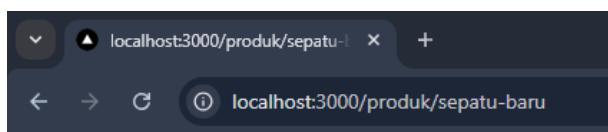


## Halaman Produk

Produk: sepatu

### c. Uji di Browser

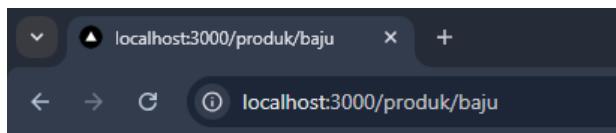
- /produk/sepatu-baru



## Halaman Produk

Produk: sepatu-baru

- /produk/baju



## Halaman Produk

Produk: baju

### 📌 Catatan Penting:

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

The screenshot shows the VS Code interface with the Explorer sidebar open, displaying the project structure. The code editor shows a file named "HalamanProduk.tsx" with the following content:

```

import { useRouter } from "next/router";
const HalamanProduk = () => {
  // const Router = useRouter();
  // console.log(Router);
  const { query } = useRouter();
  return (
    <div>
      <h1>Halaman Produk</h1>
      <p>Produk: {query.no}</p>
    </div>
  );
}
export default HalamanProduk;

```

A red arrow points from the "produk" folder in the Explorer sidebar to the "no" parameter in the code editor.

## 5. Membuat Komponen Navbar

### a. Struktur Komponen

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

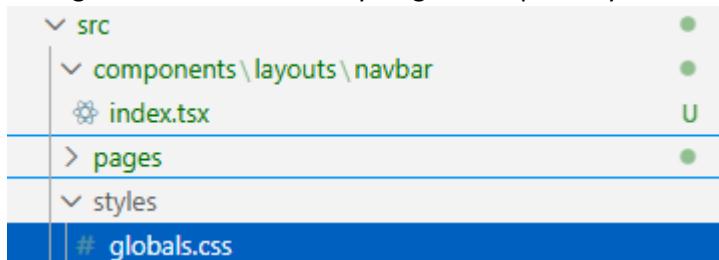


- Modifikasi index.tsx

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
```

- 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
```

- 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
```

- 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 }
```

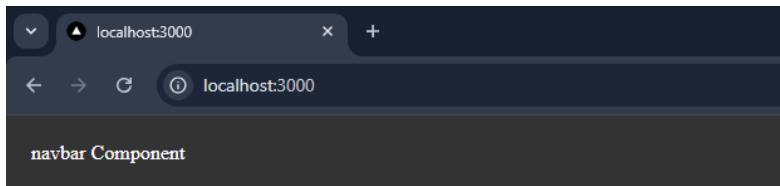
- Modifikasi index.tsx pada folder pages

```
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
```

- 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) {
5   return <Component {...pageProps} />
6 }
7
```

- 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 shows two browser windows side-by-side. The left window is at the root URL 'localhost:3000' and displays the 'Praktikum Next.js Pages Router' page with its header and content. The right window is at the URL 'localhost:3000/produk/sepatu' and displays the 'Halaman Produk' page with the text 'Produk: sepatu', illustrating that the dynamic route does not include the shared header/navbar component.

- 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 }
```

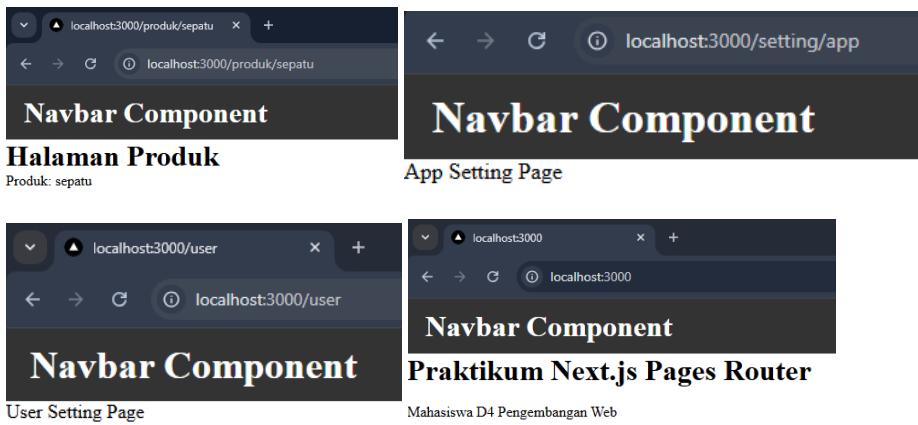
- Modifikasi \_app.tsx ( Menambahkan navbar )

```

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

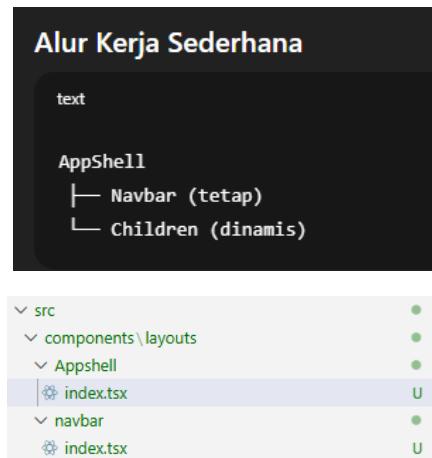
```

- 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

```

## 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

```

### ⭐ 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>

```



---

## E. Tugas Praktikum

### Tugas 1 – Routing

1. Buat halaman:
  - /profile
  - /profile/edit
2. Pastikan routing berjalan tanpa error

### Tugas 2 – Dynamic Routing

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

### Tugas 3 – Layout

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

## F. Pertanyaan Refleksi

1. Apa perbedaan routing berbasis file dan routing manual?
  2. Mengapa dynamic routing penting dalam aplikasi web?
  3. Apa keuntungan menggunakan layout global dibanding memanggil komponen satu per satu?
- 

## G. 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