



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
```

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 = 1
+=     a = a + 2  → a += 2
-=     a = a - 2  → a -= 2
*=     a = a * 2  → a *= 2
/=     a = a / 2  → a /= 2
%=     a = a % 2  → a %= 2
**=    a = a ** 2  → 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 newObjek = { 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 (propertiObjek in 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
```

DOM

Akses Element

```
document.getElementById("id")
document.getElementsByTagName("tag")
document.getElementsByClassName("class")
document.querySelectorAll("selector")
```

Manipulasi Element

```
// Ubah element
element.innerHTML
element.attribute
element.setAttribute(attribute, nilai)
```

```
// Event Click
element.onclick
```

