

LAPORAN PRAKTIKUM PEMROGRAMAN BERBASIS KERANGKA KERJA

Modul 5 dan Modul 10



NAMA : M. Januar Rizkyanto

NIM : 1203210040

PRODI : Informatika

KELAS : IF 01-03

**PRODI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI DAN BISNIS
INSTITUT TEKNOLOGI TELKOM SURABAYA
SEMESTER GENAP 2023/2024**

Modul 5

1. App/about/page.js

```
export default function Page() {  
  return (  
    <>  
    <h1>Ini Halaman About</h1>  
    </>  
  )  
}
```

File ini digunakan sebagai halaman dalam aplikasi dengan melakukan export dan function page untuk mendeklarasikan dan mengespor program sebagai komponen page. Sebagai uji coba akan menampilkan konten teks 'ini halaman about' satu program di render.

2. App/dashboard/layout.js

```
const Layout = ({ children }) => {  
  return (  
    <>  
    <div style={{ height: 400, width: 600, background: "red" }}>  
    <h2>Layout Dashboard</h2>  
    {children}  
    </div>  
    </>  
  );  
};  
export default Layout;
```

file ini digunakan untuk membuat layout sebuah komponen elemen dalam halaman. Pada uji coba program ini membuat element layout dashboard dengan ukuran 400 x 600 dengan berwarna merah sebagai layoutnya. Setelah itu program akan diekspor agar dapat digunakan pada file komponen lain.

3. App/dashboard/user/layout.js

```
const Layout = ({ children }) => {  
  return (  
    <>  
    <div style={{ height: 300, width: 500, background: "blue" }}>  
    <h2>Layout User</h2>  
    {children}  
    </div>  
  )  
}
```

```

    </>
  );
};
export default Layout;

```

file ini digunakan untuk membuat layout sebuah komponen elemen dalam halaman. Pada uji coba program ini membuat element layout user sebagai komponen children didalam komponen sebelumnya dengan ukuran 300 x 500 dengan berwarna biru sebagai layoutnya. Setelah itu akan program akan diekspor agar dapat digunakan pada file komponen lain.

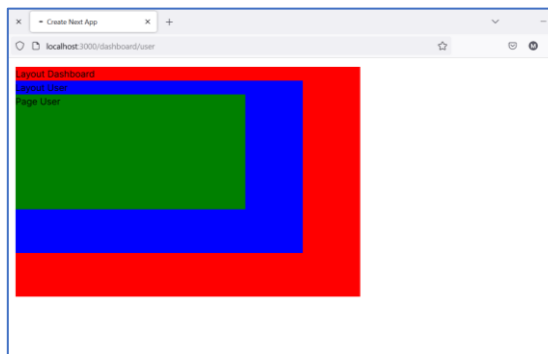
4. App/dashboard/user/page.js

```

const Page = () => {
  return (
    <>
    <div style={{ height: 200, width: 400, background: "green"
  }}>
    <h2>Page User</h2>
    </div>
    </>
  )
}
export default Page;

```

file ini digunakan untuk membuat layout sebuah komponen elemen dalam halaman. Pada uji coba program ini membuat element page user sebagai komponen didalam komponen sebelumnya dengan ukuran 200 x 400 dengan berwarna hijau sebagai layoutnya. Setelah itu akan program akan diekspor agar dapat digunakan pada file komponen lain.



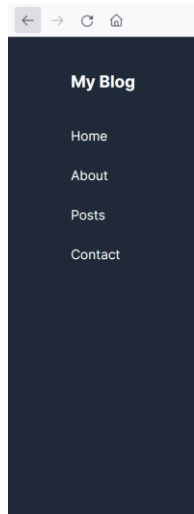
Hasil output diatas merupakan hasil run dari uji coba pembuatan layout dari ketiga file dimana terdapat perbedaan ukuran layout sesuai dengan ukuran dan warna yang telah ditentukan. Pada file ini terdapat inisialisasi program sebagai children dari program lain agar dapat memuat dan menampilkan elemen yang dimasukkan kedalam komponen yang digunakan.

5. Src/app/partials/sidebar.js

```
import Link from "next/link";
const menus = [
  {
    label: "Home",
    url: "/",
  },
  {
    label: "About",
    url: "/about",
  },
  {
    label: "Posts",
    url: "/posts",
  },
  {
    label: "Contact",
    url: "/contact",
  },
];
const Sidebar = () => {
  return (
    <div className="w-[18%] bg-gray-800 h-full flex justify-
center p-4 py-10">
      <ul>
        <li className="mb-10 text-white font-bold text-
xl">My Blog</li>
        {menus.map((menu, index) => (
          <li key={index} className="my-6 text-white">
            <Link href={menu.url}>{menu.label}</Link>
          </li>
        ))}
      </ul>
    </div>
  );
};
export default Sidebar;
```

pada kode diatas merupakan kode program yang digunakan untuk membuat komponen elemen side bar dalam halaman web dengan menggunakan tiling css . Komponen ini juga berfungsi sebagai navigasi item dengan url yang dihubungkan dengan menggunakan komponen link dari next.js sehingga ketika pengguna

memilih item pada side bar akan dihubungkan dengan url link yang telah ditentukan pada fungsi `conts` menu. Setelah itu akan program akan diekspor agar dapat digunakan pada file komponen lain.



Tampilan disamping merupakan hasil output dari pembuatan side bar. Pada element side bar terdapat beberapa komponen yaitu home, about, posts dna contact. Masing masing komponen ini sudah terhubung dnegan url yang dtelah ditentukan dan diatur dengan fungsi link yas ada pada inisialisasi `const` menu side bar.

6. `Src/app/page.js`

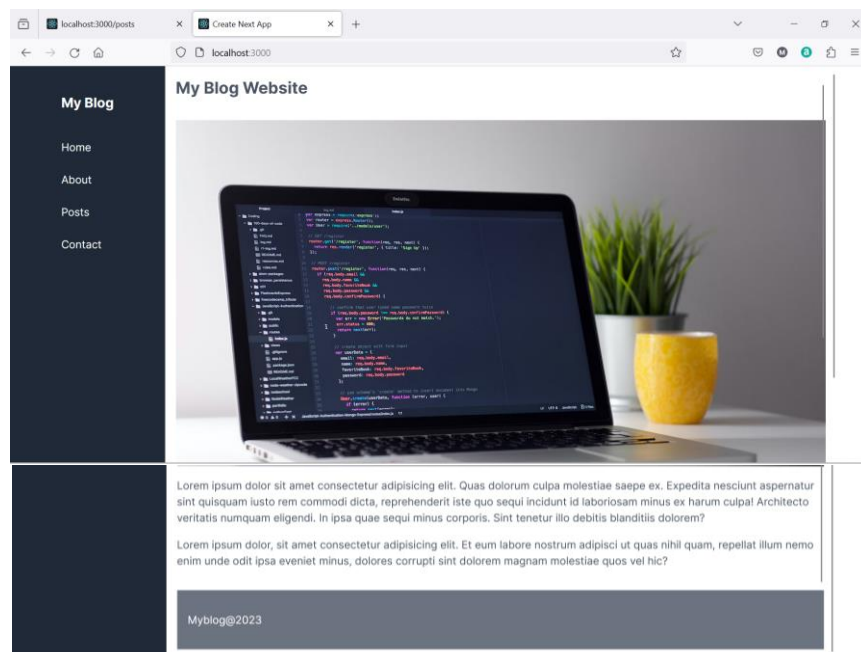
```
export default function Page() {
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">My Blog
Website</h1>
      <div className="py-4 text-gray-600">
        <div className="py-4">
          
        </div>
        <p>
          Lorem ipsum dolor sit amet consectetur adipisicing
elit. Quas dolorum
          culpa molestiae saepe ex. Expedita nesciunt aspernatur
sint quisquam
          iusto rem commodi dicta, reprehenderit iste quo sequi
incidunt id
          laboriosam minus ex harum culpa! Architecto veritatis
numquam
        </p>
      </div>
    </div>
  )
}
```

```

        eligendi. In ipsa quae sequi minus corporis. Sint
        tenetur illo debitis
        blanditiis dolorem?
    </p>
    <p className="py-4">
        Lorem ipsum dolor, sit amet consectetur adipisicing
        elit. Et eum labore nostrum adipisci ut quas nihil quam, repellat
        illum nemo enim unde odit ipsa eveniet minus, dolores corrupti
        sint dolorem magnam molestiae quos vel hic?
    </p>
</div>
</div>
);
}

```

Pada kode program file diatas berfungsi untuk membuat halaman dummy bernama page. Pada halaman in akan menampilkan judul dan konten teks dummy dari element <h1> dan <p>. pada halama in juga akan menampilkan gamabr yang akan digunakan pada halaman ini dari link url dari unsplash yang sudah mencakup parameter untuk ukuran gambar. Pada tampilan halaman ini mennggunakan komponen tailwind css untuk mengatur gara dan tata letak komponen untuk tampilan halaman.



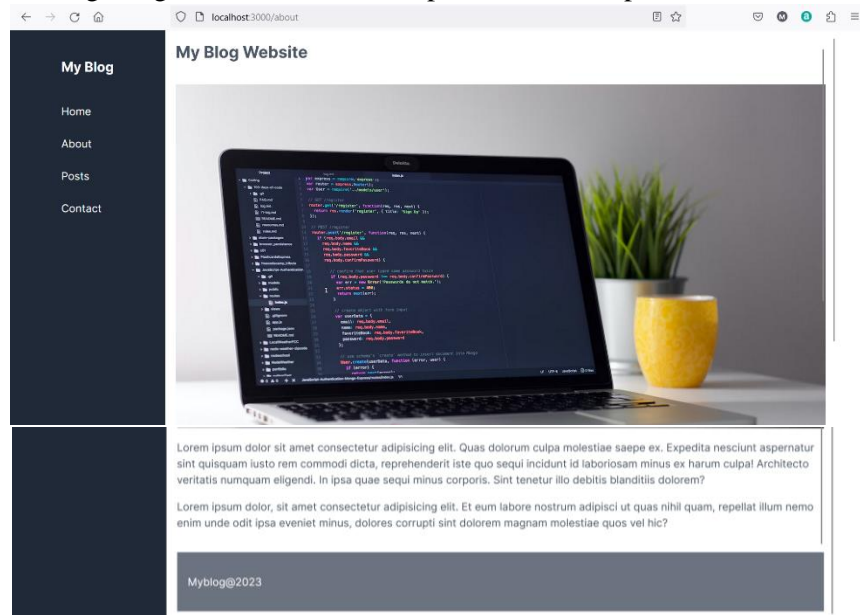
Hasil ouput diatas merupakan hasil run dari tampilan halaman dummy page.js dengan menampilkan judul halaman, komponen gambar dari link unsplash dan content dummy sesuai dengan isi content yang telah dittentukan.

7. Src/app/about/page.js

```
export default function Page() {
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">My Blog Website</h1>
      <div className="py-4 text-gray-600">
        <div className="py-4">
          
          </div>
          <p>
            Lorem ipsum dolor sit amet consectetur
            adipisicing elit. Quas dolorum
            culpa molestiae saepe ex. Expedita
            nesciunt aspernatur sint quisquam
            iusto rem commodi dicta, reprehenderit
            iste quo sequi incidunt id
            laboriosam minus ex harum culpa!
            Architecto veritatis numquam
            eligendi. In ipsa quae sequi minus
            corporis. Sint tenetur illo debitis
            blanditiis dolorem?
          </p>
          <p className="py-4">
            Lorem ipsum dolor, sit amet consectetur
            adipisicing elit. Et eum labore nostrum adipisci ut quas nihil
            quam, repellat illum nemo enim unde odit ipsa eveniet minus,
            dolores corrupti sint dolorem magnam molestiae quos vel hic?
          </p>
        </div>
      </div>
    </div>
  );
}
```

Pada kode program file diatas berfungsi untuk membuat halaman about. Pada halaman in akan menampilkan judul dan konten teks dummy dari element <h1> dan <p>. pada halama in juga akan menampilkan gamabr yang akan digunakan pada halaman ini dari link url dari unsplash yang sudah mencakup parameter untuk

ukuran gambar. Pada tampilan halaman ini menggunakan komonen tailwind css untuk mengatur gara dan tata letak komponen untuk tampilan halaman.



Hasil output halaman about.js

8. Src/app/posts/page.js

```
export const metadata = {
  title: "About Next",
  description: "Generated by create next app",
};

export default function Page() {
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">Posts</h1>
      <div className="py-4 text-gray-600">
        <p>
          Lorem ipsum dolor sit amet
          consectetur adipisicing elit. Quas dolorum
          culpa molestiae saepe ex. Expedita
          nesciunt aspernatur sint quisquam
          iusto rem commodi dicta,
          reprehenderit iste quo sequi incidunt id
          laboriosam minus ex harum culpa!
          Architecto veritatis numquam
        </p>
      </div>
    </div>
  );
}
```



```

        eligendi. In ipsa quae sequi minus
corporis. Sint tenetur illo debitis
        blanditiis dolorem?
    </p>
  </div>
</div>
);
}

```

Pada kode program file diatas berfungsi untuk membuat halaman posts. Pada halaman in akan menampilkan element content berupa teks informasi mengenai program dari halaman posts ketika pengguna memilih element post.

9. Src/app/contact/page.js

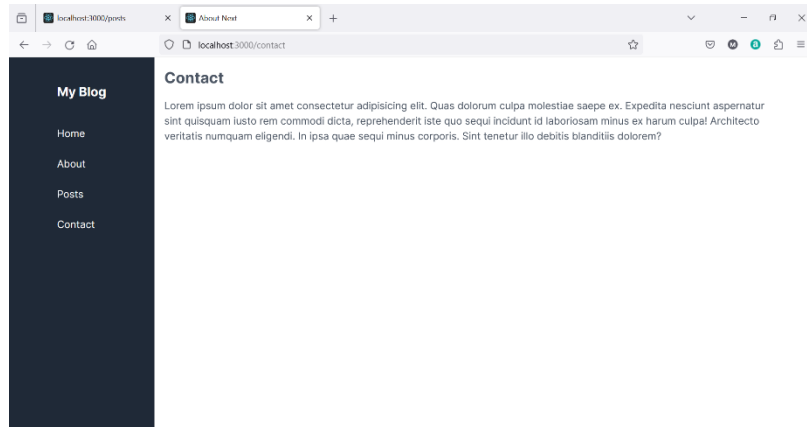
```

export const metadata = {
  title: "About Next",
  description: "Generated by create next app",
};
export default function Page() {
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">Contact</h1>
      <div className="py-4 text-gray-600">
        <p>
          Lorem ipsum dolor sit amet
consectetur adipisicing elit. Quas dolorum
          culpa molestiae saepe ex. Expedita
nesciunt aspernatur sint quisquam
          iusto rem commodi dicta,
reprehenderit iste quo sequi incidunt id
          laboriosam minus ex harum culpa!
Architecto veritatis numquam
          eligendi. In ipsa quae sequi minus
corporis. Sint tenetur illo debitis
          blanditiis dolorem?
        </p>
      </div>
    </div>
  );
}

```

Pada kode program file diatas berfungsi untuk membuat halaman contact. Pada halaman in akan menampilkan element content berupa teks informasi mengenai

program dari halaman posts ketika pengguna memilih element contact.



Hasil ouput diatas merupakan hasil run dari tampilan halaman contact yang menampilkan contenct informasi contact dan judul halaman ketika pengguna memilih komponen contact.

Modul 10

1. Src/app/posts/slug/page.js

```
export default function Page({ params }) {
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">Posts Detail</h1>
      <div className="py-4 text-gray-600">
        {params.slug}
      </div>
    </div>
  );
}
```

Pada kode program diatas digunakan untuk membuat komponen fungsi yaitu page dengan menggunakan komponen params untuk menerima dan menampilkan informasi.

2. Install library axios

```
PS C:\PBKK\week5\prak5> npm install axios

up to date, audited 333 packages in 3s

117 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
PS C:\PBKK\week5\prak5> █
```

Pada file ini juga akan menginstall library tambahan javascript yaitu axios untuk membuat permintaan http pada browser agar dapat melakukan hubungan untuk mengambil dan mengirim data.

3. Src/app/posts/page.js

```
import axios from "axios";
import Link from "next/link";
export default async function Page() {
  const { data } = await
  axios.get("https://raw.githubusercontent.com/dauditts/pbkk-static-api/main/articles.json");
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">Posts</h1>
```

```

        <div className="py-4 text-gray-600">
          {data.map((post) => (
            <>
              <div className="p-4 my-3">
                <Link href={` /posts/${post.id}`}>
                  <b className="hover:text-
blue-400">{post.title}</b>
                </Link>
                <p>{post.description}</p>
              </div>
            </>
          ))}
        </div>
      </div>
    );
  }

```

Pada kode program diatas pada file page.js digunakan untuk membuat dan mengambil nilai yang diambil dari link API yang telah ditentukan. Pada file ini akan mengimpor komponen react, axios dan link untuk melakukan navigasi dengan sisi klien. Setelah itu akan melakukan pengambilan data dnegan menggunakan fungsi fetchdata dan axios.get untk meminta dan mengambil data dari url api. Setelah itu akan merender posting dan pengambaran komponen utama.

4. Src/app/posts/slug/loading.js

```

const Loading = () => {
  return (
    <>
      <h2>Ini Loading</h2>
    </>
  )
}
export default Loading;

```

pada kode program diatas yaitu untuk membuat dan menampilkan pesan informasi pesan “ini loading” ketika program sedang dalam proses.

5. Src/app/posts/slug/page.js

```

export default async function Page({ params }) {
  const { data } = await new Promise(async (resolve) => {
    setTimeout(() => {
      resolve('Ini Data dari menunggu');
    }, 3000);
  });
}

```

```

    return (
      <div>
        <h1 className="font-bold text-2xl text-gray-600">Posts Detail</h1>
        <div className="py-4 text-gray-600">
          {params.slug}
        </div>
      </div>
    );
  }
}

```

Pada file ini memiliki fungsi untuk memberi pesan ketika di dalam halaman sedang mengambil data dari api, maka loading bahwa akan di load atau proses sampai request complete. Pesan akan muncul setelah 3 detik proses dimulai.

6. Src/app/posts/slug/not-found.js

```

export default function NotFound() {
  return (
    <div>
      <h2>Ini 404! Post</h2>
    </div>
  );
}

```

Pada file ini memiliki fungsi untuk memberi pesan 404 not found sebagai respond kepada pengguna bahwa halaman tidak ditemukan dan akna meretur elementnya.

7. Src/app/posts/slug/page.js

```

const { data } = await getArticleDetail(params.slug);

if (!data) {
  notFound();
}

```

Lalu pada file page.js ditambahkan fungsi untuk memanggil fungsi not found ketika data tidak ditemukan agar dapat memberikan pesan error sebagai respons kepada pengguna.

8. Src/middleware.js

```
import { NextResponse } from "next/server";
export function middleware(request) {
  let checkAuth = false;

  const checkToken = request.cookies.get("accessToken");
  if (checkToken) {
    checkAuth = true;
  }
  if (checkAuth) {
    return NextResponse.next();
  } else {
    const loginUrl = new URL("/login", request.url);
    return NextResponse.redirect(loginUrl);
  }
}
export const config = {
  matcher: "/posts",
};
```

Pada file kode program diatas berfungsi untuk melakukan pengecekan dan otentikasi dengan menggunakna middleware. Pengecekan ini bedasarkan token apakah ditemukan atau tidak, jika berhasil amka akan bernilai true dan akan mengembalikan nilai nextresponst, jika tidak maka akan dikembalikan ke halaman login.

9. App/api/auth/login/route.js

```
import { cookies } from "next/headers";
import { NextResponse } from "next/server";
export async function POST(request) {
  const data = await request.json();
  if (data) {
    if (data.email == "abc@gmail.com") {
      cookies().set("accessToken", "mytoken");
      return await NextResponse.json({ success: "Login
success" });
    }
  }
  return await NextResponse.json(
    { error: "User & password not match" },
    { status: 400 }
  );
}
```

Pada file diatas berfungsi senagai handler dalam menangani permintaann post ketika proses pengecekank atau otentikasi pengguna dan kata sandi inputan pengguna. Kode ini sebagai pengecekan dari kode sebelumnya bahwa inputan pengguna harus sesuai dengan kondisi pengecekan.

10. App/auth/login/page.js

```
"use client";
import axios from "axios";
import { useState } from "react";
export default function Page() {
  const [form, setForm] = useState({ email: "", password: ""
});
  const handleSubmit = (f) => {
    f.preventDefault();
    axios
      .post("http://localhost:3000/api/auth/login", form, {
        headers: {
          "Content-Type": "application/json",
        },
      })
      .then((res) => {
        console.log("res", res);
        document.location = "/posts";
      })
      .catch((err) => {
        console.log("err", err.response);
      });
  };
  return (
    <>
    <h2 className="font-bold text-3xl my-6">Login</h2>
    <div className="p-6 border-2 max-w-2xl">
      <form
        className="bg-white shadow-md rounded px-8
pt-6 pb-8 mb-4"
        onSubmit={handleSubmit}
      >
        <div className="mb-4">
          <label
            className="block text-gray-700 text-
sm font-bold mb-2"
            htmlFor="username"
```



```

    });
  }
  />
</div>
<div className="flex items-center justify-
between">
    <button className="bg-blue-500 hover:bg-
blue-700 text-white font-bold py-2 px-4 rounded focus:outline-
none focus:shadow-outline">
        Sign In
    </button>
</div>
</form>
</div>
</>
);
}

```

Pada kode program diatas berfungsi untuk mengirim permintaan post dengan menggunakan api login yaitu localhost 3000. Pada kode ini akan membuat state form untuk menyimpan value nilai inputan email dan password dari pengguna, setelah itu akan memanggil formular ketika pengguna menekan tombol handlesubmit. Setelah itu kod eakan mengirim permintaan post dengan kondisi sukses dan gagal sata proses otentikasi.

11. Src/utils/api.js

```

import axios from "axios";
const api = axios.create({
  baseURL: "https://raw.githubusercontent.com/dauditts",
});
api.interceptors.request.use(
  function (config) {
    // Do something before request is sent
    return config;
  },
  function (error) {
    // Do something with request error
    return Promise.reject(error);
  }
);
api.interceptors.response.use(
  function (response) {

```

```

        // Any status code that lie within the range of 2xx cause
this function to trigger
        // Do something with response data
        return response;
    },
    function (error) {
        // Any status codes that falls outside the range of 2xx
cause this function to trigger
        // Do something with response error
        return Promise.reject(error);
    }
);
export { api };

```

Kode program diatas berfungsi untuk melakukan dan membuat konfigurasi komponen axios sebagai const dengan menggunakan komponen axios untuk dapat melakukan permintaan dan merespons. Dalam file ini membuat instance axios dengan menghubungkan url untuk dapat melakukan permintaan. Setelah itu akan menggunakan request.use untuk merespon permintaan sebelum dikirim. Komponen axios akan membuat kondisi gagal dan sukses dalam melakukan respon permintaan. Dalam merespon menggunakan request use untuk melakukan respon permintaan sebelum dikirim, setelah itu akan diserahkan ke variable then atau catch untuk merespon kesalahan yang terjadi. Setelah itu akan diekspor agar dapat digunakan di file lain.

12. Src/utls/services/post.services.js

```

import { api } from '@/utls/api';
export const getArticle = async () => {
    // get all article
    return await api.get('/pbkk-static-api/main/articles.json', {
        transformResponse: (res) => {
            const resData = JSON.parse(res);
            return resData;
        }
    });
}

// get article by id
export const getArticleDetail = async (id) => {
    return await api.get('/pbkk-static-api/main/articles.json', {
        transformResponse: (res) => {
            const resData = JSON.parse(res);
            const article = resData.filter(item => {
                return item.id === id
            });
        }
    });
}

```

```

    })
    return article[0] ?? {};
  }
});
}

```

Pada kode program diatas berfungsi untuk program mengambil data dengan menggunakan komponen axios. Pada file ini juga menggunakan fungsi transformresponse untuk dapat mengubah format respons menjadi javascript. Proses pengambilan data akan dilakukan dengan melakukan pengecekan berdasarkan id yang ada.

13. app/posts/page.js

```

import axios from "axios";
import Link from "next/link";
import { getArticle } from "@/utils/services/post.service";
export default async function Page() {
  // const { data } = await
  axios.get("https://raw.githubusercontent.com/dauditts/pbkk-
  static-api/main/articles.json");
  const { data } = await getArticle();
  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-
  600">Posts</h1>
      <div className="py-4 text-gray-600">
        {data.map((post) => (
          <>
            <div className="p-4 my-3">
              <Link href={`/${post.id}`}>
                <b className="hover:text-blue-
  400">{post.title}</b>
              </Link>
              <p>{post.description}</p>
            </div>
          </>
        ))}
      </div>
    </div>
  );
}

```

Kode program diatas berfungsi untuk mengambil dan menampilkan daftar dari nilai api yang ditentukan dengan menggunakan fungsi getarticle yang ada pada file post.service. fungsi getarticle digunakan untuk melakukan permintaan kepada api untuk mendapat data dan nilai agar dapat dirender ke dalam program dna komponen.

14. app/posts/slug/page.js};

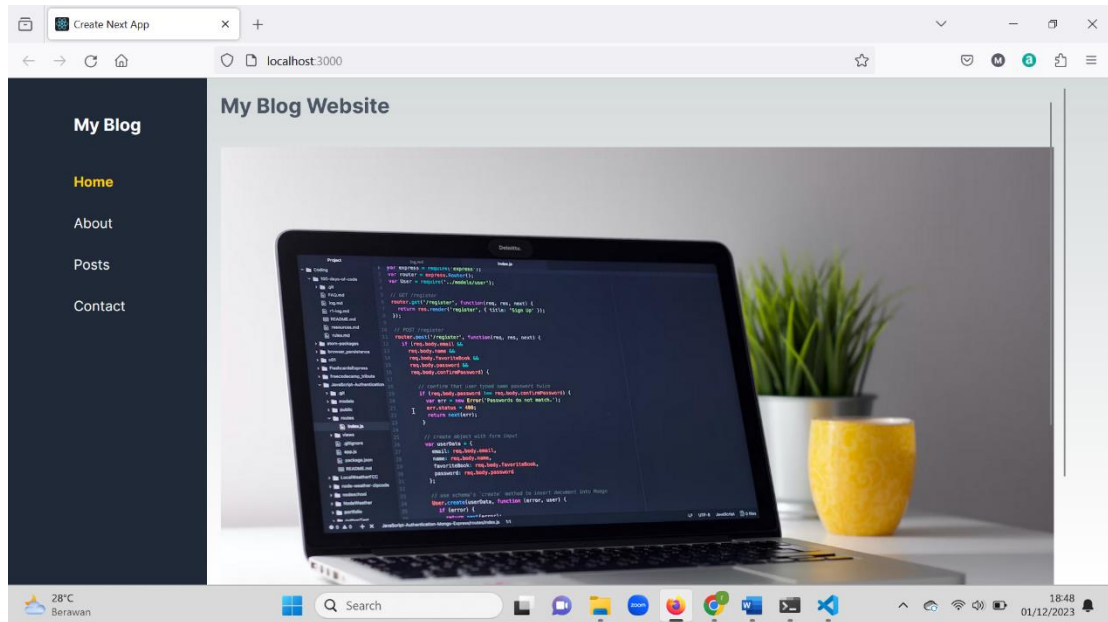
```
import { notFound } from "next/navigation";
import { getArticleDetail } from "@utils/services/post.service";
export default async function Page({ params }) {
  // const { data } = await new Promise(async (resolve) => {
  //   setTimeout(() => {
  //     // resolve('Ini Data dari menunggu');
  //     resolve(false);
  //   }, 3000);
  // });

  const { data } = await getArticleDetail(params.slug);

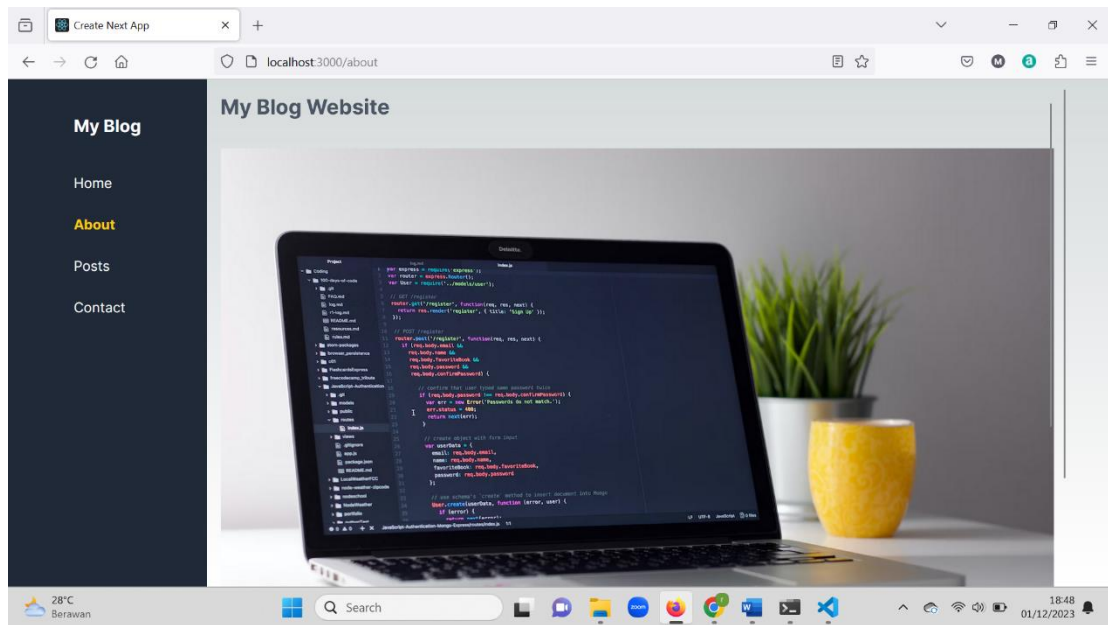
  if (!data) {
    notFound();
  }

  return (
    <div>
      <h1 className="font-bold text-2xl text-gray-600">
        Posts - {data.title}
      </h1>
      <div className="py-4 text-gray-600">
        <p>{data.content}</p>
      </div>
    </div>
  );
}
```

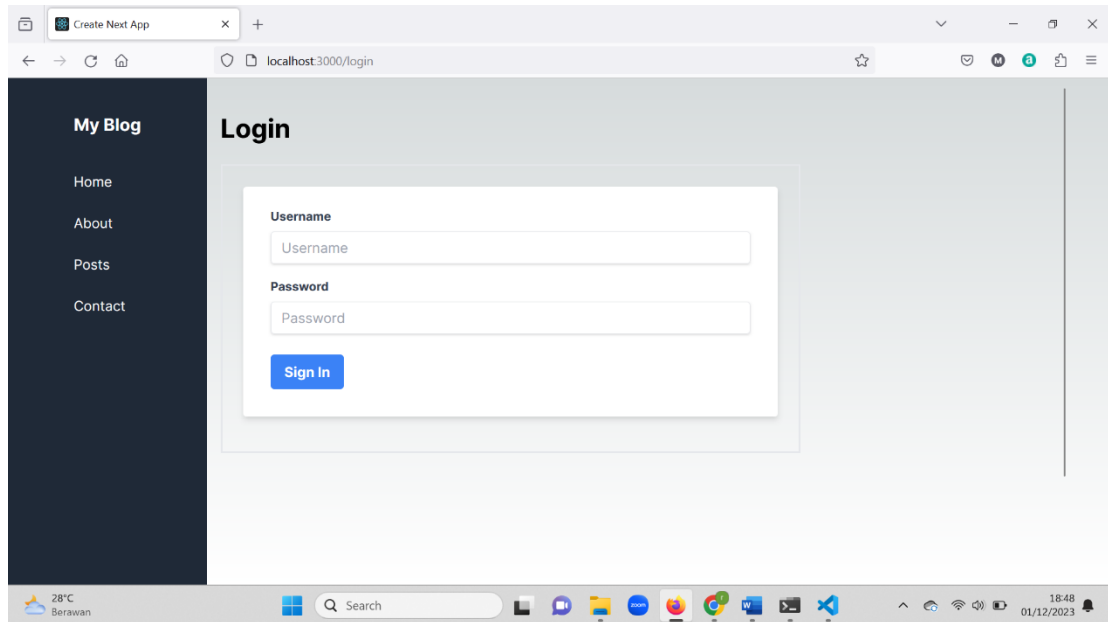
Kode program diatas berfungsi untuk menangani kondisi ketika terdapat proses pencarian data yang tidla ditemukan sehingga muncul pesan 404 not found. Dalam file ini akan menangani dnegan cara mengisialisasi bahwa konfigurasi route kedalam file program, setelah itu akan mengimport fungsi not found ketika data tidak ditemukan. Pada file ini akan memanggil fungsi getarticel yang ada pada file post.service untuk mendapatkan artikel, dalam proses ini terjadi 2 kondisi ketika gagal dan berhasil. Ketika gagal maka akan memanggil fungsi notfound untuk mendapatkan pesan 4040 not found sedangkan jika berhasil maka akan merender detail data dengan menampilkan judul dan konten.



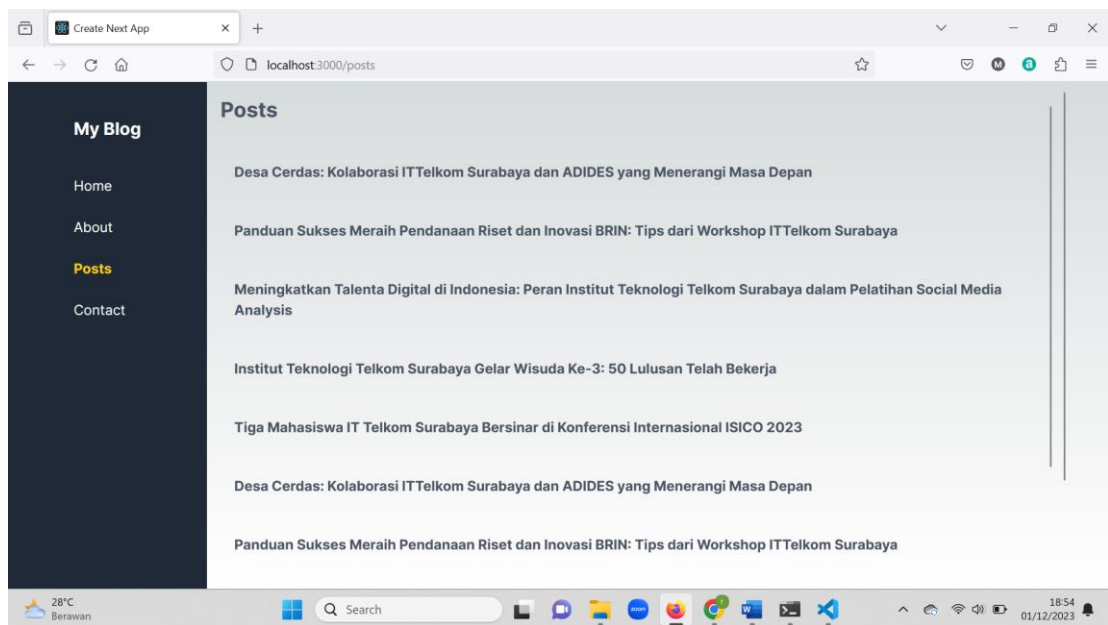
Tampilan halaman home



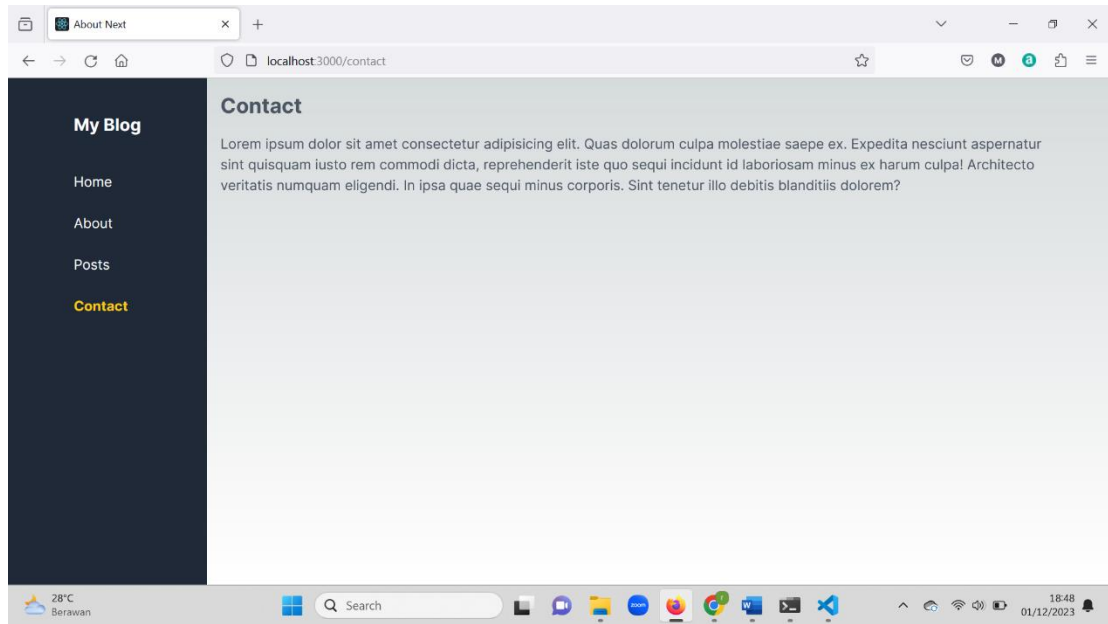
Tampilan halaman about



Tampilan awal halaman posts



Tampilan halaman posts ketika pengguna berhasil login



Tampilan halaman contact