

TUGAS PBOP
PEMROGRAMAN BERORIENTASI OBJEK PRAKTIK



D I S U S U N

OLEH

Daffa Nayottama

5230411234

Prodi : Informatika – E

UNIVERSITAS TEKNOLOGI YOGYAKARTA

FAKULTAS SAINS & TEKNOLOGI

Tahun Akademik 2023

1. Pengertian

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (DBMS) yang multithread dan multi-user. MySQL digunakan untuk mengelola dan menyimpan data dalam bentuk tabel-tabel yang dapat diakses dan dimanipulasi menggunakan perintah SQL. MySQL dapat digunakan pada berbagai platform, termasuk Windows, Linux, dan macOS. MySQL juga mendukung berbagai bahasa pemrograman, seperti PHP, Python, dan Java. Selain itu, MySQL juga dapat diintegrasikan dengan berbagai aplikasi web, seperti WordPress dan Joomla. MySQL juga memiliki fitur-fitur keamanan yang baik, seperti autentikasi pengguna dan enkripsi data.

- **Codingan:**

```
C: > kuliah > smt 3 > pemograman berbasis object praktik > uas > tugas_1 > tugas.py > ...
1  import mysql.connector
2
```

- Sebelum kalian membuat Database kalian harus menginstal ini terlebih dahulu dengan cara meng-aktifkan venv kalian melalui terminal dengan mengetikan:
- “python -m venv venv”
- “venv\Scripts\activate”
- Baru kalian bisa menginstallnya “pip install mysql.connector”

```
# Membuat koneksi ke database
try:
    conn = mysql.connector.connect(
        user="root",
        host="localhost",
        password="Dfaaoam17022005.",
        database="manajemen_produk2"
    )
    cur = conn.cursor()
```

- Kemudian kalian bisa langsung mengkoneksikan database kalian dengan mengetikan user, host, pass, dan nama dari database kalian, karena disini saya menggunakan password maka saya harus memasukan password saya terlebih dahulu agar bisa terkoneksi jika kalian tidak memakai maka kalian bisa kosongkan saya passwordnya.

```
cur.execute("CREATE DATABASE manajemen_produk2;")
print("Database 'manajemen_produk2' siap atau sudah ada.")
```

- Membuat database yang ingin kalian pakai dengan memasukkan nama database kalian yang akan digunakan untuk menyimpan table yang nanti kalian buat

```
cur.execute("""
CREATE TABLE Pegawai (
    NIK CHAR(4) NOT NULL PRIMARY KEY,
    Nama VARCHAR(50),
    Alamat VARCHAR(255)
)
""")

cur.execute("""
CREATE TABLE 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 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)
)
""")

cur.execute("""
CREATE TABLE Struk (
    No_Transaksi CHAR(4),
    Nama_Pegawai VARCHAR(50),
    Nama_Produk VARCHAR(50),
    Jumlah_Produk INT,
    Total_Harga INT,
    FOREIGN KEY (No_Transaksi) REFERENCES Transaksi(No_Transaksi)
)
""")
print("Tabel 'Struk' berhasil dibuat atau sudah ada.")
```

- Membuat table yang ingin kalian masukan ke database kalian dengan mengisi atribut apa yang ada pada table kalian dan menyesuaikan kebutuhan di table kalian misal kalian ingin menaruh “HARGA” maka kalian harus menggunakan INT karena biasanya harga di ketikan dengan angka bukan menggunakan huruf dan jika menggunakan VARCHAR kalian user akan bisa memasukan huruf di kolom harga tersebut. Oleh karena itu sesuaikan dengan apa yang kalian inginkan
- Disini saya merubah primary key menjadi forend key fungsi untuk mengambil atribut dari table lain untuk di gunakan di table yang tidak ada atribut tersebut dengan mengaksesnya melalui “REFERENCES”
- Dan jika berhasil maka akan memunculkan “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 ('P001', 'Lampu LED', 'Elektronik', 50000)
""")
conn.commit()
print("Data produk awal berhasil ditambahkan.")
```

- Saya akan memasukan uji coba data terlebih dahulu agar bisa mengecek koneksinya terhubung atau tidak pada table product dengan memakai “INSERT INTO” dan memanggil atribut sesuai dengan urutan yang saya masukan di datanya dengan total 4 data

```
# Menu aplikasi
Codeium: Refactor | Explain | Generate Docstring | X
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.")
```

- Disini saya membuat fungsi yang menyimpan sebuah tampilan menu dengan menggunakan while true, kemudian saya membuat permissalan menggunakan if else dengan memberitahukan jika user mengetikkan satu maka akan mengarah ke sebuah fungsi yang sudah saya buat sesuai dengan angka yang dimasukan.
- Jika user memasukan angka selain yang ada di menu maka akan memberikan notifikasi berupa "Pilihan tidak valid. Coba lagi."

```

Codeium: Refactor | Explain | Generate Docstring | X
def tambah_produk():
    print("\n== Tambah Produk ==")
    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("Produk berhasil ditambahkan!")

Codeium: Refactor | Explain | Generate Docstring | X
def hapus_produk():
    print("\n== Hapus Produk ==")
    kode_produk = input("Masukkan kode produk yang ingin dihapus: ")
    cur.execute("""
        DELETE FROM Produk WHERE Kode_Produk = %s
    """, (kode_produk,))
    conn.commit()
    print("Produk berhasil dihapus!")

Codeium: Refactor | Explain | Generate Docstring | X
def tampilkan_produk():
    print("\n== Daftar Produk ==")
    cur.execute("SELECT * FROM Produk")
    result = cur.fetchall()
    if not result:
        print("Belum ada produk yang terdaftar.")
    else:
        for row in result:
            print(f"Kode: {row[0]}, Nama: {row[1]}, Jenis: {row[2]}, Harga: {row[3]}")

```

- Saya membuat fungsi dimana fungsi tersebut akan bekerja Ketika user mengetikkan perintah di menu utama tadi misal user mengetikkan "1" maka akan mengarah langsung ke "tambah_produk" dan Ketika menambah produk user harus meng-inputkan kode, nama, jenis, dan juga haraga dari produk tersebut. Jika sudah mengisi maka produk tersebut akan di tambahkan ke dalam table yang sudah di buat tadi dengan "INSERT INTO" dan % akan menampungnya, jika sudah maka akan memberikan notifikasi yaitu "Produk berhasil ditambahkan!".
- Jika mengetikkan 2 maka user harus memasukan kode produk untuk mengapus produk tersebut
- Jika mengetikkan 3 maka akan menampilkan semua produk yang berada dalam table produk yang disimpan.

```

# Menjalankan menu
if __name__ == "__main__":
    menu()

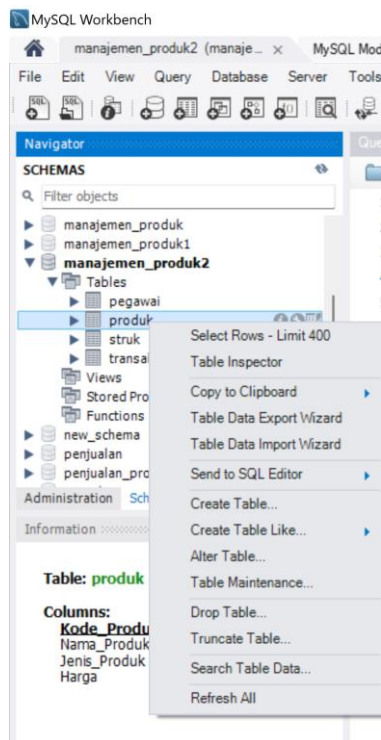
```

- Fungsi `if __name__ == "__main__":` adalah untuk menjalankan semua menu yang ada pada menu utama

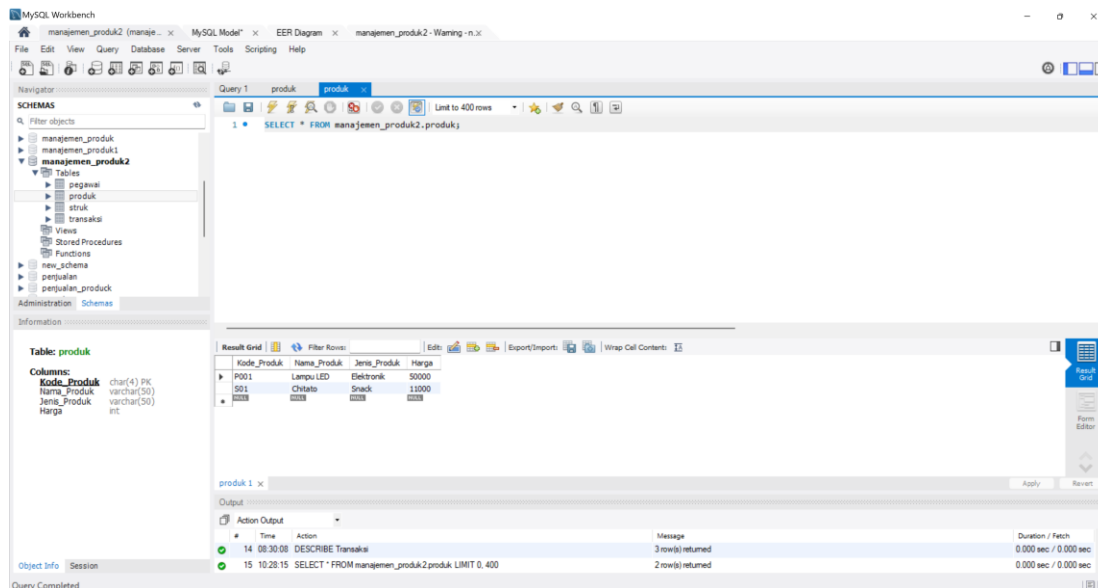
2. MySQL Workbench:



- Ketika kalian sudah membuat maka akan langsung muncul secara otomatis di MySQL Workbench kalian dan karena kita memberi nama Database kita adalah “manajemen_produk2” maka table yang kita buat akan masuk kedata



- Kalian bisa klik kanan pada produk lalu tekann yang bertulisan “Select Rows – Limit 400 untuk menampilkan data yang sebelumnya kita masukan

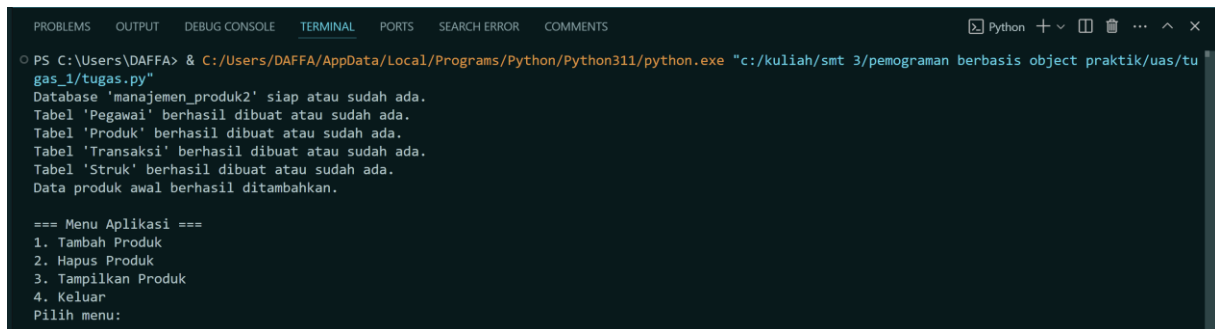


- Maka data di Dalam produk akan muncul seperti ini

SHOW TABLES;

- Kalian bisa mengetikkan “SHOW TABLES;” untuk melihat tablenya dan klik logo petir untuk me run

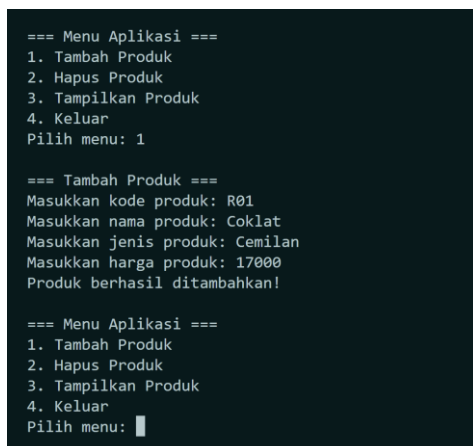
3. Implementasi Ketika di run melalui VS-code



```
PS C:\Users\DAFFA> & C:/Users/DAFFA/AppData/Local/Programs/Python/Python311/python.exe "c:/kuliah/smt 3/pemograman berbasis object praktik/uas/tugas_1/tugas.py"
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 di run VS-code akan muncul table seperti ini dan kalian bisa melakukan penambahan jika mau menambahkan suatu produk jika memasukan angka “1”



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

=== Tambah Produk ===
Masukkan kode produk: R01
Masukkan nama produk: Coklat
Masukkan jenis produk: Cemilan
Masukkan harga produk: 17000
Produk berhasil ditambahkan!

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

- Jika kalian berhasil menambahkan maka akan muncul notifikasi “Produk berhasil ditambahkan!”


```

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

=== Daftar Produk ===
Kode: P001, Nama: Lampu LED, Jenis: Elektronik, Harga: 50000
Kode: R01, Nama: Coklat, Jenis: Cemilan, Harga: 17000
Kode: S01, Nama: Chitato, Jenis: Snack, Harga: 11000

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

```

- Jika kalian menekan 3 maka akan muncul tampilan semua produk yang sudah kalian tambahkan pada menu “Tambah Produk”

```

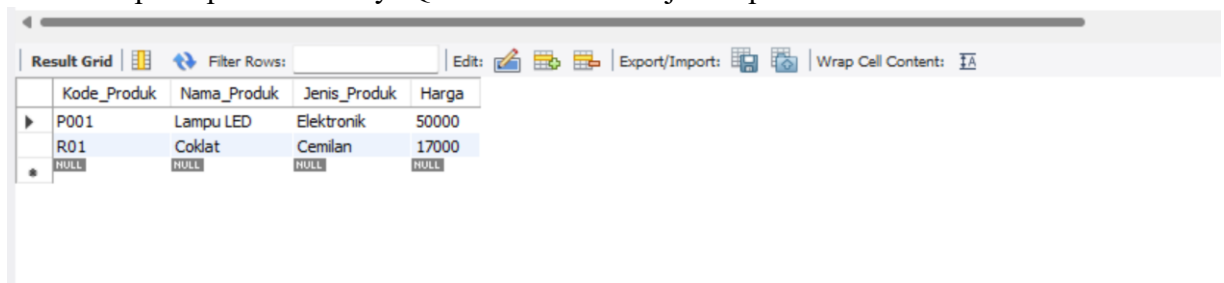
=== Menu Aplikasi ===
1. Tambah Produk
2. Hapus Produk
3. Tampilkan Produk
4. Keluar
Pilih menu: 2

=== Hapus Produk ===
Masukkan kode produk yang ingin dihapus: S01
Produk berhasil dihapus!

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

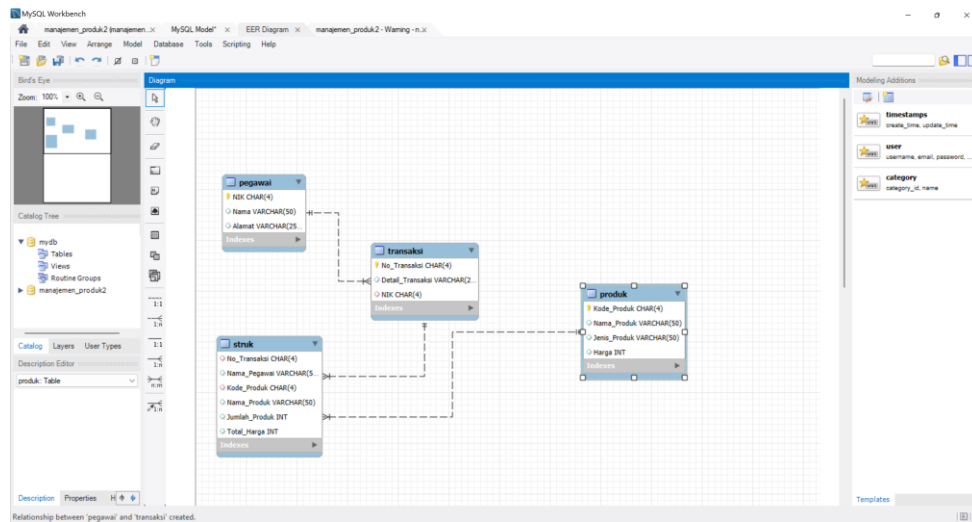
```

- Dan jika kalian ingin menghapus maka kalian harus memasukan kode produk terlebih dahulu agar bisa menghapus produk tersebut di dalam table produk
- Maka tampilan pada table MySQL kalian akan menjadi seperti ini

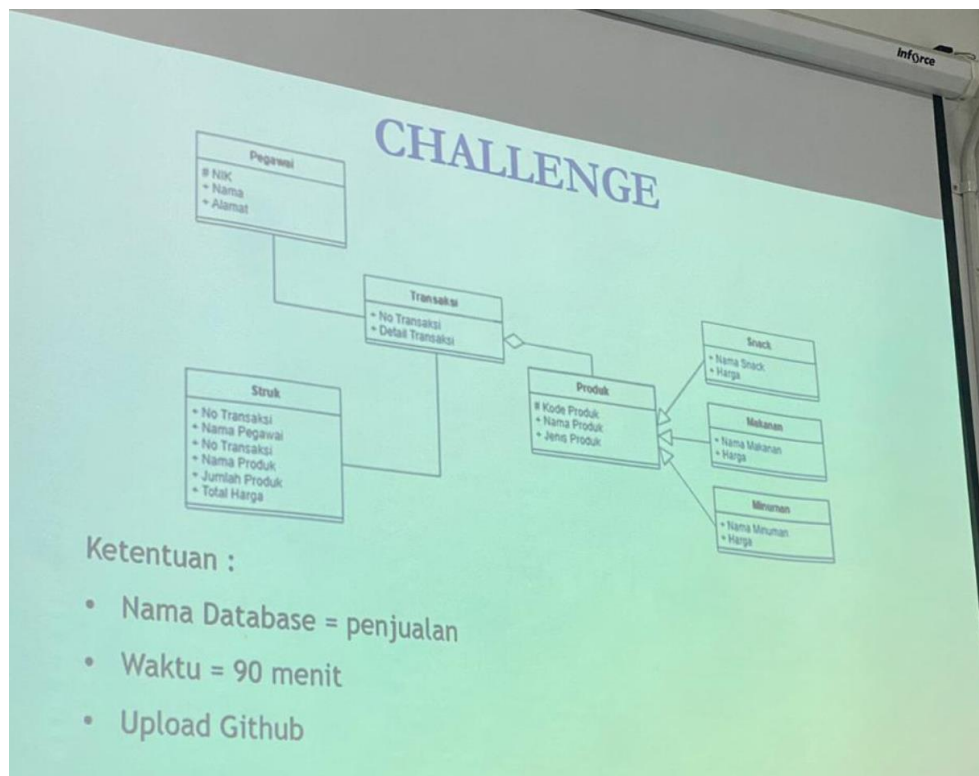


Kode_Produk	Nama_Produk	Jenis_Produk	Harga
P001	Lampu LED	Elektronik	50000
R01	Coklat	Cemilan	17000
NULL	NULL	NULL	NULL

4. Table yang terbuat dari query itu akan di implementasikan di bagian fitur models pada MySQL



Gambar 1



Gambar 2

5. Kesimpulan

Meskipun strukturnya sedikit berbeda dalam cara mereka memperlihatkan hubungan antara tabel-tabel. Tetapi gambar 1 menunjukkan hubungan antara Pegawai, Transaksi, dan Produk secara langsung melalui simbol-simbol primary key dan foreign key. Ini mencerminkan hubungan 1:1, 1:M, dan M:N antara tabel tersebut.

Gambar 2, meskipun menggunakan tabel Struk sebagai intermediary untuk menghubungkan Transaksi dan Produk, masih mencerminkan logika yang sama. Tabel Struk digunakan untuk menciptakan hubungan many-to-many antara Produk dan Transaksi, dengan relasi yang memungkinkan transaksi terhubung dengan beberapa produk dan produk terhubung dengan beberapa transaksi. Secara keseluruhan, baik gambar pertama maupun kedua menggambarkan relasi yang konsisten antara tabel-tabel dalam skema databasenya.