


PROJEK PERTEMUAN 12

1. Buatlah project baru dengan nama **PROJEK-PERTEMUAN12** di visual studio code Anda.
2. Anda diminta untuk mengimplementasikan proses CRUD koneksi database SQLite pada pertemuan sebelumnya.
3. Buat database baru dengan nama **database_fauna.db**.
4. Anda dapat menyelesaikan project ini dengan membuat file satu persatu seperti saat praktikum.
5. Ketentuan **Tabel FAUNA**:

ID FAUNA	NAMA FAUNA	JENIS	ASAL	JUMLAH SAAT INI	TAHUN TERAKHIR DITEMUKAN
1	Harimau Jawa	Mamalia	Jawa	40	2019
2	Kuskus Beruang	Mamalia	Sulawesi	30	2021
3	Beruang Madu	Mamalia	Sumatera	1000	2020
4	Pesut Mahakam	Mamalia	Kalimantan	100	2021
5	Burung Maleo	Burung	Sulawesi	7000	2023
6	Macan Dahan	Mamalia	Sumatera	400	2020
7	Kancil	Mamalia	Jawa	60	2022
8	Gajah Kalimantan	Mamalia	Kalimantan	1500	2021
9	Elang Jawa	Burung	Jawa	200	2021
10	Katak Borneo	Amfibi	Kalimantan	2000	2023

6. Ketentuan field tabel:
 - **id_fauna** : primary key, INTEGER, Auto Increment.
 - **nama_fauna** : VARCHAR(50)
 - **jenis**: VARCHAR(50)
 - **asal**: VARCHAR(50)
 - **jml_skrng**: INTEGER(10)
 - **thn_ditemukan**: INTEGER(10)
7. Ketentuan Program Query:
 - 1) **CREATE** Database dan Tabel
 - Buat file **1-create-fauna.py**
 - 2) **INSERT INTO** (Menambahkan data kedalam table)
 - Buat file **2-insert-fauna.py**
 - 3) **SELECT ALL** (Tampilkan semua data tabel)
 - Buat file **3-select-all.py**
 - **Tampilkan hasilnya.**
 - 4) **SELECT WHERE** (Tampilkan data berdasarkan parameter tertentu)
 - Buat file **4-select-where-jenis.py** dan **4-select-where-jumlah.py**
 - Tampilkan berdasarkan **jenis = mamalia** saja.
 - Tampilkan berdasarkan fauna dengan **jml_skrng** kurang dari sama dengan 1000 ekor saja.
 - **Tampilkan hasilnya.**
 - 5) **SELECT WHERE AND** (Tampilkan data berdasarkan operator AND)
 - Buat file **5-select-where-and.py**
 - Tampilkan berdasarkan **jenis (Mamalia)** dan **asal (Sulawesi)**
 - **Tampilkan hasilnya.**
 - 6) **SELECT WHERE OR**(Tampilkan data berdasarkan operator OR)
 - Buat file **6-select-where-or.py**

- Tampilkan berdasarkan **asal(Sumatera)** dan **jml_skrng** lebih dari 500 ekor.
 - **Tampilkan hasilnya.**
- 7) **SELECT SUM** (Menjumlahkan isian field tertentu)
- Buat file **7-select-sum.py**
 - Jumlahkan total populasi hewan langka saat ini (**Total Populasi**) dari **jml_skrng**.
 - **Tampilkan hasilnya.**
- 8) **SELECT ORDER BY** (Mengurutkan sebuah data)
- Buat file **8-select-order-by1.py, 8-select-order-by2.py, 8-select-order-by3.py**
 - Urutkan **nama_fauna** berdasarkan dari awal alphabetic (1).
 - Urutkan **jml_skrng** fauna berdasarkan dari yang terbanyak ke paling sedikit (2).
 - Urutkan **thn_ditemukan** fauna berdasarkan dari tahun yang terlama ke terbaru (3).
 - **Tampilkan hasilnya.**
- 9) **SELECT LIKE** (Filter karakter data)
- Buat file **9-select-like.py**
 - Cari nama fauna yang diawali dengan karakter “B”
 - Tampilkan.
- 10) **UPDATE SET** (Memperbarui data)
- Buat file **10-select-update1.py, 10-select-update2.py**
 - Update **jml_skrng** dari fauna ‘Katak Borneo’ menjadi 650 (1).
 - Update **asal** dari fauna ‘Pesut Mahakam’ menjadi ‘Kalimantan Timur’ (2).
 - **Tampilkan hasilnya.**
- 11) **DELETE FROM** (Menghapus Data)
- File **11-delete-fauna.py**
 - Hapus isian field yang memiliki **asal = Kalimantan**.
 - Buat file **delete-from.py**
 - Tampilkan **sebelum** dihapus.
 - Tampilkan **setelah** dihapus.
8. **Ketentuan lainnya:**
- Projek dikumpulkan pada pertemuan depan.
 - Projek tidak dikumpulkan di elearning, tetapi dikumpulkan pada platform github.
 - Hasil screenshot **diambil dari terminal VS Code** bukan dari DBBrowser SQLite.

	Nama	
	NPM	
	Mata Kuliah	Algoritma Pemrograman Praktik V
	Projek	Projek Pertemuan 12

Copy Paste Codingan:

1. Soal 1 (CREATE)

```

1  import sqlite3
2
3  koneksi = sqlite3.connect('database_fauna.db')
4
5  koneksi.execute('''
6      CREATE TABLE IF NOT EXISTS FAUNA (
7          id_fauna INTEGER PRIMARY KEY AUTOINCREMENT,
8          nama_fauna VARCHAR(50),
9          jenis VARCHAR(50),
10         asal VARCHAR(50),
11         jml_skrng INTEGER,
12         thn_ditemukan INTEGER
13     )
14 ''')
15
16 koneksi.close()

```

2. Soal 2 (INSERT INTO)

DB Browser for SQLite - C:\database_fauna.db\database_fauna.db

File Edit View Tools Help

Database Structure Browse Data Edit Pragmas Execute SQL

Table: FAUNA

id_fauna	nama_fauna	jenis	asal	jml_skrng	thn_ditemukan
Filter	Filter	Filter	Filter	Filter	Filter

3. Soal 3 (SELECT ALL)

Algoritma Pemrograman Praktik V– Jumat

Projek Pertemuan 12

```
3-select-all.py > ...
1 import sqlite3
2 koneksi = sqlite3.connect('database_fauna.db')
3
4 kursor = koneksi.cursor()
5 #Mengambil semua data dalam tabel dan tampilkan
6 kursor.execute("SELECT * FROM FAUNA")
7 #Tampilkan dalam bentuk baris
8 baris_tabel = kursor.fetchall()
9
10 #Membuat format tabel dengan method format()
11 print("Tabel Fauna")
12 print("="*126)
13 print("{:<10} {:<20} {:<20} {:<20} {:<20} {:<20}".format("ID Fauna", "Nama Fauna", "Jenis", "Asal", "Jumlah Saat Ini", "Tahun Terakhir Ditemukan"))
14 print("-"*126)
15 #Tampilkan data sesuai format tabel dg perulangan
16 for baris in baris_tabel:
17     print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
18     print("-"*100)
19 koneksi.close()
```

4. Soal 4 (SELECT WHERE)

- Select Where Jenis

```
5-select-where-and.py > ...
1 import sqlite3
2 koneksi = sqlite3.connect('database_fauna.db')
3
4 kursor = koneksi.cursor()
5 #Mengambil semua data dalam tabel dan tampilkan
6 kursor.execute("SELECT * FROM FAUNA WHERE jenis = 'Mamalia' AND asal = 'Sulawesi' ")
7 #Tampilkan dalam bentuk baris
8 baris_tabel = kursor.fetchall()
9
10 #Membuat format tabel dengan method format()
11 print("Data Fauna")
12 print("="*126)
13 print("{:<10} {:<20} {:<20} {:<20} {:<20} {:<20}".format("ID Fauna", "Nama Fauna", "Jenis", "Asal", "Jumlah Saat Ini", "Tahun Terakhir Ditemukan"))
14 print("-"*126)
15 #Tampilkan data sesuai format tabel dg perulangan
16 for baris in baris_tabel:
17     print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
18
19 koneksi.close()
```

- Select Where Jumlah

```
4-select-where-jumlah.py > ...
1 import sqlite3
2
3 koneksi = sqlite3.connect('database_fauna.db')
4 kursor = koneksi.cursor()
5 kursor.execute("SELECT * FROM FAUNA WHERE jml_skrng <= '1000'")
6 baris_tabel = kursor.fetchall()
7
8 #Membuat format tabel dengan method format()
9 print("Data Fauna")
10 print("="*126)
11 print("{:<10} {:<20} {:<20} {:<20} {:<20} {:<20}".format("ID Fauna", "Nama Fauna", "Jenis", "Asal", "Jumlah Saat Ini", "Tahun Terakhir Ditemukan"))
12 print("-"*126)
13 #Tampilkan data sesuai format tabel dg perulangan
14 for baris in baris_tabel:
15     print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
16
17 koneksi.close()
```

5. Soal 5 (SELECT WHERE AND)

Algoritma Pemrograman Praktik V– Jumat

Projek Pertemuan 12

5-select-where-and.py > ...

```
1 import sqlite3
2 koneksi = sqlite3.connect('database_fauna.db')
3
4 kursor = koneksi.cursor()
5 #Mengambil semua data dalam tabel dan tampilan
6 kursor.execute("SELECT * FROM FAUNA WHERE jenis = 'Mamalia' AND asal = 'Sulawesi' ")
7 #Tampilkan dalam bentuk baris
8 baris_tabel = kursor.fetchall()
9
10 #Membuat format tabel dengan method format()
11 print("Data Fauna")
12 print("="*126)
13 print("{:<10} {:<20} {:<20} {:<20} {:<20} {:<20}".format("ID Fauna", "Nama Fauna", "Jenis", "Asal", "Jumlah Saat Ini", "Tahun Terakhir Ditemukan"))
14 print("-"*126)
15 #Tampilkan data sesuai format tabel dg perulangan
16 for baris in baris_tabel:
17     print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
18
19 koneksi.close()
```

6. Soal 6 (SELECT WHERE OR)

6-select-where-or.py > ...

```
1 import sqlite3
2 koneksi = sqlite3.connect('database_fauna.db')
3
4 kursor = koneksi.cursor()
5 #Mengambil semua data dalam tabel dan tampilan
6 kursor.execute("SELECT * FROM FAUNA WHERE asal = 'Sumatera' OR jml_skrng = '500' ")
7 #Tampilkan dalam bentuk baris
8 baris_tabel = kursor.fetchall()
9
10 #Membuat format tabel dengan method format()
11 print("Data Fauna")
12 print("="*126)
13 print("{:<10} {:<20} {:<20} {:<20} {:<20} {:<20}".format("ID Fauna", "Nama Fauna", "Jenis", "Asal", "Jumlah Saat Ini", "Tahun Terakhir Ditemukan"))
14 print("-"*126)
15 #Tampilkan data sesuai format tabel dg perulangan
16 for baris in baris_tabel:
17     print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
18
19 koneksi.close()
```

7. Soal 7 (SELECT SUM)

7-select-sum.py > ...

```
1 import sqlite3
2 koneksi = sqlite3.connect('database_fauna.db')
3 kursor = koneksi.cursor()
4 #INSERT DATA KE TABEL
5
6 kursor.execute("SELECT SUM(jml_skrng) FROM FAUNA")#untuk mengganti jadi sum bagian avg yang diganti
7 total_populasi = kursor.fetchone()[0]#ambil data gaji jaidkan baris baru
8
9 print(f"Total populasi hewan langka saat ini: {total_populasi}")#SUM UNTUK MENGHITUNG TOTAL
10
11 koneksi.close()
```

8. Soal 8 (SELECT ORDER BY)

- orderby1

8_select_order_by1.py > ...

```
1 import sqlite3
2
3 # Membuat koneksi ke database atau membuat database baru jika belum ada
4 koneksi = sqlite3.connect('database_fauna.db')
5 kursor = koneksi.cursor()
6
7 # Menjalankan query SELECT dengan ORDER BY
8 kursor.execute("SELECT * FROM FAUNA ORDER BY nama_fauna ASC") #ASC|DESC
9 baris_table = kursor.fetchall()
10
11 print("Data Fauna:")
12 print("=====")
13 print("{:<5} {:<20} {:<20} {:<20} {:<10}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir Ditemukan"))
14 print("-----")
15 for baris in baris_table:
16     print("{:<5} {:<20} {:<20} {:<20} {:<10}".format(baris[0], baris[1], baris[2], baris[3], baris[4]))
17
18 koneksi.close()
```

Algoritma Pemrograman Praktik V– Jumat
Projek Pertemuan 12

- orderby2

```
8_select_order_by2.py > ...
1 import sqlite3
2
3 # Membuat koneksi ke database atau membuat database baru jika belum ada
4 koneksi = sqlite3.connect('database_fauna.db')
5 kursor = koneksi.cursor()
6
7 # Menjalankan query SELECT dengan ORDER BY
8 kursor.execute("SELECT * FROM FAUNA ORDER BY jml_skrng DESC") #ASC|DESC
9 baris_table = kursor.fetchall()
10
11 print("Data Fauna:")
12 print("=====")
13 print("{:<5} {:<20} {:<20} {:<20} {:<10}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir Ditemukan"))
14 print("-----")
15 for baris in baris_table:
16     print("{:<5} {:<20} {:<20} {:<20} {:<10}".format(baris[0], baris[1], baris[2], baris[3], baris[4]))
17
18 koneksi.close()
```

- orderby3

```
8_select_order_by3.py > ...
1 import sqlite3
2
3 # Membuat koneksi ke database atau membuat database baru jika belum ada
4 koneksi = sqlite3.connect('database_fauna.db')
5 kursor = koneksi.cursor()
6
7 # Menjalankan query SELECT dengan ORDER BY
8 kursor.execute("SELECT * FROM FAUNA ORDER BY thn_ditemukan ASC") #ASC|DESC
9 baris_table = kursor.fetchall()
10
11 print("Data Fauna:")
12 print("=====")
13 print("{:<5} {:<18} {:<10} {:<10} {:<15} {:>20}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir"))
14 print("-----")
15 for baris in baris_table:
16     print("{:<5} {:<18} {:<10} {:<10} {:<15} {:>20}".format(baris[0], baris[1], baris[2], baris[3], baris[4],baris[5]))
17
18 koneksi.close()
```

9. Soal 9 (SELECT LIKE)

```
9_select_like.py > ...
1 import sqlite3
2
3 # Membuat koneksi ke database atau membuat database baru jika belum ada
4 koneksi = sqlite3.connect('database_fauna.db')
5 kursor = koneksi.cursor()
6
7 # Menjalankan query SELECT dengan LIKE
8 nama = 'B%' # Mencari nama
9 kursor.execute(f"SELECT * FROM FAUNA WHERE nama_fauna LIKE ?", (nama,))
10 baris_table = kursor.fetchall()
11
12 print("Data Fauna :")
13 print("=====")
14 print("{:<5} {:<18} {:<10} {:<10} {:<15} {:>20}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir"))
15 print("-----")
16 for baris in baris_table:
17     print("{:<5} {:<18} {:<10} {:<10} {:<15} {:>20}".format(baris[0], baris[1], baris[2], baris[3], baris[4],baris[5]))
18
19 koneksi.close()
```

10. Soal 10 (UPDATE SET)

-update1

10_select_update1.py > ...

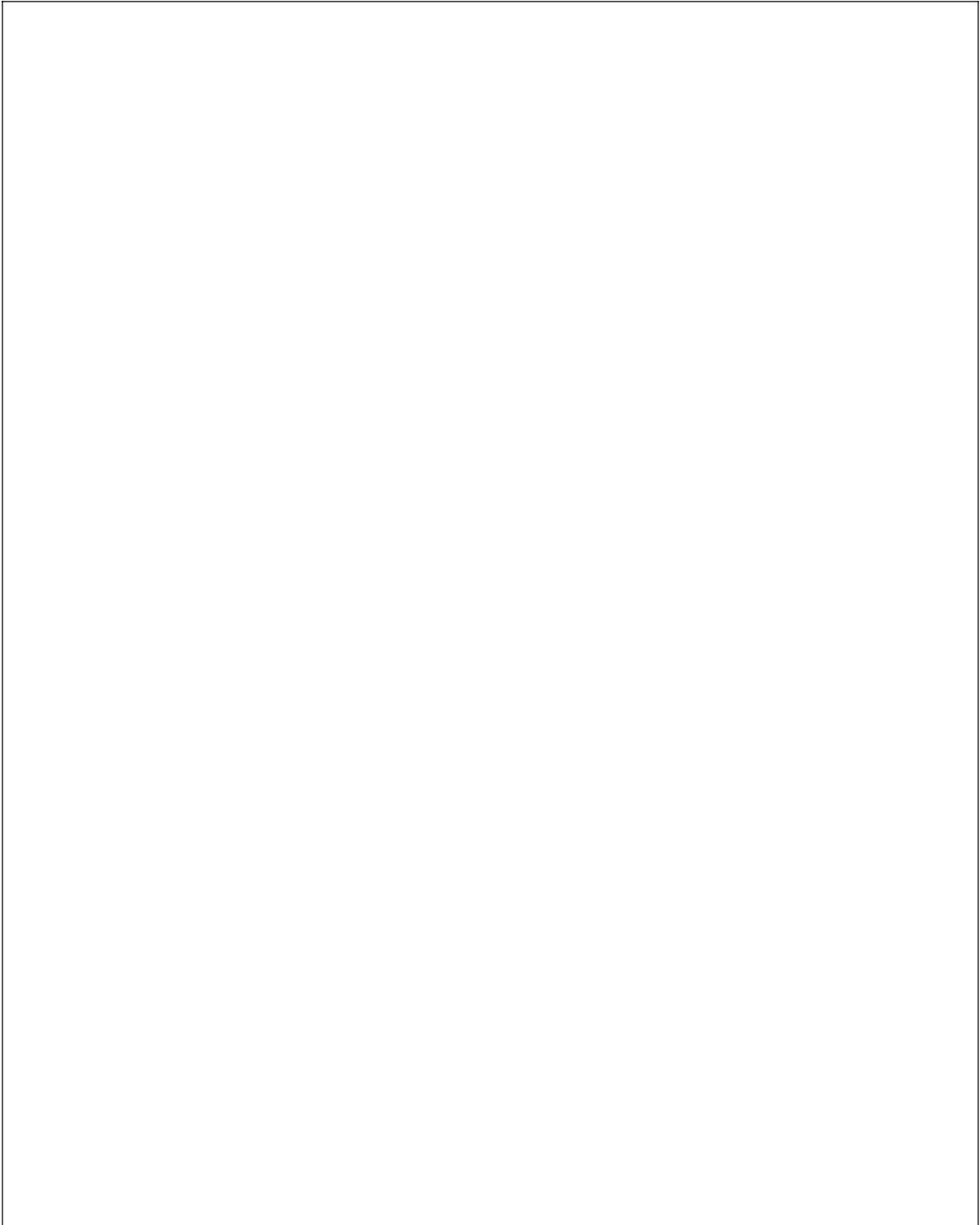
```
1  # UPDATE table_name
2  # SET column1 = value1, column2 = value2, ...
3  # WHERE condition;
4  import sqlite3
5
6  # Membuat koneksi ke database atau membuat database baru jika belum ada
7  conn = sqlite3.connect('database_fauna.db')
8  cursor = conn.cursor()
9
10 # Data yang ingin diubah
11 id_fauna = 10
12 jml_baru = 650
13
14 # Menjalankan query UPDATE
15 cursor.execute(f"UPDATE FAUNA SET jml_skrng = {jml_baru} WHERE id_fauna = {id_fauna}")
16 conn.commit()
17
18 # Menampilkan pesan setelah update berhasil
19 if cursor.rowcount > 0:
20     print(f>Data fauna dengan ID {id_fauna} berhasil diupdate.")
21 else:
22     print(f>Tidak ada data fauna dengan ID {id_fauna}.")
23
24 # Menutup koneksi
25 conn.close()
```

-update2

10_select_update2.py > ...

```
1  # UPDATE table_name
2  # SET column1 = value1, column2 = value2, ...
3  # WHERE condition;
4  import sqlite3
5
6  # Membuat koneksi ke database atau membuat database baru jika belum ada
7  conn = sqlite3.connect('database_fauna.db')
8  cursor = conn.cursor()
9
10 # Data yang ingin diubah
11 id_fauna = 4
12 asal_baru = "Kalimantan Timur"
13
14 # Menjalankan query UPDATE
15 cursor.execute(f"UPDATE FAUNA SET asal = {asal_baru} WHERE id_fauna = {id_fauna}")
16 conn.commit()
17
18 # Menampilkan pesan setelah update berhasil
19 if cursor.rowcount > 0:
20     print(f>Data fauna dengan ID {id_fauna} berhasil diupdate.")
21 else:
22     print(f>Tidak ada data fauna dengan ID {id_fauna}.")
23
24 # Menutup koneksi
25 conn.close()
```

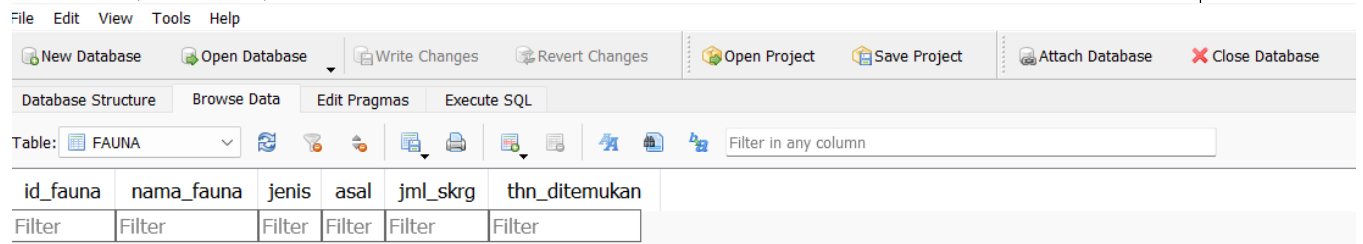
```
11_delete_fauna.py > ...  
1  import sqlite3  
2  
3  # Membuat koneksi ke database atau membuat database baru jika belum ada  
4  conn = sqlite3.connect('database_fauna.db')  
5  cursor = conn.cursor()  
6  
7  # Menjalankan query DELETE  
8  asal = "Kalimantan" # Ganti dengan kriteria yang sesuai untuk data yang ingin dihapus  
9  cursor.execute("DELETE FROM FAUNA WHERE asal = ?", (asal,))  
10 conn.commit()  
11  
12 # Menampilkan pesan setelah penghapusan berhasil  
13 if cursor.rowcount > 0:  
14 |     print(f>Data fauna dengan Asal {asal} berhasil dihapus.")  
15 else:  
16 |     print(f"Tidak ada data fauna dengan Asal {asal}.")  
17  
18 # Menutup koneksi  
19 conn.close()
```

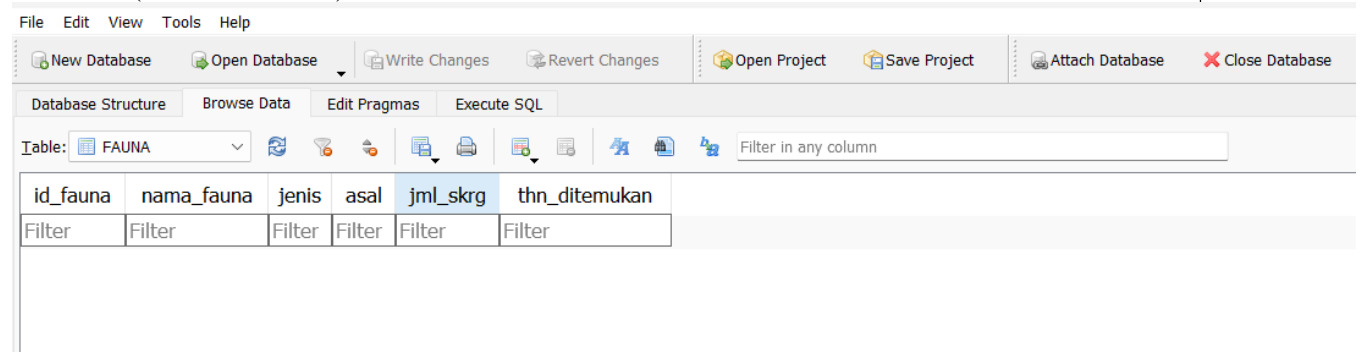
Algoritma Pemrograman Praktik V– Jumat Projek Pertemuan 12

Screenshot Hasil Program:

1. Soal 1 (CREATE)



2. Soal 2 (INSERT INTO)



3. Soal 3 (SELECT ALL)

ID Fauna	Nama Fauna	Jenis	Asal	Jumlah Saat Ini	Tahun Terakhir Ditemukan
1	Harimau Jawa	Mamalia	Jawa	40	2019
2	Kuskus Beruang	Mamalia	Sulawesi	30	2021
3	Beruang Madu	Mamalia	Sumatera	1000	2020
5	Burung Maleo	Burung	Sulawesi	7000	2023
6	Macan Dahan	Mamalia	Sumatera	400	2020
7	Kancil	Mamalia	Jawa	60	2022
9	Elang Jawa	Burung	Jawa	200	2021

4. Soal 4 (SELECT WHERE)

- Select Where Jenis

```
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/5-select-where-and.py
Data Fauna
=====
ID Fauna  Nama Fauna  Jenis      Asal      Jumlah Saat Ini  Tahun Terakhir Ditemukan
-----
PS C:\database_fauna.db>
```

- Select Where Jumlah

ID Fauna	Nama Fauna	Jenis	Asal	Jumlah Saat Ini	Tahun Terakhir Ditemukan
1	Harimau Jawa	Mamalia	Jawa	40	2019
2	Kuskus Beruang	Mamalia	Sulawesi	30	2021
3	Beruang Madu	Mamalia	Sumatera	1000	2020
6	Macan Dahan	Mamalia	Sumatera	400	2020
7	Kancil	Mamalia	Jawa	60	2022
9	Elang Jawa	Burung	Jawa	200	2021

PS C:\database_fauna.db>

5. Soal 5 (SELECT WHERE AND)

ID Fauna	Nama Fauna	Jenis	Asal	Jumlah Saat Ini	Tahun Terakhir Ditemukan
2	Kuskus Beruang	Mamalia	Sulawesi	30	2021

PS C:\database_fauna.db>

6. Soal 6 (SELECT WHERE OR)

Algoritma Pemrograman Praktik V– Jumat
Projek Pertemuan 12

ID Fauna	Nama Fauna	Jenis	Asal	Jumlah Saat Ini	Tahun Terakhir Ditemukan
3	Beruang Madu	Mamalia	Sumatera	1000	2020
6	Macan Dahan	Mamalia	Sumatera	400	2020

PS C:\database_fauna.db>

7. Soal 7 (SELECT SUM)

```
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/7-select-sum.py
Total populasi hewan langka saat ini: 8730
PS C:\database_fauna.db>
```

8. Soal 8 (SELECT ORDER BY)

- orderby1

ID	Nama	Jenis	Asal	Jumlah Sekarang
3	Beruang Madu	Mamalia	Sumatera	1000
5	Burung Maleo	Burung	Sulawesi	7000
9	Elang Jawa	Burung	Jawa	200
8	Gajah Kalimantan	Mamalia	Kalimantan	1500
1	Harimau Jawa	Mamalia	Jawa	40
7	Kancil	Mamalia	Jawa	60
10	Katak Borneo	Amfibi	Kalimantan	2000
2	Kuskus Beruang	Mamalia	Sulawesi	30
6	Macan Dahan	Mamalia	Sumatera	400
4	Pesut Mahakam	Mamalia	Kalimantan	100

PS C:\database_fauna.db>

- orderby2

```
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/8_select_order_by2.py
Traceback (most recent call last):
  File "c:\database_fauna.db\8_select_order_by2.py", line 8, in <module>
    cursor.execute("SELECT * FROM FAUNA ORDER BY jml_skrng DESC") #ASC|DESC
    ~~~~~^~~~~~
sqlite3.OperationalError: no such column: jml_skrng
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/8_select_order_by2.py
Traceback (most recent call last):
  File "c:\database_fauna.db\8_select_order_by2.py", line 8, in <module>
    cursor.execute("SELECT * FROM FAUNA ORDER BY jml_skrng DESC") #ASC|DESC
    ~~~~~^~~~~~
sqlite3.OperationalError: no such column: jml_skrng
PS C:\database_fauna.db>
```

- orderby3

ID	Nama	Jenis	Asal	Jumlah Sekarang	Tahun Terakhir
1	Harimau Jawa	Mamalia	Jawa	40	2019
3	Beruang Madu	Mamalia	Sumatera	1000	2020
6	Macan Dahan	Mamalia	Sumatera	400	2020
2	Kuskus Beruang	Mamalia	Sulawesi	30	2021
4	Pesut Mahakam	Mamalia	Kalimantan	100	2021
8	Gajah Kalimantan	Mamalia	Kalimantan	1500	2021
9	Elang Jawa	Burung	Jawa	200	2021
7	Kancil	Mamalia	Jawa	60	2022
5	Burung Maleo	Burung	Sulawesi	7000	2023
10	Katak Borneo	Amfibi	Kalimantan	2000	2023

PS C:\database_fauna.db>

9. Soal 9 (SELECT LIKE)

ID	Nama	Jenis	Asal	Jumlah Sekarang	Tahun Terakhir
3	Beruang Madu	Mamalia	Sumatera	1000	2020
5	Burung Maleo	Burung	Sulawesi	7000	2023

PS C:\database_fauna.db>

10. Soal 10 (UPDATE SET)

Update1

Algoritma Pemrograman Praktik V– Jumat

Projek Pertemuan 12

```
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/10_select_update1.py
Traceback (most recent call last):
  File "c:\database_fauna.db\10_select_update1.py", line 15, in <module>
    cursor.execute(f"UPDATE FAUNA SET jml_skrng = {jml_baru} WHERE id_fauna = {id_fauna}")
sqlite3.OperationalError: no such column: jml_skrng
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/10_select_update1.py
Traceback (most recent call last):
  File "c:\database_fauna.db\10_select_update1.py", line 15, in <module>
    cursor.execute(f"UPDATE FAUNA SET jml_skrng = {jml_baru} WHERE id_fauna = {id_fauna}")
sqlite3.OperationalError: no such column: jml_skrng
PS C:\database_fauna.db>
```

Update2

```
sqlite3.OperationalError: no such column: jml_skrng
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/10_select_update2.py
Traceback (most recent call last):
  File "c:\database_fauna.db\10_select_update2.py", line 15, in <module>
    cursor.execute(f"UPDATE FAUNA SET asal = {asal_baru} WHERE id_fauna = {id_fauna}")
sqlite3.OperationalError: near "Timur": syntax error
PS C:\database_fauna.db>
```

11. Soal 11 (DELETE FROM)

```
sqlite3.OperationalError: near "Timur": syntax error
PS C:\database_fauna.db> & C:/Users/Lenovo/AppData/Local/Programs/Python/Python312/python.exe c:/database_fauna.db/11_delete_fauna.py
Data fauna dengan Asal Kalimantan berhasil dihapus.
PS C:\database_fauna.db>
```