

UJIAN TENGAH SEMESTER

**PRAKTIK
PEMOGRAMAN PYTHON**



Disusun oleh :
Bimo Adji Kusnadi
V3922010

Dosen :
Yusuf Fadila Rachman S.Kom,M.Kom

**PS D-III TEKNIK INFORMATIKA
SEKOLAH VOKASI
UNIVERSITAS SEBELAS MARET
2023**

HASIL DAN PENJELASAN

1. Membuat database

```
import mysql.connector

dataBase = mysql.connector.connect (
    host = "localhost",
    user = "root",
    passwd = ""
)

cursorObject = dataBase.cursor()
cursorObject.execute("CREATE DATABASE db_V3922010")
```

Penjelasan :

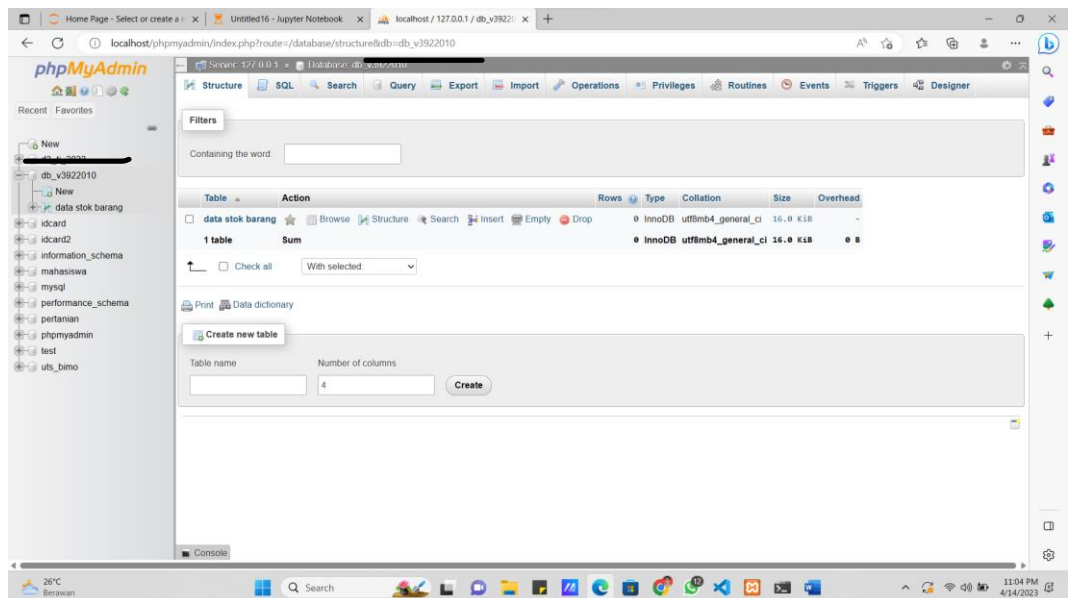
Baris 1 mengimpor modul `mysql.connector` kedalam program python.

Pada baris 2 mendefinisikan parameter koneksi ke database dengan menentukan host, user, dan password.

Baris 3 membuat objek kursor yang dapat digunakan untuk mengeksekusi perintah SQL pada database.

Baris ke 4 adalah perintah SQL untuk membuat database baru dengan nama "db_V3922010". Dieksekusi menggunakan objek kursor dengan memanggil metode `execute()`. Setelah dieksekusi, database baru akan dibuat dan siap digunakan.

Hasil:



2. Membuat tabel DATA STOK BARANG

```
import mysql.connector

dataBase = mysql.connector.connect(
    host = "localhost",
    user = "root",
    passwd = "",
    database = "db_v3922010"
)

cursorObject = dataBase.cursor()

DataBarang = """CREATE TABLE `DATA STOK BARANG` (
    Id_Barang VARCHAR(14) PRIMARY KEY,
    `Nama Barang` VARCHAR(50),
    `Harga Barang` INT,
    `Stok Awal` INT,
    `Barang Masuk` INT,
    `Barang Keluar` INT,
    `Stok Akhir` INT
)"""

cursorObject.execute(DataBarang)
dataBase.close()
```

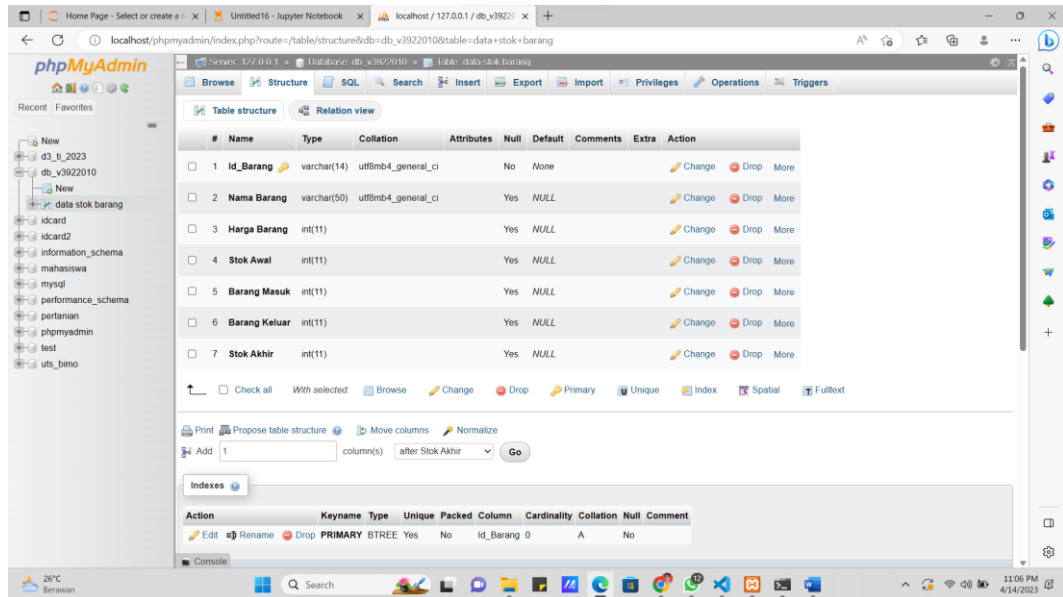
Penjelasan :

Pada baris ke 4 perintah SQL untuk membuat sebuah tabel baru dengan nama `DATA STOK BARANG`. Tabel ini memiliki tujuh kolom, yaitu `Id_Barang`, `Nama Barang`, `Harga Barang`, `Stok Awal`, `Barang Masuk`, `Barang Keluar`, `Stok Akhir`.

Baris ke 5 menjalankan perintah SQL.

Baris ke 6 menutup koneksi ke database dengan menggunakan metode `close()` pada objek `database`. Untuk memastikan bahwa sumber daya database tidak terus terbuka dan membebani sistem.

Hasil:



3. Buat fungsi insert, show, update, hapus, dan search berdasarkan table yang dibuat.

```
import mysql.connector

dataBase = mysql.connector.connect(
    user = 'root',
    host = 'localhost',
    database = 'db_v3922010'
)

cursorObject = dataBase.cursor()

def insert_data( Id_Barang, Nama_Barang, Harga_Barang, Stok_Awal, Barang_Masuk, Barang_Keluar, Stok_Akhir ):
    sql = "INSERT INTO `DATA STOK BARANG` (`Id_Barang`, `Nama_Barang`, `Harga_Barang`, `Stok_Awal`, `Barang_Masuk`, `Barang_Keluar`, `Stok_Akhir`)"
    val = (Id_Barang, Nama_Barang, Harga_Barang, Stok_Awal, Barang_Masuk, Barang_Keluar, Stok_Akhir)

    cursorObject.execute(sql, val)
    dataBase.commit()

    print(" ")
    print("Data berhasil ditambahkan")
```

Penjelasan :

Fungsi ini menggunakan string SQL untuk mengeksekusi perintah INSERT pada tabel "DATA STOK BARANG". String SQL ini memiliki placeholder %s untuk menandakan nilai-nilai yang akan dimasukkan ke dalam tabel, sehingga menghindari potensi masalah keamanan seperti SQL injection.

Setelah nilai-nilai dimasukkan ke dalam placeholder pada string SQL, fungsi akan mengeksekusi perintah SQL tersebut menggunakan objek

kursor dan menyimpan perubahan ke database menggunakan metode `commit()`. Fungsi juga mencetak pesan untuk memberitahukan pengguna bahwa data telah berhasil dimasukkan ke dalam tabel.

```
def show_data():
    query = "SELECT * FROM `DATA STOK BARANG`"

    cursorObject.execute(query)

    myresult = cursorObject.fetchall()

    for x in myresult:
        print(x)

    print(" ")
    print("Data berhasil ditampilkan")
```

Penjelasan:

Fungsi ini menggunakan string SQL `SELECT * FROM DATA STOK BARANG`` untuk mengeksekusi perintah SQL yang mengambil seluruh data dari tabel "DATA STOK BARANG". Setelah itu, fungsi akan menjalankan perintah SQL tersebut menggunakan objek kursor dan mengambil hasilnya menggunakan metode `fetchall()`, lalu menampilkannya ke layar satu per satu menggunakan perulangan `for`. Terakhir, fungsi mencetak pesan untuk memberitahukan pengguna bahwa data berhasil ditampilkan. Dengan menggunakan fungsi `show_data` pengguna bisa melihat data yang ada di dalam tabel DATA STOK BARANG.

```
def update_data(Id_Barang, Nama_Barang, Harga_Barang, Stok_Awal, Barang_Masuk, Barang_Keluar, Stok_Akhir):
    sql = "UPDATE `DATA STOK BARANG` SET `Nama Barang` = %s, `Harga Barang` = %s, `Stok Awal` = %s, `Barang Masuk` = %s, `Barang Keluar` = %s, `Stok Akhir` = %s"
    val = (Nama_Barang, Harga_Barang, Stok_Awal, Barang_Masuk, Barang_Keluar, Stok_Akhir, Id_Barang)

    cursorObject.execute(sql, val)
    database.commit()

    print(" ")
    print("Data berhasil diupdate")
```

Penjelasan:

Ini menggunakan string SQL `UPDATE DATA STOK BARANGSET ... WHEREId Barang = %s` untuk mengeksekusi perintah SQL yang mengupdate data yang ada di dalam tabel "DATA STOK BARANG" sesuai dengan `Id_Barang` yang diinputkan. Setelah itu, fungsi akan menjalankan perintah SQL tersebut menggunakan objek kursor dan mengupdate data dengan nilai-nilai yang diberikan menggunakan parameter `val`. fungsi mencetak pesan untuk memberitahukan pengguna

bahwa data berhasil diupdate, dengan menggunakan update_data pengguna dapat dengan mudah mengupdate data pada tabel DATA STOK BARANG.

```
def delete_data(Id_Barang):  
    sql = "DELETE FROM `DATA STOK BARANG` WHERE `Id_Barang` = %s"  
    val = (Id_Barang,)  
  
    cursorObject.execute(sql, val)  
    DataBase.commit()  
  
    print(" ")  
    print("Data berhasil dihapus")
```

Penjelasan :

Fungsi ini menggunakan string SQL DELETE FROM DATA STOK BARANG WHERE Id Barang = %s untuk mengeksekusi perintah SQL yang menghapus data yang ada di dalam tabel "DATA STOK BARANG" sesuai dengan Id_Barang yang diinputkan. Setelah itu, fungsi akan menjalankan perintah SQL tersebut menggunakan objek kursor dan menghapus data dengan nilai-nilai yang diberikan menggunakan parameter val. Fungsi mencetak pesan untuk memberitahukan pengguna bahwa data berhasil dihapus.

```
def search_data(id_barang):  
    sql = "SELECT * FROM `DATA STOK BARANG` WHERE `Id_Barang` = %s"  
    val = (id_barang,)  
  
    cursorObject.execute(sql, val)  
    myresult = cursorObject.fetchall()  
  
    for x in myresult:  
        print(x)  
  
    print(" ")  
    print("Data berhasil dicari")
```

Penjelasan :

Fungsi search_data adalah untuk mencari data barang berdasarkan Id_Barang. Fungsi ini akan mengeksekusi query SQL untuk mencari data pada tabel DATA STOK BARANG yang memiliki Id_Barang yang sama dengan yang diinputkan. Setelah itu, data yang dicari akan ditampilkan dan diberikan pesan bahwa pencarian data berhasil dilakukan.

```

while True:
    print(" ")
    print("=== APLIKASI DATABASE PYTHON ===")
    print("1. Insert data")
    print("2. Show data")
    print("3. Update data")
    print("4. Hapus data")
    print("5. Cari data")
    print("6. Keluar")
    print("-----")
    menu = input("Pilih menu> ")
    print(" ")

    if menu == "1":
        Id_Barang = input("Masukkan Id_Barang : ")
        Nama_Barang = input("Masukkan Nama Barang : ")
        Harga_Barang = int(input("Masukkan Harga Barang : "))
        Stok_Awal = int(input("Masukkan Stok Awal : "))
        Barang_Masuk = int(input("Masukkan Barang Masuk : "))
        Barang_Keluar = int(input("Masukkan Barang Keluar : "))

        Stok_Akhir = Stok_Awal + Barang_Masuk - Barang_Keluar

        print("Stok Akhir : ", Stok_Akhir)

        insert_data(Id_Barang, Nama_Barang, Harga_Barang, Stok_Awal, Barang_Masuk, Barang_Keluar, Stok_Akhir)

    elif menu == "2":
        show_data()

```

```

    elif menu == "3":
        Id_Barang = input("Masukkan Id_Barang yang akan diupdate : ")
        Nama_Barang = input("Masukkan Nama Barang baru : ")
        Harga_Barang = int(input("Masukkan Harga Barang baru : "))
        Stok_Awal = int(input("Masukkan Stok Awal baru : "))
        Barang_Masuk = int(input("Masukkan Barang Masuk baru : "))
        Barang_Keluar = int(input("Masukkan Barang Keluar baru : "))

        Stok_Akhir = Stok_Awal + Barang_Masuk - Barang_Keluar
        print("Stok Akhir setelah diupdate : ", Stok_Akhir)

        update_data(Id_Barang, Nama_Barang, Harga_Barang, Stok_Awal, Barang_Masuk, Barang_Keluar, Stok_Akhir)

    elif menu == "4":
        Id_Barang = input("Masukkan Id_Barang yang ingin dihapus : ")

        delete_data(Id_Barang)

    elif menu == "5":
        Id_Barang = input("Masukkan Id_Barang yang ingin dicari : ")

        search_data(Id_Barang)

    elif menu == "6":
        print("Terimakasih sudah mampir ")
        break

    else:
        print("Pilihan anda tidak valid, Mohon coba lagi dan pilihlah dengan benar")

```

Penjelasan :

Program ini menyediakan beberapa pilihan menu seperti insert data, show data, update data, delete data, search data, dan exit. Setiap pilihan menu memiliki fungsi-fungsi yang berbeda dan dapat dilakukan dengan memasukkan input yang diperlukan. Program ini juga menggunakan database MySQL untuk menyimpan dan mengelola data. Program akan terus berjalan hingga user memilih pilihan menu exit.









Hasil ketika dijalankan

Menu 1

```
=== APLIKASI DATABASE PYTHON ===
1. Insert data
2. Show data
3. Update data
4. Hapus data
5. Cari data
6. Keluar
-----|-----
Pilih menu> 1

Masukkan Id Barang : 3922010
Masukkan Nama Barang : Buku
Masukkan Harga Barang : 50000
Masukkan Stok Awal : 20
Masukkan Barang Masuk : 50
Masukkan Barang Keluar : 10
Stok Akhir : 60

Data berhasil ditambahkan
```

				Id_Barang	Nama Barang	Harga Barang	Stok Awal	Barang Masuk	Barang Keluar	Stok Akhir
<input type="checkbox"/>				3922010	Buku	50000	20	50	10	60
	<input type="checkbox"/> Check all	With selected:								

Menu 2

```
=== APLIKASI DATABASE PYTHON ===
1. Insert data
2. Show data
3. Update data
4. Hapus data
5. Cari data
6. Keluar
-----
Pilih menu> 2

('3922010', 'Buku', 50000, 20, 50, 10, 60)

Data berhasil ditampilkan
```

Menu 3

```
=== APLIKASI DATABASE PYTHON ===
1. Insert data
2. Show data
3. Update data
4. Hapus data
5. Cari data
6. Keluar
-----
Pilih menu> 3

Masukkan Id_Barang yang akan diupdate : 3922010
Masukkan Nama Barang baru : Pensil
Masukkan Harga Barang baru : 10000
Masukkan Stok Awal baru : 10
Masukkan Barang Masuk baru : 30
Masukkan Barang Keluar baru : 20
Stok Akhir setelah diupdate : 20

Data berhasil diupdate
```

<div><div><div></div><div></div></div><div></div></div>		<div>Id_Barang</div>	<div>Nama Barang</div>	<div>Harga Barang</div>	<div>Stok Awal</div>	<div>Barang Masuk</div>	<div>Barang Keluar</div>	<div>Stok Akhir</div>
<div><div><div></div><div></div></div><div></div></div>	<div><div>Edit</div><div>Copy</div><div>Delete</div></div>	<div>3922010</div>	<div>Pensil</div>	<div>10000</div>	<div>10</div>	<div>30</div>	<div>20</div>	<div>20</div>
<div><div><div></div><div></div></div><div></div></div>	<div><div>Check all</div><div>With selected:</div><div><div>Edit</div><div>Copy</div><div>Delete</div><div>Export</div></div></div>							

Menu 4

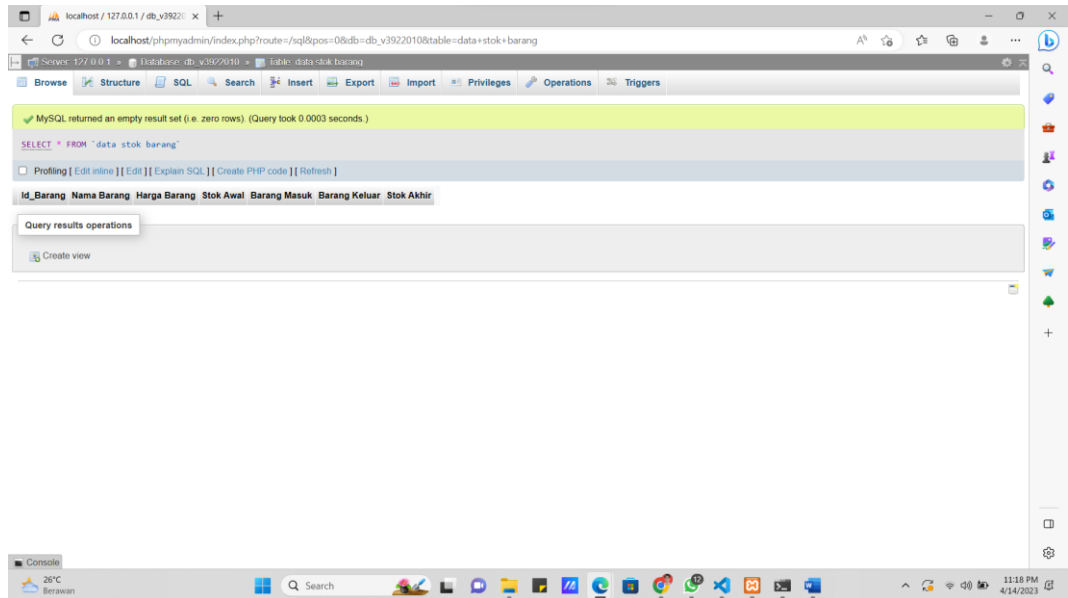
=== APLIKASI DATABASE PYTHON ===

1. Insert data
2. Show data
3. Update data
4. Hapus data
5. Cari data
6. Keluar

Pilih menu> 4

Masukkan Id_Barang yang ingin dihapus : 3922010

Data berhasil dihapus



Menu 5

=== APLIKASI DATABASE PYTHON ===

1. Insert data
2. Show data
3. Update data
4. Hapus data
5. Cari data
6. Keluar

Pilih menu> 5

Masukkan Id_Barang yang ingin dicari : 3922010
('3922010', 'Pensil', 10000, 10, 30, 20, 20)

Data berhasil dicari

Menu 6

=== APLIKASI DATABASE PYTHON ===

1. Insert data
2. Show data
3. Update data
4. Hapus data
5. Cari data
6. Keluar

Pilih menu> 6

Terimakasih sudah mampir