

# Rangkuman JavaScript

---

## Topik yang Dibahas

- **Pengantar JavaScript:** Apa itu JavaScript, cara menulis dan menjalankannya di browser.
- **Sintaks Dasar:** Variabel, operator, tipe data.
- **Kontrol Alur:** If...else, switch, loop (for, while).
- **Fungsi:** Cara membuat dan menggunakan fungsi.
- **Event:** Menangani interaksi pengguna seperti klik atau input.
- **DOM (Document Object Model):** Mengakses dan memodifikasi elemen HTML dengan JavaScript.
- **Form Validation:** Validasi data input pengguna.
- **Object dan Array:** Struktur data penting dalam JavaScript.
- **AJAX dan JSON:** Komunikasi dengan server tanpa reload halaman.

## Apa Itu JavaScript?

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat website lebih interaktif dan dinamis.

Contohnya:

- Menampilkan pesan pop-up
- Mengubah konten halaman secara otomatis
- Validasi form
- Animasi elemen, dll

## Fungsi Utama JavaScript di Web

- **Mengubah Konten HTML**  
→ Misalnya mengganti teks paragraf.
- **Mengubah Style CSS**  
→ Contohnya mengubah warna latar belakang saat tombol diklik.
- **Menyembunyikan atau Menampilkan Elemen**  
→ Misalnya menyembunyikan teks saat diklik.

## Menambahkan JavaScript ke HTML

### 1. Internal JavaScript

Ditulis di dalam tag `<script>` di file HTML:

```
html

<!DOCTYPE html>
<html>
<body>

<h2>Contoh JavaScript</h2>

<p id="demo">Hello!</p>

<script>
  document.getElementById("demo").innerHTML = "Halo dari JavaScript!";
</script>

</body>
</html>
```

### 2. External JavaScript

JavaScript ditulis dalam file .js terpisah (contoh `script.js`), lalu dipanggil di HTML:

File HTML:

```
html

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

File `script.js`:

```
javascript

document.getElementById("demo").innerHTML = "Teks dari file eksternal";
```

## JavaScript Where To

Lokasi JavaScript bisa ditempatkan di dalam dokumen HTML agar bisa berjalan dengan benar

### 1. Di Dalam `<head>`

```
html

<!DOCTYPE html>
<html>
<head>
  <script>
    function myFunction() {
      document.getElementById("demo").innerHTML = "Hello from HEAD!";
    }
  </script>
</head>
<body>
  <h2>JavaScript in Head</h2>
  <p id="demo"></p>
  <button onclick="myFunction()">Klik Saya</button>
</body>
</html>
```

#### ⚠ Catatan Penting:

Jika JavaScript berada di `<head>`, maka elemen HTML seperti `<p id="demo">` mungkin belum dimuat saat skrip dijalankan. Solusinya: letakkan fungsi di `<head>`, tapi jalankan saat event terjadi (misalnya onclick).

## 2. Di Dalam `<body>` (biasanya di bawah konten)

JavaScript juga bisa ditulis di bagian bawah sebelum penutup `</body>`. Ini adalah praktik yang umum dan lebih disarankan karena HTML sudah dimuat sepenuhnya sebelum JavaScript dijalankan.

Contoh:

```
html

<!DOCTYPE html>
<html>
<body>
  <h2>JavaScript in Body</h2>
  <p id="demo"></p>
  <button onclick="myFunction()">Klik Saya</button>

  <script>
    function myFunction() {
      document.getElementById("demo").innerHTML = "Hello from BODY!";
    }
  </script>
</body>
</html>
```

## 3. External JavaScript File (file terpisah)

JavaScript bisa ditulis dalam file .js lalu dipanggil ke HTML menggunakan tag `<script src="...">`

Script.js:

```
javascript

function myFunction() {
  document.getElementById("demo").innerHTML = "Hello from External File!";
}
```

HTML:

```
html

<!DOCTYPE html>
<html>
<body>
  <h2>External JavaScript</h2>
  <p id="demo"></p>
  <button onclick="myFunction()">Klik Saya</button>

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

## 4 Cara Umum Menampilkan Output di JavaScript

### 1. Menggunakan `innerHTML`

Menampilkan output langsung ke dalam elemen HTML, seperti paragraf, div, atau span.

```
html

<!DOCTYPE html>
<html>
<body>

<p id="demo"></p>

<script>
  document.getElementById("demo").innerHTML = "Hello World!";
</script>

</body>
</html>
```

Kelebihan:

- Sangat umum digunakan
- Memanipulasi isi halaman dengan mudah

## 2. Menggunakan `document.write()`

Langsung menulis ke halaman HTML.

Contoh:

```
html Salin Edit  
  
<script>  
  document.write("Hello World!");  
</script>
```

### ⚠ Catatan Penting:

- Digunakan hanya saat pemuatan awal halaman.
- Jika digunakan setelah halaman dimuat, maka akan menghapus seluruh isi halaman.

## 3. Menggunakan `window.alert()`

Menampilkan output dalam bentuk kotak pop-up (alert box)

Contoh:

```
javascript Salin Edit  
  
window.alert("Hello World!");
```

📌 Bisa juga ditulis lebih singkat:

```
javascript Salin Edit  
  
alert("Hello!");
```

Kelebihan:

- Bagus untuk debugging cepat atau memberi peringatan.
- Tapi mengganggu pengalaman pengguna jika digunakan terlalu sering.

## JavaScript Statements

Statement adalah instruksi satu baris yang dijalankan oleh JavaScript.

Contoh:

```
let x = 5;
```

```
let y = 6;
```

```
let z = x + y;
```

- Jalankan dari atas ke bawah
- Disarankan menggunakan titik koma (;) untuk memisahkan
- Bisa menulis beberapa statement dalam satu baris dengan titik koma

## JavaScript Syntax

Syntax (sintaks) adalah aturan atau struktur penulisan dalam bahasa pemrograman.

Dalam JavaScript:

- Syntax memberitahu browser bagaimana kode harus dibaca dan dijalankan.
- Jika sintaks salah, maka akan muncul error dan kode tidak bisa dijalankan dengan benar.

### Contoh Sintaks JavaScript Dasar:

```
javascript Salin Edit  
  
let x, y, z;    // Deklarasi variabel  
x = 5;          // Memberi nilai ke x  
y = 6;          // Memberi nilai ke y  
z = x + y;      // Menghitung jumlah x dan y, simpan ke z
```

## Penjelasan Unsur Sintaks JavaScript

### 1. Variabel

Digunakan untuk menyimpan data. JavaScript menggunakan kata kunci seperti:

- **let** (disarankan untuk penggunaan umum)
- **const** (untuk nilai tetap)
- **var** (lama, jarang direkomendasikan)

Contoh:

javascript

Salin

Edit

```
let name = "Budi";  
const PI = 3.14;
```

## 2. Operator

Digunakan untuk melakukan operasi pada nilai:

- Aritmatika: `+`, `-`, `*`, `/`
- Penugasan: `=`
- Perbandingan: `==`, `===`, `>`, `<`

Contoh:

javascript

Salin

Edit

```
let total = 10 + 5;
```

## 3. Nilai (Values)

Nilai bisa berupa:

- Tetap (Literal): seperti angka 10, string "Hello"
- Variabel: menyimpan nilai, seperti x, y

Contoh:

javascript

Salin

Edit

```
let a = 100;           // Literal  
let b = a + 20;        // Variabel + Literal
```

## 4. Komentar

Digunakan untuk memberi catatan di dalam kode (tidak dijalankan oleh program):

javascript

Salin

Edit

```
// Ini komentar satu baris  
  
/* Ini komentar  
   lebih dari satu baris */
```



## 5. Case Sensitive

JavaScript membedakan huruf besar dan kecil.

Contoh:

javascript

Salin

Edit

```
let name = "Ani";  
let Name = "Budi"; // variabel berbeda
```

### Kesimpulan:

**Variabel** `let x = 5;` Menyimpan data

**Operator** `+, -, *` Operasi matematika/logika

**Nilai** `10, "text"` Data literal

**Komentar** `// catatan` Tidak dijalankan

**Case sensitive** `var a ≠ A` Huruf besar/kecil penting