|  |  |  |
| --- | --- | --- |
| **A logo of the university of technology  Description automatically generated** | **Nama** | **Rangga Eka Pramana** |
| **NPM** | **5230411157** |
| **Mata Kuliah** | **Alpro Praktik** |
| **Projek** | **Pertemuan 12 Database Fauna** |

**1. Soal 1 (CREATE)**

import sqlite3

koneksi = sqlite3.connect("database\_fauna.db")

# insert data ke tabel

koneksi.execute('''

                CREATE TABLE FAUNA (

                    id\_fauna INTEGER PRIMARY KEY AUTOINCREMENT,

                    nama\_fauna VARCHAR(50),

                    jenis VARCHAR(50),

                    asal VARCHAR(50),

                    jml\_skrng INTEGER(10),

                    thn\_ditemukan INTEGER(10)

                )

               ''')

koneksi.close()

**2. Soal 2 (INSERT INTO)**

import sqlite3

koneksi = sqlite3.connect("database\_fauna.db")

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Harimau Jawa','Mamalia','Jawa','40','2019')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Kuskus Beruang','Mamalia','Sulawesi','30','2021')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Beruang Madu','Mamalia','Sumatera','1000','2020')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Pesut Mahakam','Mamalia','Kalimantan','100','2021')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Burung Maleo','Burung','Sulawesi','7000','2023')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Macan Dahan','Mamalia','Sumatera','400','2020')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Kancil','Mamalia','Jawa','60','2019')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Gajah Kalimantan','Mamalia','Kalimantan','1500','2021')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Elang Jawa','Burung','Jawa','200','2021')

                    ''')

koneksi.execute('''

                    INSERT INTO FAUNA(nama\_fauna, jenis, asal, jml\_skrng,thn\_ditemukan)

                    VALUES('Katak Borneo','Amfibi','Kalimantan','2000','2023')

                    ''')

koneksi.commit()

koneksi.close()

A screenshot of a computer

Description automatically generated

3. Soal 3 (SELECT ALL)

import sqlite3

koneksi = sqlite3.connect("database\_fauna.db")

kursor = koneksi.cursor()

#mengambil swmua data dalam tabel dan tampilkan

kursor.execute("SELECT \* FROM FAUNA")

# Tampilkan dalam bentuk baris

baris\_tabel = kursor.fetchall()

# membuat format table dengan method format

print("DATA FAUNA")

print("="\*90)

print("{:<5} {:18} {:10} {:10} {:<15} {:<20}".format("ID","Nama\_Fauna","Jenis","Asal","Jumlah Saat Ini","Tahun Terakhir"))

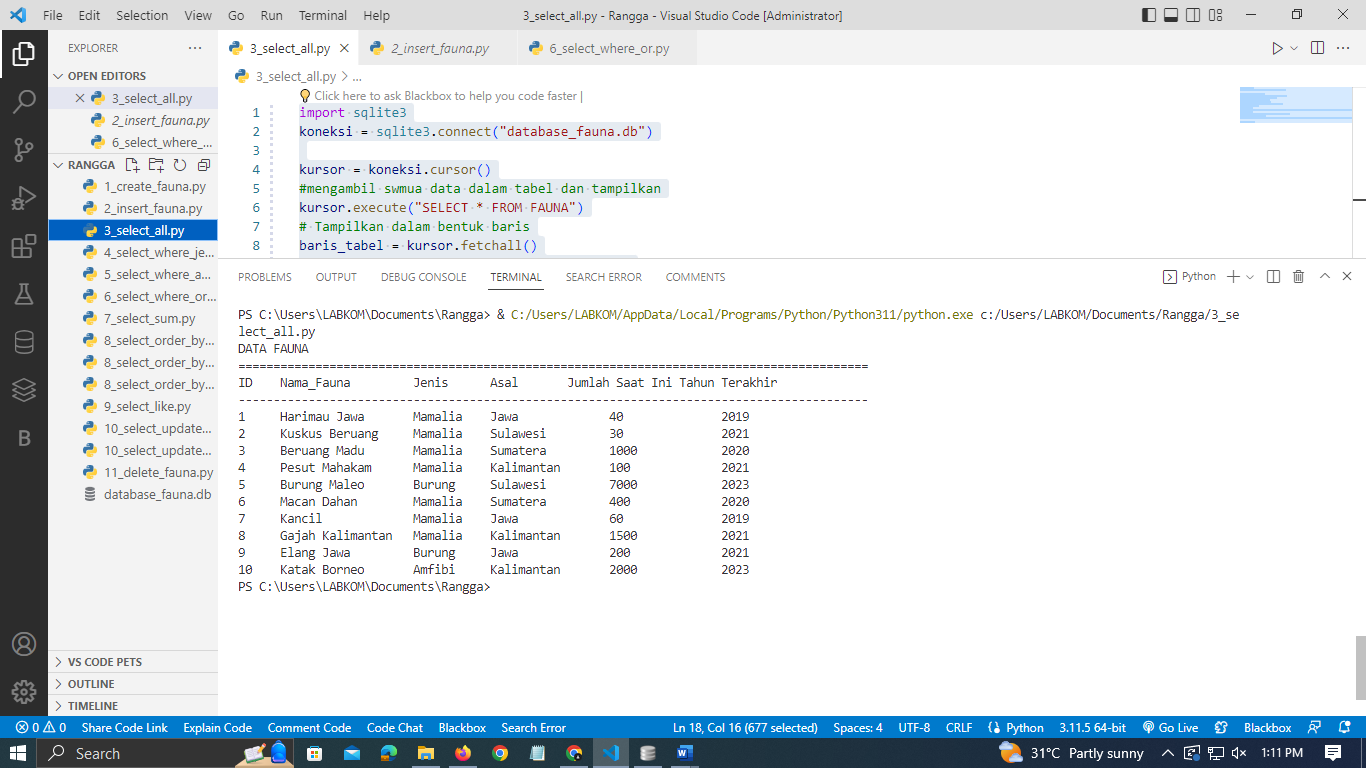
print("-"\*90)

# tampilkan data sesuai format tabel dg perulangan

for baris in baris\_tabel :

    print("{:<5} {:18} {:10} {:16} {:<15} {:<20}".format(baris[0],baris[1],baris[2],baris[3],baris[4],baris[5]))

koneksi.close()



4. Soal 4 (SELECT WHERE)

- **Select Where Jenis**

import sqlite3

koneksi = sqlite3.connect("database\_fauna.db")

kursor = koneksi.cursor()

#mengambil swmua data dalam tabel dan tampilkan

kursor.execute("SELECT \* FROM FAUNA WHERE jenis = 'Mamalia' AND jml\_skrng <'1000'")

# Tampilkan dalam bentuk baris

baris\_tabel = kursor.fetchall()

# membuat format table dengan method format

print("DATA FAUNA")

print("="\*90)

print("{:<5} {:18} {:10} {:15} {:15} {:>20}".format("ID","Nama\_Fauna","Jenis","Asal","Jumlah Saat Ini","Tahun Terakhir"))

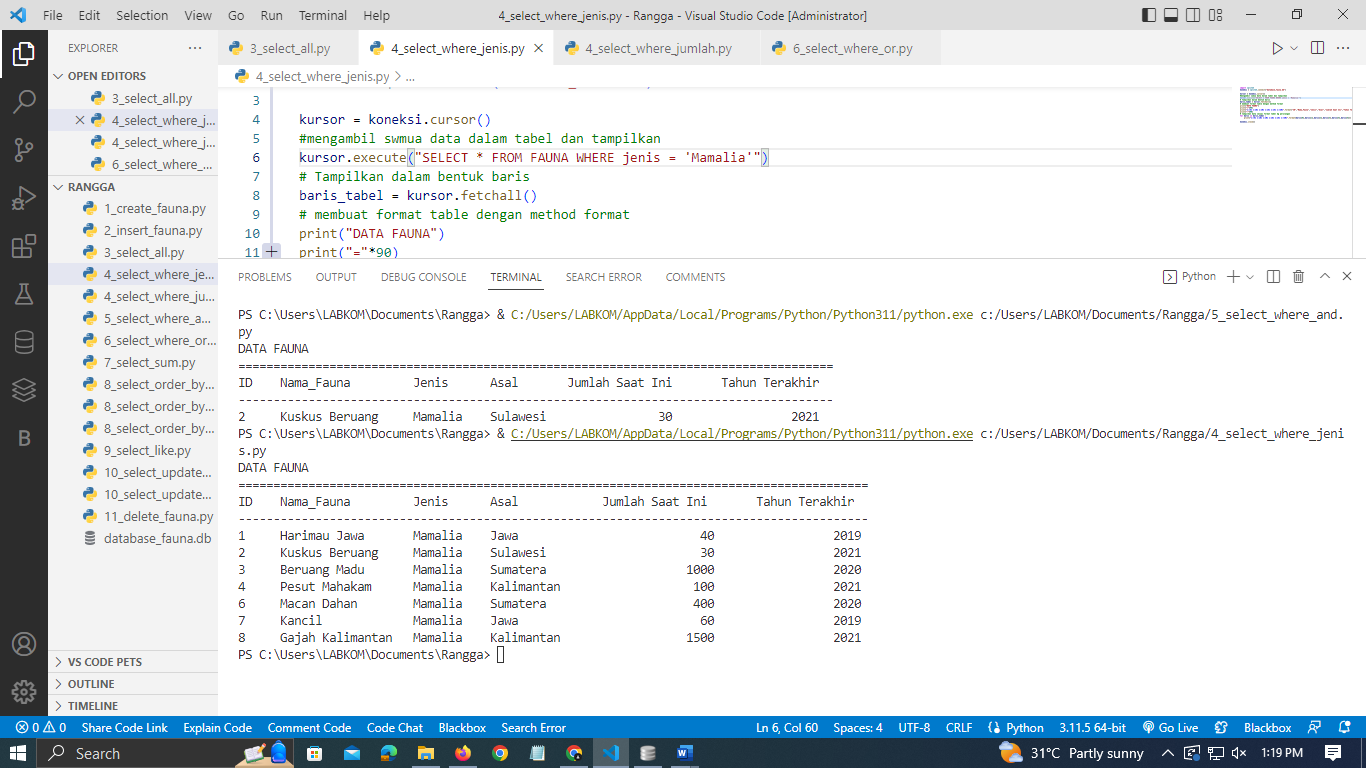
print("-"\*90)

# tampilkan data sesuai format tabel dg perulangan

for baris in baris\_tabel :

    print("{:<5} {:18} {:10} {:16} {:15} {:>20}".format(baris[0],baris[1],baris[2],baris[3],baris[4],baris[5]))

koneksi.close()



**- Select Where Jumlah**

import sqlite3

koneksi = sqlite3.connect("database\_fauna.db")

kursor = koneksi.cursor()

#mengambil swmua data dalam tabel dan tampilkan

kursor.execute("SELECT \* FROM FAUNA WHERE jml\_skrng <'1000'")

# Tampilkan dalam bentuk baris

baris\_tabel = kursor.fetchall()

# membuat format table dengan method format

print("DATA FAUNA")

print("="\*90)

print("{:<5} {:18} {:10} {:15} {:15} {:>20}".format("ID","Nama\_Fauna","Jenis","Asal","Jumlah Saat Ini","Tahun Terakhir"))

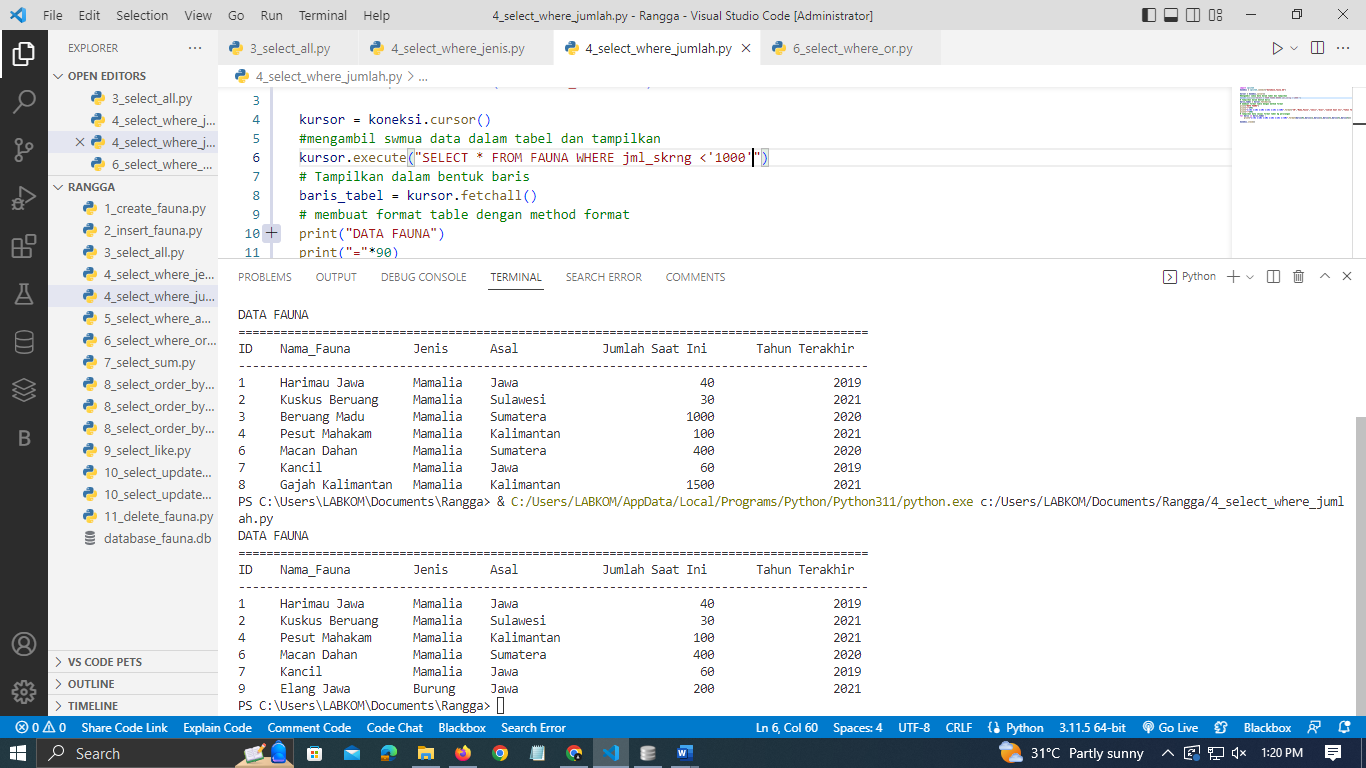
print("-"\*90)

# tampilkan data sesuai format tabel dg perulangan

for baris in baris\_tabel :

    print("{:<5} {:18} {:10} {:16} {:15} {:>20}".format(baris[0],baris[1],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 swmua data dalam tabel dan tampilkan

kursor.execute("SELECT \* FROM FAUNA WHERE jenis = 'Mamalia' AND asal= 'Sulawesi' ")

# Tampilkan dalam bentuk baris

baris\_tabel = kursor.fetchall()

# membuat format table dengan method format

print("DATA FAUNA")

print("="\*85)

print("{:<5} {:18} {:10} {:10} {:15} {:>20}".format("ID","Nama\_Fauna","Jenis","Asal","Jumlah Saat Ini","Tahun Terakhir"))

print("-"\*85)

# tampilkan data sesuai format tabel dg perulangan

for baris in baris\_tabel :

    print("{:<5} {:18} {:10} {:10} {:15} {:>20}".format(baris[0],baris[1],baris[2],baris[3],baris[4],baris[5]))

koneksi.close()

A screenshot of a computer

Description automatically generated

6. Soal 6 (SELECT WHERE OR)

import sqlite3

conn = sqlite3.connect("database\_fauna.db")

cursor = conn.cursor()

# Jalankan Query SELECT WHERE OR

cursor.execute("SELECT \* FROM Fauna WHERE asal= 'Sumatera' OR jml\_skrng > 500 ")

dataFauna = cursor.fetchall()

# Tampilkan dalam format tabel

if len(dataFauna) > 0:

    print("-" \* 100)

    print(

        "{:<5}{:<20}{:<20}{:<20}{:<20}{:<20}".format(

            "ID", "NAMA FAUNA", "JENIS", "ASAL", "JUMLAH SAAT INI", "TAHUN DITEMUKAN"

        )

    )

    print("-" \* 100)

    for value in dataFauna:

        print(

            "{:<5}{:<20}{:<20}{:<20}{:<20}{:<20}".format(

                value[0], value[1], value[2], value[3], value[4], value[5]

            )

        )

    else:

        print("\*" \* 100)

        print(

            "Data diatas, Berdasarkan Hewan yang Berasal dari Sumatera dan Jumlah Lebih dari 500"

        )

conn.close()

# Tutup Koneksi

A screenshot of a computer

Description automatically generated

7. Soal 7 (SELECT SUM)

import sqlite3

koneksi = sqlite3.connect("database\_fauna.db")

kursor = koneksi.cursor()

kursor.execute("SELECT SUM (jml\_skrng) FROM FAUNA")

total = kursor.fetchone()[0]

print(f"Total Populasi :{total}")

koneksi.close()

A screenshot of a computer

Description automatically generated

8. Soal 8 (SELECT ORDER BY)

- orderby1

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

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|DESC

baris\_table = kursor.fetchall()

print("Data Fauna:")

print("==============================================================")

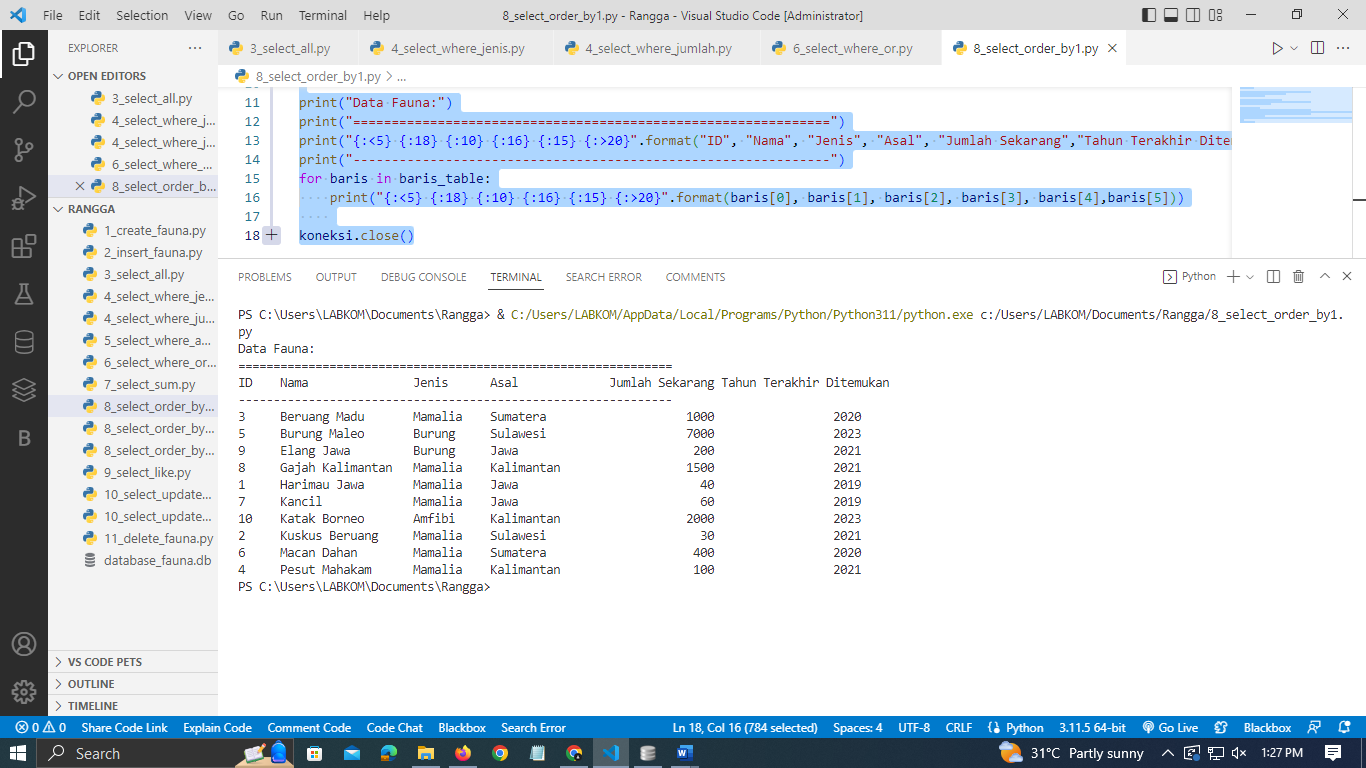
print("{:<5} {:18} {:10} {:16} {:15} {:>20}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir Ditemukan"))

print("--------------------------------------------------------------")

for baris in baris\_table:

    print("{:<5} {:18} {:10} {:16} {:15} {:>20}".format(baris[0], baris[1], baris[2], baris[3], baris[4],baris[5]))

koneksi.close()



- orderby2

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

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") #ASC|DESC

baris\_table = kursor.fetchall()

print("Data Fauna:")

print("================================================================================================")

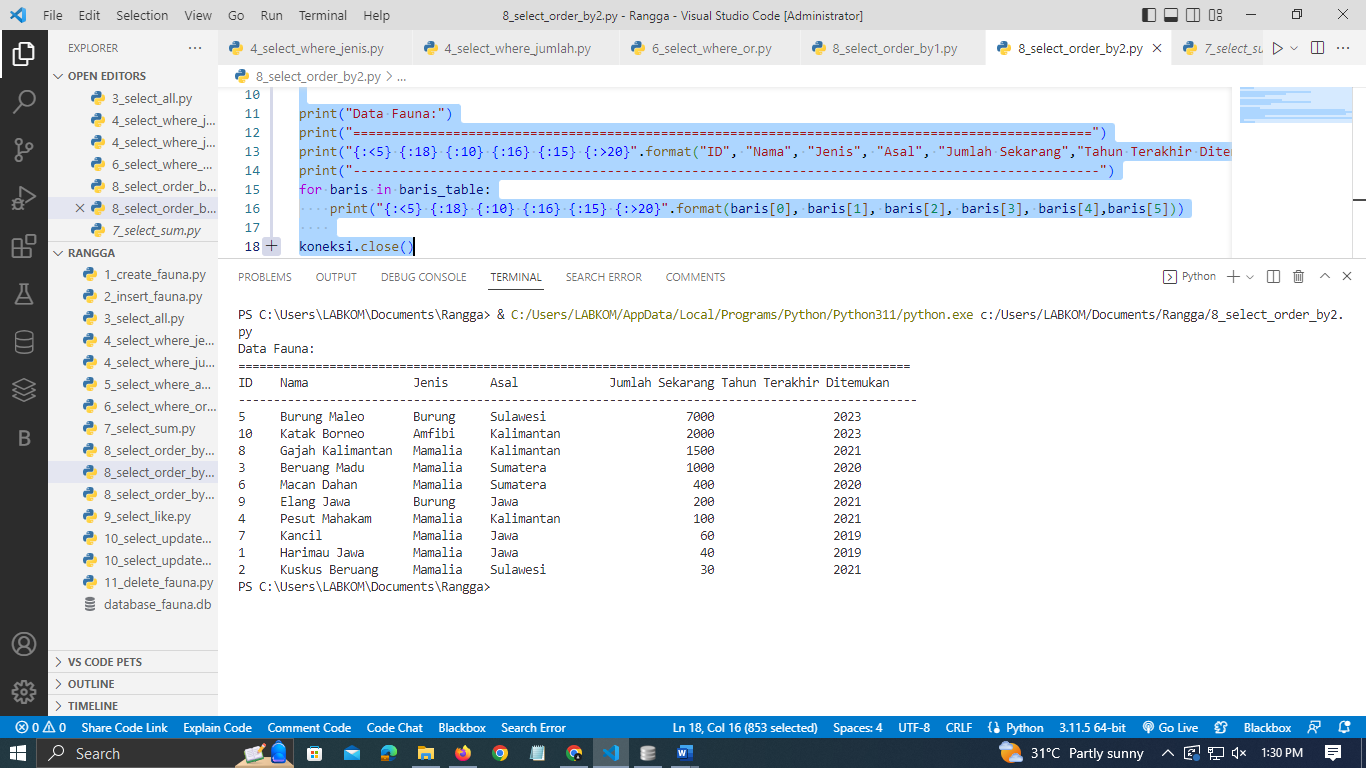
print("{:<5} {:18} {:10} {:16} {:15} {:>20}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir Ditemukan"))

print("-------------------------------------------------------------------------------------------------")

for baris in baris\_table:

    print("{:<5} {:18} {:10} {:16} {:15} {:>20}".format(baris[0], baris[1], baris[2], baris[3], baris[4],baris[5]))

koneksi.close()



- orderby3

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

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|DESC

baris\_table = kursor.fetchall()

print("Data Fauna:")

print("==============================================================================================")

print("{:<5} {:18} {:10} {:10} {:15} {:>20}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir"))

print("----------------------------------------------------------------------------------------------")

for baris in baris\_table:

    print("{:<5} {:18} {:10} {:10} {:15} {:>20}".format(baris[0], baris[1], baris[2], baris[3], baris[4],baris[5]))

koneksi.close()

A screenshot of a computer

Description automatically generated

9. Soal 9 (SELECT LIKE)

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

koneksi = sqlite3.connect('database\_fauna.db')

kursor = koneksi.cursor()

# Menjalankan query SELECT dengan LIKE

nama = 'B%'  # Mencari nama

kursor.execute(f"SELECT \* FROM FAUNA WHERE nama\_fauna LIKE ?", (nama,))

baris\_table = kursor.fetchall()

print("Data Fauna :")

print("==============================================================")

print("{:<5} {:18} {:10} {:10} {:15} {:>20}".format("ID", "Nama", "Jenis", "Asal", "Jumlah Sekarang","Tahun Terakhir"))

print("----------------------------------------------------------------------------------------------")

for baris in baris\_table:

    print("{:<5} {:18} {:10} {:10} {:15} {:>20}".format(baris[0], baris[1], baris[2], baris[3], baris[4],baris[5]))

koneksi.close()

A screenshot of a computer

Description automatically generated

10. Soal 10 (UPDATE SET)

- update 1

# UPDATE table\_name

# SET column1 = value1, column2 = value2, ...

# WHERE condition;

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

conn = sqlite3.connect('database\_fauna.db')

cursor = conn.cursor()

# Data yang ingin diubah

id\_fauna =10

jml\_baru = 650

# Menjalankan query UPDATE

cursor.execute(f"UPDATE FAUNA SET jml\_skrng = {jml\_baru} WHERE id\_fauna = {id\_fauna}")

conn.commit()

# Menampilkan pesan setelah update berhasil

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}.")

# Menutup koneksi

conn.close()

A screenshot of a computer

Description automatically generated

- Update 2

# UPDATE table\_name

# SET column1 = value1, column2 = value2, ...

# WHERE condition;

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

conn = sqlite3.connect('database\_fauna.db')

cursor = conn.cursor()

# Data yang ingin diubah

id\_fauna = 4

asal\_baru = "Kalimantan Timur"

# Menjalankan query UPDATE

cursor.execute("UPDATE FAUNA SET asal = ? WHERE id\_fauna = ?", (asal\_baru, id\_fauna))

conn.commit()

# Menampilkan pesan setelah update berhasil

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}.")

# Menutup koneksi

conn.close()

A screenshot of a computer

Description automatically generated

11. Soal 11 (DELETE FROM)

import sqlite3

# Membuat koneksi ke database atau membuat database baru jika belum ada

conn = sqlite3.connect('database\_fauna.db')

cursor = conn.cursor()

# Menjalankan query DELETE

asal = "Kalimantan"  # Ganti dengan kriteria yang sesuai untuk data yang ingin dihapus

cursor.execute("DELETE FROM FAUNA WHERE asal = ?", (asal,))

conn.commit()

# Menampilkan pesan setelah penghapusan berhasil

if cursor.rowcount > 0:

    print(f"Data fauna dengan Asal {asal} berhasil dihapus.")

else:

    print(f"Tidak ada data fauna dengan Asal {asal}.")

# Menutup koneksi

conn.close()

A computer screen shot of a computer

Description automatically generated