

LAPORAN RESMI
MODUL IV
JAVASCRIPT
PEMROGRAMAN BERBASIS WEB



NAMA	: ANNA MAULINA
N.R.P	: 230441100178
DOSEN	: SRI HERAWATI, S.Kom, M.Kom
ASISTEN	: RACHMAD AZIZ FAZARIKHA
TGL PRAKTIKUM	: 10 MEI 2024

Disetujui : 16 Mei 2024
Asisten

RACHMAD AZIZ FAZARIKHA
22.04.411.00080



LABORATORIUM BISNIS INTELIJEN SISTEM
PRODI SISTEM INFORMASI
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS TRUNOJOYO MADURA

BAB I

PENDAHULUAN

1.1 Latar Belakang

JavaScript merupakan salah satu bahasa pemrograman yang paling populer di dunia saat ini. Dikenal sebagai bahasa pemrograman yang berbasis web, JavaScript digunakan untuk membuat halaman web menjadi interaktif dan dinamis.

Pada dasarnya, JavaScript digunakan untuk mengontrol perilaku elemen-elemen HTML dan CSS pada sebuah halaman web. Dengan JavaScript, pengembang web dapat menambahkan fitur-fitur seperti validasi formulir, animasi, manipulasi data, dan banyak lagi.

Salah satu konsep penting dalam JavaScript adalah DOM (Document Object Model). DOM merupakan representasi struktur hierarkis dokumen HTML atau XML, yang memungkinkan JavaScript untuk mengakses dan memanipulasi elemen-elemen pada halaman web. Dengan menggunakan DOM, pengembang dapat secara dinamis mengubah konten, atribut, dan gaya dari elemen-elemen HTML.

Pemahaman tentang event listener juga menjadi kunci dalam pengembangan web dengan JavaScript. Event listener memungkinkan JavaScript untuk merespons interaksi pengguna seperti klik mouse, penekanan tombol keyboard, atau pengiriman formulir. Dengan menggunakan event listener, pengembang dapat membuat halaman web yang responsif dan interaktif, meningkatkan pengalaman pengguna secara keseluruhan.

Melalui pemahaman tentang JavaScript dasar, DOM, dan event listener, pengembang web dapat membangun aplikasi web yang kuat dan dinamis. Pada bab ini, kita akan menjelajahi konsep-konsep dasar tersebut secara lebih mendalam, serta menerapkan mereka dalam pembuatan aplikasi web yang nyata.

1.2 Tujuan

- Mahasiswa mampu memahami dasar pemrograman JavaScript.
- Mahasiswa mampu memahami Event dan DOM Manipulation.

BAB II

DASAR TEORI

2.1 JavaScript Dasar

2.1.1 Pengenalan dan Sejarah Singkat JavaScript

JavaScript, bahasa pemrograman yang sangat populer dalam pengembangan web, pertama kali dikembangkan oleh Brendan Eich di Netscape Communications pada tahun 1995. Awalnya dikenal sebagai Mocha dan kemudian LiveScript, sebelum akhirnya dinamai JavaScript. Pada tahun 1997, JavaScript diajukan sebagai standar ECMAScript kepada European Computer Manufacturers Association (ECMA). Sejak itu, JavaScript terus berkembang dengan penambahan fitur baru dan peningkatan kinerja melalui rilis ECMAScript yang berkala, seperti ES6 (ECMAScript 2015).

Dengan kemampuannya untuk menambahkan interaktivitas ke halaman web, JavaScript menjadi bahasa pemrograman yang sangat populer. Selain digunakan dalam pengembangan web, JavaScript juga digunakan dalam pengembangan aplikasi mobile, desktop, permainan, dan bahkan di bidang pengembangan server (Node.js). JavaScript saat ini menjadi salah satu elemen kunci dalam ekosistem pengembangan web modern, bersama dengan HTML dan CSS, memungkinkan pembuatan aplikasi web yang kompleks dan interaktif.

2.1.2 Variable dan Tipe Data

1. Variable

Variabel digunakan untuk menyimpan data dalam JavaScript. Mendeklarasikan variabel dilakukan dengan menggunakan kata kunci **var**, **let**, atau **const**.

Penjelasan :

- **var** dan **let**: digunakan untuk mendefinisikan bahwa suatu variabel tersebut nilainya dapat berubah.
- **const**: digunakan untuk mendefinisikan bahwa suatu variabel tersebut nilainya tidak dapat berubah (konstan).

Syntax umum :

- **var/let/const** [nama_variabel] = nilai;

Contoh :

- **var** namaLengkap = "otong surotong";
- **let** nama_panggilan = 'surot';
- **const** PI = 3.14;

2. Tipe Data

JavaScript memiliki beberapa tipe data dasar seperti **string**, **number**, **boolean**, **null**, **undefined**, **array**, dan **object**. Setiap tipe data memiliki peran dan karakteristiknya sendiri.

Penjelasan :

- **string** : digunakan untuk merepresentasikan teks atau urutan karakter. String dituliskan di dalam tanda kutip, baik tunggal (' ') maupun ganda (" ").
- **number** : digunakan untuk merepresentasikan nilai numerik. Ini bisa berupa bilangan bulat (integer) atau bilangan pecahan (floating point).
- **boolean** : digunakan untuk merepresentasikan 2 nilai yaitu (true) atau (false).
- **null** : digunakan untuk menunjukkan bahwa sebuah variabel tidak memiliki nilai atau tidak memiliki nilai yang valid. Secara harfiah, null berarti "tidak ada nilai".
- **undefined** : digunakan untuk menunjukkan bahwa sebuah variabel telah dideklarasikan tetapi tidak memiliki nilai yang diberikan.
- **array** : digunakan untuk menyimpan kumpulan nilai dalam satu variabel. Elemen-elemen dalam array diindeks, dimulai dari indeks 0.
- **object** : digunakan untuk merepresentasikan sebuah objek, yang terdiri dari kumpulan pasangan kunci-nilai (key-value pairs). Objek dapat memiliki properti dan metode.

Tipe Data	Contoh Nilai
string	"otong surotong", 'surot'
number	1, 3.14, -10.5
boolean	true, false

array	<code>["surotong", 1, false, 'surot']</code>
object	<pre> { namaLengkap: "otong surotong", nama_panggilan: 'surot', isAktif: true } </pre>

Setiap tipe data dalam JavaScript memiliki karakteristik dan penggunaan masing-masing, dan memahami mereka secara mendalam sangat penting dalam pengembangan aplikasi JavaScript yang efektif.

2.1.3 Operator dan Ekspresi

1. Operator

Operator adalah simbol atau kata kunci yang digunakan untuk melakukan operasi pada operand (nilai atau variabel). Di JavaScript, ada beberapa jenis operator yang berbeda.

Diantaranya :

- **Operator aritmatika** : digunakan untuk melakukan operasi matematika pada operand. Contoh (`+`, `-`, `*`, `/`, `%`).
- **Operator perbandingan** : digunakan untuk membandingkan dua nilai atau variabel dan menghasilkan nilai boolean. Contoh (`<`, `<=`, `>`, `>=`, `==`, `!=`, `===`, `!==`).
- **Operator logika** : digunakan untuk menggabungkan atau membalikkan kondisi logika dan menghasilkan nilai boolean. Contoh (`&&`, `||`, `!`).
- **Operator penugasan** : digunakan untuk memberikan nilai kepada variabel. Contoh (`=`, `+=`, `-=`, `*=`, dll).

2. Ekspresi

Ekspresi adalah kombinasi variabel, nilai, operator, dan/atau fungsi yang dievaluasi menjadi nilai tunggal. Ekspresi dapat berupa ekspresi aritmatika, ekspresi logika, atau ekspresi yang lebih kompleks.

Contoh ekspresi:

→ **Ekspresi aritmatika** : `(2 + 3)`

- **Ekspresi perbandingan** : (`2 >= -2`)
- **Ekspresi logika** : (`true && false`)
- **Ekspresi penugasan** : (`x *= 10`)
- **Ekspresi fungsi** : (`Math.max(5, 10)`)

Ekspresi dievaluasi menjadi nilai tunggal sesuai dengan aturan dan operasi yang dinyatakan di dalamnya. Ekspresi ini sering digunakan dalam kondisi, perulangan, dan pernyataan lainnya dalam JavaScript untuk mengontrol alur eksekusi program.

Memahami operator dan ekspresi adalah kunci dalam pemrograman JavaScript, karena mereka memungkinkan kita untuk melakukan berbagai macam operasi dan mengendalikan alur eksekusi program secara efektif.

2.1.4 Struktur Kontrol

Dalam pemrograman JavaScript, struktur kontrol digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi tertentu atau untuk mengulangi blok kode tertentu. Dengan memahami struktur kontrol, Anda dapat membuat program yang lebih dinamis dan responsif. Berikut adalah beberapa konsep struktur kontrol yang penting:

1. Seleksi Kondisi (**if**, **else if**, **else**)

Seleksi kondisi digunakan untuk menjalankan blok kode tertentu berdasarkan kondisi yang diberikan. Pernyataan **if** digunakan untuk mengevaluasi ekspresi boolean, jika ekspresi tersebut benar (**true**), maka blok kode di dalam **if** akan dieksekusi. Jika ekspresi tersebut salah (**false**), maka blok kode di dalam **else** akan dieksekusi. **else if** dapat digunakan untuk mengevaluasi beberapa kondisi.

Syntax umum :

```

• if (kondisi_1) {
    // pernyataan-1;
} else if (kondisi_2) {
    // pernyataan-2;
if (kondisi_3) {
    // pernyataan-3;

```

```

    }
  } else {
    // pernyataan-4;
  }

```

2. Pengulangan (for, while, do-while)

Pengulangan digunakan untuk mengeksekusi blok kode berulang kali. JavaScript menyediakan beberapa jenis pengulangan, yaitu **for**, **while**, dan **do-while**. Loop **for** digunakan ketika Anda mengetahui berapa kali iterasi yang dibutuhkan, **while** digunakan ketika Anda hanya ingin menjalankan loop selama kondisi tertentu terpenuhi, dan **do-while** digunakan ketika Anda ingin menjalankan loop setidaknya sekali sebelum mengevaluasi kondisi.

Syntax umum :

- **for** → **for** (inisialisasi; kondisi; iterasi) {
 // pernyataan yang ingin diulang;
 }
- **while** → **while** (kondisi) {
 // Blok kode yang akan diulang
 // Perhatikan bahwa kondisi akan
 dievaluasi sebelum setiap iterasi
 }
- **do-while** → **do** {
 // Blok kode yang akan diulang
 // Perhatikan bahwa blok kode akan
 dijalankan setidaknya sekali sebelum
 kondisi dievaluasi
 } **while** (kondisi);

Dengan memahami dan menguasai struktur kontrol ini, Anda dapat mengendalikan alur eksekusi program dengan lebih efisien dan fleksibel dalam pengembangan aplikasi JavaScript.

2.1.5 Fungsi

Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu. Dengan menggunakan fungsi, Anda dapat mengorganisir kode Anda menjadi

bagian-bagian yang lebih kecil dan dapat digunakan kembali. Dalam JavaScript, fungsi sangat penting dan dapat digunakan untuk berbagai tujuan, mulai dari mengeksekusi tugas sederhana hingga mengelola logika yang kompleks dalam aplikasi Anda.

Kegunaan fungsi :

- Fungsi digunakan untuk mengelompokkan serangkaian pernyataan menjadi satu unit yang dapat dipanggil.
- Saat fungsi dipanggil, blok kode di dalamnya dieksekusi.

Syntax umum :

```
→ function namaFungsi(parameter1, parameter2, ...) {  
    // Blok kode yang akan dieksekusi saat fungsi dipanggil  
}
```


BAB III

TUGAS PENDAHULUAN

3.1 Soal

1. Apa yang kalian ketahui tentang JavaScript? Jelaskan!
2. Apakah Java dengan JavaScript itu berbeda? Jelaskan!
3. Apa itu ECMAScript? Apa bedanya dengan JavaScript? Jelaskan!
4. Apakah kalian mengetahui keyword untuk mendeklarasikan variable dalam JavaScript? Jelaskan!

3.2 Jawaban

1. JavaScript adalah bahasa pemrograman yang sering digunakan untuk mengembangkan aplikasi web interaktif. JavaScript berjalan di sisi klien, yang berarti kodenya dieksekusi di dalam peramban web pengguna. Dengan kemampuannya untuk memanipulasi struktur dan konten halaman web melalui DOM (Document Object Model), JavaScript memungkinkan pengembang untuk membuat halaman web lebih dinamis dan responsif terhadap interaksi pengguna.
2. Ya. Java dan JavaScript adalah dua bahasa pemrograman yang berbeda. Java adalah bahasa yang berorientasi objek yang berarti terdiri dari objek yang berinteraksi satu sama lain yang digunakan untuk pengembangan aplikasi desktop, perangkat lunak server, aplikasi web dan perangkat seluler. Di sisi lain, JavaScript adalah bahasa pemrograman yang digunakan untuk mengembangkan aplikasi web interaktif, yang berjalan di sisi klien dan biasanya digunakan untuk menambahkan interaktivitas, validasi formulir, manipulasi DOM, dan banyak lagi.
3. ECMAScript adalah standar scripting language yang digunakan untuk membuat dan mengatur perilaku situs web. JavaScript adalah implementasi dari standar ECMAScript yang paling umum digunakan di web. Perbedaannya yaitu ECMAScript adalah standar yang menetapkan aturan dan fitur yang harus dimiliki oleh bahasa pemrograman, sementara JavaScript adalah bahasa pemrograman yang mengkuhspentifikasi ECMAScript.
4. Ya. Untuk mendeklarasikan variable dalam JavaScript menggunakan keyword var, let atau const. Var dan let digunakan untuk mendefinisikan bahwa suatu variabel tersebut nilainya dapat berubah. Const digunakan untuk mendefinisikan bahwa suatu variabel tersebut nilainya tidak dapat berubah (konstan).

BAB IV

IMPLEMENTASI

4.1 Source Code

Anda adalah seorang pengembang perangkat lunak yang sedang membuat program untuk mengelola data mahasiswa. Berikut adalah beberapa tugas yang perlu Anda selesaikan:

1. Buatlah sebuah program untuk menyimpan data mahasiswa. Setiap mahasiswa memiliki nama, npm, jurusan dan nilai (berbentuk array). Data mahasiswa disimpan dalam bentuk objek.
2. Tampilkan data mahasiswa yang telah Anda simpan ke dalam objek ke dalam console. Pastikan untuk menampilkan setiap atribut mahasiswa (nama, npm, jurusan, nilai) ke console.
3. Buatlah sebuah fungsi untuk menghitung rata-rata dari nilai mahasiswa. Fungsi ini akan menerima sebuah array angka dari nilai mahasiswa dan mengembalikan nilai rata-ratanya.
4. Buatlah sebuah fungsi untuk mengubah urutan karakter dalam sebuah string menjadi terbalik. Misalnya, jika diberikan input "Hello", fungsi ini akan mengembalikan "olleH".
5. Buatlah sebuah perulangan untuk menampilkan angka dari nilai mahasiswa ke dalam console.
6. Buatlah sebuah fungsi untuk menentukan nilai huruf berdasarkan skala nilai berikut:
 - Jika nilai lebih besar dari 80, kembalikan "A".
 - Jika nilai antara 70 dan 79, kembalikan "B".
 - Jika nilai antara 60 dan 69, kembalikan "C".
 - Jika nilai antara 50 dan 59, kembalikan "D".
 - Jika nilai kurang dari 50, kembalikan "E".
7. Terakhir, tampilkan nilai huruf dari rata-rata nilai mahasiswa yang telah Anda simpan.

// nomor 1

```

var mahasiswa = {
    nama : "Lina",
    npm : "230441100178",
    jurusan : "Sistem Informasi",
    nilai : [85, 70, 60, 45, 95]
}

// nomor 2
console.log("=== Data Mahasiswa ===");
console.log(`Nama      : ${mahasiswa.nama}`);
console.log(`NPM       : ${mahasiswa.npm}`);
console.log(`Jurusan   : ${mahasiswa.jurusan}`);
console.log(`Nilai    : ${mahasiswa.nilai}`);
console.log("");

//nomor 3
function hitungRataNilai(nilai) {
    var total = 0;
    for (let i = 0; i < nilai.length; i++) {
        total += nilai[i];
    }
    return total / nilai.length;
}
const rataRata = hitungRataNilai(mahasiswa.nilai);

//nomor 4
function ubahUrutan(karakter) {
    var urutan = "";
    for (let i = karakter.length - 1; i >= 0; i--) {
        urutan += karakter[i];
    }
    return urutan;
}
const input = "Hello";
const kebalikan = ubahUrutan(input);
console.log(`Kebalikan dari '${input}' adalah '${kebalikan}'`);
console.log("");

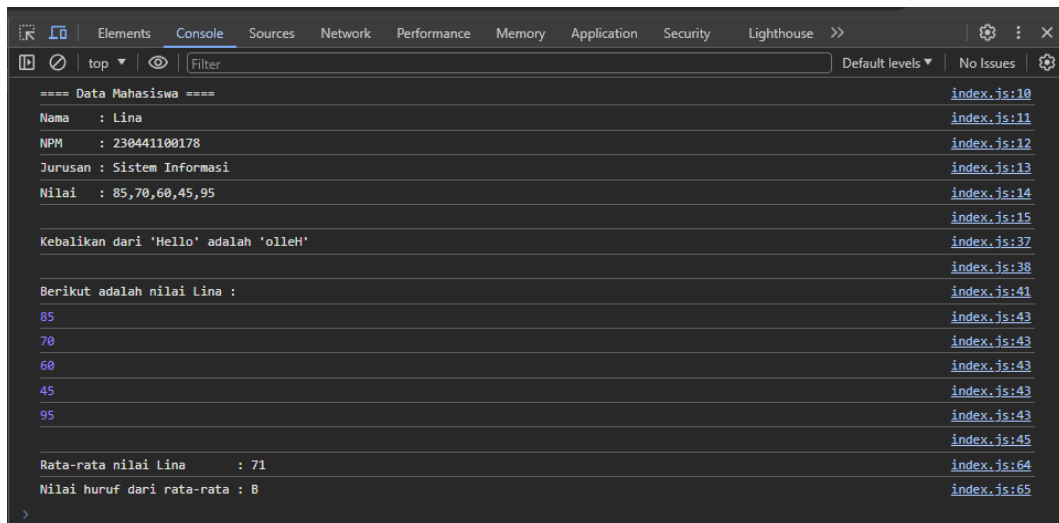
```

```
//nomor 5
console.log(`Berikut adalah nilai ${mahasiswa.nama} :`);
for (let i = 0; i < mahasiswa["nilai"].length; i++) {
    console.log(mahasiswa.nilai[i]);
}
console.log("");

//nomor 6
function nilaiHuruf(nilai) {
    if (nilai > 80) {
        return "A";
    } else if (nilai >= 70 && nilai <= 79) {
        return "B";
    } else if (nilai >= 60 && nilai <= 69) {
        return "C";
    } else if (nilai >= 50 && nilai <= 59) {
        return "D";
    } else {
        return "E";
    }
}

// nomor 7
const nilaiHrf = nilaiHuruf(rataRata);
console.log(`Rata-rata nilai ${mahasiswa.nama} : ${rataRata}`);
console.log(`Nilai huruf dari rata-rata : ${nilaiHrf}`);
```

4.2 Hasil



```
==== Data Mahasiswa ====
Nama : Lina
NPM : 230441100178
Jurusan : Sistem Informasi
Nilai : 85,70,60,45,95

Kebalikan dari 'Hello' adalah 'olleH'

Berikut adalah nilai Lina :
85
70
60
45
95

Rata-rata nilai Lina : 71
Nilai huruf dari rata-rata : B
```

4.3 Penjelasan

1. Untuk menyimpan data mahasiswa dalam bentuk objek. Pertama definisikan variabel dengan kata kunci `var` diikuti dengan nama variabel, lalu kita isi variabel tersebut dengan key dan value sesuai yang ditugaskan yang diletakkan di dalam kurung kurawal. Setiap key dan value dipisah dengan tanda `:`, untuk membuat lebih dari satu pasangan key-value dapat dipisah dengan tanda koma.
2. Untuk menampilkan setiap atribut ke console gunakan `console.log()` lalu di dalam kurung kita tuliskan yang ingin ditampilkan disini saya menggunakan template literal yang menggunakan tanda backtick (```) untuk pembuka dan penutupnya, lalu memasukkan nilai variabel ke dalam string dengan menggunakan `${}`.
3. Pertama buat fungsi `hitungRataNilai()` yang menerima satu parameter bernama `nilai`. Selanjutnya deklarasikan variabel `total` dan menginisialisasinya dengan nilai `0`. Variabel ini akan digunakan untuk menyimpan total nilai dari semua elemen dalam array. Lalu buat sebuah loop `for` yang akan melakukan iterasi melalui setiap elemen dalam array `nilai`. Loop ini akan menjumlahkan setiap nilai ke variabel `total`. Pada setiap iterasi loop, nilai pada indeks `i` dalam array `nilai` akan ditambahkan ke variabel `total` sehingga dapat menghitung jumlah total dari semua nilai dalam array. Setelah loop selesai, fungsi akan mengembalikan nilai rata-rata dari array `nilai`. Yang dilakukan dengan

membagi total jumlah nilai dengan panjang (jumlah elemen) dari array tersebut.

4. Pertama buat fungsi `ubahUrutan()` yang menerima satu parameter yaitu karakter, yang diasumsikan sebagai sebuah string. Lalu, deklarasikan variabel urutan sebagai string kosong. Variabel ini akan digunakan untuk menyimpan hasil dari proses mengubah urutan karakter. Kemudian, buat sebuah loop `for` yang akan melakukan iterasi melalui setiap karakter dalam string karakter, dimulai dari karakter terakhir hingga karakter pertama. Pada setiap iterasi loop, karakter pada indeks `i` dari string karakter akan ditambahkan ke variabel urutan. Setelah loop selesai, fungsi akan mengembalikan string urutan, yang merupakan kebalikan dari string input.
5. Pertama buat loop `for` yang akan melakukan iterasi melalui setiap nilai dalam array nilai milik objek mahasiswa. Pada setiap iterasi loop, nilai pada indeks `i` dari array nilai milik objek mahasiswa akan dicetak ke konsol yang akan mencetak setiap nilai satu per satu ke konsol.
6. Buat sebuah fungsi bernama `nilaiHuruf` yang menerima satu parameter yaitu nilai. Lalu buat kondisi `if` jika nilai yang diberikan lebih dari 80, maka fungsi akan mengembalikan string "A", menandakan nilai tersebut memiliki nilai huruf A. Lalu buat `else if` yang jika nilai berada di antara 70 dan 79, fungsi akan mengembalikan string "B", menandakan nilai tersebut memiliki nilai huruf B. Lalu buat `else if` jika nilai `>= 60 && nilai <= 69` nilai berada di antara 60 dan 69 akan mengembalikan string "C", menandakan nilai tersebut memiliki nilai huruf C. Lalu buat `else if` jika nilai `>= 50 && nilai <= 59` jika nilai berada di antara 50 dan 59 (inklusif), fungsi akan mengembalikan string "D", menandakan nilai tersebut memiliki nilai huruf D. Lalu buat `else` Jika nilai tidak memenuhi kondisi-kondisi di atas (yaitu kurang dari atau sama dengan 50), maka fungsi akan mengembalikan string "E", menandakan nilai tersebut memiliki nilai huruf E.
7. Untuk menampilkan ke konsol masukkan nilai dari variabel menggunakan template literal tadi.

BAB V

PENUTUP

5.1 Analisa

Dari hasil praktikum, praktikan menganalisa bahwa program membuat dropdown menu dari HTML, CSS, dan JavaScript dari materi yang telah didapatkan dapat berjalan dengan lancar. Kita dapat memanfaatkan JavaScript menjadi salah satu elemen kunci dalam ekosistem pengembangan web modern, bersama dengan HTML dan CSS, yang memungkinkan pembuatan aplikasi web yang kompleks dan interaktif.

Dari hasil praktikum, praktikan juga menganalisa bahwa kita dapat memahami bahwa dengan DOM, kita dapat membuat aplikasi web yang interaktif dan responsif, seperti perubahan konten halaman, validasi formulir, atau menanggapi interaksi pengguna.

5.2 Kesimpulan

1. Variabel dalam JavaScript dideklarasikan dengan menggunakan kata kunci `var`, `let`, atau `const`. JavaScript memiliki beberapa tipe data dasar seperti `string`, `number`, `boolean`, `null`, `undefined`, `array`, dan `object`.
2. Di JavaScript, ada beberapa jenis operator yaitu operator aritmatika, perbandingan, logika, dan penugasan. Ekspresi adalah kombinasi variabel, nilai, operator, dan/atau fungsi yang dievaluasi menjadi nilai tunggal.
3. Dalam pemrograman JavaScript, struktur kontrol yang digunakan yaitu seleksi kondisi dan pengulangan. Fungsi adalah blok kode yang dirancang untuk melakukan tugas tertentu yang dapat mengorganisir kode menjadi bagian-bagian yang lebih kecil dan dapat digunakan kembali.
4. DOM (Document Object Model) adalah representasi struktur dokumen HTML sebagai objek yang dapat dimanipulasi oleh bahasa pemrograman, seperti JavaScript.
5. Event listener adalah mekanisme dalam JavaScript yang digunakan untuk mendengarkan atau menanggapi aksi atau peristiwa yang terjadi dalam elemen HTML atau dokumen.