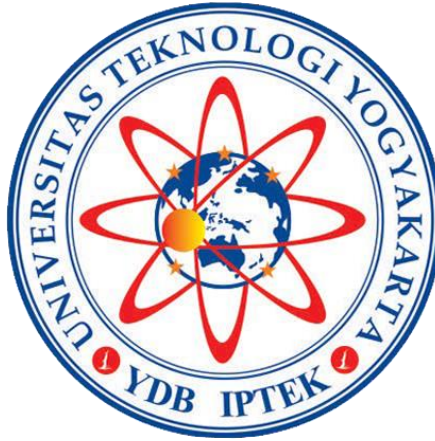


LAPORAN PEMPOROGRAMAN BERBASIS OBJEK PRAKTIK  
PERTEMUAN 10

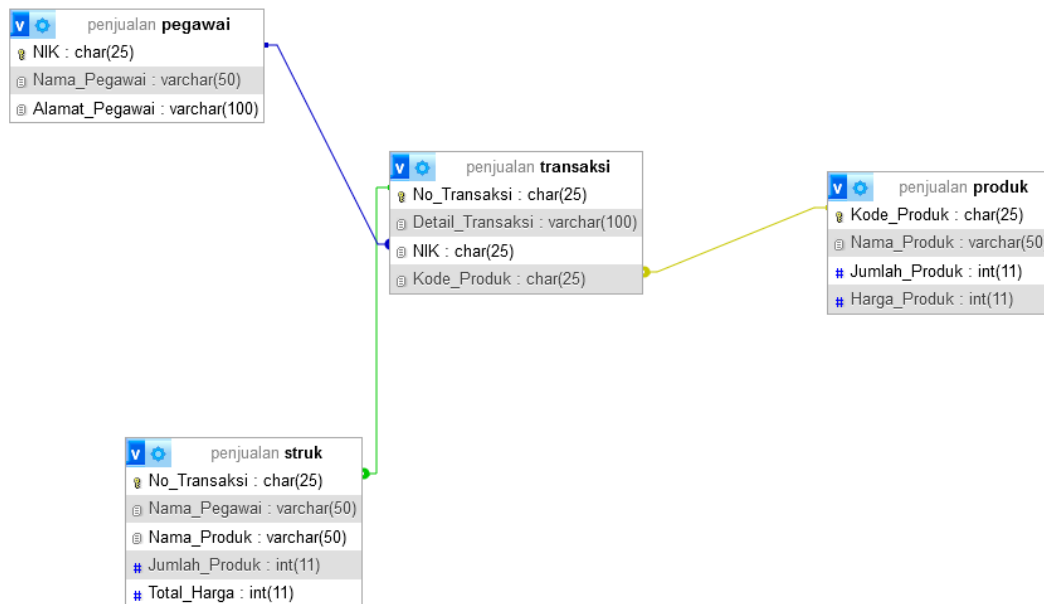


Dibuat Oleh:

Fauzi Kurniawan (5230411251)

PRODI S1 INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
2024

## Desainer phpMyAdmin



### Penjelasan

#### 1. Tabel Pegawai

- **NIK:** Nomor Induk Karyawan, menjadi kunci utama (primary key) yang unik untuk setiap pegawai.
- **Nama\_Pegawai:** Nama lengkap pegawai.
- **Alamat\_Pegawai:** Alamat lengkap pegawai.

Tabel ini digunakan untuk menyimpan data identitas pegawai.

#### 2. Tabel Produk

- **Kode\_Produk:** Kode unik untuk setiap produk, menjadi kunci utama.
- **Nama\_Produk:** Nama produk.
- **Jumlah\_Produk:** Jumlah produk yang tersedia.
- **Harga\_Produk:** Harga satuan produk.

Tabel ini menyimpan informasi mengenai produk yang dijual.

#### 3. Tabel Transaksi

- **No\_Transaksi:** Nomor transaksi yang unik untuk setiap transaksi, menjadi kunci utama.
- **Detail\_Transaksi:** Detail tambahan mengenai transaksi (misalnya, tanggal transaksi).
- **NIK:** Nomor Induk Karyawan yang melakukan transaksi, menjadi foreign key yang merujuk ke tabel Pegawai.
- **Kode\_Produk:** Kode produk yang terlibat dalam transaksi, menjadi foreign key yang merujuk ke tabel Produk.

Tabel ini mencatat semua transaksi penjualan yang terjadi, menghubungkan data pegawai dan produk yang terlibat.

#### 4. Tabel Struk

- **No\_Transaksi:** Nomor transaksi, menjadi kunci utama dan foreign key yang merujuk ke tabel Transaksi.
- **Nama\_Pegawai:** Nama pegawai yang melakukan transaksi (diambil dari tabel Pegawai).

- **Nama\_Produk:** Nama produk yang dibeli (diambil dari tabel Produk).
- **Jumlah\_Produk:** Jumlah produk yang dibeli.
- **Total\_Harga:** Total harga transaksi.

Tabel ini merepresentasikan struk penjualan yang diberikan kepada pelanggan, berisi ringkasan transaksi.

Code

```
import mysql.connector

# Koneksi ke database
conn = mysql.connector.connect(
    user="root",
    host="localhost",
    password="",
    database="penjualan"
)

cur = conn.cursor()

# Membuat tabel jika belum ada
cur.execute("""
    CREATE TABLE IF NOT EXISTS Pegawai (
        NIK CHAR(25) NOT NULL PRIMARY KEY,
        Nama_Pegawai VARCHAR(50),
        Alamat_Pegawai VARCHAR(100)
    )
""")

cur.execute("""
    CREATE TABLE IF NOT EXISTS Produk (
        Kode_Produk CHAR(25) NOT NULL PRIMARY KEY,
        Nama_Produk VARCHAR(50),
        Jumlah_Produk INT,
        Harga_Produk INT
    )
""")

cur.execute("""
    CREATE TABLE IF NOT EXISTS Transaksi (
        No_Transaksi CHAR(25) NOT NULL PRIMARY KEY,
        Detail_Transaksi VARCHAR(100),
        NIK CHAR(25) NOT NULL,
        Kode_Produk CHAR(25) NOT NULL,
        FOREIGN KEY (NIK) REFERENCES Pegawai(NIK),
        FOREIGN KEY (Kode_Produk) REFERENCES Produk(Kode_Produk)
    )
""")
```

```

cur.execute("""
CREATE TABLE IF NOT EXISTS Struk (
    No_Transaksi CHAR(25) NOT NULL PRIMARY KEY,
    Nama_Pegawai VARCHAR(50),
    Nama_Produk VARCHAR(50),
    Jumlah_Produk INT,
    Total_Harga INT,
    FOREIGN KEY (No_Transaksi) REFERENCES Transaksi(No_Transaksi)
)
""")

print("Tabel berhasil dibuat.")

while True:
    print("\n===== Aplikasi Kasir =====")
    print("1. Manajemen Data")
    print("2. Melakukan Transaksi")
    print("3. Laporan")
    print("4. Struk")
    print("5. Keluar")

    menu = input("Masukkan pilihan : ")

    if menu == '1':
        while True:
            print("\n===== Manajemen Data =====")
            print("1. Pegawai")
            print("2. Produk")
            print("3. Kembali")

            sub_menu = input("Masukkan pilihan : ")

            if sub_menu == '1':
                while True:
                    print("\n===== Manajemen Pegawai =====")
                    print("1. Tambah Pegawai")
                    print("2. Hapus Pegawai")
                    print("3. Tampilkan Pegawai")
                    print("4. Kembali")

                    pegawai_menu = input("Masukkan pilihan : ")
                    if pegawai_menu == '1':
                        print("\n===== Tambah Pegawai =====")
                        NIK = input("Masukkan NIK : ")
                        Nama_Pegawai = input("Masukkan Nama Pegawai : ")
                        Alamat = input("Masukkan Alamat : ")
                        cur.execute("""
INSERT INTO Pegawai (NIK, Nama_Pegawai, Alamat_Pegawai)
""")

```

```

        VALUES (%s, %s, %s)""" ,
        (NIK, Nama_Pegawai, Alamat))
    conn.commit()
    print("Pegawai berhasil ditambahkan.")

elif pegawai_menu == '2':
    print("\n===== Hapus Pegawai =====")
    NIK = input("Masukkan NIK : ")
    cur.execute("DELETE FROM Pegawai WHERE NIK = %s", (NIK,))
    conn.commit()
    print("Pegawai berhasil dihapus.")
elif pegawai_menu == '3':
    print("\n===== Tampilkan Pegawai =====")
    cur.execute("SELECT * FROM Pegawai")
    pegawai = cur.fetchall()
    for data in pegawai:
        print(data)

elif pegawai_menu == '4':
    break # Kembali ke Manajemen Data

elif sub_menu == '2':
    while True:
        print("\n===== Manajemen Produk =====")
        print("1. Tambah Produk")
        print("2. Hapus Produk")
        print("3. Tampilkan Produk")
        print("4. Kembali")

        produk_menu = input("Masukkan pilihan : ")
        if produk_menu == '1':
            print("\n===== Tambah Produk =====")
            Kode_Produk = input("Masukkan Kode Produk : ")
            Nama_Produk = input("Masukkan Nama Produk : ")
            Jumlah_Produk = int(input("Masukkan Jumlah Produk : "))
            Harga_Produk = int(input("Masukkan Harga Produk : "))
            cur.execute("""
                INSERT INTO Produk (Kode_Produk, Nama_Produk, Jumlah_Produk, Harga_Produk)
                VALUES (%s, %s, %s, %s)""" ,
                (Kode_Produk, Nama_Produk, Jumlah_Produk, Harga_Produk))
            conn.commit()
            print("Produk berhasil ditambahkan.")

        elif produk_menu == '2':
            print("\n===== Hapus Produk =====")
            Kode_Produk = input("Masukkan Kode Produk : ")
            cur.execute("DELETE FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
            conn.commit()

```

```

        print("Produk berhasil dihapus.")

    elif produk_menu == '3':
        print("\n===== Tampilkan Produk =====")
        cur.execute("SELECT * FROM Produk")
        produk = cur.fetchall()
        for data in produk:
            print(data)

    elif produk_menu == '4':
        break

elif sub_menu == '3':
    break

elif menu == '2':
    print("\n===== Melakukan Transaksi =====")

    print("\n===== Tampilan Data Barang ===== ")
    cur.execute("SELECT Kode_Produk, Nama_Produk, Harga_Produk FROM Produk")
    barang = cur.fetchall()
    for data in barang:
        print(f"ID: {data[0]}, Nama: {data[1]}, Harga: {data[2]}")

    print("\n===== Tampilkan Data Pegawai =====")
    cur.execute("SELECT NIK, Nama_Pegawai FROM Pegawai")
    pegawai = cur.fetchall()
    for data in pegawai:
        print(f"NIK: {data[0]}, Nama: {data[1]}")

    print("\n===== Buat Pembelian =====")
    No_Transaksi = input("Masukkan nomor transaksi : ")
    NIK = input("Masukkan NIK Pegawai : ")
    Kode_Produk = input("Masukkan Kode Produk : ")
    Jumlah_Produk = int(input("Masukkan Jumlah Produk : "))
    cur.execute("SELECT Harga_Produk, Jumlah_Produk FROM Produk WHERE Kode_Produk = %s",
(Kode_Produk,))
    produk_info = cur.fetchone()
    if produk_info:
        harga_produk, jumlah_produk = produk_info
        if Jumlah_Produk <= jumlah_produk:
            Total_Harga = harga_produk * Jumlah_Produk

    cur.execute("""
        INSERT INTO Transaksi (No_Transaksi, NIK, Kode_Produk)
        VALUES (%s, %s, %s)"""
        (No_Transaksi, NIK, Kode_Produk))
    cur.execute("""

```

```

        INSERT INTO Struk (No_Transaksi, Nama_Pegawai, Nama_Produk, Jumlah_Produk,
Total_Harga)
        VALUES (%s, (SELECT Nama_Pegawai FROM Pegawai WHERE NIK = %s), (SELECT
Nama_Produk FROM Produk WHERE Kode_Produk = %s), %s, %s)"""
        (No_Transaksi, NIK, Kode_Produk, Jumlah_Produk, Total_Harga))

    # Update jumlah produk setelah transaksi
    new_jumlah_produk = jumlah_produk - Jumlah_Produk
    cur.execute("UPDATE Produk SET Jumlah_Produk = %s WHERE Kode_Produk = %s",
(new_jumlah_produk, Kode_Produk))

    conn.commit()
    print("Pembelian berhasil dilakukan.")
else:
    print("Jumlah produk tidak cukup.")
else:
    print("Kode Produk tidak ditemukan.")

elif menu == '3':
    print("\n===== Laporan =====")
    print("1. Laporan Penjualan")
    print("2. Laporan Stok")
    print("3. Keluar ")

    laporan_menu = input("Masukkan pilihan : ")

    if laporan_menu == '1':
        print("\n===== Laporan Penjualan =====")
        cur.execute("SELECT No_Transaksi, NIK, Kode_Produk FROM Transaksi")
        penjualan = cur.fetchall()
        for data in penjualan:
            NIK = data[1]
            Kode_Produk = data[2]

            cur.execute("SELECT Nama_Pegawai FROM Pegawai WHERE NIK = %s", (NIK,))
            nama_pegawai = cur.fetchone()
            cur.execute("SELECT Nama_Produk FROM Produk WHERE Kode_Produk = %s",
(Kode_Produk,))
            nama_produk = cur.fetchone()

            if nama_pegawai and nama_produk:
                print(f"No_Transaksi: {data[0]}, Nama_Pegawai: {nama_pegawai[0]}, Nama_Produk:
{nama_produk[0]}")

    elif laporan_menu == '2':
        print("\n===== Laporan Stok =====")
        cur.execute("SELECT Kode_Produk, Nama_Produk, Jumlah_Produk FROM Produk")
        stok = cur.fetchall()

```

```

        for data in stok:
            print(f"Kode_Produk: {data[0]}, Nama_Produk: {data[1]}, Jumlah_Produk: {data[2]}")

    elif laporan_menu == '3':
        print("\nTerima kasih telah menggunakan aplikasi penjualan dan pembelian produk!")
        break

    elif menu == '4':
        print("\n===== Tampilan Transaksi ===== ")
        cur.execute("SELECT No_Transaksi FROM Transaksi")
        transaksi = cur.fetchall()
        for data in transaksi:
            print(f"No_Transaksi: {data[0]}")

        print("\n===== Pembuatan Struk =====")
        No_Transaksi = input("Masukkan No_Transaksi: ")

        # Ambil detail transaksi berdasarkan No_Transaksi
        cur.execute("SELECT Detail_Transaksi, NIK, Kode_Produk FROM Transaksi WHERE No_Transaksi = %s", (No_Transaksi,))
        transaksi = cur.fetchone()
        if transaksi:
            Detail_Transaksi, NIK, Kode_Produk = transaksi

        # Ambil jumlah produk dari transaksi
        cur.execute("SELECT Jumlah_Produk FROM Struk WHERE No_Transaksi = %s", (No_Transaksi,))
        Jumlah_Produk = cur.fetchone()
        Jumlah_Produk = Jumlah_Produk[0] if Jumlah_Produk else 0 # Ambil nilai dari tuple

        # Ambil harga produk
        cur.execute("SELECT Harga_Produk FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
        harga_produk = cur.fetchone()
        harga_produk = harga_produk[0] if harga_produk else 0 # Ambil nilai dari tuple

        if harga_produk:
            Total_Harga = Jumlah_Produk * harga_produk

        # Ambil nama pegawai
        cur.execute("SELECT Nama_Pegawai FROM Pegawai WHERE NIK = %s", (NIK,))
        nama_pegawai = cur.fetchone()

        # Ambil nama produk
        cur.execute("SELECT Nama_Produk FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
        nama_produk = cur.fetchone()

        # Tampilkan struk dengan format yang lebih menarik

```



```

print("\n==== STRUK PEMBELIAN =====")
print("=" * 30)
print(f"No Transaksi    : {No_Transaksi}")
print(f>Nama Pegawai    : {nama_pegawai[0] if nama_pegawai else 'Data tidak ditemukan'}")
print(f>Nama Produk     : {nama_produk[0] if nama_produk else 'Data tidak ditemukan'}")
print(f>Jumlah Produk   : {Jumlah_Produk}")
print(f>Total Harga     : Rp {Total_Harga:,}")
print("=" * 30)

# Simpan struk ke dalam tabel Struk hanya jika belum ada
cur.execute("SELECT * FROM Struk WHERE No_Transaksi = %s", (No_Transaksi,))
if cur.fetchone() is None: # Cek apakah struk sudah ada
    try:
        cur.execute("""
            INSERT INTO Struk (No_Transaksi, Nama_Pegawai, Nama_Produk, Jumlah_Produk,
Total_Harga)
            VALUES (%s, %s, %s, %s, %s)
            """, (No_Transaksi, nama_pegawai[0] if nama_pegawai else None,
                nama_produk[0] if nama_produk else None, Jumlah_Produk, Total_Harga))
        conn.commit()
        print("Struk berhasil disimpan.")
    except mysql.connector.IntegrityError as e:
        print(f"Error: {e}. No Transaksi sudah ada.")
    else:
        print("Struk sudah ada, tidak perlu disimpan lagi.")
    else:
        print("Harga produk tidak ditemukan.")
    else:
        print("Transaksi tidak ditemukan.")
elif menu == '5':
    print("Keluar dari aplikasi.")
    break

```

Penjelasan Code

**Koneksi ke Database**

```

challenge.py > ...
1  import mysql.connector
2
3  # Koneksi ke database
4  conn = mysql.connector.connect(
5      user="root",
6      host="localhost",
7      password="",
8      database="penjualan"
9  )
10
11  cur = conn.cursor()

```

- **Import mysql.connector:** Mengimpor pustaka **mysql.connector** untuk memungkinkan koneksi ke database MySQL.
- **Koneksi ke Database:** Membuat koneksi ke database MySQL dengan menggunakan username, host, password, dan nama database yang ditentukan.
- **Cursor:** Membuat objek cursor (**cur**) untuk mengeksekusi perintah SQL.

### Membuat Tabel

```

# Membuat tabel jika belum ada
cur.execute("""
    CREATE TABLE IF NOT EXISTS Pegawai (
        NIK CHAR(25) NOT NULL PRIMARY KEY,
        Nama_Pegawai VARCHAR(50),
        Alamat_Pegawai VARCHAR(100)
    )
""")

```

- **Membuat Tabel:** Membuat tabel **Pegawai**, **Produk**, **Transaksi**, dan **Struk** jika tabel tersebut belum ada. Setiap tabel memiliki kolom yang sesuai dengan kebutuhan aplikasi.

### Menu Utama

```

while True:
    print("\n===== Aplikasi Kasir =====")
    print("1. Manajemen Data")
    print("2. Melakukan Transaksi")
    print("3. Laporan")
    print("4. Struk")
    print("5. Keluar")

    menu = input("Masukkan pilihan : ")

```

- **Menu Utama:** Menampilkan menu utama aplikasi kasir dengan pilihan untuk manajemen data, melakukan transaksi, laporan, pembuatan struk, dan keluar dari aplikasi.

### Pilihan 1: Manajemen Data

```

if menu == '1':
    while True:
        print("\n===== Manajemen Data =====")
        print("1. Pegawai")
        print("2. Produk")
        print("3. Kembali")

        sub_menu = input("Masukkan pilihan : ")

```

- **Manajemen Data:** Memungkinkan pengguna untuk mengelola data pegawai dan produk. Pengguna dapat memilih untuk mengelola pegawai, produk, atau kembali ke menu utama.

#### Sub-Pilihan 1: Pegawai

```

if sub_menu == '1':
    while True:
        print("\n===== Manajemen Pegawai =====")
        print("1. Tambah Pegawai")
        print("2. Hapus Pegawai")
        print("3. Tampilkan Pegawai")
        print("4. Kembali")

        pegawai_menu = input("Masukkan pilihan : ")
        if pegawai_menu == '1':
            print("\n===== Tambah Pegawai =====")
            NIK = input("Masukkan NIK : ")
            Nama_Pegawai = input("Masukkan Nama Pegawai : ")
            Alamat = input("Masukkan Alamat : ")
            cur.execute("""
                INSERT INTO Pegawai (NIK, Nama_Pegawai, Alamat_Pegawai)
                VALUES (%s, %s, %s)"""
                , (NIK, Nama_Pegawai, Alamat))
            conn.commit()
            print("Pegawai berhasil ditambahkan.")

        elif pegawai_menu == '2':
            print("\n===== Hapus Pegawai =====")
            NIK = input("Masukkan NIK : ")
            cur.execute("DELETE FROM Pegawai WHERE NIK = %s", (NIK,))
            conn.commit()
            print("Pegawai berhasil dihapus.")

        elif pegawai_menu == '3':
            print("\n===== Tampilkan Pegawai =====")
            cur.execute("SELECT * FROM Pegawai")
            pegawai = cur.fetchall()
            for data in pegawai:
                print(data)

        elif pegawai_menu == '4':
            break # Kembali ke Manajemen Data

```

- **Manajemen Pegawai:** Menampilkan opsi untuk menambah, menghapus, menampilkan pegawai, atau kembali ke menu manajemen data.

## Tambah Pegawai

- **Tambah Pegawai:** Mengambil input dari pengguna untuk NIK, nama, dan alamat pegawai, kemudian menyimpannya ke dalam tabel **Pegawai**.

## Hapus Pegawai

- **Hapus Pegawai:** Menghapus pegawai dari tabel **Pegawai** berdasarkan NIK yang dimasukkan oleh pengguna.

## Tampilkan Pegawai

- **Tampilkan Pegawai:** Menampilkan semua data pegawai yang ada dalam tabel **Pegawai**.

## Sub-Pilihan 2: Produk

```
elif sub_menu == '2':
    while True:
        print("\n===== Manajemen Produk =====")
        print("1. Tambah Produk")
        print("2. Hapus Produk")
        print("3. Tampilkan Produk")
        print("4. Kembali")

        produk_menu = input("Masukkan pilihan : ")
        if produk_menu == '1':
            print("\n===== Tambah Produk =====")
            Kode_Produk = input("Masukkan Kode Produk : ")
            Nama_Produk = input("Masukkan Nama Produk : ")
            Jumlah_Produk = int(input("Masukkan Jumlah Produk : "))
            Harga_Produk = int(input("Masukkan Harga Produk : "))
            cur.execute("""
                INSERT INTO Produk (Kode_Produk, Nama_Produk, Jumlah_Produk, Harga_Produk)
                VALUES (%s, %s, %s, %s)"""
                (Kode_Produk, Nama_Produk, Jumlah_Produk, Harga_Produk))
            conn.commit()
            print("Produk berhasil ditambahkan.")

        elif produk_menu == '2':
            print("\n===== Hapus Produk =====")
            Kode_Produk = input("Masukkan Kode Produk : ")
            cur.execute("DELETE FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
            conn.commit()
            print("Produk berhasil dihapus.")

        elif produk_menu == '3':
            print("\n===== Tampilkan Produk =====")
            cur.execute("SELECT * FROM Produk")
            produk = cur.fetchall()
            for data in produk:
                print(data)

        elif produk_menu == '4':
            break
```

- **Manajemen Produk:** Menampilkan opsi untuk menambah, menghapus, menampilkan produk, atau kembali ke menu manajemen data.

## Tambah Produk

- **Tambah Produk:** Mengambil input dari pengguna untuk kode produk, nama produk, jumlah produk, dan harga produk, kemudian menyimpannya ke dalam tabel **Produk**.

## Hapus Produk

- **Hapus Produk:** Menghapus produk dari tabel **Produk** berdasarkan kode produk yang dimasukkan oleh pengguna.

### Tampilkan Produk

- **Tampilkan Produk:** Menampilkan semua data produk yang ada dalam tabel **Produk**.

### Pilihan 2: Melakukan Transaksi

```
elif menu == '2':
    print("\n===== Melakukan Transaksi =====")

    print("\n ---- Tampilan Data Barang ---- ")
    cur.execute("SELECT Kode_Produk, Nama_Produk, Harga_Produk FROM Produk")
    barang = cur.fetchall()
    for data in barang:
        print(f"ID: {data[0]}, Nama: {data[1]}, Harga: {data[2]}")

    print("\n ---- Tampilkan Data Pegawai ----")
    cur.execute("SELECT NIK, Nama_Pegawai FROM Pegawai")
    pegawai = cur.fetchall()
    for data in pegawai:
        print(f"NIK: {data[0]}, Nama: {data[1]}")

    print("\n==== Buat Pembelian ====")
    No_Transaksi = input("Masukkan nomor transaksi : ")
    NIK = input("Masukkan NIK Pegawai : ")
    Kode_Produk = input("Masukkan Kode Produk : ")
    Jumlah_Produk = int(input("Masukkan Jumlah Produk : "))
    cur.execute("SELECT Harga_Produk, Jumlah_Produk FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
    produk_info = cur.fetchone()
    if produk_info:
        harga_produk, jumlah_produk = produk_info
        if Jumlah_Produk <= jumlah_produk:
            Total_Harga = harga_produk * Jumlah_Produk

            cur.execute("""
                INSERT INTO Transaksi (No_Transaksi, NIK, Kode_Produk)
                VALUES (%s, %s, %s)"""
                (No_Transaksi, NIK, Kode_Produk))
            cur.execute("""
                INSERT INTO Struk (No_Transaksi, Nama_Pegawai, Nama_Produk, Jumlah_Produk, Total_Harga)
                VALUES (%s, (SELECT Nama_Pegawai FROM Pegawai WHERE NIK = %s), (SELECT Nama_Produk FROM Produk WHERE Kode_Produk = %s), %s, %s)"""
                (No_Transaksi, NIK, Kode_Produk, Jumlah_Produk, Total_Harga))

            # Update jumlah produk setelah transaksi
            new_jumlah_produk = jumlah_produk - Jumlah_Produk
            cur.execute("UPDATE Produk SET Jumlah_Produk = %s WHERE Kode_Produk = %s", (new_jumlah_produk, Kode_Produk))

            conn.commit()
            print("Pembelian berhasil dilakukan.")
        else:
            print("Jumlah produk tidak cukup.")
    else:
        print("Kode Produk tidak ditemukan.")
```

- **Melakukan Transaksi:** Memungkinkan pengguna untuk melakukan transaksi penjualan produk.

### Menampilkan Data Barang dan Pegawai

- **Tampilan Data Barang dan Pegawai:** Menampilkan daftar produk dan pegawai yang tersedia untuk transaksi.

### Input Transaksi

- **Input Transaksi:** Mengambil input dari pengguna untuk nomor transaksi, NIK pegawai, kode produk, dan jumlah produk yang ingin dibeli.

### Validasi dan Penyimpanan Transaksi

- **Validasi dan Penyimpanan Transaksi:** Mengecek apakah kode produk ada dan apakah jumlah produk yang diminta cukup. Jika valid, transaksi disimpan dalam tabel **Transaksi** dan **Struk**, serta jumlah produk diupdate di tabel **Produk**.

### Pilihan 3: Laporan

```
elif menu == '3':  
    print("\n==== Laporan =====")  
    print("1. Laporan Penjualan")  
    print("2. Laporan Pembelian")  
    print("3. Laporan Stok")  
    print("4. Keluar ")  
  
    laporan_menu = input("Masukkan pilihan : ")
```

- **Laporan:** Menampilkan opsi untuk melihat laporan penjualan, pembelian, stok, atau keluar dari aplikasi.

### Sub-Pilihan 1: Laporan Penjualan

```
if laporan_menu == '1':  
    print("\n==== Laporan Penjualan =====")  
    cur.execute("SELECT No_Transaksi, NIK, Kode_Produk FROM Transaksi")  
    penjualan = cur.fetchall()  
    for data in penjualan:  
        NIK = data[1]  
        Kode_Produk = data[2]  
  
        cur.execute("SELECT Nama_Pegawai FROM Pegawai WHERE NIK = %s", (NIK,))  
        nama_pegawai = cur.fetchone()  
        cur.execute("SELECT Nama_Produk FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))  
        nama_produk = cur.fetchone()  
  
        if nama_pegawai and nama_produk:  
            print(f"No_Transaksi: {data[0]}, Nama_Pegawai: {nama_pegawai[0]}, Nama_Produk: {nama_produk[0]}")
```

- **Laporan Penjualan:** Mengambil dan menampilkan semua transaksi penjualan, termasuk nama pegawai dan produk yang terlibat.

### Sub-Pilihan 2: Laporan Stok

```
elif laporan_menu == '2':  
    print("\n==== Laporan Stok =====")  
    cur.execute("SELECT Kode_Produk, Nama_Produk, Jumlah_Produk FROM Produk")  
    stok = cur.fetchall()  
    for data in stok:  
        print(f"Kode_Produk: {data[0]}, Nama_Produk: {data[1]}, Jumlah_Produk: {data[2]}")
```

- **Laporan Stok:** Menampilkan semua produk yang ada beserta jumlah stok yang tersedia.

## Pilihan 4: Pembuatan Struk

```
elif menu == '4':
    print("\n==== Tampilan Transaksi ==== ")
    cur.execute("SELECT No_Transaksi FROM Transaksi")
    transaksi = cur.fetchall()
    for data in transaksi:
        print(f"No_Transaksi: {data[0]}")

    print("\n==== Pembuatan Struk ====")
    No_Transaksi = input("Masukkan No_Transaksi: ")

    # Ambil detail transaksi berdasarkan No_Transaksi
    cur.execute("SELECT Detail_Transaksi, NIK, Kode_Produk FROM Transaksi WHERE No_Transaksi = %s", (No_Transaksi,))
    transaksi = cur.fetchone()
    if transaksi:
        Detail_Transaksi, NIK, Kode_Produk = transaksi

        # Ambil jumlah produk dari transaksi
        cur.execute("SELECT Jumlah_Produk FROM Struk WHERE No_Transaksi = %s", (No_Transaksi,))
        Jumlah_Produk = cur.fetchone()
        Jumlah_Produk = Jumlah_Produk[0] if Jumlah_Produk else 0 # Ambil nilai dari tuple

        # Ambil harga produk
        cur.execute("SELECT Harga_Produk FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
        harga_produk = cur.fetchone()
        harga_produk = harga_produk[0] if harga_produk else 0 # Ambil nilai dari tuple

        if harga_produk:
            Total_Harga = Jumlah_Produk * harga_produk

        # Ambil nama pegawai
        cur.execute("SELECT Nama_Pegawai FROM Pegawai WHERE NIK = %s", (NIK,))
        nama_pegawai = cur.fetchone()

        # Ambil nama produk
        cur.execute("SELECT Nama_Produk FROM Produk WHERE Kode_Produk = %s", (Kode_Produk,))
        nama_produk = cur.fetchone()

        # Tampilkan struk dengan format yang lebih menarik
        print("\n==== STRUK PEMBELIAN ====")
        print("=" * 30)
        print(f"No Transaksi      : {No_Transaksi}")
        print(f>Nama Pegawai       : {nama_pegawai[0] if nama_pegawai else 'Data tidak ditemukan'}")
        print(f>Nama Produk        : {nama_produk[0] if nama_produk else 'Data tidak ditemukan'}")
        print(f"Jumlah Produk       : {Jumlah_Produk}")
        print(f>Total Harga        : Rp {Total_Harga:,}")
        print("=" * 30)

        # Simpan struk ke dalam tabel Struk hanya jika belum ada
        cur.execute("SELECT * FROM Struk WHERE No_Transaksi = %s", (No_Transaksi,))
        if cur.fetchone() is None: # Cek apakah struk sudah ada
            try:
                cur.execute("""
                    INSERT INTO Struk (No_Transaksi, Nama_Pegawai, Nama_Produk, Jumlah_Produk, Total_Harga)
                    VALUES (%s, %s, %s, %s, %s)
                    """, (No_Transaksi, nama_pegawai[0] if nama_pegawai else None,
                        nama_produk[0] if nama_produk else None, Jumlah_Produk, Total_Harga))
                conn.commit()
                print("Struk berhasil disimpan.")
            except mysql.connector.IntegrityError as e:
                print(f"Error: {e}. No Transaksi sudah ada.")
            else:
                print("Struk sudah ada, tidak perlu disimpan lagi.")
        else:
            print("Harga produk tidak ditemukan.")
    else:
        print("Transaksi tidak ditemukan.")
```

- **Pembuatan Struk:** Memungkinkan pengguna untuk membuat struk berdasarkan nomor transaksi yang dimasukkan. Kode ini mengambil detail transaksi, jumlah produk, harga produk, dan nama pegawai serta produk untuk ditampilkan dalam format struk yang lebih menarik. Jika struk belum ada, maka struk akan disimpan ke dalam tabel **Struk**.

## Pilihan 5: Keluar

```
elif menu == '5':
    print("Keluar dari aplikasi.")
    break
```

- **Keluar:** Menghentikan loop dan keluar dari aplikasi.

## Bukti Percobaan Code

### Manajemen Pegawai

```
===== Manajemen Pegawai =====
1. Tambah Pegawai
2. Hapus Pegawai
3. Tampilkan Pegawai
4. Kembali
Masukkan pilihan : 3

===== Tampilkan Pegawai =====
('HoloGen2', 'Risdu', 'Kelurahan Holo id')
('HOLOGEN3', 'Kaela', 'Kelurahan Holo id')

===== Manajemen Pegawai =====
1. Tambah Pegawai
2. Hapus Pegawai
3. Tampilkan Pegawai
4. Kembali
Masukkan pilihan : 1

===== Tambah Pegawai =====
Masukkan NIK : HoloGen1
Masukkan Nama Pegawai : Yopi
Masukkan Alamat : Kelurahan Holo id
Pegawai berhasil ditambahkan.

===== Manajemen Pegawai =====
1. Tambah Pegawai
2. Hapus Pegawai
3. Tampilkan Pegawai
4. Kembali
Masukkan pilihan : 3

===== Tampilkan Pegawai =====
('HoloGen1', 'Yopi', 'Kelurahan Holo id')
('HoloGen2', 'Risdu', 'Kelurahan Holo id')
('HOLOGEN3', 'Kaela', 'Kelurahan Holo id')
```

### Manajemen Produk

```
===== Manajemen Produk =====
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Kembali
Masukkan pilihan : 1

===== Tambah Produk =====
Masukkan Kode Produk : HA01
Masukkan Nama Produk : Godzilla ayang yopi
Masukkan Jumlah Produk : 1
Masukkan Harga Produk : 1000000000
Produk berhasil ditambahkan.

===== Manajemen Produk =====
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Kembali
Masukkan pilihan : 3

===== Tampilkan Produk =====
('HA01', 'Godzilla ayang yopi', 1, 1000000000)
('MI01', 'Air keringat moona', 100, 50000)
('MK01', 'Lele Kodam Zeta', 37, 10000)
```



## Transaksi

```
===== Aplikasi Kasir =====
1. Manajemen Data
2. Melakukan Transaksi
3. Laporan
4. Struk
5. Keluar
Masukkan pilihan : 2

===== Melakukan Transaksi =====

===== Tampilan Data Barang =====
ID: HA01, Nama: Godzila ayang yopi, Harga: 1000000000
ID: MI01, Nama: Air keringat moona, Harga: 50000
ID: MK01, Nama: Lele Kodam Zeta, Harga: 10000

===== Tampilkan Data Pegawai =====
NIK: HoloGen1, Nama: Yopi
NIK: HoloGen2, Nama: Risdu
NIK: HOLOGEN3, Nama: Kaela

===== Buat Pembelian =====
Masukkan nomor transaksi : 002
Masukkan NIK Pegawai : HOLOGEN3
Masukkan Kode Produk : MK01
Masukkan Jumlah Produk : 1
Pembelian berhasil dilakukan.
```

## Laporan

```
===== Laporan =====
1. Laporan Penjualan
2. Laporan Stok
3. Keluar
Masukkan pilihan : 1

===== Laporan Penjualan =====
No_Transaksi: 001, Nama_Pegawai: Yopi, Nama_Produk: Godzila ayang yopi
No_Transaksi: EN01, Nama_Pegawai: Kaela, Nama_Produk: Lele Kodam Zeta
No_Transaksi: ID100, Nama_Pegawai: Yopi, Nama_Produk: Lele Kodam Zeta
No_Transaksi: ID200, Nama_Pegawai: Risdu, Nama_Produk: Air keringat moona
No_Transaksi: JP01, Nama_Pegawai: Risdu, Nama_Produk: Air keringat moona

===== Aplikasi Kasir =====
1. Manajemen Data
2. Melakukan Transaksi
3. Laporan
4. Struk
5. Keluar
Masukkan pilihan : 3

===== Laporan =====
1. Laporan Penjualan
2. Laporan Stok
3. Keluar
Masukkan pilihan : 2

===== Laporan Stok =====
Kode_Produk: HA01, Nama_Produk: Godzila ayang yopi, Jumlah_Produk: 0
Kode_Produk: MI01, Nama_Produk: Air keringat moona, Jumlah_Produk: 23
Kode_Produk: MK01, Nama_Produk: Lele Kodam Zeta, Jumlah_Produk: 27
```

## Struk

```
===== Tampilan Transaksi =====
No_Transaksi: 001
No_Transaksi: ID100
No_Transaksi: ID200
No_Transaksi: JP01
No_Transaksi: 002
No_Transaksi: EN01

===== Pembuatan Struk =====
Masukkan No_Transaksi: 002

===== STRUK PEMBELIAN =====
=====
No Transaksi      : 002
Nama Pegawai     : Kaela
Nama Produk      : Lele Kodam Zeta
Jumlah Produk    : 1
Total Harga      : Rp 10,000
=====
Struk sudah ada, tidak perlu disimpan lagi.
```