

Praktikum 2 - Matakuliah Pilihan 1 (Web)

Program Studi: Teknik Informatika

Lakukan praktikum dibawah ini, dan buat screenshot untuk pembuktian mengerjakan setiap poin dengan mengisi tabel dibawah, kemudian tunjukan hasil akhir dari men-share repository github yang telah dibuat.

A. Membuat Project [Next.js](#)

1. Buka command prompt atau lewat visual code
2. Ketik : `npx create-next-app@latest praktikum2`
3. Jika ada pilihan Typescript → Yes, App Router → Yes, Tailwindcss → Yes, include src → Yes.
4. Masuk ke folder proyek anda dengan
cd praktikum2
5. Jalankan proyek dengan mengetik
npm run dev (Pastikan di dalam alamat folder proyek)
Cek gunakan browser sesuai port yang dibuka dalam cmd.

B. Membuat dan Menggunakan components

1. Buat folder baru di dalam app dengan nama component. Sehingga strukturnya seperti ini /src/app/components
2. Pada folder components, buat file baru dengan nama Navbar.tsx dengan berisi kode seperti dibawah ini: (/src/app/components/Navbar.tsx)

```
src > app > components > Navbar.tsx > Navbar
1  import Link from "next/link";
2
3  export default function Navbar() {
4    return (
5      <nav style={{
6        padding: "10px",
7        background: "#333",
8        color: "#fff",
9        display: "flex",
10       gap: "20px"
11     }}>
12        <Link href="/" style={{ color: "#fff", textDecoration: "none" }}>Home</Link>
13        <Link href="/about" style={{ color: "#fff", textDecoration: "none" }}>About</Link>
14        <Link href="/blog/1" style={{ color: "#fff", textDecoration: "none" }}>Blog</Link>
15        <Link href="/contact" style={{ color: "#fff", textDecoration: "none" }}>Contact</Link>
16      </nav>
17    );
18  }
```

3. Pada file page.tsx di app, (/src/app/page.tsx) ubah kode program seperti gambar dibawah ini (kode default dihapus semua)

```
src > app > page.tsx > Home
1 import Navbar from "../components/Navbar";
2
3 export default function Home() {
4   return (
5     <Navbar></Navbar>
6   );
7 }
8
9
10
```

4. Cek lewat browser, apa yang ditampilkan? Screenshot hasilnya.
5. Kunjungi website tailwind flowbite, ambil salah satu component Navbar dan terapkan di dalam kode program Navbar.tsx (<https://flowbite.com/docs/components/navbar/>)
6. Kode HTML dari flowbite dimasukan ke dalam **tag return{ ... }**, dan "class" di replace menjadi "className" gunakan ctrl+H (Replace) agar mempermudah penulisan kode program.
7. Ganti menu menjadi Home, Blog, About dan contact dengan link menuju /, /blog/1, /about, /contact
8. Screenshot hasil dari Navbar dengan component flowbite.

C. Membuat dan Menggunakan Dynamic Components

1. Buat sebuah folder di dalam app dengan nama **about**. Sehingga strukturnya menjadi /src/app/about
2. Buat sebuah folder lagi di dalam folder about dengan nama components, sehingga strukturnya /src/app/about/components
3. Buat file Card.tsx di dalam folder components tersebut. Sehingga lokasinya ada di /src/app/about/components/Card.tsx
4. **(Warning!!) Card.tsx berbeda folder dengan Navbar.tsx**
5. Isi Card.tsx dengan kode program seperti screenshot dibawah ini

```
src > app > about > components > Card.tsx > Card
1
2 export default function Card({ title, description }) {
3   return (
4     <div style={{ border: "1px solid #ddd", margin: "10px", padding: "10px" }}>
5       <h2>{title}</h2>
6       <p>{description}</p>
7     </div>
8   );
9 }
10
```

6. Buat **File Baru** pages.tsx di dalam folder about tersebut, /src/app/about/page.tsx
7. **(Warning!!)** pages.tsx **tidak sama** dengan page.tsx pada poin C. page.tsx ini ada di dalam folder /about/
8. Isi pages.tsx dengan kode program seperti dibawah ini:

```
src > app > about > page.tsx > ...
1  import Card from "../components/Card";
2
3  export default function Home() {
4    return (
5      <div>
6        <Card title="Komponen 1" description="Ini adalah komponen pertama" />
7        <Card title="Komponen 2" description="Ini adalah komponen kedua" />
8      </div>
9    );
10  };
11 }
```

9. Gunakan browser dan menuju ke link localhost:3000/about . Screenshot hasilnya
10. Tambahkan di pages.tsx tersebut, dengan Navbar pada components navbar di parent, caranya dengan menulis Navbar di return, nanti visual code akan otomatis mengimport Navbar di atas kode program.

```
src > app > about > page.tsx > Home
1  import Navbar from "../components/Navbar";
2  import Card from "../components/Card";
3
4  export default function Home() {
5    return (
6      <>
7        <Navbar></Navbar>
8        <div>
9          <Card title="Komponen 1" description="Ini adalah komponen pertama" />
10         <Card title="Komponen 2" description="Ini adalah komponen kedua" />
11        </div>
12      </>
13    );
14  };
15 }
```

11. Penulisan <> dan </> artinya untuk menambahkan komponen-komponen yang berlainan, dan agar posisi pada html berurutan dari atas kebawah dengan membuat line baru.
12. Screenshot hasilnya.
13. Ubah Card.tsx dengan menggunakan komponen card dari flowbite
<https://flowbite.com/docs/components/card/> (Pilih salah satu)
14. Tidak lupa untuk mengganti class ke className dan menambahkan {title} dan {description}
15. Screenshot hasilnya.

D. Dynamic Routes

1. Dalam poin sebelumnya kita sudah menerapkan Static Routes, seperti /about. Pada poin kali ini kita akan mempraktekan dynamic routes
2. Buat folder blog di dalam /src/app. Sehingga strukturnya seperti ini /src/app/blog
3. Buat folder [id] di dalam folder blog, sehingga seperti ini /src/app/blog/[id]
4. Buat file dengan nama page.tsx di dalam folder [id] Sehingga file tersebut memiliki lokasi seperti ini /src/app/blog/[id]/page.tsx
5. Tulis kode program pada file [id]/page.tsx seperti pada kode program dibawah ini

```
src > app > blog > [id] > ✎ page.tsx > ...  
1  
2  
3 interface BlogProp {  
4   |   params: { id: string };  
5   |  
6   |  
7   export default function BlogDetail({ params }: BlogProp) {  
8     |   return <h1>Detail Blog dengan ID: {params.id}</h1>;  
9     |  
10    |
```

6. Coba menggunakan browser untuk menampilkan hasilnya dengan mengunjungi
localhost:3000/blog/1
localhost:3000/blog/23
localhost:3000/blog/Indonesia-Raya
7. Screenshot hasilnya
8. Bagaimana menerapkan 2 params misalkan id dan title?
9. Buat folder baru [title] dan letakan page.tsx di dalamnya sehingga struktur file dan foldernya menjadi seperti ini [id]/[title]/page.tsx
10. Pada kode program page.tsx tambahkan
params: {id: string; title: string};
Dan pada return <h1> tambahkan params.title terserah ditulis dimana.
11. Coba menggunakan browser untuk menampilkan hasilnya dengan mengunjungi
localhost:3000/blog/1945/Indonesia-Raya
12. Tambahkan folder components, tambahkan Card dari flowbite, seperti poin C-13.
ubah id dan title untuk mengisi component card tersebut.
13. Nama file dan fungsi tidak boleh sama. Karena ada di folder blog, maka gunakan nama BlogCard seperti pada kode di bawah ini

```
src > app > blog > [id] > [title] > components > BlogCard.tsx > ...
1
2 export default function BlogCard({ title, description }) {
3   return (
4     <div style={{ border: "1px solid #ddd", margin: "10px", padding: "10px" }}>
5       <h2>{title}</h2>
6       <p>{description}</p>
7     </div>
8   );
9 }
10
```

14. Screenshot hasil dari pengerjaan ini.

E. Github + Visual Code

1. Buat proyek di Github dengan nama **Latihan2**

git init

git add .

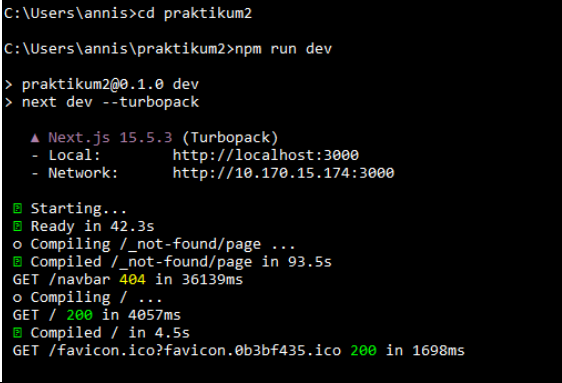
git commit -m "first commit"

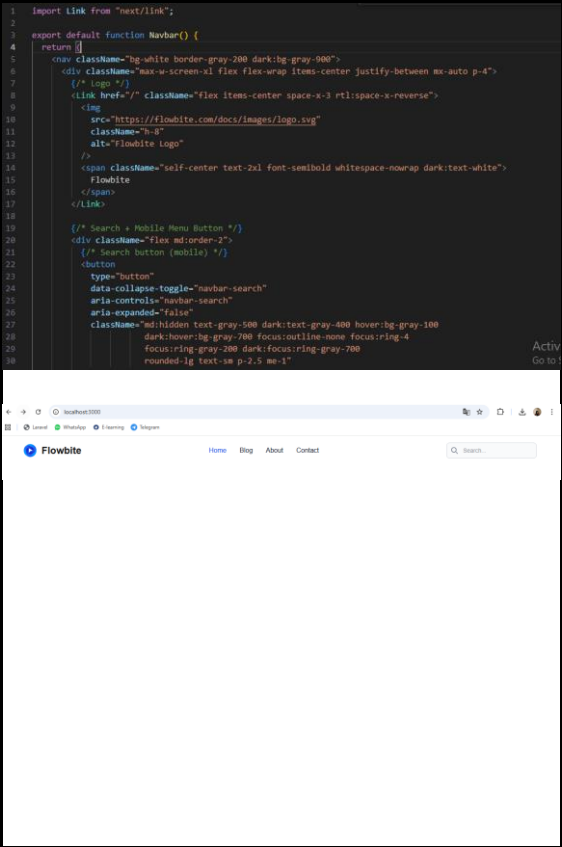
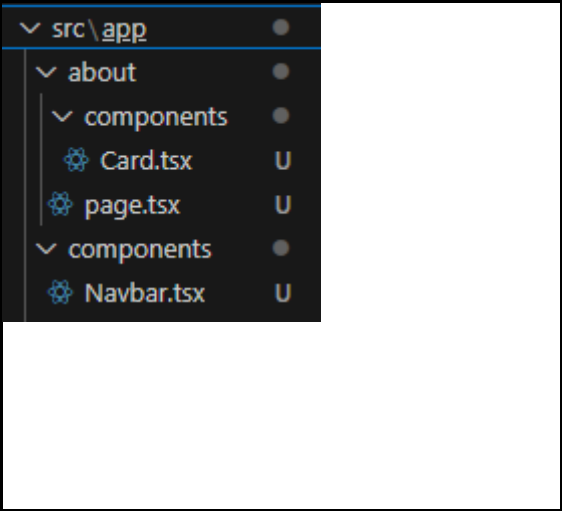
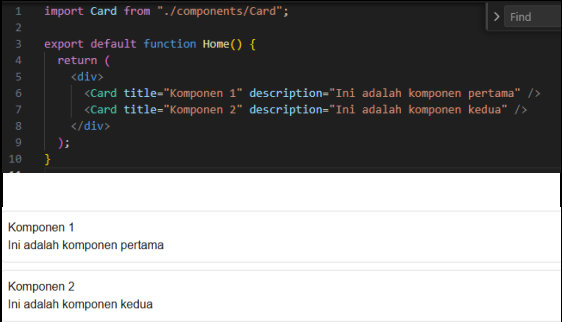
git branch -M main

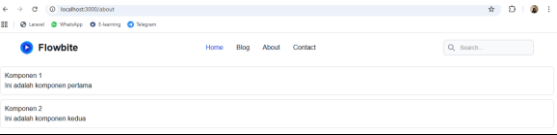
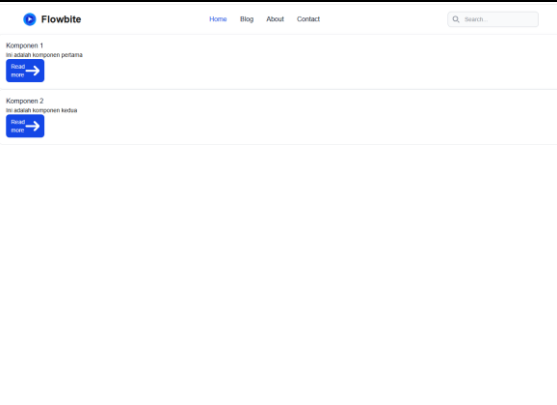
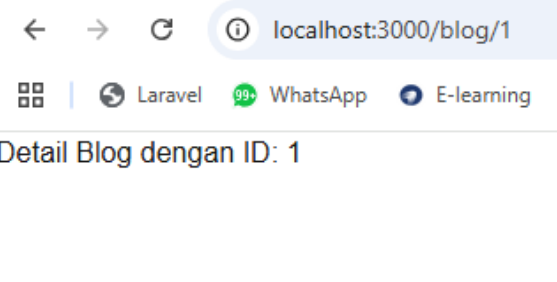
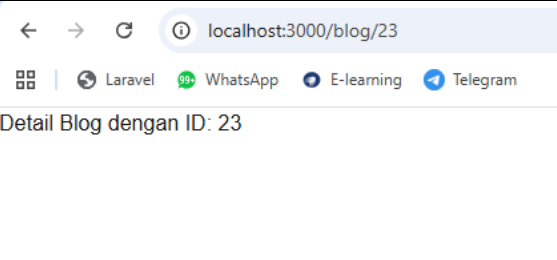
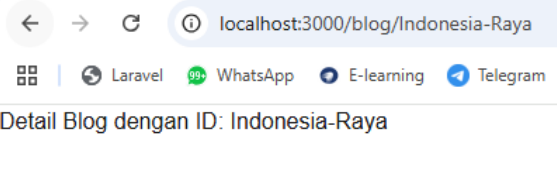
git remote add origin <https://github.com/agunghakase/Latihan1.git>

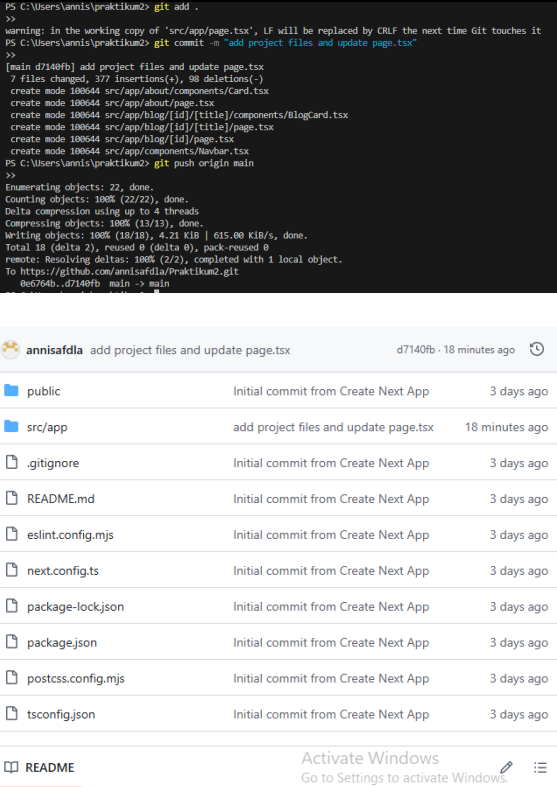
git push -u origin main

Hasil Pengerjaan

No.	Instruksi	Screenshot	Kendala/Saran
A.	Membuat Project Next.js		
1.	<p>1. Buka command prompt atau lewat visual code</p> <p>2. Ketik : <code>npx create-next-app@latest praktikum2</code> .</p> <p>3. Jika ada pilihan Typescript → Yes, App Router → Yes, Tailwindcss → Yes, include src → Yes</p>		
2.	<p>4. Masuk ke folder proyek anda dengan <code>cd praktikum2</code></p> <p>5. Jalankan proyek dengan mengetik <code>npm run dev</code> (Pastikan di dalam alamat folder proyek) Cek gunakan browser sesuai port yang dibuka dalam cmd.</p>		
B.	Membuat dan Menggunakan components		
1.	<p>1. Buat folder baru di dalam app dengan nama component. Sehingga strukturnya seperti ini <code>/src/app/components</code></p> <p>2. Pada folder components, buat file baru dengan nama <code>Navbar.tsx</code> dengan berisi kode seperti dibawah ini: <code>(/src/app/components/Navbar.tsx)</code></p>		

<p>2.</p>	<p>4. Cek lewat browser, apa yang ditampilkan? Screenshot hasilnya.</p> <p>5. Kunjungi website tailwind flowbite, ambil salah satu component Navbar dan terapkan di dalam kode program Navbar.tsx (https://flowbite.com/docs/components/navbar/)</p> <p>6. Kode HTML dari flowbite dimasukan ke dalam tag <code>return{ ... }</code>, dan "class" di replace menjadi "className" gunakan <code>ctrl+H</code> (Replace) agar mempermudah penulisan kode program.</p> <p>7. Ganti menu menjadi Home, Blog, About dan contact dengan link menuju <code>/</code>, <code>/blog/1</code>, <code>/about</code>, <code>/contact</code></p> <p>8. Screenshot hasil dari Navbar dengan component flowbite.</p>		
<p>C.</p>	<p>Membuat dan Menggunakan Dynamic Components</p>		
<p>1.</p>	<p>1. Buat sebuah folder di dalam app dengan nama about. Sehingga strukturnya menjadi <code>/src/app/about</code></p> <p>2. Buat sebuah folder lagi di dalam folder about dengan nama components, sehingga strukturnya <code>/src/app/about/components</code></p> <p>3. Buat file Card.tsx di dalam folder components tersebut. Sehingga lokasinya ada di <code>/src/app/about/components/Card.tsx</code></p>		
<p>2.</p>	<p>Gunakan browser dan menuju ke link <code>localhost:3000/about</code>. Screenshot hasilnya</p>		

3.	Tambahkan di pages.tsx tersebut, dengan Navbar pada components navbar di parent, caranya dengan menulis Navbar di return, nanti visual code akan otomatis mengimport Navbar di atas kode program.	<pre> src > app > about > pages.tsx > Home 1 import Navbar from "../components/Navbar"; 2 import Card from "../components/Card"; 3 4 export default function Home() { 5 return (6 <> 7 <Navbar></Navbar> 8 <div> 9 <Card title="Komponen 1" description="Ini adalah komponen pertama" /> 10 <Card title="Komponen 2" description="Ini adalah komponen kedua" /> 11 </div> 12 </> 13); 14} </pre> 	
4.	<p>13. Ubah Card.tsx dengan menggunakan komponen card dari flowbite https://flowbite.com/docs/components/card/ (Pilih salah satu)</p> <p>14. Tidak lupa untuk mengganti class ke className dan menambahkan {title} dan {description}</p> <p>15. Screenshot hasilnya.</p>		
D.	Dynamic Routes		
1.	<p>Coba menggunakan browser untuk menampilkan hasilnya dengan mengunjungi</p> <p>localhost:3000/blog/1</p> <p>localhost:3000/blog/23</p> <p>localhost:3000/blog/Indonesia-Raya</p> <p>7. Screenshot hasilnya</p>	  	
2.	<p>10. Pada kode program page.tsx tambahkan params: {id: string; title: string}; Dan pada return tambahkan params.title terserah ditulis dimana.</p> <p>11. Coba menggunakan browser untuk menampilkan hasilnya dengan mengunjungi localhost:3000/blog/1945/In</p>	<p>Blog Post Details</p> <p>Title: Indonesia Raya</p> <p>Post ID: 1945</p> <p>URL Title: Indonesia-Raya</p> <p>Content Preview</p> <p>Ini adalah halaman blog dengan ID 1945 dan judul "Indonesia Raya". Dynamic routing memungkinkan kita mengakses kedua parameter ini melalui URL.</p> <p>URL Parameters:</p> <ul style="list-style-type: none"> • ID: 1945 • Title: Indonesia Raya 	

	<p>donesia-Raya</p> <p>12. Tambahkan folder components, tambahkan Card dari flowbite, seperti poin C-13. ubah id dan title untuk mengisi component card tersebut.</p> <p>13. Nama file dan fungsi tidak boleh sama. Karena ada di folder blog, maka gunakan nama BlogCard seperti pada kode di bawah ini</p> <p>14. Screenshot hasil dari pengerjaan ini.</p>		
E.	Github dan Viscode		
1.	<p>git init</p> <p>git add .</p> <p>git commit -m "first commit"</p> <p>git branch -M main</p> <p>git remote add origin https://github.com/annisafdl a/Praktikum2.git</p> <p>git push -u origin main</p>	 <p>The screenshot shows a terminal window with the following commands and output:</p> <pre>PS C:\Users\annis\praktikum2> git add . >> warning: in the working copy of 'src/app/page.tsx', LF will be replaced by CRLF the next time Git touches it PS C:\Users\annis\praktikum2> git commit -m "add project files and update page.tsx" >> [main d7140fb] add project files and update page.tsx 7 files changed, 377 insertions(+), 96 deletions(-) create mode 100644 src/app/about/components/Card.tsx create mode 100644 src/app/about/page.tsx create mode 100644 src/app/blog/[id]/[title]/components/BlogCard.tsx create mode 100644 src/app/blog/[id]/[title]/page.tsx create mode 100644 src/app/blog/[id]/page.tsx create mode 100644 src/app/components/Navbar.tsx PS C:\Users\annis\praktikum2> git push origin main >> Enumerating objects: 22, done. Counting objects: 100% (22/22), done. Delta compression using up to 4 threads Compressing objects: 100% (13/13), done. Writing objects: 100% (18/18), 4.21 KiB 615.00 KiB/s, done. Total 18 (delta 2), reused 0 (delta 0), pack-reused 0 remote: Resolving deltas: 100% (2/2), completed with 1 local object. To https://github.com/annisafdl a/Praktikum2.git 06c764b..d7140fb main -> main</pre> <p>Below the terminal output is a screenshot of the GitHub repository page for 'annisafdl a/Praktikum2.git'. It shows the commit history with the following entries:</p> <ul style="list-style-type: none"> annisafdl add project files and update page.tsx (d7140fb · 18 minutes ago) <ul style="list-style-type: none"> public: Initial commit from Create Next App (3 days ago) src/app: add project files and update page.tsx (18 minutes ago) .gitignore: Initial commit from Create Next App (3 days ago) README.md: Initial commit from Create Next App (3 days ago) eslint.config.mjs: Initial commit from Create Next App (3 days ago) next.config.ts: Initial commit from Create Next App (3 days ago) package-lock.json: Initial commit from Create Next App (3 days ago) package.json: Initial commit from Create Next App (3 days ago) postcss.config.mjs: Initial commit from Create Next App (3 days ago) tsconfig.json: Initial commit from Create Next App (3 days ago) <p>At the bottom of the repository page, there is a 'README' section and a 'Activate Windows' watermark.</p>	