

TUGAS 10
PEMROGRAMAN BERORIENTASI OBJEK
“CHALLENGE Membuat Database dari Studi Kasus Penjualan”



DISUSUN OLEH :
ELSA AMBARWATI
(5230411238)

Program Studi Informatika
Fakultas Sains dan Teknologi

Universitas Teknologi Yogyakarta
Tahun Akademik
2024/2025

DAFTAR ISI

DAFTAR ISI.....	2
BAB I	3
PENDAHULUAN	3
BAB II.....	3
PEMBAHASAN.....	3
2.1 Membuat Database	3
2.2 Menambahkan Foreign Key	5
2.3 Membuat Menu Utama.....	6
2.4 Input Struk.....	11
2.4 Mengubah Data Struk	13
2.5 Hapus Data Struk	14
BAB III	14
PENUTUP.....	14
3.1 Kesimpulan	14

BAB I

PENDAHULUAN

Sistem informasi penjualan merupakan alat penting dalam mengelola transaksi dan data terkait dalam bisnis. Laporan ini membahas implementasi sistem penjualan sederhana menggunakan Python dan MySQL. Sistem ini dirancang untuk mengelola data pegawai, transaksi, produk, dan struktur penjualan dengan efisien.

Pengguna dapat menampilkan, memasukkan, mengubah, dan menghapus data melalui antarmuka berbasis teks yang sederhana. Dengan menggunakan database MySQL, sistem ini mendukung operasi CRUD (Create, Read, Update, Delete) untuk memudahkan pengelolaan informasi. Melalui Laporan ini, Saya akan menjelaskan struktur database, alur kerja sistem, dan implementasi kode program yang mendasari fungsi-fungsi tersebut. Diharapkan Laporan ini dapat memberikan wawasan mengenai pengembangan sistem informasi penjualan dalam konteks bisnis.

BAB II

PEMBAHASAN

2.1 Membuat Database










Bagian kode ini berfungsi untuk membuat struktur tabel dalam database MySQL untuk sistem penjualan.

1. **Tabel Pegawai** menyimpan data pegawai dengan kolom NIK (Nomor Induk Karyawan) sebagai primary key, serta nama dan alamat pegawai.

Python :

```
cur.execute("""CREATE TABLE Pegawai (  
    NIK INT NOT NULL PRIMARY KEY,  
    Nama_Pegawai VARCHAR(225),  
    Alamat VARCHAR(255))""")
```

Hasil di MySQL :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 NIK 	int(11)			No	None			 Change  Drop  More
<input type="checkbox"/>	2 Nama_Pegawai	varchar(225)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More
<input type="checkbox"/>	3 Alamat	varchar(255)	utf8mb4_general_ci		Yes	NULL			 Change  Drop  More

2. **Tabel Transaksi** menyimpan informasi transaksi, dengan No_Transaksi sebagai primary key dan detail transaksi.

Python :

```
cur.execute("""CREATE TABLE Transaksi (  
    No_Transaksi CHAR(6) NOT NULL PRIMARY KEY,  
    Detail_Transaksi VARCHAR(255),  
    NIK INT)""")
```

Hasil di MySQL :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 No_Transaksi	char(6)	utf8mb4_general_ci		No	None			Change Drop More
<input type="checkbox"/>	2 Detail_Transaksi	varchar(255)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 NIK	int(11)			Yes	NULL			Change Drop More

3. **Tabel Produk** menyimpan data produk, di mana Kode_Produk berfungsi sebagai primary key, serta nama, jenis, dan harga produk.

Python :

```
cur.execute("""CREATE TABLE Produk (
    Kode_Produk INT NOT NULL PRIMARY KEY,
    No_Transaksi CHAR(6),
    Nama_Produk VARCHAR(225),
    Jenis_Produk VARCHAR(225),
    Harga DECIMAL(10, 3))""")
```

Hasil di MySQL :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 Kode_Produk	int(11)			No	None			Change Drop More
<input type="checkbox"/>	2 No_Transaksi	char(6)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 Nama_Produk	varchar(225)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	4 Jenis_Produk	varchar(225)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 Harga	decimal(10,3)			Yes	NULL			Change Drop More

4. **Tabel Struk** menyimpan informasi struk penjualan, termasuk No_Struk, No_Transaksi, NIK, nama pegawai, kode produk, nama produk, jumlah produk, dan total harga.

Python :

```
cur.execute("""CREATE TABLE Struk (
    No_Struk CHAR (6),
    No_Transaksi CHAR(6),
    NIK INT,
    Nama_Pegawai VARCHAR(225),
    Kode_Produk INT(2),
    Nama_Produk VARCHAR(225),
    Jumlah_Produk INT(1),
    Total_Harga DECIMAL(10, 3))""")
```

Hasil di MySQL :

#	Name	Type	Collation	Attributes	Null	Default	Comments	Extra	Action
<input type="checkbox"/>	1 No_Struk	char(6)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	2 No_Transaksi	char(6)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	3 NIK	int(11)			Yes	NULL			Change Drop More
<input type="checkbox"/>	4 Nama_Pegawai	varchar(225)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	5 Kode_Produk	int(2)			Yes	NULL			Change Drop More
<input type="checkbox"/>	6 Nama_Produk	varchar(225)	utf8mb4_general_ci		Yes	NULL			Change Drop More
<input type="checkbox"/>	7 Jumlah_Produk	int(1)			Yes	NULL			Change Drop More
<input type="checkbox"/>	8 Total_Harga	decimal(10,3)			Yes	NULL			Change Drop More

2.2 Menambahkan Foreign Key

```
cur.execute("""ALTER TABLE Transaksi
              ADD FOREIGN KEY (NIK)
              REFERENCES Pegawai(NIK)""")

cur.execute("""ALTER TABLE Produk
              ADD FOREIGN KEY (No_Transaksi)
              REFERENCES Transaksi(No_Transaksi)""")

cur.execute("""ALTER TABLE Struk
              ADD FOREIGN KEY (No_Transaksi)
              REFERENCES Transaksi(No_Transaksi)""")

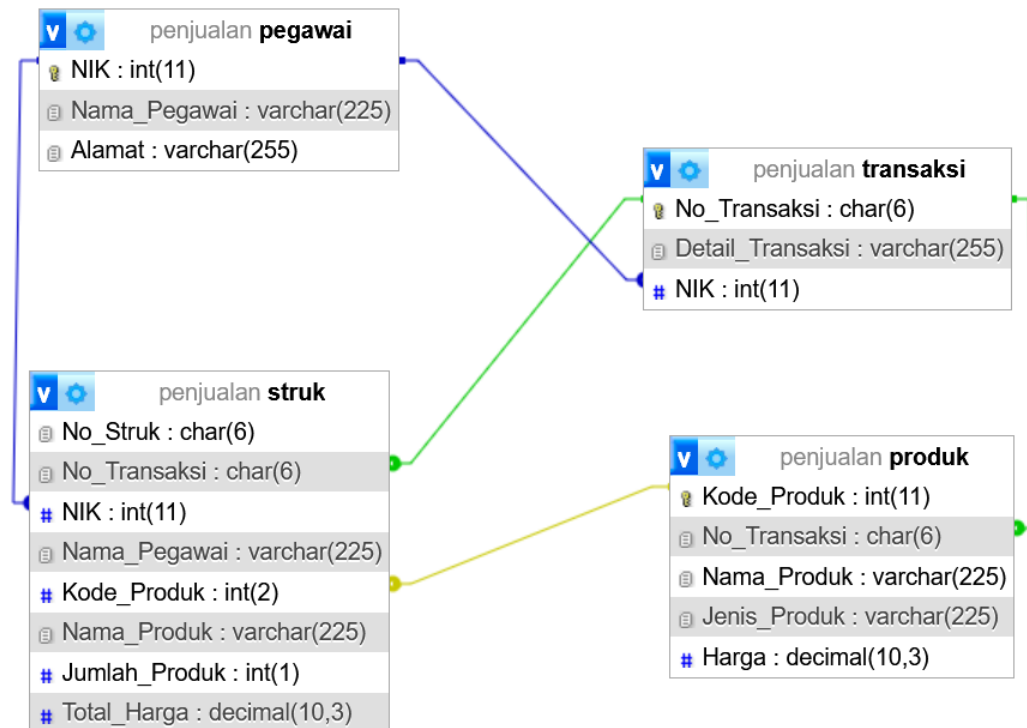
cur.execute("""ALTER TABLE Struk
              ADD FOREIGN KEY (NIK)
              REFERENCES Pegawai(NIK)""")

cur.execute("""ALTER TABLE Struk
              ADD FOREIGN KEY (Kode_Produk)
              REFERENCES Produk(Kode_Produk)""")
```

1. **Transaksi - NIK:** Menghubungkan **NIK** di tabel **Transaksi** dengan **NIK** di tabel **Pegawai**, memastikan setiap transaksi terkait dengan pegawai yang valid.
2. **Produk - No_Transaksi:** Menghubungkan **No_Transaksi** di tabel **Produk** dengan **No_Transaksi** di tabel **Transaksi**, menjaga konsistensi antara produk dan transaksi.
3. **Struk - No_Transaksi:** Menghubungkan **No_Transaksi** di tabel **Struk** dengan **No_Transaksi** di tabel **Transaksi**, memastikan struk hanya untuk transaksi yang ada.
4. **Struk - NIK:** Menghubungkan **NIK** di tabel **Struk** dengan **NIK** di tabel **Pegawai**, memastikan struk terkait dengan pegawai yang valid.

5. **Struk - Kode_Produk**: Menghubungkan **Kode_Produk** di tabel **Struk** dengan **Kode_Produk** di tabel **Produk**, menjaga konsistensi antara struk dan produk.

Hasil di MySQL :



2.3 Membuat Menu Utama

```
while True:
    print("\n===== Menu Penjualan =====")
    print("1. Tampilkan Data")
    print("2. Input Data")
    print("3. Input Struk")
    print("4. Ubah Data Struk")
    print("5. Hapus Data Struk")
    print("0. Keluar")
    menu = int(input("Masukan Pilihan Menu (1/2/3/4/5/0) :"))
```

```
PS D:\kuliah\semester 3\Pemrograman berorientasi objek>
===== Menu Penjualan =====
1. Tampilkan Data
2. Input Data
3. Input Struk
4. Ubah Data Struk
5. Hapus Data Struk
0. Keluar
Masukan Pilihan Menu (1/2/3/4/5/0) :
```

Program ini adalah sebuah aplikasi berbasis teks yang digunakan untuk mengelola data penjualan, termasuk data pegawai, transaksi, produk, dan struk. Aplikasi ini menggunakan Python dan terhubung dengan database (misalnya, MySQL) untuk menyimpan dan mengambil data.

```
# Tampilkan data
if menu == 1:
    print("\n===== Menampilkan Data =====")

    # Menampilkan Data Pegawai
    cur.execute("SELECT * FROM Pegawai")
    print("\nData Pegawai :")
    pegawai = cur.fetchall()
    if pegawai:
        for row in pegawai:
            print(row)
    else:
        print("Tidak ada data pegawai.")

    # Menampilkan Data Transaksi
    cur.execute("SELECT * FROM Transaksi")
    print("\nData Transaksi :")
    transaksi = cur.fetchall()
    if transaksi:
        for row in transaksi:
            print(row)
    else:
        print("Tidak ada data transaksi.")

    # Menampilkan Data Produk
    cur.execute("SELECT * FROM Produk")
    print("\nData Produk :")
    produk = cur.fetchall()
    if produk:
        for row in produk:
            print(row)
    else:
        print("Tidak ada data produk.")

    # Menampilkan Data Struk
    cur.execute("SELECT * FROM Struk")
    print("\nData Struk :")
    struk = cur.fetchall()
    if struk:
        for row in struk:
            print(row)
    else:
        print("Tidak ada data struk.")
```

1. Tampilkan Data

Jika pengguna memilih opsi 1, program akan menampilkan data dari empat tabel: **Pegawai**, **Transaksi**, **Produk**, dan **Struk**.

- Program akan menjalankan query SQL untuk mengambil semua data dari masing-masing tabel.
- Jika data ada, program akan mencetak setiap baris data. Jika tidak ada data, program akan menampilkan pesan yang sesuai.

Hasil yang Diharapkan: Pengguna dapat melihat semua data yang tersimpan dalam tabel-tabel tersebut.

```
===== Menampilkan Data =====  
  
Data Pegawai :  
Tidak ada data pegawai.  
  
Data Transaksi :  
Tidak ada data transaksi.  
  
Data Produk :  
Tidak ada data produk.  
  
Data Struk :  
Tidak ada data struk.
```

2. Input Data

Jika pengguna memilih opsi 2, program akan menampilkan submenu untuk memasukkan data baru ke dalam tabel:

```
elif menu == 2:  
    while True:  
        print("\n===== Input Data =====")  
        print("1. Pegawai")  
        print("2. Transaksi")  
        print("3. Produk")  
        print("0. Kembali")  
        pilihan = int(input("Masukkan Pilihan (1/2/3/0): "))  
  
        if pilihan == 1:  
            print("\n===== Input Data Pegawai =====")  
            NIK = int(input("Masukkan NIK (Angka): "))  
            Nama_Pegawai = input("Masukkan Nama Pegawai: ")  
            Alamat = input("Masukkan Alamat: ")  
  
            cur.execute("""INSERT INTO Pegawai (NIK, Nama_Pegawai,  
Alamat)
```



```

VALUES (%s, %s, %s)""" , (NIK, Nama_Pegawai,
Alamat))

    conn.commit()
    print("Data Pegawai berhasil ditambahkan.")

    elif pilihan == 2:
        print("\n===== Input Data Transaksi =====")
        No_Transaksi = input("Masukkan No Transaksi (TR): ")
        Detail_Transaksi = input("Masukkan Detail Transaksi: ")
        NIK = int(input("Masukkan NIK (angka): "))

        cur.execute("""INSERT INTO Transaksi (No_Transaksi,
Detail_Transaksi, NIK)
VALUES (%s, %s, %s)""" , (No_Transaksi,
Detail_Transaksi, NIK))
        conn.commit()
        print("Data Transaksi berhasil ditambahkan.")

    elif pilihan == 3:
        print("\n===== Input Data Produk =====")
        Kode_Produk = int(input("Masukkan Kode Produk (angka):
"))
        No_Transaksi = input("Masukkan No Transaksi (TR): ")
        Nama_Produk = input("Masukkan Nama Produk: ")
        Jenis_Produk = input("Masukkan Jenis Produk (Snack,
Makanan, Minuman): ")
        Harga = Decimal(input("Masukkan Harga (angka): "))

        cur.execute("""INSERT INTO Produk (Kode_Produk,
No_Transaksi, Nama_Produk, Jenis_Produk, Harga)
VALUES (%s, %s, %s, %s, %s)""" ,
(Kode_Produk, No_Transaksi, Nama_Produk,
Jenis_Produk, Harga))
        conn.commit()
        print("Data Produk berhasil ditambahkan.")

    elif pilihan == 0:
        print("Kembali ke menu utama.")
        break

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

```

```

===== Input Data =====
1. Pegawai
2. Transaksi
3. Produk
0. Kembali
Masukkan Pilihan (1/2/3/0): 

```

- **Pegawai:** Pengguna diminta untuk memasukkan NIK, nama pegawai, dan alamat. Data ini kemudian disimpan dalam tabel **Pegawai**.

```

===== Input Data Pegawai =====
Masukkan NIK (Angka): 123
Masukkan Nama Pegawai: elsa
Masukkan Alamat: jogja
Data Pegawai berhasil ditambahkan.

```

Hasil di MySQL :

	NIK	Nama_Pegawai	Alamat
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	123	elsa	jogja

- **Transaksi:** Pengguna diminta untuk memasukkan nomor transaksi, detail transaksi, dan NIK pegawai. Data ini disimpan dalam tabel **Transaksi**.

```

===== Input Data Transaksi =====
Masukkan No Transaksi (TR): TR001
Masukkan Detail Transaksi: Ayam
Masukkan NIK (angka): 123
Data Transaksi berhasil ditambahkan.

```

Hasil di MySQL :

	No_Transaksi	Detail_Transaksi	NIK
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	TR001	Ayam	123

- **Produk:** Pengguna diminta untuk memasukkan kode produk, nomor transaksi, nama produk, jenis produk, dan harga. Data ini disimpan dalam tabel **Produk**.

```

===== Input Data Produk =====
Masukkan Kode Produk (angka): 1
Masukkan No Transaksi (TR): TR001
Masukkan Nama Produk: Ayam
Masukkan Jenis Produk (Snack, Makanan, Minuman): Makanan
Masukkan Harga (angka): 15.000
Data Produk berhasil ditambahkan.

```

Hasil di MySQL :

	Kode_Produk	No_Transaksi	Nama_Produk	Jenis_Produk	Harga
<input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete	1	TR001	Ayam	Makanan	15.000

Hasil yang Diharapkan: Data baru berhasil ditambahkan ke dalam tabel yang sesuai, dan pengguna akan menerima konfirmasi bahwa data telah ditambahkan.

```
===== Menampilkan Data =====  
  
Data Pegawai :  
(123, 'elsa', 'jogja')  
  
Data Transaksi :  
( 'TR001', 'Ayam', 123)  
  
Data Produk :  
(1, 'TR001', 'Ayam', 'Makanan', Decimal('15.000'))
```

2.4 Input Struk

Jika pengguna memilih opsi 3, program akan meminta pengguna untuk memasukkan informasi terkait struk penjualan, termasuk nomor struk, nomor transaksi, NIK pegawai, kode produk, dan jumlah produk.

```
elif menu == 3:  
    print("\n===== Input Data Struk =====")  
    try:  
        No_Struk = input("Masukkan No Struk (ST): ")  
        No_Transaksi = input("Masukkan No Transaksi (TR): ")  
        NIK = int(input("Masukkan NIK Pegawai (angka): "))  
        Kode_Produk = int(input("Masukkan Kode Produk: "))  
        Jumlah_Produk = int(input("Masukkan Jumlah Produk (angka):  
"))  
  
        # Mengambil data Pegawai  
        cur.execute("SELECT Nama_Pegawai FROM Pegawai WHERE NIK =  
%s", (NIK,))  
        result_pegawai = cur.fetchone()  
        if result_pegawai:  
            Nama_Pegawai = result_pegawai[0]  
        else:  
            print("Pegawai dengan NIK tersebut tidak ditemukan.")  
            pass  
  
        # Mengambil data Produk  
        cur.execute("SELECT Nama_Produk, Harga FROM Produk WHERE  
Kode_Produk = %s", (Kode_Produk,))  
        result_produk = cur.fetchone()  
        if result_produk:  
            Nama_Produk, Harga = result_produk  
        else:  
            print("Produk dengan Kode Produk tersebut tidak  
ditemukan.")  
            pass
```

```

# Menghitung Total Harga
Total_Harga = Harga * Jumlah_Produk

# Insert data ke dalam Struk
cur.execute("""INSERT INTO Struk (No_Struk, No_Transaksi,
NIK, Nama_Pegawai, Kode_Produk, Nama_Produk, Jumlah_Produk,
Total_Harga)
VALUES (%s, %s, %s, %s, %s, %s, %s, %s)""",
        (No_Struk, No_Transaksi, NIK, Nama_Pegawai,
Kode_Produk, Nama_Produk, Jumlah_Produk, Total_Harga))
conn.commit()

# Menampilkan Struk
print("\n===== STRUK =====")
print(f"No Struk: {No_Struk}")
print(f"No Transaksi: {No_Transaksi}")
print(f>Nama Pegawai: {Nama_Pegawai}")
print(f>Nama Produk: {Nama_Produk}")
print(f>Jumlah Produk: {Jumlah_Produk}")
print(f>Total Harga: {Total_Harga}")

except Exception as e:
    print(f"Terjadi kesalahan: {e}")

```

- Program akan memverifikasi keberadaan pegawai dan produk berdasarkan NIK dan kode produk yang dimasukkan.
- Total harga dihitung berdasarkan harga produk dan jumlah yang dimasukkan.
- Data struk kemudian disimpan dalam tabel **Struk**.

```

===== Input Data Struk =====
Masukkan No Struk (ST): ST001
Masukkan No Transaksi (TR): TR001
Masukkan NIK Pegawai (angka): 123
Masukkan Kode Produk: 1
Masukkan Jumlah Produk (angka): 3

```

```

===== STRUK =====
No Struk: ST001
No Transaksi: TR001
Nama Pegawai: elsa
Nama Produk: Ayam
Jumlah Produk: 3
Total Harga: 45.000

```

Hasil di MySQL :

No_Struk	No_Transaksi	NIK	Nama_Pegawai	Kode_Produk	Nama_Produk	Jumlah_Produk	Total_Harga
ST001	TR001	123	elsa	1	Ayam	3	45.000

Hasil yang Diharapkan: Struk berhasil ditambahkan, dan informasi struk ditampilkan kepada pengguna.

```
Data Struk :  
( 'ST001', 'TR001', 123, 'elsa', 1, 'Ayam', 3, Decimal('45.000'))
```

2.4 Mengubah Data Struk

Jika pengguna memilih opsi 4, program akan meminta pengguna untuk memasukkan nomor struk yang ingin diubah.

```
elif menu == 4:  
    print("\n==== Ubah Data Struk =====")  
    try:  
        No_Struk = input("Masukkan No Struk yang ingin diubah (ST):  
")  
  
        # Memastikan data Struk ada  
        cur.execute("SELECT * FROM Struk WHERE No_Struk = %s",  
(No_Struk,))  
        if not cur.fetchone():  
            print("Struk dengan No Struk tersebut tidak  
ditemukan.")  
            pass  
  
        No_Transaksi = input("Masukkan No Transaksi Baru (TR): ")  
        NIK = int(input("Masukkan NIK baru (angka): "))  
        Kode_Produk = int(input("Masukkan Kode Produk Baru (angka):  
"))  
  
        Nama_Produk = input("Masukkan Nama Produk Baru: ")  
        Total_Harga = Decimal(input("Total Harga baru (angka): "))  
  
        # Update data Struk  
        cur.execute("""UPDATE Struk SET No_Transaksi = %s, NIK =  
%s, Kode_Produk = %s, Nama_Produk = %s, Total_Harga = %s  
WHERE No_Struk = %s""",  
                    (No_Transaksi, NIK, Kode_Produk, Nama_Produk,  
Total_Harga, No_Struk))  
        conn.commit()  
        print(f>Data Struk dengan No Struk {No_Struk} berhasil  
diubah.")  
  
    except Exception as e:  
        print(f"Terjadi kesalahan: {e}")
```

- Program akan memverifikasi keberadaan struk tersebut.
- Pengguna kemudian dapat memasukkan data baru untuk nomor transaksi, NIK, kode produk, nama produk, dan total harga.

- Data struk yang ada akan diperbarui dengan informasi baru.

```
===== Ubah Data Struk =====
Masukkan No Struk yang ingin diubah (ST): ST001
Masukkan No Transaksi Baru (TR): TR002
Masukkan NIK baru (angka): 321
Masukkan Kode Produk Baru (angka): 2
Masukkan Nama Produk Baru: Es
Total Harga baru (angka): 2.000
Data Struk dengan No Struk ST001 berhasil diubah.
```

Hasil yang Diharapkan: Data struk berhasil diubah, dan pengguna akan menerima konfirmasi.

```
Data Struk :
('ST001', 'TR002', 321, 'elsa', 2, 'Es', 3, Decimal('2.000'))
('ST002', 'TR002', 321, 'ambar', 2, 'Es', 4, Decimal('20.000'))
```

Hasil di MySQL :

No_Struk	No_Transaksi	NIK	Nama_Pegawai	Kode_Produk	Nama_Produk	Jumlah_Produk	Total_Harga
ST001	TR002	321	elsa	2	Es	3	2.000
ST002	TR002	321	ambar	2	Es	4	20.000

2.5 Hapus Data Struk

Jika pengguna memilih opsi 5, program akan meminta pengguna untuk memasukkan nomor struk yang ingin dihapus.

- Program akan memverifikasi keberadaan struk tersebut.
- Jika struk ada, data akan dihapus dari tabel **Struk**.

```
===== Hapus Data Struk =====
Masukkan No Struk yang ingin dihapus (ST): ST002
Data Struk dengan No Struk ST002 berhasil dihapus.
```

Hasil yang Diharapkan: Data struk berhasil dihapus, dan pengguna akan menerima konfirmasi.

```
Data Struk :
('ST001', 'TR002', 321, 'elsa', 2, 'Es', 3, Decimal('2.000'))
```

Hasil di MySQL :

No_Struk	No_Transaksi	NIK	Nama_Pegawai	Kode_Produk	Nama_Produk	Jumlah_Produk	Total_Harga
ST001	TR002	321	elsa	2	Es	3	2.000

BAB III

PENUTUP

3.1 Kesimpulan

Laporan ini membahas pengembangan sistem informasi penjualan menggunakan Python dan MySQL, yang dirancang untuk mengelola data pegawai, transaksi, produk, dan struk penjualan. Sebelum menambah input data lagi, sistem ini memungkinkan pengguna untuk melakukan operasi CRUD (Create, Read, Update, Delete) dengan mudah. Dengan struktur tabel yang jelas dan relasi antar

tabel yang terdefinisi melalui foreign key, pengguna dapat menampilkan, memasukkan, mengubah, dan menghapus data secara efisien.

Hasil yang diharapkan adalah kemudahan dalam pengelolaan data penjualan, meningkatkan produktivitas dan akurasi dalam transaksi. Sistem ini memberikan wawasan berharga untuk pengembangan sistem informasi penjualan di masa depan.