

# **LAPORAN PRAKTIKUM**

*Nama Tugas*



## **Disusun Oleh:**

**Nama** : Alyan fahrezi  
**Kelas** : XI RPL A  
**NIS** : XXXX/XXXXXX

**KONSENTRASI KEAHLIAN REKAYASA PERANGKAT**

**LUNAK SMK NEGERI 1 JENANGAN**

Jl. Niken Gandini No.98, Plampitan, Setono, Kec. Jenangan, Kabupaten Ponorogo,  
Jawa Timur 63492

## Langkah Praktikum

*(Tuliskan Langkah-langkah praktikum dengan jelas disertai dengan dokumentasi (screenshot) program)*

### 1. Pendahuluan

#### Deskripsi Tampilan Frontend

1. **Index (Daftar Siswa):** Tabel berisi daftar siswa dengan tombol aksi (Detail, Ubah, Hapus).
2. **Detail Siswa:** Menampilkan informasi lengkap siswa berdasarkan ID.
3. **Entry Siswa:** Form untuk menambahkan siswa baru (input ID, nama, jenis kelamin, kelas, tanggal, foto).
4. **Ubah Data Siswa:** Form untuk mengedit data siswa yang ada.
5. **Hapus Siswa:** Daftar siswa yang bisa dihapus dengan konfirmasi.

#### Teknologi Digunakan

- **HTML5 & CSS3:** Untuk struktur dan desain responsif.
- **JavaScript:** Untuk interaksi CRUD dengan server menggunakan fetch API.
- **PostgreSQL:** Basis data untuk menyimpan dan mengelola data siswa.
- **Node.js:** Backend untuk API yang menangani operasi CRUD.

### 2. Implementasi

Jelaskan bagaimana frontend CRUD dibuat, termasuk struktur proyek, library yang digunakan, dan kode utama. Sertakan cuplikan kode sebagai contoh.

#### A. Struktur Folder

```
B. sistem_informasi_siswa/
C. |— index.html           (Halaman utama untuk menampilkan daftar siswa)
D. |— detail_siswa.html   (Halaman untuk menampilkan detail siswa)
E. |— entry_siswa.html    (Halaman untuk menambah siswa baru)
F. |— hapus_siswa.html    (Halaman untuk menghapus siswa dari sistem)
G. |— ubah_siswa.html     (Halaman untuk mengubah data siswa yang ada)
H. |— css/                (Folder untuk file CSS)
I. |   |— styles.css      (File CSS untuk styling halaman)
J. |— js/                 (Folder untuk file JavaScript)
K. |   |— index.js        (File JavaScript untuk halaman index)
L. |   |— detail_siswa.js (File JavaScript untuk halaman detail siswa)
M. |   |— entry_siswa.js  (File JavaScript untuk halaman entry siswa)
N. |   |— hapus_siswa.js  (File JavaScript untuk halaman hapus siswa)
O. |   |— ubah_siswa.js   (File JavaScript untuk halaman ubah siswa)
P. |— img/                (Folder untuk menyimpan gambar foto siswa)
Q.
```

#### B. Code dan Penjelasan

## **1. Detail Siswa**

```

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 Siswa</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8   <link href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.0.0-beta3/css/all.min.css" rel="stylesheet">
9   <style>
10     body {
11       padding: 20px;
12       background-color: #f8d7da;
13       font-family: 'Arial', sans-serif;
14     }
15     .container {
16       background-color: white;
17       padding: 10px;
18       border-radius: 12px;
19       box-shadow: 0 4px 12px rgba(0, 0, 0, 0.1);
20       transition: transform 0.3s, box-shadow 0.3s;
21     }
22     .container:hover {
23       transform: translateY(-10px);
24       box-shadow: 0 4px 20px rgba(0, 0, 0, 0.2);
25     }
26     table th {
27       width: 30%;
28       font-weight: bold;
29       color: #333;
30       text-align: left;
31     }
32     table td {
33       color: #555;
34     }
35     .text-center img {
36       border-radius: 50%;
37       transition: transform 0.3s ease;
38     }
39     .text-center img:hover {
40       transform: scale(1.1);
41     }
42     .btn-primary {
43       background-color: #007bff;
44       transition: background-color 0.3s, box-shadow 0.3s;
45     }
46     .btn-primary:hover {
47       background-color: #0056b3;
48       box-shadow: 0 4px 8px rgba(0, 0, 0, 0.2);
49     }
50     a {
51       display: block;
52       text-align: center;
53       margin-top: 10px;
54     }
55     h2 {
56       color: #007bff;
57       font-weight: bold;
58       text-align: center;
59       margin-bottom: 20px;
60     }
61   </style>
62 </head>
63 <body>
64   <div class="container w-10 py-4 rounded-3 shadow">
65     <div class="text-center">
66       <img id="foto" src="" alt="Foto Siswa" width="130" height="130" class="shadow-sm">
67     </div>
68     <div>
69       <table class="table table-bordered">
70         <tr>
71           <th id="id_siswa" class="text-center">ID Siswa</th>
72           <td id="id_siswa"></td>
73         </tr>
74         <tr>
75           <th id="nama_lengkap" class="text-center">Nama Lengkap</th>
76           <td id="nama_lengkap"></td>
77         </tr>
78         <tr>
79           <th id="jenis_kelamin" class="text-center">Jenis Kelamin</th>
80           <td id="jenis_kelamin"></td>
81         </tr>
82         <tr>
83           <th id="tanggal_daftar" class="text-center">Tanggal Daftar</th>
84           <td id="tanggal_daftar"></td>
85         </tr>
86         <tr>
87           <th id="kelas" class="text-center">Kelas</th>
88           <td id="kelas"></td>
89         </tr>
90       </table>
91       <a href="#" id="btn" class="btn btn-primary w-50"><i class="fas fa-arrow-left"></i> Kembali</a>
92     </div>
93   </div>
94   <script>
95     document.addEventListener("DOMContentLoaded", function() {
96       const urlParams = new URLSearchParams(window.location.search);
97       const id_siswa = urlParams.get("id");
98
99       fetch(`/siswa/${id_siswa}/`)
100       .then(response => response.json())
101       .then(data => {
102         if (data) {
103           document.getElementById("foto").src = data[0].foto || 'default-avatar.png';
104           document.getElementById("id_siswa").textContent = data[0].id_siswa;
105           document.getElementById("nama_lengkap").textContent = data[0].nama_lengkap;
106           document.getElementById("jenis_kelamin").textContent = data[0].jenis_kelamin;
107           document.getElementById("tanggal_daftar").textContent = data[0].tanggal_daftar;
108           document.getElementById("kelas").textContent = data[0].kelas;
109         }
110       })
111     })
112   </script>
113 </body>
114 </html>
115

```

- **Fungsi Halaman:** Halaman ini digunakan untuk menampilkan informasi detail dari seorang siswa yang dipilih dari daftar utama. Halaman ini memberikan detail komprehensif untuk setiap siswa yang ada di dalam database.
- **Elemen yang Ditampilkan:**
  - **ID Siswa:** Nomor unik yang mengidentifikasi siswa.
  - **Nama Lengkap:** Nama lengkap siswa yang terdaftar.
  - **Jenis Kelamin:** Gender siswa, ditampilkan sebagai "Laki-laki" atau "Perempuan."
  - **Tanggal Daftar:** Tanggal kapan siswa mendaftar atau mulai terdaftar di sekolah.
  - **Kelas:** Kelas yang saat ini diikuti oleh siswa.
  - **Foto Siswa:** Gambar atau foto siswa yang ditampilkan di bagian atas halaman sebagai bentuk pengenalan visual.
- **Teknologi yang Digunakan:**
  - JavaScript digunakan untuk mengambil data detail dari server menggunakan metode `fetch` melalui API endpoint.
  - Data yang diterima ditampilkan pada halaman menggunakan DOM manipulation untuk memperbarui elemen HTML.
  - Jika data tidak ditemukan, halaman menampilkan pesan kesalahan yang relevan, seperti "Data siswa tidak ditemukan."
  - Fungsi tambahan termasuk kemampuan untuk memfilter data berdasarkan ID siswa yang dikirim melalui URL atau query string.

## 2. Entry Siswa

```

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>Entry Siswa</title>
7   <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/css/bootstrap.min.css" rel="stylesheet">
8   <style>
9     body { padding: 20px; margin: 40px; }
10  </style>
11 </head>
12 <body style="background-color: rgb(59, 93, 184);">
13   <div class="container w-50 py-2 rounded-3 shadow" style="background-color: rgb(255, 255, 255);">
14     <table id="example" class="table table-striped" style="width:100%">
15       <tr class="bg-3" align="center"><th colspan="2">Tambah Siswa</th>
16     </tr>
17     <tr id="formCreate">
18       <div class="mb-3">
19         <label class="form-label">Upload Foto</label>
20         <input type="file" class="form-control" name="foto" id="foto">
21       </div>
22       <div class="mb-3">
23         <label class="form-label">ID Siswa</label>
24         <input type="text" class="form-control" name="id_siswa" id="id_siswa" placeholder="Masukan ID">
25       </div>
26       <div class="mb-3">
27         <label class="form-label">Nama Lengkap</label>
28         <input type="text" class="form-control" name="nama_lengkap" id="nama_lengkap" placeholder="Masukan Nama Lengkap">
29       </div>
30       <div class="mb-3">
31         <label class="form-label">Jenis Kelamin</label>
32         <select class="form-select" name="jenis_kelamin" id="jenis_kelamin" aria-label="Default select example">
33           <option selected disabled>Pilih Jenis Kelamin</option>
34           <option value="laki-laki">Laki-laki</option>
35           <option value="Perempuan">Perempuan</option>
36         </select>
37       </div>
38       <div class="mb-3">
39         <label class="form-label">Tanggal Daftar</label>
40         <input type="date" class="form-control" name="tanggal_daftar" id="tanggal_daftar" placeholder="Masukan Tanggal Daftar">
41       </div>
42       <div class="mb-3">
43         <label class="form-label">Kelas</label>
44         <input type="text" class="form-control" name="kelas" id="kelas" placeholder="Masukan Kelas">
45       </div>
46       <button type="submit" class="btn btn-primary mb-3">Tambah</button>
47     </tr>
48   </table>
49 </div>
50 </div>
51 <script>
52   document.getElementById("formCreate").addEventListener("submit", function(event) {
53     event.preventDefault();
54     const formData = new FormData();
55     formData.append("foto", document.getElementById("foto").files[0]);
56     formData.append("id_siswa", document.getElementById("id_siswa").value);
57     formData.append("nama_lengkap", document.getElementById("nama_lengkap").value);
58     formData.append("jenis_kelamin", document.getElementById("jenis_kelamin").value);
59     formData.append("tanggal_daftar", document.getElementById("tanggal_daftar").value);
60     formData.append("kelas", document.getElementById("kelas").value);
61
62     fetch("http://localhost:3002/siswa", {
63       method: "POST",
64       body: formData
65     })
66     .then(response => response.text())
67     .then(result => {
68       localStorage.setItem("successMessage", "Data berhasil Disimpan")
69       window.location.href = "index.html";
70     });
71   });
72 </script>
73 </body>
74 </html>

```

- **Fungsi Halaman:** Halaman ini digunakan untuk menambahkan data siswa baru ke dalam sistem. Admin atau pengguna dapat memasukkan informasi siswa baru melalui formulir yang telah disediakan.
- **Formulir Input:**
  - **ID Siswa:** Input teks untuk memasukkan nomor ID unik siswa.
  - **Nama Lengkap:** Input teks untuk nama lengkap siswa.
  - **Jenis Kelamin:** Input pilihan untuk jenis kelamin siswa, dengan opsi “Laki-laki” atau “Perempuan.”
  - **Tanggal Daftar:** Input tanggal untuk memilih tanggal siswa mendaftar.
  - **Kelas:** Input dropdown atau pilihan untuk memilih kelas yang diikuti oleh siswa.

- **Foto:** Input untuk mengunggah foto siswa (dalam format JPEG/PNG).
- **Validasi Data:**
  - Semua input wajib diisi sebelum data dapat dikirim.
  - Validasi dilakukan pada sisi klien menggunakan JavaScript, seperti memastikan foto yang diunggah adalah format yang didukung, dan tanggal tidak boleh lebih awal dari tahun tertentu.
- **Pengiriman Data:**
  - Setelah formulir diisi, data dikirim ke server menggunakan metode `POST` melalui JavaScript.
  - Pengguna akan menerima pesan keberhasilan atau kesalahan tergantung pada hasil penyimpanan data.
  - Setelah data berhasil disimpan, pengguna dapat diarahkan kembali ke halaman utama atau halaman detail siswa baru tersebut.

### 3. Hapus Siswa



- **Fungsi Halaman:** Halaman ini menampilkan daftar siswa yang sudah terdaftar di dalam sistem dengan opsi untuk menghapus data siswa yang dipilih.
- **Daftar Siswa:**
  - Setiap siswa ditampilkan dalam bentuk tabel yang berisi kolom ID Siswa, Nama Lengkap, Jenis Kelamin, Kelas, dan Tanggal Daftar.
  - Setiap baris data memiliki tombol "Hapus" di kolom terakhir.
- **Fungsi Penghapusan:**
  - Tombol "Hapus" yang ditekan akan mengirimkan permintaan `DELETE` ke server dengan ID siswa terkait.
  - Pengguna akan diminta konfirmasi sebelum penghapusan dilakukan untuk menghindari penghapusan data secara tidak sengaja.
  - Setelah konfirmasi, JavaScript digunakan untuk memanggil API server dan menghapus data dari database.
  - Jika penghapusan berhasil, siswa tersebut akan dihapus dari tampilan tabel secara otomatis tanpa perlu me-refresh halaman.
  - Pesan sukses atau gagal penghapusan akan ditampilkan kepada pengguna.

#### **4. Ubah Data Siswa**





- **Fungsi Halaman:** Halaman ini digunakan untuk memperbarui atau mengubah data siswa yang sudah ada di dalam sistem.
- **Formulir Input:**
  - Sama seperti formulir untuk entry siswa, namun data siswa yang sudah ada akan otomatis terisi dalam formulir.
  - Pengguna dapat mengubah Nama Lengkap, Jenis Kelamin, Tanggal Daftar, Kelas, atau Foto siswa.
  - ID siswa tidak dapat diubah karena merupakan kunci unik dalam database.
- **Validasi dan Pengiriman Data:**
  - Formulir validasi memastikan bahwa semua perubahan yang dilakukan adalah sah, termasuk validasi ulang untuk foto dan tanggal.
  - Setelah perubahan selesai, data akan dikirim menggunakan metode PUT untuk memperbarui data di server.
  - JavaScript menangani pengiriman data dan memastikan perubahan langsung diterapkan pada database serta halaman detail siswa yang bersangkutan diperbarui secara otomatis.
  - Pengguna dapat diberikan pesan sukses setelah data berhasil diubah.

## 5. Index Data Siswa

```

<script>
    $(document).ready(function() {
        // Fetch data from the database
        $.ajax({
            url: 'http://localhost:3000/api/siswa',
            type: 'GET',
            success: function(data) {
                // Display data in a table
                $('#table').empty();
                $.each(data, function(index, siswa) {
                    $('#table').append(
                        <tr>
                            <td>{siswa.id}</td>
                            <td>{siswa.nama}</td>
                            <td>{siswa.jenis_kelamin}</td>
                            <td>{siswa.tanggal_daftar}</td>
                            <td>{siswa.kelas}</td>
                            <td>{siswa.foto}</td>
                        </tr>
                    );
                });
            },
            error: function(xhr) {
                console.log('Error fetching data: ' + xhr.responseText);
            }
        });

        // Fetch data for a specific student (for update form)
        $('#id').on('change', function() {
            var id = $(this).val();
            $.ajax({
                url: 'http://localhost:3000/api/siswa/' + id,
                type: 'GET',
                success: function(data) {
                    // Fill the form with the fetched data
                    $('#nama').val(data.nama);
                    $('#jenis_kelamin').val(data.jenis_kelamin);
                    $('#tanggal_daftar').val(data.tanggal_daftar);
                    $('#kelas').val(data.kelas);
                    $('#foto').val(data.foto);
                },
                error: function(xhr) {
                    console.log('Error fetching student data: ' + xhr.responseText);
                }
            });
        });

        // Update student data
        $('#update').on('click', function() {
            var id = $('#id').val();
            var data = {
                nama: $('#nama').val(),
                jenis_kelamin: $('#jenis_kelamin').val(),
                tanggal_daftar: $('#tanggal_daftar').val(),
                kelas: $('#kelas').val(),
                foto: $('#foto').val()
            };

            $.ajax({
                url: 'http://localhost:3000/api/siswa/' + id,
                type: 'PUT',
                data: JSON.stringify(data),
                success: function() {
                    // Show success message
                    $('#message').text('Data updated successfully!');
                },
                error: function(xhr) {
                    console.log('Error updating data: ' + xhr.responseText);
                }
            });
        });
    });

```

- **Fungsi Halaman:** Halaman ini merupakan halaman utama atau dashboard yang menampilkan seluruh data siswa dalam bentuk tabel.
- **Tabel Siswa:**
  - Tabel ini menampilkan daftar lengkap siswa dengan kolom-kolom berikut: ID Siswa, Nama Lengkap, Jenis Kelamin, Kelas, dan Tanggal Daftar.
  - Tombol aksi terdapat di setiap baris, termasuk tombol untuk:
    - **Lihat Detail:** Mengarahkan pengguna ke halaman detail siswa.
    - **Ubah:** Membuka halaman perubahan data siswa.
    - **Hapus:** Menghapus data siswa dengan konfirmasi terlebih dahulu.
- **Penggunaan DataTable:**
  - Untuk kemudahan pencarian, filter, dan pengurutan, tabel ini menggunakan plugin DataTable di JavaScript.
  -

## 6. Index.js

```

1  const express = require('express');
2  const bodyParser = require('body-parser');
3  const multer = require('multer');
4  const cors = require('cors');
5  const client = require('./connection');
6  const app = express();
7  const port = 3000;
8  const path = require('path');
9  const routes = require('./routes');
10
11  app.use(express.static(path.join(__dirname, 'view')));
12  app.use("/public", express.static('public'));
13  app.use('/', routes);
14  app.use(cors());
15  app.use(bodyParser.json());
16
17  app.listen(port, () => {
18    console.log('listening to port ' + port);
19  });
20
21  client.connect(err => {
22    if (err) {
23      console.log(err.message);
24    } else {
25      console.log('berhasil connect');
26    }
27  });
28
29  const storage = multer.diskStorage({
30    destination: 'public',
31    filename: (req, file, cb) => cb(null, file.originalname),
32  });
33  const upload = multer({ storage });
34
35  app.get('/siswa', (req, res) => {
36    client.query('SELECT * FROM siswa', (err, result) => {
37      if (!err) {
38        res.send(result.rows);
39      }
40    });
41  });
42
43  app.get('/siswa/:id', (req, res) => {
44    const id = parseInt(req.params.id);
45    client.query('SELECT * FROM siswa WHERE id = $1', [id], (err, result) => {
46      if (!err) {
47        res.send(result.rows);
48      }
49    });
50  });
51
52  app.post('/siswa', upload.single('foto'), (req, res) => {
53    const { id_siswa, nama_lengkap, jenis_kelamin, tanggal_daftar, kelas } = req.body;
54    const imagePath = req.file ? '/public/' + req.file.filename : null;
55
56    const query = 'INSERT INTO siswa (foto, id_siswa, nama_lengkap, jenis_kelamin, tanggal_daftar, kelas) VALUES ($1, $2, $3, $4, $5, $6)';
57
58    client.query(query, [imagePath, id_siswa, nama_lengkap, jenis_kelamin, tanggal_daftar, kelas], (err, result) => {
59      if (err) {
60        console.error('Error saat menambahkan data:', err.message);
61        res.status(500).send({ error: err.message });
62      } else {
63        res.send({ message: 'Data berhasil ditambahkan', data: req.body });
64      }
65    });
66  });
67
68  app.put('/siswa/:id', upload.single('foto'), (req, res) => {
69    const id = parseInt(req.params.id);
70    const { nama_lengkap, jenis_kelamin, kelas } = req.body;
71    const imagePath = req.file ? '/public/' + req.file.filename : null;
72
73    const query = 'UPDATE siswa SET nama_lengkap = $1, jenis_kelamin = $2, kelas = $3, foto = $4 WHERE id = $5';
74
75    client.query(query, [nama_lengkap, jenis_kelamin, kelas, imagePath, id], (err, result) => {
76      if (!err) {
77        res.send('Update Success');
78      } else {
79        console.error('Database Error:', err.message);
80        res.status(500).send('Database Error: ' + err.message);
81      }
82    });
83  });
84
85  app.delete('/siswa/:id', (req, res) => {
86    const id = parseInt(req.params.id);
87    client.query('DELETE FROM siswa WHERE id = $1', [id], (err, result) => {
88      if (!err) {
89        res.send('Delete Success');
90      } else {
91        console.error('Database Error:', err.message);
92        res.status(500).send('Database Error: ' + err.message);
93      }
94    });
95  });
96
97
98
99
100
101

```

- **Express dan middleware:** Aplikasi menggunakan **Express** untuk membuat server dan beberapa middleware seperti `express.static`, `body-parser` untuk parsing JSON, dan **CORS** untuk mengizinkan permintaan lintas domain.
- **Multer:** Digunakan untuk menangani upload file, khususnya foto siswa, dan menyimpan file di direktori `public`.
- **Routes:** Terdapat beberapa endpoint yang terkait dengan pengelolaan data siswa di database PostgreSQL:
  - **GET /siswa:** Mengambil semua data siswa.
  - **GET /siswa/:id:** Mengambil data siswa berdasarkan ID.
  - **POST /siswa:** Menambahkan data siswa baru, termasuk mengupload foto.
  - **PUT /siswa/:id:** Mengupdate data siswa berdasarkan ID.
  - **DELETE /siswa/:id:** Menghapus data siswa berdasarkan ID.
- **PostgreSQL:** Koneksi ke database diatur melalui `client.connect` untuk mengakses tabel **siswa** dan menjalankan query untuk operasi CRUD (Create, Read, Update, Delete).

### 3. Hasil Tampilan detail



#### Entry siswa



#### After



Hapus

After



ubah

A screenshot of the 'Ubah' (Edit) form for a student record. The form has a light blue background and contains the following fields: 'Upload Foto' with a 'Choose File' button and 'K0902.jpg' text; 'ID Siswa' with the value '5000'; 'Nama Lengkap' with the value 'Kobon Chan'; 'Jenis Kelamin' with the value 'Perempuan'; 'Tanggal Daftar' with the value '2023-01-22'; and 'Kelas' with the value 'X RPL'. There is a 'Simpan' (Save) button at the bottom right.

After



Index halaman utama



#### 4. Kesimpulan

Tanpa index.js maka semua program tidak dapat berjalan

#### 5. Kesulitan Yang Dihadapi

1. Sulit jika ada sedikit typo untuk sulit di cari
2. Susah menemukan masalah di file index.js
3. Terlalu singkat masa belajar pendapat saya pribadi