

COACHING PROGRAM Day 1

Javascript


Bagian 0: Apa itu Javascript?

JavaScript adalah bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi web interaktif. Ini adalah salah satu dari tiga teknologi inti dalam pengembangan web bersama dengan HTML dan CSS. Selain itu, JavaScript memiliki beragam framework yang digunakan untuk mengembangkan aplikasi web. Salah satu framework yang sangat populer adalah React. Framework ini memungkinkan pengembang untuk membangun antarmuka pengguna yang dinamis dan responsif. Selain React, ada juga banyak kerangka kerja JavaScript lainnya seperti Angular, Vue.js, Next.js, Ember.js, dll, masing-masing dengan pendekatan dan filosofi pengembangan yang berbeda. Kerangka kerja ini menyediakan berbagai alat dan fitur yang mempermudah pembuatan dan pemeliharaan aplikasi web yang kompleks serta meningkatkan produktivitas pengembang. Dengan adanya beragam pilihan kerangka kerja, pengembang dapat memilih yang paling sesuai dengan kebutuhan proyek mereka.

Untuk Membantu pembelajaran Javascript, kalian dapat menggunakan [NodeJS](https://nodejs.org/), basic js (menggunakan HTML dan console dari browser), atau online compiler seperti programiz.com


Bagian 1: Basic Javascript Syntax

Deklarasi Variabel



```
const angka = 10;  
var huruf = "abcde";  
let isValid = true;
```

Dalam JavaScript, untuk mendeklarasikan variabel, Anda dapat menggunakan tiga kata kunci yang umum digunakan: **let**, **var**, dan **const**.



```
const angka = 10;  
let isValid = true;  
  
angka = 20; // error  
isValid = false // tidak error
```

let digunakan untuk mendeklarasikan variabel dengan cakupan yang terbatas pada blok kode di mana variabel itu dideklarasikan. Variabel yang dideklarasikan dengan **let** tidak dapat di-deklarasikan ulang dalam cakupan yang sama, yang membuatnya lebih aman dan meminimalkan kesalahan.

const digunakan untuk mendeklarasikan variabel dengan nilai yang tidak dapat diubah setelah inisialisasi. Seperti **let**, variabel **const** juga memiliki cakupan berdasarkan blok kode.

COACHING PROGRAM Day 1

Javascript

Ini sangat berguna ketika Anda ingin memastikan bahwa nilai variabel tetap konstan sepanjang kode Anda.

```
if (true) {  
  const angka = 10;  
  var huruf = "abcde";  
  let isValid = true;  
  
  console.log(angka); // 10  
  console.log(huruf); // "abcde"  
  console.log(isValid); // true  
  
  if (true) {  
    console.log(angka); // 10  
    console.log(huruf); // "abcde"  
    console.log(isValid); // true  
  }  
}  
console.log(angka); // error  
console.log(huruf); // "abcde"  
console.log(isValid); // error
```

“const dan let hanya dapat diakses yang sejajar atau childrennya (bersifat block scope), sedangkan var dapat diakses dimana saja”

var, di sisi lain, memiliki cakupan berdasarkan fungsi di mana variabel tersebut dideklarasikan. Ini berarti variabel **var** dapat diakses di mana saja dalam fungsi tersebut. Variabel **var** juga dapat di-deklarasikan ulang dalam cakupan yang sama, yang dapat menyebabkan potensi kesalahan.

Pilihan antara **let**, **var**, dan **const** harus dipertimbangkan dengan hati-hati tergantung pada kebutuhan Anda dalam mengelola cakupan dan perubahan nilai variabel dalam kode JavaScript Anda, namun diusahakan untuk fokus menggunakan **let** dan **const** saja, karena deklarasi menggunakan **var** kurang baik dari segi code reading.

Pernyataan if/else

Pernyataan if dan else adalah bagian fundamental dalam pemrograman yang digunakan untuk mengatur aliran eksekusi kode. Mereka memungkinkan Anda untuk membuat keputusan berdasarkan kondisi tertentu. Berikut adalah deskripsi tentang cara kerja if dan else dalam JavaScript:

Pernyataan if digunakan untuk menguji suatu kondisi, yang merupakan ekspresi yang menghasilkan nilai true atau false. Jika kondisi tersebut benar (true), blok kode yang berada di dalam pernyataan if akan dieksekusi. Jika kondisi tersebut salah (false), blok kode tersebut akan diabaikan. Contoh:

COACHING PROGRAM Day 1

Javascript

```
const angka = 10

if (angka < 0) {
  console.log(angka + " adalah angka negatif")
} else if (angka == 0) {
  console.log(angka + " adalah angka nol")
} else {
  console.log(angka + " adalah angka positif")
}
```

Pada contoh di atas, kode akan mencetak "10 adalah angka positif" karena saat pengecekan pertama, $(angka < 0)$ menghasilkan nilai **false**, karena angka 10 itu tidak lebih kecil dari 0. Lalu pada pengecekan kedua, $(angka == 0)$ menghasilkan nilai **false**, karena angka 10 itu tidak sama dengan 0. Sehingga pada else terakhir (tidak dilakukan pengecekan), code ini dijalankan jika semua pengecekan sebelumnya bernilai false. Sehingga mencetak hasil "10 adalah angka positif".

Tips: Selain menggunakan double-quote ("lorem"), di javascript juga dapat menggunakan single quote ('lorem') dan backquote (`lorem`) dengan bebas. Namun untuk backquote, ada sedikit perbedaan, salah satunya biasa digunakan untuk menyematkan **variabel** pada string, seperti berikut:

```
console.log(`${angka} adalah angka negatif`)
```

For Loop

Pernyataan **for** adalah salah satu jenis loop yang digunakan untuk melakukan iterasi atau perulangan sejumlah tertentu kali. Ini adalah pilihan yang baik ketika Anda tahu berapa kali Anda ingin menjalankan kode. Pernyataan **for** terdiri dari tiga bagian: inisialisasi variabel, kondisi pengulangan, dan perubahan variabel. Contoh:

```
for (let i = 0; i < 5; i++) {
  console.log("Perulangan ke-" + i);
}
```

Dalam contoh ini, **let i = 0** adalah bagian inisialisasi, **i < 5** adalah kondisi pengulangan, dan **i++** adalah perubahan variabel. Loop akan berjalan selama kondisi **i < 5** terpenuhi, dan setiap iterasi akan menambahkan 1 ke nilai **i**. Sehingga menghasilkan output:

```
Perulangan ke-0
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
```

COACHING PROGRAM Day 1

Javascript

While Loop

Pernyataan **while** adalah loop yang digunakan untuk melakukan iterasi berdasarkan kondisi tertentu. Ini cocok ketika Anda tidak tahu berapa kali Anda perlu menjalankan kode, tetapi Anda ingin menjalankannya selama kondisi tertentu terpenuhi. Contoh:

```
let i = 0;
while (i < 5) {
  console.log("Perulangan ke-" + i);
  i++;
}
```

Dalam contoh ini, loop **while** akan terus berjalan selama kondisi **i < 5** benar (true), dan setiap iterasi akan menambahkan 1 ke nilai **i**. Anda perlu mengatur dan memperbarui kondisi pengulangan dan perubahan variabel secara manual dalam **while** loop. Sehingga menghasilkan output:

```
Perulangan ke-0
Perulangan ke-1
Perulangan ke-2
Perulangan ke-3
Perulangan ke-4
```

Perbedaan antara keduanya adalah bahwa **for** loop biasanya digunakan ketika Anda tahu berapa kali Anda ingin menjalankan kode, sementara **while** loop digunakan ketika Anda ingin menjalankan kode selama kondisi tertentu terpenuhi, tanpa menentukan jumlah iterasi secara eksplisit.

String editing

String editing merujuk pada proses memanipulasi atau mengubah konten dari string dalam pemrograman. Dalam JavaScript, ada beberapa operasi yang dapat Anda gunakan untuk mengedit string. Contoh:

```
let nama = "Nama saya Udin"

console.log(nama.length) // 14
console.log(nama.toLowerCase()) // nama saya udin
console.log(nama.toUpperCase()) // NAMA SAYA UDIN
console.log(nama + " Keren") // Nama saya Udin Keren
console.log(nama.replace("Udin", "John")) // Nama saya John
console.log(nama.slice(5)) // saya Udin
console.log(nama.slice(-4)) // Udin
console.log(nama.slice(0, 12)) // Nama saya Ud
console.log(nama.substring(0, 12)) // Udin
console.log(nama.split(' ')) // ['Nama', 'saya', 'Udin']
```

- `console.log(nama.length) // 14`
Ini menghitung dan mencetak jumlah karakter dalam string nama, yaitu 14 karakter.
- `console.log(nama.toLowerCase()) // nama saya udin`
Ini mengonversi semua karakter dalam string nama menjadi huruf kecil dan mencetaknya.
- `console.log(nama.toUpperCase()) // NAMA SAYA UDIN`
Ini mengonversi semua karakter dalam string nama menjadi huruf besar dan mencetaknya.

COACHING PROGRAM Day 1

Javascript

- `console.log(nama + " Keren")` // Nama saya Udin Keren
Ini menggabungkan string nama dengan string " Keren" dan mencetak hasilnya.
- `console.log(nama.replace("Udin", "John"))` // Nama saya John
Ini mengganti substring "Udin" dalam string nama dengan "John" dan mencetak hasilnya.
- `console.log(nama.slice(5))` // saya Udin
Ini memotong string nama mulai dari karakter ke-5 hingga akhir dan mencetak hasilnya.
- `console.log(nama.slice(-4))` // Udin
Ini memotong string nama mulai dari 4 karakter terakhir hingga akhir dan mencetak hasilnya.
- `console.log(nama.slice(0, 12))` // Nama saya Ud
Ini memotong string nama mulai dari karakter ke-0 hingga karakter ke-12 (tetapi tidak termasuk karakter ke-12) dan mencetak hasilnya.
- `console.log(nama.substring(0, 12))` // Udin
Ini hampir sama dengan slice, tetapi pada kasus ini, karena nilai kedua lebih kecil dari nilai pertama, itu mengembalikan potongan yang sama dengan menggunakan slice. Ini mungkin disengaja atau perlu diperiksa.
- `console.log(nama.split(' '))` // ['Nama', 'saya', 'Udin']
Ini memecah string nama menjadi array berdasarkan spasi sebagai pemisah (delimiter), sehingga setiap kata menjadi elemen array yang berbeda, dan hasilnya dicetak sebagai array.

Array

Array adalah struktur data yang digunakan dalam pemrograman untuk menyimpan sejumlah nilai dalam satu variabel. Nilai-nilai ini dapat berupa tipe data apa pun, seperti angka, string, objek, atau bahkan array lain. Array memiliki indeks numerik yang digunakan untuk mengakses nilai-nilai individu di dalamnya.

Deklarasi Array: Anda dapat mendeklarasikan array dalam JavaScript dengan menggunakan tanda kurung siku `[]` dan memasukkan nilai-nilai ke dalamnya, yang dipisahkan oleh koma.
Contoh:

```
let angka = [1, 2, 3, 4, 5];  
let buah = ["apel", "mangga", "pisang"];
```

Akses Nilai dalam Array: Anda dapat mengakses nilai dalam array dengan menggunakan indeks (urutan). Indeks dimulai dari 0 untuk elemen pertama. Contoh:

```
let buah = ["apel", "mangga", "pisang"];  
console.log(buah[0]); // Output: "apel"  
console.log(buah[1]); // Output: "mangga"
```

Panjang Array: Anda dapat mengakses panjang (jumlah elemen) dari array menggunakan properti `length`.

COACHING PROGRAM Day 1

Javascript

```
let buah = ["apel", "mangga", "pisang"];  
console.log(buah.length); // Output: 3
```

Modifikasi Array: Anda dapat menambah, mengubah, atau menghapus elemen dari array. Beberapa metode umum untuk melakukan ini adalah **push()**, **pop()**, **shift()**, **unshift()**, dan **splice()**.

```
let buah = ["apel", "mangga", "pisang"];  
buah.push("jeruk"); // Menambahkan "jeruk" ke array  
buah[1] = "stroberi"; // Mengubah nilai indeks ke-1  
buah.pop(); // Menghapus elemen terakhir ("jeruk")
```

Untuk menampilkan semua data pada array, anda dapat menggunakan for loop, while loop, forEach, map(), dan lain-lain.

Array adalah struktur data yang sangat umum dan berguna dalam pemrograman karena memungkinkan Anda untuk menyimpan dan mengelola banyak nilai dalam satu variabel. Dalam JavaScript, array adalah objek yang memiliki sejumlah metode bawaan untuk melakukan berbagai operasi terkait array.

Object

Objek (Object) adalah salah satu struktur data yang paling penting dalam pemrograman JavaScript. Objek adalah entitas yang mengelompokkan dan menyimpan data (properti) dan fungsi (metode) yang terkait dalam satu kesatuan. Properti dalam objek adalah pasangan "kunci-nilai" yang berisi informasi tentang objek tersebut.

Deklarasi Objek: Anda dapat mendeklarasikan objek dengan menggunakan tanda kurung kurawal {} dan mendefinisikan properti dan metodenya di dalamnya.

```
let orang = {  
  nama: "John",  
  usia: 30,  
  alamat: "Jl. Contoh No. 123",  
};
```

Akses Properti Objek: Anda dapat mengakses properti dalam objek dengan menggunakan operator titik (.) atau tanda kurung siku [].

```
console.log(orang.nama); // Output: "John"  
console.log(orang['usia']); // Output: 30
```

Menambah Properti Baru: Anda dapat menambahkan properti baru ke objek setelah objek dideklarasikan. Contoh:

COACHING PROGRAM Day 1

Javascript

```
orang.pekerjaan = "Insinyur";
```

Objek Bersarang (Nested Object): Objek dalam JavaScript dapat memiliki objek sebagai properti, membentuk struktur yang kompleks dengan beberapa tingkat kedalaman.

```
let mahasiswa = {  
  nama: "Alice",  
  alamat: {  
    jalan: "Jl. Contoh No. 456",  
    kota: "Bandung"  
  }  
};
```

Objek adalah struktur data yang sangat fleksibel dan sering digunakan dalam pengembangan aplikasi web. Mereka memungkinkan Anda untuk mengorganisir dan mengelola data dengan cara yang lebih terstruktur, serta merencanakan tindakan yang akan diambil dengan menggunakan metode.

Function

Fungsi (function) adalah salah satu konsep paling fundamental dalam pemrograman. Mereka adalah blok kode yang dapat dipanggil dan dieksekusi berulang kali dengan tujuan tertentu. Fungsi membantu mengorganisir kode menjadi unit-unit yang dapat digunakan kembali, memecah kode menjadi bagian yang lebih kecil dan lebih mudah dikelola.

Deklarasi Fungsi: Anda dapat mendeklarasikan fungsi dengan kata kunci function. Fungsi ini bisa memiliki parameter (input) dan mengembalikan nilai (output).

```
function sapa(nama) {  
  console.log("Halo, " + nama + "!");  
}
```

Memanggil Fungsi: Anda memanggil fungsi dengan menuliskan namanya diikuti oleh tanda kurung ().

```
sapa("Alice"); // Output: "Halo, Alice!"
```

Parameter Fungsi: Fungsi dapat menerima parameter, yang merupakan nilai yang diperlukan untuk menjalankan fungsi. Parameter ini digunakan dalam tubuh fungsi.

COACHING PROGRAM Day 1

Javascript

```
function tambah(a, b) {  
    return a + b;  
}  
let hasil = tambah(5, 3); // hasil akan berisi 8
```

Fungsi adalah konsep penting dalam JavaScript yang memungkinkan Anda untuk mengorganisir kode, mengisolasi logika tertentu, dan menjadikannya lebih mudah digunakan kembali. Mereka juga digunakan dalam banyak aspek pemrograman web, seperti menangani peristiwa (event), permintaan HTTP, dan lainnya.