

Laporan Praktikum
Mata Kuliah Pemrograman Web dan Pemrograman Berorientasi Objek



Mini Project
**“CRUD, REST API, WebSocket, Manajemen pengguna, Deployment,
Hosting”**

Dosen Pengampu :
Willdan Aprizal Arifin, S.Pd., M.Kom.

Disusun Oleh :
1. Roma Angelita Tambunan (2307955)
2. Nur'aini Dwi Anra (2301148)

PROGRAM STUDI SISTEM INFORMASI KELAUTAN
UNIVERSITAS PENDIDIKAN INDONESIA
2024

Penjelasan Teknis Aplikasi :

Aplikasi Sistem Informasi Kelautan ini dibuat untuk membantu pengelolaan data kapal nelayan di sebuah pelabuhan dengan fitur-fitur utama seperti CRUD untuk data kapal, autentikasi pengguna menggunakan JWT, notifikasi real-time melalui WebSocket, dan manajemen pengguna dengan peran (admin dan user).

Berikut adalah penjelasan detail tentang bagaimana aplikasi ini bekerja dan fitur-fitur yang telah diterapkan:

1. Desain Database

Aplikasi ini menggunakan MySQL sebagai database untuk menyimpan informasi terkait kapal nelayan dan pengguna aplikasi. Ada dua tabel utama dalam database:

- **Tabel Kapal:** Menyimpan data tentang kapal nelayan, termasuk ID kapal, nama kapal, jenis kapal, kapasitas muatan, dan waktu kapal terdaftar.
- **Tabel Pengguna (Users):** Menyimpan informasi pengguna aplikasi, termasuk username, password yang dienkripsi, dan role pengguna (admin atau user).

2. Fitur CRUD untuk Data Kapal

Aplikasi ini mendukung operasi CRUD (Create, Read, Update, Delete) untuk mengelola data kapal. Berikut adalah cara setiap fitur bekerja:

- **Create:** Admin dapat menambahkan data kapal baru melalui endpoint POST /kapal. Data kapal yang dimasukkan mencakup nama kapal, jenis kapal, dan kapasitas muatan.
- **Read:** Semua pengguna dapat melihat daftar kapal yang terdaftar di aplikasi melalui endpoint GET /kapal. Pengguna juga dapat melihat detail kapal tertentu dengan menggunakan endpoint GET /kapal/:id.
- **Update:** Hanya admin yang dapat memperbarui data kapal melalui endpoint PUT /kapal/:id. Admin bisa mengubah informasi kapal yang sudah ada.
- **Delete:** Hanya admin yang bisa menghapus data kapal melalui endpoint DELETE /kapal/:id.

Setiap data yang dimasukkan atau diubah di aplikasi akan divalidasi terlebih dahulu, seperti memastikan bahwa nama kapal tidak boleh kosong dan kapasitas muatan harus berupa angka positif.

3. Autentikasi Pengguna dengan JWT

Aplikasi ini menggunakan JWT (JSON Web Token) untuk autentikasi pengguna. Pengguna yang ingin mengakses fitur CRUD harus login terlebih dahulu untuk mendapatkan token JWT yang akan digunakan di setiap request.

- Register: Admin dapat menambahkan pengguna baru (role admin atau user) melalui endpoint POST /auth/register.
- Login: Pengguna dapat login melalui endpoint POST /auth/login. Jika login berhasil, sistem akan memberikan token JWT yang harus disertakan pada setiap request untuk mengakses endpoint CRUD.

Setelah login, token JWT yang didapatkan digunakan untuk memverifikasi identitas pengguna pada setiap request. Hanya pengguna dengan role admin yang dapat melakukan operasi Create, Update, dan Delete, sementara pengguna dengan role user hanya bisa melihat data kapal.

4. WebSocket untuk Notifikasi Real-Time

Untuk meningkatkan interaktivitas aplikasi, saya menambahkan fitur WebSocket yang memungkinkan pengiriman notifikasi real-time kepada semua pengguna yang terhubung, setiap kali terjadi perubahan data kapal (Create, Update, Delete). Setiap kali ada perubahan data kapal, notifikasi dengan format JSON akan dikirim ke semua client yang terhubung, seperti:

```
json
{
  "event": "data_changed",
  "message": "Data kapal telah diperbarui.",
  "data": [
    {
      "id_kapal": 1,
      "nama_kapal": "Laut Biru",
      "jenis_kapal": "Penangkap Ikan",
      "kapasitas_muatan": 5000,
      "waktu_terdaftar": "2024-12-17T00:00:00Z"
    }
  ]
}
```

}

Dengan fitur ini, pengguna bisa mendapatkan pembaruan secara langsung tanpa perlu melakukan refresh halaman.

5. Hosting Aplikasi

Aplikasi ini telah di-deploy ke layanan hosting gratis, seperti Heroku. Aplikasi dapat diakses secara publik melalui URL yang telah disediakan. Proses deployment mencakup:

- Mengatur server Node.js di platform hosting.
- Menghubungkan aplikasi dengan database MySQL di cloud.
- Memastikan aplikasi berjalan dengan lancar dan dapat diakses oleh pengguna.

6. Dokumentasi API

Untuk memudahkan pengguna lain dalam mengakses dan menguji API, saya telah menyediakan dokumentasi API yang bisa digunakan dengan Postman. Dokumentasi ini mencakup semua endpoint yang ada dalam aplikasi, termasuk cara menggunakan autentikasi JWT dan cara mengakses setiap fitur CRUD. Dokumentasi ini akan memudahkan pengguna untuk menguji setiap bagian dari API.

7. Keamanan

Aplikasi ini mengutamakan keamanan data pengguna dengan beberapa cara:

- Password dienkripsi menggunakan bcrypt sebelum disimpan di database untuk menghindari penyimpanan kata sandi dalam bentuk teks asli.
- JWT diterapkan untuk memastikan hanya pengguna yang terautentikasi yang dapat mengakses endpoint yang membutuhkan otorisasi.
- Validasi data dilakukan pada setiap request untuk memastikan hanya data yang valid yang dapat disimpan atau diperbarui.

URL Hosting :

<https://mini-projek-production.up.railway.app/>

Dokumentasi API :

[Dokumentasi REST API Pengelolaan Kapal](#)

Link Video YouTube :

<https://youtu.be/d5ORxZaTC-I>