

TUGAS PEMROGRAMAN WEB



Disusun Oleh :

Nama : Ghazi Alvin Karim

Nim : 121450123

Mata Kuliah : Pemrograman Web (RB)

**PROGRAM STUDI SAINS DATA
JURUSAN SAINS
INSTITUT TEKNOLOGI SUMATERA
2024/2025**

• Tugas Individu 2

1. Terdapat dua halaman

- Halaman pendaftaran yang di desain menggunakan form dengan method **POST**
- Halaman Detail informasi pendaftaran yang di buat dummy
- Desain Query string untuk melempar data dari halaman pendaftaran ke dalam detail informasi pendaftaran, hanya mendesain query string,tidak menangkap data.

2. Desain kedua halaman

- Menggunakan minimal 3 selector
- Desain layout,box, tabel dan form menggunakan **CSS**

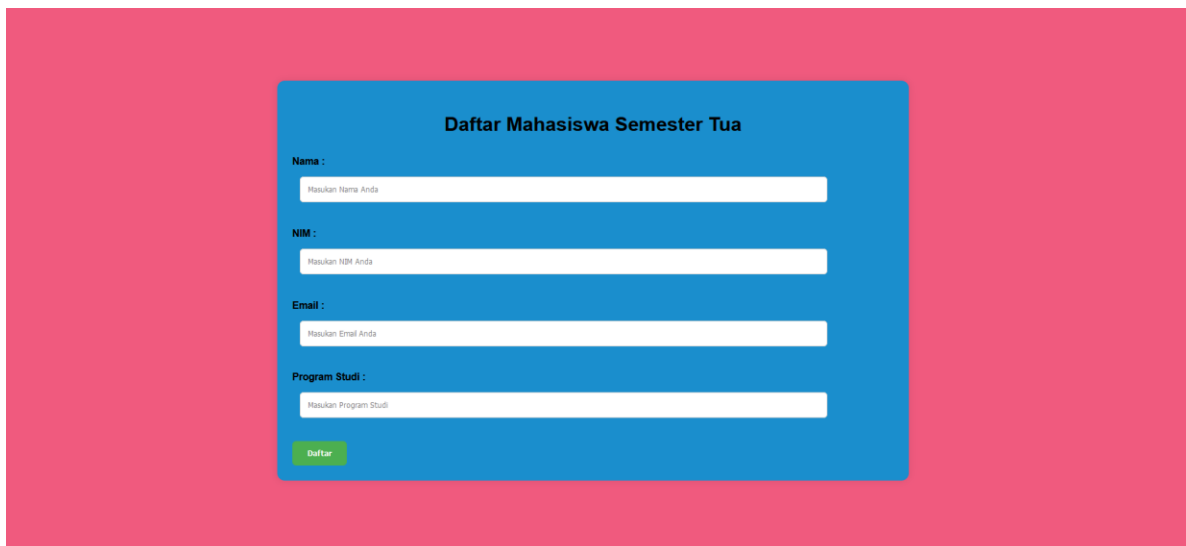
1. Halaman Pendaftaran

```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <title>Tugas Pemrograman Web</title>
7   <link rel="stylesheet" href="style.css">
8   <script src="data.js" ></script>
9
10 </head>
11 <body>
12   <div class="container">
13     <h1>Daftar Mahasiswa Semester Tua</h1>
14     <form onsubmit="saveData(); return true;" action="information.html">
15
16       <label for="nama">Nama :</label>
17       <input type="text" id="nama" name="nama" placeholder="Masukan Nama Anda" required>
18       <br><br>
19
20       <label for="Nim">NIM :</label>
21       <input type="text" id="Nim" name="Nim" placeholder="Masukan NIM Anda" required>
22       <br><br>
23
24       <label for="email">Email :</label>
25       <input type="email" id="email" name="email" placeholder="Masukan Email Anda" required>
26       <br><br>
27
28       <label for="Program Studi">Program Studi :</label>
29       <input type="text" id="Program Studi" name="Program Studi" placeholder="Masukan Program Studi" required>
30       <br><br>
31
32       <input type="submit" value="Daftar">
33     </form>
34   </div>
35 </body>
36 </html>
```

Gambar 1. Code Halaman Pendaftaran

Kode HTML pada **Gambar 1.** diatas digunakan untuk membuat sebuah halaman web sederhana yang berisi form pendaftaran mahasiswa semester akhir. Pada bagian awal, elemen `<head>` berisi metadata seperti pengaturan karakter encoding menjadi UTF-8 untuk mendukung berbagai jenis karakter, serta tag viewport yang memungkinkan halaman beradaptasi dengan berbagai ukuran layar, terutama untuk tampilan mobile. Di dalamnya juga terdapat link ke file CSS eksternal (`style.css`) untuk mengatur tampilan halaman, dan skrip JavaScript (`data.js`) yang berfungsi untuk memproses data yang diisi dalam form.

Di dalam elemen `<body>`, terdapat container dengan kelas "container" yang membungkus semua elemen yang ada. Judul halaman, "Daftar Mahasiswa Semester Tua", ditampilkan dalam elemen `<h1>` di bagian atas halaman. Setelah itu, ada elemen `<form>` yang digunakan untuk mengumpulkan input dari pengguna, dengan beberapa field seperti nama, NIM, email, dan program studi. Setiap input dilengkapi dengan atribut `required`, yang memastikan bahwa pengguna wajib mengisi setiap field sebelum mengirimkan form. Placeholder dalam setiap field memberikan panduan singkat tentang informasi yang harus diisi oleh pengguna. Ketika form dikirim, fungsi JavaScript `saveData()` akan dipanggil untuk menyimpan data, dan setelah itu pengguna akan diarahkan ke halaman `information.html`. Tombol submit dengan label "Daftar" digunakan untuk mengirim data form.



Gambar 2. Halaman Pendaftaran

2. Halaman Informasi

```
1  <!DOCTYPE html>
2  <html lang="en">
3  <head>
4      <meta charset="UTF-8">
5      <meta name="viewport" content="width=device-width, initial-scale=1.0">
6      <title>Detail Mahasiswa Semester Tua</title>
7      <link rel="stylesheet" href="style.css">
8      <script src="data.js" ></script>
9
10 </head>
11 <body onload="loadData()">
12
13     <div class="container">
14         <h1>Detail Mahasiswa Semester Tua</h1>
15         <h2>Information Detail Data</h2>
16
17         <table>
18             <tr>
19                 <th colspan="2">Data </th>
20             </tr>
21             <tr>
22                 <td>Nama</td>
23                 <td id="Nama">Nama </td>
24             </tr>
25             <tr>
26                 <td>Nim</td>
27                 <td id="Nim">Nim</td>
28             </tr>
29             <tr>
30                 <td>Email</td>
31                 <td id="email">Email</td>
32             </tr>
33             <tr>
34                 <td>Program Studi</td>
35                 <td id="Program Studi">Progam Studi</td>
36             </tr>
37         </table>
38         <br>
39         <button>
40             <a href="home.html">Home</a>
41         </button>
42
43     </div>
44
45 </body>
46 </html>
```

Gambar 3. Code Halaman Information

Kode HTML pada **Gambar 3.** di atas merupakan halaman web yang menampilkan detail informasi mahasiswa semester akhir setelah data dimasukkan. Bagian pertama terdiri dari elemen

<head>, yang mencakup metadata seperti charset UTF-8, pengaturan responsif dengan tag viewport, serta judul halaman yang bertuliskan "Detail Mahasiswa Semester Tua". Selain itu, terdapat referensi ke file CSS eksternal (`style.css`) untuk mengatur tampilan dan file JavaScript (`data.js`) yang digunakan untuk memproses atau menampilkan data mahasiswa.

Pada elemen <body>, terdapat atribut `onload="loadData()"` yang berarti ketika halaman dimuat, fungsi `loadData()` dari file `data.js` akan dijalankan. Fungsi ini bertujuan untuk mengambil data yang sebelumnya dimasukkan oleh pengguna dan menampilkannya dalam tabel. Di dalam container, judul halaman ditampilkan dengan elemen <h1>, diikuti subjudul "Information Detail Data" dalam <h2>. Berikutnya, ada sebuah tabel yang digunakan untuk menyajikan data mahasiswa. Tabel ini memiliki beberapa baris, masing-masing berisi label seperti Nama, NIM, Email, dan Program Studi. Data mahasiswa akan dimasukkan ke dalam sel tabel dengan ID seperti `Nama`, `Nim`, `email`, dan `Program Studi`, yang akan diisi oleh JavaScript sesuai dengan data yang sebelumnya disimpan atau diambil. Setelah tabel, ada sebuah tombol yang berisi link ke halaman "home.html". Tombol ini mengarahkan pengguna kembali ke halaman beranda ketika diklik.



Gambar 4. Halaman Information

3. CSS

```
1 body {
2   font-family: Arial, sans-serif;
3   background-color: #F05A7E;
4   margin: 0px;
5   padding: 0px;
6   justify-content: center;
7   line-height: 1.6;
8 }
9 .container {
10  width: 60%;
11  max-width: 800px;
12  margin: 100px auto;
13  background-color: #1a8ecd;
14  padding: 20px;
15  box-shadow: 0 0 10px rgba(0, 0, 0, 0.1);
16  border-radius: 10px;
17  justify-content: center;
18  align-items: center;
19 }
20 h1 {
21   text-align: center;
22 }
23 h2 {
24   text-align: center;
25   font-style: bold;
26 }
27 form label {
28   display: inline-block;
29   width: 100px;
30   font-weight: bold;
31 }
32 form input[type="text"], form input[type="email"], form input[type="text"] {
33   width: calc(100% - 120px);
34   padding: 10px;
35   margin-bottom: 10px;
36   margin: 10px;
37   border-radius: 5px;
38   border: 1px solid #ccc;
39 }
```

```
40 form input[type="submit"] {
41   background-color: #4CAF50;
42   color: white;
43   padding: 10px 20px;
44   border: none;
45   border-radius: 5px;
46   cursor: pointer;
47   font-weight: bold;
48   text-align: left;
49 }
50 form input[type="submit"]:hover {
51   background-color: #45a049;
52 }
53 table {
54   width: 100%;
55   border-collapse: collapse;
56   margin-top: 20px;
57 }
58 table, th, td {
59   border: 1px solid #181818;
60 }
61 th, td {
62   padding: 10px;
63   text-align: left;
64 }
65 th {
66   background-color: #41B3A2;
67   border: #181818;
68   text-align: center;
69 }
70 a {
71   color: rgb(12, 11, 11);
72   font-style: bold;
73   text-decoration: none;
74 }
```

Gambar 5. Code CSS

Kode CSS pada **Gambar 5.** di atas bertujuan untuk memberikan gaya visual pada halaman web, sehingga tampilan menjadi lebih menarik dan terstruktur. Pada bagian pertama, aturan untuk elemen `<body>` diatur dengan menggunakan font **Arial** atau jenis font **sans-serif** sebagai alternatif. Latar belakang halaman diberi warna merah muda (`#F05A7E`), dan margin serta padding diatur menjadi nol untuk memastikan elemen-elemen menempel pada batas halaman. `justify-content: center` digunakan untuk memusatkan konten secara horizontal, sementara **line-height** diatur menjadi 1.6 untuk menjaga jarak antarbaris agar mudah dibaca. Bagian selanjutnya adalah aturan untuk kelas `.container`. **Container** ini diatur dengan lebar 60% dari halaman, maksimal 800 piksel, dan secara otomatis diposisikan di tengah halaman dengan margin atas 100 piksel. Latar belakang container berwarna biru (`#1a8ecd`), dengan padding sebesar 20 piksel untuk memberikan ruang di dalamnya. Efek bayangan diterapkan di sekitar container menggunakan **box-shadow**, serta sudut container dibulatkan dengan **border-radius** 10 piksel. Semua elemen dalam container diposisikan dan disejajarkan menggunakan properti **justify-content** dan **align-items**.

Judul halaman `<h1>` dan subjudul `<h2>` dipusatkan menggunakan **text-align: center**. Gaya tambahan pada `<h2>` termasuk font yang ditebalkan. Selanjutnya, untuk label dalam form (`form label`), label ditampilkan dalam format **inline-block** dengan lebar tetap 100 piksel dan teks yang

ditebalkan menggunakan **font-weight: bold**. Sedangkan untuk input dalam form, baik jenis **text** maupun **email**, lebar input diatur menggunakan **calc(100% - 120px)**, yang artinya lebar input adalah 100% dari lebar container dikurangi 120 piksel, menyesuaikan dengan labelnya. Setiap input juga diberikan padding 10 piksel, margin bawah 10 piksel, serta margin secara keseluruhan 10 piksel. Input dikelilingi oleh border abu-abu terang (#ccc) dan sudutnya dibulatkan dengan **border-radius** 5 piksel. Tombol submit diformat dengan warna latar belakang hijau (#4CAF50) dan teks berwarna putih. Tombol ini juga diberi padding, tanpa border, dan sudut bulat dengan **border-radius** 5 piksel. Efek interaktif ditambahkan dengan mengubah warna latar belakang menjadi lebih gelap (#45a049) ketika pengguna mengarahkan kursor di atas tombol tersebut. Untuk tabel, lebar diatur menjadi 100%, dan tabel serta elemen-elemen di dalamnya seperti baris, header, dan sel memiliki border yang solid berwarna hitam (#181818). Header tabel (<th>) diberi latar belakang hijau kebiruan (#41B3A2) dan teks dipusatkan. Seluruh tabel menggunakan **padding** 10 piksel agar isi tabel terlihat rapi.

4. Java Script

```
1  // Function to save form data to localStorage
2  function saveData() {
3      const nama = document.getElementById('nama').value;
4      const nim = document.getElementById('Nim').value;
5      const email = document.getElementById('email').value;
6      const programStudi = document.getElementById('Program Studi').value;
7
8      const data = {
9          nama: nama,
10         nim: nim,
11         email: email,
12         programStudi: programStudi
13     };
14
15     localStorage.setItem('mahasiswaData', JSON.stringify(data));
16 }
17
18 // Function to load and display data on the information page
19 function loadData() {
20     const data = JSON.parse(localStorage.getItem('mahasiswaData'));
21
22     if (data) {
23         document.getElementById('Nama').textContent = data.nama;
24         document.getElementById('Nim').textContent = data.nim;
25         document.getElementById('email').textContent = data.email;
26         document.getElementById('Program Studi').textContent = data.programStudi;
27     }
28 }
```

Gambar 6. Code Java Script

Kode JavaScript pada **Gambar 6.** di atas terdiri dari dua fungsi yang digunakan untuk menyimpan data form dan menampilkannya kembali dengan menggunakan **localStorage**. Fungsi pertama, **saveData()**, bertugas untuk menyimpan data yang dimasukkan oleh pengguna melalui form pendaftaran. Data diambil dari elemen-elemen input seperti nama, NIM, email, dan program studi menggunakan `document.getElementById` untuk masing-masing input. Setelah data diambil, sebuah objek `data` dibuat yang berisi informasi dari form tersebut. Objek ini kemudian diubah menjadi string dalam format JSON menggunakan metode `JSON.stringify()` dan disimpan ke **localStorage** dengan kunci `'mahasiswaData'`. **localStorage** memungkinkan data tersimpan secara persisten di browser, sehingga meskipun halaman di-refresh atau browser ditutup, data tetap ada dan bisa diakses kembali.

Fungsi kedua, **loadData()**, bertugas untuk menampilkan kembali data yang sudah disimpan ke **localStorage** ketika halaman lain dimuat. Saat halaman dibuka, fungsi ini memanggil data dari **localStorage** menggunakan `localStorage.getItem('mahasiswaData')` dan mengubahnya kembali menjadi objek JavaScript menggunakan `JSON.parse()`. Jika data berhasil ditemukan, fungsi ini akan menampilkan informasi tersebut di elemen-elemen HTML yang sesuai, seperti menampilkan nama pada elemen dengan ID `'Nama'`, NIM pada elemen `'Nim'`, email pada elemen `'email'`, dan program studi pada elemen `'Program Studi'`. Dengan cara ini, data yang telah dimasukkan pengguna sebelumnya bisa ditampilkan kembali di halaman informasi.