Praktikum 06 Pemrograman Platform Khusus

Muhammad Zacky Heta Warman / 222212775 / 3SI1 / 28

6.6 Penugasan

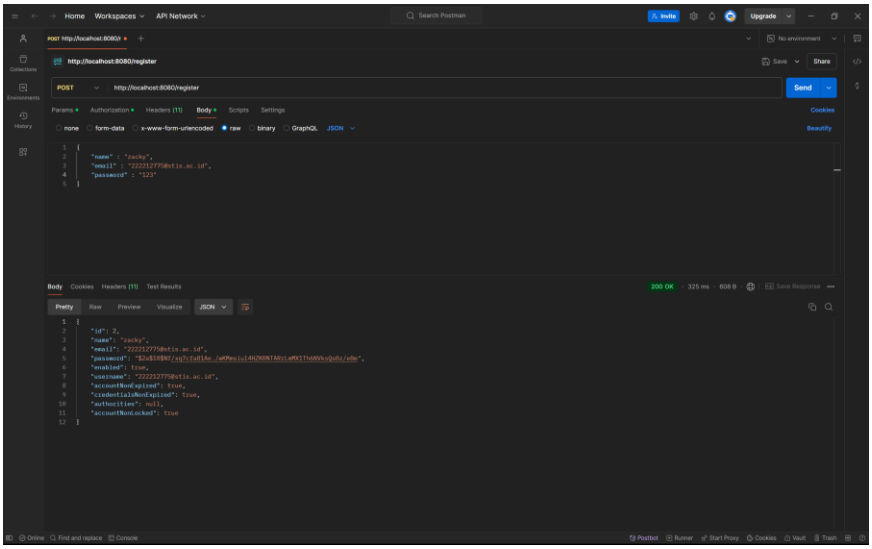
Buatlah penjelasan lengkap mengenai beberapa hal pada kegiatan praktikum di atas.

1) Jelaskan fungsi/tugas atau kegunaan masing-masing kelas yang dibuat pada praktikum ini!
2) Jelaskan alur eksekusi endpoint /login dan proses otentikasi endpoint /member! Jelaskan kelas/objek/metode apa saja yang terlibat!

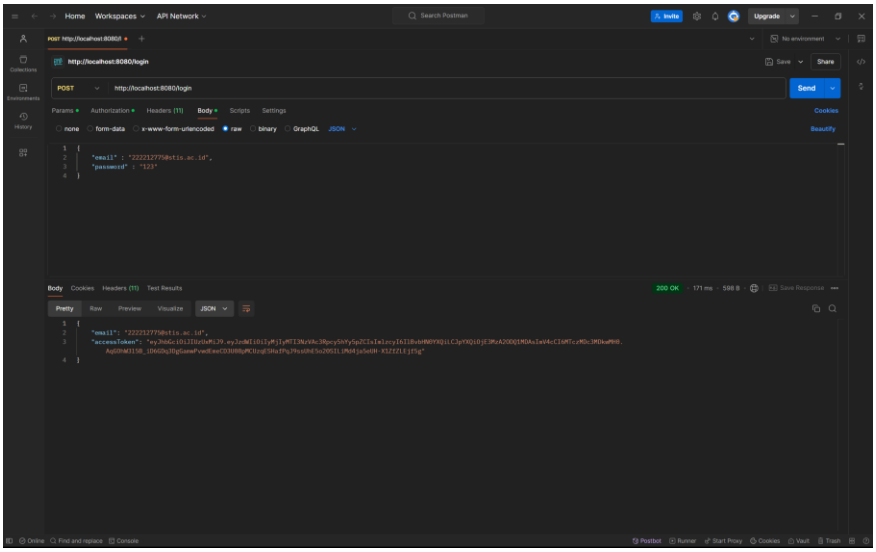
Jawab :

Sebelum menjawab pertanyaan 1 dan 2, berikut adalah tangkapan layar dari proses register, login, dan akses data member yang telah saya lakukan :

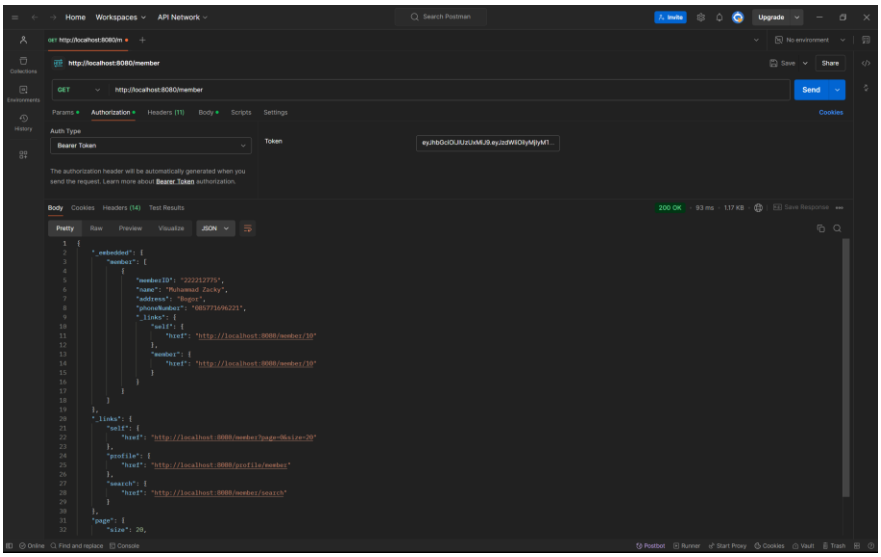
a. Register



b. Login



c. Akses data member



Berikut adalah penjelasan lengkap mengenai beberapa hal dari praktikum yang telah saya kerjakan, diantaranya yaitu :

1. Fungsi/tugas atau kegunaan masing-masing kelas yang dibuat pada praktikum ini :

- UserEntity : Merupakan kelas yang mewakili tabel users dalam database. Kelas ini menggunakan anotasi JPA untuk mendefinisikan kolom – kolom database, seperti id, name, email, dan password.
- UserRepository : Repository yang mengelola operasi CRUD untuk entitas User di database, seperti menyimpan pengguna baru dan menemukan pengguna berdasarkan email.
- UserDTO : Kelas DTO ini digunakan untuk mengirim atau menerima data pengguna anatara klien dan server. Biasanya berisikan atribut – atribut yang diperlukan untuk proses tertentu tanpa memuat keseluruhan detail dari UserEntity.
- UserMapper : Kelas ini menyediakan metode untuk mengonversi antara objek User dan UserDto, sehingga mempermudah transfer data antara layer entitas dan DTO.
- UserService : Antarmuka ini mendefinisikan operasi dasar untuk layanan pengguna, seperti createUser untuk membuat pengguna baru dan getUserByEmail untuk mendapatkan data pengguna berdasarkan email.
- UserServiceImpl : Implementasi dari UserService yang menggunakan UserRepository untuk operasi CRUD pengguna. createUser mengenkripsi password sebelum menyimpan pengguna ke database, dan getUserByEmail mengambil data pengguna berdasarkan email.
- CustomUserDetailsService : Implementasi dari UserDetailsService yang menyediakan detail pengguna berdasarkan email dari UserRepository. Digunakan oleh Spring Security untuk memuat detail pengguna selama proses otentikasi.
- JwtUtil : Kelas ini berfungsi untuk menghasilkan, memvalidasi, dan mengambil informasi dari token JWT. generateAccessToken digunakan untuk membuat token berdasarkan detail pengguna, sementara validateAccessToken memverifikasi apakah token valid atau tidak, dan getSubject mengembalikan subjek dari token (yaitu email pengguna).

- JwtFilter : Filter ini merupakan komponen OncePerRequestFilter yang digunakan untuk memverifikasi token JWT pada setiap permintaan setelah pengguna berhasil login. Jika token JWT valid, filter ini akan mengatur konteks otentikasi sehingga pengguna dapat mengakses endpoint yang memerlukan otorisasi.
- SecurityConfig : Konfigurasi keamanan yang mengatur akses endpoint, mekanisme otentikasi, serta filter otentikasi. Kelas ini mengatur kebijakan keamanan seperti menonaktifkan CSRF, membuat sesi menjadi stateless (tanpa menyimpan sesi), serta mengizinkan akses ke endpoint /login dan /register tanpa otentikasi.
- AuthRequest : DTO (Data Transfer Object) ini digunakan untuk menerima input dari pengguna pada endpoint /login, dengan atribut email dan password yang divalidasi agar tidak kosong dan memenuhi syarat tertentu.
- AuthResponse : DTO ini mengemas respons dari endpoint /login, mengirimkan kembali email pengguna dan accessToken (token JWT) yang dihasilkan jika otentikasi berhasil.
- AuthController : Kelas ini adalah controller yang mengelola endpoint otentikasi, yaitu endpoint /login dan /register. Endpoint /login digunakan untuk login, sementara /register digunakan untuk mendaftarkan pengguna baru. Pada /login, AuthController akan memverifikasi kredensial pengguna dan menghasilkan token JWT jika berhasil, sementara /register akan membuat pengguna baru menggunakan data yang diberikan.

2. Alur eksekusi endpoint /login dan proses otentikasi endpoint /member! Jelaskan kelas/objek/metode apa saja yang terlibat
   a. **Alur Eksekusi Endpoint /login**:
      - **Memanggil /login pada AuthController**: Pengguna mengirimkan request POST ke /login dengan email dan password di dalam AuthRequest.
      - **Autentikasi melalui AuthenticationManager**: AuthController menggunakan AuthenticationManager untuk melakukan autentikasi, dengan membuat objek UsernamePasswordAuthenticationToken dari email dan password pengguna yang dimasukkan. AuthenticationManager kemudian memanggil CustomUserDetailsService untuk memuat detail pengguna berdasarkan email.
      - **Validasi Password**: Jika pengguna ditemukan di database, AuthenticationManager mencocokkan password yang dikirimkan dengan password terenkripsi yang tersimpan di database menggunakan PasswordEncoder.
      - **Membuat Token JWT**: Jika autentikasi berhasil, JwtUtil menghasilkan token JWT yang berisi subjek (email pengguna) dan masa kedaluwarsa token.
      - **Mengembalikan Respons**: AuthController mengembalikan objek AuthResponse yang berisi email dan accessToken dalam respons HTTP.
   b. **Proses Otentikasi Endpoint /member**:
      - Mengirim Permintaan ke Endpoint /member: Pengguna mengakses endpoint /member setelah login. Token JWT yang dihasilkan selama login disertakan di header Authorization dalam bentuk Bearer <token>.
      - Validasi Token oleh JwtFilter: JwtFilter menangkap permintaan dan memeriksa header Authorization. Jika header valid dan token ada, JwtFilter memanggil JwtUtil untuk memvalidasi token.

- Verifikasi Token JWT: JwtUtil mengecek apakah token masih berlaku dan sesuai dengan SECRET_KEY. Jika token valid, JwtFilter menguraikan subjek dari token (email pengguna).
- Mengatur Konteks Otentikasi: JwtFilter mengatur konteks otentikasi pada SecurityContextHolder dengan UsernamePasswordAuthenticationToken, sehingga request memiliki status otentikasi sebagai pengguna yang sesuai.
- Akses ke Endpoint yang Diotorisasi: Setelah konteks otentikasi disiapkan, pengguna dapat mengakses endpoint /member atau endpoint lainnya yang memerlukan otentikasi.