

PRAKTIK PENGEMBANGAN FRONT END

23 Oktober 2024

PENGUNAAN API

a. Pengertian Api

Apa itu API?

API, atau antarmuka pemrograman aplikasi, adalah serangkaian aturan atau protokol yang memungkinkan aplikasi perangkat lunak berkomunikasi satu sama lain untuk bertukar data, fitur, dan fungsionalitas.

API menyederhanakan dan mempercepat pengembangan aplikasi dan perangkat lunak dengan memungkinkan pengembang untuk mengintegrasikan data, layanan, dan kapabilitas dari aplikasi lain, alih-alih mengembangkannya dari awal. API juga memberi pemilik aplikasi cara yang mudah dan aman untuk menyediakan data dan fungsi aplikasi mereka ke berbagai departemen dalam organisasi mereka. Pemilik aplikasi juga dapat berbagi atau memasarkan data dan fungsi ke mitra bisnis atau pihak ketiga.

API memungkinkan pembagian informasi yang diperlukan saja, dengan tetap merahasiakan detail sistem internal lainnya, yang membantu keamanan sistem. Server atau perangkat tidak harus mengekspos data sepenuhnya—API memungkinkan pembagian paket data kecil, yang relevan dengan permintaan tertentu.

Dokumentasi API seperti buku petunjuk teknis yang menyediakan detail tentang API dan informasi bagi pengembang tentang cara bekerja dengan API dan layanannya. Dokumentasi yang dirancang dengan baik akan memberikan pengalaman API yang lebih baik bagi pengguna dan secara umum akan menghasilkan API yang lebih sukses.

b. Cara Kerja Api

Berguna untuk memikirkan komunikasi API dalam hal permintaan dan respons antara klien dan server. Aplikasi yang mengirimkan permintaan adalah klien, dan server memberikan respons. API adalah jembatan yang membangun koneksi di antara keduanya.

Cara sederhana untuk memahami cara kerja API adalah dengan melihat contoh umum—pemrosesan pembayaran pihak ketiga. Saat pengguna membeli produk di situs [e-commerce](#), situs tersebut mungkin meminta pengguna untuk “Bayar dengan PayPal” atau jenis sistem pihak ketiga lainnya. Fungsi ini bergantung pada API untuk membuat koneksi.

- Saat pembeli mengklik tombol pembayaran, panggilan API dikirim untuk mengambil informasi. Ini adalah permintaan. Permintaan ini diproses dari aplikasi ke server web melalui Uniform Resource Identifier (URI) API dan mencakup kata kerja permintaan, tajuk, dan terkadang, isi permintaan.
- Setelah menerima permintaan yang valid dari halaman web produk, API memanggil program eksternal atau server web, dalam hal ini, sistem pembayaran pihak ketiga.

- Server mengirimkan respons ke API dengan informasi yang diminta.
- API mentransfer data ke aplikasi peminta awal, dalam hal ini, situs web produk.

Meskipun transfer data berbeda-beda, tergantung pada layanan web yang digunakan, permintaan dan respons semuanya terjadi melalui API. Tidak ada visibilitas pada antarmuka pengguna, artinya API bertukar data di dalam komputer atau aplikasi, dan tampak bagi pengguna sebagai koneksi yang lancar.

c. Jenis-Jenis Api

Jenis-jenis API

API dapat dikategorikan berdasarkan kasus penggunaan, termasuk API data, API sistem operasi, API jarak jauh, dan API web.

1) API data (atau basis data)

Digunakan untuk menghubungkan aplikasi dan sistem manajemen basis data.

2) API sistem operasi (lokal)

Digunakan untuk menentukan bagaimana aplikasi menggunakan layanan dan sumber daya sistem operasi.

3) API Jarak Jauh

Digunakan untuk menentukan bagaimana aplikasi pada perangkat yang berbeda berinteraksi.

4) API Web

Digunakan untuk mengaktifkan transfer data dan fungsionalitas melalui internet menggunakan protokol HTTP.

Saat ini, sebagian besar API adalah API web. API web adalah jenis API jarak jauh (artinya API menggunakan protokol untuk memanipulasi sumber daya eksternal) yang memaparkan data dan fungsionalitas aplikasi melalui internet.

4 jenis utama API web adalah:

5) API Terbuka

Open API adalah antarmuka pemrograman aplikasi sumber terbuka yang dapat Anda akses dengan protokol HTTP. Dikenal juga sebagai API publik, API ini memiliki titik akhir API dan format permintaan dan respons yang telah ditentukan.

6) API Mitra

API mitra menghubungkan mitra bisnis strategis. Biasanya, pengembang mengakses API ini dalam mode layanan mandiri melalui portal pengembang API publik. Namun, mereka perlu menyelesaikan proses pendaftaran dan mendapatkan kredensial login untuk mengakses API mitra.

7) API Internal

API internal, atau privat, tetap tersembunyi dari pengguna eksternal. API privat ini tidak tersedia bagi pengguna di luar perusahaan. Sebaliknya, organisasi menggunakannya untuk meningkatkan produktivitas dan komunikasi di berbagai tim pengembangan internal.

8) API Komposit

API Komposit menggabungkan beberapa API data atau layanan. API ini memungkinkan programmer untuk mengakses beberapa titik akhir dalam satu panggilan. API Komposit berguna dalam

arsitektur layanan mikro di mana menjalankan satu tugas mungkin memerlukan informasi dari beberapa sumber.

d. Manfaat

- 1) **Integrasi Aplikasi:** Menghubungkan aplikasi yang berbeda untuk berbagi data dan fitur.
- 2) **Menghemat Waktu dan Biaya Pengembangan:** Mengurangi pengembangan dari awal dengan menggunakan fitur yang sudah tersedia.
- 3) **Membuat Aplikasi Dinamis:** Menyediakan data real-time dan layanan berkelanjutan.
- 4) **Ekosistem yang Terbuka:** Mendukung pengembang pihak ketiga dalam menciptakan fitur tambahan.
- 5) **Keamanan dan Abstraksi:** Mengamankan data dengan abstraksi dan otentikasi.
- 6) **Fleksibilitas:** Mempermudah pengembangan lintas platform dan menciptakan modularitas.
- 7) **Meningkatkan Pengalaman Pengguna:** Menyediakan pengalaman terintegrasi, misalnya login dengan media sosial.
- 8) **Mendukung Ekonomi Digital:** API bisa digunakan untuk monetisasi dan kolaborasi bisnis.

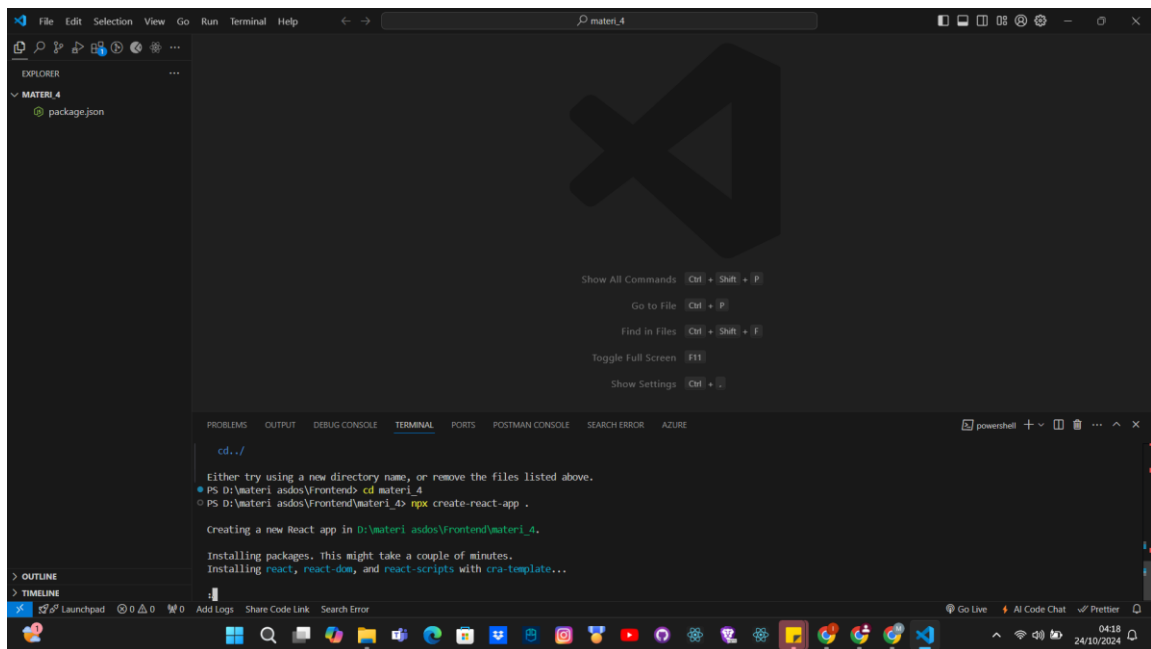
e. Contoh Penggunaan

1) Api Wilayah

a) Buatlah Folder dan installlah reactnya di folder tersebut

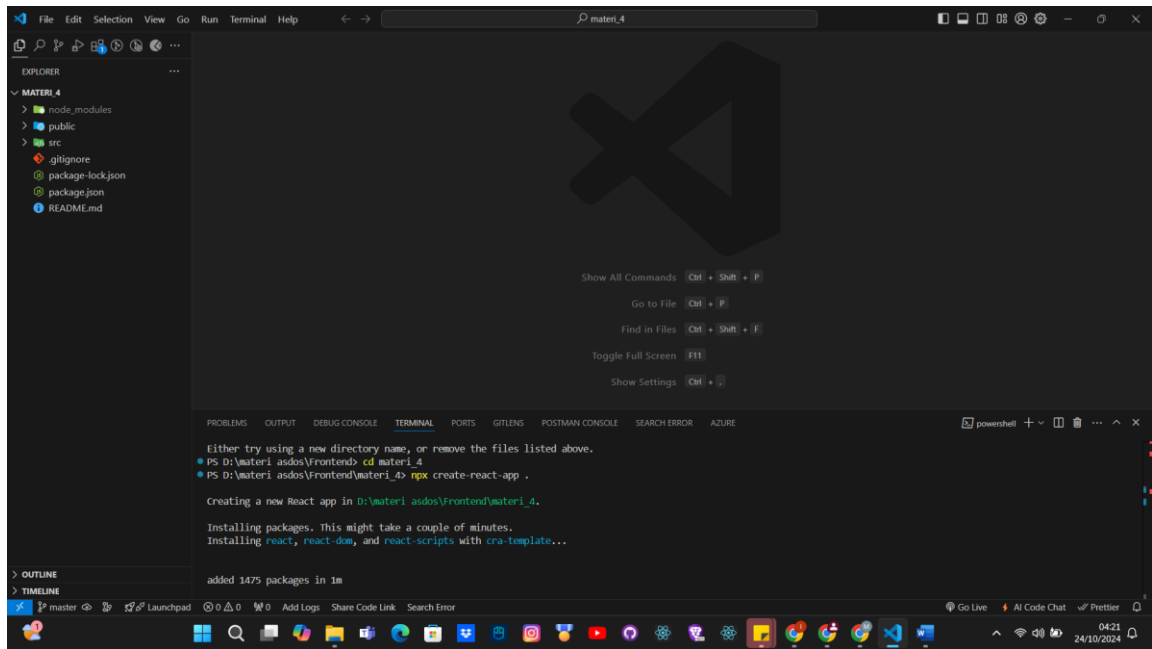
“ npx create-react-app . “ (untuk buat di folder tersebut) /

“ npx create-react-app my-app “ (untuk buat di folder baru dengan nama my-app)

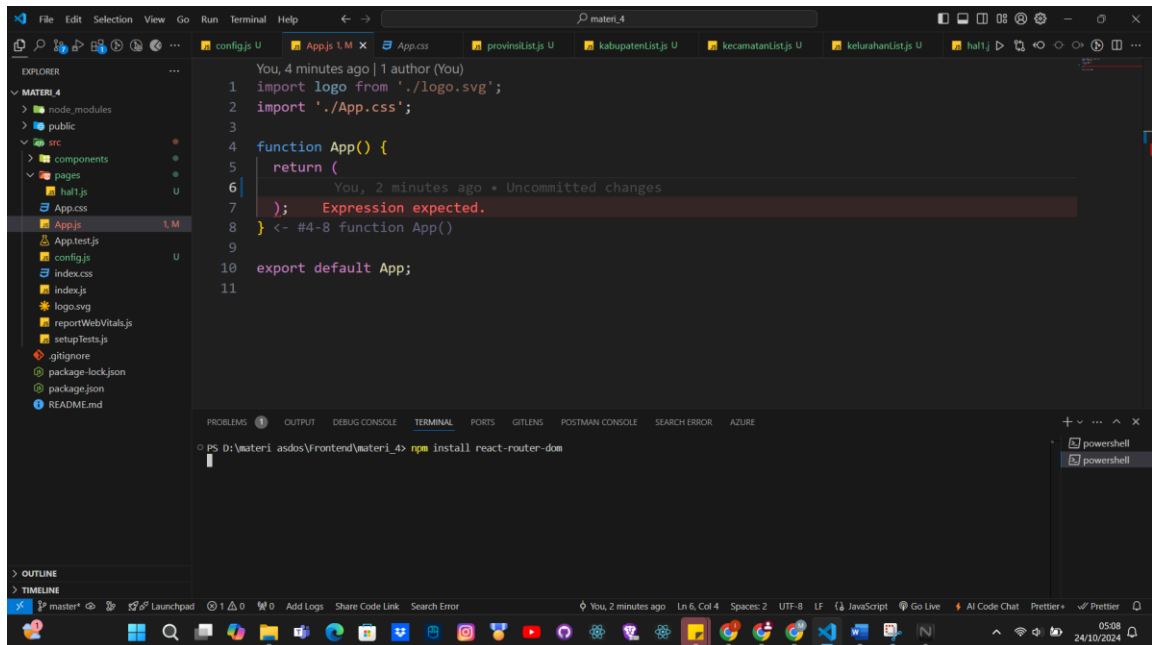


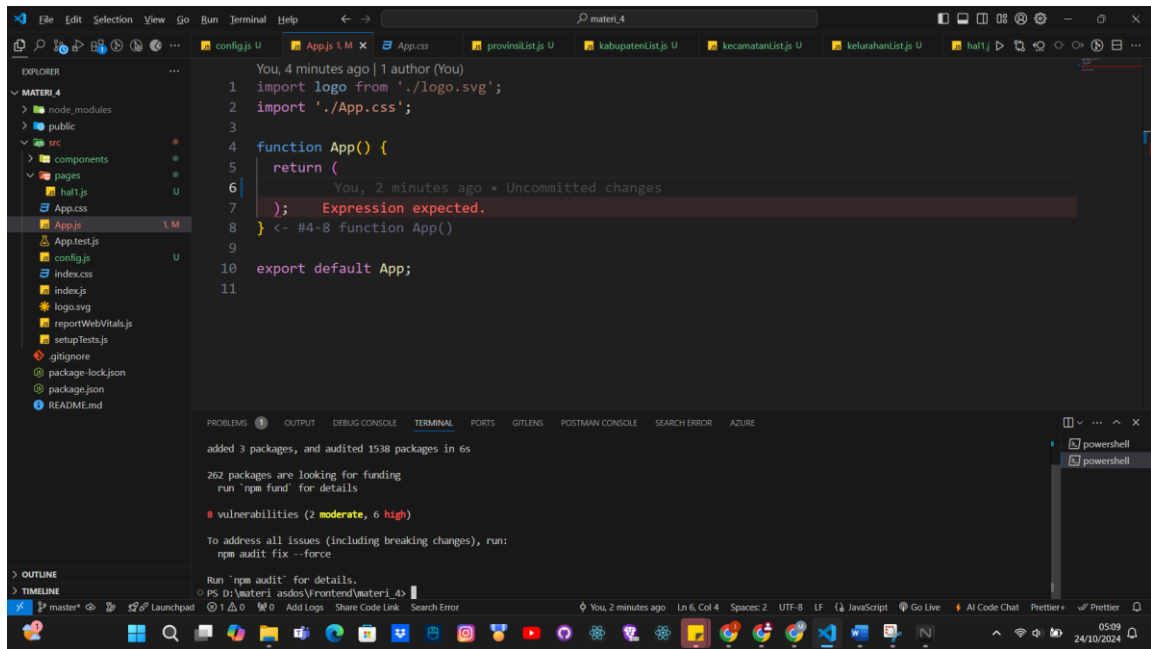
The screenshot shows the Visual Studio Code interface with a terminal window open at the bottom. The terminal is running a PowerShell session. The user has navigated to a directory and executed the command `npx create-react-app .`. The output shows the creation of a new React app and the installation of packages. The Explorer panel on the left shows a folder named 'materi_4' containing a 'package.json' file. The terminal output is as follows:

```
cd ../  
  
Either try using a new directory name, or remove the files listed above.  
PS D:\materi asdos\Frontend> cd materi_4  
PS D:\materi asdos\Frontend\materi_4> npx create-react-app .  
  
Creating a new React app in D:\materi asdos\Frontend\materi_4.  
  
Installing packages. This might take a couple of minutes.  
Installing react, react-dom, and react-scripts with cra-template...
```

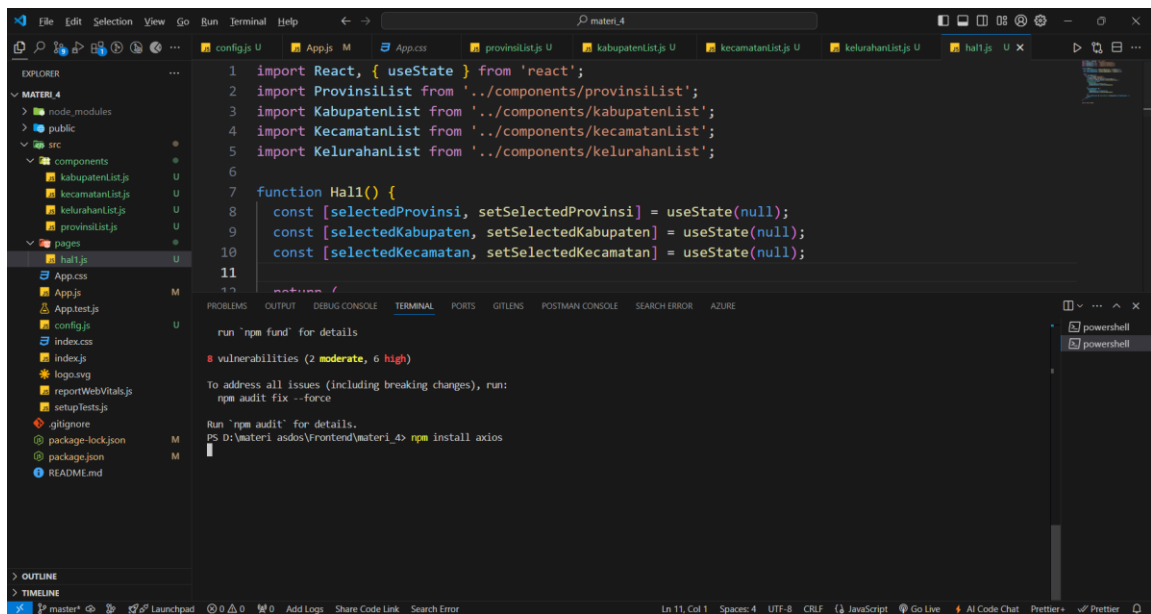


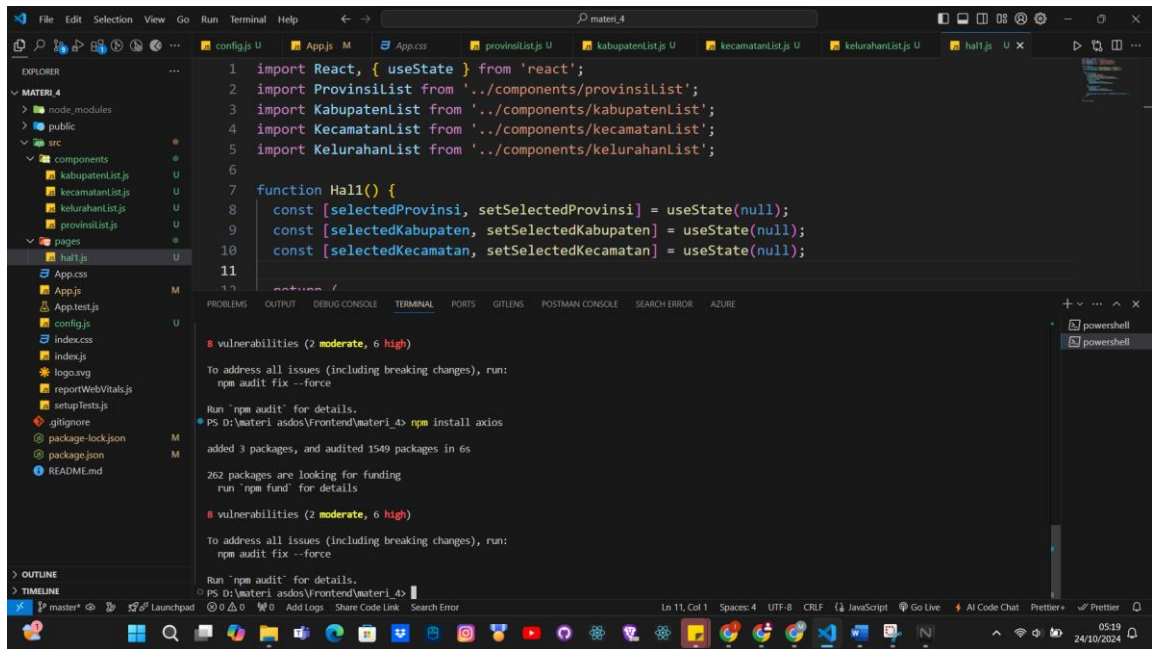
- b) Kemudian jangan lupa untuk install react-router-dom nya
“ npm install react-router-dom “



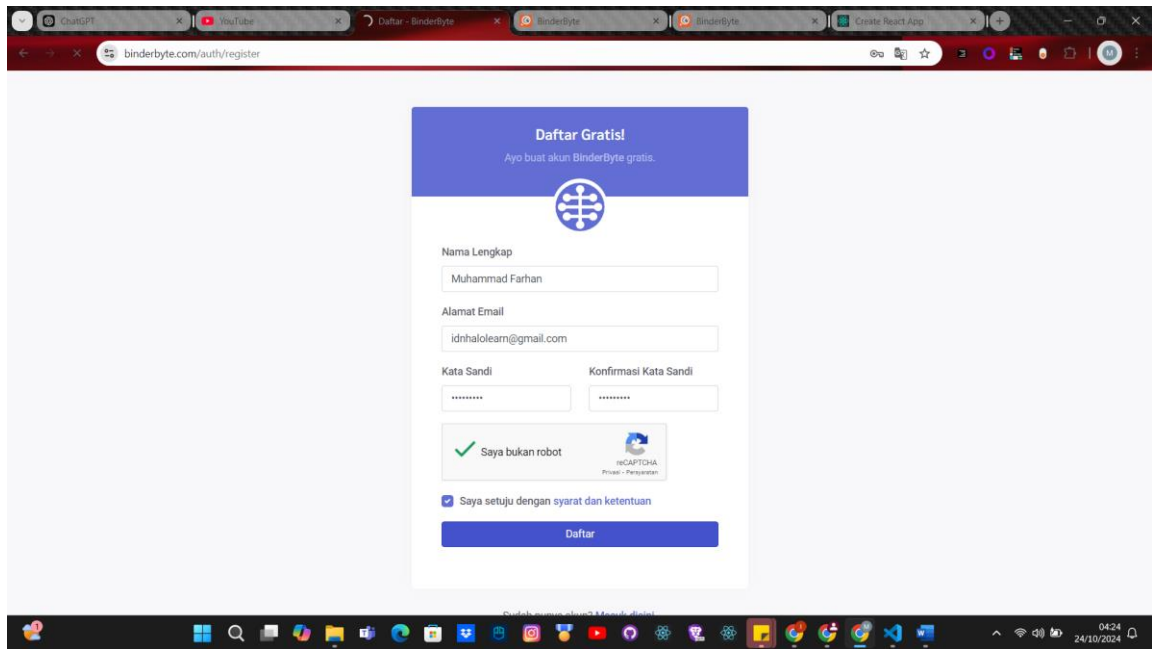


c) Kemudian jangan lupa untuk install axios juga
“npm install axios”

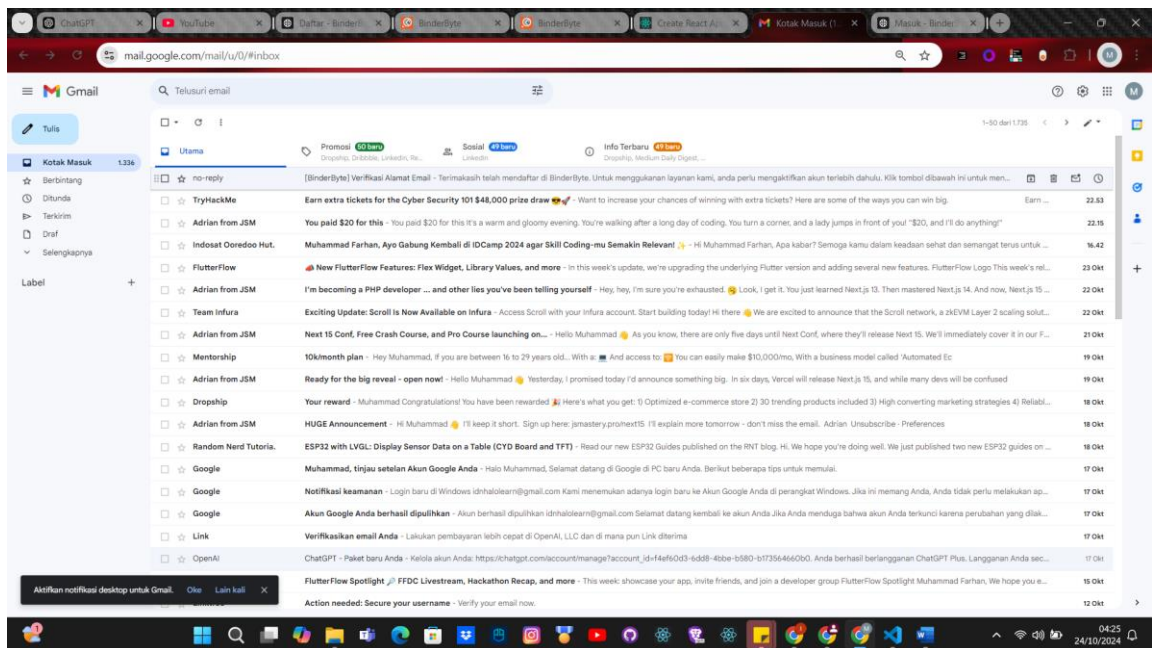




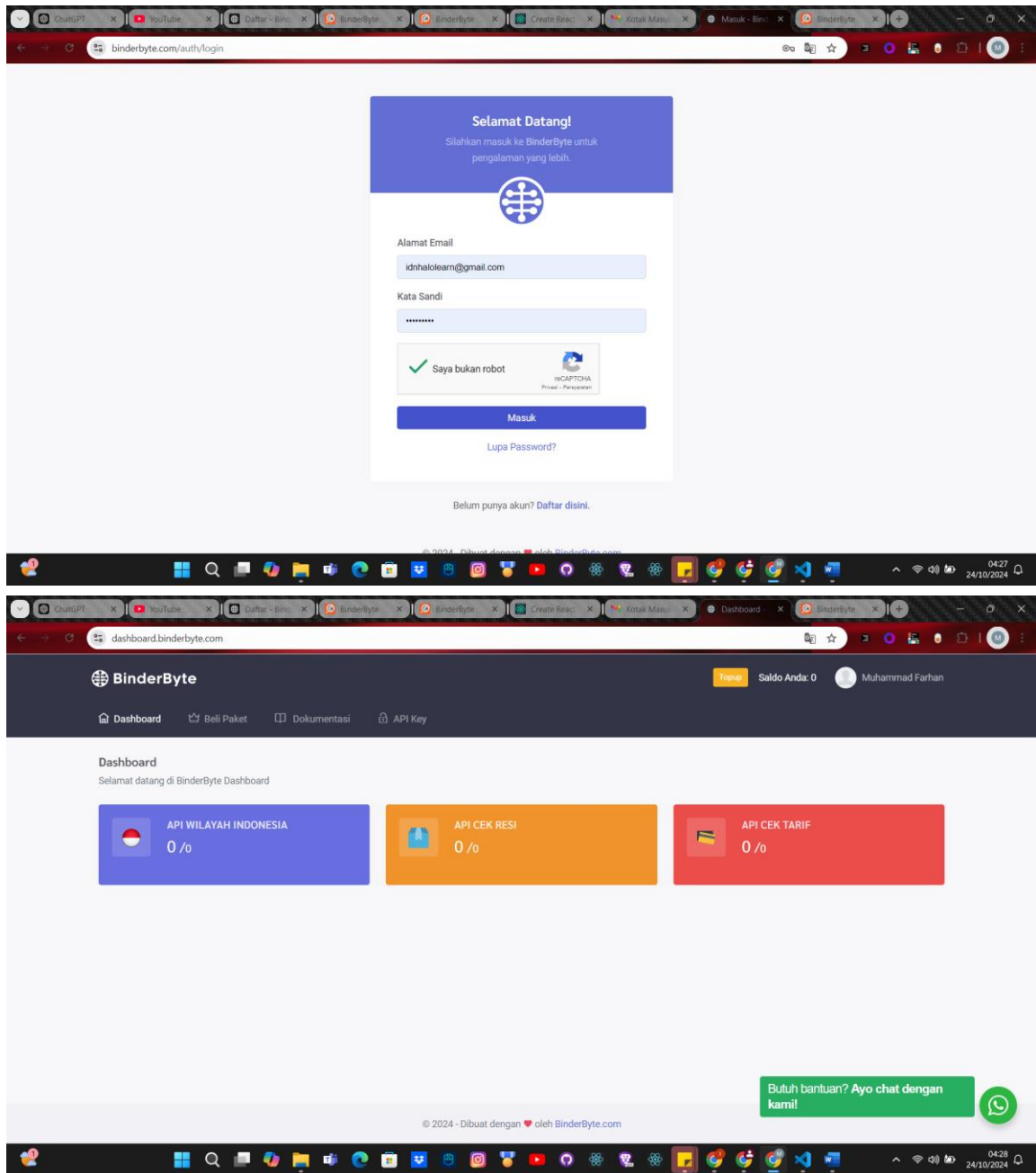
- d) Kemudian click web ini <https://docs.binderbyte.com/api/wilayah>
Kemudian daftar akun kalian



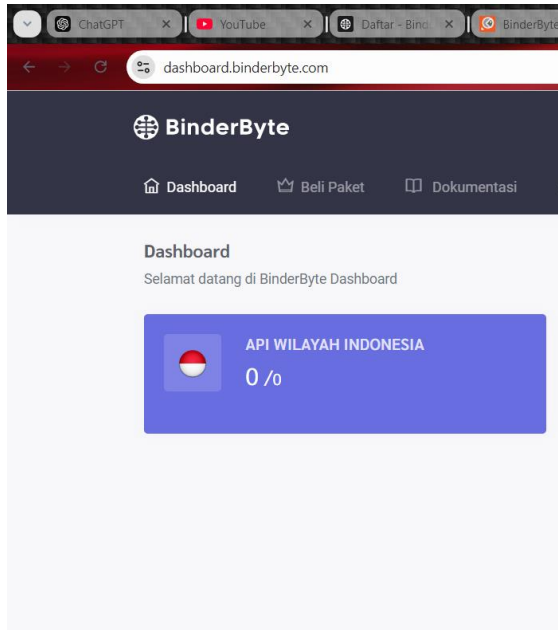
e) Setelah itu cek email kalian kemudian verifikasi akun kalian disana



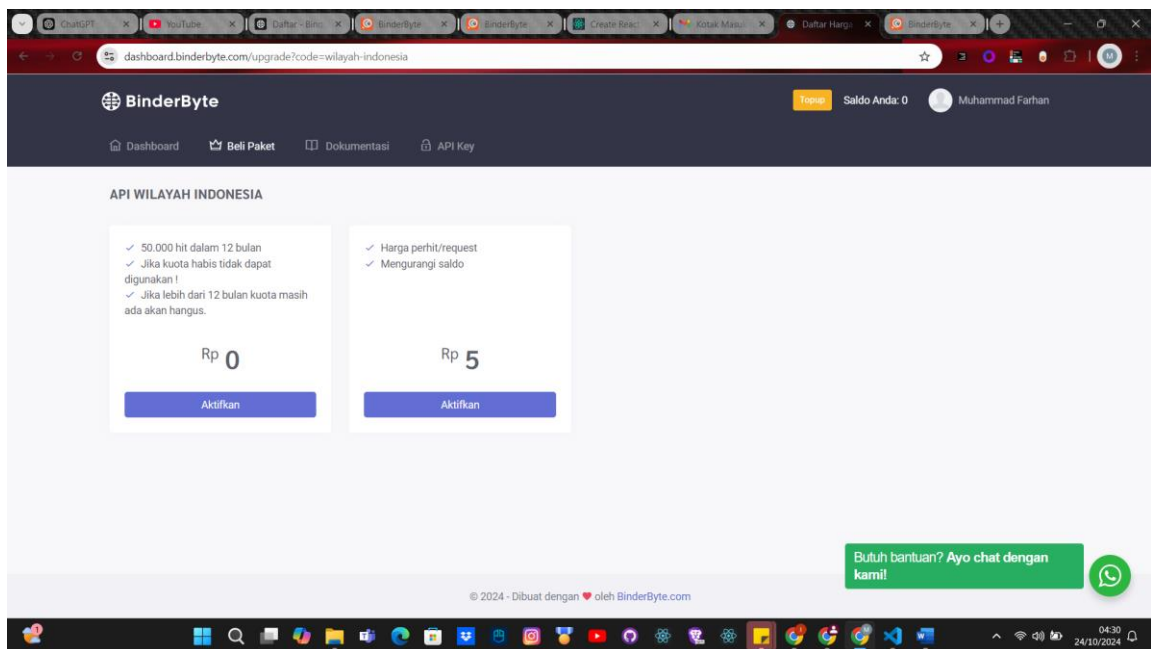
f) Kemudian login ulang dengan akun yang sudah kalian buat kemudian akan diarahkan ke dashboard

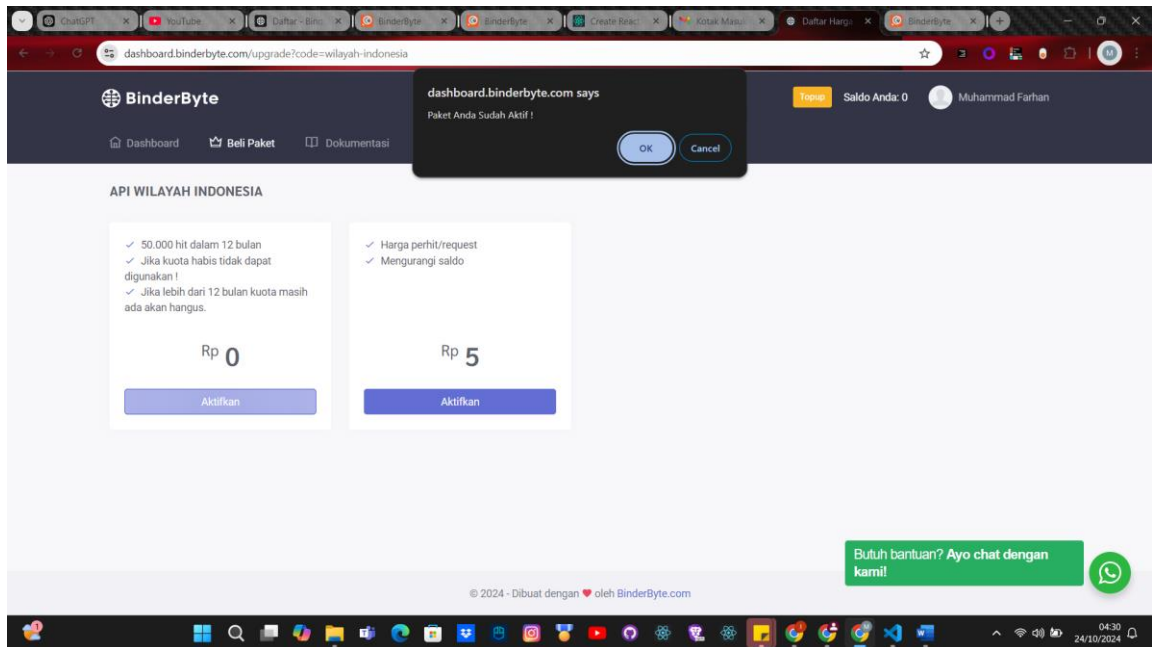


g) Setelah berada di dashboard, klik yang “Api Wilayah Indonesia”

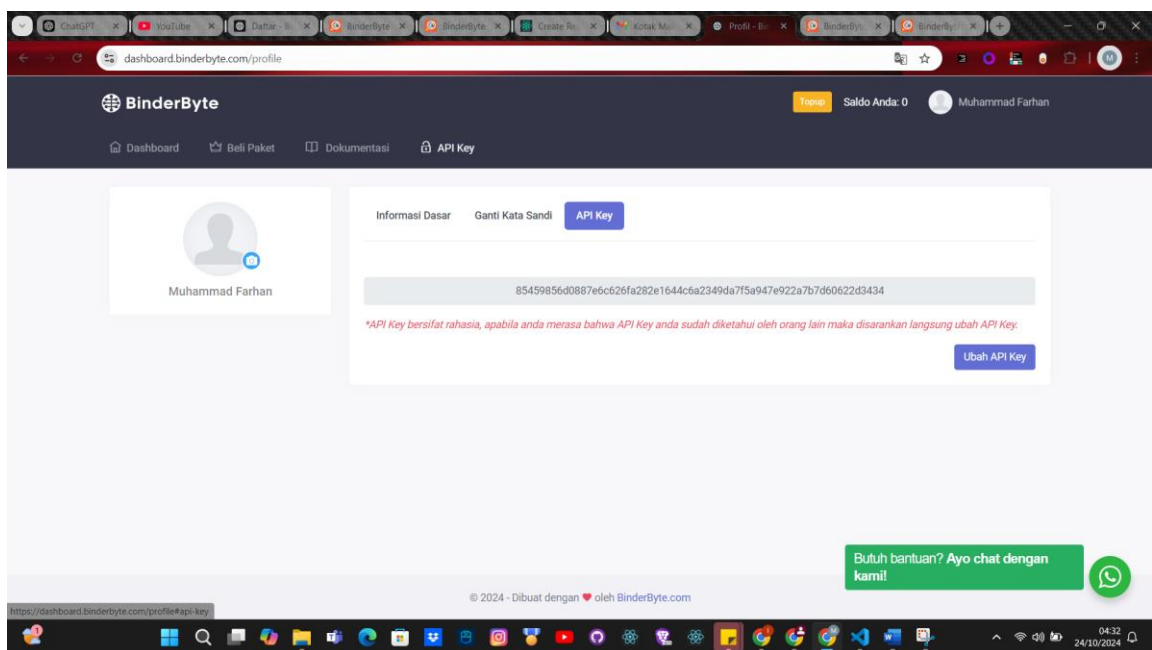
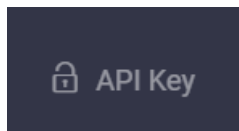


h) Kemudian aktifkan paket yang Rp. 0

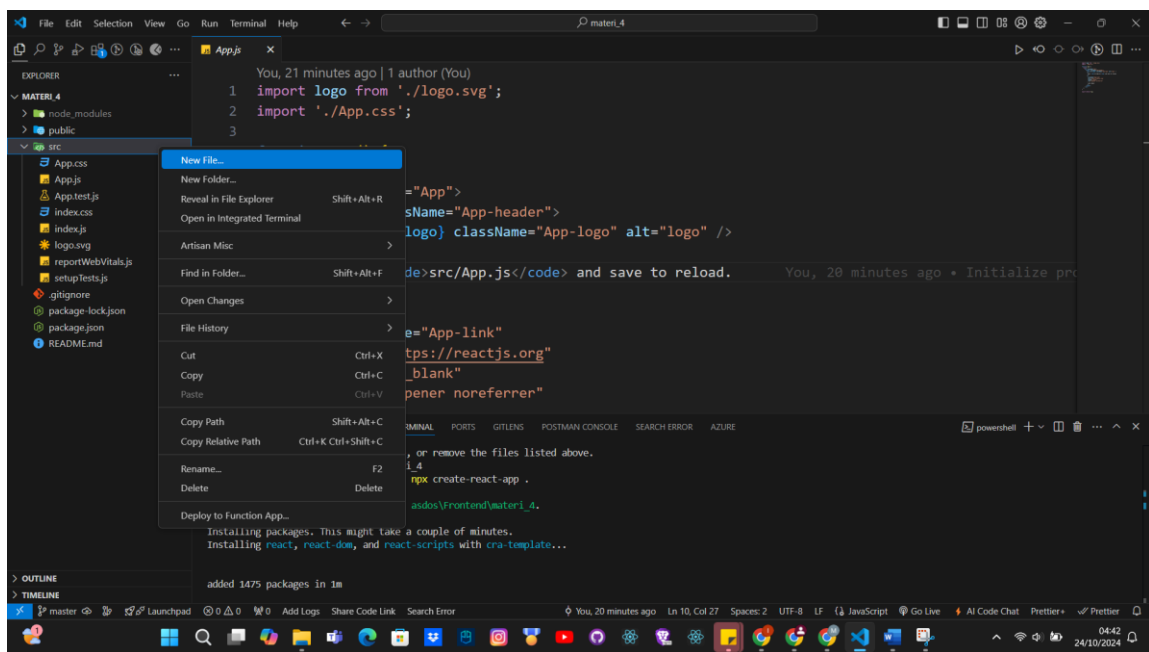
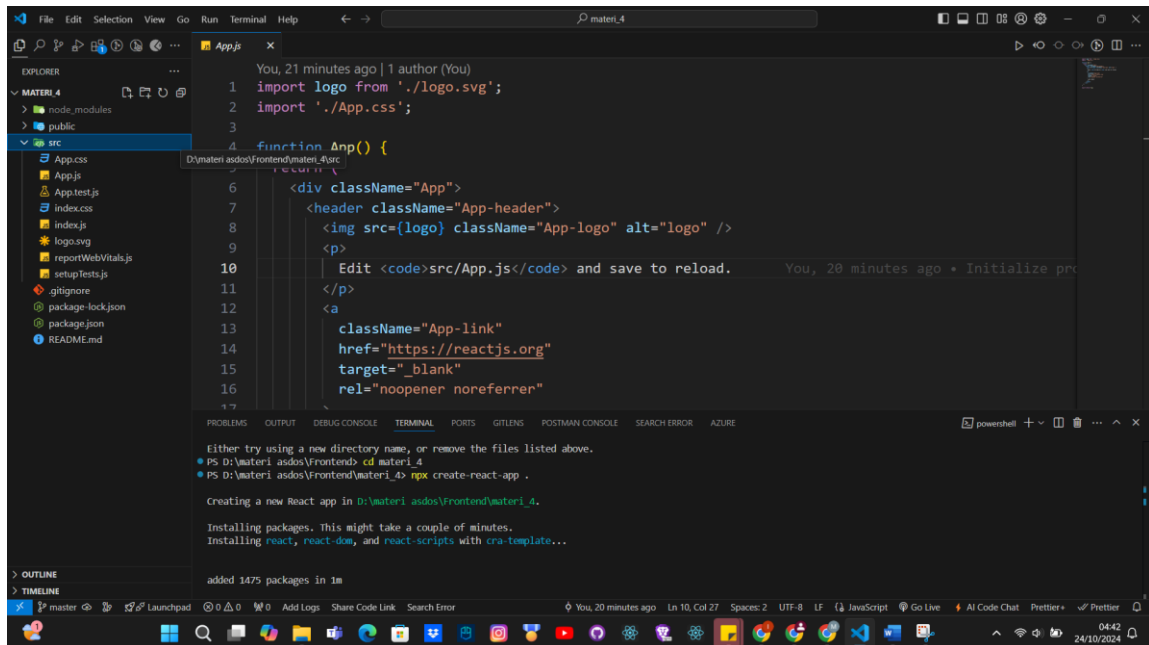




- i) Kemudian click Api-key, setelah itu klik lagi api-key yang dibawahnya, setelah itu api-keynya dicopy



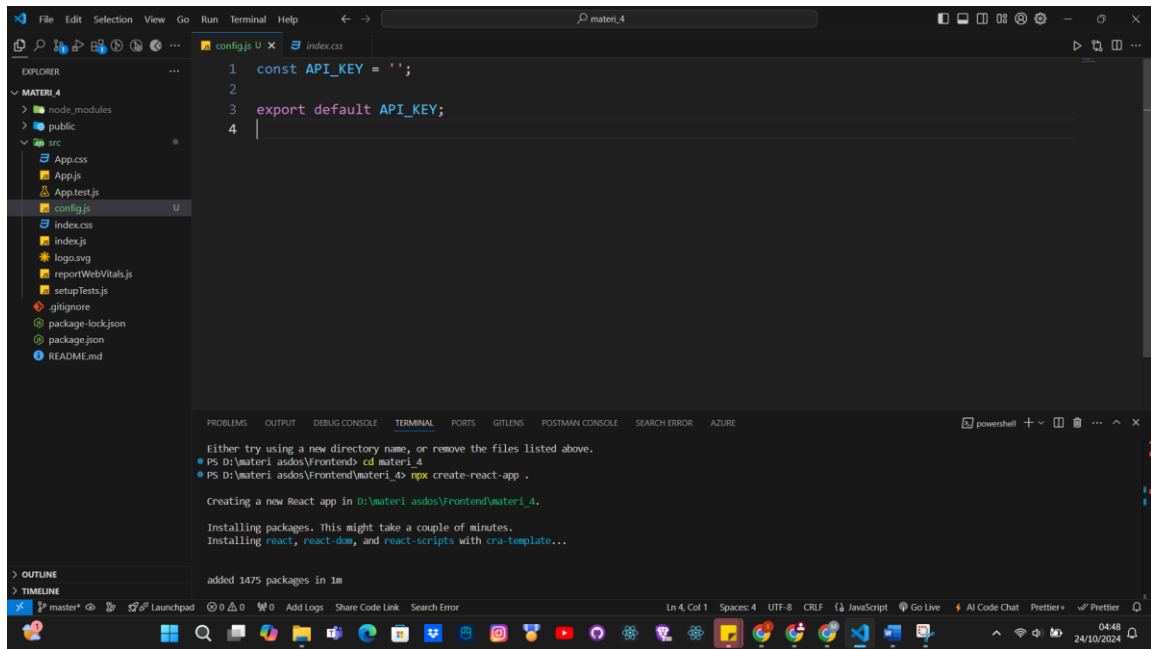
- j) Setelah di copy api-keynya tadi, kemudian balik lagi ke react, dan buatlah file baru di folder "src" dengan nama config.js



k) Kemudian isilah config.js tersebut dengan code ini

```
const API_KEY = '';

export default API_KEY;
```

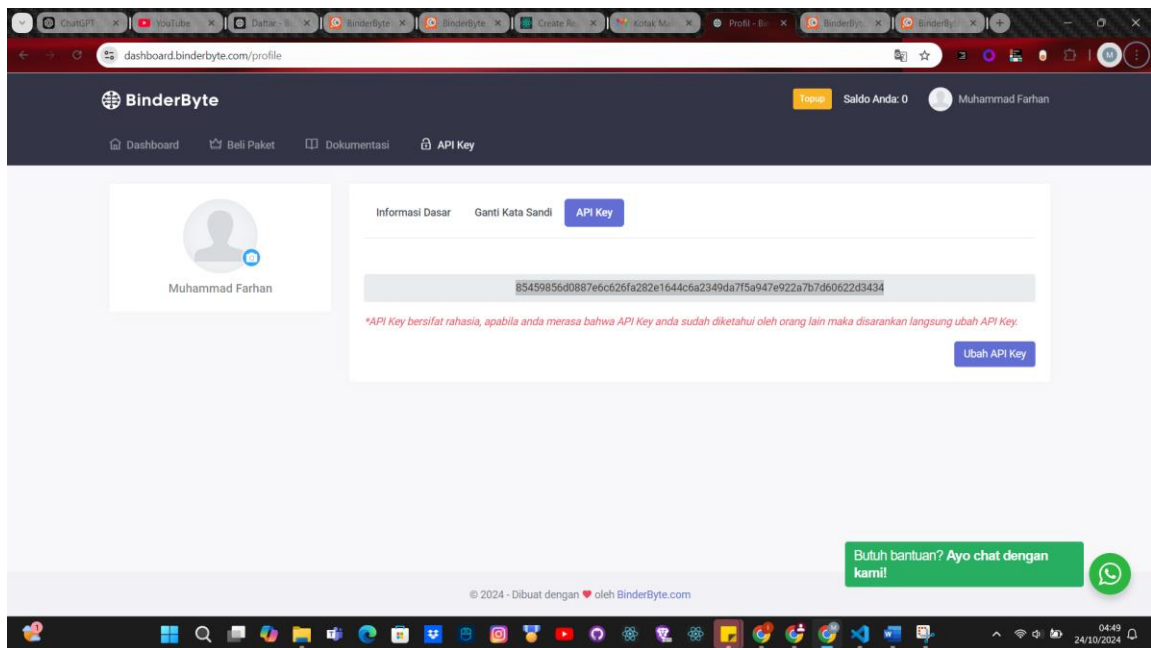


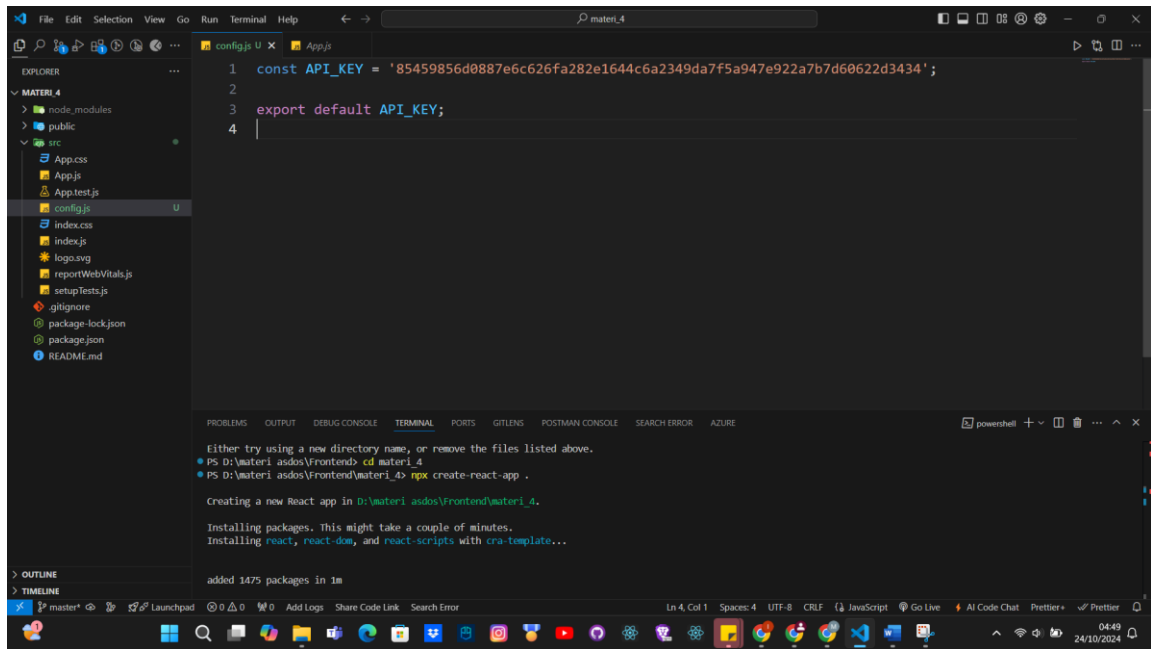
The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the file structure of a project named 'materi_4', including folders like 'node_modules', 'public', and 'src', and files like 'App.css', 'App.js', 'App.test.js', 'config.js', 'index.css', 'index.js', 'logo.svg', 'reportWebVitals.js', 'setupTests.js', 'gitignore', 'package-lock.json', 'package.json', and 'README.md'. The main editor area shows the 'config.js' file with the following code:

```
1 const API_KEY = '';  
2  
3 export default API_KEY;  
4
```

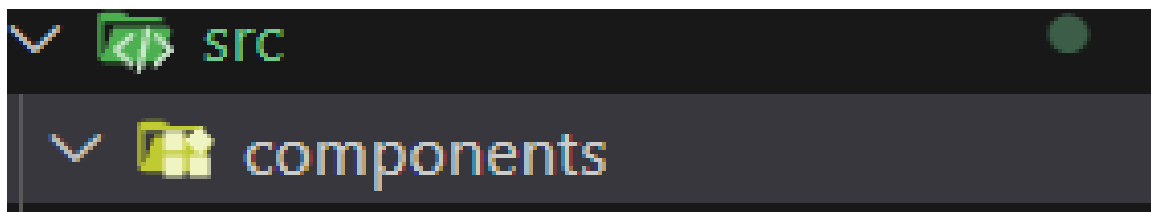
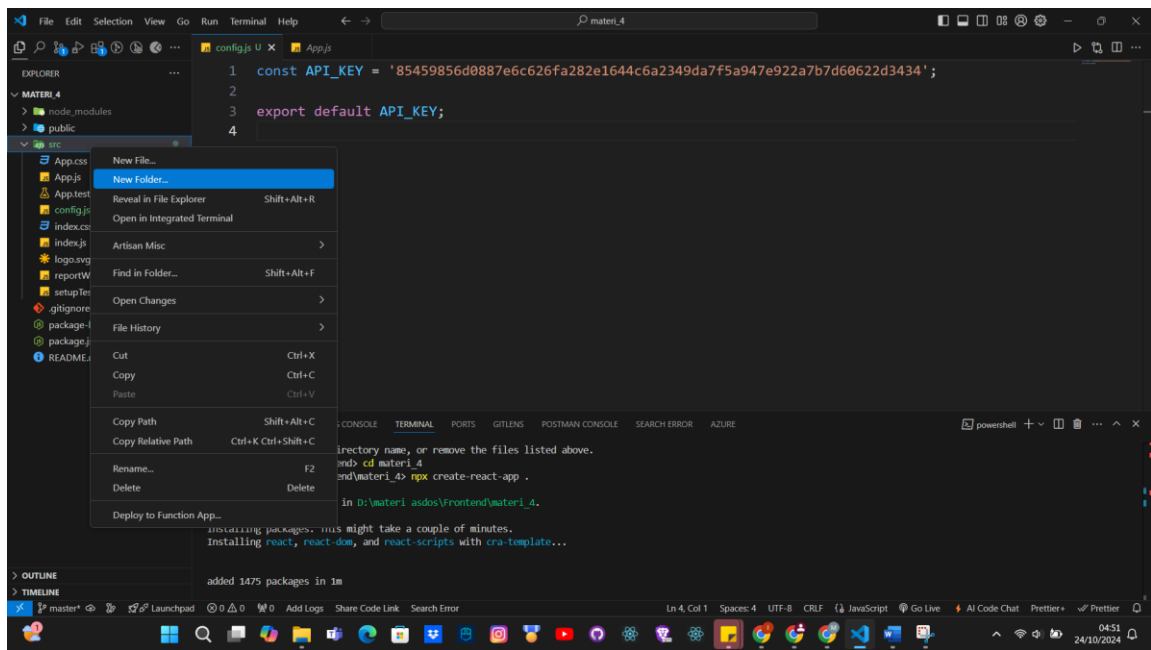
The integrated terminal at the bottom shows the command prompt output for running 'npm create-react-app' in the directory 'D:\materi_4'. The output indicates that a new React app is being created and packages are being installed.

l) Setelah itu isilah Api keynya dengan api key yang sudah kalian copy dari web binderbyte sebelumnya





m) Kemudian buatlah folder baru untuk komponen di “src” dengan nama “components”



n) Setelah itu buatlah file komponen di dalam folder components tersebut.

- ProvinsiList.js

```
import React, { useEffect, useState } from 'react';
import axios from 'axios';
import API_KEY from '../config';

const ProvinsiList = ({ onProvinsiSelect }) => {
  const [provinsiList, setProvinsiList] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    const fetchProvinsi = async () => {
      try {
        const response = await axios.get('https://api.binderbyte.com/wilayah/provinsi', {
          params: {
            api_key: API_KEY,
          },
        });
        setProvinsiList(response.data.value);
        setLoading(false);
      } catch (err) {
        setError(err.message);
        setLoading(false);
      }
    };

    fetchProvinsi();
  }, []);
```

```

    if (loading) return <p>Loading...</p>;

    if (error) return <p>Error: {error}</p>;

    return (
      <div>

        <h3>Pilih Provinsi</h3>

        <select onChange={(e) => onProvinsiSelect(e.target.value)}>

          <option value="">Pilih Provinsi</option>

          {provinsiList.map((provinsi) => (
            <option key={provinsi.id} value={provinsi.id}>

              {provinsi.name}

            </option>

          ))}

        </select>

      </div>
    );
  };
};

export default ProvinsiList;

```

- kabupatenList.js

```

import React, { useEffect, useState } from 'react';

import axios from 'axios';

import API_KEY from '../config';

const KabupatenList = ({ provinsiId, onKabupatenSelect }) => {

  const [kabupatenList, setKabupatenList] = useState([]);

  const [loading, setLoading] = useState(true);

  const [error, setError] = useState(null);

```

```

useEffect(() => {
  if (provinsiId) {
    const fetchKabupaten = async () => {
      try {
        const response = await axios.get('https://api.binderbyte.com/wilayah/kabupaten', {
          params: {
            api_key: API_KEY,
            id_provinsi: provinsiId,
          },
        });
        setKabupatenList(response.data.value);
        setLoading(false);
      } catch (err) {
        setError(err.message);
        setLoading(false);
      }
    };

    fetchKabupaten();
  }
}, [provinsiId]);

if (!provinsiId) return null;
if (loading) return <p>Loading...</p>;
if (error) return <p>Error: {error}</p>;

return (
  <div>
    <h3>Pilih Kabupaten/Kota</h3>
    <select onChange={e => onKabupatenSelect(e.target.value)}>
      <option value="">Pilih Kabupaten/Kota</option>

```

```

      {kabupatenList.map((kabupaten) => (
        <option key={kabupaten.id} value={kabupaten.id}>
          {kabupaten.name}
        </option>
      ))}
    </select>
  </div>
);
};

export default KabupatenList;

```

- kecamatanList.js

```

import React, { useEffect, useState } from 'react';
import axios from 'axios';
import API_KEY from '../config';

const KecamatanList = ({ kabupatenId, onKecamatanSelect }) => {
  const [kecamatanList, setKecamatanList] = useState([]);
  const [loading, setLoading] = useState(true);
  const [error, setError] = useState(null);

  useEffect(() => {
    if (kabupatenId) {
      const fetchKecamatan = async () => {
        try {
          const response = await axios.get('https://api.binderbyte.com/wilayah/kecamatan', {
            params: {
              api_key: API_KEY,
              id_kabupaten: kabupatenId,

```

```

        },

    });

    setKecamatanList(response.data.value);

    setLoading(false);

  } catch (err) {

    setError(err.message);

    setLoading(false);

  }

};

fetchKecamatan();

}

}, [kabupatenId]);

if (!kabupatenId) return null;

if (loading) return <p>Loading...</p>;

if (error) return <p>Error: {error}</p>;

return (

  <div>

    <h3>Pilih Kecamatan</h3>

    <select onChange={e => onKecamatanSelect(e.target.value)}>

      <option value="">Pilih Kecamatan</option>

      {kecamatanList.map((kecamatan) => (

        <option key={kecamatan.id} value={kecamatan.id}>

          {kecamatan.name}

        </option>

      ))}

    </select>

  </div>

);

};

```

```
export default KecamatanList;
```

- kelurahanList.js

```
import React, { useEffect, useState } from 'react';

import axios from 'axios';

import API_KEY from '../config';

const KelurahanList = ({ kecamatanId }) => {

  const [kelurahanList, setKelurahanList] = useState([]);

  const [loading, setLoading] = useState(true);

  const [error, setError] = useState(null);

  useEffect(() => {

    if (kecamatanId) {

      const fetchKelurahan = async () => {

        try {

          const response = await axios.get('https://api.binderbyte.com/wilayah/kelurahan', {

            params: {

              api_key: API_KEY,

              id_kecamatan: kecamatanId,

            },

          });

          setKelurahanList(response.data.value);

          setLoading(false);

        } catch (err) {

          setError(err.message);

          setLoading(false);

        }

      };

      fetchKelurahan();

    }

  });

}
```

```

    }, [kecamatanId]);

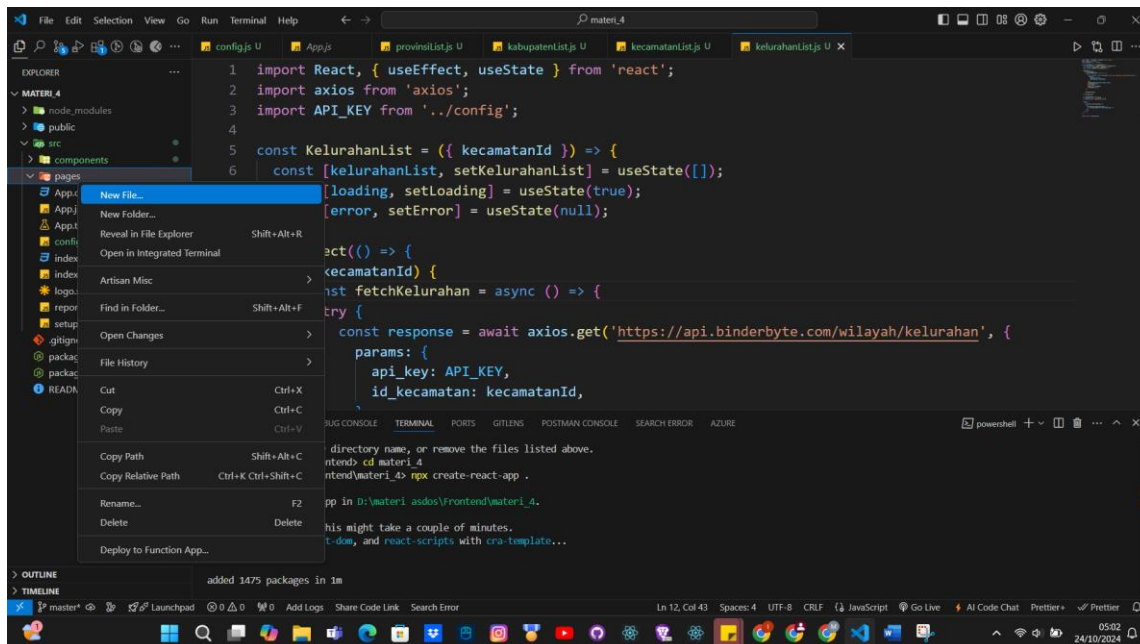
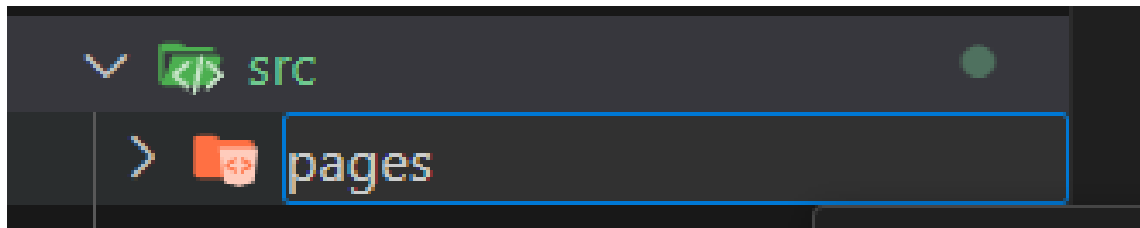
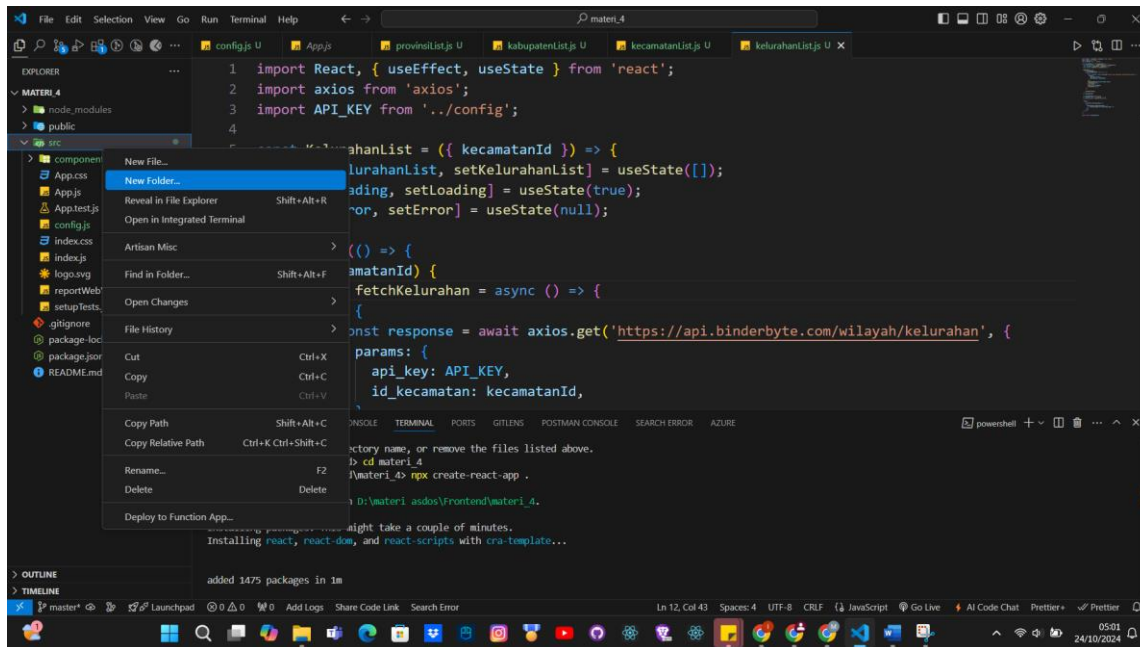
    if (!kecamatanId) return null;
    if (loading) return <p>Loading...</p>;
    if (error) return <p>Error: {error}</p>;

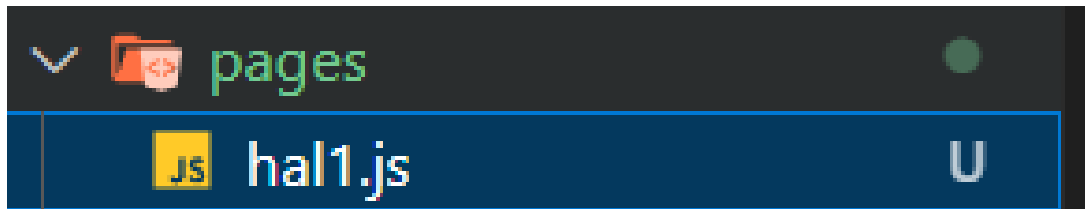
    return (
      <div>
        <h3>Daftar Kelurahan/Desa</h3>
        <ul>
          {kelurahanList.map((kelurahan) => (
            <li key={kelurahan.id}>{kelurahan.name}</li>
          ))}
        </ul>
      </div>
    );
  };
};

export default KelurahanList;

```

- o) Setelah itu buatlah folder pages di dalam folder src, kemudian buatlah file hal1.js di dalamnya





p) Setelah itu masukkanlah code ini di hal1.js nya

```
import React, { useState } from 'react';
import ProvinsiList from '../provinsiList/components/ProvinsiList';
import KabupatenList from '../kabupatenList/components/KabupatenList';
import KecamatanList from '../kecamatanList/components/KecamatanList';
import KelurahanList from '../kelurahanList/components/KelurahanList';

function Hal1() {
  const [selectedProvinsi, setSelectedProvinsi] = useState(null);
  const [selectedKabupaten, setSelectedKabupaten] = useState(null);
  const [selectedKecamatan, setSelectedKecamatan] = useState(null);

  return (
    <div className="App">
      <header className="App-header">
        <h1>Data Wilayah Indonesia</h1>
        <h2>Selamat Datang di Aplikasi Data Wilayah</h2>
        <ProvinsiList onProvinsiSelect={setSelectedProvinsi} />
        {selectedProvinsi && (
          <KabupatenList
            provinsiId={selectedProvinsi}
            onKabupatenSelect={setSelectedKabupaten}
          />
        )}
        {selectedKabupaten && (
          <KecamatanList
            kabupatenId={selectedKabupaten}

```

```

        onKecamatanSelect={setSelectedKecamatan}
      />
    )}
    {selectedKecamatan && <KelurahanList
kecamatanId={selectedKecamatan} />}
  </header>
</div>
);
}

export default Hal1;

```

q) Setelah itu buatlah route di app.js agar bisa mengaskes hal1.js

```

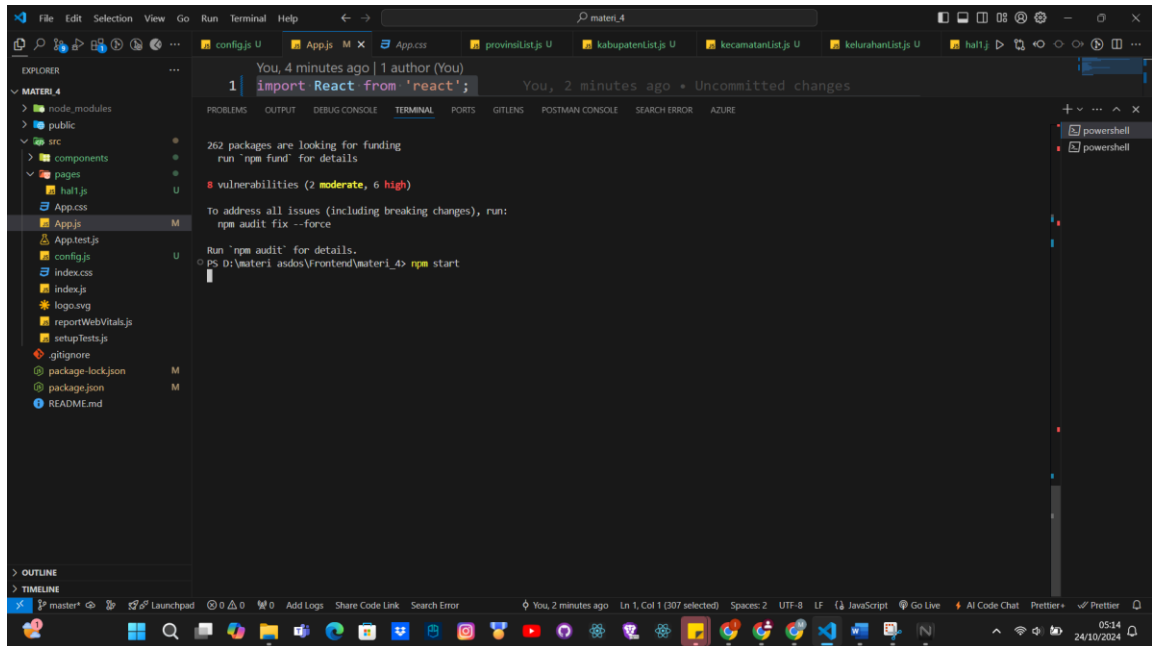
import React from 'react';
import { BrowserRouter as Router, Route, Routes,
Link } from 'react-router-dom';
import Hal1 from './pages/hal1';

function App() {
  return (
    <Router>
      <Routes>
        <Route path="/" element={<Hal1 />} />
      </Routes>
    </Router>
  );
}

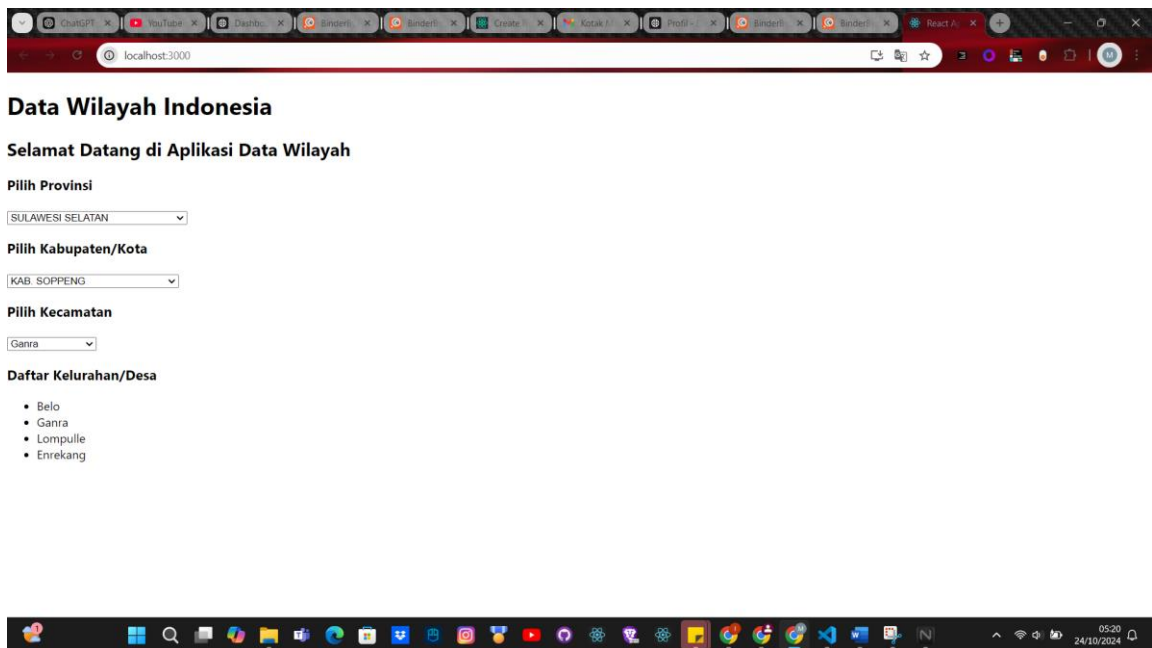
export default App;

```

- r) Kemudian Coba jalankan project yang sudah kita buat, dengan command
“npm start”



- s) Kemudian hasilnya seperti ini



2. TUGAS

Jawablah Pertanyaan dari Soal ini |

1. Cobalah untuk melakukan semua percobaan diatas
2. Kemudian buatalah terapkanlah di project kalian (nggak harus pakai api yang ini boleh search api yang lain)
3. sesuaikanlah penggunaannya dengan project kalian masing masing

PENGUMPULAN TUGAS DAN DEADLINE

- a. Pengumpulan Tugas di Google Drive
- b. Deadline = 24 Oktober 2024 | 23.00 WIB

DAFTAR PUSTAKA

<https://www.ibm.com/topics/api>

<https://binderbyte.com/>

<https://docs.binderbyte.com/api/wilayah>

<https://documenter.getpostman.com/view/12963788/TVRg69g4>

<https://github.com/farizdotid/DAFTAR-API-LOKAL-INDONESIA>