

Dasar JavaScript

Topik

1. Variabel (Variable)
2. Tipe Data (Data Types)
3. Operator
4. Kondisional/Percabangan
5. Perulangan (Loop)
6. Fungsi (Function)
7. Array
8. Objek (Object)
9. Kelas (Class)

Variabel

- Variabel digunakan untuk menyimpan data yang dapat digunakan kembali untuk proses selanjutnya.
- Dalam JavaScript, Anda dapat mendeklarasikan variabel menggunakan `let`, `const`, atau `var`.
- Contoh:

```
let usia = 25;  
const nama = "John";
```

Perbedaan `let`, `const`, dan `var`

- `let` dan `const` diperkenalkan dalam ES6 (ECMAScript 2015).
- `let` dan `const` adalah variabel yang bersifat lokal (local variable).
- `var` adalah variabel yang bersifat global (global variable).
- `const` adalah variabel yang nilainya tidak dapat diubah (immutable).
- `let` dan `var` adalah variabel yang nilainya dapat diubah (mutable).

Perbedaan `let`, `const`, dan `var`

```
const PI = 3.14;  
// PI = 3.14159; // Error: Assignment to constant variable  
  
if (true) {  
  const age = 30;  
  console.log(age); // 30  
}  
  
// console.log(age); // Error: age is not defined outside the block
```

let

```
let counter = 0;
counter = counter + 1;

if (true) {
  let counter = 10; // Different variable in this block
  console.log(counter); // 10
}

console.log(counter); // 1 (the outer variable)
```

Perbedaan `let`, `const`, dan `var`

```
var x = 5;

if (true) {
  var x = 10; // This reassigns the outer variable x
  console.log(x); // 10
}

console.log(x); // 10 (the same variable was reassigned)
```

Tipe Data

- **Number:** Nilai numerik (misalnya, 42, 3,14)
- **String:** Data teks/string (e.g., "Hello, World!")
- **Boolean:** True or false
- **Array:** Kumpulan nilai yang terindeks
- **Object:** Pasangan kunci-nilai (key-value pair)
- **Null:** Merepresentasikan ketiadaan nilai
- **Undefined:** Meresentasikan nilai yang belum ditetapkan

Type Data

```
const num = 42;  
const greeting = "Hello";  
const isTrue = true;  
const colors = ["red", "green", "blue"];  
const person = { name: "Alice", age: 25 };  
const emptyValue = null;  
  
let uninitializedVar;
```

Operator

- Operator aritmatika: `+`, `-`, `*`, `/`, `%`
- Operator perbandingan: `==`, `===`, `!=`, `!==`, `<`, `>`, `<=`, `>=`
- Operator logika: `&&` (AND), `||` (OR), `!` (NOT)

```
const x = 5;  
const y = 10;  
const sum = x + y; // 15  
const isEqual = x === y; // false  
const isGreater = x > y; // false  
const isLogical = x < 10 && y > 5; // true
```

Kondisional/Percabangan

Kondisional memungkinkan Anda untuk membuat keputusan dalam kode Anda berdasarkan kondisi tertentu. Pernyataan kondisional yang paling umum adalah if, else if, dan else.

```
const age = 25;

if (age < 18) {
  console.log("You are a minor.");
} else if (age >= 18 && age < 65) {
  console.log("You are an adult.");
} else {
  console.log("You are a senior citizen.");
}
```

Perulangan (Loop)

Perulangan digunakan mengulang kode tertentu berdasarkan kondisi tertentu.

Perulangan `for`

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

Perulangan (Loop)

Perulangan `while`

```
let i = 0;

while (i < 5) {
  console.log("Iteration " + i);
  i++;
}
```

Perulangan (Loop)

Perulangan `do...while`

```
let i = 0;

do {
  console.log("Iteration " + i);
  i++;
} while (i < 5);
```

Fungsi (Function)

Sebuah fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu. Fungsi dapat digunakan kembali di berbagai bagian dari kode Anda.

```
function add(a, b) {  
    return a + b;  
}  
  
const result = add(3, 4); // 7
```

Array

Array digunakan untuk menyimpan kumpulan nilai yang terindeks.

```
const colors = ["red", "green", "blue"];  
console.log(colors[0]); // "red"  
colors.push("yellow"); // Add an element to the end of the array  
  
const numbers = [1, 2, 3, 4, 5];  
  
const mixed = ["Hello", 42, true];
```


Objek (Object)

Objek digunakan untuk menyimpan pasangan kunci-nilai (key-value pair) untuk merepresentasikan sebuah objek nyata.

```
const person = {  
  name: "Alice",  
  age: 25,  
  isStudent: false,  
};  
  
console.log(person.name);
```

Kelas (Class)

Class adalah blueprint untuk membuat objek. Class mendefinisikan properti dan metode yang akan dimiliki oleh objek.

Class baru diperkenalkan dalam ES6 (ECMAScript 2015).

```
class Animal {  
  constructor(name) {  
    this.name = name;  
  }  
  
  speak() {  
    console.log(this.name + " makes a sound.");  
  }  
}  
  
const cat = new Animal("Whiskers");  
cat.speak(); // "Whiskers makes a sound."
```