

**LAPORAN IMPLEMENTASI SISTEM PENJUALAN
DAN PEMBUATAN STRUK TRANSAKSI
PADA DATABASE PENJUALAN**

T. A. Semester Ganjil 2024/2025



5230411285

OKTABERLIAN DIVA GRACIELA

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA**

2024

PENDAHULUAN

Pada laporan ini, akan dijelaskan pembuatan aplikasi pengelolaan data menggunakan Python dan MySQL sebagai basis data. Aplikasi ini bertujuan untuk membantu pengelolaan data pegawai, produk, transaksi, dan struk pada sebuah sistem penjualan. Sistem ini dirancang dengan struktur relasi antar tabel yang saling terhubung untuk mempermudah pengolahan data yang terintegrasi.

Tujuan dari program ini adalah untuk:

1. Membuat database dan tabel yang sesuai dengan kebutuhan pengelolaan data penjualan.
2. Mengimplementasikan fungsi input, penghapusan, dan penampilan data untuk entitas Pegawai, Produk, Transaksi, dan Struk.
3. Memastikan data yang dimasukkan terintegrasi antar tabel melalui relasi yang terdefinisi dengan baik.

PEMBAHASAN

1. Desain Database Sistem database terdiri dari beberapa tabel utama, yaitu:

- Pegawai: Berisi informasi pegawai, seperti NIK, Nama, dan Alamat.
- Produk: Menyimpan data produk seperti Kode Produk, Nama Produk, Jenis Produk, dan Harga.
- Transaksi: Mencatat transaksi yang dilakukan oleh pegawai, termasuk detail transaksi.
- Transaksi_Produk: Menghubungkan transaksi dengan produk, mencatat jumlah produk yang dibeli.
- Struk: Menyimpan informasi struk transaksi, termasuk total harga.

Kode pembuatan tabel dituliskan sebagai berikut:

```
import mysql.connector
try:
    conn = mysql.connector.connect(
        user="root",
        host="localhost",
        password="",
    )
    cur = conn.cursor()
except mysql.connector.Error as err:
    print(f"Error: {err}")
    exit(1)

# Membuat database jika belum ada
cur.execute("CREATE DATABASE IF NOT EXISTS penjualan")
cur.execute("USE penjualan")

# Membuat Tabel Pegawai
cur.execute("""
CREATE TABLE IF NOT EXISTS Pegawai (
    NIK CHAR(4) NOT NULL PRIMARY KEY,
    Nama VARCHAR(50),
    Alamat VARCHAR(255)
)
""")

# Membuat Tabel Produk dengan kolom Harga
cur.execute("""
CREATE TABLE IF NOT EXISTS Produk (
    Kode_Produk CHAR(4) NOT NULL PRIMARY KEY,
```

```

    Nama_Produk VARCHAR(50),
    Jenis_Produk VARCHAR(20),
    Harga INT
)
)
)

# Membuat Tabel Transaksi
cur.execute("""
CREATE TABLE IF NOT EXISTS Transaksi (
    No_Transaksi CHAR(4) NOT NULL PRIMARY KEY,
    NIK CHAR(4) NOT NULL,
    Detail_Transaksi VARCHAR(255),
    FOREIGN KEY (NIK) REFERENCES Pegawai(NIK)
)
)

# Membuat Tabel Relasi Transaksi dengan Produk
cur.execute("""
CREATE TABLE IF NOT EXISTS Transaksi_Produk (
    No_Transaksi CHAR(4),
    Kode_Produk CHAR(4),
    Jumlah_Produk INT,
    PRIMARY KEY (No_Transaksi, Kode_Produk),
    FOREIGN KEY (No_Transaksi) REFERENCES Transaksi(No_Transaksi),
    FOREIGN KEY (Kode_Produk) REFERENCES Produk(Kode_Produk)
)
)

# Membuat Tabel Struk
cur.execute("""
CREATE TABLE IF NOT EXISTS Struk (
    No_Transaksi CHAR(4) NOT NULL PRIMARY KEY,
    Nama_Pegawai VARCHAR(50),
    Total_Harga DECIMAL(10, 2),
    FOREIGN KEY (No_Transaksi) REFERENCES Transaksi(No_Transaksi)
)
)

print("Database, tabel, dan relasi telah dibuat sesuai dengan kebutuhan.")

```

2. Implementasi Fungsi Berbagai fungsi dikembangkan untuk mendukung pengelolaan data:

- Tampilkan dan Hapus Data: Semua entitas (Pegawai, Produk, Transaksi, dan Struk) memiliki fungsi untuk menampilkan dan menghapus data sesuai kebutuhan.
- Input Data Pegawai: Memasukkan data pegawai baru ke tabel Pegawai.

```
def input_data_pegawai():
    try:
        nik = input("Masukkan NIK: ")
        nama = input("Masukkan Nama: ")
        alamat = input("Masukkan Alamat: ")
        cur.execute("INSERT INTO Pegawai (NIK, Nama, Alamat) VALUES (%s, %s, %s)",
(nik, nama, alamat))
        conn.commit()
        print("Data Pegawai berhasil ditambahkan.")
    except mysql.connector.Error as err:
        print(f"Error saat menambahkan data Pegawai: {err}")

def tampil_data_pegawai():
    try:
        cur.execute("SELECT * FROM Pegawai")
        result = cur.fetchall()
        print("\nData Pegawai:")
        for row in result:
            print(f"NIK: {row[0]}, Nama: {row[1]}, Alamat: {row[2]}")
    except mysql.connector.Error as err:
        print(f"Error saat menampilkan data Pegawai: {err}")

def hapus_data_pegawai():
    try:
        nik = input("Masukkan NIK Pegawai yang akan dihapus: ")
        cur.execute("DELETE FROM Pegawai WHERE NIK = %s", (nik,))
        conn.commit()
        print("Data Pegawai berhasil dihapus.")
    except mysql.connector.Error as err:
        print(f"Error saat menghapus data Pegawai: {err}")
```

- Input Data Produk: Menambahkan data produk baru ke tabel Produk.

```
def input_data_produk():
    try:
        kode_produk = input("Masukkan Kode Produk: ")
        nama_produk = input("Masukkan Nama Produk: ")
        jenis_produk = input("Masukkan Jenis Produk: ")
        harga = int(input("Masukkan Harga Produk: "))
```

```

        cur.execute("INSERT INTO Produk (Kode_Produk, Nama_Produk, Jenis_Produk,
        Harga) VALUES (%s, %s, %s, %s)",
            (kode_produk, nama_produk, jenis_produk, harga))
        conn.commit()
        print("Data Produk berhasil ditambahkan.")
    except ValueError:
        print("Input harga harus berupa angka.")
    except mysql.connector.Error as err:
        print(f"Error saat menambahkan data Produk: {err}")

def tampil_data_produk():
    try:
        cur.execute("SELECT * FROM Produk")
        result = cur.fetchall()
        print("\nData Produk:")
        for row in result:
            print(f"Kode Produk: {row[0]}, Nama: {row[1]}, Jenis: {row[2]}, Harga: Rp.
            {row[3]:,}")
    except mysql.connector.Error as err:
        print(f"Error saat menampilkan data Produk: {err}")

def hapus_data_produk():
    try:
        kode_produk = input("Masukkan Kode Produk yang akan dihapus: ")
        cur.execute("DELETE FROM Produk WHERE Kode_Produk = %s", (kode_produk,))
        conn.commit()
        print("Data Produk berhasil dihapus.")
    except mysql.connector.Error as err:
        print(f"Error saat menghapus data Produk: {err}")

```

- Input Data Transaksi: Mencatat transaksi yang dilakukan oleh pegawai dan menghubungkannya dengan produk yang dibeli.

```

def input_data_transaksi():
    try:
        no_transaksi = input("Masukkan No Transaksi: ")
        nik_pegawai = input("Masukkan NIK Pegawai: ")

        cur.execute("SELECT * FROM Pegawai WHERE NIK = %s", (nik_pegawai,))
        pegawai = cur.fetchone()

        if pegawai is None:
            print("Error: NIK Pegawai tidak ditemukan.")
            return

        print("\nDaftar Produk yang Tersedia:")
        cur.execute("SELECT * FROM Produk")
        result_produk = cur.fetchall()

```

```

for row in result_produk:
    print(f"Kode Produk: {row[0]}, Nama: {row[1]}, Jenis: {row[2]}, Harga: Rp.
{row[3]:.2f}")

produk_beli = []
while True:
    kode_produk = input("Masukkan Kode Produk yang dibeli (atau ketik 'selesai'
untuk selesai): ")
    if kode_produk == 'selesai':
        break

    cur.execute("SELECT * FROM Produk WHERE Kode_Produk = %s",
(kode_produk,))
    produk = cur.fetchone()

    if produk is None:
        print("Error: Kode Produk tidak ditemukan.")
        continue

    jumlah = int(input(f"Masukkan Jumlah Produk {produk[1]}: "))
    produk_beli.append((kode_produk, jumlah))

total_harga = 0
for kode_produk, jumlah in produk_beli:
    cur.execute("SELECT Harga FROM Produk WHERE Kode_Produk = %s",
(kode_produk,))
    harga_produk = cur.fetchone()[0]
    total_harga += harga_produk * jumlah

cur.execute(
    "INSERT INTO Transaksi (No_Transaksi, NIK, Detail_Transaksi) VALUES (%s,
%s, %s)",
    (no_transaksi, nik_pegawai, f"Pembelian produk sejumlah {len(produk_beli)}
item.")
)

for kode_produk, jumlah in produk_beli:
    cur.execute(
        "INSERT INTO Transaksi_Produk (No_Transaksi, Kode_Produk,
Jumlah_Produk) VALUES (%s, %s, %s)",
        (no_transaksi, kode_produk, jumlah)
    )

conn.commit()
print("Data Transaksi dan Produk berhasil ditambahkan.")

except mysql.connector.Error as err:
    print(f"Error saat menambahkan data Transaksi: {err}")

def tampil_data_transaksi():
    try:
        cur.execute("SELECT * FROM Transaksi")
        result = cur.fetchall()
        print("\nData Transaksi:")

```

```

    if result:
        for row in result:
            print(f"No Transaksi: {row[0]}, NIK: {row[1]}, Detail: {row[2]}")
        else:
            print("Tidak ada data transaksi.")
    except mysql.connector.Error as err:
        print(f"Terjadi kesalahan saat mengambil data transaksi: {err}")

def hapus_data_transaksi():
    try:
        no_transaksi = input("Masukkan No Transaksi yang akan dihapus: ")
        cur.execute("DELETE FROM Transaksi WHERE No_Transaksi = %s",
        (no_transaksi,))
        conn.commit()

        if cur.rowcount > 0:
            print("Data Transaksi berhasil dihapus.")
        else:
            print("No Transaksi tidak ditemukan.")
    except mysql.connector.Error as err:
        print(f"Terjadi kesalahan saat menghapus data transaksi: {err}")

```

Fungsi ini juga mencakup penghitungan total harga berdasarkan produk yang dibeli.

3. Antarmuka Menu Program menyediakan menu utama dan submenu untuk mempermudah pengguna memilih operasi yang diinginkan.

```
while True:
    print("=====Menu Utama=====")
    print("1. Kelola Pegawai")
    print("2. Kelola Produk")
    print("3. Kelola Transaksi")
    print("4. Kelola Struk")
    print("0. Keluar")
    pilihan = input("Pilih menu: ")

    if pilihan == "1":
        print("=====Kelola Pegawai=====")
        print("1. Input Data")
        print("2. Tampilkan Data")
        print("3. Hapus Data")
        sub_pilihan = input("Pilih submenu: ")
        if sub_pilihan == "1":
            input_data_pegawai()
        elif sub_pilihan == "2":
            tampil_data_pegawai()
        elif sub_pilihan == "3":
            hapus_data_pegawai()

    elif pilihan == "2":
        print("=====Kelola Produk=====")
        print("1. Input Data")
        print("2. Tampilkan Data")
        print("3. Hapus Data")
        sub_pilihan = input("Pilih submenu: ")
        if sub_pilihan == "1":
            input_data_produk()
        elif sub_pilihan == "2":
            tampil_data_produk()
        elif sub_pilihan == "3":
            hapus_data_produk()

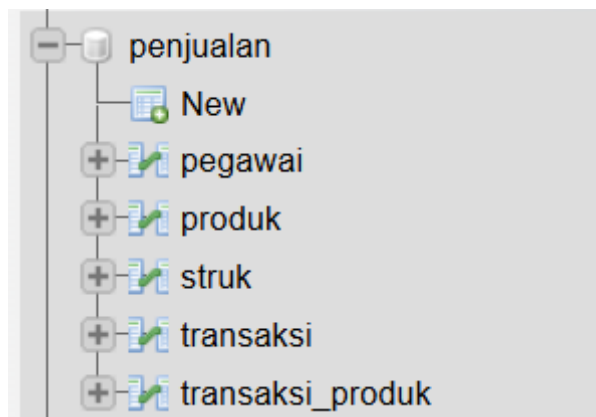
    elif pilihan == "3":
        print("=====Kelola Transaksi=====")
        print("1. Input Data")
        print("2. Tampilkan Data")
        print("3. Hapus Data")
        sub_pilihan = input("Pilih submenu: ")
        if sub_pilihan == "1":
            input_data_transaksi()
        elif sub_pilihan == "2":
            tampil_data_transaksi()
        elif sub_pilihan == "3":
            hapus_data_transaksi()
```

```
elif pilihan == "4":
    print("====Kelola Struk====")
    print("1. Input Data")
    print("2. Tampilkan Data")
    print("3. Hapus Data")
    sub_pilihan = input("Pilih submenu: ")
    if sub_pilihan == "1":
        input_data_struk()
    elif sub_pilihan == "2":
        tampil_data_struk()
    elif sub_pilihan == "3":
        hapus_data_struk()

elif pilihan == "0":
    print("Keluar dari program.")
    break

else:
    print("Pilihan tidak valid. Silakan coba lagi.")
```

HASIL IMPLEMENTASI



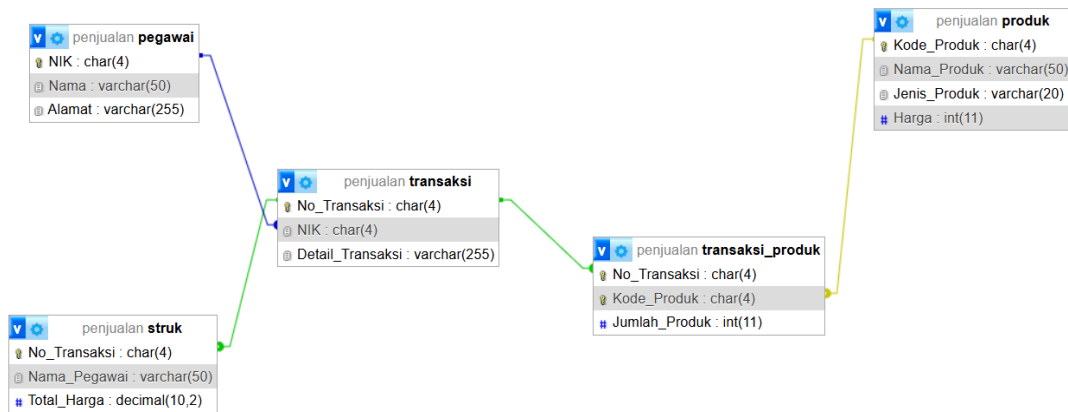
Gambar 1. Struktur Tabel di Database

1. Database penjualan:

- Memiliki lima tabel utama:
 1. pegawai
 2. produk
 3. struk
 4. transaksi
 5. transaksi_produk
- Tiap tabel sudah diatur relasinya sesuai dengan ERD.

2. Penjelasan Urutan Data:

- Langkah Insert Data:
 1. Tambahkan data pegawai (pegawai).
 2. Tambahkan data produk (produk).
 3. Catat transaksi baru di tabel transaksi.
 4. Isi detail transaksi (produk yang dibeli dan jumlahnya) di tabel transaksi_produk.
 5. Simpan total harga transaksi di tabel struk.



Gambar 2. ERD (Entity-Relationship Diagram)

1. Tabel pegawai:

○ Kolom:

- NIK (char(4)): Primary key yang merepresentasikan Nomor Induk Karyawan.
- Nama (varchar(50)): Nama karyawan.
- Alamat (varchar(255)): Alamat karyawan.

○ Relasi:

- Berelasi dengan tabel transaksi. Artinya, setiap transaksi dicatat atas nama seorang pegawai.

2. Tabel produk:

○ Kolom:

- Kode_Produk (char(4)): Primary key yang menjadi identitas produk.
- Nama_Produk (varchar(50)): Nama produk.
- Jenis_Produk (varchar(20)): Kategori produk.
- Harga (int): Harga satuan dari produk.

- Relasi:
 - Berelasi dengan tabel transaksi_produk, yang menghubungkan transaksi dengan produk yang terjual.

3. Tabel transaksi:

- Kolom:
 - No_Transaksi (char(4)): Primary key yang menjadi identitas transaksi.
 - NIK (char(4)): Foreign key yang menghubungkan ke tabel pegawai.
 - Detail_Transaksi (varchar(255)): Deskripsi atau rincian transaksi.
- Relasi:
 - Berelasi ke tabel pegawai dan tabel transaksi_produk.

4. Tabel transaksi_produk:

- Kolom:
 - No_Transaksi (char(4)): Foreign key yang menghubungkan ke tabel transaksi.
 - Kode_Produk (char(4)): Foreign key yang menghubungkan ke tabel produk.
 - Jumlah_Produk (int): Jumlah produk yang dibeli dalam transaksi.
- Relasi:
 - Menghubungkan antara transaksi dan produk.

5. Tabel struk:

- Kolom:
 - No_Transaksi (char(4)): Primary key sekaligus foreign key yang menghubungkan ke tabel transaksi.
 - Nama_Pegawai (varchar(50)): Nama pegawai yang melayani transaksi (redundan dari tabel pegawai).

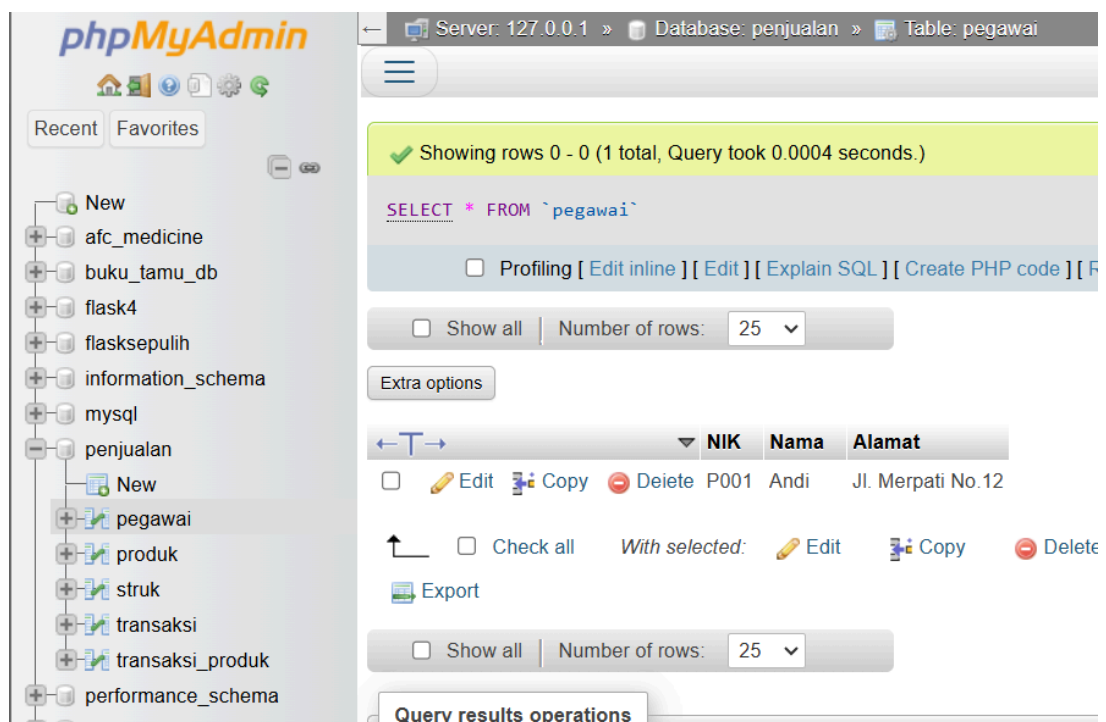
- Total_Harga (decimal(10,2)): Total harga dari semua produk dalam transaksi.
- Relasi:
 - Berelasi ke tabel transaksi.

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 1
=====Kelola Pegawai=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 1
Masukkan NIK: P001
Masukkan Nama: Andi
Masukkan Alamat: Jl. Merpati No.12
Data Pegawai berhasil ditambahkan.
```

Gambar 3. Menu Utama dan Input Data Pegawai

- Program menampilkan menu utama dengan beberapa opsi seperti:
 - Kelola Pegawai
 - Kelola Produk
 - Kelola Transaksi
 - Kelola Struk
 - Keluar

- Ketika pengguna memilih opsi "Kelola Pegawai" (pilihan menu 1), program menampilkan sub-menu untuk Kelola Pegawai, yaitu:
 - Input Data
 - Tampilkan Data
 - Hapus Data
- Pengguna memilih submenu 1, yaitu Input Data.
- Sistem meminta input dari pengguna:
 - NIK (P001)
 - Nama (Andi)
 - Alamat (Jl. Merpati No.12)
- Setelah data dimasukkan, program memberikan pesan: *"Data Pegawai berhasil ditambahkan."*



Gambar 4. Data yang Masuk pada Tabel Pegawai

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 1
=====Kelola Pegawai=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 2

Data Pegawai:
NIK: P001, Nama: Andi, Alamat: Jl. Merpati No.12
```

Gambar 5. Proses Menampilkan Data Pegawai

1. Menu Utama:

- Sistem menampilkan daftar menu utama dengan beberapa opsi, yaitu:
 - 1. Kelola Pegawai
 - 2. Kelola Produk
 - 3. Kelola Transaksi
 - 4. Kelola Struk
 - 5. Keluar
- Pengguna memilih opsi 1 (Kelola Pegawai) untuk masuk ke menu pengelolaan data pegawai.

2. Submenu Kelola Pegawai:

- Setelah memilih menu Kelola Pegawai, sistem menampilkan tiga submenu:
 - 1. Input Data: Untuk menambahkan data pegawai baru.

- 2. Tampilkan Data: Untuk menampilkan daftar pegawai yang sudah ada.
- 3. Hapus Data: Untuk menghapus data pegawai yang tidak lagi diperlukan.
- Pengguna memilih opsi 2 (Tampilkan Data) untuk melihat data pegawai.

3. Output Data Pegawai:

- Sistem memproses pilihan dan menampilkan data pegawai yang tersedia, seperti:
 - NIK: P001
 - Nama: Andi
 - Alamat: Jl. Merpati No.12

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 1
=====Kelola Pegawai=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 3
Masukkan NIK Pegawai yang akan dihapus: P001
Data Pegawai berhasil dihapus.
```

Gambar 6. Proses Menghapus Data Pegawai

1. Submenu Kelola Pegawai:
 - Setelah memilih opsi 1. Kelola Pegawai dari menu utama, sistem menampilkan tiga submenu:
 - 1. Input Data
 - 2. Tampilkan Data
 - 3. Hapus Data
 - Pengguna memilih opsi 3 (Hapus Data) untuk menghapus data pegawai.
2. Permintaan Input NIK:
 - Sistem meminta pengguna untuk memasukkan NIK (Nomor Induk Karyawan) dari pegawai yang ingin dihapus.
 - Pesan yang ditampilkan:
 - "Masukkan NIK pegawai yang akan dihapus:"
3. Input NIK:

- Pengguna mengetikkan P001, yang merupakan NIK dari pegawai yang akan dihapus.

4. Konfirmasi Penghapusan:

- Sistem memproses input NIK, mencocokkannya dengan data yang ada, dan menghapus data pegawai dengan NIK tersebut dari database.
- Setelah penghapusan berhasil, sistem menampilkan pesan konfirmasi:
 - "Data Pegawai berhasil dihapus."

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 2
=====Kelola Produk=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 1
Masukkan Kode Produk: PR001
Masukkan Nama Produk: Nasi Goreng
Masukkan Jenis Produk: Makanan
Masukkan Harga Produk: 25000
Data Produk berhasil ditambahkan.
```

Gambar 7. Proses Menambahkan Data Produk

Submenu Kelola Produk:

Setelah memilih opsi 2. Kelola Produk dari menu utama, sistem menampilkan tiga submenu:

1. Input Data
2. Tampilkan Data
3. Hapus Data

Pengguna memilih opsi 1 (Input Data) untuk menambahkan data produk baru.

Proses Input Data Produk:

Sistem meminta pengguna untuk memasukkan data produk dengan format berikut:

- Kode Produk

- Nama Produk
- Jenis Produk
- Harga Produk

Input Data:

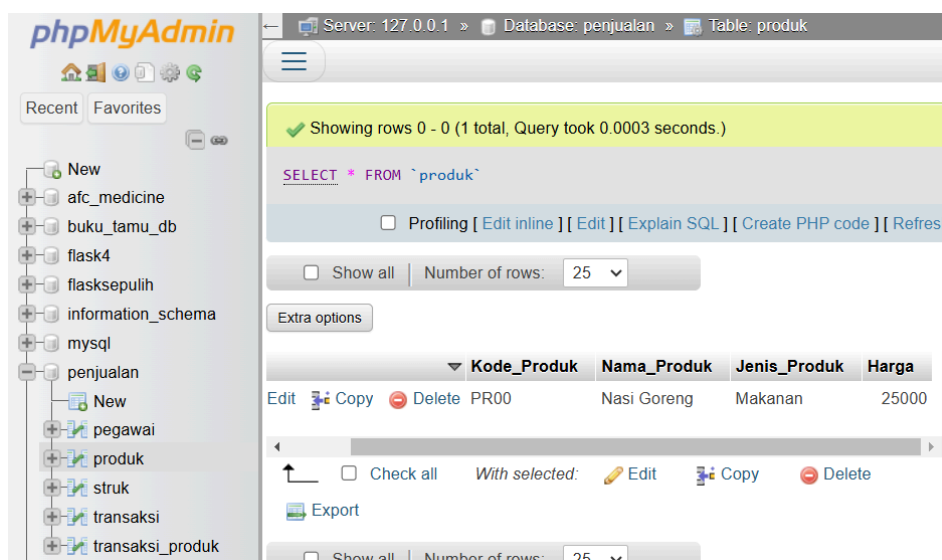
Pengguna mengetikkan informasi berikut:

- Kode Produk: PR001
- Nama Produk: Nasi Goreng
- Jenis Produk: Makanan
- Harga Produk: 25.000

Konfirmasi Penambahan Data:

Sistem memproses input data tersebut, menambahkannya ke dalam database, dan menampilkan pesan konfirmasi:

"Data Produk berhasil ditambahkan."



Gambar 8. Tampilan Data Produk di phpMyAdmin

```

=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 2
=====Kelola Produk=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 2

Data Produk:
Kode Produk: PR00, Nama: Nasi Goreng, Jenis: Makanan , Harga: Rp. 25,000

```

Gambar 9. Proses Menampilkan Data Produk

Submenu Kelola Produk:

Setelah memilih opsi 2. Kelola Produk dari menu utama, sistem menampilkan tiga submenu:

1. Input Data
2. Tampilkan Data
3. Hapus Data

Pengguna memilih opsi 2 (Tampilkan Data) untuk melihat data produk yang telah tersimpan dalam database.

Proses Menampilkan Data:

Setelah memilih submenu Tampilkan Data, sistem mengambil data dari tabel **produk** dalam database dan menampilkannya di layar terminal.

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 2
=====Kelola Produk=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 3
Masukkan Kode Produk yang akan dihapus: PR00
Data Produk berhasil dihapus.
```

Gambar 10. Proses Menghapus Data Produk

Submenu Kelola Produk:

Setelah memilih opsi 2. Kelola Produk dari menu utama, sistem menampilkan tiga submenu:

1. Input Data
2. Tampilkan Data
3. Hapus Data

Pengguna memilih opsi 3 (Hapus Data) untuk menghapus data produk yang ada di database.

Permintaan Input Kode Produk:

Sistem meminta pengguna untuk memasukkan Kode Produk dari produk yang ingin dihapus dengan menampilkan pesan:

“Masukkan Kode Produk yang akan dihapus:”

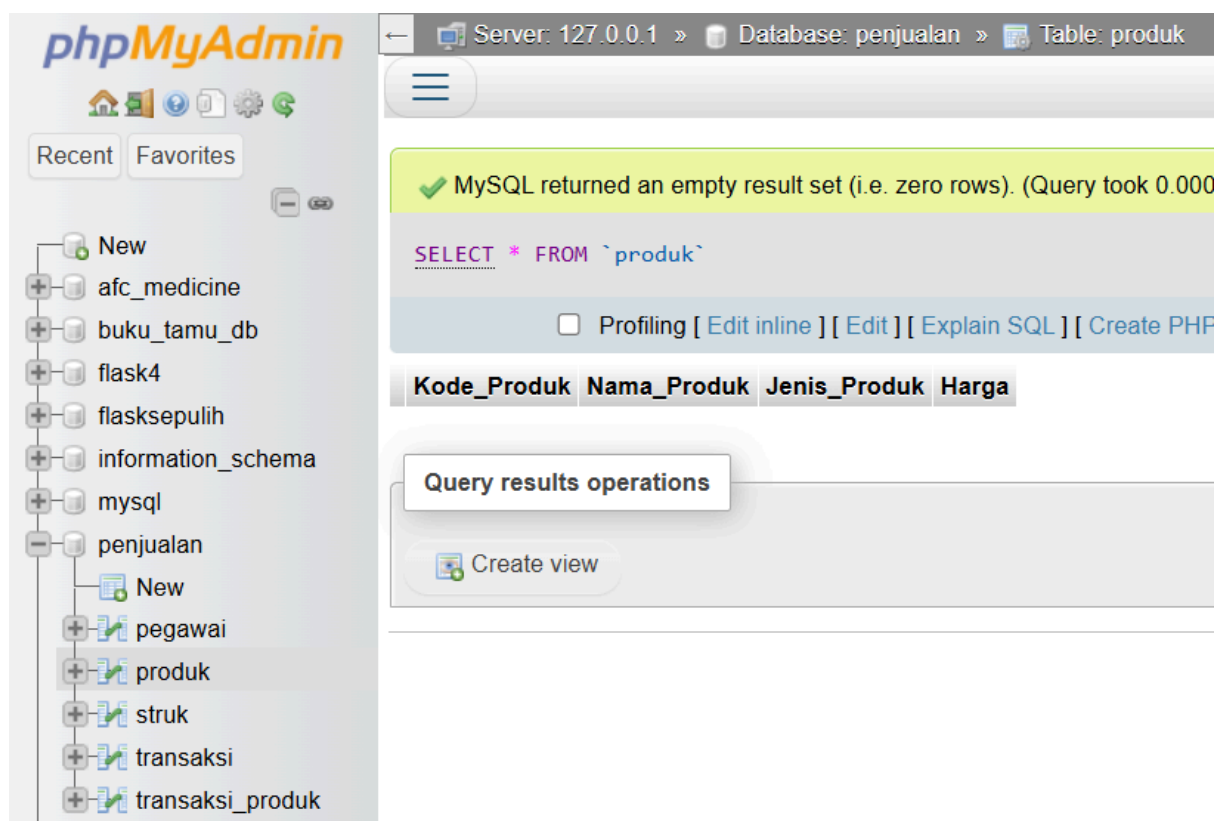
Input Kode Produk:

Pengguna mengetikkan kode produk PR00, yang merupakan kode dari produk yang ingin dihapus.

Konfirmasi Penghapusan Data:

Sistem mencocokkan kode produk yang diinputkan dengan data yang ada dalam database, menghapus data tersebut, dan menampilkan pesan konfirmasi:

“Data Produk berhasil dihapus.”



Gambar 11. Tampilan Data Produk di phpMyAdmin Setelah Data Dihapus


```

=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
0. Keluar
Pilih menu: 3
=====Kelola Transaksi=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 1
Masukkan No Transaksi: T010
Masukkan NIK Pegawai: P001

Daftar Produk yang Tersedia:
Kode Produk: PR01, Nama: Nasi Goreng, Jenis: Makanan, Harga: Rp. 25,000.00
Kode Produk: PR02, Nama: Ayam Geprek, Jenis: Makanan, Harga: Rp. 30,000.00
Kode Produk: PR03, Nama: Es Teh Manis, Jenis: Minuman, Harga: Rp. 5,000.00
Kode Produk: PR04, Nama: Kopi Susu, Jenis: Minuman, Harga: Rp. 15,000.00
Kode Produk: PR05, Nama: Keripik Singkong, Jenis: Snack, Harga: Rp. 10,000.00
Kode Produk: PR06, Nama: Roti Bakar, Jenis: Snack, Harga: Rp. 20,000.00
Masukkan Kode Produk yang dibeli (atau ketik 'selesai' untuk selesai): PR01
Masukkan Jumlah Produk Nasi Goreng: 5
Masukkan Kode Produk yang dibeli (atau ketik 'selesai' untuk selesai): P05
Error: Kode Produk tidak ditemukan.
Masukkan Kode Produk yang dibeli (atau ketik 'selesai' untuk selesai): 2
Error: Kode Produk tidak ditemukan.
Masukkan Kode Produk yang dibeli (atau ketik 'selesai' untuk selesai): selesai
Data Transaksi dan Produk berhasil ditambahkan.

```

Gambar 12. Proses Input Data Transaksi

Submenu Kelola Transaksi:

Setelah pengguna memilih opsi 3. Kelola Transaksi dari menu utama, sistem menampilkan tiga submenu:

1. Input Data
2. Tampilkan Data
3. Hapus Data

Pengguna memilih opsi 1 (Input Data) untuk menambahkan data transaksi baru ke dalam database.

Permintaan Input Data Transaksi:

Sistem meminta beberapa data untuk dimasukkan:

1. Nomor Transaksi: Pengguna memasukkan kode transaksi, contohnya *T010*.
2. NIK Pegawai: Pengguna memasukkan kode pegawai yang bertugas, contohnya *P001*.

Menampilkan Daftar Produk Tersedia:

Sistem menampilkan daftar produk yang dapat dipilih pengguna, termasuk Kode Produk, Nama, Jenis, dan Harga.

Input Kode Produk dan Jumlah:

1. Sistem meminta pengguna untuk memasukkan Kode Produk yang dibeli:
 - Pengguna mengetikkan kode produk, misalnya *PR01* untuk Nasi Goreng.
 - Sistem meminta jumlah produk yang dibeli, misalnya *5*.
2. Jika pengguna salah memasukkan kode produk (seperti *P05* yang tidak ada dalam daftar), sistem menampilkan pesan kesalahan:
 - "Error: Kode Produk tidak ditemukan."
3. Pengguna dapat mengetikkan *selesai* untuk mengakhiri input produk.

Konfirmasi Penambahan Data:

Setelah pengguna selesai memasukkan produk, sistem mencatat data transaksi berikut detail pembelian dan menampilkan pesan:

"Data Transaksi dan Produk berhasil ditambahkan."

```

=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
0. Keluar
Pilih menu: 3
=====Kelola Transaksi=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 2

Data Transaksi:
No Transaksi: T001, NIK: P001, Detail: Pembelian produk sejumlah 1 item.
No Transaksi: T002, NIK: P001, Detail: Pembelian produk sejumlah 1 item.
No Transaksi: T003, NIK: P002, Detail: Pembelian produk sejumlah 1 item.
No Transaksi: T005, NIK: P001, Detail: Pembelian produk sejumlah 1 item.
No Transaksi: T009, NIK: P001, Detail: Pembelian produk sejumlah 2 item.
No Transaksi: T010, NIK: P001, Detail: Pembelian produk sejumlah 1 item.

```

Gambar 13. Proses Menampilkan Data Transaksi

Submenu Kelola Transaksi:

Setelah memilih opsi 3. Kelola Transaksi dari menu utama, sistem menampilkan tiga submenu:

1. Input Data
2. Tampilkan Data
3. Hapus Data

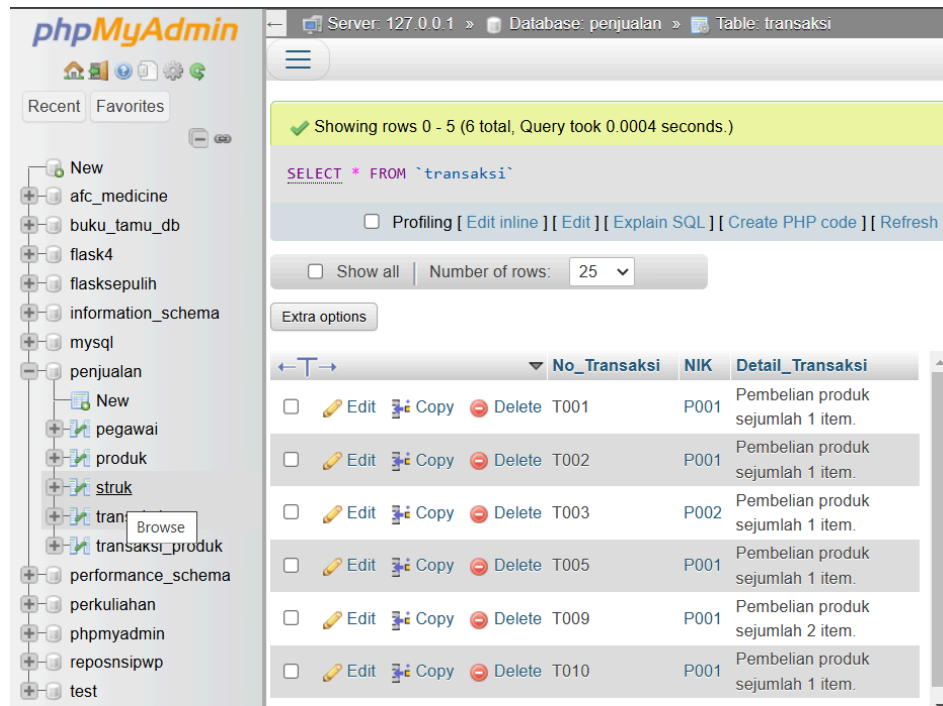
Pengguna memilih opsi 2 (Tampilkan Data) untuk melihat data transaksi yang tersimpan.

Menampilkan Data Transaksi:

Sistem menampilkan daftar semua transaksi yang ada, termasuk:

1. No Transaksi: Nomor transaksi seperti *T001* atau *T010*.

2. NIK Pegawai: Kode pegawai yang bertugas, seperti *P001* atau *P002*.
3. Detail Transaksi: Informasi mengenai jumlah item produk yang dibeli, misalnya "Pembelian produk sejumlah 1 item."



Server: 127.0.0.1 » Database: penjualan » Table: transaksi

Showing rows 0 - 5 (6 total, Query took 0.0004 seconds.)

`SELECT * FROM `transaksi``

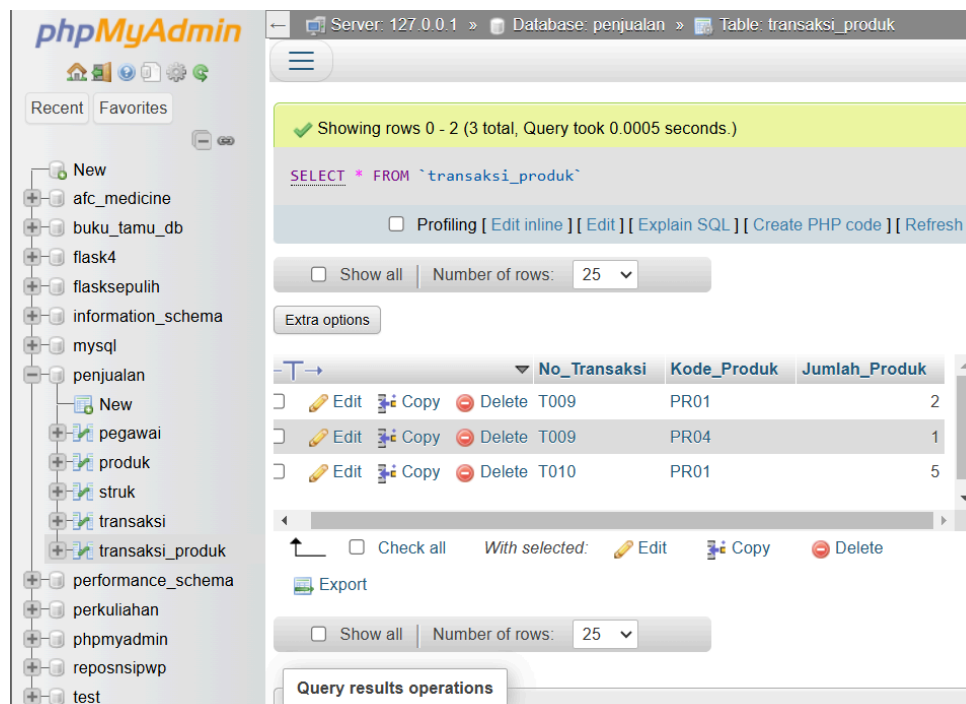
☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25

Extra options

	No_Transaksi	NIK	Detail_Transaksi
<input type="checkbox"/> Edit Copy Delete	T001	P001	Pembelian produk sejumlah 1 item.
<input type="checkbox"/> Edit Copy Delete	T002	P001	Pembelian produk sejumlah 1 item.
<input type="checkbox"/> Edit Copy Delete	T003	P002	Pembelian produk sejumlah 1 item.
<input type="checkbox"/> Edit Copy Delete	T005	P001	Pembelian produk sejumlah 1 item.
<input type="checkbox"/> Edit Copy Delete	T009	P001	Pembelian produk sejumlah 2 item.
<input type="checkbox"/> Edit Copy Delete	T010	P001	Pembelian produk sejumlah 1 item.

Gambar 14. Tampilan Data Transaksi di phpMyAdmin Setelah Data Ditambahkan ke Dalam Tabel Transaksi



Server: 127.0.0.1 » Database: penjualan » Table: transaksi_produk

Showing rows 0 - 2 (3 total, Query took 0.0005 seconds.)

`SELECT * FROM `transaksi_produk``

☐ Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

☐ Show all | Number of rows: 25

Extra options

	No_Transaksi	Kode_Produk	Jumlah_Produk
<input type="checkbox"/> Edit Copy Delete	T009	PR01	2
<input type="checkbox"/> Edit Copy Delete	T009	PR04	1
<input type="checkbox"/> Edit Copy Delete	T010	PR01	5

☐ Check all | With selected: [Edit](#) [Copy](#) [Delete](#)

[Export](#)

☐ Show all | Number of rows: 25

Query results operations

Gambar 15. Tampilan Data Transaksi di phpMyAdmin Setelah Data Ditambahkan Secara Otomatis ke Dalam Tabel Transaksi_Produk

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
0. Keluar
Pilih menu: 3
=====Kelola Transaksi=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 3
Masukkan No Transaksi yang akan dihapus: T001
Data Transaksi berhasil dihapus.
```

Gambar 16. Proses Menghapus Data Transaksi

Submenu Kelola Transaksi:

Setelah memilih opsi 3. Kelola Transaksi dari menu utama, sistem menampilkan tiga submenu:

1. Input Data
2. Tampilkan Data
3. Hapus Data

Pengguna memilih opsi 3 (Hapus Data) untuk menghapus data transaksi yang sudah tersimpan dalam database.

Permintaan Input Nomor Transaksi:

Sistem meminta pengguna untuk memasukkan Nomor Transaksi yang ingin dihapus dengan menampilkan pesan:

“Masukkan No Transaksi yang akan dihapus:”

Input Nomor Transaksi:

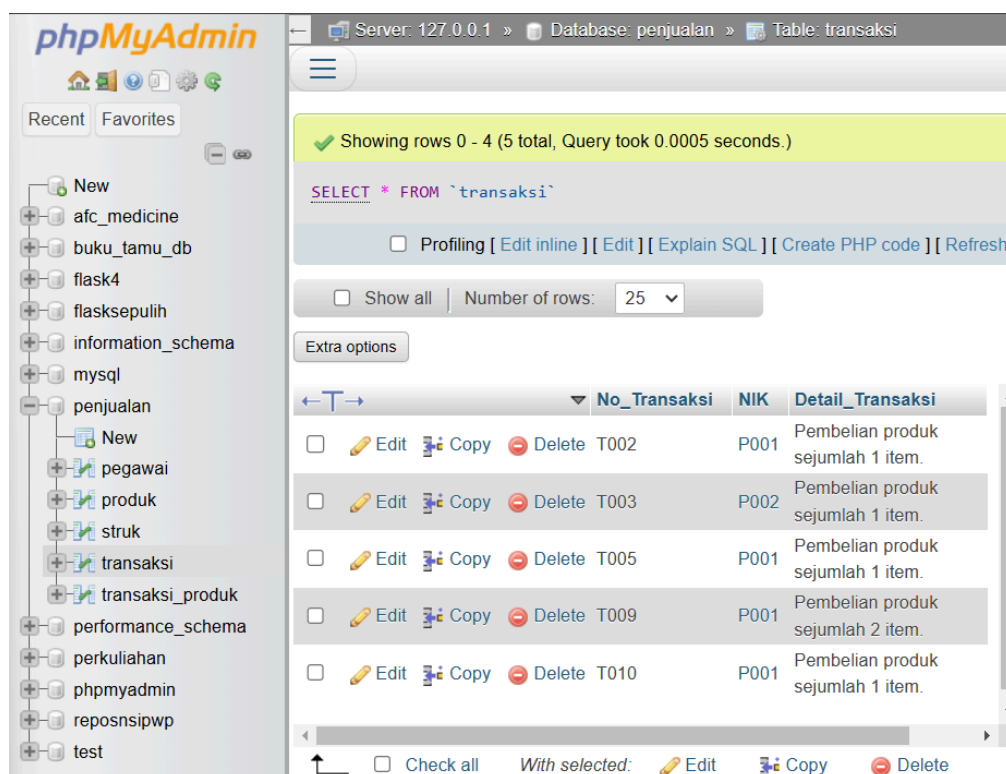
Pengguna mengetikkan nomor transaksi yang akan dihapus, misalnya: T001.

Konfirmasi Penghapusan Data:

Sistem mencocokkan nomor transaksi yang diinputkan dengan data yang ada dalam database.

Jika data ditemukan, sistem menghapus transaksi tersebut dan menampilkan pesan konfirmasi:

“Data Transaksi berhasil dihapus.”



Gambar 17. Tampilan Data Transaksi di phpMyAdmin Setelah Data Dihapus

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
0. Keluar
Pilih menu: 4
=====Kelola Struk=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 1
Masukkan No Transaksi: T010
Data Struk berhasil ditambahkan.
```

Gambar 18. Proses Menambahkan Data Struk

1. Memilih Menu Kelola Struk:

- Dari Menu Utama, pengguna memilih opsi 4. Kelola Struk, yang membawa pengguna ke submenu Kelola Struk.
- Sistem menampilkan daftar submenu berikut:
 - 1. Input Data
 - 2. Tampilkan Data
 - 3. Hapus Data

2. Memilih Submenu Input Data:

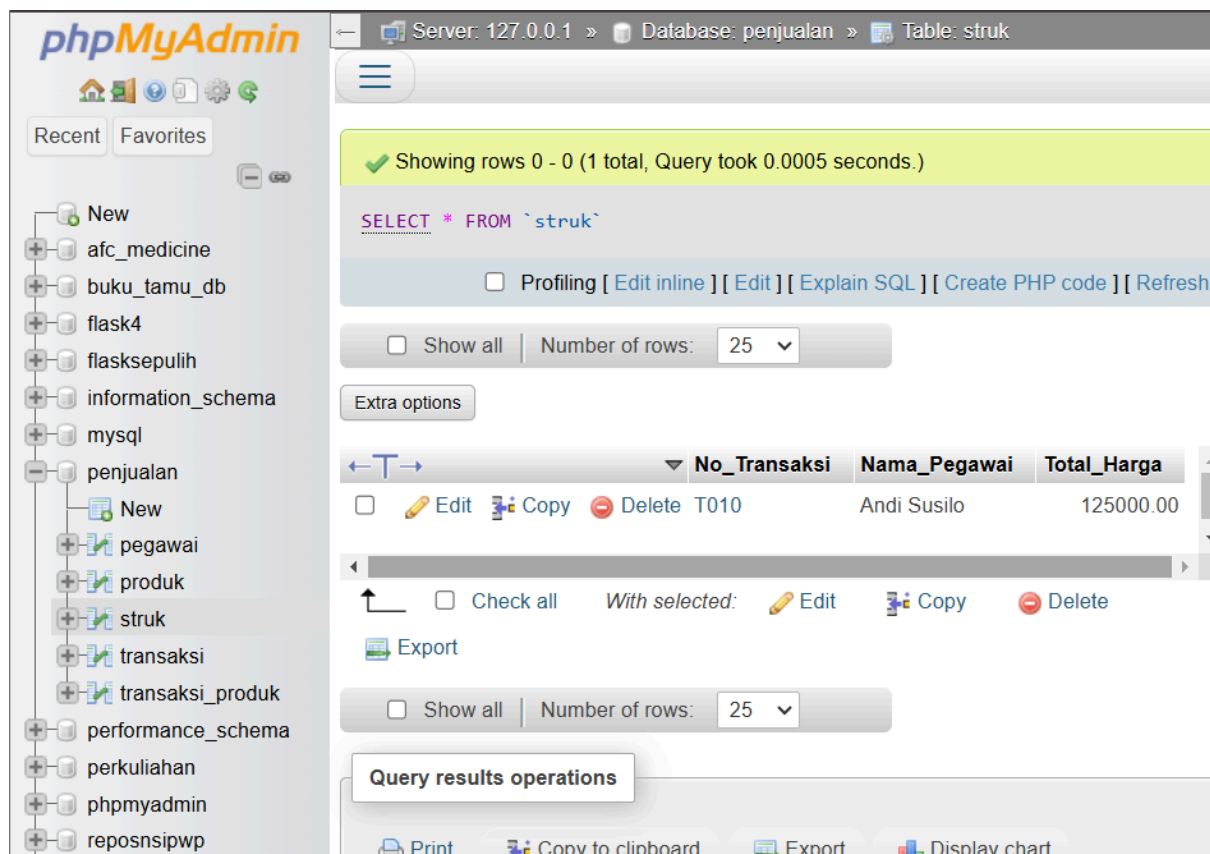
- Pengguna memilih opsi 1 (Input Data) untuk menambahkan data struk baru ke dalam sistem.
- Sistem menampilkan pesan untuk meminta pengguna memasukkan nomor transaksi:
 - "Masukkan No Transaksi:"

3. Input Nomor Transaksi:

- Pengguna mengetikkan T010, yang merupakan nomor transaksi baru yang ingin ditambahkan.
- Sistem memproses input ini dan menyimpan data struk ke dalam database.

4. Konfirmasi Penambahan Data:

- Setelah data struk berhasil ditambahkan, sistem menampilkan pesan konfirmasi:
 - "Data Struk berhasil ditambahkan."



Gambar 19. Tampilan Data Transaksi di phpMyAdmin Setelah Data Ditambahkan dalam Struk


```

=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
0. Keluar
Pilih menu: 4
=====Kelola Struk=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 2

Data Struk:
No Transaksi: T009, Kode Pegawai: None, Total Harga: Rp. 65,000.00
No Transaksi: T010, Kode Pegawai: Andi Susilo, Total Harga: Rp. 125,000.00

```

Gambar 20. Menampilkan Data Struk

1. Memilih Menu Kelola Struk:

- Dari Menu Utama, pengguna memilih opsi 4. Kelola Struk, yang membawa pengguna ke submenu Kelola Struk.
- Sistem menampilkan daftar submenu berikut:
 - 1. Input Data
 - 2. Tampilkan Data
 - 3. Hapus Data

2. Memilih Submenu Tampilkan Data:

- Pengguna memilih opsi 2 (Tampilkan Data) untuk melihat semua data struk yang telah tersimpan dalam sistem.
- Sistem memproses permintaan pengguna dan memuat data dari tabel struk di database.

3. Menampilkan Data Struk:

- Sistem menampilkan daftar semua data struk yang ada di database dalam format berikut:

- No Transaksi: Nomor transaksi yang unik untuk setiap struk.
- Nama Pegawai: Nama pegawai yang melayani transaksi.
- Total Harga: Total harga dari semua produk dalam transaksi.

```
=====Menu Utama=====
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
0. Keluar
Pilih menu: 4
=====Kelola Struk=====
1. Input Data
2. Tampilkan Data
3. Hapus Data
Pilih submenu: 3
Masukkan No Transaksi yang akan dihapus: T009
Data Struk berhasil dihapus.
```

Gambar 21. Menghapus Data Struk

1. Memilih Menu Kelola Struk:

- Dari Menu Utama, pengguna memilih opsi 4. Kelola Struk, yang membawa pengguna ke submenu Kelola Struk.
- Sistem menampilkan daftar submenu berikut:
 - 1. Input Data
 - 2. Tampilkan Data
 - 3. Hapus Data

2. Memilih Submenu Hapus Data:

- Pengguna memilih opsi 3 (Hapus Data) untuk menghapus data struk tertentu dari sistem.
- Sistem meminta pengguna untuk memasukkan No Transaksi dari struk yang akan dihapus.

3. Menginput No Transaksi:

- Pengguna mengetikkan T009 sebagai nomor transaksi yang ingin dihapus.
- Sistem memproses permintaan penghapusan berdasarkan No Transaksi yang dimasukkan.

4. Konfirmasi Penghapusan:

- Jika nomor transaksi T009 ditemukan di database, sistem menghapus data tersebut.
- Setelah penghapusan berhasil, sistem memberikan notifikasi:

“Data Struk berhasil dihapus”

KESIMPULAN

Program ini berhasil dibuat untuk memenuhi kebutuhan pengelolaan data penjualan. Dengan memanfaatkan Python dan MySQL, sistem ini mendukung fungsi input, penghapusan, serta penampilan data dengan struktur relasi antar tabel yang baik. Penggunaan antarmuka menu menjadikan program mudah digunakan oleh pengguna awam sekalipun.