Universitas Pertamina

Author: Ahmad, Farel, Hafiz 30/10/2024

PENGENALAN JAVASCRIPT

PREPARED BY

UNIVERSITAS PERTAMINA

MADE FOR

PRAKTIKUM PEMROGRAMAN WEB

Universitas Pertamina

PENGENALAN JAVASCRIPT

1. Pendahuluan

Apa Itu JavaScript? JavaScript adalah bahasa pemrograman yang digunakan untuk menambah interaktivitas pada halaman web. Bersama dengan **HTML** (untuk struktur) dan **CSS** (untuk tampilan), JavaScript adalah salah satu teknologi inti yang membentuk fondasi dari pengembangan web modern. JavaScript memungkinkan halaman web berinteraksi dengan pengguna dalam waktu nyata. Sebagai contoh, JavaScript memungkinkan Anda memvalidasi formulir sebelum dikirim, memperbarui konten tanpa memuat ulang halaman, dan membuat animasi dinamis yang membuat situs web lebih hidup.

---

2. Sejarah JavaScript

- **1995**: JavaScript pertama kali dikembangkan oleh **Brendan Eich** di Netscape sebagai bahasa pemrograman sederhana untuk browser Netscape Navigator.
- **Awalnya bernama Mocha**, kemudian berubah menjadi LiveScript, sebelum akhirnya dinamakan **JavaScript**.
- **1997**: ECMA (European Computer Manufacturers Association) membuat spesifikasi standar untuk JavaScript yang disebut **ECMAScript** (ES). Versi pertama ini dikenal sebagai **ES1**.
- **2009**: JavaScript mengalami perkembangan besar dengan dirilisnya **ES5**.
- **2015**: Versi **ES6** atau **ECMAScript 2015** membawa banyak fitur baru, termasuk let, const, kelas (class), modul, dan arrow function.

Sejak itu, versi ECMAScript diperbarui hampir setiap tahun dengan penambahan fitur dan perbaikan.

---

3. Tujuan Penggunaan JavaScript

JavaScript memiliki beberapa tujuan utama, yaitu:

1. **Interaktivitas Web**: Membuat situs web lebih dinamis dan responsif terhadap tindakan pengguna, seperti klik tombol atau input formulir.

2. **Manipulasi DOM**: Mengubah elemen HTML secara dinamis seperti menambah, menghapus, atau mengedit konten halaman tanpa harus memuat ulang seluruh halaman.
3. **Komunikasi Asinkron**: Dengan teknologi seperti **AJAX**, JavaScript memungkinkan komunikasi asinkron antara browser dan server, memungkinkan pembaruan data tanpa refresh halaman penuh.
4. **Validasi Data**: JavaScript dapat digunakan untuk memvalidasi data yang dimasukkan oleh pengguna sebelum data tersebut dikirim ke server.
5. **Game dan Aplikasi**: Banyak game dan aplikasi web yang dibangun sepenuhnya dengan JavaScript atau dengan menggunakan framework dan pustaka JavaScript modern.

---

4. Variabel Dalam Javascript

Dalam JavaScript, variabel adalah tempat untuk menyimpan nilai. Ada tiga cara utama untuk mendeklarasikan variabel: var, let, dan const. Masing-masing memiliki perbedaan dalam hal scoping, reassignability (apakah nilainya bisa diubah), dan cara JavaScript menangani variabel tersebut.

a. Var (Deklarasi Lama)

- Scoping: Variabel yang dideklarasikan dengan var memiliki function scope. Artinya, variabel ini tersedia di dalam fungsi tempat ia dideklarasikan atau global jika dideklarasikan di luar fungsi.
- Redeclaration (Redeclare): Variabel yang dideklarasikan dengan var dapat dideklarasikan ulang dalam scope yang sama.
- Reassignability (Assign Ulang): Nilai dari variabel var bisa diubah setelah dideklarasikan.

```
var x = 10;
console.log(x);  // Output: 10
x = 20;          // Reassigning
console.log(x);  // Output: 20
```

b. Let (Diperkenalkan di ES6)

- Scoping: let memiliki block scope, artinya variabel ini hanya tersedia di dalam blok kode tempat ia dideklarasikan (misalnya, di dalam {}).
- Redeclaration: Variabel let tidak dapat dideklarasikan ulang dalam scope yang sama.
- Reassignability: Nilai dari variabel let bisa diubah setelah dideklarasikan.

```
let y = 10;
console.log(y);  // Output: 10
y = 30;          // Reassigning
console.log(y);  // Output: 30

// let y = 50;   // Error: Cannot redeclare 'y' in the same scope
```

c. Const (Konstanta tetap)

- Scoping: Seperti let, const juga memiliki block scope.
- Redeclaration: Variabel yang dideklarasikan dengan const tidak dapat dideklarasikan ulang dalam scope yang sama.
- Reassignability: Nilai dari variabel const tidak dapat diubah setelah dideklarasikan. Ini berarti const hanya digunakan untuk nilai tetap (konstanta).

```
const z = 100;
console.log(z);  // Output: 100

// z = 200;  // Error: Assignment to constant variable
```

Berikut Tabel Perbandingannya:

Fitur	***Var***	***Let***	***Const***
Scoping	Function-scoped	Block-scoped	Block-scoped
Redeclaration	Bisa dideklarasikan ulang	Tidak bisa dideklarasikan ulang	Tidak bisa dideklarasikan ulang
Reassignability	Nilai bisa diubah	Nilai bisa diubah	Nilai **tidak bisa diubah** (konstan)

5. Tipe Data JavaScript

JavaScript mendukung beberapa tipe data utama, yang dibagi menjadi dua kategori besar: Primitif dan Non-Primitif.

a. Tipe Data Primitif

1. String: Teks yang dikelilingi oleh tanda kutip tunggal (') atau ganda (").
   - Contoh: 'Halo, Dunia!', "JavaScript"
2. Number: Mencakup bilangan bulat dan bilangan desimal.
   - Contoh: 10, 3.14
3. Boolean: Hanya memiliki dua nilai: true atau false.
   - Contoh: true, false
4. Undefined: Variabel yang telah dideklarasikan tetapi belum diberikan nilai.

- Contoh: let a; (nilai a adalah undefined)

5. Null: Nilai yang secara eksplisit menyatakan bahwa variabel tidak memiliki nilai.
   - Contoh: let b = null;

b. Tipe Data Non-Primitif

1. Array: Kumpulan nilai yang dikelompokkan dalam satu variabel.
   - Contoh: let buah = ["apel", "pisang", "mangga"];
2. Object: Struktur data yang menyimpan data dalam bentuk pasangan key-value (kunci-nilai).
   - Contoh: let orang = { nama: "Budi", umur: 30 };

---

6. Operator Dalam JavaScript

JavaScript memiliki berbagai jenis operator untuk melakukan operasi aritmatika, logika, perbandingan, dan lainnya. Berikut adalah tabel operator yang penting untuk dipahami.

a. Operator Aritmatika

Operator ini digunakan untuk melakukan berbagai operasi matermatika dasar.

OPERATOR	NAMA	CONTOH	PENJELASAN
+	Penjumlahan	5 + 2 = 7	Menjumlahkan dua angka
-	Pengurangan	5 – 2 = 3	Mengurangi satu angka dengan lainnya
*	Perkalian	5 * 2 = 10	Mengalikan dua angka
/	Pembagian	5 / 2 = 2.5	Membagi satu angka dengan yang lainnya
%	Modulus	5 % 2 = 1	Sisa hasil dari pembagian
++	Increment	a++	Menambah nilai satu pada variabel a
--	Decrement/	a--	Mengurangi nilai satu padavariabel a

Berikut contoh penggunaan operator aritmatika:

```
let a = 10;
let b = 5;
console.log(a + b); // 15
console.log(a - b); // 5
console.log(a * b); // 50
console.log(a / b); // 2
console.log(a % b); // 0
```

b. Operator Perbandingan

Operator ini digunakan untuk membandingkan dua nilai.

OPERATOR	NAMA	CONTOH	PNEJELASAN
= =	Sama dengan	5 = = ‘5’	Mengembalikan true jika nilainya sama tanpa memeriksa tipe data
= = =	Identik (persis sama)	5 = = = ‘5’	Mengembalikan true jika nilainya dan tipe data sama
!=	Tidak sama dengan	5 != ‘5’	Mengembalikan true jika nilainya tidak sama
!==	Tidak identik	5 != = ‘5’	Mengembalikan true jika nilainya atau tipe data tidak sama
>	Lebih besar dari	5 > 3	Mengembalikan true jika nilainya lebih besar
<	Lebih kecil dari	5 < 3	Mengembalikan true jika nilainya lebih kecil
>=	Lebih besar atau sama	5 >= 5	Mengembalikan true jika nilainya lebih besar atau sama
<=	Lebih kecil atau sama	5 <= 5	Mengembalikan true jika nilainya lebih kecil atau sama

Berikut contoh penggunaan operator perbandingan:

```
let x = 5;
let y = '5';
console.log(x == y);  // true
console.log(x === y); // false
console.log(x != y);  // false
console.log(x !== y); // true
```

c. Operator Logika

Operator logika digunakan untuk menggabungkan beberapa ekspresi logika.

OPERATOR	NAMA	CONTOH	PENJELASAN
&&	AND	a && b	Mengembalikan true jika kedua ekspresi benar
||	OR	a || b	Atau (OR)
!	Not	!a	Mengembalikan true jika ekspresi bernilai false

Berikut contoh penggunaan operator logika:

```
let a = true;
let b = false;

console.log(a && b); // false
console.log(a || b); // true
console.log(!a);     // false
```

7. Struktur Kontrol (Control Flow)

JavaScript menyediakan beberapa jenis struktur kontrol yang membantu kita dalam mengatur alur eksekusi program, seperti pernyataan kondisional dan perulangan.

a. Pernyataan Kondisional (if, else if, else)

Pernyataan kondisional memungkinkan kita menjalankan kode berdasarkan kondisi tertentu.

```
let angka = 10;

if (angka > 0) {
    console.log("Positif");
} else if (angka < 0) {
    console.log("Negatif");
} else {
    console.log("Nol");
}
```

b. Perulangan (Looping)

JavaScript menyediakan beberapa jenis perulangan yang dapat digunakan untuk mengulangi blok kode berdasarkan kondisi tertentu.

- **for loop**: Digunakan untuk perulangan dengan kondisi awal, batas, dan perubahan.

```
for (let i = 0; i < 5; i++) {
        console.log(i);  // Output: 0, 1, 2, 3, 4
}
```

- **while loop**: Digunakan ketika kondisi harus diperiksa sebelum melakukan iterasi.

```
let i = 0;
while (i < 5) {
        console.log(i);  // Output: 0, 1, 2, 3, 4
        i++;
}
```

8. Fungsi Dalam JavaScript

Fungsi adalah blok kode yang bisa dipanggil kapan saja dan berfungsi untuk menjalankan tindakan tertentu.

a. Mendefinisikan Fungsi

Fungsi dalam JavaScript dapat didefinisikan dengan sintaks seperti ini. Fungsi dapat Anda panggil pada bagian baris kode mana saja selama kode fungsi ini terdapat pada kode:

```
function salam(nama) {
            return "Halo, " + nama + "!";
          }
console.log(salam("Budi"));  // Output: Halo, Budi!
```

b. Fungsi Ekspresi

Fungsi juga bisa dideklarasikan dalam bentuk ekspresi (anonymous function). dalam bentuk ini fungsi hanya bisa dipanggil ketika telah dideklarasikan di baris kode sebelumnya:

```
const kali = (a, b) => a * b;
console.log(kali(5, 4));  // Output: 20
```