

**LAPORAN PRATIUM PERTEMUAN 5**  
**SOFTWARE ARCHITECTURE**



**DI SUSUN OLEH:**  
**RADAM GUMELAR 164231032**  
**AQILA MALFA ZAHIRA 164231036**  
**ADINDA SYARIFATUL MUNA 164231042**

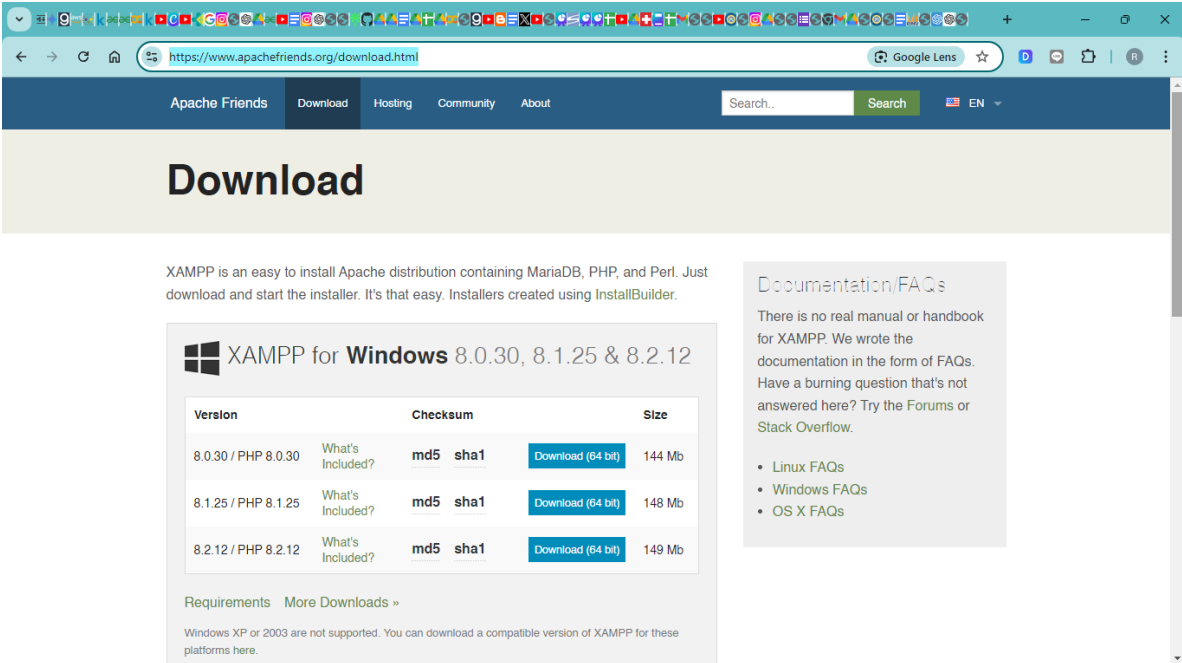
**ALGORITMA PEMROGRAMAN II**

**PROGRAM STUDI TEKNOLOGI SAINS DATA**  
**PROGRAM SARJANA**  
**UNIVERSITAS AIRLANGGA**

**2024**

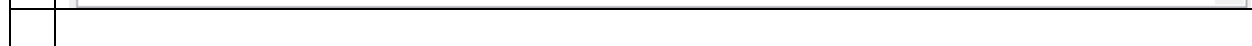
# INSTALASI XAMPP

XAMPP disini digunakan untuk mengaktifkan Apache dan database MySQL untuk keperluan pengembangan web lokal. Apache digunakan untuk menjalankan dan menguji aplikasi atau situs web langsung di laptopd tanpa harus terhubung ke server online. Sementara itu, MySQL diaktifkan untuk menyimpan dan mengelola data aplikasi web.

1	<p>Jika belum memiliki XAMPP, maka buka link <a href="https://www.apachefriends.org/download.html">https://www.apachefriends.org/download.html</a> untuk mendownload XAMPP</p> 
2	<p>Lakukan Instalasi, kemudian setelah selesai buka XAMPP hingga tampil seperti gambar berikut</p>



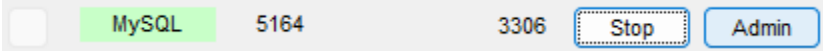
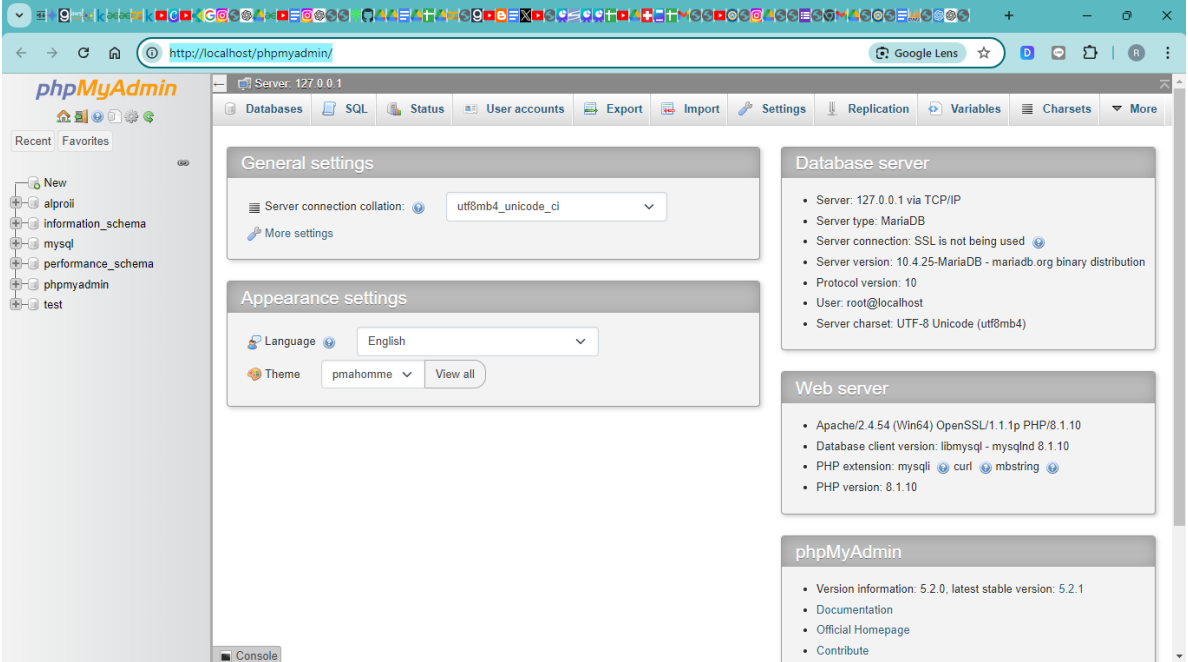

3	Nyalakan Module Apache dan MySQL
---	----------------------------------

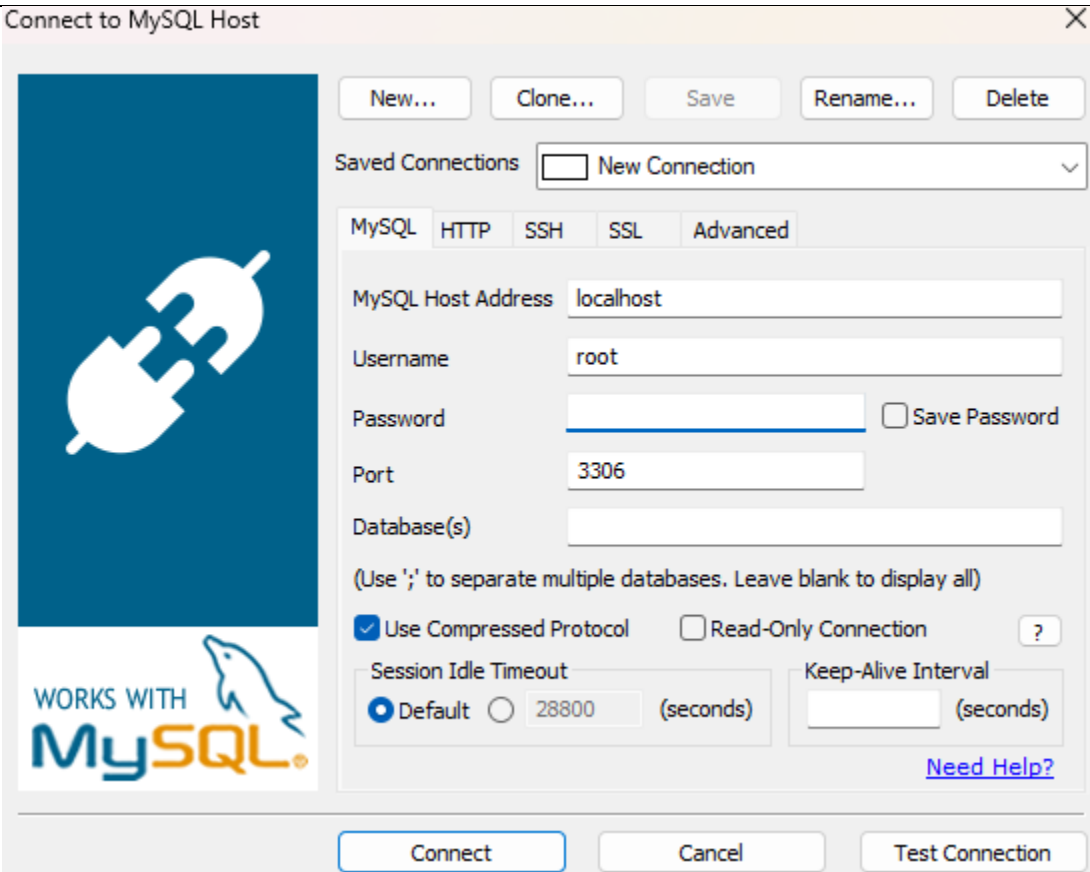
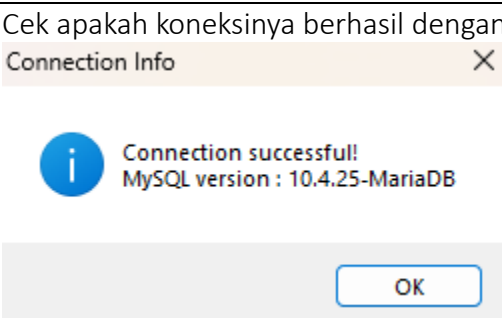


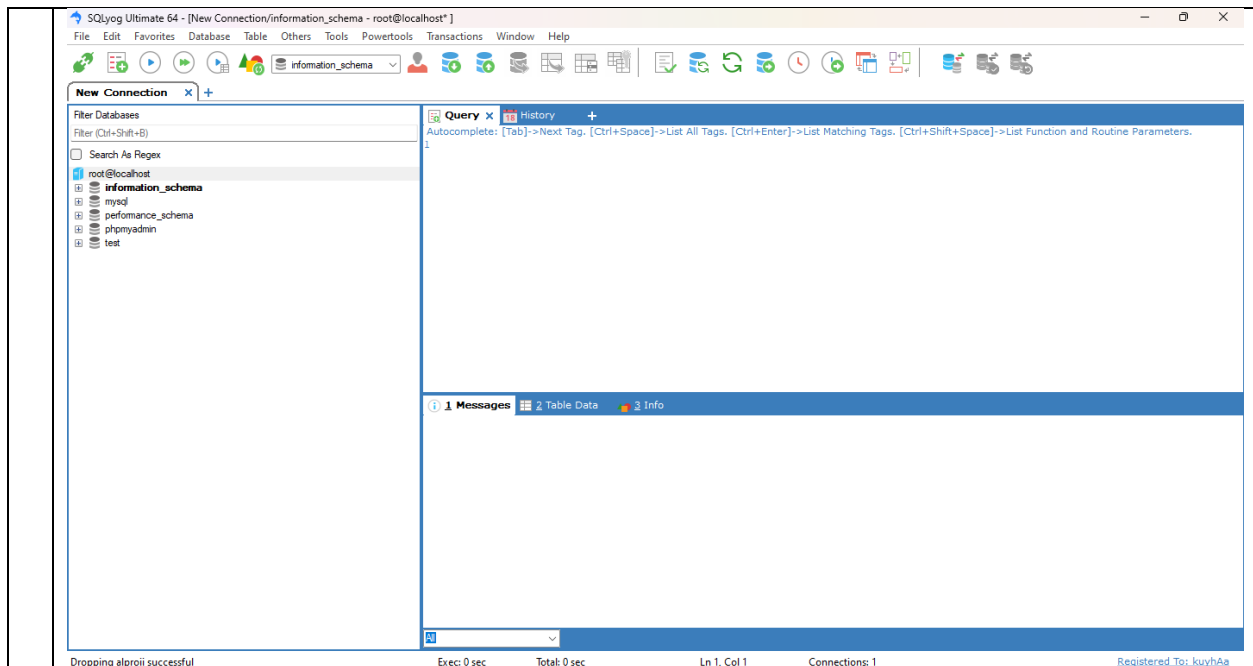
4	XAMPP Siap digunakan
---	----------------------

# PEMBUATAN DATABASE

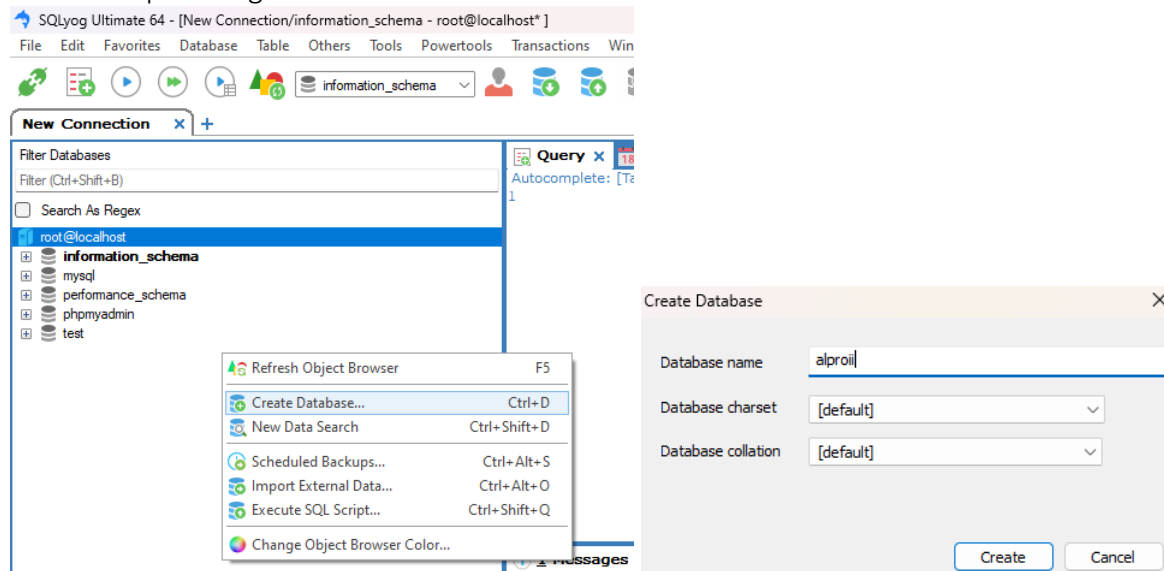
Dengan menggunakan XAMPP yang sudah dijalankan, kita dapat mengakses database lokal (PHPMyAdmin) untuk membuat suatu database yang bisa diakses tanpa harus terhubung ke server online.

1	<p>Terdapat beberapa cara untuk mengakses database, contohnya dengan klik tombol admin pada XAMPP module MySQL yang sudah dijalankan</p>  <p>Lalu akan menampilkan server lokal PHPMyAdmin</p> 
2	<p>Cara lainnya adalah dengan menggunakan aplikasi database pihak ketiga. Disini kami menggunakan SQLYog. Pastikan untuk menyalakan Apache dan MySQL pada XAMPP sebelum membuka aplikasi ini.</p>  <p>SQLYog</p> <p>salah satu GUI untuk pengelolaan database</p>
3	<p>Setelah dibuka, pertama kali akan diminta untuk menyambungkan koneksi host database yang dipunya</p>

	
4	<p>Cek apakah koneksinya berhasil dengan pilih Test Connection, jika muncul gambar berikut</p>  <p>maka koneksi telah berhasil dan tersambung dengan server</p>
5	<p>Tampilan utama SQLYog akan seperti ini</p>



6 Klik kanan pada bagian tabel di kiri lalu klik create database



Setelah membuat nama database lalu klik "Create"

7 Klik kanan pada tabel di database yang sudah di buat, lalu klik "Create table" untuk membuat tabel

8 Lakukan berulang untuk membuat 2 tabel  
Berikut adalah tabel yang kita buat dengan rincian

1. Tabel “tabel\_umum”

Table Name  Engine   
Database  Character Set   
Collation

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?
<input type="checkbox"/> Nim	varchar	9		<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Nama	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Prodi	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

2. Tabel “kuis\_alpro”

Table Name  Engine   
Database  Character Set   
Collation

Column Name	Data Type	Length	Default	PK?	Not Null?	Unsigned?
<input type="checkbox"/> Nim	varchar	9		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Nama	varchar	20		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Nilai_Kuis	int			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

9 Database telah siap digunakan dengan mengandung 2 tabel

# AKSES DATABASE DARI PYTHON DENGAN MENGGUNAKAN GUI TKINTER

Akses database dengan python bisa dilakukan dengan bantuan library mysql.connector dan ditampilkan dengan library tkinter. Metode pengaksesan database tersebut dibedakan menjadi 2 yaitu dengan cara Monolitik dan Layered

## 1. Monolitik

Pembuatan kode program dalam 1 file

```
1 import tkinter as tk
2 from tkinter import messagebox, ttk
3 import mysql.connector
4
5 # Function to establish the database connection
6 def establish_db_connection():
7     try:
8         conn = mysql.connector.connect(
9             host="localhost",      # Alamat server database (gunakan 'localhost' jika ada di komputer lokal)
10            user="root",           # Username MySQL
11            password="",           # Password MySQL (kosongkan jika tidak ada password)
12            database="alproii"     # Nama database yang ingin digunakan
13        )
14        return conn # Mengembalikan objek koneksi jika berhasil
15    except mysql.connector.Error as err:
16        # Menampilkan pesan kesalahan jika koneksi gagal
17        messagebox.showerror("Database Error", f"Error: {err}")
18        return None # Mengembalikan None jika koneksi gagal
19
20 # Function untuk menambahkan data ke tabel kuis_alpro
21 def add_user():
22     nim = nim_entry.get()         # Mendapatkan input NIM dari entry field
23     nama = nama_entry.get()       # Mendapatkan input Nama dari entry field
24     nilai_kuis = nilai_kuis_entry.get() # Mendapatkan input Nilai Kuis dari entry field
25
26     # Validasi input: memastikan NIM, Nama terisi dan Nilai Kuis adalah angka
27     if nim and nama and nilai_kuis.isdigit():
28         conn = establish_db_connection() # Membuka koneksi ke database
29         if conn:
30             cursor = conn.cursor() # Membuat cursor untuk menjalankan query SQL
31             # Menjalankan query INSERT untuk menambahkan data ke tabel 'kuis_alpro'
32             cursor.execute("INSERT INTO kuis_alpro (Nim, Nama, `Nilai Kuis`) VALUES (%s, %s, %s)"
33                             , (nim, nama, int(nilai_kuis)))
34             conn.commit() # Menyimpan perubahan ke database
35             cursor.close() # Menutup cursor
36             conn.close()  # Menutup koneksi
37
38             # Menghapus input field setelah data berhasil ditambahkan
39             nim_entry.delete(0, tk.END)
40             nama_entry.delete(0, tk.END)
41             nilai_kuis_entry.delete(0, tk.END)
42
43             # Memperbarui tampilan pengguna
44             show_users()
45         else:
46             # Menampilkan pesan peringatan jika input tidak valid
47             messagebox.showwarning("Input Error", "Harap masukkan NIM, Nama, dan Nilai Kuis yang valid")
```

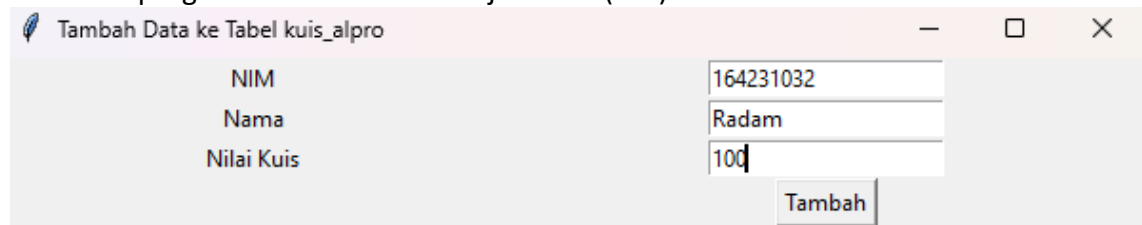


```

48
49 # Function untuk menampilkan data dari tabel kuis_alpro ke Treeview
50 def show_users():
51     conn = establish_db_connection() # Membuka koneksi ke database
52     if conn:
53         cursor = conn.cursor() # Membuat cursor untuk menjalankan query SQL
54         cursor.execute("SELECT * FROM kuis_alpro") # Mengambil semua data dari tabel 'kuis_alpro'
55         records = cursor.fetchall() # Menyimpan hasil query ke dalam variabel 'records'
56
57         # Membersihkan Treeview sebelum menampilkan data baru
58         for row in tree.get_children():
59             tree.delete(row)
60
61         # Menambahkan data ke dalam Treeview
62         for record in records:
63             tree.insert("", tk.END, values=record)
64
65         cursor.close() # Menutup cursor
66         conn.close() # Menutup koneksi setelah selesai
67
68 # Membuat window utama
69 root = tk.Tk()
70 root.title("Tambah Data ke Tabel kuis_alpro")
71
72 # Label dan Entry untuk NIM
73 nim_label = tk.Label(root, text="NIM:")
74 nim_label.pack()
75
76 nim_entry = tk.Entry(root)
77 nim_entry.pack()
78
79 # Label dan Entry untuk Nama
80 nama_label = tk.Label(root, text="Nama:")
81 nama_label.pack()
82
83 nama_entry = tk.Entry(root)
84 nama_entry.pack()
85
86 # Label dan Entry untuk Nilai Kuis
87 nilai_kuis_label = tk.Label(root, text="Nilai Kuis:")
88 nilai_kuis_label.pack()
89
90 nilai_kuis_entry = tk.Entry(root)
91 nilai_kuis_entry.pack()
92
93 # Tombol untuk menambah data ke tabel kuis_alpro
94 add_button = tk.Button(root, text="Tambah Data", command=add_user)
95 add_button.pack()
96
97 # Treeview untuk menampilkan data dari tabel kuis_alpro
98 tree = ttk.Treeview(root, columns=("NIM", "Nama", "Nilai Kuis"), show='headings')
99 tree.heading("NIM", text="NIM")
100 tree.heading("Nama", text="Nama")
101 tree.heading("Nilai Kuis", text="Nilai Kuis")
102 tree.pack()
103
104 # Memanggil fungsi untuk menampilkan data ketika aplikasi dimulai
105 show_users()
106
107 # Menjalankan loop utama untuk GUI
108 root.mainloop()

```

Hasil dari program tersebut ketika di jalankan (run) adalah:



NIM	Nama	Nilai Kuis
164231032	Radam	100

Kita bisa menginputkan NIM, Nama, dan Nilai Kuis karena memang programnya adalah mengakses tabel kuis\_alpro. Kemudian inputan tersebut akan ditampilkan setelah diinputkan

## 2. Layered

Pada versi layered, kode program dipisah menjadi 4 file sesuai dengan fungsinya masing masing, dan hasil dari programnya akan tetap sama dengan yang monolitik. File ini juga harus berada dalam 1 folder yang sama.

a. File **“database.py”** berisi kode program untuk berinteraksi dengan database

```
1 import mysql.connector
2 from tkinter import messagebox
3
4 # Fungsi untuk mengoneksikan ke database
5 def establish_db_connection():
6     try:
7         # Membuat koneksi ke database MySQL dengan parameter host, user, password, dan database
8         conn = mysql.connector.connect(
9             host="localhost", # Alamat server database (localhost jika berada di komputer lokal)
10            user="root",       # Username MySQL
11            password="",       # Password MySQL (dikosongkan jika tidak ada password)
12            database="alproii" # Nama database yang ingin diakses
13        )
14        return conn # Mengembalikan objek koneksi
15    except mysql.connector.Error as err:
16        # Menampilkan pesan kesalahan jika terjadi error saat koneksi
17        messagebox.showerror("Database Error", f"Error: {err}")
18        return None # Mengembalikan None jika koneksi gagal
19
20 # Fungsi untuk menginput data ke database
21 def insert_user(nim, nama, nilai_kuis):
22     conn = establish_db_connection() # Membuka koneksi ke database
23     if conn:
24         cursor = conn.cursor() # Membuat cursor untuk menjalankan query SQL
25         # Menjalankan query INSERT untuk memasukkan data ke dalam tabel 'kuis_alpro'
26         cursor.execute("INSERT INTO kuis_alpro (Nim, Nama, `Nilai Kuis`) VALUES (%s, %s, %s)",
27             (nim, nama, int(nilai_kuis)))
28         conn.commit() # Menyimpan perubahan di database
29         cursor.close() # Menutup cursor
30         conn.close() # Menutup koneksi
```

```

31
32 # Fungsi untuk menampilkan inputan data dari database
33 def fetch_all_users():
34     conn = establish_db_connection() # Membuka koneksi ke database
35     if conn:
36         cursor = conn.cursor() # Membuat cursor untuk menjalankan query SQL
37         cursor.execute("SELECT * FROM kuis_alpro") # Mengambil semua data dari tabel 'kuis_al
38         records = cursor.fetchall() # Menyimpan hasil query ke dalam variabel 'records'
39         cursor.close() # Menutup cursor
40         conn.close() # Menutup koneksi
41         return records # Mengembalikan hasil query sebagai list
42     return [] # Mengembalikan list kosong jika koneksi gagal

```

b. File “**service.py**” berisi kode program untuk logika program yang dijalankan

```

1 from database import insert_user, fetch_all_users # memanggil file database.py
2 from tkinter import messagebox
3
4 # Fungsi untuk menambahkan inputan dengan Validasi
5 def add_user(nim, nama, nilai_kuis):
6     # Validasi input: memastikan bahwa NIM dan Nama terisi, dan Nilai Kuis adalah angka
7     if nim and nama and nilai_kuis.isdigit():
8         insert_user(nim, nama, nilai_kuis) # Memasukkan data ke database menggunakan fungsi dari database
9         return True # Mengembalikan True jika data berhasil ditambahkan
10    else:
11        # Menampilkan pesan peringatan jika input tidak valid
12        messagebox.showwarning("Input Error", "Harap masukkan NIM, Nama, dan Nilai Kuis yang valid.")
13        return False # Mengembalikan False jika input tidak valid
14
15 # Fungsi untuk menampilkan semua inputan
16 def get_all_users():
17     return fetch_all_users() # Mengambil semua data pengguna dari database

```

c. File “**gui.py**” berisi kode program untuk membuat tampilan interface pengguna (GUI)

```

1 import tkinter as tk
2 from tkinter import ttk
3 from service import add_user, get_all_users
4
5 # Fungsi untuk memperbarui data yang telah diinputkan
6 def show_users(tree):
7     records = get_all_users() # Mengambil data pengguna dari database
8     # Menghapus data lama dari Treeview sebelum menampilkan data baru
9     for row in tree.get_children():
10         tree.delete(row)
11
12     # Menambahkan data pengguna ke Treeview
13     for record in records:
14         tree.insert("", tk.END, values=record)
15
16 # Fungsi untuk menangani jika ada inputan
17 def handle_add_user(nim_entry, nama_entry, nilai_kuis_entry, tree):
18     nim = nim_entry.get() # Mendapatkan input NIM dari Entry
19     nama = nama_entry.get() # Mendapatkan input Nama dari Entry
20     nilai_kuis = nilai_kuis_entry.get() # Mendapatkan input Nilai Kuis dari Entry
21
22     # Menambahkan pengguna dan membersihkan input jika berhasil
23     if add_user(nim, nama, nilai_kuis):
24         nim_entry.delete(0, tk.END)
25         nama_entry.delete(0, tk.END)
26         nilai_kuis_entry.delete(0, tk.END)
27         show_users(tree) # Memperbarui Treeview dengan data terbaru

```

```

29 # Fungsi untuk membuat tampilan GUI
30 def create_gui():
31     root = tk.Tk()
32     root.title("Tambah Data ke Tabel kuis_alpro")
33
34     # NIM Entry
35     nim_label = tk.Label(root, text="NIM:")
36     nim_label.pack()
37     nim_entry = tk.Entry(root)
38     nim_entry.pack()
39
40     # Nama Entry
41     nama_label = tk.Label(root, text="Nama:")
42     nama_label.pack()
43     nama_entry = tk.Entry(root)
44     nama_entry.pack()
45
46     # Nilai Kuis Entry
47     nilai_kuis_label = tk.Label(root, text="Nilai Kuis:")
48     nilai_kuis_label.pack()
49     nilai_kuis_entry = tk.Entry(root)
50     nilai_kuis_entry.pack()
51
52     # Add Button
53     add_button = tk.Button(root, text="Tambah Data",
54                             command=lambda: handle_add_user(nim_entry, nama_entry, nilai_kuis_entry, tree))
55     add_button.pack()
56
57     # Treeview for displaying data
58     tree = ttk.Treeview(root, columns=("NIM", "Nama", "Nilai Kuis"), show='headings')
59     tree.heading("NIM", text="NIM")
60     tree.heading("Nama", text="Nama")
61     tree.heading("Nilai Kuis", text="Nilai Kuis")
62     tree.pack()
63
64     # Show users on startup
65     show_users(tree)
66
67     root.mainloop() # Menjalankan loop utama untuk GUI

```

- d. File **"main.py"** berisi kode program untuk menjalankan keseluruhan program (run kode di file ini)

```

1 from gui import create_gui
2
3 if __name__ == "__main__":
4     create_gui() # Memanggil fungsi untuk membuat dan menjalankan GUI

```

Ketika file ini dijalankan, akan menampilkan tampilan GUI seperti gambar berikut:

Tambah Data ke Tabel kuis\_alpro

NIM:  
164231036

Nama:  
Malfa

Nilai Kuis:  
100

Tambah Data

NIM	Nama	Nilai Kuis
164231032	Radam	100

Kemudian jika telah berhasil menambahkan data maka akan tampil updatenya seperti gambar berikut

Tambah Data ke Tabel kuis\_alpro

NIM:

Nama:

Nilai Kuis:

Tambah Data

NIM	Nama	Nilai Kuis
164231032	Radam	100
164231036	Malfa	100