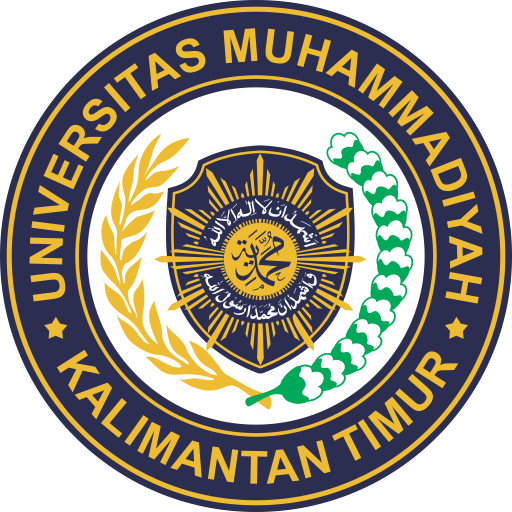
**LAPORAN PRAKTIKUM PEMROGRAMAN WEB**



**Oktavia Ramadhany**

**2411102441095**

**TEKNIK INFORMATIKA**

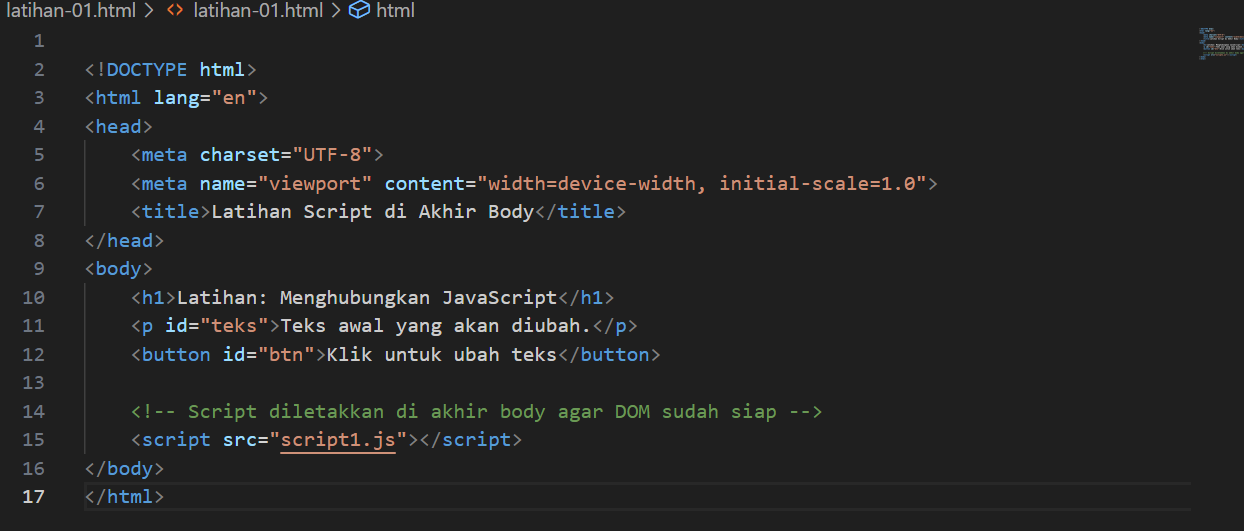
**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS MUHAMMADIYAH KALIMANTAN TIMUR**

**2025**

**-Materi Pertemuan 4 Javascript**

* **Latihan 1 HTML**



Penjelasan HTML

Kode HTML ini adalah kerangka dasar dari halaman web.

Baris <!DOCTYPE html> menunjukkan bahwa dokumen menggunakan standar HTML5.

<html lang="en"> menandakan bahasa utama dokumen adalah bahasa Inggris.

Di dalam <head> ada pengaturan karakter (UTF-8) supaya semua huruf bisa ditampilkan dengan benar, dan pengaturan viewport agar halaman bisa tampil responsif di berbagai ukuran layar, terutama di smartphone.

<title> memberi judul tab browser yaitu “Latihan Script di Akhir Body”.

Pada bagian <body>, terdapat:

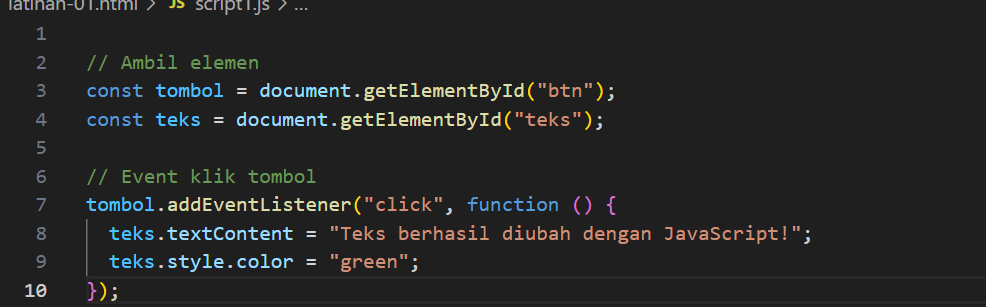
<h1> sebagai judul utama halaman.

<p id="teks"> yaitu paragraf yang teksnya akan diubah dengan bantuan JavaScript. Id teks berfungsi sebagai pengenal unik supaya JavaScript bisa dengan mudah menemukan elemen ini.

<button id="btn"> yaitu tombol dengan teks “Klik untuk ubah teks”. Sama seperti paragraf, tombol ini juga punya id yaitu btn agar bisa dipanggil dari JavaScript.

Terakhir, <script src="script1.js"></script> digunakan untuk menyambungkan file JavaScript eksternal bernama script1.js. Letaknya di bagian akhir <body> supaya ketika script dijalankan, semua elemen HTML sudah selesai dimuat (DOM sudah siap).

* **Latihan 1 JS**



Isi dari script1.js berisi kode interaktif:

1. const tombol = document.getElementById("btn");

→ JavaScript mencari tombol yang memiliki id btn, lalu menyimpannya dalam variabel tombol.

2. const teks = document.getElementById("teks");

→ JavaScript mencari paragraf dengan id teks, lalu menyimpannya dalam variabel teks.

3. tombol.addEventListener("click", function () { ... });

→ Menambahkan sebuah event listener pada tombol. Event listener ini menunggu kejadian klik. Jika tombol diklik, maka fungsi di dalam kurung kurawal {...} akan dieksekusi.

4. Di dalam fungsi:

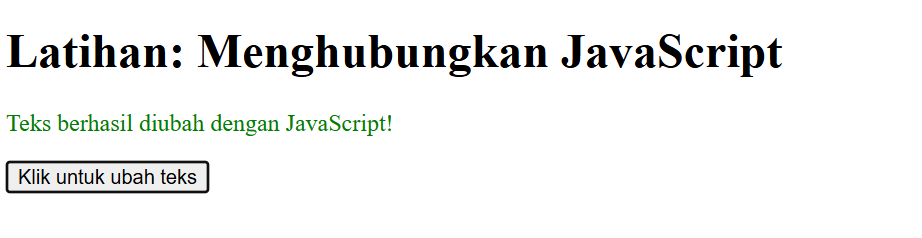
teks.textContent = "Teks berhasil diubah dengan JavaScript!";

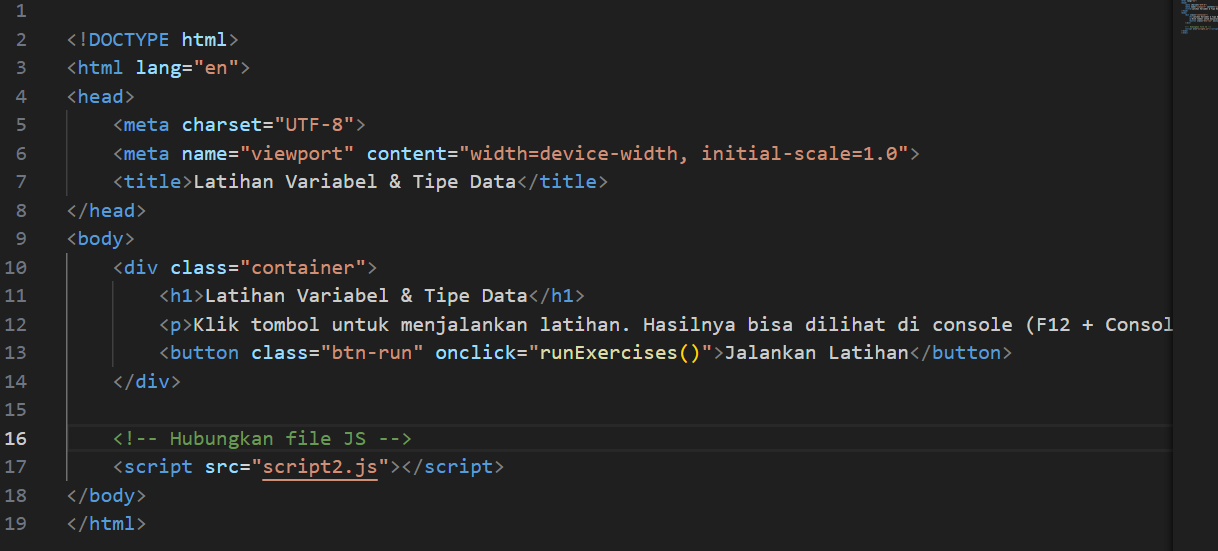
→ Mengubah isi teks paragraf menjadi kalimat baru.

teks.style.color = "green";

→ Mengubah warna teks paragraf menjadi hijau.

Hasil Web :



* **Latihan 2 HTML**

Penjelasan:

<!DOCTYPE html> → menandakan bahwa dokumen ini memakai standar HTML5

<meta charset="UTF-8"> → agar halaman bisa menampilkan semua jenis karakter, termasuk huruf spesial

<meta name="viewport" ...> → agar tampilan web responsif di perangkat mobile.

<title> → judul tab browser: “Latihan Variabel & Tipe Data”.

<div class="container"> → sebagai wadah isi halaman.

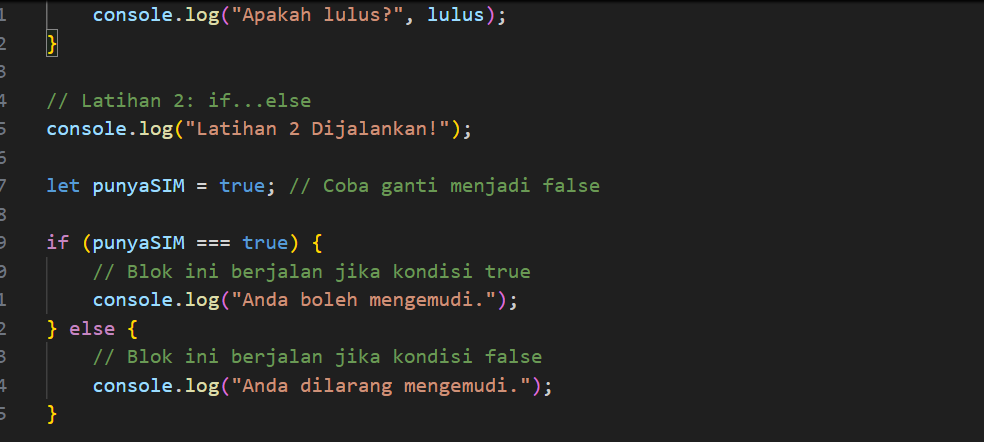
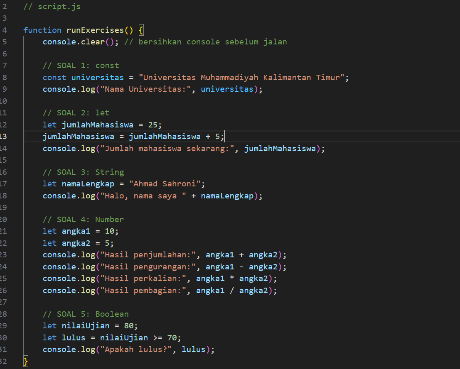
<h1> → judul besar halaman.

<p> → instruksi kepada pengguna, hasil program bisa dilihat di console (dibuka lewat F12 → tab Console).

<button> → tombol dengan teks “Jalankan Latihan”. Ada atribut onclick="runExercises()" artinya saat tombol diklik, fungsi runExercises() (yang ada di file JavaScript) akan dijalankan.

<script src="script2.js"></script> → menghubungkan HTML dengan file JavaScript eksternal bernama script2.js.

* **Latihan 2 JS**



Penjelasan:

1. function runExercises()

Fungsi ini berisi latihan soal variabel dan tipe data. Akan dijalankan ketika tombol di HTML diklik.

2. console.clear();

Membersihkan tampilan console agar hasil baru lebih rapi.

3. Soal 1: const

const universitas = "Universitas Muhammadiyah Kalimantan Timur";

Menyimpan nama universitas dalam variabel konstan (const → nilainya tidak bisa diubah).

4. Soal 2: let

let jumlahMahasiswa = 25;

Kemudian ditambah 5 → hasilnya 30. Nilai let bisa berubah.

5. Soal 3: String

let namaLengkap = "Ahmad Sahroni";

Dicetak ke console dengan kalimat sapaan.

6. Soal 4: Number

Dua angka (angka1 = 10, angka2 = 5) digunakan untuk operasi aritmatika:

Penjumlahan → 15

Pengurangan → 5

Perkalian → 50

Pembagian → 2

7. Soal 5: Boolean

nilaiUjian = 80

lulus = nilaiUjian >= 70 → menghasilkan true karena 80 ≥ 70. Dicetak: “Apakah lulus? true”.

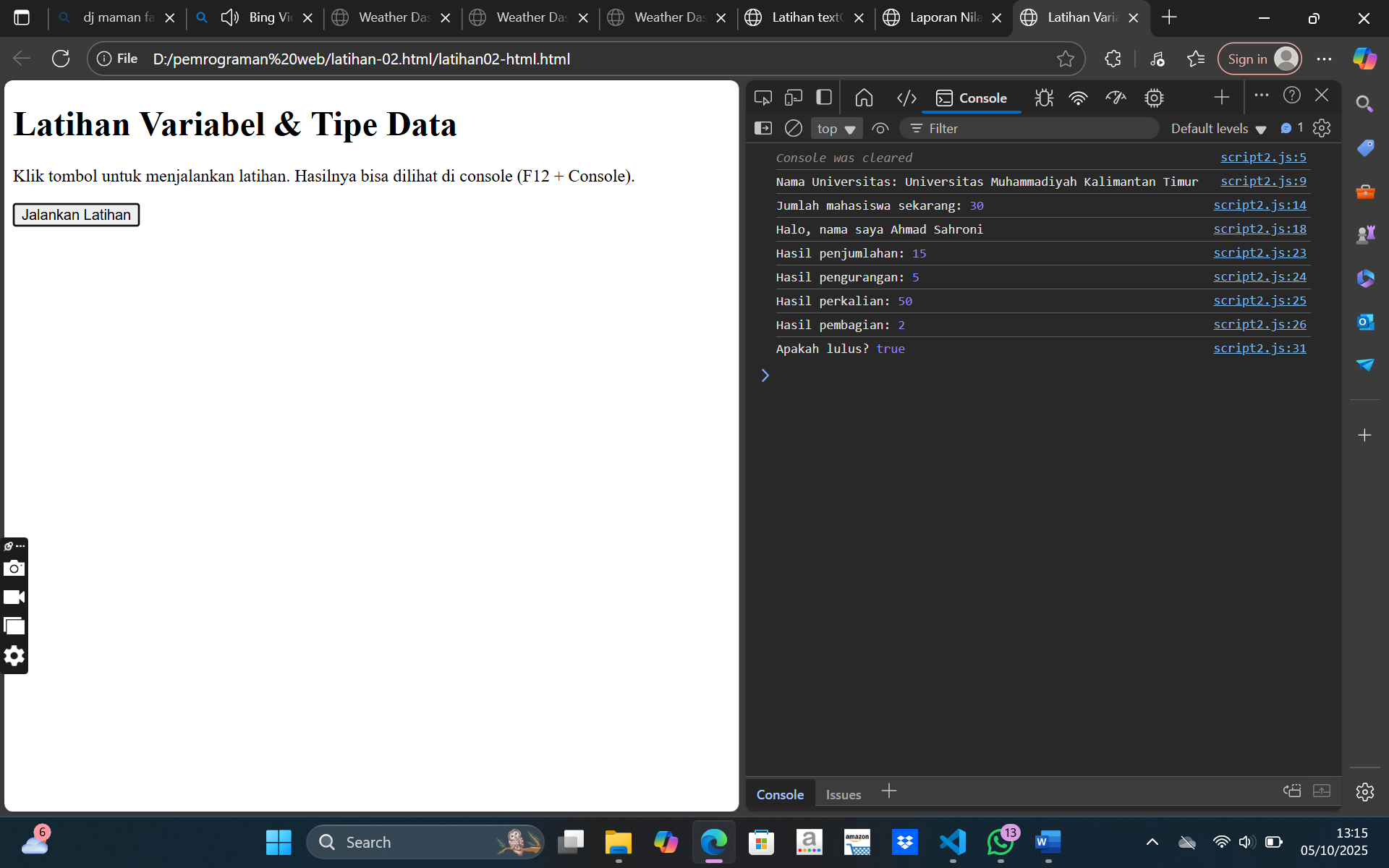
Latihan 2: if...else

Bagian ini berada di luar fungsi runExercises, jadi akan langsung dijalankan otomatis ketika halaman dibuka.

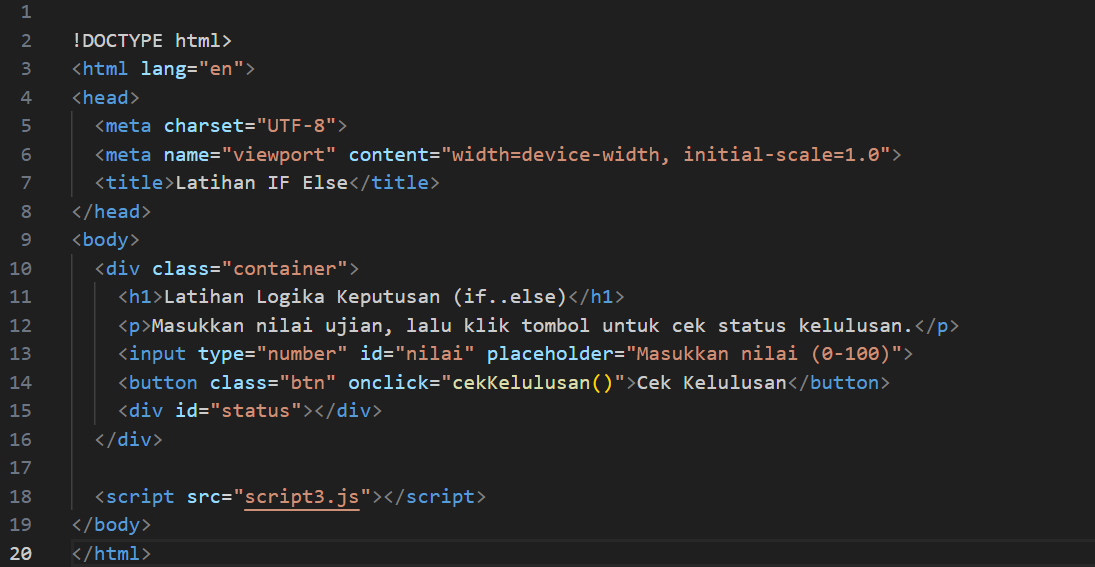
let punyaSIM = true; → variabel boolean, bisa diubah menjadi false untuk menguji.if (punyaSIM === true) → jika benar punya SIM → tampil: “Anda boleh mengemudi.”

Jika false → tampil: “Anda dilarang mengemudi.”

Hasil Web:

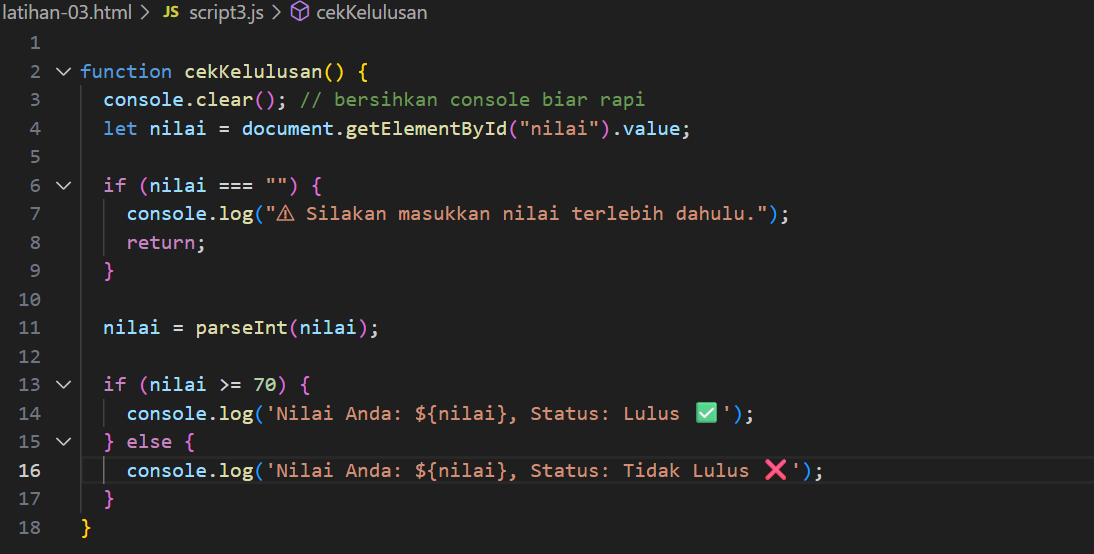


* **Latihan 3 HTML**



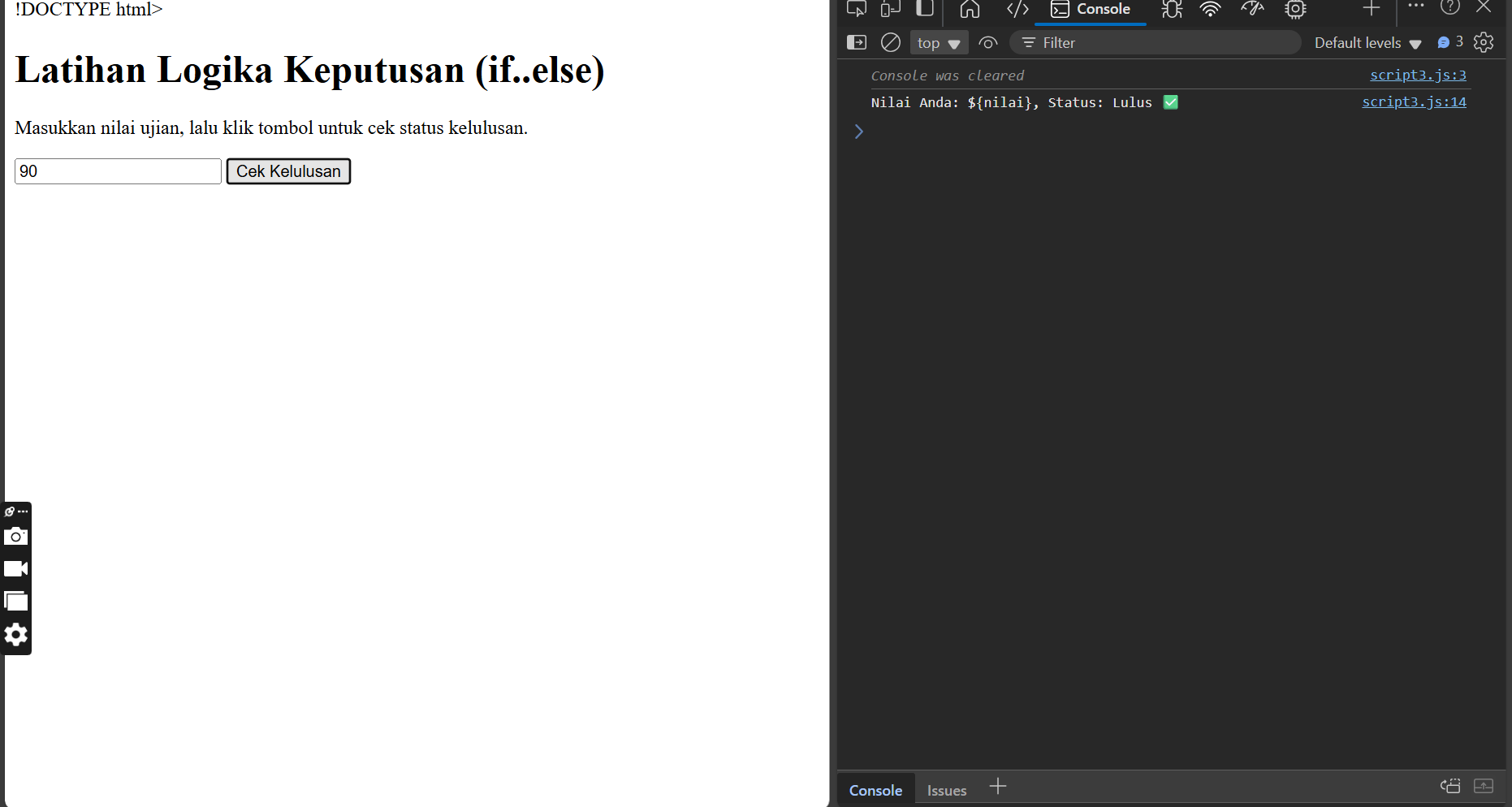
Kode HTML ini membuat halaman web sederhana untuk latihan logika if–else dalam menentukan kelulusan. Bagian <head> berisi pengaturan halaman seperti judul dan pengaturan tampilan agar responsif di semua perangkat. Bagian <body> menampilkan isi utama, yaitu judul, petunjuk, kolom input angka untuk memasukkan nilai, tombol untuk memeriksa kelulusan, dan area kosong untuk menampilkan hasil. Tombol “Cek Kelulusan” terhubung dengan fungsi JavaScript yang akan menampilkan hasil “Lulus” atau “Belum Lulus” di bagian bawah. Struktur ini membentuk tampilan halaman latihan logika keputusan yang sederhana dan mudah dipahami.

* **Latihan 3 JS**

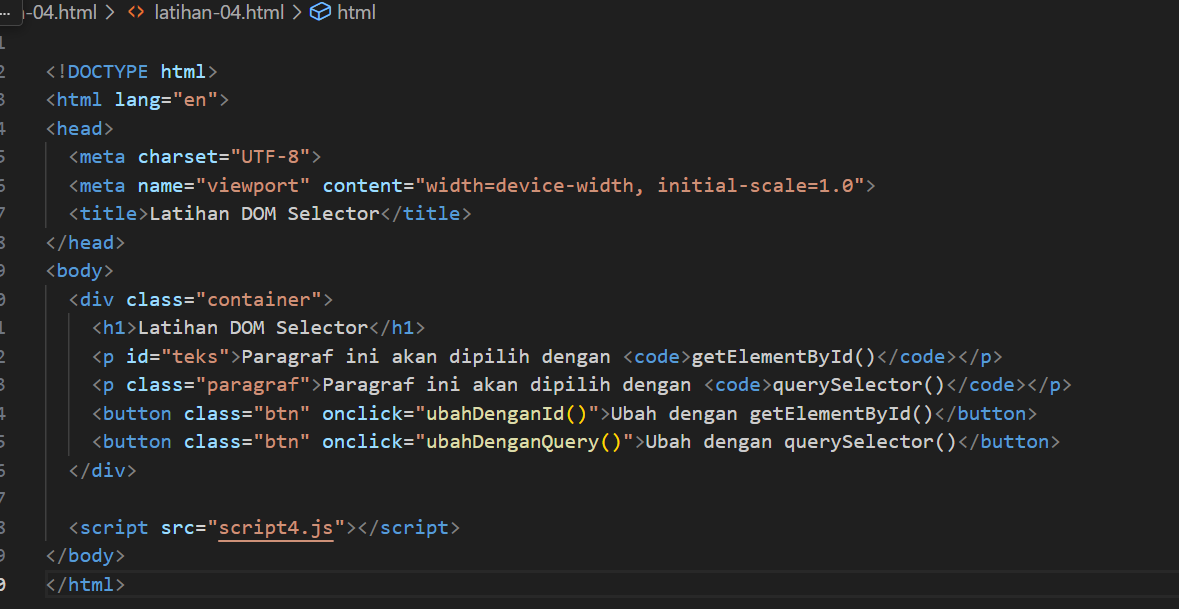


Kode JavaScript ini berfungsi untuk memeriksa status kelulusan berdasarkan nilai yang dimasukkan pengguna. Fungsi cekKelulusan() pertama-tama membersihkan tampilan console dengan console.clear() agar hasil terlihat rapi. Lalu, nilai dari input HTML dengan id "nilai" diambil dan disimpan dalam variabel nilai. Jika kolom masih kosong, program menampilkan pesan peringatan di console dan berhenti dengan return. Setelah itu, nilai diubah menjadi tipe angka menggunakan parseInt() agar bisa dibandingkan secara numerik. Kemudian dilakukan pengecekan menggunakan struktur if–else: jika nilai lebih besar atau sama dengan 70 maka muncul pesan “Lulus ✅”, sedangkan jika kurang dari 70 maka muncul pesan “Tidak Lulus ❌”. Semua hasil ditampilkan di console log.

Hasil Web:

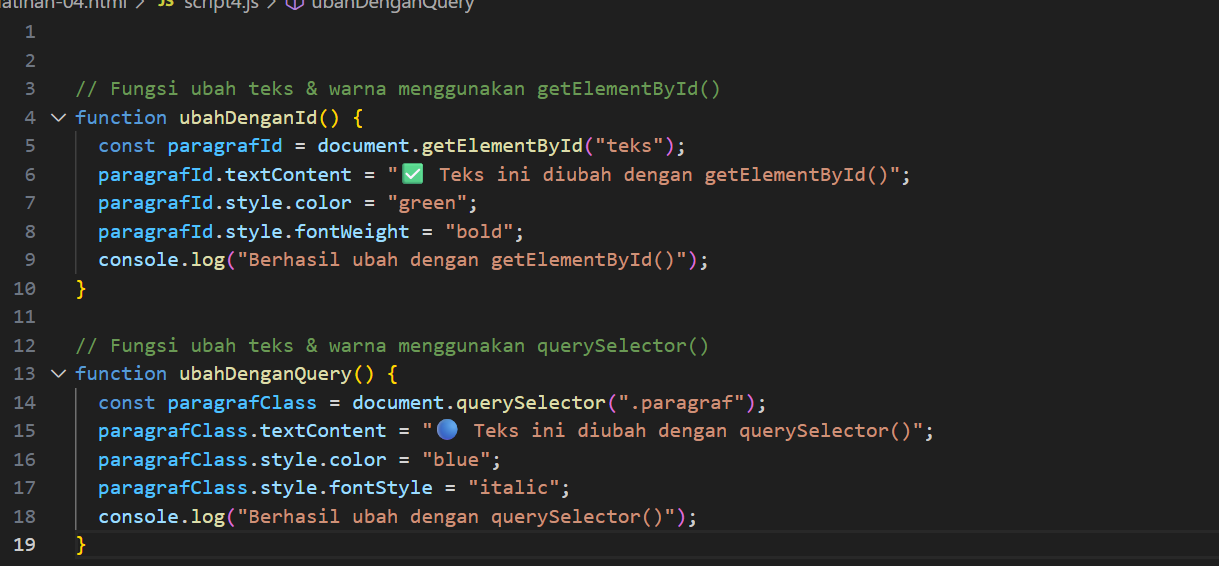


* **Latihan 4 HTMl**



Kode HTML di atas membuat halaman latihan untuk memahami DOM Selector dalam JavaScript. Struktur halaman terdiri dari judul, dua paragraf, dan dua tombol. Paragraf pertama memiliki atribut id="teks" yang akan diakses dengan metode getElementById(), sedangkan paragraf kedua memiliki kelas class="paragraf" yang akan diakses menggunakan querySelector(). Kedua tombol memiliki fungsi onclick yang masing-masing memanggil fungsi ubahDenganId() dan ubahDenganQuery() dari file JavaScript script4.js. Ketika tombol diklik, JavaScript akan memanipulasi teks pada paragraf sesuai dengan selector yang digunakan. Secara keseluruhan, halaman ini digunakan untuk latihan memilih dan mengubah elemen HTML menggunakan dua metode DOM selector berbeda.

* **Latihan 4 JS**



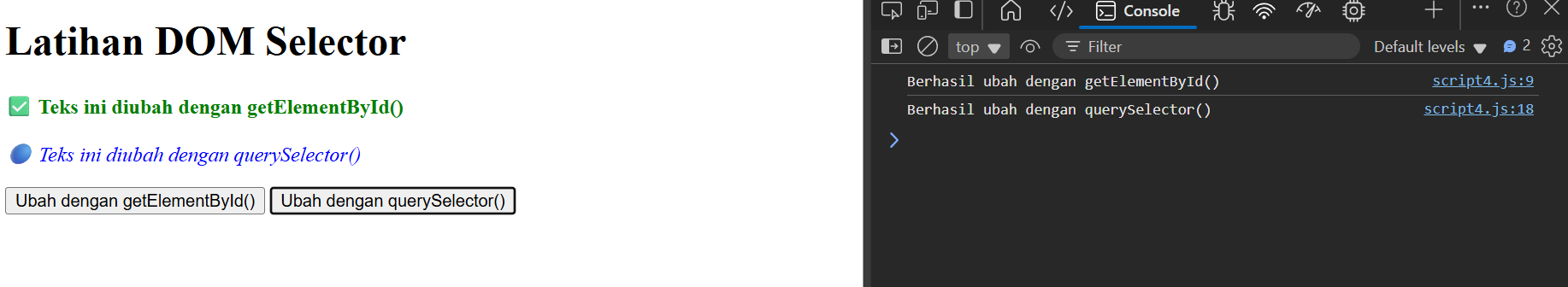
Kode JavaScript ini berfungsi untuk mengubah teks dan gaya elemen HTML menggunakan dua cara berbeda dalam DOM Selector.

Fungsi ubahDenganId() menggunakan document.getElementById("teks") untuk memilih elemen dengan id “teks”. Setelah elemen ditemukan, teksnya diubah menjadi “✅ Teks ini diubah dengan getElementById()”, warnanya diganti menjadi hijau, dan teks dibuat tebal dengan fontWeight = "bold". Pesan keberhasilan juga ditampilkan di console.

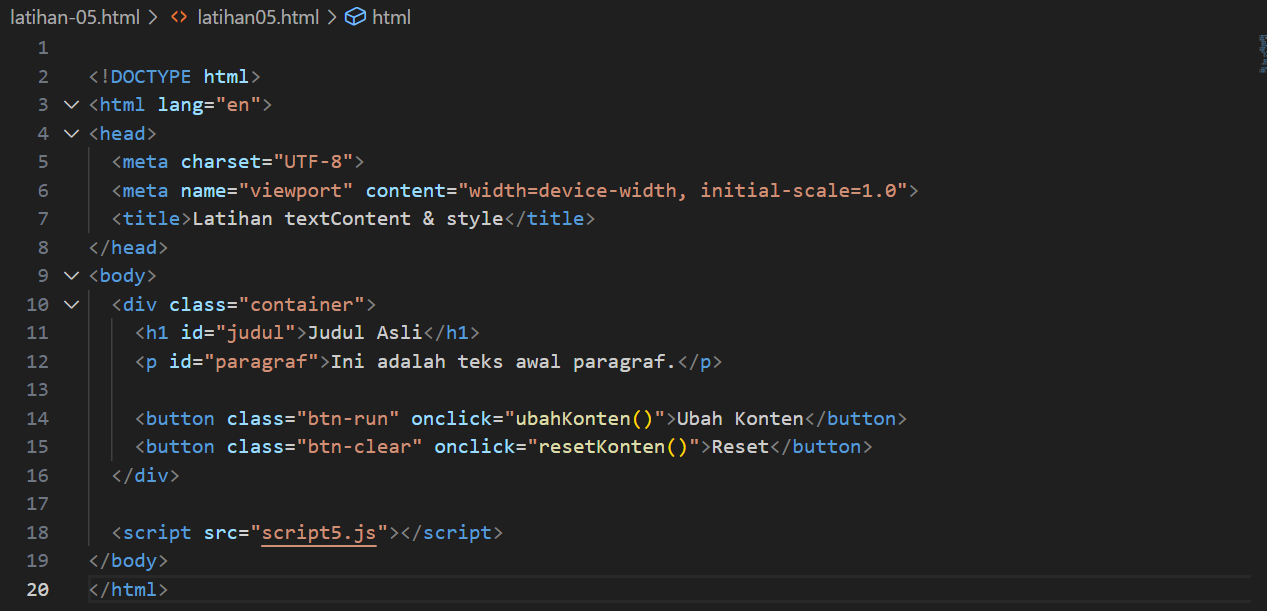
Fungsi ubahDenganQuery() menggunakan document.querySelector(".paragraf") untuk memilih elemen dengan class “paragraf”. Teksnya diubah menjadi “🔵 Teks ini diubah dengan querySelector()”, warnanya diubah menjadi biru, dan teks dibuat miring dengan fontStyle = "italic". Sama seperti sebelumnya, console menampilkan pesan bahwa perubahan berhasil.

Secara keseluruhan, kode ini menunjukkan perbedaan penggunaan dua metode DOM — getElementById() untuk elemen dengan id, dan querySelector() untuk elemen dengan class — keduanya digunakan untuk memanipulasi tampilan halaman secara dinamis.

Hasil Web:

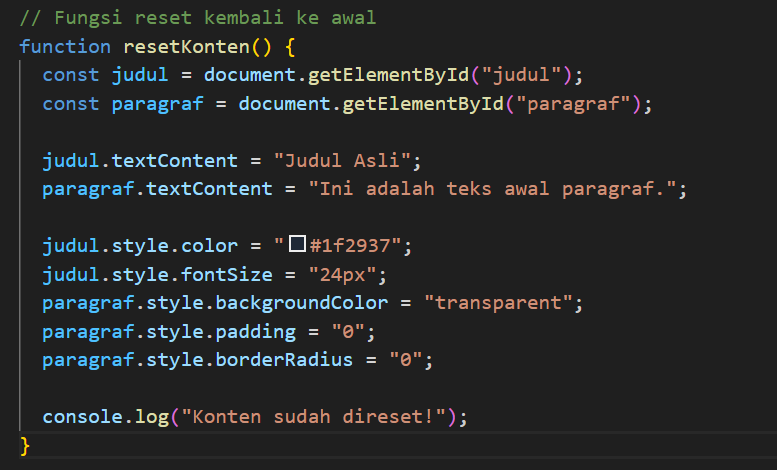
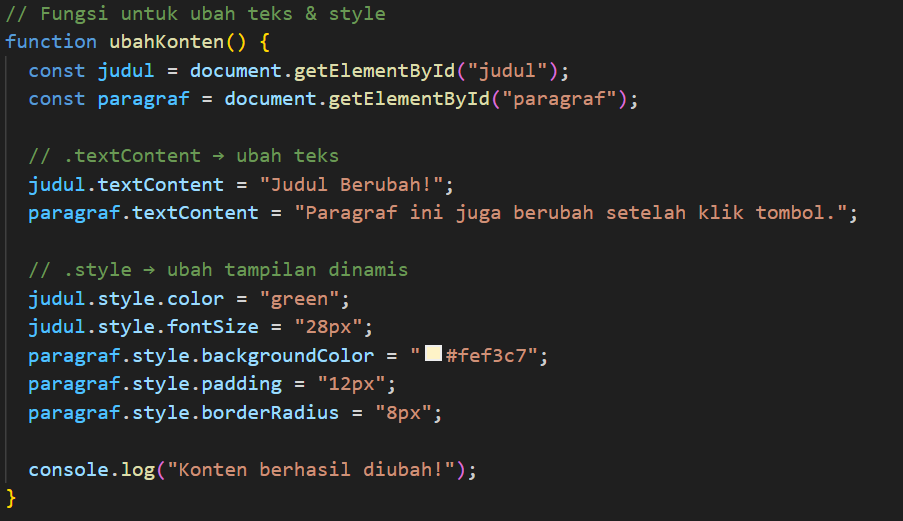


* **Latihan 5 HTML**



Kode HTML ini digunakan untuk latihan mengubah teks dan gaya elemen HTML menggunakan JavaScript. Di dalam <body> terdapat sebuah container berisi judul (<h1 id="judul">) dan paragraf (<p id="paragraf">) yang menampilkan teks awal. Dua tombol disediakan: tombol “Ubah Konten” akan memanggil fungsi ubahKonten() untuk mengganti isi dan gaya teks, sedangkan tombol “Reset” akan memanggil fungsi resetKonten() untuk mengembalikannya ke keadaan semula. Kode JavaScript untuk menjalankan fungsi tersebut terdapat di file eksternal script5.js. Secara keseluruhan, halaman ini menjadi latihan dasar manipulasi elemen menggunakan properti textContent dan style dalam JavaScript.

* **Latihan 5 JS**



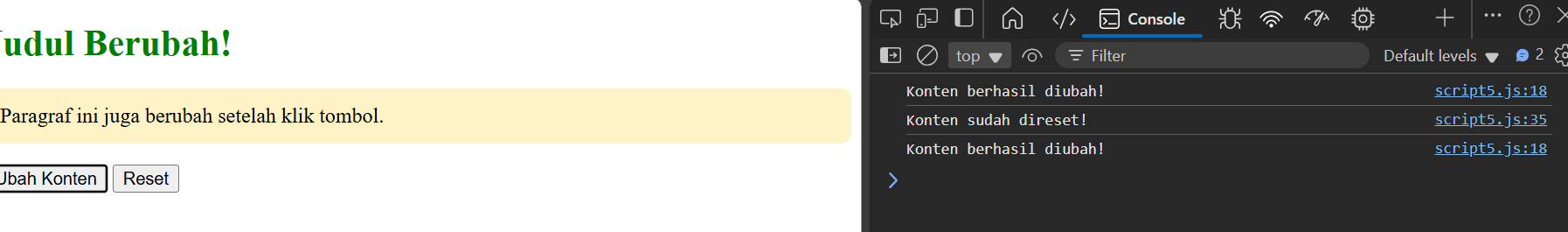
Kode JavaScript ini berfungsi untuk mengubah dan mengembalikan teks serta gaya elemen HTML secara dinamis.

Fungsi ubahKonten() pertama mengambil elemen <h1> dan <p> menggunakan document.getElementById(). Lalu, properti .textContent digunakan untuk mengubah teks menjadi “Judul Berubah!” dan “Paragraf ini juga berubah setelah klik tombol.” Setelah itu, bagian .style dipakai untuk memodifikasi tampilan: warna teks judul menjadi hijau, ukuran huruf lebih besar, latar belakang paragraf berubah warna krem (#fef3c7), diberi jarak dalam (padding), dan sudutnya dibulatkan. Semua perubahan ini ditampilkan di console sebagai pesan “Konten berhasil diubah!”.

Fungsi resetKonten() mengembalikan semua elemen ke kondisi semula: teks kembali ke “Judul Asli” dan “Ini adalah teks awal paragraf.”, warna dan ukuran teks dikembalikan seperti semula, serta gaya paragraf (warna latar, padding, dan border radius) dihapus. Terakhir, console menampilkan pesan “Konten sudah direset!”.

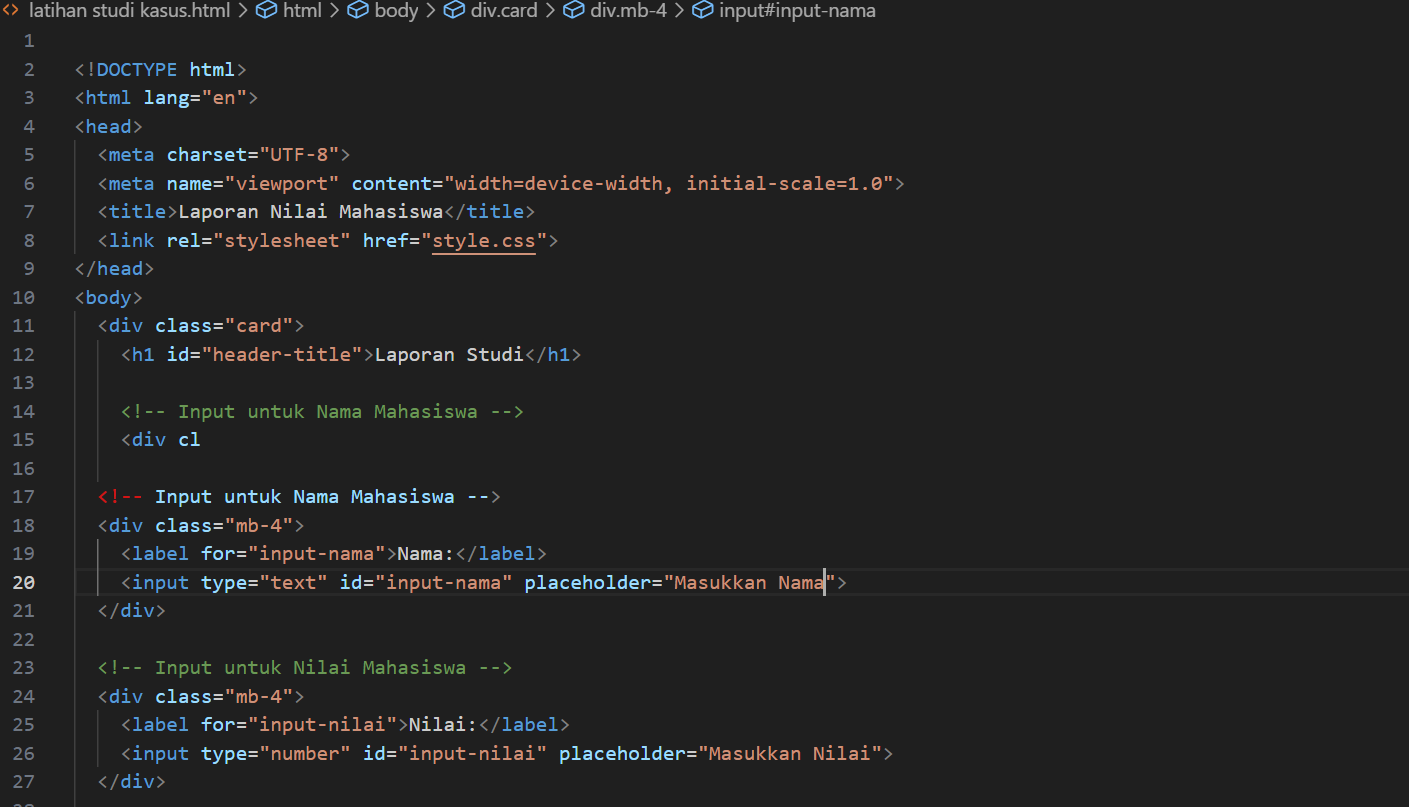
Secara keseluruhan, kode ini menunjukkan cara menggunakan textContent untuk mengubah isi teks dan style untuk memodifikasi tampilan elemen HTML melalui JavaScript.

Hasil Web:



* **Soal Latihan Studi Kasus Laporan Studi**

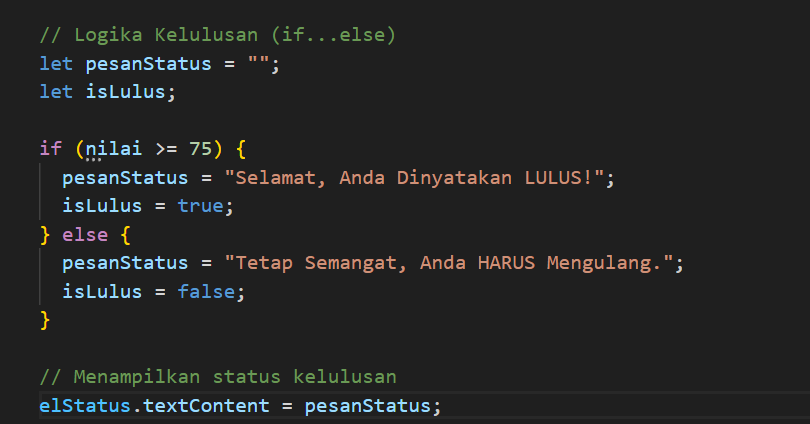
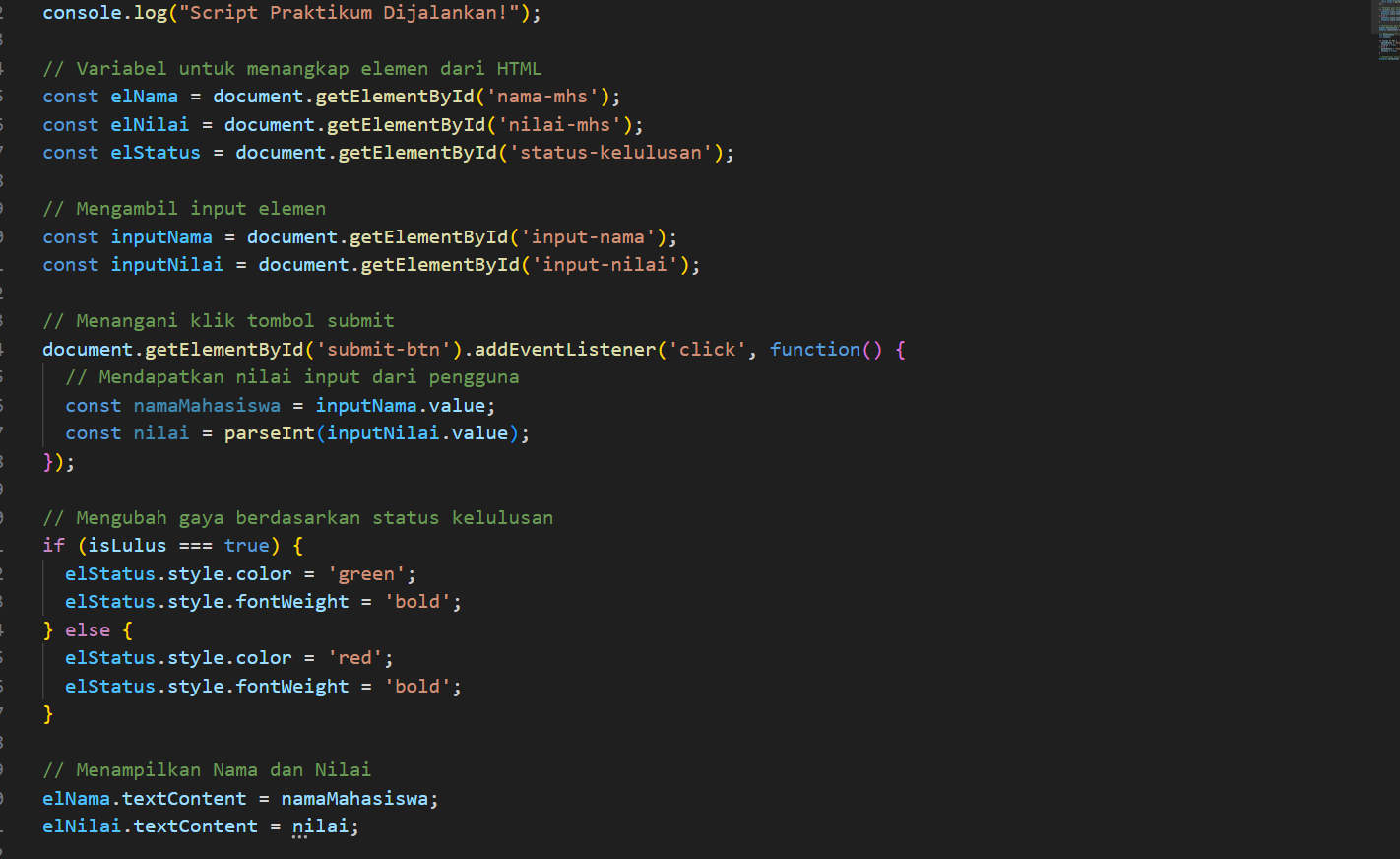
**-code HTML**

****

Bagian <head> berisi pengaturan dasar halaman seperti judul dan link ke file CSS eksternal style.css untuk memperindah tampilan. Di dalam <body>, terdapat sebuah container <div class="card"> yang berisi elemen-elemen utama:

Judul halaman (<h1 id="header-title">) menampilkan teks “Laporan Studi”.Input nama mahasiswa (<input type="text" id="input-nama">) digunakan untuk menulis nama.Input nilai mahasiswa (<input type="number" id="input-nilai">) digunakan untuk memasukkan nilai ujian.Tombol Submit (<button id="submit-btn">) digunakan untuk memproses data ketika diklik.Setelah area input, terdapat garis pemisah <hr> dan bagian untuk menampilkan hasil:<p> dengan <span id="nama mhs"> dan <span id="nilai mhs"> akan menampilkan nama serta nilai yang dimasukkan pengguna.<p id="status-kelulusan"> menampilkan status kelulusan, misalnya “Lulus” atau “Tidak Lulus”, yang akan diubah lewat JavaScript.Di bagian paling bawah, file soal.js dihubungkan menggunakan tag <script> untuk menangani logika interaktif seperti membaca input dan menampilkan hasil.

**-Code JS**

****

Kode ini digunakan untuk memproses data nama dan nilai mahasiswa, lalu menentukan status kelulusan berdasarkan input yang diberikan di halaman HTML sebelumnya.

Pertama, console.log("Script Praktikum Dijalankan!") memastikan bahwa file JavaScript sudah aktif. Kemudian beberapa variabel dibuat untuk mengambil elemen dari HTML menggunakan document.getElementById(), seperti elNama, elNilai, dan elStatus untuk menampilkan hasil, serta inputNama dan inputNilai untuk membaca data dari input pengguna.

Sebuah event listener ditambahkan pada tombol dengan id "submit-btn" menggunakan addEventListener('click', function() {...}), yang akan menjalankan perintah setiap kali tombol diklik. Di dalamnya, nilai nama dan angka dari input diambil menggunakan .value, lalu nilai angka diubah ke tipe numerik dengan parseInt().

Selanjutnya, logika if–else digunakan untuk menentukan kelulusan:

Jika nilai >= 75, maka variabel pesanStatus diisi dengan “Selamat, Anda Dinyatakan LULUS!” dan isLulus diset ke true.

Jika kurang dari 75, maka pesanStatus berisi “Tetap Semangat, Anda HARUS Mengulang.” dan isLulus diset ke false.Kemudian hasilnya ditampilkan ke halaman dengan mengubah teks pada elemen HTML:

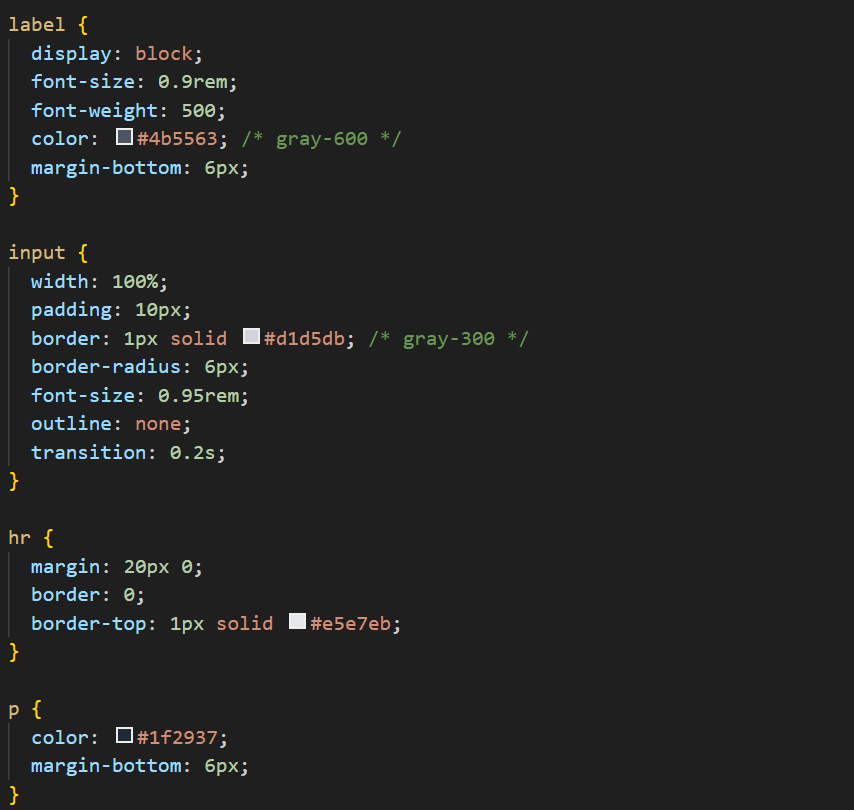
elNama.textContent menampilkan nama mahasiswa,

elNilai.textContent menampilkan nilai,

elStatus.textContent menampilkan pesan status kelulusan.

Bagian terakhir mengubah warna teks status secara visual: jika lulus maka warna hijau, jika tidak lulus maka merah, keduanya dengan gaya huruf tebal.

**-Code CSS**

****

Kode CSS ini berfungsi untuk mengatur tampilan halaman Laporan Nilai Mahasiswa agar terlihat bersih, modern, dan rapi.

Pertama, bagian reset dasar (\*) menghapus jarak bawaan (margin, padding) dan mengatur semua elemen agar memakai sistem tata letak yang konsisten (box-sizing: border-box) serta font standar sans-serif. Pada body, latar belakang diberi warna abu muda (#f3f4f6), dan isi halaman diposisikan di tengah secara vertikal dan horizontal menggunakan flex.

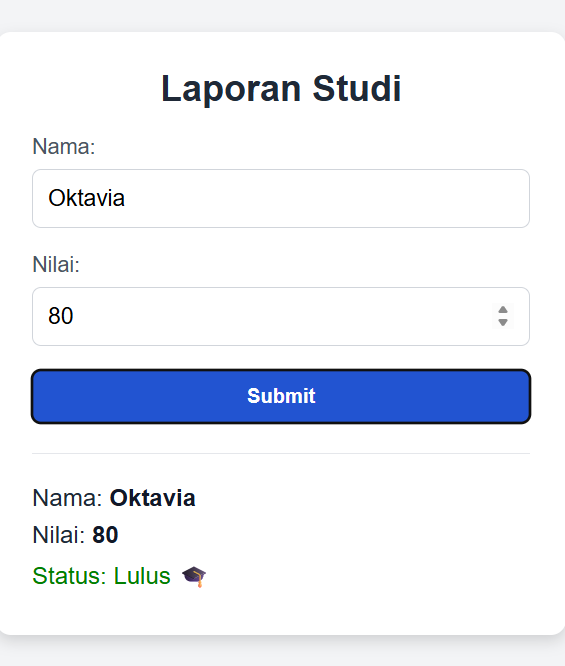
Bagian .card adalah kotak utama tempat semua konten berada. Kotak ini memiliki warna putih, sudut melengkung, bayangan halus, serta ukuran lebar 380px agar tampak seperti kartu laporan.

Elemen judul (h1) diatur agar teksnya rata tengah, berwarna abu tua, dan memiliki jarak bawah agar tidak menempel pada elemen lain. Label dan input diberi gaya agar mudah dibaca: label tampil di atas input dengan warna abu lembut, sementara input memiliki batas abu terang, sudut membulat, dan efek fokus biru (border-color dan box-shadow) ketika diklik.

Kelas .mb-4 menambahkan jarak antar bagian form, sedangkan button diberi warna biru (#2563eb), teks putih tebal, serta efek hover yang membuat warnanya lebih gelap untuk memberi kesan interaktif.

Elemen p dan span diatur untuk menampilkan hasil dengan teks rapi dan penekanan tebal pada nilai mahasiswa. Garis <hr> digunakan sebagai pembatas visual antar bagian dengan warna abu muda.

Hasil Web:



* **Latihan Javacript (Cuaca)**

-code HTML



Halaman ini merupakan antarmuka sederhana aplikasi cuaca (Weather Dashboard).

Bagian head berisi pengaturan dasar seperti jenis karakter (UTF-8), pengaturan tampilan agar responsif di berbagai perangkat, judul halaman di tab browser, serta tautan ke file CSS yang mengatur tampilannya.

Bagian body menampilkan sebuah kotak (card) yang berisi judul aplikasi, teks petunjuk agar pengguna memasukkan nama kota, kolom input untuk mengetik nama kota (misalnya “Samarinda”), dan sebuah tombol untuk menampilkan informasi cuaca.

Terakhir, di bagian bawah halaman terdapat tautan ke file JavaScript eksternal (script.js) yang berfungsi untuk menjalankan logika aplikasi—misalnya mengambil data cuaca dari API berdasarkan kota yang dimasukkan pengguna dan menampilkannya di halaman.

Secara keseluruhan, struktur ini adalah kerangka utama dari aplikasi cuaca berbasis web yang menggabungkan HTML untuk tampilan, CSS untuk gaya visual, dan JavaScript untuk fungsi interaktifnya

-code JS



Kode ini membuat simulasi aplikasi cuaca sederhana tanpa harus mengambil data dari internet (API). Semua informasi cuaca disimpan secara lokal di dalam program.

1. Penyimpanan Data Cuaca (Simulasi)

Objek weatherData berisi daftar kota sebagai key (Jakarta, Bandung, Surabaya, Samarinda, Medan),

dan setiap kota memiliki dua properti:

-temperature → suhu dalam derajat Celsius

-condition → kondisi cuaca (misalnya “Cerah”, “Mendung”, “Hujan Ringan”)

Fungsinya seperti database kecil untuk menampung data cuaca.

2. Fungsi Utama showWeather()

Fungsi ini dijalankan saat tombol “Tampilkan Cuaca” diklik.

Langkah-langkahnya:

1. Mengambil nilai kota dari kotak input dengan getElementById("cityInput").

2. Menghapus spasi berlebih dengan .trim().

3. Mencari data kota tersebut di dalam objek weatherData.

4. Jika ditemukan:

Menampilkan nama kota, suhu, dan kondisi cuaca di console browser.

5. Jika tidak ditemukan:

Menampilkan pesan “Data cuaca untuk kota ... tidak ditemukan.”

Artinya, jika pengguna mengetik “Samarinda”, konsol akan menampilkan:

Cuaca di Samarinda:

Suhu: 28°C

Kondisi: Mendung

3. Event Listener DOMContentLoaded

Baris ini memastikan bahwa kode baru dijalankan setelah seluruh halaman selesai dimuat.

Ketika halaman sudah siap:

Kode mencari tombol dengan ID btnCuaca.Lalu menambahkan event listener agar saat tombol diklik, fungsi showWeather() dijalankan.

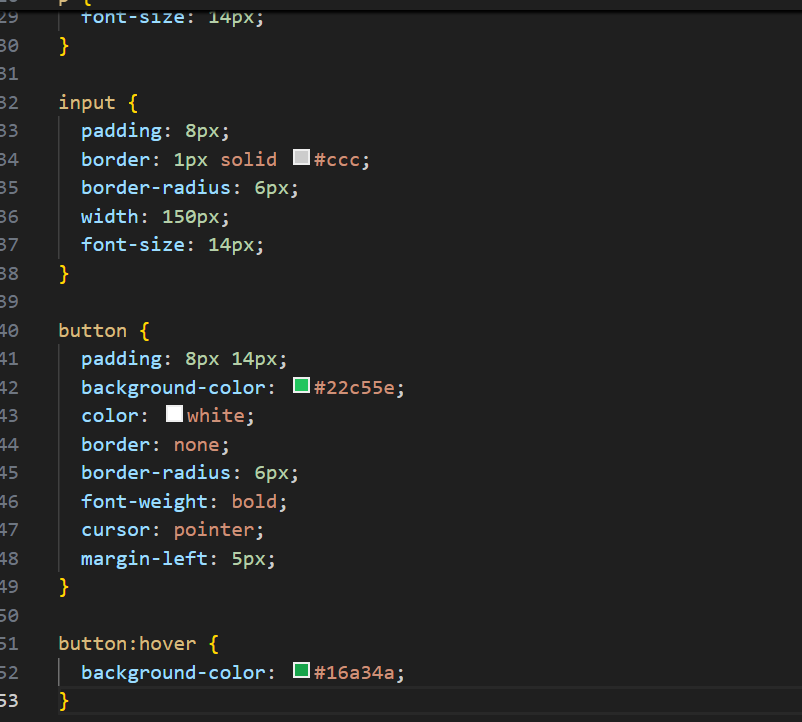
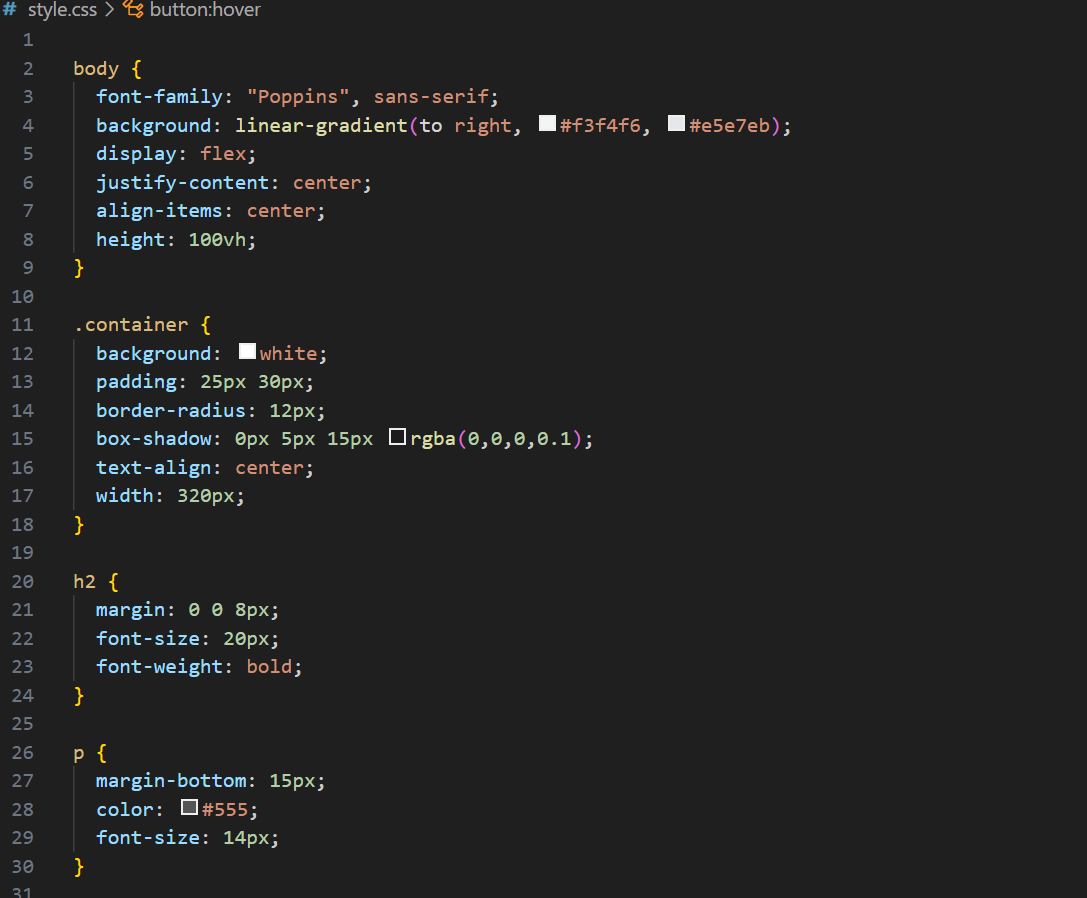
4. Kesimpulan Fungsi Program

Secara keseluruhan:

Program ini membaca nama kota dari input pengguna.

Mencocokkannya dengan daftar cuaca yang sudah disiapkan.

-code css



CSS tersebut digunakan untuk mempercantik tampilan halaman Weather Dashboard agar terlihat rapi dan modern.

Bagian reset dasar menghapus jarak bawaan elemen dan menyamakan ukuran agar mudah diatur.

Bagian body membuat latar belakang abu-abu muda dan menempatkan konten di tengah layar.

Card berfungsi sebagai wadah utama dengan warna putih, sudut membulat, dan bayangan lembut agar tampak menonjol.

Input didesain lebar, memiliki sudut halus, dan berubah warna biru saat aktif untuk menandakan fokus pengguna.

Tombol berwarna biru dengan teks putih, efek hover lebih gelap, serta transisi halus untuk kesan interaktif.

Elemen lain seperti judul, label, dan teks diberi ukuran dan warna abu-abu yang seimbang supaya mudah dibaca dan konsisten

-Hasil Web

