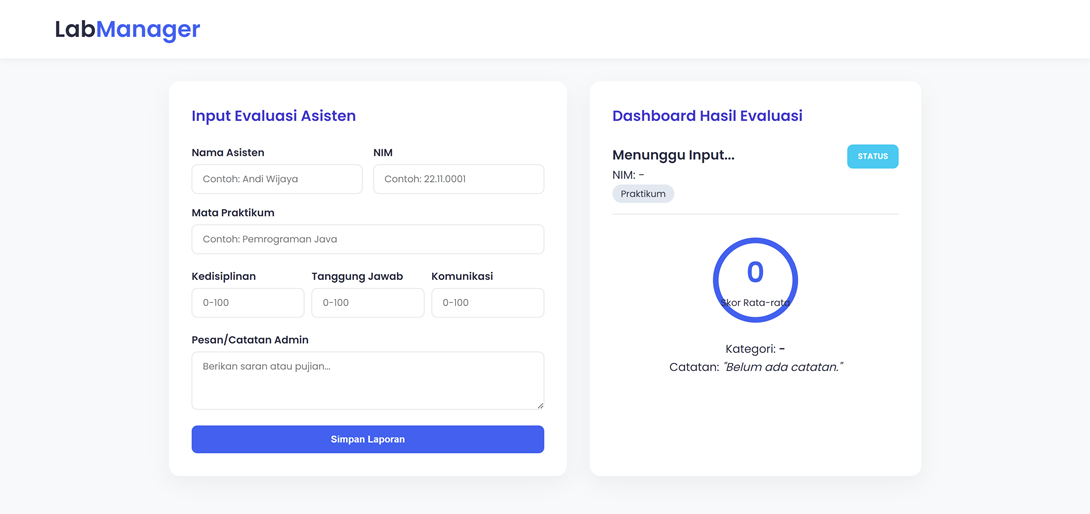
**TUGAS PROJEK**

SISTEM PENGELOLAAN PENUGASAN DAN EVALUASI KINERJA ASISTEN PRAKTIKUM

Disusun untuk memenuhi tugas projek Pemrograman Berorientasi Objek

Dosen Pengampu : Ali Tarmuji, S.T., M.Cs.



**Disusun oleh:**

**NAMA : Sheilla Rizki Fadillah**

**NIM : 2400018115**

**KELAS : B**

**LINK GITHUB :**

[**https://github.com/Sheilla1812/PBO2025-TP02-SHEILLA\_RIZKI\_FADILLAH-2400018115**](https://github.com/Sheilla1812/PBO2025-TP02-SHEILLA_RIZKI_FADILLAH-2400018115)

**PROGRAM STUDI S1 INFORMATIKA**

**FAKULTAS TEKNOLOGI INDUSTRI**

**UNIVERSITAS AHMAD DAHLAN**

**2025/2026**

# PENDAHULUAN

**1.1 Latar Belakang**

Asisten praktikum memiliki peran penting dalam menunjang kegiatan praktikum di perguruan tinggi, seperti membantu dosen pengampu, membimbing mahasiswa, serta menjaga kelancaran pelaksanaan praktikum di laboratorium. Seiring bertambahnya jumlah kelas praktikum dan asisten yang terlibat, pengelolaan penugasan dan evaluasi kinerja asisten praktikum menjadi semakin kompleks.

Pada praktiknya, proses pengelolaan penugasan dan penilaian kinerja asisten praktikum masih sering dilakukan secara manual, seperti melalui pesan singkat, dokumen terpisah, atau pencatatan sederhana. Hal ini berpotensi menimbulkan berbagai permasalahan, seperti data yang tidak terorganisir dengan baik, kesulitan dalam memantau penyelesaian tugas, serta kurangnya objektivitas dan transparansi dalam proses evaluasi kinerja.

Berdasarkan permasalahan tersebut, diperlukan sebuah sistem terkomputerisasi yang mampu mengelola penugasan dan evaluasi kinerja asisten praktikum secara terintegrasi. Oleh karena itu, pada tugas akhir ini dikembangkan **Sistem Pengelolaan Penugasan dan Evaluasi Kinerja Asisten Praktikum** yang diharapkan dapat meningkatkan efisiensi pengelolaan praktikum serta membantu dosen dan koordinator praktikum dalam melakukan evaluasi secara sistematis dan akurat.

**1.2 Rumusan Masalah**

Berdasarkan latar belakang yang telah diuraikan, maka rumusan masalah dalam tugas akhir ini adalah sebagai berikut:

1. Bagaimana merancang dan membangun sistem pengelolaan penugasan asisten praktikum yang terstruktur dan terintegrasi?
2. Bagaimana sistem dapat membantu proses evaluasi kinerja asisten praktikum secara objektif dan terdokumentasi?
3. Bagaimana sistem dapat menyajikan laporan penugasan dan kinerja asisten praktikum secara akurat dan mudah diakses?

**1.3 Tujuan Penelitian**

Tujuan dari penyusunan tugas akhir ini adalah:

1. Merancang dan membangun Sistem Pengelolaan Penugasan dan Evaluasi Kinerja Asisten Praktikum.
2. Mempermudah proses pemberian dan pemantauan penugasan asisten praktikum.
3. Membantu dosen dan koordinator praktikum dalam melakukan evaluasi kinerja asisten praktikum secara sistematis.
4. Menghasilkan laporan kinerja asisten praktikum yang akurat dan terdokumentasi.

**1.4 Manfaat Penelitian**

Adapun manfaat yang diharapkan dari tugas akhir ini adalah sebagai berikut:

1. **Bagi Dosen dan Koordinator Praktikum**  
   Membantu dalam pengelolaan penugasan dan evaluasi kinerja asisten praktikum secara lebih efisien dan terorganisir.
2. **Bagi Asisten Praktikum**  
   Memudahkan asisten dalam menerima tugas, mengumpulkan hasil pekerjaan, serta mengetahui hasil evaluasi kinerjanya.
3. **Bagi Institusi**  
   Menjadi sarana pendukung dalam meningkatkan kualitas pengelolaan kegiatan praktikum di perguruan tinggi.

**1.5 Batasan Masalah**

Untuk membatasi ruang lingkup pembahasan, maka tugas akhir ini memiliki batasan masalah sebagai berikut:

1. Sistem hanya digunakan untuk pengelolaan penugasan dan evaluasi kinerja asisten praktikum.
2. Pengguna sistem dibatasi pada admin/koordinator praktikum, dosen pengampu, dan asisten praktikum.
3. Evaluasi kinerja asisten praktikum dilakukan berdasarkan kriteria yang ditentukan oleh dosen atau koordinator praktikum.
4. Sistem tidak membahas aspek keuangan atau penggajian asisten praktikum.

#### A. RUANG LINGKUP

Ruang lingkup aplikasi **Sistem Pengelolaan Penugasan dan Evaluasi Kinerja Asisten Praktikum** meliputi pengelolaan data pengguna (koordinator praktikum, dosen, dan asisten), pengelolaan mata praktikum dan kelas, pemberian serta pemantauan penugasan asisten praktikum, pengumpulan dan penyimpanan hasil tugas, evaluasi kinerja asisten praktikum, serta penyajian laporan dan rekapitulasi kinerja sebagai bahan evaluasi dan pengambilan keputusan.

#### B. RANCANGAN APLIKASI

***1. Rancangan Struktur Program (Class Diagram)***

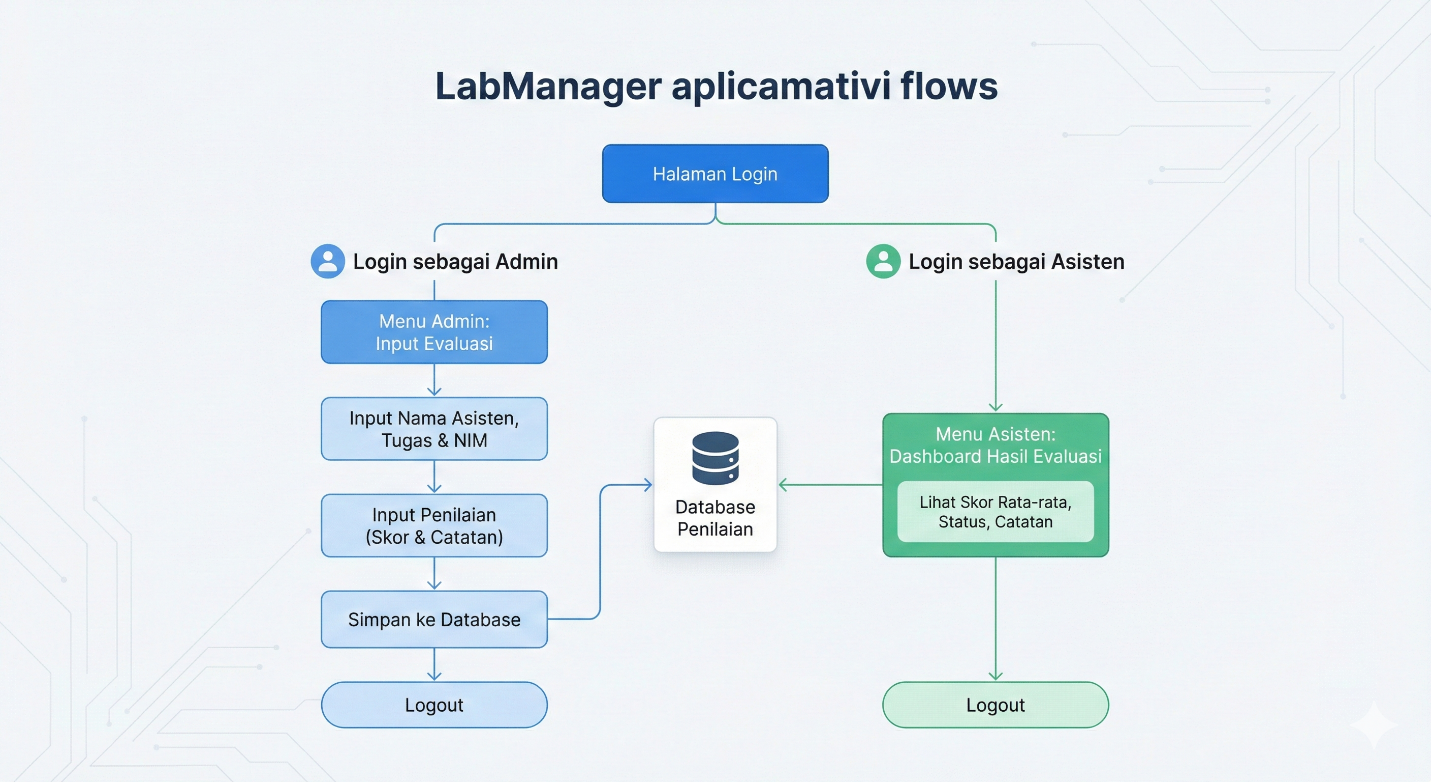


Diagram ini memvisualisasikan perjalanan pengguna (user journey) dari saat mereka masuk ke dalam aplikasi hingga menyelesaikan tugas mereka sesuai dengan hak akses masing-masing.

1. Titik Awal: Login dan Autentikasi

* Halaman Login: Semua pengguna, baik Admin maupun Asisten, memulai dari halaman login yang sama.
* Proses Autentikasi: Pengguna memasukkan kredensial mereka (misalnya, username/email dan password). Siste kemudian memverifikasi kredensial tersebut.
* Cek Role (Percabangan): Setelah verifikasi berhasil, sistem akan menentukan peran (role) dari akun tersebut. Inilah titik percabangan utama:
  + Jika perannya adalah Admin, pengguna diarahkan ke Alur Admin.
  + Jika perannya adalah Asisten, pengguna diarahkan ke Alur Asisten.

2. Alur Admin (Proses Input & Penilaian)

Alur ini berfokus pada pembuatan data (Create/Write). Admin memiliki hak penuh untuk memasukkan dan menyimpan data evaluasi.

1. Halaman Input Evaluasi Asisten: Admin diarahkan ke halaman dengan formulir kosong di sebelah kiri (seperti pada gambar antarmuka).
2. Input Identitas & Tugas: Admin mengisi data identitas asisten yang akan dievaluasi:
   * Nama Asisten
   * NIM (Nomor Induk Mahasiswa)
   * Mata Praktikum yang ditugaskan
3. Input Penilaian Kuantitatif: Admin memberikan skor angka (range 0-100) untuk tiga kriteria kinerja:
   * Kedisiplinan
   * Tanggung Jawab
   * Komunikasi
4. Input Catatan Kualitatif: Admin menuliskan pesan, saran, atau pujian di kolom "Pesan/Catatan Admin".
5. Aksi Simpan Laporan: Setelah semua data terisi, Admin mengklik tombol "Simpan Laporan".
6. Proses Sistem (Di Latar Belakang):
   * Kalkulasi: Sistem menerima tiga nilai kuantitatif tadi dan menghitung Skor Rata-rata. Sistem juga menentukan Kategori penilaian berdasarkan skor tersebut (misalnya: Sangat Baik, Baik, Cukup, Kurang).
   * Penyimpanan Database: Sistem menyimpan seluruh data (Identitas, Nilai per Kriteria, Skor Rata-rata, Kategori, Catatan Admin, dan Status "Selesai") ke dalam Database.
7. Feedback & Selesai: Sistem menampilkan notifikasi sukses kepada Admin (misalnya, "Laporan berhasil disimpan"). Alur Admin selesai untuk satu sesi evaluasi ini.

3. Alur Asisten (Proses Melihat Hasil)

Alur ini berfokus pada melihat data (Read-only). Asisten hanya dapat melihat hasil evaluasi yang telah diinput oleh Admin.

1. Halaman Dashboard Hasil Evaluasi: Asisten diarahkan ke halaman dashboard di sebelah kanan (seperti pada gambar antarmuka).
2. Proses Sistem (Pengambilan Data):
   * Segera setelah halaman dimuat, sistem secara otomatis mengambil (fetch) data penilaian terbaru yang terkait dengan akun Asisten tersebut dari Database.
3. Melihat Hasil: Data yang diambil kemudian ditampilkan di dashboard. Asisten dapat melihat:
   * Skor Rata-rata: Ditampilkan secara visual dalam lingkaran besar.
   * Status: (Misalnya, "Selesai" atau "Menunggu Input" jika belum dinilai).
   * Kategori: Kategori kinerja berdasarkan skor rata-rata.
   * Catatan: Pesan atau saran yang ditulis oleh Admin.
4. Selesai: Asisten tidak memiliki opsi untuk mengubah atau menginput data apa pun di halaman ini. Alur Asisten selesai setelah mereka melihat informasi yang dibutuhkan.

***2. Sketsa Antarmuka Pengguna (UI Mockup)***

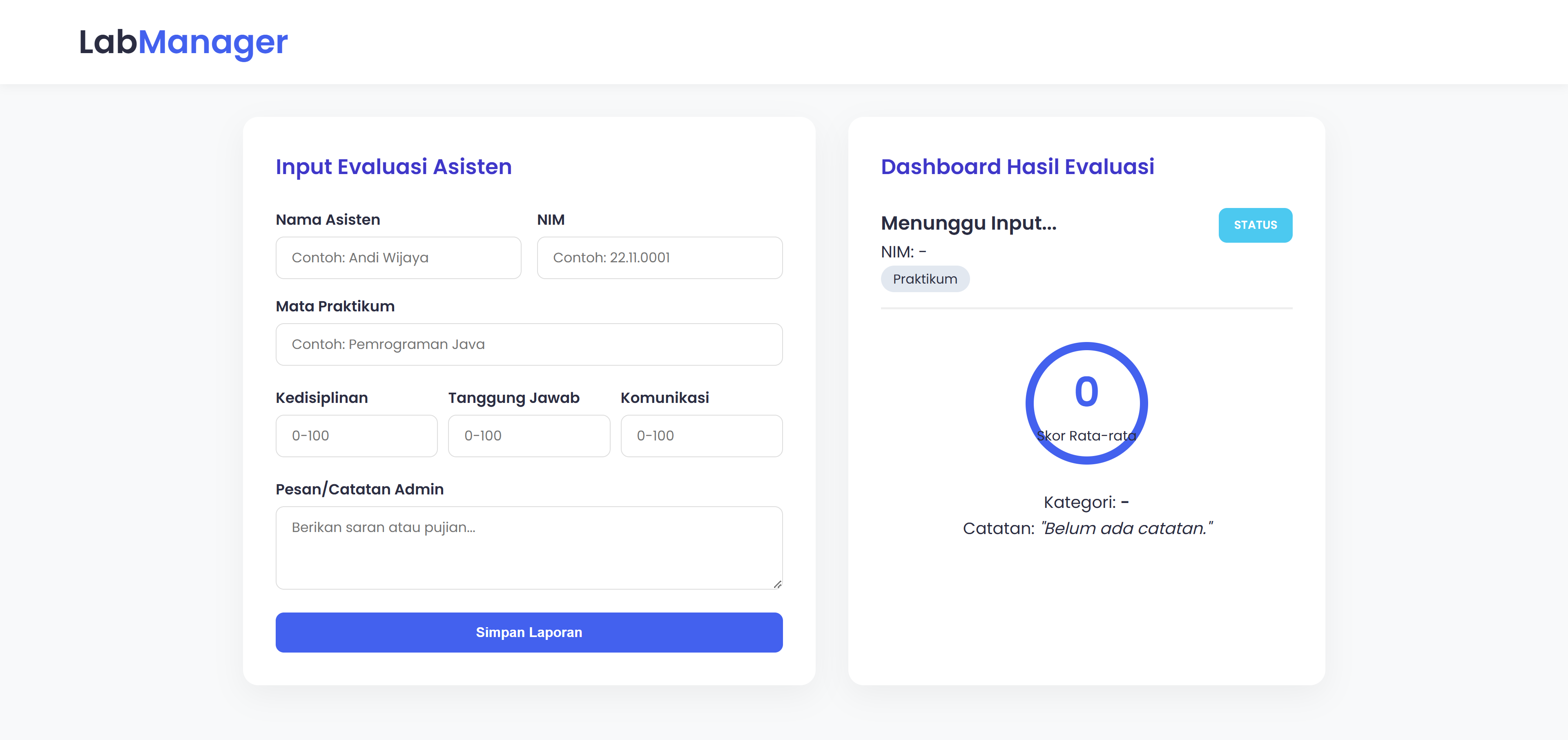
Berikut adalah visualisasi bagaimana data diinput dan ditampilkan pada layar konsol sesuai dengan alur di Main.java :

1. Input Section: Pengguna memasukkan data teks untuk identitas dan angka untuk penilaian (0-100).
2. Processing: Program menghitung rata-rata dari tiga nilai input tersebut.
3. Result Section: Program menampilkan kategori ("Sangat Baik", "Baik", atau "Perlu Evaluasi") serta status keberlanjutan asisten.

***3. Logika Penilaian***

|  |  |  |
| --- | --- | --- |
| Nilai Rata-rata | Kategori | Status di Main |
| ≥85 | Sangat Baik | Layak Dipertahankan |
| 70−84 | Baik | Layak Dipertahankan |
| <70 | Perlu Evaluasi | Perlu Pembinaan |

*4. Desain Antarmuka*



C. KODE PROGRAM

##### Asisten.java

public class Asisten {

private String nama;

private String nim;

private String praktikum;

public Asisten(String nama, String nim, String praktikum) {

this.nama = nama;

this.nim = nim;

this.praktikum = praktikum;

}

public String getNama() { return nama; }

public String getNim() { return nim; }

public String getPraktikum() { return praktikum; }

}

###### Tugas.java

public class Tugas {

private String deskripsi;

public Tugas(String deskripsi) {

this.deskripsi = deskripsi;

}

public String getDeskripsi() {

return "- " + deskripsi;

}

}

###### Evaluasi.java

public class Evaluasi {

protected int nilai;

public int hitungNilai() {

return nilai;

}

public String kategori() {

return "Belum dievaluasi";

}

}

###### EvaluasiKinerja.java

public class EvaluasiKinerja extends Evaluasi {

private int disiplin;

private int tanggungJawab;

private int komunikasi;

public EvaluasiKinerja(int disiplin, int tanggungJawab, int komunikasi) {

this.disiplin = disiplin;

this.tanggungJawab = tanggungJawab;

this.komunikasi = komunikasi;

}

@Override

public int hitungNilai() {

nilai = (disiplin + tanggungJawab + komunikasi) / 3;

return nilai;

}

@Override

public String kategori() {

if (nilai >= 85) return "Sangat Baik";

else if (nilai >= 70) return "Baik";

else return "Perlu Evaluasi";

}

public String getStatusKelayakan() {

return (nilai >= 70) ? "LAYAK DIPERTAHANKAN" : "PERLU PEMBINAAN";

}

}

###### Main.java

import java.util.Scanner;

import java.io.\*;

import java.util.ArrayList;

public class Main {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

System.out.println("===== SISTEM PENGELOLAAN ASISTEN =====");

System.out.print("Login Sebagai (admin/asisten): ");

String role = input.nextLine();

if (role.equalsIgnoreCase("admin")) {

System.out.print("Masukkan Password Admin: ");

String pass = input.nextLine();

if (pass.equals("admin123")) {

menuAdmin(input);

} else {

System.out.println("Password Salah!");

}

} else if (role.equalsIgnoreCase("asisten")) {

menuAsisten();

} else {

System.out.println("Role tidak dikenali.");

}

System.out.println("\n===== PROSES SELESAI =====");

input.close();

}

public static void menuAdmin(Scanner input) {

System.out.println("\n--- MODE INPUT DATA (ADMIN) ---");

System.out.print("Nama Asisten : ");

String nama = input.nextLine();

System.out.print("NIM : ");

String nim = input.nextLine();

System.out.print("Mata Praktikum : ");

String praktikum = input.nextLine();

// Input Tugas

System.out.print("Jumlah Tugas yang diselesaikan: ");

int jml = input.nextInt();

input.nextLine();

ArrayList<String> listTugas = new ArrayList<>();

for (int i = 0; i < jml; i++) {

System.out.print("Tugas ke-" + (i + 1) + ": ");

listTugas.add(input.nextLine());

}

// Input Nilai

System.out.println("\n--- INPUT NILAI KINERJA ---");

System.out.print("Nilai Kedisiplinan : ");

int d = input.nextInt();

System.out.print("Nilai Tanggung Jawab : ");

int tj = input.nextInt();

System.out.print("Nilai Komunikasi : ");

int k = input.nextInt();

input.nextLine();

System.out.print("Pesan/Catatan Admin : ");

String pesan = input.nextLine();

// Hitung Logika

EvaluasiKinerja ev = new EvaluasiKinerja(d, tj, k);

int skor = ev.hitungNilai();

String kat = ev.kategori();

String status = ev.getStatusKelayakan();

// Simpan ke File

try (BufferedWriter writer = new BufferedWriter(new FileWriter("kinerja.txt"))) {

writer.write("========== LAPORAN EVALUASI ASISTEN ==========\n");

writer.write("Nama : " + nama + "\n");

writer.write("NIM : " + nim + "\n");

writer.write("Praktikum : " + praktikum + "\n");

writer.write("----------------------------------------------\n");

writer.write("DAFTAR TUGAS:\n");

for (String t : listTugas) {

writer.write("- " + t + "\n");

}

writer.write("----------------------------------------------\n");

writer.write("HASIL PENILAIAN:\n");

writer.write("Rata-rata Nilai : " + skor + "\n");

writer.write("Kategori : " + kat + "\n");

writer.write("STATUS AKHIR : " + status + "\n");

writer.write("PESAN ADMIN : " + pesan + "\n");

writer.write("==============================================");

System.out.println("\n[Berhasil] Data disimpan ke kinerja.txt");

} catch (IOException e) {

System.out.println("Gagal menyimpan data: " + e.getMessage());

}

}

public static void menuAsisten() {

System.out.println("\n--- MODE LIHAT NILAI (ASISTEN) ---");

File file = new File("kinerja.txt");

if (!file.exists()) {

System.out.println("Maaf, Admin belum menginputkan evaluasi Anda.");

return;

}

try (BufferedReader reader = new BufferedReader(new FileReader(file))) {

String line;

while ((line = reader.readLine()) != null) {

System.out.println(line);

}

} catch (IOException e) {

System.out.println("Gagal membaca file: " + e.getMessage());

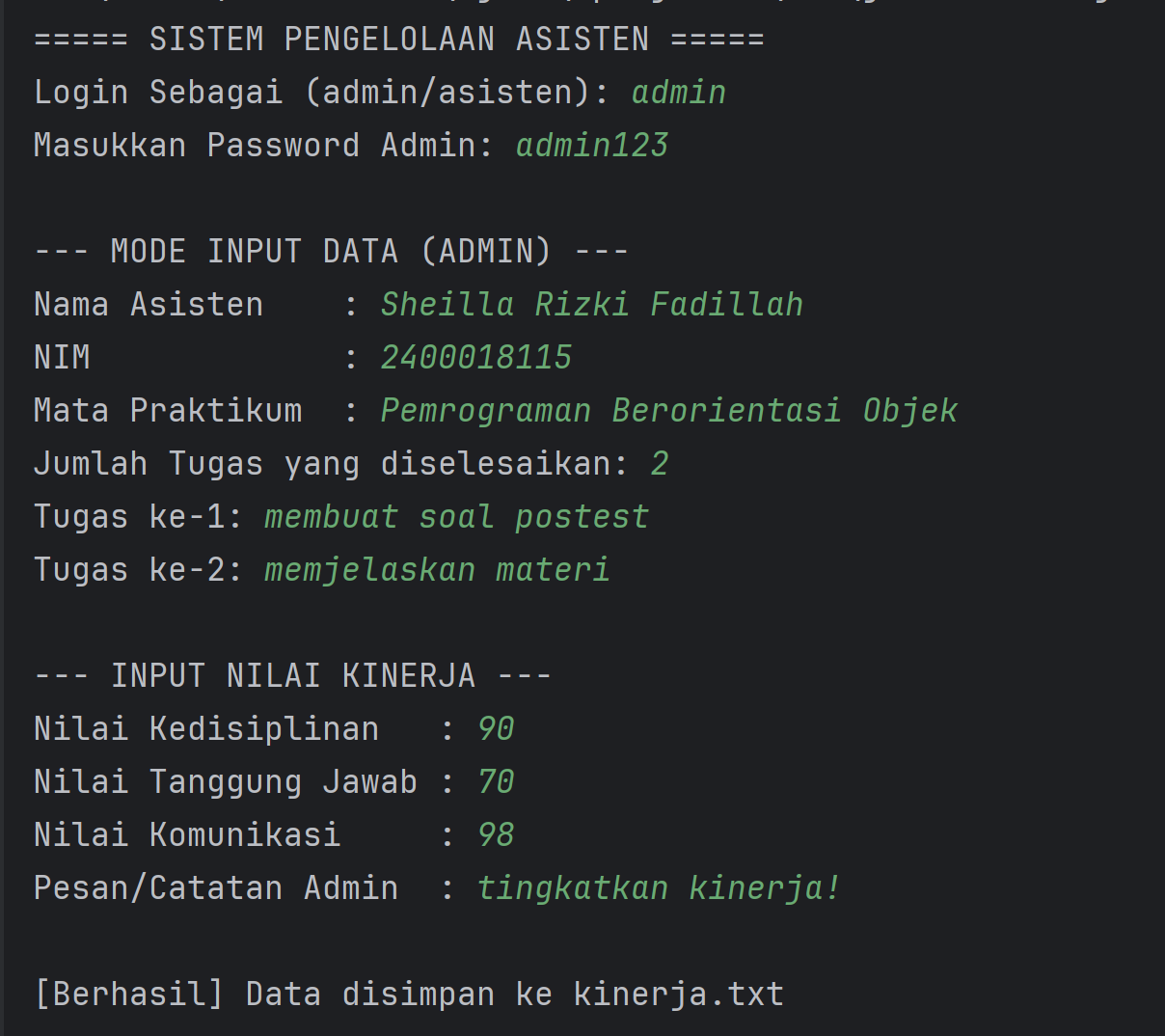
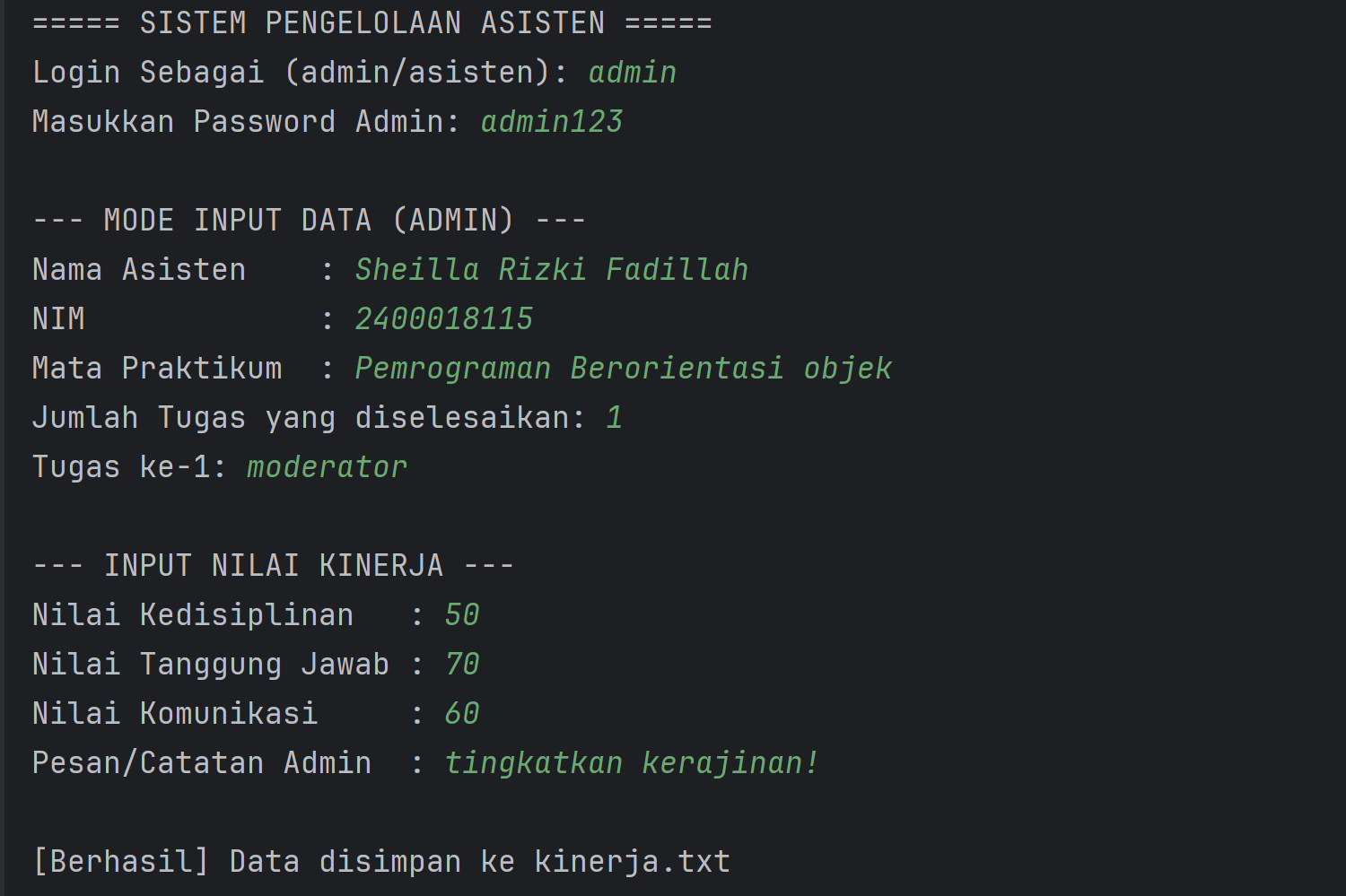
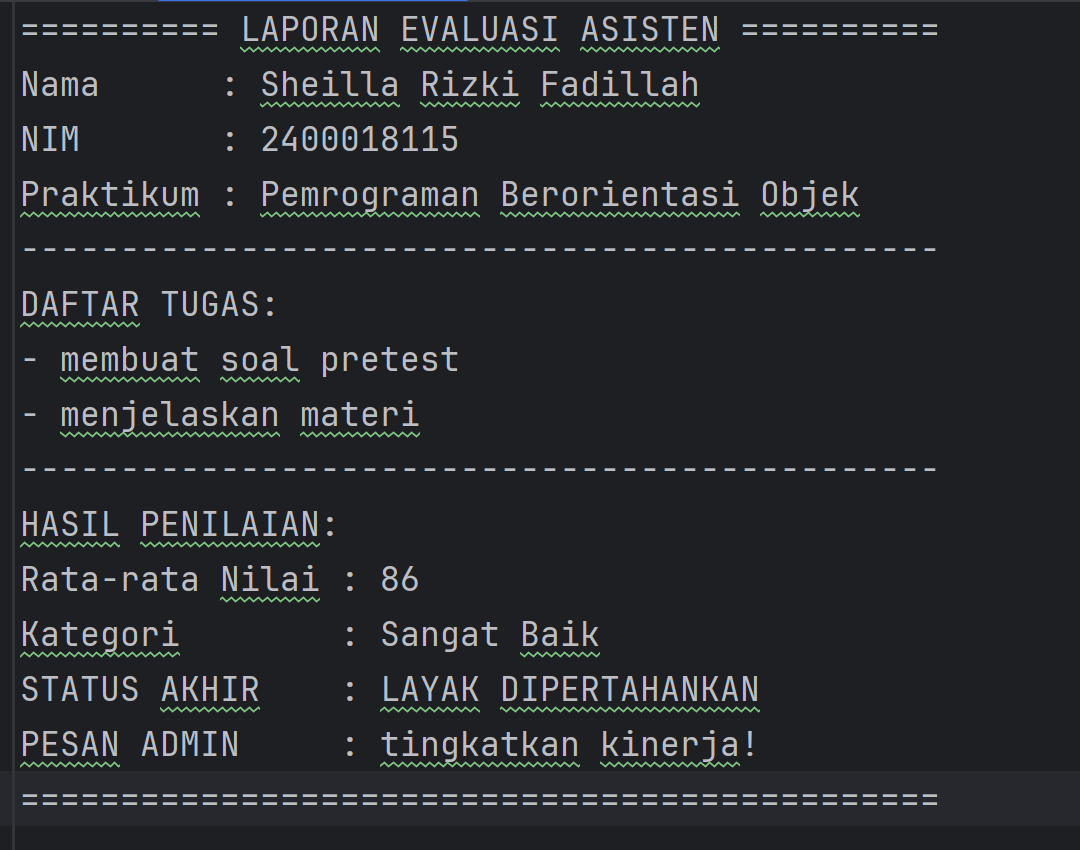
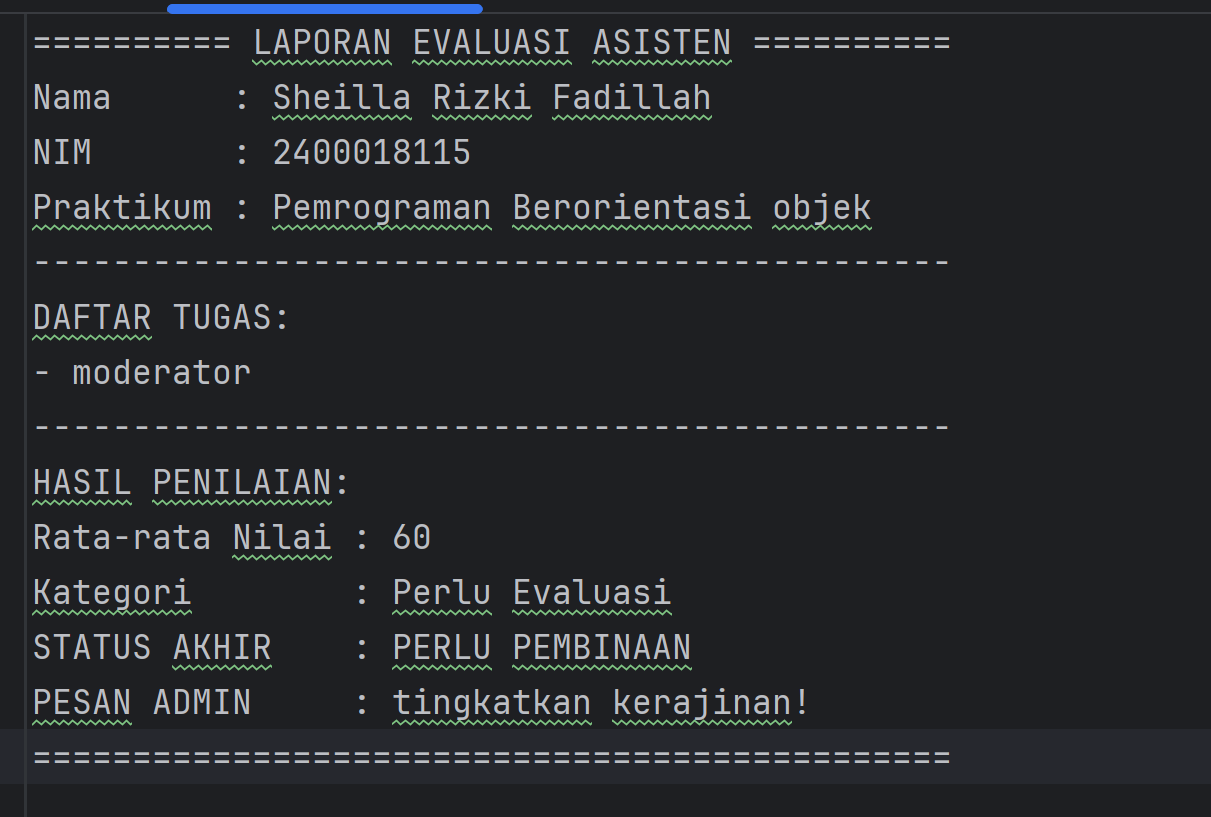
}

}

}

###### D. OUTPUT PROGRAM

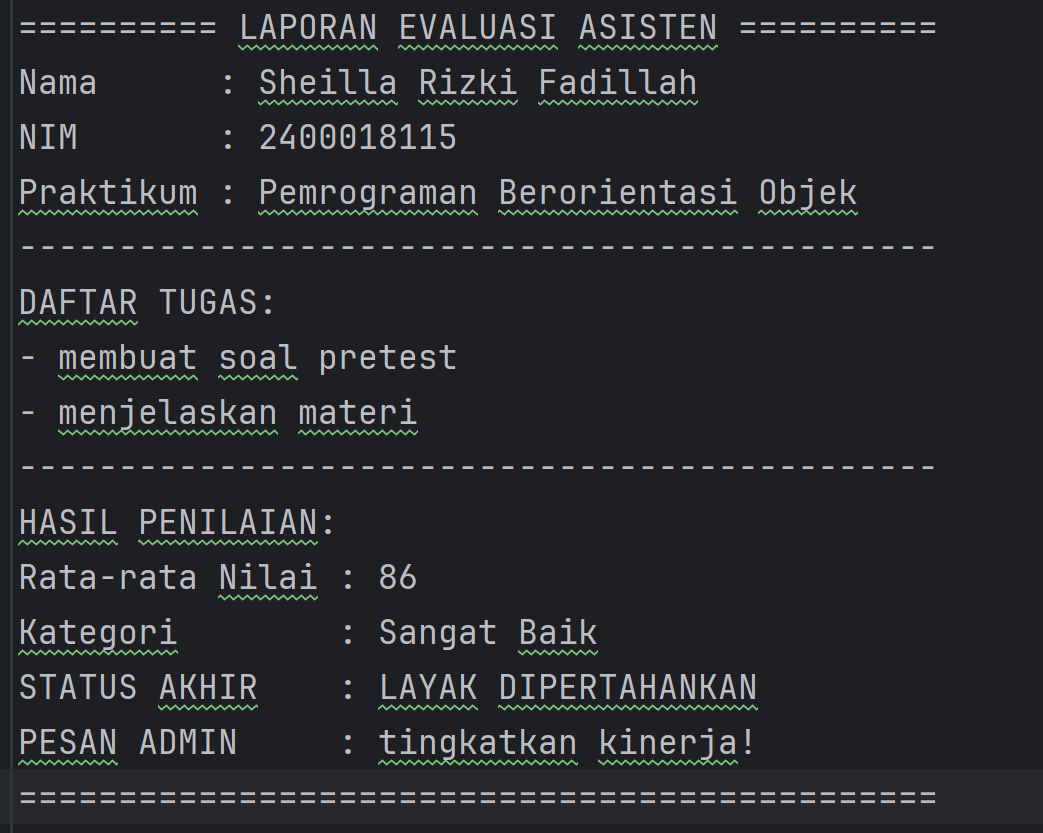
Output login sebagai admin

Output login sebagai asisten



Output pada kinerja.txt



### 1. Output untuk Admin (Konfirmasi & Penyimpanan)

Setelah Admin selesai melakukan input data dan menekan tombol "Simpan", output yang diterima adalah:

* Notifikasi Status (Feedback Visual): Sistem akan menampilkan pesan sukses, seperti *"Data berhasil disimpan"* atau *"Laporan terkirim"*. Ini adalah sinyal bagi Admin bahwa tugasnya telah selesai dan tidak ada *error*.
* Perubahan Status Data: Secara sistem, status laporan untuk asisten tersebut berubah dari *"Pending/Menunggu"* menjadi *"Selesai/Submitted"*.

### 2. Output untuk Asisten (Visualisasi Data)

Ini adalah output utama yang dilihat oleh pengguna akhir (Asisten) pada halaman Dashboard. Berdasarkan desain yang Anda kirimkan, outputnya terdiri dari 4 komponen:

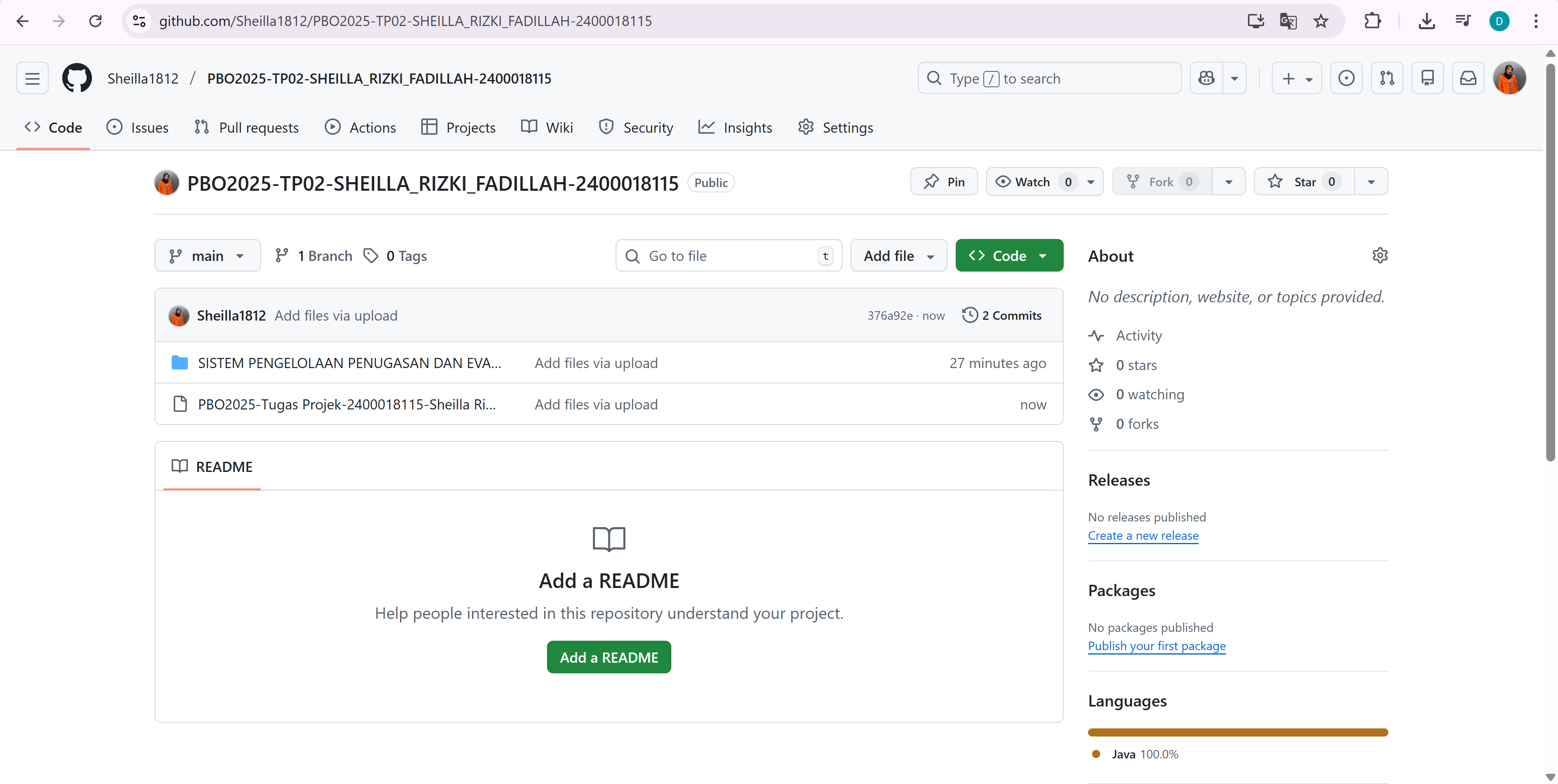
* A. Skor Rata-rata (Angka & Grafik): Ini adalah hasil kalkulasi matematika. Sistem mengambil rata-rata dari 3 nilai input (Kedisiplinan, Tanggung Jawab, Komunikasi).
  + *Contoh:* Jika nilai (90 + 80 + 85) / 3 = 85.
  + *Tampilan:* Angka "85" muncul di tengah lingkaran grafik.
* B. Status Laporan: Label yang menunjukkan apakah admin sudah menilai atau belum.
  + *Output:* Badge bertuliskan "Selesai" (atau "Status" berwarna biru seperti di gambar).
* C. Kategori Predikat: Sistem menerjemahkan angka menjadi kata-kata kualitatif.
  + *Contoh:* Jika skor 85, output teksnya mungkin "Sangat Baik" atau "A".
* D. Catatan Admin: Output teks deskriptif yang berisi saran atau kritik yang ditulis manual oleh Admin.
  + *Tampilan:* Muncul di bagian bawah dashboard (misal: *"Pertahankan kinerjanya, tingkatkan inisiatif"*).

### 3. Output di Belakang Layar (Database Record)

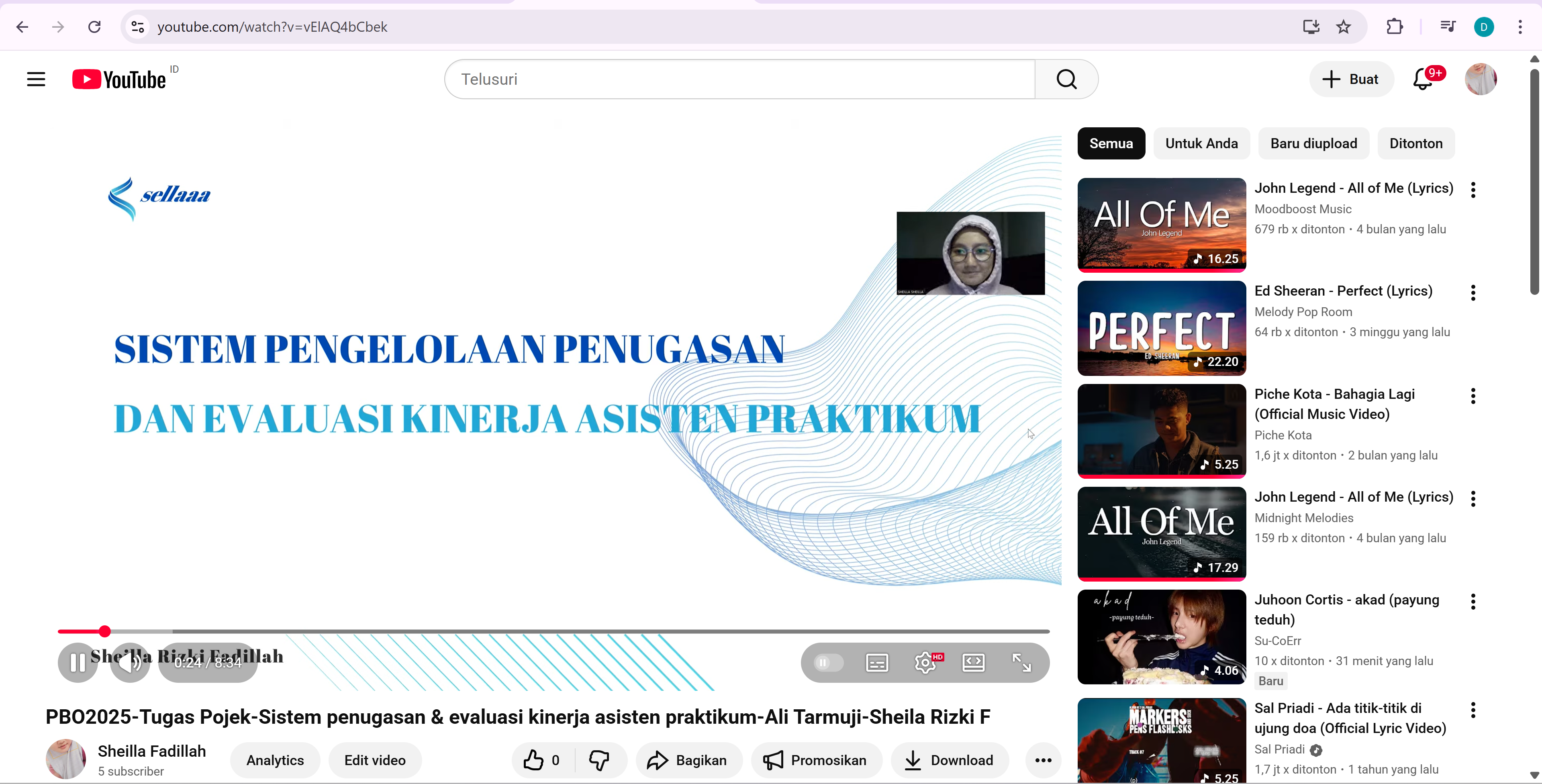
Ini adalah output yang tidak terlihat oleh user (invisible output) namun terbentuk di dalam database server

**E. BUKTI PUBLIKASI**

*Github :* <https://github.com/Sheilla1812/PBO2025-TP02-SHEILLA_RIZKI_FADILLAH-2400018115>

**

*Youtube :* [*https://youtu.be/vElAQ4bCbek*](https://youtu.be/vElAQ4bCbek)

**

F. ANALISIS PENGERJAAN PROJEK

Proyek ini saya kerjakan dalam waktu sekitar satu minggu sebagai bagian dari tugas proyek mata kuliah Pemrograman Berorientasi Objek. Selama proses pengerjaan, saya banyak mempelajari penerapan konsep OOP menggunakan bahasa Java, seperti pembuatan class, enkapsulasi, inheritance, dan overriding method. Dalam pengembangannya, saya menggunakan IntelliJ IDEA sebagai IDE utama untuk menulis, menjalankan, dan menguji program. Proyek ini tidak memerlukan biaya tambahan karena seluruh perangkat dan software yang digunakan bersifat gratis. Tantangan yang saya hadapi terutama dalam menyusun alur program, mengelola input pengguna, serta proses penyimpanan dan pembacaan data menggunakan file teks. Meskipun demikian, saya merasa puas dengan hasil akhir proyek ini karena memberikan pengalaman langsung dalam membangun aplikasi berbasis konsol yang terstruktur dan memperdalam pemahaman saya terhadap konsep pemrograman berorientasi objek.

G. KESIMPULAN

Berdasarkan hasil perancangan dan implementasi yang telah dilakukan, dapat disimpulkan bahwa Sistem Pengelolaan Penugasan dan Evaluasi Kinerja Asisten Praktikum berhasil dibangun dengan menerapkan konsep Pemrograman Berorientasi Objek menggunakan bahasa Java. Sistem ini mampu membantu proses pengelolaan data asisten, penugasan, serta evaluasi kinerja secara lebih terstruktur dan terdokumentasi. Melalui fitur login, input penilaian, perhitungan nilai otomatis, dan penyimpanan laporan ke dalam file, aplikasi ini dapat mempermudah dosen atau koordinator praktikum dalam melakukan evaluasi serta membantu asisten dalam mengetahui hasil kinerjanya. Selain itu, proyek ini juga memberikan pengalaman dan pemahaman yang lebih baik dalam menerapkan konsep OOP serta menyusun alur program secara sistematis.