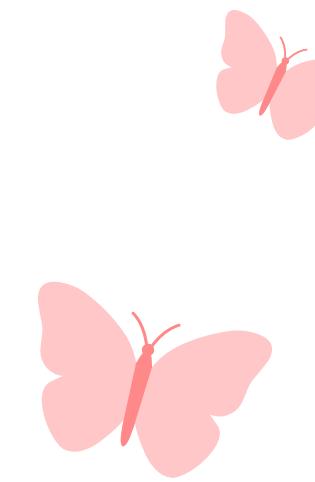


OPTIMISASI INFORMASI BIAYA PENDIDIKAN DAN PROSPEK KERJA TIAP PRODI BAGI CALON MAHASISWA UI

Database untuk Sains Data (B)

Kelompok 7

Anggota



Naufal Elban Musyaffa L. (2206053865)

Kayla Zahira Amadya (2206053890)

Meuthia Nabila Jauhari (2206820434)

Najwa Putri Faradila (2206051355)

Soraya Indira Putri D. (2206053902)



OVERVIEW

01

TABEL RELASIONAL

02

IMPLEMENTASI SQL

03

IMPLEMENTASI GUI

04

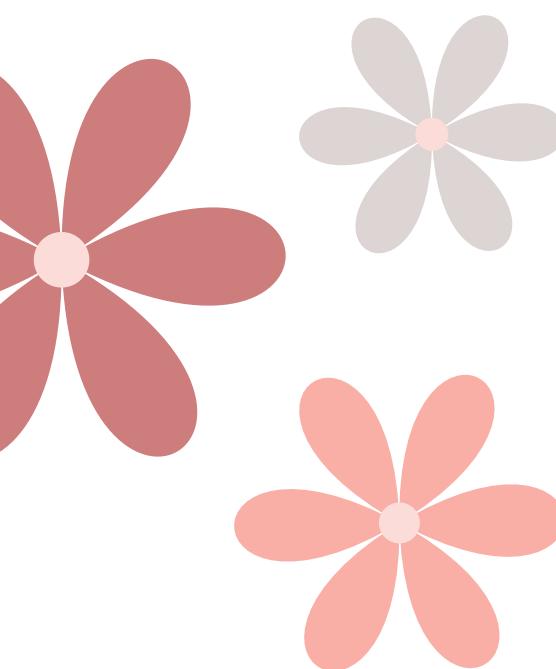
TESTING

05

KESIMPULAN

06

TAUTAN LAMPIRAN



TABEL RELASIONAL

Tabel Program pendidikan

Tabel ini berisikan informasi mengenai program-program pendidikan yang dapat diambil oleh calon mahasiswa Universitas Indonesia. Contohnya seperti program pendidikan D3, D4, dan S1. Tabel Program Pendidikan memiliki 2 degree dan 3 kardinalitas.

Nama Field	Tipe Data	Domain	ket
ID_Program	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID program pendidikan {J001, J002, J003}	PK
Jenjang	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepesentasikan jenjang pendidikan {S1, D3, D4}	



Tabel Program pendidikan

Berikut adalah tabel program pendidikan yang telah kami buat pada **Google Sheets**,

ID_Program	Jenjang
J001	S1
J002	D3
J003	D4



Tabel Rumpun

Berisi informasi mengenai rumpun pendidikan di Universitas Indonesia seperti Ilmu Kesehatan, Sains dan Teknologi, Ilmu Sosial dan Humaniora, dan Vokasi. Tabel ini terdiri dari 2 degree dan 4 kardinalitas.

Nama Field	Tipe Data	Domain	Ket
ID_Rumpun	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID rumpun yang tersedia {R001, R002, ...}	PK
Nama_Rumpun	CHAR/TEXT	Terdiri dari kumpulan karakter string yang merepresentasikan nama rumpun	

Tabel Rumpun

Berikut adalah tabel rumpun yang telah kami buat pada **Google Sheets**

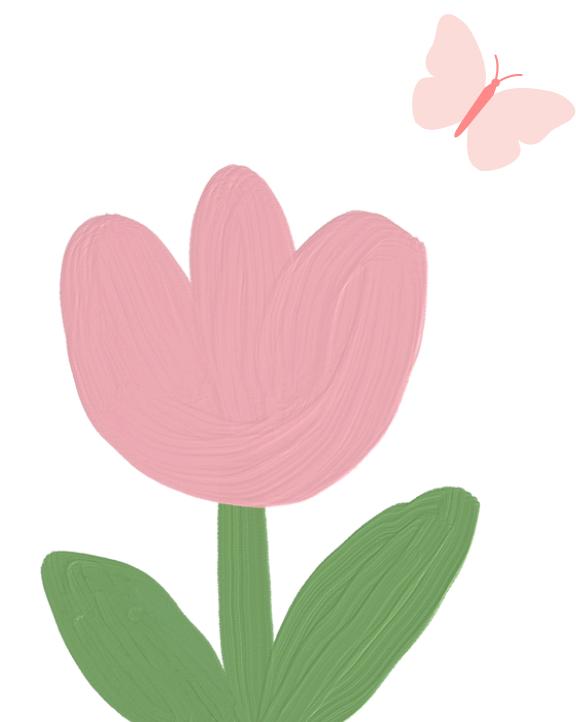
ID_Rumpun	Nama_Rumpun
R001	Ilmu Kesehatan
R002	Sains dan Teknologi
R003	Ilmu Sosial dan Humaniora
R004	Vokasi



Nama Field	Tipe Data	Domain	Ket
ID_Biaya	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID biaya pendidikan {B00101, B00102, ...}	PK
Golongan	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan golongan dari biaya pendidikan	
Harga	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan rentang harga dari suatu biaya pendidikan persemester	
ID_Rumpun	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID rumpun yang tersedia {R001, R002, ...}	FK

Tabel Biaya Pendidikan

Tabel ini berisikan informasi mengenai golongan biaya pendidikan yang harus dibayar oleh mahasiswa persemester. Golongan biaya terdiri dari 11 golongan dan besar biaya dipengaruhi oleh rumpun pendidikan. Tabel ini terdiri dari 4 degree dan 44 kardinalitas.



Tabel Biaya Pendidikan

Berikut adalah 'head' tabel biaya pendidikan yang telah kami buat pada Google Sheets

ID_Biaya	ID_Rumpun	Golongan	Harga
B00101	R001	K-1	0-500.000
B00102	R001	K-2	500.000-1.000.000
B00103	R001	K-3	1.000.000-2.000.000
B00104	R001	K-4	2.000.000-4.000.000
B00105	R001	K-5	4.000.000-6.000.000
B00106	R001	K-6	6.000.000-7.500.000
B00107	R001	K-7	7.500.000-10.000.000
B00108	R001	K-8	10.000.000-12.500.000
B00109	R001	K-9	12.500.000-15.000.000
B00110	R001	K-10	15.000.000-17.500.000
B00111	R001	K-11	17.500.000-20.000.000
B00201	R002	K-1	0-500.000
B00202	R002	K-2	500.000-1.000.000

Nama Field	Tipe Data	Domain	Ket.
ID_Fakultas	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID sebuah fakultas {F001, F002, ...}	PK
Nama_Fakultas	CHAR/TEXT	Terdiri dari kumpulan karakter string yang merepresentasikan nama fakultas	
Biaya_Golongan Terendah	CHAR/TEXT	Terdiri dari kumpulan karakter merepresentasikan biaya tertinggi dalam sebuah fakultas	
Biaya_Golongan Tertinggi	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID rumpun yang tersedia {R001, R002, ...}	
ID_Rumpun	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID rumpun yang tersedia {R001, R002, ...}	FK

Tabel Fakultas

Tabel ini berisikan informasi mengenai fakultas-fakultas yang tersedia di Universitas Indonesia seperti Fakultas Teknik, Kedokteran, Kedokteran Gigi, Matematika dan Ilmu Pengetahuan Alam, Hukum, Psikologi, Ilmu Budaya, Ilmu Sosial dan Politik, dan sebagainya. Tabel ini terdiri dari 5 degree dan 15 kardinalitas.

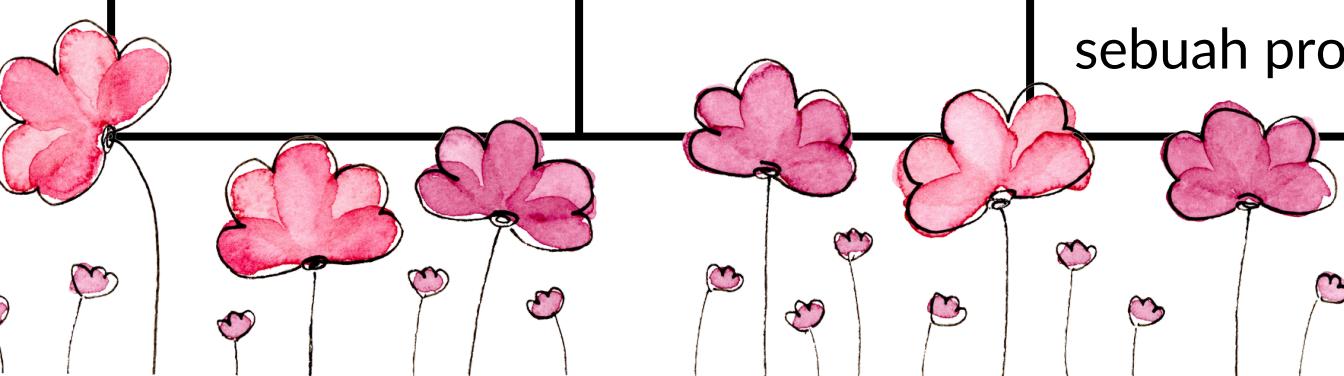
Tabel Fakultas

Berikut adalah tabel fakultas yang telah kami buat pada **Google Sheets**,

ID_Fakultas	Nama_Fakultas	Biaya_Golongan_Terendah	Biaya_Golongan_Tertinggi	ID_Rumpun
F001	Kedokteran	Rp0-500.000	Rp17.500.000-20.000.000	R001
F002	Kedokteran Gigi	Rp0-500.000	Rp17.500.000-20.000.000	R001
F003	Ilmu Keperawatan	Rp0-500.000	Rp17.500.000-20.000.000	R001
F004	Farmasi	Rp0-500.000	Rp17.500.000-20.000.000	R001
F005	Kesehatan Masyarakat	Rp0-500.000	Rp17.500.000-20.000.000	R001
F006	Matematika dan Ilmu Pengetahuan Alam	Rp0-500.000	Rp17.500.000-20.000.000	R002
F007	Teknik	Rp0-500.000	Rp17.500.000-20.000.000	R002
F008	Ilmu Komputer	Rp0-500.000	Rp17.500.000-20.000.000	R002
F009	Hukum	Rp0-500.000	Rp15.000.000-17.500.000	R003
F010	Ekonomi dan Bisnis	Rp0-500.000	Rp15.000.000-17.500.000	R003
F011	Ilmu Budaya	Rp0-500.000	Rp15.000.000-17.500.000	R003
F012	Psikologi	Rp0-500.000	Rp15.000.000-17.500.000	R003
F013	Ilmu Sosial dan Ilmu Politik	Rp0-500.000	Rp15.000.000-17.500.000	R003
F014	Ilmu Administrasi	Rp0-500.000	Rp15.000.000-17.500.000	R003
F015	Vokasi	Rp0-500.000	Rp17.500.000-20.000.000	R004

Tabel Program Studi

Nama Field	Tipe Data	Domain	Ket.	Nama Field	Tipe Data	Domain	Ket
ID_Prodi	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID sebuah program studi {P001, P002, ...}	PK	Daya Tampung	INTEGER	Terdiri dari kumpulan integer postif yang merepresentasikan jumlah daya tampung sebuah program studi	
Nama_Prodi	CHAR/TEXT	Terdiri dari kumpulan karakter string yang merepresentasikan nama program studi		ID_Fakultas	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID sebuah fakultas {F001, F002, ...}	FK
Akreditasi	CHAR/TEXT	Terdiri dari kumpulan karakter string yang merepresentasikan akreditasi nasional sebuah program studi		ID_Program	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID program pendidikan {J001, J002, J003}	FK



Tabel Program Studi

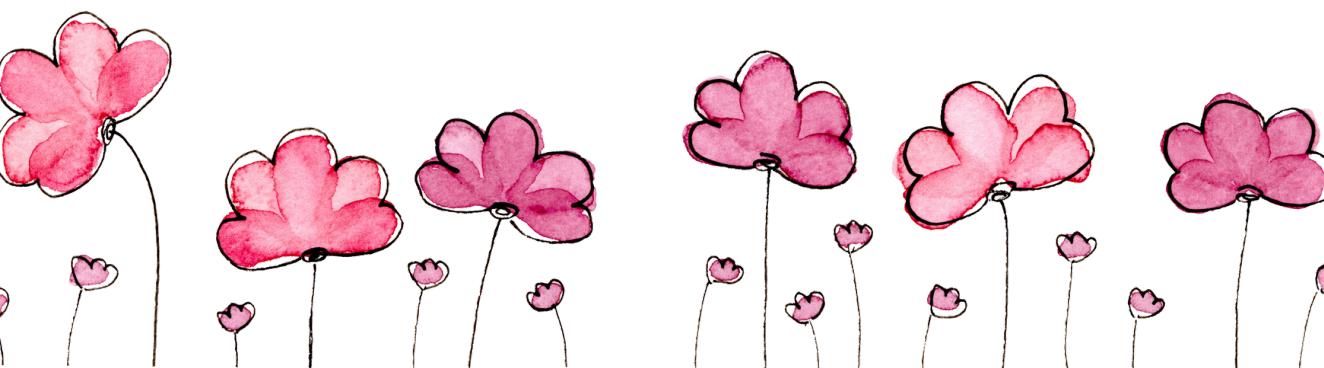
Tabel ini berisikan informasi mengenai program studi yang ditawarkan untuk calon mahasiswa baru oleh Universitas Indonesia. Contohnya seperti Pendidikan Dokter, Gizi, Farmasi, Ilmu Aktuaria, Sistem Informasi, Teknik Industri, Psikologi, Ilmu Politik, Akutansi, Administrasi Rumah Sakit dan masih banyak lagi. Tabel ini terdiri dari 6 degree dan 79 kardinalitas (terdapat 79 program studi yang ditawarkan). Berikut adalah 'head' dari tabel Program Studi :

ID_Prodi	Nama_Prodi	Akreditasi	Daya_tampung	ID_Fakultas	ID_Program
P001	Pendidikan Dokter	A	183	F001	J001
P002	Kedokteran Gigi	A	111	F002	J001
P003	Ilmu Keperawatan	A	140	F003	J001
P004	Farmasi	Unggul	90	F004	J001
P005	Gizi	Unggul	58	F005	J001
P006	Kesehatan Lingkungan	Unggul	30	F005	J001
P007	Kesehatan Masyarakat	Unggul	180	F005	J001
P008	Keselamatan dan Kesehatan Kerja	Unggul	30	F005	J001
P009	Biologi	Unggul	77	F006	J001
P010	Fisika	Unggul	77	F006	J001
P011	Geofisika	Baik Sekali	47	F006	J001
P012	Geografi	Unggul	77	F006	J001
P013	Geologi	Baik Sekali	47	F006	J001
P014	Ilmu Aktuaria	Unggul	47	F006	J001
P015	Kimia	Unggul	77	F006	J001

Nama Field	Tipe Data	Domain	Ket.
ID_Prospek	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID suatu prospek kerja {JP0001, JP0002, ...}	PK
Nama_Prospek	CHAR/TEXT	Terdiri dari kumpulan karakter string yang merepresentasikan nama prospek kerja.	
Gaji_per_bulan	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan rentang gaji perbulan suatu prospek	
ID_Prodi	CHAR/TEXT	Terdiri dari kumpulan karakter yang merepresentasikan ID sebuah program studi {P001, P002, ...}	FK

Tabel Prospek Kerja

Tabel ini berisikan informasi mengenai prospek kerja yang mungkin dari program-program studi yang ditawarkan. Tabel ini terdiri dari 4 degree dan 158 kardinalitas.

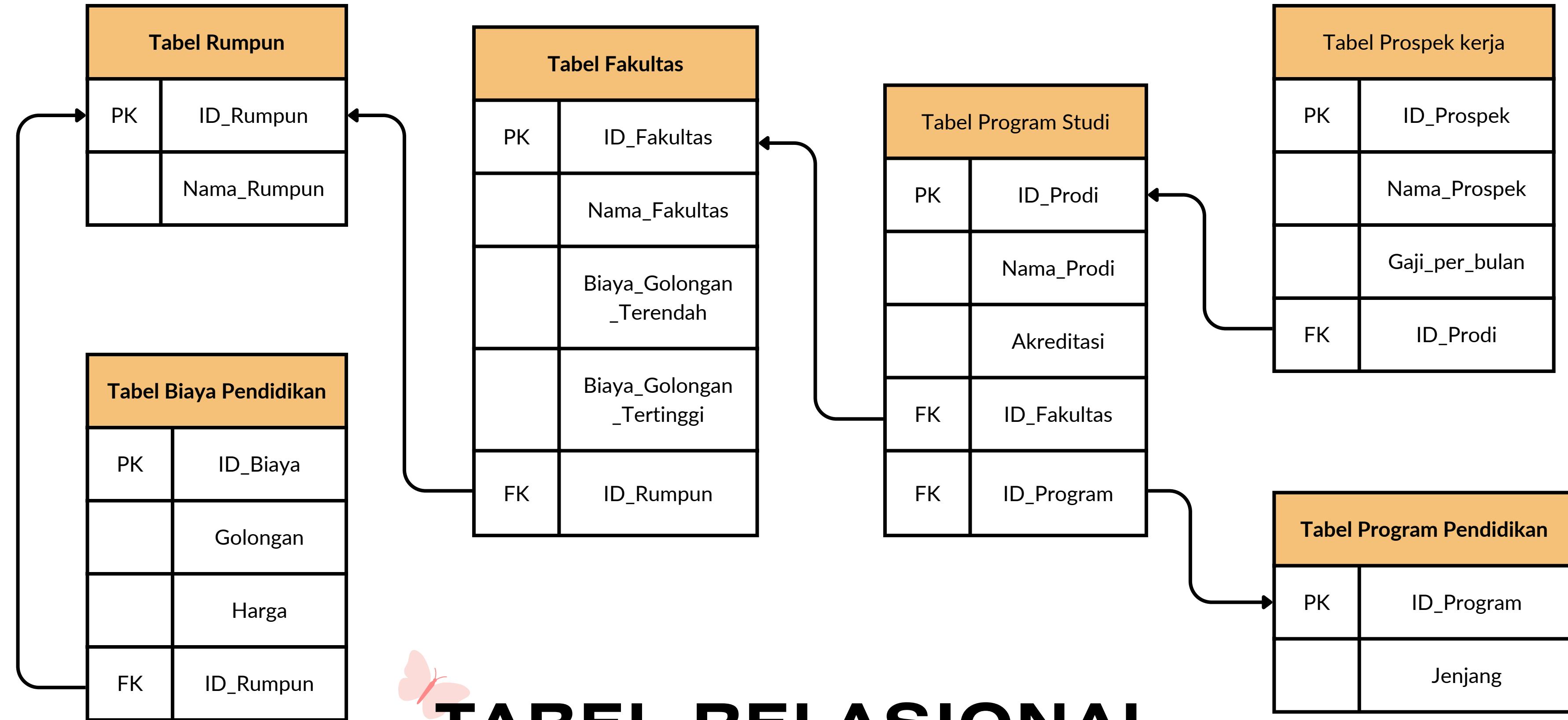


Tabel Prospek Kerja

Berikut adalah head tabel prospek kerja yang telah kami buat pada **Google Sheets** ,

ID_Prodi	ID_Prospek	Nama_Prospek	Gaji_per_bulan
P001	JP0001	Dokter umum	Rp5 juta - Rp20 juta
P002	JP0002	Dokter gigi Umum	Rp4 juta - Rp15 juta
P003	JP0003	Perawat	Rp3 juta - Rp10 juta
P004	JP0004	Apoteker	Rp5 juta - Rp15 juta
P005	JP0005	Ahli gizi	Rp4 juta - Rp10 juta
P006	JP0006	Pengawas kesehatan lingkungan	Rp4 juta - Rp10 juta
P007	JP0007	Epidemiolog	Rp7 juta - Rp17 juta
P008	JP0008	Pengawas K3	Rp5 juta - Rp12 juta
P009	JP0009	Peneliti biologi	Rp4 juta - Rp12 juta
P010	JP0010	Peneliti fisika	Rp4 juta - Rp12 juta
P011	JP0011	Geofisikawan	Rp7 juta - Rp20 juta
P012	JP0012	Analisis geografis	Rp5 juta - Rp15 juta
P013	JP0013	Geolog pertambangan	Rp10 juta - Rp30 juta
P014	JP0014	Aktuaris	Rp8 juta - Rp25 juta
P015	JP0015	Peniliti kimia	Rp5 juta - Rp15 juta
P016	JP0016	Data scientist	Rp10 juta - Rp30 juta
P017	JP0017	Analisis data	Rp7 juta - Rp20 juta

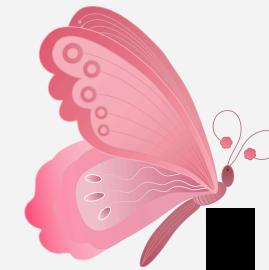


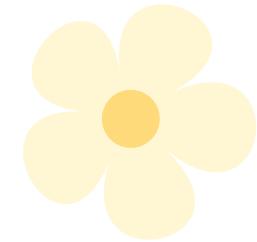


TABEL RELASIONAL



IMPLEMENTASI SQL





IMPLEMENTASI SQL

- Import package

```
import sqlite3  
import pandas as pd
```

- Membuat database

```
def create_db_and_tables():  
    conn = sqlite3.connect('C:/Users/najwa/Downloads/Database_data/Database UI.db')  
    cursor = conn.cursor()
```

- Membuat tabel rumpun

```
CREATE TABLE IF NOT EXISTS Rumpun (  
    ID_Rumpun TEXT PRIMARY KEY,  
    Nama_Rumpun TEXT);
```



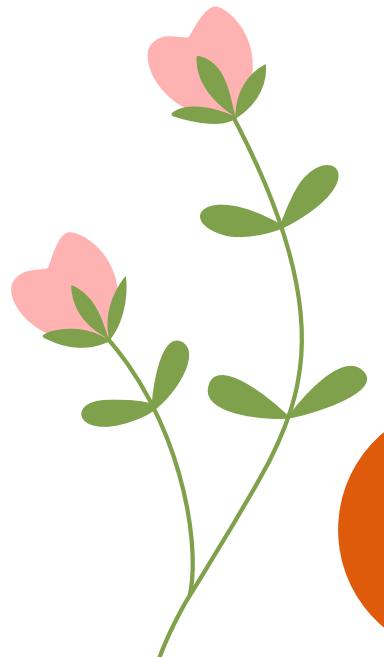
IMPLEMENTASI SQL

- Membuat tabel program studi

```
CREATE TABLE IF NOT EXISTS ProgramStudi (
    ID_Prodi TEXT PRIMARY KEY,
    Nam_Prodi TEXT,
    Akreditasi TEXT,
    Daya_Tampung INTEGER,
    ID_Fakultas TEXT,
    ID_Program TEXT,
    FOREIGN KEY (ID_Fakultas) REFERENCES Fakultas(ID_Fakultas),
    FOREIGN KEY (ID_Program) REFERENCES Program(ID_Program));
```

- Membuat tabel biaya pendidikan

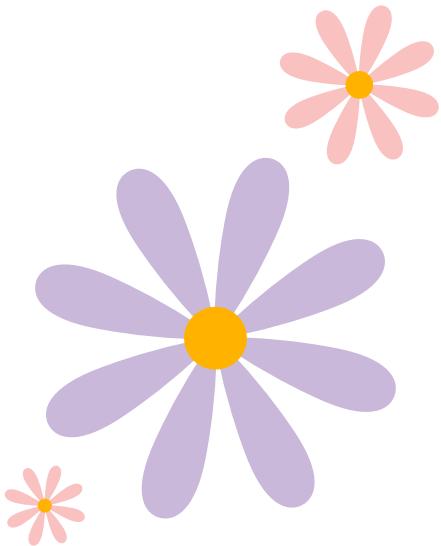
```
CREATE TABLE IF NOT EXISTS BiayaPendidikan (
    ID_Biaya TEXT PRIMARY KEY,
    Golongan TEXT,
    Harga TEXT,
    ID_Rumpun,
    FOREIGN KEY (ID_Rumpun) REFERENCES Rumpun(ID_Rumpun));
```



IMPLEMENTASI SQL

- Membuat tabel prospek kerja

```
CREATE TABLE IF NOT EXISTS ProspekKerja (
    ID_Prospek TEXT PRIMARY KEY,
    Nama_Prospek TEXT,
    Gaji_per_bulan TEXT,
    ID_Prodi,
    FOREIGN KEY (ID_Prodi) REFERENCES ProgramStudi(ID_Prodi));
```



- Membuat tabel program pendidikan

```
CREATE TABLE IF NOT EXISTS ProgramPendidikan (
    ID_Program TEXT PRIMARY KEY,
    Jenjang TEXT);
```

- Membuat tabel fakultas

```
CREATE TABLE IF NOT EXISTS Fakultas (
    ID_Fakultas TEXT PRIMARY KEY,
    Nama_Fakultas TEXT,
    Biaya_Golongan_Terendah TEXT,
    Biaya_Golongan_Tertinggi TEXT,
    ID_Rumpun,
    FOREIGN KEY (ID_Rumpun) REFERENCES Rumpun(ID_Rumpun));
```

IMPLEMENTASI SQL

- Memasukkan data ke dalam tabel

```
def load_data():
    conn = sqlite3.connect('C:/Users/najwa/Downloads/Database_data/Database UI.db')

    df_fakultas = pd.read_csv('C:/Users/najwa/Downloads/Database_data/Fakultas.csv', encoding='utf-8')
    df_fakultas.to_sql('Fakultas', conn, if_exists='replace', index=False)

    df_program_studi = pd.read_csv('C:/Users/najwa/Downloads/Database_data/Program Studi.csv', encoding='utf-8')
    df_program_studi.to_sql('ProgramStudi', conn, if_exists='replace', index=False)

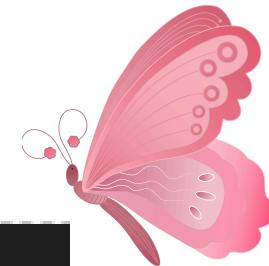
    df_biaya_pendidikan = pd.read_csv('C:/Users/najwa/Downloads/Database_data/Biaya Pendidikan.csv', encoding='utf-8')
    df_biaya_pendidikan.to_sql('BiayaPendidikan', conn, if_exists='replace', index=False)

    df_rumpun = pd.read_csv('C:/Users/najwa/Downloads/Database_data/Rumpun.csv', encoding='utf-8')
    df_rumpun.to_sql('Rumpun', conn, if_exists='replace', index=False)

    df_prospek_kerja = pd.read_csv('C:/Users/najwa/Downloads/Database_data/Prospek Kerja.csv', encoding='utf-8')
    df_prospek_kerja.to_sql('ProspekKerja', conn, if_exists='replace', index=False)

    df_program_pendidikan = pd.read_csv('C:/Users/najwa/Downloads/Database_data/Program Pendidikan.csv', encoding='utf-8')
    df_program_pendidikan.to_sql('ProgramPendidikan', conn, if_exists='replace', index=False)

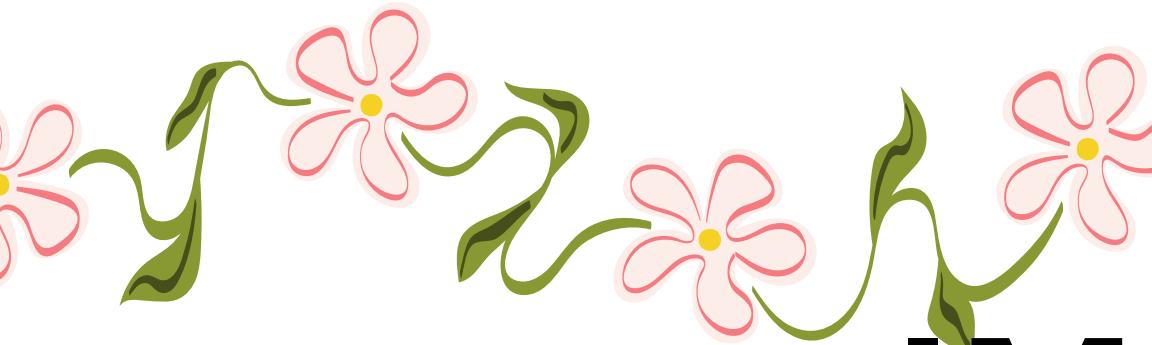
    conn.commit()
    conn.close()
```



IMPLEMENTASI SQL

- Implementasi SQL ketika memilih opsi Biaya Pendidikan
Fungsi query untuk Biaya Pendidikan berdasarkan keyword. Keyword yang akan digunakan ialah nama Program Studi yang lengkap sehingga digunakan “=”

```
def query_biaya_pendidikan(keyword):  
    conn = sqlite3.connect('C:/Users/najwa/Downloads/Database_data/Database UI.db')  
    cursor = conn.cursor()  
  
    query = """  
        SELECT b.ID_Biaya, b.Golongan, b.Harga  
        FROM BiayaPendidikan b  
        JOIN Rumpun r ON b.ID_Rumpun = r.ID_Rumpun  
        JOIN Fakultas f ON f.ID_Rumpun = r.ID_Rumpun  
        JOIN ProgramStudi ps ON f.ID_Fakultas = ps.ID_Fakultas  
        WHERE ps.Nama_Prodi = ? OR f.Nama_Fakultas = ?  
    """  
  
    cursor.execute(query, (keyword, keyword))  
    results = cursor.fetchall()  
    conn.close()  
    return results
```



IMPLEMENTASI SQL



- Implementasi SQL ketika memilih opsi Program Pendidikan dan Fakultas

Query ini memilih kolom Nama_Rumpun, Nama_Fakultas, Biaya_Golongan_Terendah, dan Biaya_Golongan_Tertinggi dari tabel ProgramPendidikan (disingkat pp). Tabel ini di-join dengan tabel ProgramStudi (ps), Fakultas (f), dan Rumpun (r) berdasarkan ID terkait. Kondisi pada WHERE menyaring data berdasarkan Jenjang yang sesuai dengan parameter jenjang yang diberikan.

```
def query_program_pendidikan(jenjang):
    conn = sqlite3.connect('C:/Users/najwa/Downloads/Database_data/Database UI.db')
    cursor = conn.cursor()
    query = """
        SELECT r.Nama_Rumpun, f.Nama_Fakultas, f.Biaya_Golongan_Terendah, f.Biaya_Golongan_Tertinggi
        FROM ProgramPendidikan pp
        JOIN ProgramStudi ps ON ps.ID_Program = pp.ID_Program
        JOIN Fakultas f ON ps.ID_Fakultas = f.ID_Fakultas
        JOIN Rumpun r ON f.ID_Rumpun = r.ID_Rumpun
        WHERE pp.Jenjang = ?
        """

    cursor.execute(query, (jenjang,))
    results = cursor.fetchall()
    conn.close()
    return results
```



IMPLEMENTASI SQL

- Implementasi SQL ketika memilih opsi Program Studi dan Prospek Kerja
Fungsi query untuk Program Studi berdasarkan keyword

```
def query_program_studi(keyword=None):
    conn = sqlite3.connect('C:/Users/najwa/Downloads/Database_data/Database UI.db')
    cursor = conn.cursor()
    if keyword:
        query = """
            SELECT ps.Nama_Prodi, ps.Akreditasi, ps.Dayatampung, f.Nama_Fakultas
            FROM ProgramStudi ps
            JOIN Fakultas f ON ps.ID_Fakultas = f.ID_Fakultas
            WHERE ps.Nama_Prodi LIKE ?
            ORDER BY ps.Nama_Prodi
            """
        cursor.execute(query, ('%' + keyword + '%',))
```

Pada bagian ini, jika keyword diberikan, query akan memilih Nama_Prodi, Akreditasi, Daya_Tampung, dan Nama_Fakultas dari tabel ProgramStudi (ps) yang berhubungan dengan tabel Fakultas (f). Kondisi WHERE menggunakan LIKE untuk mencocokkan Nama_Prodi yang mengandung keyword dan hasilnya diurutkan berdasarkan Nama_Prodi.



IMPLEMENTASI SQL

- Implementasi SQL ketika memilih opsi Program Studi dan Prospek Kerja

Jika keyword tidak diberikan (None), query akan memilih kolom yang sama dari tabel yang sama tetapi tanpa kondisi WHERE, dan hasilnya tetap diurutkan berdasarkan Nama_Prodi.

```
else:  
    query = """  
SELECT ps.Nama_Prodi, ps.Akreditasi, ps.Daya_Tampung, f.Nama_Fakultas  
FROM ProgramStudi ps  
JOIN Fakultas f ON ps.ID_Fakultas = f.ID_Fakultas  
ORDER BY ps.Nama_Prodi  
"""  
  
    cursor.execute(query)  
    results = cursor.fetchall()  
    conn.close()  
    return results
```





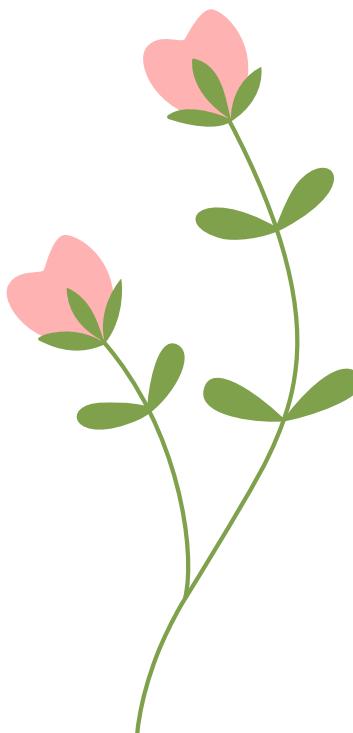
IMPLEMENTASI SQL



- Implementasi SQL ketika memilih opsi Program Studi dan Prospek Kerja

```
def query_prospek_kerja(prodi):
    conn = sqlite3.connect('C:/Users/najwa/Downloads/Database_data/Database UI.db')
    cursor = conn.cursor()
    query = """
        SELECT Nama_Prospek, Gaji_per_bulan
        FROM ProspekKerja
        JOIN ProgramStudi ON ProspekKerja.ID_Prodi = ProgramStudi.ID_Prodi
        WHERE ProgramStudi.Nama_Prodi = ?
        """

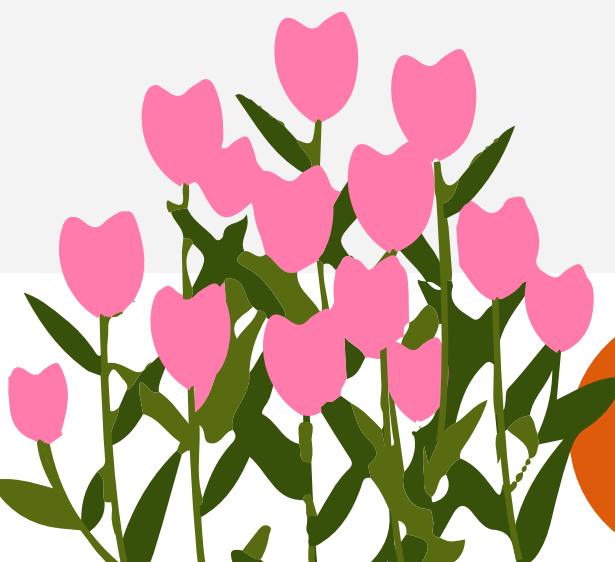
    cursor.execute(query, (prodi,))
    results = cursor.fetchall()
    conn.close()
    return results
```



Query ini memilih kolom Nama_Prospek dan Gaji_per_bulan dari tabel ProspekKerja yang di-join dengan tabel ProgramStudi berdasarkan ID_Prodi. Kondisi pada WHERE menyaring data berdasarkan Nama_Prodi yang sesuai dengan parameter prodi yang diberikan.



IMPLEMENTASI GUI



IMPLEMENTASI GUI

1. Import Package

```
import tkinter as tk  
from tkinter import ttk
```

2. Biaya Pendidikan

- Menampilkan pencarian Biaya Pendidikan

```
def show_jenjang_buttons():  
    for widget in main_frame.winfo_children():  
        widget.destroy()  
    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue").pack(pady=10)  
    separator = ttk.Separator(main_frame, orient='horizontal')  
    separator.pack(fill='x', padx=5, pady=5)  
    ttk.Label(main_frame, text="Jenjang Pendidikan di UI", font=('Verdana', 30)).pack(pady=10)  
  
    jenjang_names = ["S1", "D3", "D4"]  
    for name in jenjang_names:  
        button = ttk.Button(main_frame, text=name, style="TButton",  
                           command=lambda j=name: show_detail_jenjang(j))  
        button.pack(pady=10)  
  
    back_button = ttk.Button(main_frame, text="Kembali", command=reset_view, style="TButton")  
    back_button.pack(pady=(20, 10), side='bottom')
```

IMPLEMENTASI GUI

- Melakukan pencarian dan menampilkan hasil pencarian.

Jika terdapat kesalahan penulisan nama Program Studi, maka hasil pencarian akan menampilkan "!! Data Tidak Ditemukan !!"

```
def perform_search(keyword):
    results = query_biaya_pendidikan(keyword)
    for widget in main_frame.winfo_children():
        widget.destroy()

    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue").pack(pady=10)
    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)
    ttk.Label(main_frame, text="Hasil Pencarian", font=('Verdana', 30)).pack(pady=20)

    if not results:
        ttk.Label(main_frame, text="!! Data Tidak Ditemukan !!", font=('Verdana', 16), foreground="#8b0000").pack(pady=20)
    else:
        tree = ttk.Treeview(main_frame, columns=("Golongan", "Uang Kuliah Tunggal (Rp)"), show="headings", height=11)
        tree.heading("Golongan", text="Golongan")
        tree.heading("Uang Kuliah Tunggal (Rp)", text="Uang Kuliah Tunggal (Rp)")
        tree.column("Golongan", anchor="center", width=100)
        tree.column("Uang Kuliah Tunggal (Rp)", anchor="center", width=200)

        style = ttk.Style()
        style.configure("Treeview", background="white", fieldbackground="white")
        style.map('Treeview', background=[('selected', 'brown')])
        style.map('Treeview.Heading', background=[('active', 'yellow')])

        for id_biaya, golongan, harga in results:
            tree.insert("", "end", values=(golongan, harga))

        tree.pack(pady=10, fill='x', expand=True)

    back_button = ttk.Button(main_frame, text="Kembali", command=show_biaya_pendidikan_search)
    back_button.pack(pady=(20, 10), side='bottom')
```

IMPLEMENTASI GUI

3. Ketika memilih opsi Program Pendidikan dan Fakultas

- Menampilkan button untuk memilih Jenjang Pendidikan yang ada (S1, D3, D4)

```
def show_jenjang_buttons():
    for widget in main_frame.winfo_children():
        widget.destroy()
    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue").pack(pady=10)
    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)
    ttk.Label(main_frame, text="Jenjang Pendidikan di UI", font=('verdana', 30)).pack(pady=10)

    jenjang_names = ["S1", "D3", "D4"]
    for name in jenjang_names:
        button = ttk.Button(main_frame, text=name, style="TButton",
                            command=lambda j=name: show_detail_jenjang(j))
        button.pack(pady=10)

    back_button = ttk.Button(main_frame, text="Kembali", command=reset_view, style="TButton")
    back_button.pack(pady=(20, 10), side='bottom')
```

```
def show_detail_jenjang(jenjang_name):
    results = query_program_pendidikan(jenjang_name)
    for widget in main_frame.winfo_children():
        widget.destroy()

    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue").pack(pady=10)
    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)
    ttk.Label(main_frame, text=f"Detail untuk Jenjang {jenjang_name}", font=('verdana', 30)).pack(pady=10)

    style = ttk.Style()
    style.configure("Treeview", background="white", fieldbackground="white")
    style.map('Treeview', background=[('selected', 'brown')])
    style.map('Treeview.Heading', background=[('active', 'yellow')])

    tree = ttk.Treeview(main_frame, columns=("Rumpun", "Fakultas", "Biaya Terendah", "Biaya Tertinggi"), show="headings", height=15)
    tree.heading("Rumpun", text="Rumpun")
    tree.heading("Fakultas", text="Fakultas")
    tree.heading("Biaya Terendah", text="Uang Kuliah Tunggal Terendah")
    tree.heading("Biaya Tertinggi", text="Uang Kuliah Tunggal Tertinggi")

    fakultas_data = {}
    for rumpun, fakultas, biaya_terendah, biaya_tertinggi in results:
        if fakultas not in fakultas_data:
            fakultas_data[fakultas] = [rumpun, fakultas, biaya_terendah, biaya_tertinggi]
        else:
            existing_biaya_terendah, existing_biaya_tertinggi = fakultas_data[fakultas][2], fakultas_data[fakultas][3]
            fakultas_data[fakultas][2] = min(existing_biaya_terendah, biaya_terendah)
            fakultas_data[fakultas][3] = max(existing_biaya_tertinggi, biaya_tertinggi)

    for fakultas in fakultas_data.values():
        tree.insert("", "end", values=(fakultas[0], fakultas[1], fakultas[2], fakultas[3]))

    tree.pack(pady=10, fill='x', expand=True)

    back_button = ttk.Button(main_frame, text="Kembali", command=show_jenjang_buttons)
    back_button.pack(pady=(20,10), side='bottom')
```

IMPLEMENTASI GUI

4. Ketika memilih opsi Program Studi dan Prospek Kerja

- Menampilkan Program Studi

```
def show_program_studi():
    for widget in main_frame.winfo_children():
        widget.destroy()

    bg_color = "#ffc067"

    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue", background=bg_color).pack(pady=10)
    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)
    ttk.Label(main_frame, text="Program Studi", font=('Verdana', 30), background=bg_color).pack(pady=15)

    search_frame = tk.Frame(main_frame, bg=bg_color)
    search_frame.pack(fill='x', pady=10, padx=5)
    ttk.Label(search_frame, text="Cari Program Studi:", font=('Verdana', 12), background=bg_color).pack(side='left')
    search_entry = ttk.Entry(search_frame, width=50)
    search_entry.pack(side='left', padx=5)
    search_button = ttk.Button(search_frame, text="Cari", command=lambda: search_program_studi(search_entry.get()))
    search_button.pack(side='left')

    tree_frame = tk.Frame(main_frame, bg=bg_color)
    tree_frame.pack(fill='both', expand=True)

    global tree
    tree = ttk.Treeview(tree_frame, columns=("Nama Prodi", "Akreditasi", "Daya Tampung", "Asal Fakultas"), show="headings", height=11)
    tree.heading("Nama Prodi", text="Nama Program Studi")
    tree.heading("Akreditasi", text="Akreditasi")
    tree.heading("Daya Tampung", text="Daya Tampung")
    tree.heading("Asal Fakultas", text="Asal Fakultas")
    tree.column("Nama Prodi", anchor="center", width=200)
    tree.column("Akreditasi", anchor="center", width=100)
    tree.column("Daya Tampung", anchor="center", width=100)
    tree.column("Asal Fakultas", anchor="center", width=150)

    results = query_program_studi()
    for nama_prodi, akreditasi, daya_tampung, nama_fakultas in results:
        tree.insert("", "end", values=(nama_prodi, akreditasi, daya_tampung, nama_fakultas))

    style = ttk.Style()
    style.configure("Treeview", background="white", fieldbackground="white")
    style.map('Treeview', background=[('selected', 'brown')])
    style.map('Treeview.Heading', background=[('active', 'yellow')])

    tree.pack(pady=10, fill='both', expand=True)
    tree.bind("<Double-1>", lambda event: show_prospek_detail(tree))

    ttk.Label(main_frame, text="Klik baris untuk melihat detail informasi", font=('Verdana', 12), foreground="#8b0000", background="yellow").pack(pady=10, anchor="w")
    back_button = ttk.Button(main_frame, text="Kembali", command=reset_view)
    back_button.pack(pady=(20, 10), side='bottom')
```

- Menampilkan pencarian Program Studi

Disediakan tempat untuk melakukan pencarian karena untuk memudahkan user mencari Program Studi di antara 79 Program Studi yang tersedia. Tabel ini akan memiliki kolom-kolom sebagai berikut:

1. Nama Program Studi

2. Akreditasi

3. Daya Tampung

4. Asal Fakultas

```
def search_program_studi(keyword):
    for widget in main_frame.winfo_children():
        widget.destroy()

    bg_color = "#ffc067"

    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue", background=bg_color).pack(pady=10)
    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)
    ttk.Label(main_frame, text="Program Studi", font=('Verdana', 30), background=bg_color).pack(pady=15)

    search_frame = tk.Frame(main_frame, bg=bg_color)
    search_frame.pack(fill='x', pady=10, padx=5)
    ttk.Label(search_frame, text="Cari Program Studi:", font=('Verdana', 12), background=bg_color).pack(side='left')
    search_entry = ttk.Entry(search_frame, width=40)
    search_entry.pack(side='left', padx=5)
    search_entry.insert(0, keyword)
    search_button = ttk.Button(search_frame, text="Cari", command=lambda: search_program_studi(search_entry.get()))
    search_button.pack(side='left')

    tree_frame = tk.Frame(main_frame, bg=bg_color)
    tree_frame.pack(fill='both', expand=True)

    global tree
    tree = ttk.Treeview(tree_frame, columns=("Nama Prodi", "Akreditasi", "Daya Tampung", "Asal Fakultas"), show="headings", height=11)
    tree.heading("Nama Prodi", text="Nama Program Studi")
    tree.heading("Akreditasi", text="Akreditasi")
    tree.heading("Daya Tampung", text="Daya Tampung")
    tree.heading("Asal Fakultas", text="Asal Fakultas")
    tree.column("Nama Prodi", anchor="center", width=200)
    tree.column("Akreditasi", anchor="center", width=100)
    tree.column("Daya Tampung", anchor="center", width=100)
    tree.column("Asal Fakultas", anchor="center", width=150)

    results = query_program_studi(keyword)
    for nama_prodi, akreditasi, daya_tampung, nama_fakultas in results:
        tree.insert("", "end", values=(nama_prodi, akreditasi, daya_tampung, nama_fakultas))

    tree.pack(pady=10, fill='both', expand=True)
    tree.bind("<Double-1>", lambda event: show_prospek_detail(tree))

    ttk.Label(main_frame, text="Klik baris untuk melihat detail informasi", font=('Verdana', 12), foreground="#8b0000", background="yellow").pack(pady=10, anchor="center")
    back_button = ttk.Button(main_frame, text="Kembali", command=show_program_studi)
    back_button.pack(pady=(20, 10), side='bottom')
```

IMPLEMENTASI GUI

- Menampilkan detail Prospek Kerja untuk Program Studi yang terpilih

Detail prospek kerja untuk program studi yang dipilih akan ditampilkan ketika pengguna mengklik dua kali pada baris yang dipilih dalam tabel program studi.

```
def show_prospek_detail(tree):
    selected_item = tree.selection()[0]
    prodi = tree.item(selected_item, 'values')[0]
    results = query_prospek_kerja(prodi)

    for widget in main_frame.winfo_children():
        widget.destroy()

    ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue").pack(pady=10)
    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)
    ttk.Label(main_frame, text=f"Prospek Kerja Program Studi {prodi}", font=('Verdana', 30)).pack(pady=10)

    tree = ttk.Treeview(main_frame, columns=("Nama Prospek", "Gaji per Bulan"), show="headings", height=2)
    tree.heading("Nama Prospek", text="Nama Prospek Kerja")
    tree.heading("Gaji per Bulan", text="Gaji per Bulan")
    tree.column("Nama Prospek", anchor="center", width=200)
    tree.column("Gaji per Bulan", anchor="center", width=150)

    for nama_prospek, gaji_per_bulan in results:
        tree.insert("", "end", values=(nama_prospek, gaji_per_bulan))

    tree.pack(pady=10, fill='x', expand=True)

    back_button = ttk.Button(main_frame, text="Kembali", command=show_program_studi)
    back_button.pack(pady=(20, 10), side='bottom')
```

IMPLEMENTASI GUI

5. Mengembalikan ke tampilan awal

Menyediakan fungsi untuk mengatur ulang tampilan aplikasi ke halaman utama.

```
def reset_view():
    for widget in main_frame.winfo_children():
        widget.destroy()
    initialize_main_view()
```

6. Menginisialisasi tampilan utama aplikasi dengan opsi-opsi menu yang dapat dipilih oleh pengguna.

```
def initialize_main_view():
    global canvas, flying_text_instances, flying_text, option_var

    label_welcome = ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue")
    label_welcome.pack(pady=(20, 20))

    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)

    canvas = tk.Canvas(main_frame, height=100, bg="#ffc067", bd=0, highlightthickness=0)
    canvas.pack(fill='x', pady=20)

    flying_text = "Selamat Datang di UIAcademicPortal"
    flying_text_instances = []

    initial_text_x = root.winfo_width()
    flying_text_id = canvas.create_text(initial_text_x, 30, text=flying_text, font=('Verdana', 8), anchor='w')
    flying_text_instances.append((flying_text_id, initial_text_x))

    option_var = tk.StringVar(value="Informasi apa yang ingin Anda cari?")
    options = ["Biaya Pendidikan", "Program Pendidikan dan Fakultas", "Program Studi dan Prospek Kerja"]
    option_menu = ttk.Combobox(main_frame, textvariable=option_var, values=options, state="readonly", width=35)
    option_menu.pack(pady=20)
    option_menu.bind("<<ComboboxSelected>>", on_combobox_select)
    option_menu.bind("<FocusIn>", lambda e: option_var.set('') if option_var.get() == "Informasi apa yang ingin Anda cari?" else None)
    option_menu.bind("<FocusOut>", lambda e: option_var.set("Informasi apa yang ingin Anda cari?") if not option_var.get() else None)

    update_flying_text()
```

IMPLEMENTASI GUI

7. Fungsi untuk menampilkan pilihan dari combobox
Menangani aksi ketika pengguna memilih opsi dari
menu combobox dan menampilkan tampilan yang
sesuai.

```
def on_combobox_select(event):
    option = option_var.get()
    if option == "Biaya Pendidikan":
        show_biaya_pendidikan_search()
    elif option == "Program Pendidikan dan Fakultas":
        show_jenjang_buttons()
    elif option == "Program Studi dan Prospek Kerja":
        show_program_studi()
```

8. Fungsi untuk memperbarui flying text
Membuat efek teks berjalan yang ditampilkan pada canvas.

```
def update_flying_text():
    global flying_text_instances

    for text_id, _ in flying_text_instances:
        canvas.move(text_id, -2, 0)

    if canvas.coords(flying_text_instances[0][0])[0] < -canvas.bbox(flying_text_instances[0][0])[2]:
        canvas.delete(flying_text_instances[0][0])
        flying_text_instances.pop(0)

    last_text_id, _ = flying_text_instances[-1]
    if canvas.coords(last_text_id)[0] < root.winfo_width() - 400:
        new_text_x = root.winfo_width()
        new_text_id = canvas.create_text(new_text_x, 30, text=flying_text, font=('Verdana', 8), anchor='w')
        flying_text_instances.append((new_text_id, new_text_x))

    canvas.after(50, update_flying_text)
```

9. Fungsi mainframe untuk menjalankan tampilan aplikasi pertama kali.

Menginisialisasi dan menjalankan aplikasi Tkinter, membuat database, memuat data, dan menampilkan antarmuka pengguna

```
def main():
    global root, main_frame, option_var, label_info, option_menu, canvas, flying_text, flying_text_instances

    create_db_and_tables()
    load_data()

    root = tk.Tk()
    root.title("UIAcademicPortal")
    root.geometry("900x600")

    style = ttk.Style()
    style.theme_use('clam')
    style.configure("Custom.TFrame", background="#ffc067")
    style.configure("TLabel", background="#ffc067", font=('Verdana', 12))
    style.configure("TButton", font=('Verdana', 12))

    main_frame = ttk.Frame(root, style="Custom.TFrame", padding="10")
    main_frame.pack(fill=tk.BOTH, expand=True)

    label_welcome = ttk.Label(main_frame, text="UIAcademicPortal", font=('Verdana', 50), foreground="dark blue")
    label_welcome.pack(pady=(20, 20))

    separator = ttk.Separator(main_frame, orient='horizontal')
    separator.pack(fill='x', padx=5, pady=5)

    canvas = tk.Canvas(main_frame, height=100, bg="#ffc067", bd=0, highlightthickness=0)
    canvas.pack(fill='x', pady=20)

    flying_text = "Selamat Datang di UIAcademicPortal"
    flying_text_instances = []

    initial_text_x = root.winfo_width()
    flying_text_id = canvas.create_text(initial_text_x, 30, text=flying_text, font=('Verdana', 8), anchor='w')
    flying_text_instances.append((flying_text_id, initial_text_x))

    option_var = tk.StringVar(value="Informasi apa yang ingin Anda cari?")
    options = ["Biaya Pendidikan", "Program Pendidikan dan Fakultas", "Program Studi dan Prospek Kerja"]
    option_menu = ttk.Combobox(main_frame, textvariable=option_var, values=options, state="readonly", width=35)
    option_menu.pack(pady=20)
    option_menu.bind("<<ComboboxSelected>>", on_combobox_select)
    option_menu.bind("<FocusIn>", lambda e: option_var.set('') if option_var.get() == "Informasi apa yang ingin Anda cari?" else None)
    option_menu.bind("<FocusOut>", lambda e: option_var.set("Informasi apa yang ingin Anda cari?") if not option_var.get() else None)

    label_info = ttk.Label(main_frame, text="")
    label_info.pack(pady=10, fill='x')

    update_flying_text()

    root.mainloop()

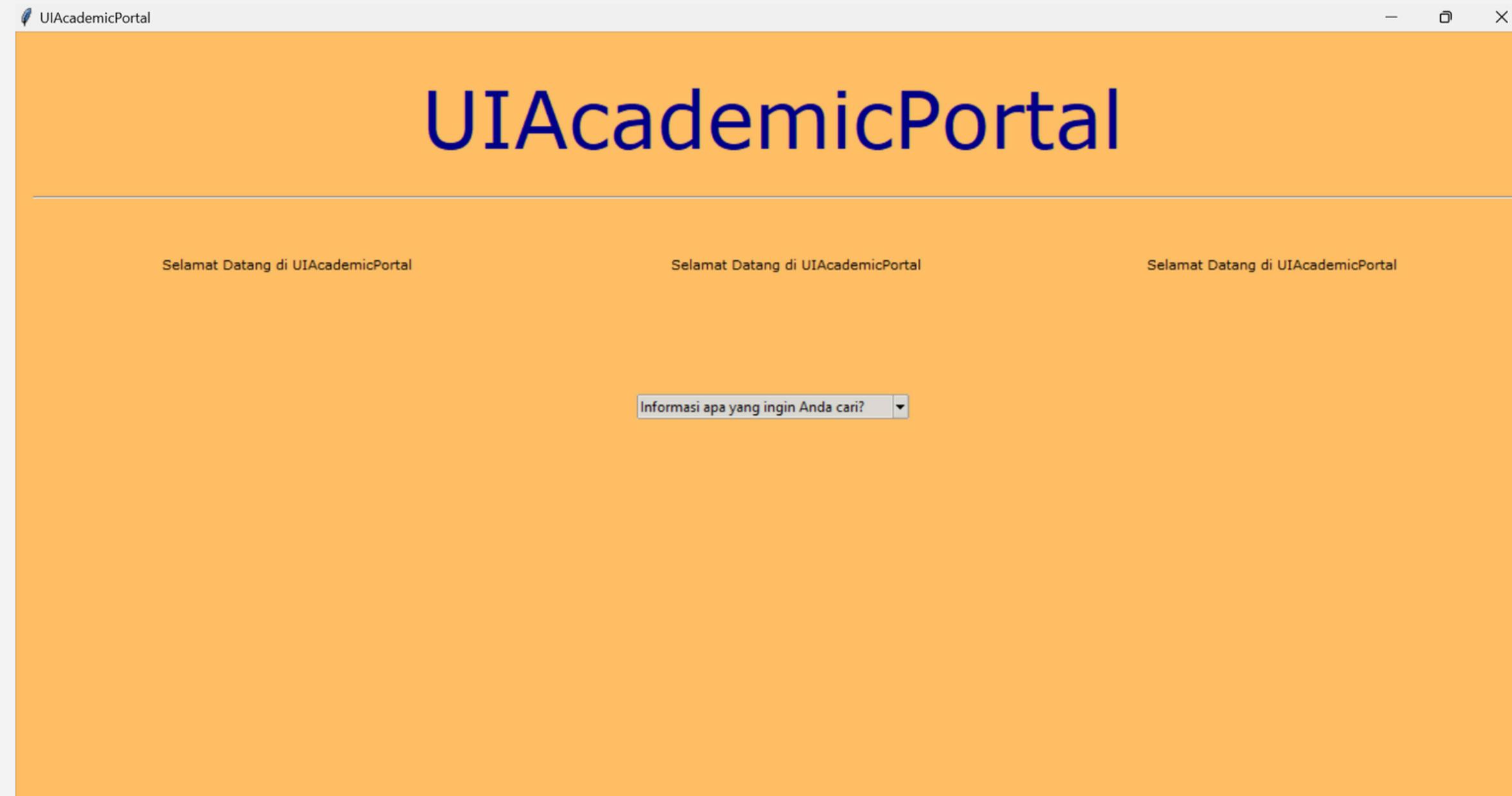
if __name__ == "__main__":
    main()
```

TESTING



TESTING

Tampilan awal



TESTING



TESTING

Ketika penulisan nama program studi benar,

UIAcademicPortal

Biaya Pendidikan di UI

Cari melalui Nama Program Studi

Contoh: Ilmu Aktuaria

Ilmu Komunikasi

Cari

Kembali

This screenshot shows the UI Academic Portal search interface. The search bar contains 'Ilmu Komunikasi'. Below the search bar, there is a placeholder text 'Contoh: Ilmu Aktuaria'. A 'Cari' (Search) button is located below the search bar. At the bottom left is a 'Kembali' (Back) button.

maka akan memberikan output yang sesuai

UIAcademicPortal

Hasil Pencarian

Golongan	Uang Kuliah Tunggal (Rp)
K-1	0-500.000
K-2	500.000-1.000.000
K-3	1.000.000-2.000.000
K-4	2.000.000-3.000.000
K-5	3.000.000-4.000.000
K-6	4.000.000-5.000.000
K-7	5.000.000-7.500.000
K-8	7.500.000-10.000.000
K-9	10.000.000-12.500.000
K-10	12.500.000-15.000.000
K-11	15.000.000-17.500.000

Kembali

This screenshot shows the search results for 'Ilmu Komunikasi'. The results are displayed in a table format. The columns are 'Golongan' (Group) and 'Uang Kuliah Tunggal (Rp)' (Single Tuition Fee). The table lists 11 groups with their corresponding tuition fee ranges. The last row, group K-9, is highlighted in red.

■ TESTING

Ketika penulisan nama program studi tidak lengkap,

The screenshot shows the UI Academic Portal search interface. The search bar contains the incomplete text "aktuaria". Below the search bar is a "Cari" button. At the bottom of the page is a "Kembali" button.

maka akan memberikan output "!! Data Tidak Ditemukan !!"

The screenshot shows the UI Academic Portal search results page. The main heading is "Hasil Pencarian". Below it, a red message box displays the text "!! Data Tidak Ditemukan !!". At the bottom of the page is a "Kembali" button.

TESTING



TESTING

Muncul jenjang atau program pendidikan yang ada di UI, yaitu S1, D3, dan D4.

Ketika memilih jenjang S1

UIAcademicPortal

Jenjang Pendidikan di UI

S1
D3
D4

Kembali

UIAcademicPortal

UIAcademicPortal

Detail untuk Jenjang S1

Rumpun	Fakultas	Uang Kuliah Tunggal Terendah	Uang Kuliah Tunggal Tertinggi
Ilmu Kesehatan	Kedokteran	Rp0-500.000	Rp17.500.000-20.000.000
Ilmu Kesehatan	Kedokteran Gigi	Rp0-500.000	Rp17.500.000-20.000.000
Ilmu Kesehatan	Ilmu Keperawatan	Rp0-500.000	Rp17.500.000-20.000.000
Ilmu Kesehatan	Farmasi	Rp0-500.000	Rp17.500.000-20.000.000
Ilmu Kesehatan	Kesehatan Masyarakat	Rp0-500.000	Rp17.500.000-20.000.000
Sains dan Teknologi	Matematika dan Ilmu Pengetahuan Alam	Rp0-500.000	Rp17.500.000-20.000.000
Sains dan Teknologi	Teknik	Rp0-500.000	Rp17.500.000-20.000.000
Sains dan Teknologi	Ilmu Komputer	Rp0-500.000	Rp17.500.000-20.000.000
Ilmu Sosial dan Humaniora	Hukum	Rp0-500.000	Rp15.000.000-17.500.000
Ilmu Sosial dan Humaniora	Ekonomi dan Bisnis	Rp0-500.000	Rp15.000.000-17.500.000
Ilmu Sosial dan Humaniora	Ilmu Budaya	Rp0-500.000	Rp15.000.000-17.500.000
Ilmu Sosial dan Humaniora	Psikologi	Rp0-500.000	Rp15.000.000-17.500.000
Ilmu Sosial dan Humaniora	Ilmu Sosial dan Ilmu Politik	Rp0-500.000	Rp15.000.000-17.500.000
Ilmu Sosial dan Humaniora	Ilmu Administrasi	Rp0-500.000	Rp15.000.000-17.500.000

Kembali

TESTING

Ketika memilih jenjang D3

UIAcademicPortal

Detail untuk Jenjang D3

Rumpun	Vokasi	Fakultas	Uang Kuliah Tunggal Terendah	Uang Kuliah Tunggal Tertinggi
Vokasi			Rp0-500.000	Rp17.500.000-20.000.000

Kembali

Ketika memilih jenjang D4

UIAcademicPortal

Detail untuk Jenjang D4

Rumpun	Vokasi	Fakultas	Uang Kuliah Tunggal Terendah	Uang Kuliah Tunggal Tertinggi
Vokasi			Rp0-500.000	Rp17.500.000-20.000.000

Kembali

TESTING



TESTING

User langsung memilih nama program studi yang diinginkannya.

The screenshot shows the 'UIAcademicPortal' interface with the title 'Program Studi'. A search bar at the top has the placeholder 'Cari Program Studi:' and a 'Cari' button. Below it is a table with columns: 'Nama Program Studi', 'Akreditasi', 'Daya Tampung', and 'Asal Fakultas'. The table lists various programs like Administrasi Asuransi dan Aktuaria, Administrasi Keuangan dan Perbankan, etc. The row for 'Antropologi Sosial' is highlighted in red, indicating it is selected. A tooltip at the bottom left says 'Klik baris untuk melihat detail informasi'. At the bottom right is a 'Kembali' button.

Nama Program Studi	Akreditasi	Daya Tampung	Asal Fakultas
Administrasi Asuransi dan Aktuaria	Unggul	50	Vokasi
Administrasi Keuangan dan Perbankan	A	50	Vokasi
Administrasi Perkantoran	A	50	Vokasi
Administrasi Perpajakan	A	50	Vokasi
Administrasi Rumah Sakit	Unggul	50	Vokasi
Akuntansi	Unggul	213	Ekonomi dan Bisnis
Akuntansi	Unggul	50	Vokasi
Antropologi Sosial	Unggul	48	Ilmu Sosial dan Ilmu Politik
Arkeologi	Unggul	50	Ilmu Budaya
Arsitektur	Unggul	67	Teknik
Arsitektur Interior	Unggul	41	Teknik

Dengan meng-klik baris yang user inginkan, maka akan langsung muncul informasi tambahan terkait prospek kerja yang sudah sesuai dengan program studinya.

The screenshot shows the 'UIAcademicPortal' interface with the title 'Prospek Kerja Program Studi Antropologi Sosial'. It displays a table with columns: 'Nama Prospek Kerja' and 'Gaji per Bulan'. Two rows are listed: 'Konsultan Sosial' with a salary range of 'Rp6 juta - Rp14 juta' and 'Peneliti Antropologi' with a salary range of 'Rp5 juta - Rp12 juta'. At the bottom right is a 'Kembali' button.

Nama Prospek Kerja	Gaji per Bulan
Konsultan Sosial	Rp6 juta - Rp14 juta
Peneliti Antropologi	Rp5 juta - Rp12 juta

TESTING

Untuk efektivitas dalam mencari beberapa program studi dari 79 program studi yang tersedia. User cukup mengetikkan “string” nama dari program studinya,

The screenshot shows the UI Academic Portal interface. At the top, it says "UIAcademicPortal" and "Program Studi". Below that is a search bar with "Cari Program Studi: kom" and a "Cari" button. A table below the search bar lists program details: Nama Program Studi (Ilmu Komputer, Ilmu Komunikasi, Teknik Komputer), Akreditasi (Unggul, A, Unggul), Daya Tampung (67, 118, 78), and Asal Fakultas (Ilmu Komputer, Ilmu Sosial dan Ilmu Politik, Teknik). A yellow callout box highlights the search bar and the resulting table. At the bottom, there's a note "Klik baris untuk melihat detail informasi" and a "Kembali" button.

Nama Program Studi	Akreditasi	Daya Tampung	Asal Fakultas
Ilmu Komputer	Unggul	67	Ilmu Komputer
Ilmu Komunikasi	A	118	Ilmu Sosial dan Ilmu Politik
Teknik Komputer	Unggul	78	Teknik

maka akan langsung muncul program studi yang namanya mengandung string yang user ketikan. User bisa langsung memilih nama program studi yang diinginkannya.

The screenshot shows the same UI Academic Portal interface as the first one, but with a different result. The search bar still has "Cari Program Studi: kom". The table now has a single row highlighted in red: Nama Program Studi (Teknik Komputer), Akreditasi (Unggul), Daya Tampung (78), and Asal Fakultas (Teknik). The rest of the table rows are white. A yellow callout box highlights the search bar and the highlighted row. At the bottom, there's a note "Klik baris untuk melihat detail informasi" and a "Kembali" button.

Nama Program Studi	Akreditasi	Daya Tampung	Asal Fakultas
Teknik Komputer	Unggul	78	Teknik
Ilmu Komputer	Unggul	67	Ilmu Komputer
Ilmu Komunikasi	A	118	Ilmu Sosial dan Ilmu Politik

■ TESTING

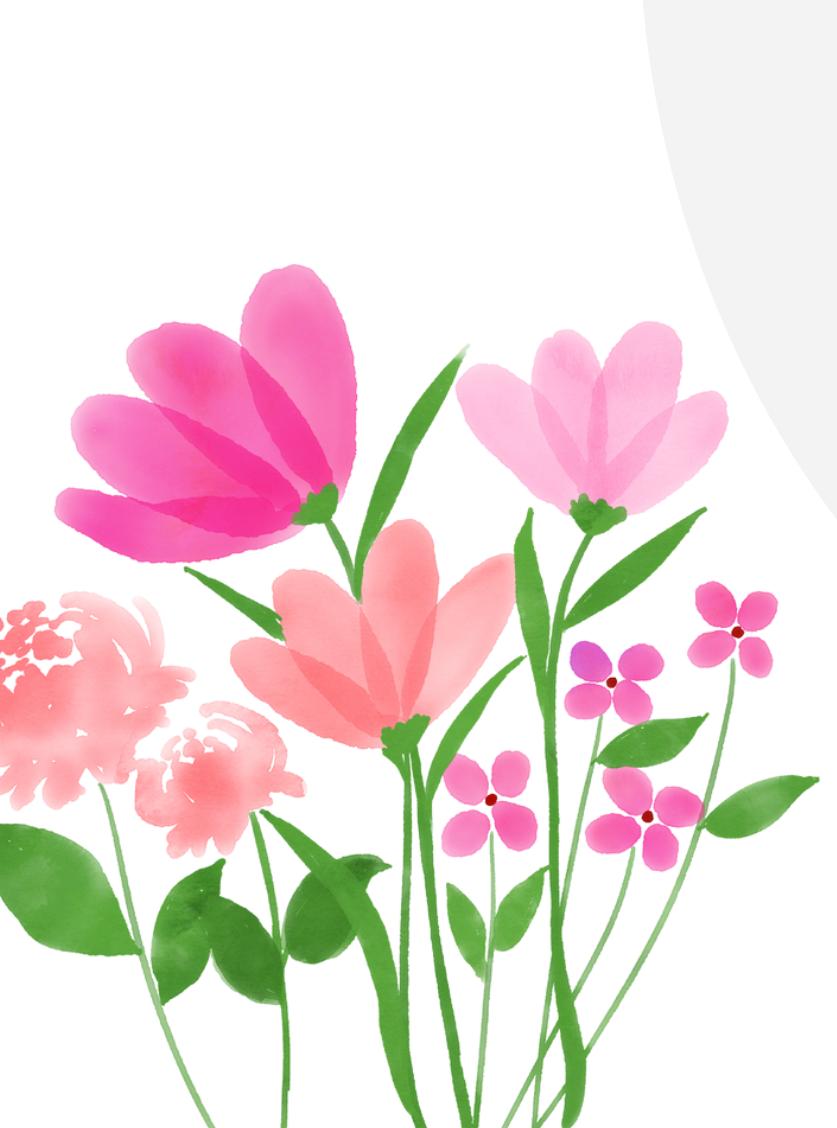
Dengan meng-klik baris yang user inginkan, maka akan langsung muncul informasi tambahan terkait prospek kerja yang sudah sesuai dengan program studinya.

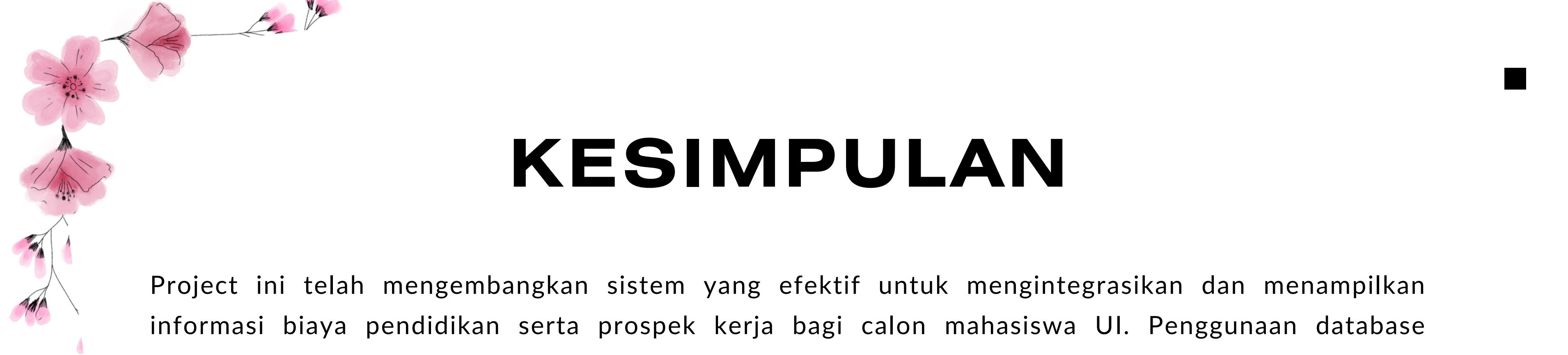
The screenshot shows a window titled "UIAcademicPortal" with the sub-header "Prospek Kerja Program Studi Teknik Komputer". Below this, there is a table displaying job prospects:

Nama Prospek Kerja	Gaji per Bulan
Insinyur sistem informasi	Rp8 juta - Rp25 juta
Pengembang perangkat lunak	Rp8 juta - Rp25 juta

At the bottom right of the table area is a "Kembali" button.

KESIMPULAN





KESIMPULAN

Project ini telah mengembangkan sistem yang efektif untuk mengintegrasikan dan menampilkan informasi biaya pendidikan serta prospek kerja bagi calon mahasiswa UI. Penggunaan database relasional dan GUI yang ramah pengguna memastikan bahwa informasi yang diberikan akurat, relevan, dan mudah diakses. Sistem ini juga fleksibel untuk dikembangkan lebih lanjut sesuai dengan kebutuhan pengguna dan perkembangan informasi di UI.

Dengan implementasi yang baik dari struktur database, SQL, dan GUI, proyek ini memberikan kontribusi signifikan dalam memudahkan calon mahasiswa UI dalam mendapatkan informasi yang mereka butuhkan untuk membuat keputusan yang tepat terkait pendidikan dan karier mereka di masa depan.

