

**Tugas 2 : Mengembangkan Layanan *Microservices***  
**II3160 - Teknologi Sistem Terintegrasi**



Disusun oleh :  
Muhammad Kevinza Faiz / 18222072

Rekan :  
Taufiq Ramadhan Ahmad / 18222060

**Program Studi Sistem dan Teknologi Informasi**  
**Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung**  
**Jl. Ganesha 10, Bandung 40132**

## Latar Belakang

Proyek ini bertujuan untuk membuat sistem pelaporan masalah untuk penghuni kos. Sistem ini terdiri dari backend berbasis API yang memungkinkan penghuni kos melaporkan masalah, melihat status laporan, dan pemilik kos mengelola laporan tersebut. Aplikasi ini dibuat menggunakan framework CodeIgniter 4 dengan XAMPP.

## Fitur Utama

Sistem ini memiliki beberapa fitur utama sebagai berikut:

- 1. Register**  
Penghuni kos harus mendaftarkan username dan password melalui fitur ini untuk dapat mengakses sistem lebih lanjut.
- 2. Login**  
Penghuni kos harus memasukkan username dan password yang sudah terdaftar dan valid melalui fitur ini untuk dapat mengakses sistem.
- 3. Pembuatan Laporan (Create Report)**  
Penghuni kos dapat membuat laporan masalah dengan informasi seperti jenis masalah, deskripsi, lokasi kamar, dan mengunggah foto terkait.
- 4. Melihat Semua Laporan (View All Reports)**  
Pemilik kos dapat melihat semua laporan yang telah dibuat oleh penghuni kos.
- 5. Memperbarui Status Laporan (Update Report Status)**  
Pemilik kos dapat memperbarui status laporan (pending, in\_process, completed).
- 6. Filter Laporan Berdasarkan Status**  
Laporan dapat di-*filter* berdasarkan statusnya.
- 7. Statistik Laporan**  
Pemilik kos dapat melihat statistik laporan berdasarkan status.
- 8. Hapus Laporan (Delete Report)**  
Sebuah laporan dapat dihapus dengan tujuan tertentu, seperti laporan dibatalkan atau laporan sudah ditangani.

## Struktur API

Link Web : <https://cornflowerblue-wolverine-266402.hostingersite.com/>

Berikut adalah *Endpoint* yang tersedia pada proyek ini :

### AuthController

#### 1. Register

| Register        |  |
|-----------------|--|
| Method          | POST   |
| Endpoint        | /auth/register   |
| Deskripsi       | Memasukkan data username dan password ke dalam database users              |
| Input           | <pre>{<br/>  "username": "String",<br/>  "password": "String",<br/>}</pre> |
| Respon Berhasil | <pre>{<br/>  "message": "User registered successfully."<br/>}</pre>        |
| Respon Gagal    | <pre>{<br/>  "message": "Failed to register user."<br/>}</pre>             |

#### 2. Login

| Register  |   |
|-----------|---|
| Method    | POST  |
| Endpoint  | /auth/login   |
| Deskripsi | Memvalidasi username dan password bagi user yang ingin mengakses method pada "ReportController" |
| Input     | <pre>{<br/>  "username": "String",<br/>  "password": "String",<br/>}</pre>                      |
| Respon    | <pre>{</pre>  |

|                 |   |
|-----------------|---|
| Berhasil        | <pre>"message": "Login successful." }</pre>                 |
| Respon<br>Gagal | <pre>{   "message": "Invalid username or password." }</pre> |

## ReportController

### 1. Create Report

| Create Report      |  |
|--------------------|--|
| Method             | POST   |
| Endpoint           | /report/create   |
| Deskripsi          | Digunakan untuk membuat sebuah laporan baru terkait permasalahan di kos-kosan  |
| Input              | <pre>{   "problem_type": "String",   "description": "String",   "room_location": "String",   "photo": "File" }</pre> |
| Respon<br>Berhasil | <pre>{   "message": "Report berhasil disubmit." }</pre>  |
| Respon<br>Gagal    | <pre>{   "message": "Gagal dalam membuat report." }</pre>  |

### 2. Get All Reports

| Get All Reports |  |
|-----------------|--|
| Method          | GET  |
| Endpoint        | /report  |
| Deskripsi       | Mengambil seluruh laporan yang ada pada database |
| Output          | <pre>[   {</pre>                                 |

|  |  |
|--|--|
|  | <pre>         "id": "4",         "problem_type": "Lantai retak",         "description": "Ada lantai retak",         "room_location": "1",         "photo":         "uploads/1735379460_85ba411bef93b6f6087f.jpg",         "status": "in_process",         "created_at": "2024-12-28 09:51:00",         "updated_at": "2024-12-28 10:00:04",         "photo_url":         "http://localhost:8080/uploads/1735379460_85ba411bef93b6         f6087f.jpg"       }     ] </pre> |
|--|--|

### 3. Update Report Status

| Update Report Status |  |
|----------------------|--|
| Method               | PUT  |
| Endpoint             | /report/updateStatus/{id}  |
| Deskripsi            | Memperbarui status dari sebuah laporan dengan status yang valid, yaitu : 'pending', 'in_process', dan 'completed'. |
| Input                | <pre> {   "status": "pending in_process completed" } </pre>  |
| Respon Berhasil      | <pre> {   "message": "Report status updated." } </pre>   |
| Respon Gagal         | <pre> {   "status": 404,   "error": "Report not found." } </pre>   |

### 4. Filter Report by Status

| Filter Report by Status |                         |
|-------------------------|-------------------------|
| Method                  | GET                     |
| Endpoint                | /report/status/{status} |

|           |   |
|-----------|---|
| Deskripsi | Mendapatkan laporan berdasarkan status tertentu.  |
| Output    | <pre>[   {     "id": "4",     "problem_type": "Lantai retak",     "description": "Ada lantai retak",     "room_location": "1",     "photo":       "uploads/1735379460_85ba411bef93b6f6087f.jpg",     "status": "in_process",     "created_at": "2024-12-28 09:51:00",     "updated_at": "2024-12-28 10:00:04",     "photo_url":       "http://localhost:8080/uploads/1735379460_85ba411bef93b6f6087f.jpg"   } ]</pre> |

## 5. Get Statistics

| Get Statistics |  |
|----------------|--|
| Method         | GET  |
| Endpoint       | /report/stats  |
| Deskripsi      | Mendapatkan rangkuman jumlah laporan untuk setiap status.                                |
| Output         | <pre>{   "total_reports": 1,   "pending": 0,   "in_process": 0,   "completed": 1 }</pre> |

## 6. Delete Report

| Delete Report |  |
|---------------|--|
| Method        | DELETE   |
| Endpoint      | /report/delete/{id}                                      |
| Deskripsi     | Menghapus sebuah report laporan berdasarkan id tertentu. |
| Respon        | {  |

|          |  |
|----------|--|
| Berhasil | "message": "Report deleted successfully."<br>} |
|----------|--|

## Struktur Database

### - Tabel users

Tabel ini digunakan untuk menyimpan informasi yang digunakan user untuk melakukan autentikasi. Tabel ini berisi id, username, password, dan waktu memasukkan data tersebut.

| Column     | Tipe Data         | Deskripsi  |
|------------|-------------------|--|
| id         | INT (Primary Key) | Identifikasi unik untuk setiap data pada <i>database</i> |
| username   | VARCHAR(255)      | Username yang digunakan user untuk melakukan login       |
| password   | VARCHAR(255)      | Password yang digunakan user untuk melakukan login       |
| created_at | timestamp         | Waktu ketika sebuah akun dibuat                          |
| updated_at | timestamp         | Waktu ketika sebuah akun diperbarui                      |

### - Tabel reports

Tabel ini digunakan untuk menyimpan informasi laporan yang dibuat oleh penghuni kos. Tabel ini berisi data penting seperti jenis masalah, deskripsi, lokasi kamar, foto, status laporan, serta waktu pembuatan dan pembaruan laporan.

| Column        | Tipe Data            | Deskripsi  |
|---------------|----------------------|--|
| id            | INT (Primary Key)    | Identifikasi unik untuk setiap data pada <i>database</i> |
| user_id       | INT (FK to users.id) | Foreign key yang merujuk ke kolom id pada table users    |
| problem_type  | VARCHAR (255)        | Jenis masalah yang terjadi di kos-kosan                  |
| description   | text                 | Deskripsi lebih lanjut dari permasalahan yang terjadi    |
| room_location | VARCHAR (255)        | Nomor kamar kos yang memiliki masalah                    |

|            |   |   |
|------------|---|---|
| photo      | VARCHAR (255)                                     | Path foto sebagai bukti laporan   |
| status     | Enum<br>(‘pending’, ‘in_process’,<br>‘completed’) | Status laporan yang terdiri dari <i>pending</i> ,<br><i>in_process</i> , <i>completed</i> |
| created_at | datetime  | Waktu laporan dibuat  |
| updated_at | datetime  | Waktu laporan diperbarui  |

## Implementasi

Dalam pengimplementasiannya, saya membuat sebuah *controller* bernama “ReportController” yang digunakan untuk mengimplementasikan berbagai *method* yang digunakan. Berikut adalah penjelasan lebih lanjut terkait *method* pada “ReportController”.

- **create method**  
*Method* ini digunakan untuk menerima *input* sebuah laporan terkait permasalahan yang dialami penghuni kos. *Input* dari *user* akan masuk ke dalam *database*, yaitu pada tabel *reports*.
- **index method**  
*Method* ini digunakan dalam mengambil seluruh data pada laporan dari *database*. Selain itu, akan ditambahkan juga URL dari foto untuk setiap laporan.
- **updateStatus method**  
*Method* ini digunakan untuk melakukan *update* pada status sebuah laporan berdasarkan ID laporan. Laporan yang bisa di-*update* hanya id laporan yang valid dan dengan nilai (‘pending’, ‘in\_process’, ‘completed’). Status tersebut juga akan ter-*update* ke dalam *database*.
- **getByStatus method**  
*Method* ini digunakan untuk mendapatkan informasi laporan dengan cara memfilter berdasarkan status tertentu.
- **stats method**  
*Method* ini digunakan untuk menampilkan statistik laporan secara keseluruhan, seperti total laporan keseluruhan dan total laporan untuk setiap status laporan.
- **deleteReport method**  
*Method* ini digunakan untuk menghapus laporan tertentu dari *database*.

Selain itu, saya juga membuat “AuthController” yang digunakan untuk mengimplementasikan *method* pada autentikasi. Berikut adalah penjelasan lebih lanjut terkait *method* pada “AuthController”.



- **register *method***

*Method* ini digunakan untuk memasukkan data username dan password ke dalam database users.

- **login *method***

*Method* ini digunakan untuk memvalidasi username dan password bagi user yang ingin mengakses *method* pada “ReportController”.

Selain *controller*, saya juga menerapkan Models bernama “ReportModel” untuk mengatur nama tabel, kolom yang diizinkan, dan penggunaan timestamp otomatis. Dilakukan juga *routing* dalam mengatur *endpoint* untuk setiap API pada file Routes.php.

Untuk *front end*, saya membuat sebuah *user interface* yang digunakan untuk menguji seluruh *endpoint* yang dibangun sehingga dapat lebih realistis. Untuk *deployment*, saya menggunakan aiven untuk database dan hostinger untuk deploy web keseluruhan.