

LAPORAN JOBSHEET PEMROGRAMAN BERBASIS FRAMEWORK



Nama : Yan Daffa Putra Liandhie

NIM: 2341720142

Kelas: TI 3E

**POLITEKNIK NEGERI MALANG
TAHUN 2026**

JOBSCHEET PRAKTIKUM

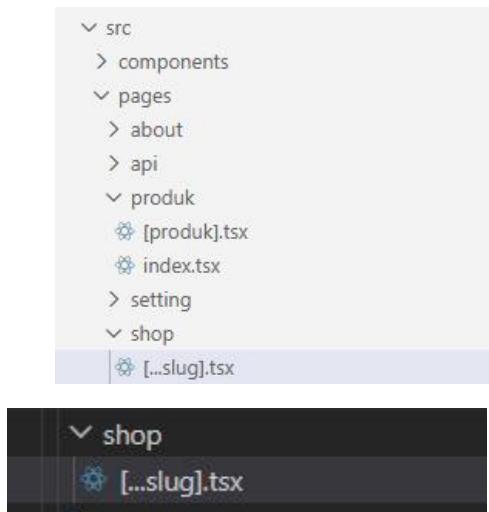
A. Langkah Kerja Praktikum

Langkah 1 – Menjalankan Project

1. Buka folder project Next.js.
 2. Jalankan server:
 3. npm run dev
 4. Akses:
 5. http://localhost:3000
-

Langkah 2 – Membuat Catch-All Route

1. Masuk ke folder pages.
2. Buat folder shop dan file [...slug].tsx:



3. Modifikasi Isi file [...slug].tsx dengan kode berikut:

```
praktikum-LinkNavigation > my-app > src > pages > shop > ⚙️ [...slug].tsx > .
  1 import { useRouter } from "next/router";
  2
  3
  4 Windsurf: Refactor | Explain | Generate JSDoc | X
  5 const halamanToko = () => {
  6   const Router = useRouter();
  7   console.log(Router);
  8   // const { query } = useRouter();
  9   return (
 10     <div>
 11       <h1>Halaman Toko</h1>
 12     </div>
 13   );
 14 };
 15
 16 export default halamanToko;
 17
```

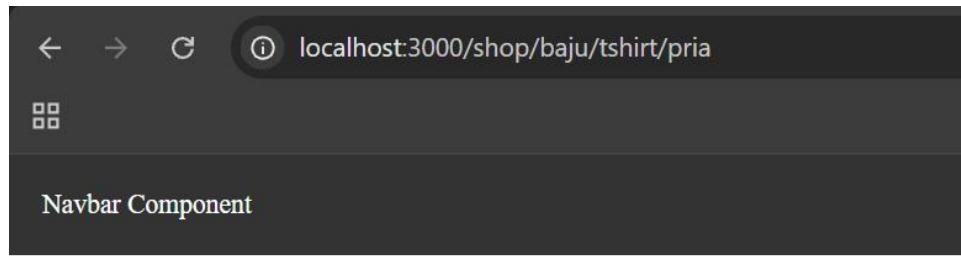
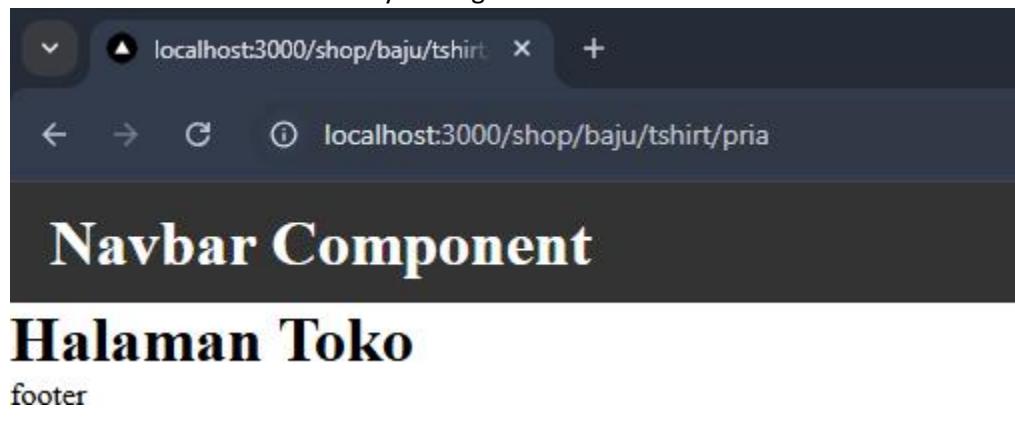
```

src > pages > shop > [...slug].tsx > [d] default
1 import { useRouter } from "next/router";
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const halamanToko = ()=> {
5   const router = useRouter();
6   const { slug } = router.query;
7   // const {query} = useRouter();
8   return (
9     <div>
10       <h1>Halaman Toko</h1>
11     </div>
12   );
13 export default halamanToko;

```

Cek menggunakan console.log apakah nilai segment berhasil didapat

- Jalankan browser dan ketik urlnya sebagai berikut



Halaman Toko

footer

- Cek Vscode jika pada console.log dapat menampilkan nilai querynya berarti berhasil

```

praktikum-LinkNavigation > my-app > src > pages > shop > [...slug].tsx > ...
1 import { useRouter } from "next/router";
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const halamanToko = ()=> {
5   const Router = useRouter();
6   console.log(Router); x15 * { pathname: '/shop/[...slug]', route: '/shop/[...slug]', query: { slug: [...] }, asPath: '/shop/baju/tshirt/pria', components:... }
7   // const { query } = Router;
8   return (
9     <div>
10       <h1>Halaman Toko</h1>
11     </div>
12   );
13 export default halamanToko;
14
15
16
17
18

```

```

1 import { useRouter } from "next/router";
2
3 Windsurf Refactor | Explain | Generate JSDoc | X
4 ist HalamanToko = () => [
5   const router = useRouter();
6   console.log(router); // { pathname: '/shop/...slug', route: '/shop/...slug', query: { slug: [...] }, asPath: '/shop/baju/tshirt/pria', components: ... }
7
8   return (
9     <div>
10       <h1>Halaman Toko</h1>
11     </div>
12   );
13
14 export default HalamanToko;

```

- Modifikasi `[...slug].tsx` untuk menampilkan nilai query

```

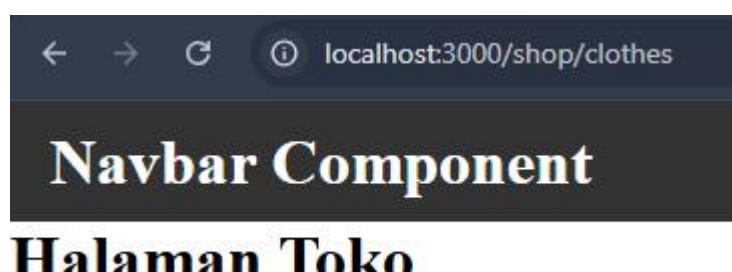
praktikum-LinkNavigation > my-app > src > pages > shop > ⚡ [...slug].tsx > ...
1 import { useRouter } from "next/router";
2
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 const halamanToko = () => {
6   // const Router = useRouter();
7   //console.log(Router);
8   const { query } = useRouter();
9   return (
10     <div>
11       <h1>Halaman Toko</h1>
12       <p>Toko: ${query.slug[0]} - ${query.slug[1]}</p> /* menggunakan backtick bukan petik satu*/
13     </div>
14   );
15 };
16
17 export default halamanToko;
src > pages > shop > ⚡ [...slug].tsx > HalamanToko
1 import { useRouter } from "next/router";
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const HalamanToko = () => [
5   // const router = useRouter();
6   // console.log(router);
7   const {query} = useRouter();
8   return (
9     <div>
10       <h1>Halaman Toko</h1>
11       <p>Toko: ${query.slug[0]} - ${query.slug[1]}</p>
12     </div>
13   );
14
15 export default HalamanToko;

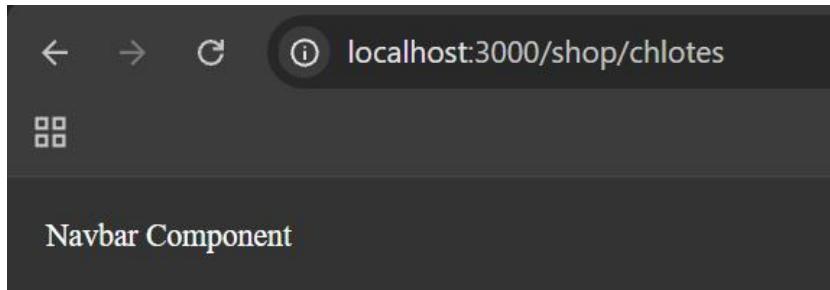
```

Langkah 3 – Pengujian Catch-All Route

Akses URL berikut di browser:

`/shop/clothes`

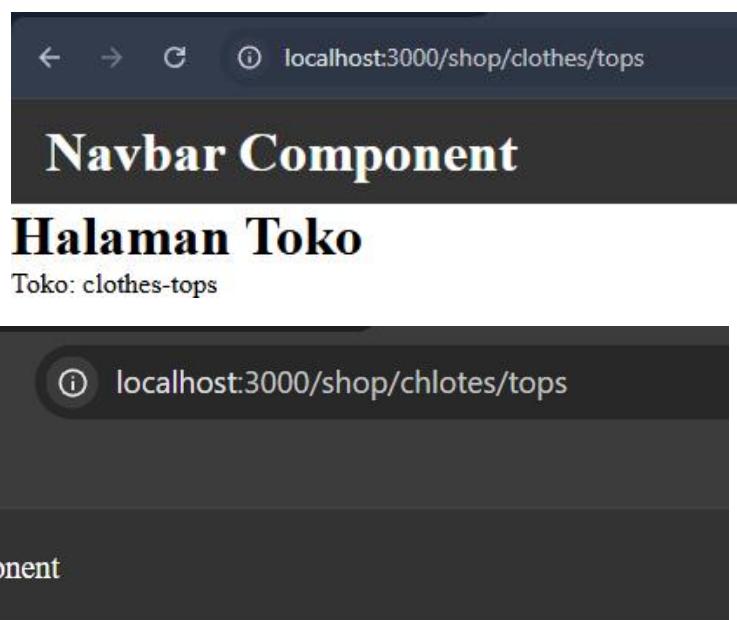




Halaman Toko

Toko: chlotes-undefined
footer

/shop/clothes/tops

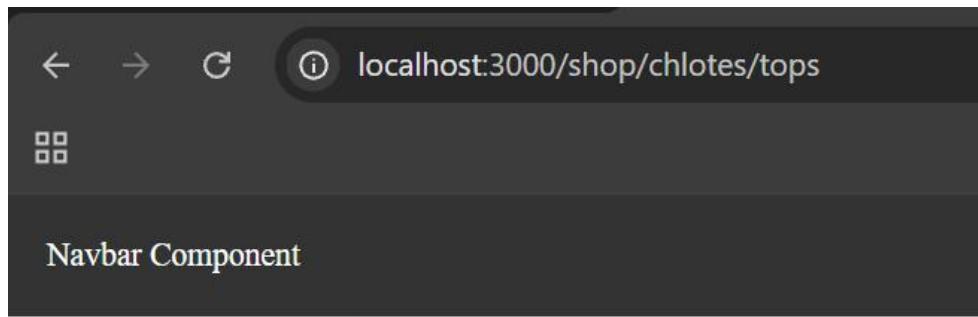


Halaman Toko

Toko: chlotes-tops
footer

/shop/clothes/tops/t-shirt





Navbar Component

Halaman Toko

Toko: chlotes-tops

Jika dilihat ada yang terbaca undefined dan ada yang tidak terbaca ini dikarena segmennya dibatasi Cuma array[0] dan array[1]. Solusinya bagaimana ?

Modifikasi [...slug].tsx menjadi berikut

```

⚙️ [...slug].tsx ×
```

```

praktikum-LinkNavigation > my-app > src > pages > shop > ⚙️ [...slug].tsx > ...
1 import { useRouter } from "next/router";
2
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 const halamanToko = () => {
6   // const Router = useRouter();
7   //console.log(Router);
8   const { query } = useRouter();
9   return (
10    <div>
11      <h1>Halaman Toko</h1>
12      {/* <p>Toko: ${query.slug[0]}-${query.slug[1]}</p> menggunakan backtick bukan petik satu */}
13      <p>
14        Toko: {Array.isArray(query.slug) ? query.slug.join("-") : query.slug}
15      </p>
16    </div>
17  );
18
19
20 export default halamanToko;
21
```

```

1 import { useRouter } from "next/router";
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const HalamanToko = () => {
5   // const router = useRouter();
6   // console.log(router);
7   const {query} = useRouter();
8   return (
9     <div>
10       <h1>Halaman Toko</h1>
11       {/* <p>Toko: ${query.slug[0]}-${query.slug[1]}-${query.slug[2]}</p> */}
12       <p>
13         Toko: {Array.isArray(query.slug) ? query.slug.join("-") : query.slug}
14       </p>
15     </div>
16   );
17
18 export default HalamanToko;

```

Jalankan browser : Berapapun banyaknya segment tetap terbaca

A screenshot of a web browser window. The address bar at the top shows the URL `localhost:3000/shop/clothes/tops/t-shirt`. The main content area has a dark background. At the top of this area, there is a dark bar containing the text "Navbar Component" in white. Below this bar, the text "Halaman Toko" is displayed in large, bold, black font. A small amount of text "Toko: clothes-tops-t-shirt" is visible below the main title.

A screenshot of a web browser window. The address bar at the top shows the URL `localhost:3000/shop/clothes/tops/t-shirt/1/2/3`. The main content area has a dark background. At the top of this area, there is a dark bar containing the text "Navbar Component" in white. Below this bar, the text "Halaman Toko" is displayed in large, bold, black font. A small amount of text "Toko: clothes-tops-t-shirt-1-2-3" is visible below the main title.

A screenshot of a web browser window. The address bar at the top shows the URL `localhost:3000/shop/chlotes/tops/t-shirt`. The main content area has a dark background. At the top of this area, there is a dark bar containing the text "Navbar Component" in white. Below this bar, the text "Halaman Toko" is displayed in large, bold, black font. A small amount of text "Toko :chlotes-tops-t-shirt" is visible below the main title.

Halaman Toko
Toko :chlotes-tops-t-shirt
Footer

A screenshot of a web browser window. The address bar at the top shows the URL `localhost:3000/shop/chlotes/tops/t-shirt/1/2/3`. The main content area has a dark background. At the top of this area, there is a dark bar containing the text "Navbar Component" in white. Below this bar, the text "Halaman Toko" is displayed in large, bold, black font. A small amount of text "Toko :chlotes-tops-t-shirt-1-2-3" is visible below the main title.

Halaman Toko
Toko :chlotes-tops-t-shirt-1-2-3
Footer

Untuk saat ini gunakan

```

praktikum-LinkNavigation > my-app > src > pages > shop > ⇝ [...slug].tsx > ...
1 import { useRouter } from "next/router";
2
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 const halamanToko = () => {
6   // const Router = useRouter();
7   //console.log(Router);
8   const { query } = useRouter();
9   return (
10    <div>
11      <h1>Halaman Toko</h1>
12      <p>Toko: ${query.slug && query.slug[0] + "-" + query.slug[1]}</p> /* menggunakan backtick bukan petik satu*/
13    </div>
14  );
15};
16
17 export default halamanToko;

src / pages / shop / ⇝ [...slug].tsx / HalamanToko
1 import { useRouter } from "next/router";
2
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const HalamanToko = () => {
5   // const router = useRouter();
6   // console.log(router);
7   const {query} = useRouter();
8   return (
9     <div>
10       <h1>Halaman Toko</h1>
11       <p>Toko: ${query.slug && query.slug[0] + "-" + query.slug[1]}</p>
12       /* <p>
13         Toko :{Array.isArray(query.slug) ? query.slug.join("-") : query.slug}
14       </p> */
15     </div>
16   );
17
18 export default HalamanToko;

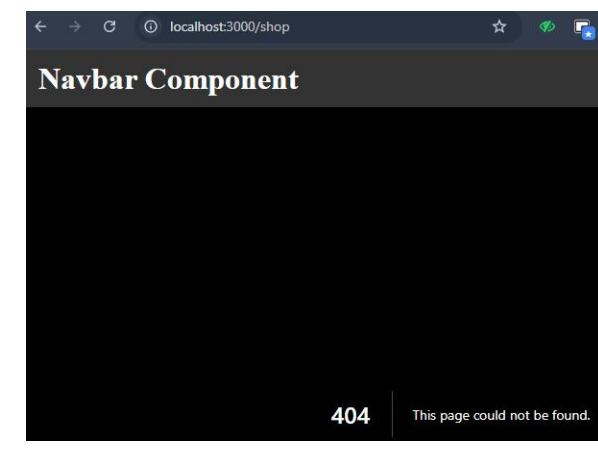
```

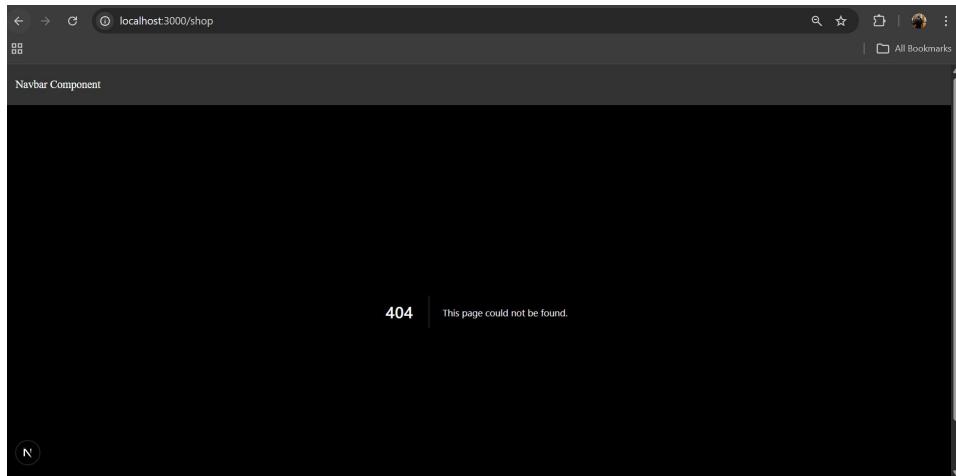
Perhatikan bahwa:

- slug berbentuk array
- Isi halaman berubah sesuai URL

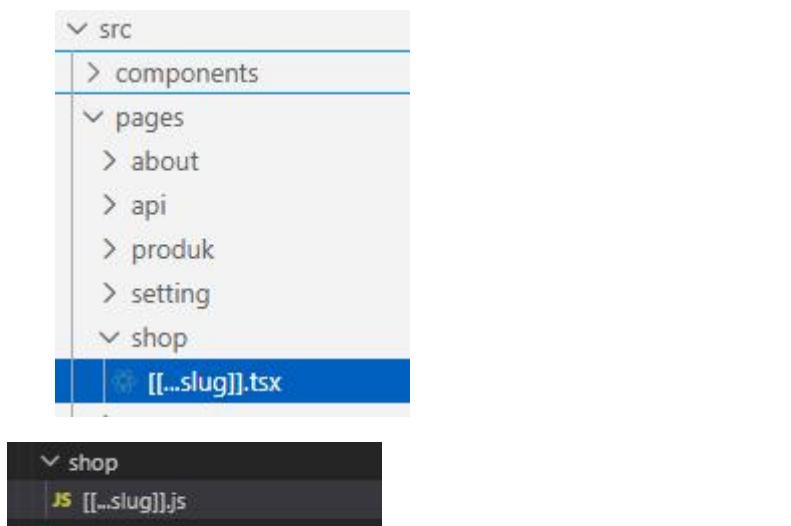
Langkah 4 – Optional Catch-All Route

1. Jika menggunakan [...slug].js maka ketika mengakses shop akan terjadi error



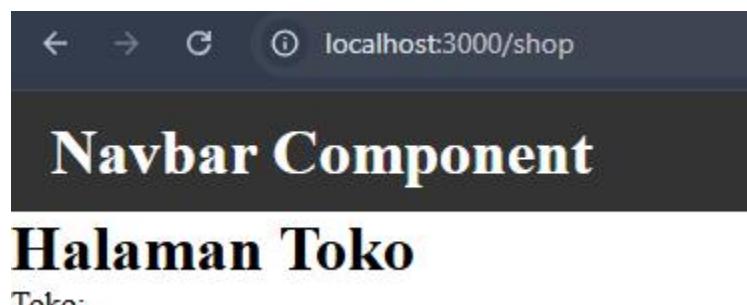


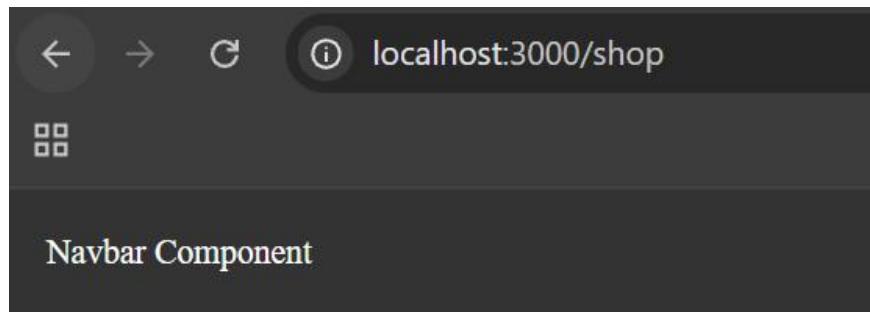
2. Solusinya dengan Rename file: [...slug].js → [[...slug]].js



3. Sekarang akses:

/shop





Halaman Toko

Toko: undefined
footer

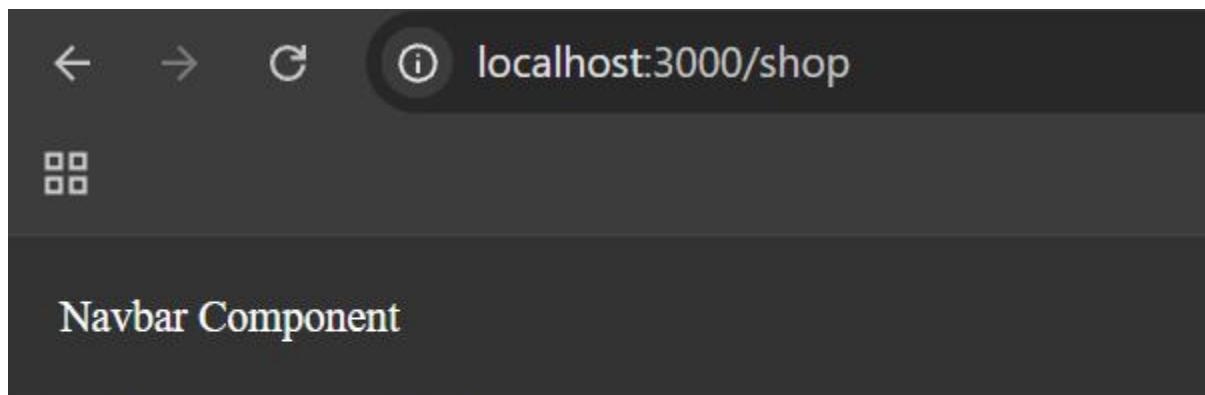
4. Halaman dapat diakses meskipun tanpa parameter.

Langkah 5 – Validasi Parameter

Tambahkan validasi agar tidak error saat slug kosong:

```
<p>  
  Kategori: {slug ? slug[0] : "Semua Kategori"}  
</p>
```

```
1  port { useRouter } from "next/router";  
2  
3  const HalamanToko = () => {  
4    // const router = useRouter();  
5    // console.log(router);  
6    const {query} = useRouter();  
7    return (  
8      <div>  
9        <h1>Halaman Toko</h1>  
10       <p>  
11         Toko: ` ${query.slug && query.slug[0] + "-" + query.slug[1]} `  
12       </p>  
13       <p>  
14         Kategori: ` ${slug ? slug[0] : "semua kategori"} `  
15       </p>  
16     </div>  
17   );  
18  
19  
20  
21  port default HalamanToko;
```

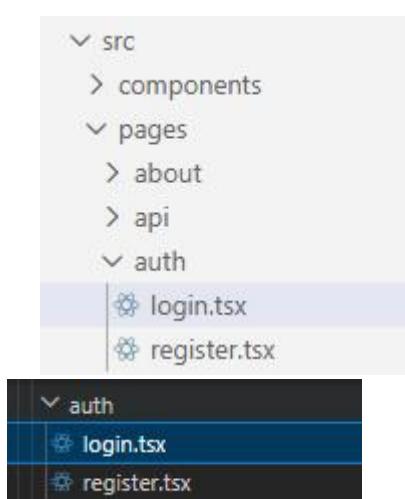


Halaman Toko

Toko: undefined
Kategori: semua kategori
footer

Langkah 6 – Membuat Halaman Login & Register

1. Buat folder:
2. pages/auth
3. Buat file:
 - o login.jsx
 - o register.jsx



Modifikasi login.jsx:

```
praktikum-LinkNavigation > my-app > src > pages > auth > login.jsx > halamanLogin
1 import Link from "next/link";
2
3 Windurf: Refactor | Explain | Generate JSDoc | X
4 const halamanLogin = () => {
5   return (
6     <div>
7       <h1>Halaman Login</h1>
8         <Link href="/auth/register">Ke Halaman Register</Link>
9     </div>
10   );
11
12 export default halamanLogin;
13
```

```
src > pages > auth > login.tsx > default
1 import Link from "next/link";
2
3 const halamanlogin = () => {
4     return (
5         <div>
6             <h1>Halaman Login</h1>
7             <Link href="/auth/register">
8                 Ke Halaman Register
9             </Link>
10        </div>
11    );
12};
13 export default halamanlogin;
```

Modifikasi register.jsx:

praktikum-LinkNavigation > my-app > src > pages > auth > register.tsx > halamanRegister

```
1 import Link from "next/link";
2
3 const halamanRegister = () => {
4     return (
5         <div>
6             <h1>Halaman Register</h1>
7             <Link href="/auth/login">Ke Halaman Login</Link>
8         </div>
9     );
10};
11
12 export default halamanRegister;
13
```

```
1 import Link from "next/link";
2
3 const halamanregister = () => {
4     return (
5         <div>
6             <h1>Halaman Register</h1>
7             <Link href="/auth/login">
8                 Ke Halaman Login
9             </Link>
10        </div>
11    );
12};
13 export default halamanregister;
```

Langkah 7 – Navigasi Imperatif (router.push)

1. Tambahkan button login:

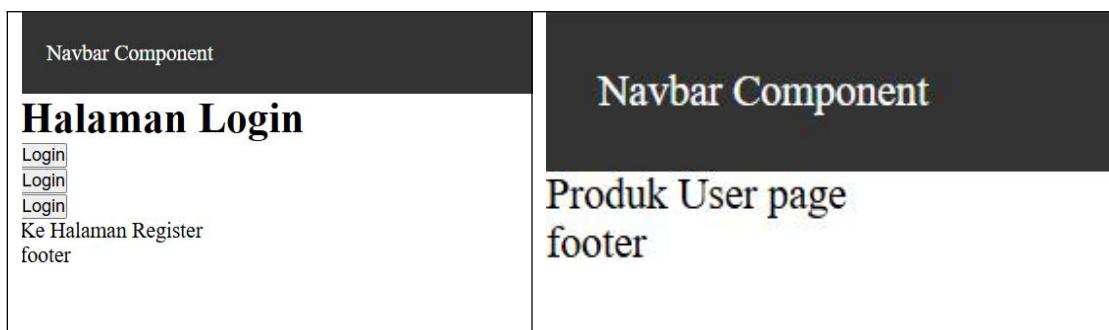
```
praktikum-LinkNavigation > my-app > src > pages > auth > login.tsx > halamanLogin
  1 import Link from "next/link";
  2 import { useRouter } from "next/router";
  3
  4 Windsurf: Refactor | Explain | Generate JSDoc | X
  5 const halamanLogin = () => {
  6   const {push} = useRouter();
  7   const handlerLogin = () => {
  8     // logic login disini
  9     push('/produk');
 10 }
 11 return (
 12   <div>
 13     <h1>Halaman Login</h1>
 14     <button onClick={handlerLogin}>Login</button> <br />
 15     <button onClick={() => push('/produk')}>Login</button> <br />
 16     <button onClick={() => handlerLogin()}>Login</button> <br />
 17     <Link href="/auth/register">Ke Halaman Register</Link>
 18   </div>
 19 );
 20 };
 21
 22 export default halamanLogin;
 23

src / pages / auth > login.tsx > halamanlogin
  1 import Link from "next/link";
  2 import { useRouter } from "next/router";
  3
  4 Windsurf: Refactor | Explain | Generate JSDoc | X
  5 const halamanlogin = () => {
  6   const {push} = useRouter();
  7   const handleLogin = () => {
  8     push("/produk");
  9   }
 10
 11   return [
 12     <div>
 13       <h1>Halaman Login</h1>
 14       <button onClick={handleLogin}>Login</button> <br />
 15       <button onClick={() => push("/produk")}>Login</button> <br />
 16       <button onClick={()=> handleLogin()}>Login</button> <br />
 17       <Link href="/auth/register">
 18         Ke Halaman Register
 19       </Link>
 20     </div>
 21   ];
 22
 23 export default halamanlogin;
```

Kode	Cara Kerja	Kapan Dipanggil	Kelebihan	Kekurangan	Rekomendasi
onClick={handlerLogin}	Mengirim referensi fungsi ke event handler	Saat tombol diklik	Paling bersih, efisien, dan best practice	Tidak bisa kirim parameter langsung	Sangat direkomendasikan
onClick={() => push('/produk')}	Arrow function memanggil fungsi di dalamnya	Saat tombol diklik	Praktis untuk aksi sederhana	Kurang reusable jika logika bertambah	Direkomendasikan untuk navigasi sederhana
onClick={() => handlerLogin()}	Arrow function membungkus pemanggilan fungsi	Saat tombol diklik	Fleksibel untuk kirim parameter	Redundant jika tanpa parameter	Direkomendasikan hanya jika perlu argumen
onClick={handlerLogin()}	Fungsi langsung dieksekusi	Saat render	Tidak ada	Bug: fungsi tidak menunggu klik	Tidak direkomendasikan

Saat ini gunakan **onClick={() => handlerLogin()}**

2. Klik tombol dan perhatikan perpindahan halaman tanpa reload.
3. Jika di klik button login maka akan menuju /produk



Note :

- Pastikan code untuk redirect tidak aktif , jika aktif maka ketika masuk ke produk akan langsung redirect ke login

```
6.praktikum-ApiRoutes > my-app > src > pages > produk > index.tsx > ...
1 import { useRouter } from "next/router";
2 import { useEffect, useState } from "react";
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 const produk = () => {
6   // const [isLoggedIn, setIsLogin] = useState(false);
7   // const { push } = useRouter();
8   // useEffect(() => {
9   //   if (!isLoggedIn) {
10   //     push("/auth/login");
11   //   }
12   // }, []);
13   return (
14     <div>Produk User Page</div>
15   );
16 };
17
18 export default produk;
```

```
src > pages > produk > index.tsx > produk
1 import {useRouter} from 'next/router';
2 import {useEffect, useState} from 'react';
3 Windsurf: Refactor | Explain | Generate JSDoc | X
4 const produk=() => {
5   // const [isLoggedIn, setIsLogin] = useState(false);
6   // const {push} = useRouter();
7   // useEffect(() => {
8   //   if(!isLoggedIn){
9   //     push("/auth/login");
10  //   }
11  // }, []);
12  return(
13    <div>
14      Produk User page
15    </div>
16  );
17
18 export default produk;
```

Langkah 8 – Simulasi Redirect (Belum Login)

1. Di halaman product, pada index.tsx tambahkan beberapa code berikut:

```
praktikum-LinkNavigation > my-app > src > pages > produk > index.tsx > ...
1 import { useRouter } from "next/router";
2 import { useEffect, useState } from "react";
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 const produk = () => {
6   const [isLoggedIn, setIsLogin] = useState(false);
7   const { push } = useRouter();
8   useEffect(() => {
9     if (!isLoggedIn) {
10       push("/auth/login");
11     }
12   }, []);
13   return (
14     <div>Produk User Page</div>
15   );
16 };
17
18 export default produk;
Windsurf: Generate (Ctrl+I)
```

```

src > pages > produk > index.tsx > default
1 import {useRouter} from 'next/router';
2 import { useEffect, useState } from 'react';
Windsurf: Refactor | Explain | Generate JSDoc | X
3 const produk=() => {
4     const [isLoggedIn, setIsLogin] = useState(false);
5     const {push} = useRouter();
6     useEffect(() => {
7         if(!isLoggedIn){
8             push("/auth/login");
9         }
10    }, []);
11    return(
12        <div>
13            Produk User page
14        </div>
15    );
16};
17
18 export default produk;

```

2. Jika Akses /product → otomatis diarahkan ke login.
-

B. Tugas Praktikum

Tugas 1 (Wajib)

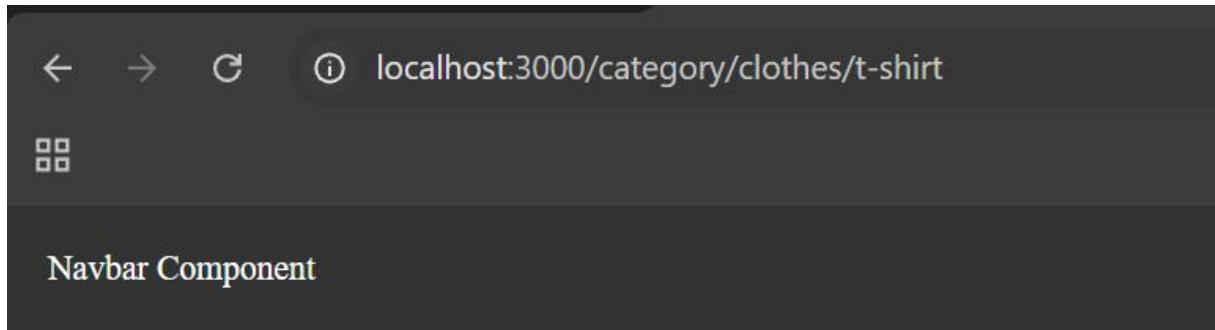
- Buat catch-all route:
- /category/[...slug].js

```

src > pages > category > [...slug].tsx > default
1 import { useRouter } from "next/router";
2
Windsurf: Refactor | Explain | Generate JSDoc | X
3 const HalamanKategori = () => {
4     const {query} = useRouter();
5     const slugArray = Array.isArray(query.slug) ? query.slug : [];
6     return (
7         <div>
8             <h1>category Slug</h1>
9             {slugArray.length > 0 ? (
10                 <ul>
11                     {slugArray.map((item, index) => (
12                         <li key={index}>{item}</li>
13                     )));
14                 </ul>
15             ) : (
16                 <p>Tidak ada parameter</p>
17             )}
18         </div>
19     );
20 }
21
22 export default HalamanKategori;

```

- Tampilkan seluruh parameter URL dalam bentuk list.

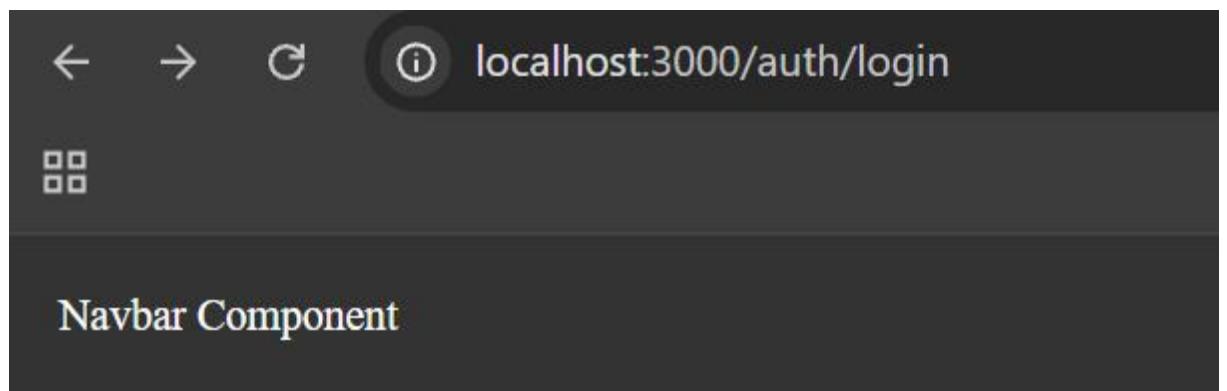


Tugas 2 (Wajib)

- Buat navigasi:
 - Login → Product (imperatif)

- Login ↔ Register (Link)

```
src > pages > auth > login.tsx > [?] halamanlogin
  1 import Link from "next/link";
  2 import { useRouter } from "next/router";
  3
  4 const halamanlogin = () => {
  5   // const {push} = useRouter();
  6   const router = useRouter();
  7
  8   const handleLogin = () => {
  9     // push("/produk");
 10     console.log("Login berhasil");
 11     localStorage.setItem("isLoggedIn", "true");
 12     router.push("/produk");
 13   }
 14
 15   return (
 16     <div>
 17       <h1>Halaman Login</h1>
 18       <button onClick={handleLogin}>Login ke produk</button> <br />
 19       {/* <button onClick={() => push("/produk")}>Login</button> <br />
 20       <button onClick={()=> handleLogin()}>Login</button> <br /> */}
 21       <Link href="/auth/register">
 22         Ke Halaman Register
 23       </Link>
 24     </div>
 25   );
 26 }
 27 export default halamanlogin;
```



Tugas 3 (Pengayaan)

- Terapkan redirect otomatis ke login jika user belum login.

```
1 import { useRouter } from "next/router";
2 import { useEffect, useState } from "react";
3
4 Windsurf: Refactor | Explain | Generate JSDoc | X
5 const Produk = () => {
6     const router = useRouter();
7     const [isChecked, setIsChecked] = useState(false);
8
9     useEffect(() => {
10         const isLogin = localStorage.getItem("isLogin");
11
12         if (isLogin !== "true") {
13             router.replace("/auth/login");
14         } else {
15             setIsChecked(true);
16         }
17     }, [router]);
18
19     if (!isChecked) {
20         return null;
21     }
22
23     return (
24         <div>
25             Produk User page
26         </div>
27     );
28
29 export default Produk;
```

C.Pertanyaan Evaluasi

1. Apa perbedaan [id].js dan [...slug].js?

Jawab : [id].js menangkap parameter berupa string sementara [...slug].js menangkap banyak segmen url sekaligus dan nilainya berupa array

2. Mengapa slug berbentuk array?

Jawab : slug berbentuk array karena berisi bagian URL yang dipisahkan oleh tanda /

3. Kapan sebaiknya menggunakan Link dan router.push()?

Jawab : link digunakan untuk navigasi biasa melalui klik. Router.push() digunakan untuk navigasi otomatis atau berdasarkan kondisi

4. Mengapa navigasi Next.js tidak me-refresh halaman?

Jawab : karena menggunakan client-side navigation, sehingga hanya mengganti komponen tanpa reload seluruh halaman