



## DEKLARASI VARIABEL

### VAR

```
var angka = 10
```

### LET

```
let kotaAsal = "Jakarta"
```

### CONST

```
const noKTP = "12345"
```

## ARRAY

### Deklarasi

```
// Menggunakan Array literal
let namaArray = [ element1, element2 ]
// index      0      1

// Menggunakan keyword new
let namaArray = new Array(element1, element2)
// index      0      1
```

### Akses

```
namaArray[nomorIndex]
```

### Destructure

```
let [ variabel1, variabel2 ] = namaArray
```

## TIPE DATA

### String

```
let negara = "Indonesia"
```

### Number

```
let tahun = 2020
```

### Boolean

```
let isChecked = true
```

### Null

```
let data = null
```

### Undefined

```
let nama
```

### Object

```
let person = {
  firstName: "John",
  lastName: "Doe",
  age: 20
}
```

## MATH

### Properti

```
Math.E // Bilangan Euler
Math.LN2 // Log 2
Math.LN10 // Log 10
Math.LOG2E // Log E di Basis 2
Math.LOG10E // Log E di Basis 10
Math.PI // Pi
Math.SQRT1_2 // Akar Kuadrat dari 0.5
Math.SQRT2 // Akar Kuadrat dari 2
```

### Method

```
Math.abs(x) // negatif ke positif
Math.pow(x, y) // bilangan pangkat
Math.sqrt(x) // akar pangkat 2
Math.cbrt(x) // akar pangkat 3
Math.round(x) // pembulatan
Math.floor(x) // pembulatan ke bawah
Math.ceil(x) // pembulatan ke atas
Math.random() // angka acak 0 ~ 1
Math.max(x, ..., n) // mencari angka terbesar
Math.min(x, ..., n) // mencari angka terkecil
```

## USE JS IN HTML

### Internal

```
<body>
...
<script>
  console.log("on page script")
</script>
</body>
```

### Eksternal

```
<body>
...
<script src="./index.js"></script>
</body>
```

## CONDITIONAL

### IF - Else

```
if (condition) {
  // kode yg akan dijalankan
} else if (condition) {
  // kode yg akan dijalankan
} else {
  // kode yg akan dijalankan
}
```

### Switch

```
switch (condition) {
  case 1:
    // kode yg akan dijalankan
    break
  case 2:
    // kode yg akan dijalankan
    break
  default:
    // kode yg akan dijalankan
}
```

## OPERATOR

### Aritmatika

```
+ penjumlahan
- pengurangan
* perkalian
/ pembagian
% modulus (sisa bagi)
** eksponen (pangkat)
++ increment (menambahkan 1)
-- decrement (mengurangi 1)

= Assignment
+= a = a + 1
-= a = a - 2
*= a = a * 2
/= a = a / 2
%= a = a % 2
**= a = a ** 2

Comparison
== sama dengan (cek nilai)
=== sama dengan (cek nilai dan tipe data)
!= tidak sama dengan (cek nilai)
!== tidak sama dengan (cek nilai dan tipe data)
> lebih dari
< kurang dari
>= lebih dari atau sama dengan
<= kurang dari atau sama dengan
? : ternary operator
```

### Logika

```
&& AND (dan) jika bertemu FALSE, maka menjadi FALSE
|| OR (atau) jika bertemu TRUE, maka menjadi TRUE
! NOT (bukan) kebalikan dari suatu kondisi

Rest dan Spread (...)
// Rest
let [ variabel1, variabel2, variabel3, ...rest ] = namaArray
let { properti1, properti2, properti3, ...rest } = namaObjek

// Spread
let newArray = [ element1, element2, ...namaArray ]
let newObj = { objek1, objek2, ...namaObjek }
```

## FUNCTION

### Deklarasi Function

```
function namaFungsi(){
  // Kode yang akan dijalankan
}
```

### Variabel Function

```
let namaVariabelFungsi = function(){
  // Kode yang akan dijalankan
}
```

### Arrow Function

```
let arrowFungsi = () => {
  // Kode yang akan dijalankan
}
```

## LOOP

### For

```
for (statement1, statement2, statement3) {
  // Kode yg akan dijalankan
}
```

### For in (obj)

```
for (property of namaObjek) {
  // Kode yg akan dijalankan
}
```

### For of (arr)

```
for (element of namaVariabel) {
  // Kode yang akan dijalankan
}
```

### While

```
while (condition) {
  // Kode yg akan dijalankan
}
```

### Do While

```
do {
  // Kode yg akan dijalankan
} while (condition)
```

## OBJECT

### Deklarasi

```
// Menggunakan Objek literal
let namaObjek = {
  namaProperti1: nilai1,
  namaProperti2: nilai2
}

// Menggunakan keyword new
let namaObjek = new Object()
namaObjek.namaProperti1 = nilai1
namaObjek.namaProperti2 = nilai2
```

### Akses

```
// Dot Notation
namaObjek.namaProperti1

// Bracket Notation
namaObjek["namaProperti2"]
```

### Destructure

```
let { properti1, properti2 } = namaObjek
```

