

LATIHAN PRAKTIKUM PEMROGRAMAN WEB

BAB : JAVASCRIPT (1)
NAMA : MONALISA MAHARANI
NIM : 215150201111004
ASISTEN : 1. AMALLIA PUTRI ANUGRAENI
2. THANA FADHILA P
TGL PRAKTIKUM: 17 MEI 2023



Modul 10 : JavaScript (1)

10.1 Waktu Pelaksanaan Praktikum

Durasi kegiatan praktikum adalah **170 menit**, dengan rincian sebagai berikut.

- 15 menit untuk pengerjaan Tes Awal atau wawancara Tugas Pendahuluan
- 60 menit untuk penyampaian materi
- 45 menit untuk pengerjaan jurnal, tes akhir atau tugas
- 50 menit **pengayaan**

10.2 Tujuan

Setelah mengikuti praktikum ini mahasiswa diharapkan dapat:

1. Memanfaatkan JavaScript (JS) Dasar untuk membuat update state atau UI Event halaman web.
2. Memanfaatkan JS Library dan Framework TensorFlow.js untuk pengembangan penerapan AI, Machine Learning dari aplikasi atau situs web.
3. Memanfaatkan JS Library dan Framework jQuery untuk pengembangan update state atau UI Event dari halaman web secara lebih mudah.
4. Memanfaatkan JS Dome untuk melakukan interaksi dinamis halaman view web, data model dan controller logic terhadap masukan pengguna.

10.3 Dasar Teori

a) JavaScript (JS) Dasar

JavaScript adalah scripting language, interpreter dan high level programming yang memungkinkan untuk melakukan pembuatan konten web baik berupa Update State maupun UI Event secara lebih dinamis dan interaktif, misal untuk mengontrol multimedia, animasi gambar, dan lainnya pada aplikasi web atau situs web.

1. *Variables*
2. *Constants*
3. *Data Types*
4. *Objects*

5. *Arrays*
6. *Functions*
7. *Conditional statements*
8. *Loops*
9. *Switch case*
10. *Print atau Display*

b) **JS Library dan Framework TensorFlow.js**

JavaScript menjadi sangat populer, salah satu alasannya adalah karena banyaknya JavaScript Library dan Frameworks yang dapat digunakan dengan mudah, misal seperti berikut:

- Node.js untuk mengembangkan aplikasi web server-side secara end-to-end hanya dengan JavaScript.
- AngularJS adalah framework web dev.-nya Google yang menyediakan banyak fitur desain utk pengembangan aplikasi modern dengan cepat.
- ReactJS adalah JavaScript framework, di mana pengelola utamanya yaitu Facebook yang sudah diterapkan pada Aplikasi Facebook dan Instagram, yang terbukti handal meski dengan arus akses yang tinggi.
- MeteorJS digunakan untuk pengembangan back-end dengan lebih cepat.
- jQuery digunakan untuk membuat situs Web menjadi lebih interaktif.
- TensorFlow.js adalah library untuk pengembangan Machine Learning (ML) model dalam JavaScript, yang tersedia dalam bentuk CDN.
- Dll.

Di mana CDN merupakan sekelompok server yang tersebar di banyak lokasi. Server ini menyimpan salinan duplikat data sehingga server dapat memenuhi permintaan data berdasarkan server mana yang paling dekat dengan pengguna. CDN membuat layanan cepat karena tidak terpengaruh oleh lalu lintas tinggi. seperti library online yang disediakan oleh suatu perusahaan pengembang yang dipasang pada platform server di berbagai lokasi berdasarkan geografis.

Untuk membuat penjumlahan bilangan, kita dapat menggunakan kode TensorFlow.js berikut contohnya:

```
1 <script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.0.0"> </script>
2
3 <input id="myButton1234" type="button" value="Use GPU math"
4 style="visibility:hidden;" onclick="{
5
6 //const myA = tf.scalar(6.1434, 'float32')
7 const myA = tf.tensor1d([10,20,30], 'int32')
8 const myB = tf.scalar(2.1, 'float32')
9 const myResult = tf.add(myA, myB);
10 //document.getElementById('myDiv1234').innerHTML
11 // = myResult.dataSync()[0].toFixed(2)
12
13 let myArray = Array.from(myResult.dataSync());
14 let result2 = myArray.map(a => a.toFixed(2));
15 document.getElementById('myDiv1234').innerHTML = result2
16
17 }"><br><br>
18 <div id='myDiv1234'>...</div><br>
```

Dapat dilihat pada bagian perintah di atas, kita menambah opsi --migration. Hal ini menginstruksikan Laravel untuk membuat migrasi untuk tabel pada basis data.

c) JS Library dan Framework jQuery

jQuery ini memudahkan dalam penanganan interaksi halaman web dengan pengguna dan lebih handal jika dibandingkan hanya menggunakan JS standard. Berikut contoh beberapa implementasinya.

- Hide Teks

```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <script
5 src="https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js"></script>
6 <script>
7 $(document).ready(function() {
8     $("p").click(function() {
9         $(this).hide();
10    });
11 });
12 </script>
13 </head>
14 <body>
15 <p>If you click on me, I will disappear.</p>
16 <p>Click me away!</p>
17 <p>Click me too!</p>
18 </body>
19 </html>
```

- Replace Teks

```
1 <script
2 src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
3 <div id="result" style="border: 1px solid red;">
4 (Klik tombol "Click Me" untuk me-replace teks ini)
5 </div>
6 <input type="button" value="Click Me" onClick="change_content('result')">
7
8 <script>
9 function change_content(id) {
10     var string = 'jQuery - Pemrograman Web';
11     $('#'+id).text(string);
12 }
13 </script>
```

- Update alert pada List

```
1 <script
2 src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.3.1/jquery.min.js"></script>
3 <p>Clicking on each number will log to console:</p>
4 <div id="div1">1</div>
5 <div id="div2">2</div>
6 <div id="div3">3</div>
7 <div id="div4">4</div>
8 <div id="div5">5</div>
9
10 <script>
11 for(let i=1; i<6; i++) {
12     $("#div" + i).click(function () { alert(i); });
```

```
13 }  
14 </script>
```

d) JS Dome

Javascript DOM (Document Object Model) adalah antarmuka pemrograman (programming interface) HTML, XML, SVG yang memungkinkan pengembang memanipulasi konten, struktur, dan style situs web secara dinamis dengan menggunakan JS, juga bisa menggunakan scripting language lainnya

- Misal menggunakan objek `document` untuk menampilkan "Hi JS DOM" pada console dengan dapat menggunakan fungsi `write` seperti berikut:

```
1 document.write("Hi JS DOM");
```

- Add elements

```
1 // membuat element b  
2 var bold = document.createElement("b");  
3  
4 // mengisi konten elemen  
5 bold.textContent = "Hi JS DOM";  
6  
7 // menambahkan elemen ke dalam tag body  
8 document.body.append(bold);
```

10.4 Prosedur Praktikum

1. Fundamental JavaScript

- a. Jawablah beberapa pertanyaan berikut!
 - o Jelaskan sejarah singkat JS, dan siapakah yang menggagasnya pada tahun berapa?

Jawab :

JavaScript (JS) adalah bahasa pemrograman yang digunakan untuk mengembangkan aplikasi web interaktif. Itu pertama kali dikembangkan oleh Brendan Eich pada tahun 1995 saat bekerja di Netscape Communications Corporation. Bahasa ini awalnya disebut "Mocha" dan kemudian "LiveScript" sebelum akhirnya dinamai "JavaScript".

Pada saat itu, web masih dalam tahap awal perkembangannya, dan Netscape sedang mencari cara untuk menyematkan skrip ke dalam browser mereka, Netscape Navigator. Brendan Eich, seorang insinyur perangkat lunak, ditugaskan untuk mengembangkan bahasa yang mudah digunakan dan dipahami oleh para pengembang web.

Pada Desember 1995, versi pertama JavaScript dirilis dengan fitur-fitur dasar seperti pengontrol aliran, fungsi, dan manipulasi elemen HTML. JavaScript memungkinkan pengembang untuk membuat halaman web yang lebih dinamis dengan mengubah elemen HTML, menangani interaksi pengguna, dan memanipulasi data secara real-time. Seiring dengan popularitas internet yang berkembang, JavaScript menjadi bahasa pemrograman yang sangat penting dalam pengembangan web. Ini memperkenalkan banyak perubahan dan peningkatan seiring berjalannya waktu. Organisasi Ecma International merilis standar JavaScript yang dikenal sebagai ECMAScript, yang

memungkinkan konsistensi dalam implementasi dan interoperabilitas di berbagai platform.

- Kenapa JavaScript (JS) disebut sebagai High Level Programming Language?

Jawab :

JavaScript (JS) disebut bahasa pemrograman tingkat tinggi karena:

- Abstraksi Tinggi: Menyediakan sintaks yang lebih mudah dipahami daripada bahasa tingkat rendah.
- Automatisasi: Menyediakan fitur dan pustaka bawaan untuk mengotomatisasi tugas-tugas umum.
- Portabilitas: Dapat dijalankan di berbagai platform dan browser tanpa modifikasi kode yang signifikan.
- Manajemen Memori Otomatis: Mengelola alokasi dan pembebasan memori secara otomatis.
- Bahasa Skrip: Berfungsi sebagai bahasa interpretatif, tidak memerlukan proses kompilasi yang rumit.

- Kenapa JavaScript (JS) disebut sebagai interpreter?

Jawab :

JavaScript (JS) disebut sebagai bahasa yang diinterpretasikan karena:

- a. Kode JavaScript dieksekusi langsung oleh interpreter saat program berjalan.
- b. Tidak memerlukan proses kompilasi terpisah sebelum eksekusi.
- c. Interpreter membaca dan menerjemahkan kode JavaScript saat program berjalan.
- d. Memiliki fleksibilitas untuk melakukan perubahan langsung pada kode sumber.
- e. Kinerja relatif lebih lambat dibandingkan dengan bahasa yang dikompilasi.

- Apakah sama JS dengan Java Programming Language, jelaskan perbedaannya?

Jawab :

JavaScript (JS) dan Java adalah dua bahasa pemrograman yang berbeda meskipun memiliki nama yang mirip:

- a. Tujuan: JavaScript digunakan untuk pengembangan aplikasi web, sementara Java digunakan untuk berbagai jenis aplikasi seperti desktop, mobile, dan enterprise.
- b. Eksekusi: JavaScript dieksekusi langsung oleh interpreter, sedangkan Java dikompilasi menjadi bytecode yang dijalankan oleh mesin virtual Java (JVM).
- c. Paradigma Pemrograman: JavaScript adalah bahasa berorientasi objek dengan dukungan untuk pemrograman fungsional dan asinkronous. Java juga berorientasi objek tetapi lebih fokus pada paradigma pemrograman berorientasi objek.
- d. Ekosistem: JavaScript memiliki banyak pustaka dan kerangka kerja untuk pengembangan web, sementara Java memiliki ekosistem yang kuat dan beragam.

- e. Kompleksitas: JavaScript cenderung lebih mudah dipelajari untuk pemula, sementara Java cenderung lebih kompleks dan memiliki kurva pembelajaran yang lebih tinggi.
- o Jelaskan perbedaan JS Lib. Vs JS Framework dan sebutkan contoh masing-masing!

Jawab :

Perbedaan antara JavaScript Library dan JavaScript Framework adalah:

- a. JavaScript Library:
 - Kumpulan fungsi-fungsi siap pakai.
 - Digunakan untuk menambahkan fungsionalitas pada aplikasi yang telah ada.
 - Pengembang memiliki kontrol penuh terhadap alur program.
 - Contoh: jQuery, Lodash, Axios.
 - b. JavaScript Framework:
 - Menyediakan struktur dan aturan untuk pengembangan aplikasi.
 - Digunakan untuk membangun aplikasi yang lebih besar dan kompleks.
 - Mengendalikan alur program dan membutuhkan pengembang untuk mengikuti konvensi dan aturan yang ditentukan.
 - Contoh: React, Angular, Vue.js.
- o Apakah JS mendukung terminology OOP?

Jawab :

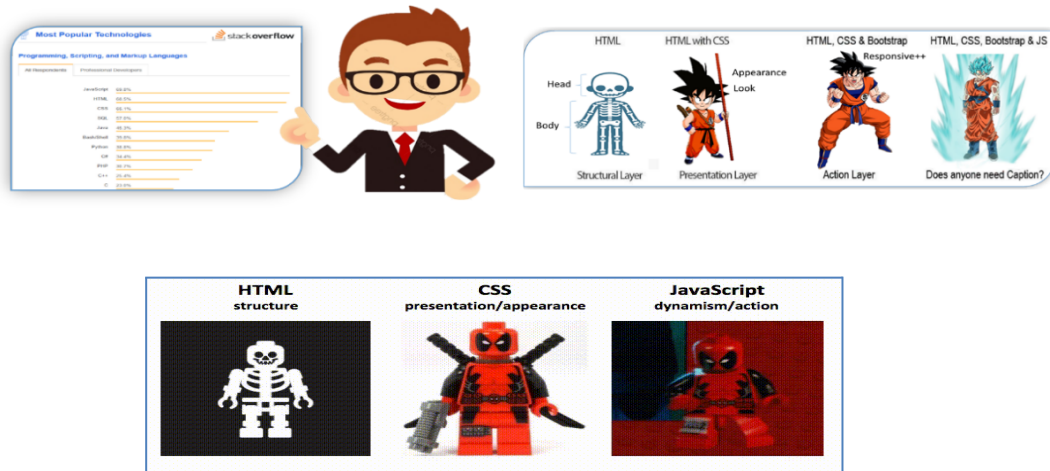
JavaScript (JS) mendukung Pemrograman Berorientasi Objek (OOP) dengan fitur berikut:

- a. Objek: Hampir semua hal dalam JavaScript diperlakukan sebagai objek.
 - b. Pewarisan: JavaScript mendukung pewarisan prototipe di mana objek dapat mewarisi properti dan metode dari objek prototipe.
 - c. Polimorfisme: Metode pada objek dapat diganti pada objek yang mewarisi objek prototipe.
 - d. Enkapsulasi: Meskipun tidak ada dukungan langsung, enkapsulasi dapat dicapai menggunakan closure atau pola modul.
 - e. JavaScript dapat mengimplementasikan konsep OOP seperti pewarisan, polimorfisme, dan enkapsulasi, meskipun dengan pendekatan yang berbeda dari bahasa pemrograman lainnya.
- o Sejak kapan JS support pada server side, siapakah penggagasnya?

Jawab :

JavaScript mulai mendukung penggunaan di sisi server melalui platform Node.js. Node.js dikembangkan oleh Ryan Dahl dan dirilis pertama kali pada tahun 2009. Ryan Dahl menggagas Node.js dengan tujuan memungkinkan JavaScript dijalankan di luar lingkungan browser dan digunakan untuk membangun aplikasi server-side yang skalabel dan efisien

b. Dari gambar berikut



Jelaskan keunggulan JS, jika dibandingkan dengan programming lainnya, dan keunggulan dari Aplikasi Web atau situs Web jika menggunakan JS, bukan hanya HTML dan CSS!

Jawab :

JavaScript (JS) memiliki sejumlah keunggulan dibandingkan dengan bahasa pemrograman lain dan memberikan nilai tambah yang signifikan saat digunakan dalam pengembangan aplikasi web. Berikut adalah beberapa keunggulan JavaScript:

- Universal Language:** JavaScript adalah bahasa pemrograman yang mendukung baik sisi klien (browser) maupun sisi server (Node.js). Ini memungkinkan pengembang menggunakan bahasa yang sama di seluruh tumpukan teknologi, mengurangi kebutuhan untuk mempelajari bahasa pemrograman tambahan.
- Kemampuan Interaktif:** JavaScript memungkinkan pembuatan interaksi dan dinamika pada halaman web. Dengan JavaScript, Anda dapat membuat elemen-elemen web menjadi responsif dan memberikan pengalaman pengguna yang lebih interaktif.
- Integrasi dengan HTML dan CSS:** JavaScript secara natif terintegrasi dengan HTML dan CSS, memungkinkan pengembang untuk mengakses dan memanipulasi elemen-elemen DOM (Document Object Model) serta mengubah tampilan dan perilaku halaman web dengan mudah.
- Pustaka dan Kerangka Kerja yang Kuat:** JavaScript memiliki ekosistem yang kaya dengan banyak pustaka dan kerangka kerja yang kuat, seperti React, Angular, dan Vue.js. Pustaka dan kerangka kerja ini menyediakan alat dan fitur yang mempercepat pengembangan aplikasi web dengan memanfaatkan komponen, manajemen state, routing, dan banyak lagi.
- Asinkronous dan Responsif:** JavaScript mendukung pemrograman asinkronous, yang memungkinkan aplikasi web untuk mengambil data dan berinteraksi dengan server secara efisien tanpa menghentikan eksekusi program. Ini menghasilkan responsivitas yang lebih baik dalam aplikasi web yang membutuhkan pemrosesan yang intensif.
- Komunitas dan Sumber Daya yang Besar:** JavaScript memiliki komunitas pengembang yang besar dan aktif serta banyak sumber daya online seperti dokumentasi, tutorial, dan forum dukungan. Ini membuatnya mudah untuk belajar dan mendapatkan bantuan jika diperlukan.

Keunggulan Aplikasi Web yang menggunakan JavaScript (JS) dibandingkan dengan hanya menggunakan HTML dan CSS adalah sebagai berikut:

- a) **Interaktivitas:** JavaScript memungkinkan pengembang untuk membuat aplikasi web yang lebih interaktif dan responsif. Dengan JavaScript, Anda dapat menambahkan efek animasi, validasi formulir secara real-time, fitur drag-and drop, dan banyak lagi.
- b) **Manipulasi DOM:** JavaScript memungkinkan manipulasi dan pemrosesan data secara dinamis melalui penggunaan API Document Object Model (DOM). Ini memungkinkan perubahan tampilan dan struktur halaman web secara dinamis berdasarkan interaksi pengguna atau data yang diperoleh dari server.
- c) **Validasi Data:** Dengan JavaScript, Anda dapat melakukan validasi data di sisi klien sebelum mengirimkannya ke server. Ini membantu memastikan bahwa data yang diinput oleh pengguna sesuai dengan persyaratan yang ditentukan dan mengurangi jumlah permintaan server yang tidak valid.

2. Konsep dan Implementasi JS Library & Framework

- a. Jelaskan Javascript (JS) dari beberapa istilah berikut jika dihubungkan untuk pengembangan Web App atau Hybrid Web /Mobile App dan khusus untuk 3 point awal, berikan contoh case sederhana beserta implementasinya:

- o JS standard

Jawab : JS Standard mengacu pada standar penulisan kode JavaScript yang umum digunakan dalam pengembangan web app atau hybrid web/mobile app. Standar ini mencakup pedoman dan praktik terbaik untuk mengorganisir, menyusun, dan menulis kode JavaScript yang mudah dibaca, dipelihara, dan diperluas

Contoh case sederhana beserta implementasi :

js standar	
1	<!DOCTYPE html>
2	<html>
3	
4	<head>
5	<title>Contoh Validasi Form dengan
6	JavaScript</title>
7	</head>
8	
9	<body>
1	<form id="myForm" onsubmit="validateForm(event)">
0	<input type="text" id="name" placeholder="Nama">
1	<input type="email" id="email"
1	placeholder="Email">
1	<input type="submit" value="Submit">
2	</form>
1	<script>
3	function validateForm(event) {

1	event.preventDefault();
4	
1	var name =
5	document.getElementById("name").value;
1	var email =
6	document.getElementById("email").value;
1	if (name === "" email === "") {
7	alert("Nama dan email harus diisi");
1	return false;
8	}
1	alert("Form berhasil dikirim!");
9	document.getElementById("myForm").reset();
2	}
0	</script>
2	</body>
1	
2	</html>
2	
2	
3	
2	
4	
2	
5	
2	
6	
2	
7	

o CDN TensorFlow.js

Jawab : TensorFlow.js adalah library JavaScript yang digunakan untuk mengembangkan aplikasi machine learning dan deep learning di sisi klien (client-side) menggunakan JavaScript. Menggunakan CDN (Content Delivery Network) TensorFlow.js memungkinkan Anda untuk mengimpor dan menggunakan TensorFlow.js melalui tautan URL eksternal, tanpa harus mengunduh atau menginstalnya secara lokal

Contoh case sederhana beserta implementasi :

CDN TensorFlow.js	
1	<!DOCTYPE html>
2	<html>
3	
4	<head>
5	<title>Contoh Klasifikasi Gambar dengan
6	TensorFlow.js</title>
7	<script
8	src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@3.8.
9	0/dist/tf.min.js"></script>

1	<script
0	src="https://cdn.jsdelivr.net/npm/@tensorflowmodels/mobi
1	lenet@2.0.1/dist/mobilenet.min.js"></script>
1	</head>
1	
2	<body>
1	<input type="file" id="imageUpload"
3	accept="image/*">
1	<img src="" id="selectedImage" width="300"
4	height="300">
1	<button
5	onclick="classifyImage()">Klasifikasi</button>
1	<script>
6	async function classifyImage() {
1	const image =
7	
1	document.getElementById("selectedImage");
8	const model = await mobilenet.load();
1	const predictions = await
9	model.classify(image);
2	console.log(predictions);
0	}
2	const imageUpload =
1	document.getElementById("imageUpload");
2	imageUpload.addEventListener("change", function
2	(event) {
2	const selectedImage =
3	
2	document.getElementById("selectedImage");
4	selectedImage.src =
2	
5	URL.createObjectURL(event.target.files[0]);
2	});
6	</script>
2	</body>
7	</html>

o jQuery

Jawab : jQuery adalah library JavaScript yang ringan dan cepat yang menyederhanakan manipulasi HTML, penanganan event, animasi, dan interaksi dengan AJAX untuk pengembangan web. jQuery memperluas fungsionalitas JavaScript dengan menyediakan sintaks yang lebih mudah digunakan dan kompatibilitas lintas browser yang lebih baik.

Contoh case sederhana beserta implementasi :

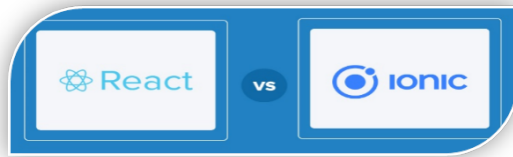
jQuery	
1	<!DOCTYPE html>

```

2 <html>
3
4 <head>
5     <title>Contoh Memuat Konten dengan jQuery</title>
6     <script
7 src="https://code.jquery.com/jquery3.6.0.min.js"></scrip
8 t>
9 </head>
10
11 <body>
12     <div id="content"></div>
13     <script>
14         $(document).ready(function () {
15             $.ajax({
16                 url: "https://api.example.com/content",
17                 success: function (data) {
18                     $("#content").html(data);
19                 },
20                 error: function () {
21                     $("#content").html("Gagal memuat
22 konten.");
23                 }
24             });
25         });
26     </script>
27 </body>
28
29 </html>
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99

```

- JS Framework yang support pada client side dan server side (Node.js) vs Native Script vs lainnya



Jawab : JS Framework yang support pada client side dan server side (Node.js) vs Native Script vs lainnya :

JS Framework yang mendukung pengembangan di sisi klien (client-side) dan sisi server (server-side), seperti Node.js, memungkinkan pengembang untuk menggunakan JavaScript sebagai bahasa pemrograman yang konsisten di seluruh aplikasi web atau hybrid web/mobile app. Framework ini menyediakan alat dan struktur yang diperlukan untuk membangun aplikasi yang efisien dan dapat diskalakan baik di sisi klien maupun server.

server side (Node.js)

```
1 // File: server.js (Node.js)
2 const express = require('express');
3 const bodyParser = require('body-parser');
4 const app = express();
5 app.use(bodyParser.json());
6 app.get('/api/data', (req, res) => {
7   const data = {
8     message: 'Halo dari server!'
9   };
10  res.json(data);
11 });
12 app.post('/api/data', (req, res) => {
13   const message = req.body.message;
14   // Melakukan penyimpanan data ke server
15   // ...
16   const response = {
17     status: 'success',
18     message: 'Data berhasil disimpan.'
19   };
20   res.json(response);
21 });
22 app.listen(3000, () => {
23   console.log('Server berjalan di
24   http://localhost:3000');
25 });
```

2	
3	
2	
4	
2	
5	
2	
6	
2	
7	

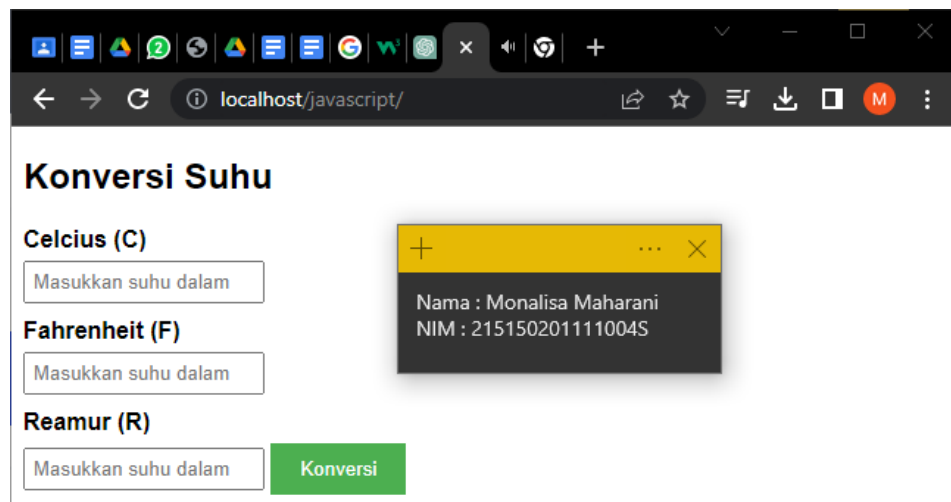
client side (index.html)	
1	<!-- File: index.html (Client-side) -->
2	<!DOCTYPE html>
3	<html>
4	
5	<head>
6	<title>Contoh Penggunaan Framework Client-side dan
7	Server-side</title>
8	<script
9	src="https://code.jquery.com/jquery3.6.0.min.js"></scrip
1	t>
0	</head>
1	
1	<body>
1	<button onclick="fetchData()">Ambil Data dari
2	Server</button>
1	<button onclick="saveData()">Simpan Data ke
3	Server</button>
1	<script>
4	function fetchData() {
1	\$.get('/api/data', function (data) {
5	console.log(data);
1	});
6	}
1	function saveData() {
7	const data = { message: 'Halo server!' };
1	\$.post('/api/data', data, function
8	(response) {
1	console.log(response);
9	});
2	}
0	</script>
2	</body>
1	
2	</html>
2	
2	
3	
2	
4	
2	
5	

2	
6	
2	
7	

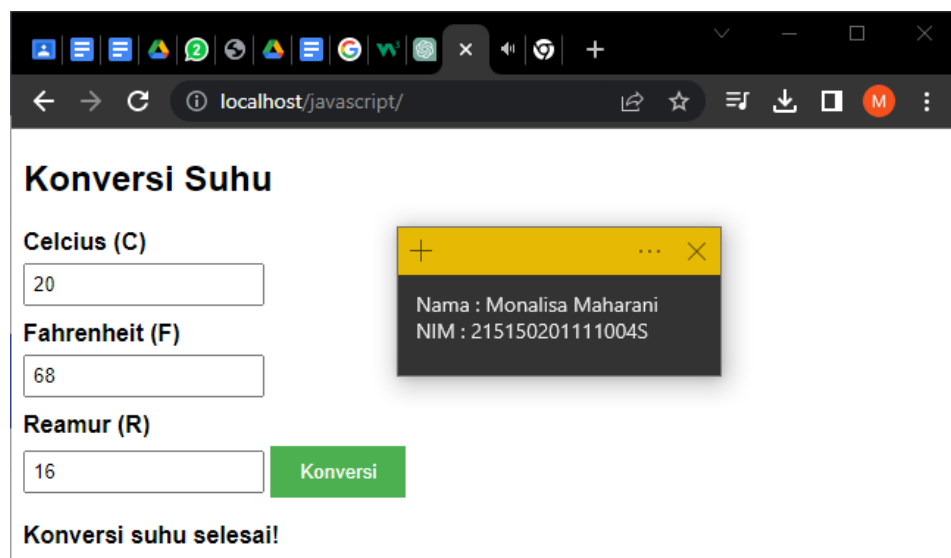
b. Berdasarkan contoh koding JS standard dengan atau tanpa CDN, misal dgn CDN TensorFlow.js, buatlah koding untuk melakukan komputasi berikut:

- o Konversi antar satuan suhu dari Celcius (C), Fahrenheit (F) dan Reamur ®

Screenshot Tampilan awal program konversi suhu.



Screenshot Tampilan hasil program konversi suhu.



index.html

```
1 <!DOCTYPE html>
2 <html>
3
4 <head>
5     <title>Konversi Suhu</title>
6     <link rel="stylesheet" type="text/css"
7 href="style.css" />
8 </head>
9
1 <body>
0     <h2>Konversi Suhu</h2>
1     <label>Celcius (C)</label>
1     <input type="number" id="celciusInput"
1 placeholder="Masukkan suhu dalam Celcius" />
2     <label>Fahrenheit (F)</label>
1     <input type="number" id="fahrenheitInput"
3 placeholder="Masukkan suhu dalam Fahrenheit" />
1     <label>Reamur (R)</label>
4     <input type="number" id="reamurInput"
1 placeholder="Masukkan suhu dalam Reamur" />
5     <button
1 onclick="convertTemperature()">Konversi</button>
6     <div id="result"></div>
1     <script src="script.js"></script>
7 </body>
1
8 </html>
```

style.css.html

```
1 body {
2     font-family: Arial, sans-serif;
3 }
4 label {
```

5	font-weight: bold;
6	display: block;
7	margin-bottom: 5px;
8	}
9	input[type="number"] {
1	padding: 5px;
0	width: 150px;
1	margin-bottom: 10px;
1	}
1	button {
2	padding: 10px 20px;
1	background-color: #4caf50;
3	color: white;
1	border: none;
4	cursor: pointer;
1	}
5	#result {
1	margin-top: 10px;
6	font-weight: bold;
1	}
7	
1	
8	
1	
9	
2	
0	
2	
1	
2	
2	
2	
3	
2	
4	
2	
5	
2	
6	
2	
7	

script.js	
1	function convertTemperature() {
2	var celciusInput =
3	document.getElementById("celciusInput");
4	var fahrenheitInput =
5	document.getElementById("fahrenheitInput");
6	var reamurInput =
7	document.getElementById("reamurInput");

1	function convertTemperature() {
2	var celciusInput =
3	document.getElementById("celciusInput");
4	var fahrenheitInput =
5	document.getElementById("fahrenheitInput");
6	var reamurInput =
7	document.getElementById("reamurInput");


```

8      var celcius = parseFloat(celciusInput.value);
9      var fahrenheit = parseFloat(fahrenheitInput.value);
10     var reamur = parseFloat(reamurInput.value);
11     if (!isNaN(celcius)) {
12         var fahrenheitResult = (celcius * 9) / 5 + 32;
13         var reamurResult = (celcius * 4) / 5;
14         fahrenheitInput.value = fahrenheitResult;
15         reamurInput.value = reamurResult;
16     } else if (!isNaN(fahrenheit)) {
17         var celciusResult = ((fahrenheit - 32) * 5) / 9;
18         var reamurResult = ((fahrenheit - 32) * 4) / 9;
19         celciusInput.value = celciusResult;
20         reamurInput.value = reamurResult;
21     } else if (!isNaN(reamur)) {
22         var celciusResult = (reamur * 5) / 4;
23         var fahrenheitResult = (reamur * 9) / 4 + 32;
24         celciusInput.value = celciusResult;
25         fahrenheitInput.value = fahrenheitResult;
26     }
27     document.getElementById("result").innerText =
28     "Konversi suhu selesai!";
29 }

```

Penjelasan :

Fungsi `convertTemperature()` berisi logika untuk mengkonversi suhu berdasarkan input yang diberikan. Pertama, fungsi ini mengambil nilai dari input suhu dalam variabel `celciusInput`, `fahrenheitInput`, dan `reamurInput`. Kemudian, fungsi ini mengubah nilai-nilai tersebut menjadi tipe data float menggunakan `parseFloat()`.

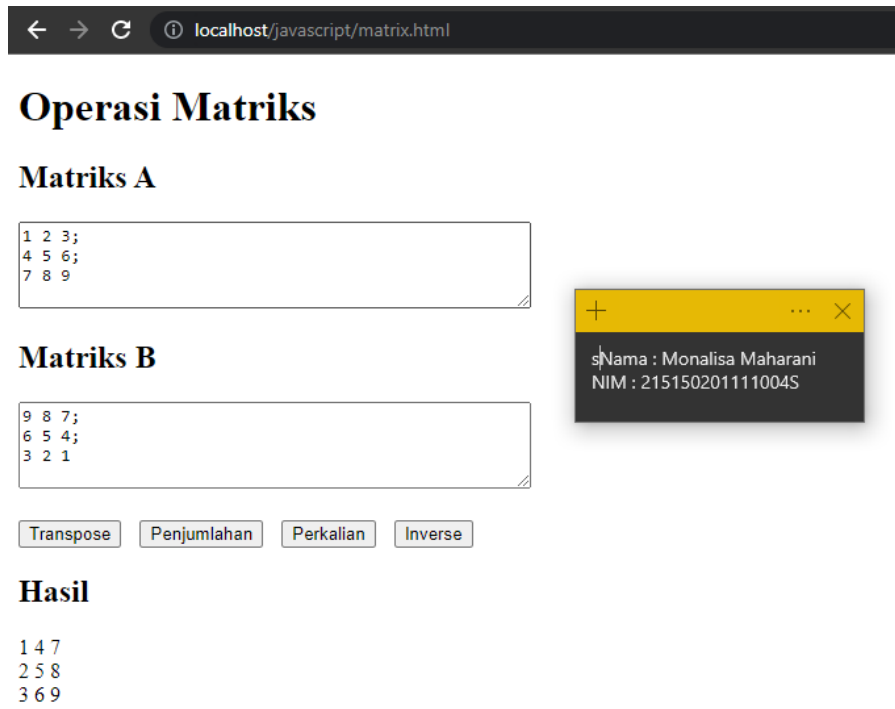
Fungsi ini kemudian memeriksa input suhu yang tidak mengandung NaN (Not-a-Number). Jika input suhu dalam Celcius valid, maka fungsi akan menghitung nilai suhu dalam Fahrenheit dan Reamur menggunakan rumus yang sesuai. Hasil konversi akan ditampilkan di input suhu Fahrenheit dan Reamur.

Jika input suhu dalam Fahrenheit valid, fungsi akan menghitung nilai suhu dalam Celcius dan Reamur menggunakan rumus yang sesuai. Hasil konversi akan ditampilkan di input suhu Celcius dan Reamur.

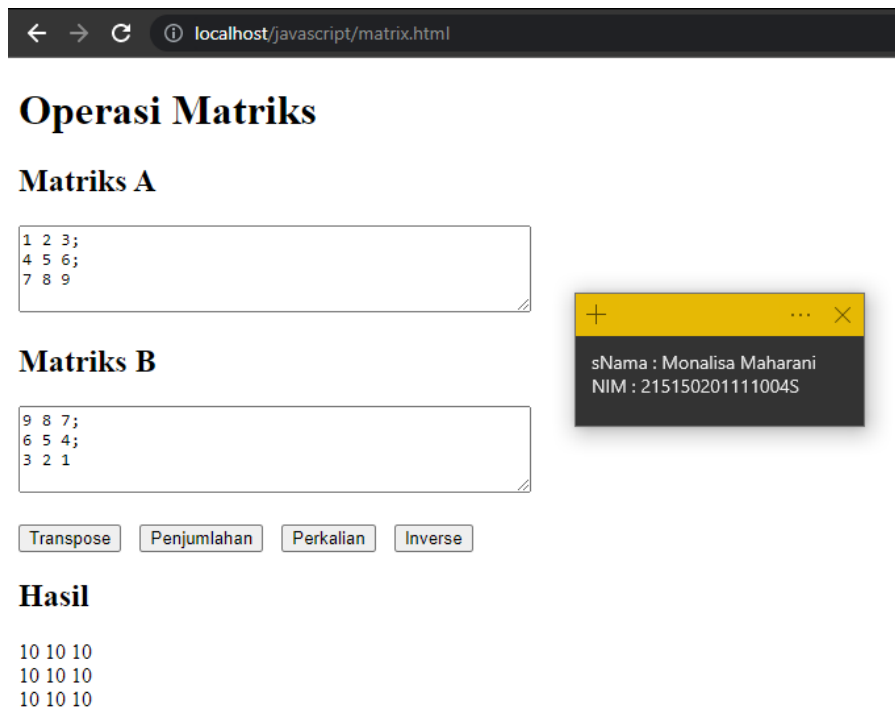
Jika input suhu dalam Reamur valid, fungsi akan menghitung nilai suhu dalam Celcius dan Fahrenheit menggunakan rumus yang sesuai. Hasil konversi akan ditampilkan di input suhu Celcius dan Fahrenheit. Terakhir, fungsi ini menampilkan pesan "Konversi suhu selesai!" di elemen dengan ID "result".

- Operasi Matrik Dasar:
 - i. Transpose, Penjumlahan,
 - ii. Perkalian dan Inverse

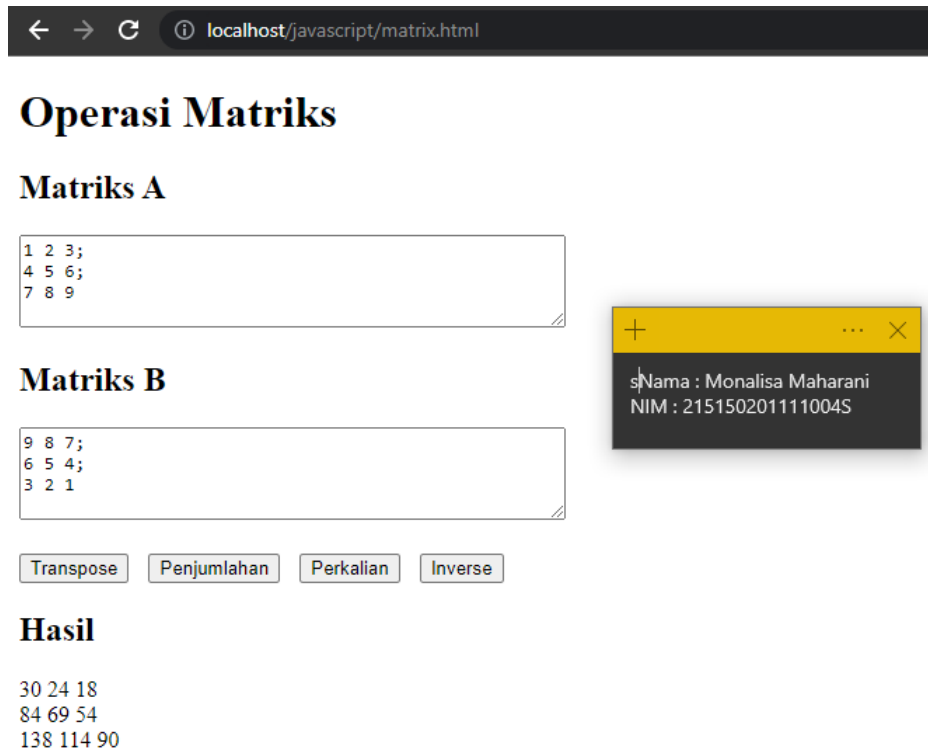
Screenshot Tampilan program operasi dasar transpose matriks.



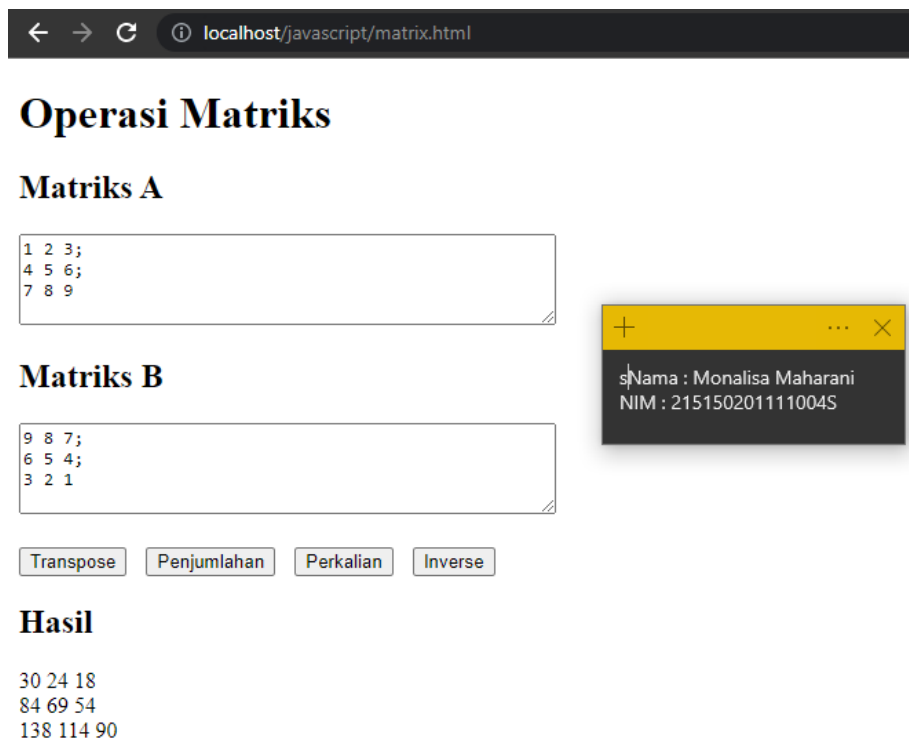
Screenshot Tampilan program operasi dasar penjumlahan matriks.



Screenshot Tampilan program operasi dasar perkalian matriks.



Screenshot Tampilan program operasi dasar inverse matriks.



matrix.html	
1	<!DOCTYPE html>

2	<html>
3	
4	<head>
5	<title>Operasi Matriks</title>
6	<link rel="stylesheet" type="text/css"
	href="matrix.css">
	</head>
	<body>
	<h1>Operasi Matriks</h1>
	<div class="matrix-container">
	<h2>Matriks A</h2>
	<textarea id="matrixAInput" rows="4"
	cols="50" placeholder="Masukkan matriks
	A..."></textarea>
	</div>
	<div class="matrix-container">
	<h2>Matriks B</h2>
	<textarea id="matrixBInput" rows="4"
	cols="50" placeholder="Masukkan matriks
	B..."></textarea>
	</div>
	<div class="buttons">
	<button
	onclick="transpose()">Transpose</button>
	<button onclick="add()">Penjumlahan</button>
	<button
	onclick="multiply()">Perkalian</button>
	<button onclick="invert()">Inverse</button>
	</div>
	<h2>Hasil</h2>
	<div id="result"></div>
	<script src="matrix.js"></script>
	</body>
	</html>

matrix.css	
1	}
2	.buttons button {
3	margin-right: 10px;
4	}
5	
6	

matrix.js	
1	function transpose() {
2	var inputA =
3	document.getElementById("matrixAInput").value;
4	var matrixA = parseMatrix(inputA);

```

5     var transposedMatrixA =
6     transposeMatrix(matrixA);
    displayMatrix(transposedMatrixA, "Hasil
Transpose");
}

function parseMatrix(input) {
    var rows = input.trim().split(';');
    var matrix = [];
    for (var i = 0; i < rows.length; i++) {
        var row = rows[i].trim().split(' ');
        matrix[i] = [];
        for (var j = 0; j < row.length; j++) {
            matrix[i][j] = parseFloat(row[j]);
        }
    }
    return matrix;
}

function transposeMatrix(matrix) {
    var transposedMatrix = [];
    for (var j = 0; j < matrix[0].length; j++) {
        transposedMatrix[j] = [];
        for (var i = 0; i < matrix.length; i++) {
            transposedMatrix[j][i] = matrix[i][j];
        }
    }
    return transposedMatrix;
}

function add() {
    var inputA =
document.getElementById("matrixAInput").value;
    var inputB =
document.getElementById("matrixBInput").value;
    var matrixA = parseMatrix(inputA);
    var matrixB = parseMatrix(inputB);
    if (matrixA.length !== matrixB.length ||
matrixA[0].length !== matrixB[0].length) {
        throw new Error("Ukuran matriks harus sama
untuk penjumlahan");
    }
    var resultMatrix = addMatrices(matrixA,
matrixB);
    displayMatrix(resultMatrix, "Hasil
Penjumlahan");
}

function addMatrices(matrix1, matrix2) {
    var rows = matrix1.length;
    var cols = matrix1[0].length;
    var resultMatrix = [];
    for (var i = 0; i < rows; i++) {
        resultMatrix[i] = [];
        for (var j = 0; j < cols; j++) {

```

```

        resultMatrix[i][j] = matrix1[i][j] +
        matrix2[i][j];
    }
}
return resultMatrix;
}

function multiply() {
    var inputA =
document.getElementById("matrixAInput").value;
    var inputB =
document.getElementById("matrixBInput").value;
    var matrixA = parseMatrix(inputA);
    var matrixB = parseMatrix(inputB);
    if (matrixA[0].length !== matrixB.length) {
        throw new Error("Jumlah kolom matriks A
harus sama dengan jumlah baris matriks B untuk
perkalian");
    }
    var resultMatrix =
multiplyMatrices(matrixA,matrixB);
    displayMatrix(resultMatrix, "Hasil Perkalian");
}

function multiplyMatrices(matrix1, matrix2) {
    var rows1 = matrix1.length;
    var cols1 = matrix1[0].length;
    var rows2 = matrix2.length;
    var cols2 = matrix2[0].length;
    if (cols1 !== rows2) {
        throw new Error("Jumlah kolom matriks
pertama harus sama dengan jumlah baris matriks
kedua");
    }
    var resultMatrix = [];
    for (var i = 0; i < rows1; i++) {
        resultMatrix[i] = [];
        for (var j = 0; j < cols2; j++) {
            resultMatrix[i][j] = 0;
            for (var k = 0; k < cols1; k++) {
                resultMatrix[i][j] += matrix1[i][k] *
matrix2[k][j];
            }
        }
    }
    return resultMatrix;
}

function invert() {
    var inputA =
document.getElementById("matrixAInput").value;
    var matrixA = parseMatrix(inputA);
    if (matrixA.length !== matrixA[0].length) {
        throw new Error("Matriks harus berbentuk
persegi untuk diinvers");
    }
}

```

```

    }
    var inverseMatrix = invertMatrix(matrixA);
    displayMatrix(inverseMatrix, "Hasil Inverse");
}

function invertMatrix(matrix) {
    var rows = matrix.length;
    var cols = matrix[0].length;
    if (rows !== cols) {
        throw new Error("Matriks harus berbentuk persegi
untuk diinvers");
    }
    var identityMatrix = createIdentityMatrix(rows);
    var augmentedMatrix = [];
    for (var i = 0; i < rows; i++) {
        augmentedMatrix[i] =
matrix[i].concat(identityMatrix[i]);
    }
    for (var k = 0; k < rows; k++) {
        var pivotRow = k;
        var pivotElement =
augmentedMatrix[pivotRow][k];
        if (pivotElement === 0) {
            throw new Error("Matriks tidak dapat
diinvers");
        }
        for (var j = 0; j < 2 * cols; j++) {
            augmentedMatrix[pivotRow][j] /=
pivotElement;
        }
        for (var i = 0; i < rows; i++) {
            if (i !== pivotRow) {
                var factor = augmentedMatrix[i][k];
                for (var j = 0; j < 2 * cols; j++) {
                    augmentedMatrix[i][j] -= factor
* augmentedMatrix[pivotRow][j];
                }
            }
        }
    }
    var inverseMatrix = [];
    for (var i = 0; i < rows; i++) {
        inverseMatrix[i] =
augmentedMatrix[i].slice(cols);
    }
    return inverseMatrix;
}

function createIdentityMatrix(size) {
    var matrix = [];
    for (var i = 0; i < size; i++) {
        matrix[i] = [];
        for (var j = 0; j < size; j++) {
            matrix[i][j] = (i === j) ? 1 : 0;
        }
    }
}

```

```

    }
    return matrix;
}

function displayMatrix(matrix) {
    var result = "";
    for (var i = 0; i < matrix.length; i++) {
        result += matrix[i].join(' ') + '<br>';
    }
    document.getElementById("result").innerHTML =
result;
}

```

Penjelasan :

parseMatrix(input): Fungsi ini digunakan untuk mengubah input matriks dalam bentuk string menjadi bentuk matriks dalam bentuk array dua dimensi. Input matriks dipisahkan berdasarkan baris dengan pemisah ; dan setiap elemen matriks dalam satu baris dipisahkan dengan spasi. Fungsi ini mengembalikan matriks dalam bentuk array dua dimensi.

transposeMatrix(matrix): Fungsi ini digunakan untuk melakukan operasi transpose pada matriks. Matriks input diterima dalam bentuk array dua dimensi. Fungsi ini menghasilkan matriks transposenya dengan menukar posisi baris dan kolom.

addMatrices(matrix1, matrix2): Fungsi ini digunakan untuk menjumlahkan dua matriks. Matriks input diterima dalam bentuk array dua dimensi. Fungsi ini memeriksa terlebih dahulu apakah ukuran matriks sama. Jika ukurannya sama, fungsi akan menjumlahkan elemen-elemen matriks dengan aturan penjumlahan matriks.

multiplyMatrices(matrix1, matrix2): Fungsi ini digunakan untuk mengalikan dua matriks. Matriks input diterima dalam bentuk array dua dimensi. Fungsi ini memeriksa terlebih dahulu apakah jumlah kolom matriks pertama sama dengan jumlah baris matriks kedua. Jika memenuhi syarat, fungsi akan mengalikan elemen-elemen matriks dengan aturan perkalian matriks.

invertMatrix(matrix): Fungsi ini digunakan untuk menghitung invers matriks. Matriks input diterima dalam bentuk array dua dimensi. Fungsi ini memeriksa terlebih dahulu apakah matriks merupakan matriks persegi. Jika matriks persegi, fungsi akan melakukan proses eliminasi Gauss-Jordan untuk mendapatkan matriks inversnya. Jika tidak memenuhi syarat, fungsi akan menghasilkan pesan kesalahan.

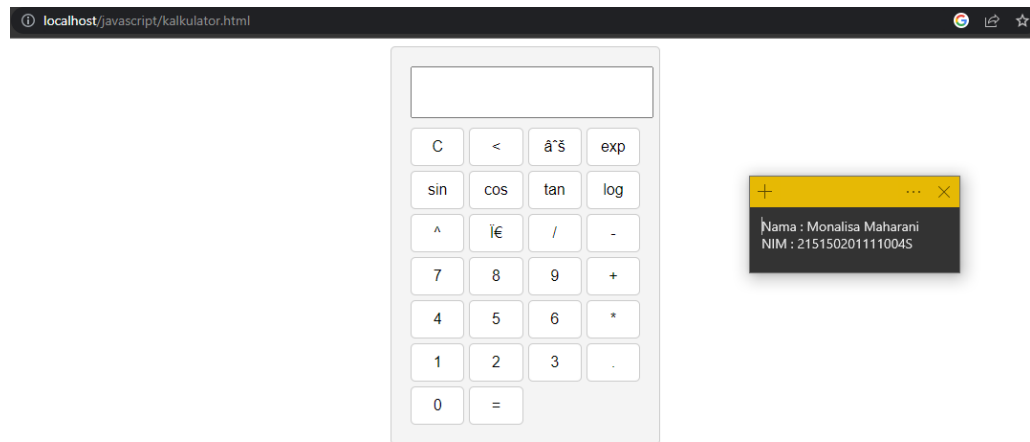
createIdentityMatrix(size): Fungsi ini digunakan untuk membuat matriks identitas dengan ukuran yang ditentukan. Matriks identitas adalah matriks persegi dengan elemen diagonal utama bernilai 1 dan elemen lainnya bernilai 0.

displayMatrix(matrix): Fungsi ini digunakan untuk menampilkan matriks ke elemen HTML dengan ID "result". Matriks diterima dalam bentuk array dua dimensi. Fungsi ini akan

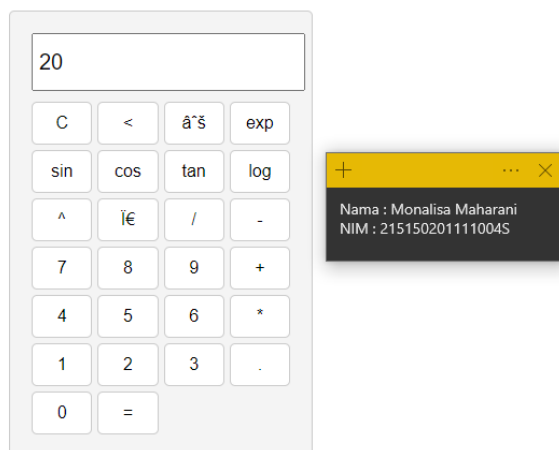
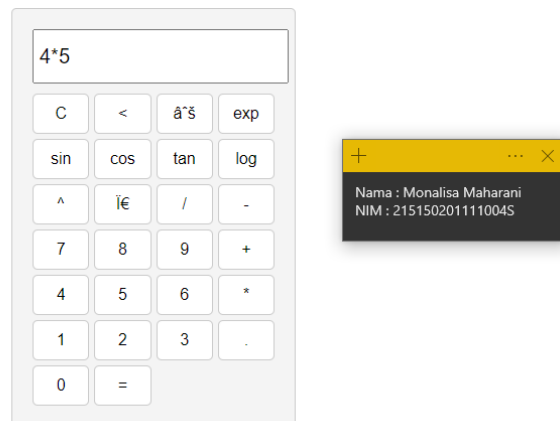
mengonversi matriks menjadi string dengan menggunakan join() dan menambahkan tag
 untuk setiap baris matriks.

- Membuat Kalkulator Scientific

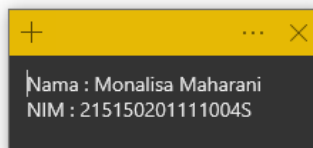
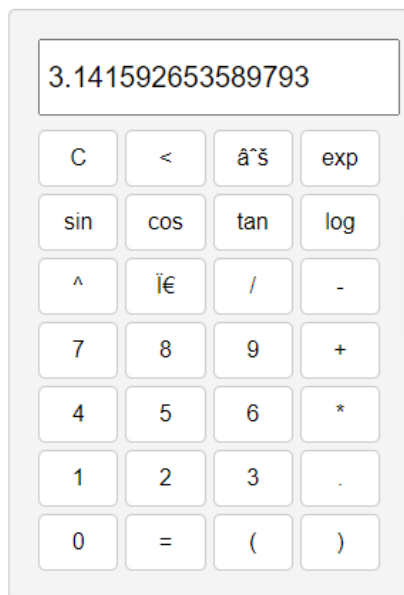
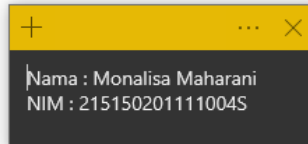
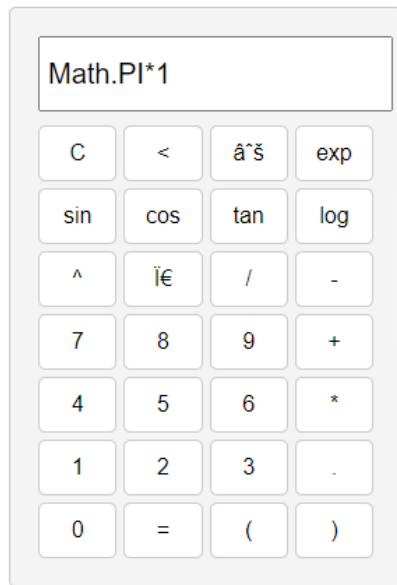
Screenshot Tampilan program kalkulator scientific.



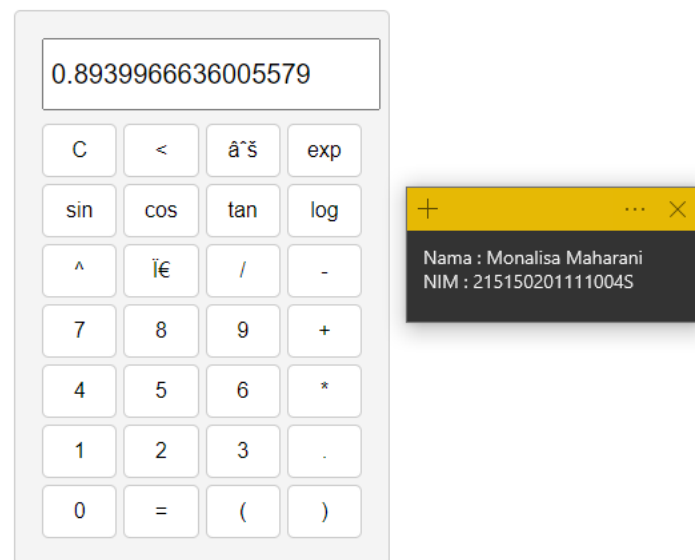
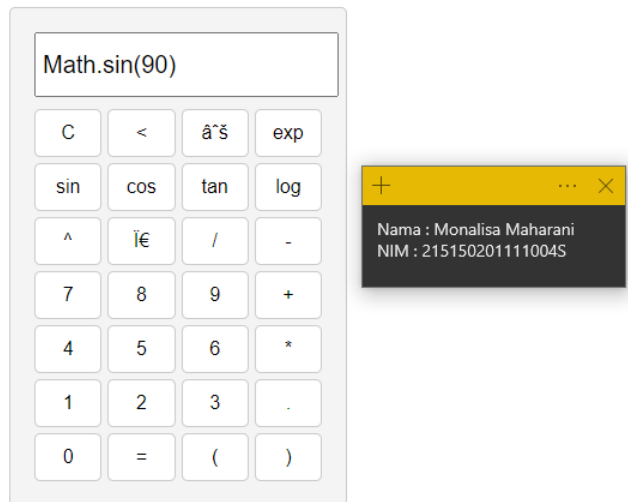
Screenshot Tampilan percobaan perhitungan dengan program kalkulator scientific.



Screenshot Tampilan percobaan ke 2 perhitungan dengan program kalkulator scientific.



Screenshot Tampilan percobaan perhitungan dengan program kalkulator scientific.



kalkulator.html	
1	<code><!DOCTYPE html></code>
2	<code><html></code>
3	<code><head></code>
4	<code> <title>Kalkulator Sains</title></code>
5	<code> <link rel="stylesheet" type="text/css"</code>
6	<code>href="kalkulator.css"></code>
	<code></head></code>
	<code><body></code>
	<code> <div class="calculator"></code>
	<code> <input type="text" id="result" readonly></code>
	<code> <div class="keypad"></code>
	<code> <button onclick="clearResult()">C</button></code>
	<code> <button onclick="deleteChar()"><<</button></code>
	<code> <button onclick="calculate('sqrt(')">√</button></code>

```

        <button
onclick="calculate('exp(')">exp</button>
        <button
onclick="calculate('Math.sin(')">sin</button>
        <button
onclick="calculate('Math.cos(')">cos</button>
        <button
onclick="calculate('Math.tan(')">tan</button>
        <button
onclick="calculate('Math.log(')">log</button>
        <button
onclick="calculate('Math.pow(')">^</button>
        <button
onclick="calculate('Math.PI'">π</button>
        <button onclick="calculate('/')">/</button>
        <button onclick="calculate('-')">-</button>
        <button
onclick="appendToResult('7'">7</button>
        <button
onclick="appendToResult('8'">8</button>
        <button
onclick="appendToResult('9'">9</button>
        <button onclick="calculate('+')">+</button>
        <button
onclick="appendToResult('4'">4</button>
        <button
onclick="appendToResult('5'">5</button>
        <button
onclick="appendToResult('6'">6</button>
        <button onclick="calculate('*')">*</button>
        <button
onclick="appendToResult('1'">1</button>
        <button
onclick="appendToResult('2'">2</button>
        <button
onclick="appendToResult('3'">3</button>
        <button
onclick="appendToResult('.')">.</button>
        <button
onclick="appendToResult('0'">0</button>
        <button onclick="calculateResult()">=</button>
        <button onclick="appendChar('(')">(</button>
        <button onclick="appendChar(')')">)</button>

    </div>
</div>
<script src="kalkulator.js"></script>
</body>
</html>

```

1	.calculator {
2	width: 240px;
3	margin: 0 auto;
4	padding: 20px;
5	background-color: #f4f4f4;
6	border: 1px solid #ccc;
7	border-radius: 5px;
8	}
9	
10	#result {
	width: 100%;
	height: 40px;
	margin-bottom: 10px;
	padding: 5px;
	font-size: 20px;
	}
	.keypad {
	display: grid;
	grid-template-columns: repeat(4, 1fr);
	grid-gap: 5px;
	}
	button {
	width: 100%;
	height: 40px;
	font-size: 16px;
	background-color: #fff;
	border: 1px solid #ccc;
	border-radius: 5px;
	cursor: pointer;
	}
	button:hover {
	background-color: #f4f4f4;
	}

kalkulator.js	
---------------	--

1	function appendToResult(value) {
2	document.getElementById('result').value +=
3	value;
4	}
5	
6	function calculate(operator) {
7	document.getElementById('result').value +=
8	operator;
9	}
10	function appendChar(char) {
	document.getElementById("result").value += char;
	}

```

function calculateResult() {
    const result =
document.getElementById('result').value;
    let calculatedResult = '';

    try {
        calculatedResult = eval(result);
    } catch (error) {
        calculated
        Result = 'Error';
    }

    document.getElementById('result').value =
calculatedResult;
}

function clearResult() {
document.getElementById('result').value = '';
}

function deleteChar() {
    const result =
document.getElementById('result').value;
    document.getElementById('result').value =
result.slice(0, -1);
}

```

Penjelasan :

appendToResult(value): Fungsi ini akan menambahkan nilai value ke dalam input hasil (elemen dengan id "result"). Fungsi ini digunakan ketika tombol angka atau tanda operasi ditekan.

calculate(operator): Fungsi ini akan menambahkan operator matematika atau fungsi matematika ke dalam input hasil. Fungsi ini digunakan ketika tombol operator atau fungsi matematika ditekan.

appendChar(char): Fungsi ini akan menambahkan karakter char ke dalam input hasil. Fungsi ini digunakan untuk menambahkan karakter khusus seperti tanda kurung.

calculateResult(): Fungsi ini akan menghitung hasil ekspresi matematika yang ada dalam input hasil. Fungsi eval() digunakan untuk mengevaluasi ekspresi matematika yang diinputkan. Jika hasil evaluasi valid, maka hasilnya akan ditampilkan dalam input hasil. Jika terjadi kesalahan selama evaluasi, maka pesan "Error" akan ditampilkan.

clearResult(): Fungsi ini akan menghapus semua nilai yang ada dalam input hasil sehingga menjadi kosong.

deleteChar(): Fungsi ini akan menghapus karakter terakhir dari input hasil. Fungsi slice() digunakan untuk membuang karakter terakhir dari string yang ada dalam input hasil.

3. Membuat Implementasi Interaktif dan mendukung CRUD dengan Full JS (Dome, dll.)

- a. Perhatikan JS DOM berikut:

```
..  
  
    <body>  
  
        <p class="id">Belajar Form</p>  
  
        <p class="id">Belajar DB MySQL</p>  
  
        <p class="id">Belajar JS</p>  
  
        <p class="id">Belajar Ajax</p>  
  
        <script>  
  
            var val = document.getElementsByClassName("id");  
  
            console.log(val);  
  
        </script>  
  
..
```

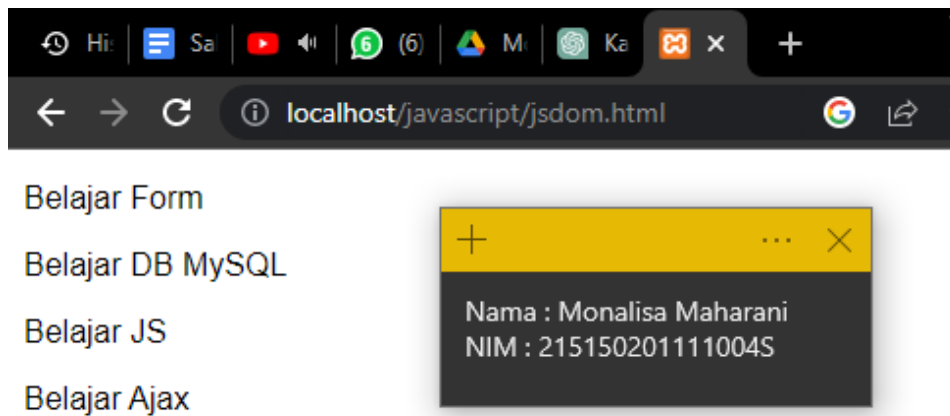
Dan

```
setInterval(function () {  
  
    val[0].style.color = "red";  
  
    val[1].style.color = "green";  
  
    val[2].style.color = "blue";  
  
    val[3].style.color = "orange";  
  
    setTimeout(function () {  
  
        val[0].style.color = "black";  
  
        val[1].style.color = "black";
```

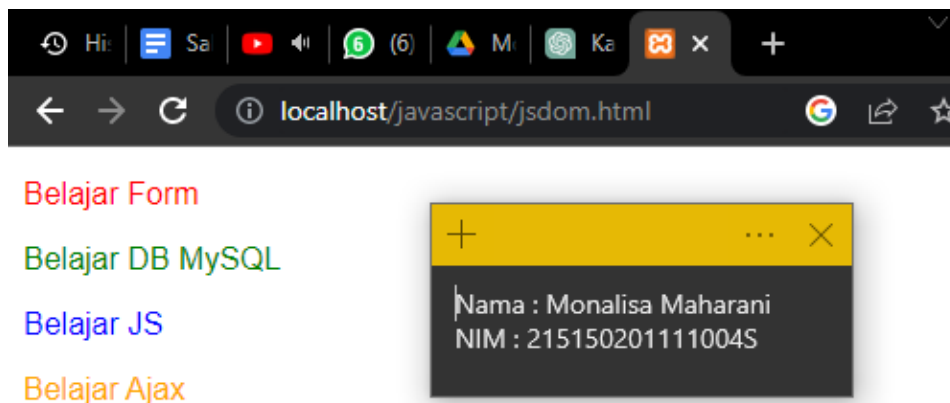
```
        val[2].style.color = "black";  
  
        val[3].style.color = "black";  
  
    }, 1000)  
  
}, 2000);
```

Bagaimana hasil yang ditampilkan jika pada “console..” diubah dengan “setInterval..” di atas!

Screenshot Tampilan program sebelum console diubah dengan setInterval.



Screenshot Tampilan program setelah console diubah dengan setInterval.



Penjelasan:

Perubahan yang terjadi adalah fungsi setInterval dieksekusi setiap 2000 ms dan di dalam fungsi ini terdapat setTimeout di mana akan mengeksekusi perintah setelah menunggu selama 1000 ms. Fungsi setInterval bertindak sebagai perulangan tanpa henti dengan delay selama 2000 ms, sedangkan fungsi setTimeout bertindak sebagai pengubah kembali ke warna hitam dengan waktu tunggu 1000 ms. Hal ini yang menyebabkan tulisan pada halaman dapat berganti-ganti warna.

- b. Buatlah Web “Students App” CRUD sederhana yang datanya pada temp. dgn kode full JS (Lib & Framework JS) + HTML + CSS, seperti tampilan berikut untuk manage data Mahasiswa secara dinamis dan interaktif!

Add Mahasiswa

1	harry	porter	46	Show	Edit	Delete
2	jonn	deep	25	Show	Edit	Delete

Screenshot Tampilan saat menuliskan data mahasiswa.

Nama:

Monalisa Maharani

NIM:

215150201111004

IPK:

4.00

Add Mahasiswa

Nama : Monalisa Maharani

NIM : 215150201111004S

Data Mahasiswa :

Nama	NIM	IPK	Update/Delete
------	-----	-----	---------------

Screenshot Tampilan saat setelah data mahasiswa ditambahkan.

Nama:

NIM:

IPK:

Add Mahasiswa

Nama : Monalisa Maharani

NIM : 215150201111004S

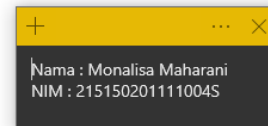
Data Mahasiswa :

Nama	NIM	IPK	Update/Delete
Monalisa Maharani	215150201111004	4.00	Show Edit Delete

Screenshot Tampilan ketika terdapat 2 atau lebih data mahasiswa pada tabel data mahasiswa.

Data Mahasiswa :

Nama	NIM	IPK	Update/Delete
Monalisa Maharani	215150201111004	4.00	<button>Show</button> <button>Edit</button> <button>Delete</button>
Ratu Veronica	215150201111005	3.98	<button>Show</button> <button>Edit</button> <button>Delete</button>
Cantika Putri	215150201111006	3.75	<button>Show</button> <button>Edit</button> <button>Delete</button>

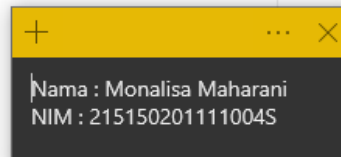


Screenshot Tampilan saat kita mengakses tombol Show.

Nama:

NIM:

IPK:



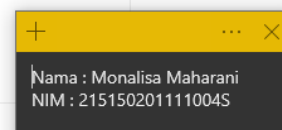
Screenshot Tampilan saat kita mengakses tombol Edit.

Nama:

NIM:

IPK:

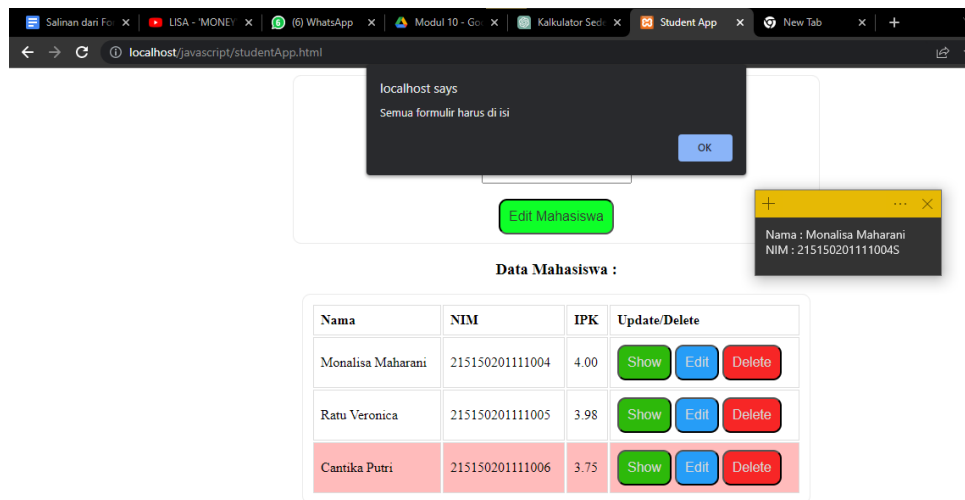
Edit Mahasiswa



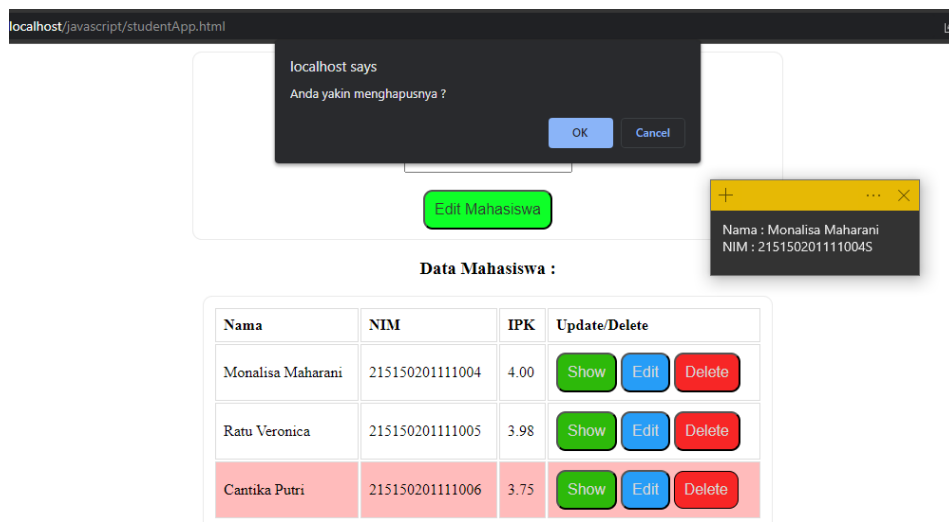
Data Mahasiswa :

Nama	NIM	IPK	Update/Delete
Monalisa Maharani	215150201111004	4.00	<button>Show</button> <button>Edit</button> <button>Delete</button>
Ratu Veronica	215150201111005	3.98	<button>Show</button> <button>Edit</button> <button>Delete</button>
Cantika Putri	215150201111006	3.75	<button>Show</button> <button>Edit</button> <button>Delete</button>

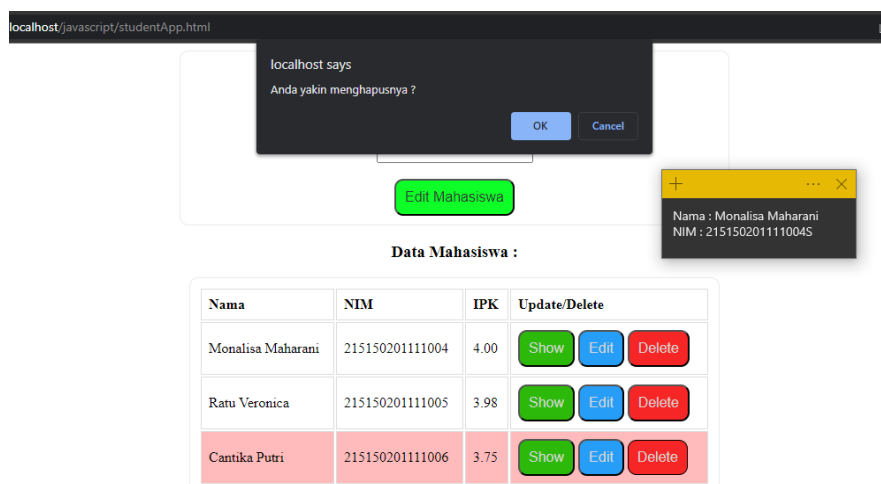
Screenshot Tampilan saat kita tidak memasukkan data pada halaman Edit.



Screenshot Tampilan saat kita mengakses tombol delete.



Screenshot Tampilan saat kita mengonfirmasi "Hapus" data mahasiswa pada tabel.



studentApp.html	
1	<!doctype html>
2	<html lang="en">
3	
4	<head>
5	<meta charset="utf-8">
6	<title>Student App</title>
7	<style>
8	body {
9	text-align: center;
10	}
11	
12	table {
13	align-items: center;
14	text-align: center;
15	max-width: 600px;
16	padding: 10px;
17	width: 100%;
18	background: #FFFFFF;
19	border-radius: 10px;
20	border: 1px solid #ebebeb;
21	margin: auto;
22	}
23	
24	form {
25	max-width: 600px;
26	padding: 10px;
27	width: 100%;
28	background: #FFFFFF;
29	border-radius: 10px;
30	border: 1px solid #ebebeb;
31	margin: auto;
32	}
33	
34	td,
35	th {
36	border: 1px solid #dddddd;
37	text-align: left;
38	padding: 8px;
39	}
40	
41	#edit {
42	background: #269df7;
43	-webkit-border-radius: 10px;
44	-moz-border-radius: 10px;
45	border-radius: 10px;
46	color: #D7D7D7;
47	font-family: Arial;
48	font-size: 16px;
49	font-weight: 100;
50	padding: 10px;
51	text-decoration: none;
52	display: inline-block;

```
53         cursor: pointer;
54         text-align: center
55     }
56
57     #delete {
58         background: #F72626;
59         -webkit-border-radius: 10px;
60         -moz-border-radius: 10px;
61         border-radius: 10px;
62         color: #D7D7D7;
63         font-family: Arial;
64         font-size: 16px;
65         font-weight: 100;
66         padding: 10px;
67         text-decoration: none;
68         display: inline-block;
69         cursor: pointer;
70         text-align: center;
71     }
72
73     #show {
74         background: #2db90a;
75         -webkit-border-radius: 10px;
76         -moz-border-radius: 10px;
77         border-radius: 10px;
78         color: #D7D7D7;
79         font-family: Arial;
80         font-size: 16px;
81         font-weight: 100;
82         padding: 10px;
83         text-decoration: none;
84         display: inline-block;
85         cursor: pointer;
86         text-align: center;
87     }
88
89     #tombolSubmit {
90         background: #0EFF28;
91         -webkit-border-radius: 10px;
92         -moz-border-radius: 10px;
93         border-radius: 10px;
94         color: #343434;
95         font-family: Arial;
96         font-size: 16px;
97         font-weight: 100;
98         padding: 10px;
99         text-decoration: none;
100        display: inline-block;
101        cursor: pointer;
102        text-align: center;
103    }
104
105    button:hover {
106        border: solid #000000 1px;
107        background: #000000;
```



```

108         background-image:
109 -webkit-linear-gradient(top, #1E62D0, #3D94F6);
110         background-image: -moz-linear-gradient(top,
111 #1E62D0, #3D94F6);
112         background-image:
113 -ms-linear-gradient(top, #1E62D0, #3D94F6);
114         background-image:
115 -o-linear-gradient(top, #1E62D0, #3D94F6);
116         background-image:
117 -webkit-border-gradient(to bottom, #1E62D0, #3D94F6);
118         -webkit-border-radius: 10px;
119         -moz-border-radius: 10px;
120         border-radius: 10px;
121         text-decoration: none;
122     }
123 </style>
124 </head>
125
126 <body>
127     <form
128 onsubmit="event.preventDefault();onFormSubmit();">
129         <label>Nama:</label><br>
130         <input type="text" id="name"><br>
131         <label>NIM:</label><br>
132         <input type="text" id="nim"><br>
133         <label>IPK:</label><br>
134         <input type="text" id="ipk"><br>
135         <br>
136         <input type="submit" id="tombolSubmit"
137 value="Add Mahasiswa">
138     </form>
139     <div id="tabel">
140         <table id="tableMahasiswa">
141             <h3>Data Mahasiswa :</h3>
142             <thead>
143                 <tr>
144                     <th>Nama</th>
145                     <th>NIM</th>
146                     <th>IPK</th>
147                     <th>Update/Delete</th>
148                 </tr>
149             </thead>
150             <tbody id="bodyTable">
151                 </tbody>
152         </table>
153     </div>
154     <script>
155         var activeRow = null
156         function resetForm() {
157
158             document.getElementById("name").value = "";
159             document.getElementById("nim").value
160                 = "";
161             document.getElementById("ipk").value
162                 = "";
163

```

```

164
165
166 document.getElementById("tombolSubmit").value = "Add
167 Mahasiswa"
168         if (activeRow != null) {
169             activeRow.style.background = "#ffffff"
170         }
171         activeRow = null;
172     }
173
174     //CREATE
175     function insertNewData(data) {
176         var table =
177
178 document.getElementById("tableMahasiswa");
179         var newrow =
180             table.insertRow(table.length)
181             var tmp = newrow.insertCell(0);
182             tmp.innerHTML = data.name;
183             var t1 = newrow.insertCell(1)
184             t1.innerHTML = data.nim;
185             var t2 = newrow.insertCell(2)
186             t2.innerHTML = data.ipk;
187             var t3 = newrow.insertCell(3)
188             t3.innerHTML = `<button
189 onClick="readData(this)" id="show">Show</button>
190     <button onClick="editData(this)"
191 id="edit">Edit</button>
192     <button onClick="deleteData(this)"
193 id="delete">Delete</button>`;
194         }
195
196     //READ
197     function readData() {
198         var data = {};
199         data["name"] =
200             document.getElementById("name").value;
201         data["nim"] =
202             document.getElementById("nim").value;
203         data["ipk"] =
204             document.getElementById("ipk").value;
205         return data;
206     }
207
208     //UPDATE
209     function editData(data) {
210         activeRow =
211             data.parentElement.parentElement;
212
213         document.getElementById("name").value =
214             activeRow.cells[0].innerHTML;
215         document.getElementById("nim").value
216             = activeRow.cells[1].innerHTML;
217         document.getElementById("ipk").value
218

```

```

219         = activeRow.cells[2].innerHTML;
220         activeRow.style.background = "#ffbfff"
221
222     document.getElementById("tombolSubmit").value = "Edit
223 Mahasiswa"
224     }
225     function updateData(data) {
226         activeRow.cells[0].innerHTML = data.name
227         activeRow.cells[1].innerHTML = data.nim
228         activeRow.cells[2].innerHTML = data.ipk
229     }
230
231     //DELETE
232     function deleteData(data) {
233         if (confirm('Anda yakin menghapusnya ? '))
234     {
235         row = data.parentElement.parentElement;
236
237
238     document.getElementById("tableMahasiswa").deleteRow(row
239 .rowIndex);
240         resetForm();
241     }
242     }
243     function onFormSubmit() {
244         isValid = false;
245         if
246         (document.getElementById("name").value
247 == ""
248         ||
249         document.getElementById("nim").value ==
250 ""
251         ||
252         document.getElementById("ipk").value ==
253 "") {
254             alert("Semua formulir harus di isi")
255         } else if
256         (document.getElementById("ipk").value <
257 0.0 ||
258         document.getElementById("ipk").value >
259 4.0) {
260             alert("IPK Harus positif dan kurang
261 dari 4.0")
262         } else {
263             var data = readData()
264             if (activeRow == null) {
265                 insertNewData(data)
266             } else {
267                 updateData(data)
268             }
269             resetForm()
270         }
271     }
272     </script>
273 </body>

```

Penjelasan :

Pada baris 115 terdapat `prevent default` yang bertujuan untuk mencegah autosubmitting form (live form) karena penyimpanan datanya tidak berkaitan dengan database. Selain itu juga method `onFormSubmit()` yang dipanggil sebagai form handler-nya sehingga jika field inputan pada form ada yang kosong maka akan menghasilkan alert apabila terdapat field yang kosong, jika ipknya melebihi dan kurang dari batas yang ditentukan juga mengeluarkan alert, dan pengelolaan form. Kemudian terdapat function `resetForm()` yang digunakan untuk menghapus form mereset kembali isi form baik setelah merefresh atau reopen tampilan tersebut. `insertNewData()` untuk menambahkan data baru pada tabel daftar mahasiswa, dalam fungsi ini terdapat `insertRow()` dan `insertCell()` untuk membentuk secara otomatis baris data yang ditambahkan. `readData()` hanya untuk membaca atau melihat tabel data mahasiswa. `editData()` untuk mengubah data pada tabel mahasiswa dengan memasukkan nilainya ke dalam form kembali jadi pengeditan dilakukan pada form dan `updateData()` untuk memperbarui data yang telah diedit tersebut. `deleteData()` untuk menghapus data pada tabel mahasiswa. Setiap melakukan pengelolaan form dan tabel memanfaatkan DOM JS untuk manipulasi model atau elemen pada HTML seperti id, class, tag, dll. Baris 6-112 adalah cssnya untuk mempercantik tampilan.