**LAPORAN PRAKTIKUM**

**PEMROGRAMAN WEB 1**



NAMA : Aef Nur Hidayah

NIM : 20230910054

KELAS : SINFC – 2023 - 03

FAKULTAS ILMU KOMPUTER

UNIVERSITAS KUNINGAN

TAHUN 2024

DAFTAR ISI

[**A.** **PRE-TEST** 3](#_Toc182414180)

[**B.** **PRAKTIKUM** 6](#_Toc182414181)

[**C.** **POST TEST** 18](#_Toc182414182)

[**D. TUGAS** 22](#_Toc182414183)

# **PRE-TEST**

1. Apa itu javaScript dan untuk apa biasanya digunakan?

JavaScript adalah bahasa pemrograman yang digunakan untuk mengembangkan situs web agar lebih dinamis dan interaktif. Dengan JavaScript, konten di dalam halaman web dapat bergerak atau berubah secara otomatis tanpa perlu memuat ulang halaman, memberikan pengalaman pengguna yang lebih menarik. Misalnya, JavaScript memungkinkan pembuatan elemen interaktif seperti animasi, menu drop-down, dan pembaruan konten secara real-time, yang sangat penting untuk meningkatkan keterlibatan pengunjung di situs web. Awalnya, JavaScript hanya dapat dijalankan di sisi klien (client-side), artinya kode ini dieksekusi di browser pengguna. Namun, dengan perkembangan teknologi seperti Node.js, JavaScript kini juga dapat digunakan di sisi server (server-side), memungkinkan pengembang untuk membangun aplikasi web yang lebih kompleks dan fungsional. Selain itu, JavaScript juga digunakan dalam pengembangan aplikasi mobile melalui framework seperti React Native.Secara keseluruhan, JavaScript berfungsi untuk membuat situs web lebih interaktif dan menarik, mengembangkan aplikasi mobile, serta memungkinkan manipulasi elemen-elemen dalam halaman web untuk meningkatkan pengalaman pengguna. Keterampilan dalam JavaScript menjadi sangat penting bagi para pengembang web untuk menciptakan aplikasi yang responsif dan dinamis.

1. Tuliskan sintaks dasar untuk mendeklarasikan variable dalam javaScript dan sebutkan perbedaan antara var, let dan const.

Kita dapat menggunakan tiga kata kunci utama: var, let, dan const. Sintaks dasar untuk mendeklarasikan variabel adalah sebagai berikut :

**var namaVariabel = nilai;**

**let namaVariabel = nilai;**

**const namaVariabel = nilai;**

Di mana namaVariabel adalah nama yang Anda berikan untuk variabel tersebut, dan nilai adalah data yang ingin disimpan dalam variabel.

Perbedaan antara var, let, dan const

1. **Var** :
   * **Cakupan (Scope)**: Variabel yang dideklarasikan dengan var memiliki cakupan fungsi (function scope) atau cakupan global jika dideklarasikan di luar fungsi. Ini berarti bahwa variabel tersebut dapat diakses di seluruh fungsi atau di seluruh skrip.
   * **Deklarasi Ulang**: Anda dapat mendeklarasikan ulang variabel yang menggunakan var tanpa menghasilkan kesalahan.
   * **Hoisting**: Variabel yang dideklarasikan dengan var akan dihoist, yang berarti mereka dapat digunakan sebelum deklarasi sebenarnya dalam kode.
2. **Let** :
   * **Cakupan (Scope)**: Variabel yang dideklarasikan dengan let memiliki cakupan blok (block scope), yang berarti hanya dapat diakses dalam blok kode tempat mereka dideklarasikan (misalnya, dalam loop atau pernyataan if).
   * **Deklarasi Ulang**: Anda tidak dapat mendeklarasikan ulang variabel yang menggunakan let dalam cakupan yang sama; ini akan menghasilkan kesalahan.
   * **Hoisting**: Meskipun juga dihoist, variabel yang menggunakan let tidak dapat diakses sebelum deklarasinya, sehingga akan menghasilkan kesalahan jika dicoba.
3. **Const** :
   * **Cakupan (Scope)**: Seperti let, variabel yang dideklarasikan dengan const juga memiliki cakupan blok.
   * **Deklarasi Ulang**: Anda tidak dapat mendeklarasikan ulang variabel yang menggunakan const, dan Anda juga tidak dapat mengubah nilai dari variabel tersebut setelah inisialisasi. Namun, jika nilai tersebut adalah objek atau array, isi dari objek atau array tersebut masih bisa dimodifikasi.
   * **Inisialisasi**: Variabel yang dideklarasikan dengan const harus diinisialisasi pada saat deklarasi; jika tidak, akan menghasilkan kesalahan.
4. Bagaimana cara membuat kondisi dalam JavaScript? Tuliskan contoh penggunaan if, else if, dan else.

Sintaks Dasar

1. **Pernyataan if**: Digunakan untuk mengevaluasi suatu kondisi. Jika kondisi tersebut benar (true), maka blok kode di dalamnya akan dieksekusi.

**if** (kondisi) {

*// kode yang dijalankan jika kondisi benar*

}

1. **Pernyataan else if**: Digunakan untuk mengevaluasi kondisi tambahan jika kondisi sebelumnya salah. Anda dapat memiliki beberapa else if untuk memeriksa berbagai kondisi.

**else** **if** (kondisi) {

*// kode yang dijalankan jika kondisi ini benar*

}

1. **Pernyataan else**: Menangani semua kemungkinan jika tidak ada dari kondisi sebelumnya yang benar.

**else** {

*// kode yang dijalankan jika semua kondisi sebelumnya salah*

}

1. Contoh Penggunaanya :

**let** nilai = 85;

**if** (nilai >= 90) {

console.log("Nilai Anda: A");

} **else** **if** (nilai >= 75) {

console.log("Nilai Anda: B");

} **else** {

console.log("Nilai Anda: C");

}

Dalam contoh di atas, program memeriksa nilai yang disimpan dalam variabel nilai. Jika nilai tersebut lebih besar atau sama dengan 90, maka akan mencetak "Nilai Anda: A". Jika tidak, tetapi nilai tersebut lebih besar atau sama dengan 75, maka akan mencetak "Nilai Anda: B". Jika kedua kondisi tersebut tidak terpenuhi, maka program akan mencetak "Nilai Anda: C".

## **PRAKTIKUM**

1. Praktikum 1

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Pengenalan JavaScript</title>  </head>  <body>  <script>  //Menampilkan pesan menggunakan alert  alert("Selamat datang di tutorial JavaScript!");  //Menampilkan pesan di konsol  console.log("Pesan ini muncul di konsol browser.");  </script>  </body>  </html> |
| Dimulai dengan deklarasi doctype <!DOCTYPE html>, yang menunjukkan bahwa dokumen ini menggunakan HTML5, sehingga membantu browser dalam merender halaman dengan cara yang tepat. Elemen <html lang="en"> menandakan awal dari dokumen dan menetapkan atribut lang ke "en" (Bahasa Inggris), yang sangat penting untuk aksesibilitas dan optimisasi mesin pencari (SEO). Di dalam elemen <head>, terdapat beberapa tag meta, termasuk <meta charset="UTF-8"> yang menentukan pengkodean karakter dokumen, serta <meta name="viewport" content="width=device-width, initial-scale=1.0"> yang mengatur tampilan responsif halaman di perangkat mobile. Judul halaman ditentukan oleh elemen <title>Pengenalan JavaScript</title>, yang akan tampil di tab browser.Selanjutnya, di dalam elemen <body>, terdapat skrip JavaScript yang menyisipkan interaksi dasar dengan pengguna. Pertama, fungsi alert("Selamat datang di tutorial JavaScript!"); digunakan untuk menampilkan kotak dialog dengan pesan sambutan kepada pengguna, memberikan umpan balik secara langsung. Selain itu, terdapat perintah console.log("Pesan ini muncul di konsol browser."); yang mengirimkan pesan ke konsol browser, berguna bagi pengembang untuk melakukan debugging atau pengujian tanpa terlihat oleh pengguna biasa. Secara keseluruhan, kode ini adalah contoh sederhana penggunaan HTML dan JavaScript untuk memberikan interaksi dasar kepada pengguna, dan bisa dikembangkan lebih lanjut dengan menambahkan elemen HTML lainnya atau logika tambahan dalam skrip JavaScript. |

1. Praktikum 2

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Document</title>  </head>  <body>  <script>  //Deklarasi variabel  var nama = "JavaScript";  let versi = 1.0;  const rilis = "2024";  //Tipe Data  let angka = 123; //number  let teks = "Belajar JavaScript"; //string  let banar = true; //boolean  //Menampilkan nilai variabel  console.log("Nama : " + nama);  console.log("Versi : " + versi);  console.log("Rilis : " + rilis);  console.log("Angka : " + angka);  console.log("Teks : " + teks);  console.log("Benar : " + benar);  //Komentar satu baris  //Ini adalah komentar  /\*  Komentar beberapa baris  Ini adalah komentar beberapa baris  \*/  </script>  </body>  </html> |
| Dokument HTML dimulai dengan deklarasi doctype (<!DOCTYPE html>) yang menunjukkan jenis dokumentasi sebagai HTML5. Kemudian, elemen <html lang="en"> menandai awal dari dokumentasi dan menetapkan atribut lang ke “en” (bahasa Inggris), yang sangat penting bagi aksesibilitas dan optimisasi mesin pencari (SEO).Di bawahnya, elemen <head> berfungsi sebagai tempat penyimpanan informasi metadata seperti pengkodean karakter (<meta charset="UTF-8">), konfigurasi viewport (<meta name="viewport" content="width=device-width, initial-scale=1.0">), dan judul halaman (<title>Document</title>).Elemen <body> kemudian dipilih karena merupakan area utama yang berisi isi konten yang nantinya akan ditampilkan kepada pengguna. Didalam elemen <body>, terdapat tag script (<script>).Di dalam tag script tersebut, terdapat deklarasi variasi variable menggunakan sintaks modern JavaScript:   1. Variabel nama didefinisikan dengan string "JavaScript". 2. Variabel versi didefinisikan dengan number 1.0. 3. Variabel rilis juga didefinisikan dengan string "2024" menggunakan keyword const. 4. Selain itu, terdapat contoh variasi tipe data dasar dalam JavaScript: 5. Variabel angka didefinisikan dengan number 123. 6. Variabel teks didefinisikan dengan string "Belajar JavaScript". 7. Variabel benar didefinisikan dengan boolean value true.   Setiap variabel kemudian dievaluasi dan ditampilkan melalui console.log() untuk memastikan nilainya sudah benar.Terakhir, kode juga menampilkan cara membuat komentar single line ("//Ini adalah komentar") dan multiple lines ("/\* ... \*/"). |

1. Praktikum 3

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Variable dan Tipe Data</title>  </head>  <body>  <script>  //Variabel dengan var, let dan const  var varNum = 10;  let letStr = "Hello";  const constBool = true;  //Tipe Data  let num = 123; // Number  let str = "This is a string"; // String  let bool = false; // Boolean  let obj = {Name : "John", age : 30}; // Object  let arr = [1, 2, 3, 4, 5]; // Array  let und; // Undefined  let nul = null; // Null  //Menampilkan hasil  console.log("varNum : " + varNum);  console.log("letStr : " + letStr);  console.log("constBool : " + constBool);  console.log("num : " + num);  console.log("str : " + str);  console.log("boll : " + bool);  console.log("obj " + obj);  console.log("arr : " + arr);  console.log("und : " + und);  console.log("nul : " + nul);  </script>  </body>  </html> |
| Dimulai dengan deklarasi doctype <!DOCTYPE html>, yang menunjukkan bahwa dokumen ini merupakan HTML5, diikuti dengan elemen <html lang="en"> yang menetapkan bahasa dokumen sebagai Bahasa Inggris, penting untuk aksesibilitas dan SEO. Dalam elemen <head>, terdapat tag meta yang mengatur pengkodean karakter (<meta charset="UTF-8">) dan viewport (<meta name="viewport" content="width=device-width, initial-scale=1.0">), serta judul halaman yang ditetapkan sebagai "Variable dan Tipe Data".Di dalam elemen <body>, terdapat tag <script> yang berisi kode JavaScript. Kode ini dimulai dengan mendeklarasikan tiga jenis variabel: var, let, dan const. Variabel varNum didefinisikan dengan nilai numerik 10, letStr dengan string "Hello", dan constBool dengan boolean true. Ini menunjukkan perbedaan antara ketiga kata kunci tersebut, di mana var memiliki cakupan fungsi, let memiliki cakupan blok, dan const digunakan untuk mendeklarasikan nilai konstan yang tidak dapat diubah.Selanjutnya, kode ini memperkenalkan berbagai tipe data dalam JavaScript. Tipe data yang didefinisikan meliputi:   1. **Number**: num dengan nilai 123. 2. **String**: str dengan nilai "This is a string". 3. **Boolean**: bool dengan nilai false. 4. **Object**: obj, yang merupakan objek dengan properti nama dan usia. 5. **Array**: arr, yang merupakan array berisi angka dari 1 hingga 5. 6. **Undefined**: variabel und dideklarasikan tanpa nilai. 7. **Null**: variabel nul didefinisikan dengan nilai null.   Setelah mendeklarasikan semua variabel, kode ini menampilkan hasilnya menggunakan console.log(), yang mencetak nilai-nilai variabel ke konsol browser. |

1. Praktikum 4

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Operator dan Ekspresi</title>  </head>  <body>  <script>  //Operator aritmatika  let a = 10;  let b = 5;  console.log("a + b = " + (a + b));  console.log("a - b = " - (a - b));  console.log("a \* b = " \* (a \* b));  console.log("a / b = " / (a / b));  console.log("a % b = " % (a % b));  //Operator perbandingan  console.log("a == b : " + (a == b));  console.log("a === b : " + (a === b));  console.log("a != b : " + (a != b));  console.log("a > b : " + (a > b));  console.log("a < b : " + (a < b));  //Operator Logika  let x = true;  let y = false;  console.log("x && y : " + (x && y));  console.log("x || y : " + (x || y));  console.log("!x : " + (!x));  //Ekspresi  let c = a + b \* 2; // Ekspresi logika  console.log("c = a + b \* 2 : " + c);  let isTrue = (a > b) && (b < 10); // Ekspresi Logika  console.log("isTrue = (a < b) && (b < 10) : " + isTrue);  </script>  </body>  </html> |
| Deklarasi doctype <!DOCTYPE html>, yang menunjukkan bahwa dokumen ini merupakan HTML5, diikuti oleh elemen <html lang="en"> yang menetapkan bahasa dokumen sebagai Bahasa Inggris, penting untuk aksesibilitas dan SEO. Di dalam elemen <head>, terdapat tag meta yang mengatur pengkodean karakter (<meta charset="UTF-8">) dan viewport (<meta name="viewport" content="width=device-width, initial-scale=1.0">), serta judul halaman yang ditetapkan sebagai "Operator dan Ekspresi".Di dalam elemen <body>, terdapat tag <script> yang berisi kode JavaScript. Kode ini dimulai dengan demonstrasi operator aritmatika menggunakan dua variabel, a dan b, yang masing-masing diatur ke nilai 10 dan 5. Kode ini mencetak hasil dari operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulus ke konsol. Namun, terdapat kesalahan sintaksis pada bagian operator pengurangan, perkalian, dan pembagian karena penggunaan tanda minus (-), bintang (\*), dan garis miring (/) yang tidak tepat. Seharusnya, operator tersebut tidak diikuti oleh tanda kutip.Selanjutnya, kode ini memperlihatkan penggunaan operator perbandingan untuk membandingkan nilai a dan b. Hasil dari perbandingan ini juga dicetak ke konsol untuk menunjukkan apakah nilai-nilai tersebut sama, tidak sama, lebih besar, atau lebih kecil.Kemudian, kode berlanjut dengan operator logika menggunakan dua variabel boolean, x dan y, yang diatur ke true dan false. Kode ini mencetak hasil dari operasi logika AND (&&), OR (||), dan NOT (!) ke konsol.Selain itu, terdapat beberapa ekspresi yang dihitung dan ditampilkan. Ekspresi pertama adalah let c = a + b \* 2;, yang menunjukkan bagaimana urutan operasi (precedence) bekerja dalam JavaScript. Ekspresi kedua adalah let isTrue = (a > b) && (b < 10);, yang mengevaluasi dua kondisi logika sekaligus. |

1. Praktikum 5

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Kondisi dalam JavaScript</title>  </head>  <body>  <script>  let nilai = 85;  //Kondisi if  if (nilai >= 90) {  console.log("Nilai Anda : A");  }  //Kondisi if else  else if (nilai >= 75) {  console.log("Nilai Anda : B");  } else {  console.log("Nilai Anda : C");  }  //Kondisi switch  switch(nilai) {  case 90:  console.log("Nilai Anda : A");  break;  case 80:  console.log("Nilai Anda : B");  default:  console.log("Nilai di luar rentang");  }  </script>  </body>  </html> |
| Dimulai dengan deklarasi doctype <!DOCTYPE html>, yang menandakan bahwa dokumen ini adalah HTML5, diikuti oleh elemen <html lang="en"> yang menetapkan bahasa dokumen sebagai Bahasa Inggris, penting untuk aksesibilitas dan optimisasi mesin pencari (SEO). Di dalam elemen <head>, terdapat tag meta yang mengatur pengkodean karakter (<meta charset="UTF-8">) dan viewport (<meta name="viewport" content="width=device-width, initial-scale=1.0">), serta judul halaman yang ditetapkan sebagai "Kondisi dalam JavaScript".Di dalam elemen <body>, terdapat tag <script> yang berisi kode JavaScript. Kode ini dimulai dengan mendeklarasikan variabel nilai yang diatur ke 85. Kemudian, terdapat penggunaan struktur kondisi if untuk menentukan grade berdasarkan nilai tersebut. Jika nilai lebih besar atau sama dengan 90, maka akan mencetak "Nilai Anda : A" ke konsol. Jika tidak, struktur else if mengecek apakah nilai berada di rentang 75 hingga 89; jika iya, maka akan mencetak "Nilai Anda : B". Jika nilai kurang dari 75, maka akan mencetak "Nilai Anda : C". Ini merupakan contoh penggunaan kondisi sederhana untuk mengklasifikasikan nilai.Selanjutnya, kode ini juga menggunakan struktur kondisi switch untuk memberikan hasil yang sama berdasarkan nilai. Namun, terdapat beberapa kesalahan dalam implementasi switch. Misalnya, tidak ada case untuk nilai 85 yang sesuai dengan variabel nilai, dan case untuk 80 tidak memiliki break, sehingga dapat menyebabkan eksekusi berlanjut ke case berikutnya. Selain itu, ada kesalahan penempatan default, yang seharusnya berada di akhir untuk menangani semua kasus yang tidak ditentukan sebelumnya. |

1. Praktikum 6

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Looping dalam JavaScript</title>  </head>  <body>  <script>  // Looping For  console.log("Looping for :");  for (let i = 0; i <= 5; i++) {  console.log("Angka : " + i);  }  // Looping While  let j = 1;  console.log("Angka : " + j);  while (j <= 5) {  console.log("Angka : " + j);  j++;  }  // Looping do while  let k = 1;  console.log("Looping do while : ");  do {  console.log("Angka : " + k);  k++;  } while (k <= 5);  </script>  </body>  </html> |
| Dimulai dengan deklarasi doctype <!DOCTYPE html>, yang menandakan bahwa dokumen ini adalah HTML5, diikuti oleh elemen <html lang="en"> yang menetapkan bahasa dokumen sebagai Bahasa Inggris, penting untuk aksesibilitas dan optimisasi mesin pencari (SEO). Di dalam elemen <head>, terdapat tag meta yang mengatur pengkodean karakter (<meta charset="UTF-8">) dan viewport (<meta name="viewport" content="width=device-width, initial-scale=1.0">), serta judul halaman yang ditetapkan sebagai "Looping dalam JavaScript".Di dalam elemen <body>, terdapat tag <script> yang berisi kode JavaScript. Kode ini dimulai dengan penggunaan looping for, di mana variabel i diinisialisasi dengan nilai 0 dan akan terus bertambah hingga mencapai 5. Setiap iterasi mencetak nilai i ke konsol dengan format "Angka : " diikuti oleh nilai tersebut. Ini adalah cara yang umum digunakan untuk melakukan iterasi dengan jumlah langkah yang sudah ditentukan sebelumnya.Selanjutnya, kode ini menggunakan looping while. Di sini, variabel j diinisialisasi dengan nilai 1, dan selama nilai j kurang dari atau sama dengan 5, program akan mencetak nilai j ke konsol. Setelah setiap cetakan, nilai j akan bertambah satu. Looping while berguna ketika jumlah iterasi tidak diketahui sebelumnya dan tergantung pada kondisi tertentu.Terakhir, kode ini menunjukkan penggunaan looping do while. Dalam struktur ini, blok kode akan dieksekusi setidaknya sekali sebelum kondisi dievaluasi. Variabel k diinisialisasi dengan nilai 1, dan selama kondisi k <= 5 terpenuhi, program akan mencetak nilai k. Ini memastikan bahwa meskipun kondisi tidak terpenuhi pada iterasi pertama, setidaknya satu output tetap dicetak. |

1. Praktikum 7

|  |
| --- |
| <!DOCTYPE html>  <html lang="en">  <head>  <meta charset="UTF-8">  <meta name="viewport" content="width=device-width, initial-scale=1.0">  <title>Fungsi dan Scope dalam JavaScript</title>  </head>  <body>  <script>  // Fungsi sederhana  function tambah(a, b) {  return a + b;  }  // Fungsi dengan parameter default  function greet(nama = "Pengguna") {  console.log("Halo, " + nama + "!");  }  // Fungsi dengan fungsi di dalamnya  function outerFunction() {  let outerVar = "Variabel Luar";  function innerFuncition() {  let innerVar = "Variabel Dalam";  console.log(outerVar); // Akses variabel luar  console.log(innerVar); // Akses variabel dalam  }  innerFuncition();  }  //Memanggil fungsi  console.log("Hasil penjumlahan : " + tambah(5, 10));  greet("John");  outerFunction();  // Scope  let globalVar = "Variabel Global";  function checkScope() {  let localVar = "Variabel Lokal";  console.log(globalVar); // Akses variabel global  console.log(localVar); // Akses variabel lokal  }  checkScope();  //console.log(localVar); // Ini akan menghasilkan error karena localVar tidak dapat diakses di sini  </script>  </body>  </html> |
| Dimulai dengan deklarasi doctype <!DOCTYPE html>, yang menunjukkan bahwa dokumen ini adalah HTML5, diikuti oleh elemen <html lang="en"> yang menetapkan bahasa dokumen sebagai Bahasa Inggris, penting untuk aksesibilitas dan optimisasi mesin pencari (SEO). Di dalam elemen <head>, terdapat tag meta yang mengatur pengkodean karakter (<meta charset="UTF-8">) dan viewport (<meta name="viewport" content="width=device-width, initial-scale=1.0">), serta judul halaman yang ditetapkan sebagai "Fungsi dan Scope dalam JavaScript".Di dalam elemen <body>, terdapat tag <script> yang berisi kode JavaScript. Kode ini dimulai dengan mendefinisikan fungsi sederhana bernama tambah, yang menerima dua parameter dan mengembalikan hasil penjumlahan keduanya. Fungsi ini menunjukkan cara dasar dalam mendeklarasikan dan menggunakan fungsi di JavaScript.Selanjutnya, kode ini memperkenalkan fungsi dengan parameter default melalui fungsi greet, yang mencetak pesan sapaan ke konsol. Jika tidak ada argumen yang diberikan saat pemanggilan fungsi, maka "Pengguna" akan digunakan sebagai nilai default. Ini merupakan fitur yang berguna untuk memberikan nilai awal tanpa harus selalu menyertakan parameter saat memanggil fungsi.Kemudian, terdapat contoh fungsi bersarang dengan outerFunction, di mana terdapat variabel outerVar di luar dan innerFuncition di dalamnya. Fungsi dalam dapat mengakses variabel dari fungsi luar berkat konsep closure dalam JavaScript. Ketika innerFuncition dipanggil, ia mencetak kedua variabel ke konsol, menunjukkan bagaimana scope bekerja.Setelah mendefinisikan fungsi-fungsi tersebut, kode ini memanggil masing-masing fungsi untuk menunjukkan hasilnya. Hasil penjumlahan dari tambah(5, 10) dicetak ke konsol, diikuti dengan sapaan untuk "John" dan eksekusi dari outerFunction.Bagian terakhir dari kode ini membahas scope dengan mendeklarasikan variabel global globalVar dan variabel lokal localVar di dalam fungsi checkScope. Fungsi ini mencetak kedua variabel ke konsol, menekankan bahwa variabel lokal hanya dapat diakses dalam konteks fungsinya sendiri. Komentar pada bagian akhir menunjukkan bahwa mencoba mengakses localVar di luar fungsinya akan menghasilkan error, karena variabel tersebut tidak tersedia di scope global. |

# **POST TEST**

1. Apa perbedaan antara var, let, dan const dalam hal scope? Berikan contoh kode yang menunjukkan perbedaan tersebut.
2. var

Scope : Variabel yang dideklarasikan dengan var memiliki function scope atau global scope. Ini berarti bahwa variabel tersebut dapat diakses di seluruh fungsi tempat ia dideklarasikan, atau jika dideklarasikan di luar fungsi, ia dapat diakses di seluruh skrip.

Contoh:

**var x = 10;**

**if (true) {**

**var y = 20; // y dideklarasikan dengan var**

**console.log(x); // Output: 10**

**console.log(y); // Output: 20**

**}**

**console.log(y); // Output: 20 (y dapat diakses di luar blok)**

1. let

Scope : Variabel yang dideklarasikan dengan let memiliki block scope. Ini berarti bahwa variabel tersebut hanya dapat diakses dalam blok kode tempat ia dideklarasikan (misalnya, dalam loop atau pernyataan if).

Contoh:

**let a = 10;**

**if (true) {**

**let b = 20; // b dideklarasikan dengan let**

**console.log(a); // Output: 10**

**console.log(b); // Output: 20**

**}**

**console.log(b); // Error: b is not defined (b tidak dapat diakses di luar blok)**

1. const

Scope : Seperti let, variabel yang dideklarasikan dengan const juga memiliki block scope. Namun, nilai dari variabel yang dideklarasikan dengan const tidak dapat diubah setelah inisialisasi.

Contoh:

**const c = 10;**

**if (true) {**

**const d = 20; // d dideklarasikan dengan const**

**console.log(c); // Output: 10**

**console.log(d); // Output: 20**

**}**

**console.log(d); // Error: d is not defined (d tidak dapat diakses di luar blok)**

Var memungkinkan akses variabel di luar blok tempat ia dideklarasikan, sedangkan let dan const membatasi akses variabel hanya dalam blok tersebut. Hal ini membuat penggunaan let dan const lebih aman dan lebih mudah diprediksi, sehingga disarankan untuk digunakan dalam pengembangan JavaScript modern.

1. Tulis fungsi JavaScript yang mengembalikan nilai kuadrat dari suatu bilangan. Bagaimana anda dapat memanggil fungsi tersebut untuk menghitung kuadrat dari angka 7?

Untuk membuat fungsi JavaScript yang mengembalikan nilai kuadrat dari suatu bilangan bisa menuliskan sintaks seperti :

**function kuadrat(angka) {**

**return angka \* angka; // Mengembalikan hasil kuadrat dari angka**

**}**

Fungsi ini menerima satu parameter, yaitu angka, dan mengembalikan hasil perkalian angka dengan dirinya sendiri, yang merupakan definisi dari kuadrat.

Untuk memanggilnya gunakan sintaks :

**let hasilKuadrat = kuadrat(7);**

**console.log("Kuadrat dari 7 adalah: " + hasilKuadrat);**

1. Dalam konteks scope, jelaskan apa yang dimaksud dengan global scope dan local scope. Berikan contoh Dimana variable yang dideklarasikan dalam local scope tidak dapat diakses dari global scope.
2. Global Scope

Global scope adalah ruang lingkup di mana variabel dideklarasikan di luar fungsi atau blok mana pun. Variabel yang memiliki global scope dapat diakses dari mana saja dalam kode, termasuk di dalam fungsi dan blok lainnya. Ini seperti "ruang publik" di mana semua bagian program dapat melihat dan menggunakan variabel tersebut.

1. Local Scope

Sebaliknya, local scope adalah ruang lingkup di mana variabel dideklarasikan di dalam sebuah fungsi. Variabel yang memiliki local scope hanya dapat diakses dari dalam fungsi tempat mereka dideklarasikan. Jika Anda mencoba mengakses variabel lokal dari luar fungsi, itu akan menghasilkan kesalahan karena variabel tersebut tidak dikenali di luar lingkupnya.

1. Contohnya :

|  |
| --- |
| // Variabel global  let globalVar = "Ini adalah variabel global";  function contohScope() {  // Variabel lokal  let localVar = "Ini adalah variabel lokal";    console.log(globalVar); // Output: Ini adalah variabel global  console.log(localVar); // Output: Ini adalah variabel lokal  }  // Memanggil fungsi  contohScope();  // Mengakses variabel global  console.log(globalVar); // Output: Ini adalah variabel global  // Mengakses variabel lokal dari luar fungsi  console.log(localVar); // Error: localVar is not defined |

globalVar dideklarasikan di luar fungsi contohScope, sehingga dapat diakses baik dari dalam maupun luar fungsi. Namun, localVar dideklarasikan di dalam fungsi, sehingga hanya dapat diakses dari dalam fungsi tersebut. Saat mencoba mengakses localVar dari luar fungsi, akan muncul kesalahan "localVar is not defined", karena variabel tersebut tidak ada dalam lingkup global.

# **D. TUGAS**

1. Lengkapi web alumni pada tugas sebelumnya dengan menambahkan beberapa ketentuan sebagai berikut :

1. Tambahkan interaktivitas sederhana seperti efek hover pada profile alumni.
2. Buat fitur pencarian sederhana untuk memfilter alumni berdasarkan nama atau jurusan.
3. Simpan seluruh praktikum pada modul sebelumnya kedalam repository git yang sudah anda buat.

|  |
| --- |
| **HTML :**  <!DOCTYPE html>  <html lang="en">  <head>      <meta charset="UTF-8">      <meta name="viewport" content="width=device-width, initial-scale=1.0">      <title>Web Alumni</title>      <link rel="stylesheet" href="style.css">      <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/5.15.4/css/all.min.css">  </head>  <body>      <div class="container">          <h1>Daftar Alumni</h1>          <input type="text" id="searchInput" placeholder="Cari alumni berdasarkan nama atau jurusan..." onkeyup="filterAlumni()">            <div id="alumniList" class="alumni-list">              <!-- Daftar alumni -->              <div class="alumni-profile" data-nama="John Doe" data-jurusan="Teknik Informatika">                  <i class="fas fa-user-graduate"></i>                  <h2>Ahmad Malik</h2>                  <p>Jurusan: Teknik Informatika</p>              </div>              <div class="alumni-profile" data-nama="Jane Smith" data-jurusan="Sistem Informasi">                  <i class="fas fa-user-graduate"></i>                  <h2>Aef Nur Hidayah</h2>                  <p>Jurusan: Sistem Informasi</p>              </div>              <div class="alumni-profile" data-nama="Alice Johnson" data-jurusan="Design Komunikasi Visual">                  <i class="fas fa-user-graduate"></i>                  <h2>Yusuf Fauzin</h2>                  <p>Jurusan: Design Komunikasi Visual</p>              </div>          </div>      </div>      <script src="main.js"></script>  </body>  </html> |
| **CSS :**  body {      font-family: 'Arial', sans-serif;      background-color: #f0f4f8;      margin: 0;      padding: 20px;  }  .container {      max-width: 900px;      margin: auto;      background: white;      padding: 20px;      border-radius: 8px;      box-shadow: 0 4px 20px rgba(0, 0, 0, 0.1);  }  h1 {      text-align: center;      color: #333;  }  input[type="text"] {      width: 100%;      padding: 12px;      border-radius: 5px;      border: 1px solid #ccc;      margin-bottom: 20px;  }  .alumni-list {      display: grid;      grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));      gap: 20px;  }  .alumni-profile {      background-color: #e9ecef;      padding: 15px;      border-radius: 5px;      text-align: center;      transition: transform 0.3s, box-shadow 0.3s; /\* Transisi untuk efek hover \*/  }  .alumni-profile:hover {      transform: translateY(-5px); /\* Efek mengangkat \*/      box-shadow: 0 4px 15px rgba(0, 0, 0, 0.2); /\* Bayangan saat hover \*/  }  .alumni-profile i {      font-size: 40px; /\* Ukuran ikon \*/      color: #007bff; /\* Warna ikon \*/  }  .alumni-profile h2 {      margin-top: 10px;  }  .alumni-profile p {      color: #666; /\* Warna teks jurusan \*/  } |
| **JAVASCRIPT :**  function filterAlumni() {      const input = document.getElementById('searchInput');      const filter = input.value.toLowerCase();      const alumniProfiles = document.getElementsByClassName('alumni-profile');      for (let i = 0; i < alumniProfiles.length; i++) {          const name = alumniProfiles[i].getAttribute('data-nama').toLowerCase();          const jurusan = alumniProfiles[i].getAttribute('data-jurusan').toLowerCase();            if (name.includes(filter) || jurusan.includes(filter)) {              alumniProfiles[i].style.display = "";          } else {              alumniProfiles[i].style.display = "none";          }      }  } |
| **HASIL RUN :** |