TUGAS PEMROGRAMAN BERORIENTASI OBJEK PRAKTIK

Disusun oleh :

Daphne Aisya Raflian | 5230411325

PRODI STUDI INFORMATIKA

UNIVERSITAS TEKNOLOGI YOGYAKARTA

FAKULTAS SAINS & TEKNOLOGI

2023

```
import mysql.connector
```

Sebelum membuat database, aktifkan venv menggunakan terminal dengan mengetik "python -m venv venv" dan "venv Scripts activate". Setelah itu, kalian dapat menginstalnya dengan mengetik "pip install mysql.connector".

```
try:
    conn = mysql.connector.connect(
        user="root",
        host="localhost",
        password="",
        database="Penjualan"
    )
    cur = conn.cursor()

    # Membuat database jika belum ada
    cur.execute("CREATE DATABASE IF NOT EXISTS Penjualan;")
    print("Database 'Penjualan' siap atau sudah ada.")
```

Kemudian Anda dapat menghubungkan database Anda dengan mengetikan user, hots, pass, dan nama database Anda. Karena saya tidak menggunakan password, Anda harus tidak harus memberikan password terlebih dahulu agar dapat terkoneksi.

```
# Membuat tabel jika belum ada
cur.execute("""
CREATE TABLE IF NOT EXISTS Pegawai (
    NIK CHAR(4) NOT NULL PRIMARY KEY,
    Nama VARCHAR(50),
    Alamat VARCHAR(255)
)
""")
print("Tabel 'Pegawai' berhasil dibuat atau sudah ada.")

cur.execute("""
CREATE TABLE IF NOT EXISTS Produk (
    Kode_Produk CHAR(4) NOT NULL PRIMARY KEY,
    Nama_Produk VARCHAR(50),
    Jenis_Produk VARCHAR(50),
    Harga INT
)
""")
print("Tabel 'Produk' berhasil dibuat atau sudah ada.")

cur.execute("""
CREATE TABLE IF NOT EXISTS Transaksi (
    No_Transaksi CHAR(20) NOT NULL PRIMARY KEY,
    Detail_Transaksi VARCHAR(255),
    Nama VARCHAR(50),
    Kode_Produk CHAR(4),
    FOREIGN KEY (Nama) REFERENCES Pegawai(Nama),
    FOREIGN KEY (Kode_Produk) REFERENCES Produk(Kode_Produk)
)
""")

print("Tabel 'Transaksi' berhasil dibuat atau sudah ada.")
```

Mengisi table yang ingin Anda masukan ke database dengan mengisi semua atributnya dan menyesuaikannya sesuai kebutuhan. Misalnya, jika Anda ingin menaruh "HARGA", Anda harus menggunakan INT karena harga biasanya ditulis dengan angka daripada huruf, dan jika Anda menggunakan VARCHAR, pengguna dapat memasukan huruf ke kolom harga. Karena itu, sesuaikan dengan apa yang Anda inginkan. Misalnya, saya merubah tombol utama menjadi tombol akhir untuk fungsi mengambil atribut dari table lain untuk digunakan di table yang tidak memilikinya dengan menggunakan "REFERENCES". Jika berhasil, akan muncul pesan, "Table "Struk" berhasil dibuat atau sudah ada."

```
# Menambahkan satu data awal ke tabel Produk
cur.execute("""
INSERT IGNORE INTO Produk (Kode_Produk, Nama_Produk, Jenis_Produk, Harga)
VALUES ('Z01', 'Kopi', 'Minuman', 35000)
""")
conn.commit()
print("Data produk awal berhasil ditambahkan.")
```

Untuk memulai, saya akan memasukkan uji coba data untuk melihat apakah koneksinya terhubung ke table produk. Saya akan menggunakan "INSERT INTO" dan memanggil atribut sesuai dengan urutan yang saya masukan di datanya, yang memiliki total empat data.

```python
# Menu aplikasi
Codeium: Refactor | Explain | Generate Docstring | ×
def menu():
    while True:
        print("\n=== Menu Aplikasi ===")
        print("1. Tambah Produk")
        print("2. Hapus Produk")
        print("3. Tampilkan Produk")
        print("4. Keluar")
        pilihan = input("Pilih menu: ")

        if pilihan == "1":
            tambah_produk()
        elif pilihan == "2":
            hapus_produk()
        elif pilihan == "3":
            tampilkan_produk()
        elif pilihan == "4":
            print("Keluar dari aplikasi.")
            break
        else:
            print("Pilihan tidak valid. Coba lagi.")
```

Di sini, saya menggunakan while true untuk membuat fungsi yang dapat menyimpan tampilan menu, dan kemudian saya membuat permisalan menggunakan if else. Jika pengguna mengetikan satu, itu akan mengarah ke fungsi yang sudah saya buat sesuai dengan angka yang dimasukkan. Jika pengguna memasukan angka yang tidak ada di menu, maka akan muncul notifikasi berupa "Pilihan tidak valid. Coba lagi."

Saya membuat fungsi yang akan bekerja. Dengan mengetikan perintah di menu utama, seperti "1", fungsi akan mengarah ke "tambah_produk". Saat ingin menambah produk, pengguna harus mengisi kode, nama, jenis, dan harga produk. Setelah mengisi, produk akan dimasukkan ke dalam table yang telah saya buat sebelumnya dengan "INSERT INTO", dan % akan menampungnya.

```
Database 'manajemen_produk2' siap atau sudah ada.
Tabel 'Pegawai' berhasil dibuat atau sudah ada.
Tabel 'Produk' berhasil dibuat atau sudah ada.
Tabel 'Transaksi' berhasil dibuat atau sudah ada.
Tabel 'Struk' berhasil dibuat atau sudah ada.
Data produk awal berhasil ditambahkan.

=== Menu Aplikasi ===
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Keluar
Pilih menu:
```

Ketika Anda menjalankan VS-kode, table seperti ini akan muncul. Jika Anda memasukkan angka "1", Anda dapat melakukan penambahan.

```
=== Menu Aplikasi ===
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Keluar
Pilih menu: 1

=== Tambah Produk ===
Masukkan kode produk: Y01
Masukkan nama produk: Susu
Masukkan jenis produk: Minuman
Masukkan harga produk: 5000
Produk berhasil ditambahkan!

=== Menu Aplikasi ===
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Keluar
Pilih menu:
```

Setelah menambahkan, notifikasi akan muncul, "Produk berhasil ditambahkan!"

```
Kode: T01, Nama: Jagung, Jenis: Makanan, Harga: 6000
Kode: Y01, Nama: Susu, Jenis: Minuman, Harga: 5000

=== Menu Aplikasi ===
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Keluar
Pilih menu:
```

Tampilkan semua produk opsi ke 3

Menghapus id T01 dengan opsi ke 3

Kesimpulan:
Program ini menghubungkan ke database MySQL bernama "Penjualan" dan memastikan bahwa tabel seperti "Pegawai", "Produk", "Transaksi", dan "Struk" ada. Jika tidak ada, tabel tersebut akan dibuat. Agar tidak ada duplikat, perintah "INSERT IGNORE" digunakan untuk memasukkan data awal produk ke dalam tabel "Produk". Selain itu, aplikasi ini memiliki menu interaktif yang memungkinkan pengguna menambah, menghapus, dan menampilkan produk yang sudah ada dalam database. Struktur exception handling memungkinkan program untuk menangani kesalahan yang mungkin terjadi saat berinteraksi dengan database. Secara keseluruhan, aplikasi ini menyediakan antarmuka yang mudah digunakan untuk mengelola data produk di sistem penjualan.