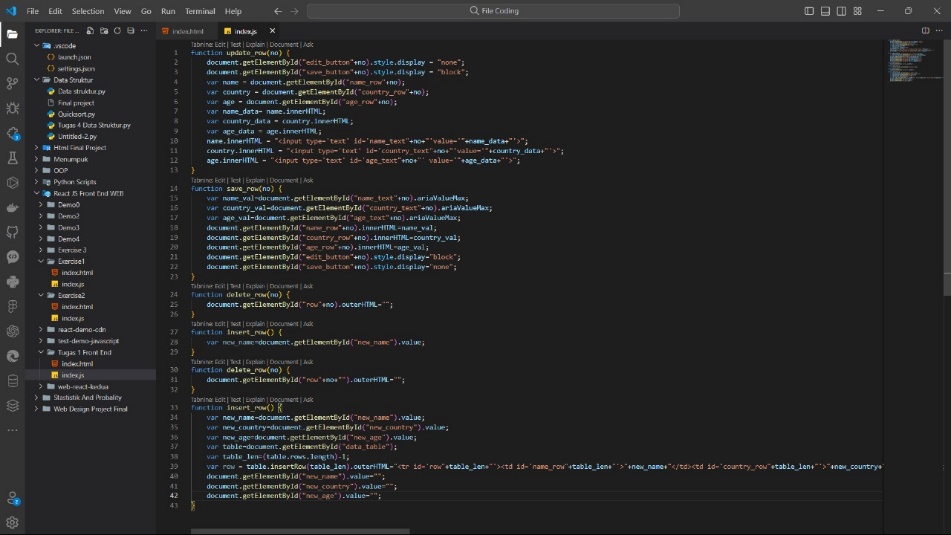
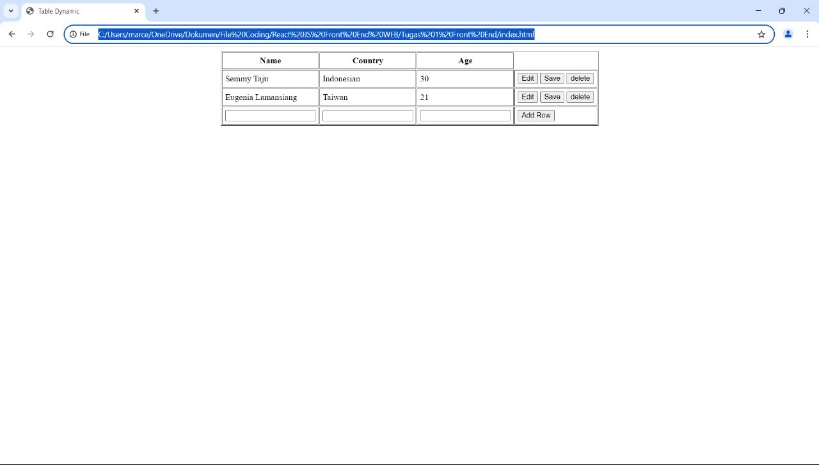
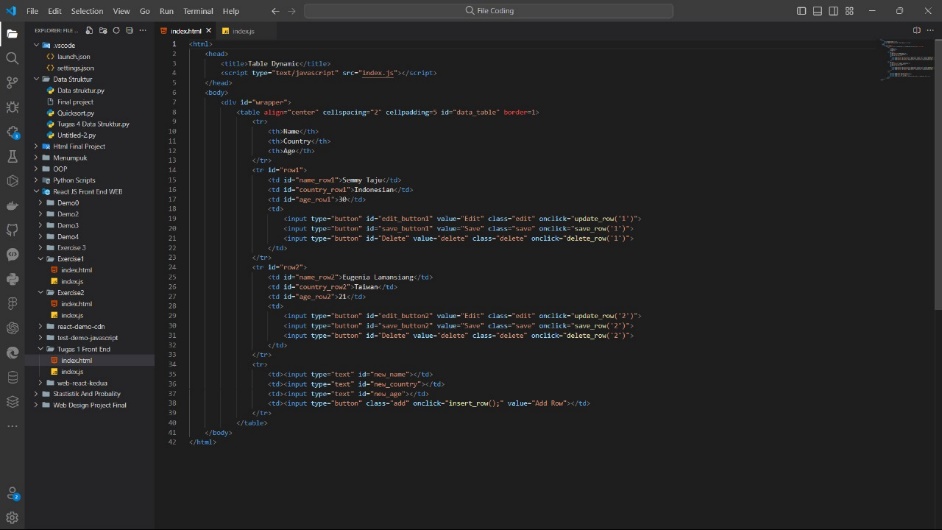
**FRONT-END WEB DEVELOPMENT**

* **Nanalis, Arlino Abraham Timotius**
* **Sorongan, Michelle Heavenly Faith**
* **Kaseger, Ivan Ibrani**
* **Sijadi, Marcello Bryan**

**Exercise 1 - Table Dynamically**

****

Tag <html> dan <head>:

Tag <html> menandakan awal dari dokumen HTML.

Di dalam tag <head>, terdapat tag <title> yang memberikan judul pada halaman, serta tag <script> yang mengimpor file JavaScript eksternal bernama index.js. File ini diharapkan berisi fungsi-fungsi yang diperlukan untuk mengelola interaksi dengan tabel.

Tag <body> dan <div>:

Tag <body> berisi konten yang akan ditampilkan di halaman.

Tag <div> dengan id wrapper digunakan untuk membungkus tabel, memberikan struktur yang lebih baik.

Tag <table>:

Tabel ini diatur agar berada di tengah halaman dengan atribut align="center".

Atribut cellspacing dan cellpadding mengatur jarak antar sel dan padding dalam sel.

Tabel memiliki tiga kolom: Name, Country, dan Age.

Baris Data

Setiap baris data diwakili oleh tag <tr>. Dalam contoh ini, terdapat dua baris data yang sudah ada, masing-masing dengan informasi nama, negara, dan usia.

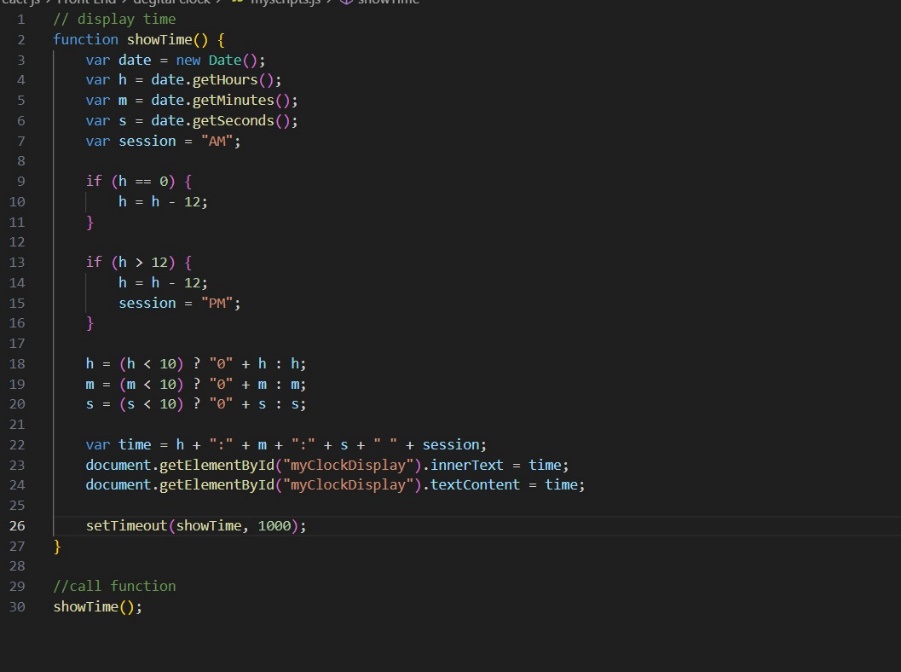
Setiap baris juga memiliki tiga tombol: Edit, Save, dan Delete, yang masing-masing memanggil fungsi JavaScript untuk mengedit, menyimpan, atau menghapus baris tersebut.

Baris Input untuk Menambah Data:

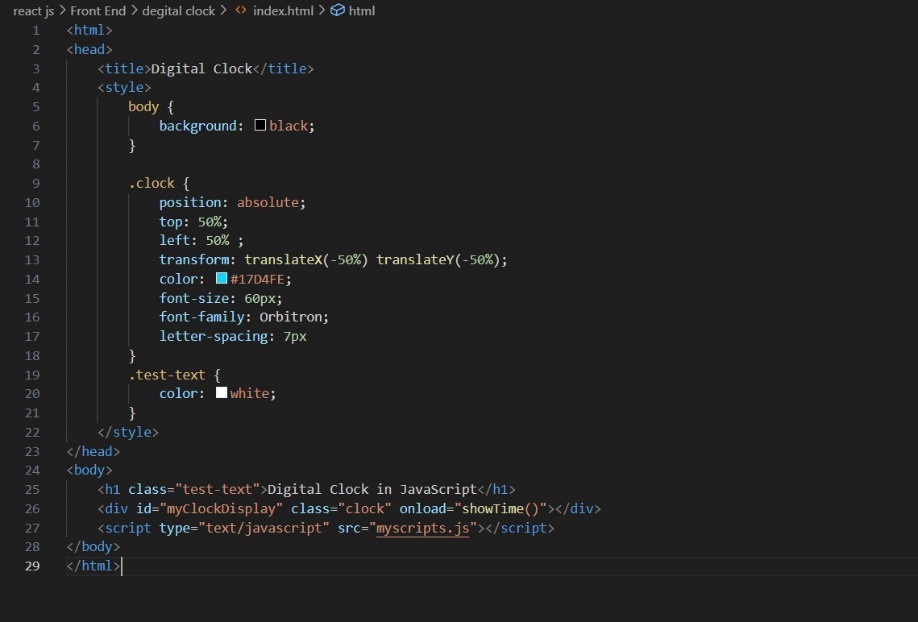
Baris terakhir dalam tabel menyediakan input untuk menambah data baru. Terdapat tiga input teks untuk nama, negara, dan usia, serta sebuah tombol Add Row yang memanggil fungsi insert\_row() untuk menambahkan data baru ke dalam tabel.

Secara keseluruhan, kode ini memberikan kerangka dasar untuk membuat tabel dinamis yang dapat diubah oleh pengguna. Dengan menambahkan logika JavaScript yang sesuai dalam file index.js, pengguna dapat berinteraksi dengan tabel ini secara efektif.

**Exercise 2 - Digital Clock**

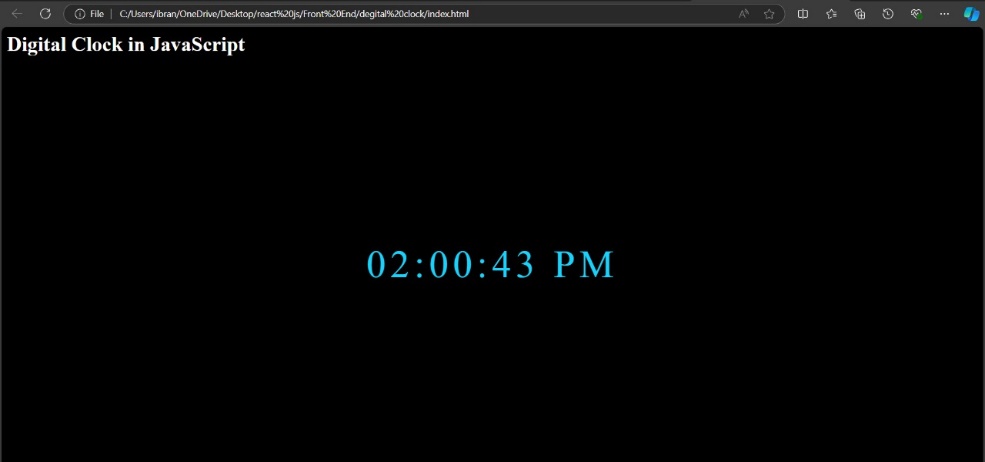
****

Pertama dalam elemen <style> halaman diberikan latar belakang hitam dan dua kelas untuk menampilkan jam digital di tengah layar dengan teks biru besar dan judul teks putih. Di bagian <body>, sebuah judul "Digital Clock in JavaScript" ditampilkan di bagian atas halaman dengan gaya teks putih, sementara elemen <div> yang berfungsi sebagai wadah untuk jam digital ditempatkan tepat di tengah halaman, dan terakhir.

****

yang pertama-tama membuat objek Date untuk mengambil waktu saat ini, kemudian mengonversi jam dari format 24 jam menjadi 12 jam dan menentukan apakah waktu tersebut adalah AM atau PM

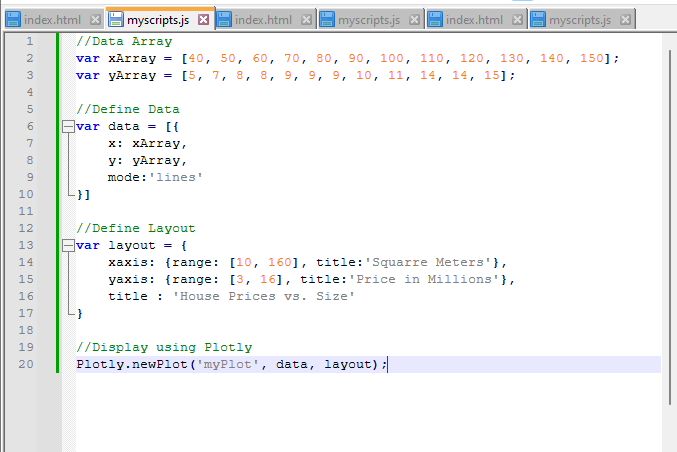
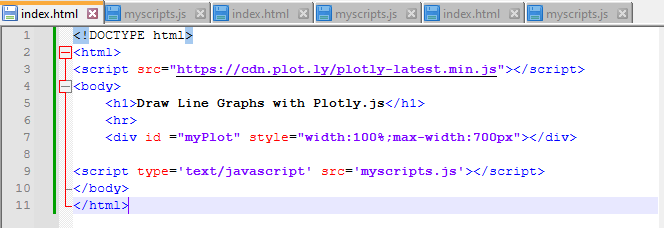
kemudian ditampilkan dalam elemen <div> yang memiliki id myClockDisplay melalui DOM. Fungsi ini terus memanggil dirinya sendiri setiap detik menggunakan setTimeout, sehingga waktu yang ditampilkan pada layar selalu diperbarui secara real-time.

****

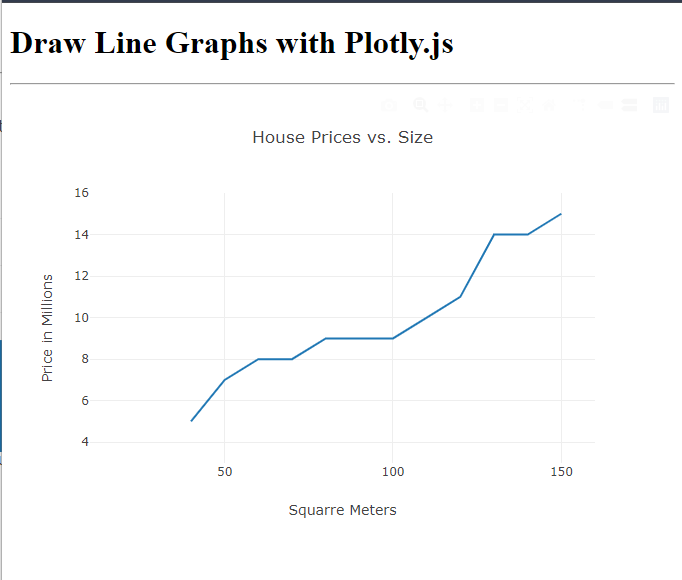
Output yang yang di tampilkan memperlihatkan jam digital untuk menunjukan waktu saat ini dalam format 12 jam, dan angka di jam digital itu akan otomatis di perbaharui setiap detik secara real time.

**Exercise 3 – Graphing Library**

* **Line Graph**

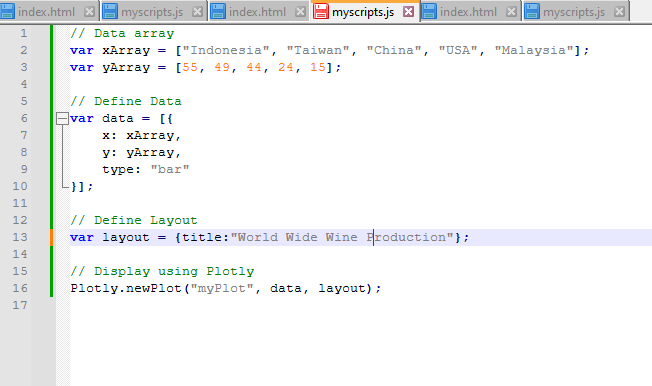
****

Menunjukan langkah-langkah untuk membuat grafik garis sederhana menggunakan Plotly.js, sebuah pustaka JavaScript untuk visualisasi data interaktif. Gambar pertama menampilkan kode HTML (index.html) dan JavaScript (myscripts.js). File HTML menyertakan tag script untuk membuat pustaka Plotly melalui CDN dan sebuah div dengan ID “myplot”, tempat grafik akan ditampilkan.

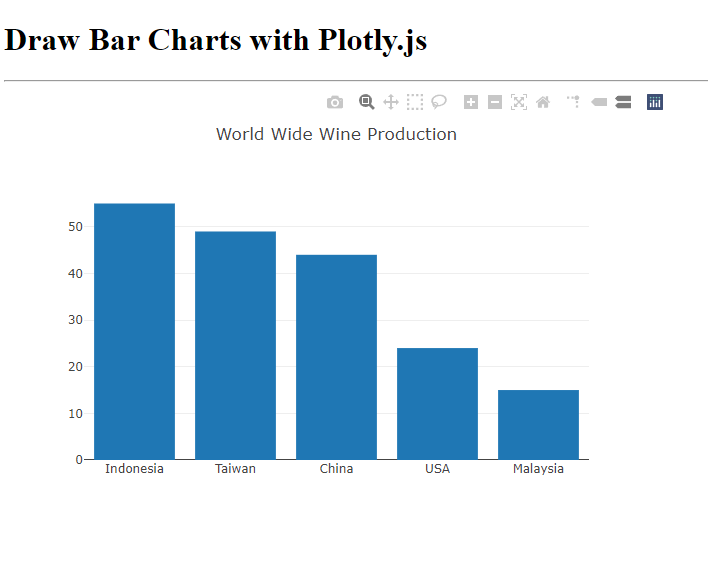
****

File JavaScript mendefinisikan array untuk data sumbu x dan y, mengatur konfigurasi data dan tata letak, dan akhirnya menggunakan Plotly.newPlot

* **Bar Charts**

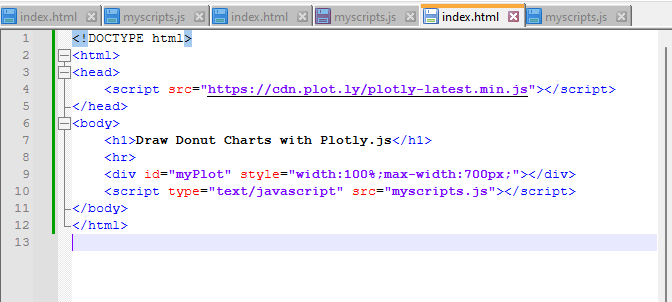
****

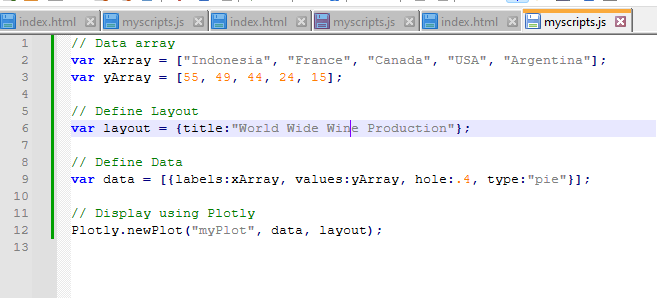
Menunjukkan proses pembuatan grafik batang menggunakan Plotly.js. Gambar pertama menampilkan kode HTML (index.html) dan JavaScript (myscripts.js). File HTML menyertakan tag script untuk memuat pustaka Plotly melalui CDN dan sebuah div dengan ID "myPlot" yang akan digunakan untuk menampilkan grafik batang. File JavaScript mendefinisikan array untuk data sumbu x dan y, yang dalam hal ini berisi nama negara dan nilai produksi anggur mereka. Selanjutnya, data dan tata letak grafik diatur, termasuk judul grafik "World Wide Wine Production".

****

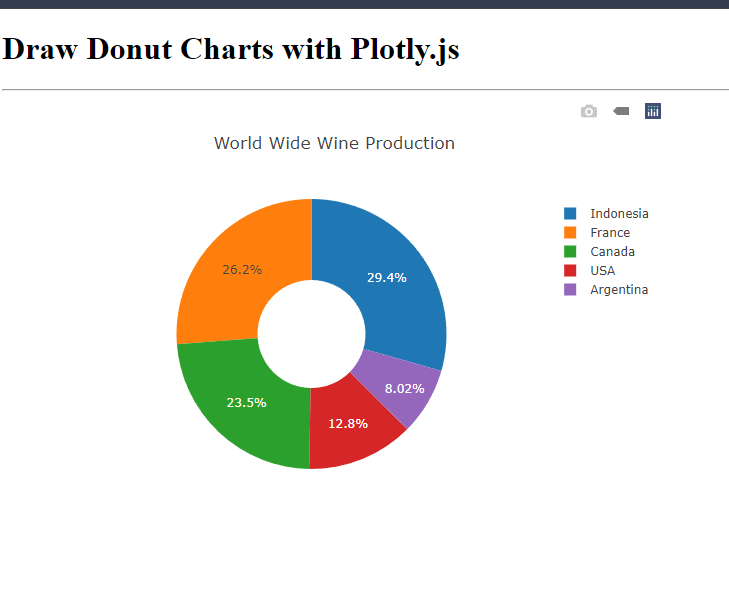
Akhirnya, grafik batang dihasilkan dan ditampilkan di halaman web menggunakan Plotly.newPlot. Gambar kedua menunjukkan hasil akhir grafik batang yang dihasilkan, dengan masing-masing batang mewakili produksi anggur di berbagai negara.

* **Donut Charts**

****

****

Menunjukkan langkah-langkah pembuatan diagram donat menggunakan Plotly.js dalam proyek web sederhana. Bagian pertama adalah kode HTML yang memuat pustaka Plotly.js dan menyediakan struktur dasar, termasuk <div> untuk menampilkan diagram. Bagian kedua adalah kode JavaScript yang mendefinisikan data dan tata letak diagram dengan menggunakan dua array (xArray dan yArray) untuk label negara dan nilai produksi, serta mengatur diagram menjadi tipe "pie" dengan lubang di tengah, menciptakan efek donat. Kode ini kemudian memanggil Plotly.newPlot() untuk merender diagram di dalam <div> yang disediakan. Bagian terakhir menampilkan hasil akhir berupa diagram donat di browser, yang menggambarkan distribusi produksi anggur di beberapa negara dengan persentase yang ditunjukkan di masing-masing segmen.

****

**Exercise 4 – Dynamic Donut**

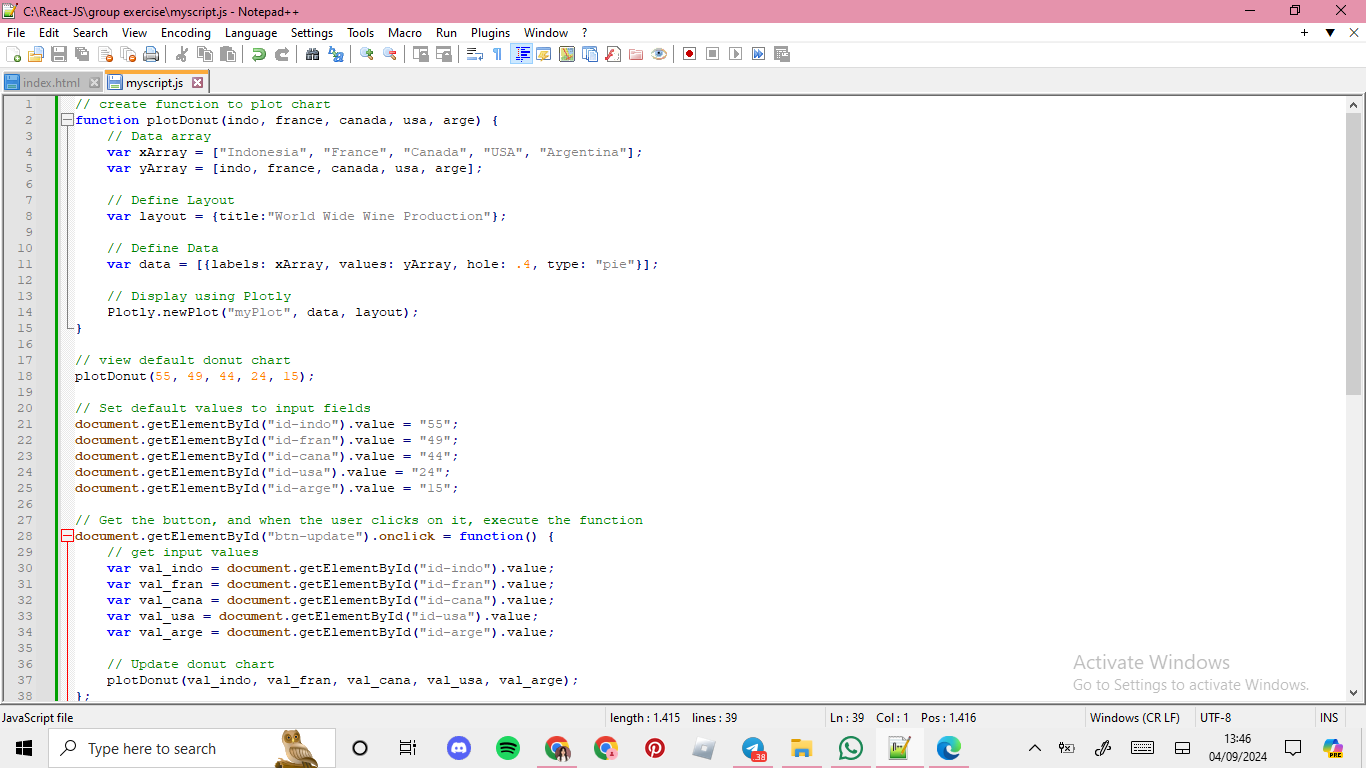
****

Input Fields : Ada beberapa input bertipe "number" untuk negara-negara seperti Prancis, Kanada, USA, dan Argentina. Setiap input memiliki id unik (misalnya id-fran untuk Prancis) dan batasan nilai minimum dan maksimum.

Button : Ada tombol dengan id="btn-update" yang akan memicu pembaruan chart ketika diklik.

Div untuk Chart : Sebuah div dengan id="myPlot" yang digunakan untuk menampilkan chart.

Script : Pada bagian akhir, sebuah script eksternal (myscript.js) dimuat yang berisi logika untuk chart dan manipulasi DOM.

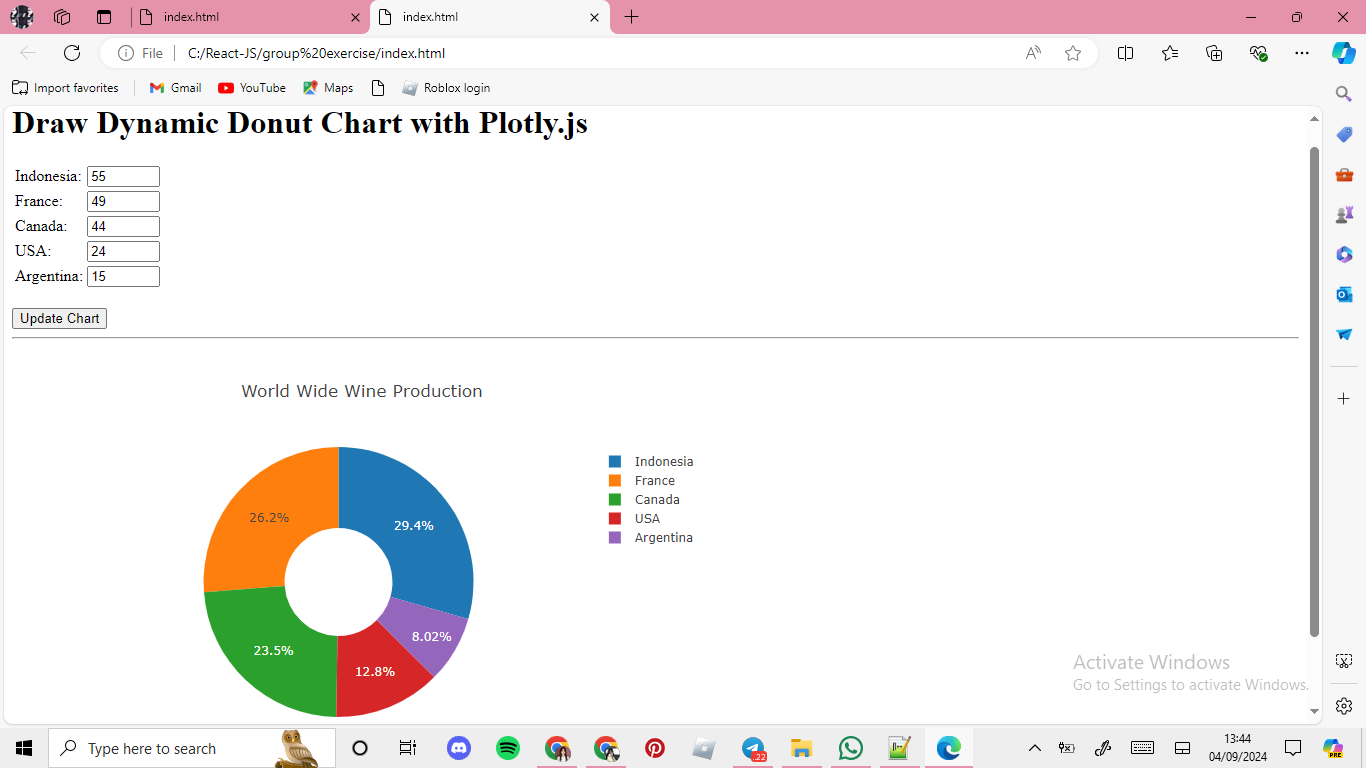
****

Function plotDonut : Fungsi ini digunakan untuk membuat chart berbentuk donut dengan menggunakan library Plotly. Fungsi ini menerima lima parameter yang akan menjadi nilai untuk setiap negara.

Inisialisasi Chart : Pada awal script, fungsi plotDonut dipanggil dengan nilai default untuk membuat chart donut.

Set Default Values : Script juga mengatur nilai default untuk setiap input field dengan nilai yang sama seperti yang ditampilkan dalam chart awal.

Event Listener : Script menambahkan event listener untuk tombol "Update Chart". Ketika tombol diklik, nilai dari setiap input field diambil dan digunakan untuk memperbarui chart donut dengan memanggil kembali plotDonut menggunakan nilai-nilai baru

****

Kode tersebut membuat sebuah halaman web yang menampilkan input untuk beberapa negara dan sebuah tombol "Update Chart". Ketika pengguna memasukkan angka ke dalam input dan menekan tombol, grafik berbentuk donut diperbarui sesuai dengan nilai input tersebut, menggunakan library Plotly untuk visualisasi data.