



JAVASCRIPT

swono@polibatam.ac.id

DEFINISI

- Bahasa pemrograman yang digunakan untuk membuat halaman web menjadi lebih interaktif dan dinamis.
- JavaScript adalah bahasa pemrograman yang berjalan di sisi **klien (client-side)**, yang berarti dijalankan di browser pengguna, meskipun saat ini juga bisa digunakan di sisi **server (server-side)** dengan platform seperti **Node.js**.
- Programs diembedded pada HTML Web page, untuk mengidentifikasi komponen program

Contoh : `<script type="text/javascript"> ... </script>`

KOMPONEN UTAMA

Variabel

- Variabel digunakan untuk menyimpan data atau nilai yang bisa digunakan kembali dalam kode. Ada tiga kata kunci utama yang digunakan untuk mendeklarasikan variabel:
- **var:** Dapat digunakan di seluruh kode, tapi memiliki cakupan yang lebih longgar
- **let:** Variabel dengan cakupan blok, digunakan untuk nilai yang bisa berubah.
- **const:** Variabel dengan cakupan blok, digunakan untuk nilai yang konstan (tidak bisa diubah setelah diinisialisasi).
- Contoh :

```
let name = "Alice";  
const age = 25;  
var nim = 12345;
```

Tipe Data

Tipe data menentukan jenis nilai yang bisa disimpan dalam variabel. Tipe data umum di JavaScript:

- String: Teks, dideklarasikan dengan tanda kutip (" ", ' ').
- Number: Angka, bisa berupa integer atau float.
- Boolean: Nilai true atau false.
- Object: Koleksi data dalam bentuk pasangan kunci-nilai.
- Array: Daftar data yang diatur dalam urutan.

```
let str = "Hello World";  
let num = 42;  
let isActive = true;  
let person = { name: "Alice", age: 25 };  
let fruits = ["Apple", "Banana", "Cherry"];
```

Operator

Operator digunakan untuk melakukan operasi pada nilai-nilai atau variabel.

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

DOM (DOCUMENT OBJECT MODEL)

- Representasi struktural dari halaman web yang memungkinkan JavaScript untuk berinteraksi dengan dan memanipulasi konten, struktur, serta gaya dari sebuah halaman HTML.
- DOM mengubah dokumen HTML menjadi pohon objek yang dapat diakses dan dimodifikasi oleh JavaScript.
- Setiap elemen tersebut diwakili sebagai objek dalam DOM, dan elemen-elemen ini bisa dimanipulasi, ditambahkan, dihapus, atau diubah dengan menggunakan JavaScript.

METODE DOM SECARA UMUM

Mengakses Elemen DOM

- **getElementById()** : Mengakses elemen berdasarkan ID, metode ini sangat umum karena ID bersifat unik di setiap halaman.
- **getElementsByClassName()** : Mengembalikan semua elemen yang memiliki class tertentu sebagai koleksi (HTMLCollection).
- **getElementsByTagName()** : Mengambil semua elemen dengan tag tertentu (misalnya, semua <p> atau <div>) sebagai HTMLCollection.
- **querySelector()** : Mengembalikan elemen pertama yang cocok dengan selektor CSS yang diberikan.
- **querySelectorAll()** : Mengembalikan semua elemen yang cocok dengan selektor CSS sebagai NodeList.

Mengubah atau Memanipulasi Konten Elemen

- **innerHTML** : Mengambil atau mengubah konten HTML dari sebuah elemen.
- **textContent** : Mengambil atau mengubah teks dari elemen, termasuk teks yang tersembunyi, tetapi tanpa HTML.

Memanipulasi Atribut Elemen

- **setAttribute()** : Menambahkan atau memperbarui atribut elemen.
- **getAttribute()** : Mengambil nilai dari atribut yang ditentukan pada elemen.
- **removeAttribute()** : Menghapus atribut dari elemen.

Mengubah Gaya Elemen

- **Style** : Mengubah gaya CSS langsung pada elemen.

Mengelola Elemen DOM (Menambah, Menghapus, dan Mengganti Elemen)

- **createElement()** : Membuat elemen HTML baru.
- **appendChild()** : Menambahkan elemen baru ke dalam elemen lain sebagai anak terakhir.
- **removeChild()** : Menghapus elemen anak dari elemen induk.
- **replaceChild()** : Mengganti elemen anak dengan elemen baru.

CONTOH

```
<script>
  // Mendapatkan elemen DOM
  const addItemButton = document.getElementById('addItemButton');
  const newItemInput = document.getElementById('newItemInput');
  const itemList = document.getElementById('itemList');

  // Membuat elemen <li> baru
  const newItem = document.createElement('li');
  newItem.textContent = newItemText + ' '; // Menambahkan teks ke item
  // Menambahkan tombol hapus ke item baru
  newItem.appendChild(removeButton);
  // Menambahkan item baru ke dalam daftar <ul>
  itemList.appendChild(newItem);
</script>
```

EVENT HANDLING

- JavaScript memungkinkan situs web merespons aksi pengguna seperti klik, hover, atau input keyboard.
- Misalnya, membuka menu dropdown, menampilkan notifikasi saat tombol diklik, atau menampilkan tooltip saat mouse diarahkan ke elemen tertentu.

METODE DAN TEKNIK EVENT

Event Listener

- **addEventListener()** : Metode ini menambahkan event listener pada elemen, yang akan mengeksekusi fungsi callback ketika event tertentu terjadi.
- **removeEventListener()** : Metode ini menghapus event listener dari elemen. Event listener harus ditambahkan dengan cara yang sama (fungsi dan parameter) agar bisa dihapus.

Contoh :

```
function handleClick() {  
    alert("Button clicked!");  
}  
button.addEventListener("click", handleClick);  
button.removeEventListener("click", handleClick);
```

Event Types

Berikut adalah beberapa tipe peristiwa umum yang bisa ditangani di JavaScript:

- click: Terjadi ketika elemen diklik.
- dblclick: Terjadi ketika elemen diklik dua kali.
- mousedown: Terjadi ketika tombol mouse ditekan.
- mouseup: Terjadi ketika tombol mouse dilepas.
- mouseover: Terjadi ketika mouse bergerak di atas elemen.
- mouseout: Terjadi ketika mouse bergerak keluar dari elemen.

```
let box = document.getElementById("box");
box.addEventListener("mouseover", function () {
  box.style.backgroundColor = "lightblue";
});
box.addEventListener("mouseout", function () {
  box.style.backgroundColor = "";
});
```

Form Events

- submit: Terjadi ketika formulir disubmit.
- change: Terjadi ketika nilai elemen form berubah.
- focus: Terjadi ketika elemen form mendapat fokus.
- blur: Terjadi ketika elemen form kehilangan fokus.

```
let form = document.getElementById("myForm");

form.addEventListener("submit", function (event) {
  event.preventDefault();
  alert("Form submitted!");
});
```

PENEMPATAN JAVASCRIPT

Inline JavaScript

- JavaScript dapat disisipkan langsung dalam atribut HTML menggunakan event handler.
- Ini adalah metode yang sering digunakan untuk aksi sederhana.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Inline JavaScript</title>
</head>
<body>
  <button onclick="alert('Hello, World!')">Klik Saya</button>
</body>
</html>
```

Internal JavaScript

- JavaScript dapat disisipkan langsung dalam tag `<script>` di dalam dokumen HTML.
- Sering digunakan untuk kode JavaScript yang spesifik untuk halaman tertentu.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Internal JavaScript</title>
</head>
<body>
  <button id="myButton">Klik Saya</button>
  <p id="demo">Teks ini akan diubah.</p>
  <script>
    document.getElementById("myButton").addEventListener("click", function () {
      document.getElementById("demo").textContent = "Teks telah diubah!";
    });
  </script>
</body>
</html>
```


External JavaScript

- JavaScript dapat disimpan dalam file terpisah dengan ekstensi .js dan dihubungkan ke dokumen HTML menggunakan tag `<script>` dengan atribut `src`.
- Memungkinkan pemisahan kode JavaScript dari HTML, membuatnya lebih terorganisir.

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>External JavaScript</title>
  <script src="script.js" defer></script>
</head>
<body>
  <button id="myButton">Klik Saya</button>
  <p id="demo">Teks ini akan diubah.</p>
</body>
</html>
```

kode html

```
<script>
document.addEventListener("DOMContentLoaded", function() {
  document.getElementById("myButton").addEventListener("click", function() {
    document.getElementById("demo").textContent = "Teks telah diubah!";
  });
});
```

script.js

JENDELA DIALOG

- Jendela kecil yang muncul di atas halaman web dan digunakan untuk berinteraksi dengan pengguna.
- Fungsi-fungsi ini memungkinkan untuk menampilkan pesan, meminta input, atau meminta konfirmasi dari pengguna.
- JavaScript menyediakan beberapa fungsi bawaan untuk membuat jendela dialog ini.

JENIS-JENIS JENDELA DIALOG

Alert

- Jendela **alert** digunakan untuk menampilkan pesan sederhana kepada pengguna.
- Memberikan informasi tanpa interaksi lebih lanjut selain menutup kotak dialog tersebut.
- Pengguna hanya bisa menutup kotak dialog dengan mengklik tombol "OK".

```
<body>
  <h1>Contoh Alert</h1>
  <button onclick="showAlert()">Tampilkan Alert</button>
  <script>
    function showAlert() {
      alert("Ini adalah pesan alert!");
    }
  </script>
</body>
```

New Tab New Tab Gmail YouTube Maps sid.pol

Contoh Alert

Tampilkan Alert

This page says

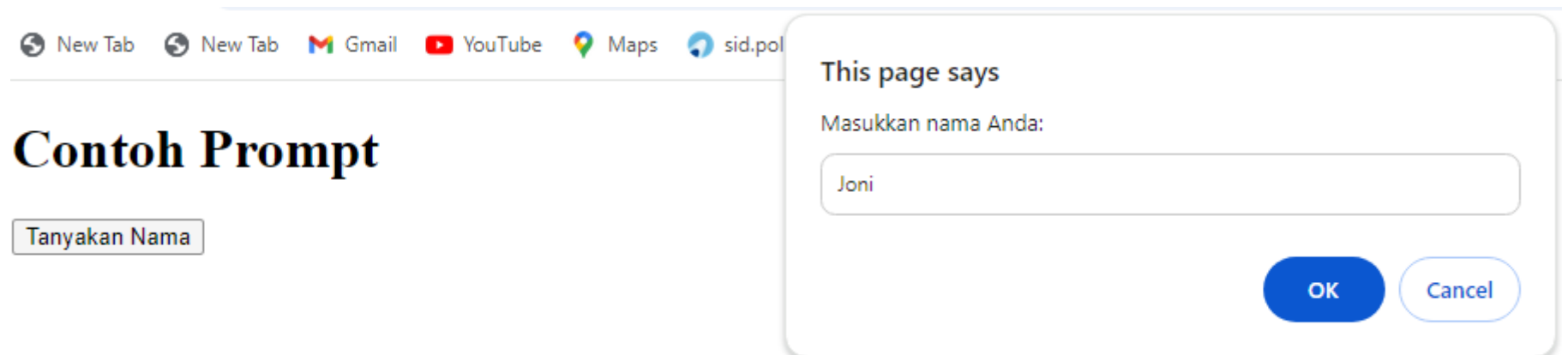
Ini adalah pesan alert!

OK

Prompt (Input dari Pengguna)

- Jendela **prompt** digunakan untuk meminta input dari pengguna.
- Pengguna dapat mengetikkan sesuatu ke dalam kotak teks, dan hasil input tersebut dapat digunakan dalam kode JavaScript.

```
<body>
  <h1>Contoh Prompt</h1>
  <button onclick="askName()">Tanyakan Nama</button>
  <script>
    function askName() {
      var name = prompt("Masukkan nama Anda:", "Nama");
      if (name != null && name != "") {
        alert("Halo, " + name + "! Selamat datang.");
      }
    }
  </script>
</body>
```



Confirm

- Jendela confirm digunakan untuk meminta konfirmasi dari pengguna.
- Kotak dialog ini menampilkan pesan dengan dua tombol: OK dan Cancel.
- Hasil dari kotak dialog ini adalah boolean (true jika pengguna mengklik "OK", dan false jika mereka mengklik "Cancel").

```
<body>
  <h1>Contoh Confirm</h1>
  <button onclick="confirmAction()">Konfirmasi Aksi</button>
  <script>
    function confirmAction() {
      var result = confirm("Apakah Anda yakin ingin melanjutkan?");
      if (result) {
        alert("Anda memilih OK!");
      } else {
        alert("Anda memilih Cancel!");
      }
    }
  </script>
</body>
```

New Tab New Tab Gmail YouTube Maps sid.pol

Contoh Confirm

Konfirmasi Aksi

This page says

Apakah Anda yakin ingin melanjutkan?

OK

Cancel

MANFAAT JAVASCRIPT

- **Interaktivitas:** JavaScript memungkinkan pembuatan elemen interaktif pada halaman web, seperti tombol, formulir, animasi, dan efek visual.
- **Validasi Formulir:** JavaScript bisa digunakan untuk memvalidasi data yang diinput oleh pengguna di formulir sebelum data tersebut dikirim ke server.
- **Pembuatan Grafik dan Visualisasi Data:** Dengan bantuan library seperti **Chart.js** atau **D3.js**, JavaScript memungkinkan pembuatan grafik interaktif, visualisasi data, dan dashboard yang menampilkan data dengan cara yang menarik dan mudah dipahami oleh pengguna.
- **Penyimpanan Data di Sisi Klien:** JavaScript bisa digunakan untuk menyimpan data di browser pengguna, baik sementara (session storage) maupun secara persisten (local storage), seperti menyimpan item dalam keranjang belanja, menyimpan preferensi pengguna, atau melacak status login tanpa memerlukan database eksternal

TERIMAKASIH