

## **TUGAS PROJECT SISTEM BASIS DATA**



### **"AGROMAME" : SISTEM PENDATAAN PRODUKSI DAN STOK EDAMAME DI PT MITRATANI DUA TUJUH**

Disusun Oleh  
Kelompok C01 :

Laili Rizqi Aulia	252410102035
Eka Lailatur Rosyidah	252410102069
Rizka Akhnas	252410102076
Gabriela Christtodevi P.K.	252410102099

**PROGRAM STUDI TEKNOLOGI INFORMASI  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS JEMBER**

**2025**

# **BAB 1**

## **REQUIREMENT ANALYSIS**

### **1.1. Deskripsi sistem yang akan dibuat**

Mitratani Dua Tujuh merupakan perusahaan swasta yang bergerak di bidang agroindustri, dengan edamame sebagai produk unggulan. Perusahaan ini mengelola proses produksi mulai dari menerima hasil panen dari petani edamame, melakukan pengolahan edamame menjadi produk dengan tiga jenis (fresh, frozen dan jus), kemudian distribusi produk edamame beku untuk pasar ekspor. Dalam operasionalnya, PT Mitratani Dua Tujuh masih menggunakan sistem pencatatan manual untuk pengelolaan data hasil panen, stok gudang, dan pelaporan produksi. Hal ini menyebabkan tingginya risiko kesalahan input, duplikasi data, serta keterlambatan dalam pelaporan. Oleh karena itu, sistem ini dikembangkan untuk mendigitalisasi proses pencatatan dan pelaporan, guna meningkatkan efisiensi, akurasi data, dan mendukung pengambilan keputusan yang lebih cepat dan tepat.

Sistem ini memiliki beberapa fitur yang dapat membantu dalam mengelola proses produksi hasil panen dari petani mitra hingga tahap penyimpanan produk di gudang. Sistem ini akan digunakan 4 jenis pengguna yang terdiri dari Admin Produksi, Admin Gudang, Manajer Produksi, dan Pelanggan dengan fitur yang disesuaikan dengan kebutuhan setiap pengguna. Admin Produksi akan memiliki akses untuk mencatat data hasil panen dari petani mitra, dan menginput data produksi harian. Admin Produksi juga memiliki akses untuk melihat laporan hasil produksi harian dan bulanan untuk memantau perkembangan kegiatan produksi.

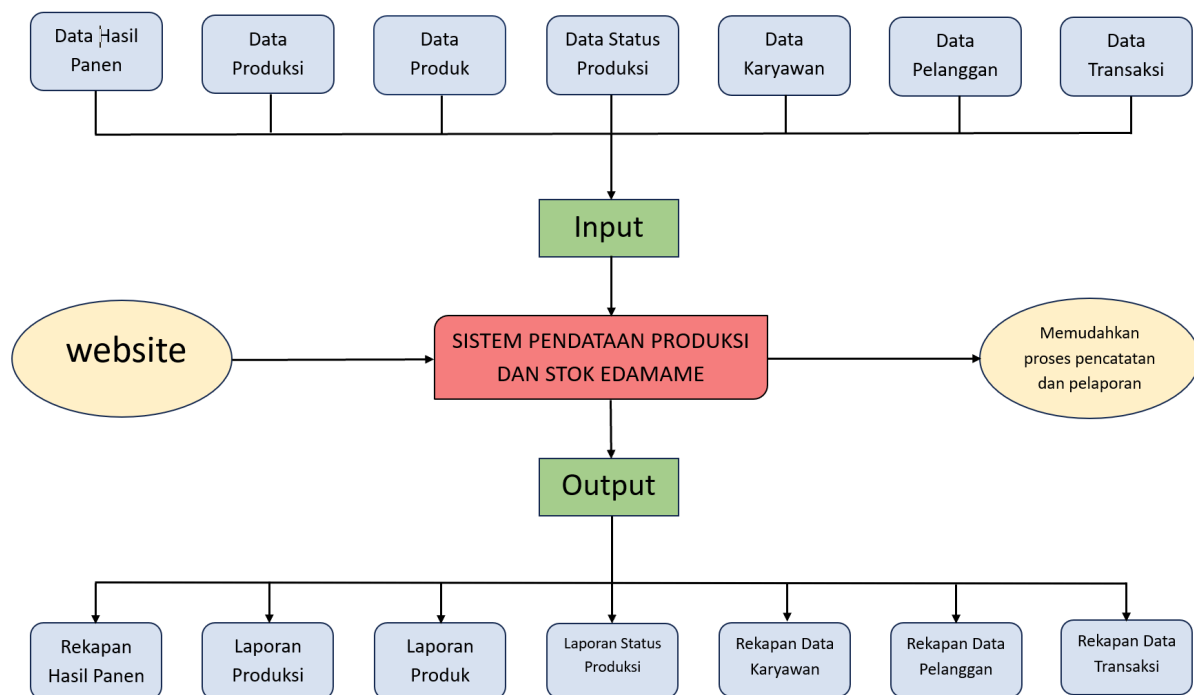
Kemudian, Admin Gudang juga mengelola data stok yang ada di gudang dengan mengupdate stok masuk melalui entity produk, memasukkan produk berdasarkan jenis produknya (fresh, frozen, dan jus), dan memantau stok keluar melalui entity transaksi. Sistem akan otomatis menghitung jumlah stok yang tersisa.

Selanjutnya, Manajer Produksi akan memantau seluruh aktivitas dengan menggunakan laporan dalam bentuk tabel yang disediakan oleh sistem. Laporan tersebut terdiri dari data hasil panen, dan jumlah stok bahan. Selain itu, Manajer Produksi memiliki akses untuk mengelola akun pengguna seperti Admin Produksi dan Admin Gudang. Manajer Produksi juga memiliki akses dalam melihat riwayat transaksi yang dilakukan oleh pelanggan.

Kemudian, pelanggan dapat melakukan registrasi untuk membuat akun lalu login untuk mengakses sistem dan logout saat selesai menggunakan layanan. Setelah masuk, pelanggan dapat melakukan transaksi dengan cara memilih produk yang tersedia, menentukan jumlah yang diinginkan, dan melakukan proses pembayaran sesuai metode yang disediakan. Selain itu, pelanggan juga dapat memperbarui biodata mereka, seperti nama, alamat, nomor telepon, dan informasi lainnya untuk memastikan data yang tersimpan tetap akurat dan terkini.

Sistem baru ini diharapkan mampu mengoptimalkan pencatatan dan pelaporan produksi PT Mitratani Dua Tujuh, membuatnya lebih terstruktur, akurat, dan efisien. Selain itu, Sistem ini juga memungkinkan perusahaan untuk menjaga efisiensi dan skalabilitas proses kerja, bahkan ketika volume produksi dan jumlah mitra petani terus meningkat.

## 1.2 Business process organisasi



Sistem Pendataan Produksi dan Stok Edamame di PT Mitratani Dua Tujuh menggunakan sistem berbasis console. Sistem pendataan ini bertujuan untuk mencatat

data terkait hasil panen, stok, serta proses produksi edamame agar memudahkan bagian produksi dan gudang dalam memonitoring kegiatan produksi serta mempercepat proses pelaporan. Sistem ini juga membantu Manajer Produksi dalam mengontrol jalannya produksi secara efisien dan akurat. Adapun input dan output dari sistem pengelolaan di atas sebagai berikut:

#### 1. Data Hasil Panen

Data Hasil Panen merupakan catatan hasil panen edamame yang diperoleh dari petani mitra PT Mitratani Dua Tujuh. Data hasil panen diinput oleh Admin Produksi setelah proses panen selesai dilakukan. Inputan dari data hasil panen yaitu ID hasil panen, tanggal panen, dan jumlah panen dalam satuan kg. Output dari data hasil panen berupa rekapitan data hasil panen.

#### 2. Data Produksi

Data Produksi merupakan catatan aktivitas produksi edamame yang dilakukan setiap hari. Data ini diinput oleh Admin Produksi untuk mendokumentasikan seluruh proses pengolahan bahan baku menjadi produk siap olah. Inputan dari data produksi yaitu ID produksi, tanggal produksi, tanggal selesai, Status Produksi dan jumlah produksi. Output dari data produksi berupa laporan produksi yang menampilkan rincian proses produksi setiap hari beserta hasil yang dicapai.

#### 3. Data Produk

Data Produk merupakan data yang berisi hasil produksi edamame berdasarkan kategori jenis produk, yaitu frozen, fresh, dan jus. Data ini digunakan untuk mengelompokkan hasil akhir dari proses produksi. Inputan dari data hasil olahan yaitu ID Produk, jenis produk, stok, tanggal update stok, dan harga produk. Output dari data hasil olahan berupa laporan hasil olahan yang menampilkan jumlah stok yang dihasilkan pada setiap kategori(Fresh, Frozen dan jus).

#### 4. Data Status Produksi

Data Status Produksi berisi penjelasan mengenai berbagai tahapan atau kondisi yang menggambarkan proses dari produksi seperti proses, ditunda, dan selesai. Status “proses” menunjukkan bahwa kegiatan produksi sedang berlangsung. Status “ditunda”

menggambarkan ketika produksi mengalami berhenti sementara karena mengalami hambatan di beberapa faktor seperti keterbatasan bahan baku, hambatan pada tenaga kerja, dan masalah teknis lainnya. Status yang terakhir adalah “selesai”, status ini menunjukkan tahap produksi sudah di tahap selesai. Pada tahap ini, produk hasil produksi sudah siap untuk melanjutkan ke tahap berikutnya.

#### 5. Data Karyawan

Data Karyawan berisi informasi seluruh pengguna yang terlibat dalam sistem, meliputi Admin Produksi, Admin Gudang, dan Manajer Produksi. Setiap peran memiliki jumlah pengguna berbeda, di mana admin bisa lebih dari satu, sedangkan manajer hanya satu. Inputan dari data karyawan meliputi ID karyawan, nama, jabatan, username, kata sandi, gender, tanggal lahir, no telpon, tanggal masuk, status karyawan dan gaji karyawan, sedangkan outputnya berupa rekapan data seluruh karyawan.

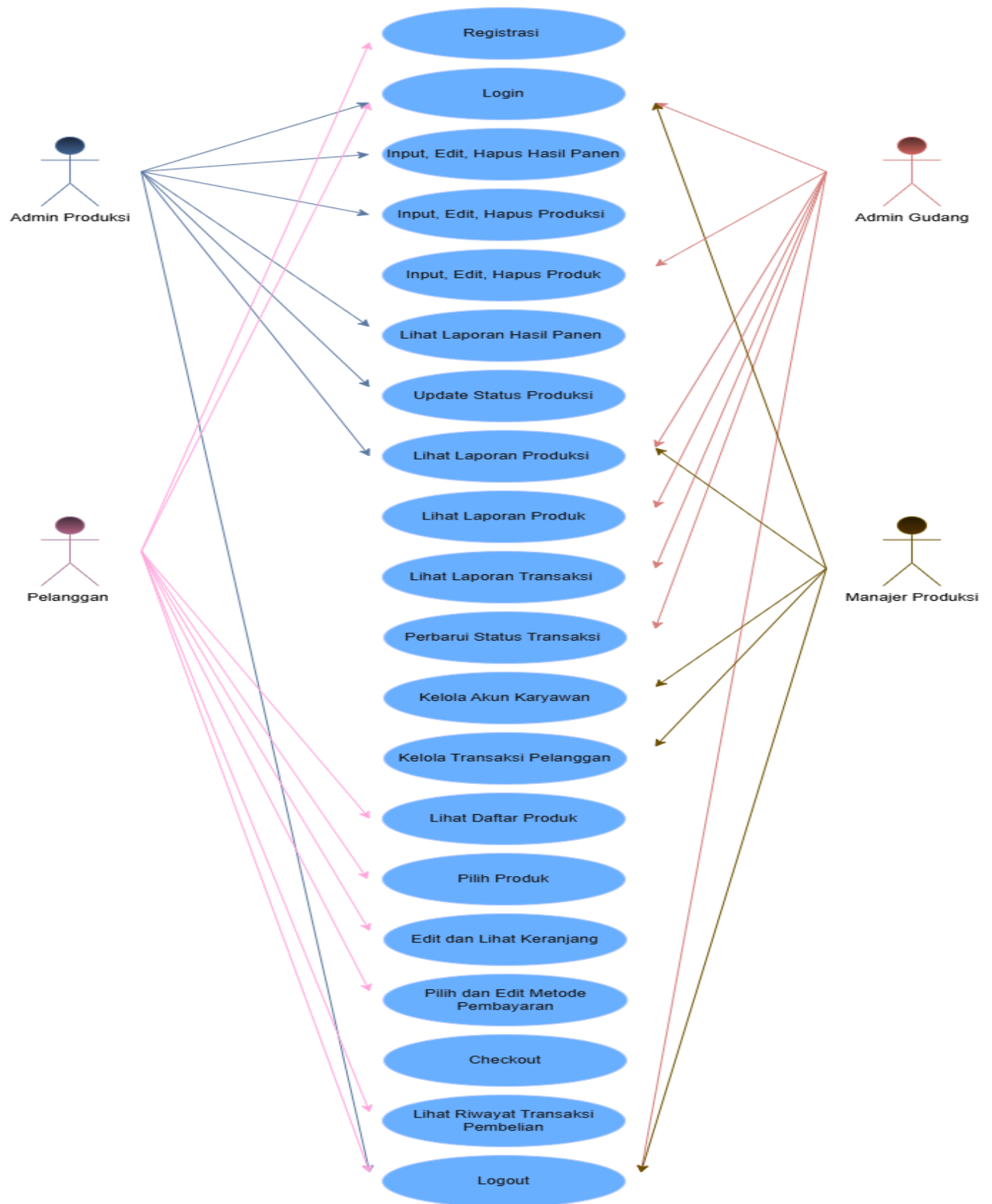
#### 6. Data Pelanggan

Data Pelanggan merupakan informasi mengenai pihak luar yang berhubungan dengan perusahaan yaitu seperti pelanggan. Data ini digunakan untuk mencatat identitas pelanggan yang melakukan pemesanan atau pembelian produk dari PT Mitratani Dua Tujuh. Inputan dari data pelanggan meliputi ID Pelanggan, nama pelanggan, nomor telepon pelanggan, alamat, kata sandi pelanggan, username, dan tanggal registrasi. Output dari data pelanggan berupa rekapan data seluruh pelanggan yang menampilkan informasi tentang pelanggan.

#### 7. Data Transaksi

Data Transaksi merupakan catatan aktivitas jual beli produk edamame yang dilakukan antara perusahaan dan pelanggan. Data ini berfungsi untuk mendokumentasikan proses permintaan produk, jumlah barang yang keluar, serta kesesuaian antara stok dan distribusi. Inputan dari data transaksi meliputi Id transaksi, status transaksi, metode pembayaran, tanggal transaksi. Output dari data transaksi berupa rekapan transaksi yang menampilkan detail pembelian yang dilakukan oleh pelanggan.

### 1.3. Use Case Diagram



Use case dalam Sistem Pendataan Produksi dan Stok Edamame di PT Mitratani Dua Tujuh memiliki empat aktor utama, yaitu Admin Produksi, Admin Gudang, Manajer Produksi dan Pelanggan. Masing-masing aktor memiliki hak akses dan batasan yang berbeda sesuai dengan peran dan fungsinya dalam sistem. Berikut penjelasan dari use case tersebut.

### **1. Manajer Produksi**

Manajer Produksi adalah pengguna yang bertugas memantau seluruh proses produksi edamame. Manajer Produksi dapat mengakses sistem untuk melihat laporan dan mengelola akun pengguna. Berikut adalah hak akses dan fitur yang dapat dilakukan oleh Manajer Produksi:

- ❖ Login : Manajer Produksi masuk ke sistem menggunakan akun yang telah didaftarkan
- ❖ Lihat Laporan Produksi : Manajer Produksi dapat melihat seluruh laporan hasil panen
- ❖ Kelola Akun Karyawan : Manajer bisa lihat, tambah edit, dan hapus akun Admin Produksi dan Admin Gudang
- ❖ Kelola Transaksi Pelanggan : Manajer dapat memantau jalannya transaksi pelanggan dan melihat riwayat transaksi
- ❖ Logout : Setelah selesai memantau laporan dan mengelola akun pengguna, Manajer Produksi keluar dari sistem untuk menandai bahwa sesi itu telah selesai

### **2. Admin Produksi**

Admin Produksi adalah pengguna yang bertugas mencatat dan mengelola data hasil panen edamame. Admin Produksi dapat mengakses sistem untuk melakukan berbagai aktivitas terkait pencatatan produksi. Berikut adalah hak akses dan fitur yang dapat dilakukan oleh Admin Produksi:

- ❖ Login : Admin Produksi masuk ke sistem menggunakan akun yang telah didaftarkan
- ❖ Input, Edit, Hapus Hasil Panen : Admin Produksi dapat menginput, mengedit, dan menghapus hasil panen
- ❖ Input, Edit, Hapus Produksi : Admin produksi juga bisa menginput, mengedit, dan menghapus hasil produksi
- ❖ Lihat Laporan Hasil Panen : Admin Produksi dapat melihat hasil panennya
- ❖ Update Status Produksi : Dapat memperbarui status proses produksi (dalam proses, ditunda atau selesai)
- ❖ Lihat Laporan Produksi : Admin Produksi dapat melihat seluruh laporan hasil panen
- ❖ Logout : Keluar dari sistem setelah proses pencatatan selesai

### **3. Admin Gudang**

Admin Gudang adalah pengguna yang bertugas mencatat dan mengelola data stok edamame di gudang. Admin Gudang dapat mengakses sistem untuk melakukan berbagai aktivitas terkait pengelolaan stok. Berikut adalah hak akses dan fitur yang dapat dilakukan oleh Admin Gudang:

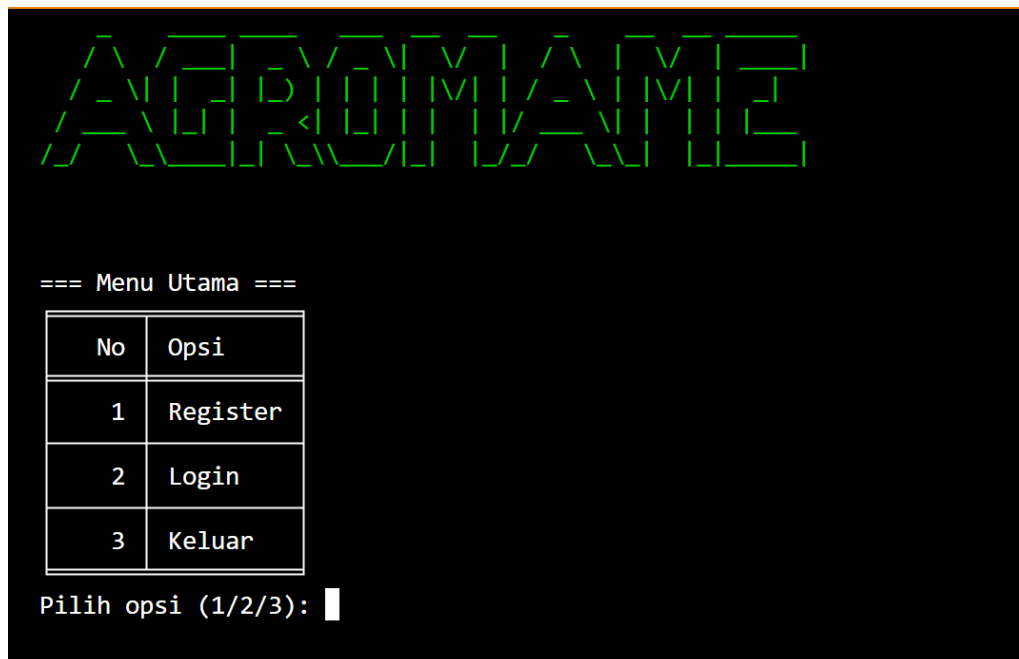
- ❖ Login : Admin Gudang masuk ke sistem menggunakan akun yang telah didaftarkan.
- ❖ Input, Edit, Hapus Produk : Admin Gudang bisa menginput, mengedit, dan menghapus data produk
- ❖ Lihat Laporan Produksi : Admin Gudang dapat melihat seluruh laporan produksi
- ❖ Lihat Laporan Produk : Admin Gudang dapat melihat seluruh laporan produk
- ❖ Lihat Laporan Transaksi : Admin Gudang dapat memantau transaksi pelanggan
- ❖ Perbarui Status Transaksi : Admin Gudang dapat memperbarui status transaksi pelanggan seperti selesai, proses, dan dibatalkan
- ❖ Logout : Keluar dari sistem setelah proses pencatatan selesai

#### **4. Pelanggan**

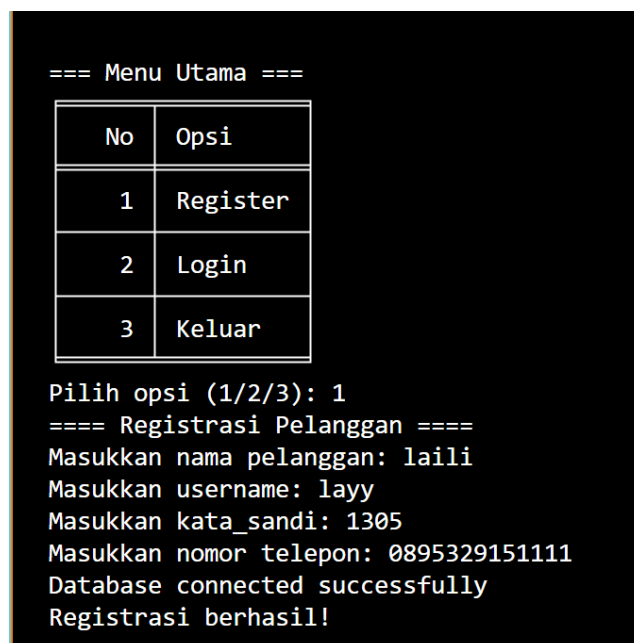
- ❖ Registrasi : Pelanggan dapat membuat akun baru dengan mengisi data diri pada sistem
- ❖ Login : Pelanggan masuk ke sistem menggunakan akun yang telah didaftarkan
- ❖ Lihat Daftar Produk : Pelanggan dapat melihat daftar produk edamame beserta ketersediaan stok dan detail harga
- ❖ Pilih Produk : Pelanggan dapat melakukan pemesanan produk sesuai kategori pembeli
- ❖ Edit dan Lihat Keranjang : Pelanggan dapat melihat pesanan yang sudah di input waktu pilih produk di keranjang dan juga bisa mengeditnya jika ada yang ingin diubah dan dihapus
- ❖ Pilih dan Edit Metode Pembayaran : Melakukan pemilihan metode pembayaran atas pesanan yang telah dibuat dan dapat diedit jika ingin diubah
- ❖ Lihat Riwayat Transaksi Pembelian : Pelanggan dapat melihat riwayat transaksi yang pernah dilakukan
- ❖ Logout : Keluar dari sistem setelah selesai melakukan pemesanan atau memperbarui data



## 1.4. Desain User Interface



Gambar 1.4.1 Halaman Utama



Gambar 1.4.2 Registrasi

```
=== Menu Utama ===  


| No | Opsi     |
|----|----------|
| 1  | Register |
| 2  | Login    |
| 3  | Keluar   |

  
Pilih opsi (1/2/3): 2  
  
==== Login ====  


| No. | Opsi      |
|-----|-----------|
| 1   | Karyawan  |
| 2   | Pelanggan |

  
Pilih opsi (1/2): 2  
==== Login Pelanggan ====  
Masukkan username: layy  
Masukkan kata_sandi: 1305  
Database connected successfully
```

Gambar 1.4.3 Login

=== Data Login Karyawan ===

Data	Keterangan
Nama Lengkap	Rosyid
Username	ros
Password	***
Role	Admin Produksi

Selamat datang Rosyid! Anda login sebagai Admin Produksi.

=== Halaman Menu Admin Produksi ===

No	Menu
1	Lihat Biodata
2	Edit Biodata
3	Input Hasil Panen
4	Hapus Hasil Panen
5	Lihat Laporan Hasil Panen
6	Edit Hasil Panen
7	Input Produksi
8	Edit Produksi
9	Hapus Produksi
10	Perbarui Status Produksi
11	Lihat Laporan Produksi
12	Logout

Gambar 1.4.4 Halaman Admin Produksi

=== Data Login Karyawan ===

Data	Keterangan
Nama Lengkap	Bhiyuga
Username	bhi
Password	***
Role	Admin Gudang

Selamat datang Bhiyuga! Anda login sebagai Admin Gudang.

=== Halaman Menu Admin Gudang ===

No	Menu
1	Lihat Biodata
2	Edit Biodata
3	Lihat Laporan Produksi
4	Input Produk
5	Hapus Produk
6	Lihat Laporan Produk
7	Edit Data Produk
8	Update Status Transaksi
9	Lihat Laporan Transaksi
10	Logout

Pilih menu:

Gambar 1.4.5 Halaman Admin Gudang

Pilih opsi (1/2): 1

==== Login Karyawan ====

Username: layy

Password: 0506

Database connected successfully

=== Data Login Karyawan ===

Data	Keterangan
Nama Lengkap	fak
Username	layy
Password	****
Role	Manajer Produksi

Selamat datang fak! Anda login sebagai Manajer Produksi.

=== Halaman Menu Manajer Produksi ===

No	Menu
1	Lihat Biodata
2	Edit Biodata
3	Kelola Akun Karyawan
4	Lihat Laporan Produksi
5	Kelola Transaksi Pelanggan
6	Pergantian Manager Baru
7	Logout

Pilih menu:

Gambar 1.4.6 Halaman Manajer Produksi

=== Data Login Pelanggan ===

Data	Keterangan
Nama Lengkap	laili
Username	layy
Password	****

Selamat datang laili! Anda berhasil login sebagai Pelanggan.

=== AGROMAME: Menu Utama Pembelian ===

1. Lihat Biodata
2. Edit Biodata
3. Lihat Daftar Produk
4. Pilih Produk
5. Lihat Keranjang
6. Edit Keranjang
7. Pilih Metode Pembayaran
8. Edit Metode Pembayaran
9. Checkout
10. Lihat Riwayat Transaksi Pembelian
11. Logout

Pilih menu:

Gambar 1.4.7 Halaman Pelanggan

## BAB 2

### IDENTIFIKASI ENTITY DAN RELATIONSHIP

#### 2.1 Identifikasi entity dan attribute

Entity	Hasil Panen
Kategori	Benda Nyata
Deskripsi	Menyimpan data hasil panen edamame yang diperoleh dari petani mitra PT Mitratani Dua Tujuh
Alasan dimasukkan sebagai entity	Karena hasil panen merupakan data utama yang menjadi dasar proses produksi. Data ini penting untuk mencatat jumlah bahan baku yang tersedia sebelum diproses lebih lanjut.
Atribut	<ul style="list-style-type: none"> <li>· id_karyawan</li> <li>· id_hasil_panen</li> <li>· tanggal_panen</li> <li>· jumlah_panen</li> </ul>
keterangan	Data hasil panen diinput oleh Admin Produksi

Entity	Produksi
Kategori	Benda Abstrak
Deskripsi	Menyimpan data aktivitas produksi edamame yang dilakukan setiap hari
Alasan dimasukkan sebagai entity	Karena produksi merupakan inti kegiatan perusahaan untuk mengubah hasil panen menjadi produk siap jual.
Atribut	<ul style="list-style-type: none"> <li>· id_produksi</li> <li>· tgl_produksi</li> <li>· id_hasil_panen</li> <li>· id_karyawan</li> <li>· id_status_produksi</li> <li>· tanggal_selesai</li> <li>· jumlah_produksi</li> </ul>
keterangan	Data ini diinput oleh Admin Produksi untuk mendokumentasikan proses pengolahan bahan baku setiap hari.

Entity	Produk
Kategori	Benda Nyata
Deskripsi	Menyimpan informasi hasil olahan edamame berdasarkan kategori produk, yaitu frozen, fresh, dan jus.
Alasan dimasukkan sebagai entity	Karena produk merupakan hasil akhir proses produksi dan menjadi objek utama dalam penyimpanan serta distribusi.
Atribut	<ul style="list-style-type: none"> <li>· id_produk</li> <li>· id_produksi</li> <li>· id_jenis_produk</li> <li>· harga_produk</li> <li>· stok</li> <li>· tanggal_update_stok</li> </ul>
keterangan	Data ini diinput oleh admin gudang.

Entity	Karyawan
Kategori	Benda Nyata
Deskripsi	Menyimpan data seluruh pengguna sistem yang terdiri dari Admin Produksi, Admin Gudang, Petani Mitra, dan Manajer Produksi.
Alasan dimasukkan sebagai entity	Karena setiap pengguna memiliki peran dan tanggung jawab yang berbeda, sehingga perlu dikelola secara terpisah.
Atribut	<ul style="list-style-type: none"> <li>· id_karyawan</li> <li>· username</li> <li>· password</li> <li>· nama_karyawan</li> <li>· gender</li> <li>· tgl_lahir</li> <li>· no_telepon</li> <li>· id_jabatan</li> <li>· tgl_masuk</li> <li>· status_karyawan</li> <li>· gaji_karyawan</li> </ul>
keterangan	Data ini dikelola oleh Manajer Produksi untuk mengatur akses setiap pengguna dalam sistem.



Entity	Pelanggan
Kategori	Benda Nyata
Deskripsi	Menyimpan informasi mengenai pihak yang melakukan pembelian produk edamame dari PT Mitratani Dua Tujuh.
Alasan dimasukkan sebagai entity	Karena pelanggan merupakan pihak yang berhubungan langsung dengan proses transaksi penjualan. Data pelanggan diperlukan agar proses transaksi dapat dicatat dan ditelusuri dengan jelas.
Atribut	<ul style="list-style-type: none"> <li>· id_pelanggan</li> <li>· username</li> <li>· password</li> <li>· nama_pelanggan</li> <li>· no_telp_pelanggan</li> <li>· alamat</li> </ul>
keterangan	Data pelanggan diinput ketika ada pihak yang melakukan pemesanan atau pembelian produk, sehingga perusahaan dapat menyimpan informasi kontak dan kebutuhan pelanggan.

Entity	Transaksi
Kategori	Benda Abstrak
Deskripsi	Menyimpan catatan transaksi penjualan produk edamame kepada pelanggan.
Alasan dimasukkan sebagai entity	Karena transaksi merupakan aktivitas penting yang menunjukkan pergerakan produk dari gudang menuju pelanggan dan menjadi dasar pencatatan stok keluar.
Atribut	<ul style="list-style-type: none"> <li>· id_transaksi</li> <li>· id_pelanggan</li> <li>· tgl_transaksi</li> <li>· id-karyawan</li> <li>· id_metode_pembayaran</li> <li>· id_status_transaksi</li> </ul>
keterangan	Data transaksi dipantau oleh admin gudang untuk memantau stok keluar.

## 2.2 Relationship Matrix

	Hasil Panen	Produksi	Produk	Karyawan	Pelanggan	Transaksi
Hasil Panen		✓		✓		
Produksi	✓		✓	✓		
Produk		✓				✓
Karyawan	✓	✓				✓
Pelanggan						✓
Transaksi			✓	✓	✓	

### Relationship

#### 1. Hasil Panen :

Hasil Panen – Produksi

Hasil Panen – Karyawan

#### 2. Produksi :

Produksi – Hasil Panen

Produksi – Produk

Produksi – Karyawan

#### 3. Produk

Produk – Produksi

Produk - Transaksi

#### 4. Karyawan :

Karyawan – Hasil Panen

Karyawan – Produksi

Karyawan – Transaksi

## 5. Pelanggan

Pelanggan – Transaksi

## 6. Transaksi

Transaksi – Produk

Transaksi – Karyawan

Transaksi – Pelanggan

## Cardinality

### 1. Relationship Hasil Panen – Karyawan

– Cardinality

Many to One : Many disisi Hasil Panen dan One disisi Karyawan

– Optionality

Optionality Hasil Panen : Wajib

Optionality Produk : Tidak Wajib

### 2. Relationship Hasil Panen – Produksi

– Cardinality

One to Many : One disisi Hasil Panen dan Many disisi Produksi

– Optionality

Optionality Hasil Panen : Tidak Wajib

Optionality Produk : Wajib

### 3. Relationship Produksi – Produk

– Cardinality

One to Many : One disisi Produksi dan Many disisi Produk

– Optionality

Optionality Produksi : Tidak Wajib

Optionality Produk : Wajib

### 4. Relationship Produksi – Karyawan

– Cardinality

Many to One : Many disisi Produksi dan One disisi Karyawan

– Optionality

Optionality Produksi : Wajib

Optionality Karyawan : Tidak Wajib

## 5. Relationship Produk – Transaksi

– Cardinality

Many to Many : Many disisi Produk dan Many disisi Transaksi

– Optionality

Optionality Produk : Tidak Wajib

Optionality Transaksi : Wajib

## 6. Relationship Karyawan - Transaksi

– Cardinality

One to Many : One disisi Karyawan dan Many disisi Transaksi

– Optionality

Optionality Karyawan : Tidak Wajib

Optionality Transaksi : Wajib

## 7. Relationship Pelanggan - Transaksi

– Cardinality

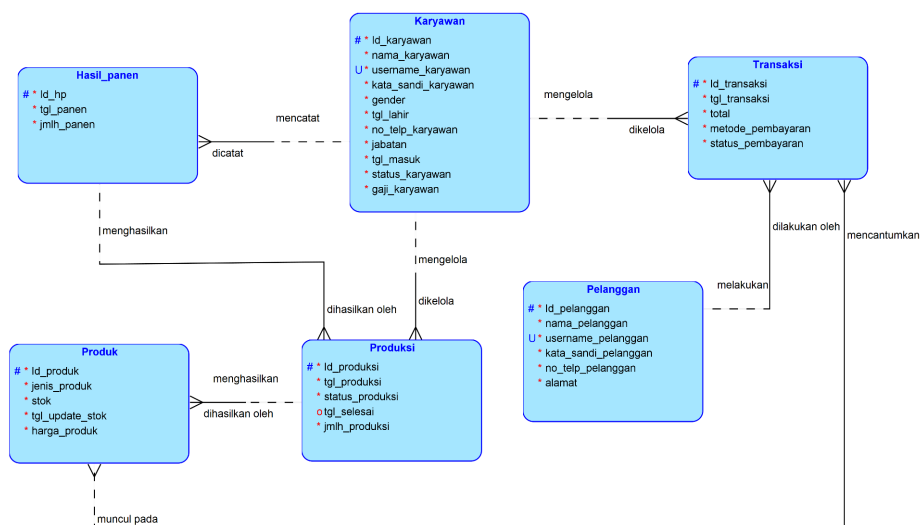
One to Many : One disisi Pelanggan dan Many disisi Transaksi

– Optionality

Optionality Pelanggan : Tidak Wajib

Optionality Transaksi : Wajib

### 2.3. Conceptual design awal (ERD awal)



Link Gambar:

[https://drive.google.com/file/d/1HhsJkwA5CcKGNd-00h71q38g\\_uXyWM30/view?usp=sharing](https://drive.google.com/file/d/1HhsJkwA5CcKGNd-00h71q38g_uXyWM30/view?usp=sharing)

	Hasil Panen	Produksi	Produk	Karyawan	Pelanggan	Transaksi
Hasil Panen		menghasilkan		dicatat		
Produksi	dihasilkan oleh		menghasilkan	dikelola		
Produk		dihasilkan oleh				muncul pada
Karyawan	mencatat	mengelola				mengelola
Pelanggan						melakukan
Transaksi			mencantumkan	dikelola	dilakukan oleh	

### Keterangan:

#### 1. Hasil Panen

Menyimpan data terkait hasil panen dengan atribut sebagai berikut	
id_hp (#*)	<ul style="list-style-type: none"><li>- Tipe data INTEGER</li><li>- Sebagai Primary Key</li><li>- Tidak Boleh Kosong</li></ul>
tgl_panen (*)	<ul style="list-style-type: none"><li>- Tipe data DATE</li><li>- Tidak Boleh Kosong</li></ul>
jmlh_panen (*)	<ul style="list-style-type: none"><li>- Tipe data FLOAT</li><li>- Tidak Boleh Kosong</li></ul>

#### 2. Produksi

Menyimpan data terkait produk dengan atribut sebagai berikut	
id_produksi (#*)	<ul style="list-style-type: none"><li>- Tipe data INTEGER</li><li>- Sebagai Primary Key</li><li>- Tidak Boleh Kosong</li></ul>
tgl_produksi (*)	<ul style="list-style-type: none"><li>- Tipe data DATE</li><li>- Tidak Boleh Kosong</li></ul>
status_produksi (*)	<ul style="list-style-type: none"><li>- Tipe data VARCHAR size 10</li><li>- Tidak Boleh Kosong</li></ul>

tgl_selesai (0)	<ul style="list-style-type: none"> <li>- Tipe data DATE</li> <li>- Boleh Kosong</li> </ul>
jmlh_produksi (*)	<ul style="list-style-type: none"> <li>- Tipe data FLOAT</li> <li>- Tidak Boleh Kosong</li> </ul>

### 3. Produk

Menyimpan data terkait produk dengan atribut sebagai berikut	
id_produk (#*)	<ul style="list-style-type: none"> <li>- Tipe data INTEGER</li> <li>- Sebagai Primary Key</li> <li>- Tidak Boleh Kosong</li> </ul>
jenis_produk (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 100</li> <li>- Tidak Boleh Kosong</li> </ul>
stok (*)	<ul style="list-style-type: none"> <li>- Tipe data INTEGER</li> <li>- Tidak Boleh Kosong</li> </ul>
tgl_update_stok (*)	<ul style="list-style-type: none"> <li>- Tipe data DATE</li> <li>- Tidak Boleh Kosong</li> </ul>
harga_produk (*)	<ul style="list-style-type: none"> <li>- Tipe data MONEY</li> <li>- Tidak Boleh Kosong</li> </ul>

### 4. Pelanggan

Menyimpan data terkait pelanggan dengan atribut sebagai berikut	
id_pelanggan (#*)	<ul style="list-style-type: none"> <li>- Tipe data INTEGER</li> <li>- Sebagai Primary Key</li> <li>- Tidak Boleh Kosong</li> </ul>
nama_pelanggan (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 128</li> <li>- Tidak Boleh Kosong</li> </ul>
username_pelanggan (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 20</li> <li>- Tidak Boleh Kosong</li> </ul>
kata_sandi_pelanggan (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 8</li> <li>- Tidak Boleh Kosong</li> </ul>
no_telp_pelanggan (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 20</li> <li>- Tidak Boleh Kosong</li> </ul>
alamat (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 128</li> <li>- Tidak Boleh Kosong</li> </ul>

## 5. Karyawan

Menyimpan data terkait karyawan dengan atribut sebagai berikut	
id_karyawan (#*)	<ul style="list-style-type: none"><li>- Tipe data INTEGER</li><li>- Sebagai Primary Key</li><li>- Tidak boleh kosong</li></ul>
nama_karyawan (*)	<ul style="list-style-type: none"><li>- Tipe data VARCHAR size 128</li><li>- Tidak Boleh Kosong</li></ul>
username_karyawan (*)	<ul style="list-style-type: none"><li>- Tipe data VARCHAR size 20</li><li>- Tidak Boleh Kosong</li></ul>
kata_sandi_karyawan (*)	<ul style="list-style-type: none"><li>- Tipe data VARCHAR size 8</li><li>- Tidak Boleh Kosong</li></ul>
gender (*)	<ul style="list-style-type: none"><li>- Tipe data BOOLEAN size 1</li><li>- Tidak Boleh Kosong</li></ul>
tgl_lahir (*)	<ul style="list-style-type: none"><li>- Tipe data DATE</li><li>- Tidak Boleh Kosong</li></ul>
no_telp_karyawan (*)	<ul style="list-style-type: none"><li>- Tipe data VARCHAR size 20</li><li>- Tidak Boleh Kosong</li></ul>
jabatan (*)	<ul style="list-style-type: none"><li>- Tipe data VARCHAR size 100</li><li>- Tidak Boleh Kosong</li></ul>
tgl_masuk (*)	<ul style="list-style-type: none"><li>- Tipe data DATE</li><li>- Tidak Boleh Kosong</li></ul>
status_karyawan (*)	<ul style="list-style-type: none"><li>- Tipe data BOOLEAN size 1</li><li>- Tidak Boleh Kosong</li></ul>
gaji_karyawan (*)	<ul style="list-style-type: none"><li>- Tipe data MONEY</li><li>- Tidak Boleh Kosong</li></ul>

## 6. Transaksi

Menyimpan data terkait transaksi dengan atribut sebagai berikut	
id_transaksi (#*)	<ul style="list-style-type: none"><li>- Tipe data INTEGER</li><li>- Sebagai Primary Key</li><li>- Tidak boleh kosong</li></ul>
tgl_transaksi (*)	<ul style="list-style-type: none"><li>- Tipe data DATE</li><li>- Tidak Boleh Kosong</li></ul>
total (*)	<ul style="list-style-type: none"><li>- Tipe data INTEGER</li><li>- Tidak Boleh Kosong</li></ul>

metode_pembayaran (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 10</li> <li>- Tidak Boleh Kosong</li> </ul>
status_transaksi (*)	<ul style="list-style-type: none"> <li>- Tipe data VARCHAR size 10</li> <li>- Tidak Boleh Kosong</li> </ul>



## BAB 3

### PENYEMPURNAAN ERD

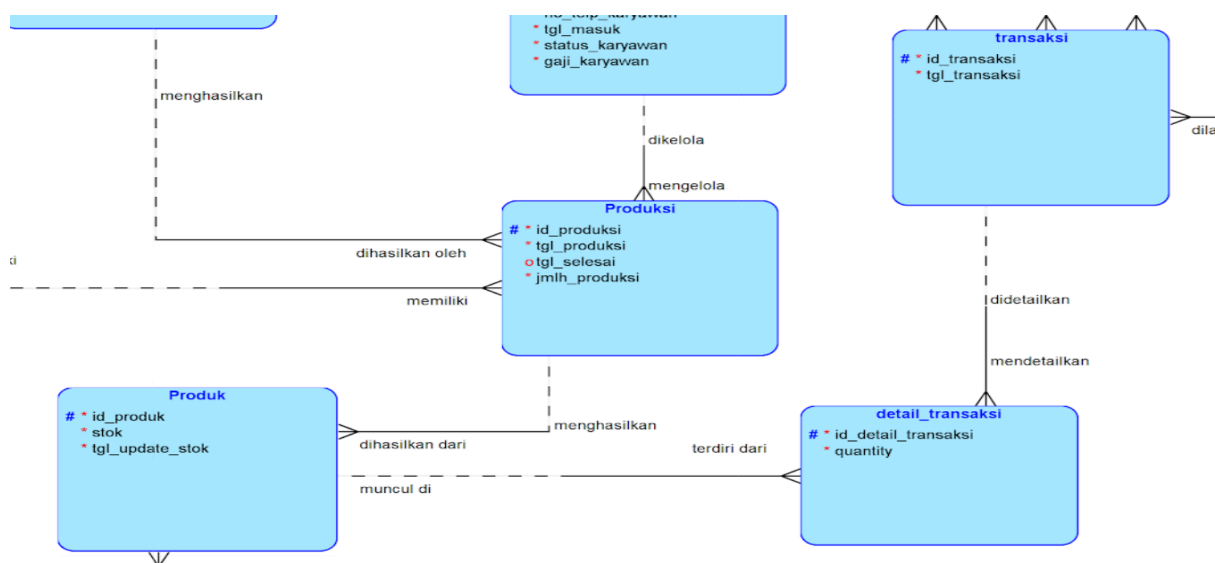
#### 3.1 Pemecahan many to many attribute

Relasi Many to many dalam rancangan ERD ini terjadi antara Entitas Produk dan Entitas Transaksi. Hal tersebut dikarenakan pada saat sebuah transaksi dilakukan, produk yang dibeli bisa lebih dari satu, dan satu produk juga bisa muncul dalam banyak transaksi yang berbeda. Oleh karena itu, relasi antara produk dan transaksi adalah many to many.

Apabila suatu entitas memiliki relasi many to many dengan entitas lain, maka di antara kedua entitas tersebut pasti terdapat atribut yang bersifat multivalue (memiliki lebih dari satu nilai). Untuk menghilangkan multivalue tersebut, perlu dibuat sebuah Entitas penghubung yang menjembatani relasi antara dua entitas tersebut. Dengan adanya entitas penghubung, relasi yang awalnya many to many akan berubah menjadi one to many.

Dalam konteks rancangan ERD ini yaitu untuk menghilangkan relasi many to many antara produk dan transaksi yang akan dilakukan beberapa langkah sebagai berikut :

- Pembuatan entitas baru yaitu detail\_transaksi, dimana relasi many to many dari entitas produk ke entitas transaksi dihapus dan digantikan dengan relasi one to many dari masing-masing entitas ke entitas detail transaksi seperti gambar di bawah ini.



- Dalam entitas detail\_transaksi terdapat atribut utama yaitu id\_detail\_transaksi, quantity, dan harga serta foreign key dari entitas produk dan entitas transaksi. Relasi

ini bersifat many to one terhadap masing-masing entitasnya. Dengan adanya entitas detail\_transaksi ini tidak akan terjadinya multivalued pada atribut relasi antara produk dan transaksi.

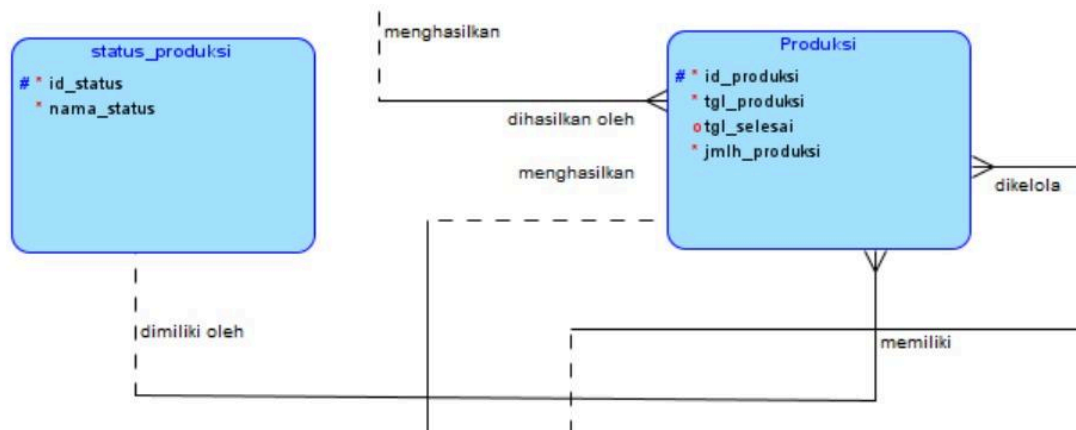
### 3.2 Supertype/subtype dan arc

Pada studi kasus kami, khususnya pada ERD ( Entity Relationship Diagram ) tidak memerlukan supertype/subtype maupun arc.

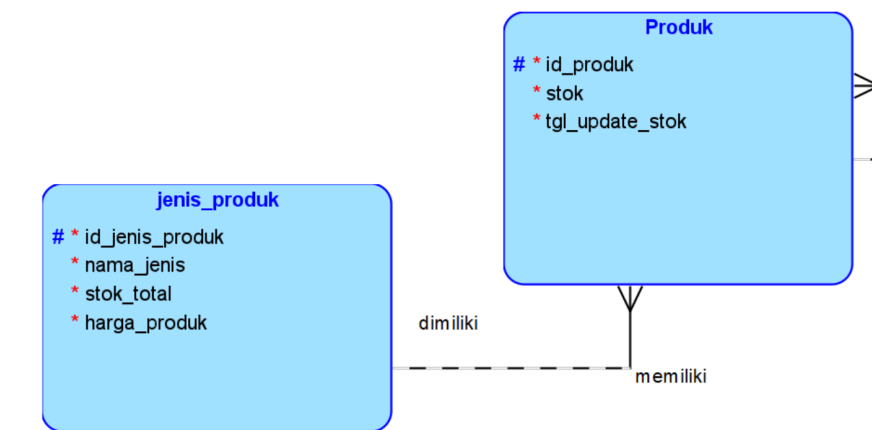
### 3.3 Normalisasi (jika ada tabel yang belum 3NF)

#### 1. 1NF (tidak boleh ada multivalued, redundant, dan composite)

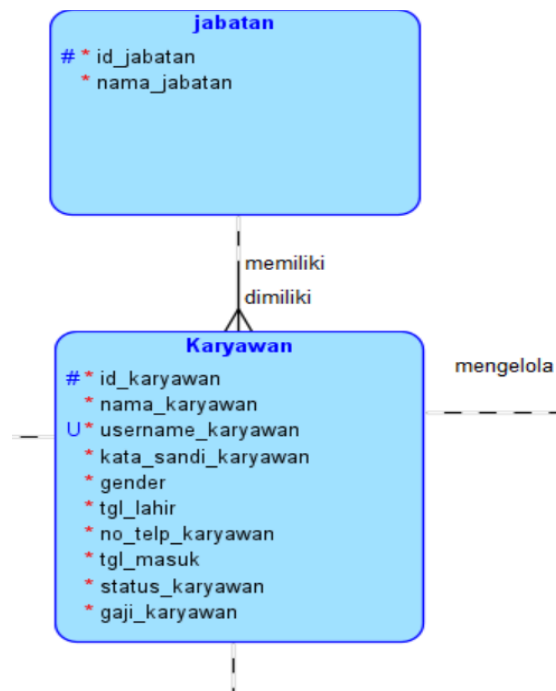
- Pemecahan atribut status produksi yang awalnya berada dalam entitas produksi menjadi entitas baru status\_produksi untuk menghindari duplikasi data dan memastikan setiap status tersimpan sebagai data tersendiri.



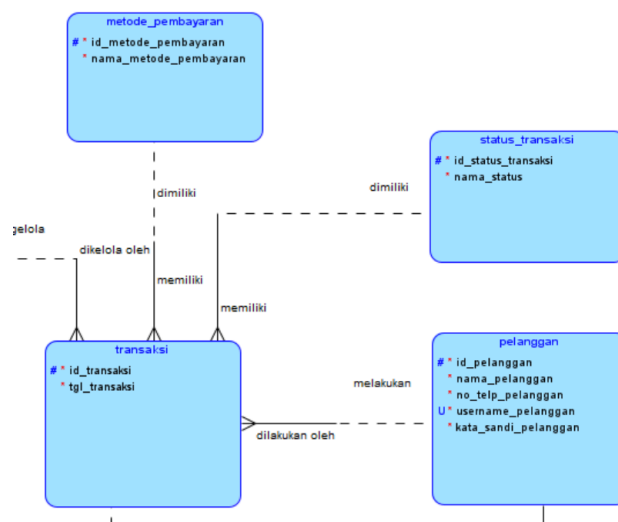
- Pemecahan atribut jenis produk dari entitas produk menjadi entitas jenis produk, karena jenis produk bersifat redundant (frozen, fresh, jus) dan harus disimpan sebagai tabel terpisah.



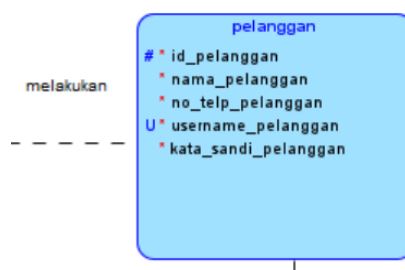
- Pemecahan atribut jabatan pada entitas karyawan menjadi entitas jabatan karena atribut tersebut dapat digunakan oleh banyak karyawan, sehingga harus dipisahkan agar tidak terjadi redudansi data.



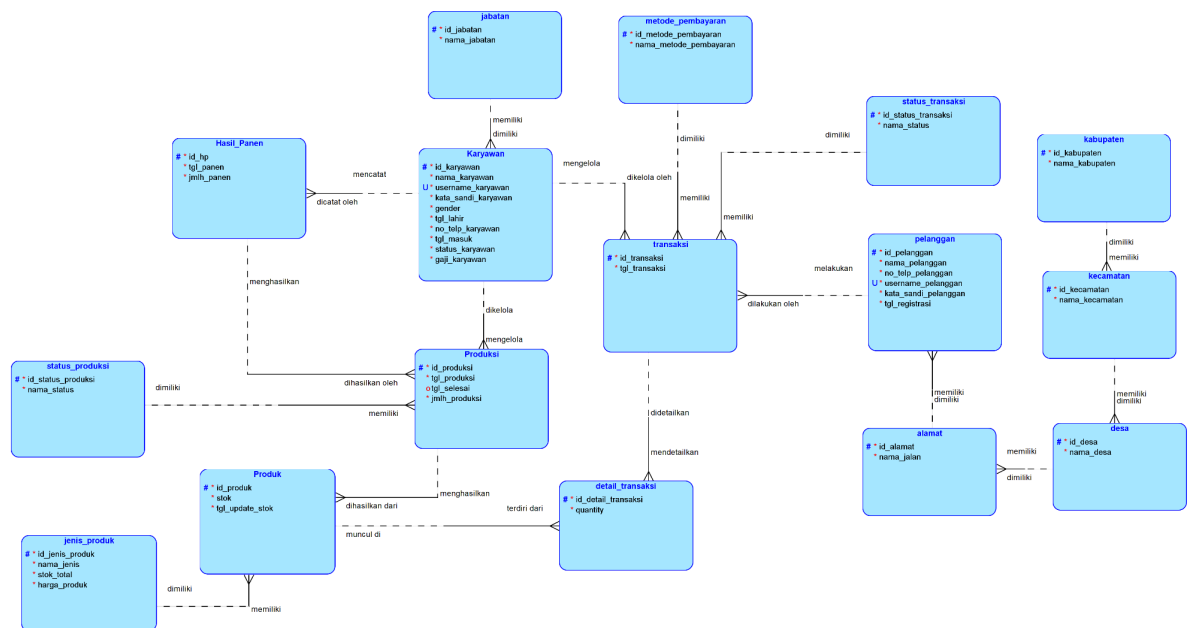
- Pemecahan atribut metode dan status transaksi dari entitas transaksi menjadi entitas metode transaksi dan status transaksi untuk menghindari nilai berulang dan konsistensi data.



- Pemecahan atribut alamat dari entity pelanggan, dikarenakan atribut alamat termasuk dalam atribut yang composite.



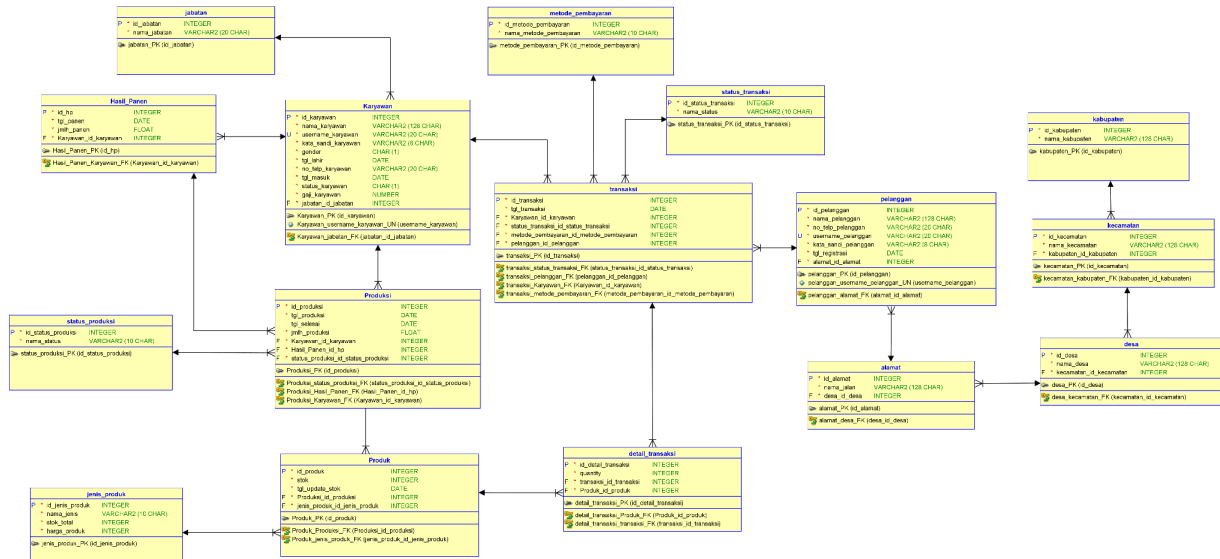
2. 2NF (memenuhi 1NF dan setiap entitas harus memiliki primary key dan semua atribut yang ada harus bergantung dalam satu primary key dalam sebuah tabel) memenuhi 1NF dan setiap entitas harus memiliki primary key, serta seluruh atribut non-key harus bergantung sepenuhnya pada satu primary key yang dimiliki tabel tersebut. Dengan kata lain, tidak boleh ada partial dependency pada sebagian primary key. Dalam rancangan ERD ini, seluruh entitas telah memenuhi 2NF karena setiap entitas sudah memiliki primary key tunggal, dan semua atribut yang terdapat di dalam masing-masing entitas sepenuhnya bergantung pada satu primary key tersebut
3. 3NF (memenuhi 1NF dan setiap atribut hanya bergantung kepada primary key dan tidak bergantung pada kolom non primary key). Dalam rancangan database Agromame sudah memenuhi 3NF karena setiap atribut dalam entitas tidak ada yang bergantung kepada kolom non primary key dan hanya bergantung kepada primary key masing masing entitas.



Link gambar:

## BAB 4

## 4.1 PDM



Link gambar: [https://drive.google.com/file/d/1r9a2nPMYhI4T3zuPfvzQ6CENml-08SUc/view?usp=drive\\_link](https://drive.google.com/file/d/1r9a2nPMYhI4T3zuPfvzQ6CENml-08SUc/view?usp=drive_link)

## 4.2 Deskripsi Table

PRODUKSI				
Nama Kolom	Type	Null	Key	Default
id_produk	Integer	No	Primary Key	No
tgl_produk	Date	No	-	No
tgl_selesai	Date	-	-	No
jmlh_produk	Float	No	-	No
karyawan_id_karyawan	Integer	No	Foreign Key	No
Hasil_panen_id_hp	Integer	No	Foreign Key	No
Status_produk id status produk	Integer	No	Foreign Key	No
HASIL PANEN				
Nama Kolom	Type	Null	Key	Default

HASIL PANEN				
id_hasil_panen	Integer	No	Primary Key	No
tgl_panen	Date	No	-	No
jumlah_panen	Float	No	-	No
Karyawan_id_karyawan	Integer	No	Foreign Key	No

JENIS PRODUK				
Nama Kolom	Type	Null	Key	Default
nama_jenis	Varchar(10)	No	-	No
id_jenis_produk	Integer	No	Primary Key	No
stok_total	Integer	No	-	No
harga_produk	Integer	No	-	No

PRODUK				
Nama Kolom	Type	Null	Key	Default
id_produk	Integer	No	Primary Key	No
stok	Integer	No	-	No
tgl_update_stok	Date	No	-	No
Produksi_id_produk	Integer	No	Foreign Key	No
Jenis_produk_id_jenis_produk	Integer	No	Foreign Key	No

PELANGGAN				
Nama Kolom	Type	Null	Key	Default
username_pelanggan	Varchar(20)	No	Secondary Key	No
kata_sandi_pelanggan	Varchar(8)	No	-	No
nama_pelanggan	Varchar(128)	No	-	No
id_pelanggan	Integer	No	Primary Key	No
no_telp_pelanggan	Varchar(20)	No	-	No
alamat_id_alamat	Integer	No	Foreign Key	No

STATUS PRODUKSI				
Nama Kolom	Type	Null	Key	Default
id_status_produk	Integer	No	Primary Key	No
nama_status	Varchar(10)	No	-	No

KECAMATAN				
Nama Kolom	Type	Null	Key	Default
id_kecamatan	Integer	No	Primary Key	No
nama_kecamatan	Varchar(128)	No	-	No
kabupaten_id_kabupaten	Integer	No	Foreign Key	No

JABATAN				
Nama Kolom	Type	Null	Key	Default
id_jabatan	Integer	No	Primary Key	No
nama_jabatan	Varchar(20)	No	-	No

KARYAWAN				
Nama Kolom	Type	Null	Key	Default
username_karyawan	Varchar(20)	No	Secondary Key	No
kata_sandi_karyawan	Varchar(8)	No	-	No
id_karyawan	Integer	No	Primary Key	No
nama_karyawan	Varchar(128)	No	-	No
gender	Char(1)	No	-	No
tgl_lahir	Date	No	-	No
no_telp	Varchar(20)	No	Secondary Key	No
tgl_masuk	Date	No	-	No
status_karyawan	Char(1)	No	-	No
jabatan_id_jabatan	Integer	No	Foreign Key	No
gaji_karyawan	Integer	No	-	No



METODE PEMBAYARAN				
Nama Kolom	Type	Null	Key	Default
id_metode_pembayaran	Integer	No	Primary Key	No
nama_metode_pembayaran	Varchar(10)	No	-	No

TRANSAKSI				
Nama Kolom	Type	Null	Key	Default
id_transaksi	Integer	No	Primary Key	No
tgl_transaksi	Date	No	-	No
pelanggan_id_pelanggan	Integer	No	Foreign Key	No
karyawan_id_karyawan	Integer	No	Foreign Key	No
status_transaksi_id_status_transaksi	Integer	No	Foreign Key	No
metode_pembayaran_id_metode_pembayaran	Integer	No	Foreign Key	No

ALAMAT				
Nama Kolom	Type	Null	Key	Default
id_alamat	Integer	No	Primary Key	No
nama_jalan	Varchar(128)	No	-	No
desa_id_desa	Integer	No	Foreign Key	No

DETAIL TRANSAKSI				
Nama Kolom	Type	Null	Key	Default
id_detail_transaksi	Integer	No	Primary Key	No
quantity	Integer	No	-	No
transaksi_id_transaksi	Integer	No	Foreign Key	No
produk_id_produk	Integer	No	Foreign Key	No

DESA				
Nama Kolom	Type	Null	Key	Default
id_desa	Integer	No	Primary Key	No
nama_desa	Varchar(128)	No	-	No
kecamatan_id_kecamatan	Integer	No	Foreign Key	No

STATUS TRANSAKSI				
Nama Kolom	Type	Null	Key	Default
id_status_transa	Integer	No	Primary Key	No

ksi				
nama_status	Varchar(10)	No	-	No

#### 4.3 DDL sesuai deskripsi tabel

PRODUKSI
<p><b>– TABEL PRODUKSI</b></p> <pre>CREATE TABLE Produksi (   id_produk SERIAL PRIMARY KEY ,   tgl_produk DATE NOT NULL ,   tgl_selesai DATE NULL,   jmlh_produk FLOAT NOT NULL,   id_karyawan INTEGER references karyawan(id_karyawan),   id_hp INTEGER references hasil_panen (id_hp),   id_status_produk INTEGER references status_produk (id_status_produk) )</pre>

STATUS PRODUKSI
<p><b>– TABEL STATUS PRODUKSI</b></p> <pre>CREATE TABLE status_produk (   id_status_produk SERIAL PRIMARY KEY ,   nama_status VARCHAR (10) NOT NULL )</pre>

PRODUK
<p><b>– TABEL PRODUK</b></p> <pre>CREATE TABLE Produk (   id_produk SERIAL PRIMARY KEY ,   stok INTEGER NOT NULL ,   id_produk INTEGER references produksi (id_produk),   id_jenis_produk INTEGER references jenis_produk (id_jenis_produk),   tgl_update_stok DATE NOT NULL )</pre>

JENIS PRODUK
--------------

**– TABEL JENIS PRODUK**

```
CREATE TABLE jenis_produk
(
  id_jenis_produk SERIAL PRIMARY KEY ,
  nama_jenis VARCHAR (10) NOT NULL,
  stok_total INTEGER NOT NULL,
  harga INTEGER NOT NULL
)
```

**HASIL PANEN**

**– TABEL HASIL PANEN**

```
CREATE TABLE Hasil_Panen
(
  id_hp SERIAL PRIMARY KEY ,
  tgl_panen DATE NOT NULL ,
  "jmlh_panen" FLOAT NOT NULL,
  id_karyawan integer references karyawan (id_karyawan)
)
```

**KARYAWAN**

**– TABEL KARYAWAN**

```
CREATE TABLE Karyawan
(
  id_karyawan SERIAL PRIMARY KEY ,
  nama_karyawan VARCHAR (128) NOT NULL ,
  gender BOOLEAN NOT NULL ,
  tgl_lahir DATE NOT NULL ,
  no_telp VARCHAR (20) NOT NULL ,
  tgl_masuk DATE NOT NULL ,
  status_karyawan BOOLEAN NOT NULL ,
  gaji_karyawan Integer NULL,
  id_jabatan INTEGER REFERENCES jabatan (id_jabatan),
  username VARCHAR(20) UNIQUE,
  kata_sandi VARCHAR(8) NOT NULL
)
```

**JABATAN**

**– TABEL JABATAN**

```
CREATE TABLE jabatan
(
  id_jabatan SERIAL PRIMARY KEY ,
  nama_jabatan VARCHAR (20) NOT NULL
)
```

**TRANSAKSI**

**– TABEL TRANSAKSI**

```
CREATE TABLE transaksi
(
  id_transaksi SERIAL PRIMARY KEY ,
  tgl_transaksi DATE NOT NULL,
  id_pelanggan INTEGER references pelanggan (id_pelanggan),
  id_karyawan INTEGER references karyawan (id_karyawan),
  id_status_transaksi INTEGER references status_transaksi (id_status_transaksi),
  id_metode_pembayaran INTEGER references metode_pembayaran
(id_metode_pembayaran)
)
```

**DETAIL TRANSAKSI**

**– TABEL DETAIL TRANSAKSI**

```
CREATE TABLE detail_transaksi
(
  id_detail_transaksi SERIAL PRIMARY KEY,
  quantity INTEGER NOT NULL ,
  id_transaksi INTEGER references transaksi (id_transaksi),
  id_produk INTEGER references produk (id_produk)
)
```

**METODE PEMBAYARAN**

**– TABEL METODE PEMBAYARAN**

```
CREATE TABLE metode_pembayaran
(
  id_metode_pembayaran SERIAL PRIMARY KEY ,
  nama_metode VARCHAR (10) NOT NULL
)
```

## STATUS TRANSAKSI

### – TABEL STATUS TRANSAKSI

```
CREATE TABLE status_transaksi
(
  id_status_transaksi SERIAL PRIMARY KEY,
  nama_status VARCHAR (10) NOT NULL
)
```

## PELANGGAN

### – TABEL PELANGGAN

```
CREATE TABLE pelanggan
(
  id_pelanggan SERIAL PRIMARY KEY ,
  nama_pelanggan VARCHAR (128) NOT NULL ,
  no_telp_pelanggan VARCHAR (20) NOT NULL ,
  tgl_registrasi DATE NOT NULL,
  id_alamat INTEGER references alamat (id_alamat),
  username VARCHAR(20) UNIQUE,
  kata_sandi VARCHAR(8) NOT NULL
)
```

## DESA

### – TABEL DESA

```
CREATE TABLE desa
(
  id_desa SERIAL PRIMARY KEY,
  nama_desa VARCHAR (128) NOT NULL,
  id_kecamatan SERIAL REFERENCES kecamatan (id_kecamatan)
)
```

## ALAMAT

### – TABEL ALAMAT

```
CREATE TABLE alamat
(
  id_alamat SERIAL PRIMARY KEY,
  nama_jalan VARCHAR (128) NOT NULL,
  id_desa SERIAL REFERENCES desa (id_desa)
)
```

## KECAMATAN

### – TABEL KECAMATAN

```
CREATE TABLE kecamatan
(
  id_kecamatan SERIAL PRIMARY KEY ,
  nama_kecamatan VARCHAR (128) NOT NULL,
  id_kabupaten INTEGER references kabupaten (id_kabupaten)
)
```

## KABUPATEN

### – TABEL KABUPATEN

```
CREATE TABLE kabupaten
(
  id_kabupaten SERIAL PRIMARY KEY,
  nama_kabupaten VARCHAR (128) NOT NULL
)
```



## BAB 5

### BUILD KE DATABASE

#### PRODUKSI

##### INPUT :

INSERT INTO produksi (tgl\_produk, tgl\_selesai, jmlh\_produk, id\_karyawan, id\_hp, id\_status\_produk)

VALUES

('2025-11-20', '2025-11-27', 170, 1, 1, 1),  
( '2025-11-21', '2025-11-28', 185, 2, 2, 2),  
( '2025-11-22', '2025-11-29', 0, 2, 3, 3);

##### OUTPUT :

	id_produk [PK] integer	tgl_produk date	tgl_selesai date	jmlh_produk double precision	id_karyawan integer	id_hp integer	id_status_produk integer
1	1	2025-11-20	2025-11-27	170	1	1	1
2	2	2025-11-21	2025-11-28	185	2	2	2
3	3	2025-11-22	2025-11-29	0	2	3	3

#### STATUS PRODUKSI

##### INPUT :

INSERT INTO status\_produk (nama\_status)

VALUES

('selesai'),  
( 'ditunda'),  
( 'proses');

##### OUTPUT :

	id_status_produk [PK] integer	nama_status character varying (10)
1	1	selesai
2	2	ditunda
3	3	proses

#### PRODUK

##### INPUT :

INSERT INTO produk (stok, id\_produk, id\_jenis\_produk, tgl\_update\_stok)

VALUES

(300, 2, 1, '2025-11-23'),  
(200, 3, 2, '2025-11-24'),  
(180, 4, 3, '2025-11-25');

**OUTPUT :**

	id_produk [PK] integer	stok integer	id_produksi integer	id_jenis_produk integer	tgl_update_stok date
1	4	300	2	1	2025-11-23
2	5	200	3	2	2025-11-24
3	6	180	4	3	2025-11-25

## JENIS PRODUK

**INPUT :**

INSERT INTO jenis\_produk (nama\_jenis, stok\_total, harga\_produk)

VALUES

('fresh', 100, 20000),  
( 'frozen', 100, 22500),  
( 'jus', 80, 8000);

**OUTPUT :**

	id_jenis_produk [PK] integer	nama_jenis character varying (10)	stok_total integer	harga_produk integer
1	1	fresh	100	20000
2	2	frozen	100	22500
3	3	jus	80	8000

## HASIL PANEN

**INPUT :**

INSERT INTO hasil\_panen (tgl\_panen, jmlh\_panen, id\_karyawan)

VALUES

('2025-11-20', 200, 1),  
( '2025-11-21', 190.5, 2),  
( '2025-11-22', 31.5, 2);

**OUTPUT :**

	id_hp [PK] integer	tgl_panen date	jmlh_panen double precision	id_karyawan integer
1	1	2025-11-20	200	1
2	2	2025-11-21	190.5	2
3	3	2025-11-22	31.5	2

## KARYAWAN

### INPUT :

INSERT INTO karyawan (nama\_karyawan, gender, tgl\_lahir, no\_telp, tgl\_masuk, status\_karyawan, gaji\_karyawan, id\_jabatan, username\_karyawan, kata\_sandi\_karyawan)  
VALUES

('Rosa', 'true', '2006-02-12', 082330302030, '2022-03-10', 'true', '4000000', 2, 'rosy', '321'),  
('Rosyid', 'false', '2006-02-11', 082345678901, '2021-03-10', 'true', '4000000', 2, 'ros', '345'),  
('Rizka', 'true', '2006-02-10', 089796954345, '2023-02-11', 'true', '3500000', 3, 'astoria', '4651'),  
('Axel', 'false', '2007-01-10', 083456789012, '2022-02-11', 'true', '3500000', 3, 'ax', '4567'),  
('Laili', 'true', '2006-05-13', 0895329151111, '2020-01-05', 'true', '5000000', 1, 'layy', '0506');

### OUTPUT :

	id_karyawan [PK] integer	nama_karyawan character varying (20)	username_karyawan character varying (20)	kata_sandi_karyawan character varying (8)	gender boolean	tgl_lahir date	no_telp character varying (20)	tgl_masuk date	status_karyawan boolean	gaji_karyawan integer	id_jabatan integer
1	1	Rosa	rosy	321	true	2006-02-12	82330302030	2022-03-10	true	4000000	2
2	2	Rosyid	ros	345	false	2006-02-11	82345678901	2021-03-10	true	4000000	2
3	3	Rizka	astoria	4651	true	2006-02-10	89796954345	2023-02-11	true	3500000	3
4	4	Axel	ax	4567	false	2007-01-10	83456789012	2022-02-11	true	3500000	3
5	5	Laili	layy	0506	true	2006-05-13	895329151111	2020-01-05	true	5000000	1

## JABATAN

### INPUT :

INSERT INTO jabatan (nama\_jabatan)  
VALUES

('Manajer Produksi'),  
('Admin Produksi'),  
('Admin Gudang');

### OUTPUT :

	id_jabatan [PK] integer	nama_jabatan character varying (20)
1	1	Manajer Produksi
2	2	Admin Produksi
3	3	Admin Gudang

## TRANSAKSI

### INPUT :

INSERT INTO transaksi (tgl\_transaksi, id\_pelanggan, id\_karyawan, id\_status\_transaksi, id\_metode\_pembayaran)

VALUES

('2025-12-20', 1, 3, 1, 1),

('2025-12-21', 2, 4, 2, 2),

('2025-12-23', 3, 3, 3, 3);

### OUTPUT :

	id_transaksi [PK] integer	tgl_transaksi date	id_pelanggan integer	id_karyawan integer	id_status_transaksi integer	id_metode_pembayaran integer
1	1	2025-12-20	1	3	1	1
2	2	2025-12-21	2	4	2	2
3	3	2025-12-23	3	3	3	3

## DETAIL TRANSAKSI

### INPUT :

INSERT INTO detail\_transaksi (quantity, id\_transaksi, id\_produk)

VALUES

(2, 1, 4),

(3, 2, 5),

(4, 3, 6);

### OUTPUT :

	id_detail_transaksi [PK] integer	quantity integer	id_transaksi integer	id_produk integer
1	4	2	1	4
2	5	3	2	5
3	6	4	3	6

## METODE PEMBAYARAN

### INPUT :

```
INSERT INTO metode_pembayaran (nama_metode)
VALUES
('tunai'),
('qris'),
('transfer');
```

### OUTPUT :

	id_metode_pembayaran [PK] integer	nama_metode character varying (10)
1	1	tunai
2	2	qris
3	3	transfer

## STATUS TRANSAKSI

### INPUT :

```
INSERT INTO status_transaksi (nama_status)
VALUES
('selesai'),
('proses'),
('dibatalkan');
```

### OUTPUT :

	id_status_transaksi [PK] integer	nama_status character varying (10)
1	1	selesai
2	2	proses
3	3	dibatalkan

## PELANGGAN

### INPUT :

```
INSERT INTO pelanggan (nama_pelanggan, no_telp_pelanggan, tgl_registrasi,
username_pelanggan, kata_sandi_pelanggan, id_alamat)
values
('Aurora Leona', 089876543210, '2025-12-01', 'raa', '321', 1),
('Leonardo Gabriel', 089765432109, '2025-12-02', 'leeee', '123', 2),
('Teona Lolly', 089654321098, '2025-12-02', 'starr', '456', 3),
('Muhammad Benz', 089543210987, '2025-12-07', 'muh', '789', 4);
```

### OUTPUT :

	id_pelanggan [PK] integer	nama_pelanggan character varying (128)	no_telp_pelanggan character varying (20)	username_pelanggan character varying (20)	kata_sandi_pelanggan character varying (8)	tgl_registrasi date	id_alamat integer
1	1	Aurora Leona	89876543210	raa	321	2025-12-01	9
2	2	Leonardo Gabriel	89765432109	leeee	123	2025-12-02	10
3	3	Teona Lolly	89654321098	starr	456	2025-12-02	11
4	4	Muhammad Benz	89543210987	muh	789	2025-12-07	12

## ALAMAT

### INPUT :

INSERT INTO alamat (nama\_jalan, id\_desa)

VALUES

('sumatra', 1),

('jawa', 2),

('brantas', 3),

('riau', 3);

### OUTPUT :

	id_alamat [PK] integer	nama_jalan character varying (128)	id_desa integer
1	4	sumatra	1
2	5	jawa	2
3	6	brantas	3

## DESA

### INPUT :

INSERT INTO desa (nama\_desa, id\_kecamatan)

VALUES

('kebonsari', 3),

('karangrejo', 3),

('tegalgede', 3);

### OUTPUT :

	id_desa [PK] integer	nama_desa character varying (128)	id_kecamatan integer
1	1	kebonsari	3
2	2	karangrejo	3
3	3	tegalgede	3

## KECAMATAN

### INPUT :

INSERT INTO kecamatan (nama\_kecamatan, id\_kabupaten)

### VALUES

('kaliwates', 1),  
('patrang', 1),  
('sumbersari', 1);

### OUTPUT :

	id_kecamatan [PK] integer	nama_kecamatan character varying (128)	id_kabupaten integer
1	1	kaliwates	1
2	2	patrang	1
3	3	sumbersari	1

## KABUPATEN

### INPUT :

INSERT INTO kabupaten (nama\_kabupaten)

### VALUES

('jember'),  
('lumajang'),  
('bondowoso');

### OUTPUT :

	id_kabupaten [PK] integer	nama_kabupaten character varying (128)
1	1	jember
2	2	lumajang
3	3	bondowoso

## BAB 6

### IMPLEMENTASI DENGAN BAHASA PEMROGRAMAN

```
import psycopg2
import pandas as pd
from tabulate import tabulate
from datetime import datetime
import logo

def connectDB():
    try:
        conn = psycopg2.connect(host="localhost", user="postgres",
password="13mei2006", dbname="basda")
        cur = conn.cursor()
        print("Database connected successfully")
        return conn, cur
    except Exception as e:
        print("Failed to connect to database")
        print("Error detail:")
        return None, None

def main():
    while True:
        Menu = [
            ["No", "Opsi"],
            [1., "Register"],
            [2., "Login"],
            [3., "Keluar"]
        ]
        print("\n=== Menu Utama ===")
        print(tabulate(Menu, headers="firstrow", tablefmt="fancy_grid"))
        pilihan = int(input("Pilih opsi (1/2/3): "))
        if pilihan == 1:
            registrasi()
        elif pilihan == 2:
            login()
        elif pilihan == 3:
            print("Program selesai. Terima kasih!")
            break
        else:
            print("Pilihan tidak valid, coba lagi.")
```



```

def registrasi():
    print("==== Registrasi Pelanggan ====")
    nama_pelanggan= input("Masukkan nama pelanggan: ")
    username_pelanggan = input("Masukkan username: ")
    kata_sandi_pelanggan = input("Masukkan kata_sandi: ")
    no_telp_pelanggan = input("Masukkan nomor telepon: ")

    conn, cur = connectDB()
    query = """
        INSERT INTO pelanggan (nama_pelanggan, username_pelanggan,
kata_sandi_pelanggan, no_telp_pelanggan, tgl_registrasi)
        VALUES (%s, %s, %s, %s, %s)
        """
    cur.execute(query, (nama_pelanggan, username_pelanggan,
kata_sandi_pelanggan, no_telp_pelanggan, datetime.now()))
    conn.commit()
    print("Registrasi berhasil!")
    main()
    cur.close()
    conn.close()

def login():
    while True:
        Login = [
            ["No.", "Opsi"],
            [1., "Karyawan"],
            [2., "Pelanggan"]
        ]
        print("\n==== Login ====")
        print(tabulate(Login, headers="firstrow", tablefmt="fancy_grid"
))

        pilihan = int(input("Pilih opsi (1/2): "))
        if pilihan == 1:
            login_karyawan()
        elif pilihan == 2:
            login_pelanggan()
            break
        else:
            print("Pilihan tidak valid, coba lagi.")

def login_karyawan():
    print("==== Login Karyawan ====")
    username = input("Username: ")

```

```

password = input("Password: ")

karyawan = cari_karyawan_di_database(username, password)

if karyawan :
    nama_lengkap = karyawan['nama_karyawan']
    role = karyawan['role']

log_karyawan = [
    ["Nama Lengkap", nama_lengkap],
    ["Username", username],
    ["Password", "*" * len(password)],
    ["Role", role]
]
print("\n=== Data Login Karyawan ===")
print(tabulate(log_karyawan, headers=["Data", "Keterangan"],
tablefmt="fancy_grid"))
print(f"\nSelamat datang {nama_lengkap}! Anda login sebagai
{role}.\n")

if karyawan:
    role = karyawan['role']
    if role == 'Manajer Produksi':
        halaman_menu_manajer_produksi()
    elif role == 'Admin Produksi':
        halaman_menu_admin_produksi()
    elif role == 'Admin Gudang':
        halaman_menu_admin_gudang()
    else:
        print("Role tidak dikenali.")
else:
    print("Login gagal. Username atau password salah.")

def cari_karyawan_di_database(username, password):
    conn, cur = connectDB()
    query = """SELECT k.nama_karyawan, k.username_karyawan,
k.kata_sandi_karyawan, j.nama_jabatan
FROM karyawan k
JOIN jabatan j ON k.id_jabatan = j.id_jabatan
WHERE k.username_karyawan = %s AND k.kata_sandi_karyawan
= %s"""
    cur.execute(query, (username, password))
    result = cur.fetchone()

```

```

    if result:
        karyawan = {
            'nama_karyawan': result[0],
            'username': result[1],
            'password': result[2],
            'role': result[3]
        }
        cur.close()
        conn.close()
        return karyawan
    cur.close()
    conn.close()
    return None

def login_pelanggan():
    print("==== Login Pelanggan ====")
    username_pelanggan = input("Masukkan username: ")
    kata_sandi_pelanggan = input("Masukkan kata_sandi: ")

    conn, cur = connectDB()
    query = """SELECT id_pelanggan, nama_pelanggan, username_pelanggan,
kata_sandi_pelanggan
                FROM pelanggan
                WHERE username_pelanggan = %s AND kata_sandi_pelanggan =
%s"""
    cur.execute(query, (username_pelanggan, kata_sandi_pelanggan))
    result = cur.fetchone()

    if result:
        nama_pelanggan = result[1]
        username = result[2]
        password = result[3]

        log_pelanggan = [
            ["Nama Lengkap", nama_pelanggan],
            ["Username", username],
            ["Password", "*" * len(password)]
        ]

        print("\n==== Data Login Pelanggan ====")
        print(tabulate(log_pelanggan, headers=["Data", "Keterangan"],
tablefmt="fancy_grid"))
        print(f"\nSelamat datang {nama_pelanggan}! Anda berhasil login
sebagai Pelanggan.\n")

```

```

        halaman_menu_pelanggan(id_pelanggan = result[0])

    else:
        print("Login gagal! Username atau kata sandi salah.")
        cur.close()
        conn.close()
def halaman_menu_admin_produksi():
    while True:
        menu = [
            [1, "Lihat Biodata"],
            [2, "Edit Biodata"],
            [3, "Input Hasil Panen"],
            [4, "Hapus Hasil Panen"],
            [5, "Lihat Laporan Hasil Panen"],
            [6, "Edit Hasil Panen"],
            [7, "Input Produksi"],
            [8, "Edit Produksi"],
            [9, "Hapus Produksi"],
            [10, "Perbarui Status Produksi"],
            [11, "Lihat Laporan Produksi"],
            [12, "Logout"]
        ]
        print("\n=== Halaman Menu Admin Produksi ===")
        print(tabulate(menu, headers=["No", "Menu"],
tablefmt="fancy_grid"))
        pilihan_adp = int(input("Pilih menu: "))

        if pilihan_adp == 1:
            lihat_biodata_adp()
        elif pilihan_adp == 2:
            edit_bio_adp()
        elif pilihan_adp == 3:
            input_hp()
        elif pilihan_adp == 4:
            hapus_hp()
        elif pilihan_adp == 5:
            lihat_laporan_hp()
        elif pilihan_adp == 6:
            edit_hp()
        elif pilihan_adp == 7:
            input_produksi()
        elif pilihan_adp == 8:
            edit_produksi()

```

```

        elif pilihan_adp == 9:
            hapus_produksi()
        elif pilihan_adp == 10:
            perbarui_status_produksi()
        elif pilihan_adp == 11:
            lihat_laporan_produksi()
        elif pilihan_adp == 12:
            logout()
            exit()
        else:
            print("Pilihan tidak valid, coba lagi.")

def edit_bio_adp():
    print("=== Edit Biodata Admin Produksi ===")
    conn, cur = connectDB()
    username_karyawan = input("Masukkan username: ")
    query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
    cur.execute(query, (username_karyawan,))
    result = cur.fetchone()

    if result:
        print("Masukkan data baru (isi kembali data untuk tidak mengubah):")
        nama_baru = input(f>Nama ({result[1]}): ") or result[1]
        username_baru = input(f"Username ({result[2]}): ") or result[2]
        kata_sandi_baru = input(f">Kata Sandi ({result[3]}): ") or result[3]
        no_telp_baru = input(f">No Telepon ({result[6]}): ") or result[6]
    else:
        print("Data biodata tidak ditemukan.")
    cur.close()
    conn.close()

def lihat_biodata_adp():
    print("=== Lihat Biodata Admin Produksi ===")
    conn, cur = connectDB()
    username_karyawan = input("Masukkan username: ")
    query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
    cur.execute(query, (username_karyawan,))
    result = cur.fetchone()

    if result:

```

```

        biodata = [
            ["Nama", result[1]],
            ["Username", result[2]],
            ["Kata Sandi", result[3]],
            ["Gender", "Perempuan" if result[4] else "Laki-laki"],
            ["Tgl Lahir", result[5]],
            ["No Telepon", result[6]],
            ["Tgl Bergabung", result[7]],
            ["Gaji", result[9]]
        ]
        print("\n=== Biodata Admin Produksi ===")
        print(tabulate(biodata, headers=["Data", "Keterangan"],
tablefmt="fancy_grid"))
    else:
        print("Data biodata tidak ditemukan.")
    cur.close()
    conn.close()

def input_hp():
    print("=== Input Hasil Panen ===")
    conn, cur = connectDB()
    hasil_panen = float(input("Masukkan jumlah hasil panen (kg): "))
    tanggal_panen = input("Masukkan tanggal panen (YYYY-MM-DD): ")
    id_karyawan = int(input("Masukkan ID karyawan: "))
    query = """
        INSERT INTO hasil_panen (jmlh_panen, tgl_panen, id_karyawan)
        VALUES (%s, %s, %s)

    """
    cur.execute(query, (hasil_panen, tanggal_panen, id_karyawan))
    conn.commit()
    print("Input hasil panen berhasil!")
    cur.close()
    conn.close()

def hapus_hp():
    print("=== Hapus Hasil Panen ===")
    conn, cur = connectDB()
    id_hp = int(input("Masukkan ID hasil panen yang akan dihapus: "))
    query = "DELETE FROM hasil_panen WHERE id_hp = %s"
    cur.execute(query, (id_hp,))
    conn.commit()
    print("Hasil Panen berhasil dihapus.")

```

```

cur.close()
conn.close()

def edit_hp():
    print("=== Edit Hasil Panen ===")
    conn, cur = connectDB()
    id_hp = int(input("Masukkan ID hasil panen yang akan diedit: "))
    query = "SELECT * FROM hasil_panen WHERE id_hp = %s"
    cur.execute(query, (id_hp,))
    result = cur.fetchone()

    if result:
        print("Masukkan data baru (isi ulang untuk data yang tidak diubah):")
        tgl_panen_baru = input(f"Tanggal Panen ({result[1]}): ") or result[1]
        jmlh_panen_baru = input(f"Jumlah Panen ({result[2]}): ") or result[2]
        id_karyawan_baru = input(f"ID Karyawan ({result[3]}): ") or result[3]

        update_query = """
            UPDATE hasil_panen
            SET tgl_panen = %s, jmlh_panen = %s, id_karyawan = %s
            WHERE id_hp = %s
        """
        cur.execute(update_query, (tgl_panen_baru, jmlh_panen_baru, id_karyawan_baru, id_hp))
        conn.commit()
        print("Hasil Panen berhasil diperbarui.")
    else:
        print("Data hasil panen tidak ditemukan.")

    cur.close()
    conn.close()

def lihat_laporan_hp():
    print("=== Lihat Laporan Hasil Panen ===")
    conn, cur = connectDB()
    query = """
        SELECT hp.id_hp, hp.jmlh_panen, hp.tgl_panen, k.nama_karyawan
        FROM hasil_panen hp
        JOIN karyawan k ON hp.id_karyawan = k.id_karyawan
    """

```

```

"""
cur.execute(query)
results = cur.fetchall()
df = pd.DataFrame(results, columns=['ID Hasil Panen', 'Hasil Panen
(kg)', 'Tanggal Panen', 'Nama Karyawan'])
print(tabulate(df, headers='keys', tablefmt='fancy_grid',
showindex=False))
cur.close()
conn.close()

def input_produksi():
    print("=== Input Produksi ===")
    conn, cur = connectDB()
    tanggal_produksi = input("Masukkan tanggal produksi (YYYY-MM-DD): ")
    jumlah_produksi = float(input("Masukkan jumlah produksi (kg): "))
    id_karyawan = int(input("Masukkan ID karyawan: "))
    id_hp = int(input("Masukkan ID hasil panen: "))
    id_status_produksi = int(input("Masukkan ID status panen (1.
selesai/2. ditunda/3. proses): "))
    query = """
        INSERT INTO produksi (tgl_produksi, jmlh_produksi, id_karyawan,
id_hp, id_status_produksi)
        VALUES (%s, %s, %s, %s, %s)
    """
    cur.execute(query, (tanggal_produksi, jumlah_produksi, id_karyawan,
id_hp, id_status_produksi))
    conn.commit()
    print("Input produksi berhasil!")
    cur.close()
    conn.close()

def perbarui_status_produksi():
    print("=== Perbarui Status Produksi ===")
    conn, cur = connectDB()
    id_produksi = int(input("Masukkan ID produksi yang akan diperbarui:
"))
    status_baru = int(input("Masukkan status produksi baru (1.
selesai/2. ditunda/3. proses): "))

    if status_baru == 1:
        query = """
            UPDATE produksi
            SET id_status_produksi = %s, tgl_selesai = %s

```



```

        WHERE id_produkSi = %s"""
        cur.execute(query, (status_baru, datetime.now(), id_produkSi))
    else:
        query = """
            UPDATE produksi
            SET id_status_produkSi = %s, tgl_selesai = NULL
            WHERE id_produkSi = %s"""
        cur.execute(query, (status_baru, id_produkSi))
    conn.commit()
    print("Status produksi berhasil diperbarui!")
    cur.close()
    conn.close()

def lihat_laporan_produkSi():
    print("=== Lihat Laporan Produksi ===")
    conn, cur = connectDB()
    query = """
        SELECT p.id_produkSi, p.tgl_produkSi, p.tgl_selesai,
        p.jmlh_produkSi,
            k.nama_karyawan, hp.jmlh_panen, sp.nama_status
        FROM produksi p
        JOIN karyawan k ON p.id_karyawan = k.id_karyawan
        JOIN hasil_panen hp ON p.id_hp = hp.id_hp
        JOIN status_produkSi sp ON p.id_status_produkSi =
        sp.id_status_produkSi"""
    cur.execute(query)
    results = cur.fetchall()
    df = pd.DataFrame(results, columns=['ID Produksi', 'Tanggal
    Produksi', 'Tanggal Selesai', 'Jumlah Produksi', 'Nama Karyawan', 'Jumlah
    Hasil Panen (kg)', 'Status Produksi' ])
    print(tabulate(df, headers='keys', tablefmt='fancy_grid',
    showindex=False))
    cur.close()
    conn.close()

def edit_produkSi():
    print("=== Edit Produksi ===")
    conn, cur = connectDB()
    id_produkSi = int(input("Masukkan ID produksi yang akan diedit: "))
    query = "SELECT * FROM produksi WHERE id_produkSi = %s"
    cur.execute(query, (id_produkSi,))
    result = cur.fetchone()

```

```

        if result:
            print("Masukkan data baru (isi kembali untuk tidak mengubah):")
            tgl_produksi_baru = input(f"Tanggal Produksi ({result[1]}): ")
            or result[1]
            jmlh_produksi_baru = input(f"Jumlah Produksi ({result[3]}): ")
            or result[3]
            id_karyawan_baru = input(f"ID Karyawan ({result[4]}): ") or
            result[4]
            id_hp_baru = input(f"ID Hasil Panen ({result[5]}): ") or
            result[5]
            id_status_produksi_baru = input(f"ID Status Produksi
            ({result[6]}): ") or result[6]

            if id_status_produksi_baru == 1:
                query = """
                    UPDATE produksi
                    SET id_status_produksi = %s, tgl_selesai = %s
                    WHERE id_produksi = %s"""
                cur.execute(query, (id_status_produksi_baru, datetime.now(),
            id_produksi))
            else:
                query = """
                    UPDATE produksi
                    SET id_status_produksi = %s, tgl_selesai = NULL
                    WHERE id_produksi = %s"""
                cur.execute(query, (id_status_produksi_baru, id_produksi))

                update_query = """
                    UPDATE produksi
                    SET tgl_produksi = %s, tgl_selesai = %s, jmlh_produksi =
            %s, id_karyawan = %s, id_hp = %s, id_status_produksi = %s
                    WHERE id_produksi = %s
                """
                cur.execute(update_query, (tgl_produksi_baru,
            datetime.now(), jmlh_produksi_baru, id_karyawan_baru, id_hp_baru,
            id_status_produksi_baru, id_produksi))
                conn.commit()
                print("Produksi berhasil diperbarui.")
            else:
                print("Data produksi tidak ditemukan.")

        cur.close()
        conn.close()

```

```

def hapus_produksi():
    print("=== Hapus Produksi ===")
    conn, cur = connectDB()
    id_produksi = int(input("Masukkan ID produksi yang akan dihapus: "))
    query = "DELETE FROM produksi WHERE id_produksi = %s"
    cur.execute(query, (id_produksi,))
    conn.commit()
    print("Produksi berhasil dihapus.")
    cur.close()
    conn.close()

def logout():
    print("Anda telah keluar dari sistem.")
    print("=== Logout ===")

def halaman_menu_admin_gudang():
    while True:
        print("=== Halaman Menu Admin Gudang ===")
        print("1. Lihat Biodata")
        print("2. Edit Biodata")
        print("3. Lihat Laporan Produksi")
        print("4. Input Produk")
        print("5. Hapus Produk")
        print("6. Lihat Laporan Produk")
        print("7. Edit Data Produk")
        print("8. Update Status Transaksi")
        print("9. Lihat Laporan Transaksi")
        print("10. Logout")
        pilihan_adg = int(input("Pilih menu: "))

        if pilihan_adg == 1:
            lihat_biodata_adg()
        elif pilihan_adg == 2:
            edit_bio_adg()
        elif pilihan_adg == 3:
            lihat_laporan_produksi()
        elif pilihan_adg == 4:
            input_produk()
        elif pilihan_adg == 5:
            hapus_produk()
        elif pilihan_adg == 6:
            lihat_laporan_produk()

```

```

        elif pilihan_adg == 7:
            edit_produk()
        elif pilihan_adg == 8:
            update_status_transaksi()
        elif pilihan_adg == 9:
            lihat_laporan_transaksi()
        elif pilihan_adg == 10:
            logout()
            exit()
        else:
            print("Pilihan tidak valid, coba lagi.")

def lihat_biodata_adg():
    print("=== Lihat Biodata Admin Gudang ===")
    conn, cur = connectDB()
    username_karyawan = input("Masukkan username: ")
    query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
    cur.execute(query, (username_karyawan,))
    result = cur.fetchone()

    if result:
        data = [
            ["Nama", result[1]],
            ["Username", result[2]],
            ["Kata Sandi", result[3]],
            ["Gender", "Perempuan" if result[4] else "Laki-laki"],
            ["Tgl Lahir", result[5]],
            ["No Telepon", result[6]],
            ["Tgl Bergabung", result[7]],
            ["Gaji", result[9]]
        ]
        print(tabulate(data, headers=["Data", "Keterangan"],
tablefmt="fancy_grid"))
    else:
        print("Data biodata tidak ditemukan.")
    cur.close()
    conn.close()

def edit_bio_adg():
    print("=== Edit Biodata Admin Gudang ===")
    conn, cur = connectDB()
    username_karyawan = input("Masukkan username: ")
    query = "SELECT * FROM karyawan WHERE username_karyawan = %s"

```

```

cur.execute(query, (username_karyawan,))
result = cur.fetchone()

if result:
    print("Masukkan data baru (isi kembali untuk tidak mengubah):")
    nama_baru = input(f>Nama ({result[1]}): ") or result[1]
    username_baru = input(f"Username ({result[2]}): ") or result[2]
    kata_sandi_baru = input(f">Kata Sandi ({result[3]}): ") or
result[3]
    no_telp_baru = input(f">No Telepon ({result[6]}): ") or result[6]

    update_query = """
        UPDATE karyawan
        SET nama_karyawan = %s, kata_sandi_karyawan = %s, no_telp =
%s, username_karyawan = %s
        WHERE username_karyawan = %s
    """
    cur.execute(update_query, (nama_baru, kata_sandi_baru,
no_telp_baru, username_baru, username_karyawan,))
    conn.commit()
    print("Biodata berhasil diperbarui.")
else:
    print("Data biodata tidak ditemukan.")

cur.close()
conn.close()

def input_produk():
    print("=== Input Produk ===")
    conn, cur = connectDB()
    jumlah_dibutuhkan = int(input("Masukkan jumlah stok: "))
    id_jenis_produk = int(input("Masukkan id jenis (1.Fresh, 2.Frozen,
3.Jus): "))

    query = """
        SELECT id_produk, jmlh_produk
        FROM produksi
        WHERE jmlh_produk > 0
        ORDER BY tgl_produk ASC
    """
    cur.execute(query)
    produksis = cur.fetchall()

```

```

    sisa = jumlah_dibutuhkan
    for id_produksi, jmlh_produksi in produksis:
        if sisa <= 0:
            break
        if jmlh_produksi >= sisa:
            cur.execute("""
                UPDATE produksi
                SET jmlh_produksi = jmlh_produksi - %s
                WHERE id_produksi = %s
            """, (sisa, id_produksi))
            sisa = 0
        else:
            cur.execute("""
                UPDATE produksi
                SET jmlh_produksi = 0
                WHERE id_produksi = %s
            """, (id_produksi,))
            sisa -= jmlh_produksi

    if sisa > 0:
        print("Stok habis! Tidak bisa memenuhi semua permintaan.")
    else:
        print("Input produk berhasil! Jumlah produksi berkurang.")
    conn.commit()
    cur.close()
    conn.close()

def lihat_laporan_produk():
    print("=== Lihat Laporan Produk ===")
    conn, cur = connectDB()
    query = """
        SELECT p.id_produk, j.nama_jenis, p.stok, p.tgl_update_stok,
ps.jmlh_produksi
        FROM produk p
        JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
        JOIN produksi ps ON p.id_produksi = ps.id_produksi
    """
    cur.execute(query)
    results = cur.fetchall()
    df = pd.DataFrame(results, columns=['ID Produk', 'Jenis Produk',
    'Jumlah Produk', 'Tanggal Update Stok', 'Jumlah Produksi'])
    print(tabulate(df, headers='keys', tablefmt='fancy_grid',
showindex=False))

```

```

cur.close()
conn.close()

def edit_produk():
    print("=== Edit Produk ===")
    conn, cur = connectDB()
    id_produk = int(input("Masukkan ID produk yang akan diedit: "))
    query = "SELECT * FROM produk WHERE id_produk = %s"
    cur.execute(query, (id_produk,))
    result = cur.fetchone()

    if result:
        print("Masukkan data baru (isi kembali jika tidak mengubah data):")
        stok_baru = int(input(f"Stok baru ({result[1]}): ") or result[1])

        update_query = """
            UPDATE produk
            SET stok = %s, tgl_update_stok = %s
            WHERE id_produk = %s
        """
        cur.execute(update_query, (stok_baru, datetime.now(), id_produk))
        conn.commit()
        print("Produk berhasil diperbarui.")
    else:
        print("Data produk tidak ditemukan.")

    cur.close()
    conn.close()

def hapus_produk():
    print("=== Hapus Produk ===")
    conn, cur = connectDB()
    id_produk = int(input("Masukkan ID produk yang akan dihapus: "))
    query = "DELETE FROM produk WHERE id_produk = %s"
    cur.execute(query, (id_produk,))
    conn.commit()
    print("Produk berhasil dihapus.")
    cur.close()
    conn.close()

```

```

def update_status_transaksi():
    print("=== Input Status Transaksi ===")
    conn, cur = connectDB()
    id_transaksi = int(input("Masukkan ID transaksi yang akan
diperbarui: "))
    status_baru = int(input("Masukkan status produksi baru (1. selesai/
2. proses /3. dibatalkan): "))

    query = """
        UPDATE transaksi
        SET id_status_transaksi = %s
        WHERE id_transaksi = %s
    """
    cur.execute(query, (status_baru, id_transaksi))
    conn.commit()
    print("Status transaksi berhasil diperbarui!")
    cur.close()
    conn.close()

def lihat_laporan_transaksi():
    print("=== Lihat Laporan Transaksi ===")
    conn, cur = connectDB()
    query = """
        SELECT dt.id_detail_transaksi, t.id_transaksi, t.tgl_transaksi,
jp.nama_jenis, dt.quantity, st.nama_status
        FROM transaksi t
        JOIN status_transaksi st ON t.id_status_transaksi =
st.id_status_transaksi
        JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
        JOIN produk p ON dt.id_produk = p.id_produk
        JOIN jenis_produk jp ON p.id_jenis_produk = jp.id_jenis_produk
    """
    cur.execute(query)
    results = cur.fetchall()
    df = pd.DataFrame(results, columns=['ID Detail', 'ID Transaksi',
'Tanggal', 'Produk', 'Jumlah Produk', 'Status Transaksi'])
    print(tabulate(df, headers='keys', tablefmt='fancy_grid',
showindex=False))
    cur.close()
    conn.close()

```



```
def halaman_menu_pelanggan(id_pelanggan):
    while True:
        print("=== AGROMAME: Menu Utama Pembelian ===")
        print("1. Lihat Biodata")
        print("2. Edit Biodata")
        print("3. Lihat Daftar Produk")
        print("4. Pilih Produk")
        print("5. Lihat Keranjang")
        print("6. Edit Keranjang")
        print("7. Pilih Metode Pembayaran")
        print("8. Edit Metode Pembayaran")
        print("9. Checkout")
        print("10. Lihat Riwayat Transaksi Pembelian")
        print("11. Logout")

        try:
            pilihan = int(input("Pilih menu: "))
        except ValueError:
            print("Input harus angka!")
            continue

        if pilihan == 1:
            lihat_biodata(id_pelanggan)
        elif pilihan == 2:
            edit_biodata(id_pelanggan)
        elif pilihan == 3:
            lihat_daftar_produk()
        elif pilihan == 4:
            pilih_produk(id_pelanggan)
        elif pilihan == 5:
            lihat_keranjang(id_pelanggan)
        elif pilihan == 6:
            edit_keranjang(id_pelanggan)
        elif pilihan == 7:
            pilih_metode_pembayaran(id_pelanggan)
        elif pilihan == 8:
            edit_metode_pembayaran(id_pelanggan)
        elif pilihan == 9:
            checkout(id_pelanggan)
        elif pilihan == 10:
            lihat_riwayat_transaksi_pembelian(id_pelanggan)
        elif pilihan == 11:
            print("Logout berhasil.")
```

```

        exit()
    else:
        print("Pilihan tidak valid!")

def lihat_biodata(id_pelanggan):
    conn, cur = connectDB()

    query = """
        SELECT p.nama_pelanggan, p.username_pelanggan,
p.kata_sandi_pelanggan, p.no_telp_pelanggan,
            a.nama_jalan, d.nama_desa, k.nama_kecamatan,
kb.nama_kabupaten
        FROM pelanggan p
        LEFT JOIN alamat a ON p.id_alamat = a.id_alamat
        LEFT JOIN desa d ON a.id_desa = d.id_desa
        LEFT JOIN kecamatan k ON d.id_kecamatan = k.id_kecamatan
        LEFT JOIN kabupaten kb ON k.id_kabupaten = kb.id_kabupaten
        WHERE p.id_pelanggan = %s
    """

    cur.execute(query, (id_pelanggan,))
    data = cur.fetchone()

    if data:
        print("=== BIODATA ===")
        if data[4]:
            headers = ["Nama", "Username", "Kata Sandi", "No Telp",
            "Jalan", "Desa", "Kecamatan", "Kabupaten"]
            print(tabulate([data], headers=headers,
tablefmt="fancy_grid"))
        else:
            headers= ["Nama", "Username", "Kata Sandi", "No Telp",
            "Alamat"]
            row = [data[0], data[1], data[2], data[3], "Belum diisi"]
            print(tabulate([row], headers=headers,
tablefmt="fancy_grid"))
        else:
            print("Data tidak ditemukan.")

    cur.close(); conn.close()

def edit_biodata(id_pelanggan):

```

```

print("=== Edit Biodata ===")
conn, cur = connectDB()

query = """
    SELECT p.nama_pelanggan, p.username_pelanggan,
p.kata_sandi_pelanggan,
        p.no_telp_pelanggan, a.id_alamat, a.nama_jalan,
d.id_desa, d.nama_desa
    FROM pelanggan p
    LEFT JOIN alamat a ON p.id_alamat = a.id_alamat
    LEFT JOIN desa d ON a.id_desa = d.id_desa
    WHERE p.id_pelanggan = %s
"""
cur.execute(query, (id_pelanggan,))
data = cur.fetchone()

if data:
    print("Isi data baru (tekan Enter jika tidak ingin mengubah):")
    nama = input(f>Nama ({data[0]}): ") or data[0]
    user = input(f>Username ({data[1]}): ") or data[1]
    sandi = input(f>Kata Sandi ({data[2]}): ") or data[2]
    telp = input(f>No Telp ({data[3]}): ") or data[3]

    if data[4]:
        jalan = input(f>Nama Jalan ({data[5]}): ") or data[5]
        desa = input(f>ID Desa ({data[6]} - {data[7]}): ") or
data[6]

        cur.execute("""
            UPDATE alamat SET nama_jalan = %s, id_desa = %s
            WHERE id_alamat = %s
            """, (jalan, desa, data[4]))
    else:
        print("Alamat belum ada, silakan isi:")
        jalan = input("Nama Jalan: ")
        desa = input("ID Desa: ")

        cur.execute("""
            INSERT INTO alamat (nama_jalan, id_desa)
            VALUES (%s, %s) RETURNING id_alamat
            """, (jalan, desa))
        id_alamat_baru = cur.fetchone()[0]

```

```

        cur.execute("""
            UPDATE pelanggan SET id_alamat = %s
            WHERE id_pelanggan = %s
            """, (id_alamat_baru, id_pelanggan))

    cur.execute("""
        UPDATE pelanggan
        SET nama_pelanggan = %s,
            username_pelanggan = %s,
            kata_sandi_pelanggan = %s,
            no_telp_pelanggan = %s
        WHERE id_pelanggan = %s
        """, (nama, user, sandi, telp, id_pelanggan))

    conn.commit()
    print("Biodata berhasil diperbarui.")
else:
    print("Data tidak ditemukan.")

cur.close(); conn.close()

def lihat_daftar_produk():
    print("=== Lihat Daftar Produk ===")
    conn, cur = connectDB()

    query = """
    SELECT j.nama_jenis, p.stok, j.harga_produk, p.tgl_update_stok
    FROM produk p
    JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
    ORDER BY j.nama_jenis
    """

    cur.execute(query)
    data = cur.fetchall()

    if data:
        headers = ["Nama Jenis", "Stok", "Harga Produk (Rp)", "Update
Stok"]
        print(tabulate(data, headers=headers, tablefmt="fancy_grid"))
    else:
        print("Belum ada produk tersedia.")

    cur.close(); conn.close()

```

```

def pilih_produk(id_pelanggan):
    print("=== Pilih Produk ===")
    conn, cur = connectDB()

    cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE
nama_status = 'proses'")
    s = cur.fetchone()
    if not s:
        print("Status 'proses' tidak ada di status_transaksi.")
        cur.close(); conn.close()
        return
    id_status = s[0]

    cur.execute("""
        SELECT t.id_transaksi
        FROM transaksi t
        WHERE t.id_pelanggan = %s AND t.id_status_transaksi = %s
        ORDER BY t.id_transaksi DESC
        LIMIT 1
    """, (id_pelanggan, id_status))
    t = cur.fetchone()

    if t:
        id_transaksi = t[0]
    else:
        cur.execute("""
            INSERT INTO transaksi (id_pelanggan, id_status_transaksi,
tgl_transaksi)
            VALUES (%s, %s, CURRENT_DATE) RETURNING id_transaksi
        """, (id_pelanggan, id_status))
        id_transaksi = cur.fetchone()[0]

    while True:
        cur.execute("""
            SELECT p.id_produk, j.nama_jenis, p.stok, j.harga_produk,
p.tgl_update_stok
            FROM produk p
            JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
            ORDER BY p.id_produk
        """)
        daftar_produk = cur.fetchall()

```

```

    if not daftar_produk:
        print("Belum ada produk tersedia.")
        break

    headers = ["ID Produk", "Jenis Produk", "Stok", "Harga (Rp)",
"Update Stok"]
    print(tabulate(daftar_produk, headers=headers,
tablefmt="fancy_grid"))

    try:
        id_produk = int(input("Masukkan ID Produk: "))
        quantity = int(input("Masukkan jumlah: "))
    except ValueError:
        print("Input harus angka!")
        continue

    cur.execute("SELECT stok FROM produk WHERE id_produk = %s",
(id_produk,))
    row = cur.fetchone()
    if not row:
        print("Produk tidak ditemukan.")
        continue

    stok_tersedia = row[0]
    if quantity > stok_tersedia:
        print("Stok tidak cukup. Stok tersedia:", stok_tersedia)
        continue

    cur.execute("""
        SELECT j.harga_produk
        FROM produk p
        JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
        WHERE p.id_produk = %s
    """, (id_produk,))
    row = cur.fetchone()
    if not row:
        print("Harga produk tidak ditemukan.")
        continue

    harga_satuan = row[0]
    total_harga = harga_satuan * quantity

    cur.execute("""

```

```

        INSERT INTO detail_transaksi (id_transaksi, id_produk,
quantity, harga)
        VALUES (%s, %s, %s, %s)
        """ , (id_transaksi, id_produk, quantity, total_harga))
    conn.commit()
    print(f"Produk {id_produk} x{quantity} berhasil ditambahkan ke
keranjang.")

    lanjut = input("Tambah produk lain? (y/n): ").strip().lower()
    if lanjut != "y":
        break

    cur.close(); conn.close()
    print("Selesai memilih produk. Semua pilihan sudah masuk ke
keranjang.")

def lihat_keranjang(id_pelanggan):
    print("=== Lihat Keranjang ===")
    conn, cur = connectDB()

    cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE
nama_status = 'proses'")
    s = cur.fetchone()
    if not s:
        print("Status 'proses' tidak ada di status_transaksi.")
        cur.close(); conn.close()
        return
    id_status = s[0]

    cur.execute("""
        SELECT id_transaksi, id_metode_pembayaran
        FROM transaksi
        WHERE id_pelanggan = %s AND id_status_transaksi = %s
        ORDER BY id_transaksi DESC
        LIMIT 1
        """, (id_pelanggan, id_status))
    t = cur.fetchone()

    if not t:
        print("Keranjang kosong. Belum ada transaksi aktif.")
        cur.close(); conn.close()
        return

```

```

id_transaksi, id_metode = t

cur.execute("""
    SELECT j.nama_jenis, d.quantity, d.harga
    FROM detail_transaksi d
    JOIN produk p ON d.id_produk = p.id_produk
    JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
    WHERE d.id_transaksi = %s
""", (id_transaksi,))
items = cur.fetchall()

if not items:
    print("Keranjang masih kosong.")
else:
    headers = ["Produk", "Jumlah", "Subtotal (Rp)"]
    print(tabulate(items, headers=headers, tablefmt="fancy_grid"))

    total = sum([row[2] for row in items])
    print("\nTotal Keranjang: Rp", total)

if id_metode:
    cur.execute("SELECT id_metode_pembayaran, nama_transaksi FROM
metode_pembayaran WHERE id_metode_pembayaran = %s", (id_metode,))
    metode = cur.fetchone()
    if metode:
        print(f"\nMetode Pembayaran Aktif: {metode[1]} (ID:
{metode[0]})")
    else:
        print("\nMetode pembayaran belum dipilih.")
else:
    print("\nMetode pembayaran belum dipilih.")

cur.close(); conn.close()

def edit_keranjang(id_pelanggan):
    print("=== Edit Keranjang ===")
    conn, cur = connectDB()

    cur.execute("""
        SELECT t.id_transaksi
        FROM transaksi t
    """)

```



```

        JOIN status_transaksi s ON t.id_status_transaksi =
s.id_status_transaksi
        WHERE t.id_pelanggan = %s AND s.nama_status = 'proses'
        ORDER BY t.id_transaksi DESC
        LIMIT 1
    """ , (id_pelanggan,))
    t = cur.fetchone()

    if not t:
        print("Keranjang kosong. Belum ada transaksi aktif.")
        cur.close(); conn.close()
        return

    id_transaksi = t[0]

    cur.execute("""
        SELECT d.id_detail_transaksi, j.nama_jenis, d.quantity, d.harga
        FROM detail_transaksi d
        JOIN produk p ON d.id_produk = p.id_produk
        JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
        WHERE d.id_transaksi = %s
    """ , (id_transaksi,))
    items = cur.fetchall()

    if not items:
        print("Keranjang masih kosong.")
        cur.close(); conn.close()
        return

    headers = ["ID Detail", "Produk", "Jumlah", "Subtotal (Rp)"]
    print(tabulate(items, headers=headers, tablefmt="fancy_grid"))

    try:
        id_detail = int(input("Masukkan ID Detail yang ingin
diubah/hapus: "))
        pilihan = input("Ketik 'ubah' untuk ubah jumlah, 'hapus' untuk
hapus produk: ").lower()
    except ValueError:
        print("Input harus angka!")
        cur.close(); conn.close()
        return

    if pilihan == "ubah":

```

```

try:
    new_qty = int(input("Masukkan jumlah baru: "))
except ValueError:
    print("Input harus angka!")
    cur.close(); conn.close()
    return

cur.execute("""
    SELECT j.harga_produk
    FROM detail_transaksi d
    JOIN produk p ON d.id_produk = p.id_produk
    JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
    WHERE d.id_detail_transaksi = %s
""", (id_detail,))
row = cur.fetchone()
if not row:
    print("Produk tidak ditemukan.")
    cur.close(); conn.close()
    return

harga_satuan = row[0]
total_harga = harga_satuan * new_qty

cur.execute("""
    UPDATE detail_transaksi
    SET quantity = %s, harga = %s
    WHERE id_detail_transaksi = %s
""", (new_qty, total_harga, id_detail))
conn.commit()
print("Jumlah produk berhasil diubah.")

elif pilihan == "hapus":
    cur.execute("DELETE FROM detail_transaksi WHERE
id_detail_transaksi = %s", (id_detail,))
    conn.commit()
    print("Produk berhasil dihapus dari keranjang.")

else:
    print("Pilihan tidak valid.")

cur.close(); conn.close()

def pilih_metode_pembayaran(id_pelanggan):

```

```

print("=== Pilih Metode Pembayaran ===")
conn, cur = connectDB()

cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE
nama_status = 'proses'")
s = cur.fetchone()
if not s:
    print("Status 'proses' tidak ditemukan.")
    cur.close(); conn.close()
    return
id_status = s[0]

cur.execute("""
    SELECT id_transaksi
    FROM transaksi
    WHERE id_pelanggan = %s AND id_status_transaksi = %s
    ORDER BY id_transaksi DESC
    LIMIT 1
""", (id_pelanggan, id_status))
t = cur.fetchone()

if not t:
    print("Tidak ada transaksi aktif. Silakan pilih produk dulu.")
    cur.close(); conn.close()
    return

id_transaksi = t[0]

cur.execute("SELECT id_metode_pembayaran, nama_transaksi FROM
metode_pembayaran")
metode = cur.fetchall()

if not metode:
    print("Belum ada metode pembayaran tersedia.")
    cur.close(); conn.close()
    return

for m in metode:
    print(f"{m[0]}. {m[1]}")

try:
    pilihan = int(input("Masukkan ID Metode Pembayaran: "))
except ValueError:

```

```

        print("Input harus angka!")
        cur.close(); conn.close()
        return

    cur.execute("SELECT id_metode_pembayaran FROM metode_pembayaran
WHERE id_metode_pembayaran = %s", (pilihan,))
    cek = cur.fetchone()
    if not cek:
        print("Metode pembayaran tidak valid.")
        cur.close(); conn.close()
        return

    cur.execute("""
        UPDATE transaksi
        SET id_metode_pembayaran = %s
        WHERE id_transaksi = %s
    """, (pilihan, id_transaksi))
    conn.commit()

    print("Metode pembayaran berhasil dipilih.")

    cur.close(); conn.close()

from tabulate import tabulate

def edit_metode_pembayaran(id_pelanggan):
    print("=== Edit Metode Pembayaran ===")
    conn, cur = connectDB()
    if not conn: return

    # cari transaksi aktif (status = proses) milik pelanggan
    cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE
nama_status = 'proses'")
    s = cur.fetchone()
    if not s:
        print("Status 'proses' tidak ditemukan.")
        cur.close(); conn.close()
        return
    id_status = s[0]

    cur.execute("""
        SELECT id_transaksi, id_metode_pembayaran

```

```

        FROM transaksi
        WHERE id_pelanggan = %s AND id_status_transaksi = %s
        ORDER BY id_transaksi DESC
        LIMIT 1
    """ , (id_pelanggan, id_status))
    t = cur.fetchone()
    if not t:
        print("Tidak ada transaksi aktif untuk pelanggan ini.")
        cur.close(); conn.close()
        return
    id_transaksi = t[0]

    # tampilkan daftar metode pembayaran (tetap 3 pilihan: tunai, qris,
transfer)
    cur.execute("SELECT id_metode_pembayaran, nama_transaksi FROM
metode_pembayaran ORDER BY id_metode_pembayaran")
    metode = cur.fetchall()
    headers = ["ID Metode", "Nama Metode"]
    print(tabulate(metode, headers=headers, tablefmt="fancy_grid"))

    try:
        pilihan = int(input("Masukkan ID Metode Pembayaran baru: "))
    except ValueError:
        print("Input harus angka!")
        cur.close(); conn.close()
        return

    cur.execute("SELECT id_metode_pembayaran FROM metode_pembayaran
WHERE id_metode_pembayaran = %s", (pilihan,))
    cek = cur.fetchone()
    if not cek:
        print("Metode pembayaran tidak valid.")
        cur.close(); conn.close()
        return

    # update transaksi aktif dengan metode baru
    cur.execute("""
        UPDATE transaksi
        SET id_metode_pembayaran = %s
        WHERE id_transaksi = %s
    """, (pilihan, id_transaksi))
    conn.commit()
    print("Metode pembayaran berhasil diganti untuk transaksi aktif.")

```

```

        cur.close(); conn.close()

def checkout(id_pelanggan):
    print("=== Checkout ===")
    conn, cur = connectDB()

    cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE
nama_status = 'proses'")
    s_proses = cur.fetchone()
    cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE
nama_status = 'selesai'")
    s_selesai = cur.fetchone()

    if not s_proses or not s_selesai:
        print("Status transaksi tidak lengkap (butuh 'proses' dan
'selesai').")
        cur.close(); conn.close()
        return

    id_status_proses = s_proses[0]
    id_status_selesai = s_selesai[0]

    cur.execute("""
        SELECT id_transaksi
        FROM transaksi
        WHERE id_pelanggan = %s AND id_status_transaksi = %s
        ORDER BY id_transaksi DESC
        LIMIT 1
    """, (id_pelanggan, id_status_proses))
    t = cur.fetchone()

    if not t:
        print("Tidak ada transaksi aktif untuk checkout.")
        cur.close(); conn.close()
        return

    id_transaksi = t[0]

    cur.execute("""
        SELECT d.id_produk, d.quantity, d.harga
        FROM detail_transaksi d
        WHERE d.id_transaksi = %s
    """, (id_transaksi,))

```

```

        """', (id_transaksi,))
    items = cur.fetchall()

    if not items:
        print("Keranjang kosong, tidak bisa checkout.")
        cur.close(); conn.close()
        return

    total_belanja = 0
    for item in items:
        id_produk, qty, subtotal = item
        total_belanja += subtotal

        cur.execute("UPDATE produk SET stok = stok - %s WHERE id_produk"
= %s", (qty, id_produk))

    cur.execute("""
        UPDATE transaksi
        SET id_status_transaksi = %s,
            tgl_transaksi = CURRENT_DATE
        WHERE id_transaksi = %s
    """, (id_status_selesai, id_transaksi))

    conn.commit()

    print("Checkout berhasil!")
    print("Total belanja: Rp", total_belanja)

    cur.close(); conn.close()

def lihat_riwayat_transaksi_pembelian(id_pelanggan):
    print("=== Riwayat Transaksi Pembelian ===")
    conn, cur = connectDB()

    cur.execute("""
        SELECT t.id_transaksi, t.tgl_transaksi,
        COALESCE(m.nama_transaksi, '-') AS metode_bayar,
            SUM(d.harga) AS total_belanja
        FROM transaksi t
        LEFT JOIN metode_pembayaran m ON t.id_metode_pembayaran =
m.id_metode_pembayaran
        JOIN detail_transaksi d ON t.id_transaksi = d.id_transaksi
    """)

```

```

        JOIN status_transaksi s ON t.id_status_transaksi =
s.id_status_transaksi
        WHERE t.id_pelanggan = %s AND s.nama_status = 'selesai'
        GROUP BY t.id_transaksi, t.tgl_transaksi, m.nama_transaksi
        ORDER BY t.tgl_transaksi DESC
        """ , (id_pelanggan,))
    riwayat = cur.fetchall()

    if not riwayat:
        print("Belum ada transaksi selesai.")
    else:
        headers = ["ID Transaksi", "Tanggal", "Metode Bayar", "Total
Belanja (Rp)"]
        print(tabulate(riwayat, headers=headers, tablefmt="fancy_grid"))

    cur.close(); conn.close()

def halaman_menu_manajer_produksi():
    while True:
        print("=== Halaman Menu Manajer Produksi ===")
        print("1. Lihat Biodata")
        print("2. Edit Biodata")
        print("3. Kelola Akun Karyawan")
        print("4. Lihat Laporan Produksi")
        print("5. Kelola Transaksi Pelanggan")
        print("6. Pergantian Manager Baru")
        print("7. Logout")
        pilihan_mgr = int(input("Pilih menu: "))

        if pilihan_mgr == 1:
            lihat_biodata_mgr()
        elif pilihan_mgr == 2:
            edit_bio_mgr()
        elif pilihan_mgr == 3:
            kelola_akun_karyawan()
        elif pilihan_mgr == 4:
            lihat_laporan_produksi()
        elif pilihan_mgr == 5:
            kelola_transaksi_pelanggan()
        elif pilihan_mgr == 6:
            ganti_mgr_baru()
        elif pilihan_mgr == 7:
            logout()

```



```

        exit()
    else:
        print("Pilihan tidak valid, coba lagi.")

def lihat_biodata_mgr():
    print("=== Lihat Biodata Manager ===")
    conn, cur = connectDB()
    username_karyawan = input("Masukkan username: ")
    query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
    cur.execute(query, (username_karyawan,))
    result = cur.fetchone()

    if result:
        biodata = [
            ["Nama", result[1]],
            ["Username", result[2]],
            ["Kata Sandi", result[3]],
            ["Gender", "Perempuan" if result[4] else "Laki-laki"],
            ["Tgl Lahir", result[5]],
            ["No Telepon", result[6]],
            ["Tgl Bergabung", result[7]],
            ["Gaji", result[9]]
        ]
        print("\n=== Biodata Manager ===")
        print(tabulate(biodata, headers=["Data", "Keterangan"],
tablefmt="fancy_grid"))
    else:
        print("Data biodata tidak ditemukan.")
    cur.close()
    conn.close()

def edit_bio_mgr():
    print("=== Edit Biodata Manager ===")
    conn, cur = connectDB()
    username_karyawan = input("Masukkan username manager: ")

    query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
    cur.execute(query, (username_karyawan,))
    result = cur.fetchone()

    if result:
        print("Masukkan data baru (kosongkan untuk tidak mengubah):")
        nama_baru = input(f>Nama ({result[1]}): ") or result[1]

```

```

        tgl_lahir_baru = input(f"Tgl Lahir ({result[5]} - YYYY-MM-DD): ")
    ) or result[5]

    no_telp_baru = input(f"No Telepon ({result[6]}): ") or result[6]

    update_query = """
        UPDATE karyawan
        SET nama_karyawan = %s, tgl_lahir = %s, no_telp = %s
        WHERE username_karyawan = %s
    """

    cur.execute(update_query, (nama_baru, tgl_lahir_baru,
no_telp_baru, username_karyawan))
    conn.commit()
    print("Biodata manager berhasil diperbarui.")
else:
    print("Data biodata manager tidak ditemukan.")
cur.close()
conn.close()

```

```

def kelola_akun_karyawan():
    while True:
        print("\n=== Kelola Akun Karyawan ===")
        print("1. Lihat Karyawan")
        print("2. Tambah Karyawan")
        print("3. Edit Karyawan")
        print("4. Hapus Karyawan")
        print("5. Kembali")
        pilihan = input("Pilih menu: ")

        if pilihan == "1":
            lihat_karyawan()
        elif pilihan == "2":
            tambah_karyawan()
        elif pilihan == "3":
            edit_karyawan()
        elif pilihan == "4":
            hapus_karyawan()
        elif pilihan == "5":
            break
        else:
            print("Pilihan tidak valid!")

```

```

def lihat_karyawan():

```

```

conn, cur = connectDB()
if not conn:
    return

cur.execute("SELECT * FROM karyawan")
data = cur.fetchall()

if data:
    print("\n=== Daftar Karyawan ===")
    headers = [desc[0] for desc in cur.description]
    print(tabulate(data, headers=headers, tablefmt="fancy_grid"))
else:
    print("Belum ada data karyawan.")

cur.close()
conn.close()

def tambah_karyawan():
    conn, cur = connectDB()
    if not conn: return

    nama = input("Nama: ")
    username = input("Username: ")
    sandi = input("Password: ")
    gender_input = int(input("Gender (1. Perempuan / 0. Laki-laki): "))
    gender = True if gender_input == 1 else False
    status_input = int(input("Status (1. Aktif / 0. Nonaktif): "))
    status = True if status_input == 1 else False
    tgl_lahir = input("Tanggal Lahir (YYYY-MM-DD): ")
    no_telp = input("No Telepon: ")
    id_jabatan = input("ID Jabatan: ")
    gaji = int(input("Gaji: "))

    cur.execute("""
    INSERT INTO karyawan
    (nama_karyawan, username_karyawan, kata_sandi_karyawan, gender,
status_karyawan, tgl_lahir, no_telp, tgl_masuk, id_jabatan,
gaji_karyawan)
    VALUES (%s, %s, %s, %s, %s, %s, %s, %s, %s, %s)
    """, (nama, username, sandi, gender, status, tgl_lahir, no_telp,
datetime.now().date(), id_jabatan, gaji))
    conn.commit()
    print("Karyawan berhasil ditambahkan.")

```

```

        cur.close(); conn.close()

def edit_karyawan():
    conn, cur = connectDB()
    if not conn: return
    id_karyawan = input("Masukkan ID Karyawan: ")
    cur.execute("SELECT * FROM karyawan WHERE id_karyawan = %s",
(id_karyawan,))
    result = cur.fetchone()
    if result:
        nama_baru = input(f>Nama ({result[1]}): ") or result[1]
        username_baru = input(f"Username ({result[2]}): ") or result[2]
        sandi_baru = input(f>Password ({result[3]}): ") or result[3]

        cur.execute("""
            UPDATE karyawan
            SET nama_karyawan = %s, username_karyawan = %s,
kata_sandi_karyawan = %s
            WHERE id_karyawan = %s
            """, (nama_baru, username_baru, sandi_baru, id_karyawan))
        conn.commit()
        print("Data karyawan berhasil diperbarui.")
    else:
        print("Karyawan tidak ditemukan.")
    cur.close(); conn.close()

def hapus_karyawan():
    conn, cur = connectDB()
    if not conn: return
    id_karyawan = input("Masukkan ID Karyawan: ")
    cur.execute("DELETE FROM karyawan WHERE id_karyawan = %s",
(id_karyawan,))
    conn.commit()
    if cur.rowcount > 0:
        print("Karyawan berhasil dihapus.")
    else:
        print("Karyawan tidak ditemukan.")
    cur.close(); conn.close()

def ganti_mgr_baru():
    print("=== Ganti Manager Baru ===")

```

```

conn, cur = connectDB()
if not conn: return

username_karyawan = input("Masukkan username manager lama: ")

query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
cur.execute(query, (username_karyawan,))
result = cur.fetchone()

if result:
    nama_baru = input("Nama Manager Baru: ")
    gender = int(input("Gender (1. Perempuan / 0. Laki-laki): "))
    gender_baru = True if gender == 1 else False
    tgl_lahir_baru = input("Tgl Lahir (YYYY-MM-DD): ")
    no_telp_baru = input("No Telepon: ")
    tgl_masuk_baru = datetime.now().date()

    update_query = """
        UPDATE karyawan
        SET nama_karyawan = %s, gender = %s, tgl_lahir = %s, no_telp
= %s, tgl_masuk = %s
        WHERE username_karyawan = %s
    """
    cur.execute(update_query, (nama_baru, gender_baru,
tgl_lahir_baru, no_telp_baru, tgl_masuk_baru, username_karyawan))
    conn.commit()
    print("Manager baru berhasil diganti. Tanggal masuk otomatis
diisi hari ini.")
else:
    print("Data manager tidak ditemukan.")

cur.close()
conn.close()

def kelola_transaksi_pelanggan():
    while True:
        print("\n=== Kelola Transaksi Pelanggan ===")
        print("1. Lihat Transaksi")
        print("2. Lihat Riwayat Transaksi")
        print("3. Kembali")
        pilihan = input("Pilih menu: ")

```

```

        if pilihan == "1":
            lihat_transaksi()
        elif pilihan == "2":
            lihat_riwayat_transaksi()
        elif pilihan == "3":
            break
        else:
            print("Pilihan tidak valid.")

from tabulate import tabulate

def lihat_transaksi():
    print("=== Lihat Transaksi ===")
    conn, cur = connectDB()
    if not conn:
        return

    cur.execute("SELECT * FROM transaksi")
    result = cur.fetchall()

    if result:
        print("\n=== Daftar Transaksi ===")
        headers = [desc[0] for desc in cur.description]
        print(tabulate(result, headers=headers, tablefmt="fancy_grid"))
    else:
        print("Transaksi tidak ditemukan.")

    cur.close()
    conn.close()

from tabulate import tabulate

def lihat_riwayat_transaksi():
    print("=== Lihat Riwayat Transaksi ===")
    conn, cur = connectDB()
    if not conn: return

    cur.execute("""

```

```
        SELECT id_transaksi, tgl_transaksi, id_pelanggan,
id_status_transaksi, id_metode_pembayaran
        FROM transaksi
        WHERE id_status_transaksi = 1
        ORDER BY tgl_transaksi DESC
    """)
    data = cur.fetchall()

    if data:
        headers = ["ID Transaksi", "Tanggal", "ID Pelanggan", "Status",
"Metode Pembayaran"]
        print(tabulate(data, headers=headers, tablefmt="fancy_grid"))
    else:
        print("Belum ada riwayat transaksi dengan status 'proses'.")

    cur.close()
    conn.close()

main()
```