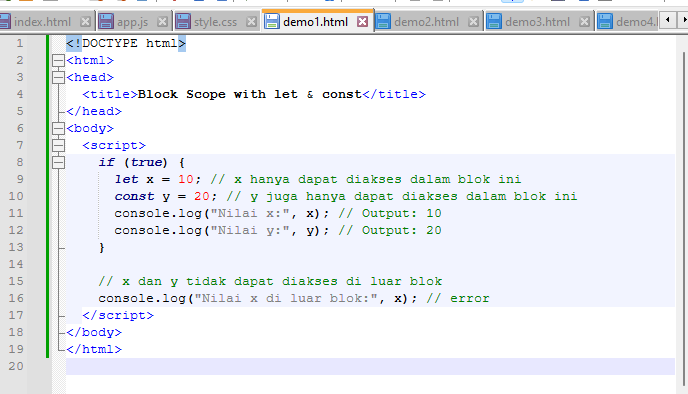
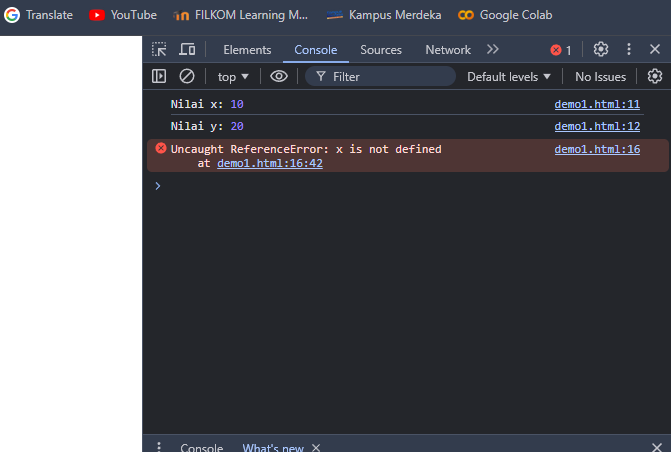
**Nanalis, Arlino Abraham Timotius**

**FRONT-END WEB DEVELOPMENT**

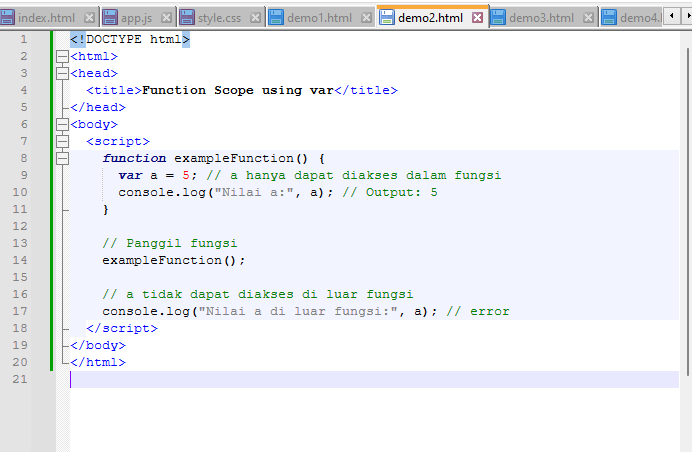
Exercise Lecture 9

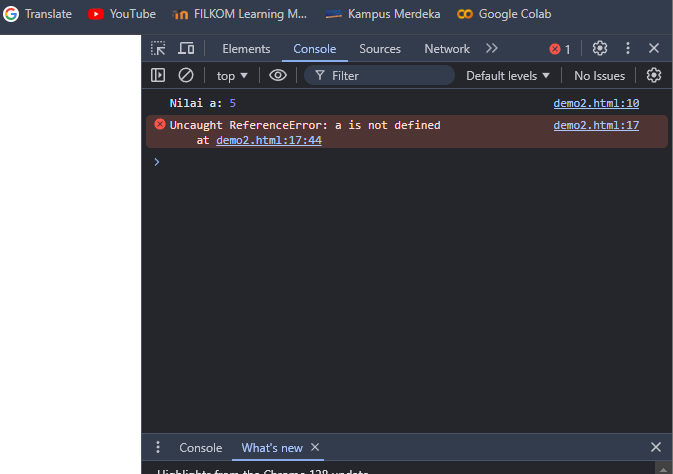
* Code 1



Penjelasan :

Kode di atas menunjukkan bagaimana variabel `let` dan `const` bekerja di dalam blok kode, seperti pada blok `if`. Di dalam blok itu, variabel `x` dan `y` dideklarasikan dengan nilai masing-masing 10 dan 20, dan bisa diakses serta dicetak di dalam blok tersebut. Tapi, saat mencoba mencetak `x` di luar blok, akan muncul error, karena `let` dan `const` hanya bisa digunakan di dalam area tempat mereka dideklarasikan, yaitu blok `if` tersebut.

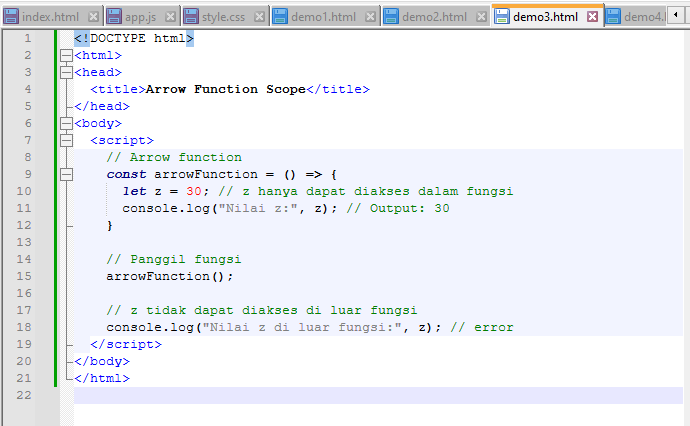
* Code 2

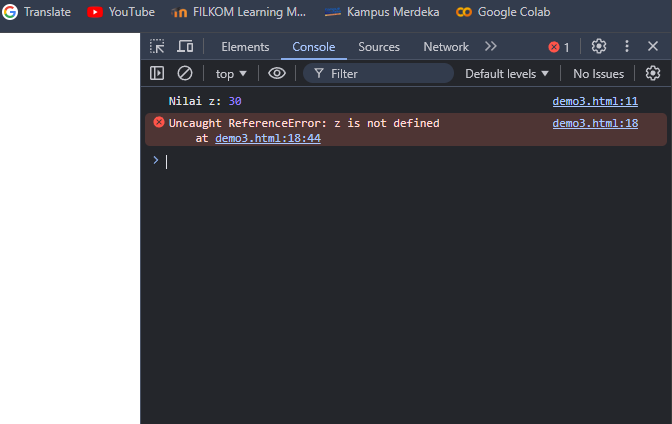


Penjelasan :

Kode di atas menunjukkan bagaimana variabel `var` memiliki cakupan fungsi. Di dalam fungsi `exampleFunction`, variabel `a` dideklarasikan dengan nilai 5 dan bisa dicetak menggunakan `console.log` di dalam fungsi tersebut. Namun, saat mencoba mengakses variabel `a` di luar fungsi, akan terjadi error karena variabel `var` hanya bisa diakses di dalam fungsi tempat variabel itu dideklarasikan. Jadi, variabel `a` tidak ada di luar fungsi `exampleFunction`.

* Code 3

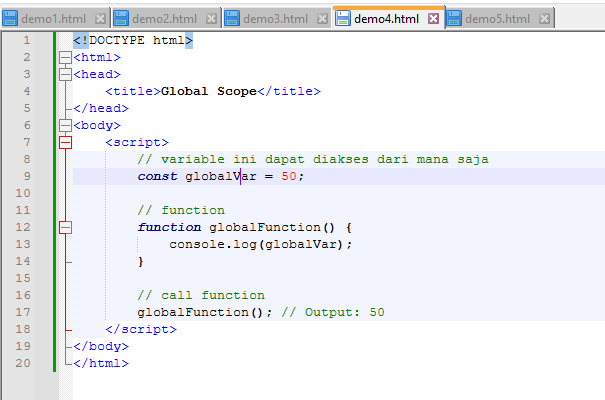


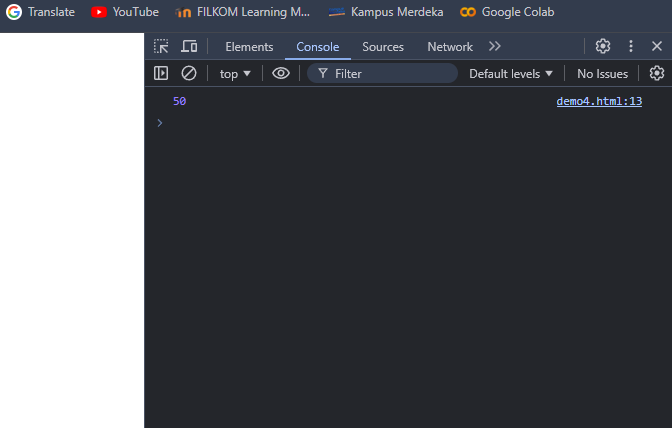


Penjelasan :

Kode di atas menunjukkan penggunaan arrow function, yaitu cara menulis fungsi yang lebih singkat menggunakan tanda panah (`=>`). Di dalam fungsi tersebut, variabel `z` dideklarasikan dengan nilai 30 menggunakan `let`, dan nilainya bisa dicetak di dalam fungsi dengan `console.log`. Namun, jika mencoba mengakses variabel `z` di luar fungsi, akan muncul error karena `let` hanya berlaku di dalam fungsi atau blok tempat variabel itu dideklarasikan.

* Code 4

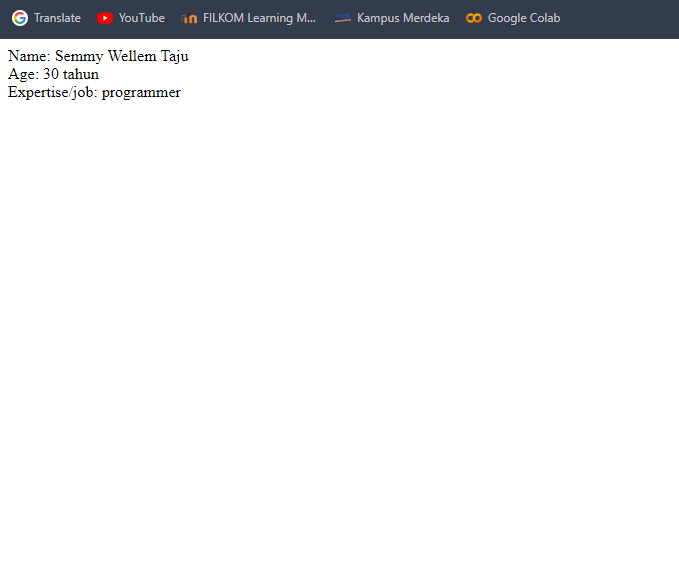
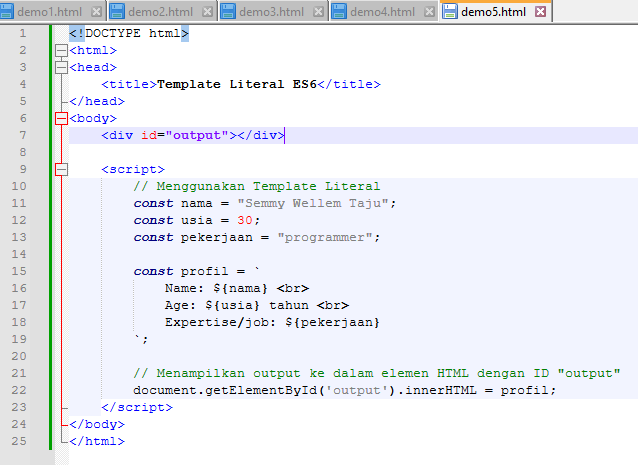




Penjelasan :

Kode di atas menunjukkan penggunaan variabel dengan cakupan global. Variabel `globalVar` dideklarasikan di luar fungsi menggunakan `const`, sehingga bisa diakses dari mana saja dalam kode, baik di dalam fungsi maupun di luar fungsi. Fungsi `globalFunction` mencetak nilai `globalVar` menggunakan `console.log`. Ketika fungsi dipanggil, nilai `globalVar`, yaitu 50, dicetak karena variabel ini memiliki cakupan global, yang artinya dapat diakses di seluruh bagian kode.

* Code 5

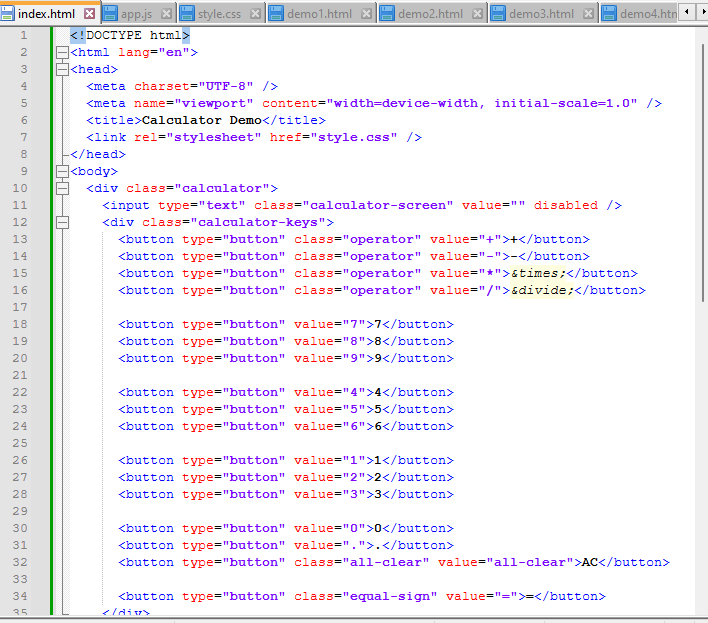
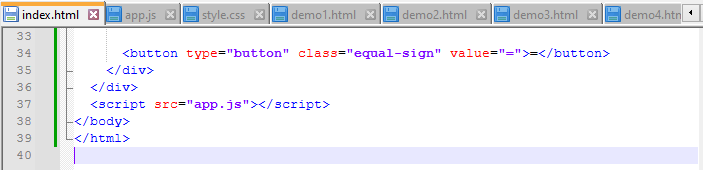


Penjelasan :

Kode di atas menunjukkan cara menggunakan \*\*Template Literal\*\* di ES6 untuk menyusun teks dengan lebih mudah. Tiga variabel, yaitu `nama`, `usia`, dan `pekerjaan`, masing-masing menyimpan informasi "Semmy Wellem Taju", 30, dan "programmer". Dengan menggunakan Template Literal (ditandai dengan tanda backtick \`), kita bisa menyisipkan variabel langsung ke dalam teks menggunakan `${}`. Teks yang telah disusun kemudian ditampilkan ke dalam elemen HTML dengan ID "output" menggunakan `innerHTML`. Hasilnya, informasi profil akan ditampilkan di halaman web dengan format yang rapi.

**Calculator**

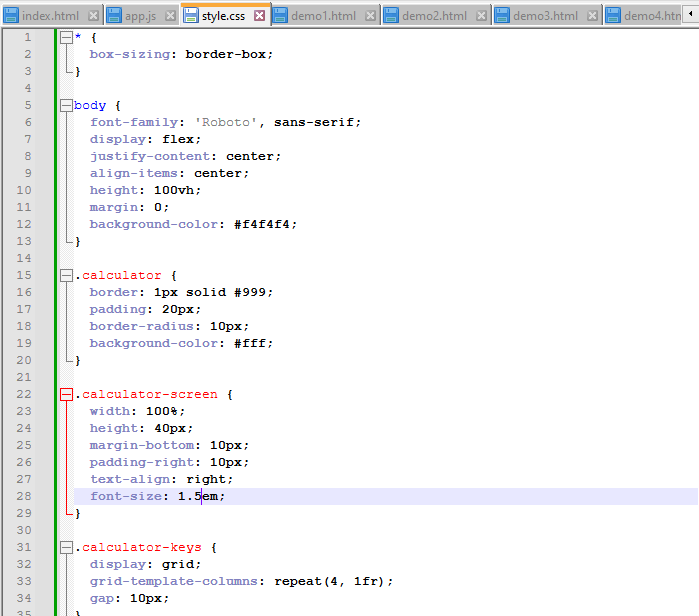
* Code HTML

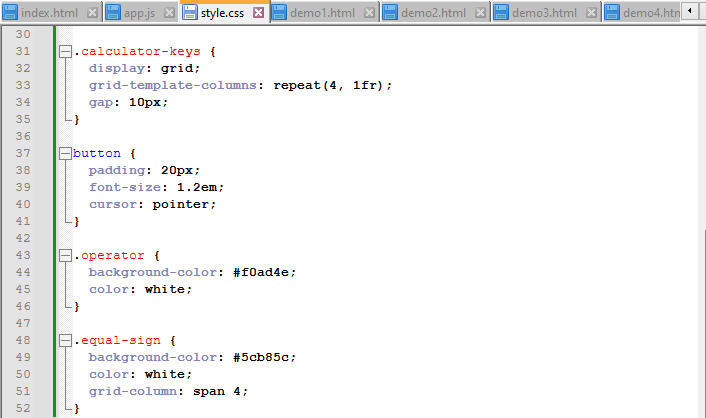


Penjelasan :

Kode di atas adalah sebuah struktur dasar dari kalkulator sederhana berbasis web. Di dalam elemen HTML, terdapat layar kalkulator (`input` dengan kelas "calculator-screen") yang menampilkan angka yang diketik atau hasil perhitungan. Ada juga tombol-tombol (dalam div dengan kelas "calculator-keys") untuk angka, operator seperti `+`, `-`, `\*`, dan `/`, serta tombol untuk titik desimal, tombol "AC" untuk menghapus semua input, dan tombol "=" untuk menghitung hasil. Kode JavaScript yang mengatur fungsionalitas kalkulator ini dimuat dari file eksternal bernama `app.js`.

* Code CSS

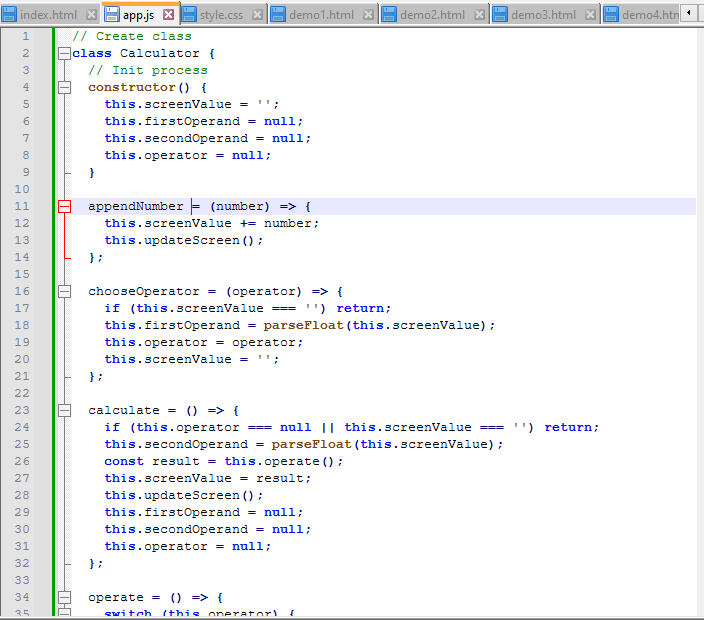


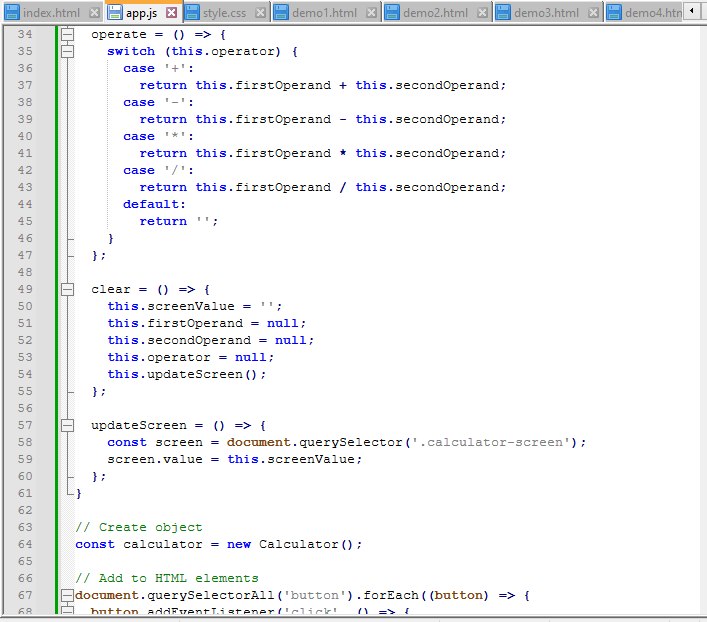


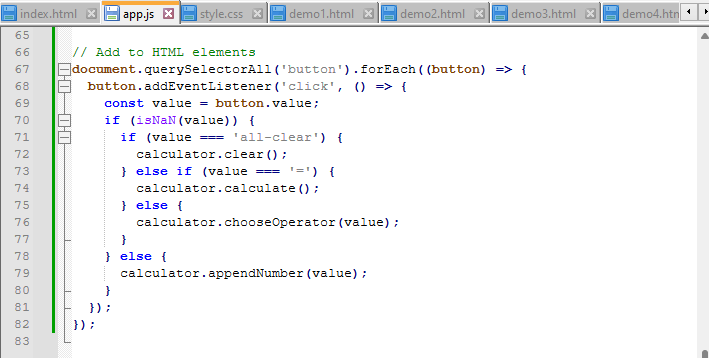
Penjelasan :

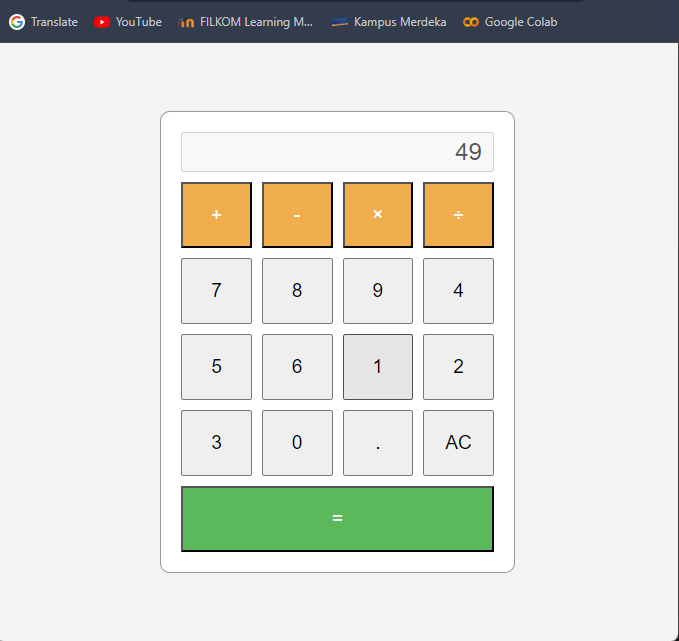
Kode di atas adalah \*\*CSS\*\* yang digunakan untuk menata tampilan kalkulator sederhana pada halaman web. Secara keseluruhan, kode ini mengatur tampilan elemen agar terlihat rapi dan menarik. Bagian `body` menggunakan font "Roboto", mengatur halaman agar kalkulator berada di tengah layar, dengan latar belakang abu-abu muda. Kalkulator memiliki batas, padding, dan sudut membulat untuk tampilan yang lebih halus. Layar kalkulator diatur agar teksnya rata kanan dan lebih besar. Tombol-tombol pada kalkulator ditata dalam grid dengan 4 kolom dan jarak antar tombol. Tombol operator diberi warna oranye, tombol "=" diberi warna hijau, dan tombol lain berukuran besar dengan gaya interaktif (pointer berubah saat di-hover).

* Code JS









Penjelasan :

Kode di atas membuat kalkulator dengan JavaScript menggunakan kelas `Calculator`. Kelas ini mengelola semua fungsi kalkulator, seperti menambahkan angka ke layar, memilih operator matematika, melakukan perhitungan, dan menghapus semua data. Metode `appendNumber` menambah angka yang ditekan ke layar, `chooseOperator` menyimpan operator matematika, dan `calculate` menghitung hasil berdasarkan operator. Metode `operate` melakukan operasi matematika sesuai dengan operator yang dipilih. Metode `clear` menghapus semua data. `updateScreen` memperbarui tampilan layar kalkulator. Setelah objek kalkulator dibuat, kode menambahkan event listener ke semua tombol di halaman, sehingga saat tombol diklik, kode memeriksa apakah tombol tersebut adalah angka, operator, atau tombol "AC" dan menjalankan fungsi yang sesuai dari objek kalkulator.