

# JavaScript 1

Putra Prasetya

# Objective

- ❖ **What is Javascript**
- ❖ **Data Types**
- ❖ **Variables**
- ❖ **Condition**
- ❖ **Looping**
- ❖ **String Literal**
- ❖ **Spread Operator**
- ❖ **Destructuring**

# Apa Itu JavaScript?

- JavaScript adalah **bahasa pemrograman** untuk web.
- JavaScript dapat memperbarui dan mengubah HTML dan CSS.
- JavaScript dapat menghitung, memanipulasi dan memvalidasi data.

# Mengapa Belajar JavaScript ?

JavaScript adalah salah satu dari 3 bahasa yang **harus** dipelajari semua pengembang web :

- HTML untuk menentukan konten halaman web.
- CSS untuk menentukan tata letak halaman web.
- JavaScript untuk memprogram perilaku halaman web.

# Tipe Data

JavaScript mempunyai 8 tipe data :

1. String
2. Number
3. Bigint
4. Boolean
5. Undefined
6. Null
7. Symbol
8. Object

Tipe data Object :

1. An object
2. An array
3. A date

# Contoh Tipe Data

## 1. String

```
pesan = "Halo Dunia !";
```

## 2. Number

```
umur = 23;
```

## 3. BigInt

```
kuadriliun = BigInt("10000000000000000");
```

## 4. Boolean

```
laki-laki = true;
```

## 5. Undefined

```
nilaiTes;
```

## 6. Null

```
biayaTransportasi = null;
```

## 7. Symbol

```
id = Symbol("id");
```

## 8. Object

```
test = {  
  nama: 'Putra',  
  nilai: function(nPertanyaan,  
nBenar) {  
    return 100/nPertanyaan *  
nBenar;  
  }  
}
```

- **Array**

```
hobi = [belajar, berenang,  
bulu tangkis];
```

- **Date**

```
bootcampDimulai = new  
Date("2023-08-07");
```

# Variabel

Variabel JavaScript dapat dideklarasikan dalam 4 cara :

1. Otomatis

```
pesan = "Halo Dunia!"
```

2. Menggunakan var

```
var umur = 23;
```

3. Menggunakan let

```
let nama = "Putra"
```

4. Menggunakan const

```
const laki-laki = true;
```

# Kondisi

Sering kali bila kita menuliskan kode, kita ingin melakukan tindakan yang berbeda untuk kondisi yang berbeda.

Kita dapat menggunakan *conditional statement* untuk melakukan ini :

- Gunakan **if** untuk menentukan blok kode yang akan dieksekusi. Jika kondisi yang ditentukan benar.
- Gunakan **else if** untuk menentukan kondisi baru yang akan diuji, jika kondisi pertama salah.
- Gunakan **else** untuk menentukan blok kode yang akan dieksekusi. Jika semua kondisi **if** tidak dipenuhi.
- Gunakan **switch** untuk menentukan banyak blok kode alternatif yang akan dieksekusi.



# Contoh Kondisi

1. Buat sebuah blok kode yang bisa mengenali bilangan positif, negatif, dan bilangan nol.
2. Buat sebuah blok kode untuk menentukan nama bulan berdasarkan nomor bulannya.

## Contoh Kondisi (2)

```
const angka = -1;

if (angka > 0) {
    console.log("bilangan positif");
} else if (angka < 0) {
    console.log("bilangan negatif");
} else {
    console.log("bilangan nol");
}
```

# Contoh Kondisi (3)

```
const nomorBulan = 11;
switch (nomorBulan) {
  case 1:
    namaBulan = "Januari"; break;
  case 2:
    namaBulan = "Februari"; break;
  case 3:
    namaBulan = "Maret"; break;
  case 4:
    namaBulan = "April"; break;
  case 5:
    namaBulan = "Mei"; break;
  case 6:
    namaBulan = "Juni"; break;
```

```
  case 7:
    namaBulan = "Juli"; break;
  case 8:
    namaBulan = "Agustus"; break;
  case 9:
    namaBulan = "September";
break;
  case 10:
    namaBulan = "Oktober"; break;
  case 11:
    namaBulan = "November"; break;
  case 12:
    namaBulan = "Desember"; break;
}
console.log(namaBulan);
```

# Perulangan

- Perulangan dapat mengeksekusi blok kode beberapa kali.
- Perulangan berguna, jika kita ingin menjalankan kode yang sama berulang kali atau setiap kali dengan nilai yang berbeda.
- JavaScript mendukung berbagai jenis perulangan :
  - for → perulangan melalui blok kode beberapa kali.
  - for..in → perulangan melalui property objek.
  - for..of → perulangan melalui nilai objek yang dapat diubah.
  - while → perulangan melalui blok kode saat kondisi yang ditentukan benar.
  - do..while → blok kode dijalankan terlebih dahulu baru cek kondisi, jika kondisi benar maka blok kode dijalankan kembali.

# For

## Sintak :

```
for (inisialisasi; kondisi; iterasi) {  
    // blok kode yang akan diulang  
}
```

## Contoh :

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

# For in

## Sintak :

```
for (key in object) {  
    // blok kode yang akan diulang  
}
```

## Contoh :

```
let identitas = {  
    nama: "Putra",  
    umur: 23,  
    alamat: "Jakarta Selatan"  
}  
  
for ( i in identitas ) {  
    console.log(i+ ": " +identitas[i]);  
}
```

# For of

## Sintak :

```
for (variabel of iterable) {  
    // blok kode yang akan diulang  
}
```

## Contoh :

```
const alfabet = ['A', 'B', 'C', 'D',  
    'E', 'F'];  
  
for (const elemen of alfabet) {  
    console.log(elemen);  
}
```

# While

## Sintak :

```
while (kondisi) {  
    // blok kode yang akan diulang  
}
```

## Contoh :

```
let i = 1;  
  
while (i < 5) {  
    console.log("Perulangan ke-"+i);  
    i++;  
}
```



# Do While

Sintak :

```
do {  
    // code block to be executed  
} while (condition);
```

Contoh :

```
let i = 6;  
  
do {  
    console.log("Perulangan ke-"+i);  
    i++;  
} while (i < 5);
```

# String Literal

- String Literal / Template Literal menggunakan back-ticks (``) untuk mendefinisikan sebuah string.
- Fungsinya :
  - Menuliskan string multi-baris
  - Menyisipkan kutip di dalam string
  - Interpolasi variable
  - Menyematkan ekspresi dalam string

# String Literal (2)

- Menuliskan string multi-baris

```
const identitas = `  
  nama: Putra,  
  umur: 23,  
  hobi: programming  
`;
```

- Menggunakan kutip di dalam string

```
const pesan = `Dihari jum'at ada kegiatan "Bersih-Bersih"`;
```

- Interpolasi variable

```
const namaDepan="Putra";  
const namaBelakang="Prasetya";  
let selamatDatang = `Halo ${namaDepan}, ${namaBelakang}`;
```

- Menyematkan ekspresi dalam string

```
let s = 10;  
let luasPersegi = `Luas Persegi: ${s*s}`;
```

# Spread Operator

- Spread operator memperluas iterable (seperti array) menjadi lebih banyak elemen.
- Contoh :

```
const array1 = [1,2,3];
```

```
const array2 = [4,5];
```

```
array1 = [...array1, ...array2];
```

```
console.log(array1);
```

# Destructuring

- Destructuring adalah ekspresi JavaScript yang memungkinkan untuk membongkar nilai dari array, atau properti dari objek, ke dalam variabel yang berbeda.
- Contoh :

```
const nama = ["Putra", "Tono", "Nabil", Teguh];  
let nama1 = nama[0];  
let nama2 = nama[1];  
  
console.log(nama1);  
console.log(nama2);
```



**Selesai**