

BAB V

PEMBELAJARAN JAVASCRIPT

A. Pengenalan JavaScript

5.1.1 Apa Itu JavaScript?

JavaScript adalah bahasa pemrograman yang digunakan untuk membuat halaman web lebih dinamis dan interaktif. JavaScript dapat digunakan untuk menangani event (seperti klik tombol), memvalidasi input form, mengubah elemen HTML secara real-time, dan bahkan mengembangkan aplikasi berbasis web yang kompleks. Dengan JavaScript, pengembang dapat membuat pengalaman pengguna yang lebih menarik dengan berbagai fitur seperti animasi, validasi form otomatis, serta komunikasi dengan server menggunakan AJAX untuk mendapatkan data tanpa perlu me-refresh halaman. Selain itu, JavaScript juga memungkinkan manipulasi Document Object Model (DOM), yang memberikan kontrol penuh terhadap elemen-elemen HTML.

JavaScript juga memiliki berbagai pustaka dan framework yang mempermudah pengembangan aplikasi web. Contohnya adalah jQuery, sebuah pustaka JavaScript yang mempermudah manipulasi elemen HTML dan AJAX. Selain itu, ada pula React, Vue.js, dan Angular yang merupakan framework modern yang banyak digunakan dalam pengembangan aplikasi berbasis Single Page Application (SPA). Dengan bantuan framework ini, pengembang dapat membangun aplikasi dengan arsitektur yang lebih terstruktur dan efisien.

Selain digunakan di sisi klien (frontend), JavaScript juga dapat digunakan untuk pengembangan di sisi server (backend) dengan bantuan Node.js. Dengan Node.js, pengembang dapat membuat aplikasi server-side yang cepat dan scalable. Hal ini menjadikan JavaScript sebagai bahasa pemrograman full-stack, yang memungkinkan pengembang membangun aplikasi web dari frontend hingga backend hanya dengan satu bahasa.

Seiring perkembangan teknologi, JavaScript juga digunakan dalam pengembangan aplikasi mobile dan desktop. Dengan framework seperti React Native, pengembang dapat membangun aplikasi mobile berbasis iOS dan Android menggunakan JavaScript. Untuk pengembangan aplikasi desktop, Electron memungkinkan pengembang membuat aplikasi lintas platform dengan teknologi web.

Dengan fleksibilitas dan dukungan komunitas yang besar, JavaScript terus berkembang menjadi salah satu bahasa pemrograman yang paling banyak digunakan di dunia. Hampir semua website modern menggunakan JavaScript untuk meningkatkan interaktivitas dan fungsionalitasnya. Oleh karena itu, menguasai JavaScript menjadi keterampilan yang sangat penting bagi siapa saja yang ingin terjun ke dunia pengembangan web dan teknologi informasi secara umum.

5.2.1 Cara Menjalankan JavaScript

JavaScript dapat dijalankan langsung di browser modern seperti Google Chrome, Mozilla Firefox, Microsoft Edge, dan Safari. Setiap browser memiliki mesin JavaScript sendiri untuk menjalankan kode JavaScript dengan cepat dan efisien. Misalnya, Google Chrome menggunakan mesin V8, Mozilla Firefox menggunakan SpiderMonkey, dan Microsoft Edge menggunakan Chakra. Mesin-mesin ini memungkinkan JavaScript untuk dieksekusi dengan performa tinggi, sehingga pengalaman pengguna menjadi lebih baik.

Untuk menyisipkan JavaScript ke dalam halaman HTML, terdapat tiga cara utama yang dapat digunakan, yaitu Inline JavaScript, Internal JavaScript, dan External JavaScript. Masing-masing metode ini memiliki kelebihan dan kekurangan tergantung pada kebutuhan pengembangan.

a. JavaScript di dalam Elemen HTML (Inline JavaScript)

Inline JavaScript memungkinkan kita menulis kode JavaScript langsung dalam atribut HTML menggunakan event seperti onclick,

onmouseover, atau onchange. Cara ini sering digunakan untuk menangani event sederhana seperti menampilkan pesan pop-up atau mengubah elemen HTML tanpa perlu menulis kode di dalam tag `<script>`.

Contoh:

```
<button onclick="alert('Halo! Ini dari JavaScript')">Klik  
Saya</button>
```

Ketika tombol diklik, sebuah popup alert akan muncul dengan pesan "Halo! Ini dari JavaScript".

b. JavaScript di dalam Tag `<script>` (Internal JavaScript)

Internal JavaScript memungkinkan kita menulis kode JavaScript langsung di dalam tag `<script>` dalam file HTML. Cara ini lebih terstruktur dibandingkan Inline JavaScript karena semua kode JavaScript dikumpulkan di satu tempat, biasanya di bagian `<head>` atau sebelum penutupan tag `<body>`. Dengan metode ini, pengembang dapat dengan mudah mengelola kode tanpa mencampurnya dengan elemen HTML, sehingga halaman lebih rapi dan mudah dipelihara. Internal JavaScript sangat berguna untuk proyek dengan skala kecil hingga menengah yang memerlukan interaksi dengan berbagai elemen HTML dalam satu halaman tanpa memerlukan file JavaScript eksternal.

Contoh:

```
<html>  
  
<head>  
  
  <title>Belajar JavaScript</title>  
  
</head>
```

```
<body>

  <h1 id="judul">Selamat Datang</h1>

  <button onclick="ubahTeks()">Ubah Teks</button>


  <script>

    function ubahTeks() {

      document.getElementById("judul").innerHTML = "Teks telah
berubah!";

    }

  </script>

</body>
</html>
```

Kode ini mengubah teks pada elemen <h1> saat tombol diklik.

C. JavaScript di dalam File Eksternal (External JavaScript)

External JavaScript adalah metode di mana kode JavaScript disimpan dalam file terpisah dengan ekstensi .js, lalu dipanggil dalam file HTML menggunakan tag <script src="script.js"></script>. Metode ini sangat disarankan untuk proyek berskala besar karena memungkinkan pemisahan kode JavaScript dari HTML, sehingga meningkatkan keterbacaan dan kemudahan pemeliharaan kode. Dengan menggunakan External JavaScript, kita dapat menggunakan

satu file JavaScript untuk beberapa halaman HTML sekaligus. Ini mengurangi duplikasi kode dan memungkinkan pengembang untuk memperbaiki fungsi JavaScript hanya dari satu tempat tanpa perlu mengedit setiap halaman HTML yang menggunakannya.

Contoh :

Simpan kode dibawah ini dengan nama **file script.js**

```
function salam() {  
    alert("Selamat datang di website kami!");  
}
```

Dan simpan kode HTML dibawah ini dengan nama salam.html silahkan anda akses file tersebut untuk melihat hasilnya.

B. Variabel dan Tipe Data dalam JavaScript

5.3.1 Variabel dalam JavaScript

Variabel adalah tempat penyimpanan data dalam JavaScript yang memungkinkan kita menyimpan, mengubah, dan mengambil data di dalam program. JavaScript menyediakan tiga cara utama untuk mendeklarasikan variabel, yaitu var, let, dan const. Pemilihan jenis variabel ini penting untuk menghindari kesalahan dalam pengelolaan data, terutama saat bekerja dengan skala proyek yang lebih besar.

a. Penggunaan let untuk deklarasi variable

let menjadi metode standar untuk mendeklarasikan variabel yang dapat berubah. let memiliki block scope, yang berarti variabel yang dideklarasikan di dalam {} hanya berlaku di dalam blok tersebut dan tidak dapat diakses di luar blok.

Contoh:

```
function contohLet() {  
    let y = 10;  
    if (true) {  
        let y = 20;  
        console.log("Nilai dalam blok:", y);  
    }  
    console.log("Nilai luar blok:", y);  
}  
contohLet();
```

b. Penggunaan const untuk deklarasi variable

const digunakan untuk mendeklarasikan variabel yang nilainya tidak dapat diubah setelah dideklarasikan. const juga memiliki block scope seperti let

Contoh :

```
const z = 100;  
z = 200; // Akan menyebabkan error
```

5.4.1 Tipe Data dalam JavaScript

JavaScript adalah bahasa pemrograman yang bersifat dinamis, artinya kita tidak perlu mendeklarasikan tipe data secara eksplisit seperti dalam bahasa pemrograman lain. JavaScript secara otomatis menentukan tipe data berdasarkan nilai yang diberikan pada variabel. Secara umum, terdapat dua kategori utama dalam tipe data JavaScript, yaitu tipe data primitif dan tipe data non-primitif (referensi) kemudian untuk tipe data Non primitif (referensi) menyimpan referensi ke lokasi memori, bukan nilai langsung. Secara sederhana tipe data dalam JavaScript dibagi menjadi primitif (String, Number, Boolean, Undefined, Null) dan non-primitif (Object, Array, Function). Memahami tipe data sangat penting dalam pengelolaan variabel dan pengembangan aplikasi berbasis JavaScript.

Contoh :

```
let nama = "Budi"; // String
let umur = 30; // Number
let isStudent = false; // Boolean
let buah = ["Apel", "Mangga", "Jeruk"]; // Array
let mahasiswa = { nama: "Rina", usia: 20, jurusan: "Informatika" };
// Object

console.log(nama); // Output: Budi
console.log(buah[1]); // Output: Mangga
console.log(mahasiswa.jurusan); // Output: Informatika
```

C. Pembuatan Kalkulator Sederhana

5.5.1 Deskripsi Studi Kasus

Pada bagian ini, kita akan membuat sebuah **kalkulator sederhana** menggunakan HTML, CSS, dan JavaScript, silahkan anda ketik kode dibawah ini dan simpan dalam extensi html.

```
<html>

<head>

  <title>Kalkulator Sederhana</title>

</head>

<body>

  <h2>Kalkulator</h2>

  <input type="number" id="num1">

  <input type="number" id="num2">

  <button onclick="hitung()">Hitung</button>

  <p>Hasil: <span id="hasil"></span></p>

  <script>

    function hitung() {

      let num1 = document.getElementById("num1").value;

      let num2 = document.getElementById("num2").value;
```



```

    let hasil = parseInt(num1) + parseInt(num2);

    document.getElementById("hasil").innerText = hasil;

}

</script>

</body>

</html>

```

5.6.1 Tugas buat program javascript sederhana

Buat sebuah halaman web yang memungkinkan pengguna untuk memasukkan NIM (Nomor Induk Mahasiswa) dan nilai mata kuliah. Setelah pengguna menekan tombol Cek Nilai, sistem akan menentukan huruf mutu berdasarkan skala berikut:

Rentang Nilai	Huruf Mutu
80 - 100	A
70 - 79	B
60 - 69	C
50 - 59	D
0 - 49	E

Jika nilai yang dimasukkan di luar rentang 0 - 100, tampilkan pesan error "Nilai tidak valid!", ketentuan tugas sebagai berikut :

- Buat struktur HTML dengan input untuk NIM dan nilai mata kuliah serta tombol Cek Nilai.
- Gunakan JavaScript untuk menangani event klik tombol.
- Gunakan kondisi if-else untuk menentukan huruf mutu berdasarkan rentang nilai yang dimasukkan.
- Tampilkan huruf mutu atau pesan error di bawah form setelah tombol ditekan.