## **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)
    - o Buat file 3-select-all.py
    - o Tampilkan hasilnya.
  - 4) **SELECT WHERE** (Tampilkan data berdasarkan parameter tertentu)
    - o Buat file 4-select-where-jenis.py dan 4-select-where-jumlah.py
    - o Tampilkan berdasarkan **jenis = mamalia** saja.
    - o Tampilkan berdasarkan fauna dengan **jml\_skrng** kurang dari sama dengan 1000 ekor saja.
    - o Tampilkan hasilnya.
  - 5) **SELECT WHERE AND** (Tampilkan data berdasarkan operator AND)
    - o Buat file **5-select-where-and.pv**
    - o Tampilkan berdasarkan jenis (Mamalia) dan asal (Sulawesi)
    - o Tampilkan hasilnya.
  - 6) **SELECT WHERE OR**(Tampilkan data berdasarkan operator OR)
    - o Buat file **6-select-where-or.py**

- Tampilkan berdasarkan asal(Sumatera) dan jml\_skrng lebih dari 500 ekor.
- o Tampilkan hasilnya.
- 7) **SELECT SUM** (Menjumlahkan isian field tertentu)
  - o Buat file **7-select-sum.py**
  - Jumlahkan total populasi hewan langka saat ini (Total Populasi) dari jml\_skrng.
  - o 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
  - o 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).
  - o Tampilkan hasilnya.
- 9) **SELECT LIKE** (Filter karakter data)
  - o Buat file **9-select-like.py**
  - o Cari nama fauna yang diawali dengan karakter "B"
  - o Tampilkan.
- 10) **UPDATE SET** (Memperbarui data)
  - o Buat file 10-select-update1.py, 10-select-update2.py
  - o Update jml\_skrng dari fauna'Katak Borneo' menjadi 650 (1).
  - o Update asal dari fauna 'Pesut Mahakam' menjadi 'Kalimantan Timur' (2).
  - o Tampilkan hasilnva.
- 11) **DELETE FROM** (Menghapus Data)
  - o File 11-delete-fauna.py
  - O Hapus isian field yang memiliki asal = Kalimantan.
  - o Buat file **delete-from.pv**
  - o Tampilkan **sebelum** dihapus.
  - o Tampilkan **setelah** dihapus.

#### 8. Ketentuan lainnya:

- o 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	Yolisia Enjel	
NPM	5230411144	
Mata Kuliah	Algoritma Pemrograman Praktik V	
Projek	Projek Pertemuan 12	

## Copy Paste Codingan:

# 1. Soal 1 (CREATE)

## 2. Soal 2 (INSERT INTO)

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	Mamlia	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

#### 3. Soal 3 (SELECT ALL)

```
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Mengambil semua data dalam table dan tampilkan
kursor.execute("SELECT * FROM FAUNA")
# Tampilkan dalam bentuk baris
baris_tabel = kursor.fetchall()
# Membuat format table dengan method format()
print("Tabel FAUNA :")
print("="*120)
print("{:<10} {:<20} {:<20} {:<20} {:<20}".format("ID FAUNA", "NAMA FAUNA",
"JENIS", "ASAL", "JUMLAH SAAT INI", "TAHUN TERAKHIR DITEMUKAN"))
print("="*120)</pre>
# Tampilkan data sesuai format table dengan perulangan
for baris in baris tabel:
    print("{:<10} {:<19} {:<20} {:<22} {:<22}".format(baris[0], baris[1],</pre>
baris[2], baris[3], baris[4], baris[5]))
koneksi.close()
```

#### 4. Soal 4 (SELECT WHERE)

- Select Where Jenis

```
# SELECT ALL DATA FAUNA
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Mengambil semua data dalam table dan tampilkan
kursor.execute("SELECT * FROM FAUNA WHERE jenis = 'Mamalia'")
# Tampilkan dalam bentuk baris
baris table = kursor.fetchall()
# Membuat format table dengan method format()
print("Table FAUNA :")
-----<del>-----</del>
-----")
# Tampilkan data sesuai format table dengan perulangan
for baris in baris table:
   print("{:<10} {:<19} {:<20} {:<22} {:<22}".format(baris[0], baris[1],</pre>
baris[2], baris[3], baris[4], baris[5]))
# Tutup Koneksi
koneksi.close()
```

#### - Select Where Jumlah

```
SELECT ALL DATA FAUNA
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Mengambil semua data dalam table dan tampilkan
kursor.execute("SELECT * FROM FAUNA WHERE jml_skrng <= '1000'")</pre>
# Tampilkan dalam bentuk baris
baris_table = kursor.fetchall()
# Membuat format table dengan method format()
print("Tabel FAUNA :")
# Tampilkan data sesuai format table dengan perulangan
for baris in baris_table:
  print("{:<10} {:<19} {:<20} {:<25} {:<22} ".format(baris[0], baris[1],</pre>
baris[2], baris[3], baris[4], baris[5]))
koneksi.close()
```

#### 5. Soal 5 (SELECT WHERE AND)

```
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Mengambil semua data dalam table dan tampilkan
kursor.execute(f"SELECT * FROM FAUNA WHERE jenis = 'Mamalia' AND asal =
'Sulawesi'")
# Tampilkan dalam bentuk baris
baris_table = kursor.fetchall()
# Membuat format table dengan method format()
print("Tabel FAUNA :")
# Tampilkan data sesuai format table dengan perulangan
for baris in baris table:
  print("{:<10} {:<19} {:<20} {:<25} {:<22} ".format(baris[0], baris[1],</pre>
baris[2], baris[3], baris[4], baris[5]))
koneksi.close()
```

#### 6. Soal 6 (SELECT WHERE OR)

```
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Mengambil semua data dalam table dan tampilkan
kursor.execute(f"SELECT * FROM FAUNA WHERE asal = 'Sumatera' OR jml_skrng >=
'500'")
# Tampilkan dalam bentuk baris
baris_table = kursor.fetchall()
# Membuat format table dengan method format(
# Tampilkan data sesuai format table dengan perulangan
for baris in baris_table:
  print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1],</pre>
baris[2], baris[3], baris[4], baris[5]))
koneksi.close()
```

## 7. Soal 7 (SELECT SUM)

```
# Impor Module dan Cek koneksi database
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
cursor = koneksi.cursor()

# Menjalankan query SUM
cursor.execute("SELECT SUM(jml_skrng) FROM FAUNA")
total_populasi = cursor.fetchone()[0]

print(f"Jumlah total populasi hewan langka saat ini : {total_populasi}")

# Menutup koneksi
koneksi.close()
```

## 8. Soal 8 (SELECT ORDER BY)

- orderby1

```
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Menjalankan query SELECT dengan ORDER BY
kursor.execute("SELECT * FROM FAUNA ORDER BY nama fauna ASC") #ASC
baris table = kursor.fetchall()
# Membuat format table dengan method format()
·----")
# Tampilkan data sesuai format table dengan perulangan
for baris in baris_table:
print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))</pre>
# Tutup Koneksi
koneksi.close()
```

- orderby2

```
# Impor Module dan Cek koneksi database
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Menjalankan query SELECT dengan ORDER BY
kursor.execute("SELECT * FROM FAUNA ORDER BY jml skrng DESC") #DESC
baris_table = kursor.fetchall()
# Membuat format table dengan method format()
print("Table fauna :")
print("{:<10} {:<20} {:<20} {:<20} {:<20}".format("ID FAUNA", "NAMA FAUNA",
# Tampilkan data sesuai format table dengan perulangan
for baris in baris_table:
print("{:<10} {:<19} {:<20} {:<25} {:<22} {:<22}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))</pre>
koneksi.close()
```

#### - orderby3

```
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Menjalankan query SELECT dengan ORDER BY
kursor.execute("SELECT * FROM FAUNA ORDER BY thn_ditemukan ASC") #ASC
baris_table = kursor.fetchall()
# Membuat format table dengan method format()
print("Tabel FAUNA :")
print("{:<10} {:<20} {:<20} {:<20} {:<20}".format("ID FAUNA", "NAMA FAUNA",</pre>
# Tampilkan data sesuai format table dengan perulangan
for baris in baris table:
  print("{:<10} {:<19} {:<20} {:<25} {:<22} ".format(baris[0], baris[1],</pre>
baris[2], baris[3], baris[4], baris[5]))
koneksi.close()
```

#### 9. Soal 9 (SELECT LIKE)

```
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
kursor = koneksi.cursor()
# Menjalankan query SELECT dengan LIKE
nama fauna = 'B%' # Mencari nama yang dimulai dengan 'B'
kursor.execute(f"SELECT * FROM FAUNA WHERE nama_fauna LIKE ?", (nama_fauna,))
baris_table = kursor.fetchall()
# Membuat format table dengan method format()
print("Tabel FAUNA :")
print("-----
=========""
print("{:<10} {:<20} {:<20} {:<20} {:<20}".format("ID FAUNA", "NAMA FAUNA",
"JENIS", "ASAL", "JUMLAH SAAT INI", "TAHUN TERAKHIR DITEMUKAN"))
print("-----
# Tampilkan data sesuai format table dengan perulangan
for baris in baris_table:
print("{:<10} {:<19} {:<20} {:<25} {:<22} ".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))</pre>
koneksi.close()
```

## 10. Soal 10 (UPDATE SET)

- Update1

```
# UPDATE table_name
 # WHERE condition;
 import sqlite3
 # Membuat koneksi ke database atau membuat database baru jika belum ada
 koneksi = sqlite3.connect('database_fauna.db.')
 cursor = koneksi.cursor()
 # Data yang ingin diubah
 id_fauna = 10
 jml_skrng_baru = 650
 # Menjalankan query UPDATE
 cursor.execute(f"UPDATE FAUNA SET jml_skrng = {jml_skrng_baru} WHERE id_fauna =
 {id fauna}")
 koneksi.commit()
 if cursor.rowcount > 0:
     print(f"Data fauna dengan ID {id_fauna} berhasil diupdate.")
 else:
     print(f"Tidak ada data fauna dengan ID {id_fauna}.")
 koneksi.close()
- Update2
 # UPDATE table_name
 # SET column1 = value1, column2 = value2, ...
```

```
# UPDATE table_name
# SET column1 = value1, column2 = value2, ...
# WHERE condition;
import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada
koneksi = sqlite3.connect('database_fauna.db.')
cursor = koneksi.cursor()

# Data yang ingin diubah

# Menjalankan query UPDATE
cursor.execute(f"UPDATE FAUNA SET asal = 'Kalimantan Timur' WHERE nama_fauna = 'Pesut Mahakam'")
koneksi.commit()

# Menampilkan pesan setelah update berhasil
if cursor.rowcount > 0:
    print(f"Data fauna berhasil diupdate")
else:
    print(f"Tidak ada data fauna")

# Menutup koneksi
koneksi.close()
```

## 11. Soal 11 (DELETE FROM)

```
# Impor module dan Cek koneksi database
import sqlite3
koneksi = sqlite3.connect('database_fauna.db.')
cursor = koneksi.cursor()
# Jalankan Query DELETE dan simpan perubahan
cursor.execute("DELETE FROM Fauna WHERE asal = 'Kalimantan'")
koneksi.commit()
# Jalankan Query SELECT
cursor.execute("SELECT * FROM Fauna")
dataFauna = cursor.fetchall()
# Tampilkan data dalam format tabel setelah di hapus
if len(dataFauna) > 0:
print("="*100)
    print("{:<5}{:<20}{:<20}{:<20}{:<20}".format("ID", "NAMA FAUNA", "JENIS",
"ASAL", "JUMLAH SAAT INI", "TAHUN DITEMUKAN"))
    print("="*100)</pre>
     for value in dataFauna:
        print("{:<5}{:<20}{:<20}{:<20}{:<20}".format(value[0], value[1],</pre>
value[2], value[3], value[4],value[5]))
    else:
         print("="*100)
print("Data Fauna Yang berasal dari Kalimantan Berhasil Di hapus")
# Tutup Koneksi
koneksi.clos
```

## Screenshot Hasil Program:

## 1. Soal 1 (CREATE)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS D:\PROJEK-PERTEMUAN 12> & C:/Users/ACER/AppData/Local/Programs/Python/Python311/python.exe "d:/PROJEK-PERTEMUAN 12/create_fauna.py"

Tabel Database Berhasil Dibuat

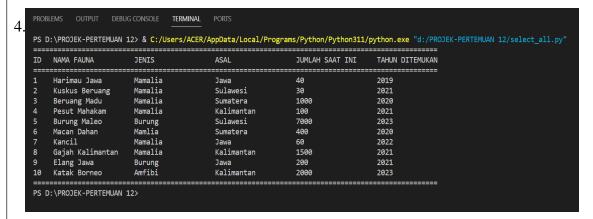
PS D:\PROJEK-PERTEMUAN 12>
```

## 2. Soal 2 (INSERT INTO)

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

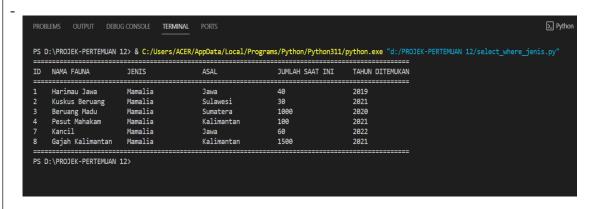
PS D:\PROJEK-PERTEMUAN 12> & C:/Users/ACER/AppData/Local/Programs/Python/Python311/python.exe "d:/PROJEK-PERTEMUAN 12/insert_fauna.py"
PS D:\PROJEK-PERTEMUAN 12>
```

## 3. Soal 3 (SELECT ALL)

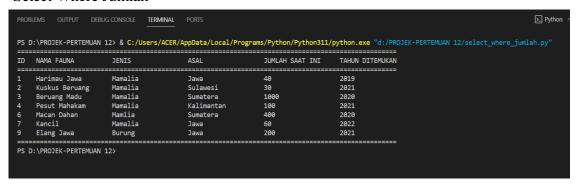


## 4. Soal 4 (SELECT WHERE)

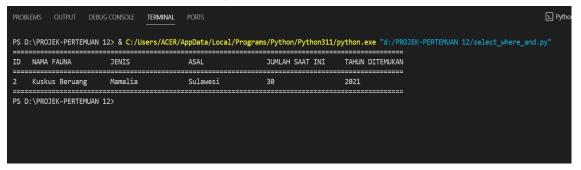
- Select Where Jenis



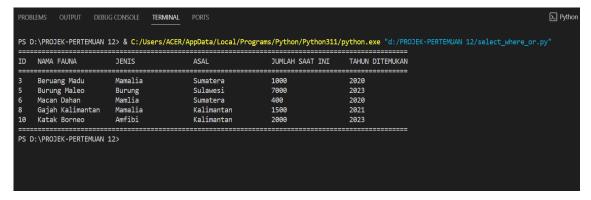
#### - Select Where Jumlah



# 5. Soal 5 (SELECT WHERE AND)



#### 6. Soal 6 (SELECT WHERE OR)

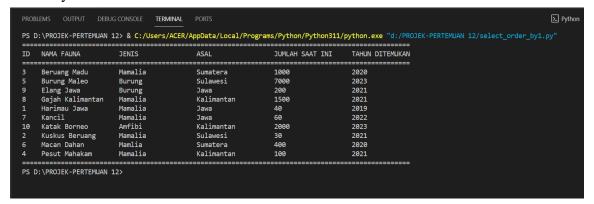


#### 7. Soal 7 (SELECT SUM)

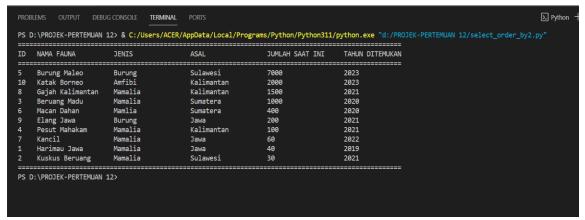


## 8. Soal 8 (SELECT ORDER BY)

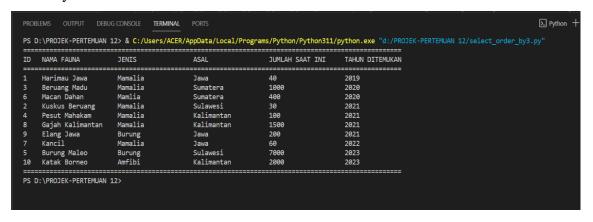
## - orderby1



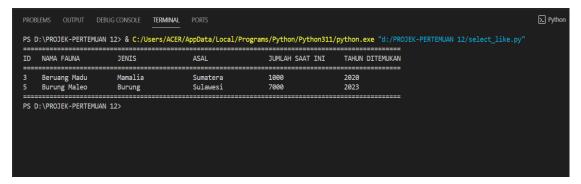
## - orderby2



## - orderby3

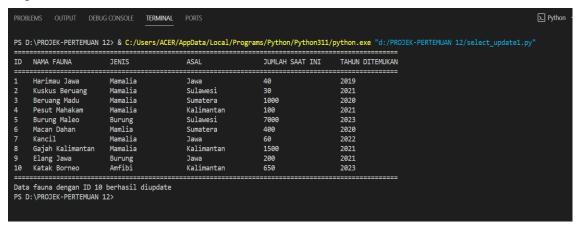


## 9. Soal 9 (SELECT LIKE)

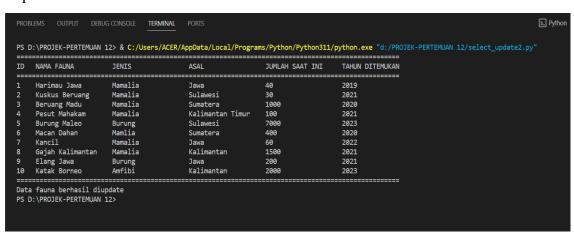


## 10. Soal 10 (UPDATE SET)

- update1

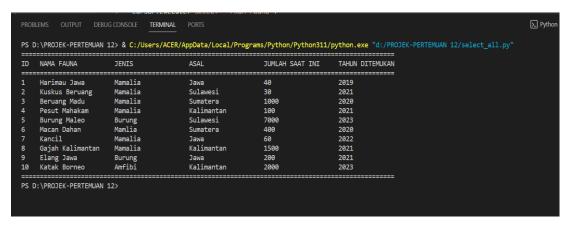


#### - update2



## 11. Soal 11 (DELETE FROM)

- Tampilan sebelum dihapus



- Tampilan setelah dihapus

