Laporan Proyek — Implementasi JWT & RBAC (Java Spring)

1. Pendahuluan Proyek

Deskripsi Singkat Proyek

Proyek ini merupakan implementasi autentikasi menggunakan **JSON Web Token (JWT)** dan **Role-Based Access Control (RBAC)** pada aplikasi berbasis **Java Spring Boot**. Tujuan utama aplikasi adalah menyediakan mekanisme login aman, manajemen user, serta pembatasan akses berdasarkan peran (role) seperti *Admin* dan *User*.

Target pengguna proyek ini adalah developer atau instansi yang membutuhkan sistem autentikasi modern yang aman, terukur, dan sesuai standar industri. Proyek ini menyelesaikan masalah umum dalam pengembangan aplikasi yaitu pengamanan endpoint, manajemen hak akses, dan verifikasi identitas pengguna.

Lingkup Aplikasi

Lingkup pengembangan mencakup:

- Registrasi dan login menggunakan JWT
- Middleware untuk validasi token
- Role-Based Access Control (RBAC)
- Management user dasar

Spesifikasi teknis dasar:

- **Backend**: Java Spring Boot
- **Security**: Spring Security, JWT
- **Database**: MySQL
- **Testing**: Postman

---

2. Flow Sistem

Diagram Alur (Flowchart)

1. User melakukan login
2. Sistem melakukan validasi kredensial
3. Jika valid → sistem mengembalikan JWT
4. Setiap request selanjutnya harus membawa JWT
5. Sistem memvalidasi token → cek role
6. Jika role sesuai → akses diperbolehkan

Deskripsi Detail Flow

- **Alur Autentikasi**: User mengirim username & password → token dibuat → token dipakai untuk setiap request.
- **Alur Role**: Sistem membaca isi token → mengecek role → menentukan apakah endpoint dapat diakses.
- **Interaksi Pengguna**: User hanya dapat mengakses endpoint tertentu sesuai role.

---

3. Gambaran Keseluruhan Fitur

Fitur Utama

- Registrasi User
- Login + Generate JWT
- Validasi JWT otomatis
- Role-Based Access Control (RBAC)
- Endpoint spesifik Admin/User

Deskripsi Fitur

- **Register**: User baru dapat dibuat dengan role tertentu
- **Login**: Sistem mengeluarkan JWT
- **RBAC**: Admin dapat mengakses endpoint khusus Admin, User hanya endpoint tertentu
- **Token Filtering**: Sistem memeriksa token setiap request

Skalabilitas & Pengembangan Lanjutan

- Penambahan Refresh Token
- Multi-role dan permission lebih detail
- Audit log aktivitas user
- Integrasi Single Sign-On (SSO)

---

4. Struktur Database

ERD

Tabel utama: `user`, `role`, `user_roles`

Detail Struktur Tabel

Table: user

- id (int)
- username (varchar)
- password (varchar)
- enabled (boolean)

Table: role

- id (int)
- name (varchar) → ADMIN / USER

Relasi

- user ↔ role = many-to-many

Alasan Pemilihan Struktur

Struktur many-to-many dipilih karena user dapat memiliki lebih dari satu role dan dapat diskalakan untuk permission lebih lanjut.

---

5. Deskripsi Teknis Integrasi API

Spesifikasi API

- `POST /auth/register`
- `POST /auth/login`
- `GET /user/profile`
- `GET /admin/dashboard`

Format data: JSON

Autentikasi

Menggunakan **JWT Bearer Token**, dikirim melalui header:

```
Authorization: Bearer <token>
```

Error Handling

- 401: Token invalid/expired
- 403: Role tidak sesuai

- 400: Input tidak valid

Fallback: Mengembalikan pesan error standar JSON.

---

6. Hasil Pengujian dengan Berbagai Skenario

Metodologi Pengujian

- Integration Testing menggunakan Postman
- Pengujian skenario login, akses sesuai role, akses ditolak

Contoh Skenario

1. Login Berhasil

- Expected: Mendapatkan token
- Actual: Sesuai

2. Akses Endpoint Admin menggunakan User Role

- Expected: 403 Forbidden
- Actual: 403 Forbidden

3. Token Expired

- Expected: 401 Unauthorized
- Actual: 401 Unauthorized

Screenshot hasil pengujian dapat ditambahkan pada lampiran.

---

7. Panduan Penggunaan & Instalasi

Persiapan Lingkungan

- Install JDK 17+
- Install MySQL
- Setup database sesuai schema
- Configure application.yml

Menjalankan Aplikasi

```
mvnd spring-boot:run
```

Akses API (Testing dengan Postman)

- Import Collection
- Tes login → simpan token → gunakan untuk request lain

---

8. Kesimpulan & Rekomendasi

Ringkasan Pencapaian

Proyek berhasil mengimplementasikan sistem autentikasi berbasis JWT dan kontrol akses berbasis role.

Pembelajaran Teknis

- Spring Security advanced configuration
- JWT lifecycle
- Implementasi middleware security

Pengembangan Lanjutan

- Refresh Token
- Permission system (ACL)
- Monitoring login

---

9. Lampiran

- Screenshot Postman
- Log transaksi
- Contoh payload JWT