

## Pertemuan 11

### Database MySQL



#### CAPAIAN PEMBELAJARAN

---

1. Membuat program memanipulasi database: menambah, membaca, mengubah rekaman tabel menggunakan Python
2. Menerapkan dengan kasus



#### KEBUTUHAN ALAT/BAHAN/SOFTWARE

---

1. Sistem Operasi Linux/Window 10
2. Database MySQL (XAMPP)
3. `mysql.connector`
4. Teks Editor



#### DASAR TEORI

---

#### Koneksi database dengan Python

Koneksi Database Python MySQL menggunakan Konektor MySQL

Dengan Python, bisa modul berikut untuk berkomunikasi dengan MySQL: MySQL Connector Python, PyMySQL, MySQLDB, MySQLClient, OurSQL

```
pip install mysql-connector-python
```

#### Cara Koneksi :

Berikut ini koneksi dari Python ke MySQL

```
import mysql.connector

mydb = mysql.connector.connect(
    host="localhost",
```

```
user="nama_user",  
password="kunci"  
)
```

Terpadat beberapa method yang digunakan untuk memanipulasi database:

- method `connect()`, dengan argumen yang diperlukan. Maka akan mengembalikan objek `MySQLConnection`
- method `cursor()`, untuk membuat objek kursor untuk melakukan berbagai operasi SQL.
- Method `execute()` menjalankan kueri SQL dan mengembalikan hasilnya.
- Ekstrak hasil menggunakan `fetchall()`
- Gunakan `cursor.fetchall()` atau `fetchone()` atau `fetchmany()` untuk membaca hasil query.
- menutup cursor dan objek koneksi gunakan method `cursor.close()` dan `connection.close()`



## PRAKTIK

---

### 1. Instalasi `mysql-connector-python`

sebelum membuat program koneksi ke data base MySQL Anda harus menambahkan paket driver `mysql-connector-python` dengan perintah:

```
pip install mysql-connector-python
```

### 2. Siapkan database, jalankan Server database MySQL nya, Anda bisa membuat database baru seperti pada perintah berikut:

```
C:\xampp\mysql\bin\> mysql -u root
```

```
MySQL>CREATE DATABASE db_jual1;
```

### 3. Memuat program untuk koneksi ke database `db_jual1` dengan menggunakan Python, buatlah program berikut:

```
*****  
# praktik11-1.py  
# NIM :.....  
# NAMA :.....  
*****  
import mysql.connector
```

```

from mysql.connector import Error

try:
    koneksi = mysql.connector.connect(host='localhost',
                                     database='db_jual1',
                                     user='ana',
                                     password='123***')

    #meminta koneksi
    if koneksi.is_connected():
        db_Info = koneksi.get_server_info()
        print("Koneksi ke MySQL Server versi: ", db_Info)
        cursor = koneksi.cursor()
        cursor.execute("select database();")
        record = cursor.fetchone()
        print("Anda Telah koneksi ke Database : ", record)
#cek jika terjadi kesalahan
except koneksi.Error as e:
    print("Terjadi kesalahan ", e)
finally:
    if koneksi.is_connected():
        cursor.close()
        koneksi.close()
        print("Koneksi ke MySQL Telah ditutup")

```

Uji hasilnya, buatlah analisis:

```

Koneksi ke MySQL Server versi: 5.5.5-10.4.17-MariaDB
Anda Telah koneksi ke Database : ('db_jual1',)
Koneksi ke MySQL Telah ditutup
PS C:\Users\Badi> 

```

4. Membuat struktur tabel barang dengan menggunakan Python, menggunakan method `execute()`, dengan rancangan struktur tabel berikut:

Field Name	Datatype	Len	PK?	Not Null?
kode	char	5	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
nama_barang	varchar	50	<input type="checkbox"/>	<input type="checkbox"/>
satuan	varchar	20	<input type="checkbox"/>	<input type="checkbox"/>
harga	int	11	<input type="checkbox"/>	<input type="checkbox"/>
jumlah	int	11	<input type="checkbox"/>	<input type="checkbox"/>

Membuat sub program `koneksi.py` kemudian simpan ke folder kerja yang sama, seperti berikut:

```

#*****
# koneksi.py
# NIM :.....
# NAMA :.....
#*****
import mysql.connector
from mysql.connector import Error

try:
    konek = mysql.connector.connect(host='localhost',
                                   database='db_jual1',

```

```

        user='ana',
        password='123***)

#cek jika terjadi kesalahan
except Error as e:
    print("Tidak bisa koneksi, terjadi kesalahan ", e)

```

catatan: nama **database** dan **user** dan **password** bisa disesuaikan diservernya masing-masing.

sub program/file **koneksi.py** akan digunakan untuk beberapa aplikasi lainnya, cara menggunakan:

```

import koneksi
cursor = koneksi.konek.cursor()

```

Untuk membuat struktur tabel **barang** maka, buatlah program dengan nama **praktik11-1.py** seperti pada kode berikut:

```

#*****
# praktik11-2.py
# NIM :.....
# NAMA :.....
#*****
import koneksi

try:
    #skrip sql membuat tabel barang
    sql = """ CREATE TABLE barang (
        kode char(5) NOT NULL,
        nama_barang varchar(50),
        satuan varchar(20),
        harga int(11),
        jumlah int(11),
        PRIMARY KEY (kode)) """
    cursor = koneksi.konek.cursor()
    result = cursor.execute(sql)
    print("Membuat tabel barang berhasil ")

#cek jika terjadi kesalahan
except koneksi.Error as e:
    print("Terjadi kesalahan ", e)
finally:
    if koneksi.konek.is_connected():
        cursor.close()
        koneksi.konek.close()
    print("Koneksi ke MySQL Telah ditutup")

```

Untuk melihat hasilnya Anda login ke database MySQL Lihat hasilnya menggunakan perintah **describe barang**

- Menambah 1 rekaman tabel barang, menggunakan method `execute()` dan `commit()`, seperti pada kode program berikut:

```

#####
# praktik11-3.py
# NIM :.....
# NAMA :.....
#####
import koneksi

try:
    #skrip sql menambah 1 rekaman tabel barang
    sql = """INSERT INTO barang(kode,nama_barang,satuan,harga,jumlah)
            VALUES('10001','Meja Tulis','Unit',1500000,5)"""
    cursor = koneksi.konek.cursor()
    result = cursor.execute(sql)
    koneksi.konek.commit()
    print("Menambahkan tabel barang sebanyak=",cursor.rowcount,"rekaman")
#cek jika terjadi kesalahan
except koneksi.Error as e:
    print("Terjadi kesalahan ", e)
finally:
    if koneksi.konek.is_connected():
        cursor.close()
        koneksi.konek.close()
    print("Koneksi ke MySQL Telah ditutup")

```

Untuk melihat hasilnya, gunakan perintah sql SELECT \* FROM barang, seperti pada gambar berikut:

```

MariaDB [db_jual1]> select * from barang;
+-----+-----+-----+-----+-----+
| kode | nama_barang | satuan | harga | jumlah |
+-----+-----+-----+-----+-----+
| 10001 | Meja Tulis | Unit | 1500000 | 5 |
+-----+-----+-----+-----+-----+
1 row in set (0.001 sec)

```

6. Menambah rekaman tabel barang menggunakan sub program, dan parameter field, seperti pada kode program berikut:

```

#####
# praktik11-4.py
# NIM :.....
# NAMA :.....
#####
import koneksi

def insertData(vkode, vnama_barang, vsatuan, vharga, vjumlah):
    try:
        #skrip sql menambah 1 rekaman tabel barang
        cursor = koneksi.konek.cursor()
        sql = """INSERT INTO barang(kode,nama_barang,satuan,harga,jumlah)
                VALUES(%s,%s,%s,%s,%s)"""
        record = (vkode, vnama_barang, vsatuan, vharga, vjumlah)
        result = cursor.execute(sql, record)
        koneksi.konek.commit()
        print("Menambahkan tabel barang sebanyak=",cursor.rowcount,"rekaman")
    #cek jika terjadi kesalahan

```

```

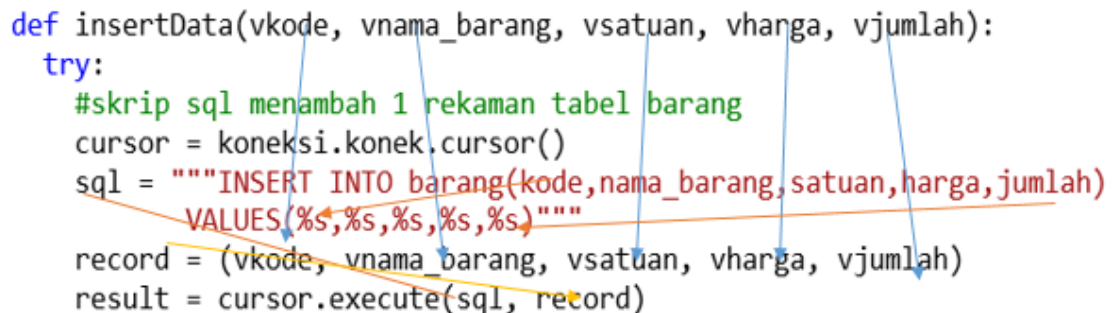
except koneksi.Error as e:
    print("Terjadi kesalahan ", e)
finally:
    if koneksi.konek.is_connected():
        cursor.close()

#memanggil sub program
insertData('10002', 'Meja Komputer', 'Unit', 1500000, 4)
insertData('10003', 'Meja Belajar', 'Unit', 1200000, 4)

```

Lihat Hasilnya menggunakan SELECT \* FROM barang

Penjelasan:



```

def insertData(vkode, vnama_barang, vsatuan, vharga, vjumlah):
    try:
        #skrip sql menambah 1 rekaman tabel barang
        cursor = koneksi.konek.cursor()
        sql = """INSERT INTO barang(kode,nama_barang,satuan,harga,jumlah)
                VALUES(%s,%s,%s,%s,%s)"""
        record = (vkode, vnama_barang, vsatuan, vharga, vjumlah)
        result = cursor.execute(sql, record)

```

The diagram illustrates the mapping of arguments from the function call to the SQL statement. Blue arrows point from the function parameters (vkode, vnama\_barang, vsatuan, vharga, vjumlah) to the corresponding placeholders (%s) in the SQL VALUES clause. An orange arrow points from the record tuple (vkode, vnama\_barang, vsatuan, vharga, vjumlah) to the record parameter in the cursor.execute(sql, record) call.

## 7. Menambahkan rekaman menggunakan list, atau tuple, dan executemany()

```

# *****
# praktik11-5.py
# NIM :.....
# NAMA :.....
# mengubah rekaman
#*****
import koneksi

def insertData(data=[()]):
    try:
        #skrip sql menambah banyak rekaman tabel barang
        cursor = koneksi.konek.cursor()
        sql = """INSERT INTO barang(kode,nama_barang,satuan,harga,jumlah)
                VALUES(%s,%s,%s,%s,%s)"""
        result = cursor.executemany(sql, data)
        koneksi.konek.commit()
        print("Menambahkan tabel barang sebanyak=", cursor.rowcount, "rekaman")
    #cek jika terjadi kesalahan
    except koneksi.Error as e:
        print("Terjadi kesalahan ", e)
    finally:
        if koneksi.konek.is_connected():
            cursor.close()
            koneksi.konek.close()
            print("Koneksi ke MySQL Telah ditutup")

#data dalam list dan tuple
data=[('10004', 'Kursi Lipat', 'Unit', 300000, 4),
      ('10005', 'Kursi Biasa', 'Unit', 100000, 5),
      ('10007', 'Papan Tulis', 'Unit', 500000, 2)]

```

```
#memanggil fungsi/sub program
insertData(data)
```

amati hasilnya

8. Membaca semua rekaman tabel menggunakan sub program dan method `execute()`, `fetchall()`, seperti pada kode program berikut:

```
*****
# praktik11-6.py
# NIM :.....
# NAMA :.....
*****
import koneksi

def bacaData():
    try:
        cursor = koneksi.konek.cursor()
        sql = """SELECT * FROM barang """
        cursor.execute(sql)
        rekaman = cursor.fetchall()
        print("Jumlah rekaman yang dibaca: ", cursor.rowcount)
        #cetak menggunakan loop
        for brs in rekaman:
            print("Kode      : ", brs[0])
            print("Nama Barang : ", brs[1])
            print("Satuan    : ", brs[2])
            print("Harga     : ", brs[3])
            print("Jumlah    : ", brs[4], "\n")

        #cek jika terjadi kesalahan
    except koneksi.Error as e:
        print("Terjadi kesalahan ", e)
    finally:
        if koneksi.konek.is_connected():
            cursor.close()
            koneksi.konek.close()
            print("Koneksi ke MySQL Telah ditutup")

#memanggil fungsi/sub program
bacaData()
```

Setelah di eksekusi apa hasilnya?

9. Dengan menggunakan program di atas untuk memeriksa jika rekaman tidak ada dengan menambahkan perintah `if rekaman == []:`, seperti pada kode program berikut:

```
*****
# praktik11-7.py
# NIM :.....
# NAMA :.....
*****
import koneksi
import os
```

```

def bacaData(kode):
    try:
        cursor = koneksi.konek.cursor()
        sql = """SELECT * FROM barang where kode = %s"""
        cursor.execute(sql,(kode,))
        rekaman = cursor.fetchall()
        #jika rekaman tidak ditemukan
        if rekaman == []:
            print("Data tidak ditemukan")
        else:
            #cetak menggunakan loop
            for brs in rekaman:
                print("Kode      : ", brs[0])
                print("Nama Barang : ", brs[1])
                print("Satuan     : ", brs[2])
                print("Harga      : ", brs[3])
                print("Jumlah     : ", brs[4], "\n")

        #cek jika terjadi kesalahan
        except koneksi.Error as e:
            print("Terjadi kesalahan ", e)
        finally:
            if koneksi.konek.is_connected():
                cursor.close()

    #memanggil fungsi/sub program
    lagi='Y'
    while lagi=='Y' or lagi=='y':
        os.system('cls')
        kode=input("Masukkan Kode Barang [0000] = ")
        bacaData(kode)
        lagi=input("Mencoba Lagi Y/T : ")

```



## LATIHAN/TUGAS

---

1. Buatlah program untuk mengubah kolom harga, menggunakan kunci pencarian kode
2. Buatlah program untuk mengubah kolom harga, dan jumlah menggunakan kunci pencarian kode
3. Buatlah program untuk menghapus rekaman menggunakan kunci pencarian kode



## REFERENSI

---

Python MySQL Update Table, [https://www.w3schools.com/python/python\\_mysql\\_update.asp](https://www.w3schools.com/python/python_mysql_update.asp)

Python MySQL Delete From By, [https://www.w3schools.com/python/python\\_mysql\\_delete.asp](https://www.w3schools.com/python/python_mysql_delete.asp)



<https://yoi.home.blog/2021/01/28/membuat-sistem-crud-sederhana-dengan-mysql-dan-python/>