

## 1. Pengantar dan Dasar JavaScript

JavaScript adalah bahasa pemrograman yang dibuat oleh Brendan Eich pada tahun 1995 di perusahaan Netscape, bertujuan untuk membuat situs web lebih interaktif. Meski namanya mirip, JavaScript berbeda dengan Java. Bahasa ini kini bisa dijalankan juga di sisi server dengan Node.js. Sebelum mempelajarinya, pemahaman dasar HTML dan CSS sangat disarankan. Pengembangan JavaScript bisa dilakukan di browser seperti Chrome dan editor teks seperti VS Code. Kode dapat disisipkan langsung dalam HTML atau ditulis di file .js terpisah. Meskipun tanda titik koma bersifat opsional, penggunaannya secara konsisten dianjurkan. JavaScript memiliki berbagai tipe data seperti angka, teks, boolean, array, objek, undefined, null, dan NaN.

---

## 2. Variabel, Operator, dan Struktur Kontrol

Dalam JavaScript, variabel dideklarasikan dengan `var`, `let`, atau `const`. `let` digunakan untuk nilai yang bisa berubah, sementara `const` untuk nilai tetap. JavaScript menerapkan **dynamic typing**, sehingga tipe data variabel bisa berubah. Operator yang tersedia antara lain operator aritmatika, perbandingan, dan logika. Untuk keperluan debugging, digunakan `console.log()`. Template literal memanfaatkan tanda backtick ( ``` ) dan `${}` untuk menyisipkan variabel. Konversi tipe data bisa dilakukan dengan `parseInt()`, `Number()`, `toString()`, dan sebagainya. Struktur kontrol meliputi `if-else`, `switch`, operator ternary, `??` (nullish coalescing), dan `?.` (optional chaining). Konsep **`**truthy**`** dan **`**falsy**`** membedakan nilai yang dianggap benar atau salah dalam logika. Untuk perulangan, digunakan `for`, `while`, `do-while`, `for...in`, dan `for...of`, serta pengendali `break` dan `continue`.

---

## 3. Fungsi dan Fitur Lanjutan

Fungsi dideklarasikan dengan `function` dan bisa memiliki parameter serta nilai balik. JavaScript mendukung parameter opsional, default, rest (`...args`), dan spread syntax. Fungsi bisa menjadi nilai, anonim (tanpa nama), bersarang (inner), atau rekursif. Arrow function (`=>`) merupakan bentuk singkat namun tidak memiliki `this` dan `arguments` sendiri. Fungsi generator menggunakan `yield` untuk menghasilkan nilai secara bertahap. Fungsi juga bisa menjadi method dalam objek, dengan `this` yang mengacu pada pemilik objek, kecuali pada arrow function. Getter dan setter memungkinkan pengaturan properti dengan logika khusus. Destructuring digunakan untuk memecah array atau objek ke dalam variabel, dan mendukung nilai default serta penggantian nama.

---

## 4. Mode dan Tools Pengembangan

Penggunaan `"use strict"` mengaktifkan mode ketat agar kode lebih aman dan terstruktur. `Statement with` memudahkan akses properti objek, namun tidak disarankan karena membingungkan. Debugger digunakan untuk menghentikan program dan memeriksa data. Setelah menguasai dasar, materi lanjutan meliputi OOP di JavaScript, modul (module system), manipulasi DOM, pemrograman asinkron, dan Web API. Penguasaan dasar ini akan menjadi pondasi kuat untuk membuat aplikasi JavaScript yang efisien dan kompleks.

---

## Rangkuman JavaScript OOP

### 1. Dasar OOP dan Class di JavaScript

Object Oriented Programming (OOP) adalah paradigma yang memusatkan pada objek yang memiliki properti dan metode. Class berperan sebagai cetakan untuk membuat objek. JavaScript awalnya bersifat prosedural, namun sejak ES6 diperkenalkan struktur class yang lebih modern. Sebelum ES6, objek dibuat dengan constructor function dan keyword new. Properti objek didefinisikan menggunakan this, dan pewarisan dilakukan dengan memanggil constructor lain menggunakan .call(this, ...). JavaScript menggunakan pewarisan berbasis prototype. Mulai ES6, class mendukung constructor, extends, super, getter/setter, dan pengecekan instanceof.

## **2. Fitur Lanjutan dan Penanganan Error**

Fitur terbaru JavaScript mencakup properti public dan private dalam class, di mana private ditandai dengan simbol # dan hanya dapat diakses dari dalam class. Keyword static digunakan untuk mendeklarasikan properti atau metode yang hanya dimiliki class, bukan objek instance-nya. Penanganan error menggunakan class Error, dan dapat dibuat custom error class dengan throw, try-catch-finally. JavaScript juga mengenal konsep Iterable dan Iterator, yang membuat objek bisa di-loop dengan for...of. Setelah memahami OOP, materi lanjutan mencakup pustaka standar, modul, manipulasi DOM, async programming, dan integrasi Web API.

---

### **JavaScript Standard Library**

Dokumentasi ini mencakup fitur-fitur penting dari pustaka standar JavaScript untuk pengembangan web. Materi meliputi manipulasi data Number, String, dan Array (termasuk implementasi stack dan queue), serta operasi seperti filter dan transformasi data. Juga dibahas pengelolaan objek melalui method seperti Object.freeze(), Object.seal(), dan Object.assign(). Untuk pertukaran data, JavaScript menyediakan dukungan JSON.

Bagian lanjutan mencakup BigInt untuk angka besar, Date untuk waktu, serta Math untuk perhitungan matematis. Struktur data seperti Map, Set, Symbol, dan RegExp juga dibahas. Fitur canggih seperti Proxy dan Reflect memungkinkan kontrol mendalam atas perilaku objek. Materi diakhiri dengan topik praktis seperti encoding URL, konversi Base64, dan eval untuk mengeksekusi kode dinamis, dengan peringatan soal risiko keamanannya.

---

## **Rangkuman PHP Native dan MySQL**

### **1. Instalasi dan Persiapan Lingkungan**

Tujuan praktikum ini adalah membuat aplikasi CRUD sederhana menggunakan PHP Native dan MySQL yang terhubung ke frontend HTML. Langkah awal meliputi instalasi Laragon sebagai server lokal dan konfigurasi PHP 8.3.17. Setelah itu, dilakukan penyesuaian environment Windows agar Laragon bisa diakses dari command line, serta pembuatan folder proyek di direktori www.

### **2. Backend dan Koneksi ke Database**

Bagian backend dimulai dengan file database.php untuk mengatur koneksi database via PDO. Struktur folder dibagi ke dalam model User.php yang memuat semua method CRUD seperti create(), read(), update(), dan delete(). Tabel users dibuat melalui phpMyAdmin atau HeidiSQL dengan field seperti id, nama, email, created\_at, dan updated\_at.

### **3. Integrasi Frontend dan Backend**

Integrasi dilakukan melalui file PHP yang menggabungkan kode HTML dan PHP. File index.php menampilkan daftar user, edit.php digunakan untuk mengedit data berdasarkan parameter GET, dan proses update dilakukan melalui method POST. Penghapusan data dilakukan melalui file khusus yang menerima parameter id, memanggil method delete(), lalu mengarahkan kembali ke halaman utama.