**LAPORAN RESMI**

**MODUL IV**

**JAVASCRIPT**

**PEMOGRAMAN BERBASIS WEB**



**NAMA : Dylan Akhtarreza**

**N.R.P : 230441100027**

**DOSEN : Firli Irhamni, St., M.kom.**

**ASISTEN : Muhammad Irham Alkarim**

**TGL PRAKTIKUM : 14 Mei 2024**

**Disetujui : XX Mei 2024**

**Asisten**

**Muhammad Irham Alkarim**

**20.04.411.00111**

**LABORATORIUM BISNIS INTELIJEN SISTEM**

**PRODI SISTEM INFORMASI**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

# **BAB I PENDAHULUAN**

## **Latar Belakang**

pentingnya pemahaman dasar-dasar pemrograman JavaScript dalam pengembangan web dalam era digital saat ini, website dan aplikasi web menjadi salah satu sarana utama untuk berinteraksi dengan pengguna. Oleh karena itu, pemahaman yang kuat tentang JavaScript, termasuk Event dan DOM Manipulation, sangat diperlukan untuk menciptakan pengalaman pengguna yang interaktif, responsif, dan dinamis. JavaScript, sebagai bahasa pemrograman yang digunakan secara luas dalam pengembangan web, memungkinkan pengembang untuk membuat website yang lebih menarik dan fungsional. Dengan menggunakan Event Listener, pengembang dapat merespons aksi pengguna seperti klik mouse atau input pengguna dengan menjalankan fungsi tertentu. Hal ini memungkinkan pembuatan aplikasi web yang lebih interaktif dan dinamis. Selain itu, pemahaman tentang DOM Manipulation memungkinkan pengembang untuk mengakses dan memanipulasi elemen HTML dalam halaman web. Dengan kemampuan ini, pengembang dapat mengubah konten HTML, menambah atau menghapus elemen, serta mengubah atribut elemen sesuai kebutuhan aplikasi. Dengan demikian, pemahaman yang mendalam tentang JavaScript, termasuk Event dan DOM Manipulation, akan memberikan landasan yang kuat dalam pengembangan aplikasi web yang efektif, responsif, dan menarik bagi pengguna. Melalui penerapan konsep-konsep ini, pengembang dapat menciptakan pengalaman pengguna yang lebih baik dan meningkatkan kualitas aplikasi web yang dibuat. Dengan demikian, Modul IV JavaScript memberikan dasar yang penting bagi mahasiswa untuk memahami konsep-konsep dasar pemrograman JavaScript dan menerapkannya dalam pengembangan aplikasi web yang inovatif dan berkualitas.

## **Tujuan**

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

# **BAB II DASAR TEORI**

## **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.



## **Variable Dan Tipe Data**

**B.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;

**B.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** | [“surotong”, 1, false, ‘surot’] |
| **object** | {   namaLengkap: “otong surotong”,  nama\_panggilan: ‘surot’,   isAktif: true |

|  |  |
| --- | --- |
|  | } |

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

## **2.3** **Operator dan Ekspresi**

**C.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 ).

**C.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.

## **2.4 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:

**D.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;

}

**Tantangan** :

• Buatlah 2 buah seleksi kondisi. Seleksi kondisi sederhana dan seleksi kondisi  bersarang !

**D.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**);

**Tantangan** :

• Praktikan 3 macam perulangan, dan coba pahami cara kerjanya!

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

**E. 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**

**(TULIS DIKERTAS, DISCAN DAN MASUKKAN KE BAGIAN INI)**

# **BAB IV IMPLEMENTASI**

## **Soal**

1. Buatlah sebuah validasi form register dengan ketentuan seperti berikut:

1. Form input first name dan last name tidak boleh kurang dari 3 huruf dan tidak boleh lebih dari 10 huruf, jika validasi yang di inputkan user kurang sesuai maka tampilkan pesan validasi “First Name tidak boleh kurang dari 3 dan melebihi 10 character”, “Last Name tidak boleh kurang dari 3 dan melebihi 10 character”.

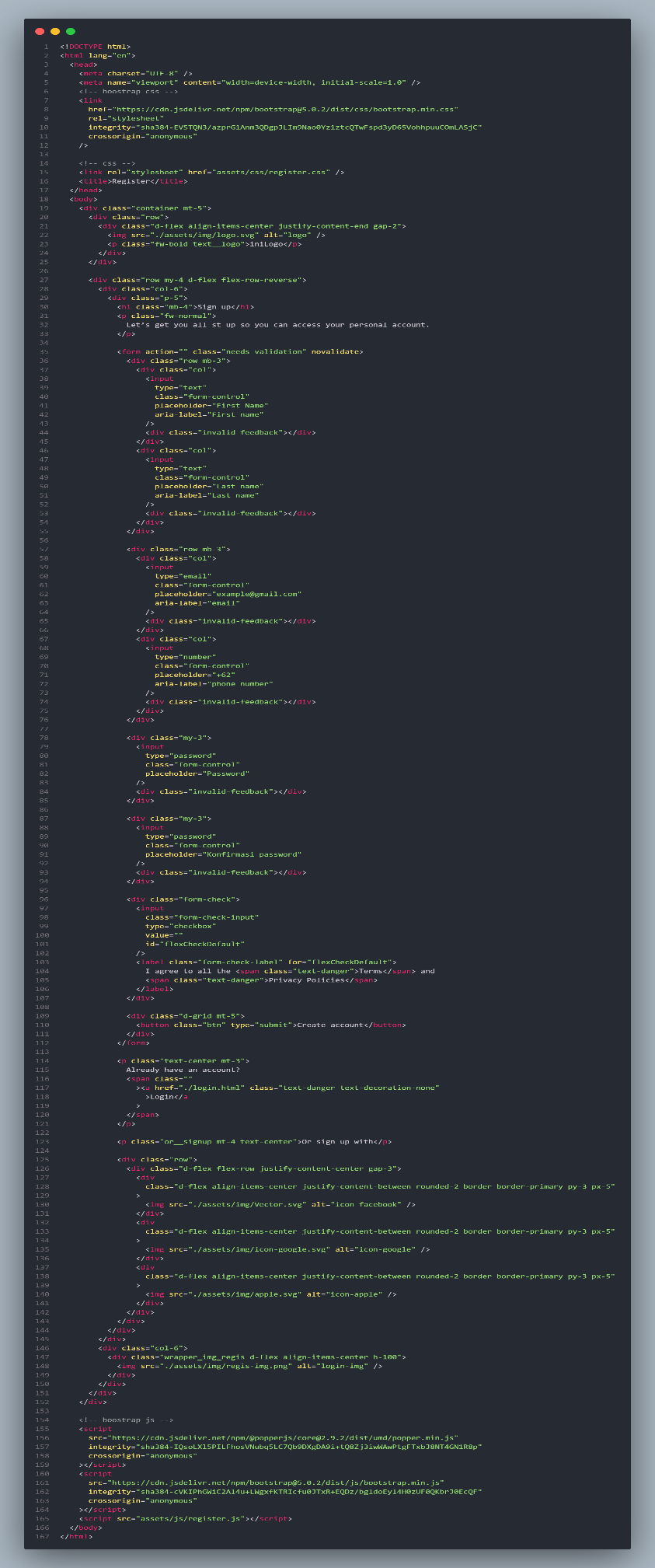
2. Form email tidak boleh kosong, format form email harus valid, jika validasi yang di inputkan user kurang sesuai maka tampilkan pesan validasi “email tidak boleh kosong”, “maaf email tidak valid”.

3. Form nomor telepon tidak boleh kosong, form nomor telepon tidak boleh melebihi 12, jika validasi yang di inputkan user kurang sesuai maka tampilkan pesan validasi “no telepon tidak boleh kosong”, “no telepon tidak boleh lebih dari 12”,.

4. Form password password harus min 8 character, huruf besar, huruf kecil, angka, dan karakter spesial, serta form password tidak boleh kosong. jika validasi yang di inputkan user kurang sesuai maka tampilkan pesan validasi “password tidak boleh kosong”, “password harus min 8 character, huruf besar, huruf kecil, angka, dan karakter spesial”.

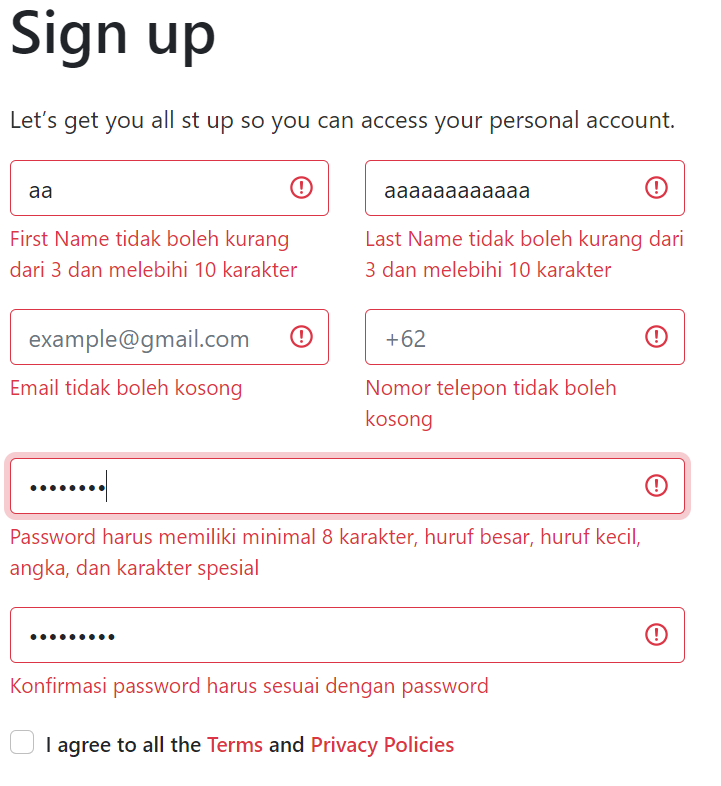
5. Form konfrimasi password harus sama atau sesuai dengan password, jika tidak maka akan muncul pesan validasi “Konfirmasi password harus sesuai dengan password”.

## **Source Code**





## **Hasil**



Membuat sebuah validasi jika melakukan sebuah kondisi tertenu pada register yang akan ditampilkan pada pengguna.

# **BAB V PENUTUP**

## **Analisa**

Dalam pemrograman JavaScript, pemahaman tentang konsep dasar seperti variabel, tipe data, struktur kontrol (seleksi kondisi, pengulangan), DOM manipulation, event listener, dan fungsi sangat penting. Variabel digunakan untuk menyimpan data, tipe data menentukan jenis nilai yang dapat disimpan, struktur kontrol digunakan untuk mengontrol alur eksekusi program berdasarkan kondisi tertentu, DOM manipulation memungkinkan manipulasi elemen HTML dalam dokumen, event listener digunakan untuk menanggapi interaksi pengguna, dan fungsi digunakan untuk mengorganisir kode menjadi bagian-bagian yang dapat digunakan kembali.

Mahasiswa diharapkan dapat memahami konsep dasar ini dan mampu mengimplementasikannya dalam pembuatan program sederhana menggunakan JavaScript. Dengan pemahaman yang baik tentang konsep dasar JavaScript, mahasiswa dapat mengembangkan aplikasi web yang interaktif dan responsif.

## **Kesimpulan**

1. Pemahaman konsep dasar JavaScript, seperti variabel, tipe data, struktur kontrol, DOM manipulation, event listener, dan fungsi, merupakan fondasi yang penting dalam pengembangan aplikasi web yang interaktif dan responsif.
2. JavaScript memainkan peran kunci dalam ekosistem pengembangan web modern, memungkinkan pengembang untuk menciptakan aplikasi yang dinamis dan menarik bagi pengguna.