

**LAPORAN PRAKTIKUM
PEMROGRAMAN PYTHON**

UJIAN TENGAH SEMESTER



Disusun oleh :
Fabianus Jan Krisna Wijaya
V3922017

Dosen
Yusuf Fadlila Rachman. S.Kom., M.Kom

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

BAB II

HASIL DAN PEMBAHASAN

```
In [1]: import sqlite3

conn = sqlite3.connect('test.db')

print("Opened database successfully")

Opened database successfully
```

```
In [2]: import mysql.connector

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

#preparing a cursor
cursorObject = dataBase.cursor()

#create database
cursorObject.execute("CREATE DATABASE db_sales_V3922017")
```

1. Membuat sintaks sqlite3 untuk memungkinkan Python berinteraksi dengan database SQLite.
2. Membuat koneksi ke database SQLite dengan 'test.db'. Untuk fungsi 'connect()' dari modul 'sqlite3' digunakan untuk membuka koneksi dan mengembalikan object koneksi SQLite.
3. Selanjutnya membuat database dengan nama db_sales_V3922017

```
In [3]: import mysql.connector

dataBase = mysql.connector.connect(
    host="localhost",
    user="root",
    passwd="",
    database="db_sales_V3922017"
)

# preparing cursor object
cursorObject = dataBase.cursor()

# creating table
studentRecord = """CREATE TABLE data_stok_barang (
    id_barang VARCHAR(15) NOT NULL PRIMARY KEY,
    nama_barang VARCHAR(70) NOT NULL,
    harga_barang INT,
    stokawal INT,
    barang_masuk INT,
    barang_keluar INT,
    stok_akhir INT
)"""

# table created
cursorObject.execute(studentRecord)

# disconnect from server
dataBase.close()
```

4. Selanjutnya membuat table di dalam database yang telah dibuat tadi dengan nama data_stok_barang.

```
In [*]: import mysql.connector

database = mysql.connector.connect(
    host = 'localhost',
    user = 'root',
    passwd = '',
    database = 'db_sales_V3922017'
)

def insert_data(id_barang, nama_barang, harga_barang, stok_awal, barang_masuk,
               barang_keluar, stok_akhir):
    cursorObject = database.cursor()
    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(cursorObject.rowcount, "Data berhasil ditambahkan")

def show_data():
    cursorObject = database.cursor()
    cursorObject.execute("SELECT * FROM data_stok_barang")
    myresult = cursorObject.fetchall()
    for x in myresult:
        print(x)

def update_data(id_barang, nama_barang, harga_barang, stok_awal, barang_masuk, barang_keluar, stok_akhir):
    cursorObject = database.cursor()
    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(cursorObject.rowcount, "Data berhasil diupdate")
```

```
def delete_data(id_barang):
    cursorObject = database.cursor()
    sql = "DELETE FROM data_stok_barang WHERE id_barang = %s"
    val = (id_barang,)
    cursorObject.execute(sql, val)
    database.commit()
    print(cursorObject.rowcount, "Data berhasil dihapus")

def search_data(keyword):
    cursorObject = database.cursor()
    sql = "SELECT * FROM data_stok_barang WHERE nama_barang LIKE %s"
    val = ("% " + keyword + "%",)
    cursorObject.execute(sql, val)
    myresult = cursorObject.fetchall()
    for x in myresult:
        print(x)

print("=== Aplikasi Database Python ===")
print("1. Insert Data")
print("2. Tampilkan Data")
print("3. Update Data")
print("4. Hapus Data")
print("5. Cari Data")
print("0. Keluar")
print("-----")

menu = input("Pilih Menu: ")
```

```
while menu != "0":
    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: "))
        barang_masuk = int(input("Masukkan Barang Masuk: "))
        barang_keluar = int(input("Masukkan Barang Keluar: "))
        stok_akhir = stok_awal + barang_masuk - barang_keluar

        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 Barang 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

        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 akan dihapus: ")
        delete_data(id_barang)

    elif menu == "5":
        keyword = input("Masukkan Kata Kunci: ")
        search_data(keyword)

    else:
        print("Menu tidak tersedia")

    print("-----")
    menu = input("Pilih Menu: ")

print("Program selesai")
```

5. Membuat script dimana program tersebut harus memenuhi kriteria CRUD.
6. `insert_data()`: fungsi ini digunakan untuk memasukkan data ke dalam database. Data yang dimasukkan antara lain `id_barang`, `nama_barang`, `harga_barang`, `stok_awal`, `barang_masuk`, `barang_keluar`, dan `stok_akhir`.
7. `show_data()`: fungsi ini digunakan untuk menampilkan semua data yang ada di dalam database.
8. `update_data()`: fungsi ini digunakan untuk mengupdate data yang sudah ada di dalam database.
9. `delete_data()`: fungsi ini digunakan untuk menghapus data yang sudah ada di dalam database.
10. `search_data()`: fungsi ini digunakan untuk mencari data berdasarkan kata kunci.

```

=== Aplikasi Database Python ===
1. Insert Data
2. Tampilkan Data
3. Update Data
4. Hapus Data
5. Cari Data
0. Keluar
-----
Pilih Menu: 1
Masukkan ID Barang: 1000
Masukkan Nama Barang: Laptop
Masukkan Harga Barang: 5000000
Masukkan Stok Awal Barang: 5
Masukkan Barang Masuk: 4
Masukkan Barang Keluar: 3
1 Data berhasil ditambahkan
-----
Pilih Menu: 2
('1000', 'Laptop', 5000000, 5, 4, 3, 6)
-----
Pilih Menu: 3
Masukkan ID Barang yang akan diupdate: 1000
Masukkan Nama Barang Baru: Handphone
Masukkan Harga Barang Baru: 2000000
Masukkan Stok Awal Barang Baru: 4
Masukkan Barang Masuk Baru: 3
Masukkan Barang Keluar Baru: 2
1 Data berhasil diupdate
-----

```

11. Berikut adalah penjelasan saat program berjalan
12. Ketika kita memilih nomor 1 maka akan memilih insert data. Kita dapat memasukkan ID barang sebagai primary key, nama barang, harga barang, stok awal, barang masuk, barang keluar.
13. Ketika memilih nomor 2 maka akan memilih tampilkan data. Untuk tampilkan data kita dapat melihat data barang yang telah di inputkan tadi tanpa membuka localhost MySQL.

✓ Menampilkan baris 0 - 0 (total 1, Pencarian dilakukan dalam 0,0002 detik.)

`SELECT * FROM `data_stok_barang``

☐ Profil [\[Edit dikotak \]](#) [\[Ubah \]](#) [\[Jelaskan SQL \]](#) [\[Buat kode PHP \]](#) [\[Segarkan \]](#)

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris:

Extra options

	<code>id_barang</code>	<code>nama_barang</code>	<code>harga_barang</code>	<code>stok_awal</code>	<code>barang_masuk</code>	<code>barang_keluar</code>	<code>stok_akhir</code>
<input type="checkbox"/> Ubah <input type="checkbox"/> Salin <input type="checkbox"/> Hapus	1000	Handphone	2000000	4	3	2	5

☐ Pilih Semua
 Dengan pilihan:
 ☐ Ubah
 ☐ Salin
 ☐ Hapus
 ☐ Ekspor

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris:

14. Jika ingin melihat data barang database dapat membuka pada MySQL localhost.
15. Ketika memilih nomor 3 maka akan memilih update data barang dalam database tanpa update data melalui localhost.



SELECT * FROM `data_stok_barang`

☐ Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini

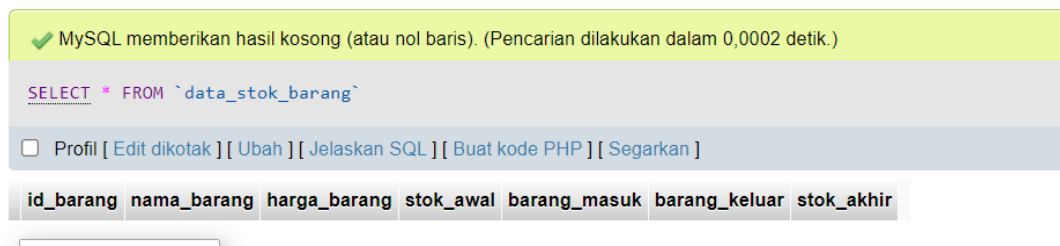
Extra options

				id_barang	nama_barang	harga_barang	stok_awal	barang_masuk	barang_keluar	stok_akhir
<input type="checkbox"/>	Ubah	Salin	Hapus	1000	laptop	5000000	5	6	4	7

↑ ☐ Pilih Semua Dengan pilihan: ☐ Ubah ☐ Salin ☐ Hapus ☐ Ekspor

☐ Tampilkan semua | Jumlah baris: 25 | Saring baris: Cari di tabel ini

16. Berikut adalah perubahan data yang semulanya nama barang handphone diubah menjadi laptop.
17. Dapat juga melihat perubahan data tersebut tanpa membuka localhost dengan cara memilih nomor 2 untuk cek apakah data tersebut sudah terganti atau belum.



✓ MySQL memberikan hasil kosong (atau nol baris). (Pencarian dilakukan dalam 0,0002 detik.)

SELECT * FROM `data_stok_barang`

☐ Profil [Edit dikotak] [Ubah] [Jelaskan SQL] [Buat kode PHP] [Segarkan]

id_barang	nama_barang	harga_barang	stok_awal	barang_masuk	barang_keluar	stok_akhir
-----------	-------------	--------------	-----------	--------------	---------------	------------

18. Ketika memilih nomor 4 maka akan memilih hapus data. Sebagai contohnya data yang dihapus adalah data dengan id_barang 1000 yaitu laptop.
19. Ketika memilih nomor 4 maka akan memilih cari data yang telah ada. Dapat mencari data dengan kata kunci Nama barang atau ID barang
20. Terakhir saat memilih nomor 6 maka program itu akan berhenti.

Pilih Menu: 5
Masukkan Kata Kunci: laptop
('12', 'laptop', 5000000, 5, 6, 4, 7)

Pilih Menu: