



Pemrograman Sisi Cleint Javascript

Muhammad Syaifur Rohman, S.Kom, M.CS



Definisi

FYI



JavaScript dibuat oleh Brendan Eich pada tahun 1995 dan awalnya dikenal sebagai Mocha, kemudian namanya diubah menjadi LiveScript, dan akhirnya menjadi JavaScript. JavaScript dikembangkan untuk Netscape Navigator, sebuah browser web populer pada era 1990-an. Sejak saat itu, JavaScript telah berkembang menjadi salah satu bahasa pemrograman paling populer di dunia, digunakan oleh developer web di seluruh dunia untuk menambahkan interaktivitas dan dinamika ke halaman web.

Seiring waktu, spesifikasi JavaScript (ECMAScript) terus berkembang. ECMAScript 6 (ES6), dirilis pada tahun 2015, membawa banyak perubahan signifikan dan peningkatan pada bahasa tersebut, termasuk sintaks baru untuk variabel (let dan const), arrow functions, class, dan banyak lagi. Saat ini, JavaScript tidak hanya digunakan untuk pengembangan web di sisi klien, tetapi juga untuk pengembangan server dengan Node.js, aplikasi mobile, game, dan bahkan pengembangan hardware.



Sintax Dasar



```
var name = "John Doe"  
let age = 30;  
const birthday = "1 January 1990";
```

Variable

- Gunakan var jika Anda membutuhkan variabel dengan cakupan fungsi atau global yang dapat diubah dan dideklarasikan ulang.
- Gunakan let untuk variabel yang cakupannya terbatas pada blok kode dan nilainya mungkin berubah.
- Gunakan const untuk mendeklarasikan konstanta yang nilainya tidak boleh berubah setelah deklarasi awal.



Variable



```
var name = "John Doe";
var name = "Jane Doe"; // Valid, name dapat dideklarasikan ulang
```

Penggunaan Ulang: Variabel yang dideklarasikan dengan `var` dapat dideklarasikan ulang dalam cakupan yang sama.

Cakupan (Scope): `let` memiliki cakupan blok (block scope), yang berarti hanya dapat diakses dalam blok kode (misalnya, dalam kurung kurawal `{}`) tempat ia dideklarasikan.



```
let age = 30;
age = 31; // Valid, nilai age dapat diubah
// let age = 32; akan menghasilkan error karena age sudah dideklarasikan
```



```
const birthday = "1 January 1990";
// birthday = "2 February 1991"; akan menghasilkan error karena birthday adalah const
```

Penggunaan Ulang: Variabel yang dideklarasikan dengan `const` tidak dapat dideklarasikan ulang dalam cakupan yang sama, dan nilainya juga tidak dapat diubah. Ini digunakan untuk konstanta yang nilainya seharusnya tetap sama sepanjang kode.

Sintax Dasar

Tipe Data

- Primitif: String, Number, Boolean, null, undefined, Symbol.
- Non-primitif: Object.
- Contoh penggunaan tipe data.



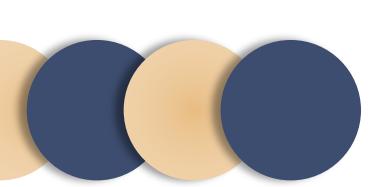
```
let name = 'Alice'; // String  
let age = 25; // Number  
let isActive = true; // Boolean
```

Operator

- Aritmatika: +, -, *, /, %.
- Perbandingan: ==, ===, !=, !==, <, >, <=, >=.
- Logika: &&, ||, !.
- Contoh penggunaan operator.



```
let sum = 10 + 5; // 15  
let isAdult = age >= 18; // true
```



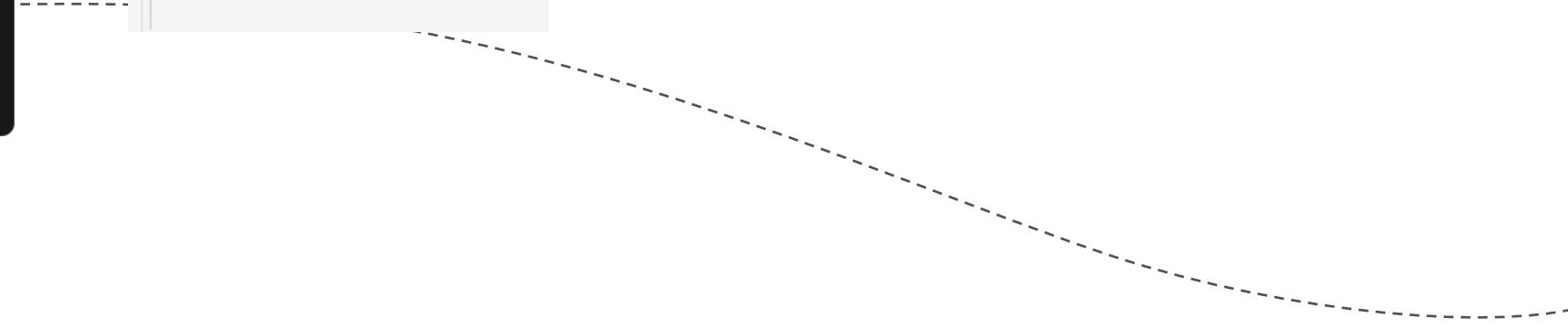
Function



```
function greet(name) {  
  return `Hello, ${name}!`;  
}  
  
console.log(greet('Alice'));
```

Output

```
node /tmp/BOHAQNynSu.js  
Hello, Alice!
```



Output

```
node /tmp/EGePQHgaDs.js  
Hello, Bob!
```

```
const greet = name => `Hello, ${name}!`;  
console.log(greet('Bob'));
```

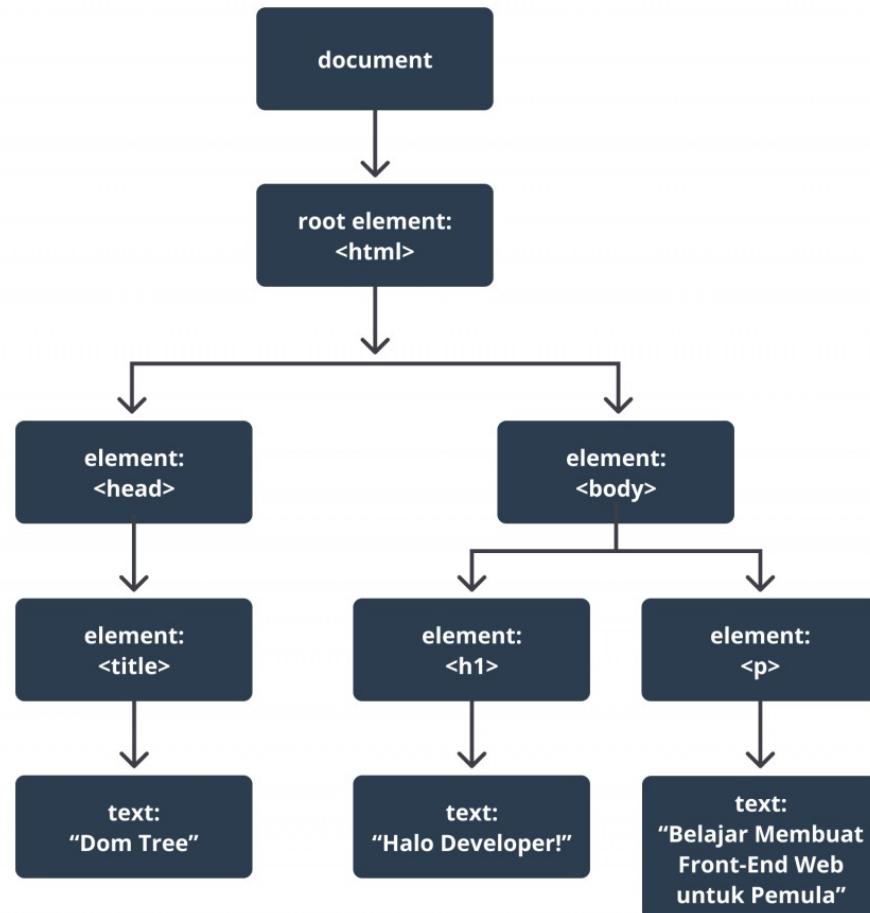
DOM Manipulation

Definisi

Dalam pengembangan web, manipulasi Document Object Model (DOM) adalah teknik yang penting untuk membuat halaman web interaktif. DOM merupakan representasi terstruktur dari dokumen HTML yang memungkinkan JavaScript untuk mengakses dan memodifikasi elemen serta atributnya, memberikan respons terhadap event pengguna, dan secara dinamis mengubah tampilan dan perilaku halaman web. Berikut ini adalah beberapa konsep utama dalam manipulasi DOM menggunakan JavaScript.

[Create Form](#) [Delete Form](#)

DOM Manipulation



Uppercase Lowercase Reset

Output:

GeeksforGeeks
Get HTML for DOM element

DOM Manipulation

How it works

Untuk memanipulasi elemen di dalam DOM, pertama-tama kita harus memilih elemen tersebut.

`document.getElementById(id)`

Mengembalikan elemen dengan ID yang spesifik.
ID dalam sebuah dokumen harus unik.



```
var element = document.getElementById("myElement");
```

`document.getElementsByTagName(name)`

Mengembalikan sebuah koleksi semua elemen dalam dokumen dengan nama tag yang spesifik.



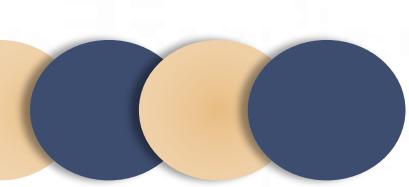
```
var elements = document.getElementsByTagName("div");
```

`document.getElementsByClassName(name)`

Mengembalikan semua elemen dengan kelas yang spesifik.



```
var elements = document.getElementsByClassName("myClass");
```



DOM Manipulation



```
// Properti innerHTML memungkinkan kita untuk mendapatkan atau  
// mengatur kode HTML yang ada di dalam elemen.  
  
// Mengatur konten HTML  
element.innerHTML = "<strong>Hello World!</strong>";  
  
// Mendapatkan konten HTML  
var content = element.innerHTML;
```



```
// Properti textContent menyediakan cara untuk  
// mendapatkan atau mengatur konten teks sebuah elemen.  
  
// Mengatur teks elemen  
element.textContent = "Hello World!";  
  
// Mendapatkan teks elemen  
var text = element.textContent;
```



```
/*
```

Untuk membuat halaman web interaktif, kita perlu merespons terhadap event yang dilakukan oleh pengguna, seperti klik mouse atau pengetikan keyboard. Kita dapat menambahkan listener event ke elemen menggunakan addEventListener.

```
*/
```

```
element.addEventListener("click", function() {  
    alert("Element was clicked!");  
});
```

DOM Manipulation

```
<!DOCTYPE html>
<html>
<head>
    <title>DOM Manipulation Example</title>
</head>
<body>
    <div id="myElement" class="content">Hello!</div>
    <button id="changeContent">Change Content</button>

    <script>
        // Memilih elemen
        var element = document.getElementById("myElement");
        var button = document.getElementById("changeContent");

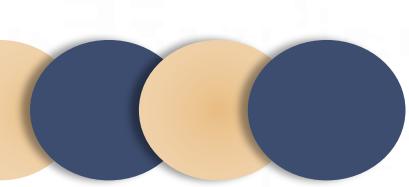
        // Menambahkan event listener ke button
        button.addEventListener("click", function() {
            // Mengubah konten elemen
            element.innerHTML = "<strong>Content Changed!</strong>";
        });
    </script>
</body>
</html>
```

TRY...



ES6 (ECMAScript 2015)





Definisi

FYI

- ES6 adalah versi dari JavaScript, bukan bahasa yang berbeda. Ketika orang berbicara tentang "JavaScript modern", mereka sering kali merujuk pada penggunaan fitur-fitur yang diperkenalkan di ES6 dan versi ECMAScript berikutnya.
- Penerapan ES6 oleh browser modern sangat baik, tetapi untuk memastikan kompatibilitas dengan semua pengguna, kode ES6 seringkali ditranspilasi kembali ke versi JavaScript yang lebih lama menggunakan alat seperti Babel. Ini memungkinkan pengembang untuk manfaatkan fitur-fitur modern bahasa sambil tetap mendukung pengguna dengan browser yang lebih lama.
- Secara keseluruhan, ES6 merupakan tonggak penting dalam evolusi JavaScript, membawa banyak fitur dan peningkatan yang telah lama ditunggu-tunggu oleh komunitas pengembang.

Fitur Khusus

JavaScript ES6, juga dikenal sebagai ECMAScript 2015, membawa banyak perubahan dan peningkatan yang membuat penulisan kode JavaScript menjadi lebih efisien dan mudah dipahami.



```
const y = 30;  
// y = 40; // Akan menyebabkan error
```



```
let x = 10;  
if (x === 10) {  
  let x = 20; // Variabel ini berbeda dengan variabel x di luar blok  
  console.log(x); // 20  
}  
console.log(x); // 10
```



```
let nama = "Dunia";  
console.log(`Halo, ${nama}!`); // "Halo, Dunia!"
```



```
// Arrow Functions: Sintaks Pendek untuk Fungsi  
const arr = [1, 2, 3, 4];  
const doubled = arr.map((item) => item * 2);  
console.log(doubled); // [2, 4, 6, 8]
```



```
// Destructuring Assignment: Cara Baru untuk Mengekstrak Data  
const obj = {a: 1, b: 2, c: 3};  
const {a, c} = obj;  
console.log(a); // 1  
console.log(c); // 3
```

How to Use

Penggunaan ES6 (ECMAScript 2015) dibandingkan dengan versi JavaScript yang lebih lama (sering disebut sebagai "JavaScript biasa") tidak memerlukan cara khusus dalam hal sintaks dasar bahasa; itu lebih tentang memanfaatkan fitur-fitur baru yang diperkenalkan oleh ES6 untuk membuat kode lebih efisien, mudah dibaca, dan modern.

Default Parameters



```
// Mengumpulkan semua argumen yang tersisa dari sebuah
// fungsi ke dalam sebuah array.
function sum(...numbers) {
    return numbers.reduce((prev, curr) => prev + curr, 0);
}
```

Spread Parameter

```
// Memungkinkan elemen dari sebuah objek iterabel (seperti array)
// untuk diperluas menjadi elemen individual.
let arr1 = [1, 2, 3];
let arr2 = [4, 5, 6];
let combined = [...arr1, ...arr2]; // [1, 2, 3, 4, 5, 6]
```

Export dan Import Module

```
// Export: Menandai variabel atau fungsi dalam
// modul agar dapat digunakan oleh modul lain.
```

```
// file math.js
export const add = (a, b) => a + b;
```

```
// Import: Mengakses variabel atau fungsi dari modul lain.
```

```
// file app.js
import { add } from './math.js';
console.log(add(2, 3)); // Output: 5
```



Rest Parameters

```
// Memungkinkan fungsi untuk memiliki parameter
// default jika tidak ada argumen yang diberikan.
function greet(name = "Stranger") {
    return `Hello, ${name}!`;
}
```

Full of fun

HTML

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>ES6 Demo</title>
</head>
<body>
  <h1>ES6 Features Demo</h1>
  <div id="greeting"></div>
  <ul id="userList"></ul>
  <button id="addUser">Add New User</button>

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

TRY...



```
// Default Parameters & Template Literals
const greet = (name = 'Visitor') => `Hello, ${name}! Welcome to the ES6 Demo.`;
document.getElementById('greeting').innerHTML = greet();
```

```
// Promise for fetching data (simulating)
```

```
const fetchUsers = () => {
  return new Promise(resolve => {
    setTimeout(() => {
      resolve([
        { name: "John Doe", age: 30 },
        { name: "Jane Doe", age: 25 }
      ]);
    }, 1000);
};
```

```
// Async/Await for using the Promise
```

```
const displayUsers = async () => {
  const users = await fetchUsers();
  const userListElement = document.getElementById('userList');
```

```
// Destructuring & Template Literals
```

```
users.forEach(user => {
  const { name, age } = user;
  userListElement.innerHTML += `<li>${name}, Age: ${age}</li>`;
});
```

```
// Arrow Functions & Spread Operator
```

```
document.getElementById('addUser').addEventListener('click', async () => {
  const newUser = { name: "New User", age: 22 };
  const users = await fetchUsers();
  const allUsers = [...users, newUser]; // Using spread to add new user

  document.getElementById('userList').innerHTML = ''; // Clear current list
  allUsers.forEach(user => {
    const { name, age } = user; // Destructuring
    document.getElementById('userList').innerHTML += `<li>${name}, Age: ${age}</li>`;
  });
}

// Initial display
displayUsers();
```

TRY...



Thanks

Muhammad Syaifur Rohman, S.Kom, M.CS

