

## Rangkuman JavaScript Dasar

### 1. Pengenalan dan Dasar-Dasar JavaScript

JavaScript dikembangkan oleh Brendan Eich pada tahun 1995 di Netscape dengan tujuan membuat halaman web lebih interaktif. Bahasa ini berbeda dengan Java meskipun namanya mirip, dan kini dapat dijalankan di server melalui Node.js. Sebelum mempelajari JavaScript, disarankan sudah memahami HTML dan CSS. Untuk pengembangan, diperlukan browser seperti Chrome atau Firefox dan text editor seperti VS Code. Kode JavaScript dapat ditulis langsung dalam HTML atau file .js terpisah, dengan penggunaan titik koma yang opsional namun disarankan konsisten. JavaScript memiliki berbagai tipe data seperti Number (bilangan), Boolean (true/false), String (teks), Array (daftar data), Object (pasangan key-value), Undefined (variabel belum diisi), Null (nilai kosong sengaja), dan NaN (bukan angka).

### 2. Variabel, Operator, dan Kontrol Program

Deklarasi variabel menggunakan var (tidak direkomendasikan), let (dapat diubah), atau const (tidak dapat diubah), dengan JavaScript mendukung dynamic typing dimana tipe data dapat berubah. Operator yang tersedia meliputi aritmatika (+, -, \*, /, %), perbandingan (==, ===, !=), dan logika (&&, ||, !). Console digunakan untuk debugging dengan console.log() dan method lainnya. String template menggunakan backtick dan \${} untuk interpolasi. Konversi tipe data dapat dilakukan dengan parseInt(), parseFloat(), Number(), dan toString(). Percabangan menggunakan if-else, switch statement, ternary operator, nullish coalescing (??), dan optional chaining (?.). JavaScript mengenal konsep falsy (false, 0, "", null, undefined, NaN) dan truthy (selain falsy). Perulangan dapat dilakukan dengan for, while, do-while, for...in, dan for...of, dengan kontrol menggunakan break dan continue.

### 3. Function dan Fitur Lanjutan

Function dideklarasikan dengan kata kunci function dan dapat menerima parameter serta mengembalikan nilai. JavaScript mendukung optional parameter, default parameter, rest parameter (...args), dan spread syntax. Function dapat diperlakukan sebagai value, dibuat tanpa nama (anonymous), didefinisikan dalam function lain (inner), atau memanggil dirinya sendiri (recursive). Arrow function menggunakan sintaks => yang lebih ringkas namun memiliki keterbatasan seperti tidak memiliki this dan arguments. Function generator menggunakan yield untuk iterasi lazy. Object method adalah function sebagai property object, dengan keyword this yang merujuk pemilik function namun nilainya bergantung konteks. Arrow function tidak memiliki this sendiri. Getter dan setter memungkinkan kontrol property dengan function untuk validasi. Destructuring memecah data array atau object ke variabel terpisah dengan dukungan nested, default value, dan penggantian nama.

### 4. Mode dan Tools Pengembangan

Strict mode diaktifkan dengan "use strict" untuk aturan lebih ketat dan mencegah error tersembunyi. With statement memungkinkan akses property object tanpa menyebut nama object secara eksplisit namun tidak direkomendasikan karena membingungkan. Debugger adalah alat untuk menghentikan eksekusi dan memeriksa data saat debugging. Setelah menguasai dasar JavaScript, materi lanjutan yang dapat dipelajari meliputi JavaScript OOP untuk pemrograman berorientasi objek, JavaScript Modules untuk organisasi kode, Document Object Model (DOM) untuk manipulasi halaman web, JavaScript Async untuk pemrograman asinkron, dan Web API untuk integrasi dengan layanan web. Pemahaman mendalam terhadap

konsep-konsep dasar ini akan menjadi fondasi kuat untuk mengembangkan aplikasi JavaScript yang kompleks dan efisien.

## **Rangkuman JavaScript Object Oriented Programming**

### **1. Konsep Dasar OOP dan Class dalam JavaScript**

Object Oriented Programming adalah paradigma pemrograman yang berfokus pada konsep "objek" yang berisi properties dan methods. Object adalah entitas yang memiliki field/properties/attributes dan method/function/behavior, sedangkan Class adalah blueprint atau cetakan untuk membuat object. JavaScript awalnya adalah bahasa prosedural, namun seiring perkembangan ECMAScript, implementasi OOP semakin diperbaiki. Sebelum ES6, pembuatan class menggunakan Constructor Function dengan kata kunci new untuk membuat object instance. Constructor Function dapat memiliki property dengan keyword this, method, parameter, dan bahkan inheritance dengan memanggil constructor lain menggunakan NamaConstructor.call(this, parameter). JavaScript menggunakan konsep prototype inheritance dimana setiap constructor function memiliki prototype, dan object instance mewarisi dari prototype tersebut. Sejak ES6, diperkenalkan kata kunci class yang lebih mudah digunakan daripada constructor function. Class mendukung constructor, property, method, inheritance dengan extends, super constructor untuk memanggil parent constructor, super method untuk mengakses parent method, getter dan setter, serta operator instanceof untuk mengecek apakah object adalah instance dari class tertentu.

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

JavaScript modern mendukung public dan private class field dengan proposal yang masih dalam tahap pengembangan. Public class field dapat dideklarasikan di luar constructor selevel dengan method, sedangkan private class field menggunakan tanda # dan hanya dapat diakses dari dalam class. Static class field dan method menggunakan kata kunci static dan menjadi milik class itu sendiri, bukan instance object, sehingga dapat diakses langsung melalui nama class. Error handling merupakan bagian penting dalam OOP JavaScript dengan class Error sebagai superclass untuk semua jenis error. Error dapat dilempar menggunakan kata kunci throw dan ditangkap menggunakan try-catch-finally statement. Developer dapat membuat custom error class dengan mewarisi dari class Error. Iterable dan Iterator adalah fitur ES6 yang memungkinkan object untuk diiterasi menggunakan for...of loop dengan mengimplementasikan kontrak Iterable dan Iterator. Setelah memahami JavaScript OOP, materi selanjutnya yang dapat dipelajari meliputi JavaScript Standard Library, Modules, DOM, Async Programming, dan Web API untuk pengembangan aplikasi yang lebih kompleks dan modern.

## **JavaScript Standard Library**

Dokumen JavaScript Standard Library ini merupakan panduan komprehensif untuk mempelajari berbagai fungsi dan metode bawaan JavaScript yang penting untuk pengembangan web. Materi mencakup manipulasi tipe data dasar seperti Number (dengan static properties dan methods untuk konversi dan validasi), String (dengan berbagai instance methods untuk manipulasi teks), dan Array (termasuk konsep struktur data Queue dan Stack, serta methods untuk pencarian, filtering, dan transformasi data). Selain itu,

dokumen juga membahas pengelolaan Object dengan methods seperti freeze(), seal(), dan assign(), serta penggunaan JSON untuk pertukaran data antara server dan client.

Bagian lanjutan membahas tipe data dan fitur JavaScript yang lebih advanced, termasuk BigInt untuk angka besar, Date untuk manipulasi waktu dan tanggal dengan format epoch timestamp, serta Math untuk operasi matematika. Dokumen juga menjelaskan struktur data modern seperti Map dan Set, penggunaan Symbol untuk identifier unik, RegExp untuk pattern matching, dan fitur advanced seperti Proxy untuk intercepting operations dan Reflect untuk meta-programming. Materi diakhiri dengan topik praktis seperti encoding/decoding URL, Base64 untuk binary-to-text conversion, dan Eval untuk eksekusi kode dinamis, dengan catatan penting tentang keamanan penggunaannya.

## **Rangkuman Materi PHP Native dan MySQL**

### **Persiapan dan Instalasi Lingkungan Pengembangan**

Materi praktikum ini bertujuan untuk membuat aplikasi CRUD (Create, Read, Update, Delete) sederhana menggunakan PHP Native dan MySQL dengan integrasi frontend HTML. Proses instalasi dimulai dengan mengunduh dan menginstal Laragon 6.0.0 sebagai server lokal, kemudian menginstal PHP 8.3.17 versi Thread Safe ke dalam direktori Laragon. Langkah penting lainnya meliputi konfigurasi environment variables Windows agar Laragon dapat diakses dari command line, serta penyiapan direktori project di folder www Laragon untuk memudahkan pengembangan aplikasi web.

### **Struktur Backend dan Koneksi Database**

Pengembangan backend dimulai dengan membuat file konfigurasi database.php untuk menangani koneksi ke MySQL menggunakan PDO (PHP Data Objects). Struktur project diorganisir dengan folder models yang berisi file User.php sebagai class model untuk menangani semua operasi CRUD terhadap tabel users. Class User ini dilengkapi dengan method-method seperti create(), read(), update(), dan delete() yang memanfaatkan prepared statements untuk keamanan database. Database schema dibuat melalui phpMyAdmin atau HeidiSQL dengan struktur tabel users yang mencakup field id, nama, email, dan timestamp untuk created\_at dan updated\_at.

### **Integrasi Frontend dan Backend**

Integrasi antara frontend dan backend dilakukan melalui file-file PHP yang menggabungkan HTML dengan kode PHP untuk menampilkan dan memanipulasi data. File index.php berfungsi sebagai halaman utama yang menampilkan daftar users dalam format tabel HTML dengan data yang diambil dari database melalui method read() pada class User. Halaman edit.php dibuat untuk menangani form update data user dengan mengambil data existing melalui parameter GET dan memproses update melalui method POST. Sedangkan operasi delete ditangani melalui file terpisah yang menerima parameter ID user dan memanggil method delete() dari class User, kemudian melakukan redirect kembali ke halaman index setelah operasi berhasil.