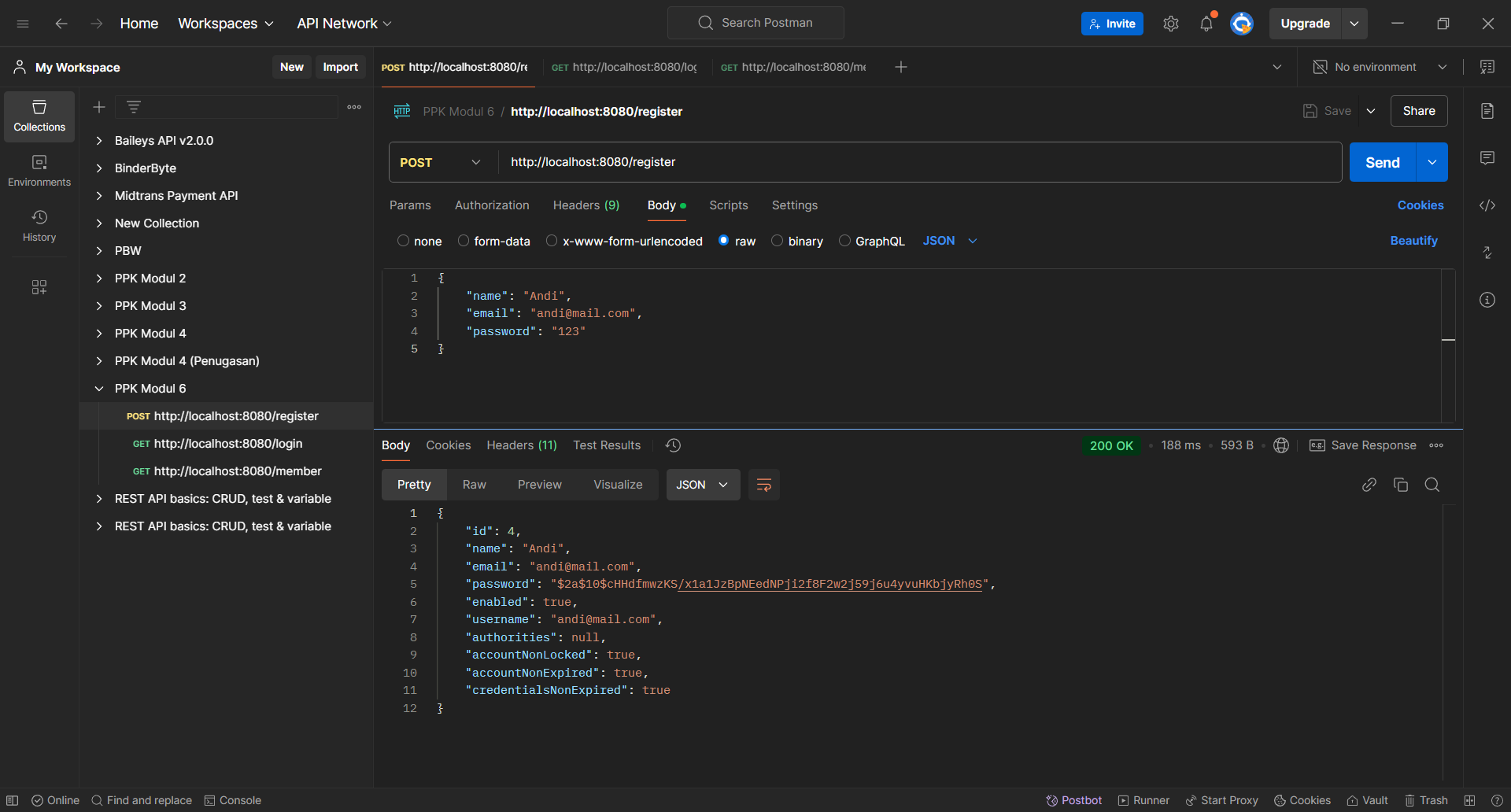
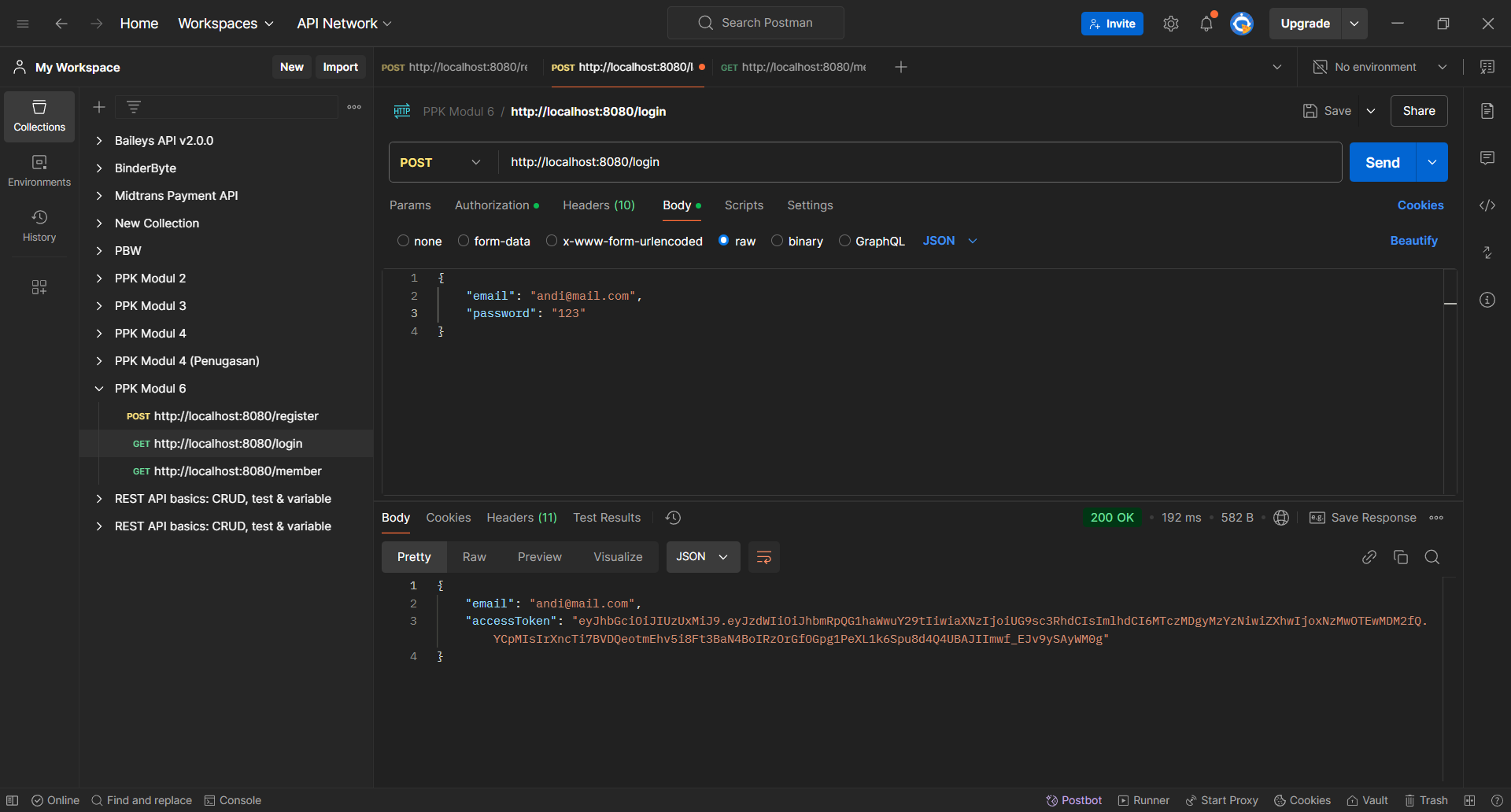
**LAPORAN PRAKTIKUM PEMROGRAMAN PLATFORM KHUSUS**

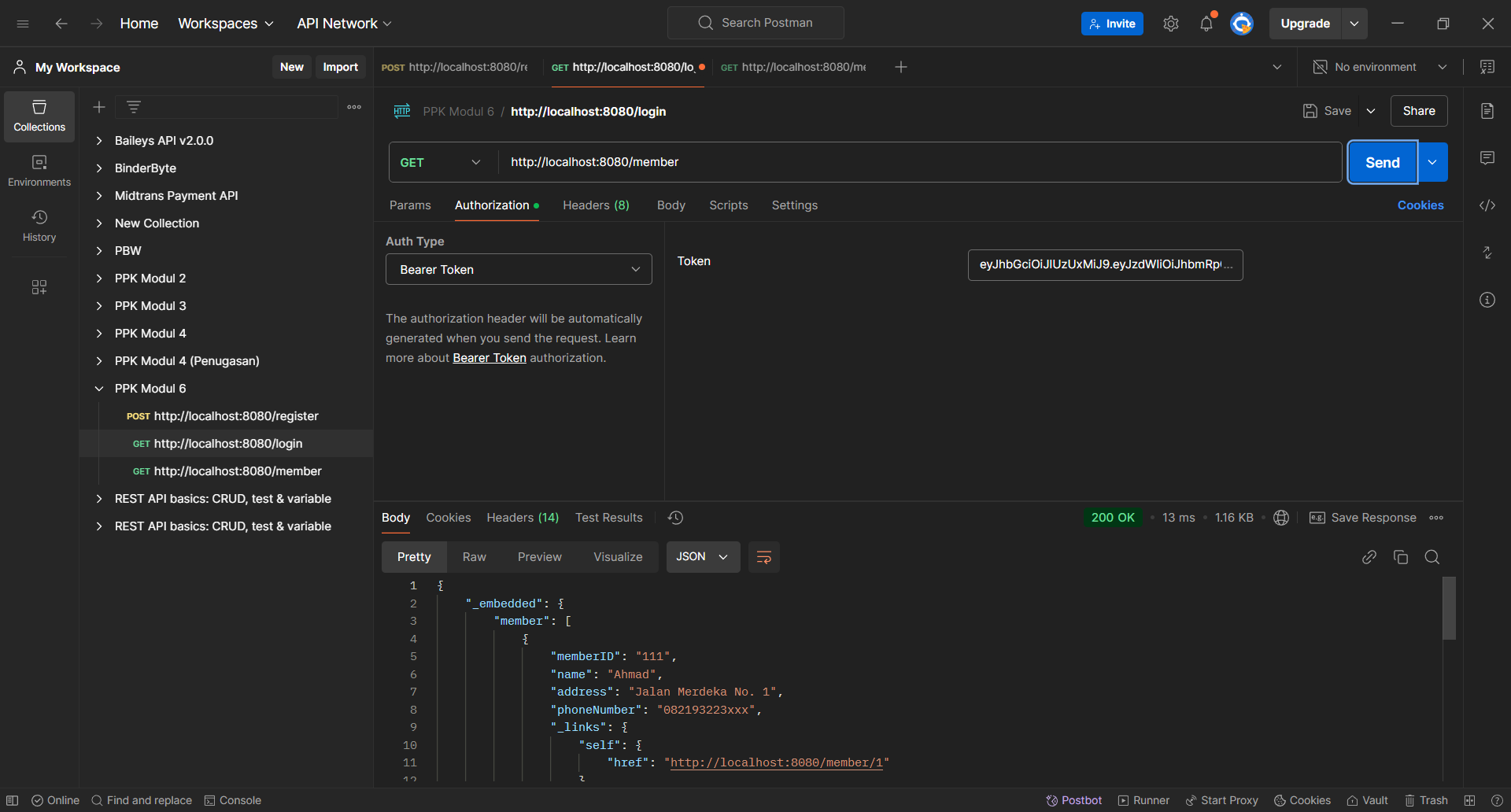
**MODUL 6**

**Json Web Token (JWT)**

Link Repository: <https://github.com/afrzl/Pemrograman-Platform-Khusus/tree/master/Pertemuan%206/perpustakaan>







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:**

1. Entity.User

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

1. 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

1. 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

1. Mapper.UserMapper

* Berfungsi untuk mengkonversi antara entity dan DTO
* Memiliki method mapToUser() untuk konversi dari DTO ke entity
* Memiliki method mapToUserDto() untuk konversi dari entity ke DTO

1. Service.UserService

* Mendefinisikan kontrak layanan untuk operasi user
* Menyediakan method createUser dan getUserByEmail

1. Service.UserServiceImpl

* Implementasi dari UserService interface
* Menangani logika bisnis seperti enkripsi password
* Menggunakan UserRepository untuk operasi database

1. Service.CustomUserDetail

* Implementasi UserDetailsService untuk Spring Security
* Menangani proses autentikasi user
* Memuat data user untuk keperluan keamanan

1. Auth.JwtUtil

* Utilitas untuk generate dan validasi JWT token

1. Auth.JwtFilter

* Filter untuk memvalidasi JWT token pada setiap request

1. Auth.SecurityConfig

* Konfigurasi keamanan aplikasi
* Mendefinisikan aturan autentikasi dan otorisasi
* Mengatur endpoint mana yang perlu autentikasi

1. Auth.AuthRequest

* Model untuk request login/autentikasi

1. Auth.AuthResponse

* Model untuk response setelah autentikasi berhasil

1. Auth.AuthController

* Menangani endpoint /login dan /register
* Mengatur proses autentikasi dan registrasi user
* Menghasilkan JWT token saat login berhasil

1. Jelaskan alur eksekusi endpoint /login dan proses otentikasi endpoint /member! Jelaskan kelas/objek/metode apa saja yang terlibat!

**Jawab:**

1. Endpoint /login
2. Request masuk melalui AuthController dalam bentuk AuthRequest (email dan password).
3. AuthController memanggil AuthenticationManager dan mengirimkan requestnya.
4. AuthenticationManager melakukan autentikasi dengan memanggil CustomUserDetailsService dan mengembalikan user yang berhasil diautentikasi.
5. Kelas JwtUtil men-generate token untuk user yang telah berhasil diautentikasi.
6. AuthController mengembalikan sebuah response berupa AuthResponse yang berisi email dan akses token.
7. Endpoint /member
8. Request masuk melalui JwtFilter untuk diperiksa headernya oleh function hasAuthorizationBearer(). Jika ada token, maka extract token tersebut.
9. JwtUtil memvalidasi token yang telah diextract oleh JwtFilter melalui function validateAccessToken().
10. setAuthenticationContext() di JwtFilter dan disimpan sebagai autentikasi user.
11. SecurityConfig memeriksa apakah route tujuan memerlukan token atau tidak.
12. Request diteruskan ke MemberController.