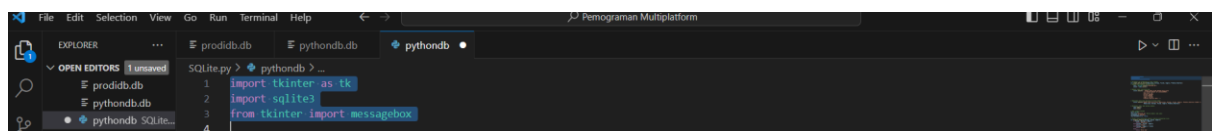


CODE PYHTON

1. **'import tkinter as tk' →** Untuk mengimpor modul tkinter, yang merupakan toolkit GUI standar untuk Python. Alias 'tk' digunakan untuk merujuk modul 'tkinter' dengan lebih singkat. 'tkinter' menyediakan seperangkat alat untuk membuat antarmuka pengguna grafis, seperti jendela, tombol, kotak teks, dan sebagainya.

import sqlite3 → Untuk mengimpor modul sqlite3, yang merupakan modul bawaan di Python yang menyediakan cara sederhana untuk berinteraksi dengan database SQLite. SQLite adalah mesin database ringan yang tidak memerlukan server dan sering digunakan dalam banyak aplikasi.

from tkinter import messagebox → Untuk mengimpor kelas messagebox dari modul tkinter. Kelas messagebox digunakan untuk membuat kotak pesan pop-up untuk menampilkan informasi, peringatan, kesalahan, atau meminta input pengguna.



2. **Fungsi simpan_data_ke_sqlite →** untuk menyimpan data siswa ke dalam database SQLite.

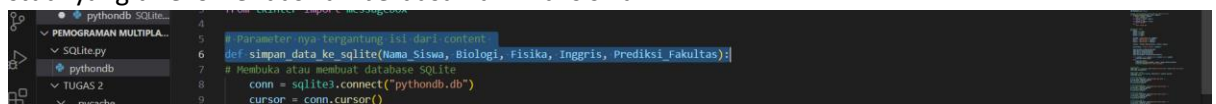
Nama_Siswa → Parameter berisi nama siswa atau identifikasi unik siswa.

Biologi → Parameter berisi nilai atau informasi terkait mata pelajaran Biologi.

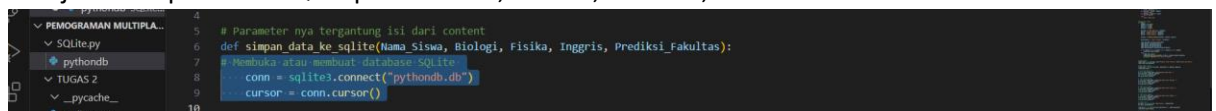
Fisika → Parameter berisi nilai atau informasi terkait mata pelajaran Fisika.

Inggris → Parameter berisi nilai atau informasi terkait mata pelajaran Bahasa Inggris.

Prediksi_Fakultas → Parameter berisi informasi atau prediksi terkait fakultas atau program studi yang direkomendasikan berdasarkan nilai siswa.



3. **sqlite3.connect("pythondb.db") →** Membuka koneksi ke database SQLite dengan nama file "pythondb.db". Jika file ini tidak ada, SQLite akan membuatnya. Fungsi connect dari modul sqlite3 mengembalikan objek koneksi (conn) yang merepresentasikan koneksi ke database. **conn.cursor() →** Membuat objek kursor dari objek koneksi (conn). Kursor digunakan untuk mengeksekusi pernyataan SQL pada database. Objek kursor (cursor) memungkinkan untuk menjalankan perintah SQL seperti SELECT, INSERT, UPDATE, dan DELETE.

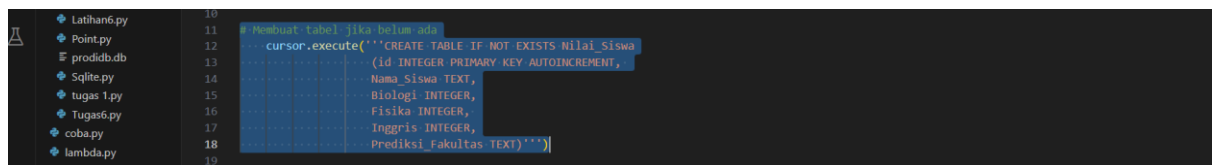


4. **CREATE TABLE IF NOT EXISTS Nilai_Siswa →** Membuat tabel dengan nama "Nilai_Siswa" jika tabel tersebut belum ada. Pernyataan ini memberikan opsi untuk membuat tabel hanya jika tabel tersebut belum ada dalam database. (id INTEGER PRIMARY KEY AUTOINCREMENT, Nama_Siswa TEXT, Biologi INTEGER, Fisika INTEGER, Inggris INTEGER, Prediksi_Fakultas TEXT) → Mendefinisikan kolom-kolom dalam tabel.

id INTEGER PRIMARY KEY AUTOINCREMENT → Kolom ID sebagai kunci utama (primary key) dengan tipe data INTEGER dan menggunakan penambahan otomatis (AUTOINCREMENT), yang berarti nilai ID akan bertambah secara otomatis untuk setiap baris baru yang ditambahkan.

Nama_Siswa TEXT → Kolom untuk menyimpan nama siswa dengan tipe data TEXT. Biologi INTEGER, Fisika INTEGER, Inggris INTEGER: Kolom-kolom untuk menyimpan nilai dalam mata pelajaran Biologi, Fisika, dan Bahasa Inggris, masing-masing dengan tipe data INTEGER.

Prediksi_Fakultas TEXT → Kolom untuk menyimpan prediksi fakultas atau program studi yang direkomendasikan dengan tipe data TEXT.



```

10
11 # Membuat tabel jika belum ada
12 cursor.execute("""CREATE TABLE IF NOT EXISTS Nilai_Siswa
13 (id INTEGER PRIMARY KEY AUTOINCREMENT,
14 Nama_Siswa TEXT,
15 Biologi INTEGER,
16 Fisika INTEGER,
17 Inggris INTEGER,
18 Prediksi_Fakultas TEXT)""")
19

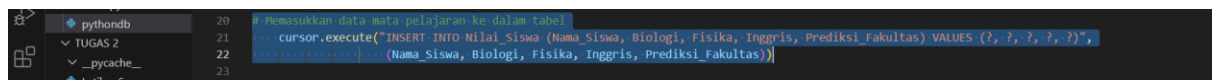
```

- INSERT INTO Nilai_Siswa** → Untuk memberitahu database bahwa akan menyisipkan (insert) data ke dalam tabel "Nilai_Siswa".

(Nama_Siswa, Biologi, Fisika, Inggris, Prediksi_Fakultas) → Daftar kolom yang akan diisi dengan nilai.

VALUES (?, ?, ?, ?, ?) → Tempat untuk memasukkan nilai-nilai ke dalam kolom-kolom yang sesuai. Tanda tanya (?) adalah parameter placeholder yang kemudian diisi dengan nilai-nilai yang diberikan dalam tuple pada bagian selanjutnya.

(Nama_Siswa, Biologi, Fisika, Inggris, Prediksi_Fakultas) → Tuple yang berisi nilai-nilai yang akan dimasukkan ke dalam tabel. Nilai-nilai ini sesuai dengan urutan kolom yang dijelaskan sebelumnya.



```

20
21 # Memasukkan data mata pelajaran ke dalam tabel
22 cursor.execute("INSERT INTO Nilai_Siswa (Nama_Siswa, Biologi, Fisika, Inggris, Prediksi_Fakultas) VALUES (?, ?, ?, ?, ?)",
23 (Nama_Siswa, Biologi, Fisika, Inggris, Prediksi_Fakultas))

```

- conn.commit()** → Perintah (commit) semua perubahan yang telah dilakukan dalam database. Jika melakukan operasi seperti penyisipan data (INSERT), pembaruan data (UPDATE), atau penghapusan data (DELETE), Anda perlu mengonfirmasi perubahan tersebut dengan menggunakan commit. Tanpa commit, perubahan tersebut tidak akan disimpan secara permanen dalam database.

conn.close() → Setelah mengonfirmasi perubahan dengan commit, perlu menutup koneksi ke database menggunakan perintah close(). untuk melepaskan sumber daya dan memastikan bahwa tidak ada koneksi yang tetap terbuka setelah selesai bekerja dengan database. Menutup koneksi juga memungkinkan file database untuk dibuka oleh aplikasi atau proses lain jika perlu.



```

23
24 # Melakukan commit dan menutup koneksi
25 conn.commit()
26 conn.close()
27

```

7. **top = tk.Tk()** → Membuat objek utama dari kelas Tk dalam modul tkinter. Objek ini merepresentasikan jendela utama atau top-level window dari aplikasi GUI .
- top.title("pythondb.db")** → Mengatur judul jendela aplikasi. Lalu, judulnya disetel menjadi "pythondb.db".
- top.geometry("500x500")** → Mengatur ukuran awal jendela aplikasi. Ukuran ini disetel menjadi lebar 500 piksel dan tinggi 500 piksel.
- top.resizable(False, False)** → Mencegah jendela aplikasi agar tidak dapat diubah ukurannya oleh pengguna. Dengan mengatur kedua parameter menjadi False, memastikan bahwa pengguna tidak dapat menyesuaikan lebar dan tinggi jendela.

```

27
28 # Buat Jendela Halaman
29 top = tk.Tk()
30 top.title("pythondb.db") # untuk memberi nama judul
31 top.geometry("500x500")
32 top.resizable(False, False)
33

```

8. **if Fisika < Biologi > Inggris:** → Ini adalah kondisi pertama. Jika nilai Biologi lebih tinggi dari Fisika dan Inggris, maka fakultas yang direkomendasikan adalah "Kedokteran".
- elif Biologi < Fisika > Inggris:** → Ini adalah kondisi kedua. Jika nilai Fisika lebih tinggi dari Biologi dan Inggris, maka fakultas yang direkomendasikan adalah "Teknik".
- elif Biologi < Inggris > Fisika:** → Ini adalah kondisi ketiga. Jika nilai Inggris lebih tinggi dari Biologi dan Fisika, maka fakultas yang direkomendasikan adalah "Bahasa".
- Else:** → Ini adalah kondisi terakhir. Jika tidak ada kondisi sebelumnya yang terpenuhi, maka fungsi akan mengembalikan string "Belum ada", yang mungkin menunjukkan bahwa tidak ada rekomendasi fakultas yang dapat diberikan.

```

34 # Fungsi untuk menentukan prediksi fakultas berdasarkan nilai
35 def Prediksi_Fakultas(Biologi, Fisika, Inggris):
36     if Fisika < Biologi > Inggris:
37         return "Kedokteran"
38     elif Biologi < Fisika > Inggris:
39         return "Teknik"
40     elif Biologi < Inggris > Fisika:
41         return "Bahasa"
42     else:
43         return "Belum ada"
44

```

9. - Menggunakan fungsi Prediksi_Fakultas untuk mendapatkan hasil prediksi
- Mengupdate label-label pada antarmuka pengguna dengan hasil informasi dan prediksi
- Jika tidak ada data yang dimasukkan, maka frame_hasil tidak ditampilkan
- Jika ada data yang dimasukkan, maka frame_hasil ditampilkan
- Menyimpan data ke dalam database SQLite
- Menampilkan pesan informasi bahwa data tersimpan

```

45 # Menampilkan fungsi
46 def show():
47     namaMhs = h1.get()
48     matkul1 = h1.get()
49     matkul2 = f1.get()
50     matkul3 = f1.get()
51
52     hasilMhs = "Nama Mahasiswa: {namaMhs}"
53     hasil1 = "Mata Kuliah 1: {matkul1}"
54     hasil2 = "Mata Kuliah 2: {matkul2}"
55     hasil3 = "Mata Kuliah 3: {matkul3}"
56
57     # Menggunakan fungsi prediksi_fakultas untuk mendapatkan hasil prediksi
58     prediksi = Prediksi_Fakultas(matkul1, matkul2, matkul3)
59
60     hasilPrediksi = "Hasil Prediksi: {prediksi}"
61     # Mengupdate label-label pada antarmuka pengguna dengan hasil informasi dan prediksi
62     label_hasilMhs.config(text=hasilMhs)
63     label_hasil1.config(text=hasil1)
64     label_hasil2.config(text=hasil2)
65     label_hasil3.config(text=hasil3)
66     label_hasilprediksi.config(text=hasilPrediksi)
67
68     # Jika tidak ada data yang dimasukkan, maka frame_hasil tidak ditampilkan
69     if not matkul1 and not matkul2 and not matkul3 and not namaMhs:
70         frame_hasil.pack_forget()
71     else:
72         frame_hasil.pack() # Jika ada data yang dimasukkan maka frame_hasil ditampilkan
73         simpan_data_ke_sqlite(namaMhs, matkul1, matkul2, matkul3, prediksi) # Menyimpan data ke dalam database SQLite
74         messagebox.showinfo("Info", "Data Tersimpan") # Menampilkan pesan informasi bahwa data tersimpan
75

```

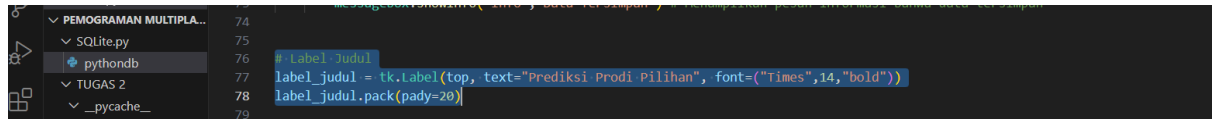
10. **tk.Label** → digunakan untuk membuat label.

Top → yang mewakili jendela utama aplikasi tkinter.

text="Prediksi Prodi Pilihan" → menetapkan teks label sebagai "Prediksi Prodi Pilihan".

font=("Times", 14, "bold") → menetapkan font label sebagai Times dengan ukuran 14 dan tebal.

label_judul.pack(pady=20) → mengemas (pack) label ke dalam tata letak (layout) dan memberikan jarak pola (padding) sebesar 20 piksel di sekitar label.



```

74
75
76 # Label Judul
77 label_judul = tk.Label(top, text="Prediksi Prodi Pilihan", font=("Times",14,"bold"))
78 label_judul.pack(pady=20)
79

```

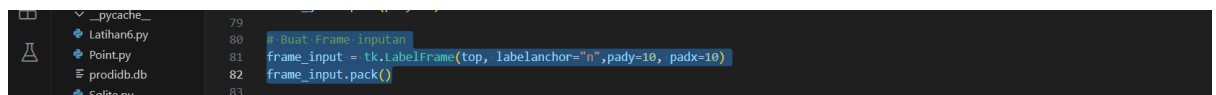
11. **tk.LabelFrame** → Digunakan untuk membuat kerangka (frame) yang dapat memiliki label.

top: merupakan objek dari kelas Tk, yang mewakili jendela utama aplikasi tkinter.

labelanchor="n" → Menetapkan anchor (titik tumpuan) untuk label di dalam kerangka. "n" mengindikasikan utara (north), yang berarti label akan ditempatkan di bagian atas kerangka.

pady=10, padx=10 → Menetapkan padding (jarak pola) di sekitar kerangka. pady adalah padding di sepanjang sumbu vertikal, dan padx adalah padding di sepanjang sumbu horizontal. Dalam hal ini, diberikan padding sebesar 10 piksel di setiap sisi.

frame_input.pack() → Mengemas (pack) kerangka ke dalam tata letak (layout). Bahwa kerangka akan ditampilkan di jendela aplikasi tkinter.



```

79
80
81 # Buat frame inputan
82 frame_input = tk.LabelFrame(top, labelanchor="n",pady=10, padx=10)
83 frame_input.pack()
84

```

12. **tk.Label** → Membuat objek label untuk menampilkan teks atau informasi di antarmuka pengguna.

frame_input → Menunjukkan bahwa label akan ditempatkan di dalam kerangka yang sebelumnya telah dibuat (dalam contoh sebelumnya, frame_input).

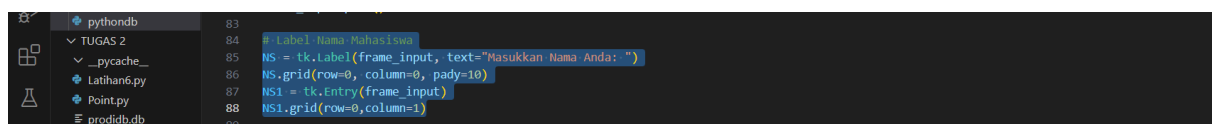
text="Masukkan Nama Anda: " → Menetapkan teks label sebagai "Masukkan Nama Anda: ".

grid(row=0, column=0, pady=10) → Menempatkan label di dalam grid (kisi) yang terletak di baris 0 dan kolom 0 dari kerangka (frame_input). pady=10 memberikan jarak pola (padding) sebesar 10 piksel di sepanjang sumbu vertikal.

tk.Entry → Membuat objek input yang memungkinkan pengguna untuk memasukkan teks atau data.

frame_input → Menunjukkan bahwa kotak masukan (Entry) akan ditempatkan di dalam kerangka yang sama (frame_input).

grid(row=0, column=1) → Menempatkan kotak masukan di dalam grid di baris 0 dan kolom 1 dari kerangka (frame_input).



```

83
84 # Label Nama Mahasiswa
85 NS = tk.Label(frame_input, text="Masukkan Nama Anda: ")
86 NS.grid(row=0, column=0, pady=10)
87 NS1 = tk.Entry(frame_input)
88 NS1.grid(row=0, column=1)
89

```

13. **B = tk.Label(frame_input, text="Nama Mata kuliah Biologi: ")** → Membuat label dengan teks "Nama Mata kuliah Biologi:" dan menetapkan ke variabel B. Label ini nantinya akan digunakan untuk menunjukkan instruksi atau informasi di antarmuka pengguna.
- B.grid(row=1, column=0, pady=10)** → Menempatkan label B di dalam grid (kisi) di dalam objek frame yang disebut frame_input. row=1 menunjukkan bahwa label tersebut akan ditempatkan pada baris kedua, column=0 menunjukkan kolom pertama, dan pady=10 adalah jarak (padding) di sekitar label pada sumbu Y (vertikal).
- B1 = tk.Entry(frame_input)** → Membuat objek Entry (input) dan menetapkan ke variabel B1. Entry ini nantinya akan digunakan untuk pengguna memasukkan data atau teks.
- B1.grid(row=1, column=1)** → Menempatkan objek Entry B1 di dalam grid di dalam frame_input. row=1 dan column=1 menunjukkan bahwa Entry tersebut akan ditempatkan pada baris kedua dan kolom kedua.

```

90 #Label Nilai Biologi
91 B = tk.Label(frame_input, text="Nama Mata kuliah Biologi: ")
92 B.grid(row=1, column=0, pady=10)
93 B1 = tk.Entry(frame_input)
94 B1.grid(row=1, column=1)
95

```

14. **F = tk.Label(frame_input, text="Nama Mata kuliah Fisika: ")** → Membuat label dengan teks "Nama Mata kuliah Fisika:" dan menetapkan ke variabel F. Label ini akan digunakan untuk memberikan instruksi atau informasi kepada pengguna terkait dengan pengisian nama mata kuliah Fisika.
- F.grid(row=2, column=0, pady=10)** → Menempatkan label F di dalam grid pada objek frame frame_input. Dengan row=2 dan column=0, label ini akan ditempatkan pada baris ketiga dan kolom pertama dalam grid frame_input. pady=10 menunjukkan jarak (padding) di sekitar label pada sumbu Y (vertikal).
- F1 = tk.Entry(frame_input)** → Membuat objek Entry (input) dan menetapkan ke variabel F1. Entry ini digunakan untuk memungkinkan pengguna memasukkan data atau teks dengan mata kuliah Fisika.
- F1.grid(row=2, column=1)** → Menempatkan objek Entry F1 di dalam grid pada objek frame frame_input. Dengan row=2 dan column=1, Entry ini akan ditempatkan pada baris ketiga dan kolom kedua dalam grid frame_input.

```

95 #Label Nilai Fisika
96 F = tk.Label(frame_input, text="Nama Mata kuliah Fisika: ")
97 F.grid(row=2, column=0, pady=10)
98 F1 = tk.Entry(frame_input)
99 F1.grid(row=2, column=1)
100
101

```

15. **I = tk.Label(frame_input, text="Nama Mata kuliah Inggris: ")** → Membuat label dengan teks "Nama Mata kuliah Inggris:" dan menetapkan ke variabel I. Label ini akan berfungsi sebagai instruksi atau informasi kepada pengguna tentang apa yang diharapkan dalam pengisian nama mata kuliah Inggris.
- I.grid(row=3, column=0, pady=10)** → Menempatkan label I di dalam grid pada objek frame frame_input. Dengan row=3 dan column=0, label ini akan ditempatkan pada baris keempat dan kolom pertama dalam grid frame_input. pady=10 menunjukkan jarak (padding) di sekitar label pada sumbu Y (vertikal).
- I1 = tk.Entry(frame_input)** → Membuat objek Entry (input) dan menetapkan ke variabel I1. Entry ini memberikan tempat bagi pengguna untuk memasukkan data atau teks terkait dengan mata kuliah Inggris.

I1.grid(row=3, column=1) → Menempatkan objek Entry I1 di dalam grid pada objek frame frame_input. Dengan row=3 dan column=1, Entry ini akan ditempatkan pada baris keempat dan kolom kedua dalam grid frame_input.

```

102 #Label Hasil Inggris
103 I = tk.Label(frame_input, text="Nama Mata kuliah Inggris: ")
104 I.grid(row=3, column=0, pady=10)
105 I1 = tk.Entry(frame_input)
106 I1.grid(row=3, column=1)
107

```

16. **btn_hasil = tk.Button(top, text="Submit", command=show) →** Membuat objek tombol (Button) dengan teks "Submit" dan menentukannya ke variabel btn_hasil. Fungsi show ditetapkan sebagai fungsi yang akan dijalankan saat tombol ditekan. Artinya, ketika tombol ini ditekan, fungsi show akan dijalankan.

btn_hasil.pack(pady=10) → Menempatkan tombol btn_hasil di dalam antarmuka pengguna (widget utama) dengan menggunakan metode pack. pady=10 menunjukkan jarak (padding) di atas dan di bawah tombol pada sumbu Y (vertikal).

```

107
108
109 btn_hasil = tk.Button(top, text="Submit", command=show)
110 btn_hasil.pack(pady=10)
111

```

17. **frame_hasil = tk.LabelFrame(top, labelanchor="n", padx=10, pady=10) →** Membuat objek LabelFrame dengan induknya sebagai objek top (jendela utama atau frame utama). labelanchor="n" menunjukkan bahwa label di LabelFrame akan diatur di bagian utara (north). padx=10 dan pady=10 adalah jarak antara tepi frame dan elemen-elemen di dalamnya pada sumbu X dan Y.

frame_hasil.pack_forget() → Metode pack_forget() digunakan untuk menyembunyikan atau menghilangkan objek frame_hasil dari tampilan. Untuk menyembunyikan frame pada awalnya dan kemudian menampilkannya, misalnya, sebagai respons terhadap suatu peristiwa seperti menekan tombol.

```

111
112 frame_hasil = tk.LabelFrame(top, labelanchor="n", padx=10, pady=10)
113 frame_hasil.pack_forget()
114

```

18. **label_hasilMhs, label_hasil1, label_hasil2, label_hasil3 dan label_hasilprediksi = tk.Label(frame_hasil, text="") →** Membuat objek label (Label) dengan induknya sebagai objek frame_hasil. Label ini awalnya memiliki teks kosong (text=""), tetapi dapat mengubah atau mengatur teksnya sesuai dengan kebutuhan.

label_hasilMhs.pack(), label_hasil1.pack(), label_hasil2.pack(), label_hasil3.pack() dan label_hasilprediksi.pack() → Metode pack() digunakan untuk menempatkan label (label_hasilMhs) di dalam frame (frame_hasil). Tanpa argumen tambahan, label akan ditempatkan di posisi default di dalam frame tersebut.

```

115 #Label Hasil
116 label_hasilMhs = tk.Label(frame_hasil, text="")
117 label_hasilMhs.pack()
118
119 label_hasil1 = tk.Label(frame_hasil, text="")
120 label_hasil1.pack()
121
122 label_hasil2 = tk.Label(frame_hasil, text="")
123 label_hasil2.pack()
124
125 label_hasil3 = tk.Label(frame_hasil, text="")
126 label_hasil3.pack()
127
128 label_hasilprediksi = tk.Label(frame_hasil, text="")
129 label_hasilprediksi.pack()
130

```

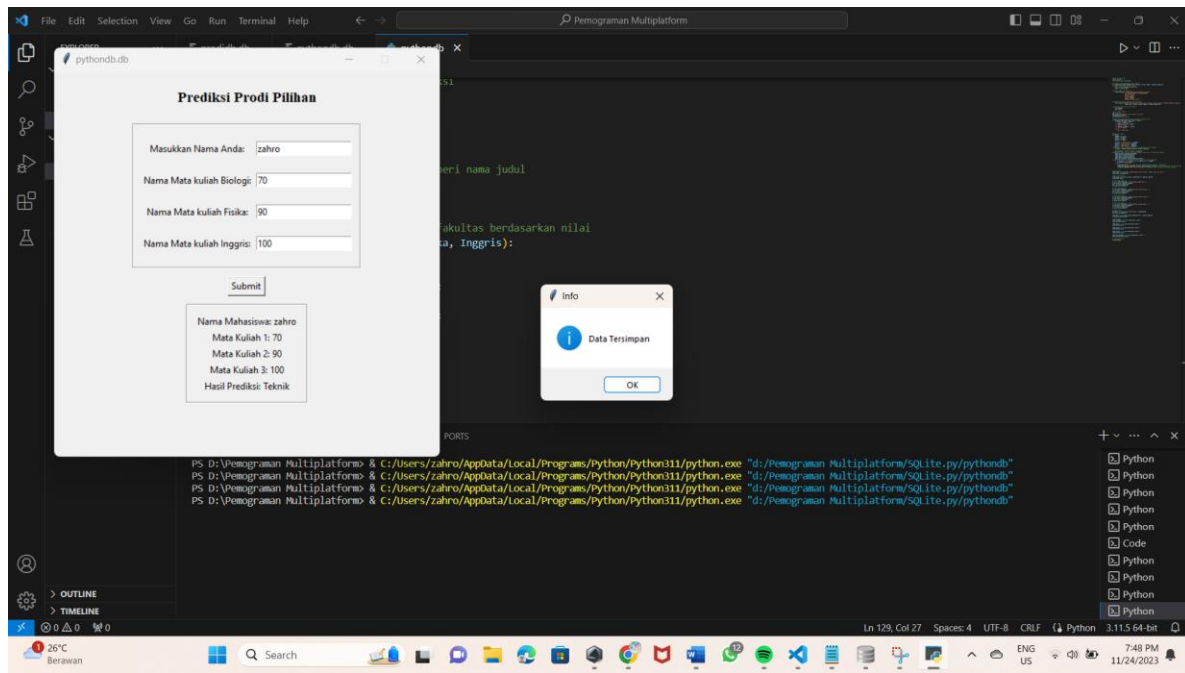
19. **top.mainloop()** → Untuk menahan eksekusi program dan terus berjalan hingga pengguna menutup jendela aplikasi atau program secara eksplisit dihentikan.

```

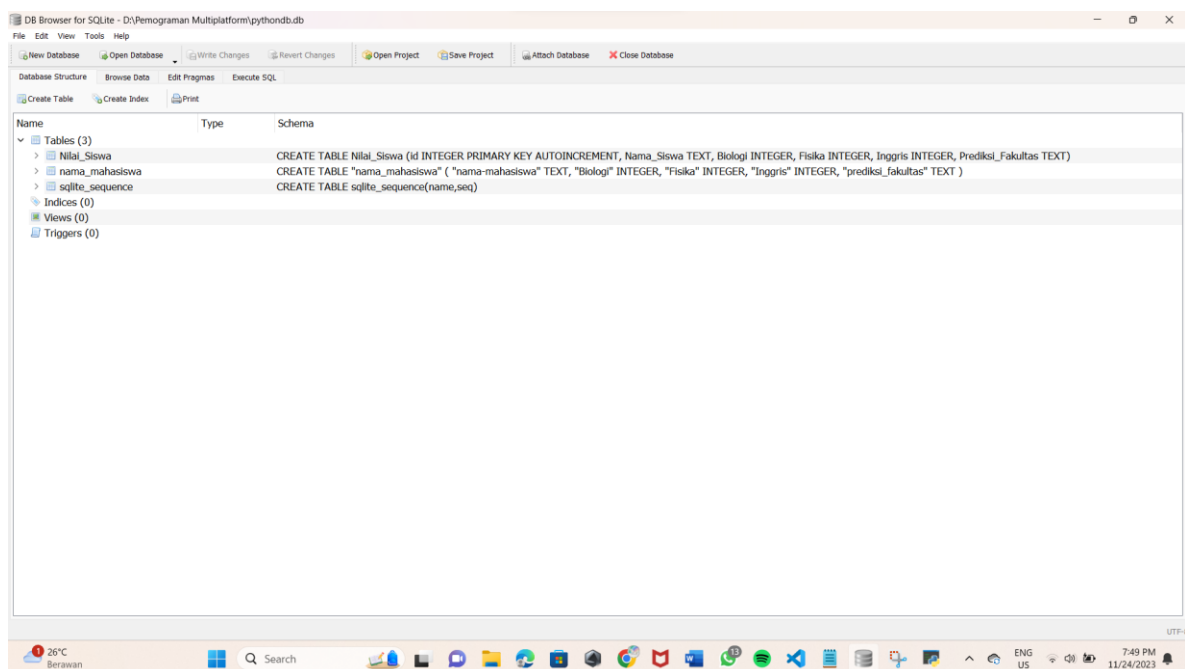
131 # Jalankan Aplikasi
132 top.mainloop()

```

HASIL TKINTER



DATA STRUCTURE



EXECUTE SQL

The screenshot shows the DB Browser for SQLite application window. The title bar reads "DB Browser for SQLite - D:\Pemrograman Multiplatform\python\python.db". The menu bar includes File, Edit, View, Tools, and Help. The toolbar contains buttons for New Database, Open Database, Write Changes, Revert Changes, Open Project, Save Project, Attach Database, and Close Database. The main interface has tabs for Database Structure, Browse Data, Edit Pragma, and Execute SQL. The Execute SQL tab is active, showing a SQL query: `SELECT * FROM Nilai_Siswa`. Below the query editor, a table displays the results of the query. The table has five columns: Id, Nama_Siswa, Biologi, Fisika, and Prediksi_Fakultas. It contains five rows of data. Below the table, a status message indicates: "Execution finished without errors. Result: 13 rows returned in 9ms. At line 1: SELECT * FROM Nilai_Siswa". The Windows taskbar at the bottom shows the system clock as 7:52 PM on 11/24/2023, along with various system icons and a search bar.

Id	Nama_Siswa	Biologi	Fisika	Prediksi_Fakultas
9	nita	10	80	20 Teknik
10	hafiz	50	40	100 Kedokteran
11	linda	40	100	80 Bahasa
12	shevi	30	100	80 Bahasa
13	zahro	70	90	100 Teknik

Execution finished without errors.
Result: 13 rows returned in 9ms
At line 1:
SELECT * FROM Nilai_Siswa