**HASIL TUGAS CLIENT SIDE**

**PEMROGRAMAN BERBASIS WEB**

1. **Materi**

Dalam web development, client-side berarti segala sesuatu yang dilakukan atau dijalankan di sisi pengguna (komputer pengguna).

Client-side Scripting adalah bahasa pemrograman web yang pengolahan datanya dilakukan oleh komputer pengguna/pengunjung. Jadi, ketika seseorang berkunjung ke sebuah web, komputernya akan mendownload data/script yang bersifat client-side di web tersebut.

Client-side script meliputi kode-kode yang ditampilkan ketika anda mengklik kanan pada sebuah halaman web dan melihat sumber halaman (View Page Source). Contoh dari Client-side Script adalah HTML, CSS, JavaScript dan XML.

**Dasar Teori**

Cara menyambungkan Javascript ke HTML

Untuk menyambungkan JavaScript ke dalam dokumen HTML, Anda dapat menggunakan tag script di dalam elemen head atau body dokumen HTML. Berikut adalah contoh kode dasar:

Masukkan kode JavaScript di dalam tag script: html

Copy code

<!DOCTYPE html>

<html>

<head>

<title>Contoh Penyambungan Javascript ke HTML</title>

<script>

// Masukkan kode JavaScript di sini

</script>

</head>

<body>

<!-- Isi dokumen HTML di sini -->

</body>

</html>

Sambungkan file JavaScript eksternal dengan menggunakan atribut src di dalam tag script:

html

Copy code

<!DOCTYPE html>

<html>

<head>

<title>Contoh Penyambungan Javascript ke HTML</title>

<script src="script.js"></script>

</head>

<body>

<!-- Isi dokumen HTML di sini -->

</body>

</html>

Pastikan untuk menempatkan file JavaScript eksternal pada direktori yang benar dan menyesuaikan nama file dengan benar. Jika Anda ingin menggunakan framework JavaScript tertentu seperti React atau Vue, Anda dapat menggunakan cara yang lebih spesifik untuk menyambungkan JavaScript ke dalam dokumen HTML.

# Deklarasi Variabel

* 1. **Var**

Deklarasi variabel menggunakan kata kunci var pada JavaScript digunakan untuk membuat variabel dan memberi tahu interpreter JavaScript bahwa kita akan menggunakan nama variabel tersebut. Variabel dapat berisi nilai yang berbeda selama program dijalankan.

Berikut adalah contoh penggunaan deklarasi variabel menggunakan var:

javascript Copy code

var nama = "John"; var umur = 25;

var sudahMenikah = false;

console.log(nama); // output: John console.log(umur); // output: 25 console.log(sudahMenikah); // output: false

Dalam contoh di atas, kita membuat tiga variabel: nama, umur, dan sudahMenikah. nama adalah variabel bertipe data string, umur adalah variabel bertipe data number, dan sudahMenikah adalah variabel bertipe data boolean.

Variabel yang dideklarasikan dengan var dapat diakses dan dimodifikasi di mana saja dalam program JavaScript. Variabel yang dideklarasikan tanpa nilai akan memiliki nilai undefined secara default.

javascript Copy code var a;

console.log(a); // output: undefined

Namun, perlu diingat bahwa variabel yang dideklarasikan dengan var memiliki scope fungsi. Artinya, jika variabel dideklarasikan di dalam fungsi, variabel

javascript Copy code

var a;

console.log(a); // output: undefined

Namun, perlu diingat bahwa variabel yang dideklarasikan dengan var memiliki scope fungsi. Artinya, jika variabel dideklarasikan di dalam fungsi, variabel tersebut hanya dapat diakses dari dalam fungsi tersebut.

javascript Copy code function tes() { var b = 2;

}

console.log(b); // output: error

Dalam contoh di atas, variabel b hanya dapat diakses dari dalam fungsi tes(). Jika kita mencoba mengakses variabel b dari luar fungsi, maka akan muncul error.

# Let

let adalah kata kunci dalam JavaScript yang digunakan untuk mendeklarasikan variabel yang nilainya dapat diubah atau diinisialisasi ulang di masa depan. Variabel yang dideklarasikan dengan let memiliki scope block-level, yang berarti bahwa variabel hanya dapat diakses di dalam blok di mana variabel tersebut dideklarasikan.

Berikut adalah contoh penggunaan let untuk deklarasi variabel:

javascript Copy code

let nama = "John"; nama = "Doe";

console.log(nama); // output: Doe

Dalam contoh di atas, kita membuat variabel nama menggunakan let. Setelah itu, kita mengubah nilainya menjadi "Doe". Kita dapat melakukannya karena variabel yang dideklarasikan menggunakan let dapat diubah nilainya.

Variabel yang dideklarasikan dengan let hanya dapat diakses di dalam blok di mana variabel tersebut dideklarasikan. Blok adalah bagian dari kode yang diapit oleh kurung kurawal {}. Sebagai contoh:

javascript Copy code let x = 10; if (x > 5) { let y = 5;

console.log(x); // output: 10 console.log(y); // output: 5

}

console.log(x); // output: 10 console.log(y); // output: error

Dalam contoh di atas, variabel x dideklarasikan menggunakan const di dalam blok if. Variabel x hanya dapat diakses di dalam blok if tersebut, sehingga jika kita mencoba mengakses variabel x di luar blok if, maka akan muncul error.

# Function dan Kondisi

Fungsi di JavaScript

Fungsi adalah blok kode yang dapat digunakan kembali dalam program. Dalam JavaScript, fungsi dapat didefinisikan dengan menggunakan kata kunci function. Berikut adalah contoh fungsi sederhana yang menerima dua argumen dan mengembalikan hasil penjumlahannya:

javascript Copy code

function tambah(a, b) { return a + b;

}

Dalam contoh di atas, kita mendefinisikan fungsi tambah dengan dua argumen a dan b. Fungsi ini mengembalikan hasil penjumlahan dari dua argumen tersebut.

Kita dapat memanggil fungsi tambah dengan memberikan dua argumen:

javascript Copy code

let hasil = tambah(2, 3); console.log(hasil); // output: 5

Dalam contoh di atas, kita memanggil fungsi tambah dengan memberikan dua argumen 2 dan 3. Hasil penjumlahan dari dua argumen tersebut kemudian disimpan dalam variabel hasil, yang kemudian dicetak menggunakan console.log.

Kondisi di JavaScript

Kondisi adalah bagian dari program yang menentukan apakah blok kode tertentu akan dijalankan atau tidak, berdasarkan kondisi yang diberikan. Dalam JavaScript, kita dapat menggunakan beberapa kata kunci untuk membuat kondisi:

if

if adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika kondisi tertentu terpenuhi. Berikut adalah contoh penggunaan if:

javascript Copy code let x = 10; if (x > 5) {

console.log("x lebih besar dari 5");

}

Dalam contoh di atas, kita menggunakan if untuk mengecek apakah nilai x lebih besar dari 5. Jika kondisi terpenuhi, maka blok kode di dalam kurung kurawal akan dieksekusi.

else

else adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika kondisi pada if tidak terpenuhi. Berikut adalah contoh penggunaan if-else:

javascript

Copy code let x = 3; if (x > 5) {

console.log("x lebih besar dari 5");

} else {

console.log("x kurang dari atau sama dengan 5");

Dalam contoh di atas, kita menggunakan if-else untuk mengecek apakah nilai x lebih besar dari 5. Jika kondisi pada if tidak terpenuhi, maka blok kode di dalam else akan dieksekusi.

else if

else if adalah kata kunci yang digunakan untuk mengeksekusi blok kode jika beberapa kondisi sebelumnya tidak terpenuhi. Berikut adalah contoh penggunaan if-else if:

javascript Copy code let x = 3; if (x > 5) {

console.log("x lebih besar dari 5");

} else if (x > 2) {

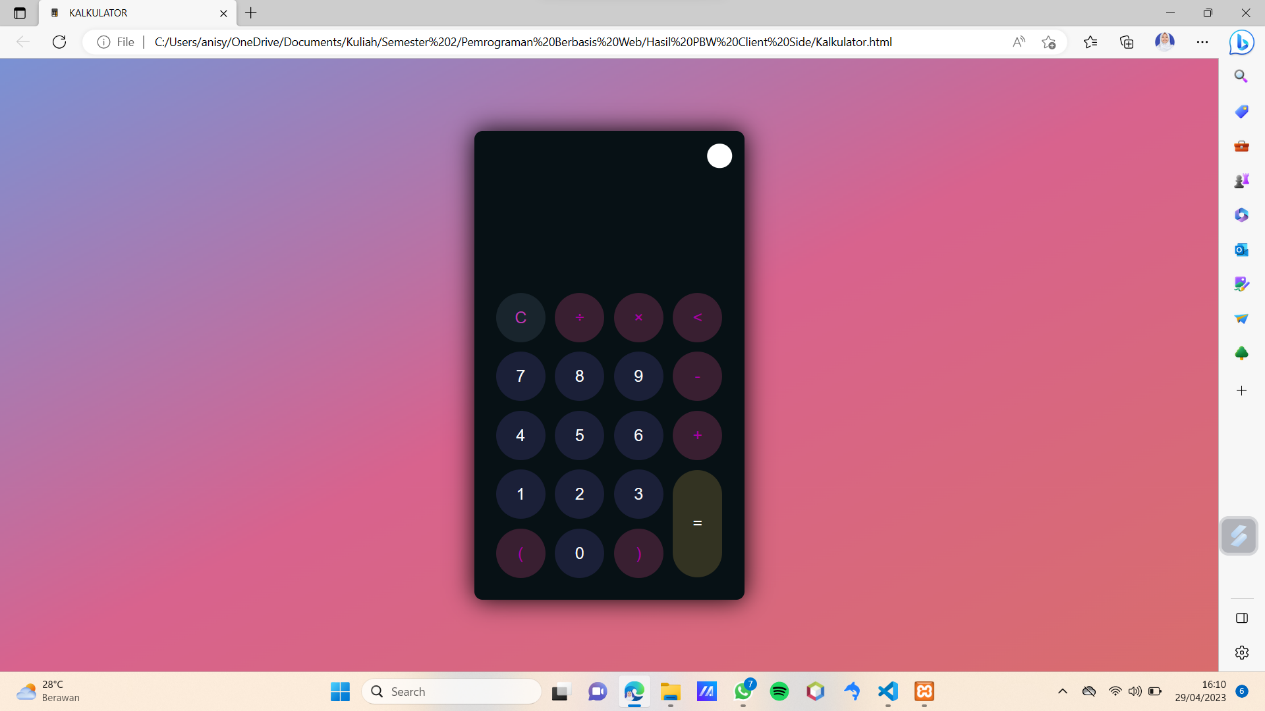
console.log("x lebih besar dari 2 dan kurang dari atau sama dengan 5");

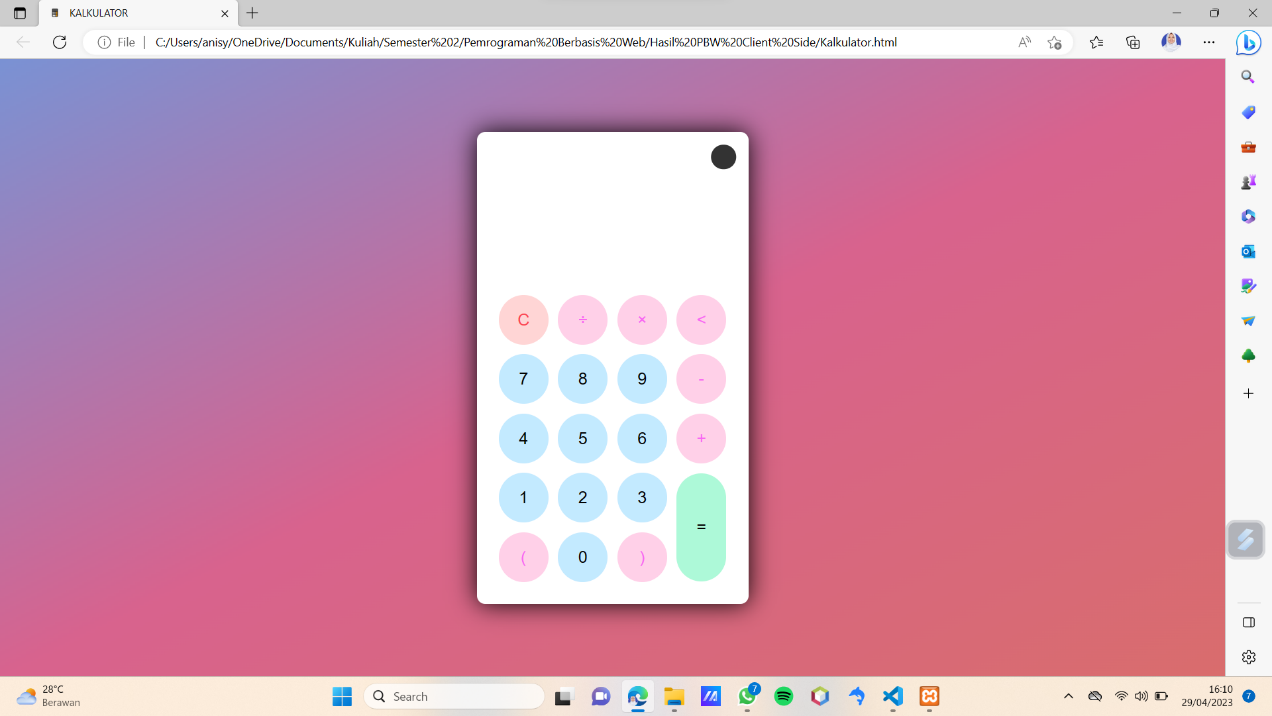
} else {

console.log("x kurang dari atau sama dengan 2");

}

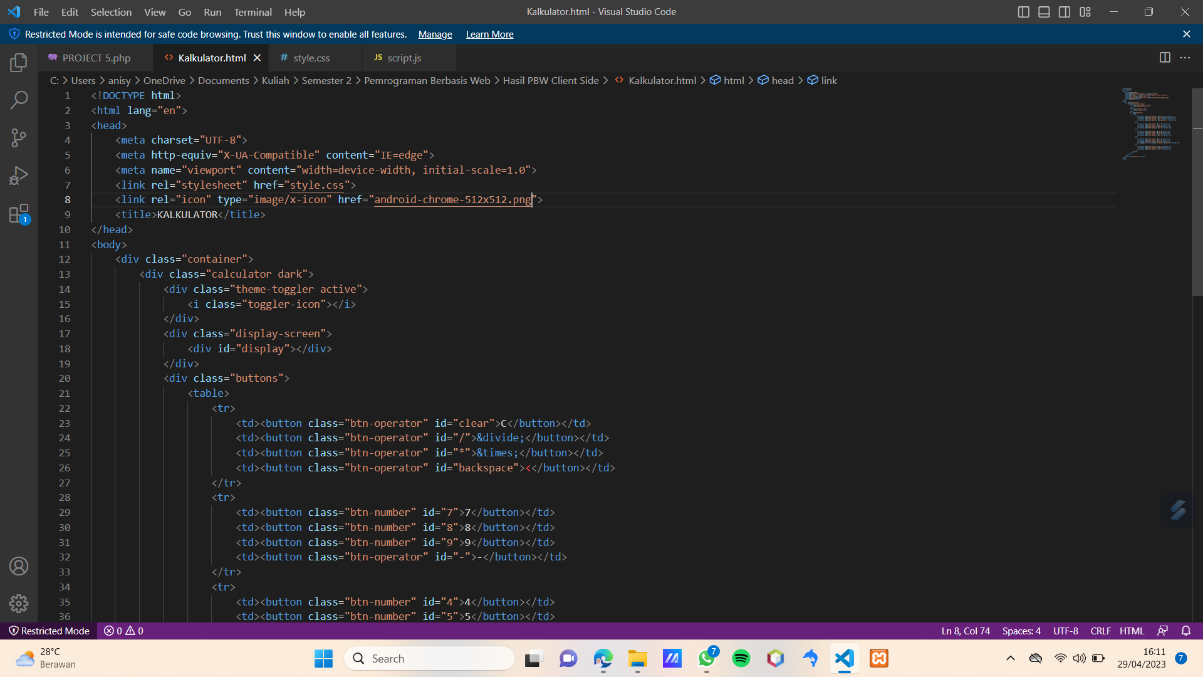
1. **Hasil**

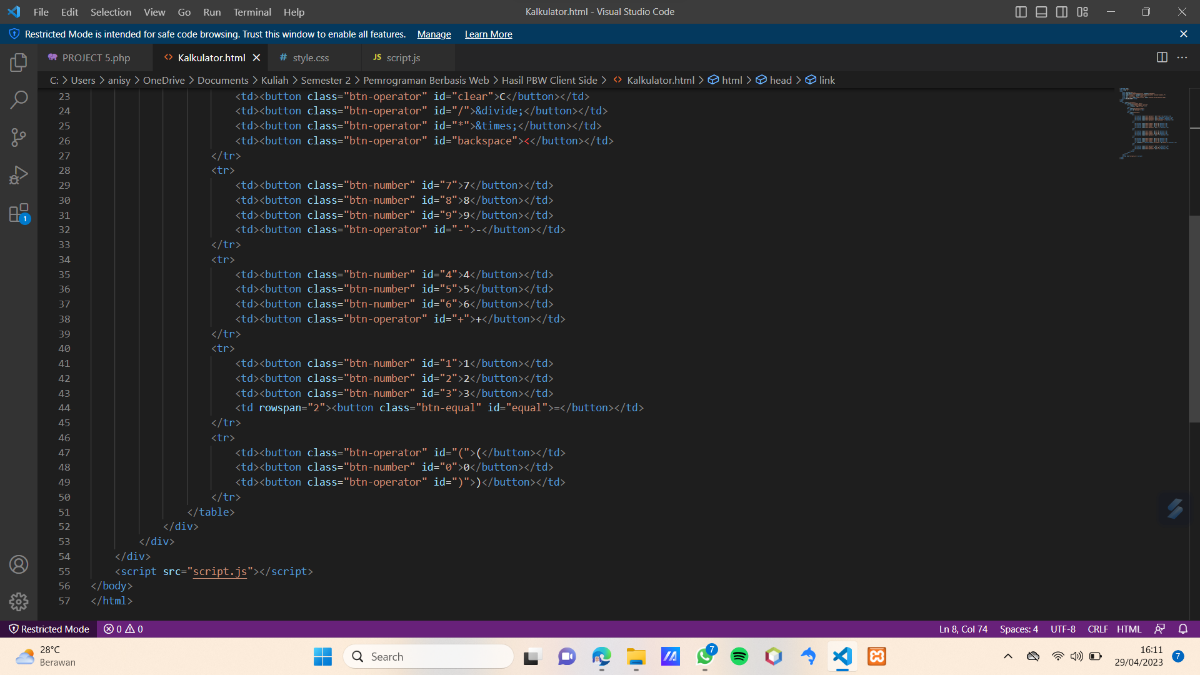
****

****

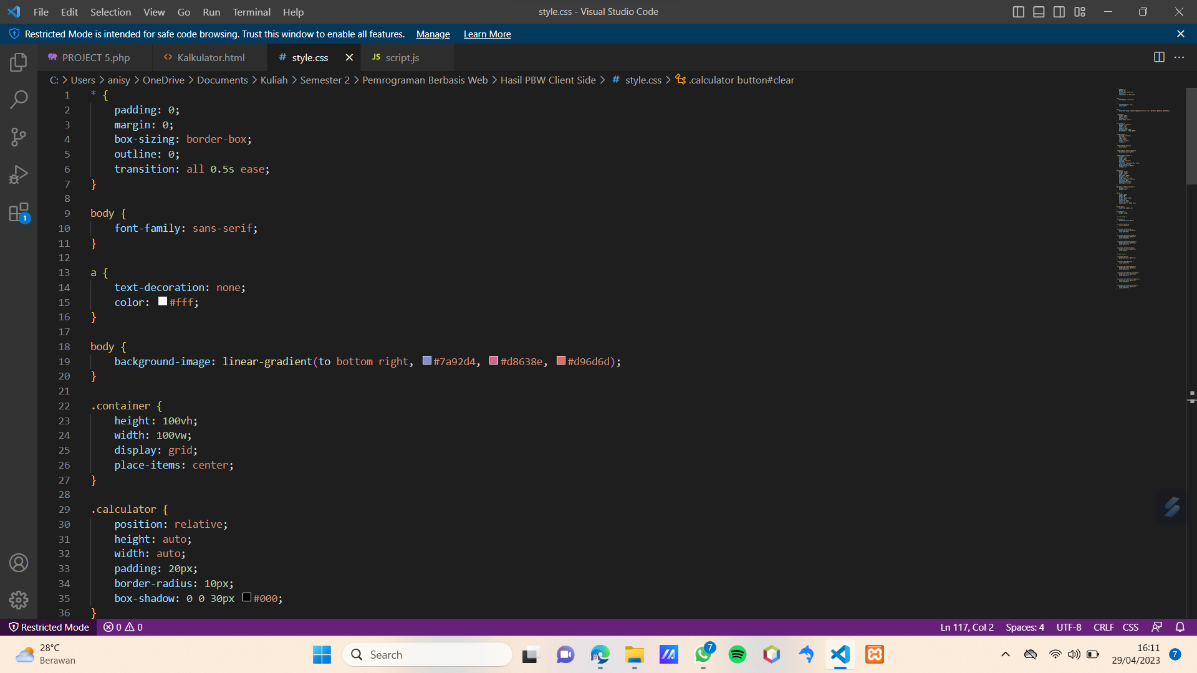
1. **Source Code**

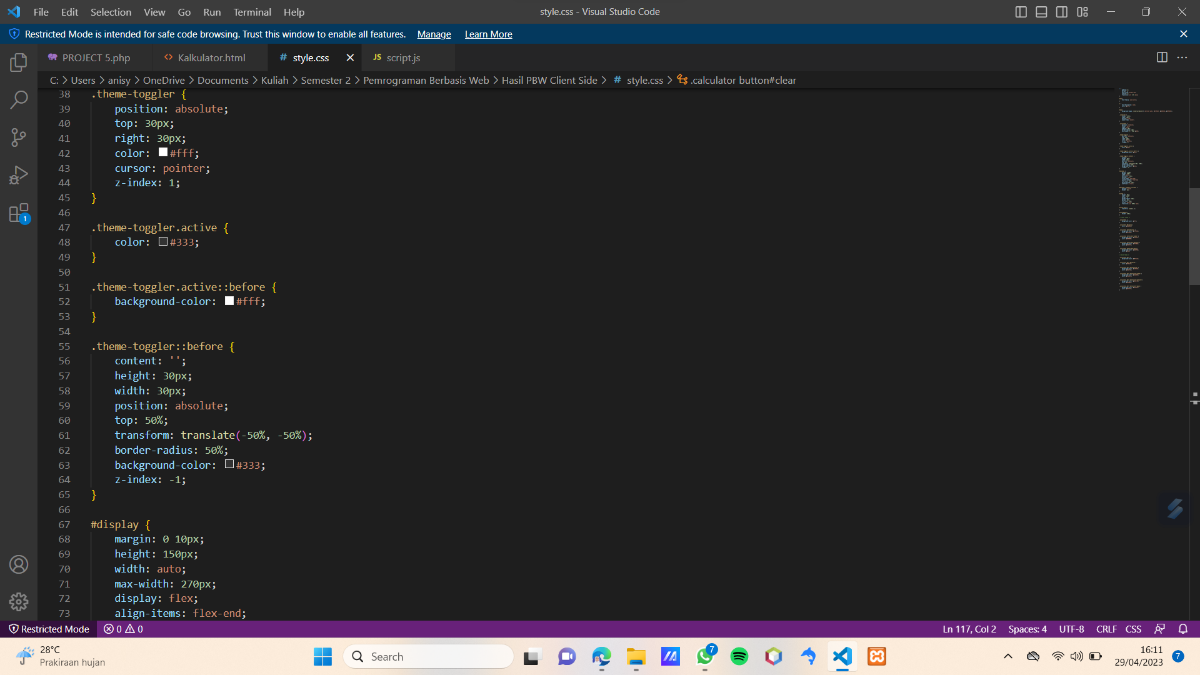
* **HTML**

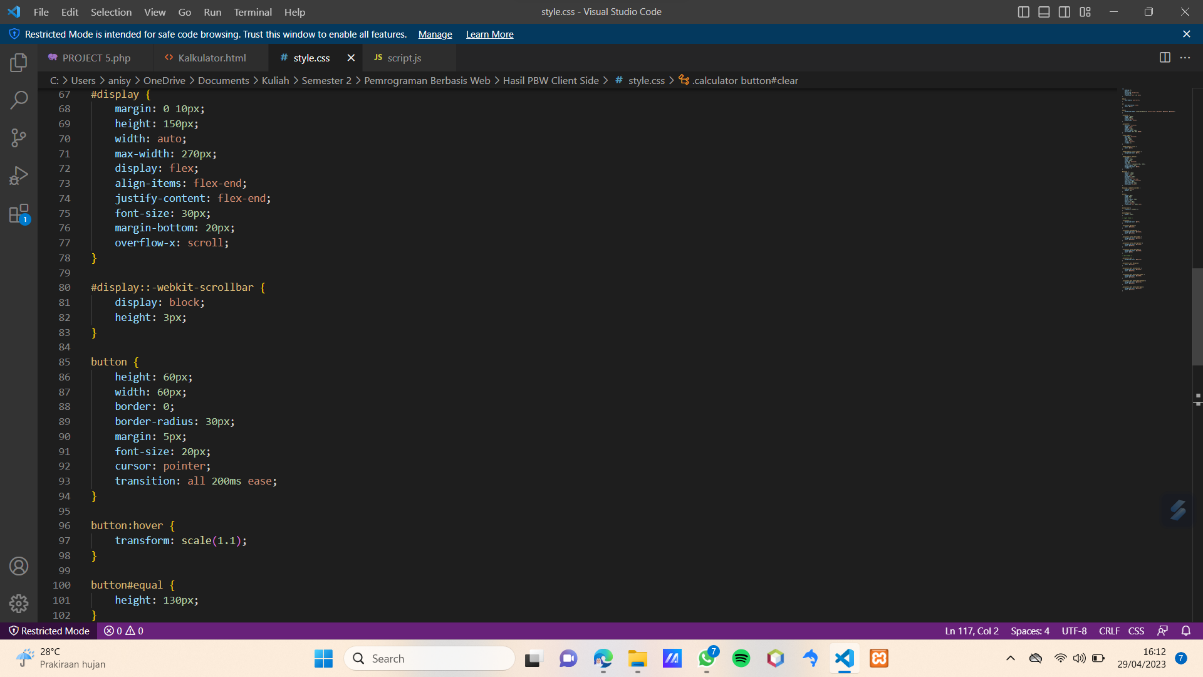
****

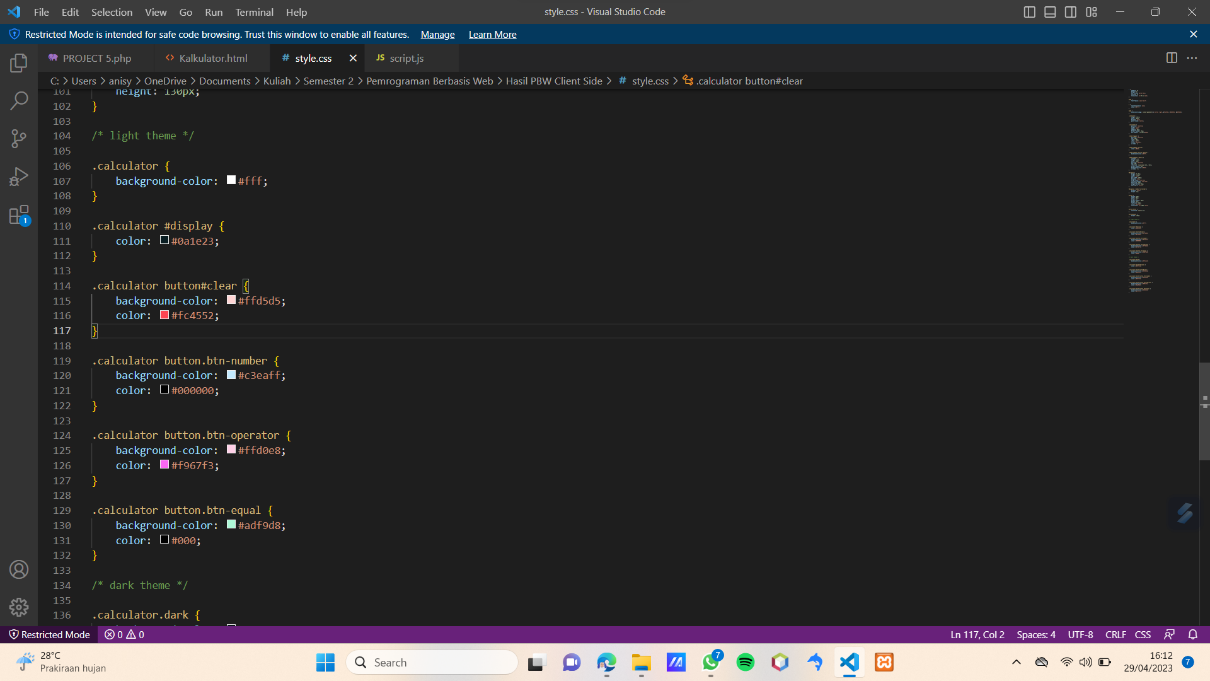
****

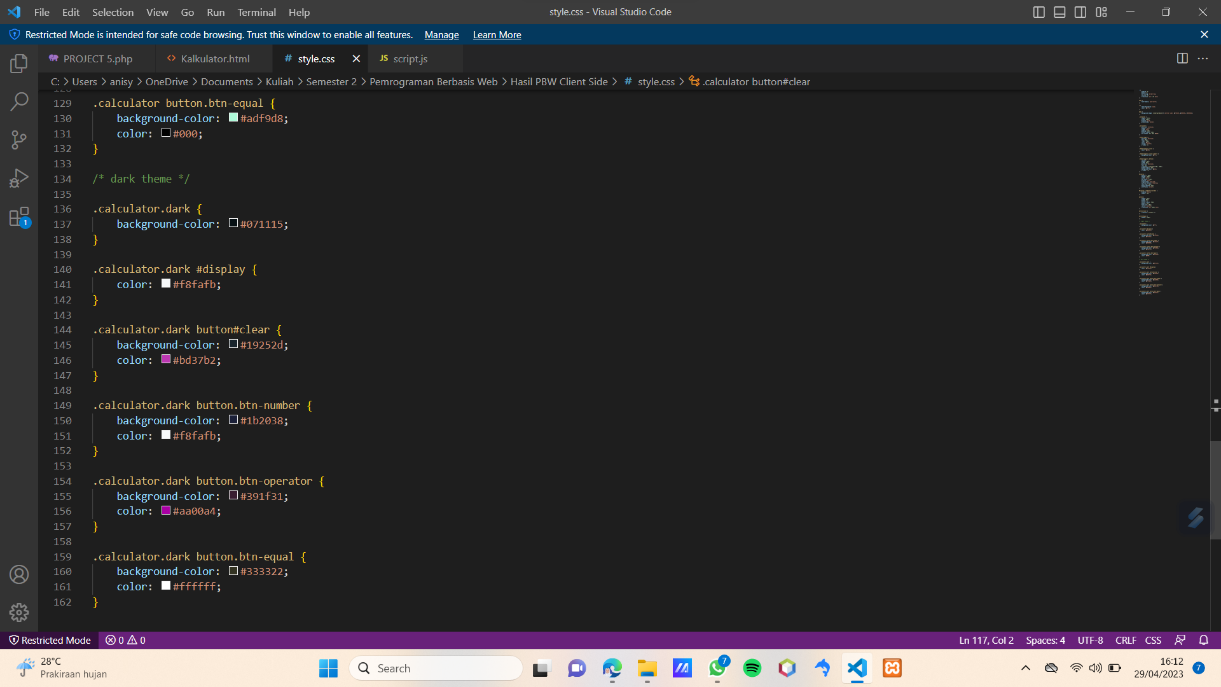
* **CSS**

****

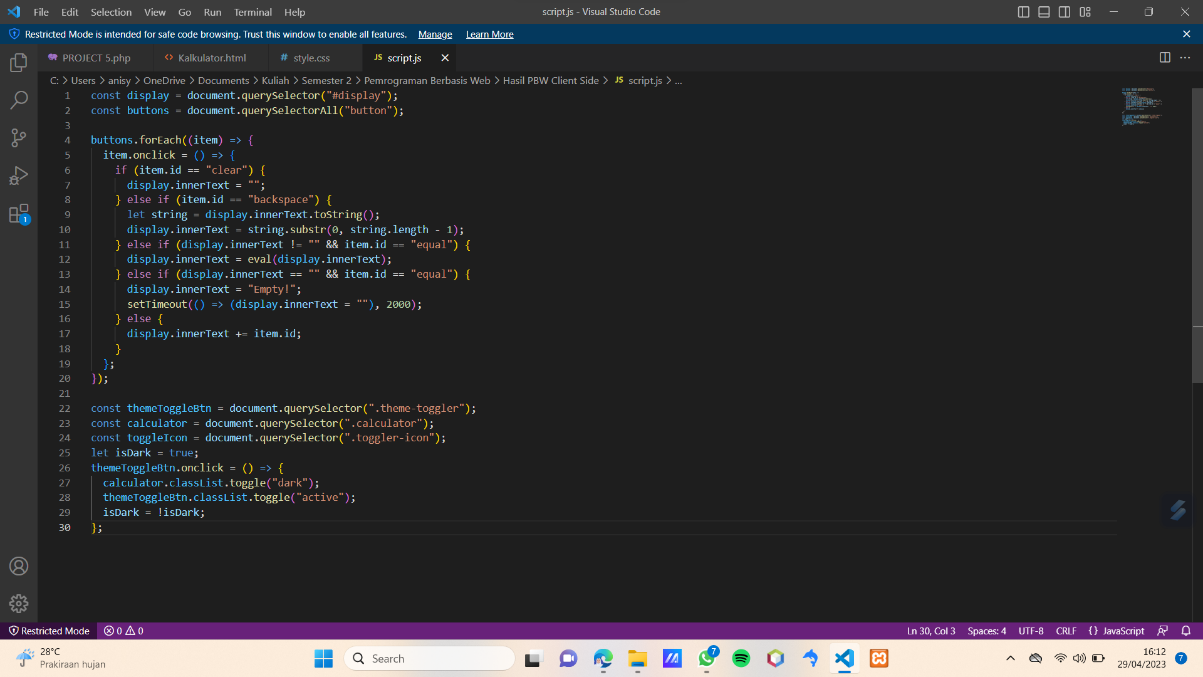
****

****

****

****

* **JavaScript**

****