

# Documentation System Dating App

## GitHub Repository Url

<https://github.com/achmad-acm11/DatingApp.git>

## List Fungsional dan Non-Fungsional

Fungsional	Non-Fungsional
Sistem harus memungkinkan pengguna untuk mendaftar dengan email dan kata sandi.	Sistem harus menggunakan enkripsi untuk menyimpan kata sandi pengguna.
Sistem harus memungkinkan pengguna untuk melihat daftar calon pasangan yang cocok.	Sistem harus merespons permintaan pengguna dalam waktu kurang dari 2 detik.
Sistem harus memungkinkan pengguna untuk menyukai (like) atau tidak menyukai (dislike) calon pasangan.	Kode sumber harus terdokumentasi dengan baik untuk memudahkan pemeliharaan dan pengembangan lebih lanjut.
Sistem harus menyimpan dan mengelola data pembayaran jika ada fitur berbayar.	Sistem harus memiliki logging dan monitoring untuk memudahkan diagnosis masalah.

## Tech Stacks

### Bahasa Pemrograman

- **Go (Golang)**
  - **Kinerja Tinggi:** Go dikenal karena kinerjanya yang cepat dan efisiensi memori.
  - **Konkuren:** Go memiliki dukungan bawaan untuk concurrency, yang memungkinkan penanganan banyak permintaan secara bersamaan dengan efisiensi tinggi.
  - **Kompilasi Statis:** Go dikompilasi menjadi binary yang berdiri sendiri, sehingga mudah untuk dideploy dan dijalankan di berbagai lingkungan.

### Framework Web

- **Gin (Go)**
  - **Ringan dan Cepat:** Gin adalah framework web yang sangat cepat dan ringan untuk Go, yang ideal untuk membangun API RESTful.
  - **Middleware:** Mendukung middleware yang memudahkan penanganan otentikasi, logging, dan error handling.
  - **JSON Handling:** Memiliki fitur yang baik untuk menangani request dan response JSON, yang umum dalam API modern.

## Database

- **MySQL**
  - **Relational Database:** MySQL adalah database relasional yang populer dan memiliki fitur-fitur handal.
  - **Kinerja Tinggi:** MySQL dikenal karena kinerjanya yang cepat dalam berbagai skenario, dari aplikasi kecil hingga besar.
  - **Komunitas dan Dukungan:** MySQL memiliki komunitas besar dan dukungan yang luas, membuatnya mudah untuk menemukan solusi dan dukungan.

## Autentikasi dan Otorisasi

- **JWT (JSON Web Tokens)**
  - **Stateless Authentication:** JWT memungkinkan autentikasi yang stateless, sehingga tidak perlu menyimpan sesi di server.
  - **Fleksibilitas:** Dapat digunakan untuk otentikasi pengguna serta komunikasi antar layanan dengan keamanan yang baik.

## API Documentation

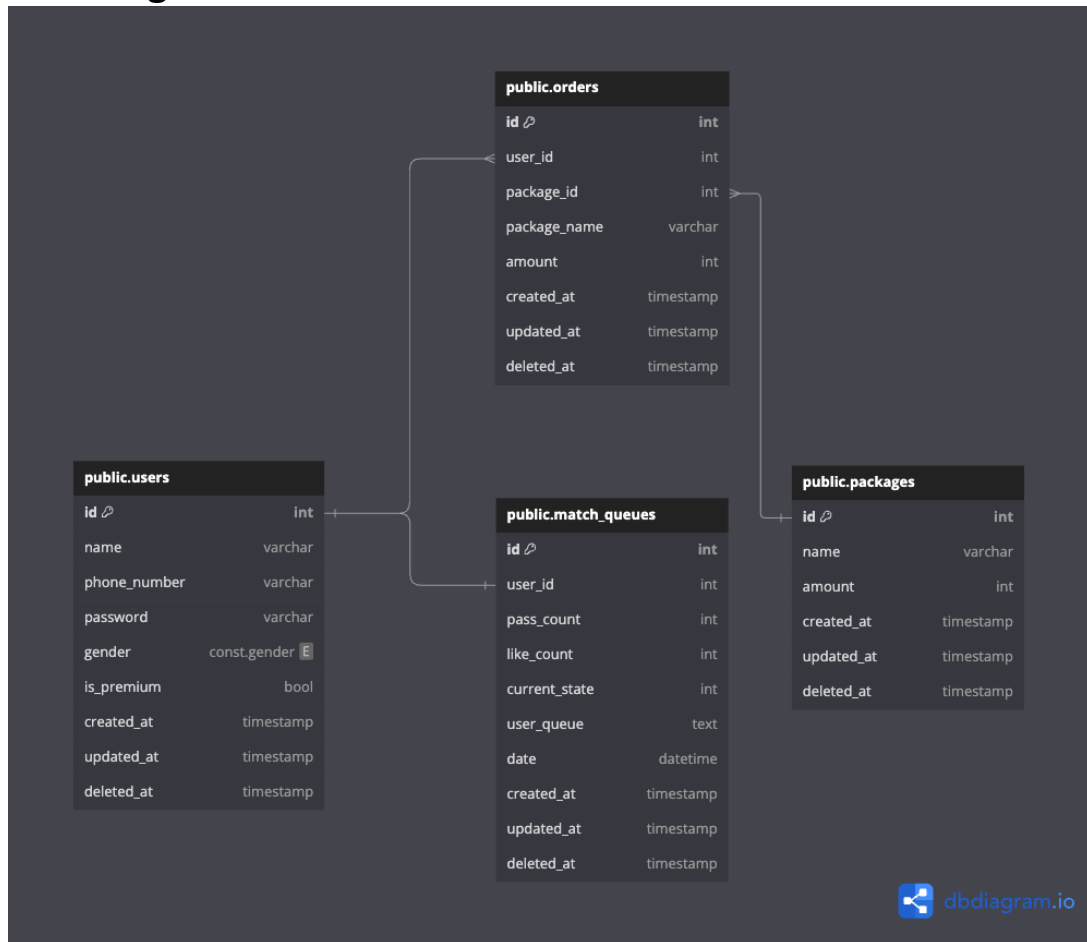
- **Postman**
  - **Interaktif dan Mudah Digunakan:** Postman memungkinkan pengujian API yang mudah dan dokumentasi yang interaktif.
  - **Koleksi yang Dapat Dibagikan:** Anda dapat membuat koleksi endpoint yang dapat dibagikan dengan tim untuk pengujian dan pengembangan.

## Alasan di Balik Pemilihan Teknologi

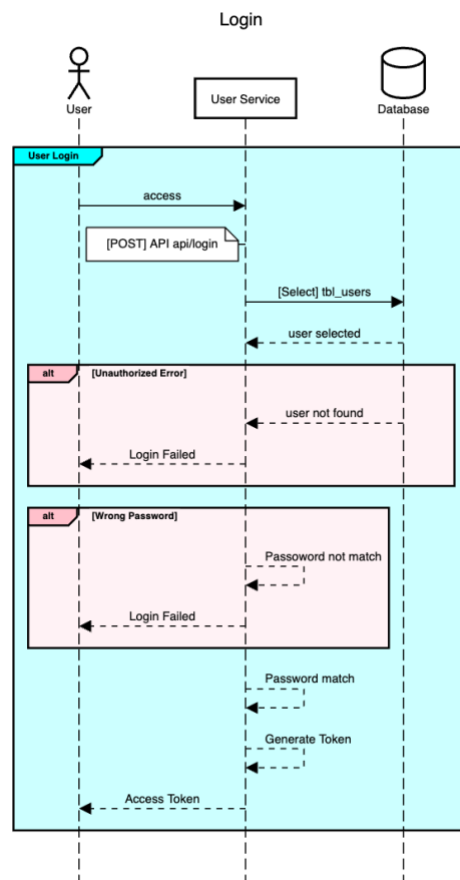
1. **Kinerja dan Skalabilitas:** Go, MySQL dipilih karena kinerja tinggi dan kemampuan skalabilitasnya.
2. **Efisiensi Pengembangan:** Framework Gin dan Postman memungkinkan pengembangan yang cepat dan efisien, dengan dokumentasi API yang baik.
3. **Keamanan:** JWT memastikan otentikasi yang aman dan komunikasi antar layanan yang andal.

## Design of The System

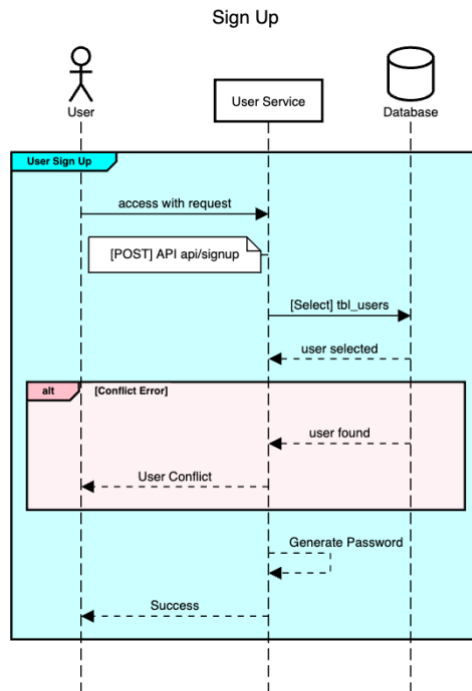
- ERD Diagram



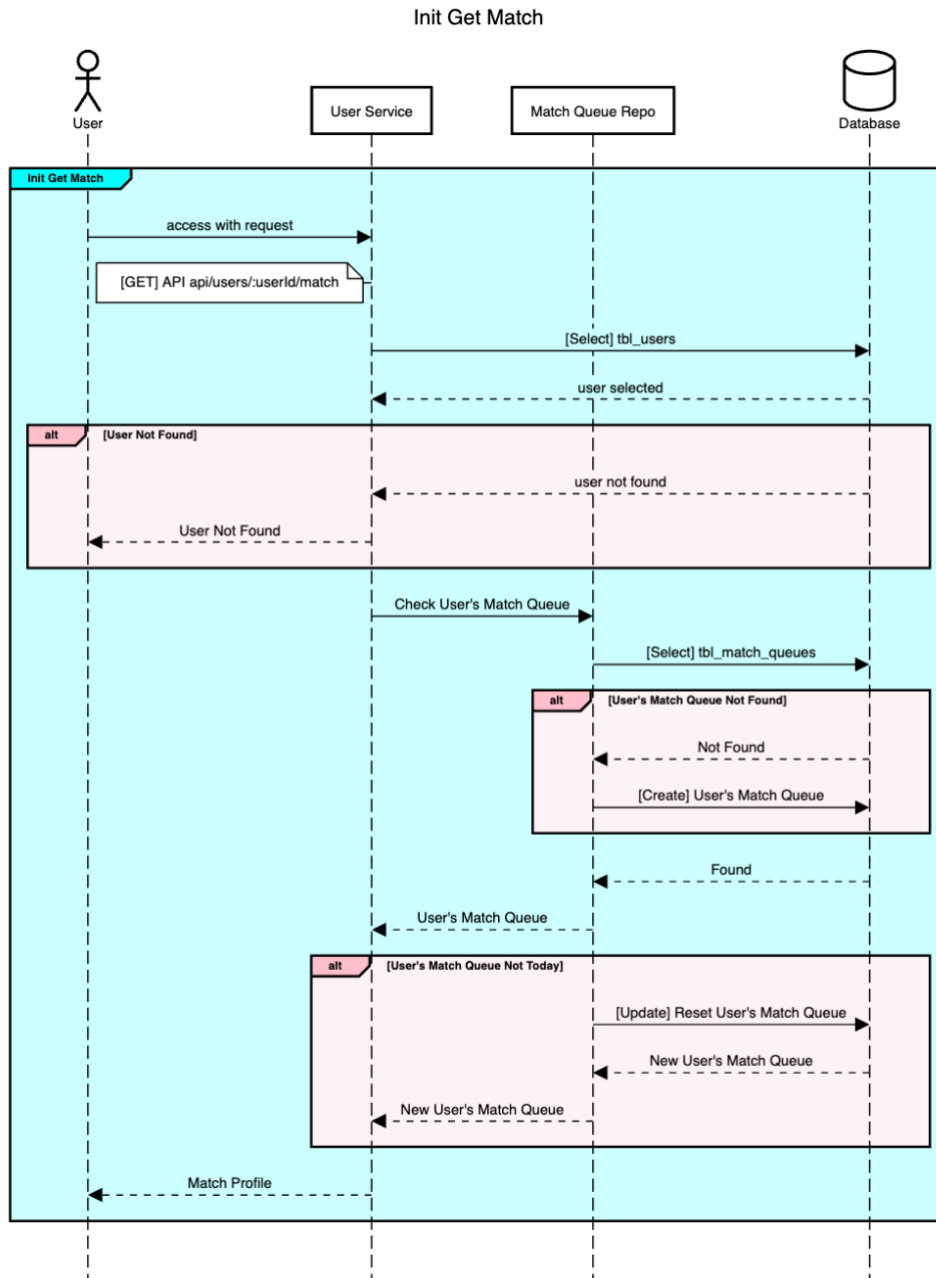
- Sequence Diagram



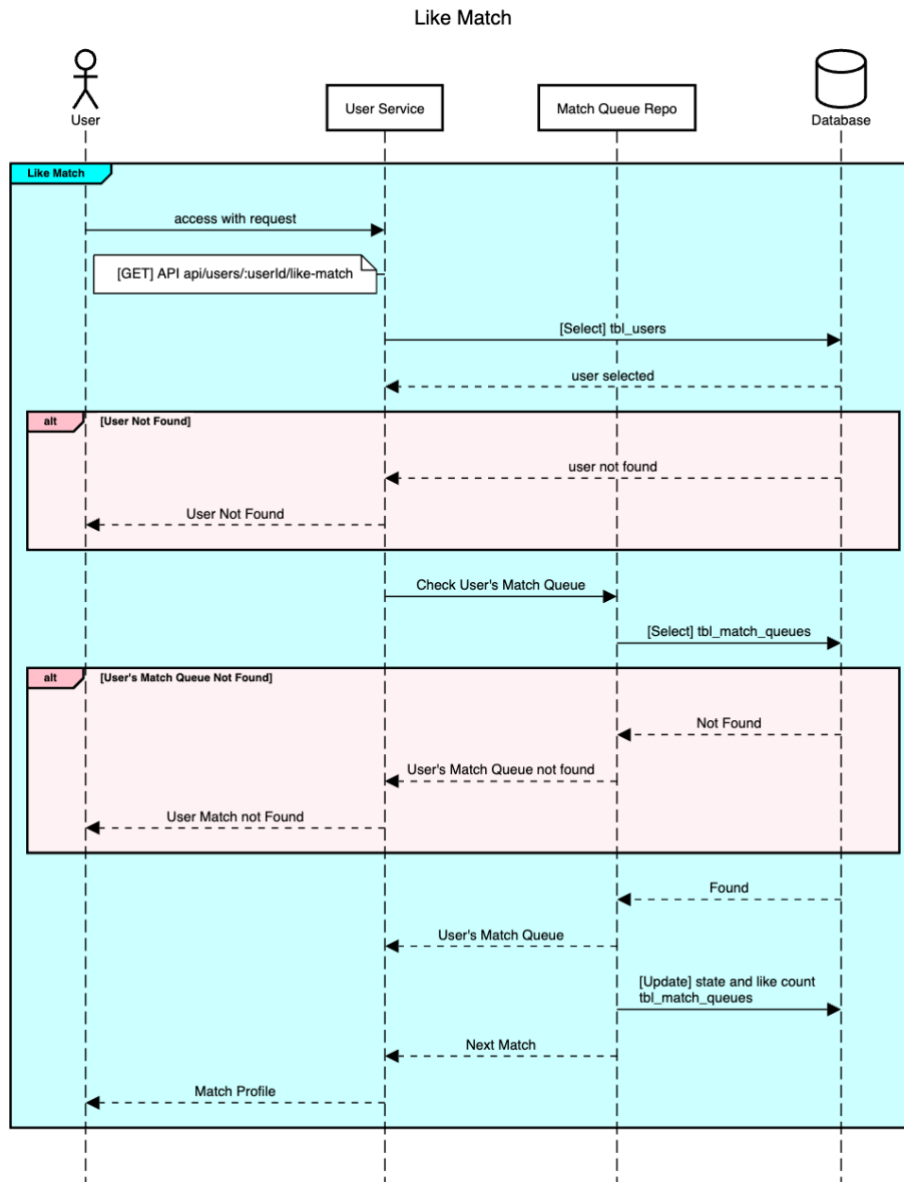
Gambar 1 Login Diagram



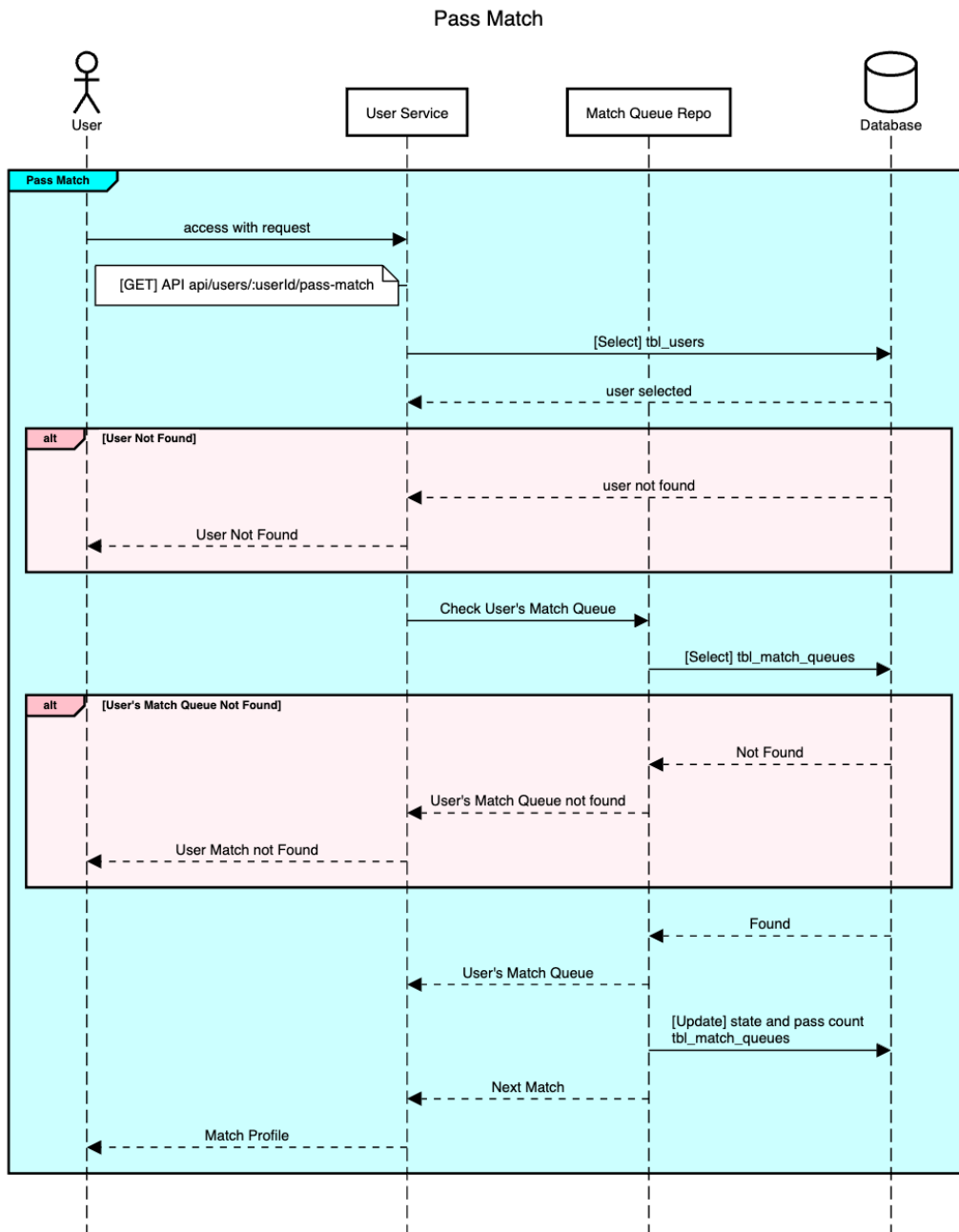
Gambar 2 SignUp Diagram



Gambar 3 Init Get Match Diagram

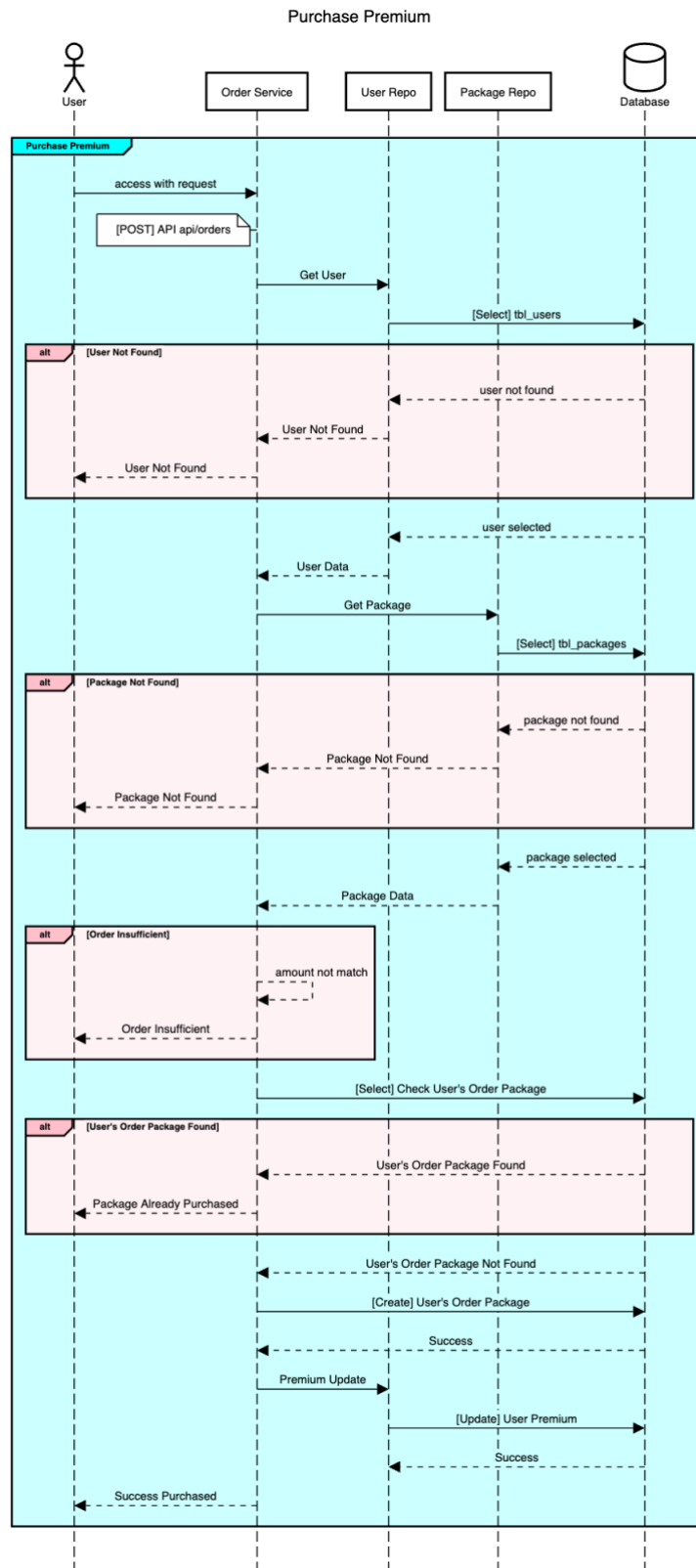


Gambar 4 Like Match Diagram



Gambar 5 Pass Match Diagram





Gambar 6 Purchased Diagram

## List Test Cases

### User SignUp

- **TC-01: Success SignUp**
  - **Input:** Semua field terisi dan valid.
  - **Expected Result:** User berhasil terdaftar, memberi response status 200.
- **TC-02: Email Conflict**
  - **Input:** Email yang sudah terdaftar.
  - **Expected Result:** User gagal terdaftar, memberi response status 409.
- **TC-03: Bad Request**
  - **Input:** Semua field tidak terisi.
  - **Expected Result:** Memberi response status 400.

### User Login

- **TC-04: Success Login**
  - **Input:** Semua field terisi dan valid.
  - **Expected Result:** User berhasil login dan memberi response status 200 serta token.
- **TC-05: Unauthorized Error**
  - **Input:** Menggunakan email yang belum terdaftar.
  - **Expected Result:** User gagal login dan memberi response status 401
- **TC-06: Wrong Password**
  - **Input:** Menggunakan password yang salah.
  - **Expected Result:** User gagal login dan memberi response status 401

### Init Get Match

- **TC-07: Success Get Match**
  - **Input:** Access menggunakan Header token
  - **Expected Result:** User berhasil mendapatkan user pasangan dan memberi response status 200.

- **TC-08: Unauthorized Error**

- **Input:** Access tanpa menggunakan Header token
- **Expected Result:** User gagal mendapatkan user pasangan dan memberi response status 401.

### **Like Match**

- **TC-09: Success Like Match**

- **Input:** Access menggunakan Header token
- **Expected Result:** User berhasil menyukai user pasangan dan memberi response status 200.

- **TC-10: User Match Not Found**

- **Input:** Access tanpa menggunakan Header token
- **Expected Result:** User gagal menyukai user pasangan dan memberi response status 401.

### **Pass Match**

- **TC-11: Success Pass Match**

- **Input:** Access menggunakan Header token
- **Expected Result:** User berhasil mengabaikan user pasangan dan memberi response status 200.

- **TC-12: User Match Not Found**

- **Input:** Access tanpa menggunakan Header token
- **Expected Result:** User gagal mengabaikan user pasangan dan memberi response status 401.

### **Purchased Premium**

- **TC-13: Success Purchased**

- **Input:** Menggunakan packageId yang valid dan belum di beli.
- **Expected Result:** User berhasil membayar package premium dan memberi response status 200.

- **TC-14: Package Not Found**

- **Input:** Menggunakan packageId yang tidak valid.
- **Expected Result:** User gagal membayar package premium dan memberi response status 404.

- **TC-15: Already Purchased Package**

- **Input:** Menggunakan packageId yang tidak valid.
- **Expected Result:** User gagal membayar package premium dan memberi response status 409.