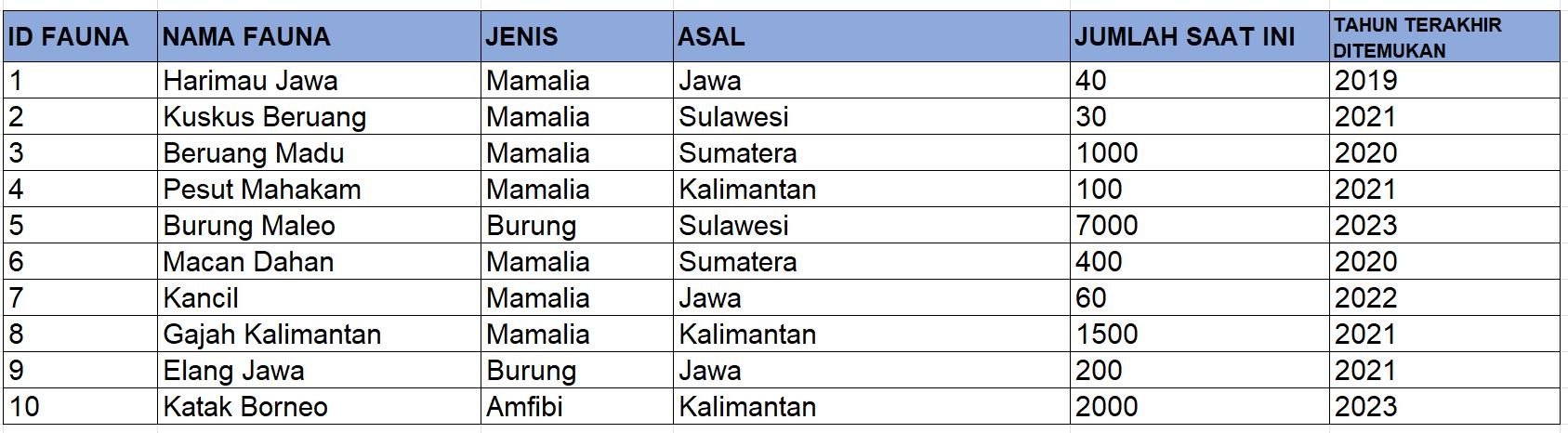
PROJEK PERTEMUAN 12

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:**
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:
8. **CREATE** Database dan Tabel
   * Buat file **1-create-fauna.py**
9. **INSERT INTO** (Menambahkan data kedalam table)
   * Buat file **2-insert-fauna.py**
10. **SELECT ALL** (Tampilkan semua data tabel)

* Buat file **3-select-all.py**

# Tampilkan hasilnya.

1. **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.

1. **SELECT WHERE AND** (Tampilkan data berdasarkan operator AND)

* Buat file **5-select-where-and.py**
* Tampilkan berdasarkan **jenis (Mamalia)** dan **asal (Sulawesi)**

# Tampilkan hasilnya.

1. **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.

1. **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.

1. **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.

1. **SELECT LIKE** (Filter karakter data)

* Buat file **9-select-like.py**
* Cari nama fauna yang diawali dengan karakter “B”
* Tampilkan.

1. **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.

1. **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.

# 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 | Adhyasta Adwaya Al Karim |
| NPM | 5230411165 |
| Mata Kuliah | Algoritma Pemrograman Praktik V |
| Projek | Projek Pertemuan 12 |

*Copy Paste Codingan:*

1. Soal 1 (CREATE)
2. import sqlite3
3. koneksi =sqlite3.connect("database\_fauna.db")
4. koneksi.execute('''
5. CREATE TABLE FAUNA(
6. id\_fauna INTEGER PRIMARY KEY AUTOINCREMENT,
7. nama\_fauna VARCHAR(50),
8. jenis VARCHAR(50),
9. asal VARCHAR(50),
10. jml\_skrng INTEGER(50),
11. thn\_ditemukan INTEGER(50)
12. )
13. ''')
14. koneksi.close()
15. Soal 2 (INSERT INTO)
16. import sqlite3
17. koneksi = sqlite3.connect('database\_fauna.db')
18. koneksi.execute(f'''
19. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
20. VALUES('Harimau Jawa','Mamalia','Jawa','40','2019')
21. ''')
22. koneksi.execute(f'''
23. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
24. VALUES('Kuskus Beruang','Mamalia','Sulawesi','30','2021')
25. ''')
26. koneksi.execute(f'''
27. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
28. VALUES('Beruang Madu','Mamalia','Sumatra','1000','2020')
29. ''')
30. koneksi.execute(f'''
31. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
32. VALUES('Pesut Mahakam','Mamalia','Kalimatan','100','2021')
33. ''')
34. koneksi.execute(f'''
35. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
36. VALUES('Burung Maleo','Burung','Sulawesi','7000','2023')
37. ''')
38. koneksi.execute(f'''
39. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
40. VALUES('Macan Dahan','Mamalia','Sumatra','400','2020')
41. ''')
42. koneksi.execute(f'''
43. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
44. VALUES('Kancil','Mamalia','Jawa','60','2022')
45. ''')
46. koneksi.execute(f'''
47. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
48. VALUES('Gajah Kalimantan','Mamalia','Kalimantan','1500','2021')
49. ''')
50. koneksi.execute(f'''
51. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
52. VALUES('Elang Jawa','Burung','Jawa','200','2021')
53. ''')
54. koneksi.execute(f'''
55. INSERT INTO FAUNA (nama\_fauna,jenis,asal,jml\_skrng,thn\_ditemukan)
56. VALUES('Katak Borneo','Amfibi','Kalimantan','2000','2023')
57. ''')
58. koneksi.commit()
59. koneksi.close()

Soal 3 (SELECT ALL) import sqlite3

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

kursor = koneksi.cursor()

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

baris\_tabel = kursor.fetchall()

print("Data Pegawai Konoha 2023")

print("="\*80)

print("{:<5}{:<20}{:<20}{:<20}{:<20}".format("ID"," nama\_fauna","jenis","asal"," jml\_skrng","thn\_ditemukan"))

print("="\*80)

for baris in baris\_tabel:

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

koneksi.close()

1. Soal 4 (SELECT WHERE)

import sqlite3

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

kursor = koneksi.cursor()

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

baris\_table = kursor.fetchall()

print("Data Fauna:")

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

print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<20}".format("ID"," nama\_fauna","jenis","asal"," jml\_skrng","thn\_ditemukan"))

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

for baris in baris\_table:

    print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<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()

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

baris\_table = kursor.fetchall()

print("Data Fauna:")

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

print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<20}".format("ID"," nama\_fauna","jenis","asal"," jml\_skrng","thn\_ditemukan"))

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

for baris in baris\_table:

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

koneksi.close()

1. Soal 5 (SELECT WHERE AND)
2. import sqlite3
3. # Membuat koneksi ke database atau membuat database baru jika belum ada
4. koneksi = sqlite3.connect('database\_fauna.db')
5. kursor = koneksi.cursor()
6. # Menjalankan query SELECT dengan ORDER BY
7. # AND harus dua-duanya terpenuhi
8. kursor.execute(f"SELECT \* FROM FAUNA WHERE jenis = 'Mamalia' AND asal = 'Sulawesi'")
9. baris\_table = kursor.fetchall()
10. print("Data Fauna:")
11. print("==============================================================")
12. print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format("ID"," nama\_fauna","jenis","asal","jml\_skrng","thn\_ditemukan"))
13. print("--------------------------------------------------------------")
14. for baris in baris\_table:
15. print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
17. koneksi.close()
18. Soal 6 (SELECT WHERE OR)
19. import sqlite3
20. # Membuat koneksi ke database atau membuat database baru jika belum ada
21. koneksi = sqlite3.connect('database\_fauna.db')
22. kursor = koneksi.cursor()
23. # Menjalankan query SELECT dengan ORDER BY
24. # OR cukup salah satu terpenuhi maka dapat dieksekusi
25. kursor.execute(f"SELECT \* FROM FAUNA WHERE asal = 'Sumatra' OR jml\_skrng >= '500'")
26. baris\_table = kursor.fetchall()
27. print("Data Fauna:")
28. print("==============================================================")
29. print("{:<5} {:<20} {:<20} {:<20} {:<10}".format("ID", "Nama", "Jabatan", "Kota", "Gaji"))
30. print("--------------------------------------------------------------")
31. for baris in baris\_table:
32. print("{:<5} {:<20} {:<20} {:<20} {:<10}".format(baris[0], baris[1], baris[2], baris[3], baris[4]))
34. koneksi.close()
35. Soal 7 (SELECT SUM)
36. import sqlite3
37. # Membuat koneksi ke database atau membuat database baru jika belum ada
38. conn = sqlite3.connect('database\_fauna.db')
39. cursor = conn.cursor()
40. # Menjalankan query SUM
41. cursor.execute("SELECT SUM(jml\_skrng) FROM FAUNA")
42. total\_populasi = cursor.fetchone()[0]
43. print(f"Total jumlah populaso seluruh fauna: {total\_populasi}")
44. # Menutup koneksi
45. conn.close()
46. 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
* # AND harus dua-duanya terpenuhi
* kursor.execute(f"SELECT \* FROM FAUNA ORDER BY nama\_fauna ASC ")
* baris\_table = kursor.fetchall()
* print("Data Fauna:")
* print("==============================================================")
* print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format("ID"," nama\_fauna","jenis","asal","jml\_skrng","thn\_ditemukan"))
* print("--------------------------------------------------------------")
* for baris in baris\_table:
* print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".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
* # AND harus dua-duanya terpenuhi
* kursor.execute(f"SELECT \* FROM FAUNA ORDER BY jml\_skrng ASC ")
* baris\_table = kursor.fetchall()
* print("Data Fauna:")
* print("==============================================================")
* print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format("ID"," nama\_fauna","jenis","asal","jml\_skrng","thn\_ditemukan"))
* print("--------------------------------------------------------------")
* for baris in baris\_table:
* print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".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

# AND harus dua-duanya terpenuhi

kursor.execute(f"SELECT \* FROM FAUNA ORDER BY thn\_ditemukan ASC ")

baris\_table = kursor.fetchall()

print("Data Fauna:")

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

print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format("ID"," nama\_fauna","jenis","asal","jml\_skrng","thn\_ditemukan"))

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

for baris in baris\_table:

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

koneksi.close()

1. Soal 9 (SELECT LIKE)

* update1
* 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 yang dimulai dengan 'John'
* kursor.execute(f"SELECT \* FROM FAUNA WHERE nama\_fauna LIKE ?", (nama,))
* baris\_table = kursor.fetchall()
* print("Data Fauna:")
* print("==============================================================")
* print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format("ID"," nama\_fauna","jenis","asal","jml\_skrng","thn\_ditemukan"))
* print("--------------------------------------------------------------")
* for baris in baris\_table:
* print("{:<5} {:<20} {:<20} {:<20} {:<10} {:<10}".format(baris[0], baris[1], baris[2], baris[3], baris[4], baris[5]))
* koneksi.close()
* update2

1. Soal 10 (UPDATE SET)
2. import sqlite3
3. koneksi = sqlite3.connect('database\_fauna.db')
4. kursor = koneksi.cursor()
6. kursor.execute("""UPDATE FAUNA SET asal = 'Kalimantan Timur'  WHERE id\_fauna = 4
7. """)
8. koneksi.commit()
9. # cek apakah data berhasil diubah atau belum
10. if kursor.rowcount > 0: # cek berdasarkan adanya baris atau tidak
11. print(f"Data berhasil Diubah!")
12. else:
13. print(f"Tidak ada data dengan ID tersebut!")

16. koneksi.close()
17. Soal 11 (DELETE FROM)

import sqlite3

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

kursor = koneksi.cursor()

kursor.execute(f"DELETE FROM FAUNA WHERE asal = 'Kalimantan' ")

koneksi.commit()

# cek apakah data berhasil diubah atau belum

if kursor.rowcount > 0: # cek berdasarkan adanya baris atau tidak

    print(f"Data dengan ID  berhasil Dihapus!")

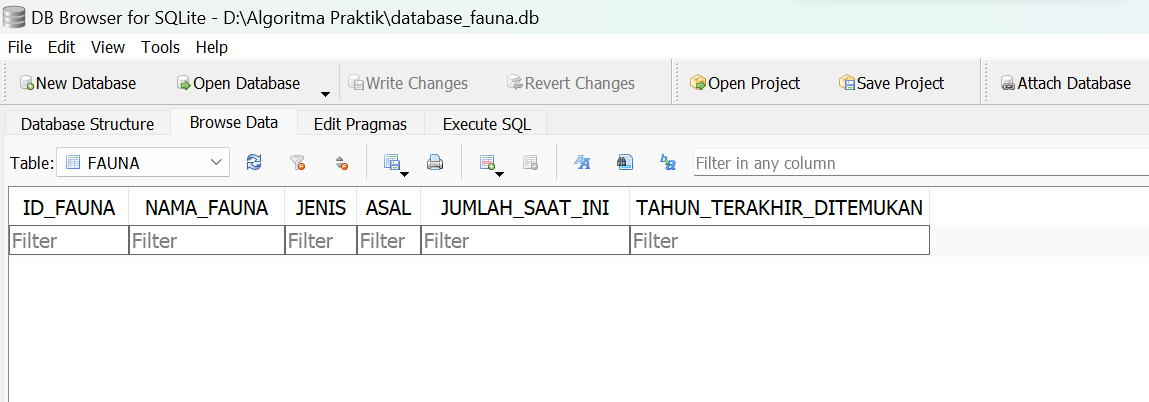
else:

    print(f"Tidak ada data pegawai dengan ID!")

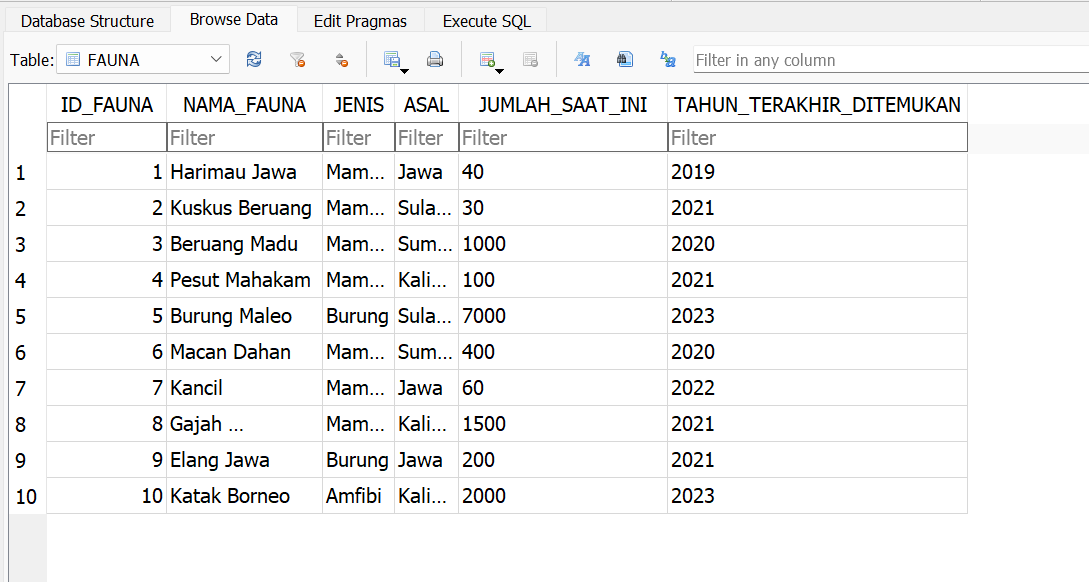
koneksi.close()

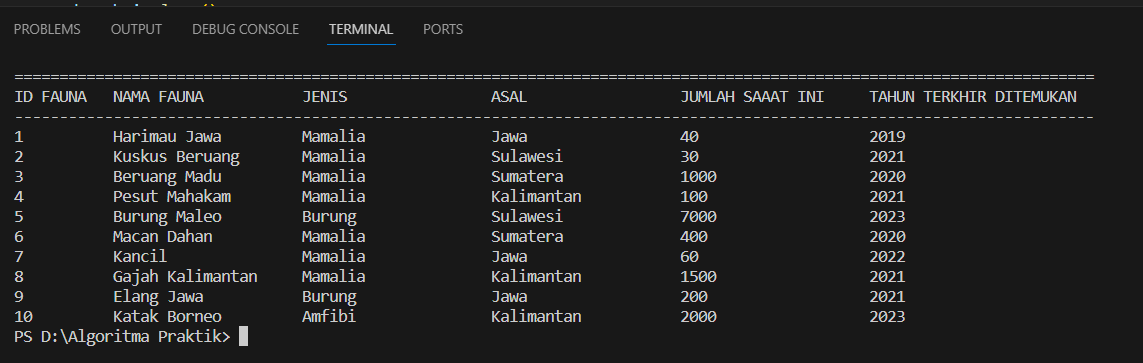
*Screenshot Hasil Program:*

1. Soal 1 (CREATE)

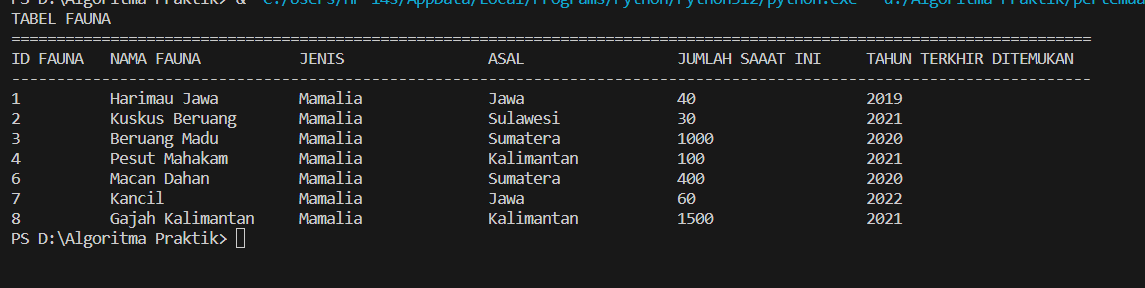


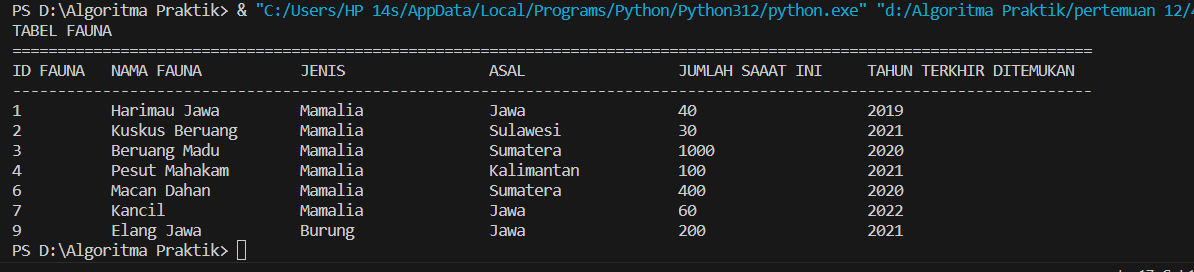
1. Soal 2 (INSERT INTO)

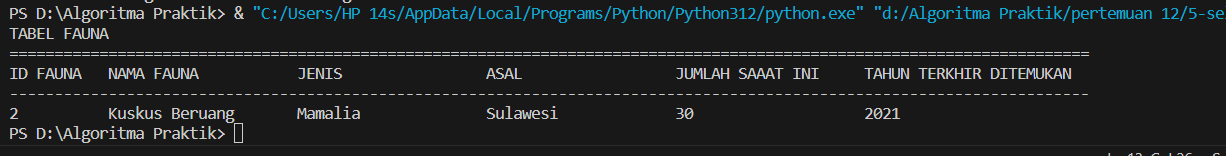


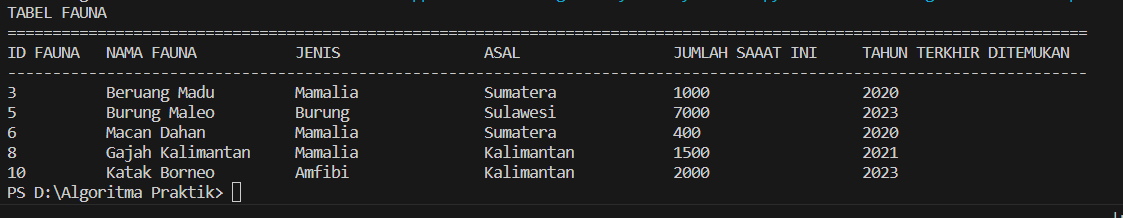
1. Soal 3 (SELECT ALL) 
2. Soal 4 (SELECT WHERE)

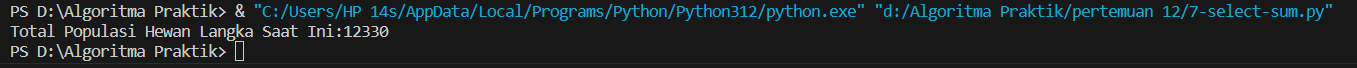
* Select Where Jenis

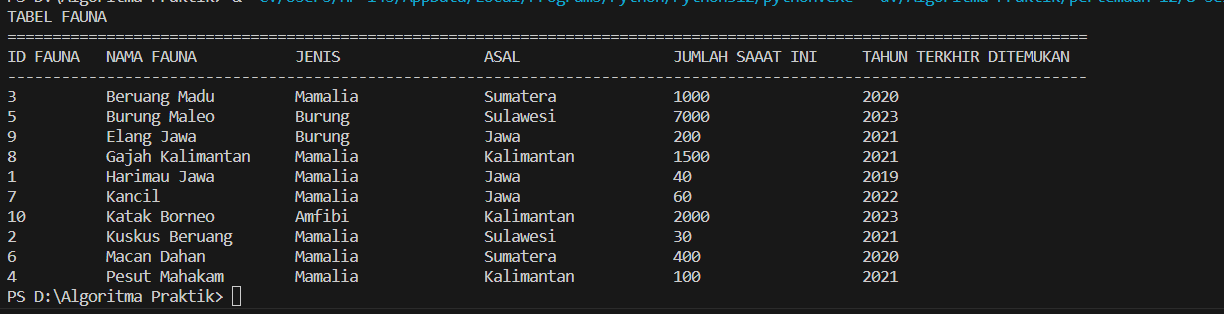


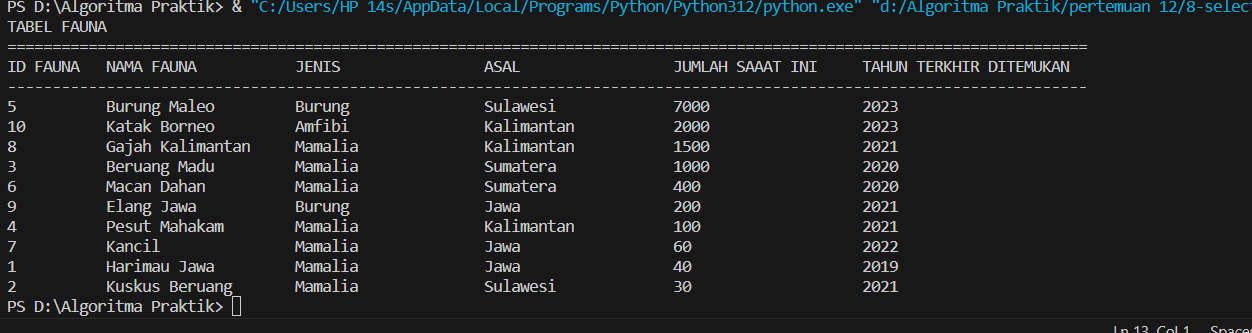
* Select Where Jumlah
* 

1. Soal 5 (SELECT WHERE AND) 
2. Soal 6 (SELECT WHERE OR)



1. Soal 7 (SELECT SUM) 
2. Soal 8 (SELECT ORDER BY)

* orderby1
* 
* orderby2



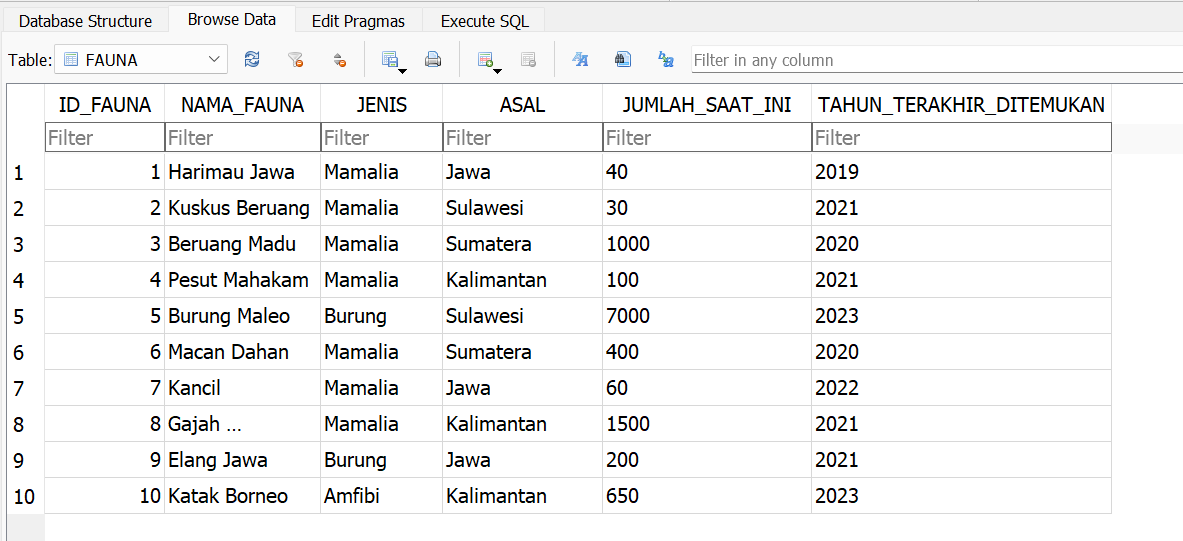
* orderby3

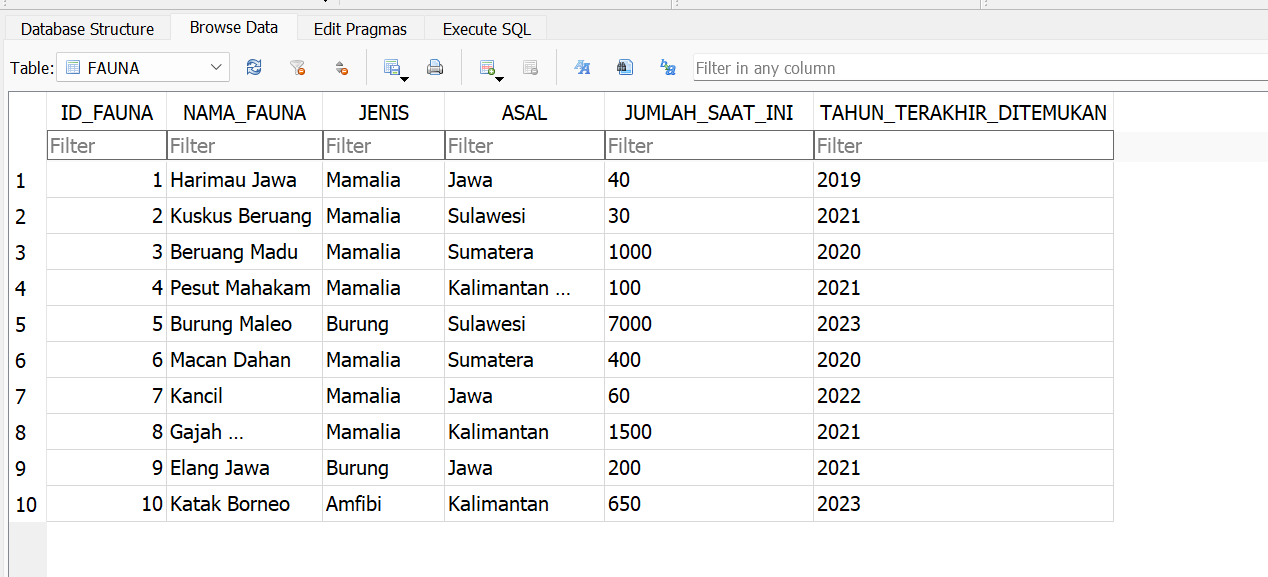


1. Soal 9 (SELECT LIKE)

* update1



1. Soal 10 (UPDATE SET) 



1. Soal 11 (DELETE FROM)

