

KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya, sehingga kami dapat menyelesaikan modul praktikum Pemrograman Berorientasi Objek dengan menggunakan bahasa pemrograman Python.

Modul praktikum ini disusun untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek di Universitas Muhammadiyah Cirebon. Modul ini berisi materi-materi praktikum pemrograman berorientasi objek dengan menggunakan bahasa pemrograman Python.

Modul ini disusun dengan tujuan untuk memberikan pemahaman yang komprehensif tentang pemrograman berorientasi objek dengan menggunakan bahasa pemrograman Python. Modul ini juga diharapkan dapat menjadi panduan bagi mahasiswa dalam mengerjakan tugas-tugas praktikum pemrograman berorientasi objek.

Kami menyadari bahwa modul ini masih jauh dari sempurna. Oleh karena itu, kami mengharapkan kritik dan saran yang membangun dari para pembaca. Semoga modul ini dapat bermanfaat bagi mahasiswa dan para pembaca lainnya.

Cirebon, Januari 2024

DAFTAR ISI

[**KATA PENGANTAR ii**](#_Toc157010174)

[**DAFTAR ISI iii**](#_Toc157010175)

[**PENDAHULUAN iv**](#_Toc157010176)

[**PERTEMUAN 1 1**](#_Toc157010177)

[**PERTEMUAN 2 2**](#_Toc157010178)

[**PERTEMUAN 3 4**](#_Toc157010179)

[**PERTEMUAN 4 7**](#_Toc157010180)

[**PERTEMUAN 5 10**](#_Toc157010181)

[**PERTEMUAN 6 11**](#_Toc157010182)

[**PERTEMUAN 7 14**](#_Toc157010183)

[**PERTEMUAN 8 17**](#_Toc157010184)

[**PERTEMUAN 9 18**](#_Toc157010185)

[**PERTEMUAN 10 21**](#_Toc157010186)

[**PERTEMUAN 11 23**](#_Toc157010187)

[**PERTEMUAN 12 26**](#_Toc157010188)

[**PERTEMUAN 13 29**](#_Toc157010189)

PENDAHULUAN

Modul praktikum ini disusun untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek di Universitas Muhammadiyah Cirebon. Modul ini berisi materi-materi praktikum pemrograman berorientasi objek dengan menggunakan bahasa pemrograman Python.

Modul ini disusun dengan tujuan untuk memberikan pemahaman yang komprehensif tentang pemrograman berorientasi objek dengan menggunakan bahasa pemrograman Python. Modul ini juga diharapkan dapat menjadi panduan bagi mahasiswa dalam mengerjakan tugas-tugas praktikum pemrograman berorientasi objek.

1. Teori singkat

Pada setiap pertemuan, materi teori singkat akan diberikan untuk memberikan pemahaman dasar tentang materi yang akan dipraktikkan. Materi teori singkat akan disajikan dalam bentuk penjelasan singkat, gambar, atau contoh program.

1. Latihsn soal

Pada setiap pertemuan, latihan soal akan diberikan untuk menguji pemahaman mahasiswa tentang materi yang telah dipelajari. Latihan soal akan disajikan dalam bentuk soal pilihan ganda, soal uraian, atau soal praktikum.

1. Tugas

Pada setiap pertemuan, tugas akan diberikan untuk mengaplikasikan materi yang telah dipelajari. Tugas dapat berupa program dan laporan

PERTEMUAN 1

1. Pengantar PBO

Pemrograman Berorientasi Objek (PBO) adalah paradigma pemrograman yang berfokus pada objek. Objek adalah entitas yang memiliki data dan perilaku. PBO menggunakan objek untuk mewakili berbagai hal di dunia nyata, seperti manusia, hewan, benda, dan sebagainya.

PBO memiliki beberapa keuntungan, seperti:

* Meningkatkan keterbacaan dan kejelasan kode
* Meningkatkan kemudahan perawatan kode
* Meningkatkan fleksibilitas kode
* Meningkatkan reusabilitas kode

1. Pengenalan Python  
   Python adalah bahasa pemrograman tingkat tinggi yang bersifat interpreted dan berorientasi objek. Python mudah dipelajari dan digunakan, serta memiliki komunitas pengguna yang besar. Python digunakan untuk berbagai keperluan, seperti:

* Pengembangan web
* Pengembangan perangkat lunak
* Data science
* Machine learning
* Artificial intelligence

1. Instalasi python  
   Untuk menginstal Python, Anda dapat mengunjungi situs web resmi Python di:  
   https://www.python.org (https://www.python.org/).  
   Setelah mengunduh installer, Anda dapat menginstal Python dengan mengikuti petunjuk yang tertera di installer.
2. Variabel  
   Variabel adalah tempat penyimpanan data dalam program. Variabel dapat digunakan untuk menyimpan berbagai jenis data, seperti angka, teks, dan boolean.

PERTEMUAN 2

1. Materi Teori Singkat

* Variabel
* Tipe data
* Operator
* Kontrol flow
* Fungsi
* Class dan object

1. Bangun Ruang Sederhana

Bangun ruang adalah benda tiga dimensi yang memiliki panjang, lebar, dan tinggi. Beberapa contoh bangun ruang sederhana adalah balok, kubus, prisma segitiga, dan tabung.

Untuk membuat program untuk menghitung luas bangun ruang sederhana, kita dapat menggunakan konsep variabel, tipe data, operator, dan kontrol flow.

Berikut adalah contoh program untuk menghitung luas persegi panjang:

*panjang = 10*

*lebar = 5*

*luas = panjang \* lebar*

*print("Luas persegi panjang:", luas)*

Program ini akan mencetak luas persegi panjang dengan panjang 10 dan lebar 5, yaitu 50.

Anda dapat mengubah program ini untuk menghitung luas bangun ruang sederhana lainnya. Misalnya, untuk menghitung luas kubus, Anda dapat menggunakan kode berikut:

*sisi = 5*

*luas = 6 \* sisi \* sisi*

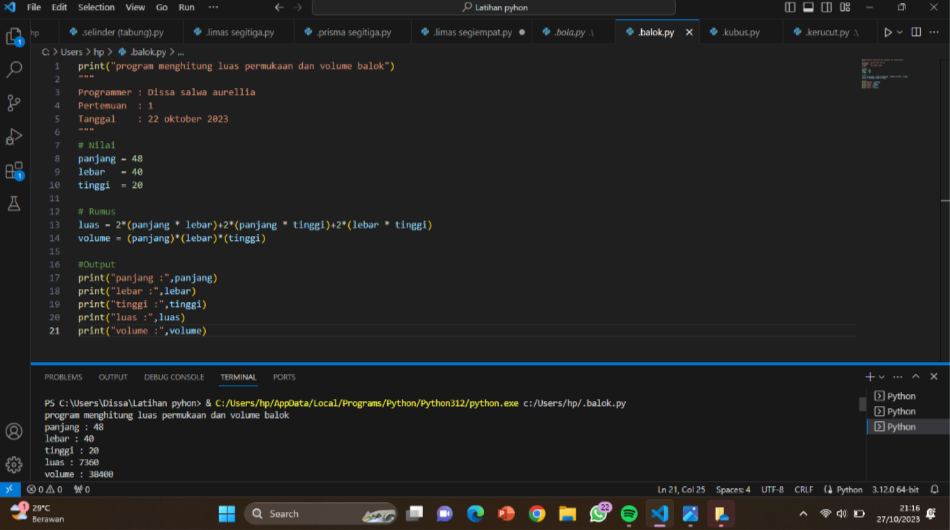
*print("Luas kubus:", luas)*

Program ini akan mencetak luas kubus dengan sisi 5, yaitu 150.

1. Latihan Soal

Berikut adalah beberapa latihan soal yang dapat Anda gunakan untuk mengasah pemahaman Anda tentang materi bangun ruang sederhana:

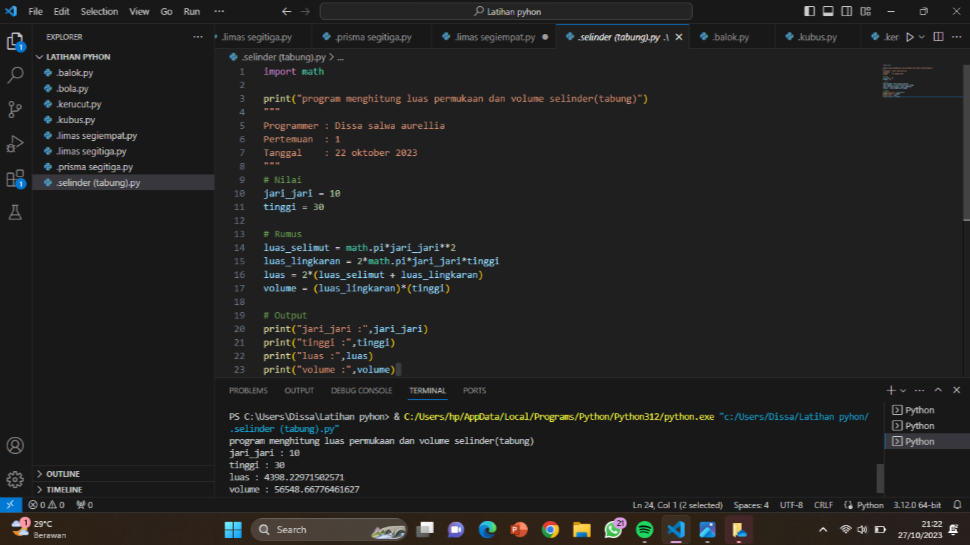
* Buat program untuk menghitung luas permukaan dan volume balok dengan panjang 48, lebar 40, dan tinggi 20



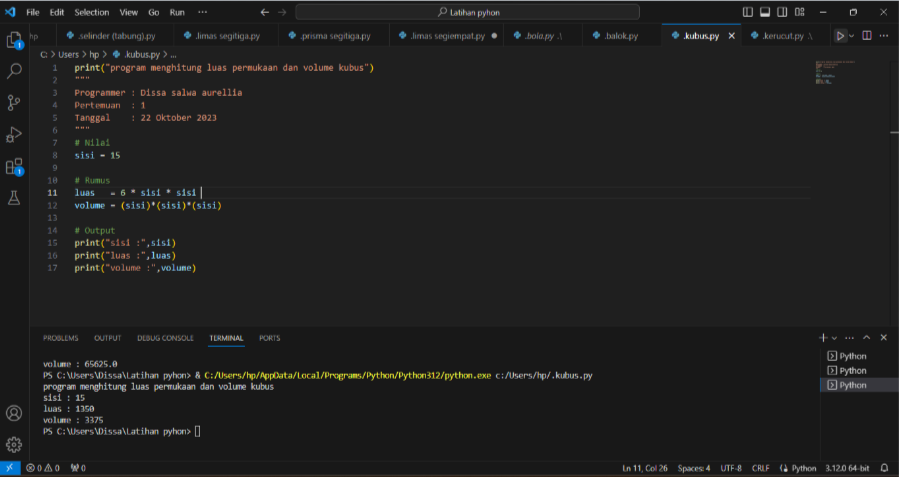
* Buat program untuk menghitung luas permukaan dan volume prisma segitiga dengan alas 70, tinggi 35, dan sisi tegak 38.



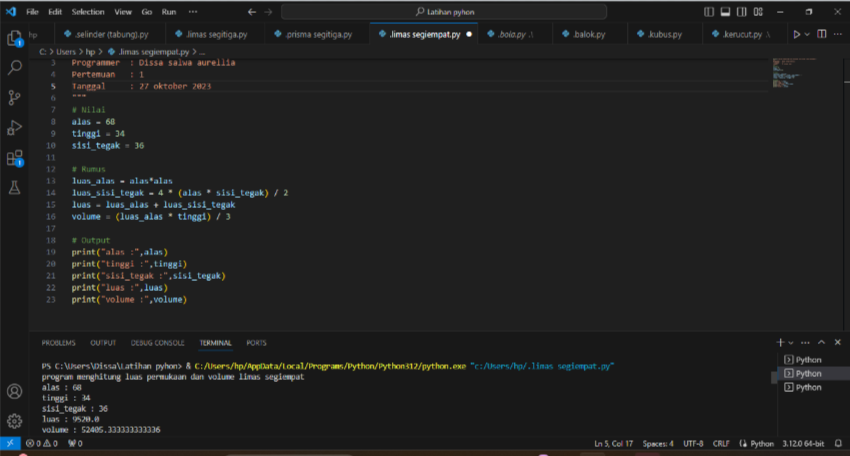
* Buat program untuk menghitung luas permukaan dan volume tabung dengan jari-jari 10 dan tinggi 30.



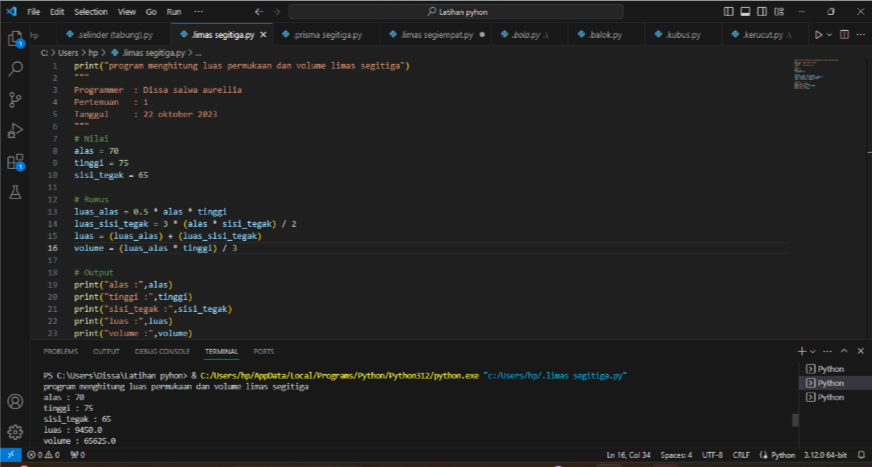
* Buat program untuk menghitung luas permukaan dan volume kubus dengan sisi 15



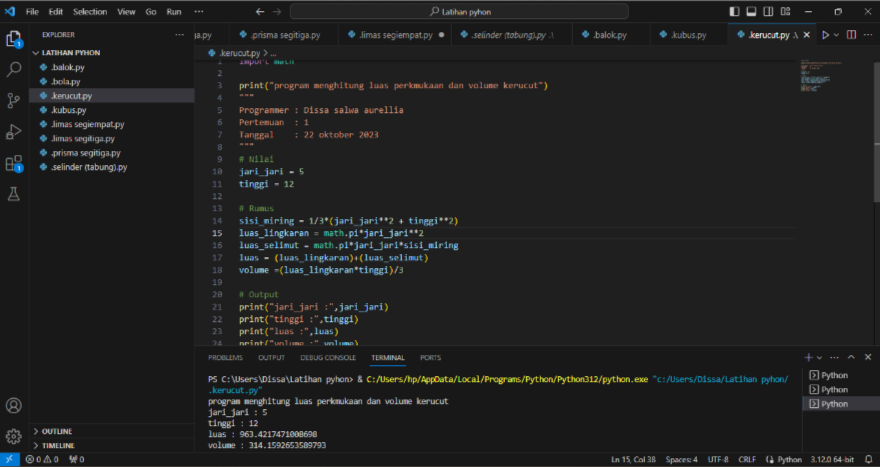
* Buat program untuk menghitung luas permukaan dan volume limas segiempat dengan alas 68, tinggi 34, dan sisi tegak 36



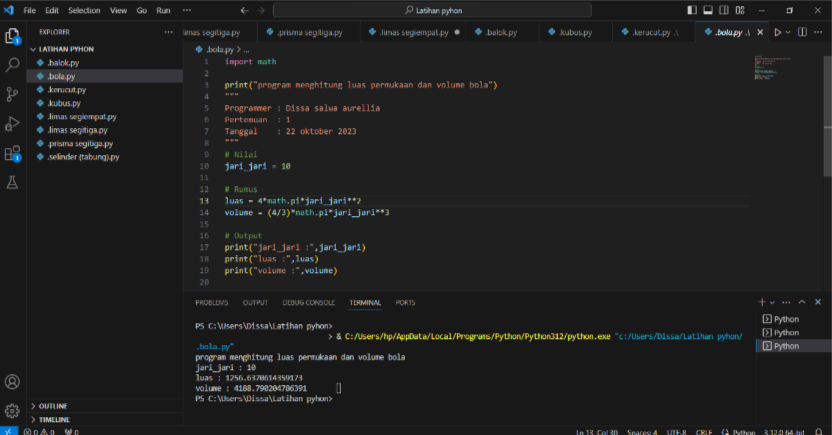
* Buat program untuk menghitung luas permukaan dan volume limas segitiga dengan alas 70, tinggi 75, dan sisi tegak 65



* Buat program untuk menghitung luas permukaan dan volume kerucut dengan jari-jari 5 dan tinggi 12.



* Buat program untuk menghitung luas permukaan dan volume bola dengan jari-jari 10.



PERTEMUAN 3

1. Variabel

Variabel adalah tempat penyimpanan data dalam program. Variabel dapat digunakan untuk menyimpan berbagai jenis data, seperti angka, teks, dan boolean.

Untuk mendeklarasikan variabel, Anda dapat menggunakan tanda sama dengan (=).

1. Tipe data

Python memiliki berbagai tipe data, seperti:

* Angka
* Teks
* Boolean
* List
* Tuple
* Dictionary
* Set

Untuk mengetahui tipe data dari suatu variabel, Anda dapat menggunakan fungsi *type().*

1. Tkinter

Tkinter adalah modul Python yang digunakan untuk membuat antarmuka pengguna grafis (GUI).

Tkinter menyediakan berbagai widget yang dapat digunakan untuk membuat berbagai elemen GUI, seperti label, button, dan entry.

Untuk menginstal Tkinter, Anda dapat menggunakan perintah berikut:

*pip install tkinter*

Aplikasi Tkinter untuk menghitung bangun ruang

Untuk membuat aplikasi Tkinter untuk menghitung bangun ruang, kita dapat menggunakan konsep variabel, tipe data, dan widget Tkinter.

Berikut adalah contoh program untuk membuat aplikasi Tkinter untuk menghitung luas persegi panjang:

*import tkinter as tk*

*# Deklarasi variabel*

*panjang = tk.StringVar()*

*lebar = tk.StringVar()*

*luas = tk.StringVar()*

*# Buat window*

*window = tk.Tk()*

*# Buat label*

*label\_panjang = tk.Label(window, text="Panjang:")*

*label\_lebar = tk.Label(window, text="Lebar:")*

*label\_luas = tk.Label(window, text="Luas:")*

*# Buat entry*

*entry\_panjang = tk.Entry(window, textvariable=panjang)*

*entry\_lebar = tk.Entry(window, textvariable=lebar)*

*entry\_luas = tk.Entry(window, textvariable=luas)*

*# Buat button*

*button\_hitung = tk.Button(window, text="Hitung")*

*# Buat fungsi untuk menghitung luas*

*def hitung():*

*panjang = float(entry\_panjang.get())*

*lebar = float(entry\_lebar.get())*

*luas.set(panjang \* lebar)*

*# Tampilkan label dan entry*

*label\_panjang.grid(row=0, column=0)*

*entry\_panjang.grid(row=0, column=1)*

*label\_lebar.grid(row=1, column=0)*

*entry\_lebar.grid(row=1, column=1)*

*label\_luas.grid(row=2, column=0)*

*entry\_luas.grid(row=2, column=1)*

*# Tampilkan button*

*button\_hitung.grid(row=3, column=0)*

*# Tampilkan window*

*window.mainloop()*

Program ini akan membuat window dengan beberapa label, entry, dan button. Label digunakan untuk menampilkan teks, entry digunakan untuk memasukkan data, dan button digunakan untuk menjalankan fungsi.

Fungsi hitung() digunakan untuk menghitung luas persegi panjang. Fungsi ini menerima dua parameter, yaitu panjang dan lebar. Nilai panjang dan lebar diperoleh dari nilai yang dimasukkan ke dalam entry entry\_panjang dan entry\_lebar.

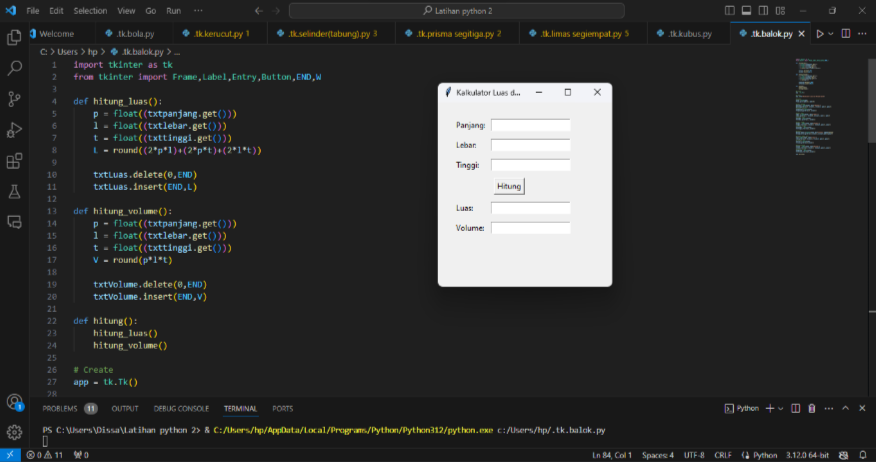
Setelah luas dihitung, nilainya ditampilkan ke dalam entry entry\_luas.

Anda dapat mengubah program ini untuk menghitung luas bangun ruang lainnya, seperti luas kubus, luas prisma segitiga, dan luas tabung.

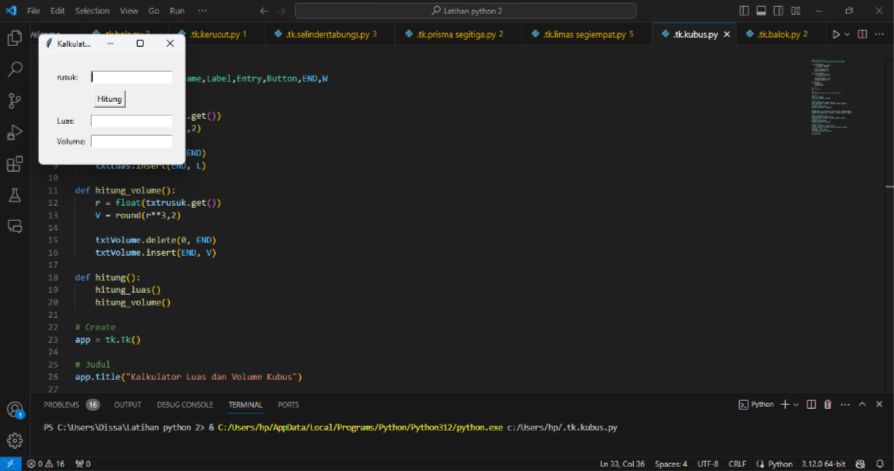
1. Latihan Soal

Berikut adalah beberapa latihan soal yang dapat Anda gunakan untuk mengasah pemahaman Anda tentang materi aplikasi Tkinter untuk menghitung bangun ruang:

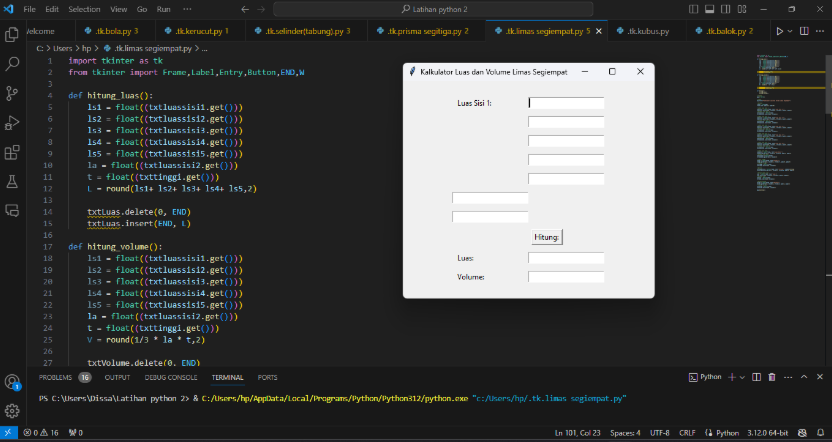
* Buat aplikasi Tkinter untuk menghitung luas balok.



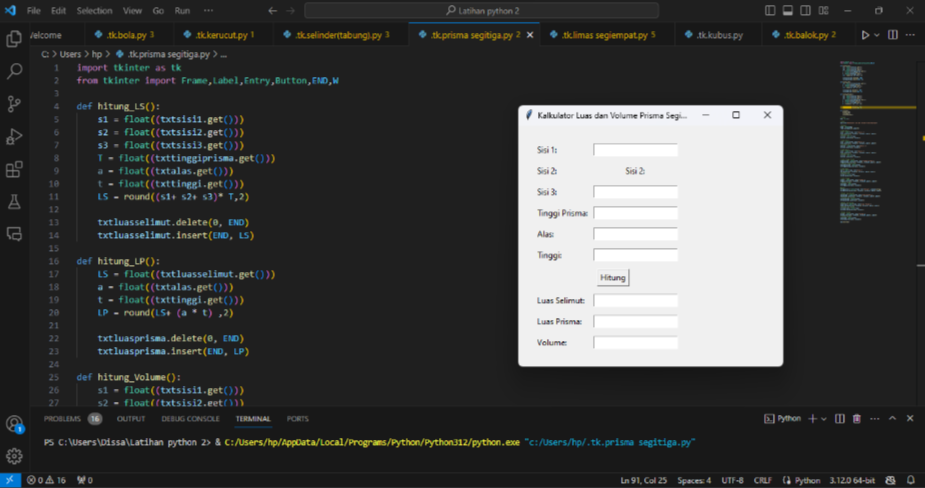
* Buat aplikasi Tkinter untuk menghitung luas kubus.



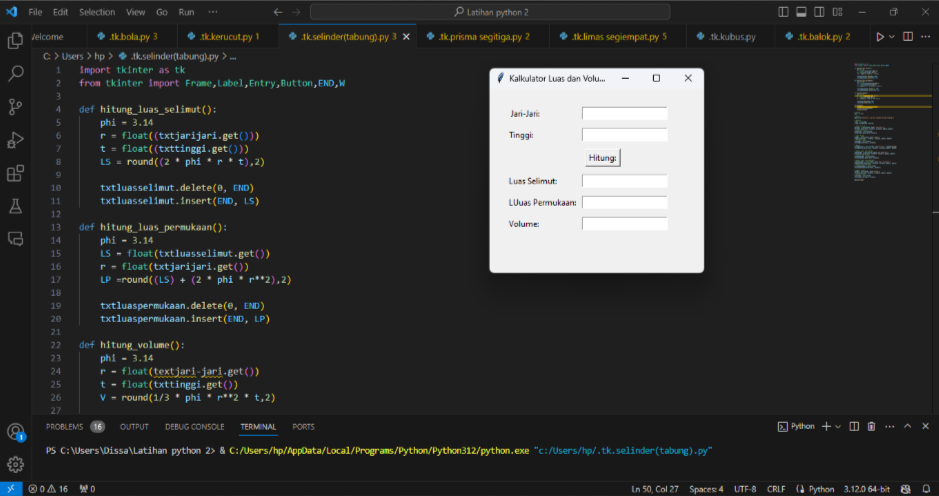
* Buat aplikasi Tkinter untuk menghitung luas limas segiempat.



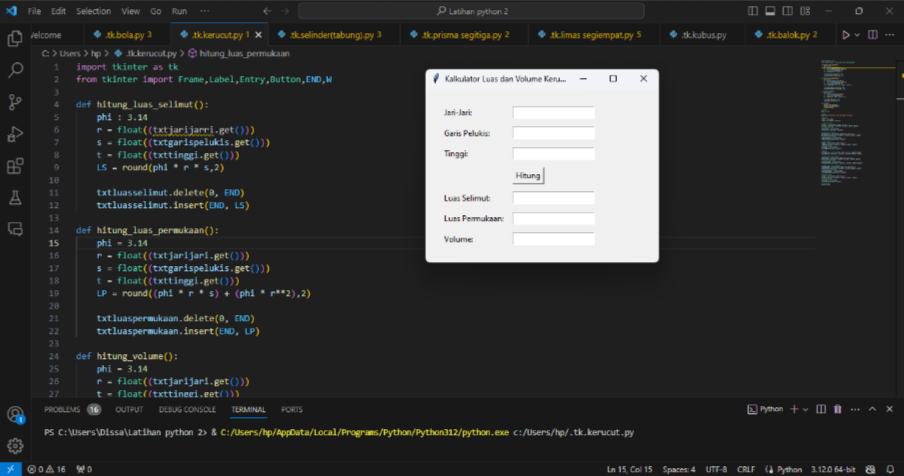
* Buat aplikasi Tkinter untuk menghitung luas prisma segitiga.



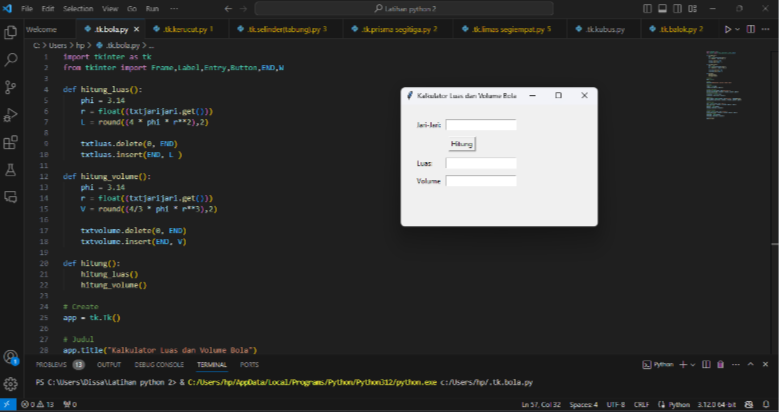
* Buat aplikasi Tkinter untuk menghitung luas tabung.



* Buat aplikasi Tkinter untuk menghitung luas kerucut.



* Buat aplikasi Tkinter untuk menghitung luas bola.



PERTEMUAN 4

1. Struktur kontrol flow

Struktur kontrol flow adalah struktur yang digunakan untuk mengendalikan aliran program. Python memiliki berbagai struktur kontrol flow, seperti:

1. Struktur if

Struktur if digunakan untuk memeriksa kondisi tertentu. Jika kondisi tersebut terpenuhi, maka blok kode yang berada di bawah struktur if akan dijalankan.

1. Struktur else

Struktur else digunakan untuk menjalankan blok kode jika kondisi yang diperiksa oleh struktur if tidak terpenuhi.

1. Struktur elif

Struktur elif digunakan untuk memeriksa kondisi lain jika kondisi yang diperiksa oleh struktur if tidak terpenuhi.

1. Struktur for

Struktur for digunakan untuk mengulang blok kode beberapa kali.

1. Struktur while

Struktur while digunakan untuk mengulang blok kode selama kondisi tertentu terpenuhi.

1. Aplikasi Tkinter MP3 player

Untuk membuat aplikasi Tkinter MP3 player, kita dapat menggunakan konsep kontrol flow, widget Tkinter, dan modul Pygame.

Berikut adalah contoh program untuk membuat aplikasi Tkinter MP3 player:

*import tkinter as tk*

*import pygame*

*# Deklarasi variabel*

*musik = None*

*# Buat window*

*window = tk.Tk()*

*# Buat label*

*label\_judul = tk.Label(window, text="Judul:")*

*label\_penyanyi = tk.Label(window, text="Penyanyi:")*

*label\_status = tk.Label(window, text="Status:")*

*# Buat entry*

*entry\_judul = tk.Entry(window)*

*entry\_penyanyi = tk.Entry(window)*

*# Buat button*

*button\_play = tk.Button(window, text="Play")*

*button\_pause = tk.Button(window, text="Pause")*

*button\_stop = tk.Button(window, text="Stop")*

*# Buat fungsi untuk memutar musik*

*def play():*

*global musik*

*# Membaca musik dari file*

*musik = pygame.mixer.Sound("musik.mp3")*

*# Memutar musik*

*musik.play()*

*# Menampilkan status "Play"*

*label\_status.config(text="Play")*

*# Buat fungsi untuk menjeda musik*

*def pause():*

*global musik*

*# Menjeda musik*

*musik.pause()*

*# Menampilkan status "Pause"*

*label\_status.config(text="Pause")*

*# Buat fungsi untuk menghentikan musik*

*def stop():*

*global musik*

*# Menghentikan musik*

*musik.stop()*

*# Menampilkan status "Stop"*

*label\_status.config(text="Stop")*

*# Tampilkan label dan entry*

*label\_judul.grid(row=0, column=0)*

*entry\_judul.grid(row=0, column=1)*

*label\_penyanyi.grid(row=1, column=0)*

*entry\_penyanyi.grid(row=1, column=1)*

*label\_status.grid(row=2, column=0)*

*# Tampilkan button*

*button\_play.grid(row=3, column=0)*

*button\_pause.grid(row=4, column=0)*

*button\_stop.grid(row=5, column=0)*

*# Tampilkan window*

*window.mainloop()*

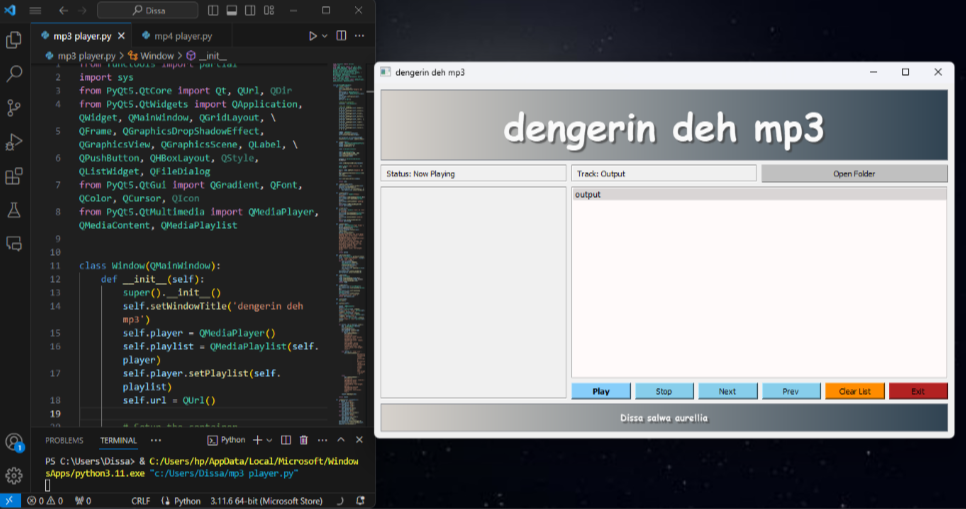
Program ini akan membuat window dengan beberapa label, entry, dan button. Label digunakan untuk menampilkan teks, entry digunakan untuk memasukkan data, dan button digunakan untuk menjalankan fungsi.

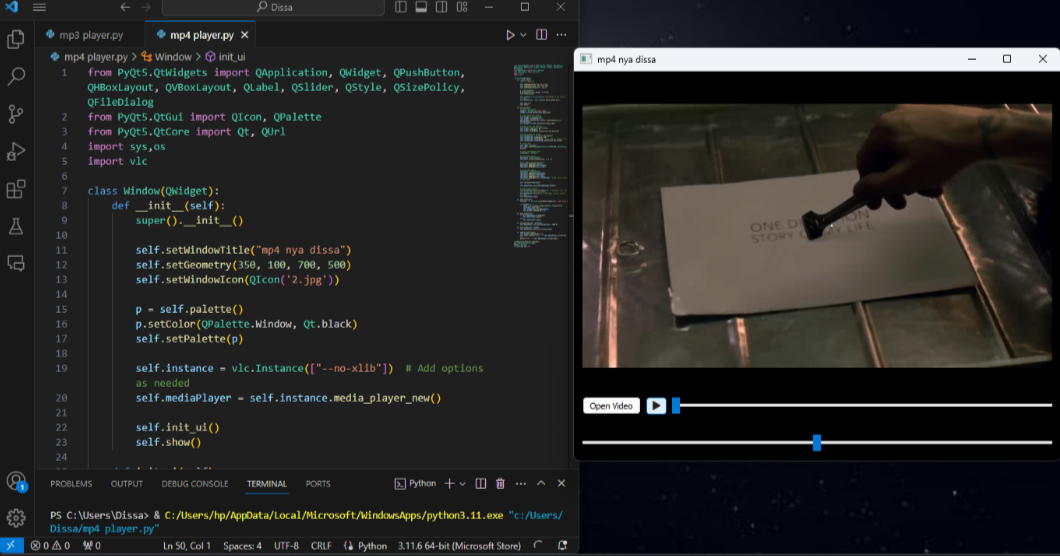
Fungsi *play()* digunakan untuk memutar musik. Fungsi ini membaca musik dari file musik.mp3 dan memainkannya.

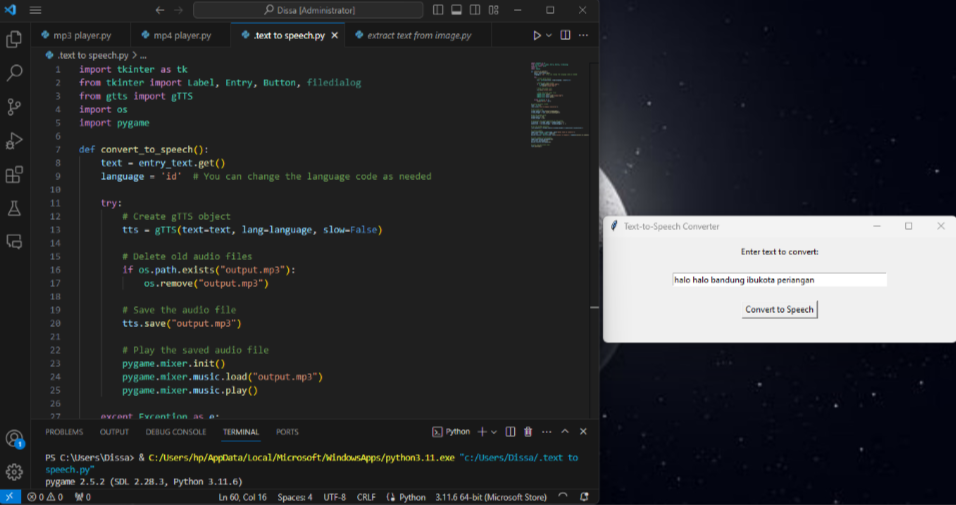
Fungsi *pause()* digunakan untuk menjeda musik. Fungsi ini menjeda musik yang sedang diputar.

Fungsi *stop()* digunakan untuk menghentikan musik. Fungsi ini menghentikan musik yang sedang diputar.

Contoh :







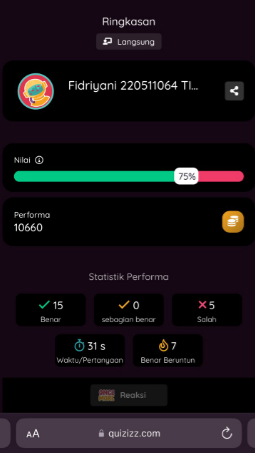
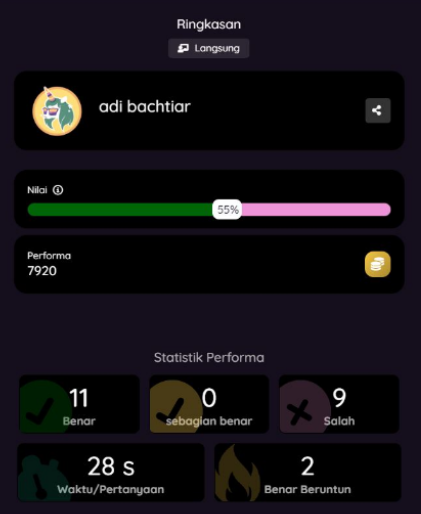
PERTEMUAN 5

1. Kuis

Kuis dirancang untuk mendukung pemahaman lebih lanjut setelah mengikuti pertemuan 1 sampai 4, tentunya materi materi pada kuis mencakup pertemuan 1 sampai 4, berikut contoh soal yang ada pada kuis:

* Implementasikan loop for dalam Python untuk menampilkan angka 1 hingga 5.
* Jelaskan perbedaan antara list dan tuple dalam Python. Berikan contoh penggunaan keduanya.
* Apa perbedaan antara pernyataan if, elif, dan else dalam struktur kontrol alur (control flow) Python?

Contoh :



PERTEMUAN 6

1. Aplikasi Tkinter berupa kalkulator

Untuk membuat aplikasi Tkinter berupa kalkulator, kita dapat menggunakan konsep variabel, tipe data, widget Tkinter, dan struktur kontrol flow.

Berikut adalah contoh program untuk membuat aplikasi Tkinter berupa kalkulator:

*import tkinter as tk*

*# Deklarasi variabel*

*bil1 = tk.StringVar()*

*bil2 = tk.StringVar()*

*hasil = tk.StringVar()*

*# Buat window*

*window = tk.Tk()*

*# Buat label*

*label\_bil1 = tk.Label(window, text="Bilangan 1:")*

*label\_bil2 = tk.Label(window, text="Bilangan 2:")*

*label\_hasil = tk.Label(window, text="Hasil:")*

*# Buat entry*

*entry\_bil1 = tk.Entry(window, textvariable=bil1)*

*entry\_bil2 = tk.Entry(window, textvariable=bil2)*

*entry\_hasil = tk.Entry(window, textvariable=hasil)*

*# Buat button*

*button\_tambah = tk.Button(window, text="+")*

*button\_kurang = tk.Button(window, text="-")*

*button\_kali = tk.Button(window, text="\*")*

*button\_bagi = tk.Button(window, text="/")*

*# Buat fungsi untuk menghitung*

*def hitung(operator):*

*bil1 = float(bil1.get())*

*bil2 = float(bil2.get())*

*if operator == "+":*

*hasil.set(bil1 + bil2)*

*elif operator == "-":*

*hasil.set(bil1 - bil2)*

*elif operator == "\*":*

*hasil.set(bil1 \* bil2)*

*elif operator == "/":*

*hasil.set(bil1 / bil2)*

*# Tampilkan label dan entry*

*label\_bil1.grid(row=0, column=0)*

*entry\_bil1.grid(row=0, column=1)*

*label\_bil2.grid(row=1, column=0)*

*entry\_bil2.grid(row=1, column=1)*

*label\_hasil.grid(row=2, column=0)*

*entry\_hasil.grid(row=2, column=1)*

*# Tampilkan button*

*button\_tambah.grid(row=3, column=0)*

*button\_kurang.grid(row=3, column=1)*

*button\_kali.grid(row=4, column=0)*

*button\_bagi.grid(row=4, column=1)*

*# Tampilkan event handler*

*button\_tambah.config(command=lambda: hitung("+"))*

*button\_kurang.config(command=lambda: hitung("-"))*

*button\_kali.config(command=lambda: hitung("\*"))*

*button\_bagi.config(command=lambda: hitung("/"))*

*# Tampilkan window*

*window.mainloop()*

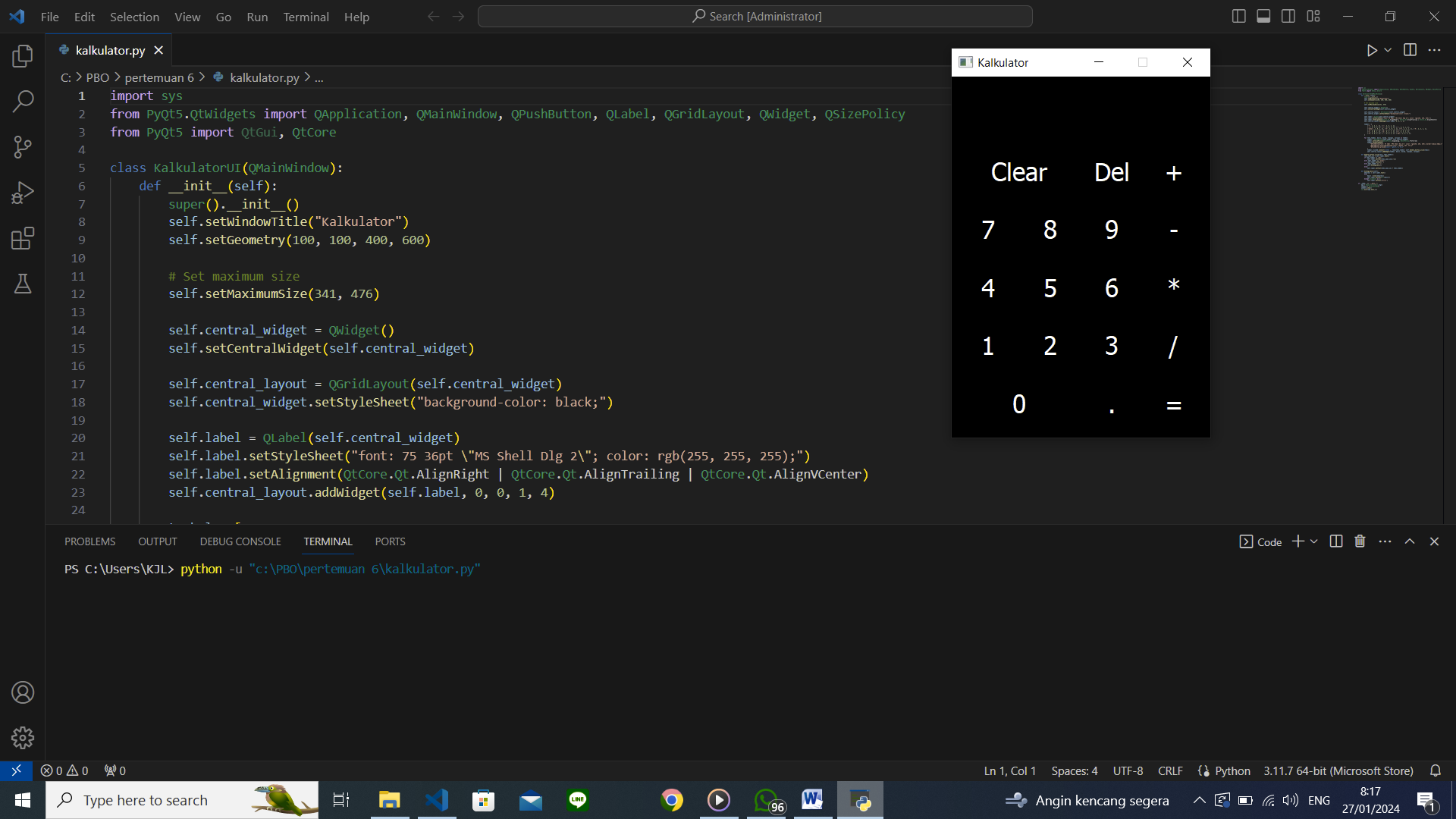
Program ini akan membuat window dengan beberapa label, entry, dan button. Label digunakan untuk menampilkan teks, entry digunakan untuk memasukkan data, dan button digunakan untuk menjalankan fungsi.

Fungsi *hitung()* digunakan untuk menghitung hasil operasi kalkulator. Fungsi ini menerima satu parameter, yaitu operator. Operator yang dapat digunakan adalah *+, -, \*,* dan */.*

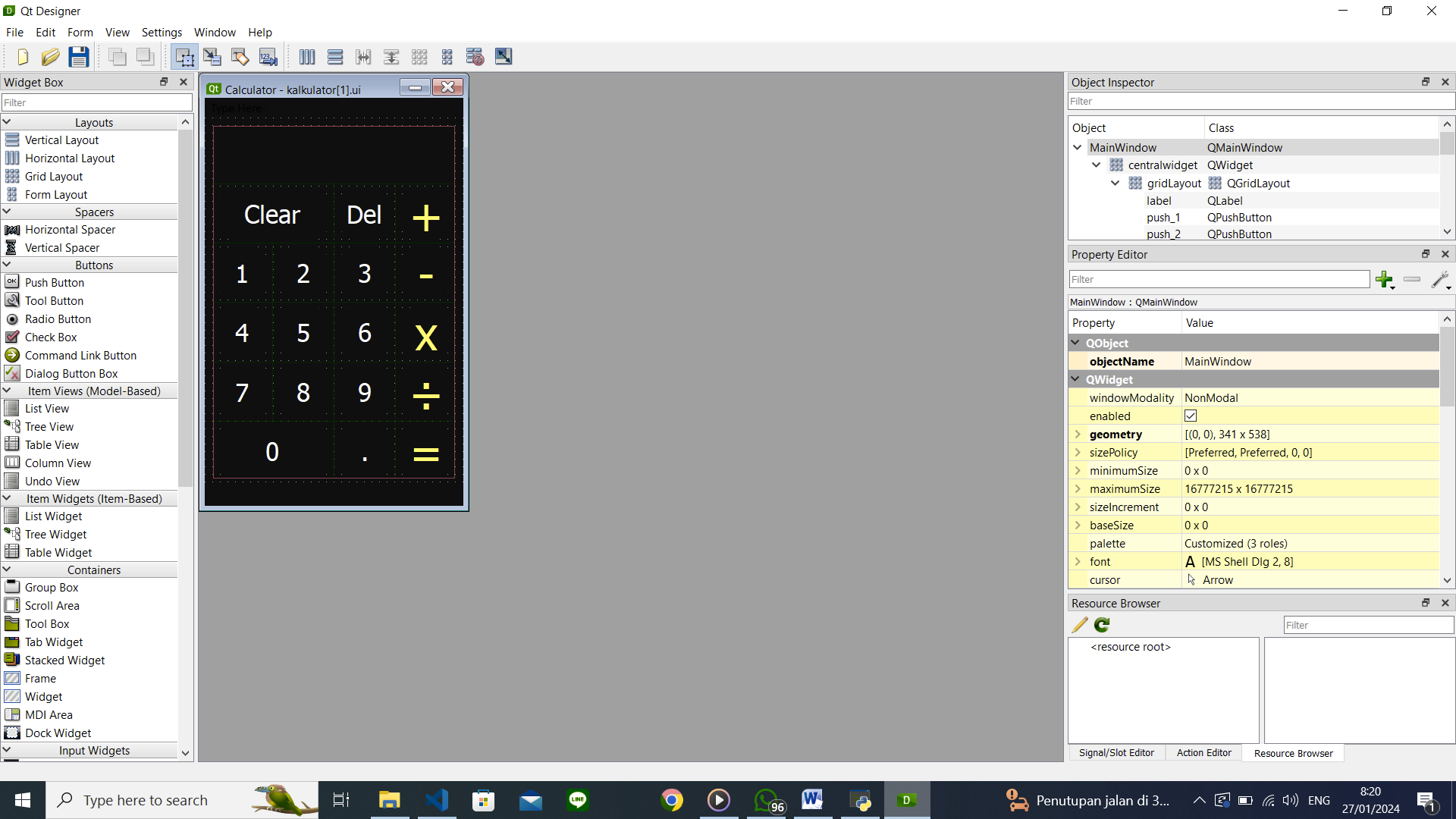
Setelah operasi selesai, hasil ditampilkan ke dalam entry *entry\_hasil.*

Anda dapat mengubah program ini untuk menambahkan fungsi-fungsi lain, seperti tombol untuk menghapus data, tombol untuk mengosongkan layar, dan sebagainya.

Contoh :



Contoh Aplikasi Kalkulator melalui Qt Designer :



PERTEMUAN 7

1. Konversi suhu

Pada pertemuan ke 7 mahasiswa ditugaskan untuk membuat aplikasi tkinter berupa aplikasi konversi suhu.

Berikut adalah formula konversi suhu yang umum digunakan:

* Celcius ke Fahrenheit: *F = (C \* 9/5) + 32*
* Fahrenheit ke Celcius: *C = (F - 32) / 9 \* 5*
* Celcius ke Kelvin: *K = C + 273.15*
* Kelvin ke Celcius: *C = K - 273.15*

1. Aplikasi Tkinter untuk menghitung konversi suhu

Untuk membuat aplikasi Tkinter untuk menghitung konversi suhu, kita dapat menggunakan konsep variabel, tipe data, widget Tkinter, dan struktur kontrol flow.

Berikut adalah contoh program untuk membuat aplikasi Tkinter untuk menghitung konversi suhu:

*import tkinter as tk*

*# Deklarasi variabel*

*suhu = tk.StringVar()*

*satuan = tk.StringVar()*

*hasil = tk.StringVar()*

*# Buat window*

*window = tk.Tk()*

*# Buat label*

*label\_suhu = tk.Label(window, text="Suhu:")*

*label\_satuan = tk.Label(window, text="Satuan:")*

*label\_hasil = tk.Label(window, text="Hasil:")*

*# Buat entry*

*entry\_suhu = tk.Entry(window, textvariable=suhu)*

*entry\_satuan = tk.Entry(window, textvariable=satuan)*

*entry\_hasil = tk.Entry(window, textvariable=hasil)*

*# Buat button*

*button\_konversi = tk.Button(window, text="Konversi")*

*# Buat fungsi untuk menghitung*

*def konversi():*

*suhu = float(suhu.get())*

*satuan = satuan.get()*

*if satuan == "C":*

*if entry\_satuan.get() == "C":*

*hasil.set(suhu)*

*elif entry\_satuan.get() == "F":*

*hasil.set((suhu \* 9/5) + 32)*

*elif entry\_satuan.get() == "K":*

*hasil.set(suhu + 273.15)*

*elif satuan == "F":*

*if entry\_satuan.get() == "C":*

*hasil.set((suhu - 32) / 9 \* 5)*

*elif entry\_satuan.get() == "F":*

*hasil.set(suhu)*

*elif entry\_satuan.get() == "K":*

*hasil.set((suhu - 32) / 9 \* 5 + 273.15)*

*elif satuan == "K":*

*if entry\_satuan.get() == "C":*

*hasil.set(suhu - 273.15)*

*elif entry\_satuan.get() == "F":*

*hasil.set((suhu - 273.15) \* 9/5 + 32)*

*elif entry\_satuan.get() == "K":*

*hasil.set(suhu)*

*# Tampilkan label dan entry*

*label\_suhu.grid(row=0, column=0)*

*entry\_suhu.grid(row=0, column=1)*

*label\_satuan.grid(row=1, column=0)*

*entry\_satuan.grid(row=1, column=1)*

*label\_hasil.grid(row=2, column=0)*

*entry\_hasil.grid(row=2, column=1)*

*# Tampilkan button*

*button\_konversi.grid(row=3, column=0)*

*# Tampilkan event handler*

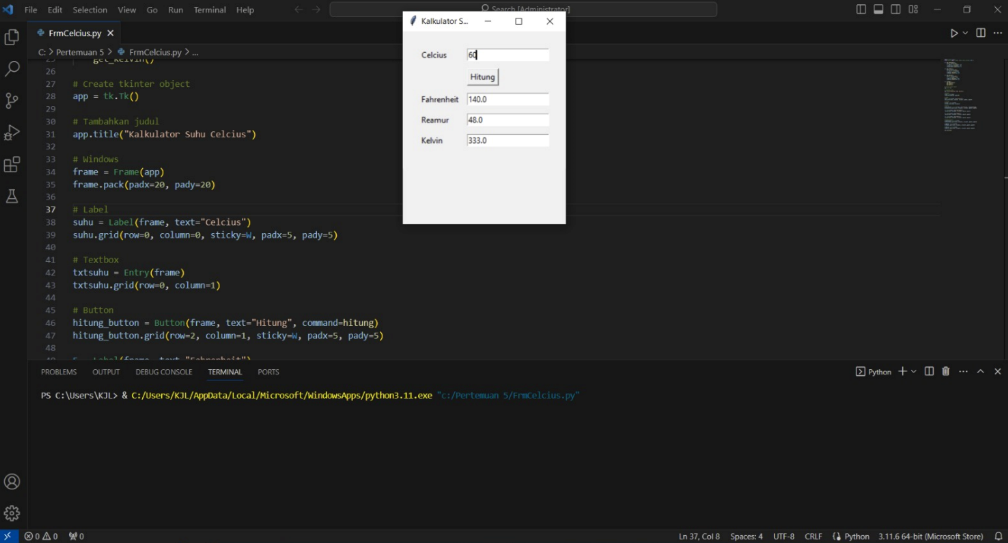
*button\_konversi.config(command=konversi)*

*# Tampilkan window*

*window.mainloop()*

Program ini akan membuat window dengan beberapa label, entry, dan button. Label digunakan untuk menampilkan teks, entry digunakan untuk memasukkan data, dan button digunakan untuk menjalankan fungsi

Contoh :



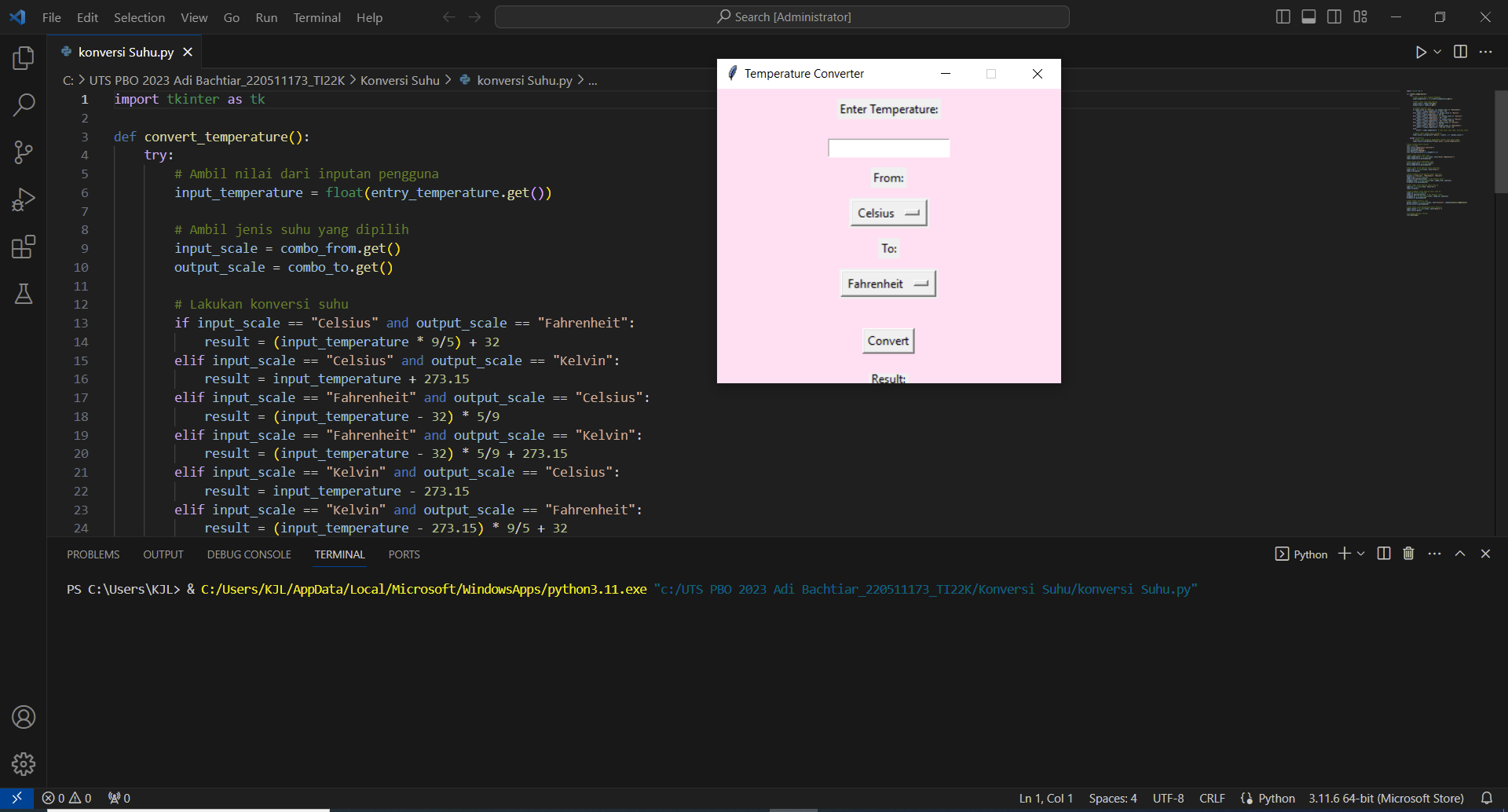
PERTEMUAN 8

1. Ulangan Tengah semester

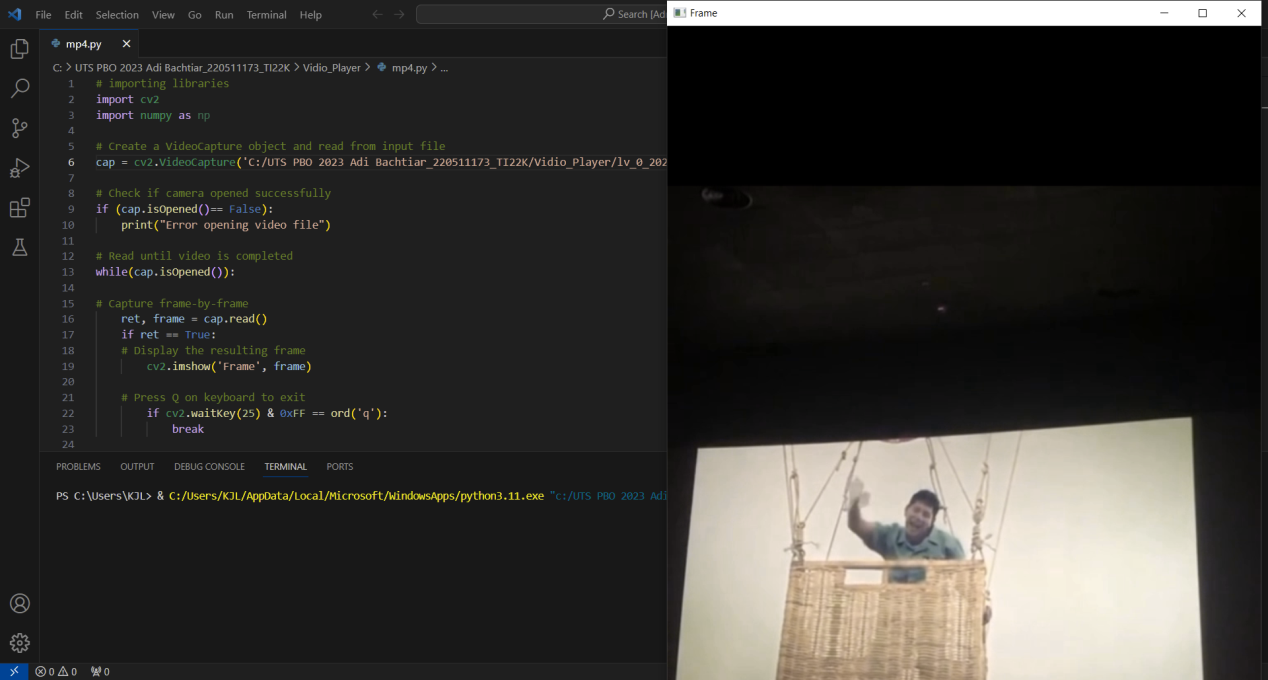
Ulangan tengah semester bertujuan untuk mengevaluasi pemahaman dan perkembangan mahasiswa dalam materi semester tersebut. Melalui ujian ini, Dosen dapat mengidentifikasi area yang perlu diperkuat dan memberikan umpan balik yang konstruktif. Bagi mahasiswa, ulangan tengah semester menjadi titik referensi penting untuk menilai tingkat pemahaman mereka dan menyesuaikan pendekatan belajar mereka ke depan. Selain itu, ujian ini juga dapat memberikan dorongan motivasi dan meningkatkan keterlibatan mahasiswa dalam proses pembelajaran.

Berikut contoh soal ulangan Tengah semester:

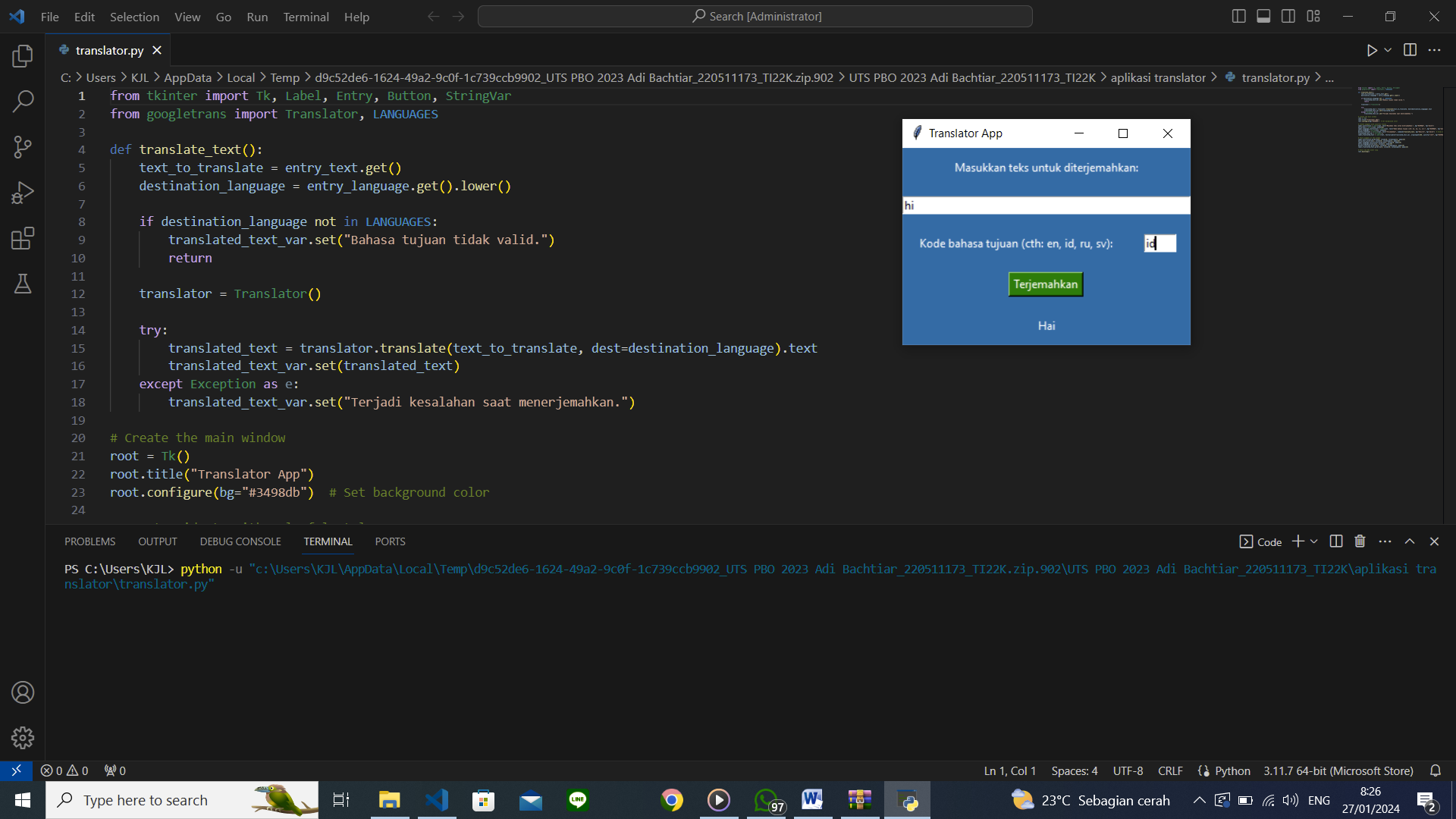
* Buatlah aplikasi konversi suhu
* import tkinter as tk
* def convert\_temperature():
* try:
* # Ambil nilai dari inputan pengguna
* input\_temperature = float(entry\_temperature.get())
* # Ambil jenis suhu yang dipilih
* input\_scale = combo\_from.get()
* output\_scale = combo\_to.get()
* # Lakukan konversi suhu
* if input\_scale == "Celsius" and output\_scale == "Fahrenheit":
* result = (input\_temperature \* 9/5) + 32
* elif input\_scale == "Celsius" and output\_scale == "Kelvin":
* result = input\_temperature + 273.15
* elif input\_scale == "Fahrenheit" and output\_scale == "Celsius":
* result = (input\_temperature - 32) \* 5/9
* elif input\_scale == "Fahrenheit" and output\_scale == "Kelvin":
* result = (input\_temperature - 32) \* 5/9 + 273.15
* elif input\_scale == "Kelvin" and output\_scale == "Celsius":
* result = input\_temperature - 273.15
* elif input\_scale == "Kelvin" and output\_scale == "Fahrenheit":
* result = (input\_temperature - 273.15) \* 9/5 + 32
* else:
* result = input\_temperature  # Jika jenis suhu sama, hasilnya tetap
* # Update label dengan hasil konversi
* label\_result.config(text=f"Result: {result:.2f} {output\_scale}")
* except ValueError:
* # Tangani jika pengguna memasukkan sesuatu yang bukan angka
* label\_result.config(text="Please enter a valid temperature")
* # Buat instance dari Tkinter
* root = tk.Tk()
* root.title("Temperature Converter")
* root.config(bg="pink")
* root.geometry("350x300")
* root.resizable(width=False,height=False)
* # Buat label untuk input suhu
* label\_temperature = tk.Label(root, text="Enter Temperature:")
* label\_temperature.pack(pady=10)
* # Buat entry untuk memasukkan suhu
* entry\_temperature = tk.Entry(root)
* entry\_temperature.pack(pady=10)
* # Buat label untuk memilih jenis suhu dari
* label\_from = tk.Label(root, text="From:")
* label\_from.pack()
* # Buat dropdown untuk memilih jenis suhu dari
* options = ["Celsius", "Fahrenheit", "Kelvin"]
* combo\_from = tk.StringVar()
* combo\_from.set(options[0])  # Set default value
* dropdown\_from = tk.OptionMenu(root, combo\_from, \*options)
* dropdown\_from.pack(pady=10)
* # Buat label untuk memilih jenis suhu ke
* label\_to = tk.Label(root, text="To:")
* label\_to.pack()
* # Buat dropdown untuk memilih jenis suhu ke
* combo\_to = tk.StringVar()
* combo\_to.set(options[1])  # Set default value
* dropdown\_to = tk.OptionMenu(root, combo\_to, \*options)
* dropdown\_to.pack(pady=10)
* # Buat tombol konversi suhu
* button\_convert = tk.Button(root, text="Convert", command=convert\_temperature)
* button\_convert.pack(pady=20)
* # Buat label untuk menampilkan hasil konversi
* label\_result = tk.Label(root, text="Result:")
* label\_result.pack()
* # Jalankan aplikasi Tkinter
* root.mainloop()
* Hasil dari codingan diatas



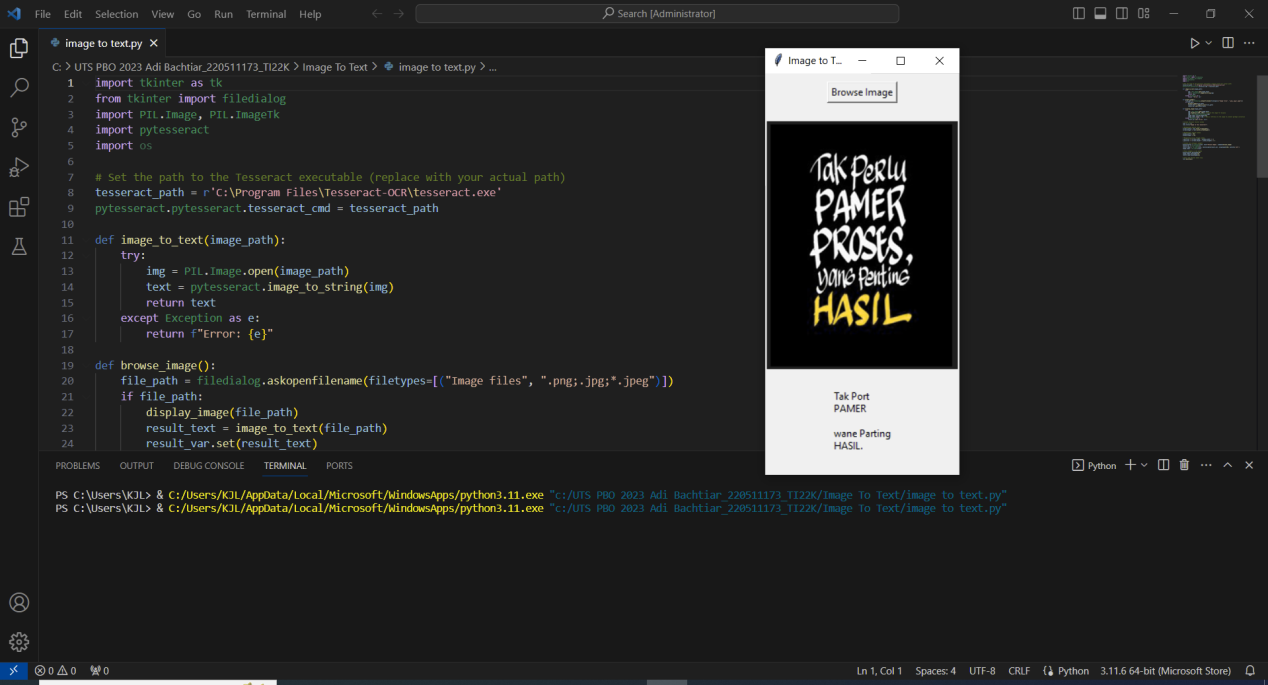
* Buatlah aplikasi mp4 player
* # importing libraries
* import cv2
* import numpy as np
* # Create a VideoCapture object and read from input file
* cap = cv2.VideoCapture('C:/UTS PBO 2023 Adi Bachtiar\_220511173\_TI22K/Vidio\_Player/lv\_0\_20220909205403.mp4')
* # Check if camera opened successfully
* if (cap.isOpened()== False):
* print("Error opening video file")
* # Read until video is completed
* while(cap.isOpened()):
* # Capture frame-by-frame
* ret, frame = cap.read()
* if ret == True:
* # Display the resulting frame
* cv2.imshow('Frame', frame)
* # Press Q on keyboard to exit
* if cv2.waitKey(25) & 0xFF == ord('q'):
* break
* # Break the loop
* else:
* break
* # When everything done, release
* # the video capture object
* cap.release()
* # Closes all the frames
* cv2.destroyAllWindows()
* Hasil Dari codingan diatas



* Buatlah aplikasi translator
* from tkinter import Tk, Label, Entry, Button, StringVar
* from googletrans import Translator, LANGUAGES
* def translate\_text():
* text\_to\_translate = entry\_text.get()
* destination\_language = entry\_language.get().lower()
* if destination\_language not in LANGUAGES:
* translated\_text\_var.set("Bahasa tujuan tidak valid.")
* return
* translator = Translator()
* try:
* translated\_text = translator.translate(text\_to\_translate, dest=destination\_language).text
* translated\_text\_var.set(translated\_text)
* except Exception as e:
* translated\_text\_var.set("Terjadi kesalahan saat menerjemahkan.")
* # Create the main window
* root = Tk()
* root.title("Translator App")
* root.configure(bg="#3498db")  # Set background color
* # Create widgets with colorful styles
* label\_instruction = Label(root, text="Masukkan teks untuk diterjemahkan:", bg="#3498db", fg="white")
* entry\_text = Entry(root, width=50)
* label\_language\_instruction = Label(root, text="Kode bahasa tujuan (cth: en, id, ru, sv):", bg="#3498db", fg="white")
* entry\_language = Entry(root, width=5)
* button\_translate = Button(root, text="Terjemahkan", command=translate\_text, bg="#2ecc71", fg="white")  # Green button
* translated\_text\_var = StringVar()
* label\_translated\_text = Label(root, textvariable=translated\_text\_var, wraplength=300, justify="left", bg="#3498db", fg="white")
* # Place widgets on the window
* label\_instruction.grid(row=0, column=0, columnspan=2, pady=10)
* entry\_text.grid(row=1, column=0, columnspan=2, pady=10)
* label\_language\_instruction.grid(row=2, column=0, pady=10)
* entry\_language.grid(row=2, column=1, pady=10)
* button\_translate.grid(row=3, column=0, columnspan=2, pady=10)
* label\_translated\_text.grid(row=4, column=0, columnspan=2, pady=10)
* # Start the GUI event loop
* root.mainloop()
* Hasil codingan diatas



* Buatlah aplikasi extract text dari gambar
* import tkinter as tk
* from tkinter import filedialog
* import PIL.Image, PIL.ImageTk
* import pytesseract
* import os
* # Set the path to the Tesseract executable (replace with your actual path)
* tesseract\_path = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
* pytesseract.pytesseract.tesseract\_cmd = tesseract\_path
* def image\_to\_text(image\_path):
* try:
* img = PIL.Image.open(image\_path)
* text = pytesseract.image\_to\_string(img)
* return text
* except Exception as e:
* return f"Error: {e}"
* def browse\_image():
* file\_path = filedialog.askopenfilename(filetypes=[("Image files", ".png;.jpg;\*.jpeg")])
* if file\_path:
* display\_image(file\_path)
* result\_text = image\_to\_text(file\_path)
* result\_var.set(result\_text)
* def display\_image(image\_path):
* try:
* img = PIL.Image.open(image\_path)
* img.thumbnail((300, 300))  # Resize the image for display
* img = PIL.ImageTk.PhotoImage(img)
* image\_label.config(image=img)
* image\_label.image = img  # Keep a reference to the image to prevent garbage collection
* except Exception as e:
* result\_var.set(f"Error: {e}")
* # Create the main Tkinter window
* root = tk.Tk()
* root.title("Image to Text Converter")
* # Mendapatkan ukuran layar
* screen\_width = root.winfo\_screenwidth()
* screen\_height = root.winfo\_screenheight()
* # Mendapatkan ukuran window
* window\_width = 300
* window\_height = 150
* # Menghitung posisi tengah window
* x\_position = (screen\_width - window\_width) // 2
* y\_position = (screen\_height - window\_height) // 2
* # Create and configure widgets
* browse\_button = tk.Button(root, text="Browse Image", command=browse\_image)
* result\_var = tk.StringVar()
* result\_label = tk.Label(root, textvariable=result\_var, wraplength=400, justify='left')
* image\_label = tk.Label(root)
* # Place widgets in the window
* browse\_button.pack(pady=10)
* image\_label.pack(pady=10)
* result\_label.pack(pady=10)
* # Start the Tkinter event loop
* root.mainloop()
* Hasil Dari Codingan diatas



PERTEMUAN 9

1. Pemrograman Berorientasi Objek (OOP)

* Konsep OOP

OOP adalah paradigma pemrograman yang berfokus pada objek. Objek adalah entitas yang memiliki data dan perilaku.

* Objek

Objek adalah representasi dari sesuatu di dunia nyata. Objek memiliki data dan perilaku. Data objek disebut atribut, sedangkan perilaku objek disebut method.

* Kelas

Kelas adalah blueprint dari objek. Kelas mendefinisikan atribut dan method yang dimiliki oleh objek.

* Atribut

Atribut adalah data yang dimiliki oleh objek. Atribut dapat berupa data primitif, data kompleks, atau objek.

* Method

Method adalah perilaku yang dimiliki oleh objek. Method dapat digunakan untuk mengubah data objek, melakukan operasi, atau berinteraksi dengan objek lain.

* Enkapsulasi

Enkapsulasi adalah teknik untuk menyembunyikan data dan method dari objek lain. Enkapsulasi dapat digunakan untuk menjaga kerahasiaan data dan perilaku objek.

* Pewarisan

Pewarisan adalah teknik untuk membuat objek baru berdasarkan objek yang sudah ada. Objek baru akan mewarisi semua atribut dan method dari objek yang sudah ada.

* Polimorfisme

Polimorfisme adalah teknik untuk menggunakan objek yang berbeda dengan cara yang sama. Polimorfisme dapat digunakan untuk membuat kode yang lebih efisien dan mudah dibaca.

1. Aplikasi suhu dengan OOP

Untuk membuat aplikasi suhu dengan OOP, kita dapat menggunakan konsep kelas dan objek.

Berikut adalah contoh program untuk membuat aplikasi suhu dengan OOP:

*class Suhu:*

*def \_\_init\_\_(self, suhu, satuan):*

*self.suhu = suhu*

*self.satuan = satuan*

*def konversi(self, satuan\_tujuan):*

*if self.satuan == "C" and satuan\_tujuan == "C":*

*return self.suhu*

*elif self.satuan == "C" and satuan\_tujuan == "F":*

*return (self.suhu \* 9/5) + 32*

*elif self.satuan == "C" and satuan\_tujuan == "K":*

*return self.suhu + 273.15*

*elif self.satuan == "F" and satuan\_tujuan == "C":*

*return (self.suhu - 32) / 9 \* 5*

*elif self.satuan == "F" and satuan\_tujuan == "F":*

*return self.suhu*

*elif self.satuan == "F" and satuan\_tujuan == "K":*

*return (self.suhu - 32) / 9 \* 5 + 273.15*

*elif self.satuan == "K" and satuan\_tujuan == "C":*

*return self.suhu - 273.15*

*elif self.satuan == "K" and satuan\_tujuan == "F":*

*return (self.suhu - 273.15) \* 9/5 + 32*

*elif self.satuan == "K" and satuan\_tujuan == "K":*

*return self.suhu*

*if \_\_name\_\_ == "\_\_main\_\_":*

*suhu = Suhu(25, "C")*

*print(suhu.konversi("K"))*

Program ini akan membuat kelas *Suhu*. Kelas *Suhu* memiliki dua atribut, yaitu *suhu* dan *satuan*. Kelas *Suhu* juga memiliki satu method, yaitu *konversi().*

Method *konversi()* digunakan untuk mengkonversi *suhu* dari satu satuan ke satuan lainnya.

Pada program di atas, objek *suhu* dibuat dengan nilai *25* dan satuan *C*. Nilai suhu dikonversi ke satuan *K* dengan menggunakan method *konversi()*. Hasil konversi ditampilkan ke layar.

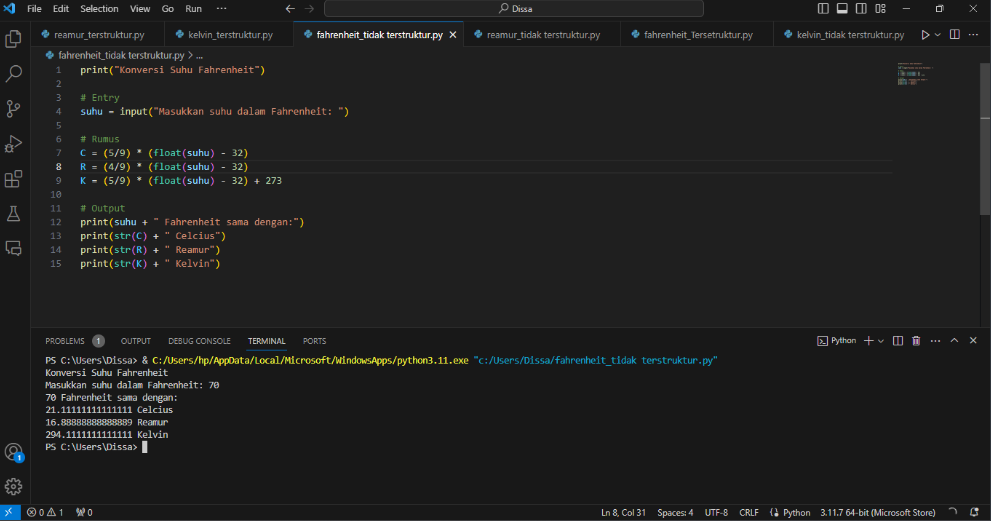
Anda dapat mengembangkan program ini dengan menambahkan fungsi-fungsi lain, seperti tombol untuk menghapus data, tombol untuk mengosongkan layar, dan sebagainya.

Berikut adalah beberapa tips untuk membuat aplikasi suhu dengan OOP:

* Gunakan kelas untuk mewakili konsep suhu.
* Gunakan atribut untuk menyimpan data suhu.

Contoh :

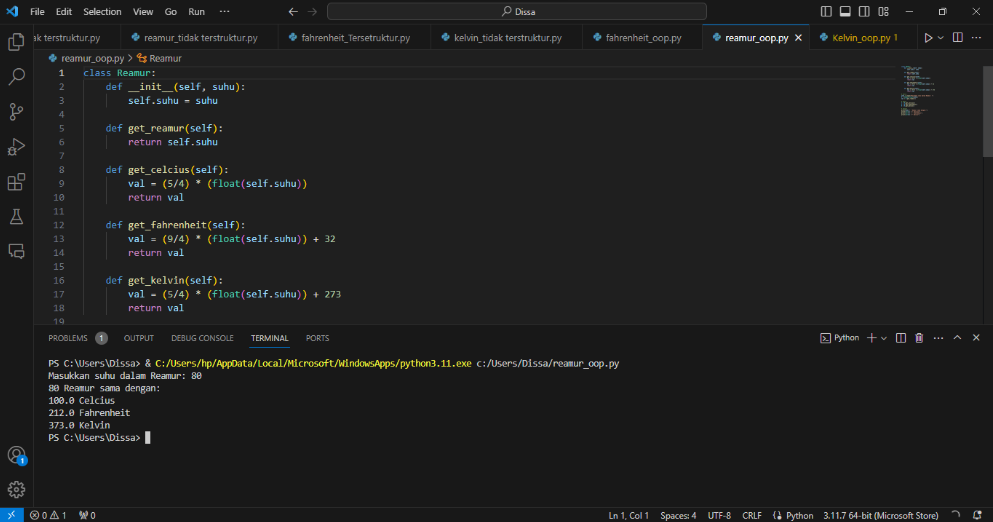
* Fahrenheit tidak terstruktur



* kelvin Terstruktur



* Reamur OOP



PERTEMUAN 10

1. XAMPP

XAMPP adalah paket perangkat lunak yang digunakan untuk menjalankan aplikasi web lokal. XAMPP terdiri dari beberapa komponen, yaitu:

* Apache: server web
* MySQL: database
* PHP: bahasa pemrograman web

1. Apache

Apache adalah server web yang paling populer di dunia. Apache digunakan untuk melayani permintaan web dari klien.

1. MySQL

MySQL adalah sistem manajemen basis data relasional (RDBMS) yang populer. MySQL digunakan untuk menyimpan data web.

1. PHP

PHP adalah bahasa pemrograman web yang populer. PHP digunakan untuk membuat halaman web dinamis.

1. Apache dan MySQL

Apache dan MySQL adalah komponen penting dari XAMPP. Apache digunakan untuk melayani permintaan web dari klien, sedangkan MySQL digunakan untuk menyimpan data web.

1. Instalasi XAMPP

Untuk menginstal XAMPP, Anda dapat mengikuti langkah-langkah berikut:

1. Unduh paket XAMPP dari situs web resmi XAMPP.
2. Ekstrak paket XAMPP ke folder yang diinginkan.
3. Jalankan file *xampp-control.exe.*
4. Klik tombol "Start" pada komponen Apache dan MySQL.

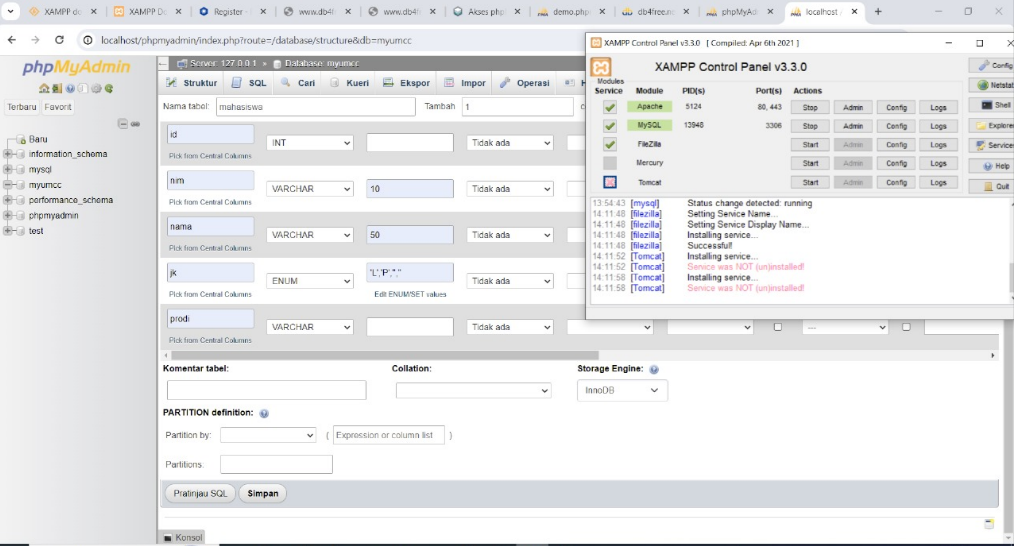
Setelah XAMPP diinstal, Anda dapat mengakses aplikasi web lokal Anda dengan membuka browser dan memasukkan alamat *localhost.*

1. Tips untuk menggunakan XAMPP

Berikut adalah beberapa tips untuk menggunakan XAMPP:

* Gunakan folder *htdocs* untuk menyimpan file aplikasi web Anda.
* Gunakan folder *xampp/phpmyadmin* untuk mengakses MySQL.
* Gunakan folder *xampp/logs* untuk melihat log XAMPP.

Contoh :



PERTEMUAN 11

1. Python Database

Python memiliki dukungan bawaan untuk database. Python dapat digunakan untuk mengakses database yang berbeda, seperti MySQL, PostgreSQL, dan SQLite.

Untuk mengakses database MySQL dari Python, kita dapat menggunakan modul *mysql.connector*. Modul *mysql.connector* menyediakan fungsi-fungsi untuk menghubungkan Python dengan MySQL, menjalankan query, dan memanipulasi data.

Untuk menghubungkan Python dengan MySQL, kita dapat menggunakan kode berikut:

*import mysql.connector*

*# Koneksi ke database*

*conn = mysql.connector.connect(*

*host="localhost",*

*user="root",*

*password="",*

*database="my\_database"*

*)*

Kode di atas akan membuat koneksi ke database *my\_database* di server *localhost* dengan username *root* dan password kosong.

1. Membuat aplikasi Tkinter dengan database MySQL

Untuk membuat aplikasi Tkinter dengan database MySQL, kita dapat menggunakan kode berikut:

*import mysql.connector*

*import tkinter as tk*

*# Koneksi ke database*

*conn = mysql.connector.connect(*

*host="localhost",*

*user="root",*

*password="",*

*database="my\_database"*

*)*

*# Buat window*

*window = tk.Tk()*

*# Buat label*

*label\_nama = tk.Label(window, text="Nama:")*

*label\_alamat = tk.Label(window, text="Alamat:")*

*# Buat entry*

*entry\_nama = tk.Entry(window)*

*entry\_alamat = tk.Entry(window)*

*# Buat button*

*button\_simpan = tk.Button(window, text="Simpan")*

*# Buat fungsi untuk menyimpan data*

*def simpan():*

*# Ambil data dari entry*

*nama = entry\_nama.get()*

*alamat = entry\_alamat.get()*

*# Masukkan data ke database*

*cursor = conn.cursor()*

*cursor.execute("INSERT INTO `data` (`nama`, `alamat`) VALUES (%s, %s)", (nama, alamat))*

*conn.commit()*

*# Kosongkan entry*

*entry\_nama.delete(0, "end")*

*entry\_alamat.delete(0, "end")*

*# Tampilkan label dan entry*

*label\_nama.grid(row=0, column=0)*

*entry\_nama.grid(row=0, column=1)*

*label\_alamat.grid(row=1, column=0)*

*entry\_alamat.grid(row=1, column=1)*

*button\_simpan.grid(row=2, column=0)*

*# Tampilkan window*

*window.mainloop()*

Kode di atas akan membuat window dengan dua label, dua entry, dan satu button. Label pertama untuk menampilkan teks "Nama:", label kedua untuk menampilkan teks "Alamat:", entry pertama untuk memasukkan nama, dan entry kedua untuk memasukkan alamat. Button "Simpan" digunakan untuk menyimpan data ke database.

Fungsi *simpan()* akan mengambil data dari entry dan memasukkannya ke database. Untuk memasukkan data ke database, kita dapat menggunakan objek *cursor*. Objek *cursor* digunakan untuk menjalankan query SQL.

Setelah data dimasukkan ke database, entry akan dikosongkan.

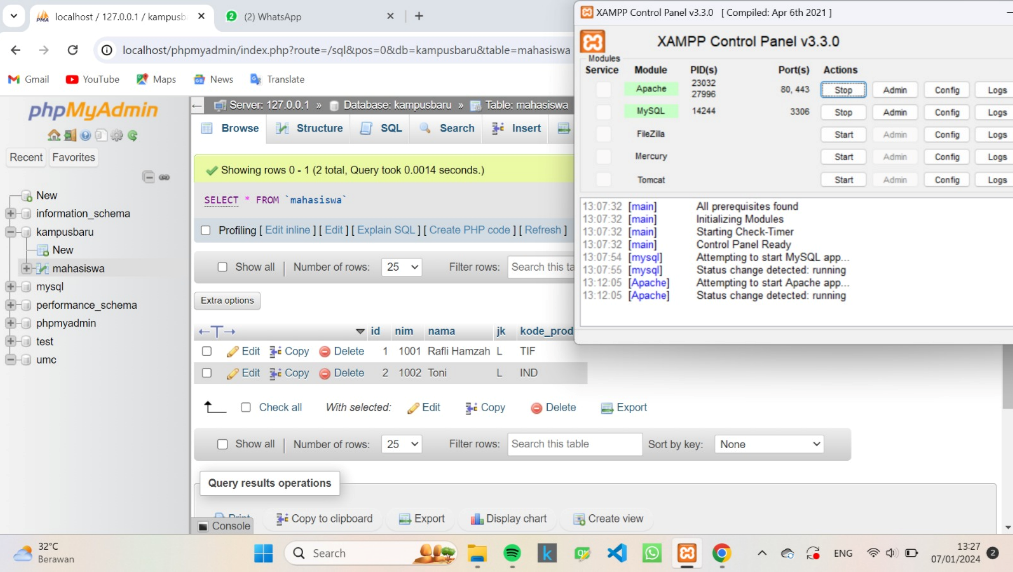
Anda dapat mengembangkan aplikasi ini dengan menambahkan fungsi-fungsi lain, seperti tombol untuk menghapus data, tombol untuk mengosongkan layar, dan sebagainya.

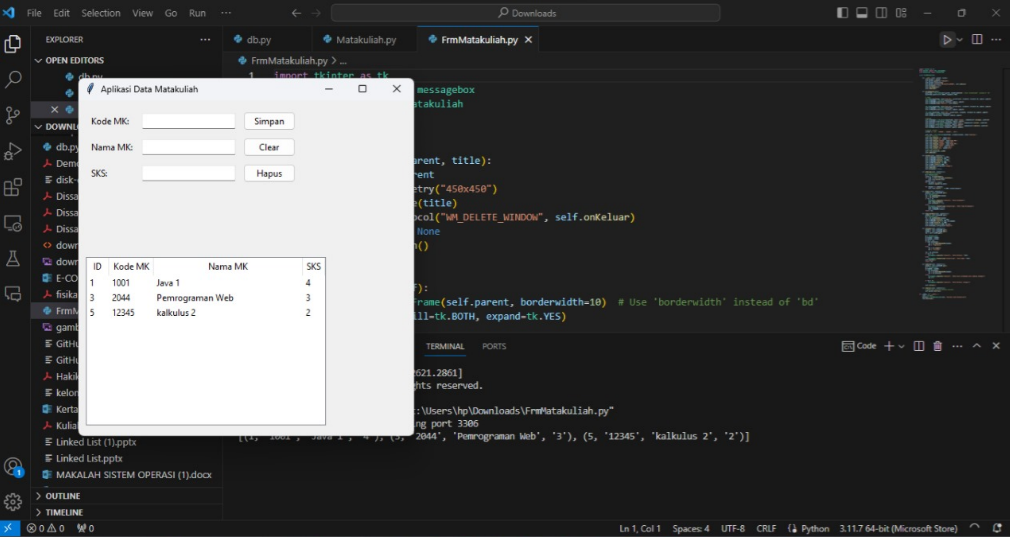
1. Tips untuk membuat aplikasi Python Database dan Tkinter

Berikut adalah beberapa tips untuk membuat aplikasi Python Database dan Tkinter:

* Gunakan modul *mysql.connector* untuk menghubungkan Python dengan MySQL.
* Gunakan objek *cursor* untuk menjalankan query SQL.
* Gunakan fungsi *commit()* untuk menyimpan perubahan ke database.
* Gunakan fungsi *rollback()* untuk membatalkan perubahan ke database.
* Pisahkan kode untuk koneksi database dan kode untuk aplikasi GUI.

Contoh :





PERTEMUAN 12

1. Membuat aplikasi Python Database berbasis MySQL Server

Untuk membuat aplikasi Python Database berbasis MySQL Server, kita dapat menggunakan langkah-langkah berikut:

1. Buat koneksi ke MySQL Server.
2. Buat database.
3. Buat tabel.
4. Masukkan data ke tabel.
5. Tampilkan data dari tabel.

Berikut adalah contoh kode untuk membuat aplikasi Python Database berbasis MySQL Server:

*import mysql.connector*

*# Koneksi ke MySQL Server*

*conn = mysql.connector.connect(*

*host="localhost",*

*user="root",*

*password="",*

*database="my\_database"*

*)*

*# Buat database*

*cursor = conn.cursor()*

*cursor.execute("CREATE DATABASE my\_database")*

*# Buat tabel*

*cursor.execute("CREATE TABLE `data` (*

*`id` INT NOT NULL AUTO\_INCREMENT,*

*`nama` VARCHAR(255) NOT NULL,*

*`alamat` VARCHAR(255) NOT NULL,*

*PRIMARY KEY (`id`)*

*)")*

*# Masukkan data ke tabel*

*cursor.execute("INSERT INTO `data` (`nama`, `alamat`) VALUES ('John Doe', 'Jakarta')")*

*cursor.execute("INSERT INTO `data` (`nama`, `alamat`) VALUES ('Jane Doe', 'Bandung')")*

*# Tampilkan data dari tabel*

*cursor.execute("SELECT \* FROM `data`")*

*for row in cursor:*

*print(row)*

*# Tutup koneksi*

*conn.close()*

Kode di atas akan membuat koneksi ke MySQL Server dengan username *root* dan password kosong. Kode juga akan membuat database bernama *my\_database* dan tabel bernama *data*. Tabel *data* memiliki dua kolom, yaitu *nama* dan *alamat*.

Kemudian, kode akan memasukkan dua data ke tabel *data*. Data pertama adalah *John Doe* dengan alamat *Jakarta*, dan data kedua adalah *Jane Doe* dengan alamat *Bandung*.

Terakhir, kode akan menampilkan data dari tabel *data*.

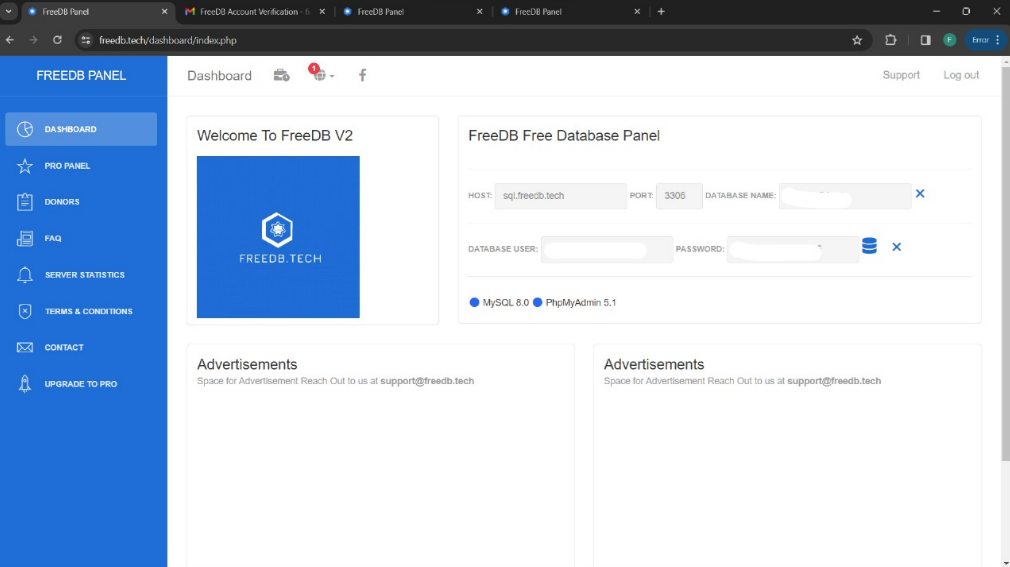
Anda dapat mengembangkan aplikasi ini dengan menambahkan fungsi-fungsi lain, seperti tombol untuk menghapus data, tombol untuk mengosongkan layar, dan sebagainya.

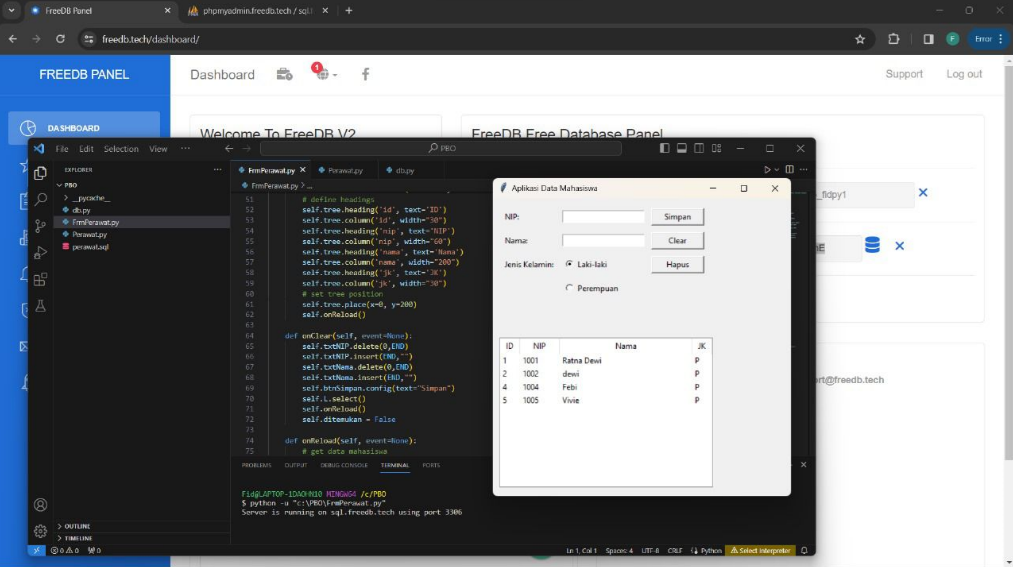
Tips untuk membuat aplikasi Python Database berbasis MySQL Server

Berikut adalah beberapa tips untuk membuat aplikasi Python Database berbasis MySQL Server:

* Gunakan modul *mysql.connector* untuk menghubungkan Python dengan MySQL Server.
* Gunakan objek *cursor* untuk menjalankan query SQL.
* Gunakan fungsi *commit()* untuk menyimpan perubahan ke database.
* Gunakan fungsi *rollback()* untuk membatalkan perubahan ke database.
* Pisahkan kode untuk koneksi database dan kode untuk aplikasi GUI.

Contoh :





PERTEMUAN 13

1. Materi Teori Singkat

* Konsep login
* Aplikasi login menggunakan Tkinter
* Menghubungkan Python dengan MySQL
* Menjalankan query SQL untuk login

1. Konsep login

Login adalah proses masuk ke suatu sistem atau aplikasi. Login biasanya dilakukan dengan memasukkan username dan password.

Aplikasi login biasanya memiliki dua form, yaitu form login dan form utama. Form login digunakan untuk memasukkan username dan password. Form utama digunakan untuk menampilkan konten aplikasi.

1. Aplikasi login menggunakan Tkinter

Tkinter adalah modul Python yang digunakan untuk membuat antarmuka pengguna grafis (GUI). Tkinter menyediakan widget yang dapat digunakan untuk membuat berbagai elemen GUI, seperti label, button, dan entry.

Untuk membuat aplikasi login menggunakan Tkinter, kita dapat menggunakan langkah-langkah berikut:

1. Buat window.
2. Tambahkan label dan entry untuk username dan password.
3. Tambahkan button untuk login.
4. Tambahkan event handler untuk button login.
5. Menghubungkan Python dengan MySQL

Untuk menghubungkan Python dengan MySQL, kita dapat menggunakan modul *mysql.connector*. Modul *mysql.connector* menyediakan fungsi-fungsi untuk menghubungkan Python dengan MySQL, menjalankan query SQL, dan memanipulasi data.

1. Menjalankan query SQL untuk login

Untuk menjalankan query SQL untuk login, kita dapat menggunakan objek *cursor*. Objek *cursor* digunakan untuk menjalankan query SQL.

Berikut adalah contoh kode untuk menjalankan query SQL untuk login:

*import mysql.connector*

*# Koneksi ke MySQL Server*

*conn = mysql.connector.connect(*

*host="localhost",*

*user="root",*

*password="",*

*database="my\_database"*

*)*

*# Buat cursor*

*cursor = conn.cursor()*

*# Jalankan query SQL*

*cursor.execute("SELECT \* FROM `users` WHERE `username` = %s AND `password` = %s", (username, password))*

*# Periksa hasil query*

*if cursor.rowcount == 1:*

*# Login berhasil*

*print("Login berhasil")*

*else:*

*# Login gagal*

*print("Login gagal")*

*# Tutup koneksi*

*conn.close()*

Kode di atas akan menjalankan query SQL untuk mengambil data dari tabel *users*. Query SQL akan mencari data dengan username dan password yang dimasukkan oleh pengguna.

Jika data ditemukan, maka login berhasil. Jika data tidak ditemukan, maka login gagal.

1. Aplikasi Login Python Database

Berikut adalah contoh kode untuk membuat aplikasi login Python Database menggunakan Tkinter dan MySQL:

*import mysql.connector*

*import tkinter as tk*

*# Koneksi ke MySQL Server*

*conn = mysql.connector.connect(*

*host="localhost",*

*user="root",*

*password="",*

*database="my\_database"*

*)*

*# Buat window*

*window = tk.Tk()*

*# Buat label dan entry untuk username*

*label\_username = tk.Label(window, text="Username:")*

*entry\_username = tk.Entry(window)*

*# Buat label dan entry untuk password*

*label\_password = tk.Label(window, text="Password:")*

*entry\_password = tk.Entry(window, show="\*")*

*# Buat button untuk login*

*button\_login = tk.Button(window, text="Login")*

*# Tambahkan event handler untuk button login*

*def login():*

*# Ambil data dari entry*

*username = entry\_username.get()*

*password = entry\_password.get()*

*# Jalankan query SQL untuk login*

*cursor = conn.cursor()*

*cursor.execute("SELECT \* FROM `users` WHERE `username` = %s AND `password` = %s", (username, password))*

*# Periksa hasil query*

*if cursor.rowcount == 1:*

*# Login berhasil*

*window.destroy()*

*else:*

*# Login gagal*

*tk.messagebox.showerror("Login Gagal", "Username atau password salah")*

*# Tampilkan label dan entry*

*label\_username.grid(row=0, column=0)*

*entry\_username.grid(row=0, column=1)*

*label\_password.grid(row=1, column=0*

*# Tambahkan button login*

*button\_login.grid(row=2, column=0)*

*# Tampilkan window*

*window.mainloop()*

*# Tutup koneksi*

*conn.close()*

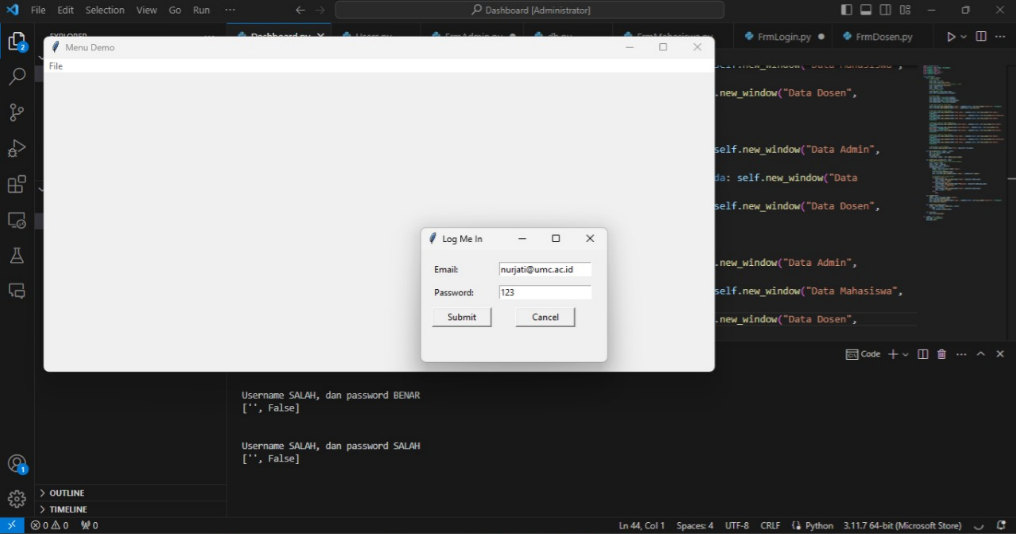
Kode di atas akan membuat aplikasi login dengan dua form, yaitu form login dan form utama. Form login akan menampilkan label dan entry untuk username dan password. Form utama akan ditampilkan setelah login berhasil.

Ketika pengguna menekan button login, kode akan menjalankan fungsi *login()*. Fungsi *login()* akan mengambil data dari entry dan menjalankan query SQL untuk login.

Jika login berhasil, maka form login akan ditutup dan form utama akan ditampilkan. Jika login gagal, maka akan muncul pesan error.

Anda dapat mengembangkan aplikasi ini dengan menambahkan fungsi-fungsi lain, seperti tombol untuk mendaftar, tombol untuk lupa password, dan sebagainya.

Contoh :



**TERIMA KASIH**

**\*\*\***