

# Dokumentasi Aplikasi MyTiket

## MyTiket

Sistem pemesanan tiket kolam renang berbasis aplikasi mobile

Platform: Mobile (Android/iOS)

### Tech Stack:

- Frontend: React Native + TypeScript
- Backend: REST API (Node.js / Express) (*bisa disesuaikan*)
- Database: PostgreSQL
- Komunikasi Data: JSON over HTTP

MyTiket adalah aplikasi mobile yang menyediakan layanan pemesanan tiket kolam renang secara digital. Aplikasi ini terintegrasi dengan backend API sebagai pusat pemrosesan data dan PostgreSQL sebagai sistem penyimpanan.

### Permasalahan

- Pemesanan tiket masih dilakukan secara manual
- Informasi harga dan ketersediaan tiket tidak terpusat
- Proses antrian tinggi pada jam kunjungan
- Tidak ada sistem untuk pencatatan transaksi yang terstruktur

### Alasan Pengembangan

Digitalisasi dibutuhkan untuk memberi pengalaman pemesanan yang lebih cepat, efisien, dan mudah diakses, baik bagi pengguna maupun pengelola kolam renang.

### 3. Tujuan Pengembangan (Objective Teknis)

#### Tujuan utama implementasi sistem:

- Digitalisasi proses pemesanan tiket  
Menghilangkan proses manual dan antrean fisik.
- Integrasi aplikasi mobile dengan backend API  
Menyediakan alur transaksi real-time.
- Persistensi data di PostgreSQL  
Menjamin data transaksi terkelola dan terstruktur.

- Optimasi pengalaman pengguna  
Pengguna dapat memesan tiket kapan saja tanpa datang ke lokasi.
- Manajemen data bagi pengelola  
Pengelola dapat mengakses data transaksi untuk evaluasi dan operasional.

Tujuan teknis tambahan:

- Membangun codebase yang modular dan mudah diperluas
- Menyediakan endpoint API dengan kontrak yang jelas
- Menyiapkan fondasi untuk fitur lanjutan (QR check-in, pembayaran online, dashboard admin, dsb.)

#### 4. Manfaat Sistem (System-Level Benefit)

Manfaat untuk Pengguna

- Pemesanan tiket lebih cepat & fleksibel
- Informasi harga dan kolam tersedia secara transparan
- Tidak perlu antri secara fisik di lokasi

Manfaat untuk Pemilik/Pengelola Kolam

- Data transaksi terdokumentasi dan dapat dianalisis
- Kontrol terhadap kuota pengunjung (jika dibutuhkan)
- Pengurangan beban kerja manual
- Modernisasi sistem operasional

Manfaat untuk Admin Sistem

- Dapat mengelola data kolam, tiket, dan transaksi
- Akses ke database terstruktur untuk kebutuhan reporting
- Skalabilitas untuk pengembangan fitur lebih lanjut

## Operasi CRUD pada Sistem MyTiket

Aplikasi MyTiket minimal mendukung empat operasi dasar dalam REST API yaitu GET, POST, PUT, dan DELETE. Operasi ini digunakan untuk berinteraksi dengan data tiket, kolam, pengguna, dan pemesanan.

### 1. GET – Mengambil Data

Tujuan:

Digunakan untuk mengambil/membaca data dari server tanpa melakukan perubahan.

Contoh Implementasi di MyTiket:

- GET daftar kolam renang
- GET detail harga tiket
- GET riwayat pemesanan
- GET data pengguna (profil)

Contoh Endpoint:

- GET /kolam
- GET /pesanan/history
- GET /user/me

Karakteristik Teknis:

- Tidak membutuhkan body
- Bisa menggunakan query params untuk filter/sort (opsional)
- Idempotent (request berulang → hasil sama)

### 2. POST – Menambahkan Data Baru

Tujuan:

Digunakan untuk membuat atau mengirim data baru ke server.

Contoh Implementasi di MyTiket:

- POST register pengguna
- POST login (autentikasi)
- POST membuat pesanan tiket baru
- POST menambahkan kolam baru (*untuk sisi admin*)

Contoh Endpoint:

- POST /auth/register
- POST /auth/login
- POST /pesanan

Karakteristik Teknis:

- Mengirim data via request body
- Server melakukan validasi & menyimpan ke database
- Biasanya mengembalikan status 201 Created atau 200 OK

### 3. PUT – Memperbarui Data

Tujuan:

Digunakan untuk memperbarui data yang sudah ada di server.

Contoh Implementasi di MyTiket:

- PUT update profil pengguna
- PUT update data kolam renang (*admin*)
- PUT update status pemesanan (*jika ada*)

Contoh Endpoint:

- PUT /user/me
- PUT /kolam/:id
- PUT /pesanan/:id

Karakteristik Teknis:

- Mengirim data lengkap untuk mengganti data yang lama
- Bisa juga menggunakan PATCH jika hanya update sebagian
- Membutuhkan parameter id untuk menentukan data target

### 4. DELETE – Menghapus Data

Tujuan:

Digunakan untuk menghapus data yang sudah ada dalam sistem.

Contoh Implementasi di MyTiket:

- DELETE kolam renang (*admin*)

- DELETE pesanan tiket (jika ada fitur cancel)
- DELETE akun pengguna (*opsional*)

Contoh Endpoint:

- DELETE /kolam/:id
- DELETE /pesanan/:id

Karakteristik Teknis:

- Hanya membutuhkan id
- Setelah sukses → data dihapus dari DB atau berubah status menjadi deleted

#### A. Tabel Operasi CRUD pada MyTiket

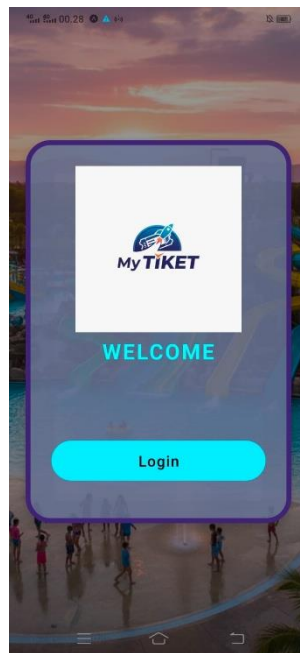
Operasi	Deskripsi Umum	Contoh Penggunaan di MyTiket	Endpoint Contoh	Data yang Dikirim	Data yang Dikembalikan
<b>GET</b>	Mengambil data dari server	Menampilkan daftar kolam renang & riwayat pemesanan	GET /kolam GET /pesanan/history	Tidak ada (optional: query)	List data dalam format JSON
<b>POST</b>	Menambahkan data baru ke server	Registrasi, login, dan pemesanan tiket	POST /auth/register POST /pesanan	JSON (body payload)	JSON (objek yang baru dibuat atau token)
<b>PUT</b>	Memperbarui data yang sudah ada	Update data kolam (admin) atau profil pengguna	PUT /kolam/:id PUT /user/me	JSON + parameter id	JSON (data yang diperbarui)
<b>DELETE</b>	Menghapus data yang sudah ada	Menghapus kolam / membatalkan pesanan	DELETE /kolam/:id DELETE /pesanan/:id	Parameter id	Status success + optional JSON

## B. Fitur Aplikasi

Fitur pada versi pengembangan saat ini memiliki fokus utama pada alur pemesanan tiket oleh pengguna. Adapun fitur yang sudah diimplementasikan adalah sebagai berikut:

### Halaman Awal (Pre-Login)

Halaman awal yang ditampilkan ketika pengguna belum login. Berfungsi sebagai entry point pertama untuk mengarahkan pengguna ke proses autentikasi.



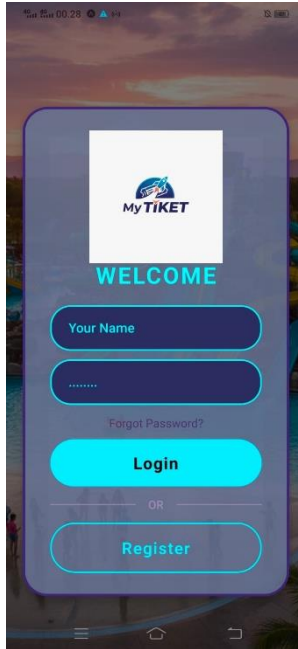
### Fungsi Utama

- Menampilkan pesan selamat datang (Welcome Screen)
- Menampilkan tombol aksi untuk login
- Menjadi gateway sebelum masuk ke sistem pemesanan

## 5.1 Login

Deskripsi Fungsi:

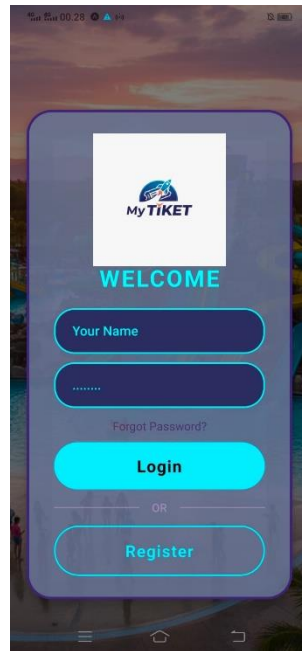
Pengguna dapat melakukan autentikasi menggunakan akun yang telah terdaftar untuk mengakses fitur aplikasi.



Detail Teknis:

- Method: POST /auth/login
- Input minimum: email/username + password
- Output: JWT Token untuk sesi autentikasi
- Validasi sisi frontend & backend
- Penyimpanan token menggunakan secure local storage (AsyncStorage, dsb.)
- Token digunakan untuk akses endpoint yang membutuhkan autentikasi

## 5.2 Register



Deskripsi Fungsi:

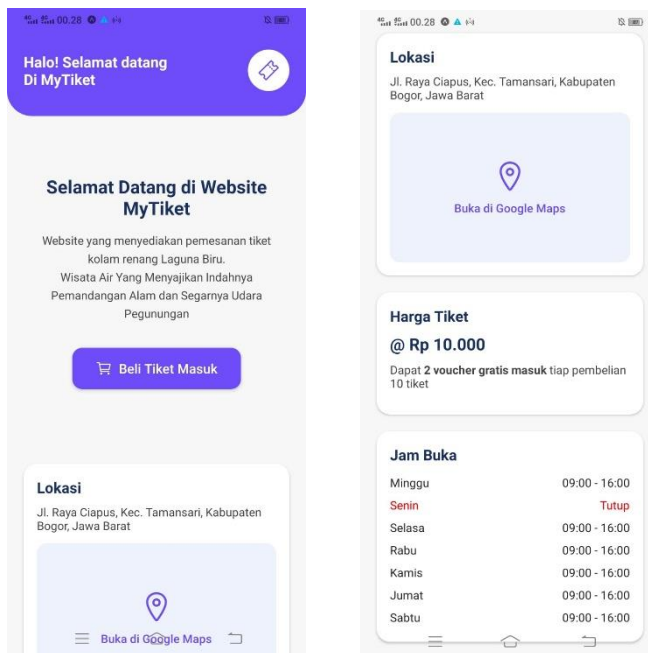
Pengguna baru dapat membuat akun untuk dapat melakukan pemesanan tiket.

Detail Teknis:

- Method: POST /auth/register
- Field: nama, email/username, password
- Validasi input (format email, panjang password, dsb.)
- Hashing password di sisi backend sebelum disimpan
- Data disimpan pada tabel pengguna (PostgreSQL)



## 5.3 Halaman Awal



Komponen tampilan:

- Pesan selamat datang
- Informasi jam buka kolam renang
- Lokasi atau informasi operasional dasar

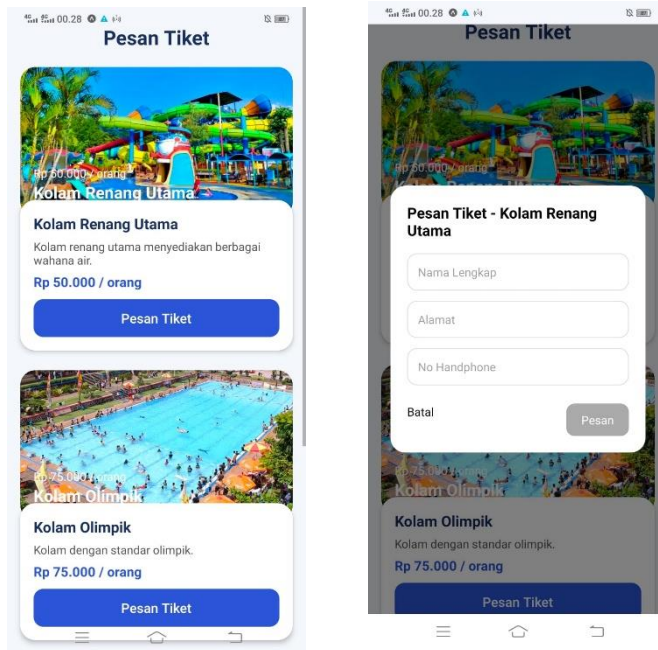
Tujuan Fungsi:

- Memberikan informasi kontekstual sebelum pengguna melakukan pemesanan

Catatan Teknis:

- Data jam buka/lokasi bisa statis atau dinamis melalui endpoint
- Menjadi entry point navigasi menuju halaman pemesanan tiket

## 5.4 Halaman Pesan Tiket



Deskripsi:

Menampilkan daftar kolam renang yang bekerja sama, beserta harga tiket.

Komponen Fungsi:

- List kolam renang
- Harga tiket per kolam
- Tombol aksi “Pesan” untuk melakukan reservasi

Teknis Backend/Frontend:

- Method: GET /kolam (list kolam + harga)
- Data disimpan di tabel kolam dan harga\_tiket (PostgreSQL)
- Pemesanan dilakukan melalui:
  - POST /pesanan (create order)
- Belum terdapat pembayaran (jika ada nanti disambung ke payment gateway)

## 5.5 Halaman Riwayat Pemesanan



### Deskripsi:

Pengguna dapat melihat daftar tiket yang pernah dipesan.

### Fungsi Utama:

- Menampilkan histori transaksi tiket berdasarkan akun pengguna
- Detail seperti: nama kolam, harga, tanggal pemesanan, status pemesanan

### Teknis Backend:

- Method: GET /pesanan/history
- Filtering berdasarkan user\_id
- Data tersimpan pada tabel pesanan

### Manfaat Teknis:

- Memvalidasi bahwa sistem sudah mencatat transaksi
- Membuka peluang fitur lanjutan seperti:
  - QR Code tiket
  - Invoice digital
  - Cancel order
  - Check-in validation