

Kelas / No : 3SI1 / 22

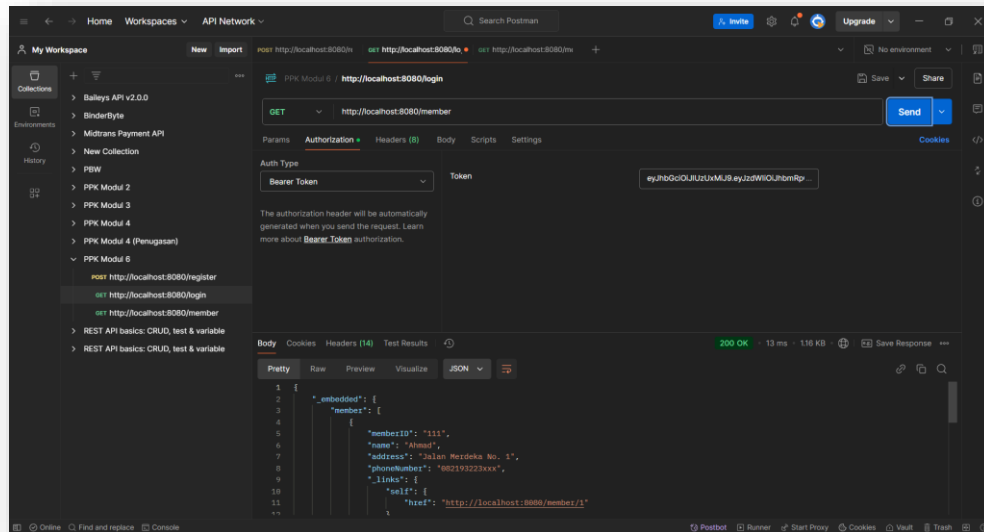
## Json Web Token (JWT)

[illegible]

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 3SI1 / 22



Buatlah penjelasan lengkap mengenai beberapa hal pada kegiatan praktikum di atas.

1) Jelaskan fungsi/tugas atau kegunaan masing-masing kelas yang dibuat pada praktikum ini!

**Jawab:**

a) Entity.User

- Berfungsi sebagai representasi tabel dalam database
- Menggunakan anotasi JPA untuk mapping ke database
- Mendefinisikan struktur data untuk user (id, name, email, password)

b) Repository.User

- Interface yang menyediakan operasi database untuk entity User
- Extends JpaRepository untuk mendapatkan operasi CRUD dasar
- Menambahkan method custom findByEmail untuk mencari user berdasarkan email

c) Dto.UserDto

- Data Transfer Object untuk memisahkan data yang ditransfer dari entity
- Implements UserDetails untuk kebutuhan Spring Security
- Menyediakan method-method keamanan seperti isEnabled(), getAuthorities(), dll

d) Mapper.UserMapper

- Berfungsi untuk mengkonversi antara entity dan DTO

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 3SI1 / 22

- Memiliki method `mapToUser()` untuk konversi dari DTO ke entity
  - Memiliki method `mapToUserDto()` untuk konversi dari entity ke DTO
- e) `Service.UserService`
- Mendefinisikan kontrak layanan untuk operasi user
  - Menyediakan method `createUser` dan `getUserByEmail`
- f) `Service.UserServiceImpl`
- Implementasi dari `UserService` interface
  - Menangani logika bisnis seperti enkripsi password
  - Menggunakan `UserRepository` untuk operasi database
- g) `Service.CustomUserDetail`
- Implementasi `UserDetailsService` untuk Spring Security
  - Menangani proses autentikasi user
  - Memuat data user untuk keperluan keamanan
- h) `Auth.JwtUtil`
- Utilitas untuk generate dan validasi JWT token
- i) `Auth.JwtFilter`
- Filter untuk memvalidasi JWT token pada setiap request
- j) `Auth.SecurityConfig`
- Konfigurasi keamanan aplikasi
  - Mendefinisikan aturan autentikasi dan otorisasi
  - Mengatur endpoint mana yang perlu autentikasi
- k) `Auth.AuthRequest`
- Model untuk request login/autentikasi
- l) `Auth.AuthResponse`
- Model untuk response setelah autentikasi berhasil
- m) `Auth.AuthController`
- Menangani endpoint `/login` dan `/register`
  - Mengatur proses autentikasi dan registrasi user
  - Menghasilkan JWT token saat login berhasil

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 3SI1 / 22

- 2) Jelaskan alur eksekusi endpoint /login dan proses otentikasi endpoint /member! Jelaskan kelas/objek/metode apa saja yang terlibat!

**Jawab:**

a) Endpoint /login

1. Request masuk melalui `AuthController` dalam bentuk `AuthRequest` (email dan password).
2. `AuthController` memanggil `AuthenticationManager` dan mengirimkan requestnya.
3. `AuthenticationManager` melakukan autentikasi dengan memanggil `CustomUserDetailsService` dan mengembalikan user yang berhasil diautentikasi.
4. Kelas `JwtUtil` men-generate token untuk user yang telah berhasil diautentikasi.
5. `AuthController` mengembalikan sebuah response berupa `AuthResponse` yang berisi email dan akses token.

b) Endpoint /member

1. Request masuk melalui `JwtFilter` untuk diperiksa headernya oleh function `hasAuthorizationBearer()`. Jika ada token, maka extract token tersebut.
2. `JwtUtil` memvalidasi token yang telah diextract oleh `JwtFilter` melalui function `validateAccessToken()`.
3. `setAuthenticationContext()` di `JwtFilter` dan disimpan sebagai autentikasi user.
4. `SecurityConfig` memeriksa apakah route tujuan memerlukan token atau tidak.
5. Request diteruskan ke `MemberController`.