

# **TUGAS BESAR ALGORITMA DAN PEMROGRAMAN I**



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

### **TIM PENYUSUN**

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

### **TAHUN AKADEMIK**

**2024/2025 GASAL**

# DAFTAR ISI

<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>I. PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar belakang.....	1
1.2 Tujuan.....	1
1.3 Batasan masalah.....	2
<b>II. RANCANGAN APLIKASI.....</b>	<b>3</b>
2.1 Kebutuhan Sistem.....	3
2.3 Rancangan navigasi dan tampilan.....	9
<b>III. IMPLEMENTASI RANCANGAN.....</b>	<b>11</b>
3.1 Fitur Halaman Utama.....	11
3.2 Fitur Registrasi Pelanggan.....	11
3.3.1 Fitur Login (main).....	12
3.3.2 Fitur Login Karyawan.....	13
3.3.3 Fitur Login Pelanggan.....	15
3.4 Fitur Halaman Menu Admin Produksi.....	15
3.5 Fitur Biodata Karyawan.....	16
3.6 Fitur Biodata Pelanggan.....	17
3.7 Fitur Lihat Biodata Karyawan.....	18
3.8 Fitur Edit Biodata Karyawan.....	19
3.9 Fitur Hasil Panen.....	21
3.10 Fitur Input Hasil Panen.....	21
3.11 Fitur Hapus Hasil Panen.....	23
3.12 Fitur Lihat Laporan Hasil Panen.....	23
3.13 Fitur Edit Hasil Panen.....	24
3.14 Fitur Produksi.....	26
3.14 Fitur Input Produksi.....	26
3.15 Fitur Edit Produksi.....	27
3.16 Fitur Hapus Produksi.....	28
3.18 Fitur Lihat Laporan Produksi.....	28
3.17 Fitur Perbarui Status Produksi.....	29
3.19 Fitur Logout.....	30
3.20 Fitur Halaman Menu Admin Gudang.....	30
3.21 Fitur Produk.....	31
3.21 Fitur Input Produk.....	32
3.22 Fitur Hapus Produk.....	33
3.23 Fitur Lihat Laporan Produk.....	34
3.24 Fitur Edit Produk.....	35

3.25	Fitur Transaksi.....	36
3.26	Fitur Update Status Transaksi.....	37
3.26	Fitur Lihat Laporan Transaksi.....	38
3.27	Fitur Halaman Menu Manager Produksi.....	39
3.28	Fitur Kelola Akun Karyawan.....	39
3.25.2	Lihat Karyawan.....	40
3.25.3	Tambah Karyawan.....	41
3.25.4	Edit Karyawan.....	43
3.25.5	Hapus Karyawan.....	43
3.29	Fitur Kelola Transaksi Pelanggan.....	44
3.26.1	Menu Utama Kelola Transaksi Pelanggan.....	44
3.26.2	Lihat Transaksi.....	45
3.26.3	Lihat Riwayat Transaksi.....	45
3.30	Fitur Pergantian Manager Baru.....	46
3.31	Fitur Halaman Menu Pelanggan.....	47
3.31	Fitur Lihat Biodata (Pelanggan).....	48
3.32	Fitur Edit Biodata (Pelanggan).....	49
3.33	Fitur Lihat Daftar Produk.....	51
3.34	Fitur Pilih Produk.....	51
3.35	Fitur Belanja Produk.....	53
3.36	Fitur Lihat Keranjang.....	54
3.37	Fitur Edit Keranjang.....	56
3.38	Fitur Pilih Metode Pembayaran.....	57
3.39	Fitur Edit Metode Pembayaran.....	58
3.40	Fitur Checkout.....	59
3.41	Fitur Lihat Riwayat Transaksi Pembelian.....	61
<b>IV.</b>	<b>HASIL DAN PEMBAHASAN.....</b>	<b>62</b>
4.1	Fitur Halaman Menu Utama.....	62
4.2	Fitur Registrasi Pelanggan.....	63
4.3	Fitur Login.....	63
4.3.1	Fitur Login (main).....	64
4.3.2	Fitur Login Karyawan.....	64
4.3.3	Fitur Login Pelanggan.....	65
4.4	Fitur Halaman Menu Admin Produksi.....	65
4.5	Fitur Lihat Biodata Karyawan.....	66
4.6	Fitur Edit Biodata Karyawan.....	67
4.7	Fitur Input Hasil Panen.....	67
4.8	Fitur Hapus Hasil Panen.....	68
4.9	Fitur Lihat Laporan Hasil Panen.....	69
4.10	Fitur Edit Hasil Panen.....	70
4.11	Fitur Input Produksi.....	71
4.12	Fitur Edit Produksi.....	72
4.13	Fitur Hapus Produksi.....	73
4.14	Fitur Lihat Laporan Produksi.....	74

4.15	Fitur Logout.....	74
4.16	Fitur Halaman Menu Admin Gudang.....	75
4.17	Fitur Input Produk.....	76
4.18	Fitur Hapus Produk.....	76
4.19	Fitur Lihat Laporan Produk.....	77
4.20	Fitur Edit Produk.....	78
4.21	Fitur Update Status Transaksi.....	79
4.22	Fitur Lihat Laporan Transaksi.....	79
4.23	Fitur Halaman Menu Manager Produksi.....	80
4.24	Fitur Kelola Akun Karyawan.....	80
4.24.1	Menu Utama Kelola Akun Karyawan.....	81
4.24.2	Lihat Karyawan.....	81
4.24.3	Tambah Karyawan.....	82
4.24.4	Edit Karyawan.....	83
4.24.5	Hapus Karyawan.....	84
4.25	Fitur Kelola Transaksi Pelanggan.....	84
4.25.1	Menu Utama Kelola Transaksi Pelanggan.....	84
4.25.2	Lihat Transaksi.....	85
4.25.3	Lihat Riwayat Transaksi.....	86
4.26	Fitur Pergantian Manager Baru.....	86
4.27	Fitur Halaman Menu Pelanggan.....	87
4.28	Fitur Lihat Biodata (Pelanggan).....	87
4.29	Fitur Edit Biodata (Pelanggan).....	88
4.30	Fitur Pilih Produk.....	89
4.31	Fitur Lihat Keranjang.....	90
4.32	Fitur Edit Keranjang.....	90
4.33	Fitur Pilih Metode Pembayaran.....	91
4.34	Fitur Edit Metode Pembayaran.....	92
4.35	Fitur Checkout.....	92
4.36	Fitur Lihat Riwayat Transaksi Pembelian.....	93
<b>DAFTAR PUSTAKA.....</b>		<b>94</b>
<b>LAMPIRAN.....</b>		<b>95</b>

# **I. PENDAHULUAN**

## **1.1 Latar belakang**

PT Mitratani Dua Tujuh merupakan perusahaan swasta yang bergerak di bidang agroindustri dengan edamame sebagai produk unggulan. Perusahaan ini mengelola proses produksi mulai dari budidaya, panen, pengolahan, hingga distribusi produk edamame beku. Dalam operasional sehari-hari, pencatatan hasil panen, stok gudang, dan pelaporan produksi masih dilakukan secara manual. Kondisi ini menimbulkan berbagai kendala, seperti risiko kesalahan input, duplikasi data, serta keterlambatan dalam penyusunan laporan.

Seiring meningkatnya volume produksi dan jumlah petani mitra, sistem manual menjadi semakin tidak efisien. Perusahaan membutuhkan solusi digital yang mampu mendukung pencatatan dan pelaporan secara lebih terstruktur, akurat, dan cepat. Oleh karena itu, dikembangkanlah sistem Agromame yang berfungsi untuk mendigitalisasi proses pencatatan hasil panen, stok gudang, transaksi, serta pelaporan produksi.

Selain fitur pengelolaan produksi dan stok, sistem Agromame juga memiliki fitur pemesanan pelanggan dan transaksi penjualan. Fitur ini memungkinkan pelanggan untuk melakukan pemesanan produk secara langsung melalui sistem, dengan harga yang disesuaikan berdasarkan kategori. Setiap pemesanan akan otomatis tercatat sebagai transaksi, lengkap dengan detail produk, jumlah, metode pembayaran, serta status transaksi. Dengan adanya fitur ini, perusahaan tidak hanya dapat mengelola produksi secara internal, tetapi juga membangun hubungan yang lebih transparan dengan pelanggan melalui pencatatan riwayat pembelian yang jelas.

Dengan adanya sistem ini, diharapkan pengelolaan data produksi edamame dan transaksi penjualan dapat dilakukan secara real time, transparan, dan efisien. Selain itu, sistem juga mendukung pengambilan keputusan manajerial dengan menyediakan laporan yang lebih akurat dan mudah diakses, sekaligus memperluas jangkauan distribusi produk kepada pelanggan.

## **1.2 Tujuan**

- Mengimplementasikan pencatatan digital untuk data panen edamame yang lebih terstruktur dan valid.
- Meningkatkan akurasi data dengan meminimalisasi kesalahan input yang sering terjadi pada sistem manual.
- Mempermudah pengelolaan stok gudang bahan baku dan hasil produksi.
- Menyediakan fitur pencatatan transaksi produksi secara efisien dan terdokumentasi.
- Meningkatkan kecepatan dan ketepatan dalam penyusunan laporan.
- Memfasilitasi pemesanan produk oleh pelanggan secara langsung melalui sistem.

- Mencatat transaksi penjualan secara otomatis dengan detail produk dan status pembayaran.
- Menyediakan riwayat pembelian pelanggan yang jelas dan terintegrasi.

### **1.3 Batasan masalah**

Batasan masalah dalam pengembangan sistem Agromame difokuskan hanya pada proses operasional edamame, yaitu pencatatan hasil panen, pengelolaan stok gudang, proses produksi, dan penyusunan laporan internal. Sistem ini tidak membahas detail aktivitas budidaya di lapangan, tidak menangani aspek keuangan, dan tidak terintegrasi dengan mesin atau alat produksi sehingga seluruh data tetap diinput oleh petugas secara manual.

Selain itu, sistem ini hanya digunakan untuk kebutuhan internal perusahaan dengan peran pengguna terbatas pada admin produksi, admin gudang, dan manajer produksi. Selain itu digunakan oleh pelanggan untuk melakukan pemesanan produk yang diperjual belikan. Dengan demikian, Agromame dibatasi sebagai sistem pencatatan dan pelaporan yang sederhana namun akurat untuk mendukung kelancaran operasional perusahaan.

## II. RANCANGAN APLIKASI

### 2.1 Kebutuhan Sistem

- a. Aplikasi memiliki fitur register sebagai pendaftaran akun Pelanggan.
- b. Aplikasi memiliki fitur login sebagai Admin Produksi yang memiliki akses untuk menginput data hasil panen, produksi, dan produk.
- c. Aplikasi memiliki fitur login sebagai Admin Gudang yang memiliki akses untuk menginput data stok dan
- d. Aplikasi memiliki fitur login sebagai Manager Produksi yang memiliki akses untuk mengelola transaksi yang dilakukan oleh Pelanggan.
- e. Aplikasi memiliki fitur login sebagai Pelanggan yang memiliki akses untuk memperbarui biodata Pelanggan agar data Pelanggan yang tersimpan akurat, melakukan pembelian produk, dan melakukan transaksi.
- f. Aplikasi memiliki fitur riwayat transaksi bagi Pelanggan dan Manajer Produksi untuk menampilkan catatan pembelian dan lebih mudah untuk mengetahui produk mana yang sering dibeli maupun jarang keluar.
- g. Aplikasi memiliki fitur logout bagi Admin Produksi, Admin Gudang, Manajer Produksi, dan Pelanggan.
- h. Aplikasi membagi fitur untuk berbagi tipe pengguna sebagai berikut :
  1. Manajer Produksi
  2. Admin Produksi
  3. Admin Gudang
  4. Pelanggan

### 2.2 Struktur Data

#### 2.2.1 Admin Produksi

No.	Fitur	Data	Tipe data	Penyimpanan
1.	Login	Username Kata Sandi	String String	Postgresql
2.	Biodata (Lihat Biodata)	Nama Username Kata Sandi Gender Tanggal Lahir No Telp Tanggal Bergabung Gaji	String String String String Date String Date Integer	Postgresql
3.	Biodata (Edit Biodata)	Nama Username Kata Sandi	String String String	Postgresql

		No Telp	String	
4.	Hasil Panen (Input Hasil Panen)	Jumlah Panen Tanggal Panen Nama karyawan	Float Date String	Postgresql
5.	Hasil Panen (Hapus Hasil Panen)	Id_hp	Integer	Postgresql
6.	Hasil Panen (Edit Hasil Panen)	Id_hp Tanggal panen baru Jumlah Panen Baru Nama karyawan	Integer Date Integer String	Postgresql
7.	Hasil Panen (Lihat Laporan Hasil Panen)	Id_hp Jumlah Panen Tgl Panen Nama karyawan	Integer Integer Date String	Postgresql
8.	Produksi (Input Produksi)	Tgl Produksi Jumlah Produksi Nama Karyawan Id_hp Id_status_produksi	Date Float String Integer Integer	Postgresql
9.	Produksi (Edit Produksi)	Id_produksi Tgl Produksi Baru Jumlah Produksi Baru Nama karyawan Id_hp_baru Id_status_produksi_baru Tgl Produksi Selesai	Integer Date Integer  String Integer Integer Date	Postgresql
10.	Produksi (Lihat Laporan Produksi)	Id_produksi Tgl Produksi Tgl Selesai Jumlah Produksi Nama karyawan Jumlah Hasil Panen Status Produksi	Integer Date Date Integer String Float String	Postgresql
11.	Produksi (Hapus Produksi)	Id_produksi	Integer	Postgresql
12.	Logout			Postgresql

### 2.2.2 Admin Gudang

No.	Fitur	Data	Tipe data	Penyimpanan
1.	Login	Username Kata Sandi	String String	Postgresql
2.	Biodata (Lihat Biodata)	Nama Username Kata Sandi Gender Tanggal lahir No Telepon Tgl Bergabung Gaji	Integer Date String String Date String Date Integer	Postgresql
3.	Biodata (Edit Biodata)	Username Nama Baru Username Baru Kata Sandi Baru No Telp Baru	String String String String String	Postgresql
5.	Produk (Input Produk)	Jumlah stok Id_jenis_produk	Integer Integer	Postgresql
6.	Produk (Lihat Laporan Produk)	Id_produk Nama Jenis Stok Harga Produk Tgl Update Stok Jumlah Produksi	Integer String Integer Integer Date Integer	Postgresql
7.	Produk (Edit Data Produk)	Id_produk Stok Baru Tgl Update Baru	Integer Integer Date	Postgresql
8.	Produk (Hapus Produk)	Id_produk	Integer	Postgresql
9.	Transaksi (Lihat Laporan Transaksi)	Id_detail_transaksi Id_transaksi Tgl_transaksi Nama Jenis Quantity Nama Status	Integer Integer Date String Integer Integer	Postgresql
10.	Transaksi (Perbarui Status Transaksi)	Id_transaksi Status Baru	Integer Integer	Postgresql
11.	Logout			Postgresql

### 2.2.3 Manajer Produksi

No.	Fitur	Data	Tipe data	Penyimpanan
1.	Login	Username Kata Sandi	String String	Postgresql
2.	Lihat Laporan Transaksi	Id_transaksi Nama Jenis Produk Jumlah Produk Total	Integer String Integer Integer	Postgresql
3.	Biodata (Lihat Biodata)	Nama Username Kata Sandi Gender Tanggal lahir No Telepon Tgl Bergabung Gaji	Integer Date String String Date String Date Integer	Postgresql
4.	Biodata (Edit Biodata)	Nama Tanggal lahir Nomor Telepon	String Date Integer	Postgresql
5.	Kelola Akun Karyawan (Lihat Karyawan)	Nama Username Kata Sandi Gender Tanggal lahir No Telepon Tgl Bergabung Status Karyawan Gaji Jabatan	Integer Date String String Date String Date Boolean Integer Integer	Postgresql
6.	Kelola Akun Karyawan (Tambah Karyawan)	Nama Username Kata Sandi Gender Tanggal lahir No Telepon Tgl Bergabung Status Karyawan Gaji Jabatan	Integer Date String String Date String Date Boolean Integer Integer	Postgresql
7.	Kelola Akun Karyawan	Nama Username	Integer Date	Postgresql

#### 2.2.4 untuk Pelanggan

No.	Fitur	Data	Tipe data	Penyimpanan
1.	Register	Username Password Nama Nomor Telepon	String String String String	Postgresql
2.	Login	Username Kata Sandi	String String	Postgresql

3.	Biodata (Lihat Biodata)	Nama Alamat Nomor Telepon	String String Integer	Postgresql
4.	Biodata (Edit Biodata)	Nama Alamat Nomor Telepon	String String Integer	Postgresql
5.	Belanja Produk (Pilih Produk)	Id_produk Jumlah Produk yang dibeli Harga	Integer Integer Integer	Postgresql
6.	Belanja Produk (Lihat Keranjang)	Id_barang Nama barang Jumlah produk yang dibeli Harga Total	Integer String Integer Integer Integer	Postgresql
7.	Belanja Produk (Edit Keranjang)	Id_produk	Integer	Postgresql
8.	Belanja Produk (Pilih Metode Pembayaran)	Id_barang Jumlah produk yang dibeli Harga	Integer Integer Integer	Postgresql
9.	Belanja Produk (Edit Metode Pembayaran)	Id_metode_pembayaran	Integer	Postgresql
10.	Belanja Produk (Checkout)	Id_transaksi Id_produk Nama Jenis Produk Jumlah produk yang dibeli Harga Total	Integer Integer String Integer Integer Integer	Postgresql
11.	Lihat Riwayat Transaksi Pembelian	Id_transaksi Nama Jenis Produk Jumlah Total	Integer String Integer Integer	Postgresql
13.	Logout			Postgresql

## 2.3 Rancangan navigasi dan tampilan

Definisi navigasi atau alur aplikasi ketika Manajer Produksi, Admin Produksi, Admin Gudang, dan Pelanggan mengoperasikan aplikasi yang kami buat adalah sebagai berikut :

### – Alur Manajer Produksi

1. Manajer Produksi login terlebih dulu menginputkan username dan kata sandi.
2. Setelah melalui proses login, manajer produksi bisa dapat langsung mengelola akun pengguna yaitu admin produksi, admin gudang, dan pelanggan.
3. Selain itu, Manajer Produksi dapat memantau aktivitas pengguna seperti melihat laporan produksi, laporan stok yang tersedia, dan melihat riwayat transaksi pengguna.

### – Alur Admin Produksi

1. Admin Produksi terlebih dahulu melakukan login dengan mengisi username dan kata sandi yang telah ditentukan.
2. Jika sudah berhasil login maka, Admin Produksi akan masuk ke halaman menu.
3. Admin Produksi dapat melakukan input dan edit terkait hasil panen, produk, jenis produk, dan produksi serta status produksi.

### – Alur Admin Gudang

1. Admin Gudang terlebih dahulu melakukan login dengan mengisi username dan kata sandi yang telah ditentukan.
2. Jika sudah berhasil login maka Admin Gudang akan masuk ke halaman menu.
3. Selanjutnya, admin gudang dapat mengelola transaksi, seperti dapat melihat riwayat pembayaran untuk mengecek stok keluar
4. Admin gudang juga dapat mengupdate stok.

### – Alur Pelanggan

1. Pelanggan melakukan register untuk membuat akun.
2. Lalu pelanggan dapat melakukan login dengan username dan kata sandi yang telah dibuat sebelumnya.
3. Jika sudah berhasil login, pelanggan akan masuk ke halaman menu dan dapat melengkapi data atau mengedit biodata jika terdapat kesalahan ketika membuat akun.
4. Kemudian pelanggan dapat memulai melakukan aktivitas seperti melihat produk, melihat harga produk, dan jumlah stok yang tersedia

5. Selanjutnya pelanggan dapat melanjutkan aktivitas pembelian produk, pelanggan juga dapat mengedit inputan data tersebut seperti menghapus produk belanja dan mengubah produk belanjaan.
6. Jika pelanggan sudah melihat produk apa saja yang dibeli dan sudah sesuai, maka Pelanggan dapat memilih metode pembayaran seperti tunai atau non tunai.
7. Setelah memilih metode pembayaran, pelanggan dapat lanjut ke proses checkout.
8. Kemudian, pelanggan dapat melihat riwayat transaksi yang telah dilakukan.

## III. IMPLEMENTASI RANCANGAN

### 3.1 Fitur Halaman Utama

#### Kode Program Halaman Utama

```
1 def main():
2     while True:
3         clear_screen()
4         banner()
5         spinner(0.6, "Memuat menu")
6
7         pilihan = questionnaire.select(
8             "Pilih opsi:",
9             choices=[
10                 "Register",
11                 "Login",
12                 "Keluar"
13             ]
14         ).ask()
15
16         if pilihan == "Register":
17             registerasi()
18         elif pilihan == "Login":
19             login()
20         elif pilihan == "Keluar":
21             print(Fore.MAGENTA + "Program selesai. Terima kasih!" + Style.RESET_ALL)
22             break
23
```

Kode program diatas berguna untuk memberi koneksi ke database PostgreSQL dan menampilkan menu utama ke user. Di menu tersebut, user bisa pilih untuk daftar, login, atau keluar dari program. Kalau pilihannya salah, sistem kasih peringatan. Semua interaksi pakai tampilan tabel biar rapi dan mudah dibaca.

### 3.2 Fitur Registrasi Pelanggan

#### Kode Program Registrasi Pelanggan

```

1  def registrasi():
2      clear_screen()
3      banner()
4      print(Fore.CYAN + "==== Registrasi Pelanggan ==== " + Style.RESET_ALL)
5      nama_pelanggan= input("Masukkan nama pelanggan: ").strip()
6      username_pelanggan = input("Masukkan username: ").strip()
7      kata_sandi_pelanggan = getpass.getpass("Masukkan kata_sandi: ").strip()
8      no_telp_pelanggan = input("Masukkan nomor telepon: ").strip()
9
10     if not nama_pelanggan or not username_pelanggan or not kata_sandi_pelanggan:
11         print(Fore.RED + "Nama/username/password tidak boleh kosong." + Style.RESET_ALL)
12         prompt_enter()
13         return
14
15     conn, cur = connectDB()
16     if not conn:
17         prompt_enter(); return
18     try:
19         # cek username unik
20         cur.execute("SELECT 1 FROM pelanggan WHERE username_pelanggan = %s", (username_pelanggan,))
21         if cur.fetchone():
22             print(Fore.RED + "Username sudah dipakai. Pilih yang lain." + Style.RESET_ALL)
23             conn.close(); cur.close()
24             prompt_enter(); return
25
26         query = """
27             INSERT INTO pelanggan (nama_pelanggan, username_pelanggan, kata_sandi_pelanggan, no_telp_pelanggan)
28             VALUES (%s, %s, %s, %s)
29         """
30         cur.execute(query, (nama_pelanggan, username_pelanggan, kata_sandi_pelanggan, no_telp_pelanggan))
31         conn.commit()
32         print(Fore.GREEN + "Registrasi berhasil!" + Style.RESET_ALL)
33     except Exception as e:
34         print(Fore.RED + "Gagal registrasi:", e, Style.RESET_ALL)
35         if conn:
36             conn.rollback()
37     finally:
38         try:
39             cur.close(); conn.close()
40         except:
41             pass
42     prompt_enter()

```

Kode program diatas dipakai untuk menyimpan data pelanggan baru ke database. User diminta untuk mengisi nama, username, password, dan nomor telepon, lalu datanya langsung dimasukin ke tabel pelanggan bareng tanggal registrasi. Kalau berhasil, muncul pesan sukses dan balik lagi ke menu utama.

## 3.3 Fitur login

### 3.3.1 Fitur Login (main)

**Kode Program Login (main)**

```
1 def login():
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "==== Login ==== " + Style.RESET_ALL)
6
7         pilihan = questionnaire.select(
8             "Pilih opsi:",
9             choices=[
10                 "Karyawan",
11                 "Pelanggan",
12                 "Kembali"
13             ]
14         ).ask()
15
16         if pilihan == "Karyawan":
17             login_karyawan()
18         elif pilihan == "Pelanggan":
19             login_pelanggan()
20         elif pilihan == "Kembali":
21             return
```

Kode program diatas berfungsi untuk menampilkan menu login, user bisa pilih login sebagai karyawan atau pelanggan. Kalau pilihannya valid, fungsi login yang sesuai akan dijalankan, kalau tidak valid, maka sistem minta input ulang.

### 3.3.2 Fitur Login Karyawan

#### Kode Program Login Karyawan

```

1 def login_karyawan():
2     clear_screen()
3     banner()
4     print(Fore.CYAN + "==== Login Karyawan ===" + Style.RESET_ALL)
5     username = input("Username: ").strip()
6     password = getpass.getpass("Password: ").strip()
7
8     karyawan = cari_karyawan_di_database(username, password)
9
10    if karyawan :
11        nama_lengkap = karyawan['nama_karyawan']
12        role = karyawan['role']
13
14        log_karyawan = [
15            ["Nama Lengkap", nama_lengkap],
16            ["Username", username],
17            ["Password", "*" * len(password)],
18            ["Role", role]
19        ]
20        print("\n==== Data Login Karyawan ===")
21        print(tabulate(log_karyawan, headers=["Data", "Keterangan"], tablefmt="fancy_grid"))
22        print(Fore.GREEN + f"\nSelamat datang {nama_lengkap}! Anda login sebagai {role}.\n" + Style.RESET_ALL)
23
24        if role == 'Manajer Produksi':
25            halaman_menu_manager_produksi()
26        elif role == 'Admin Produksi':
27            halaman_menu_admin_produksi()
28        elif role == 'Admin Gudang':
29            halaman_menu_admin_gudang()
30        else:
31            print(Fore.YELLOW + "Role tidak dikenali." + Style.RESET_ALL)
32    else:
33        print(Fore.RED + "Login gagal. Username atau password salah." + Style.RESET_ALL)
34    prompt_enter()
35

```

```

1 def cari_karyawan_di_database(username, password):
2     conn, cur = connectDB()
3     if not conn:
4         return None
5     try:
6         query = """SELECT k.nama_karyawan, k.username_karyawan, k.kata_sandi_karyawan, j.nama_jabatan
7                     FROM karyawan k
8                     JOIN jabatan j ON k.id_jabatan = j.id_jabatan
9                     WHERE k.username_karyawan = %s AND k.kata_sandi_karyawan = %s"""
10        cur.execute(query, (username, password))
11        result = cur.fetchone()
12        if result:
13            karyawan = {
14                'nama_karyawan': result[0],
15                'username': result[1],
16                'password': result[2],
17                'role': result[3]
18            }
19            return karyawan
20        return None
21    except Exception as e:
22        print(Fore.RED + "Error saat cari karyawan:", e, Style.RESET_ALL)
23        return None
24    finally:
25        try:
26            cur.close(); conn.close()
27        except:
28            pass
29

```

Kode program diatas berfungsi untuk memeriksa login karyawan ke database, kalau sudah benar, sistem ambil data jabatan dan arahkan ke menu sesuai peran. Kalau gagal login atau role-nya tidak dikenali, akan muncul pesan error.

### 3.3.3 Fitur Login Pelanggan

#### Kode Program Login Pelanggan

```
1 def login_pelanggan():
2     clear_screen()
3     banner()
4     print(Fore.CYAN + "==== Login Pelanggan ====" + Style.RESET_ALL)
5     username_pelanggan = input("Masukkan username: ").strip()
6     kata_sandi_pelanggan = getpass.getpass("Masukkan kata_sandi: ").strip()
7
8     conn, cur = connectDB()
9     if not conn:
10        prompt_enter(); return
11    try:
12        query = """SELECT id_pelanggan, nama_pelanggan, username_pelanggan, kata_sandi_pelanggan
13                     FROM pelanggan
14                     WHERE username_pelanggan = %s AND kata_sandi_pelanggan = %s"""
15        cur.execute(query, (username_pelanggan, kata_sandi_pelanggan))
16        result = cur.fetchone()
17
18        if result:
19            nama_pelanggan = result[1]
20            username = result[2]
21            password = result[3]
22
23            log_pelanggan = [
24                ["Nama Lengkap", nama_pelanggan],
25                ["Username", username],
26                ["Password", "*" * len(password)]
27            ]
28            print("\n--- Data Login Pelanggan ---")
29            print(tabulate(log_pelanggan, headers=["Data", "Keterangan"], tablefmt="fancy_grid"))
30            print(Fore.GREEN + f"\nSelamat datang {nama_pelanggan}! Anda berhasil login sebagai Pelanggan.\n" + Style.RESET_ALL)
31            halaman_menu_pelanggan(id_pelanggan = result[0])
32        else:
33            print(Fore.RED + "Login gagal! Username atau kata sandi salah." + Style.RESET_ALL)
34    except Exception as e:
35        print(Fore.RED + "Error saat login pelanggan:", e, Style.RESET_ALL)
36    finally:
37        try:
38            cur.close(); conn.close()
39        except:
40            pass
41    prompt_enter()
42
```

Kode program diatas berfungsi untuk login pelanggan dengan cara cek username dan password ke database. Kalau sudah cocok, maka sistem akan langsung menampilkan data pelanggan dan akan mengarahkan ke menu pelanggan, kalau salah makan akan muncul pesan gagal login.

## 3.4 Fitur Halaman Menu Admin Produksi

#### Kode Program Halaman Menu Admin Produksi

```
1 def halaman_menu_admin_produksi():
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "=== Halaman Menu Admin Produksi ===" + Style.RESET_ALL)
6
7         pilihan_adp = questionnaire.select(
8             "Pilih menu:",
9             choices=[
10                 "Biodata",
11                 "Hasil Panen",
12                 "Produksi",
13                 "Logout"
14             ]
15         ).ask()
16
17         if pilihan_adp == "Biodata":
18             biodata()
19         elif pilihan_adp == "Hasil Panen":
20             hasil_panen()
21         elif pilihan_adp == "Produksi":
22             produksi()
23         elif pilihan_adp == "Logout":
24             if questionnaire.confirm("Yakin ingin logout?").ask():
25                 print("logout...")
26                 spinner(0.4)
27                 return
28
29     prompt_enter()
```

Kode program diatas dibuat untuk menampilkan menu admin produksi dengan berbagai pilihan aksi, seperti input, edit, hapus, dan lihat data panen atau produksi. Sistem jalankan fungsi sesuai pilihan user, dan kalau input salah, muncul pesan error.

### 3.5 Fitur Biodata Karyawan

Kode Program Biodata Karyawan

```
1 def biodata():
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "=== Menu Biodata Karyawan ===" + Style.RESET_ALL)
6
7         pilihan_bio = questionnaire.select(
8             "Pilih menu:",
9             choices=[
10                 "Lihat Biodata",
11                 "Edit Biodata",
12                 "Kembali"
13             ]
14         ).ask()
15
16         if pilihan_bio == "Lihat Biodata":
17             lihat_biodata_karyawan()
18         elif pilihan_bio == "Edit Biodata":
19             edit_bio_karyawan()
20         elif pilihan_bio == "Kembali":
21             return
22         else:
23             print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
24
25     prompt_enter()
```

Kode program diatas berfungsi untuk menampilkan biodata karyawan berdasarkan username yang dimasukkan. Kalau data sudah ditemukan, sistem akan menampilkan info lengkap seperti nama, gender, tanggal lahir, dan gaji dalam format tabel. Kalau tidak maka akan muncul pesan error.

## 3.6 Fitur Biodata Pelanggan

Kode Program Biodata Pelanggan

```

1  def biodata_pelanggan(id_pelanggan):
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== AGROMAME: Menu Biodata ===" + Style.RESET_ALL)
6
7          pilihan = questionnaire.select(
8              "Pilih menu:",
9              choices=[
10                 "Lihat Biodata",
11                 "Edit Biodata",
12                 "Kembali"
13             ]
14          ).ask()
15
16          if pilihan == "Lihat Biodata":
17              lihat_biodata(id_pelanggan)
18          elif pilihan == "Edit Biodata":
19              edit_biodata(id_pelanggan)
20          elif pilihan == "Kembali":
21              break
22          else:
23              print(Fore.RED + "Pilihan tidak valid!" + Style.RESET_ALL)
24              prompt_enter()

```

### 3.7 Fitur Lihat Biodata Karyawan

Kode Program Lihat Biodata Karyawan

```

1 def lihat_biodata_karyawan():
2     clear_screen()
3     print("=== Lihat Biodata ===")
4     conn, cur = connectDB()
5     if not conn:
6         prompt_enter(); return
7     username_karyawan = input("Masukkan username: ")
8     try:
9         query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
10        cur.execute(query, (username_karyawan,))
11        result = cur.fetchone()
12
13        if result:
14            biodata = [
15                ["Nama", result[1]],
16                ["Username", result[2]],
17                ["Kata Sandi", result[3]],
18                ["Gender", "Perempuan" if result[4] else "Laki-laki"],
19                ["Tgl Lahir", result[5]],
20                ["No Telepon", result[6]],
21                ["Tgl Bergabung", result[7]],
22                ["Gaji", result[9]]
23            ]
24            print("\n=== Biodata Admin Produksi ===")
25            print(tabulate(biodata, headers=["Data", "Keterangan"], tablefmt="fancy_grid"))
26        else:
27            print(Fore.YELLOW + "Data biodata tidak ditemukan." + Style.RESET_ALL)
28    except Exception as e:
29        print(Fore.RED + "Error:", e, Style.RESET_ALL)
30    finally:
31        try:
32            cur.close(); conn.close()
33        except:
34            pass

```

Kode program diatas berfungsi untuk menampilkan biodata karyawan berdasarkan username yang dimasukkan. Kalau data sudah ditemukan, sistem akan menampilkan info lengkap seperti nama, gender, tanggal lahir, dan gaji dalam format tabel. Kalau tidak maka akan muncul pesan error.

### 3.8 Fitur Edit Biodata Karyawan

**Kode Program Edit Biodata Karyawan**

```

1  def edit_bio_karyawan():
2      clear_screen()
3      print("=== Edit Biodata ===")
4      conn, cur = connectDB()
5      if not conn:
6          prompt_enter()
7          return
8
9      username_karyawan = input("Masukkan username: ")
10
11     try:
12         query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
13         cur.execute(query, (username_karyawan,))
14         result = cur.fetchone()
15
16         if result:
17             print("Masukkan data baru (tekan Enter untuk tidak mengubah):")
18
19             nama_baru = input(f>Nama ({result[1]}): ") or result[1]
20             username_baru = input(f>Username ({result[2]}): ") or result[2]
21             kata_sandi_baru = input(f>Kata Sandi ({result[3]}): ") or result[3]
22             no_telp_baru = input(f>No Telepon ({result[6]}): ") or result[6]
23
24             update_q = """
25                 UPDATE karyawan
26                 SET nama_karyawan=%s, username_karyawan=%s, kata_sandi_karyawan=%s, no_telp=%s
27                 WHERE username_karyawan=%s
28             """
29             cur.execute(update_q, (nama_baru, username_baru, kata_sandi_baru, no_telp_baru, username_karyawan))
30             conn.commit()
31
32             print(Fore.GREEN + "Biodata berhasil diperbarui." + Style.RESET_ALL)
33             lihat_biodata_karyawan()
34         else:
35             print(Fore.YELLOW + "Data biodata tidak ditemukan." + Style.RESET_ALL)
36
37     except Exception as e:
38         print(Fore.RED + f"Error: {e}" + Style.RESET_ALL)
39         conn.rollback()
40     finally:
41         try:
42             cur.close()
43             conn.close()
44         except:
45             pass
46
47     def lihat_biodata_karyawan():
48         clear_screen()
49         print("=== Lihat Biodata ===")
50         conn, cur = connectDB()
51         if not conn:
52             prompt_enter(); return
53         username_karyawan = input("Masukkan username: ")
54         try:
55             query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
56             cur.execute(query, (username_karyawan,))
57             result = cur.fetchone()
58
59             if result:
60                 biodata = [
61                     ["Nama", result[1]],
62                     ["Username", result[2]],
63                     ["Kata Sandi", result[3]],
64                     ["Gender", "Perempuan" if result[4] else "Laki-laki"],
65                     ["Tgl Lahir", result[5]],
66                     ["No Telepon", result[6]],
67                     ["Tgl Bergabung", result[7]],
68                     ["Gaji", result[9]]
69                 ]
70                 print("\n=== Biodata Admin Gudang ===")
71                 print(tabulate(biodata, headers=["Data", "Keterangan"], tablefmt="fancy_grid"))
72             else:
73                 print(Fore.YELLOW + "Data biodata tidak ditemukan." + Style.RESET_ALL)
74         except Exception as e:
75             print(Fore.RED + "Error:", e, Style.RESET_ALL)
76         finally:
77             try:
78                 cur.close(); conn.close()
79             except:
80                 pass

```

Kode program diatas berfungsi untuk mengedit biodata karyawan di database. User menginput username, lalu sistem ambil data lama dan kasih opsi buat ganti nama, username, password, atau nomor telepon. Kalau username tidak ditemukan, muncul pesan error.

### 3.9 Fitur Hasil Panen

#### Kode Program Hasil Panen

```
1 def hasil_panen():
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "=== Menu Hasil Panen ===" + Style.RESET_ALL)
6
7         pilihan_hp = questionnaire.select(
8             "Pilih menu:",
9             choices=[
10                "Input Hasil Panen",
11                "Edit Hasil Panen",
12                "Hapus Hasil Panen",
13                "Lihat Laporan Hasil Panen",
14                "Kembali"
15            ]
16        ).ask()
17
18        if pilihan_hp == "Input Hasil Panen":
19            input_hp()
20        elif pilihan_hp == "Hapus Hasil Panen":
21            hapus_hp()
22        elif pilihan_hp == "Edit Hasil Panen":
23            edit_hp()
24        elif pilihan_hp == "Lihat Laporan Hasil Panen":
25            lihat_laporan_hp()
26        elif pilihan_hp == "Kembali":
27            return
28        else:
29            print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
30
31        prompt_enter()
```

Kode program diatas berguna untuk menyimpan data hasil panen ke database. User memasukkan jumlah panen, tanggal, dan ID karyawan, lalu sistem akan simpan datanya dan akan memberi konfirmasi jika berhasil.

### 3.10 Fitur Input Hasil Panen

#### Kode Program Input Hasil Panen

```

1 def input_hp():
2     clear_screen()
3     lihat_laporan_hp()
4     print("=== Input Hasil Panen ===")
5
6     hasil_panen_in = input("Masukkan jumlah hasil panen (kg) [Enter untuk batal]: ")
7     if hasil_panen_in.strip() == "":
8         print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
9         return
10    try:
11        hasil_panen = float(hasil_panen_in)
12    except ValueError:
13        print(Fore.RED + "Input harus angka." + Style.RESET_ALL)
14        prompt_enter()
15        return
16
17    tanggal_panen = input("Masukkan tanggal panen (YYYY-MM-DD) [Enter untuk batal]: ")
18    if tanggal_panen.strip() == "":
19        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
20        return
21
22    nama_karyawan = input("Masukkan nama karyawan [Enter untuk batal]: ").strip().lower()
23    if nama_karyawan == "":
24        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
25        return
26
27    print("\n--- Konfirmasi Data ---")
28    print(f"Jumlah Panen : {hasil_panen} kg")
29    print(f"Tanggal Panen: {tanggal_panen}")
30    print(f>Nama Karyawan: {nama_karyawan}")
31    konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
32    if konfirmasi != "y":
33        print(Fore.YELLOW + "Input dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
34        return
35
36    conn, cur = connectDB()
37    if not conn:
38        prompt_enter()
39        return
40
41    try:
42        cur.execute("SELECT id_karyawan FROM karyawan WHERE nama_karyawan = %s", (nama_karyawan,))
43        result = cur.fetchone()
44
45        if not result:
46            print(Fore.RED + "Nama karyawan tidak ditemukan." + Style.RESET_ALL)
47            conn.close()
48            return
49
50        id_karyawan = result[0]
51
52        query = """
53            INSERT INTO hasil_panen (jmlh_panen, tgl_panen, id_karyawan)
54            VALUES (%s, %s, %s)
55        """
56        cur.execute(query, (hasil_panen, tanggal_panen, id_karyawan))
57        conn.commit()
58        print(Fore.GREEN + "Input hasil panen berhasil!" + Style.RESET_ALL)
59        lihat_laporan_hp()
60
61    except Exception as e:
62        print(Fore.RED + "Gagal input:", e, Style.RESET_ALL)
63        conn.rollback()
64    finally:
65        try:
66            cur.close()
67            conn.close()
68        except:
69            pass

```

Kode program diatas berguna untuk menyimpan data hasil panen ke database. User memasukkan jumlah panen, tanggal, dan ID karyawan, lalu sistem akan simpan datanya dan akan memberi konfirmasi jika berhasil.

### 3.11 Fitur Hapus Hasil Panen

#### Kode Program Hapus Hasil Panen

```
1 def hapus_hp():
2     clear_screen()
3     lihat_laporan_hp()
4     print("=== Hapus Hasil Panen ===")
5     lihat_laporan_hp()
6     id_hp = safe_int("Masukkan ID hasil panen yang akan dihapus: ", default=None)
7     if not confirm("Yakin ingin menghapus hasil panen ini? (y/n): "):
8         print("Batal menghapus."); prompt_enter(); return
9     conn, cur = connectDB()
10    if not conn:
11        prompt_enter(); return
12    try:
13        query = "DELETE FROM hasil_panen WHERE id_hp = %s"
14        cur.execute(query, (id_hp,))
15        conn.commit()
16        print(Fore.GREEN + "Hasil Panen berhasil dihapus." + Style.RESET_ALL)
17        lihat_laporan_hp()
18    except Exception as e:
19        print(Fore.RED + "Gagal hapus:", e, Style.RESET_ALL)
20        conn.rollback()
21    finally:
22        try:
23            cur.close(); conn.close()
24        except:
25            pass
```

Kode program diatas berfungsi untuk menghapus data hasil panen dari database berdasarkan ID yang dimasukkan user, lalu sistem eksekusi query dan konfirmasi kalau data berhasil dihapus.

### 3.12 Fitur Lihat Laporan Hasil Panen

#### Kode Program Laporan Hasil Panen

```

1 def lihat_laporan_hp():
2     clear_screen()
3     print("=== Laporan Hasil Panen ===")
4     conn, cur = connectDB()
5     if not conn:
6         prompt_enter(); return
7     try:
8         query = """
9             SELECT hp.id_hp, hp.jmlh_panen, hp.tgl_panen, k.nama_karyawan
10            FROM hasil_panen hp
11            JOIN karyawan k ON hp.id_karyawan = k.id_karyawan
12            ORDER BY hp.tgl_panen
13        """
14        cur.execute(query)
15        results = cur.fetchall()
16        if not results:
17            print(Fore.YELLOW + "Tidak ada data hasil panen." + Style.RESET_ALL)
18        else:
19            df = pd.DataFrame(results, columns=['ID Hasil Panen', 'Hasil Panen (kg)', 'Tanggal Panen', 'Nama Karyawan'])
20            print(tabulate(df, headers='keys', tablefmt='fancy_grid', showindex=False))
21        except Exception as e:
22            print(Fore.RED + "Gagal mengambil laporan:", e, Style.RESET_ALL)
23        finally:
24            try:
25                cur.close(); conn.close()
26            except:
27                pass

```

Kode program diatas dibuat untuk menampilkan laporan hasil panen dari database, menggabungkan data panen dan nama karyawan, lalu ditampilkan dalam bentuk tabel yang rapi.

### 3.13 Fitur Edit Hasil Panen

#### Kode Program Edit Hasil Panen

```

1  def edit_hp():
2      clear_screen()
3      lihat_laporan_hp()
4      print("=== Edit Hasil Panen ===")
5      lihat_laporan_hp()
6
7      id_hp_in = input("Masukkan ID hasil panen yang akan diedit [Enter untuk batal]: ")
8      if id_hp_in.strip() == "":
9          print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
10         return
11     try:
12         id_hp = int(id_hp_in)
13     except ValueError:
14         print(Fore.RED + "ID harus angka." + Style.RESET_ALL)
15         return
16
17     conn, cur = connectDB()
18     if not conn:
19         prompt_enter()
20         return
21
22     try:
23         query = "SELECT * FROM hasil_panen WHERE id_hp = %s"
24         cur.execute(query, (id_hp,))
25         result = cur.fetchone()
26
27         if result:
28             print("Masukkan data baru (Enter untuk tidak mengubah / batal jika kosong di awal):")
29
30             tgl_panen_baru = input(f"Tanggal Panen ({result[1]}): ") or result[1]
31             jmlh_panen_baru = input(f"Jumlah Panen ({result[2]}): ") or result[2]
32             nama_karyawan = input(f>Nama Karyawan ({result[3]}) [Enter untuk batal]: ").strip().lower()
33             if nama_karyawan == "":
34                 print(Fore.YELLOW + "Edit dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
35                 return
36
37             cur.execute("SELECT id_karyawan FROM karyawan WHERE LOWER(nama_karyawan) = LOWER(%s)", (nama_karyawan,))
38             result_karyawan = cur.fetchone()
39
40             if not result_karyawan:
41                 print(Fore.RED + "Nama karyawan tidak ditemukan." + Style.RESET_ALL)
42                 conn.close()
43                 return
44
45             id_karyawan = result_karyawan[0]
46
47             print("\n--- Konfirmasi Data ---")
48             print(f"Tanggal Panen : {tgl_panen_baru}")
49             print(f"Jumlah Panen : {jmlh_panen_baru}")
50             print(f>Nama Karyawan : {nama_karyawan}")
51             konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
52             if konfirmasi != "y":
53                 print(Fore.YELLOW + "Edit dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
54                 return
55
56             update_query = """
57             UPDATE hasil_panen
58             SET tgl_panen = %s, jmlh_panen = %s, id_karyawan = %s
59             WHERE id_hp = %s
60             """
61             cur.execute(update_query, (tgl_panen_baru, jmlh_panen_baru, id_karyawan, id_hp))
62             conn.commit()
63             print(Fore.GREEN + "Hasil Panen berhasil diperbarui." + Style.RESET_ALL)
64             lihat_laporan_hp()
65         else:
66             print(Fore.YELLOW + "Data hasil panen tidak ditemukan." + Style.RESET_ALL)
67     except Exception as e:
68         print(Fore.RED + "Gagal edit:", e, Style.RESET_ALL)
69         conn.rollback()
70     finally:
71         try:
72             cur.close()
73             conn.close()
74         except:
75             pass

```

Kode program diatas berfungsi untuk mengedit data hasil panen di database berdasarkan ID. Jika data sudah ditemukan, user dapat mengubah tanggal, jumlah panen, dan ID karyawan, lalu sistem otomatis menyimpan perubahan dan memberikan konfirmasi. Kalau datanya tidak ada, maka akan muncul pesan error.

### 3.14 Fitur Produksi

#### Kode Program Produksi

```
1  def produksi():
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== Menu Produksi ===" + Style.RESET_ALL)
6
7          pilihan_prod = questionnaire.select(
8              "Pilih menu:",
9              choices=[
10                 "Input Produksi",
11                 "Edit Produksi",
12                 "Hapus Produksi",
13                 "Lihat Laporan Produksi",
14                 "Kembali"
15             ]
16          ).ask()
17
18          if pilihan_prod == "Input Produksi":
19              input_produk()
20          elif pilihan_prod == "Lihat Laporan Produksi":
21              lihat_laporan_produk()
22          elif pilihan_prod == "Edit Produksi":
23              edit_produk()
24          elif pilihan_prod == "Hapus Produksi":
25              hapus_produk()
26          elif pilihan_prod == "Kembali":
27              return
28          else:
29              print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
30
31      prompt_enter()
```

### 3.14 Fitur Input Produksi

#### Kode Program Input Produksi

```

1 def input_produksi():
2     print("=== Input Produksi ===")
3     conn, cur = connectDB()
4     tanggal_produksi = input("Masukkan tanggal produksi (YYYY-MM-DD): ")
5     jumlah_produksi = float(input("Masukkan jumlah produksi (kg): "))
6     id_karyawan = int(input("Masukkan ID karyawan: "))
7     id_hp = int(input("Masukkan ID hasil panen: "))
8     id_status_produksi = int(input("Masukkan ID status panen (1. selesai/2. ditunda/3. proses): "))
9     query = """
10     INSERT INTO produksi (tgl_produksi, jmlh_produksi, id_karyawan, id_hp, id_status_produksi)
11     VALUES (%s, %s, %s, %s, %s)
12     """
13     cur.execute(query, (tanggal_produksi, jumlah_produksi, id_karyawan, id_hp, id_status_produksi))
14     conn.commit()
15     print("Input produksi berhasil!")
16     cur.close()
17     conn.close()

```

Kode program tersebut berfungsi untuk menyimpan data produksi ke database. User memasukkan tanggal, jumlah produksi, ID karyawan, ID hasil panen, dan status produksi, lalu sistem menyimpan datanya lalu mengkonfirmasi jika berhasil

### 3.15 Fitur Edit Produksi

#### Kode Program Edit Produksi

```

1 def edit_produksi():
2     print("=== Edit Produksi ===")
3     conn, cur = connectDB()
4     id_produk = int(input("Masukkan ID produksi yang akan diedit: "))
5     query = "SELECT * FROM produksi WHERE id_produk = %s"
6     cur.execute(query, (id_produk,))
7     result = cur.fetchone()
8
9     if result:
10        print("Masukkan data baru (isi kembali untuk tidak mengubah):")
11        tgl_produk_baru = input(f"Tanggal Produksi ({result[1]}): ") or result[1]
12        jmlh_produk_baru = input(f"Jumlah Produksi ({result[2]}): ") or result[2]
13        id_karyawan_baru = input(f"ID Karyawan ({result[3]}): ") or result[3]
14        id_hp_baru = input(f"ID Hasil Panen ({result[4]}): ") or result[4]
15        id_status_produk_baru = input(f"ID Status Produksi ({result[5]}): ") or result[5]
16
17        if id_status_produk_baru == 1:
18            query = """
19            UPDATE produksi
20            SET id_status_produk = %s, tgl_selesai = %s
21            WHERE id_produk = %s"""
22            cur.execute(query, (id_status_produk_baru, datetime.now(), id_produk))
23        else:
24            query = """
25            UPDATE produksi
26            SET id_status_produk = %s, tgl_selesai = NULL,
27            WHERE id_produk = %s"""
28            cur.execute(query, (id_status_produk_baru, id_produk))
29
30        update_query = """
31        UPDATE produksi
32        SET tgl_produk = %s, tgl_selesai = %s, jmlh_produk = %s, id_karyawan = %s, id_hp = %s, id_status_produk = %s
33        WHERE id_produk = %s
34        """
35        cur.execute(update_query, (tgl_produk_baru, datetime.now(), jmlh_produk_baru, id_karyawan_baru, id_hp_baru, id_status_produk_baru, id_produk))
36        conn.commit()
37        print("Produksi berhasil diperbarui.")
38    else:
39        print("Data produksi tidak ditemukan.")
40
41    cur.close()
42    conn.close()

```

Kode program diatas dibuat untuk mengedit data produksi di database berdasarkan ID. Kalau data ditemukan, user bisa mengubah tanggal, jumlah produksi, ID karyawan, ID hasil panen yang baru, dan status produksi. Kalau statusnya selesai, sistem otomatis isi tanggal selesai, kalau belum selesai, maka tanggal selesai dikosongkan. Setelah itu data diperbarui dan disimpan. Kalau ID-nya tidak ditemukan, muncul pesan error.

### 3.16 Fitur Hapus Produksi

#### Kode Program Hapus Produksi

```
1 def hapus_produksi():
2     clear_screen()
3     lihat_laporan_produksi()
4     print("=== Hapus Produksi ===")
5     id_produksi = safe_int("Masukkan ID produksi yang akan dihapus: ", default=None)
6     if not confirm("Yakin menghapus produksi ini? (y/n): "):
7         print("Batal penghapusan."); prompt_enter(); return
8     conn, cur = connectDB()
9     if not conn:
10        prompt_enter(); return
11    try:
12        cur.execute("DELETE FROM produksi WHERE id_produksi = %s", (id_produksi,))
13        conn.commit()
14        print(Fore.GREEN + "Produksi berhasil dihapus." + Style.RESET_ALL)
15        lihat_laporan_produksi()
16    except Exception as e:
17        print(Fore.RED + "Gagal menghapus produksi:", e, Style.RESET_ALL)
18        conn.rollback()
19    finally:
20        try:
21            cur.close(); conn.close()
22        except:
23            pass
```

Kode program tersebut dibuat agar dapat menghapus data produksi dari database berdasarkan ID yang dimasukkan user, lalu sistem jalankan query dan konfirmasi kalau data berhasil dihapus

### 3.18 Fitur Lihat Laporan Produksi

#### Kode Program Lihat Laporan Produksi

```

1 def lihat_laporan_produk():
2     print("=== Lihat Laporan Produksi ===")
3     conn, cur = connectDB()
4     query = """
5     SELECT p.id_produk, p.tgl_produk, p.tgl_selesai, p.jmlh_produk,
6           k.nama_karyawan, hp.jmlh_panen, sp.nama_status
7     FROM produksi p
8     JOIN karyawan k ON p.id_karyawan = k.id_karyawan
9     JOIN hasil_panen hp ON p.id_hp = hp.id_hp
10    JOIN status_produk sp ON p.id_status_produk = sp.id_status_produk"""
11    cur.execute(query)
12    results = cur.fetchall()
13    df = pandas.DataFrame(results, columns=['ID Produk', 'Tanggal Produk', 'Tanggal Selesai', 'Jumlah Produk', 'Nama Karyawan', 'Jumlah Hasil Panen (kg)', 'Status Produk'])
14    print(tabulate(df, headers='keys', tablefmt='fancy_grid', showindex=False))
15    cur.close()
16    conn.close()

```

Kode program tersebut berfungsi untuk menampilkan laporan produksi dari database, menggabungkan data produksi, karyawan, hasil panen, dan status produksi, lalu ditampilkan dalam bentuk tabel yang rapi.

### 3.17 Fitur Perbarui Status Produksi

#### Kode Program Perbarui Status Produksi

```

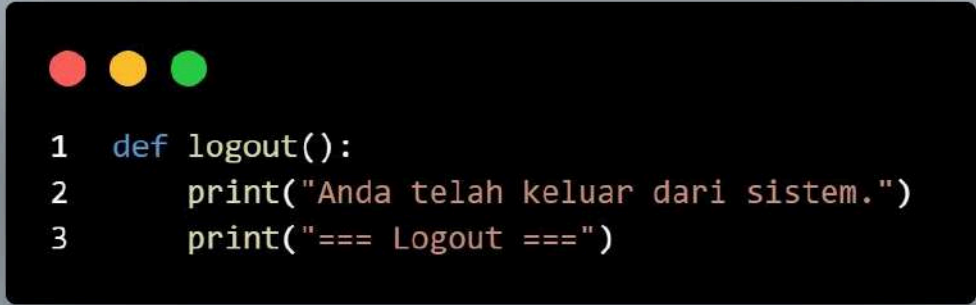
1 def perbarui_status_produk():
2     print("=== Perbarui Status Produksi ===")
3     conn, cur = connectDB()
4     id_produk = int(input("Masukkan ID produksi yang akan diperbarui: "))
5     status_baru = int(input("Masukkan status produksi baru (1. selesai/2. ditunda/3. proses): "))
6
7     if status_baru == 1:
8         query = """
9         UPDATE produksi
10        SET id_status_produk = %s, tgl_selesai = %s
11        WHERE id_produk = %s"""
12        cur.execute(query, (status_baru, datetime.now(), id_produk))
13    else:
14        query = """
15        UPDATE produksi
16        SET id_status_produk = %s, tgl_selesai = NULL
17        WHERE id_produk = %s"""
18        cur.execute(query, (status_baru, id_produk))
19    conn.commit()
20    print("Status produksi berhasil diperbarui!")
21    cur.close()
22    conn.close()

```

Kode program tersebut berfungsi untuk memperbarui status produksi di database berdasarkan ID. Kalau statusnya selesai, sistem isi tanggal selesai dengan waktu sekarang, kalau statusnya belum selesai, tanggal selesai dikosongkan. Setelah itu data disimpan dan muncul pesan berhasil.

### 3.19 Fitur Logout

#### Kode Program Logout



```
1 def logout():  
2     print("Anda telah keluar dari sistem.")  
3     print("=== Logout ===")
```

Kode program tersebut berfungsi untuk logout dari sistem, tampilan `print("Anda telah keluar dari sistem.")` merupakan tanda bahwa user sudah keluar dan proses logout sudah selesai.

### 3.20 Fitur Halaman Menu Admin Gudang

#### Kode Program Halaman Menu Admin Gudang

```

1  def halaman_menu_admin_gudang():
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== Halaman Menu Admin Gudang ===" + Style.RESET_ALL)
6
7          pilihan_adg = questionnaire.select(
8              "Pilih menu:",
9              choices=[
10                 "Biodata",
11                 "Produk",
12                 "Transaksi",
13                 "Logout"
14             ]
15          ).ask()
16
17          if pilihan_adg == "Biodata":
18              biodata()
19          elif pilihan_adg == "Produk":
20              produk()
21          elif pilihan_adg == "Transaksi":
22              transaksi()
23          elif pilihan_adg == "Logout":
24              if questionnaire.confirm("Yakin ingin logout?").ask():
25                  print("Logout...")
26                  spinner(0.4)
27                  return
28          else:
29              print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
30
31          prompt_enter()

```

Kode program diatas berfungsi untuk menampilkan menu admin gudang, user dapat memilih menu seperti lihat biodata, input produk, edit data, atau melihat laporan. Sistem menjalankan fungsi sesuai pilihan dan kasih pesan kalau input nggak valid.

### 3.21 Fitur Produk

Kode Program Produk

```

1  def produk():
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== Menu Produk ===" + Style.RESET_ALL)
6
7          pilihan_prod = questionnaire.select(
8              "Pilih menu:",
9              choices=[
10                 "Input Produk",
11                 "Edit Produk",
12                 "Hapus Produk",
13                 "Lihat Laporan Produk",
14                 "Kembali"
15             ]
16          ).ask()
17
18          if pilihan_prod == "Input Produk":
19              input_produk()
20          elif pilihan_prod == "Lihat Laporan Produk":
21              lihat_laporan_produk()
22          elif pilihan_prod == "Edit Produk":
23              edit_produk()
24          elif pilihan_prod == "Hapus Produk":
25              hapus_produk()
26          elif pilihan_prod == "Kembali":
27              return
28          else:
29              print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
30
31          prompt_enter()

```

Kode program tersebut digunakan untuk menampilkan menu produk. Pengguna memilih aksi seperti input produk, edit produk, hapus produk, atau melihat laporan. Setelah pilihan dipilih, sistem akan menjalankan fungsi yang sesuai. Jika memilih "Kembali", menu langsung ditutup. Jika pilihannya tidak valid, sistem memberi pesan error dan meminta pengguna mencoba lagi.

### 3.21 Fitur Input Produk

Kode Program Input Produk

```

1  def input_produk():
2      clear_screen()
3      lihat_laporan_produk()
4      print(Fore.CYAN + "=== Input Produk ===" + Style.RESET_ALL)
5      jumlah_dibutuhkan = safe_int("Masukkan jumlah stok: ", default=None)
6      id_jenis_produk = safe_int("Masukkan id jenis (1.Fresh, 2.Frozen, 3.Jus): ", default=None, allowed=[1, 2, 3])
7
8      conn, cur = connectDB()
9      if not conn:
10         prompt_enter()
11         return
12
13     try:
14         query = """
15             SELECT id_produk, jmlh_produk
16             FROM produksi
17             WHERE jmlh_produk > 0
18             ORDER BY tgl_produk ASC
19         """
20         cur.execute(query)
21         produksis = cur.fetchall()
22
23         sisa = jumlah_dibutuhkan
24         total_diambil = 0
25         id_produk_terpakai = []
26
27         for id_produk, jmlh_produk in produksis:
28             if sisa <= 0:
29                 break
30             if jmlh_produk >= sisa:
31                 cur.execute("""
32                     UPDATE produksi
33                     SET jmlh_produk = jmlh_produk - %s
34                     WHERE id_produk = %s
35                     """, (sisa, id_produk))
36                 total_diambil += sisa
37                 id_produk_terpakai.append(id_produk)
38                 sisa = 0
39             else:
40                 cur.execute("""
41                     UPDATE produksi
42                     SET jmlh_produk = 0
43                     WHERE id_produk = %s
44                     """, (id_produk,))
45                 total_diambil += jmlh_produk
46                 id_produk_terpakai.append(id_produk)
47                 sisa -= jmlh_produk
48
49         if sisa == 0:
50             cur.execute("""
51                 INSERT INTO produk (stok, id_jenis_produk, tgl_update_stok)
52                 VALUES (%s, %s, NOW())
53                 """, (jumlah_dibutuhkan, id_jenis_produk))
54
55             print(Fore.GREEN + "Input produk berhasil! Jumlah produksi berkurang dan produk tercatat." + Style.RESET_ALL)
56             lihat_laporan_produk()
57         else:
58             print(Fore.YELLOW + "Stok habis! Tidak bisa memenuhi semua permintaan." + Style.RESET_ALL)
59
60         conn.commit()
61
62     except Exception as e:
63         print(Fore.RED + "Gagal input produk:", e, Style.RESET_ALL)
64         conn.rollback()
65     finally:
66         try:
67             cur.close()
68             conn.close()
69         except:
70             pass

```

Kode program tersebut dibuat untuk menginput produk ke database agar admin produksi dapat cek stok produksi. Sistem kurangi jumlah produksi sesuai permintaan, kalau stok cukup maka input berhasil, kalau kurang muncul pesan stok habis.

## 3.22 Fitur Hapus Produk

### Kode Program Hapus Produk

```
1 def hapus_produk():
2     clear_screen()
3     lihat_laporan_produk()
4     print("=== Hapus Produk ===")
5     id_produk = safe_int("Masukkan ID produk yang akan dihapus: ", default=None)
6     if not confirm("Yakin menghapus produk ini? (y/n): "):
7         print("Batal menghapus."); prompt_enter(); return
8     conn, cur = connectDB()
9     if not conn:
10        prompt_enter(); return
11    try:
12        cur.execute("DELETE FROM produk WHERE id_produk=%s", (id_produk,))
13        conn.commit()
14        print(Fore.GREEN + "Produk berhasil dihapus." + Style.RESET_ALL)
15        lihat_laporan_produk()
16    except Exception as e:
17        print(Fore.RED + "Gagal hapus produk:", e, Style.RESET_ALL)
18        conn.rollback()
19    finally:
20        try:
21            cur.close(); conn.close()
22        except:
23            pass
```

Kode program diatas berfungsi untuk menghapus data produk dari database berdasarkan ID yang dimasukkan user, lalu sistem jalankan query dan tampilkan pesan kalau produk berhasil dihapus.

## 3.23 Fitur Lihat Laporan Produk

### Kode Program Lihat Laporan Produk

```

1 def lihat_laporan_produk():
2     clear_screen()
3     print("=== Laporan Produk ===")
4     conn, cur = connectDB()
5     if not conn:
6         prompt_enter(); return
7     try:
8         query = """
9             SELECT p.id_produk, j.nama_jenis, p.stok, p.tgl_update_stok, ps.jmlh_produksi
10            FROM produk p
11            JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
12            LEFT JOIN produksi ps ON p.id_produk = ps.id_produk
13            """
14         cur.execute(query)
15         results = cur.fetchall()
16         if not results:
17             print(Fore.YELLOW + "Belum ada produk tersedia." + Style.RESET_ALL)
18         else:
19             df = pd.DataFrame(results, columns=['ID Produk', 'Jenis Produk', 'Jumlah Produk', 'Tanggal Update Stok', 'Jumlah Produksi'])
20             print(tabulate(df, headers='keys', tablefmt='fancy_grid', showindex=False))
21     except Exception as e:
22         print(Fore.RED + "Gagal:", e, Style.RESET_ALL)
23     finally:
24         try:
25             cur.close(); conn.close()
26         except:
27             pass

```

Kode program tersebut dibuat untuk menampilkan laporan produk dari database, gabungkan data produk, jenis produk, dan produksi, lalu ditampilkan dalam bentuk tabel yang rapi.

## 3.24 Fitur Edit Produk

### Kode Program Edit Produk

```

1  def edit_produk():
2      clear_screen()
3      lihat_laporan_produk()
4      print("=== Edit Produk ===")
5      conn, cur = connectDB()
6      if not conn:
7          prompt_enter(); return
8
9      id_produk = safe_int("Masukkan ID produk yang akan diedit: ", default=None)
10     if id_produk is None:
11         print(Fore.YELLOW + "ID produk tidak valid." + Style.RESET_ALL)
12         return
13
14     try:
15         cur.execute("SELECT * FROM produk WHERE id_produk = %s", (id_produk,))
16         result = cur.fetchone()
17         if result:
18             stok_baru = input(f"Stok baru ({result[1]}), Enter untuk skip: ")
19
20             if stok_baru.strip() == "":
21                 print(Fore.YELLOW + "Tidak ada perubahan stok." + Style.RESET_ALL)
22             else:
23                 stok_baru = int(stok_baru) # pastikan integer
24                 update_query = """
25                     UPDATE produk
26                     SET stok = %s, tgl_update_stok = %s
27                     WHERE id_produk = %s
28                 """
29                 cur.execute(update_query, (stok_baru, datetime.now(), id_produk))
30                 conn.commit()
31                 print(Fore.GREEN + "Produk berhasil diperbarui." + Style.RESET_ALL)
32                 lihat_laporan_produk()
33             else:
34                 print(Fore.YELLOW + "Data produk tidak ditemukan." + Style.RESET_ALL)
35         except Exception as e:
36             print(Fore.RED + "Gagal edit produk:", e, Style.RESET_ALL)
37             conn.rollback()
38     finally:
39         try:
40             cur.close(); conn.close()
41         except:
42             pass

```

Kode program ini dibuat agar admin gudang dapat mengedit data produk di database berdasarkan ID. Kalau produk ditemukan, user bisa ubah stok dan sistem simpan perubahan dengan update waktu sekarang. Kalau nggak ditemukan, muncul pesan error.

### 3.25 Fitur Transaksi

Kode Transaksi

```

1  def transaksi():
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== Menu Transaksi ===" + Style.RESET_ALL)
6          pilihan_trans = questionnaire.select(
7              "Pilih menu:",
8              choices=[
9                  "Lihat Laporan Transaksi",
10                 "Update Status Transaksi",
11                 "Kembali"
12             ]
13          ).ask()
14          if pilihan_trans == "Update Status Transaksi":
15              update_status_transaksi()
16          elif pilihan_trans == "Lihat Laporan Transaksi":
17              lihat_laporan_transaksi()
18          elif pilihan_trans == "Kembali":
19              return
20          else:
21              print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
22
23      prompt_enter()

```

## 3.26 Fitur Update Status Transaksi

Kode Program Update Status Transaksi

```

1 def update_status_transaksi():
2     clear_screen()
3     lihat_laporan_transaksi()
4     print("=== Input Status Transaksi ===")
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9     id_transaksi = safe_int("Masukkan ID transaksi yang akan diperbarui: ", default=None)
10    status_baru = safe_int("Masukkan status produksi baru (1. selesai/ 2. proses /3. dibatalkan): ", default=None, allowed=[1,2,3])
11    try:
12        query = """
13            UPDATE transaksi
14            SET id_status_transaksi = %s
15            WHERE id_transaksi = %s
16        """
17        cur.execute(query, (status_baru, id_transaksi))
18        conn.commit()
19        print(Fore.GREEN + "Status transaksi berhasil diperbarui!" + Style.RESET_ALL)
20        lihat_laporan_transaksi()
21    except Exception as e:
22        print(Fore.RED + "Gagal update status:", e, Style.RESET_ALL)
23        conn.rollback()
24    finally:
25        try:
26            cur.close(); conn.close()
27        except:
28            pass

```

Kode program tersebut berfungsi untuk memperbarui status transaksi di database berdasarkan ID. Sistem ganti status sesuai input user dan simpan perubahan, lalu tampilkan pesan berhasil.

## 3.26 Fitur Lihat Laporan Transaksi

### Kode Program Lihat Laporan Transaksi

```

1 def lihat_laporan_transaksi():
2     clear_screen()
3     print("=== Laporan Transaksi ===")
4     conn, cur = connectDB()
5     if not conn:
6         prompt_enter()
7         return
8     try:
9         query = """
10            SELECT dt.id_detail_transaksi, t.id_transaksi, t.tgl_transaksi, jp.nama_jenis, dt.quantity, st.nama_status
11            FROM transaksi t
12            JOIN status_transaksi st ON t.id_status_transaksi = st.id_status_transaksi
13            JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
14            JOIN produk p ON dt.id_produk = p.id_produk
15            JOIN jenis_produk jp ON p.id_jenis_produk = jp.id_jenis_produk
16            ORDER BY t.tgl_transaksi
17        """
18        cur.execute(query)
19        results = cur.fetchall()
20        if not results:
21            print(Fore.YELLOW + "Belum ada transaksi." + Style.RESET_ALL)
22        else:
23            df = pd.DataFrame(results, columns=['ID Detail', 'ID Transaksi', 'Tanggal', 'Produk', 'Jumlah Produk', 'Status Transaksi'])
24            print(tabulate(df, headers='keys', tablefmt='fancy_grid', showindex=False))
25    except Exception as e:
26        print(Fore.RED + "Gagal ambil laporan transaksi:", e, Style.RESET_ALL)
27    finally:
28        try:
29            cur.close(); conn.close()
30        except:
31            pass

```

Kode program diatas berfungsi untuk menampilkan laporan transaksi dari database, gabungin data transaksi, produk, jenis produk, dan status, lalu ditampilkan dalam bentuk tabel yang rapi.

## 3.27 Fitur Halaman Menu Manager Produksi

### Kode Program Halaman Menu Manager Produksi

```
1 def halaman_menu_manager_produksi():
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "=== Halaman Menu Manager Produksi ===" + Style.RESET_ALL)
6
7         pilihan_mng = questionnaire.select(
8             "Pilih menu:",
9             choices=[
10                "Biodata",
11                "Kelola akun karyawan",
12                "Kelola transaksi pelanggan",
13                "Pergantian manager produksi",
14                "Kembali"
15            ]
16        ).ask()
17
18        if pilihan_mng == "Biodata":
19            biodata()
20        elif pilihan_mng == "Kelola akun karyawan":
21            kelola_akun_karyawan()
22        elif pilihan_mng == "Kelola transaksi pelanggan":
23            kelola_transaksi_pelanggan()
24        elif pilihan_mng == "Pergantian manager produksi":
25            ganti_mgr_baru()
26        elif pilihan_mng == "Kembali":
27            return
28        else:
29            print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
30
31        prompt_enter()
```

Kode program tersebut berfungsi untuk menampilkan menu manajer produksi, user pilih aksi seperti lihat biodata, kelola akun, atau laporan produksi, lalu sistem jalankan fungsi sesuai pilihan dan validasi inputnya.

## 3.28 Fitur Kelola Akun Karyawan

### 3.25.1 Menu Utama Kelola Akun Karyawan

### Kode Program Menu Utama Kelola Akun Karyawan

```

1  def kelola_akun_karyawan():
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== Kelola Akun Karyawan ===" + Style.RESET_ALL)
6
7          pilihan_akun = questionnaire.select(
8              "Pilih menu:",
9              choices=[
10                 "Tambah Karyawan",
11                 "Edit Karyawan",
12                 "Hapus Karyawan",
13                 "Lihat Laporan Karyawan",
14                 "Kembali"
15             ]
16         ).ask()
17
18         if pilihan_akun == "Tambah Karyawan":
19             tambah_karyawan()
20         elif pilihan_akun == "Edit Karyawan":
21             edit_karyawan()
22         elif pilihan_akun == "Hapus Karyawan":
23             hapus_karyawan()
24         elif pilihan_akun == "Lihat Laporan Karyawan":
25             lihat_karyawan()
26         elif pilihan_akun == "Kembali":
27             return
28         else:
29             print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
30
31     prompt_enter()

```

Kode program tersebut dibuat untuk menampilkan menu kelola akun karyawan, user bisa pilih lihat, tambah, edit, atau hapus data karyawan, dan sistem jalankan fungsi sesuai pilihan sampai user keluar.

### 3.25.2 Lihat Karyawan

#### Kode Program Lihat Karyawan

```

1 def lihat_karyawan():
2     clear_screen()
3     print("=== Lihat Data Karyawan ===")
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        cur.execute("""
12            SELECT k.id_karyawan, k.nama_karyawan, k.username_karyawan, k.kata_sandi_karyawan, k.gender, k.tgl_lahir,
13            k.no_telp, k.tgl_masuk, k.status_karyawan, k.gaji_karyawan, j.nama_jabatan
14            FROM karyawan k
15            JOIN jabatan j ON k.id_jabatan = j.id_jabatan
16            ORDER BY k.id_karyawan
17            """)
18        data = cur.fetchall()
19
20        if data:
21            print(Fore.GREEN + "\n=== Daftar Karyawan ===" + Style.RESET_ALL)
22            headers = ['Id Karyawan', 'Nama Karyawan', 'Username', 'Password', 'Gender', 'Tanggal Lahir', 'No Telepon', 'Tanggal Masuk', 'Status', 'Gaji', 'Jabatan']
23            print(tabulate(data, headers=headers, tablefmt="grid"))
24        else:
25            print(Fore.YELLOW + "Belum ada data karyawan." + Style.RESET_ALL)
26    except Exception as e:
27        print(Fore.RED + "Gagal menampilkan data:", e, Style.RESET_ALL)
28    finally:
29        try:
30            cur.close()
31            conn.close()
32        except:
33            pass
34
35    prompt_enter()

```

Kode program tersebut berfungsi untuk menampilkan semua data karyawan dari database dalam bentuk tabel, kalau datanya kosong muncul pesan belum ada karyawan.

### 3.25.3 Tambah Karyawan

#### Kode Program Tambah Karyawan

```

1 def tambah_karyawan():
2     clear_screen()
3     print("=== Tambah Karyawan ===")
4
5     nama = input("Nama [Enter untuk batal]: ")
6     if nama.strip() == "":
7         print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
8         return
9
10    username = input("Username [Enter untuk batal]: ")
11    if username.strip() == "":
12        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
13        return
14
15    sandi = input("Password [Enter untuk batal]: ")
16    if sandi.strip() == "":
17        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
18        return
19
20    gender_in = input("Gender (1. Perempuan / 0. Laki-laki) [Enter untuk batal]: ")
21    if gender_in.strip() == "":
22        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
23        return
24    try:
25        gender_input = int(gender_in)
26        gender = True if gender_input == 1 else False
27    except ValueError:
28        print(Fore.RED + "Gender harus angka 1 atau 0." + Style.RESET_ALL)
29        return
30
31    status_in = input("Status (1. Aktif / 0. Nonaktif) [Enter untuk batal]: ")
32    if status_in.strip() == "":
33        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
34        return
35    try:
36        status_input = int(status_in)
37        status = True if status_input == 1 else False
38    except ValueError:
39        print(Fore.RED + "Status harus angka 1 atau 0." + Style.RESET_ALL)
40        return
41
42    tgl_lahir = input("Tanggal Lahir (YYYY-MM-DD) [Enter untuk batal]: ")
43    if tgl_lahir.strip() == "":
44        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
45        return
46
47    no_telp = input("No Telepon [Enter untuk batal]: ")
48    if no_telp.strip() == "":
49        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
50        return
51
52    id_jabatan_in = input("ID Jabatan (2. Admin Produksi / 3. Admin Gudang) [Enter untuk batal]: ")
53    if id_jabatan_in.strip() == "":
54        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
55        return
56    try:
57        id_jabatan = int(id_jabatan_in)
58    except ValueError:
59        print(Fore.RED + "ID Jabatan harus angka." + Style.RESET_ALL)
60        return
61
62    gaji_in = input("Gaji [Enter untuk batal]: ")
63    if gaji_in.strip() == "":
64        print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
65        return
66    try:
67        gaji = int(gaji_in)
68    except ValueError:
69        print(Fore.RED + "Gaji harus angka." + Style.RESET_ALL)
70        return
71
72    print("\n--- Konfirmasi Data ---")
73    print(f"Nama      : {nama}")
74    print(f"Username   : {username}")
75    print(f"Password   : {sandi}")
76    print(f"Gender     : {'Perempuan' if gender else 'Laki-laki'}")
77    print(f"Status     : {'Aktif' if status else 'Nonaktif'}")
78    print(f"Tgl Lahir  : {tgl_lahir}")
79    print(f"No Telepon : {no_telp}")
80    print(f"ID Jabatan : {id_jabatan}")
81    print(f"Gaji       : {gaji}")
82    konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
83    if konfirmasi != "y":
84        print(Fore.YELLOW + "Input karyawan dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
85        return
86
87    conn, cur = connectDB()
88    if not conn:
89        prompt_enter()
90        return

```

Kode program tersebut berfungsi untuk menambah data karyawan ke database, ambil input dari user seperti nama, username, gender, status, dan gaji, lalu simpan datanya lewat query INSERT.

### 3.25.4 Edit Karyawan

#### Kode Program Edit Karyawan

```
1 def edit_karyawan():
2     conn, cur = connectDB()
3     if not conn: return
4     id_karyawan = input("Masukkan ID Karyawan: ")
5     cur.execute("SELECT * FROM karyawan WHERE id_karyawan = %s", (id_karyawan,))
6     result = cur.fetchone()
7     if result:
8         print("Isi data baru (isi kembali jika tidak ingin mengubah):")
9         nama_baru = input(f>Nama ({result[1]}): ") or result[1]
10        username_baru = input(f"Username ({result[2]}): ") or result[2]
11        sandi_baru = input(f>Password ({result[3]}): ") or result[3]
12        gender_baru = int(input(f"Gender ({result[4]}) (1. Perempuan / 0. Laki-laki): "))
13        gender = True if gender_baru == 1 else False
14        status_baru = int(input(f>Status ({result[8]}) (1. Aktif / 0. Nonaktif): "))
15        status = True if status_baru == 1 else False
16        gaji_baru = int(input(f"Gaji ({result[9]}): ") or result[9])
17        jabatan_baru = input(f>ID Jabatan ({result[10]}) (2. Admin Produksi/ 3. Admin Gudang): ") or result[10]
18
19        cur.execute("""
20        UPDATE karyawan
21        SET nama_karyawan = %s, username_karyawan = %s, kata_sandi_karyawan = %s, gender = %s, status_karyawan = %s,
22        gaji_karyawan = %s, id_jabatan = %s
23        WHERE id_karyawan = %s
24        """, (nama_baru, username_baru, sandi_baru, gender, status, gaji_baru, jabatan_baru, id_karyawan))
25
26        conn.commit()
27        print("Data karyawan berhasil diperbarui.")
28        lihat_karyawan()
29    else:
30        print("Karyawan tidak ditemukan.")
31    cur.close()
32    conn.close()
```

Kode program tersebut berguna untuk edit data karyawan di database berdasarkan ID, user bisa ubah nama, username, dan password, lalu sistem simpan perubahan kalau data ditemukan.

### 3.25.5 Hapus Karyawan

#### Kode Program Edit Karyawan

```

1 def hapus_karyawan():
2     conn, cur = connectDB()
3     if not conn: return
4     id_karyawan = input("Masukkan ID Karyawan: ")
5     cur.execute("DELETE FROM karyawan WHERE id_karyawan = %s", (id_karyawan,))
6     conn.commit()
7     if cur.rowcount > 0:
8         print("Karyawan berhasil dihapus.")
9     else:
10        print("Karyawan tidak ditemukan.")
11    cur.close(); conn.close()

```

Kode program tersebut dibuat untuk menghapus data karyawan dari database berdasarkan ID, kalau data ditemukan sistem jalankan query DELETE dan menampilkan pesan berhasil, kalau tidak ditemukan muncul pesan error.

## 3.29 Fitur Kelola Transaksi Pelanggan

### 3.26.1 Menu Utama Kelola Transaksi Pelanggan

#### Kode Program Menu Utama Kelola Transaksi Pelanggan

```

1 def kelola_transaksi_pelanggan():
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "=== Kelola Transaksi Pelanggan ===" + Style.RESET_ALL)
6         pilihan_transaksi = questionnaire.select(
7             "Pilih menu:",
8             choices=[
9                 "Lihat Laporan Transaksi",
10                "Lihat Riwayat Transaksi",
11                "Kembali"
12            ]
13        ).ask()
14        if pilihan_transaksi == "Lihat Laporan Transaksi":
15            lihat_transaksi()
16        elif pilihan_transaksi == "Lihat Riwayat Transaksi":
17            lihat_riwayat_transaksi()
18        elif pilihan_transaksi == "Kembali":
19            return
20        else:
21            print(Fore.RED + "Pilihan tidak valid, coba lagi." + Style.RESET_ALL)
22
23    prompt_enter()

```

Kode program tersebut dibuat untuk menampilkan menu kelola transaksi pelanggan, user bisa memilih untuk melihat transaksi, melihat riwayat, atau kembali, dan sistem akan menjalankan fungsi sesuai pilihan.

### 3.26.2 Lihat Transaksi

#### Kode Program Lihat Transaksi

```
1 def lihat_transaksi():
2     clear_screen()
3     print("=== Lihat Transaksi ===")
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        query = """
12            SELECT
13                t.id_transaksi, t.tgl_transaksi, jp.nama_jenis, dt.quantity, jp.harga_produk,
14                (dt.quantity * jp.harga_produk) AS subtotal, st.nama_status
15            FROM transaksi t
16            JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
17            JOIN produk p ON dt.id_produk = p.id_produk
18            JOIN jenis_produk jp ON p.id_jenis_produk = jp.id_jenis_produk
19            JOIN status_transaksi st ON t.id_status_transaksi = st.id_status_transaksi
20        """
21        cur.execute(query)
22        result = cur.fetchall()
23
24        if result:
25            print(Fore.GREEN + "\n=== Daftar Transaksi ===" + Style.RESET_ALL)
26            headers = ['ID Transaksi', 'Tanggal', 'Jenis Produk',
27                        'Jumlah', 'Harga Satuan', 'Subtotal', 'Status']
28            print(tabulate(result, headers=headers, tablefmt="fancy_grid"))
29        else:
30            print(Fore.YELLOW + "Transaksi tidak ditemukan." + Style.RESET_ALL)
31
32    except Exception as e:
33        print(Fore.RED + "Gagal menampilkan transaksi:", e, Style.RESET_ALL)
34    finally:
35        try:
36            cur.close()
37            conn.close()
38        except:
39            pass
40
41    prompt_enter()
```

Kode program tersebut dibuat untuk menampilkan semua data transaksi dari tabel database, jika terdapat data yang diinputkan maka akan ditampilkan dalam bentuk tabel, jika kosong maka akan muncul pesan tidak ditemukan.

### 3.26.3 Lihat Riwayat Transaksi

#### Kode Program Menu Utama Kelola Transaksi Pelanggan

```

1 def lihat_riwayat_transaksi():
2     clear_screen()
3     print("=== Lihat Riwayat Transaksi ===")
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        query = """
12        SELECT
13            t.id_transaksi, t.tgl_transaksi, jp.nama_jenis AS jenis_produk, dt.quantity, jp.harga_produk,
14            (dt.quantity * jp.harga_produk) AS subtotal, pl.nama_pelanggan, mp.nama_metode_pembayaran, st.nama_status
15        FROM transaksi t
16        JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
17        JOIN produk p ON dt.id_produk = p.id_produk
18        JOIN jenis_produk jp ON p.id_jenis_produk = jp.id_jenis_produk
19        JOIN status_transaksi st ON t.id_status_transaksi = st.id_status_transaksi
20        JOIN pelanggan pl ON t.id_pelanggan = pl.id_pelanggan
21        JOIN metode_pembayaran mp ON t.id_metode_pembayaran = mp.id_metode_pembayaran
22        WHERE st.nama_status = 'selesai'
23        ORDER BY t.tgl_transaksi
24        """
25        cur.execute(query)
26        data = cur.fetchall()
27
28        if data:
29            print(Fore.GREEN + "\n=== Daftar Riwayat Transaksi ===" + Style.RESET_ALL)
30            headers = ["ID Transaksi", "Tanggal", "Jenis Produk", "Jumlah",
31                       "Harga Satuan", "Subtotal", "Nama Pelanggan",
32                       "Metode Pembayaran", "Status"]
33            print(tabulate(data, headers=headers, tablefmt="fancy_grid"))
34        else:
35            print(Fore.YELLOW + "Belum ada riwayat transaksi dengan status 'selesai'." + Style.RESET_ALL)
36
37    except Exception as e:
38        print(Fore.RED + "Gagal menampilkan riwayat transaksi:", e, Style.RESET_ALL)
39    finally:
40        try:
41            cur.close()
42            conn.close()
43        except:
44            pass
45
46    prompt_enter()

```

Kode program tersebut untuk menampilkan riwayat transaksi yang statusnya selesai, kemudian mengambil data dari beberapa tabel dan ditampilkan dalam bentuk tabel lengkap dan rapi.

### 3.30 Fitur Pergantian Manager Baru

#### Kode Program Pergantian Manager Baru

```

1 def ganti_mer_baru():
2     clear_screen()
3     print("=== Ganti Manager Baru ===")
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        username_karyawan = input("Masukkan username manager lama [Enter untuk batal]: ")
12        if username_karyawan.strip() == "":
13            print(Fore.YELLOW + "Dibatalakan, kembali ke menu utama." + Style.RESET_ALL)
14            return
15
16        query = "SELECT * FROM karyawan WHERE username_karyawan = %s"
17        cur.execute(query, (username_karyawan,))
18        result = cur.fetchone()
19
20        if result:
21            print("Masukkan data manager baru (Enter untuk batal):")
22
23            nama_baru = input("Nama Manager Baru [Enter untuk batal]: ")
24            if nama_baru.strip() == "":
25                print(Fore.YELLOW + "Dibatalakan." + Style.RESET_ALL)
26                return
27
28            gender_in = input("Gender (1. Perempuan / 0. Laki-laki) [Enter untuk batal]: ")
29            if gender_in.strip() == "":
30                print(Fore.YELLOW + "Dibatalakan." + Style.RESET_ALL)
31                return
32            try:
33                gender = int(gender_in)
34                gender_baru = True if gender == 1 else False
35            except ValueError:
36                print(Fore.RED + "Gender harus angka 1 atau 0." + Style.RESET_ALL)
37                return
38
39            tgl_lahir_baru = input("Tgl Lahir (YYYY-MM-DD) [Enter untuk batal]: ")
40            if tgl_lahir_baru.strip() == "":
41                print(Fore.YELLOW + "Dibatalakan." + Style.RESET_ALL)
42                return
43
44            no_telp_baru = input("No Telepon [Enter untuk batal]: ")
45            if no_telp_baru.strip() == "":
46                print(Fore.YELLOW + "Dibatalakan." + Style.RESET_ALL)
47                return
48
49            tgl_masuk_baru = datetime.now().date()
50
51            print("\n--- Konfirmasi Data ---")
52            print(f"Nama Manager Baru : {nama_baru}")
53            print(f"Gender          : {'Perempuan' if gender_baru else 'Laki-laki'}")
54            print(f"Tgl Lahir         : {tgl_lahir_baru}")
55            print(f"No Telepon        : {no_telp_baru}")
56            print(f"Tgl Masuk         : {tgl_masuk_baru}")
57            konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
58            if konfirmasi != "y":
59                print(Fore.YELLOW + "Update dibatalakan." + Style.RESET_ALL)
60                return
61
62            update_query = """
63            UPDATE karyawan
64            SET nama_karyawan = %s, gender = %s, tgl_lahir = %s, no_telp = %s, tgl_masuk = %s
65            WHERE username_karyawan = %s
66            """
67            cur.execute(update_query, (nama_baru, gender_baru, tgl_lahir_baru, no_telp_baru, tgl_masuk_baru, username_karyawan))
68            conn.commit()
69            print(Fore.GREEN + "Manager baru berhasil diganti. Tanggal masuk otomatis diisi hari ini." + Style.RESET_ALL)
70        else:
71            print(Fore.YELLOW + "Data manager tidak ditemukan." + Style.RESET_ALL)
72
73    except Exception as e:
74        print(Fore.RED + "Gagal ganti manager:", e, Style.RESET_ALL)
75        conn.rollback()
76    finally:
77        try:
78            cur.close()
79            conn.close()
80        except:
81            pass

```

Kode program tersebut dibuat ketika terjadi pergantian manager berdasarkan username, user isi data baru dan sistem akan mengupdate ke database kalau datanya ditemukan.

### 3.31 Fitur Halaman Menu Pelanggan

#### Kode Program Halaman Menu Pelanggan

```
1 def halaman_menu_pelanggan(id_pelanggan):
2     while True:
3         clear_screen()
4         banner()
5         print(Fore.CYAN + "=== AGROMAME: Menu Utama Pembelian ===" + Style.RESET_ALL)
6
7         pilihan = questionnaire.select(
8             "Pilih menu:",
9             choices=[
10                 "Biodata",
11                 "Belanja Produk",
12                 "Lihat Riwayat Transaksi Pembelian",
13                 "Logout"
14             ]
15         ).ask()
16
17         if pilihan == "Biodata":
18             biodata_pelanggan(id_pelanggan)
19         elif pilihan == "Belanja Produk":
20             belanja_produk(id_pelanggan)
21         elif pilihan == "Lihat Riwayat Transaksi Pembelian":
22             lihat_riwayat_transaksi_pembelian(id_pelanggan)
23         elif pilihan == "Logout":
24             print("Logout berhasil.")
25             break
26         else:
27             print(Fore.RED + "Pilihan tidak valid!" + Style.RESET_ALL)
28
29     prompt_enter()
```

Kode program tersebut digunakan untuk menampilkan menu pelanggan, user bisa memilih fitur seperti lihat biodata, produk, keranjang, metode pembayaran, atau riwayat transaksi, dan sistem akan menjalankan fungsi sesuai inputnya.

### 3.31 Fitur Lihat Biodata (Pelanggan)

**Kode Program Lihat Biodata (Pelanggan)**

```

1 def lihat_biodata(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Lihat Biodata Pelanggan ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        query = """
12        SELECT p.nama_pelanggan, p.username_pelanggan, p.kata_sandi_pelanggan, p.no_telp_pelanggan,
13              a.nama_jalan, d.nama_desa, k.nama_kecamatan, kb.nama_kabupaten
14        FROM pelanggan p
15        LEFT JOIN alamat a ON p.id_alamat = a.id_alamat
16        LEFT JOIN desa d ON a.id_desa = d.id_desa
17        LEFT JOIN kecamatan k ON d.id_kecamatan = k.id_kecamatan
18        LEFT JOIN kabupaten kb ON k.id_kabupaten = kb.id_kabupaten
19        WHERE p.id_pelanggan = %s
20        """
21        cur.execute(query, (id_pelanggan,))
22        data = cur.fetchone()
23
24        if data:
25            print(Fore.GREEN + "=== BIODATA ===" + Style.RESET_ALL)
26            if data[4]:
27                headers = ["Nama", "Username", "Kata Sandi", "No Telp", "Jalan", "Desa", "Kecamatan", "Kabupaten"]
28                print(tabulate([data], headers=headers, tablefmt="fancy_grid"))
29            else:
30                headers = ["Nama", "Username", "Kata Sandi", "No Telp", "Alamat"]
31                row = [data[0], data[1], data[2], data[3], "Belum diisi"]
32                print(tabulate([row], headers=headers, tablefmt="fancy_grid"))
33        else:
34            print(Fore.YELLOW + "Data biodata pelanggan tidak ditemukan." + Style.RESET_ALL)
35
36    except Exception as e:
37        print(Fore.RED + "Gagal menampilkan biodata:", e, Style.RESET_ALL)
38    finally:
39        try:
40            cur.close()
41            conn.close()
42        except:
43            pass
44
45    prompt_enter()

```

Kode program tersebut dibuat untuk menampilkan biodata pelanggan dari database berdasarkan ID, gabungkan data alamat lengkap dan tampilkan dalam tabel, jika data tidak ada maka akan muncul pesan “Data tidak ditemukan”.

### 3.32 Fitur Edit Biodata (Pelanggan)

#### Code Program Edit Biodata (Pelanggan)

```

1 def edit_biodata(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Edit Biodata ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        query = """
12            SELECT p.nama_pelanggan, p.username_pelanggan, p.kata_sandi_pelanggan,
13                   p.no_telp_pelanggan, a.id_alamat, a.nama_jalan, d.id_desa, d.nama_desa
14            FROM pelanggan p
15            LEFT JOIN alamat a ON p.id_alamat = a.id_alamat
16            LEFT JOIN desa d ON a.id_desa = d.id_desa
17            WHERE p.id_pelanggan = %s
18        """
19        cur.execute(query, (id_pelanggan,))
20        data = cur.fetchone()
21
22        if data:
23            print("Isi data baru (Enter untuk tidak mengubah / batal jika kosong di awal):")
24
25            nama = input(f>Nama ({data[0]}): ") or data[0]
26            user = input(f>Username ({data[1]}): ") or data[1]
27            sandi = input(f>Kata Sandi ({data[2]}): ") or data[2]
28            telp = input(f>No Telp ({data[3]}): ") or data[3]
29
30            if data[4]:
31                jalan = input(f>Nama Jalan ({data[5]}): ") or data[5]
32                desa = input(f>ID Desa ({data[6]} - {data[7]}): ") or data[6]
33
34                cur.execute("""
35                    UPDATE alamat SET nama_jalan = %s, id_desa = %s
36                    WHERE id_alamat = %s
37                """, (jalan, desa, data[4]))
38            else:
39                print("Alamat belum ada, silakan isi:")
40                jalan = input("Nama Jalan [Enter untuk batal]: ")
41                if jalan.strip() == "":
42                    print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
43                    return
44                desa = input("ID Desa [Enter untuk batal]: ")
45                if desa.strip() == "":
46                    print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
47                    return
48
49                cur.execute("""
50                    INSERT INTO alamat (nama_jalan, id_desa)
51                    VALUES (%s, %s) RETURNING id_alamat
52                """, (jalan, desa))
53                id_alamat_baru = cur.fetchone()[0]
54
55                cur.execute("""
56                    UPDATE pelanggan SET id_alamat = %s
57                    WHERE id_pelanggan = %s
58                """, (id_alamat_baru, id_pelanggan))
59
60            print("\n--- Konfirmasi Data ---")
61            print(f>Nama      : {nama}")
62            print(f>Username : {user}")
63            print(f>Kata Sandi: {sandi}")
64            print(f>No Telp  : {telp}")
65            konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
66            if konfirmasi != "y":
67                print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
68                return
69
70            cur.execute("""
71                UPDATE pelanggan
72                SET nama_pelanggan = %s,
73                    username_pelanggan = %s,
74                    kata_sandi_pelanggan = %s,
75                    no_telp_pelanggan = %s
76                WHERE id_pelanggan = %s
77            """, (nama, user, sandi, telp, id_pelanggan))
78
79            conn.commit()
80            print(Fore.GREEN + "Biodata berhasil diperbarui." + Style.RESET_ALL)
81        else:
82            print(Fore.YELLOW + "Data tidak ditemukan." + Style.RESET_ALL)
83
84    except Exception as e:
85        print(Fore.RED + "Gagal edit biodata:", e, Style.RESET_ALL)
86        conn.rollback()
87    finally:
88        try:
89            cur.close()
90            conn.close()
91        except:
92            pass
93
94    prompt_enter()

```

Kode program tersebut dibuat untuk mengedit biodata pelanggan, sistem akan mengambil data lama dari database, kemudian user mengisi data baru dan sistem akan mengupdate data tersebut termasuk alamat jika belum terisi.

### 3.33 Fitur Lihat Daftar Produk

#### Kode Program Lihat Daftar Produk

```
1 def lihat_daftar_produk():
2     print("== Lihat Daftar Produk ==")
3     conn, cur = connectDB()
4
5     query = """
6     SELECT j.nama_jenis, p.stok, j.harga_produk, p.tgl_update_stok
7     FROM produk p
8     JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
9     ORDER BY j.nama_jenis
10    """
11    cur.execute(query)
12    data = cur.fetchall()
13
14    if data:
15        headers = ["Nama Jenis", "Stok", "Harga Produk (Rp)", "Update Stok"]
16        print(tabulate(data, headers=headers, tablefmt="fancy_grid"))
17    else:
18        print("Belum ada produk tersedia.")
19
20    cur.close(); conn.close()
21
```

Kode program tersebut dibuat untuk menampilkan daftar produk dari database, menggabungkan data jenis produk dan menampilkan data dalam tabel rapi, jika kosong maka akan muncul pesan “Belum ada produk tersedia”.

### 3.34 Fitur Pilih Produk

#### Kode Program Pilih Produk

```

1 def pilih_produk(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Pilih Produk ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'proses'")
12        s = cur.fetchone()
13        if not s:
14            print(Fore.RED + "Status 'proses' tidak ada di status_transaksi." + Style.RESET_ALL)
15            return
16        id_status = s[0]
17
18        cur.execute("""
19            SELECT t.id_transaksi
20            FROM transaksi t
21            WHERE t.id_pelanggan = %s AND t.id_status_transaksi = %s
22            ORDER BY t.id_transaksi DESC
23            LIMIT 1
24            """, (id_pelanggan, id_status))
25        t = cur.fetchone()
26
27        if t:
28            id_transaksi = t[0]
29        else:
30            cur.execute("""
31                INSERT INTO transaksi (id_pelanggan, id_status_transaksi, tgl_transaksi)
32                VALUES (%s, %s, CURRENT_DATE) RETURNING id_transaksi
33                """, (id_pelanggan, id_status))
34            id_transaksi = cur.fetchone()[0]
35
36        while True:
37            cur.execute("""
38                SELECT p.id_produk, j.nama_jenis, p.stok, j.harga_produk, p.tgl_update_stok
39                FROM produk p
40                JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
41                ORDER BY p.id_produk
42                """)
43            daftar_produk = cur.fetchall()
44
45            if not daftar_produk:
46                print(Fore.YELLOW + "Belum ada produk tersedia." + Style.RESET_ALL)
47                break
48
49            headers = ["ID Produk", "Jenis Produk", "Stok", "Harga (Rp)", "Update Stok"]
50            print(tabulate(daftar_produk, headers=headers, tablefmt="fancy_grid"))
51
52            id_produk_in = input("Masukkan ID Produk [Enter untuk batal]: ")
53            if id_produk_in.strip() == "":
54                print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
55                break
56            qty_in = input("Masukkan jumlah [Enter untuk batal]: ")
57            if qty_in.strip() == "":
58                print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
59                break
60
61            try:
62                id_produk = int(id_produk_in)
63                quantity = int(qty_in)
64            except ValueError:
65                print(Fore.RED + "Input harus angka!" + Style.RESET_ALL)
66                continue
67
68            cur.execute("SELECT stok FROM produk WHERE id_produk = %s", (id_produk,))
69            row = cur.fetchone()
70            if not row:
71                print(Fore.RED + "Produk tidak ditemukan." + Style.RESET_ALL)
72                continue
73
74            stok_tersedia = row[0]
75            if quantity > stok_tersedia:
76                print(Fore.RED + f"Stok tidak cukup. Stok tersedia: {stok_tersedia}" + Style.RESET_ALL)
77                continue

```

```

1
2     cur.execute("""
3         SELECT j.harga_produk
4         FROM produk p
5         JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
6         WHERE p.id_produk = %s
7     """, (id_produk,))
8     row = cur.fetchone()
9     if not row:
10        print(Fore.RED + "Harga produk tidak ditemukan." + Style.RESET_ALL)
11        continue
12     harga_satuan = row[0]
13     total_harga = harga_satuan * quantity
14
15     print("\n--- Konfirmasi Produk ---")
16     print(f"ID Produk : {id_produk}")
17     print(f"Jumlah : {quantity}")
18     print(f"Harga Satuan: {harga_satuan}")
19     print(f"Total Harga : {total_harga}")
20     konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
21     if konfirmasi != "y":
22        print(Fore.YELLOW + "Produk batal ditambahkan." + Style.RESET_ALL)
23        continue
24
25     cur.execute("""
26         INSERT INTO detail_transaksi (id_transaksi, id_produk, quantity, harga)
27         VALUES (%s, %s, %s, %s)
28     """, (id_transaksi, id_produk, quantity, total_harga))
29
30     cur.execute("""
31         UPDATE produk
32         SET stok = stok - %s, tgl_update_stok = CURRENT_DATE
33         WHERE id_produk = %s
34     """, (quantity, id_produk))
35
36     conn.commit()
37     print(Fore.GREEN + f"Produk {id_produk} x{quantity} berhasil ditambahkan ke keranjang." + Style.RESET_ALL)
38
39     lanjut = input("Tambah produk lain? (y/n): ").strip().lower()
40     if lanjut != "y":
41        break
42
43     print(Fore.GREEN + "Selesai memilih produk. Semua pilihan sudah masuk ke keranjang." + Style.RESET_ALL)
44
45 except Exception as e:
46     print(Fore.RED + "Gagal memilih produk:", e, Style.RESET_ALL)
47     conn.rollback()
48 finally:
49     try:
50         cur.close()
51         conn.close()
52     except:
53         pass
54     prompt_enter()
55

```

Kode program tersebut dibuat untuk proses transaksi pembelian produk, user memilih produk dan jumlah yg diinginkan, sistem cek stok, hitung total, sistem menyimpan detail ke database, lalu mengurangi stok dan menutup transaksi.

### 3.35 Fitur Belanja Produk

#### Kode Program Belanja Produk

```

1  def belanja_produk(id_pelanggan):
2      while True:
3          clear_screen()
4          banner()
5          print(Fore.CYAN + "=== AGROMAME: Menu Belanja Produk ===" + Style.RESET_ALL)
6
7          pilihan = questionnaire.select(
8              "Pilih menu:",
9              choices=[
10                 "Pilih Produk",
11                 "Lihat Keranjang",
12                 "Edit Keranjang",
13                 "Pilih Metode Pembayaran",
14                 "Edit Metode Pembayaran",
15                 "Checkout",
16                 "Kembali"
17             ]
18         ).ask()
19
20         if pilihan == "Pilih Produk":
21             pilih_produk(id_pelanggan)
22         elif pilihan == "Lihat Keranjang":
23             lihat_keranjang(id_pelanggan)
24         elif pilihan == "Edit Keranjang":
25             edit_keranjang(id_pelanggan)
26         elif pilihan == "Pilih Metode Pembayaran":
27             pilih_metode_pembayaran(id_pelanggan)
28         elif pilihan == "Edit Metode Pembayaran":
29             edit_metode_pembayaran(id_pelanggan)
30         elif pilihan == "Checkout":
31             checkout(id_pelanggan)
32         elif pilihan == "Kembali":
33             break
34         else:
35             print(Fore.RED + "Pilihan tidak valid!" + Style.RESET_ALL)
36             prompt_enter()
37

```

### 3.36 Fitur Lihat Keranjang

Kode Program Lihat Keranjang

```

1 def lihat_keranjang(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Lihat Keranjang ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'proses'")
12        s = cur.fetchone()
13        if not s:
14            print(Fore.RED + "Status 'proses' tidak ada di status_transaksi." + Style.RESET_ALL)
15            return
16        id_status = s[0]
17
18        cur.execute("""
19            SELECT id_transaksi, id_metode_pembayaran
20            FROM transaksi
21            WHERE id_pelanggan = %s AND id_status_transaksi = %s
22            ORDER BY id_transaksi DESC
23            LIMIT 1
24            """, (id_pelanggan, id_status))
25        t = cur.fetchone()
26
27        if not t:
28            print(Fore.YELLOW + "Keranjang kosong. Belum ada transaksi aktif." + Style.RESET_ALL)
29            return
30
31        id_transaksi, id_metode = t
32
33        cur.execute("""
34            SELECT j.nama_jenis, d.quantity, d.harga
35            FROM detail_transaksi d
36            JOIN produk p ON d.id_produk = p.id_produk
37            JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
38            WHERE d.id_transaksi = %s
39            """, (id_transaksi,))
40        items = cur.fetchall()
41
42        if not items:
43            print(Fore.YELLOW + "Keranjang masih kosong." + Style.RESET_ALL)
44        else:
45            print(Fore.GREEN + "\n=== Isi Keranjang ===" + Style.RESET_ALL)
46            headers = ["Produk", "Jumlah", "Subtotal (Rp)"]
47            print(tabulate(items, headers=headers, tablefmt="fancy_grid"))
48
49            total = sum([row[2] for row in items])
50            print(Fore.GREEN + f"\nTotal Keranjang: Rp {total}" + Style.RESET_ALL)
51
52            if id_metode:
53                cur.execute("""
54                    SELECT id_metode_pembayaran, nama_transaksi
55                    FROM metode_pembayaran
56                    WHERE id_metode_pembayaran = %s
57                    """, (id_metode,))
58                metode = cur.fetchone()
59                if metode:
60                    print(Fore.GREEN + f"\nMetode Pembayaran Aktif: {metode[1]} (ID: {metode[0]})" + Style.RESET_ALL)
61                else:
62                    print(Fore.YELLOW + "\nMetode pembayaran belum dipilih." + Style.RESET_ALL)
63            else:
64                print(Fore.YELLOW + "\nMetode pembayaran belum dipilih." + Style.RESET_ALL)
65
66        except Exception as e:
67            print(Fore.RED + "Gagal menampilkan keranjang:", e, Style.RESET_ALL)
68        finally:
69            try:
70                cur.close()
71                conn.close()
72            except:
73                pass
74
75        prompt_enter()

```

Kode program tersebut dibuat untuk menampilkan isi keranjang pelanggan dari transaksi yang statusnya masih proses, jika belum terdapat transaksi aktif maka muncul pesan “Keranjang kosong”.

## 3.37 Fitur Edit Keranjang

### Kode Program Edit Keranjang

```
1 def edit_keranjang(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Edit Keranjang ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        cur.execute("""
12            SELECT t.id_transaksi
13            FROM transaksi t
14            JOIN status_transaksi s ON t.id_status_transaksi = s.id_status_transaksi
15            WHERE t.id_pelanggan = %s AND s.nama_status = 'proses'
16            ORDER BY t.id_transaksi DESC
17            LIMIT 1
18            """, (id_pelanggan,))
19        t = cur.fetchone()
20
21        if not t:
22            print(Fore.YELLOW + "Keranjang kosong. Belum ada transaksi aktif." + Style.RESET_ALL)
23            return
24
25        id_transaksi = t[0]
26
27        cur.execute("""
28            SELECT d.id_detail_transaksi, j.nama_jenis, d.quantity, d.harga
29            FROM detail_transaksi d
30            JOIN produk p ON d.id_produk = p.id_produk
31            JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
32            WHERE d.id_transaksi = %s
33            """, (id_transaksi,))
34        items = cur.fetchall()
35
36        if not items:
37            print(Fore.YELLOW + "Keranjang masih kosong." + Style.RESET_ALL)
38            return
39
40        headers = ["ID Detail", "Produk", "Jumlah", "Subtotal (Rp)"]
41        print(tabulate(items, headers=headers, tablefmt="fancy_grid"))
42
43        id_detail_in = input("Masukkan ID Detail yang ingin diubah/hapus [Enter untuk batal]: ")
44        if id_detail_in.strip() == "":
45            print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
46            return
47
48        try:
49            id_detail = int(id_detail_in)
50        except ValueError:
51            print(Fore.RED + "ID Detail harus angka." + Style.RESET_ALL)
52            return
53
54        pilihan = input("Ketik 'ubah' untuk ubah jumlah, 'hapus' untuk hapus produk [Enter untuk batal]: ").lower()
55        if pilihan.strip() == "":
56            print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
57            return
58
59        if pilihan == "ubah":
60            new_qty_in = input("Masukkan jumlah baru [Enter untuk batal]: ")
61            if new_qty_in.strip() == "":
62                print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
63                return
64
65            try:
66                new_qty = int(new_qty_in)
67            except ValueError:
68                print(Fore.RED + "Jumlah harus angka." + Style.RESET_ALL)
69                return
```

```

1
2     cur.execute("""
3         SELECT j.harga_produk
4         FROM detail_transaksi d
5         JOIN produk p ON d.id_produk = p.id_produk
6         JOIN jenis_produk j ON p.id_jenis_produk = j.id_jenis_produk
7         WHERE d.id_detail_transaksi = %s
8     """, (id_detail,))
9     row = cur.fetchone()
10    if not row:
11        print(Fore.RED + "Produk tidak ditemukan." + Style.RESET_ALL)
12        return
13
14    harga_satuan = row[0]
15    total_harga = harga_satuan * new_qty
16
17    print("\n--- Konfirmasi Ubah ---")
18    print(f"ID Detail : {id_detail}")
19    print(f"Jumlah Baru: {new_qty}")
20    print(f"Subtotal : {total_harga}")
21    konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
22    if konfirmasi != "y":
23        print(Fore.YELLOW + "Edit dibatalkan." + Style.RESET_ALL)
24        return
25
26    cur.execute("""
27        UPDATE detail_transaksi
28        SET quantity = %s, harga = %s
29        WHERE id_detail_transaksi = %s
30    """, (new_qty, total_harga, id_detail))
31    conn.commit()
32    print(Fore.GREEN + "Jumlah produk berhasil diubah." + Style.RESET_ALL)
33
34    elif pilihan == "hapus":
35        print("\n--- Konfirmasi Hapus ---")
36        print(f"ID Detail : {id_detail}")
37        konfirmasi = input("Apakah yakin ingin menghapus produk ini? (y/n): ").strip().lower()
38        if konfirmasi != "y":
39            print(Fore.YELLOW + "Hapus dibatalkan." + Style.RESET_ALL)
40            return
41
42        cur.execute("DELETE FROM detail_transaksi WHERE id_detail_transaksi = %s", (id_detail,))
43        conn.commit()
44        print(Fore.GREEN + "Produk berhasil dihapus dari keranjang." + Style.RESET_ALL)
45
46    else:
47        print(Fore.RED + "Pilihan tidak valid." + Style.RESET_ALL)
48
49    except Exception as e:
50        print(Fore.RED + "Gagal edit keranjang:", e, Style.RESET_ALL)
51        conn.rollback()
52    finally:
53        try:
54            cur.close()
55            conn.close()
56        except:
57            pass
58
59    prompt_enter()

```

Kode program tersebut dibuat untuk mengedit isi keranjang pelanggan dengan mengambil transaksi aktif, cek item yang sudah dipilih, lalu tampilkan dalam bentuk tabel jikalau ada.

### 3.38 Fitur Pilih Metode Pembayaran

Kode Program Pilih Metode Pembayaran

```

1 def pilih_metode_pembayaran(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Pilih Metode Pembayaran ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'proses'")
12        s = cur.fetchone()
13        if not s:
14            print(Fore.RED + "Status 'proses' tidak ditemukan." + Style.RESET_ALL)
15            return
16        id_status = s[0]
17
18        cur.execute("""
19            SELECT id_transaksi
20            FROM transaksi
21            WHERE id_pelanggan = %s AND id_status_transaksi = %s
22            ORDER BY id_transaksi DESC
23            LIMIT 1
24            """, (id_pelanggan, id_status))
25        t = cur.fetchone()
26
27        if not t:
28            print(Fore.YELLOW + "Tidak ada transaksi aktif. Silakan pilih produk dulu." + Style.RESET_ALL)
29            return
30
31        id_transaksi = t[0]
32
33        cur.execute("SELECT id_metode_pembayaran, nama_metode_pembayaran FROM metode_pembayaran")
34        metode = cur.fetchall()
35
36        if not metode:
37            print(Fore.YELLOW + "Belum ada metode pembayaran tersedia." + Style.RESET_ALL)
38            return
39
40        print(Fore.GREEN + "\n=== Daftar Metode Pembayaran ===" + Style.RESET_ALL)
41        for m in metode:
42            print(f"{m[0]}. {m[1]}")
43
44        pilihan_in = input("Masukkan ID Metode Pembayaran [Enter untuk batal]: ")
45        if pilihan_in.strip() == "":
46            print(Fore.YELLOW + "Dibatalkan, kembali ke menu utama." + Style.RESET_ALL)
47            return
48
49        try:
50            pilihan = int(pilihan_in)
51        except ValueError:
52            print(Fore.RED + "Input harus angka!" + Style.RESET_ALL)
53            return
54
55        cur.execute("SELECT id_metode_pembayaran, nama_metode_pembayaran FROM metode_pembayaran WHERE id_metode_pembayaran = %s", (pilihan,))
56        cek = cur.fetchone()
57        if not cek:
58            print(Fore.RED + "Metode pembayaran tidak valid." + Style.RESET_ALL)
59            return
60
61        print("\n--- Konfirmasi Metode Pembayaran ---")
62        print(f"ID : {cek[0]}")
63        print(f>Nama : {cek[1]}")
64        konfirmasi = input("Apakah data sudah benar? (y/n): ").strip().lower()
65        if konfirmasi != "y":
66            print(Fore.YELLOW + "Update dibatalkan." + Style.RESET_ALL)
67            return
68
69        cur.execute("""
70            UPDATE transaksi
71            SET id_metode_pembayaran = %s
72            WHERE id_transaksi = %s
73            """, (pilihan, id_transaksi))
74        conn.commit()
75
76        print(Fore.GREEN + "Metode pembayaran berhasil dipilih." + Style.RESET_ALL)
77
78    except Exception as e:
79        print(Fore.RED + "Gagal memilih metode pembayaran:", e, Style.RESET_ALL)
80        conn.rollback()
81
82    finally:
83        try:
84            cur.close()
85            conn.close()
86        except:
87            pass
88        prompt_enter()

```

Kode program tersebut dibuat untuk user dapat memilih metode pembayaran untuk transaksi aktif pelanggan, sistem cek status, dan menampilkan daftar metode, kemudian inputan dari user akan disimpan ke database.

### 3.39 Fitur Edit Metode Pembayaran

## Kode Program Edit Metode Pembayaran

```
1 def edit_metode_pembayaran(id_pelanggan):
2     print(Fore.CYAN + "=== Edit Metode Pembayaran ===" + Style.RESET_ALL)
3     conn, cur = connectDB()
4     if not conn: return
5
6     cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'proses'")
7     s = cur.fetchone()
8     if not s:
9         print("Status 'proses' tidak ditemukan.")
10        cur.close(); conn.close()
11        return
12    id_status = s[0]
13
14    cur.execute("""
15        SELECT id_transaksi, id_metode_pembayaran
16        FROM transaksi
17        WHERE id_pelanggan = %s AND id_status_transaksi = %s
18        ORDER BY id_transaksi DESC
19        LIMIT 1
20    """, (id_pelanggan, id_status))
21    t = cur.fetchone()
22    if not t:
23        print("Tidak ada transaksi aktif untuk pelanggan ini.")
24        cur.close(); conn.close()
25        return
26    id_transaksi = t[0]
27
28    cur.execute("SELECT id_metode_pembayaran, nama_metode_pembayaran FROM metode_pembayaran ORDER BY id_metode_pembayaran")
29    metode = cur.fetchall()
30    headers = ["ID Metode", "Nama Metode"]
31    print(tabulate(metode, headers=headers, tablefmt="fancy_grid"))
32
33    try:
34        pilihan = int(input("Masukkan ID Metode Pembayaran baru: "))
35    except ValueError:
36        print("Input harus angka!")
37        cur.close(); conn.close()
38        return
39
40    cur.execute("SELECT id_metode_pembayaran FROM metode_pembayaran WHERE id_metode_pembayaran = %s", (pilihan,))
41    cek = cur.fetchone()
42    if not cek:
43        print("Metode pembayaran tidak valid.")
44        cur.close(); conn.close()
45        return
46
47    cur.execute("""
48        UPDATE transaksi
49        SET id_metode_pembayaran = %s
50        WHERE id_transaksi = %s
51    """, (pilihan, id_transaksi))
52    conn.commit()
53    print("Metode pembayaran berhasil diganti untuk transaksi aktif.")
54
55    cur.close(); conn.close()
56    prompt_enter()
```

Kode program tersebut dibuat ketika user ingin mengganti metode pembayaran transaksi aktif, sistem akan menampilkan daftar metode dan mengambil input id metode pembayaran yang baru, lalu update ke dalam database.

## 3.40 Fitur Checkout

### Kode Program Checkout

```

1  def checkout(id_pelanggan):
2      clear_screen()
3      print(Fore.CYAN + "=== Checkout ===" + Style.RESET_ALL)
4
5      conn, cur = connectDB()
6      if not conn:
7          prompt_enter()
8          return
9
10     try:
11         cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'proses'")
12         s = cur.fetchone()
13         if not s:
14             print(Fore.RED + "Status 'proses' tidak ada di status_transaksi." + Style.RESET_ALL)
15             return
16         id_status_proses = s[0]
17
18         cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'selesai'")
19         s = cur.fetchone()
20         if not s:
21             print(Fore.RED + "Status 'selesai' tidak ada di status_transaksi." + Style.RESET_ALL)
22             return
23         id_status_selesai = s[0]
24
25         cur.execute("SELECT id_status_transaksi FROM status_transaksi WHERE nama_status = 'dibatal'")
26         s = cur.fetchone()
27         if not s:
28             print(Fore.RED + "Status 'dibatal' tidak ada di status_transaksi." + Style.RESET_ALL)
29             return
30         id_status_batal = s[0]
31
32         cur.execute("""
33             SELECT id_transaksi FROM transaksi
34             WHERE id_pelanggan = %s AND id_status_transaksi = %s
35             ORDER BY id_transaksi DESC LIMIT 1
36             """, (id_pelanggan, id_status_proses))
37         transaksi = cur.fetchone()
38         if not transaksi:
39             print(Fore.YELLOW + "Tidak ada transaksi aktif." + Style.RESET_ALL)
40             return
41         id_transaksi = transaksi[0]
42
43         cur.execute("""
44             SELECT d.id_produk, d.quantity, jp.harga_produk
45             FROM detail_transaksi d
46             JOIN produk p ON d.id_produk = p.id_produk
47             JOIN jenis_produk jp ON p.id_jenis_produk = jp.id_jenis_produk
48             WHERE d.id_transaksi = %s
49             """, (id_transaksi,))
50         items = cur.fetchall()
51         if not items:
52             print(Fore.YELLOW + "Keranjang kosong." + Style.RESET_ALL)
53             return
54
55         total_belanja = sum(qty * harga for _, qty, harga in items)
56
57         print("\n--- Ringkasan Transaksi ---")
58         print(f"ID Transaksi : {id_transaksi}")
59         print(f"Jumlah Item : {len(items)}")
60         print(f"Total Belanja: Rp {total_belanja}")
61         aksi = input("Lanjut checkout atau dibatalkan? (checkout/dibatal) [Enter untuk batal]: ").strip().lower()
62         if aksi == "":
63             print(Fore.YELLOW + "Checkout dibatalkan." + Style.RESET_ALL)
64             return
65
66         if aksi == "checkout":
67             cur.execute("""
68                 UPDATE transaksi
69                 SET id_status_transaksi = %s, tgl_transaksi = CURRENT_DATE
70                 WHERE id_transaksi = %s
71                 """, (id_status_selesai, id_transaksi))
72             conn.commit()
73             print(Fore.GREEN + "Checkout berhasil!" + Style.RESET_ALL)
74             print(Fore.GREEN + f"Total belanja: Rp {total_belanja}" + Style.RESET_ALL)
75
76         elif aksi == "dibatal":
77             for id_produk, qty, _ in items:
78                 cur.execute("""
79                     UPDATE produk
80                     SET stok = stok - %s, tgl_update_stok = CURRENT_DATE
81                     WHERE id_produk = %s
82                     """, (qty, id_produk))
83
84             cur.execute("""
85                 UPDATE transaksi
86                 SET id_status_transaksi = %s
87                 WHERE id_transaksi = %s
88                 """, (id_status_batal, id_transaksi))
89             conn.commit()
90             print(Fore.GREEN + f"Transaksi {id_transaksi} dibatalkan, stok dikembalikan." + Style.RESET_ALL)
91
92         else:
93             print(Fore.RED + "Pilihan tidak valid." + Style.RESET_ALL)
94
95     except Exception as e:
96         print(Fore.RED + "Gagal checkout:", e, Style.RESET_ALL)
97         conn.rollback()
98     finally:
99         try:
100             cur.close()
101             conn.close()
102         except:
103             pass
104     prompt_enter()

```

Kode program tersebut dibuat untuk proses checkout transaksi dan melihat riwayat pembelian, sistem akan mengubah status menjadi selesai jika isi transaksi valid, lalu menampilkan daftar transaksi beserta detail produknya.

### 3.41 Fitur Lihat Riwayat Transaksi Pembelian

#### Kode Program Lihat Riwayat Transaksi Pembelian

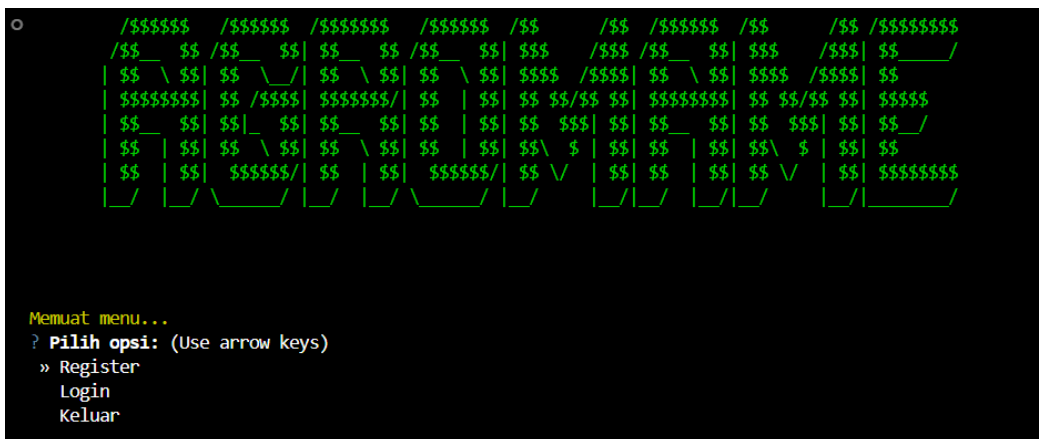
```
1 def lihat_riwayat_transaksi_pembelian(id_pelanggan):
2     clear_screen()
3     print(Fore.CYAN + "=== Riwayat Transaksi Pembelian ===" + Style.RESET_ALL)
4
5     conn, cur = connectDB()
6     if not conn:
7         prompt_enter()
8         return
9
10    try:
11        query = """
12            SELECT t.id_transaksi, t.tgl_transaksi, COALESCE(mp.nama_metode_pembayaran, '-') AS metode_bayar,
13                SUM(dt.quantity * jp.harga_produk) AS total_belanja
14            FROM transaksi t
15            LEFT JOIN metode_pembayaran mp ON t.id_metode_pembayaran = mp.id_metode_pembayaran
16            JOIN detail_transaksi dt ON t.id_transaksi = dt.id_transaksi
17            JOIN produk p ON dt.id_produk = p.id_produk
18            JOIN jenis_produk jp ON p.id_jenis_produk = jp.id_jenis_produk
19            JOIN status_transaksi st ON t.id_status_transaksi = st.id_status_transaksi
20            WHERE t.id_pelanggan = %s AND st.nama_status = 'selesai'
21            GROUP BY t.id_transaksi, t.tgl_transaksi, mp.nama_metode_pembayaran
22            ORDER BY t.tgl_transaksi DESC
23        """
24        cur.execute(query, (id_pelanggan,))
25        riwayat = cur.fetchall()
26
27        if not riwayat:
28            print(Fore.YELLOW + "Belum ada transaksi selesai." + Style.RESET_ALL)
29        else:
30            print(Fore.GREEN + "\n=== Daftar Riwayat Transaksi Pembelian ===" + Style.RESET_ALL)
31            headers = ["ID Transaksi", "Tanggal", "Metode Bayar", "Total Belanja (Rp)"]
32            print(tabulate(riwayat, headers=headers, tablefmt="fancy_grid"))
33
34    except Exception as e:
35        print(Fore.RED + "Gagal menampilkan riwayat transaksi:", e, Style.RESET_ALL)
36    finally:
37        try:
38            cur.close()
39            conn.close()
40        except:
41            pass
42
43    prompt_enter()
44
```

Kode program tersebut dibuat untuk menampilkan riwayat transaksi pembelian pelanggan, sistem akan mengambil data total belanja dan total bayar dari database, lalu menampilkan data dalam bentuk tabel.

## IV. HASIL DAN PEMBAHASAN

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

### 4.1 Fitur Halaman Menu Utama



```
o      /$$$$$ /$$$$$ /$$$$$ /$$$$$ /$ /$ /$$$$$ /$ /$ /$$$$$$
      /$ _ $ /$ _ $ $ _ $ /$ _ $ $ $ /$ /$ _ $ $ $ /$ $ /$ _ $
| $ $ \ $ $ $ \ $ / $ $ \ $ $ \ $ $ $ /$ $ $ \ $ $ $ $ /$ $ $
| $$$$ $ $ /$ $ $$$$ /$ $ | $ $ $ $ /$ $ $$$$ $ $ $ $ /$ $ $
| $ _ $ $ $ | $ _ $ $ _ $ $ $ | $ $ $ $ $ $ | $ _ $ $ $ /
| $ $ | $ $ $ $ \ $ $ $ \ $ $ | $ $ $ $ \ $ | $ $ $ \ $ | $ $ $
| $ $ | $ $ $$$$ / $ $ | $ $ $$$$ / $ $ \ $ | $ $ $ $ \ $ | $ $ $$$$
| / | / \ / | / | / \ / | / | / \ / | / | / \ / | / | / \ /

Memuat menu...
? Pilih opsi: (Use arrow keys)
» Register
  Login
  Keluar
```

Gambar 4.1 Tampilan Fitur Halaman Menu Utama

Pada bagian ini, sistem menampilkan menu awal yang menjadi pintu masuk pengguna ke aplikasi. Menu utama terdiri dari tiga pilihan yaitu *Register*, *Login*, dan *Keluar*. Saat program dijalankan, pengguna akan melihat tampilan tabel rapi yang dibuat dengan pustaka tabulate. Dari sini, pengguna cukup memasukkan angka 1, 2, atau 3 untuk menentukan langkah berikutnya.

Struktur data pada fitur ini sederhana karena berbasis console. Menu ditulis dalam bentuk list berisi nomor dan opsi, lalu pengguna memilih dengan memasukkan angka (integer). Alur program dikendalikan menggunakan perulangan *while* dan kondisi *if-else* untuk mengeksekusi pilihan yang dimasukkan.

Langkah penggunaan fitur ini dimulai dengan menjalankan program sehingga menu utama tampil di layar. Setelah itu, pengguna memilih salah satu opsi dengan mengetik angka sesuai kebutuhan. Jika memilih 1 maka sistem akan memanggil fungsi *registerasi()*, jika memilih 2 maka sistem menjalankan fungsi *login()*, dan jika memilih 3 maka program berhenti dengan pesan penutup. Dengan alur ini, pengguna dapat langsung memahami navigasi awal tanpa kebingungan.

## 4.2 Fitur Registrasi Pelanggan

```

    /$$$$$ /$$$$$ /$$$$$ /$$$$$ /$ /$ /$$$$$ /$ /$ /$$$$$
    /$ _ $ /$ _ $| $ _ $ /$ _ $| $$$ /$$$ /$ _ $| $$$ /$$$| $ _ $/
    | $ $ \ $ $| $ $ \ _/| $ $ \ $ $| $ $ \ $ $| $$$ /$$$| $ $ \ $ $| $$$ /$$$| $ $
    | $$$$$$| $ $ /$$$| $$$$$$/| $ $ | $ $| $ $ $/$ $ $| $$$$$$| $ $ $/$ $ $| $$$$$$
    | $ _ $ $| $ $| $ _ $ $| $ _ $ $| $ $ | $ $| $ $ $$$| $ $| $ _ $ $| $ $ $$$| $ $| $ _ $/
    | $ $ | $ $| $ $ \ $ $| $ $ \ $ $| $ $ | $ $| $ $ \ $ | $ $| $ $ | $ $ \ $ | $ $| $ $
    | $ $ | $ $| $$$$$$/| $ $ | $ $| $$$$$$/| $ $ \ | $ $| $ $ | $ $| $ $ \ | $ $| $$$$$$
    |_/ |_/ \ _/_/ |_/ |_/ \ _/_/ |_/ |_/ \ _/_/ |_/ |_/ \ _/_/ |_/ |_/ \ _/_/

==== Registrasi Pelanggan ====
Masukkan nama pelanggan: Yaya
Masukkan username: yaya
Masukkan kata_sandi:
Masukkan nomor telepon: 081012345678901
Database connected successfully
Registrasi berhasil!

Tekan Enter untuk melanjutkan...|
```

Gambar 4.2 Tampilan Fitur Registrasi Pelanggan

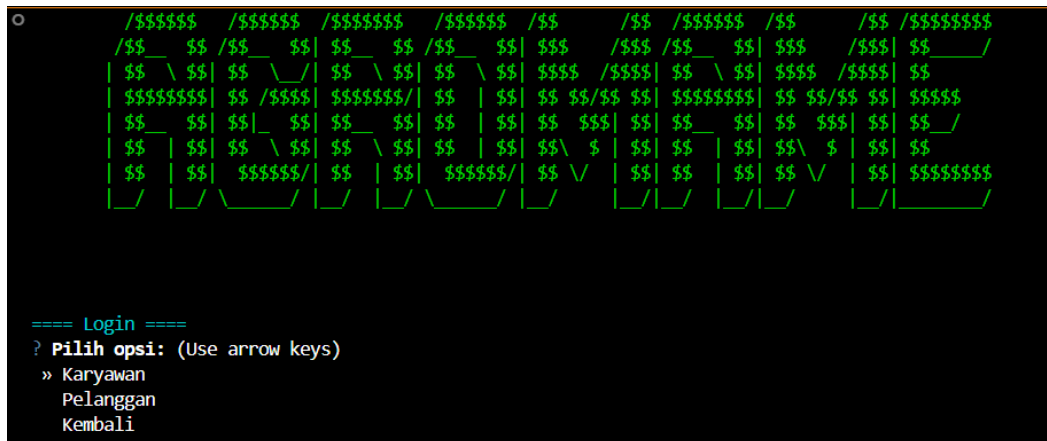
Fitur ini digunakan agar pengguna baru dapat membuat akun di sistem. Saat dijalankan, program meminta input berupa nama, username, kata sandi, dan nomor telepon. Data yang dimasukkan kemudian disimpan ke dalam tabel pelanggan di database PostgreSQL dengan tambahan tanggal registrasi. Setelah proses berhasil, sistem menampilkan pesan konfirmasi dan kembali ke menu utama.

Struktur data pada fitur ini berisi informasi yang dimasukkan pengguna, yaitu nama pelanggan, username, kata sandi, dan nomor telepon yang semuanya berupa string. Selain itu, sistem juga otomatis menambahkan tanggal registrasi dengan fungsi `datetime.now()` untuk mencatat waktu akun dibuat.

Langkah penggunaan fitur ini dimulai dengan memilih opsi registrasi pada menu utama. Selanjutnya pengguna mengisi data sesuai permintaan program. Setelah semua input diberikan, sistem mengeksekusi query `INSERT` untuk menyimpan data ke database. Jika berhasil, muncul pesan “Registrasi berhasil!” dan pengguna diarahkan kembali ke menu utama untuk melanjutkan aktivitas.

## 4.3 Fitur Login

### 4.3.1 Fitur Login (main)



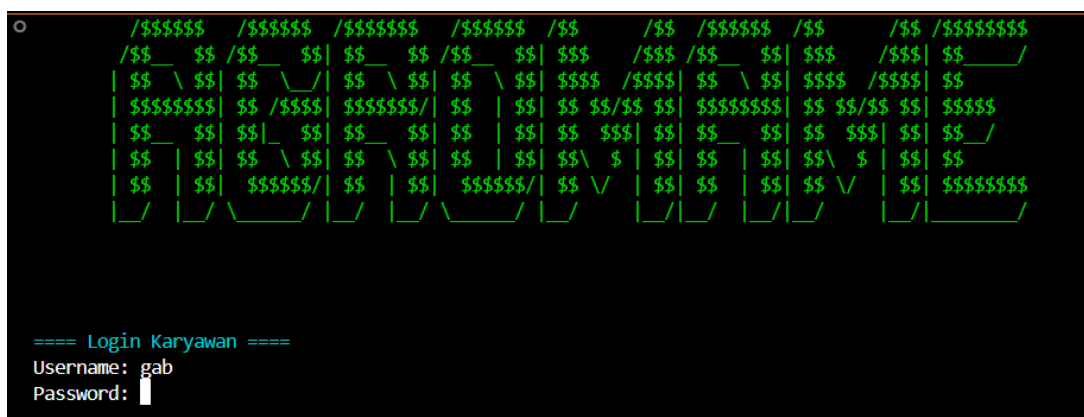
Gambar 4.3.1 Tampilan Fitur Login (main)

Fitur login utama dipakai agar pengguna bisa masuk sesuai perannya. Saat dijalankan, sistem menampilkan pilihan login untuk Karyawan atau Pelanggan. Dengan begitu, pengguna dapat memilih sesuai kebutuhan sebelum melanjutkan ke menu berikutnya.

Data yang dipakai di fitur ini sederhana. Menu login ditulis dalam bentuk list berisi nomor dan opsi, lalu input pengguna berupa angka (1 atau 2). Pilihan ini yang menentukan fungsi mana yang akan dijalankan, apakah `login_karyawan()` atau `login_pelanggan()`.

Langkah penggunaan dimulai dengan menampilkan menu login di layar. Pengguna mengetik angka sesuai peran yang dipilih. Jika memilih 1 maka sistem masuk ke login karyawan, jika memilih 2 maka sistem masuk ke login pelanggan. Jika input tidak sesuai, sistem akan menampilkan pesan “Pilihan tidak valid” dan meminta pengguna mencoba lagi.

### 4.3.2 Fitur Login Karyawan



Gambar 4.3.2 Tampilan Fitur Login Karyawan

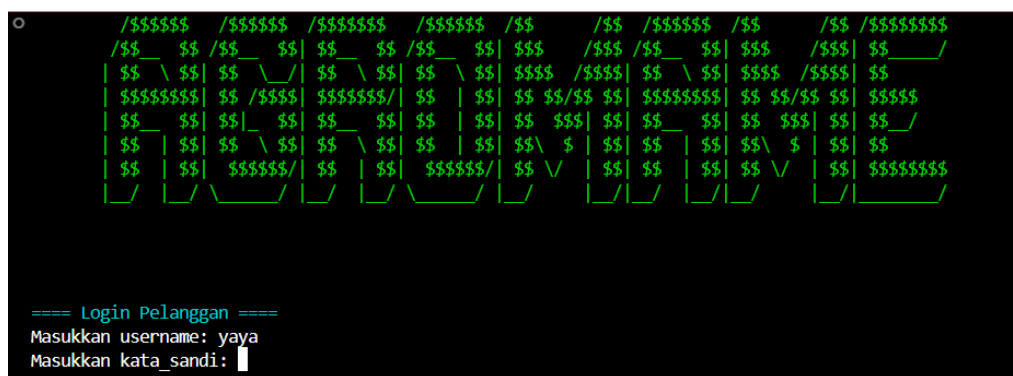
Fitur ini digunakan agar karyawan bisa masuk ke sistem sesuai peran masing-masing. Setelah memilih login sebagai karyawan, pengguna diminta memasukkan username dan password. Sistem

akan mencocokkan data tersebut dengan isi database. Kalau cocok, akan ditampilkan data login seperti nama lengkap, username, password dan role karyawan.

Data yang diproses di sini meliputi nama karyawan, username, password, dan jabatan. Semua data diambil dari tabel *karyawan* dan *jabatan* yang saling terhubung. Password ditampilkan dalam bentuk bintang agar tidak terlihat langsung.

Langkah penggunaannya yaitu karyawan memasukkan username dan password, lalu sistem mencari data di database. Kalau berhasil login, sistem akan menampilkan informasi karyawan dan langsung mengarahkan ke menu sesuai role yaitu Manajer Produksi, Admin Produksi, atau Admin Gudang. Kalau data tidak cocok, akan muncul pesan bahwa login gagal.

### 4.3.3 Fitur Login Pelanggan



Gambar 4.3.3 Tampilan Fitur Login Pelanggan

Fitur ini digunakan agar pelanggan yang sudah terdaftar bisa masuk ke sistem. Saat dijalankan, pengguna diminta memasukkan username dan kata sandi. Sistem akan mencocokkan data tersebut dengan isi tabel pelanggan di database. Kalau sesuai, akan ditampilkan informasi login seperti nama lengkap, username, dan password.

Data yang digunakan dalam fitur ini meliputi nama pelanggan, username, dan kata sandi. Semua data diambil dari tabel *pelanggan*. Kata sandi ditampilkan dalam bentuk bintang agar tidak terlihat langsung.

Langkah penggunaannya yaitu pelanggan memasukkan username dan kata sandi, lalu sistem mencari data di database. Kalau cocok, muncul pesan “Selamat datang!” dan pengguna langsung diarahkan ke menu pelanggan. Kalau tidak cocok, sistem akan menampilkan pesan bahwa login gagal.

## 4.4 Fitur Halaman Menu Admin Produksi



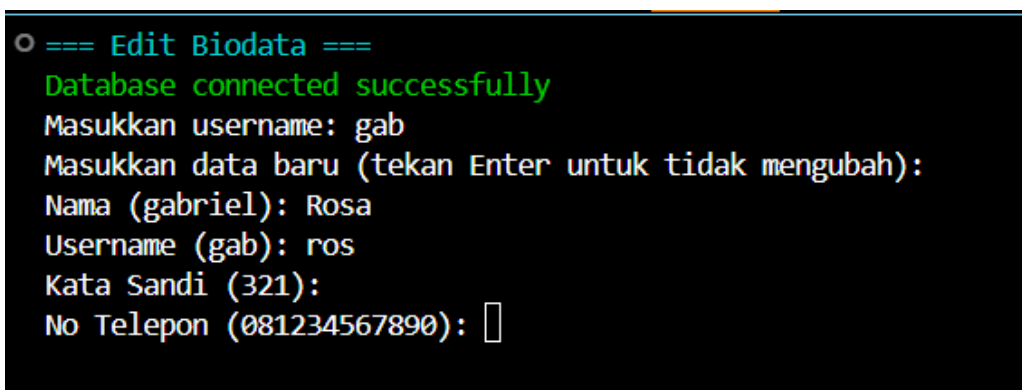
Gambar 4.5 Tampilan Fitur Lihat Biodata Karyawan

Fitur ini digunakan untuk menampilkan informasi pribadi karyawan berdasarkan username yang dimasukkan. Setelah pengguna mengetik username, sistem akan mencari data di tabel *karyawan* dan menampilkan biodata lengkap jika ditemukan.

Data yang ditampilkan meliputi nama, username, kata sandi, jenis kelamin, tanggal lahir, nomor telepon, tanggal bergabung, dan gaji. Semua informasi ditampilkan dalam bentuk tabel agar mudah dibaca.

Langkah penggunaannya yakni ketika pengguna memasukkan username, lalu sistem mengambil data dari database. Jika data ditemukan, biodata akan ditampilkan di layar. Jika tidak ada, sistem akan memberi pesan bahwa data tidak ditemukan.

## 4.6 Fitur Edit Biodata Karyawan



```
o === Edit Biodata ===
Database connected successfully
Masukkan username: gab
Masukkan data baru (tekan Enter untuk tidak mengubah):
Nama (gabriel): Rosa
Username (gab): ros
Kata Sandi (321):
No Telepon (081234567890):
```

Gambar 4.6 Tampilan Fitur Edit Biodata Karyawan

Fitur ini digunakan untuk memperbarui data karyawan yang sudah ada di sistem. Setelah memasukkan username, sistem akan menampilkan data lama dan meminta pengguna mengisi ulang data yang ingin diubah. Jika ada bagian yang tidak ingin diubah, pengguna cukup menekan enter agar data lama tetap dipakai.

Data yang bisa diedit meliputi nama, username, kata sandi, dan nomor telepon. Semua input baru akan menggantikan data lama jika diisi. Kalau tidak diisi, sistem otomatis menggunakan data sebelumnya.

Langkah penggunaannya dimulai dengan memasukkan username karyawan. Jika data ditemukan, sistem akan menampilkan form edit dan meminta pengguna mengisi ulang data. Setelah selesai, data akan diperbarui di database. Kalau username tidak ditemukan, sistem akan menampilkan pesan bahwa data tidak tersedia.

## 4.7 Fitur Input Hasil Panen

```

o === Laporan Hasil Panen ===
Database connected successfully

```

ID Hasil Panen	Hasil Panen (kg)	Tanggal Panen	Nama Karyawan
13	205.8	2025-11-11	Rosa
6	200	2025-11-20	Rosa
7	65.7	2025-11-21	Rosa
5	78.1	2025-11-27	Rosa
8	300	2025-12-06	Rosa
17	0	2025-12-08	Rosa
14	0	2025-12-12	Rosa

```

=== Input Hasil Panen ===
Masukkan jumlah hasil panen (kg) [Enter untuk batal]: 78.90
Masukkan tanggal panen (YYYY-MM-DD) [Enter untuk batal]: 2025-12-13
Masukkan nama karyawan [Enter untuk batal]: rosa

--- Konfirmasi Data ---
Jumlah Panen : 78.9 kg
Tanggal Panen: 2025-12-13
Nama Karyawan: rosa
Apakah data sudah benar? (y/n): y

```

Gambar 4.7 Tampilan Fitur Input Hasil Panen

Fitur ini digunakan untuk mencatat data hasil panen ke dalam sistem. Saat dijalankan, pengguna diminta mengisi jumlah panen dalam satuan kilogram, tanggal panen, dan ID karyawan yang bertugas. Setelah semua data diinput, sistem akan menyimpan informasi tersebut ke database dan menampilkan pesan bahwa input berhasil.

Data yang dicatat meliputi jumlah hasil panen (float), tanggal panen (YYYY-MM-DD), dan ID karyawan (integer). Semua data langsung dimasukkan ke tabel `hasil_panen` menggunakan query `INSERT`.

Langkah penggunaannya yaitu pengguna mengisi data sesuai instruksi di terminal, lalu sistem menyimpan ke database. Kalau input sudah lengkap dan formatnya benar, akan muncul pesan “Input hasil panen berhasil!” sebagai konfirmasi.

## 4.8 Fitur Hapus Hasil Panen

```

o === Laporan Hasil Panen ===
Database connected successfully

```

ID Hasil Panen	Hasil Panen (kg)	Tanggal Panen	Nama Karyawan
13	205.8	2025-11-11	Rosa
6	200	2025-11-20	Rosa
7	65.7	2025-11-21	Rosa
5	78.1	2025-11-27	Rosa
8	300	2025-12-06	Rosa
17	0	2025-12-08	Rosa
14	0	2025-12-12	Rosa
20	97.7	2025-12-13	Rosa

```

Masukkan ID hasil panen yang akan dihapus: 20
Yakin ingin menghapus hasil panen ini? (y/n): y
Database connected successfully
Hasil Panen berhasil dihapus.

Tekan Enter untuk melanjutkan...

```

Gambar 4.8 Tampilan Fitur Hapus Hasil Panen

Fitur ini digunakan untuk menghapus data hasil panen yang sudah tercatat di sistem. Admin Produksi cukup memasukkan ID hasil panen yang ingin dihapus, lalu sistem akan menjalankan perintah penghapusan dari database.

Data yang dipakai hanya berupa ID hasil panen dalam bentuk angka. Sistem menggunakan query DELETE untuk menghapus data dari tabel hasil\_panen berdasarkan ID yang dimasukkan.

Langkah penggunaannya yaitu pengguna mengetik ID hasil panen, lalu sistem menghapus data tersebut dari database. Jika berhasil, muncul pesan “Hasil Panen berhasil dihapus” sebagai konfirmasi.

## 4.9 Fitur Lihat Laporan Hasil Panen

o === Laporan Hasil Panen ===  
Database connected successfully

ID Hasil Panen	Hasil Panen (kg)	Tanggal Panen	Nama Karyawan
13	205.8	2025-11-11	Rosa
6	200	2025-11-20	Rosa
7	65.7	2025-11-21	Rosa
5	78.1	2025-11-27	Rosa
8	300	2025-12-06	Rosa
17	0	2025-12-08	Rosa
14	0	2025-12-12	Rosa
20	97.7	2025-12-13	Rosa

Tekan Enter untuk melanjutkan...

Gambar 4.9 Tampilan Fitur Lihat Laporan Hasil Panen

Fitur ini digunakan untuk menampilkan data hasil panen yang sudah tercatat di sistem. Saat dijalankan, sistem akan mengambil semua data dari tabel hasil\_panen dan menampilkannya dalam bentuk tabel, lengkap dengan informasi jumlah panen, tanggal panen, dan nama karyawan yang bertugas.

Data yang ditampilkan meliputi ID hasil panen, jumlah panen dalam kilogram, tanggal panen, dan nama karyawan. Data diambil dari dua tabel yang saling terhubung, yaitu hasil\_panen dan karyawan.

Langkah penggunaannya yaitu sistem menjalankan query untuk mengambil semua data hasil panen, lalu menampilkannya dalam format tabel menggunakan tabulate. Tampilan ini memudahkan admin produksi untuk melihat laporan panen secara keseluruhan.

## 4.10 Fitur Edit Hasil Panen

o === Laporan Hasil Panen ===  
Database connected successfully

ID Hasil Panen	Hasil Panen (kg)	Tanggal Panen	Nama Karyawan
13	205.8	2025-11-11	Rosa
6	200	2025-11-20	Rosa
7	65.7	2025-11-21	Rosa
5	78.1	2025-11-27	Rosa
8	300	2025-12-06	Rosa
17	0	2025-12-08	Rosa
14	0	2025-12-12	Rosa
20	78.9	2025-12-13	Rosa

Masukkan ID hasil panen yang akan diedit [Enter untuk batal]: 20  
Database connected successfully  
Masukkan data baru (Enter untuk tidak mengubah / batal jika kosong di awal):  
Tanggal Panen (2025-12-13):  
Jumlah Panen (78.9): 97.7  
Nama Karyawan (4) [Enter untuk batal]: rosa

--- Konfirmasi Data ---  
Tanggal Panen : 2025-12-13  
Jumlah Panen : 97.7  
Nama Karyawan : rosa  
Apakah data sudah benar? (y/n): y

Gambar 4.10 Tampilan Fitur Edit Hasil Panen

Fitur ini digunakan untuk memperbarui data hasil panen yang sudah tercatat di sistem. Pengguna cukup memasukkan ID hasil panen yang ingin diedit, lalu sistem akan menampilkan data lama dan meminta input baru. Jika ada bagian yang tidak ingin diubah, pengguna bisa langsung tekan enter agar data lama tetap dipakai.

Data yang bisa diedit meliputi tanggal panen, jumlah panen, dan ID karyawan. Sistem akan mengganti data lama dengan input baru jika diisi, lalu menyimpan perubahan ke database menggunakan query UPDATE.

Langkah penggunaannya dimulai dengan memasukkan ID hasil panen. Jika data ditemukan, sistem menampilkan form edit dan meminta pengguna mengisi ulang data. Setelah selesai, sistem menyimpan perubahan dan menampilkan pesan bahwa data berhasil diperbarui. Kalau ID tidak ditemukan, sistem akan menampilkan pesan bahwa data tidak tersedia.

## 4.11 Fitur Input Produksi

```

o === Laporan Hasil Panen ===
Database connected successfully

```

ID Hasil Panen	Hasil Panen (kg)	Tanggal Panen	Nama Karyawan
13	205.8	2025-11-11	Rosa
6	200	2025-11-20	Rosa
7	65.7	2025-11-21	Rosa
5	78.1	2025-11-27	Rosa
8	300	2025-12-06	Rosa
17	0	2025-12-08	Rosa
14	0	2025-12-12	Rosa

```

=== Input Produksi ===
Masukkan tanggal produksi (YYYY-MM-DD) [Enter untuk batal]: 2025-12-13
Masukkan jumlah produksi (kg) [Enter untuk batal]: 179.6
Masukkan ID hasil panen [Enter untuk batal]: 13
Masukkan ID status produksi (2. ditunda / 3. proses) [Enter untuk default=3]: 3
Masukkan nama karyawan [Enter untuk batal]: rosa
Database connected successfully

--- Konfirmasi Data ---
Tanggal Produksi : 2025-12-13
Jumlah Produksi : 179.6 kg
ID Hasil Panen : 13
Status Produksi : 3
Nama Karyawan : rosa
Apakah data sudah benar? (y/n): y

```

Gambar 4.11 Tampilan Fitur Input Produksi

Fitur ini digunakan untuk mencatat data produksi ke dalam sistem. Pengguna diminta mengisi tanggal produksi, jumlah produksi dalam kilogram, ID karyawan, ID hasil panen, dan ID status produksi. Setelah semua data diinput, sistem menyimpan informasi tersebut ke database dan menampilkan pesan bahwa input berhasil.

Data yang dicatat meliputi tanggal produksi (YYYY-MM-DD), jumlah produksi (float), ID karyawan, ID hasil panen, dan ID status produksi. Semua data langsung dimasukkan ke tabel produksi menggunakan query INSERT.

Langkah penggunaannya dimulai dengan Pengguna mengisi data sesuai instruksi di terminal, lalu sistem menyimpan ke database. Jika input sudah lengkap dan formatnya benar, akan muncul pesan “Input produksi berhasil!” sebagai konfirmasi.

## 4.12 Fitur Edit Produksi

0 === Laporan Produksi ===  
Database connected successfully

ID Produksi	Tanggal Produksi	Tanggal Selesai	Jumlah Produksi	Nama Karyawan	ID Hasil Panen	Jumlah Hasil Panen (kg)	Status Produksi
16	2025-12-15	2025-12-11	200.7	Rosa	14	0	Selesai
23	2025-12-13		179.6	Rosa	13	26.2	Proses
17	2025-12-10	2025-12-12	195.8	Rosa	17	0	Selesai
15	2025-12-07	2025-12-07	0	Rosa	8	300	Selesai
18	2025-12-03		0	Rosa	6	200	Proses
11	2025-11-22	2025-12-06	0	Astoria Malfoy	8	300	Selesai
10	2025-11-21	2025-11-28	0	Laili	7	65.7	Proses
9	2025-11-20	2025-12-06	0	Rosa	6	200	Selesai

=== Edit Produksi ===  
Masukkan ID produksi yang akan diedit [Enter untuk batal]: 23  
Database connected successfully  
Masukkan data baru (Enter untuk tidak mengubah):  
Tanggal Produksi (2025-12-13):  
Jumlah Produksi (179.6):  
Nama Karyawan (4) [Enter untuk batal]: rosa  
ID Hasil Panen (13):  
ID Status Produksi (1. Selesai/2. Ditunda/3. Proses) (3): 1  
--- Konfirmasi Data ---  
Tanggal Produksi : 2025-12-13  
Jumlah Produksi : 179.6  
Nama Karyawan : rosa  
ID Hasil Panen : 13  
Status Produksi : 1  
Apakah data sudah benar? (y/n): y

Gambar 4.12 Tampilan Fitur Edit Produksi

Fitur ini digunakan untuk memperbarui data produksi yang sudah tercatat di sistem. Pengguna cukup memasukkan ID produksi yang ingin diedit, lalu sistem akan menampilkan data lama dan meminta input baru. Kalau ada bagian yang tidak ingin diubah, pengguna bisa langsung tekan enter agar data lama tetap dipakai.

Data yang bisa diedit meliputi tanggal produksi, jumlah produksi, ID karyawan, ID hasil panen, dan ID status produksi. Kalau status produksi diubah menjadi “selesai”, sistem akan otomatis menambahkan tanggal selesai menggunakan `datetime.now()`.

Langkah penggunaannya dimulai dengan memasukkan ID produksi. Jika data ditemukan, sistem menampilkan form edit dan meminta pengguna mengisi ulang data. Setelah selesai, sistem menyimpan perubahan ke database dan menampilkan pesan bahwa data berhasil diperbarui. Kalau ID tidak ditemukan, sistem akan menampilkan pesan bahwa data tidak tersedia.

## 4.13 Fitur Hapus Produksi

0 === Laporan Produksi ===  
Database connected successfully

ID Produksi	Tanggal Produksi	Tanggal Selesai	Jumlah Produksi	Nama Karyawan	ID Hasil Panen	Jumlah Hasil Panen (kg)	Status Produksi
16	2025-12-15	2025-12-11	200.7	Rosa	14	0	Selesai
23	2025-12-13	2025-12-13	179.6	Rosa	13	0	Selesai
17	2025-12-10	2025-12-12	195.8	Rosa	17	0	Selesai
15	2025-12-07	2025-12-07	0	Rosa	8	300	Selesai
18	2025-12-03		0	Rosa	6	200	Proses
11	2025-11-22	2025-12-06	0	Astoria Malfoy	8	300	Selesai
10	2025-11-21	2025-11-28	0	Laili	7	65.7	Proses
9	2025-11-20	2025-12-06	0	Rosa	6	200	Selesai

=== Hapus Produksi ===  
Masukkan ID produksi yang akan dihapus: 23  
Yakin menghapus produksi ini? (y/n): y

Gambar 4.13 Tampilan Fitur Hapus Produksi

Fitur ini digunakan untuk menghapus data produksi yang sudah tercatat di sistem. Pengguna cukup memasukkan ID produksi yang ingin dihapus, lalu sistem akan menjalankan perintah penghapusan dari database.

Data yang dipakai hanya berupa ID produksi dalam bentuk angka. Sistem menggunakan query DELETE untuk menghapus data dari tabel produksi berdasarkan ID yang dimasukkan.

Langkah penggunaannya dimulai dengan Pengguna mengetik ID produksi, lalu sistem menghapus data tersebut dari database. Jika berhasil, muncul pesan “Produksi berhasil dihapus” sebagai konfirmasi.

## 4.14 Fitur Lihat Laporan Produksi

o --- Laporan Produksi ---  
Database connected successfully

ID Produksi	Tanggal Produksi	Tanggal Selesai	Jumlah Produksi	Nama Karyawan	ID Hasil Panen	Jumlah Hasil Panen (kg)	Status Produksi
16	2025-12-15	2025-12-11	200,7	Rosa	14	0	Selesai
23	2025-12-13	2025-12-13	179,6	Rosa	13	0	Selesai
17	2025-12-10	2025-12-12	195,8	Rosa	17	0	Selesai
15	2025-12-07	2025-12-07	0	Rosa	8	300	Selesai
18	2025-12-03		0	Rosa	6	200	Proses
11	2025-11-22	2025-12-06	0	Astoria Malfoy	8	300	Selesai
10	2025-11-21	2025-11-28	0	Lalli	7	65,7	Proses
9	2025-11-20	2025-12-06	0	Rosa	6	200	Selesai

Tekan Enter untuk melanjutkan...

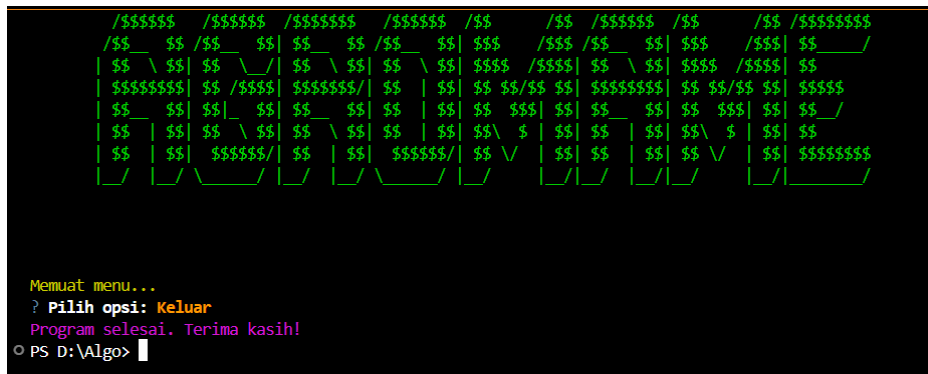
Gambar 4.14 Tampilan Fitur Lihat Laporan Produksi

Fitur ini digunakan untuk menampilkan data produksi yang sudah tercatat di sistem. Saat dijalankan, sistem akan mengambil semua data dari tabel *produksi* dan menampilkannya dalam bentuk tabel, lengkap dengan informasi tanggal produksi, tanggal selesai, jumlah produksi, nama karyawan, jumlah hasil panen, dan status produksi.

Data yang ditampilkan berasal dari beberapa tabel yang saling terhubung, yaitu produksi, karyawan, hasil\_panen, dan status\_produksi. Semua data digabungkan menggunakan query JOIN, lalu ditampilkan dalam format tabel agar mudah dibaca.

Langkah penggunaannya cukup langsung. Sistem menjalankan query untuk mengambil semua data produksi, lalu menampilkannya menggunakan tabulate. Tampilan ini memudahkan admin untuk melihat laporan produksi secara keseluruhan dan memantau statusnya.

## 4.15 Fitur Logout



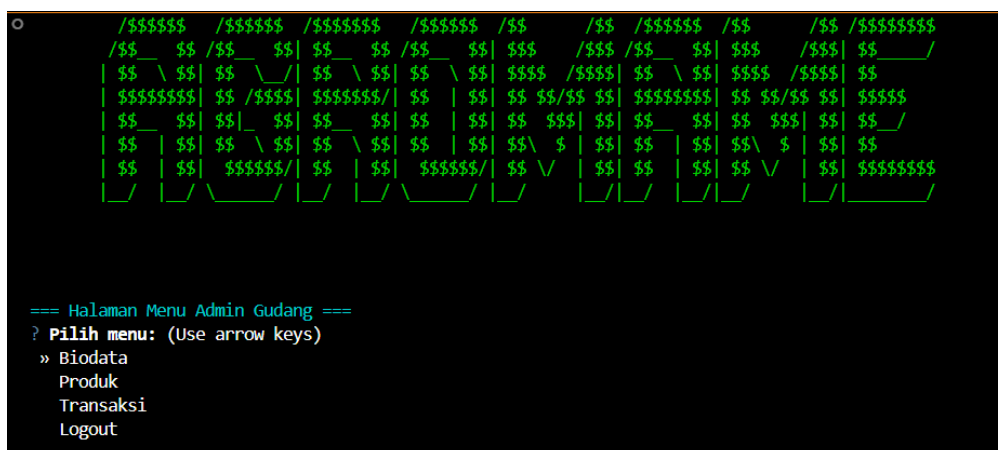
Gambar 4.15 Tampilan Fitur Logout

Fitur ini digunakan untuk keluar dari sistem setelah pengguna selesai menggunakan aplikasi. Saat dijalankan, sistem akan menampilkan pesan bahwa pengguna telah logout, lalu proses aplikasi akan dihentikan.

Data yang diproses di fitur ini sangat sederhana, hanya berupa perintah untuk menampilkan pesan dan mengakhiri sesi. Tidak ada input tambahan atau hubungan dengan database.

Langkah penggunaannya setelah memilih menu logout, sistem akan menampilkan pesan “Anda telah keluar dari sistem” sebagai konfirmasi bahwa sesi telah berakhir.

## 4.16 Fitur Halaman Menu Admin Gudang



Gambar 4.16 Tampilan Fitur Halaman Menu Admin Gudang

Fitur ini digunakan oleh admin gudang untuk mengakses berbagai menu yang berkaitan dengan data produksi, produk, dan transaksi. Saat dijalankan, sistem menampilkan 10 pilihan menu dalam bentuk tabel, mulai dari melihat biodata hingga logout.

Data menu ditampilkan dalam bentuk list dua kolom: nomor dan nama menu. Tampilan diformat menggunakan tabulate agar lebih rapi dan mudah dibaca. Input dari pengguna berupa angka 1 sampai 10 sesuai pilihan yang tersedia.

Langkah penggunaannya yakni Admin Gudang memilih menu dengan mengetik angka, lalu sistem menjalankan fungsi yang sesuai. Jika input tidak valid, sistem akan menampilkan pesan kesalahan dan meminta pengguna mencoba lagi.

## 4.17 Fitur Input Produk

o === Laporan Produksi ===  
Database connected successfully

ID Produksi	Tanggal Produksi	Tanggal Selesai	Jumlah Produksi	Nama Karyawan	ID Hasil Panen	Jumlah Hasil Panen (kg)	Status Produksi
16	2025-12-15	2025-12-11	200.7	Rosa	14	0	Selesai
17	2025-12-10	2025-12-12	179.8	Rosa	17	0	Selesai
15	2025-12-07	2025-12-07	0	Rosa	8	300	Selesai
18	2025-12-03		0	Rosa	6	200	Proses
11	2025-11-22	2025-12-06	0	Astoria Malfoy	8	300	Selesai
10	2025-11-21	2025-11-28	0	Laili	7	65.7	Proses
9	2025-11-20	2025-12-06	0	Rosa	6	200	Selesai

=== Input Produk ===  
Masukkan jumlah stok: 10  
Masukkan id jenis (1.Fresh, 2.Frozen, 3.Jus): 2

Gambar 4.17 Tampilan Fitur Input Produk

Fitur ini digunakan oleh admin gudang untuk menambahkan produk baru ke sistem berdasarkan stok yang tersedia dari proses produksi. Saat dijalankan, sistem akan meminta jumlah stok yang dibutuhkan dan jenis produk yang dipilih. Setelah itu, sistem akan mencari data produksi yang masih memiliki stok dan mengurangnya sesuai kebutuhan.

Data produksi yang ditemukan akan diupdate satu per satu berdasarkan urutan tanggal produksi. Jika stok produksi cukup, sistem langsung mengurangi sesuai jumlah yang diminta. Jika stok tidak cukup, sistem akan menghabiskan stok yang tersedia dan melanjutkan ke produksi berikutnya sampai kebutuhan terpenuhi. Semua perubahan disimpan ke database agar data tetap akurat.

Langkah penggunaannya yakni Admin Gudang memilih menu dengan mengetik angka, lalu sistem menjalankan fungsi yang sesuai. Jika input tidak valid, sistem akan menampilkan pesan kesalahan dan meminta pengguna mencoba lagi.

## 4.18 Fitur Hapus Produk

```

o === Laporan Produk ===
Database connected successfully

```

ID Produk	Jenis Produk	Jumlah Produk	Tanggal Update Stok	Jumlah Produksi
6	jus	180	2025-11-28	0
7	fresh	104	2025-12-04	0
8	fresh	300	2025-12-05	0
9	fresh	90	2025-12-06	0
5	frozen	150	2025-12-12	0
10	frozen	270	2025-12-13	0
15	jus	90	2025-12-13	nan

```

=== Hapus Produk ===
Masukkan ID produk yang akan dihapus: 15
Yakin menghapus produk ini? (y/n): y

```

Gambar 4.18 Tampilan Fitur Hapus Produk

Fitur ini digunakan oleh admin gudang untuk menghapus data produk dari sistem. Saat dijalankan, sistem akan meminta input ID produk yang ingin dihapus. ID ini digunakan sebagai acuan untuk mencari produk yang sesuai di database.

Data produksi yang ditemukan akan dikurangi sesuai jumlah stok yang diminta. Kalau stok cukup, sistem langsung kurangi dan tampilkan pesan berhasil. Kalau stok tidak cukup, sistem akan memberi peringatan bahwa permintaan tidak bisa dipenuhi semua.

Langkah penggunaannya yakni admin cukup menjalankan fungsi, lalu isi jumlah stok dan pilih jenis produk. Sistem akan otomatis memproses dan memperbarui data produksi. Fitur ini membantu menjaga agar stok produk sesuai dengan hasil produksi yang tersedia.

## 4.19 Fitur Lihat Laporan Produk

```

o === Laporan Produk ===
Database connected successfully

```

ID Produk	Jenis Produk	Jumlah Produk	Tanggal Update Stok	Jumlah Produksi
6	jus	180	2025-11-28	0
7	fresh	104	2025-12-04	0
8	fresh	300	2025-12-05	0
9	fresh	90	2025-12-06	0
5	frozen	150	2025-12-12	0
10	frozen	270	2025-12-13	0

```

Tekan Enter untuk melanjutkan...

```

Gambar 4.19 Tampilan Fitur Lihat Laporan Produk

Fitur ini digunakan oleh admin gudang untuk melihat data produk yang tersimpan di sistem. Saat dijalankan, sistem akan mengambil informasi dari tabel produk, jenis produk, dan produksi. Data yang ditampilkan meliputi ID produk, jenis produk, jumlah stok, tanggal update stok, dan jumlah produksi.

Data produk ditampilkan dalam bentuk tabel yang rapi menggunakan tabulate. Format ini memudahkan admin untuk membaca informasi stok dan produksi secara langsung tanpa harus membuka detail satu per satu. Semua data ditampilkan apa adanya sesuai isi database.

Langkah penggunaannya yakni admin cukup menjalankan fungsi `lihat_laporan_produk()`, lalu sistem otomatis menampilkan tabel laporan produk. Jika data kosong, sistem tetap menampilkan tabel tanpa error. Fitur ini sangat membantu dalam memantau stok harian dan mencatat hasil produksi dengan cepat dan jelas.

## 4.20 Fitur Edit Produk



```
o === Laporan Produk ===
Database connected successfully
```

ID Produk	Jenis Produk	Jumlah Produk	Tanggal Update Stok	Jumlah Produksi
6	jus	180	2025-11-28	0
7	fresh	104	2025-12-04	0
8	fresh	300	2025-12-05	0
9	fresh	90	2025-12-06	0
5	frozen	150	2025-12-12	0
10	frozen	290	2025-12-12	0

```
=== Edit Produk ===
Database connected successfully
Masukkan ID produk yang akan diedit: 10
Stok baru (290), Enter untuk skip: 270
```

Gambar 4.20 Tampilan Fitur Edit Produk

Fitur ini digunakan oleh admin gudang untuk mengedit data stok produk yang sudah tercatat di sistem. Saat dijalankan, sistem akan meminta input ID produk yang ingin diedit, lalu menampilkan stok lama dan meminta pengguna mengisi stok baru. Jika tidak ingin mengubah, pengguna cukup menekan Enter agar nilai lama tetap digunakan.

Data produk yang dipilih akan diperbarui langsung ke database menggunakan perintah SQL UPDATE, dengan mengisi kolom stok dan mencatat waktu update di `tgl_update_stok`. Sistem akan menyimpan perubahan menggunakan `commit()` dan menampilkan pesan bahwa produk berhasil diperbarui. Jika ID produk tidak ditemukan, sistem akan menampilkan pesan bahwa data tidak tersedia.

Langkah penggunaannya yakni admin cukup menjalankan fungsi, lalu memasukkan ID produk dan stok baru sesuai kebutuhan. Tampilan input dibuat fleksibel agar pengguna bisa memilih untuk mengubah atau mempertahankan data lama. Fitur ini sangat membantu dalam pengelolaan stok harian agar data tetap akurat dan ter-update.

## 4.21 Fitur Update Status Transaksi

o === Laporan Transaksi ===  
Database connected successfully

ID Detail	ID Transaksi	Tanggal	Produk	Jumlah Produk	Status Transaksi
2	1	2025-12-02	frozen	2	dibatalkan
3	4	2025-12-12	frozen	50	selesai
4	5	2025-12-12	frozen	10	selesai
5	6	2025-12-12	frozen	10	selesai

=== Input Status Transaksi ===  
Database connected successfully  
Masukkan ID transaksi yang akan diperbarui: 1  
Masukkan status produksi baru (1. selesai/ 2. proses /3. dibatalkan): 2

Gambar 4.21 Tampilan Fitur Update Status Transaksi

Fitur ini digunakan oleh admin gudang untuk memperbarui status transaksi yang sudah tercatat di sistem. Saat dijalankan, sistem akan meminta input ID transaksi yang ingin diubah, lalu meminta status baru yang dipilih dari tiga opsi yaitu selesai, proses, atau dibatalkan. Input status ditulis dalam bentuk angka 1 sampai 3 sesuai urutan pilihan yang tersedia.

Data status transaksi diperbarui langsung ke database menggunakan perintah SQL UPDATE, dengan mengisi kolom `id_status_transaksi` berdasarkan input pengguna. Sistem akan mengeksekusi perubahan dan menyimpan hasilnya menggunakan `commit()`, lalu menampilkan pesan bahwa status transaksi berhasil diperbarui. Proses ini memastikan bahwa status transaksi selalu sesuai kondisi terbaru di lapangan.

Langkah penggunaannya yakni admin cukup menjalankan fungsi, lalu memasukkan ID transaksi dan memilih status baru sesuai kebutuhan. Jika input valid, sistem akan langsung memperbarui data tanpa error. Fitur ini sangat membantu dalam pengelolaan transaksi harian, terutama untuk mencatat apakah transaksi sudah selesai, masih diproses, atau dibatalkan.

## 4.22 Fitur Lihat Laporan Transaksi

o === Laporan Transaksi ===  
Database connected successfully

ID Detail	ID Transaksi	Tanggal	Produk	Jumlah Produk	Status Transaksi
2	1	2025-12-02	frozen	2	proses
3	4	2025-12-12	frozen	50	selesai
4	5	2025-12-12	frozen	10	selesai
5	6	2025-12-12	frozen	10	selesai

Tekan Enter untuk melanjutkan...

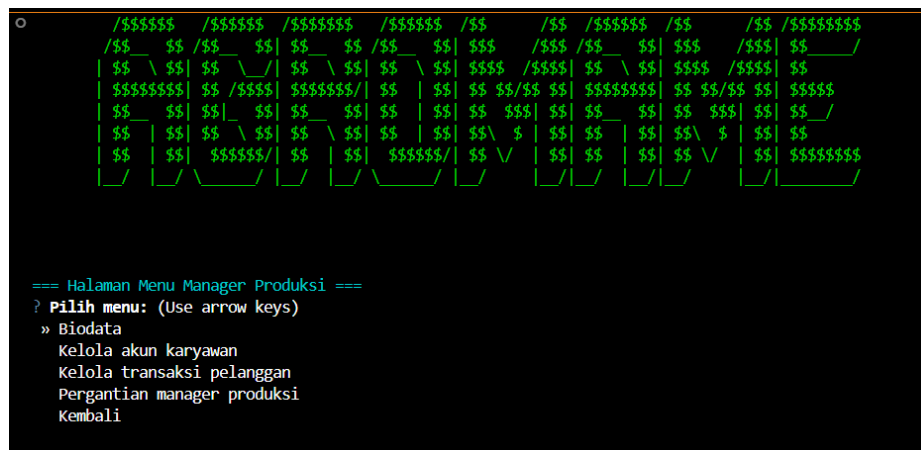
Gambar 4.22 Tampilan Fitur Lihat Laporan Transaksi

Fitur ini digunakan oleh admin gudang untuk melihat laporan transaksi yang sudah tercatat di sistem. Saat dijalankan, sistem akan terhubung ke database dan mengambil data dari tabel transaksi, detail transaksi, produk, jenis produk, serta status transaksi. Informasi yang ditampilkan meliputi ID detail, ID transaksi, tanggal, nama produk, jumlah produk, dan status transaksi.

Data laporan ditampilkan dalam bentuk tabel dengan dua kolom utama yaitu nama field dan isi data. Tampilan diformat menggunakan tabulate dengan gaya fancy\_grid sehingga hasilnya rapi, mudah dibaca, dan siap dipakai untuk monitoring maupun dokumentasi. Semua status transaksi ditampilkan apa adanya, baik yang selesai maupun dibatalkan.

Langkah penggunaannya yakni admin cukup menjalankan fungsi lihat\_laporan\_transaksi(). Setelah itu sistem otomatis menampilkan tabel laporan transaksi. Jika tidak ada data, sistem tetap menampilkan tabel kosong tanpa error sehingga proses tetap berjalan lancar.

## 4.23 Fitur Halaman Menu Manager Produksi



Gambar 4.23 Tampilan Fitur Halaman Menu Manager Produksi

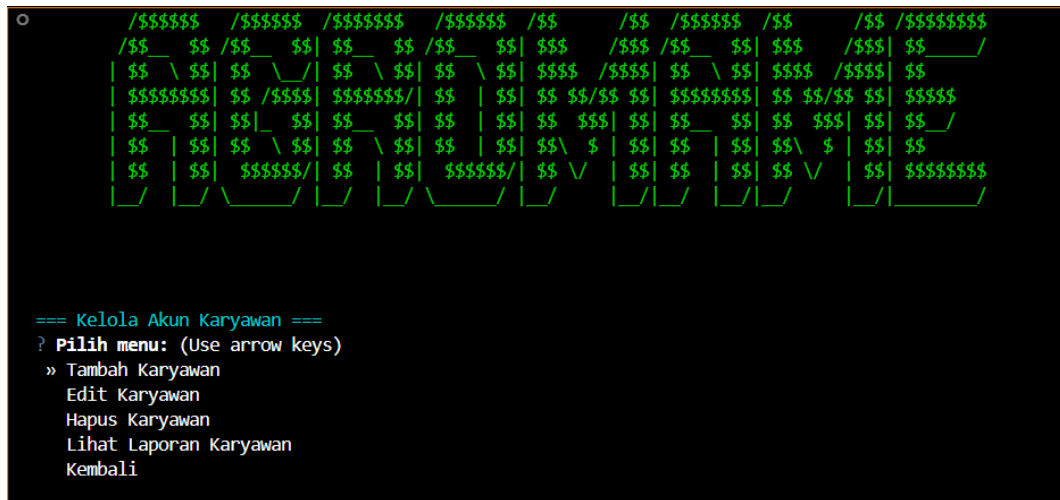
Fitur ini digunakan oleh admin gudang untuk mengakses berbagai menu yang berkaitan dengan data produksi, produk, dan transaksi. Saat dijalankan, sistem menampilkan 10 pilihan menu dalam bentuk tabel, mulai dari melihat biodata hingga logout.

Data menu ditampilkan dalam bentuk list dua kolom: nomor dan nama menu. Tampilan diformat menggunakan tabulate agar lebih rapi dan mudah dibaca. Input dari pengguna berupa angka 1 sampai 10 sesuai pilihan yang tersedia.

Langkah penggunaannya yakni Admin Gudang memilih menu dengan mengetik angka, lalu sistem menjalankan fungsi yang sesuai. Jika input tidak valid, sistem akan menampilkan pesan kesalahan dan meminta pengguna mencoba lagi.

## 4.24 Fitur Kelola Akun Karyawan

#### 4.24.1 Menu Utama Kelola Akun Karyawan



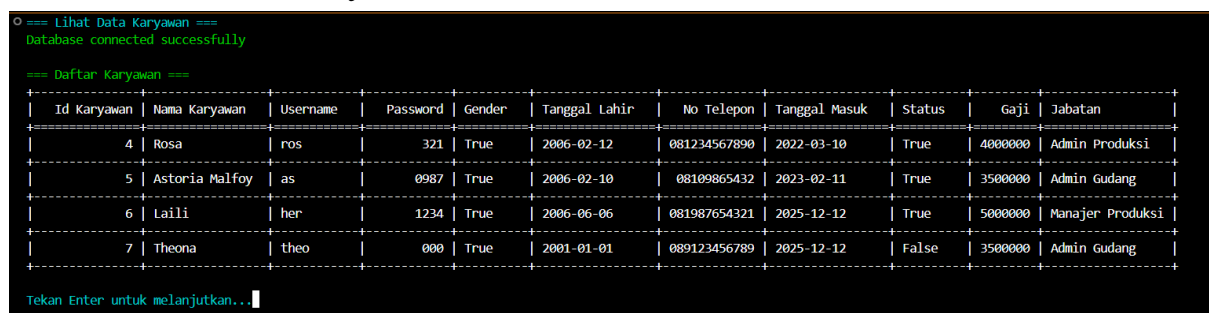
Gambar 4.24.1 Tampilan Fitur Menu Utama Kelola Akun Karyawan

Fitur ini digunakan oleh Manager Produksi untuk mengelola data karyawan di sistem. Saat dijalankan, sistem akan menampilkan lima pilihan menu: melihat data karyawan, menambah karyawan baru, mengedit data karyawan, menghapus karyawan, atau kembali ke menu sebelumnya.

Setiap pilihan akan menjalankan fungsi sesuai kebutuhan. Jika memilih “Lihat Karyawan”, sistem menampilkan daftar karyawan. Jika memilih “Tambah”, “Edit”, atau “Hapus”, sistem akan meminta input data yang diperlukan dan langsung memprosesnya ke database. Semua proses dilakukan secara interaktif melalui terminal.

Langkah penggunaannya yakni Manager Produksi cukup memilih angka menu yang tersedia. Sistem akan menjalankan fungsi yang sesuai dan menampilkan hasilnya langsung. Fitur ini memudahkan pengelolaan data karyawan secara lengkap dan terpusat.

#### 4.24.2 Lihat Karyawan



Gambar 4.24.2 Tampilan Fitur Lihat Karyawan

Fitur ini digunakan oleh Manager Produksi untuk melihat seluruh data karyawan yang tersimpan di sistem. Saat dijalankan, sistem akan mengambil data langsung dari tabel karyawan di database. Informasi yang ditampilkan meliputi ID, nama, username, password, gender, tanggal lahir, nomor telepon, tanggal masuk, status, gaji, dan ID jabatan.

Data karyawan ditampilkan dalam bentuk tabel menggunakan tabulate sehingga tampilannya rapi dan mudah dibaca. Jika tidak ada data, sistem akan menampilkan pesan bahwa belum ada karyawan yang tercatat.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu sistem otomatis menampilkan daftar karyawan. Fitur ini membantu dalam memantau dan mengelola data pegawai secara menyeluruh dan terstruktur.

#### 4.24.3 Tambah Karyawan

```
o === Tambah Karyawan ===
Nama [Enter untuk batal]: Michael
Username [Enter untuk batal]: mic
Password [Enter untuk batal]: 999
Gender (1. Perempuan / 0. Laki-laki) [Enter untuk batal]: 0
Status (1. Aktif / 0. Nonaktif) [Enter untuk batal]: 1
Tanggal Lahir (YYYY-MM-DD) [Enter untuk batal]: 2004-04-04
No Telepon [Enter untuk batal]: 081790123456321
ID Jabatan (2. Admin Produksi / 3. Admin Gudang) [Enter untuk batal]: 3
Gaji [Enter untuk batal]: 3500000

--- Konfirmasi Data ---
Nama      : Michael
Username  : mic
Password  : 999
Gender    : Laki-laki
Status    : Aktif
Tgl Lahir : 2004-04-04
No Telepon : 081790123456321
ID Jabatan : 3
Gaji      : 3500000
Apakah data sudah benar? (y/n): y
```

Gambar 4.24.3 Tampilan Fitur Kelola Akun Karyawan

Fitur ini digunakan oleh Manager Produksi untuk menambahkan data karyawan baru ke sistem. Saat dijalankan, sistem akan meminta input data seperti nama, username, password, gender, status, tanggal lahir, nomor telepon, ID jabatan, dan gaji. Semua data ini akan disimpan sebagai satu entitas karyawan baru.

Data yang dimasukkan akan langsung disimpan ke database menggunakan perintah INSERT. Sistem juga otomatis mencatat tanggal masuk karyawan sesuai tanggal hari ini. Setelah proses selesai, sistem akan menampilkan pesan bahwa karyawan berhasil ditambahkan.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu mengisi data sesuai instruksi yang muncul. Fitur ini membantu memastikan bahwa semua karyawan baru tercatat dengan lengkap dan rapi di sistem.

#### 4.24.4 Edit Karyawan

```
=== Edit Karyawan ===
Masukkan Username Karyawan [Enter untuk batal]: mic
Database connected successfully
Isi data baru (Enter untuk tidak mengubah):
Nama (Michael):
Username (mic):
Password (999):
Gender (False) (1. Perempuan / 0. Laki-laki):
Status (True) (1. Aktif / 0. Nonaktif):
Gaji (3500000): 4000000
ID Jabatan (3) (2. Admin Produksi / 3. Admin Gudang): 2

--- Konfirmasi Data ---
Nama      : Michael
Username  : mic
Password  : 999
Gender    : Laki-laki
Status    : Aktif
Gaji      : 4000000
Jabatan   : 2
Apakah data sudah benar? (y/n): y
Data karyawan berhasil diperbarui.

Tekan Enter untuk melanjutkan...
```

Gambar 4.24.4 Tampilan Fitur Edit Karyawan

Fitur ini digunakan oleh Manager Produksi untuk mengedit data karyawan yang sudah tersimpan di sistem. Saat dijalankan, sistem akan meminta ID karyawan sebagai acuan, lalu menampilkan data lama dan meminta input baru. Jika tidak ingin mengubah, pengguna cukup menekan Enter agar data lama tetap digunakan.

Data karyawan akan diperbarui langsung ke database menggunakan perintah UPDATE. Kolom yang bisa diubah meliputi nama, username, password, gender, status, gaji, dan ID jabatan. Sistem juga akan menampilkan pesan bahwa data berhasil diperbarui jika proses berjalan lancar.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu mengisi data baru sesuai instruksi. Jika ID valid dan data ditemukan, sistem akan menyimpan perubahan. Fitur ini membantu menjaga agar data karyawan tetap akurat dan bisa diperbarui kapan saja sesuai kebutuhan.

## 4.24.5 Hapus Karyawan

```
=== Lihat Data Karyawan ===
Database connected successfully

=== Daftar Karyawan ===
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| Id Karyawan | Nama Karyawan | Username | Password | Gender | Tanggal Lahir | No Telepon | Tanggal Masuk | Status | Gaji | Jabatan |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | Rosa | ros | 321 | True | 2006-02-12 | 081234567890 | 2022-03-10 | True | 4000000 | Admin Produksi |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 5 | Astoria Malfoy | as | 0987 | True | 2006-02-10 | 08109865432 | 2023-02-11 | True | 3500000 | Admin Gudang |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 6 | Laili | her | 1234 | True | 2006-06-06 | 081987654321 | 2025-12-12 | True | 5000000 | Manajer Produksi |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 7 | Theona | theo | 000 | True | 2001-01-01 | 089123456789 | 2025-12-12 | False | 3500000 | Admin Gudang |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+
| 9 | Michael | mic | 999 | False | 2004-04-04 | 081790123456321 | 2025-12-13 | True | 4000000 | Admin Produksi |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+

Tekan Enter untuk melanjutkan...
=== Hapus Karyawan ===
Masukkan ID Karyawan [Enter untuk batal]: 9
Database connected successfully

--- Konfirmasi Hapus ---
Nama : Michael
Username : mic
Apakah yakin ingin menghapus karyawan ini? (y/n): y
```

Gambar 4.24.5 Tampilan Fitur Hapus Karyawan

Fitur ini digunakan oleh Manager Produksi untuk menghapus data karyawan dari sistem. Saat dijalankan, sistem akan meminta input ID karyawan yang ingin dihapus. ID ini digunakan sebagai acuan untuk mencari data karyawan di database.

Jika data ditemukan, sistem akan menjalankan perintah DELETE untuk menghapus karyawan tersebut, lalu menyimpan perubahan dengan commit(). Sistem akan menampilkan pesan bahwa karyawan berhasil dihapus. Jika ID tidak ditemukan, sistem akan memberi peringatan bahwa data tidak tersedia.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu memasukkan ID karyawan yang ingin dihapus. Proses ini membantu menjaga agar data karyawan tetap rapi dan hanya berisi pegawai yang aktif atau relevan.

## 4.25 Fitur Kelola Transaksi Pelanggan

### 4.25.1 Menu Utama Kelola Transaksi Pelanggan

```

/##### /##### /##### /##### /## /## /##### /## /## /#####
/##_ ## /##_ ##| ##_ ## /##_ ##| $$$ /## /##_ ##| $$$ /##| ##_ /
| $$$ \ ##| $$$ \_/| $$$ \ ##| $$$ \ ##| $$$ /#####| $$$ \ ##| $$$ /#####| $$$
| #####| $$$ /#####| #####/| $$$ | $$$| $$$ /##| #####| $$$ /##| $$$
| ##_ ##| $$$_ ##| $$$_ ##| $$$ | $$$| $$$ $$$| $$$| $$$_ $$$| $$$ $$$| $$$| $$$ /
| $$$ | $$$| $$$ \ $$$| $$$ \ $$$| $$$ | $$$| $$$\ $| $$$| $$$| $$$| $$$\ $| $$$| $$$
| $$$ | $$$| #####/| $$$ | $$$| #####/| $$$ \ | $$$| $$$| $$$| $$$ \ | $$$| #####
|_/ |_/ \_/ |_/ |_/ \_/ |_/ |_/|_/ |_/|_/ |_/|_/ |_/|_/

=== Kelola Transaksi Pelanggan ===
? Pilih menu: (Use arrow keys)
» Lihat Laporan Transaksi
  Lihat Riwayat Transaksi
  Kembali
```

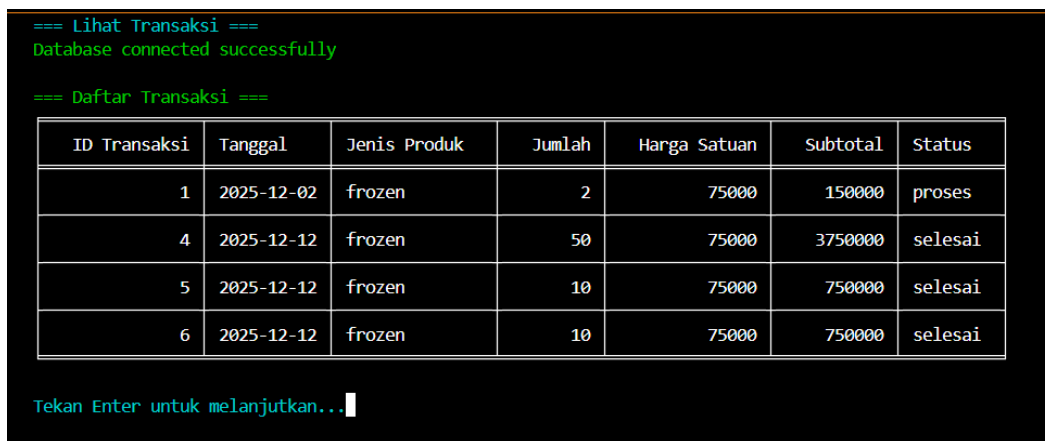
Gambar 4.25.1 Tampilan Fitur Kelola Transaksi Pelanggan

Fitur ini digunakan oleh Manager Produksi untuk mengelola data transaksi pelanggan. Saat dijalankan, sistem akan menampilkan tiga pilihan menu: melihat transaksi, melihat riwayat transaksi, atau kembali ke menu sebelumnya. Menu ini membantu Manager Produksi dalam memantau aktivitas pembelian yang dilakukan oleh pelanggan.

Jika memilih “Lihat Transaksi”, sistem akan menampilkan semua transaksi yang tercatat, baik yang selesai maupun yang dibatalkan. Jika memilih “Lihat Riwayat Transaksi”, sistem hanya menampilkan transaksi yang sudah selesai. Semua data ditampilkan dalam bentuk tabel yang rapi agar mudah dibaca dan dipahami.

Langkah penggunaannya yakni Manager Produksi cukup memilih menu sesuai kebutuhan. Sistem akan menjalankan fungsi yang sesuai dan menampilkan hasilnya langsung di layar. Fitur ini memudahkan pengelolaan transaksi pelanggan secara terpusat dan efisien.

## 4.25.2 Lihat Transaksi



```
=== Lihat Transaksi ===
Database connected successfully

=== Daftar Transaksi ===
```

ID Transaksi	Tanggal	Jenis Produk	Jumlah	Harga Satuan	Subtotal	Status
1	2025-12-02	frozen	2	75000	150000	proses
4	2025-12-12	frozen	50	75000	3750000	selesai
5	2025-12-12	frozen	10	75000	750000	selesai
6	2025-12-12	frozen	10	75000	750000	selesai

```
Tekan Enter untuk melanjutkan...
```

Gambar 4.25.2 Tampilan Fitur Lihat Transaksi

Fitur ini digunakan oleh Manager Produksi untuk melihat semua transaksi yang tercatat di sistem, baik yang selesai maupun yang dibatalkan. Sistem akan mengambil data dari beberapa tabel yang saling terhubung, seperti transaksi, detail transaksi, produk, jenis produk, dan status transaksi. Informasi yang ditampilkan meliputi ID transaksi, tanggal, jenis produk, jumlah, harga satuan, subtotal, dan status.

Data transaksi ditampilkan dalam bentuk tabel menggunakan tabulate sehingga tampilannya rapi dan mudah dibaca. Semua transaksi ditampilkan apa adanya tanpa filter status, sehingga Manager Produksi bisa melihat keseluruhan riwayat transaksi yang terjadi.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu sistem otomatis menampilkan tabel transaksi. Jika belum ada data, sistem akan menampilkan pesan bahwa transaksi tidak ditemukan. Fitur ini berguna untuk memantau aktivitas penjualan dan memastikan semua transaksi tercatat dengan baik.

### 4.25.3 Lihat Riwayat Transaksi

```
=== Lihat Riwayat Transaksi ===  
Database connected successfully  
  
=== Daftar Riwayat Transaksi ===
```

ID Transaksi	Tanggal	Jenis Produk	Jumlah	Harga Satuan	Subtotal	Nama Pelanggan	Metode Pembayaran	Status
4	2025-12-12	frozen	50	75000	3750000	laili	gris	selesai
5	2025-12-12	frozen	10	75000	750000	fajar	transfer	selesai
6	2025-12-12	frozen	10	75000	750000	fajar	transfer	selesai

```
Tekan Enter untuk melanjutkan...
```

Gambar 4.25.3 Tampilan Fitur Lihat Riwayat Transaksi

Fitur ini digunakan oleh Manager Produksi untuk melihat riwayat transaksi yang sudah selesai. Sistem akan mengambil data dari beberapa tabel yang saling berelasi, seperti transaksi, detail transaksi, produk, jenis produk, pelanggan, metode pembayaran, dan status transaksi. Informasi yang ditampilkan meliputi ID transaksi, tanggal, jenis produk, jumlah, harga satuan, subtotal, nama pelanggan, metode pembayaran, dan status.

Data riwayat ditampilkan dalam bentuk tabel menggunakan tabulate sehingga tampilannya rapi dan mudah dibaca. Sistem hanya menampilkan transaksi yang berstatus “selesai”, sehingga laporan ini fokus pada transaksi yang sudah berhasil diproses. Format tabel ini memudahkan Manager Produksi untuk memantau aktivitas penjualan dan preferensi pelanggan.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu sistem otomatis menampilkan tabel riwayat transaksi. Jika belum ada transaksi yang selesai, sistem akan menampilkan pesan bahwa data belum tersedia. Fitur ini sangat membantu dalam evaluasi penjualan dan dokumentasi laporan harian.

## 4.26 Fitur Pergantian Manager Baru

```
=== Ganti Manager Baru ===  
Database connected successfully  
Masukkan username manager lama [Enter untuk batal]: her  
Masukkan data manager baru (Enter untuk batal):  
Nama Manager Baru [Enter untuk batal]: Elena  
Gender (1. Perempuan / 0. Laki-laki) [Enter untuk batal]: 1  
Tgl Lahir (YYYY-MM-DD) [Enter untuk batal]: 2002-02-20  
No Telepon [Enter untuk batal]: 085123456987321  
  
--- Konfirmasi Data ---  
Nama Manager Baru : Elena  
Gender : Perempuan  
Tgl Lahir : 2002-02-20  
No Telepon : 085123456987321  
Tgl Masuk : 2025-12-13  
Apakah data sudah benar? (y/n): y  
Manager baru berhasil diganti. Tanggal masuk otomatis diisi hari ini.  
  
Tekan Enter untuk melanjutkan...
```

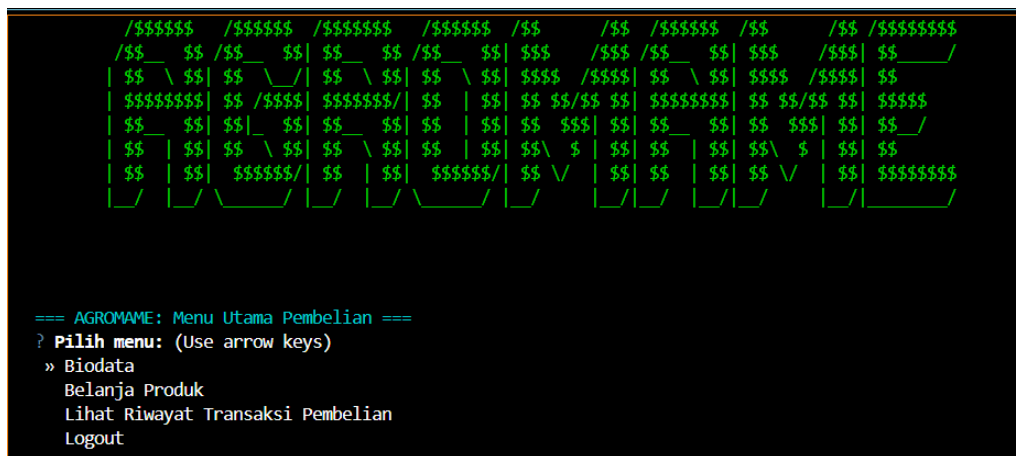
Gambar 4.26 Tampilan Fitur Pergantian Manager Baru

Fitur ini digunakan oleh Manager Produksi untuk mengganti data manager lama dengan data manager baru. Sistem akan meminta username manager lama sebagai acuan, lalu menampilkan form input untuk mengisi data baru seperti nama, gender, tanggal lahir, dan nomor telepon.

Data manager produksi akan diperbarui langsung ke database menggunakan perintah UPDATE. Sistem juga otomatis mencatat tanggal masuk manager baru sesuai tanggal hari ini. Jika username yang dimasukkan tidak ditemukan, sistem akan menampilkan pesan bahwa data manager tidak tersedia.

Langkah penggunaannya yakni Manager Produksi cukup menjalankan fungsi, lalu mengisi data baru sesuai instruksi. Jika semua data valid, sistem akan menyimpan perubahan dan menampilkan pesan bahwa pergantian Manager Produksi berhasil. Dengan cara ini, data manajerial tetap akurat dan sesuai kondisi terbaru.

## 4.27 Fitur Halaman Menu Pelanggan



Gambar 4.27 Tampilan Fitur Halaman Menu Pelanggan

Fitur ini digunakan oleh pelanggan untuk mengakses semua menu pembelian di aplikasi AGROMAME. Saat dijalankan, sistem akan menampilkan daftar pilihan seperti melihat biodata, memilih produk, mengatur keranjang, memilih metode pembayaran, hingga melihat riwayat transaksi.

Setiap pilihan akan menjalankan fungsi sesuai kebutuhan pelanggan. Misalnya, jika memilih “Pilih Produk”, pelanggan bisa menambahkan barang ke keranjang. Jika memilih “Checkout”, sistem akan memproses transaksi. Semua proses dilakukan lewat input angka yang mudah dipahami.

Langkah penggunaannya cukup sederhana: pelanggan memilih nomor menu, lalu sistem langsung menjalankan fungsinya. Jika input bukan angka atau tidak sesuai, sistem akan memberi peringatan. Fitur ini memudahkan pelanggan untuk belanja dan mengelola transaksi secara mandiri.

## 4.28 Fitur Lihat Biodata (Pelanggan)

```
=== Lihat Biodata Pelanggan ===
Database connected successfully
=== BIODATA ===
```

Nama	Username	Kata Sandi	No Telp	Alamat
Yaya	yaya	333	081012345678901	Belum diisi

```
Tekan Enter untuk melanjutkan...
```

Gambar 4.28 Tampilan Fitur Lihat Biodata (Pelanggan)

Fitur ini digunakan oleh pelanggan untuk melihat biodata pribadi yang tersimpan di sistem. Saat dijalankan, sistem akan menampilkan informasi seperti nama, username, kata sandi, nomor telepon, dan alamat lengkap jika sudah diisi.

Jika alamat belum tersedia, sistem akan menampilkan keterangan “Belum diisi” agar pelanggan tahu data masih kosong. Tampilan biodata disusun dalam bentuk tabel rapi menggunakan tabulate, sehingga mudah dibaca langsung di terminal.

Langkah penggunaannya cukup dengan menjalankan fungsi dan memasukkan ID pelanggan. Sistem akan otomatis menampilkan biodata sesuai ID yang dimasukkan. Fitur ini membantu pelanggan memastikan data pribadinya sudah benar dan lengkap.

## 4.29 Fitur Edit Biodata (Pelanggan)

```
=== Edit Biodata ===
Database connected successfully
Isi data baru (Enter untuk tidak mengubah / batal jika kosong di awal):
Nama (Yaya):
Username (yaya):
Kata Sandi (333):
No Telp (081012345678901):
Alamat belum ada, silakan isi:
Nama Jalan [Enter untuk batal]: kaliurang
ID Desa [Enter untuk batal]: 3

--- Konfirmasi Data ---
Nama      : Yaya
Username  : yaya
Kata Sandi: 333
No Telp   : 081012345678901
Apakah data sudah benar? (y/n): y
Biodata berhasil diperbarui.

Tekan Enter untuk melanjutkan...
```

Gambar 4.29 Tampilan Fitur Edit Biodata (Pelanggan)

Fitur ini digunakan oleh pelanggan untuk mengubah biodata pribadi di sistem. Saat dijalankan, sistem akan menampilkan data lama dan meminta input baru. Jika pelanggan tidak ingin mengubah, cukup tekan Enter agar data lama tetap digunakan.

Jika alamat sudah ada, sistem akan memperbarui nama jalan dan desa sesuai input. Jika belum ada, sistem akan meminta pelanggan mengisi alamat baru, lalu menyimpan ke database. Setelah itu, semua data biodata akan diperbarui dan disimpan.

Langkah penggunaannya cukup dengan menjalankan fungsi dan mengisi data sesuai instruksi. Sistem akan menampilkan pesan bahwa biodata berhasil diperbarui. Fitur ini membantu pelanggan menjaga agar data pribadinya tetap lengkap dan terbaru.

## 4.30 Fitur Pilih Produk

```

=== Pilih Produk ===
Database connected successfully

```

ID Produk	Jenis Produk	Stok	Harga (Rp)	Update Stok
5	frozen	150	75000	2025-12-12
6	jus	180	35000	2025-11-28
7	fresh	104	50000	2025-12-04
8	fresh	300	50000	2025-12-05
9	fresh	90	50000	2025-12-06
10	frozen	270	75000	2025-12-13

```

Masukkan ID Produk [Enter untuk batal]: 8
Masukkan jumlah [Enter untuk batal]: 10

--- Konfirmasi Produk ---
ID Produk   : 8
Jumlah      : 10
Harga Satuan: 50000
Total Harga : 500000
Apakah data sudah benar? (y/n): y
Produk 8 x10 berhasil ditambahkan ke keranjang.
Tambah produk lain? (y/n): n
Selesai memilih produk. Semua pilihan sudah masuk ke keranjang.

Tekan Enter untuk melanjutkan...

```

Gambar 4.30 Tampilan Fitur Pilih Produk

Fitur ini digunakan oleh pelanggan untuk memilih produk dan menambahkannya ke keranjang belanja. Sistem akan mengecek apakah pelanggan sudah punya transaksi dengan status “proses”. Jika belum, sistem akan membuat transaksi baru secara otomatis.

Setelah itu, sistem menampilkan daftar produk lengkap dengan stok, harga, dan tanggal update. Pelanggan bisa memilih produk berdasarkan ID dan jumlah yang diinginkan. Jika stok cukup, produk akan ditambahkan ke keranjang dan disimpan ke database.

Langkah penggunaannya cukup dengan memilih produk dan jumlahnya, lalu sistem akan memproses dan menanyakan apakah ingin menambah produk lain. Fitur ini membantu pelanggan belanja dengan mudah dan memastikan semua pilihan masuk ke keranjang.

## 4.31 Fitur Lihat Keranjang

```
=== Lihat Keranjang ===
Database connected successfully

=== Isi Keranjang ===
```

Produk	Jumlah	Subtotal (Rp)
fresh	10	500000

```
Total Keranjang: Rp 500000
Metode pembayaran belum dipilih.
Tekan Enter untuk melanjutkan...
```

Gambar 4.31 Tampilan Fitur Lihat Keranjang

Fitur ini digunakan oleh pelanggan untuk melihat isi keranjang belanja yang sedang aktif. Sistem akan mencari transaksi dengan status “proses” milik pelanggan, lalu menampilkan daftar produk yang sudah dipilih beserta jumlah dan subtotalnya.

Jika keranjang berisi produk, sistem akan menampilkan tabel rapi berisi nama produk, jumlah, dan subtotal. Di bawahnya, sistem juga menampilkan total belanja dan status metode pembayaran. Jika belum dipilih, akan muncul keterangan bahwa metode pembayaran belum tersedia.

Langkah penggunaannya cukup dengan menjalankan fungsi, lalu sistem otomatis menampilkan isi keranjang. Fitur ini membantu pelanggan mengecek belanjaan sebelum melanjutkan ke proses checkout.

## 4.32 Fitur Edit Keranjang

```
=== Edit Keranjang ===
Database connected successfully
```

ID Detail	Produk	Jumlah	Subtotal (Rp)
6	fresh	10	500000

```
Masukkan ID Detail yang ingin diubah/hapus [Enter untuk batal]: 6
Ketik 'ubah' untuk ubah jumlah, 'hapus' untuk hapus produk [Enter untuk batal]: ubah
Masukkan jumlah baru [Enter untuk batal]: 5

--- Konfirmasi Ubah ---
ID Detail : 6
Jumlah Baru: 5
Subtotal   : 250000
Apakah data sudah benar? (y/n): y
Jumlah produk berhasil diubah.

Tekan Enter untuk melanjutkan...
```

Gambar 4.32 Tampilan Fitur Edit Keranjang

FFitur ini digunakan oleh pelanggan untuk mengubah atau menghapus isi keranjang belanja yang masih aktif. Sistem akan mencari transaksi dengan status “proses”, lalu menampilkan daftar produk yang sudah dipilih lengkap dengan ID detail, nama produk, jumlah, dan subtotal.

Pelanggan bisa memilih apakah ingin mengubah jumlah produk atau menghapusnya dari keranjang. Jika memilih “ubah”, sistem akan menghitung ulang subtotal berdasarkan jumlah baru. Jika memilih “hapus”, produk akan dihapus dari keranjang. Semua perubahan langsung disimpan ke database.

Langkah penggunaannya cukup dengan memilih ID detail produk, lalu menentukan aksi yang diinginkan. Fitur ini membantu pelanggan mengatur isi keranjang sebelum melanjutkan ke proses checkout.

### 4.33 Fitur Pilih Metode Pembayaran

```
=== Pilih Metode Pembayaran ===
Database connected successfully

=== Daftar Metode Pembayaran ===
1. tunai
2. qris
3. transfer
Masukkan ID Metode Pembayaran [Enter untuk batal]: 2

--- Konfirmasi Metode Pembayaran ---
ID : 2
Nama : qris
Apakah data sudah benar? (y/n): y
Metode pembayaran berhasil dipilih.

Tekan Enter untuk melanjutkan...
```

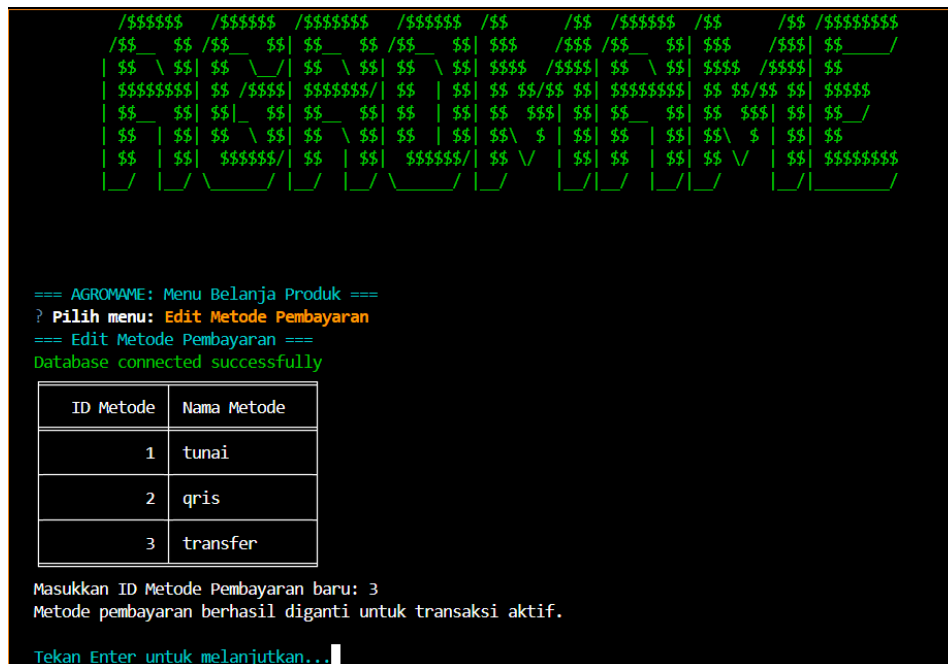
Gambar 4.33 Tampilan Fitur Pilih Metode Pembayaran

Fitur ini digunakan oleh pelanggan untuk memilih metode pembayaran dalam transaksi yang sedang berlangsung. Sistem akan mencari transaksi dengan status “proses” milik pelanggan, lalu menampilkan daftar metode pembayaran yang tersedia seperti tunai, QRIS, atau transfer.

Pelanggan cukup memilih ID metode pembayaran sesuai daftar. Sistem akan mengecek apakah pilihan valid, lalu menyimpan metode tersebut ke transaksi aktif. Jika belum ada transaksi, sistem akan memberi peringatan agar pelanggan memilih produk terlebih dahulu.

Langkah penggunaannya cukup dengan menjalankan fungsi dan memilih metode yang diinginkan. Fitur ini membantu pelanggan menyelesaikan proses belanja dengan metode pembayaran yang sesuai.

### 4.34 Fitur Edit Metode Pembayaran



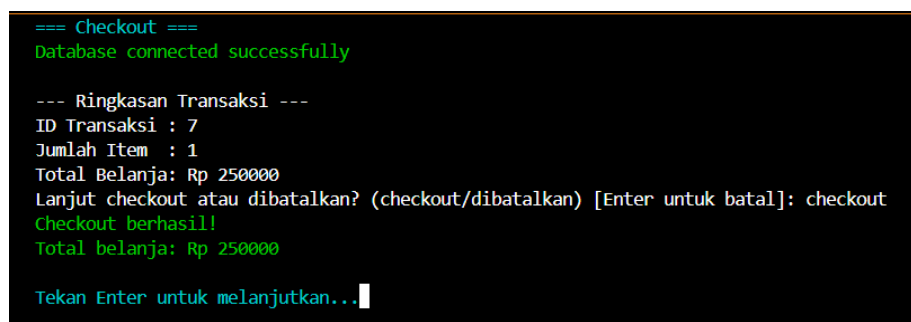
Gambar 4.34 Tampilan Fitur Edit Metode Pembayaran

Fitur ini digunakan oleh pelanggan untuk mengganti metode pembayaran pada transaksi yang masih aktif. Sistem akan mencari transaksi dengan status “proses”, lalu menampilkan daftar metode pembayaran yang tersedia dalam bentuk tabel.

Pelanggan cukup memilih ID metode baru yang diinginkan. Sistem akan mengecek apakah ID valid, lalu memperbarui metode pembayaran di transaksi tersebut. Jika tidak ada transaksi aktif, sistem akan memberi peringatan.

Langkah penggunaannya cukup dengan menjalankan fungsi dan memilih metode baru. Fitur ini membantu pelanggan mengubah pilihan pembayaran sebelum menyelesaikan proses checkout.

### 4.35 Fitur Checkout



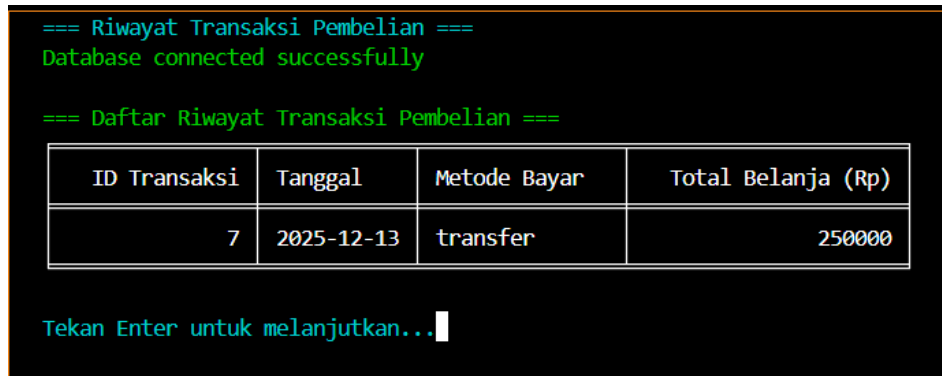
Gambar 4.35 Tampilan Fitur Checkout

Fitur ini digunakan oleh pelanggan untuk menyelesaikan transaksi belanja. Sistem akan mencari transaksi aktif dengan status “proses”, lalu mengecek isi keranjang dan stok produk. Jika semua produk tersedia, sistem akan memproses checkout dan mengubah status transaksi menjadi “selesai”.

Sistem juga akan mengurangi stok produk dan stok total jenis produk sesuai jumlah yang dibeli. Jika ada produk yang stoknya tidak cukup, checkout akan dibatalkan dan sistem menampilkan peringatan. Setelah berhasil, sistem akan menampilkan total belanja pelanggan.

Langkah penggunaannya cukup dengan menjalankan fungsi, dan sistem akan otomatis memproses transaksi aktif. Fitur ini memastikan bahwa transaksi pelanggan tercatat dengan benar dan stok produk selalu terupdate.

## 4.36 Fitur Lihat Riwayat Transaksi Pembelian



```
=== Riwayat Transaksi Pembelian ===
Database connected successfully

=== Daftar Riwayat Transaksi Pembelian ===
```

ID Transaksi	Tanggal	Metode Bayar	Total Belanja (Rp)
7	2025-12-13	transfer	250000

```
Tekan Enter untuk melanjutkan...
```

Gambar 4.36 Tampilan Fitur Lihat Riwayat Transaksi Pembelian

Fitur ini digunakan oleh pelanggan untuk melihat daftar transaksi yang sudah selesai. Sistem akan mencari semua transaksi dengan status “selesai” milik pelanggan, lalu menampilkan informasi seperti ID transaksi, tanggal, metode pembayaran, dan total belanja.

Data ditampilkan dalam bentuk tabel rapi menggunakan tabulate, sehingga pelanggan bisa melihat riwayat belanja dengan jelas. Jika belum ada transaksi yang selesai, sistem akan menampilkan pesan bahwa data belum tersedia.

Langkah penggunaannya cukup dengan menjalankan fungsi, lalu sistem otomatis menampilkan riwayat transaksi. Fitur ini membantu pelanggan memantau histori pembelian dan mencatat pengeluaran dengan mudah.

# DAFTAR PUSTAKA

- Kartahadimaja, J., Wentasari, R., & Sesanti, R.N. (2010). *Pertumbuhan dan Produksi Polong Segar Edamame Varietas Rioko pada Empat Jenis Pupuk*. Agrovigor: Jurnal Agroekoteknologi, 3(2), 131–136.
- Anggraini, Y., Pasha, D., & Setiawan, A. (2020). *Sistem Informasi Penjualan Sepeda Berbasis Web Menggunakan Framework CodeIgniter (Studi Kasus: Orbit Station)*. Jurnal Teknologi dan Sistem Informasi (JTISI), 1(2), 64–70.
- Yuwariah, Y., Hindersah, R., & Simarmata, T. (2018). *Pengaruh Pemupukan Organik terhadap Produktivitas Edamame di Lahan Tropis*. Jurnal Agronomi Indonesia, 46(1), 45–52.

# LAMPIRAN

Link Github : <https://github.com/rizkaan17/Projek-Algo>