

**LAPORAN PRAKTIKUM MATA KULIAH
PEMROGRAMAN BERORIENTASI OBJEK**

APLIKASI MANAJEMEN DATA KOMIK (CRUD JPA + REPORTING)

Dosen Pengampu: Bayu Adhi Nugroho, Ph.D.



**UIN SUNAN AMPEL
S U R A B A Y A**

**Disusun oleh:
Husain Asrarillah
(09020624035)**

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA
TAHUN 2025**

BAB 1: PENDAHULUAN

1.1 Latar Belakang

Dalam pengelolaan data perpustakaan atau koleksi pribadi, hubungan antara buku (Komik) dan pembuatnya (Pengarang) seringkali bersifat kompleks. Satu komik bisa dikerjakan oleh banyak orang (misal: Penulis Cerita dan Ilustrator), dan satu pengarang bisa membuat banyak komik. Oleh karena itu, diperlukan sebuah sistem database relasional *Many-to-Many* dan aplikasi berbasis Desktop (Java Swing) untuk mengelolanya.

1.2 Tujuan

1. Memahami implementasi JPA (Java Persistence API) untuk koneksi database.
2. Menerapkan konsep CRUD (Create, Read, Update, Delete) pada relasi tabel *Many-to-Many*.
3. Membuat antarmuka pengguna (GUI) yang responsif menggunakan Java Swing dan JDialog.
4. Mengimplementasikan fitur tambahan seperti Login/Register, Upload/Download CSV, dan Cetak Laporan (JasperReport).

BAB 2: PERANCANGAN DATABASE

Aplikasi ini menggunakan database PostgreSQL. Skema database dirancang untuk menangani relasi *Many-to-Many* antara Komik dan Pengarang melalui tabel penghubung (Junction Table) bernama `detail_karya`.

2.1 Skema Tabel (ERD)

1. **Tabel login:** Menyimpan data admin (Username, Password, Security Question).
2. **Tabel komik:** Menyimpan data buku komik (Master Data).
3. **Tabel pengarang:** Menyimpan data author/mangaka (Master Data).
4. **Tabel detail_karya:** Tabel relasi yang menghubungkan Komik dan Pengarang beserta Peran spesifiknya (misal: *Story Writer*, *Illustrator*).

2.2 Script SQL

Berikut adalah *query* untuk pembentukan database:

SQL

-- 1. Tabel Login

```
CREATE TABLE login (  
    username VARCHAR(50) PRIMARY KEY,  
    password VARCHAR(255) NOT NULL,  
    full_name VARCHAR(100),  
    security_question VARCHAR(255),  
    security_answer VARCHAR(255)  
);
```

-- 2. Tabel Komik

```
CREATE TABLE komik (  
    komik_id SERIAL PRIMARY KEY,  
    judul VARCHAR(150) NOT NULL,  
    genre VARCHAR(50),  
    tahun_terbit INT,  
    harga DECIMAL(12, 2)  
);
```

-- 3. Tabel Pengarang

```
CREATE TABLE pengarang (  
    pengarang_id SERIAL PRIMARY KEY,  
    nama VARCHAR(100) NOT NULL,  
    negara VARCHAR(50),  
    status VARCHAR(20)  
);
```

-- 4. Tabel Detail Karya (Junction)

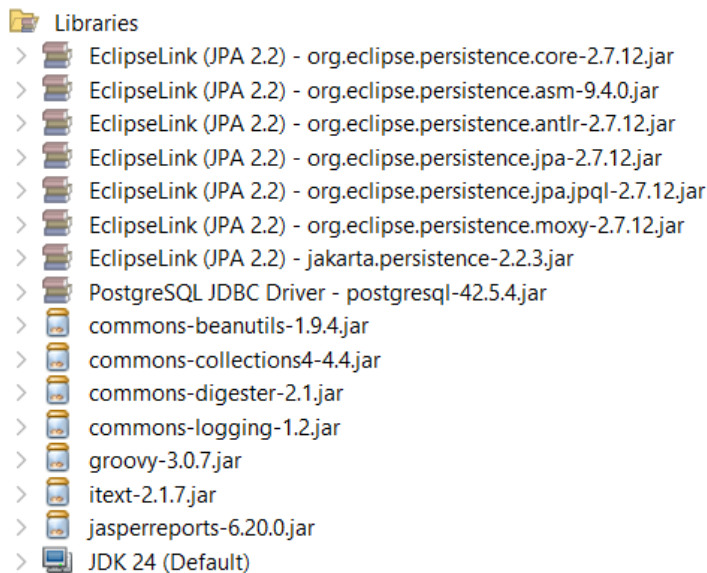
```
CREATE TABLE detail_karya (  
    id SERIAL PRIMARY KEY,  
    komik_id INT REFERENCES komik(komik_id) ON DELETE CASCADE,  
    pengarang_id INT REFERENCES pengarang(pengarang_id) ON DELETE CASCADE,  
    peran VARCHAR(50)  
);
```

BAB 3: LANGKAH-LANGKAH PEMBUATAN PROGRAM

Berikut adalah tahapan implementasi aplikasi menggunakan NetBeans IDE.

3.1 Persiapan Project & Library

1. **Buat Project Baru:** Java Application di NetBeans.
2. **Tambahkan Library (JAR):**
 - o postgresql-jdbc.jar (Driver Database).
 - o eclipselink.jar & javax.persistence.jar (Untuk JPA).
 - o jasperreports.jar & dependencies (Untuk fitur cetak).

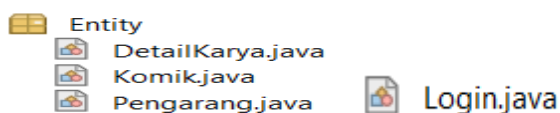


3. Konfigurasi persistence.xml:

- o Membuat file konfigurasi di folder META-INF.
- o Mengatur koneksi JDBC (URL: jdbc:postgresql://localhost:5432>NamaDB, User: postgres, Password: ...).

3.2 Pembuatan Entity Class (Mapping Database)

Menggunakan fitur *Entity Classes from Database* di NetBeans untuk men-generate kode Java secara otomatis dari tabel.



- **Komik.java & Pengarang.java:**

- Penting: Melakukan **Override method toString()** agar saat objek dimasukkan ke ComboBox, yang tampil adalah Judul>Nama, bukan alamat memori.

```
@Override
```

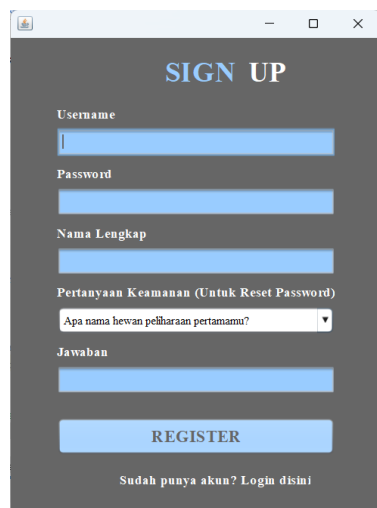
```
public String toString() { return this.judul; } // Di Class Komik
```

- **DetailKarya.java:** Memiliki relasi @ManyToOne ke Komik dan Pengarang.
- **Login.java:** Untuk mapping tabel user admin.

3.3 Implementasi Fitur Autentikasi (Login & Register)

1. Register Page (DaftarPage.java):

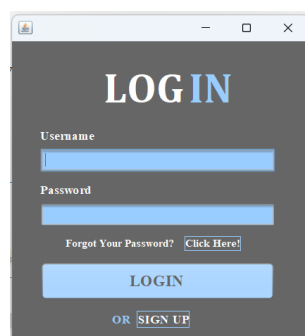
- Membuat form input Username, Password, dan **Pertanyaan Keamanan**.
- Logic: Mengecek apakah username sudah ada menggunakan em.find(). Jika belum, simpan data baru.



The screenshot shows a web browser window with a dark gray background and a light blue title bar. The page is titled "SIGN UP" in large, bold, light blue letters. Below the title, there are five input fields: "Username", "Password", "Nama Lengkap", "Pertanyaan Keamanan (Untuk Reset Password)" (which is a dropdown menu showing "Apa nama hewan peliharaan pertamamu?"), and "Jawaban". Each input field has a light blue border and a light blue background. Below the input fields is a light blue button labeled "REGISTER". At the bottom of the form, there is a link that says "Sudah punya akun? Login disini".

2. Login Page (LoginPage.java):

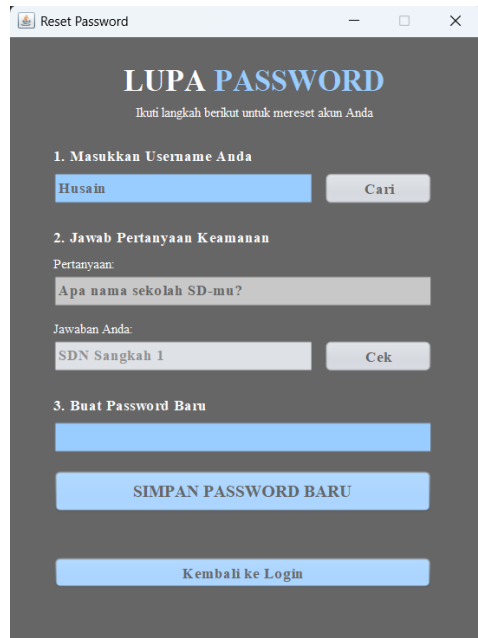
- Logic: Mencocokkan input username & password dengan data di database. Jika cocok, masuk ke MainPage.



The screenshot shows a web browser window with a dark gray background and a light blue title bar. The page is titled "LOGIN" in large, bold, light blue letters. Below the title, there are two input fields: "Username" and "Password". Each input field has a light blue border and a light blue background. Below the input fields is a light blue button labeled "LOGIN". Above the "LOGIN" button, there is a link that says "Forgot Your Password? Click Here!". Below the "LOGIN" button, there is a link that says "OR SIGN UP".

3. Lupa Password (ForgotPass.java):

- Fitur pemulihan akun. User memasukkan username -> Sistem menampilkan pertanyaan keamanan -> User menjawab. Jika benar, user bisa mereset password.



3.4 Implementasi Dialog Form (Reusable Dialog)

Untuk efisiensi kode, form input (Insert/Update) dibuat menggunakan **JDialog** dan didesain agar bisa digunakan ulang.



- **Strategi:** Satu file Dialog menangani Insert dan Edit sekaligus.
- **Logika:**
 - Jika parameter objek null maka Mode **TAMBAH** (Form kosong).

- Jika parameter objek ada isinya maka Mode **EDIT** (Form terisi otomatis).
- **File Dialog:**
 1. KomikDialog.java: Input Judul, Genre, Tahun, Harga.
 2. PengarangDialog.java: Input Nama, Negara, Status.
 3. InputDetailDialog.java: Menggunakan **ComboBox** untuk memilih Komik dan Pengarang, serta TextField untuk input Peran.

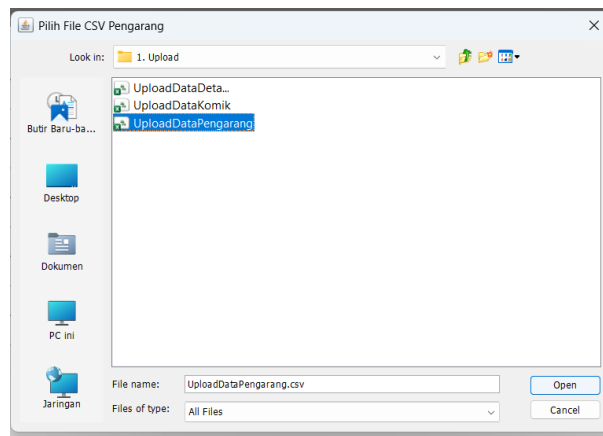
3.5 Implementasi Halaman Utama (MainPage.java)

Halaman ini adalah pusat kontrol aplikasi (Dashboard).

1. **Desain Layout:**
 - Menggunakan **JTabbedPane** untuk memisahkan tab "Data Komik", "Data Pengarang", dan "Detail Karya".
 - Menggunakan **BorderLayout** agar tabel memenuhi layar secara responsif.
2. **Fitur CRUD (Create, Read, Update, Delete):**
 - **Read:** Menggunakan DefaultTableModel dan query JPA (createNamedQuery atau createQuery) untuk menampilkan data ke JTable.
 - **Create & Update:** Memanggil JDialog yang telah dibuat sebelumnya.
 - **Delete:** Menghapus data berdasarkan ID yang dipilih di tabel menggunakan em.remove().
3. **Fitur Refresh Otomatis:** Menambahkan ChangeListener pada TabbedPane agar saat pindah tab, data (terutama ComboBox) diperbarui otomatis.

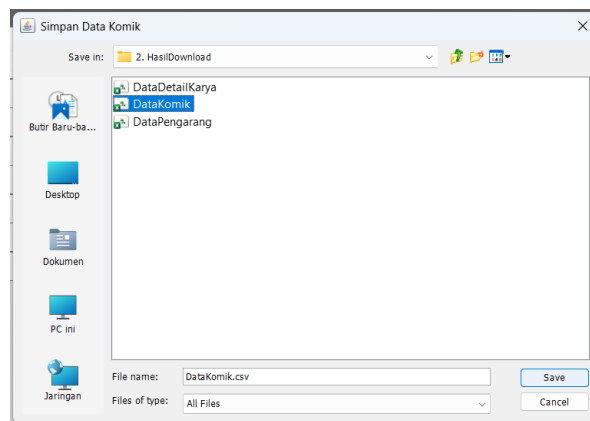
3.6 Implementasi Fitur Tambahan (Tools)

1. **Upload CSV (Import):**
 - Membaca file .csv menggunakan BufferedReader.
 - Memparsing data (dipisah koma), lalu melakukan query pencarian (findByJudul) untuk memastikan relasi data valid sebelum disimpan.



2. Download CSV (Export):

- Mengambil list data dari database, lalu menulisnya ke file teks dengan format CSV menggunakan PrintWriter.



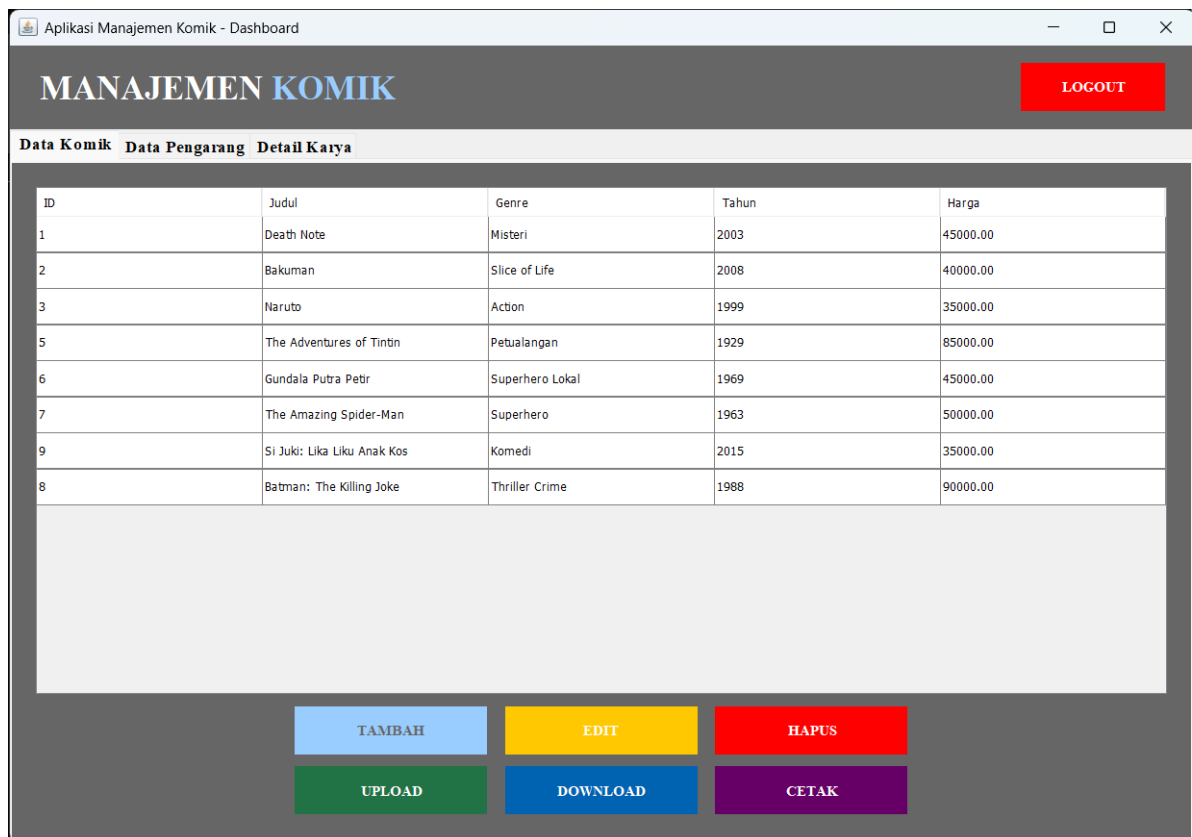
3. Cetak Laporan (JasperReport):

- Mendesain layout laporan (.jrxml) di Jaspersoft Studio.
- Di Java, menggunakan JasperFillManager untuk mengisi laporan dengan data dari koneksi JDBC dan menampilkannya dengan JasperViewer.

PENGARANG			
Data Pengarang			
pengarang_id	nama	negara	status
1	Tsugumi Ohba	Jepang	Aktif
2	Takeshi Obata	Jepang	Aktif
3	Masashi Kishimoto	Jepang	Hiatus
6	Brian Bolland	Inggris	Aktif
7	Steve Ditko	Amerika Serikat	Almarhum
8	Faza Meonk	Indonesia	Aktif
9	Stan Lee	Amerika Serikat	Almarhum
10	Hasmi	Indonesia	Almarhum
11	Alan Moore	Inggris	Aktif
5	Herge	Belgia	Almarhum

BAB 4: HASIL DAN PEMBAHASAN

4.1 Tampilan Aplikasi



1. **Halaman Login & Register:** Tampilan dengan tema gelap (Dark Mode) dan validasi input.
2. **Dashboard Utama:** Menampilkan tabel data dengan tombol aksi yang presisi di tengah.
3. **Dialog Input:** Form pop-up modal untuk menambah dan mengedit data.

4.2 Analisis Logika *Many-to-Many*

Pada tab "Detail Karya", aplikasi berhasil menghubungkan satu komik dengan banyak pengarang.

- *Contoh:* Komik "Naruto" memiliki dua entri di tabel detail: satu sebagai *Story Writer* dan satu sebagai *Illustrator*, meskipun pengarangnya sama (Masashi Kishimoto). Ini memenuhi kaidah Normalisasi Database (1NF).

BAB 5: KESIMPULAN

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. Penggunaan **JPA** sangat memudahkan proses manipulasi database dibandingkan JDBC konvensional karena kita bekerja dengan Objek (Entity), bukan query SQL mentah yang panjang.
2. Desain **GUI dengan JDialog** terpisah membuat kode program lebih bersih, terstruktur, dan mudah dirawat (*maintainable*).
3. Struktur database dengan tabel **Junction (Detail Karya)** adalah solusi paling efektif untuk menangani hubungan kompleks seperti Komik dan Pengarang, memungkinkan pencatatan peran yang spesifik.
4. Integrasi fitur eksternal seperti manipulasi file CSV dan JasperReport meningkatkan nilai guna aplikasi menjadi setara dengan aplikasi manajemen profesional.