

Laporan Tugas Pemrograman Berbasis Objek
“Program CRUD”

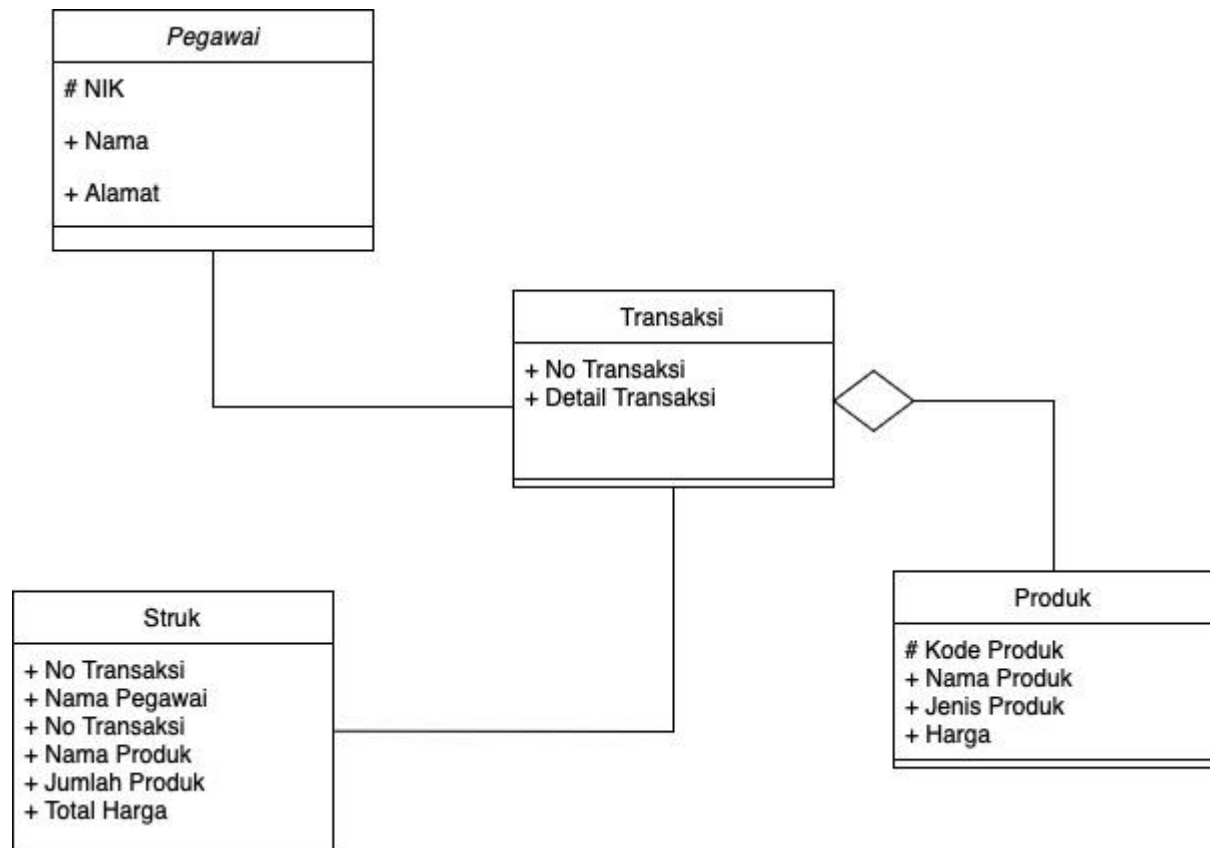


Maulana Rizal Alvani

5230411264

Sarjana Informatika
Universitas Teknologi Yogyakarta

A. Class diagram



B. Menghubungkan Program Python ke Database

```
# Koneksi ke database
try:
    mydb = mysql.connector.connect(
        user="root", password="", host="localhost", database="challenge"
    )
    if mydb.is_connected():
        print("Berhasil terhubung ke database!")
except Error as e:
    print(f"Error: {e}")
    exit()
```

Kode ini memastikan bahwa koneksi ke database MySQL berhasil dibuat, dan jika tidak, program akan memberikan informasi error tanpa melanjutkan ke langkah berikutnya.

Dengan parameter yang diberikan user:"root", host="localhost, database="challenge"
Menentukan nama database yang ingin diakses, yaitu (**challenge**).

C. Membuat Database dan Tabel

```
-- Tabel Pegawai
CREATE TABLE IF NOT EXISTS pegawai (
    nik VARCHAR(20) PRIMARY KEY,
    nama VARCHAR(100) UNIQUE NOT NULL, -- nama dibuat UNIQUE agar dapat
    dirujuk sebagai FOREIGN KEY
    alamat TEXT NOT NULL
);

-- Tabel Produk
CREATE TABLE IF NOT EXISTS produk (
    kode_produk VARCHAR(20) PRIMARY KEY,
    nama_produk VARCHAR(100) UNIQUE NOT NULL, -- nama_produk dibuat UNIQUE
    agar dapat dirujuk sebagai FOREIGN KEY
    jenis_produk VARCHAR(50) NOT NULL,
    harga DECIMAL(10, 2) NOT NULL
);

-- Tabel Transaksi
CREATE TABLE IF NOT EXISTS transaksi (
    no_transaksi VARCHAR(20) PRIMARY KEY,
    detail_transaksi TEXT NOT NULL
);

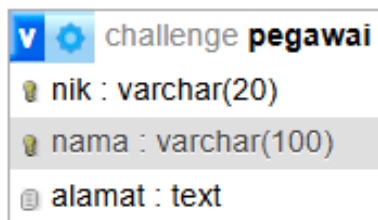
-- Tabel Struk
CREATE TABLE IF NOT EXISTS struk (
    id_struk INT AUTO_INCREMENT PRIMARY KEY,
    no_transaksi VARCHAR(20) NOT NULL,
    nama_pegawai VARCHAR(100) NOT NULL,
    nama_produk VARCHAR(100) NOT NULL,
    jumlah_produk INT NOT NULL,
    total_harga DECIMAL(15, 2) NOT NULL,
    FOREIGN KEY (no_transaksi) REFERENCES transaksi(no_transaksi) ON DELETE
    CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (nama_produk) REFERENCES produk(nama_produk) ON DELETE CASCADE
    ON UPDATE CASCADE,
    FOREIGN KEY (nama_pegawai) REFERENCES pegawai(nama) ON DELETE CASCADE ON
    UPDATE CASCADE
);
```

Databases:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> pegawai		1	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> produk		2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
<input type="checkbox"/> struk		0	InnoDB	utf8mb4_general_ci	64.0 KiB	-
<input type="checkbox"/> transaksi		2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
4 tables	Sum	5	InnoDB	utf8mb4_general_ci	144.0 KiB	0 B

Tabel yang sudah dibuat pada phpMyAdmin

1. Pegawai



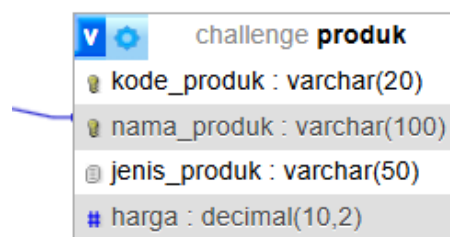
	challenge pegawai
🔑	nik : varchar(20)
🔑	nama : varchar(100)
📄	alamat : text

Kelas ini bertanggung jawab untuk menyimpan data pegawai.

Atribut:

- NIK: Nomor identifikasi pegawai (primary key).
- Nama: Nama pegawai.
- Alamat: Alamat pegawai.

2. Produk



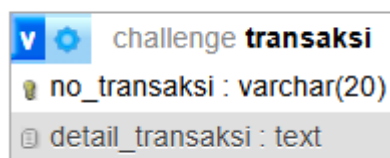
	challenge produk
🔑	kode_produk : varchar(20)
🔑	nama_produk : varchar(100)
📄	jenis_produk : varchar(50)
#	harga : decimal(10,2)

Kelas ini bertugas untuk menyimpan informasi produk.

Atribut:

- Kode Produk: Kode unik untuk identifikasi produk (primary key).
- Nama Produk: Nama produk.
- Jenis Produk: Jenis atau kategori produk.
- Harga: Harga produk.

3. Transaksi



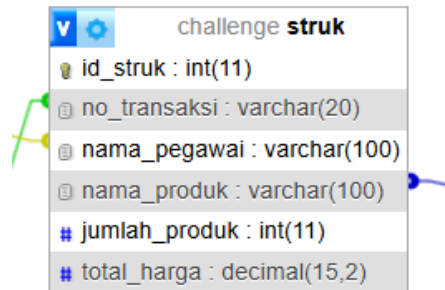
	challenge transaksi
🔑	no_transaksi : varchar(20)
📄	detail_transaksi : text

Kelas ini merepresentasikan data transaksi.

Atribut:

- No Transaksi: Nomor unik transaksi (primary key).
- Detail Transaksi: Detail dari transaksi terkait.

4. Struk



Kelas ini digunakan untuk menyimpan informasi struk transaksi.

Atribut:

- No Transaksi: Nomor transaksi terkait (foreign key ke Transaksi).
- Nama Pegawai: Pegawai yang menangani transaksi.
- Nama Produk: Produk yang dibeli.
- Jumlah Produk: Jumlah item yang dibeli.
- Total Harga: Total harga transaksi.

D. Kode Program

```
class DatabaseCRUD:
    def __init__(self, connection):
        self.conn = connection
        self.cur = connection.cursor()

    def __del__(self):
        self.cur.close()
        self.conn.close()

    # CRUD Pegawai
    def create_pegawai(self, nik, nama, alamat):
        query = "INSERT INTO pegawai (nik, nama, alamat) VALUES (%s, %s, %s)"
        self.execute_query(query, (nik, nama, alamat))

    def read_pegawai(self):
        query = "SELECT * FROM pegawai"
        return self.fetch_query(query)

    def update_pegawai(self, nik, nama, alamat):
        query = "UPDATE pegawai SET nama = %s, alamat = %s WHERE nik = %s"
        self.execute_query(query, (nama, alamat, nik))

    def delete_pegawai(self, nik):
        query = "DELETE FROM pegawai WHERE nik = %s"
        self.execute_query(query, (nik,))
```

```

# CRUD Produk
def create_produk(self, kode_produk, nama_produk, jenis_produk, harga):
    query = "INSERT INTO produk (kode_produk, nama_produk, jenis_produk,
harga) VALUES (%s, %s, %s, %s)"
    self.execute_query(query, (kode_produk, nama_produk, jenis_produk,
harga))

def read_produk(self):
    query = "SELECT * FROM produk"
    return self.fetch_query(query)

def update_produk(self, kode_produk, nama_produk, jenis_produk, harga):
    query = "UPDATE produk SET nama_produk = %s, jenis_produk = %s, harga
= %s WHERE kode_produk = %s"
    self.execute_query(query, (nama_produk, jenis_produk, harga,
kode_produk))

def delete_produk(self, kode_produk):
    query = "DELETE FROM produk WHERE kode_produk = %s"
    self.execute_query(query, (kode_produk,))

# CRUD Transaksi
def create_transaksi(self, no_transaksi, detail_transaksi):
    query = "INSERT INTO transaksi (no_transaksi, detail_transaksi) VALUES
(%s, %s)"
    self.execute_query(query, (no_transaksi, detail_transaksi))

def read_transaksi(self):
    query = "SELECT * FROM transaksi"
    return self.fetch_query(query)

def update_transaksi(self, no_transaksi, detail_transaksi):
    query = "UPDATE transaksi SET detail_transaksi = %s WHERE no_transaksi
= %s"
    self.execute_query(query, (detail_transaksi, no_transaksi))

def delete_transaksi(self, no_transaksi):
    query = "DELETE FROM transaksi WHERE no_transaksi = %s"
    self.execute_query(query, (no_transaksi,))

# CRUD Struk
def create_struk(self, no_transaksi, nama_pegawai, nama_produk,
jumlah_produk, total_harga):
    query = ""
    INSERT INTO struk (no_transaksi, nama_pegawai, nama_produk,
jumlah_produk, total_harga)
VALUES (%s, %s, %s, %s, %s)

```

```

        """
        self.execute_query(query, (no_transaksi, nama_pegawai, nama_produk,
jumlah_produk, total_harga))

    def read_struk(self):
        query = "SELECT * FROM struk"
        return self.fetch_query(query)

    def update_struk(self, no_transaksi, nama_pegawai, nama_produk,
jumlah_produk, total_harga):
        query = """
        UPDATE struk SET nama_pegawai = %s, nama_produk = %s, jumlah_produk =
%s, total_harga = %s
        WHERE no_transaksi = %s
        """
        self.execute_query(query, (nama_pegawai, nama_produk, jumlah_produk,
total_harga, no_transaksi))

    def delete_struk(self, no_transaksi):
        query = "DELETE FROM struk WHERE no_transaksi = %s"
        self.execute_query(query, (no_transaksi,))

# General Methods
def execute_query(self, query, params=None):
    try:
        self.cur.execute(query, params)
        self.conn.commit()
    except Error as e:
        print(f"Error: {e}")

def fetch_query(self, query, params=None):
    try:
        self.cur.execute(query, params)
        return self.cur.fetchall()
    except Error as e:
        print(f"Error: {e}")
    return []

# Menu dan Submenu
if __name__ == "__main__":
    crud = DatabaseCRUD(mydb)

    while True:
        print("\n=== MENU UTAMA ===")
        print("1. Kelola Pegawai")
        print("2. Kelola Produk")
        print("3. Kelola Transaksi")
        print("4. Kelola Struk")

```

```

print("5. Keluar")

menu = input("Pilih menu: ")

if menu == "1":
    while True:
        print("\n--- Kelola Pegawai ---")
        print("1. Tambah Pegawai")
        print("2. Lihat Pegawai")
        print("3. Update Pegawai")
        print("4. Hapus Pegawai")
        print("5. Kembali ke Menu Utama")

        submenu = input("Pilih submenu: ")

        if submenu == "1":
            nik = input("Masukkan NIK: ")
            nama = input("Masukkan Nama: ")
            alamat = input("Masukkan Alamat: ")
            crud.create_pegawai(nik, nama, alamat)
            print("Pegawai berhasil ditambahkan.")
        elif submenu == "2":
            data = crud.read_pegawai()
            for row in data:
                print(row)
        elif submenu == "3":
            nik = input("Masukkan NIK Pegawai yang ingin diupdate: ")
            nama = input("Masukkan Nama Baru: ")
            alamat = input("Masukkan Alamat Baru: ")
            crud.update_pegawai(nik, nama, alamat)
            print("Pegawai berhasil diupdate.")
        elif submenu == "4":
            nik = input("Masukkan NIK Pegawai yang ingin dihapus: ")
            crud.delete_pegawai(nik)
            print("Pegawai berhasil dihapus.")
        elif submenu == "5":
            break
        else:
            print("Pilihan tidak valid.")

    elif menu == "2":
        while True:
            print("\n--- Kelola Produk ---")
            print("1. Tambah Produk")
            print("2. Lihat Produk")
            print("3. Update Produk")
            print("4. Hapus Produk")
            print("5. Kembali ke Menu Utama")

```



```

submenu = input("Pilih submenu: ")

if submenu == "1":
    kode_produk = input("Masukkan Kode Produk: ")
    nama_produk = input("Masukkan Nama Produk: ")
    jenis_produk = input("Masukkan Jenis Produk: ")
    harga = float(input("Masukkan Harga: "))
    crud.create_produk(kode_produk, nama_produk, jenis_produk,
harga)

    print("Produk berhasil ditambahkan.")
elif submenu == "2":
    data = crud.read_produk()
    for row in data:
        print(row)
elif submenu == "3":
    kode_produk = input("Masukkan Kode Produk yang ingin
diupdate: ")

    nama_produk = input("Masukkan Nama Baru: ")
    jenis_produk = input("Masukkan Jenis Baru: ")
    harga = float(input("Masukkan Harga Baru: "))
    crud.update_produk(kode_produk, nama_produk, jenis_produk,
harga)

    print("Produk berhasil diupdate.")
elif submenu == "4":
    kode_produk = input("Masukkan Kode Produk yang ingin
dihapus: ")

    crud.delete_produk(kode_produk)
    print("Produk berhasil dihapus.")
elif submenu == "5":
    break
else:
    print("Pilihan tidak valid.")

elif menu == "3":
    while True:
        print("\n--- Kelola Transaksi ---")
        print("1. Tambah Transaksi")
        print("2. Lihat Transaksi")
        print("3. Update Transaksi")
        print("4. Hapus Transaksi")
        print("5. Kembali ke Menu Utama")

        submenu = input("Pilih submenu: ")

        if submenu == "1":
            no_transaksi = input("Masukkan No Transaksi: ")
            detail_transaksi = input("Masukkan Detail Transaksi: ")

```

```

        crud.create_transaksi(no_transaksi, detail_transaksi)
        print("Transaksi berhasil ditambahkan.")
    elif submenu == "2":
        data = crud.read_transaksi()
        for row in data:
            print(row)
    elif submenu == "3":
        no_transaksi = input("Masukkan No Transaksi yang ingin
diupdate: ")
        detail_transaksi = input("Masukkan Detail Transaksi Baru:
")
        crud.update_transaksi(no_transaksi, detail_transaksi)
        print("Transaksi berhasil diupdate.")
    elif submenu == "4":
        no_transaksi = input("Masukkan No Transaksi yang ingin
dihapus: ")
        crud.delete_transaksi(no_transaksi)
        print("Transaksi berhasil dihapus.")
    elif submenu == "5":
        break
    else:
        print("Pilihan tidak valid.")

elif menu == "4":
    while True:
        print("\n--- Kelola Struk ---")
        print("1. Tambah Struk")
        print("2. Lihat Struk")
        print("3. Update Struk")
        print("4. Hapus Struk")
        print("5. Kembali ke Menu Utama")

        submenu = input("Pilih submenu: ")

        if submenu == "1":
            no_transaksi = input("Masukkan No Transaksi: ")
            nama_pegawai = input("Masukkan Nama Pegawai: ")
            nama_produk = input("Masukkan Nama Produk: ")
            jumlah_produk = int(input("Masukkan Jumlah Produk: "))
            total_harga = float(input("Masukkan Total Harga: "))
            crud.create_struk(no_transaksi, nama_pegawai, nama_produk,
jumlah_produk, total_harga)
            print("Struk berhasil ditambahkan.")
        elif submenu == "2":
            data = crud.read_struk()
            for row in data:
                print(row)
        elif submenu == "3":

```

```

        no_transaksi = input("Masukkan No Transaksi yang ingin
diupdate: ")
        nama_pegawai = input("Masukkan Nama Pegawai Baru: ")
        nama_produk = input("Masukkan Nama Produk Baru: ")
        jumlah_produk = int(input("Masukkan Jumlah Produk Baru:
"))
        total_harga = float(input("Masukkan Total Harga Baru: "))
        crud.update_struk(no_transaksi, nama_pegawai, nama_produk,
jumlah_produk, total_harga)
        print("Struk berhasil diupdate.")
    elif submenu == "4":
        no_transaksi = input("Masukkan No Transaksi yang ingin
dihapus: ")
        crud.delete_struk(no_transaksi)
        print("Struk berhasil dihapus.")
    elif submenu == "5":
        break
    else:
        print("Pilihan tidak valid.")

elif menu == "5":
    print("Keluar dari program.")
    break
else:
    print("Pilihan tidak valid.")

```

Terdapat lima pilihan utama:

- Tambah Data: Menambah data baru ke database.
- Lihat Data: Menampilkan data yang ada di database.
- Update Data: Mengubah data yang sudah ada.
- Hapus Data: Menghapus data dari database.
- Kembali: Mengakhiri program.

Operasi CRUD

1. Create (Menambah data)

Operasi ini digunakan untuk menambahkan data baru ke dalam basis data.







```

=== MENU UTAMA ===
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 1

--- Kelola Pegawai ---
1. Tambah Pegawai
2. Lihat Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan NIK: 8890
Masukkan Nama: noni
Masukkan Alamat: jogja
Pegawai berhasil ditambahkan.

--- Kelola Pegawai ---
1. Tambah Pegawai
2. Lihat Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan NIK: 8891
Masukkan Nama: agus
Masukkan Alamat: sleman
Pegawai berhasil ditambahkan.

```

					nik	nama	alamat
<input type="checkbox"/>		Edit		Copy		Delete	8890 noni jogja
<input type="checkbox"/>		Edit		Copy		Delete	8891 agus sleman

```

=== MENU UTAMA ===
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 2

--- Kelola Produk ---
1. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan Kode Produk: 12
Masukkan Nama Produk: Pizza
Masukkan Jenis Produk: makanan
Masukkan Harga: 50000
Produk berhasil ditambahkan.

--- Kelola Produk ---
1. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan Kode Produk: 13
Masukkan Nama Produk: Burger
Masukkan Jenis Produk: makanan
Masukkan Harga: 50000
Produk berhasil ditambahkan.

```

<div><div><div>←</div><div>T</div><div>→</div></div></div>					kode_produk	nama_produk	jenis_produk	harga
<div><div><div><div></div></div></div><div><div><div>Edit</div><div><div><div></div><div></div><div></div></div></div><div>Copy</div><div><div><div></div></div></div><div>Delete</div></div></div></div>	12	Pizza	makanan	50000.00				
<div><div><div><div></div></div></div><div><div><div>Edit</div><div><div><div></div><div></div><div></div></div></div><div>Copy</div><div><div><div></div></div></div><div>Delete</div></div></div></div>	13	Burger	makanan	50000.00				







```

=== MENU UTAMA ===
1. Kelola Pegawai
2. Kelola Produk
3. Kelola Transaksi
4. Kelola Struk
5. Keluar
Pilih menu: 3

--- Kelola Transaksi ---
1. Tambah Transaksi
2. Lihat Transaksi
3. Update Transaksi
4. Hapus Transaksi
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan No Transaksi: 122
Masukkan Detail Transaksi: Fresh
Transaksi berhasil ditambahkan.

--- Kelola Transaksi ---
1. Tambah Transaksi
2. Lihat Transaksi
3. Update Transaksi
4. Hapus Transaksi
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan No Transaksi: 123
Masukkan Detail Transaksi: Fresh
Transaksi berhasil ditambahkan.

```

				no_transaksi	detail_transaksi
<input type="checkbox"/>	 Edit	 Copy	 Delete	122	Fresh
<input type="checkbox"/>	 Edit	 Copy	 Delete	123	Fresh

```

--- Kelola Struk ---
1. Tambah Struk
2. Lihat Struk
3. Update Struk
4. Hapus Struk
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan No Transaksi: 122
Masukkan Nama Pegawai: agus
Masukkan Nama Produk: Burger
Masukkan Jumlah Produk: 5
Masukkan Total Harga: 25000
Struk berhasil ditambahkan.

--- Kelola Struk ---
1. Tambah Struk
2. Lihat Struk
3. Update Struk
4. Hapus Struk
5. Kembali ke Menu Utama
Pilih submenu: 1
Masukkan No Transaksi: 123
Masukkan Nama Pegawai: noni
Masukkan Nama Produk: Pizza
Masukkan Jumlah Produk: 4
Masukkan Total Harga: 200000
Struk berhasil ditambahkan.

```

				id_struk	no_transaksi	nama_pegawai	nama_produk	jumlah_produk	total_harga
<input type="checkbox"/>	 Edit	 Copy	 Delete	6	122	agus	Burger	5	25000.00
<input type="checkbox"/>	 Edit	 Copy	 Delete	7	123	noni	Pizza	4	200000.00

2. Read (Menampilkan Data)

Operasi ini digunakan untuk membaca atau menampilkan data yang ada dalam basis data.

--- Kelola Produk ---

1. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Kembali ke Menu Utama

Pilih submenu: 2

('12', 'Pizza', 'makanan', Decimal('50000.00'))
('13', 'Burger', 'makanan', Decimal('50000.00'))

--- Kelola Transaksi ---

1. Tambah Transaksi
2. Lihat Transaksi
3. Update Transaksi
4. Hapus Transaksi
5. Kembali ke Menu Utama

Pilih submenu: 2

('122', 'Fresh')
('123', 'Fresh')

--- Kelola Struk ---

1. Tambah Struk
2. Lihat Struk
3. Update Struk
4. Hapus Struk
5. Kembali ke Menu Utama

Pilih submenu: 2

(6, '122', 'agus', 'Burger', 5, Decimal('25000.00'))
(7, '123', 'noni', 'Pizza', 4, Decimal('20000.00'))

--- Kelola Pegawai ---

1. Tambah Pegawai
2. Lihat Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Kembali ke Menu Utama

Pilih submenu: 2

('8890', 'noni', 'jogja')
('8891', 'agus', 'sleman')

C. Update (Memperbarui Data)

Operasi ini digunakan untuk memperbarui data yang sudah ada dalam basis data.

--- Kelola Produk ---

1. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Kembali ke Menu Utama

Pilih submenu: 3

Masukkan Kode Produk yang ingin diupdate: 13
Masukkan Nama Baru: Spagheti
Masukkan Jenis Baru: Makanan
Masukkan Harga Baru: 60000
Produk berhasil diupdate.

--- Kelola Transaksi ---

1. Tambah Transaksi
2. Lihat Transaksi
3. Update Transaksi
4. Hapus Transaksi
5. Kembali ke Menu Utama

Pilih submenu: 3

Masukkan No Transaksi yang ingin diupdate: 122
Masukkan Detail Transaksi Baru: Mantab
Transaksi berhasil diupdate.

--- Kelola Struk ---

1. Tambah Struk
2. Lihat Struk
3. Update Struk
4. Hapus Struk
5. Kembali ke Menu Utama

Pilih submenu: 3

Masukkan No Transaksi yang ingin diupdate: 122
Masukkan Nama Pegawai Baru: Bambang
Masukkan Nama Produk Baru: Pizza
Masukkan Jumlah Produk Baru: 5
Masukkan Total Harga Baru: 250000
Struk berhasil diupdate.

--- Kelola Pegawai ---

1. Tambah Pegawai
2. Lihat Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Kembali ke Menu Utama

Pilih submenu: 3

Masukkan NIK Pegawai yang ingin diupdate: 8890
Masukkan Nama Baru: Bambang
Masukkan Alamat Baru: bantul
Pegawai berhasil diupdate.

D. Delete (Menghapus Data)

Operasi ini digunakan untuk menghapus data dari basis data. Misalnya, menghapus data transaksi yang sudah tidak diperlukan.

```
--- Kelola Pegawai ---
```

1. Tambah Pegawai
2. Lihat Pegawai
3. Update Pegawai
4. Hapus Pegawai
5. Kembali ke Menu Utama

```
Pilih submenu: 4
```

```
Masukkan NIK Pegawai yang ingin dihapus: 8890  
Pegawai berhasil dihapus.
```

```
--- Kelola Produk ---
```

1. Tambah Produk
2. Lihat Produk
3. Update Produk
4. Hapus Produk
5. Kembali ke Menu Utama

```
Pilih submenu: 4
```

```
Masukkan Kode Produk yang ingin dihapus: 12  
Produk berhasil dihapus.
```

```
--- Kelola Transaksi ---
```

1. Tambah Transaksi
2. Lihat Transaksi
3. Update Transaksi
4. Hapus Transaksi
5. Kembali ke Menu Utama

```
Pilih submenu: 4
```

```
Masukkan No Transaksi yang ingin dihapus: 122  
Transaksi berhasil dihapus.
```

```
--- Kelola Struk ---
```

1. Tambah Struk
2. Lihat Struk
3. Update Struk
4. Hapus Struk
5. Kembali ke Menu Utama

```
Pilih submenu: 4
```

```
Masukkan No Transaksi yang ingin dihapus: 122  
Struk berhasil dihapus.
```

Kesimpulan

Implementasi CRUD dengan Python dan MySQL dalam laporan praktikum ini adalah bahwa program ini berhasil menunjukkan cara dasar untuk mengelola data dalam basis data menggunakan operasi CRUD (Create, Read, Update, Delete). Program ini menggunakan pustaka mysql.connector untuk menghubungkan Python dengan MySQL dan menyediakan antarmuka menu yang memungkinkan pengguna untuk menambah, menampilkan, mengubah, dan menghapus data dalam tabel-tabel basis data seperti pegawai, produk, transaksi, dan struk.