



**ISCOM Programming & IoT**

# JAVASCRIPT DASAR

Pertemuan minggu ke-4

# Agenda

- Pengantar JavaScript
- Variabel
- Tipe Data
- Operator
- Struktur Kontrol (Percabangan dan Perulangan)
- Function
- Array
- Document Object Model (DOM)

# Pengantar JavaScript



JS

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman website yang dinamis dan interaktif. Dulunya JavaScript hanya berjalan di sisi browser (client), tetapi sekarang JavaScript bisa juga digunakan di sisi server (backend) dengan Node JS. Selain membuat website, JavaScript juga bisa digunakan untuk membuat aplikasi berbasis mobile, desktop, dan machine learning.

## Contoh Syntax dasar JavaScript:



```
index.html  index.js  style.css
index.js
1 console.log("Hello World!");
```

# Variabel

JavaScript memiliki aturan penulisan kode yang harus diikuti agar tidak error

## Variabel:

Variabel dalam bahasa pemrograman digunakan untuk menyimpan nilai, dan di JavaScript ada tiga keyword dalam mendeklarasikan variabel:

```
3  
4  var nama = "Hafidz"  
5  let umur = 20  
6  const jenisKelamin = "laki-laki"
```

- **var** (Jarang digunakan karena memiliki scope yang luas)
- **let** (nilainya bisa diubah)
- **const** (Tidak bisa diubah setelah dideklarasikan, karena sifatnya konstanta)

# Tipe Data

Tipe Data dalam bahasa pemrograman digunakan untuk menentukan jenis nilai yang dapat disimpan dan diolah oleh suatu variabel dalam program.

## Tipe Data:

JavaScript sendiri memiliki tipe data dasar, diantaranya:

```
9  const nama = "Abyan" // string
10 const umur = 19 // number
11 const bool = true // boolean
12 const kosong = null // null --> Variabel untuk nilai yang kosong
13 const tidakAda = undefined // undefined --> Variabel yang belum memiliki nilai
14 const array = [1, 2, 3, 4, 5] // array
15
16 const obj = {
17   nama: "Abyan",
18   umur: 19,
19   jenisKelamin: "laki-laki",
20   kuliah: "UPN Veteran Jawa Timur"
21 } // object
```

# Operator

Operator dalam bahasa pemrograman digunakan untuk untuk melakukan operasi matematika, logika, dan manipulasi data pada variabel atau nilai tertentu.

## Operator

Operator dalam bahasa pemrograman ada:

- Aritmatika: +, -, \*, /, %, \*\*
- Perbandingan: <, >, <=, >=, !=, !==, ==, ===
- Logika: AND (&&) OR (||) NOT (!)
- Operator Ternary: Menggunakan simbol ? :

# Struktur Kontrol

Untuk mengontrol jalannya program, kita bisa menggunakan perbandingan seperti (if-else dan switch-case), kemudian untuk looping seperti (for, do-while, dan while).

## Pengkondisian (if else):

Digunakan untuk menentukan jalur program berdasarkan kondisi yang nilainya True

```
23 // Kondisi
24 nilai = 80
25 if ( nilai >= 75 ) {
26     console.log("Selamat Anda Lulus!")
27 } else {
28     console.log("Maaf Anda Tidak Lulus!")
29 }
```

# Struktur Kontrol

Untuk mengontrol jalannya program, kita bisa menggunakan perbandingan seperti (if-else, if else if dan switch-case), kemudian untuk looping seperti (for, do-while, dan while).

## Pengkondisian (if else if):

Digunakan untuk menentukan jalur program berdasarkan kondisi yang nilainya True

```
2
3  const nilai = 70;
4
5  if (nilai >= 85 && nilai <= 100) {
6    console.log("Anda mendapatkan nilai A");
7  } else if (nilai >= 75 && nilai <= 84) {
8    console.log("Anda mendapatkan nilai B")
9  } else if (nilai >= 65 && nilai <= 74) {
10   console.log("Anda mendapatkan nilai C");
11 } else if (nilai >= 55 && nilai <= 64) {
12   console.log("Anda mendapatkan nilai D");
13 } else {
14   console.log("Mohon maaf, anda tidak lulus ujian dan mengulang tahun depan!");
15 }
```



# Struktur Kontrol

Untuk mengontrol jalannya program, kita bisa menggunakan perbandingan seperti (if-else dan switch-case), kemudian untuk looping seperti (for, do-while, dan while).

## Pengkondisian (switch case):

Digunakan untuk menentukan jalur program berdasarkan kondisi

```
31  const grade = "A"
32
33  switch (grade) {
34
35      case "A":
36          console.log("Good Job");
37          break;
38
39      case "B":
40          console.log("Not Bad");
41          break;
42
43      case "C":
44          console.log("Try Harder");
45          break;
46
47      default:
48          console.log("Try Again Next Year");
49  }
50
```

# Struktur Kontrol

Untuk mengontrol jalannya program, kita bisa menggunakan perbandingan seperti (if-else dan switch-case), kemudian untuk looping seperti (for, do-while, dan while).

## Perulangan (for):

Digunakan untuk melakukan iterasi nilai pada saat kondisinya bernilai true

```
53 // PERULANGAN
54 for (let i = 0; i < 5; i++) {
55     console.log("Perulangan ke-" + i)
56 }
57
```

# Struktur Kontrol

Untuk mengontrol jalannya program, kita bisa menggunakan perbandingan seperti (if-else dan switch-case), kemudian untuk looping seperti (for, do-while, dan while).

## Perulangan (while):

Digunakan untuk melakukan iterasi nilai pada saat kondisinya bernilai true

```
59   let i = 0;
60   while (i <= 5) {
61       console.log("Perulangan ke-" + i)
62       i++
63   }
64
```

# Struktur Kontrol

Untuk mengontrol jalannya program, kita bisa menggunakan perbandingan seperti (if-else dan switch-case), kemudian untuk looping seperti (for, do-while, dan while).

## Perulangan (do-while):

Digunakan untuk melakukan iterasi nilai pada saat kondisinya bernilai true


```
70 let angka = 1;
71 do {
72     console.log(angka);
73     angka++;
74 } while (angka <= 19);
```

# Function

Function secara sederhana adalah blok kode yang bisa digunakan kembali pada saat kita memanggil nama functionnya

## Function

Dengan function kita bisa menghindari penulisan kode yang berulang dan membuat kode lebih terstruktur dan mudah dikelola.

```
Qodo Gen: Options | Test this function
78  function salam(nama) {
79       return console.log("Halo " + nama);
80  }
81  salam("Abyan")
```

# Array

Array dalam JavaScript merupakan tipe data Object, Array sendiri kurang lebih kegunaannya untuk menyimpan lebih dari satu nilai, sama seperti bahasa pemrograman pada umumnya,.

## Array

Array adalah tipe data yang digunakan untuk menyimpan data yang nilainya lebih dari satu, kurang lebih sama seperti tipe data variabel cuman kalau variabel hanya menyimpan satu nilai saja.

```
const mobil = ['Honda', 'Toyota', 'Suzuki', 'Daihatsu', 'Mitsubishi', 'Mercedes']  
console.log("mobil");  
// Output: ['Honda', 'Toyota', 'Suzuki', 'Daihatsu', 'Mitsubishi', 'Mercedes']
```

# Operasi pada Array

## Macam-macam operasi dasar Array:

- **.Push():** Menambah elemen di akhir array
- **.pop():** Menghapus elemen terakhir pada array
- **.unshift():** Menambahkan elemen di awal array
- **.shift():** Menghapus elemen pertama pada array
- **.length:** Mendapatkan jumlah elemen di dalam array
- **forEach(), for(...of...), .map:** Untuk melakukan iterasi data pada array
- **.slice():** Mengambil sebagian Array
- **.splice():** Menghapus atau menyisipkan array
- **.includes(), .indexOf():** Mencari elemen
- **.find(), .findindex():** Mencari elemen berdasarkan kondisi

# Object

Object dalam JavaScript merupakan tipe data Object, Object sendiri kegunaannya untuk menyimpan data dalam bentuk pasangan key: value.

## Object

Object adalah struktur data yang digunakan untuk menyimpan data dalam bentuk pasangan (pairs), format tipe data object adalah berupa key: value

```
const identitas = {  
  nama: 'John Doe',  
  umur: 20,  
  alamat: 'Surabaya',  
  pekerjaan: 'Programmer',  
}  
  
console.log(identitas);  
// Output: { nama: 'John Doe', umur: 20, alamat: 'Surabaya', pekerjaan: 'Programmer' }
```



# Document Object Model (DOM)

Document Object Model (DOM) merupakan representasi struktur dokumen HTML dalam bentuk objek, sehingga JavaScript bisa berinteraksi dengan elemen HTML.

## Cara mengakses elemen DOM pada JavaScript:

Di JavaScript ada 5 metode atau syntax untuk bisa mengakses elemen html atau DOM, diantaranya ada:

- **document.getElementById():** Mengambil elemen berdasarkan nama id
- **document.getElementsByTagName():** Mengambil elemen berdasarkan nama tag html
- **document.getElementsByClassName():** Mengambil elemen berdasarkan nama class
- **document.querySelector():** Mengambil elemen pertama sesuai selector yang dipilih
- **document.querySelectorAll():** Mengambil semua elemen sesuai selector yang dipilih

# Document Object Model (DOM)

Document Object Model (DOM) merupakan representasi struktur dokumen HTML dalam bentuk objek, sehingga JavaScript bisa berinteraksi dengan elemen HTML.

## Cara memanipulasi elemen DOM pada JavaScript:

### 1. Mengubah konten elemen:

```
// DOM Manipulation:  
// MENGUBAH KONTEN ELEMEN PADA HTML:  
  
const judul = document.getElementById('judul')  
const teks = document.getElementsByClassName('teks');  
const paragraf = document.getElementsByTagName('p');  
const container = document.querySelector('.container');  
const list = document.querySelectorAll('.item');
```

```
// CONTOH IMPLEMENTASI YANG BENAR:  
judul.innerHTML = "Selamat datang di Website kami!";  
paragraf[0].innerText = "Ini adalah paragraf yang sangat panjang";  
teks[0].textContent = 'Dunia Programming sangat menyenangkan';  
container.textContent = "DOM Manipulation menyenangkan sekali!!!";  
  
list.forEach((item, index) => {  
  item.textContent = `Ini adalah item ke-${index}`;  
})
```

# Document Object Model (DOM)

Document Object Model (DOM) merupakan representasi struktur dokumen HTML dalam bentuk objek, sehingga JavaScript bisa berinteraksi dengan elemen HTML.

## Cara memanipulasi elemen DOM pada JavaScript:

2. Mengubah elemen dengan style:

```
// Mengubah elemen dengan styling css:  
paragraf[0].style.color = "red";  
paragraf[0].style.fontSize = "20px";  
teks[0].style.backgroundColor = "yellow";  
teks[0].style.fontSize = "18px";  
container.style.border = "2px solid black";
```

# Document Object Model (DOM)

Document Object Model (DOM) merupakan representasi struktur dokumen HTML dalam bentuk objek, sehingga JavaScript bisa berinteraksi dengan elemen HTML.

## Cara memanipulasi elemen DOM pada JavaScript:

### 3. Menambah dan menghapus elemen:

*// Membuat dan menghapus elemen html:*

```
const nama = document.createElement('h1');  
nama.textContent = "Nama saya abyan";  
document.body.appendChild(nama); // ---> ini akan menambahkan elemen <h1> ke dalam body HTML
```

```
// Untuk menghapus elemen yang sudah kita buat  
nama.remove();
```

# Document Object Model (DOM)

Document Object Model (DOM) merupakan representasi struktur dokumen HTML dalam bentuk objek, sehingga JavaScript bisa berinteraksi dengan elemen HTML.

## Cara memanipulasi elemen DOM pada JavaScript:

3. Menambahkan event pada elemen:

```
const btn = document.getElementById('tbl');  
  
btn.addEventListener('click', function() {  
    alert("Tombol sudah di klik!");  
})
```

`addEventListener()` dipakai untuk merespon aksi pengguna, contoh event yang sering digunakan:

1. 'click'
2. 'mouseover'
3. 'keydown'
4. 'input', dll