

**TUGAS AKHIR**  
**ALGORITMA DAN PEMROGRAMAN**



**SISTEM INFORMASI PEMANTAUAN PERTUMBUHAN**  
**DAN EVALUASI KELAYAKAN PENGOLAHAN**  
**TEBAKAU**

**TIM PENYUSUN :**

Laili Rizqi Aulia (252410102035)

Adis Zalina Ali (252410102039)

Afifatul Muthi'ah Zahidah (252410102063)

**TAHUN AKADEMIK**

**2025/2026 GASAL**

# DAFTAR ISI

<b>BAB I</b>	
<b>PENDAHULUAN.....</b>	<b>3</b>
<b>1.1 Latar Belakang.....</b>	<b>3</b>
1.2 Tujuan.....	3
1.3 Batasan Masalah.....	4
<b>BAB II</b>	
<b>RANCANGAN APLIKASI.....</b>	<b>5</b>
2.1 Kebutuhan Sistem.....	5
2.2 Struktur Data.....	15
2.3 Rancangan Navigasi dan Tampilan.....	23
<b>BAB III</b>	
<b>IMPLEMENTASI RANCANGAN.....</b>	<b>27</b>
3.1 Fitur Registrasi.....	27
3.2 Fitur Login.....	30
3.3 Fitur Data Petani.....	32
3.4 Fitur Data Lahan.....	33
3.5 Fitur Pemeriksaan Tembakau.....	35
3.6 Fitur Data Hama.....	43
3.7 Fitur Evaluasi Kelayakan.....	45
3.8 Fitur Transaksi.....	47
3.9 Fitur Melihat Users.....	51
3.10 Fitur Melihat Pengajuan.....	52
3.11 Fitur Melihat Transaksi.....	53
3.12 Fitur Alamat.....	53
<b>BAB IV</b>	
<b>HASIL DAN PEMBAHASAN.....</b>	<b>56</b>
4.1 Fitur Register.....	56
4.2 Fitur Login.....	58
4.3 Fitur Data Petani.....	59
4.4 Fitur Data Lahan.....	60
4.5 Fitur Pemeriksaan Tembakau.....	62
4.6 Fitur Data Hama.....	63
4.7 Fitur Evaluasi Kelayakan.....	64
4.8 Fitur Transaksi.....	66
4.9 Fitur Melihat Users.....	69
4.10 Fitur Melihat Pengajuan.....	70
4.11 Fitur Melihat Transaksi.....	71
<b>DAFTAR PUSTAKA.....</b>	<b>73</b>

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Industri tembakau memiliki peran penting dalam perekonomian Indonesia. Selain menjadi sumber pendapatan negara, sektor ini juga memberikan lapangan kerja bagi banyak petani. Proses budidaya hingga pengolahan tembakau memerlukan pengawasan yang cermat karena setiap tahap memengaruhi mutu hasil akhir.

Namun, pengelolaan data pertumbuhan dan kualitas tembakau masih banyak dilakukan secara manual. Pencatatan yang tidak terpusat sering menimbulkan kesalahan, keterlambatan analisis, serta kesulitan dalam pengambilan keputusan. Kondisi ini menyebabkan proses produksi kurang efisien dan kualitas produk sulit dipantau secara konsisten.

Untuk mengatasi hal tersebut, diperlukan sistem informasi yang dapat mengintegrasikan seluruh data secara terstruktur dan real-time. Sistem ini harus mampu mencatat data pertumbuhan tanaman, kondisi lingkungan, penggunaan pupuk, hingga hasil evaluasi kualitas daun tembakau. Dengan data yang terkelola baik, proses analisis dapat dilakukan lebih cepat, risiko gagal panen bisa ditekan, dan mutu produk dapat dijaga.

Perancangan basis data menjadi bagian penting dalam membangun sistem informasi tersebut. Basis data yang dirancang dengan baik akan menjaga keakuratan informasi, mempercepat pengolahan data, serta mendukung pengembangan sistem ke depan. Oleh karena itu, penelitian ini berfokus pada perancangan model basis data untuk mendukung sistem informasi pemantauan pertumbuhan dan evaluasi kelayakan tembakau yang efisien dan terintegrasi.

### **1.2 Tujuan**

Berikut merupakan tujuan dibangunnya sistem ini, yaitu:

- a. Untuk menyimpan data pertumbuhan tanaman tembakau yang meliputi tinggi tanaman, kelembaban tanah, kadar air daun, dan parameter pertumbuhan lainnya secara sistematis agar lebih terstruktur dan efisien
- b. Agar dapat mendokumentasikan informasi pertumbuhan tembakau secara digital sehingga data tidak mudah hilang dan dapat ditelusuri dengan baik
- c. Untuk membantu petani tembakau dalam memantau pertumbuhan tanaman dan mengambil keputusan berbasis data.
- d. Untuk menggantikan pencatatan metode konvensional dalam pengelolaan data pertumbuhan tembakau pada sektor agroindustri.
- e. Untuk membantu pihak Agroindustri untuk menemukan kualitas tembakau yang mereka inginkan dengan harga yang sesuai.

### **1.3 Batasan Masalah**

- a. Proyek ini hanya berfokus pada perancangan basis data
- b. Sistem memiliki tiga jenis pengguna, yaitu:
  - Admin: mengelola akun pengguna dan memeriksa data.
  - Petani: menginput data pertumbuhan tanaman seperti tinggi, lebar kelembapan, kondisi daun dan lain-lain.
  - Pihak pabrik: melihat hasil kelayakan tembakau dari para petani, dan menentukan apakah akan membeli tembakau dengan hasil kelayakan tersebut.
- c. Data yang digunakan hanya mencakup data pengguna, data pertumbuhan tanaman, dan hasil evaluasi kelayakan.
- d. Evaluasi kelayakan hanya berdasarkan data pertumbuhan tanaman, tidak membahas proses pengolahan atau faktor ekonomi.
- e. Database PostgreSQL hanya local tidak online.

## **BAB II**

### **RANCANGAN APLIKASI**

#### **2.1 Kebutuhan Sistem**

Aplikasi ini dirancang untuk mendukung aktivitas industri tembakau untuk pemantauan pertumbuhan dan evaluasi kelayakan pengolahan tembakau dengan menyediakan fitur login dan logout yang memisahkan akses berdasarkan role yaitu petani, pihak pabrik sebagai pengelola tembakau, dan admin sistem.

Kebutuhan sistem ini didasarkan pada hasil observasi terhadap proses industri tembakau sehingga dapat disimpulkan detail dari layanan-layanan utama sebagai berikut:

##### **A. Kebutuhan Umum Sistem**

1. Aplikasi ini memiliki fitur registrasi meliputi username, password, role
2. Aplikasi ini memiliki fitur login dengan mengautentikasi username dan password untuk mengakses sistem sesuai role masing-masing pengguna
3. Aplikasi ini memiliki fitur logout untuk mengakhiri sesi pengguna dan kemudian dialihkan ke halaman login
4. Aplikasi ini memiliki fitur penyimpanan data yang menyimpan seluruh data dalam database PostgreSQL untuk memudahkan pengelolaan dan analisis

##### **B. Kebutuhan Fitur untuk Petani**

Petani berperan sebagai pengguna yang melakukan input data pertumbuhan tanaman tembakau dan melihat hasil evaluasi kelayakan tembakau. Fitur yang tersedia untuk petani meliputi:

#### 1. Fitur Registrasi dan Login

- a. Petani dapat melakukan pendaftaran akun dengan memasukkan username, password, dan memilih role “Petani”
- b. Petani dapat login menggunakan akun yang telah didaftarkan untuk mengakses sistem
- c. Data registrasi dan login disimpan dalam tabel user

#### 2. Fitur Input Data Petani

- a. Petani dapat menginput data profil yang meliputi:
  - ID Petani (otomatis oleh sistem)
  - Nama Petani
  - No Telepon
  - Email
- b. Data disimpan dalam tabel user

#### 3. Fitur Input Data Lahan

- a. Petani dapat memasukkan informasi lahan yang digunakan untuk penanaman tembakau yang meliputi:

- ID Lahan (otomatis dari sistem)
- Lokasi Lahan
- Luas Lahan (M<sup>2</sup>)
- Jenis Tanah

b. Data disimpan dalam tabel lahan

#### 4. Fitur Input Data Pertumbuhan Tembakau

a. Petani dapat menginput data pertumbuhan tembakau secara berkala untuk setiap periode pengamatan yang meliputi:

- ID Data Pertumbuhan (otomatis dari sistem)
- Tanggal Pengamatan
- Tinggi Tembakau (cm)
- Warna Daun
- Lebar Daun (cm)
- Tekstur Daun
- Kelembaban Tanah(%)
- Kadar Air (%)

b. Data disimpan dalam tabel lahan

#### 5. Fitur Edit dan Hapus Data Pertumbuhan Tembakau

- a. Petani dapat mengedit data pertumbuhan yang telah diinput jika terdapat kesalahan atau perubahan
- b. Petani dapat menghapus data pertumbuhan berdasarkan ID Data yang dipilih

#### 6. Fitur Melihat Riwayat Pemeriksaan Pertumbuhan

- a. Petani dapat melihat riwayat data pertumbuhan tanaman tembakau yang sebelumnya telah diinput
- b. Data ditampilkan dalam bentuk tabel agar mudah dibaca

#### 7. Fitur Input Data Hama

- a. Petani dapat menginput data hama yang menyerang tembakau untuk dokumentasi dan analisis
- b. Data hama meliputi:
  - Nama Hama
  - Jenis Hama
  - Tingkat Serangan

#### 8. Fitur Melihat Hasil Evaluasi Kelayakan

- a. Setelah petani menginput data pertumbuhan, sistem secara otomatis melakukan evaluasi kelayakan berdasarkan parameter yang telah ditetapkan oleh pihak pabrik



b. Petani dapat melihat hasil evaluasi yang meliputi:

- Hasil Evaluasi (Layak/Tidak Layak)
- Skor Kualitas (0-100)
- Indeks Evaluasi (A/B/C/D)
- Catatan berupa saran tindakan yang direkomendasikan oleh sistem untuk meningkatkan kualitas tembakau dapat berupa rekomendasi pemupukan, penyiraman, penanganan hama atau waktu panen yang optimal

c. Data disimpan dalam tabel evaluasi

#### 9. Fitur Pengajuan Penjualan Tembakau

- a. Petani dapat melakukan pengajuan penjualan kepada pihak pabrik.
- b. Pengajuan mencakup data lahan, jumlah tembakau yang tersedia dan harga yang diajukan.

#### 10. Fitur Melihat Transaksi Penjualan Tembakau

- a. Setelah pengajuan disetujui oleh pihak pabrik, petani dapat melakukan transaksi penjualan
- b. Petani dapat melihat detail transaksi yang telah disepakati yang meliputi:
  - ID Transaksi

- Tanggal Transaksi
- Jumlah Tembakau (kg)
- Harga (per kg)
- Total Pembayaran

c. Data transaksi disimpan dalam tabel transaksi

## 11. Fitur Logout

- a. Petani dapat keluar dari sistem dengan aman untuk pengamanan data

## C. Kebutuhan Fitur untuk Pihak Pabrik sebagai Pengelola Tembakau

Pihak pabrik sebagai pihak yang memfilter dan melakukan transaksi pembelian tembakau dari petani. Fitur yang tersedia meliputi:

### 1. Fitur Registrasi dan Login

- a. Pabrik dapat melakukan pendaftaran akun dengan memasukkan username, password, dan memilih role “Pabrik”
- b. Pabrik dapat login menggunakan akun yang telah didaftarkan untuk mengakses sistem.

### 2. Fitur Melihat Pengajuan Penjualan dari Petani

- a. Pihak pabrik dapat melihat daftar pengajuan penjualan tembakau yang telah diajukan oleh petani

b. Pengajuan mencakup:

- ID Pengajuan Penjualan
- Nama Petani
- Harga yang Diajukan
- Skor Kelayakan Tembakau
- Hasil Evaluasi

3. Fitur Menyetujui atau Menolak Pengajuan

- a. Pihak pabrik dapat menyetujui atau menolak pengajuan penjualan dari petani
- b. Status pengajuan (Disetujui/Ditolak) akan tersimpan

4. Fitur Input Transaksi Pembelian

- a. Setelah menyetujui pengajuan penjualan dari petani, pihak pabrik dapat melakukan input transaksi pembelian dengan mencatat:
  - ID Pengajuan Penjualan yang disetujui
  - Jumlah Pembelian Tembakau (kg)
  - Harga (per kg)
- b. Data disimpan dalam entity transaksi

## 5. Fitur Melihat Riwayat Transaksi

- a. Pihak pabrik dapat melihat riwayat semua transaksi pembelian yang telah dilakukan.

## 6. Fitur Logout

- a. Pihak pabrik dapat keluar dari sistem dengan aman

### D. Kebutuhan Fitur untuk Admin Sistem

Admin sistem berperan sebagai pengelola utama aplikasi yang mengatur data pengguna, memverifikasi data dan melakukan monitoring sistem. Fitur yang tersedia meliputi:

#### 1. Fitur Login

- a. Admin dapat login dengan menginput username dan password

#### 2. Fitur Manajemen Akun Pengguna

- a. Admin dapat mengelola akun pengguna dengan operasi:
  - Lihat Daftar Pengguna untuk melihat semua akun yang terdaftar di sistem
  - Hapus Pengguna untuk menghapus akun yang tidak aktif atau melanggar ketentuan
- b. Data pengguna disimpan dalam entity user

#### 3. Fitur Melihat Pengajuan Penjualan Tembakau

- a. Admin dapat melihat pengajuan penjualan dari petani ke pihak pabrik
- 4. Fitur Melihat Riwayat Transaksi
  - a. Admin dapat melihat riwayat transaksi tembakau yang telah disetujui oleh pihak pabrik
- 5. Fitur Logout
  - a. Admin dapat keluar dari sistem dengan aman setelah selesai melakukan pengelolaan

#### E. Alur Kerja Sistem

Berikut adalah alur kerja umum dari sistem informasi pemantauan pertumbuhan dan evaluasi kelayakan pengolahan tembakau:

1. Alur Petani:
  - a. Petani melakukan registrasi dengan menginput username, password dan memilih role “Petani”
  - b. Petani login ke sistem menggunakan akun yang telah didaftarkan
  - c. Petani menginput data profil
  - d. Petani menginput data lahan
  - e. Petani menginput data pertumbuhan tembakau secara berkala
  - f. Petani menginput data hama jika ada

- g. Sistem secara otomatis melakukan evaluasi kelayakan berdasarkan parameter yang telah ditetapkan oleh pihak pabrik
- h. Petani dapat melihat hasil evaluasi kelayakan pengolahan tembakau
- i. Petani dapat melakukan pengajuan penjualan kepada pihak pabrik
- j. Setelah pengajuan disetujui, petani dapat melakukan transaksi dan melihat detail transaksi
- k. Setelah selesai, petani dapat logout dari sistem

## 2. Alur Pihak Pabrik

- a. Pihak pabrik melakukan registrasi dengan menginput username, password dan memilih role “Pabrik”
- b. Pihak pabrik login ke sistem menggunakan akun yang telah didaftarkan
- c. Pihak Pabrik dapat melihat semua hasil evaluasi kelayakan tembakau dari berbagai petani
- d. Pihak Pabrik dapat melihat pengajuan penjualan tembakau dari petani
- e. Pihak Pabrik dapat menyetujui atau menolak pengajuan penjualan
- f. Setelah menyetujui, pihak pabrik dapat melakukan input transaksi pembelian tembakau

- g. Pihak Pabrik dapat melihat riwayat transaksi yang telah dilakukan
- h. Setelah selesai, pihak pabrik dapat logout dari sistem

### 3. Alur Admin

- a. Admin login dengan memasukkan username dan password
- b. Admin dapat mengelola akun pengguna
- c. Admin dapat melihat semua pengajuan penjualan dari petani ke pihak pabrik
- d. Admin dapat melihat riwayat transaksi tembakau yang telah disetujui oleh pihak pabrik
- e. Setelah selesai, admin dapat logout dari sistem

## 2.2 Struktur Data

No	Fitur	Data	Tipe Data	Penyimpanan
1.	Tampilan Awal	-	String	-
2.	Registrasi	Username Password Role	String String String	PostgreSQL
3.	Login	Username	String	PostgreSQL

		Password	String	
4.	Input data petani (petani)	Nama No telpon Email	String Integer String	PostgreSQL
5.	Input data lahan (petani)	Id users Id lahan Lokasi Luas lahan Jenis tanah	Integer Integer String Float String	PostgreSQL
6.	Input pemeriksaan tembakau (Petani)	Id pemeriksaan Id users Id lahan Tanggal pengamatan Tinggi tembakau Tekstur daun Lebar daun Warna daun	Integer Integer Integer Date Float String Float String Float	PostgreSQL



		Kadar air		
7.	Input data hama (petani)	Id hama  Id pemeriksaan  Nama hama  Jenis hama  Tingkat Serangan hama	Integer  Integer  String  String  String	PostgreSQL
8.	Edit data pemeriksaan (petani)	Id pemeriksaan	Integer	PostgreSQL
9.	Hapus data pemeriksaan (petani)	Id pemeriksaan	Integer	PostgreSQL

10.	Lihat riwayat pemeriksaan (petani)	Id pemeriksaan Id users Id lahan Tanggal pengamatan Tinggi tembakau Tekstur daun Lebar daun Warna daun Kadar air	Integer Integer Integer Date Float String Float String Float	PostgreSQL
11.	Evaluasi Kelayakan (otomatis) (petani)	Id evaluasi kelayakan Id pemeriksaan Hasil evaluasi Skor evaluasi Grade evaluasi Rekomendasi	Integer Integer String Float String String Datetime	PostgreSQL

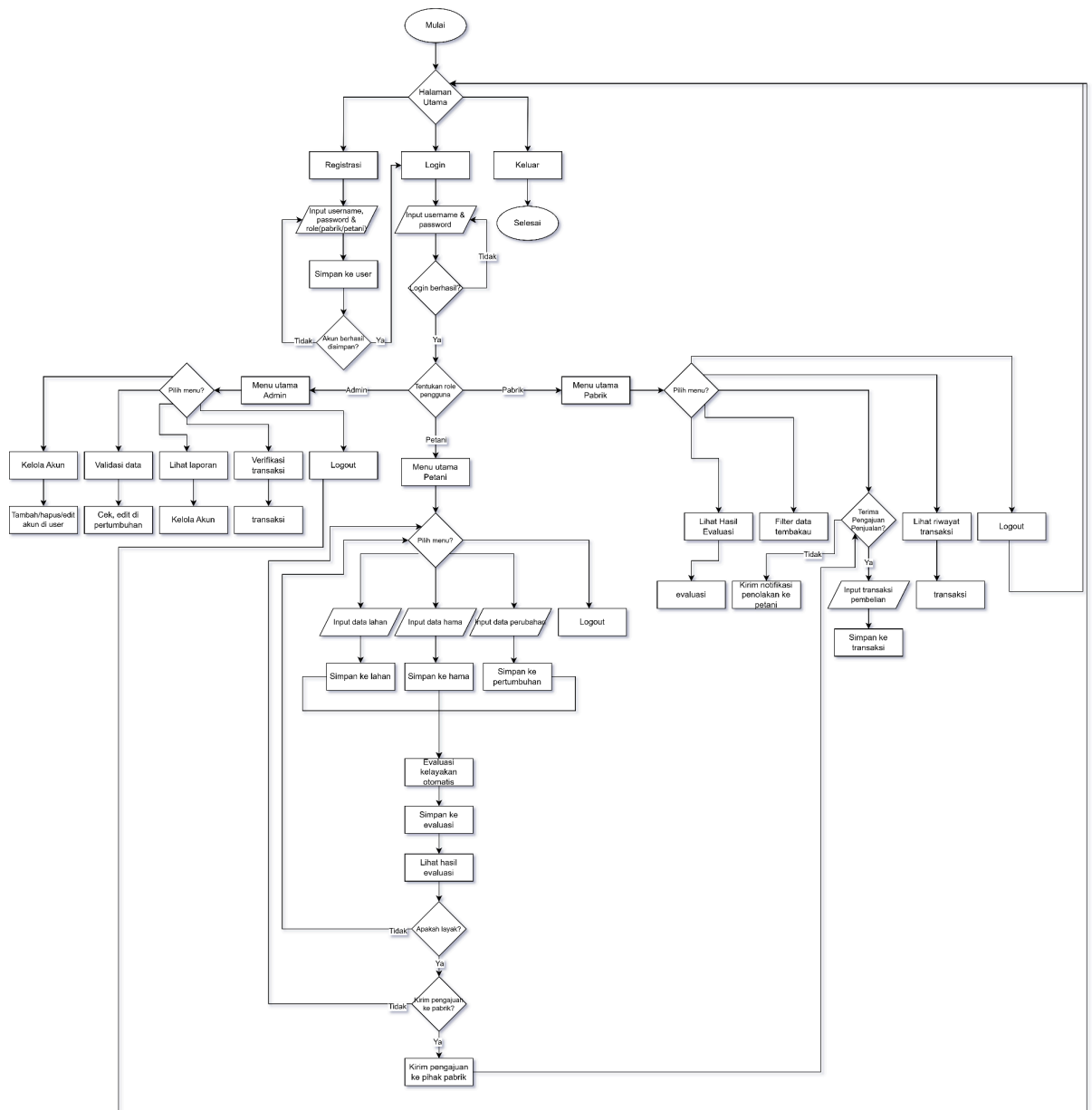
		Tanggal evaluasi		
12.	Lihat hasil evaluasi (petani dan pabrik)	Id users Id evaluasi kelayakan	Integer Integer	PostgreSQL
13.	Pengajuan transaksi tembakau (petani)	Id pengajuan Id users Id evaluasi Id transaksi Tanggal pengajuan Jumlah (kg) Pengajuan harga	Integer Integer Integer Integer Datetime Float Integer	PostgreSQL

14.	Lihat pengajuan penjualan (admin dan pabrik)	Id pengajuan Id users Id evaluasi Id transaksi Tanggal pengajuan Jumlah (kg) Pengajuan harga	Integer Integer Integer Integer Datetime Float Integer	PostgreSQL
15.	Memutuskan pengajuan (pihak pabrik)	Id transaksi Id users Id pembayaran Tanggal transaksi Jumlah (kg) Harga sepakat Total harga Status transaksi Keterangan	Integer Integer Integer Datetime Float Integer Integer Integer String String	PostgreSQL

16.	Transaksi pembelian (pabrik)	Id pembayaran Id users Id transaksi Tanggal pembayaran Metode pembayaran Status pembayaran	Integer Integer Integer Datetime String String	PostgreSQL
17.	Lihat riwayat transaksi (admin)	Id pembayaran Id users Id transaksi Tanggal pembayaran Metode pembayaran Status pembayaran	Integer Integer Integer Datetime String String	PostgreSQL
18.	Lihat semua users (admin)	Id user Username Role	Integer String String	PostgreSQL

19.	Hapus users (admin)	Id user	Integer	PostgreSQL
20.	Lihat semua pengajuan (admin)	Id Pengajuan  Id User  Harga  Status	Integer  Integer  Float  String	PostgreSQL
21.	Lihat semua transaksi (admin)	Id Transaksi  Id user  Harga  Jumlah  Total	Integer  Integer  Float  Float  Float	PostgreSQL
22.	Logout	-	String	-

## 2.3 Rancangan Navigasi dan Tampilan



Definisi navigasi atau alur aplikasi ketika admin, petani, dan pihak pabrik mengoperasikan aplikasi yang dibuat adalah sebagai berikut:

- Alur Admin

1. Admin login dengan memasukkan username dan password.

2. Setelah berhasil masuk, admin diarahkan ke menu utama admin yang berisi: Melihat akun user, Hapus user, Lihat semua pengajuan, dan Lihat semua transaksi.
  3. Pada menu melihat akun user, admin dapat mengetahui siapa saja yang telah melakukan registrasi, dan login.
  4. Menu hapus user, admin dapat menghapus akun-akun yang dinilai mencurigakan.
  5. Melalui lihat semua pengajuan, admin dapat mengetahui petani yang melakukan pengajuan penjualan, beserta informasi tambahannya terkait harga dan lain-lain.
  6. Opsi lihat semua transaksi menampilkan kegiatan transaksi yang telah dilakukan oleh pihak petani dan pabrik dengan harga yang telah disepakati.
  7. Admin dapat kembali ke menu utama dengan menekan enter.
  8. Untuk keluar dari aplikasi, admin memilih menu logout.
- Alur Petani
    1. Petani login dengan memasukkan username dan password. Apabila belum memiliki akun, petani harus melakukan pendaftaran terlebih dahulu dengan mengisi username, password dan memilih role. Setelah itu, petani dapat login menggunakan akun yang sudah terdaftar.
    2. Setelah berhasil masuk, petani akan diarahkan ke menu utama yang berisi pilihan: input data lahan, input data pertumbuhan, input data hama, edit/hapus data pertumbuhan, riwayat pertumbuhan, hasil evaluasi, pengajuan penjualan, dan transaksi.



3. Pada menu input data lahan, sistem menampilkan formulir untuk mengisi informasi lahan seperti lokasi, luas, dan jenis tanah.
  4. Melalui menu input data pertumbuhan, petani dapat mengisi data pengamatan harian seperti tinggi tanaman, warna daun, lebar daun, kadar air, serta kelembapan tanah.
  5. Menu input data hama digunakan untuk mencatat jenis hama, dan tingkat serangan.
  6. Pada menu edit data, petani memilih data pertumbuhan yang ingin diperbarui, lalu memperbaiki bagian yang diperlukan.
  7. Menu hapus data memungkinkan petani menghapus data pertumbuhan tertentu dari sistem.
  8. Opsi riwayat pertumbuhan menampilkan seluruh data pertumbuhan yang pernah dicatat.
  9. Melalui menu hasil evaluasi, petani dapat melihat skor kelayakan tanaman serta rekomendasi dari sistem.
  10. Pada menu pengajuan penjualan, petani mengisi jumlah tembakau yang akan dijual dan harga yang diajukan untuk dikirim ke pihak pabrik.
  11. Fitur transaksi menampilkan rincian transaksi pembelian yang telah disetujui pabrik.
  12. Petani dapat kembali ke menu utama dengan menekan enter.
  13. Untuk keluar dari aplikasi, pilih menu logout.
- Alur Pihak Pabrik

1. Pabrik login dengan memasukkan username dan password. Apabila belum memiliki akun, pabrik harus melakukan pendaftaran terlebih dahulu dengan mengisi data diri dan memilih role. Setelah itu, pabrik dapat login menggunakan akun yang sudah terdaftar.
2. Setelah berhasil masuk, sistem menampilkan menu utama pabrik, berisi: Lihat Pengajuan Penjualan, Putuskan Pengajuan, Transaksi Pembelian, dan Riwayat Transaksi.
3. Menu Lihat Pengajuan Penjualan menampilkan pengajuan penjualan tembakau dari seluruh petani.
4. Fitur Putuskan Pengajuan memungkinkan pabrik untuk menyetujui atau tidak untuk melanjutkan ke kegiatan transaksi.
5. Melalui Transaksi Pembelian, pabrik melakukan pembelian tembakau dengan mengisi berapa kilogram jumlah tembakau yang dibeli, dan berapa harga sepakat antara kedua belah pihak.
6. Pada riwayat transaksi, sistem menampilkan seluruh transaksi yang telah diproses sebelumnya.
7. Pabrik dapat kembali ke menu utama dengan menekan enter.
8. Untuk keluar dari aplikasi, pilih logout

## BAB III

### IMPLEMENTASI RANCANGAN

Bab ini berisi penjelasan mengenai implementasi kode program dari aplikasi yang telah dirancang. Setiap fitur yang ada dalam sistem dijelaskan secara detail berdasarkan daftar fitur yang telah dipaparkan pada Bab II. Pembahasan mencakup seluruh bagian kode yang digunakan, yang disusun secara terstruktur sesuai dengan fungsi masing-masing fitur. Untuk memudahkan pemahaman, setiap fitur dijelaskan dalam bentuk subbab tersendiri, sehingga alur logika dan struktur program dapat dilihat dengan jelas. Adapun penjabaran kode program tersebut disampaikan sebagai berikut.

#### 3.1 Fitur Registrasi

Fitur registrasi digunakan oleh pengguna baru untuk membuat akun sebelum dapat mengakses sistem. Pada tahapan ini, pengguna diminta mengisi data dasar berupa username, password, nama lengkap, nomor telepon, email, memilih jenis pengguna (petani, atau pabrik) dan menginput alamat users. Sistem kemudian melakukan pengecekan apakah username tersebut sudah pernah digunakan. Jika data valid dan belum ada duplikasi, maka akun berhasil dibuat dan disimpan. Jika tidak, sistem memberikan peringatan dan meminta pengguna mengulangi registrasi.

```
def register_manual():
    while True:
        username = input("Username: ")
        if not username.strip():
            print("username tidak boleh kosong!")
            continue
        if not username.isalnum():
            print("username hanya boleh angka dan huruf!")
            continue
        break
```

```

while True:
    password = input("Password: ")
    if not password.strip():
        print("password tidak boleh kosong!")
        continue
    break

while True:
    nama = input("Nama lengkap: ")
    if not nama.strip():
        print("nama tidak boleh kosong!")
    if not re.match(r'^[A-Za-z ]+$', nama):
        print("Nama hanya boleh huruf dan spasi")
        continue
    break

while True:
    no_telp = input("No Telepon: ")
    if not no_telp.strip():
        print("No telp tidak boleh kosong")
    if not no_telp.isdigit():
        print("no telp hanya boleh angka!")
        continue
    break

email = input("Email: ").strip()

print("\nPilih Jenis Pengguna:")
print("1. Petani")
print("2. Pabrik")
pilihan = input("Jenis (1/2): ").strip()
role_name = "petani" if pilihan == "1" else "pabrik" if pilihan == "2"
else None
if not role_name:
    print("Pilihan tidak valid!")

```

```

        return

    print("\n--- Input Alamat User ---")
    id_alamat = pilih_alamat()

    conn = get_connection()
    if not conn:
        return

    try:
        cur = conn.cursor()
        cur.execute("SELECT id_users FROM users WHERE username = %s",
(username,))
        if cur.fetchone():
            print("Username sudah digunakan!")
            return

        cur.execute("SELECT id_role FROM role WHERE nama_role ILIKE %s",
(role_name,))
        role_row = cur.fetchone()
        id_role = role_row[0] if role_row else None

        cur.execute(
            "INSERT INTO users (nama, no_kontak, email, username, password,
id_role, id_alamat) "
            "VALUES (%s,%s,%s,%s,%s,%s,%s)",
            (nama, no_telp, email, username, password, id_role, id_alamat)
        )
        conn.commit()
        print("Registrasi berhasil!")

    except Exception as e:
        print("Error registrasi:", e)
    finally:
        cur.close()

```

```
conn.close()
```

Gambar 3.1.1 Fitur Registrasi

### 3.2 Fitur Login

Setelah akun dibuat, pengguna dapat masuk ke sistem melalui proses login. Pengguna memasukkan username dan password, lalu sistem mencocokkannya dengan data yang tersimpan. Jika sesuai, pengguna diarahkan ke menu utama sesuai perannya. Jika tidak sesuai, akan muncul pesan kesalahan dan login harus diulang.

```
def login():
    while True:
        username = input("Username: ")
        if not username.strip():
            print("username tidak boleh kosong!")
            continue
        if not username.isalnum():
            print("username hanya boleh angka dan huruf!")
            continue
        break

    while True:
        password = input("Password: ")
        if not password.strip():
            print("password tidak boleh kosong!")
            continue
        break

    if username == "admin" and password == "admin123":
        print("Login sebagai Admin")
        dashboard_admin()
        return

    conn = get_connection()
```

```

if not conn:
    return

try:
    cur = conn.cursor()
    # ambil role name via join untuk memudahkan cek
    cur.execute("""
        SELECT u.id_users, r.nama_role
        FROM users u
        LEFT JOIN role r ON u.id_role = r.id_role
        WHERE u.username = %s AND u.password = %s
    """, (username, password))
    row = cur.fetchone()
    if row:
        id_user, role_name = row
        print(f"Login berhasil sebagai {role_name}!")
        if role_name and role_name.lower() == "petani":
            dashboard_petani(username)
        elif role_name and role_name.lower() == "pabrik":
            dashboard_pabrik(username)
        else:
            input("Menu untuk role ini belum dibuat. Tekan Enter untuk
kembali...")
    else:
        print("Username atau password salah!")
except Exception as e:
    print(f"Error login: {e}")
finally:
    cur.close()
    conn.close()

```

Gambar 3.2.1 Fitur Login

### 3.3 Fitur Data Petani

Melalui fitur ini, petani dapat memperbarui nama, nomor telepon, email, dan alamat mereka jika ada kesalahan atau pembaruan data.

```
def input_data_petani(username):  
    """  
    Update profil user di tabel users (nama, no_kontak, email ...).  
    """  
    conn = get_connection()  
    if not conn:  
        return  
    try:  
        cur = conn.cursor()  
        cur.execute("SELECT id_users, nama, no_kontak, email, id_alamat FROM  
users WHERE username = %s", (username,))  
        row = cur.fetchone()  
        if not row:  
            print("User tidak ditemukan.")  
            cur.close(); conn.close(); return  
        id_user = row[0]  
        print("\n--- Update Profil ---")  
        while True:  
            nama = input("Nama lengkap: ")  
            if not nama.strip():  
                print("nama tidak boleh kosong!")  
            if not re.match(r'^[A-Za-z ]+$', nama):  
                print("Nama hanya boleh huruf dan spasi")  
                continue  
            break  
  
        while True:  
            telp = input(f"No Telepon ({row[2] or ''}): ") or row[2]  
            if not telp.isdigit():  
                print("no telp tidak boleh selain angka!")  
            if not telp.strip():
```



```

        print("no telp tidak boleh kosong!")
        break
    while True:
        email = input(f"Email({row[3] or ''}):") or row[3]
        if not email.strip():
            print("email tidak boleh kosong!")
            break

    pilih = input("Apakah ingin mengubah alamat? (y/n): ").lower()
    if pilih == "y":
        id_alamat = pilih_alamat()
    else:
        id_alamat = row[4] # id_alamat lama dari SELECT

    cur.execute("UPDATE users SET nama=%s, no_kontak=%s, email=%s,
id_alamat=%s WHERE id_users=%s", (nama, telp, email, id_alamat, id_user))
    conn.commit()
    print("Profil berhasil diperbarui!")
except Exception as e:
    print(f"Error update profil: {e}")
finally:
    cur.close(); conn.close()

```

Gambar 3.3.1 Fitur Data Petani

### 3.4 Fitur Data Lahan

Fitur ini digunakan petani untuk mencatat informasi mengenai lahan tempat menanam tembakau. Data yang dicatat meliputi lokasi lahan, luas lahan, ketinggian, dan jenis tanah. Informasi ini menjadi dasar untuk proses pemeriksaan dan evaluasi tembakau pada tahap berikutnya.

```

def input_data_lahan(id_petani):

    id_alamat = pilih_alamat()

```

```

if not id_alamat:
    print("Gagal menyimpan alamat!")
    return

luas = input("Luas Lahan (M²): ")
print("\nPilihan Jenis Tanah:")
print("1. Tanah Aluvial")
print("2. Tanah Andosol")
pilih = input("Jenis Tanah (1/2): ")
jenis = "Aluvial" if pilih == "1" else "Andosol" if pilih == "2" else None

if not jenis:
    print("Pilihan tidak valid!")
    return

conn = get_connection()
if not conn:
    return

try:
    cur = conn.cursor()
    cur.execute(
        "INSERT INTO lahan (luas_lahan, ketinggian_lahan, jenis_tanah,
id_users, id_alamat) "
        "VALUES (%s,%s,%s,%s,%s) RETURNING id_lahan",
        (float(luas), 0.0, jenis, id_petani, id_alamat)
    )
    id_lahan = cur.fetchone()[0]
    conn.commit()
    print(f>Data lahan disimpan! ID Lahan = {id_lahan}")
except Exception as e:
    print(f"Error simpan lahan: {e}")
finally:
    cur.close(); conn.close()

```

Gambar 3.4.1 Fitur Data Lahan

### 3.5 Fitur Pemeriksaan Tembakau

Melalui fitur ini, petani dapat mencatat kondisi tanaman tembakau pada saat pengamatan. Data yang dicatat meliputi tinggi tanaman, tekstur daun, warna daun, lebar daun, kadar air, tanggal pengamatan, dan keterangan tambahan tentang kondisi tanamannya. Informasi tersebut menjadi data utama dalam proses analisis sistem. Selain menginputkan, data pemeriksaan tembakau juga dapat diedit atau dihapus serta petani dapat melihat riwayat data yang telah diinputkan.

```
def input_data_pertumbuhan(id_petani):
    lahan_ids = get_petani_lahan_ids(id_petani)
    if not lahan_ids:
        print("Belum ada lahan.")
        return
    conn = get_connection()
    if not conn:
        return
    try:
        query = "SELECT id_lahan, luas_lahan FROM lahan WHERE id_users = %s"
        df = pd.read_sql(query, conn, params=(id_petani,))
        for _, row in df.iterrows():
            print(f"{row['id_lahan']} - Luas: {row['luas_lahan']}")
    except Exception as e:
        print(f"Error: {e}")
        conn.close()
        return

    pilih = input("ID Lahan: ")
    if int(pilih) not in lahan_ids:
        print("ID lahan bukan milik Anda.")
        conn.close()
        return
```

```

    tanggal = input(f"Tanggal (Enter = {datetime.now().strftime('%Y-%m-%d')}): ") or datetime.now().strftime("%Y-%m-%d")

    # validasi angka
    while True:
        tinggi = input("Tinggi (cm): ")
        if tinggi.replace(".", "", 1).isdigit():
            break
        print("❌ Tinggi harus angka!")

    warna = input("Warna Daun: ")

    while True:
        lebar = input("Lebar Daun (cm): ")
        if lebar.replace(".", "", 1).isdigit():
            break
        print("❌ Lebar harus angka!")

    tekstur = input("Tekstur Daun: ")

    while True:
        lembab = input("Kelembapan (%): ")
        if lembab.replace(".", "", 1).isdigit():
            break
        print("❌ Kelembapan harus angka!")

    while True:
        kadar = input("Kadar Air (%): ")
        if kadar.replace(".", "", 1).isdigit():
            break
        print("❌ Kadar air harus angka!")

    try:
        cur = conn.cursor()
        cur.execute(

```

```

        """INSERT INTO pemeriksaan (tanggal_pemeriksaan, tinggi_tembakau,
tekstur_daun, lebar_daun, warna_daun, kadar_air, keterangan, id_lahan,
id_users)

        VALUES (%s,%s,%s,%s,%s,%s,%s,%s,%s,%s) RETURNING
id_pemeriksaan""",

        (tanggal, float(tinggi), tekstur, float(lebar), warna,
float(kadar), '', int(pilih), id_petani)
    )
    id_pemeriksaan = cur.fetchone()[0]
    conn.commit()
    print("✓ Data pemeriksaan tersimpan!")
    buat_evaluasi_otomatis(id_pemeriksaan, tinggi, warna, lebar, tekstur,
lembab, kadar)
except Exception as e:
    print(f"Error simpan pemeriksaan: {e}")
finally:
    cur.close(); conn.close()

```

Gambar 3.5.1 Fitur Pemeriksaan Tembakau

```

def edit_data_pertumbuhan(id_petani):
    lahan_ids = get_petani_lahan_ids(id_petani)
    if not lahan_ids:
        print("Belum ada lahan.")
        return
    conn = get_connection()
    if not conn:
        return
    try:
        print("\n===== DATA PEMERIKSAAN =====")
        query = """
            SELECT p.id_pemeriksaan, p.id_lahan, p.tanggal_pemeriksaan,
p.tinggi_tembakau,
                p.warna_daun, p.lebar_daun, p.tekstur_daun, p.kadar_air
            FROM pemeriksaan p

```

```

        WHERE p.id_lahan = ANY(%s)
        ORDER BY p.id_pemeriksaan DESC
    """

    df = pd.read_sql(query, conn, params=(lahan_ids,))
    if df.empty:
        print("Belum ada data pemeriksaan.")
        return

    for _, row in df.iterrows():
        print(f"ID: {row['id_pemeriksaan']} | Lahan: {row['id_lahan']} |
Tgl: {row['tanggal_pemeriksaan']} | Tinggi: {row['tinggi_tembakau']} | Warna:
{row['warna_daun']}")

    id_edit = input("\nMasukkan ID Pemeriksaan yang ingin diedit: ")
    cur = conn.cursor()
    cur.execute("SELECT id_lahan FROM pemeriksaan WHERE id_pemeriksaan =
%s", (id_edit,))
    cek = cur.fetchone()
    if not cek or cek[0] not in lahan_ids:
        print("ID tidak valid atau bukan milik Anda!")
        cur.close(); conn.close(); return

    # ambil data lama
    cur.execute("""SELECT tanggal_pemeriksaan, tinggi_tembakau,
warna_daun, lebar_daun, tekstur_daun, kadar_air
FROM pemeriksaan WHERE id_pemeriksaan = %s""",
(id_edit,))
    old = cur.fetchone()

    print("\n--- Masukkan data baru (Enter untuk tetap pakai data lama)
---")

    tanggal = input(f"Tanggal ({old[0]}): ") or old[0]

    # validasi angka untuk tinggi
    while True:
        tinggi = input(f"Tinggi ({old[1]}): ") or old[1]

```

```

        if str(tinggi).replace(".", "", 1).isdigit():
            break
        print("✗ Tinggi harus angka!")

warna = input(f"Warna Daun ({old[2]}): ") or old[2]

# validasi angka untuk lebar
while True:
    lebar = input(f"Lebar Daun ({old[3]}): ") or old[3]
    if str(lebar).replace(".", "", 1).isdigit():
        break
    print("✗ Lebar harus angka!")

tekstur = input(f"Tekstur Daun ({old[4]}): ") or old[4]

# validasi angka untuk kadar air
while True:
    kadar = input(f"Kadar Air ({old[5]}): ") or old[5]
    if str(kadar).replace(".", "", 1).isdigit():
        break
    print("✗ Kadar air harus angka!")

# update pemeriksaan
cur.execute(
    """UPDATE pemeriksaan SET tanggal_pemeriksaan=%s,
tinggi_tembakau=%s, warna_daun=%s,
    lebar_daun=%s, tekstur_daun=%s, kadar_air=%s WHERE
id_pemeriksaan=%s""",
    (tanggal, float(tinggi), warna, float(lebar), tekstur,
float(kadar), id_edit)
)
conn.commit()
print("✓ Data pemeriksaan berhasil diperbarui!")

# update evaluasi lama (tidak delete, langsung update)

```

```

        buat_evaluasi_otomatis(id_edit, tinggi, warna, lebar, tekstur, None,
kadar)

    except Exception as e:
        print(f"Error edit: {e}")
    finally:
        try:
            cur.close()
        except:
            pass
        conn.close()

```

Gambar 3.5.2 Fitur Edit Data Pemeriksaan Tembakau

```

def hapus_data_pertumbuhan(id_petani):
    print("\n===== HAPUS DATA PEMERIKSAAN =====")
    lahan_ids = get_petani_lahan_ids(id_petani)
    if not lahan_ids:
        print("Anda belum memiliki lahan.")
        return
    conn = get_connection()
    if not conn:
        return
    try:
        query = """
            SELECT p.id_pemeriksaan, p.tanggal_pemeriksaan, p.tinggi_tembakau,
p.warna_daun, p.id_lahan
            FROM pemeriksaan p
            WHERE p.id_lahan = ANY(%s)
            ORDER BY p.tanggal_pemeriksaan DESC
        """
        df = pd.read_sql(query, conn, params=(lahan_ids,))
        if df.empty:
            print("Belum ada data pemeriksaan.")
            return
    
```



```

print("\nDaftar Pemeriksaan:")
for _, row in df.iterrows():
    print(f"ID: {row['id_pemeriksaan']} | Tanggal: {row['tanggal_pemeriksaan']} | Tinggi: {row['tinggi_tembakau']} cm | Warna: {row['warna_daun']} | Lahan: {row['id_lahan']}")

id_hapus = input("\nMasukkan ID Pemeriksaan yang ingin dihapus: ")
if int(id_hapus) not in df['id_pemeriksaan'].values:
    print("❌ ID tidak ditemukan atau bukan milik Anda!")
    return

yakin = input(f"Yakin ingin menghapus ID {id_hapus}? (y/n): ")
if yakin.lower() != "y":
    print("Dibatalkan.")
    return

cur = conn.cursor()

# cek apakah evaluasi pemeriksaan ini dipakai di pengajuan
cur.execute("""
    SELECT COUNT(*) FROM pengajuan
    WHERE id_evaluasi IN (SELECT id_evaluasi FROM evaluasi WHERE
id_pemeriksaan=%s)
""", (id_hapus,))
if cur.fetchone()[0] > 0:
    print("❌ Data pemeriksaan tidak bisa dihapus karena evaluasi sudah dipakai di pengajuan!")
    return

# kalau aman, hapus evaluasi & hama lalu pemeriksaan
cur.execute("DELETE FROM evaluasi WHERE id_pemeriksaan = %s",
(id_hapus,))
cur.execute("DELETE FROM hama WHERE id_pemeriksaan = %s", (id_hapus,))
cur.execute("DELETE FROM pemeriksaan WHERE id_pemeriksaan = %s",
(id_hapus,))

```

```

        conn.commit()
        print("✓ Data pemeriksaan berhasil dihapus!")

except Exception as e:
    print(f"Error: {e}")
finally:
    try:
        cur.close()
    except:
        pass
    conn.close()

```

Gambar 3.5.3 Fitur Edit Data Pemeriksaan Tembakau

```

def lihat_riwayat_pertumbuhan(id_petani):
    lahan_ids = get_petani_lahan_ids(id_petani)
    if not lahan_ids:
        print("Belum ada lahan.")
        return
    conn = get_connection()
    if not conn:
        return
    print("\n===== RIWAYAT PEMERIKSAAN =====")
    try:
        query = """
            SELECT id_pemeriksaan, tanggal_pemeriksaan, tinggi_tembakau,
warna_daun
            FROM pemeriksaan
            WHERE id_lahan = ANY(%s)
            ORDER BY tanggal_pemeriksaan DESC
        """
        df = pd.read_sql(query, conn, params=(lahan_ids,))
        if df.empty:
            print("Belum ada data.")
        else:
            for _, row in df.iterrows():

```

```

        print(f"ID: {row['id_pemeriksaan']} | 
{row['tanggal_pemeriksaan']} | Tinggi: {row['tinggi_tembakau']} cm | Warna: 
{row['warna_daun']}")
    except Exception as e:
        print(f"Error: {e}")
    finally:
        conn.close()

```

Gambar 3.5.4 Fitur Lihat Riwayat Pemeriksaan Tembakau

### 3.6 Fitur Data Hama

Fitur ini memungkinkan petani mencatat jika ada serangan hama di lahan mereka. Data yang dimasukkan berupa nama hama, jenis hama, dan tingkat serangan. Catatan ini membantu dalam menilai kondisi tanaman secara menyeluruh dan dapat berpengaruh pada hasil evaluasi.

```

def input_data_hama(id_petani):
    lihat_riwayat_pertumbuhan(id_petani)
    id_pemeriksaan = input("\nID Pemeriksaan terkena hama: ")
    lahan_ids = get_petani_lahan_ids(id_petani)
    conn = get_connection()
    if not conn:
        return
    try:
        cur = conn.cursor()
        cur.execute("SELECT id_lahan FROM pemeriksaan WHERE id_pemeriksaan = 
%s", (id_pemeriksaan,))
        result = cur.fetchone()
        if not result or result[0] not in lahan_ids:
            print("ID tidak valid atau bukan milik Anda!")
            cur.close(); conn.close(); return
        nama_hama = input("Nama Hama: ").strip()
        jenis = input("Jenis Hama (nama jenis): ").strip()
        tingkat = input("Tingkat Serangan (nama tingkat): ").strip()
        # pastikan jenis_hama ada

```

```

        id_jenis = None
        if jenis:
            cur.execute("SELECT id_jenis_hama FROM jenis_hama WHERE jenis_hama
ILIKE %s", (jenis,))
            r = cur.fetchone()
            if r:
                id_jenis = r[0]
            else:
                cur.execute("INSERT INTO jenis_hama (jenis_hama) VALUES (%s)
RETURNING id_jenis_hama", (jenis,))
                id_jenis = cur.fetchone()[0]
        # pastikan tingkat_serangan ada
        id_tingkat = None
        if tingkat:
            cur.execute("SELECT id_tingkat_serangan FROM tingkat_serangan
WHERE tingkat_serangan ILIKE %s", (tingkat,))
            r = cur.fetchone()
            if r:
                id_tingkat = r[0]
            else:
                cur.execute("INSERT INTO tingkat_serangan (tingkat_serangan)
VALUES (%s) RETURNING id_tingkat_serangan", (tingkat,))
                id_tingkat = cur.fetchone()[0]
        # insert hama
        cur.execute(
            "INSERT INTO hama (nama_hama, id_jenis_hama, id_tingkat_serangan,
id_pemeriksaan) VALUES (%s,%s,%s,%s)",
            (nama_hama, id_jenis, id_tingkat, id_pemeriksaan)
        )
        conn.commit()
        print("Data hama tersimpan!")
    except Exception as e:
        print(f"Error: {e}")
    finally:
        try:

```

```

        cur.close()
    except:
        pass
    conn.close()

```

Gambar 3.6.1 Fitur Data Hama

### 3.7 Fitur Evaluasi Kelayakan

Evaluasi kelayakan dilakukan secara otomatis berdasarkan data pemeriksaan yang dimasukkan petani. Sistem menggunakan beberapa parameter, seperti tinggi tanaman, kondisi daun, dan kadar air, untuk menentukan apakah tembakau berada dalam kondisi layak, tidak layak, beserta menampilkan skor kualitas dan juga indeks evaluasinya. Hasil evaluasi ini disimpan dan dapat dilihat oleh petani maupun pihak pabrik.

```

def lihat_hasil_evaluasi(id_petani):
    lahan_ids = get_petani_lahan_ids(id_petani)
    if not lahan_ids:
        print("Belum ada lahan.")
        return
    conn = get_connection()
    if not conn:
        return
    print("\n" + "="*70)
    print("          HASIL EVALUASI KELAYAKAN TEMBAKAU")
    print("="*70)
    try:
        # join ke pemeriksaan supaya batasi hanya milik lahan petani
        query = """
            SELECT ev.id_evaluasi, ev.tanggal_evaluasi, ev.skor_evaluasi,
ev.catatan,
                    ge.grade_evaluasi, he.hasil_evaluasi, ev.id_pemeriksaan
            FROM evaluasi ev
            LEFT JOIN grade_evaluasi ge ON ev.id_grade_evaluasi =
ge.id_grade_evaluasi

```

```

        LEFT JOIN hasil_evaluasi he ON ev.id_hasil_evaluasi =
he.id_hasil_evaluasi
        JOIN pemeriksaan p ON ev.id_pemeriksaan = p.id_pemeriksaan
        WHERE p.id_lahan = ANY(%s)
        ORDER BY ev.tanggal_evaluasi DESC
    """
    df = pd.read_sql(query, conn, params=(lahan_ids,))
    if df.empty:
        print("    Belum ada evaluasi.")
    else:
        for _, row in df.iterrows():
            print(f"\nID Evaluasi      : {row['id_evaluasi']}")
            print(f"ID Pemeriksaan : {row['id_pemeriksaan']}")
            print(f"Tanggal        : {row['tanggal_evaluasi']}")
            print(f"Skor           : {row['skor_evaluasi']}")
            print(f"Grade          : {row.get('grade_evaluasi')}")
            print(f"Hasil          : {row.get('hasil_evaluasi')}")
            catatan = row['catatan'] or ''
            if " || Rekomendasi: " in catatan:
                detail, rekom = catatan.split(" || Rekomendasi: ", 1)
                items = detail.split(" | ")
                print("\nDetail Penilaian:")
                for it in items:
                    if it.strip():
                        print(" " + it)
                print(f"\nRekomendasi      : {rekom}")
            else:
                print(f"Catatan          : {catatan}")
            print("-" * 70)
    except Exception as e:
        print(f"Error: {e}")
    finally:
        conn.close()

```

Gambar 3.7.1 Fitur Evaluasi Kelayakan Tembakau

### 3.8 Fitur Transaksi

Fitur transaksi merupakan rangkaian proses yang melibatkan petani, pihak pabrik, dan admin.

#### a. Pengajuan transaksi (petani)

Setelah tembakau dinyatakan layak berdasarkan hasil evaluasi, petani dapat mengajukan penjualan melalui sistem. Pada tahap ini, petani mengisi data berupa jumlah tembakau yang akan dijual serta harga yang diharapkan. Pengajuan ini disimpan dan menunggu proses tindak lanjut dari pihak pabrik.

```
def pengajuan_penjualan(id_petani):
    lahan_ids = get_petani_lahan_ids(id_petani)
    if not lahan_ids:
        print("Belum ada lahan.")
        return
    conn = get_connection()
    if not conn:
        return
    try:
        query = "SELECT id_lahan, luas_lahan FROM lahan WHERE id_users = %s"
        df = pd.read_sql(query, conn, params=(id_petani,))
        for _, row in df.iterrows():
            print(f"{row['id_lahan']} - Luas: {row['luas_lahan']} m²")

        pilih = input("ID Lahan: ")
        if int(pilih) not in lahan_ids:
            print("✗ Lahan tidak valid!")
            return

        # validasi angka untuk jumlah
        while True:
            jumlah = input("Jumlah (kg): ")
            if jumlah.replace(".", "", 1).isdigit():
                break
```

```

        print("❌ Jumlah harus angka!")

# validasi angka untuk harga
while True:
    harga = input("Harga per kg: ")
    if harga.replace(".", "", 1).isdigit():
        break
    print("❌ Harga harus angka!")

cur = conn.cursor()
# ambil evaluasi terbaru untuk lahan tsb
cur.execute("""
    SELECT ev.id_evaluasi FROM evaluasi ev
    JOIN pemeriksaan p ON ev.id_pemeriksaan = p.id_pemeriksaan
    WHERE p.id_lahan = %s
    ORDER BY ev.tanggal_evaluasi DESC LIMIT 1
""", (pilih,))
ev = cur.fetchone()
id_evaluasi = ev[0] if ev else None

cur.execute(
    "INSERT INTO pengajuan (tanggal_pengajuan, pengajuan_harga,
id_status_pengajuan, id_evaluasi, id_users) VALUES (%s,%s,%s,%s,%s)",
    (datetime.now(), float(harga), None, id_evaluasi, id_petani)
)
conn.commit()
print("✅ Pengajuan berhasil dikirim!")
except Exception as e:
    print(f"Error: {e}")
finally:
    try:
        cur.close()
    except:
        pass
conn.close()

```



Gambar 3.8.1 Fitur Pengajuan Penjualan Tembakau

b. Putuskan pengajuan (pihak pabrik)

Setelah menerima pengajuan, pihak pabrik dapat memutuskan untuk membuat transaksi pembelian dengan petani yang mana. Pabrik mencatat data transaksi seperti tanggal transaksi, jumlah tembakau yang dibeli, harga kesepakatan, serta keterangan tambahan bila diperlukan. Informasi ini menjadi dasar dalam proses pembayaran dan laporan penjualan.

```
def putuskan_pengajuan(username):  
    conn = get_connection()  
    if not conn:  
        return  
    try:  
        id_pengajuan = input("Masukkan ID Pengajuan yang ingin diputuskan: ")  
        keputusan = input("Keputusan (setuju/tolak): ").lower()  
  
        cur = conn.cursor()  
        # sesuaikan dengan nama kolom di tabel status_pengajuan  
        cur.execute("SELECT id_status_pengajuan FROM status_pengajuan WHERE  
status_pengajuan ILIKE %s", (keputusan,))  
        status_row = cur.fetchone()  
        if not status_row:  
            print("✗ Status tidak valid! Harus 'setuju' atau 'tolak'.")  
            return  
        id_status = status_row[0]  
  
        cur.execute("UPDATE pengajuan SET id_status_pengajuan=%s WHERE  
id_pengajuan=%s", (id_status, id_pengajuan))  
        conn.commit()  
        print(f"✓ Pengajuan {id_pengajuan} berhasil di-{keputusan}.")  
    except Exception as e:  
        print("Error putuskan pengajuan:", e)  
    finally:
```

```

try:
    cur.close()
except:
    pass
conn.close()

```

Gambar 3.8.2 Fitur Putuskan Pengajuan Penjualan Tembakau

c. Transaksi pembelian (pihak pabrik)

Dalam fitur ini, sistem menyimpan detail transaksi meliputi jumlah tembakau yang akan dibeli, harga per kg yang telah disepakati dan total harga yang harus dibayar.

```

def transaksi_pembelian(username):
    conn = get_connection()
    if not conn:
        return
    try:
        id_pengajuan = input("ID Pengajuan yang disetujui: ")

        # validasi angka jumlah
        while True:
            jumlah = input("Jumlah (kg): ")
            if jumlah.replace(".", "", 1).isdigit():
                break
            print("✗ Jumlah harus angka!")

        # validasi angka harga
        while True:
            harga = input("Harga sepakat per kg: ")
            if harga.replace(".", "", 1).isdigit():
                break
            print("✗ Harga harus angka!")

```

```

total = float(jumlah) * float(harga)

cur = conn.cursor()
# ambil id_users petani dari pengajuan
cur.execute("SELECT id_users FROM pengajuan WHERE id_pengajuan=%s",
(id_pengajuan,))
petani_row = cur.fetchone()
id_petani = petani_row[0] if petani_row else None

# simpan transaksi dengan id_users petani
cur.execute("""
    INSERT INTO transaksi (tanggal_transaksi, jumlah_kg,
harga_sepakat, id_pengajuan, id_users)
    VALUES (%s,%s,%s,%s,%s)
    """, (datetime.now(), float(jumlah), float(harga), id_pengajuan,
id_petani))
conn.commit()
print(f"✓ Transaksi pembelian tercatat untuk petani {id_petani}.
Total: Rp {total:,.0f}")
except Exception as e:
    print("Error transaksi:", e)
finally:
    try:
        cur.close()
    except:
        pass
conn.close()

```

Gambar 3.8.2 Fitur Transaksi Pembelian Tembakau

### 3.9 Fitur Melihat Users

Fitur ini diterapkan pada role admin untuk melihat semua user yang masuk ke sistem baik dari role petani maupun role pabrik

```
def lihat_semua_user():
    conn = get_connection()
    if not conn: return
    try:
        df = pd.read_sql("SELECT id_users, username, id_role FROM users ORDER
BY id_users", conn)
        print("\n===== DAFTAR USER =====")
        for _, r in df.iterrows():
            print(f"ID: {r['id_users']} | Username: {r['username']} | Role:
{r['id_role']}")
    except Exception as e:
        print("Error lihat user:", e)
    finally:
        conn.close()
```

Gambar 3.9.1 Fitur Melihat Semua Users

### 3.10 Fitur Melihat Pengajuan

Fitur ini diterapkan pada role admin untuk melihat pengajuan penjualan tembakau dari petani ke pihak pabrik

```
def lihat_semua_pengajuan():
    conn = get_connection()
    if not conn: return
    try:
        df = pd.read_sql("SELECT * FROM pengajuan ORDER BY tanggal_pengajuan
DESC", conn)
        print("\n===== SEMUA PENGAJUAN =====")
        for _, r in df.iterrows():
            print(f"ID: {r['id_pengajuan']} | User: {r['id_users']} | Harga:
Rp {r['pengajuan_harga']} | Status: {r['id_status_pengajuan']}")
    except Exception as e:
        print("Error lihat pengajuan:", e)
    finally:
```

```
conn.close()
```

Gambar 3.10.1 Fitur Melihat Semua Pengajuan Penjualan

### 3.11 Fitur Melihat Transaksi

Fitur ini diterapkan pada role admin untuk melihat transaksi yang dilakukan oleh petani dan pihak pabrik

```
def lihat_semua_transaksi():  
    conn = get_connection()  
    if not conn: return  
    try:  
        df = pd.read_sql("SELECT * FROM transaksi ORDER BY tanggal_transaksi  
DESC", conn)  
        print("\n===== SEMUA TRANSAKSI =====")  
        for _, r in df.iterrows():  
            total = float(r['jumlah_kg']) * float(r['harga_sepakat'])  
            print(f"ID: {r['id_transaksi']} | User: {r['id_users']} | Jumlah:  
{r['jumlah_kg']} kg | Harga: Rp {r['harga_sepakat']} | Total: Rp  
{total:,.0f}")  
    except Exception as e:  
        print("Error lihat transaksi:", e)  
    finally:  
        conn.close()
```

Gambar 3.11.1 Fitur Melihat Semua Transaksi Tembakau

### 3.12 Fitur Pilih Alamat

Fitur ini diterapkan pada saat petani dan pabrik melakukan registrasi setelah menginput data. Pada fitur pilih alamat ini disediakan kabupaten, kecamatan, desa dan menginput nama jalan.

```
def pilih_alamat():  
    conn = get_connection()
```

```

if not conn:
    return None
cur = conn.cursor()

cur.execute("SELECT id_kabupaten, nama_kabupaten FROM kabupaten ORDER BY
nama_kabupaten")
kab = cur.fetchall()
print("\nPilih Kabupaten:")
for k in kab:
    print(f"{k[0]}. {k[1]}")
pilih_kab = input("ID Kabupaten: ")

cur.execute("SELECT id_kecamatan, nama_kecamatan FROM kecamatan WHERE
id_kabupaten=%s ORDER BY nama_kecamatan",
            (pilih_kab,))
kec = cur.fetchall()
print("\nPilih Kecamatan:")
for c in kec:
    print(f"{c[0]}. {c[1]}")
pilih_kec = input("ID Kecamatan: ")

cur.execute("SELECT id_desa, nama_desa FROM desa WHERE id_kecamatan=%s
ORDER BY nama_desa",
            (pilih_kec,))
des = cur.fetchall()
print("\nPilih Desa:")
for d in des:
    print(f"{d[0]}. {d[1]}")
pilih_des = input("ID Desa: ")

nama_jalan = input("Nama Jalan: ")

cur.execute("INSERT INTO alamat (nama_jalan, id_desa) VALUES (%s,%s)
RETURNING id_alamat",
            (nama_jalan, pilih_des))

```

```
id_alamat = cur.fetchone()[0]
conn.commit()

cur.close()
conn.close()
return id_alamat
```

Gambar 3.12.1 Fitur Pilih Alamat

## BAB IV

### HASIL DAN PEMBAHASAN

Deskripsi hasil pembuatan aplikasi dengan cara menampilkan screenshot pengoperasian fitur aplikasinya. Seperti pada Bab III, setiap fitur dituliskan dalam satu sub bab seperti berikut ini.

#### 4.1 Fitur Register

Fitur registrasi merupakan fungsi yang digunakan untuk membuat akun baru dalam Sistem Informasi Pemantauan Tembakau. Melalui fitur ini, pengguna (petani atau pabrik) dapat memasukkan informasi awal berupa username, password, dan jenis pengguna. Data tersebut kemudian disimpan ke dalam database PostgreSQL sehingga pengguna dapat melakukan login dan mengakses menu sesuai perannya.

```
===== SISTEM INFORMASI PEMANTAUAN TEMBAKAU =====
1. Registrasi
2. Login
3. Keluar
Pilih: 1
Username: azarine
Password: 7007
Nama lengkap: Regini
No Telepon: 081290007000
Email: hivi@gmail.com

Pilih Jenis Pengguna:
1. Petani
2. Pabrik
Jenis (1/2): 1

--- Input Alamat User ---
koneksi berhasil

Pilih Kabupaten:
2. BANYUWANGI
4. BONDOWOSO
1. JEMBER
5. LUMAJANG
3. SITUBONDO
ID Kabupaten: 5

Pilih Kecamatan:
9. CANDIPURO
10. JATIROTO
ID Kecamatan: 10

Pilih Desa:
30. JATIROTO
32. KARANGHARJO
31. KEDUNGBANTENG
ID Desa: 31
Nama Jalan: Jl. Resapi
koneksi berhasil
Registrasi berhasil!
```



### Gambar 4.1.1 Tampilan Fitur Registrasi

#### 1. Kebutuhan Sistem

- Menyediakan proses pendaftaran akun baru untuk Petani dan Pabrik.
- Menyimpan data pribadi pengguna ke tabel users.
- Melakukan input alamat lengkap berdasarkan tabel referensi:
  - Kabupaten
  - Kecamatan
  - Desa
- Menyimpan alamat ke tabel alamat yang digunakan sistem.
- Melakukan validasi koneksi database agar data tersimpan dengan benar.

#### 2. Struktur Data

##### Tabel Users

- ID Users: integer
- Username: varchar
- Password: varchar
- Nama Lengkap: varchar
- Telepon: varchar
- Email: varchar
- Jenis Pengguna: varchar (petani / pabrik)

##### Tabel Alamat

- ID Alamat: integer
- ID Users: integer
- ID Kabupaten: integer
- ID Kecamatan: integer
- ID Desa: integer
- Nama Jalan: varchar

##### Tabel Referensi

- Kabupaten
  - Kecamatan
  - Desa
3. Rancangan Langkah penggunaan
- a. Pengguna memilih menu Registrasi.
  - b. Sistem meminta pengguna mengisi:
    - Username
    - Password
    - Nama lengkap
    - Nomor telepon
    - Email
  - c. Sistem menampilkan pilihan jenis pengguna:
    - Petani
    - Pabrik
  - d. Setelah memilih jenis pengguna, sistem masuk ke proses input alamat:
    1. Menampilkan daftar kabupaten → pilih ID
    2. Menampilkan daftar kecamatan berdasarkan kabupaten → pilih ID
    3. Menampilkan daftar desa berdasarkan kecamatan → pilih ID
    4. Input nama jalan
  - e. Sistem menyimpan:
    - Data akun ke tabel users
    - Data alamat ke tabel alamat\_user
    - Sistem menampilkan pesan "Registrasi berhasil" apabila seluruh data tersimpan.

#### **4.2 Fitur Login**

Fitur login digunakan untuk masuk ke sistem menggunakan akun yang sudah terdaftar. Setelah berhasil login, pengguna akan diarahkan ke menu sesuai perannya.

```
===== SISTEM INFORMASI PEMANTAUAN TEMBAKAU =====  
1. Registrasi  
2. Login  
3. Keluar  
Pilih: 2  
Username: Laili  
Password: 1305  
koneksi berhasil  
Login berhasil sebagai petani!  
koneksi berhasil
```

Gambar 4.2.1 Tampilan Fitur Login

1. Kebutuhan Sistem

- Pengguna memasukkan username dan password.
- Sistem mengecek kecocokan data di database.
- Sistem mengarahkan ke menu Petani jika role = Petani.
- Sistem menolak login bila data salah.

2. Struktur Data

- Username : string
- Password : string
- Role : string

3. Rancangan Langkah Penggunaan

- Pada menu utama, pilih **2** (Login).
- Masukkan username dan password.
- Sistem mencocokkan data.
- Jika cocok: masuk ke menu Petani/Pabrik.
- Jika salah: tampil pesan error.

### 4.3 Fitur Data Petani

Fitur ini memungkinkan petani memperbarui nama dan nomor telepon mereka.

```
===== MENU PETANI =====
1. Input & edit Data Profil
2. Input Data Lahan
3. Input Data Pemeriksaan (Pertumbuhan)
4. Edit Data Pemeriksaan
5. Hapus Data Pemeriksaan
6. Lihat Riwayat Pemeriksaan
7. Input Data Hama
8. Lihat Hasil Evaluasi
9. Pengajuan Penjualan
10. Lihat Transaksi
11. Logout

Pilih (1-11): 1
koneksi berhasil

--- Update Profil ---
Nama lengkap: dina nur
No Telepon (087232341234):
Email(dina@gmail.com):
Apakah ingin mengubah alamat? (y/n): n
Profil berhasil diperbarui!
```

Gambar 4.3.1 Tampilan Input Profil

#### 1. Kebutuhan Sistem

- Sistem mengambil data profil lama dari database.
- Petani boleh mengosongkan input untuk mempertahankan nilai lama.
- Data diperbarui di tabel profil\_petani.

#### 2. Struktur Data

- ID Petani : integer
- Nama : string
- No Telp : string

#### 3. Rancangan Langkah Penggunaan

- Pilih menu 1 di menu petani.
- Sistem menampilkan profil lama.
- Masukkan data baru (boleh kosong).
- Sistem menyimpan perubahan.

### 4.4 Fitur Data Lahan

Fitur ini digunakan untuk menambahkan lahan baru oleh petani.

```

===== MENU PETANI =====
1. Input & edit Data Profil
2. Input Data Lahan
3. Input Data Pemeriksaan (Pertumbuhan)
4. Edit Data Pemeriksaan
5. Hapus Data Pemeriksaan
6. Lihat Riwayat Pemeriksaan
7. Input Data Hama
8. Lihat Hasil Evaluasi
9. Pengajuan Penjualan
10. Lihat Transaksi
11. Logout

Pilih (1-11): 2
koneksi berhasil

Pilih Kabupaten:
2. BANYUWANGI
4. BONDOWOSO
1. JEMBER
5. LUMAJANG
3. SITUBONDO
ID Kabupaten: 1

Pilih Kecamatan:
2. AMBULU
1. SUMBERSARI
ID Kecamatan: 1

```

```

Pilih Desa:
2. SUMBERSARI
30. SUMBERSARI
3. TEGALBESAR
31. TEGALBESAR
1. WIROLEGI
29. WIROLEGI
ID Desa: 2
Nama Jalan: jawa
Luas Lahan (M²): 200

Pilihan Jenis Tanah:
1. Tanah Aluvial
2. Tanah Andosol
Jenis Tanah (1/2): 1
koneksi berhasil
Data lahan disimpan! ID Lahan = 1

```

Gambar 4.4.1 Tampilan Input Lahan

#### 1. Kebutuhan Sistem

- Petani memasukkan lokasi, luas, dan jenis tanah.
- Sistem menyimpan data ke tabel lahan.
- Setelah tersimpan, ID lahan ditampilkan.

## 2. Struktur Data

- ID Lahan : integer
- ID Petani : integer
- Lokasi : string
- Luas : float
- Jenis Tanah : string

## 3. Rancangan Langkah Penggunaan

- Pilih menu 2
- Masukkan lokasi, luas, dan pilih jenis tanah.
- Sistem menyimpan data dan menampilkan ID lahan.

### 4.5 Fitur Pemeriksaan Tembakau

Fitur untuk mencatat pemeriksaan pertumbuhan tanaman tembakau.

```
===== MENU PETANI =====
1. Input & edit Data Profil
2. Input Data Lahan
3. Input Data Pemeriksaan (Pertumbuhan)
4. Edit Data Pemeriksaan
5. Hapus Data Pemeriksaan
6. Lihat Riwayat Pemeriksaan
7. Input Data Hama
8. Lihat Hasil Evaluasi
9. Pengajuan Penjualan
10. Lihat Transaksi
11. Logout

Pilih (1-11): 3
koneksi berhasil
koneksi berhasil
1.0 - Luas: 200.0
ID Lahan: 1
Tanggal (Enter = 2025-12-05):
Tinggi (cm): 100
Warna Daun: coklat tua
Lebar Daun (cm): 80
Tekstur Daun: halus
Kelembapan (%): 90
Kadar Air (%): 90
✓ Data pemeriksaan tersimpan!
koneksi berhasil
```

Gambar 4.5.1 Input Pemeriksaan Tembakau

## 1. Kebutuhan Sistem

- Ambil daftar lahan milik petani.
- Petani memilih ID lahan.
- Input: tinggi, warna daun, lebar daun, tekstur, kelembapan, kadar air.

- Sistem menyimpan ke tabel pemeriksaan.

## 2. Struktur Data

- ID Pemeriksaan : integer
- ID Lahan : integer
- Tanggal : datetime
- Tinggi : float
- Warna : string
- Lebar Daun : float
- Tekstur : string
- Kelembapan : integer
- Kadar Air : integer

## 3. Rancangan Langkah Penggunaan

- Pilih menu 3.
- Pilih lahan.
- Isi semua parameter pertumbuhan.
- Data tersimpan.

## 4.6 Fitur Data Hama

Untuk mencatat serangan hama berdasarkan pemeriksaan tertentu.

```
===== MENU PETANI =====
1. Input & edit Data Profil
2. Input Data Lahan
3. Input Data Pemeriksaan (Pertumbuhan)
4. Edit Data Pemeriksaan
5. Hapus Data Pemeriksaan
6. Lihat Riwayat Pemeriksaan
7. Input Data Hama
8. Lihat Hasil Evaluasi
9. Pengajuan Penjualan
10. Lihat Transaksi
11. Logout

Pilih (1-11): 7
koneksi berhasil
koneksi berhasil

===== RIWAYAT PEMERIKSAAN =====
ID: 1 | 2025-12-05 00:00:00 | Tinggi: 100.0 cm | Warna: coklat tua
ID: 2 | 2025-12-05 00:00:00 | Tinggi: 80.0 cm | Warna: hijau

ID Pemeriksaan terkena hama: 1
koneksi berhasil
koneksi berhasil
Nama Hama: keong
Jenis Hama (nama jenis): moluska
Tingkat Serangan (nama tingkat): rendah
Data hama tersimpan!
```

Gambar 4.6.1 Input Data Hama

### 1. Kebutuhan Sistem

- Menampilkan daftar pemeriksaan.
- Memilih pemeriksaan yang diserang.
- Memasukkan jenis hama dan tingkat serangan.
- Menyimpan ke tabel hama.

### 2. Struktur Data

- ID Hama : integer
- ID Pemeriksaan : integer
- Nama Hama : string
- Jenis Hama : string
- Tingkat Serangan : string

### 3. Rancangan Langkah Penggunaan

- Pilih menu 7.
- Pilih pemeriksaan yang terkena hama.
- Isi detail hama.
- Sistem menyimpan data.

## **4.7 Fitur Evaluasi Kelayakan**

Fitur yang menilai kualitas tembakau berdasarkan data pemeriksaan.



```
===== MENU PETANI =====
1. Input Data Profil
2. Input Data Lahan
3. Input Data Pemeriksaan (Pertumbuhan)
4. Edit Data Pemeriksaan
5. Hapus Data Pemeriksaan
6. Lihat Riwayat Pemeriksaan
7. Input Data Hama
8. Lihat Hasil Evaluasi
9. Pengajuan Penjualan
10. Lihat Transaksi
11. Logout

Pilih (1-11): 8
koneksi berhasil
koneksi berhasil

=====
HASIL EVALUASI KELAYAKAN TEMBAKAU
=====

ID Evaluasi      : 9
ID Pemeriksaan   : 4
Tanggal          : 2025-11-27 23:31:41.937629
Skor             : 55.0
Grade           : C
Hasil            : KURANG LAYAK

Detail Penilaian:
  Δ Tinggi kurang optimal, pertimbangkan pemupukan
  ✓ Lebar daun sempurna
  X Kelembapan tidak sesuai standar
  X Kadar air bermasalah
  ✓ Tekstur daun baik

Rekomendasi      : Perlu perbaikan signifikan.
=====
```

Gambar 4.7.1 Hasil Evaluasi Otomatis

## 1. Kebutuhan Sistem

- Sistem mengambil data pemeriksaan.
- Melakukan perhitungan skor.
- Menghasilkan grade, skor, rekomendasi.
- Menampilkan hasil lengkap.

## 2. Struktur Data

- ID Evaluasi : integer
- ID Pemeriksaan : integer
- Skor : float
- Grade : string
- Rekomendasi : string

3. Rancangan Langkah Penggunaan
  - Pilih menu 8.
  - Sistem menghitung evaluasi otomatis.
  - Tampilkan hasil evaluasi.

#### 4.8 Fitur Transaksi

1. Pengajuan transaksi (petani)

Fitur ini digunakan petani untuk mengajukan penjualan ke pabrik.

```
===== MENU PETANI =====
1. Input Data Profil
2. Input Data Lahan
3. Input Data Pemeriksaan (Pertumbuhan)
4. Edit Data Pemeriksaan
5. Hapus Data Pemeriksaan
6. Lihat Riwayat Pemeriksaan
7. Input Data Hama
8. Lihat Hasil Evaluasi
9. Pengajuan Penjualan
10. Lihat Transaksi
11. Logout

Pilih (1-11): 9
koneksi berhasil
koneksi berhasil
9.0 - Luas: 100.0 m2
10.0 - Luas: 150.0 m2
ID Lahan: 10
Jumlah (kg): 50
Harga per kg: 50000
Pengajuan berhasil dikirim!
```

Gambar 4.8.1 Pengajuan Penjualan

- a. Kebutuhan Sistem
  - Menampilkan daftar lahan petani.
  - Input jumlah tembakau dan harga per kilo.
  - Menyimpan ke tabel pengajuan.
- b. Struktur Data

- ID Pengajuan: integer
  - ID Lahan: integer
  - Jumlah (kg): integer
  - Harga per kg: integer
- c. Rancangan Langkah Penggunaan
- Pilih menu 9
  - Pilih lahan.
  - Masukkan jumlah dan harga.
  - Pengajuan tersimpan.
2. Putuskan pembelian (pihak pabrik)

Fitur ini digunakan oleh pihak pabrik untuk menentukan akan melanjutkan transaksi dengan transaksi dengan petani yang mana.

```
===== MENU PABRIK =====
1. Lihat Pengajuan Penjualan
2. Putuskan Pengajuan
3. Transaksi Pembelian
4. Riwayat Transaksi
5. Edit Profil
6. Logout

Pilih (1-6): 2
koneksi berhasil
Masukkan ID Pengajuan yang ingin diputuskan: 1
Keputusan (setuju/tolak): setuju
✓ Pengajuan 1 berhasil di-setuju.
```

#### a. Kebutuhan Sistem

- Menampilkan daftar pengajuan dari petani.
- Pihak pabrik memilih satu pengajuan.
- Menentukan keputusan: diterima / ditolak.
- Menyimpan keputusan ke tabel pengajuan.

#### b. Struktur Data

- ID Pengajuan: integer

- ID Pabrik (users): integer
- Jumlah (kg): integer
- Harga per kg: integer
- Status Pengajuan: string (setuju/tidak)

#### c. Rancangan Langkah Penggunaan

- Pilih menu “Putuskan Pembelian”.
- Sistem menampilkan daftar pengajuan.
- Pilih ID pengajuan.
- Tentukan keputusan (terima/tolak).
- Status dan keterangan tersimpan di database.

#### 3. Transaksi pembelian (pihak pabrik)

Fitur ini digunakan oleh pihak pabrik untuk melanjutkan ke kegiatan transaksi dengan mengisi data-data yang ada, seperti hal nya harga yang telah disepakati.

```
===== MENU PABRIK =====
1. Lihat Pengajuan Penjualan
2. Putuskan Pengajuan
3. Transaksi Pembelian
4. Riwayat Transaksi
5. Edit Profil
6. Logout

Pilih (1-6): 3
koneksi berhasil
ID Pengajuan yang disetujui: 1
Jumlah (kg): 50
Harga sepakat per kg: 50000
✓ Transaksi pembelian tercatat untuk petani 2. Total: Rp 2,500,000
```

#### a. Kebutuhan Sistem

- Membuat transaksi pembelian berdasarkan pengajuan yang sudah diterima.
- Mengisi detail transaksi seperti jumlah, harga sepakat, dan total harga.
- Menyimpan transaksi ke database

#### b. Struktur Data

- ID Transaksi: integer
- ID Pengajuan: integer

- ID Users (Petani): integer
  - ID Users (Pabrik): integer
  - Tanggal Transaksi: date
  - Jumlah (kg): integer
  - Harga Sepakat (per kg): integer
  - Total Harga: integer
  - Status Transaksi: string
- c. Rancangan Langkah Penggunaan
- Pilih menu “Transaksi Pembelian”.
  - Pilih ID pengajuan yang telah diterima.
  - Masukkan harga sepakat.
  - Sistem menghitung total: jumlah  $\times$  harga.
  - Simpan transaksi ke database.

#### 4.9 Fitur Melihat Users

Fitur ini memungkinkan admin untuk melihat siapa saja orang yang telah melakukan registrasi dan melakukan login.

```
===== MENU ADMIN =====
1. Lihat Semua User
2. Hapus User
3. Lihat Semua Pengajuan
4. Lihat Semua Transaksi
5. Logout

Pilih (1-5): 1
koneksi berhasil

===== DAFTAR USER =====
ID: 3 | Username: excel | Role: 2
ID: 4 | Username: Rosy | Role: 2
ID: 5 | Username: Laili | Role: 2
ID: 6 | Username: Eli | Role: 2
ID: 7 | Username: rosy | Role: 2
ID: 8 | Username: layee | Role: 2
ID: 10 | Username: Lili | Role: 2
ID: 11 | Username: eLI | Role: 3
ID: 12 | Username: lili | Role: 2
ID: 13 | Username: edi | Role: 3
ID: 14 | Username: dina | Role: 2
ID: 15 | Username: lililay | Role: 2
ID: 16 | Username: lila | Role: 2
```

1. kebutuhan sistem

- Menampilkan seluruh data pengguna (petani maupun pabrik).
  - Menampilkan data dari tabel users.
2. Struktur data
    - ID Users: integer
    - Nama: string
    - Role: string
  3. Rancangan Langkah Penggunaan
    - Pilih menu “Lihat Semua Users”.
    - Sistem mengambil seluruh data dari tabel users.
    - Data ditampilkan dalam bentuk tabel.

#### 4.10 Fitur Melihat Pengajuan

Fitur ini memungkinkan admin untuk melihat pengajuan transaksi yang dilakukan oleh para petani.

```
===== MENU ADMIN =====
1. Lihat Semua User
2. Hapus User
3. Lihat Semua Pengajuan
4. Lihat Semua Transaksi
5. Logout

Pilih (1-5): 3
koneksi berhasil

===== SEMUA PENGAJUAN =====
ID: 8 | User: 14 | Harga: Rp 50000.0 | Status: nan
ID: 7 | User: 14 | Harga: Rp 30000.0 | Status: 1.0
ID: 6 | User: 12 | Harga: Rp 50000.0 | Status: 1.0
ID: 5 | User: 10 | Harga: Rp 50000.0 | Status: 1.0
ID: 4 | User: 8 | Harga: Rp 50000.0 | Status: nan
ID: 3 | User: 5 | Harga: Rp 20000.0 | Status: nan
ID: 2 | User: 5 | Harga: Rp 50000.0 | Status: 1.0
ID: 1 | User: 4 | Harga: Rp 50000.0 | Status: nan
```

1. kebutuhan sistem
  - Menampilkan seluruh data pengajuan yang masuk ke pabrik.
  - Data bersumber dari tabel pengajuan.
2. Struktur data
  - ID Pengajuan
  - ID Lahan
  - Jumlah (kg)

- Harga per kg
  - Status Pengajuan
  - Tanggal Pengajuan
3. Rancangan langkah penggunaan
    - Pilih menu “Lihat Semua Pengajuan”.
    - Sistem mengambil data dari tabel pengajuan.
    - Tampilkan status

#### 4.11 Fitur Melihat Transaksi

Fitur ini memungkinkan admin untuk melihat semua transaksi yang dilakukan oleh pihak pabrik, yang mana hal tersebut berasal dari pengajuan yang dilakukan oleh petani.

```
===== MENU ADMIN =====
1. Lihat Semua User
2. Hapus User
3. Lihat Semua Pengajuan
4. Lihat Semua Transaksi
5. Logout

Pilih (1-5): 4
koneksi berhasil

===== SEMUA TRANSAKSI =====
ID: 5 | User: 5 | Jumlah: 5.0 kg | Harga: Rp 50000.0 | Total: Rp 250,000
ID: 4 | User: 14 | Jumlah: 70.0 kg | Harga: Rp 70000.0 | Total: Rp 4,900,000
ID: 3 | User: 14 | Jumlah: 20.0 kg | Harga: Rp 30000.0 | Total: Rp 600,000
ID: 2 | User: 12 | Jumlah: 30.0 kg | Harga: Rp 50000.0 | Total: Rp 1,500,000
ID: 1 | User: 10 | Jumlah: 10.0 kg | Harga: Rp 40000.0 | Total: Rp 400,000
```

1. Kebutuhan Sistem
  - Menampilkan daftar transaksi pembelian yang telah dilakukan pabrik.
  - Data berasal dari database.
2. Struktur Data
  - ID Transaksi
  - Tanggal Transaksi
  - ID Pengajuan
  - Jumlah (kg)
  - Harga Sepakat
  - Total Harga

- Status Transaksi
3. Rancangan langkah penggunaan
- Pilih menu “Lihat Transaksi”.
  - Sistem menampilkan seluruh transaksi pembelian.
  - Data disajikan dalam bentuk tabel.



## DAFTAR PUSTAKA

- Wardhono, A., Arifandi, J. A., & Indrawati, Y. (2019). *Standar dan mutu tembakau besuki na-oogst*. Pustaka Abadi.
- Wardhono, A., Arifandi, R., & Indrawati, D. (2019). *Buku standar dan mutu tembakau*. Universitas Jember.
- Saputri, A. (2017). *Kajian karakteristik lahan tembakau (Nicotiana tabacum) varietas Kemloko di Kabupaten Temanggung* (Skripsi). Universitas Brawijaya.
- Alvianto, R. (2017). *Analisis efisiensi produksi usahatani tanaman tembakau di Desa Sumberejo, Kecamatan Mranggen, Kabupaten Demak* (Skripsi). Universitas Diponegoro.