**Dokumen Teknis**

**Pengembangan Aplikasi E-Ticketing System Bus Danau Toba**

**Menggunakan Arsitektur Microservice**

**Tugas Proyek Akhir Semester**

**Mata Kuliah: Pengembangan Aplikasi Terdistribusi**

**Dipersiapkan Oleh:**

| 11422018 | Andien Laura T. Panjaitan |
| --- | --- |
| 11422037 | Roberto Samuel ButarButar |
| 11422042 | Vivaldi A. Simangunsong |
| 11422045 | Syahrial Jeremia Sinaga |
| 11422048 | Gladys Eunice Barbara Sitinjak |

**Untuk:**

**Institut Teknologi Del**

**2024**

# **DAFTAR ISI**

[**DAFTAR ISI 2**](#_kbvitc8e0gpn)

[**1 Pendahuluan 4**](#_qatb084lsr92)

[1.1 Deskripsi Umum Aplikasi 4](#_1oofk66x9mfz)

[1.2 Karakteristik Pengguna Aplikasi 4](#_76tegyln0ri1)

[1.3 Fungsi pada Aplikasi 6](#_j7f1b4uzcyhd)

[**2 Arsitektur Monolith dan Microservice 7**](#_4i3fiuwlhk6p)

[**3 Arsitektur Layanan Mikro 9**](#_prm4132uwb1g)

[3.1 Basis Data 9](#_tqkmzmllgrx7)

[3.2 Server 10](#_e36mq8qzuc5d)

[3.3 Model Tarik dan Dorong (Push and Pull Model) 10](#_wdwsu67f4bs1)

[3.4 Antarmuka Pengguna 10](#_cf312d1r5l2)

# **Pendahuluan**

Pada bagian ini menjelaskan deskripsi umum aplikasi, karakteristik pengguna, dan fungsi yang ada dalam aplikasi.

## **Deskripsi Umum Aplikasi**

Proyek pengembangan aplikasi web ini bertujuan untuk menyediakan sistem pemesanan tiket bus di sekitar Danau Toba yang dapat diakses oleh berbagai peran, termasuk Supir Admin, Admin Kantor, Admin Loket, Supir, dan penumpang. Dalam aplikasi ini, Supir Admin bertanggung jawab mengelola merek bus yang akan tergabung dalam sistem, sementara Admin Kantor mengurus jadwal bus, keuangan, komisi, loket, armada, rute, dan data supir. Admin Loket bertugas mencatat setiap pembelian tiket, memastikan pencatatan transaksi yang akurat, serta memfasilitasi penjualan tiket dengan efisien di loket. Supir dapat melihat riwayat perjalanan, mengedit data pribadi, dan mengetahui titik penjemputan penumpang, sementara Pengunjung (penumpang) dapat membaca informasi, melakukan pembayaran, melihat riwayat perjalanan, memesan tiket, mengedit data pribadi, dan mengubah kata sandi.

Aplikasi ini menggunakan framework Laravel dan melibatkan dua bahasa pemrograman yang berbeda. Bagian back-end dikembangkan dengan bahasa Go, sementara antarmuka pengguna (front-end) dirancang menggunakan Vue. Setiap layanan memiliki basis data dengan port yang sama, meskipun setiap layanan menggunakan port yang berbeda. Ini bertujuan untuk memisahkan layanan satu sama lain, sehingga ketika satu layanan tidak tersedia, layanan lain masih dapat beroperasi.

## **Karakteristik Pengguna Aplikasi**

Pada aplikasi Bus Danau Toba, terdapat lima kategori pengguna yang masing-masing memiliki peran dan karakteristik khusus. **Super admin** memiliki hak akses tertinggi untuk mengelola seluruh sistem dan pengguna lainnya. **Admin kantor** bertanggung jawab atas administrasi dan operasi di tingkat kantor pusat, sementara **admin loket** mengelola transaksi dan operasional di loket bus. **Supir** memiliki akses untuk melihat jadwal perjalanan dan detail rute yang harus dilalui. **Pengunjung** atau **penumpang** adalah pengguna yang memanfaatkan layanan aplikasi untuk mencari jadwal, memesan tiket, dan mengelola perjalanan mereka. Masing-masing kategori ini memiliki akses dan fungsi yang disesuaikan dengan peran mereka dalam ekosistem aplikasi, sebagaimana dijelaskan dalam Tabel 1.

**Tabel 1. Karakteristik Pengguna Aplikasi**

| **Kategori Pengguna** | **Fungsi** | **Hak Akses ke Aplikasi** |
| --- | --- | --- |
| *Super Admin* | *Super Admin* perlu mengatur partisipasi semua merek bus yang ingin tergabung dalam sistem Aplikasi Tiket Sistem Bus di Kawasan Danau Toba. | *Super* *Admin* dapat mengelola merek bus yang ingin bergabung dalam sistem Aplikasi Tiket Sistem Bus di Kawasan Danau Toba. |
| *Admin* Kantor | *Admin* Kantor memiliki akses penting untuk mengelola berbagai aspek operasional perusahaan transportasi, termasuk jadwal bus, loket, mobil, rute, dan supir. Mereka juga dapat mengakses catatan keuangan untuk mengelola anggaran, komisi, dan membuat laporan keuangan. | *Admin* Kantor dapat mengelola jadwal bus-bus sesuai merek masing-masing, melihat catatan keuangan, melihat komisi, mengelola loket, mengelola mobil, mengelola rute, dan mengelola supir. |
| *Admin* Loket | *Admin* Loket mampu untuk mencatat pembelian tiket tidak hanya memfasilitasi proses penjualan tiket secara efisien, tetapi juga memungkinkan mereka untuk memantau setiap transaksi dengan teliti dan mencatatnya secara akurat. | *Admin* Loket memiliki hak akses untuk mencatat setiap pembelian tiket yang dilakukan, memastikan bahwa semua transaksi tercatat dengan akurat dan memungkinkan pengelolaan yang efisien dalam penjualan tiket di loket. |
| Supir | Sopir dapat melihat riwayat perjalanan, mengedit data diri, dan melihat titik penjemputan penumpang meningkatkan efisiensi layanan transportasi serta memungkinkan evaluasi kinerja, pembaruan data diri, dan perencanaan rute yang lebih baik. | Supir memiliki akses untuk melihat riwayat perjalanan yang telah dilakukan, mengedit data diri mereka sendiri, dan melihat titik penjemputan penumpang. |
| Pengunjung (penumpang) | Penumpang dapat membaca informasi, melakukan pembayaran, melihat riwayat perjalanan, memesan tiket, mengedit data diri, dan mengubah kata sandi meningkatkan kenyamanan, fleksibilitas, dan keamanan dalam penggunaan layanan transportasi. | Penumpang dapat membaca informasi, melakukan pembayaran, melihat riwayat perjalanan, memesan tiket, mengedit data diri, dan mengubah password. |

## **Fungsi pada Aplikasi**

Adapun fungsi yang terdapat pada sistem pemesanan tiket bus di sekitar Danau Toba ini, yaitu:

1. Fungsi Registrasi

Fungsi registrasi digunakan oleh *user* untuk mendaftarkan akunnya sebagai *Super Admin, Admin* Kantor*, Admin* Loket*,* Sopir atau Pengunjung (Penumpang) pada aplikasi.

1. Fungsi *Login*

Fungsi *login* digunakan oleh *Super Admin, Admin* Kantor*, Admin* Loket*,* Sopir atau Pengunjung (Penumpang) untuk dapat masuk dan mengakses aplikasi.

1. Fungsi *Add Schedules*

Fungsi *add schedules* merupakan fitur yang digunakan oleh *Admin* Kantor untuk menambahkan jadwal perjalanan baru ke dalam sistem. Tujuannya adalah agar informasi jadwal perjalanan dapat ditampilkan kepada penumpang melalui aplikasi. Dengan demikian, penumpang dapat melihat jadwal perjalanan yang tersedia dan melakukan reservasi tiket dengan lebih mudah.

1. Fungsi *Add Lokets*

Fungsi *add lokets* merupakan fitur yang digunakan oleh *Admin* Kantor untuk menambahkan loket baru ke dalam sistem. Hal ini dilakukan agar informasi loket tersedia dan dapat ditampilkan kepada penumpang melalui aplikasi. Dengan adanya informasi loket yang jelas, penumpang dapat dengan mudah mengetahui lokasi loket dan memudahkan mereka dalam proses pembelian tiket atau penyelesaian masalah terkait loket.

1. Fungsi *Add Routes*

Fungsi *add routes* merupakan fitur yang digunakan oleh *Admin* Kantor untuk menambahkan rute perjalanan baru ke dalam sistem. Tujuannya adalah agar informasi tentang rute perjalanan dapat ditampilkan kepada penumpang melalui aplikasi. Dengan demikian, penumpang dapat dengan mudah melihat rute perjalanan yang tersedia, memilih rute yang sesuai dengan kebutuhan mereka, dan melakukan *reservasi* tiket dengan lebih tepat.

1. Fungsi *Add Buses*

Fungsi *add buses* merupakan fitur yang digunakan oleh *Admin* Kantor untuk menambahkan bus baru ke dalam sistem aplikasi. Tujuannya adalah agar informasi tentang bus yang tersedia dapat ditampilkan kepada penumpang melalui aplikasi. Dengan adanya fitur ini, penumpang dapat dengan mudah melihat jenis bus yang tersedia yang sesuai dengan kebutuhan perjalanan mereka.

1. Fungsi *Edit User*

Fungsi *edit user* merupakan fitur yang digunakan oleh *user* untuk mengedit data diri pada aplikasi. Tujuannya agar data diri yang dimiliki dapat di *update* apabila ada data yang perlu diubah, misalnya pada data penumpang terdapat perubahan nomor telepon maka, penumpang dapat mengedit datanya sendiri.

1. Fungsi *Edit Routes*

Fungsi *edit routes* merupakan fitur yang disediakan bagi *Admin* Kantor untuk memperbarui atau mengubah informasi rute perjalanan dalam aplikasi. Tujuannya adalah memungkinkan *Admin* Kantor untuk melakukan perubahan pada rute perjalanan yang diperlukan, seperti penambahan, penghapusan, atau perubahan rute yang ada, sehingga informasi tentang rute perjalanan selalu terkini dan akurat dalam sistem aplikasi.

1. Fungsi *Edit Buses*

Fungsi *edit buses* adalah fitur yang diperuntukkan bagi Admin untuk memperbarui informasi tentang bus dalam aplikasi. Tujuannya adalah memungkinkan Admin untuk melakukan perubahan pada data bus yang ada, seperti mengubah jenis bus, kapasitas penumpang, atau status ketersediaan bus, sehingga informasi tentang bus selalu terkini dan akurat dalam sistem aplikasi.

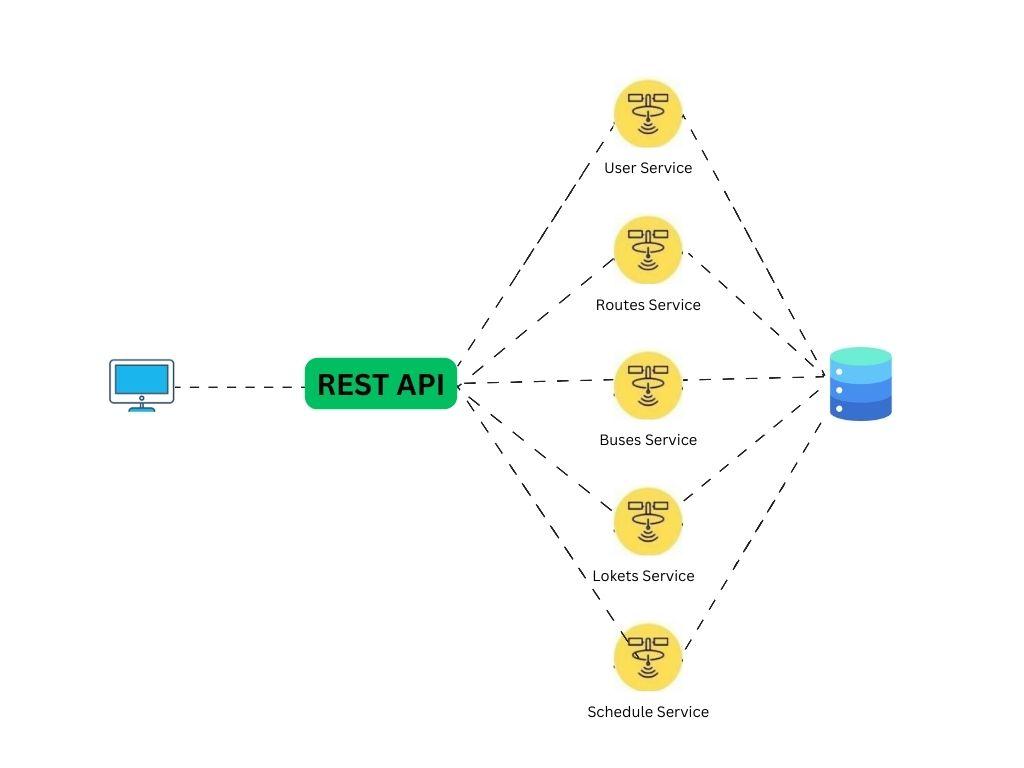
1. Fungsi *Edit Lokets*

# 

# **Arsitektur Monolith dan Microservice**

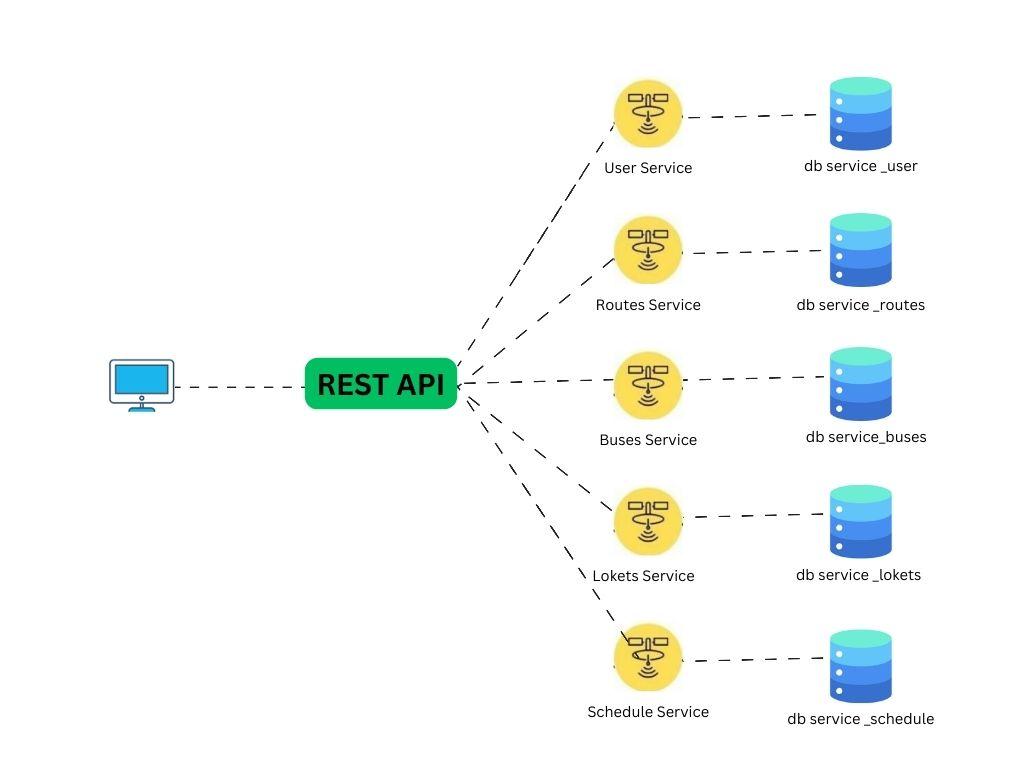
Pada bab ini dijelaskan tentang perbedaan arsitektur Monolith dan Microservice.

1. Arsitektur monolith menggabungkan semua komponen aplikasi menjadi satu unit yang besar dan terintegrasi, yang memudahkan debugging dan dapat memberikan performa yang lebih cepat karena tidak adanya overhead komunikasi antar layanan. Namun, arsitektur monolith cenderung kurang fleksibel, lebih sulit untuk diskalakan, dan rentan terhadap kegagalan sistem karena ketergantungan yang tinggi antar komponen. Dalam konteks pengembangan dan pemeliharaan, arsitektur microservice lebih kompleks dibandingkan monolith tetapi lebih adaptif terhadap perubahan dan perkembangan teknologi.



**Gambar . Monolith**

1. Aplikasi ini menerapkan arsitektur microservice, yang sesuai dengan konsepnya, bertujuan untuk memecah layanan menjadi unit yang lebih kecil. Sehingga, setiap layanan memiliki basis data sendiri. Service yang ada pada aplikasi adalah service\_user, service\_routes, service\_buses, service\_loket, service\_schedules. Setiap layanan beroperasi secara independen tanpa memengaruhi layanan lainnya. Meskipun terpisah, layanan-layanan tersebut tetap terhubung dan berkomunikasi melalui HTTP Request yang mengikuti standar REST API. Keuntungan arsitektur ini adalah skalabilitas, ketahanan, kemudahan pengembangan, dan fleksibilitas. Namun, arsitektur ini juga lebih kompleks, performanya bisa lebih lambat, dan debuggingnya lebih sulit.



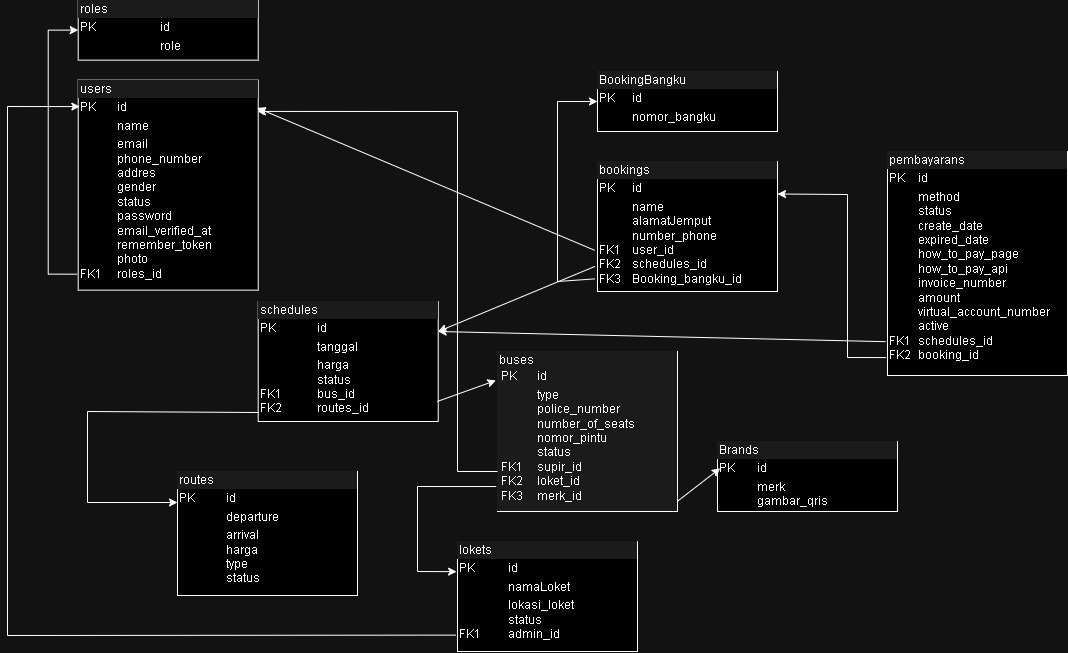
**Gambar . Arsitektur Microservice**

# **Arsitektur Layanan Mikro**

Arsitektur layanan mikro terdiri dari beberapa prosesor independen. Masing-masing mencakup komponen umum dari aplikasi perusahaan seperti, antarmuka pengguna, basis data, server, model tarik dan dorong.

## **Basis Data**

Pada bagian ini akan dijelaskan basis data yang digunakan melalui *relational* model*.* *Relational* model merupakan jenis data yang menyimpan dan menyediakan akses ke data lain yang saling berkaitan. Pada Gambar 2 memuat setiap data yang berkaitan pada aplikasi. Aplikasi Bus Danau Toba ini memiliki 9 entitas yaitu *user, roles, schedule, booking* bangku, pembayaran, *buses, routes, brands,* dan *lokets.* *Relational database* ini memberikan representasi yang jelas dan terstruktur dari sistem pemesanan tiket bus, memfasilitasi pemahaman dan pengelolaan data yang efektif untuk operasi pemesanan tiket yang lancar.



**Gambar . *Relational Diagram***

## **Server**

Server adalah komputer atau sistem komputer yang menyediakan layanan atau sumber daya kepada komputer-komputer lainnya melalui jaringan, seperti penyimpanan data, penggunaan jaringan, menjalankan aplikasi, komputasi berat, dan pengontrol akses.

* 1. **Model Tarik dan Dorong (Push and Pull Model)**

Dalam arsitektur microservice model push dan pull mengacu pada bagaimana data atau informasi dipindahkan antara layanan-layanan tersebut. Berikut merupakan model push dan pull pada aplikasi pemesanan tiket bus di sekitar Danau Toba.

1. Push

Server secara proaktif mengirimkan data, informasi, atau layanan kepada klien tanpa menunggu permintaan klien. Contohnya adalah server email yang mengirimkan pesan ke kotak masuk pengguna segera setelah pesan tersebut diterima oleh server. Jadi, ketika seseorang mengirim email kepada Anda, server email Anda akan secara otomatis mendorong pesan itu ke kotak masuk Anda tanpa Anda perlu melakukan tindakan aktif untuk mengunduhnya.

1. Pull

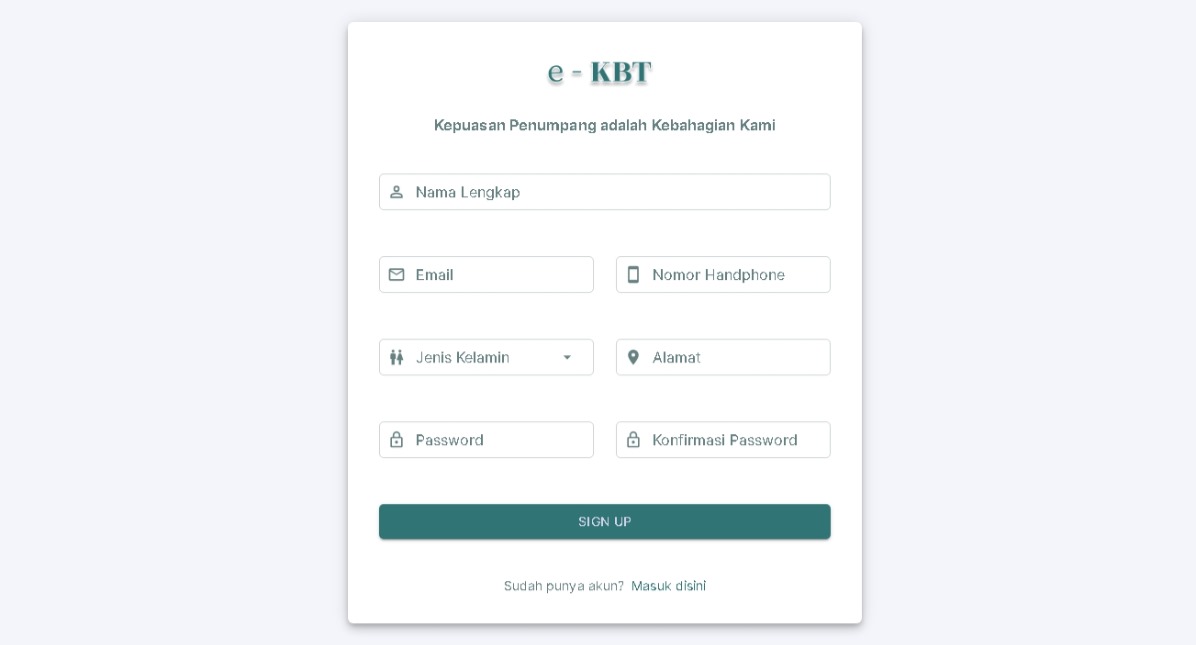
Klien atau pengguna mengambil inisiatif untuk meminta atau menarik data, informasi, atau layanan dari server sesuai dengan kebutuhan mereka. Contohnya adalah ketika Anda mengakses situs web melalui peramban web Anda. Peramban membuat permintaan kepada server untuk mengirimkan halaman web tertentu, dan server merespons dengan mengirimkan halaman web tersebut ke peramban Anda. Dalam hal ini, Anda adalah yang menarik data dari server sesuai dengan permintaan Anda.

## **Antarmuka Pengguna**

Pada bagian ini akan dijelaskan antarmuka pengguna pada aplikasi yang telah dikembangkan.

1. Tampilan Halaman Registrasi

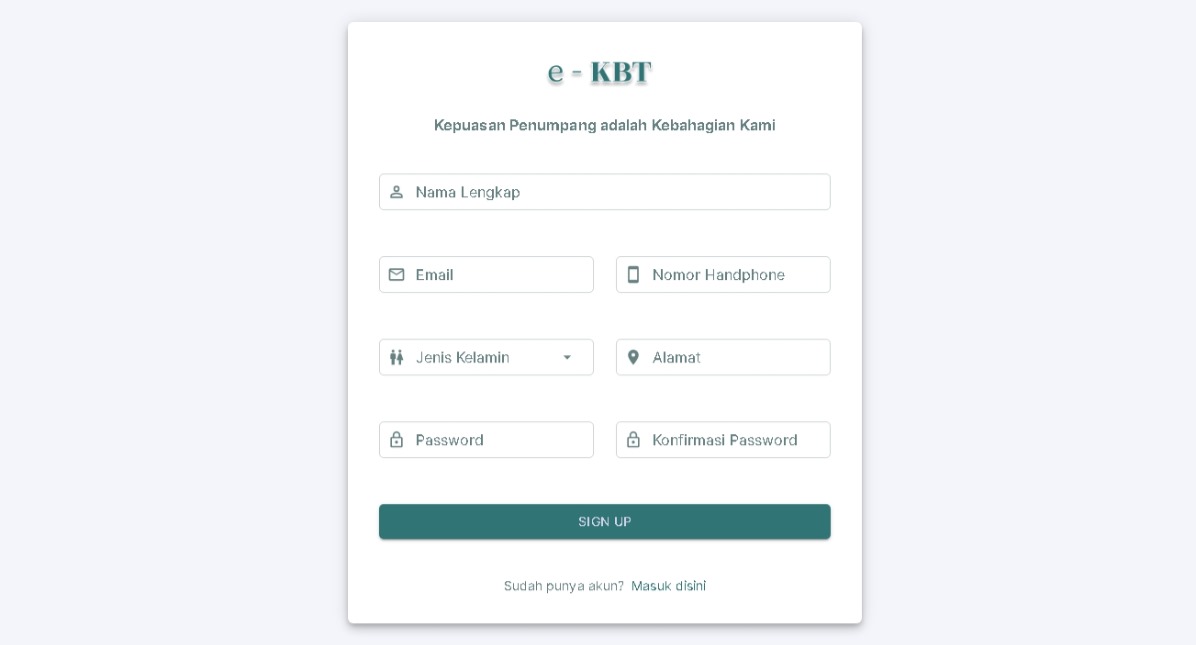
Tampilan halaman registrasi terlampir pada Gambar



**Gambar . Tampilan Halaman Registrasi**

1. Tampilan Halaman Login

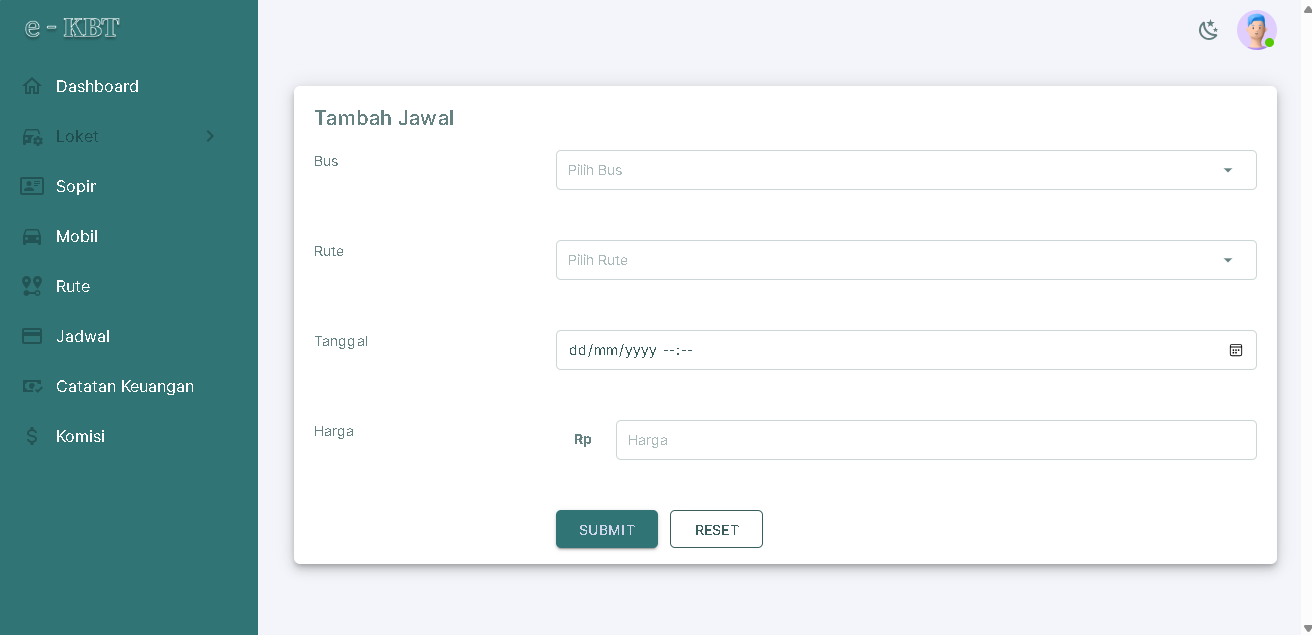
Tampilan halaman login terlampir pada Gambar



**Gambar . Tampilan Halaman Login**

1. Tampilan Halaman *Add Schedules*

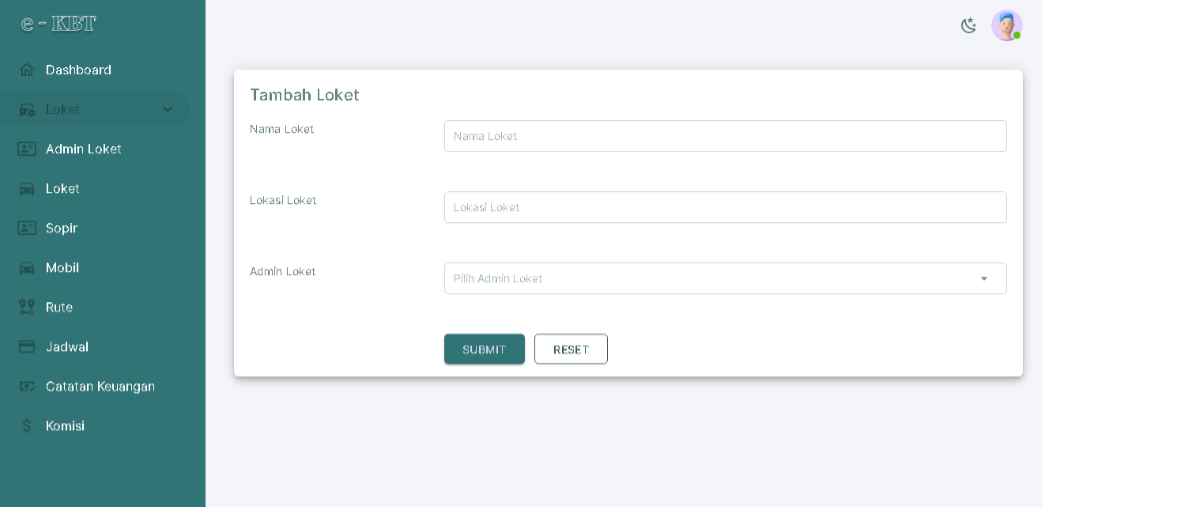
Tampilan halaman *add schedules* terlampir pada Gambar



**Gambar . Tampilan Halaman Add Schedules**

1. Tampilan Halaman *Add Lokets*

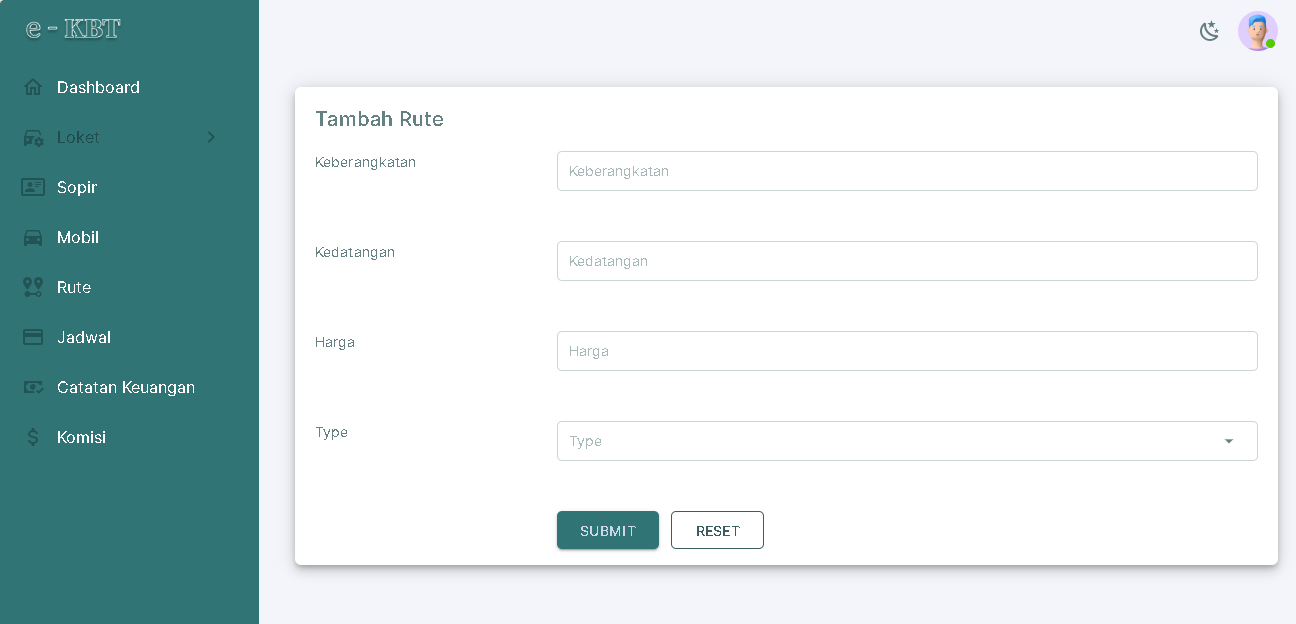
Tampilan halaman *add loket terlampir* pada Gambar



**Gambar . Tampilan Halaman Add Lokets**

1. Tampilan Halaman *Add Routes*

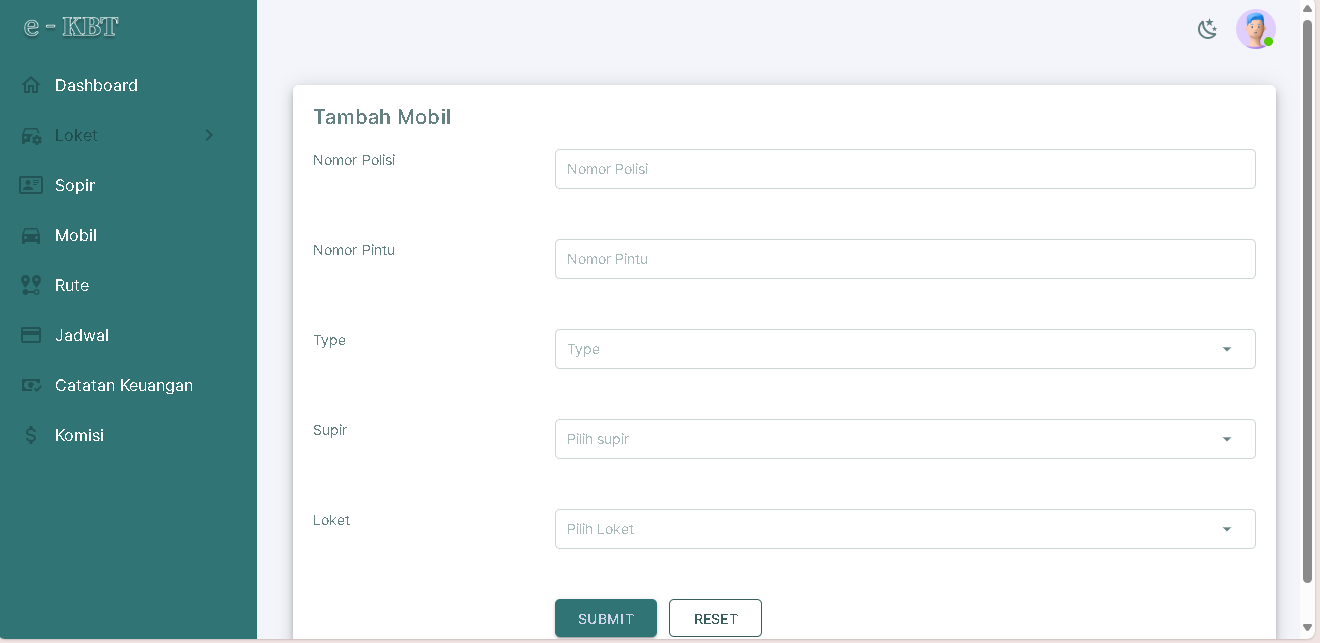
Tampilan halaman *add routes* terlampir pada Gambar



**Gambar . Tampilan Halaman Add Routes**

1. Tampilan Halaman *Add Buses*

Tampilan halaman *add buses* terlampir pada Gambar



**Gambar . Tampilan Halaman Add Buses**

1. Tampilan Halaman *Edit User*

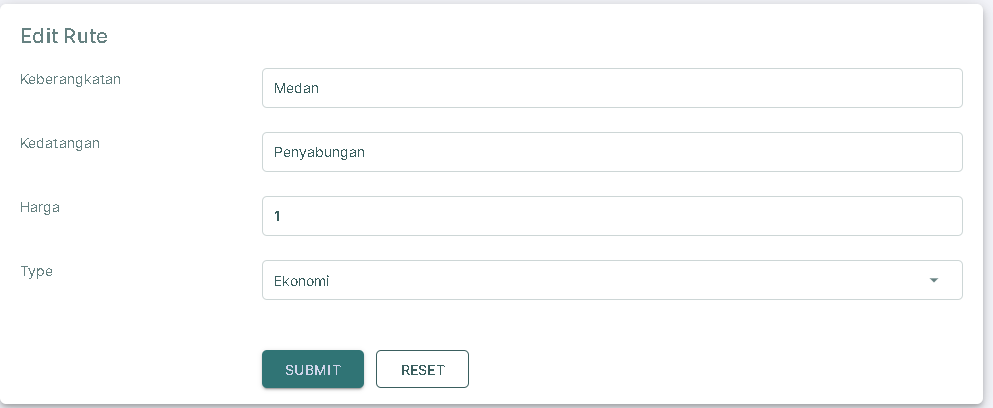
Tampilan halaman *edit user* terlampir pada Gambar



**Gambar . Tampilan Halaman Edit User**

1. Tampilan Halaman *Edit Routes*

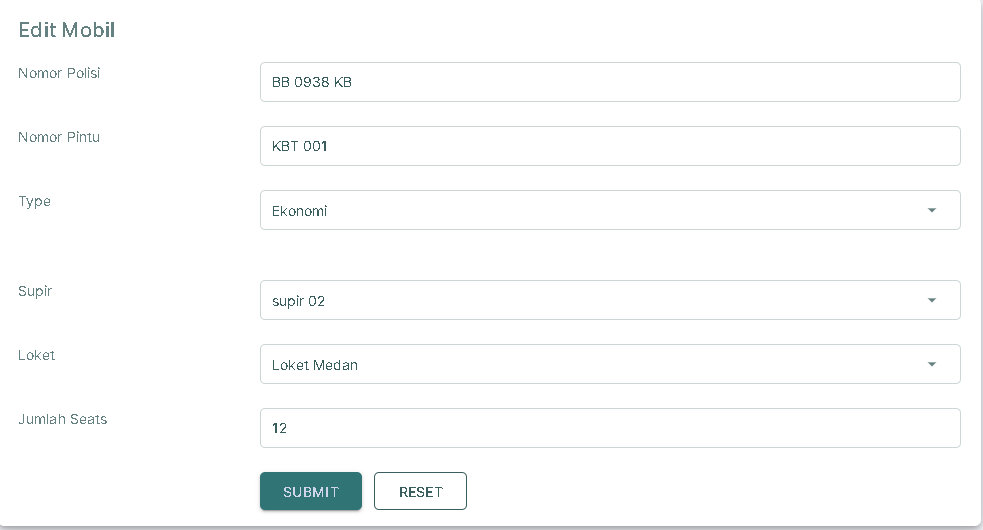
Tampilan halaman *edit routes* terlampir pada Gambar



**Gambar . Tampilan Halaman Edit Routes**

1. Tampilan Halaman *Edit Buses*

Tampilan halaman *edit buses* terlampir pada Gambar

****

**Gambar . Tampilan Halaman Edit Buses**