

Kata Pengantar

Puji syukur kita panjatkan kehadirat Allah Swt. yang telah memberikan rahmat dan hidayah-Nya sehingga saya dapat menyelesaikan tugas yang berjudul "Modul Praktikum PBO" ini tepat pada waktunya.

Adapun tujuan dari penulisan Modul Praktikum ini adalah untuk memenuhi tugas mata kuliah Pemrograman II dan juga untuk memperdalam pemahaman saya tentang Pemrograman.

Saya ingin mengucapkan terima kasih kepada Bapak Freddy Wicaksono selaku Dosen Pemrograman yang telah memberikan tugas ini. Terima kasih atas bimbingannya yang diberikan dalam proses penyelesaian Modul Praktikum ini.

Tidak lupa pula saya mengucapkan terima kasih kepada semua pihak yang telah membantu, meskipun tidak dapat disebutkan satu per satu. Semua bantuan dan dukungan yang diberikan sangat berarti bagi kelancaran penulisan Modul Praktikum ini.

Saya menyadari bahwa Modul ini masih memiliki kekurangan dan kelemahan. Oleh karena itu, segala kritik dan saran yang membangun sangat saya harapkan untuk perbaikan di masa mendatang.

Akhir kata, semoga laporan ini dapat bermanfaat bagi pembaca dan turut menyumbangkan pengetahuan yang berguna di bidangnya masing-masing.

Cirebon, 25 Januari 2024

Intan Sulastri

DAFTAR ISI

Kata Pengantar ………………………………………………………………………………..1

Daftar Isi ………………………………………………………………………………………2

Pertemuan 1 (Pengenalan Python) ……………………………………………………………3

Pertemuan 2 (Menghitung bangun ruang) …………………………………………………….6

Pertemuan 3 (Function, Tkinter) …………………………………………………………….11

Pertemuan 4 (Mp3 dan Mp4) ……………………………………………………………….20

Pertemuan 5 (Quiz) …………………………………………………………………………25

Pertemuan 6 (Membuat Kalkulator) ………………………………………………………..26

Pertemuan 7 (Konverter Suhu) ……………………………………………………………..29

Pertemuan 8 (UTS) …………………………………………………………………………31

Pertemuan 9 (Konversi suhu dengan OOP terstruktur) …………………………………….38

Pertemuan 10 (Pengenalan MySQL) ……………………………………………………….40

Pertemuan 11 (Python Database) …………………………………………………………...41

Pertemuan 12 (Aplikasi Python Database berbasis MySQL) ………………………………

Pertemuan 13 (Database menggunakan Tkinter dan MySQL) ……………………………..

Pertemuan 14 (Modul Praktikum) ………………………………………………………….

Penutup

Pertemuan 1

( Pengenalan Python)

Bahasa pemrograman python adalah Bahasa pemrograman yang dikembangkan oleh Diego Van Rossum pada tahun 1991. Bahasa pemrograman ini banyak digunakan dalam berbagai bidang seperti pengembangan perangkat lunak, web development, analisis data,dan kecerdasan buatan, Python merupakan bahaa pemrograman yang berorientasi pada objek, bersifat dinamis dan menyediakan Pustaka untuk tugas-tugas umum, makanya Bahasa pemrograman ini sangat popular dan mudah untuk dipelajari.

Setelah penjelasan singkat diatas tentu kita harus melakukan Langkah lain untuk mempelajari Bahasa pemrograman Python, sebelum praktek adakalanya kita menguasai secara teori terlebih dahulu.

Maka dari itu ada beberapa soal yang dapat kita kerjakan untuk bisa mengenal lebih dalam apa itu Python.

# Soal Pertama

Jelaskan apa itu pemrograman python?

# Jawaban: Pemrograman Python adalah suatu bentuk pemrograman computer yang menggunakan Bahasa pemrograman Python. Python dikenal karena sintaksis yang mudah dibaca dan digunakan, serta memiliki berbagai Pustaka yang mendukung berbagai fungsi.

# Soal Kedua

Jelaskan secara singkat python

# Jawaban: Python dikembangkan oleh Guido Van Rossum dan dirilis pada tahun 1991. Bahasa pemrograman ini berkembang pesat dan menjadi Bahasa pemrograman paling popular karena kemudahannya untuk dipelajari.

# Soal Ketiga

Bagaimana proses istalasi Python?

# Jawaban: Proses istalasi python melibatkan disteribusi python dari situs resminya yaitu (python.org) dan kita harus mengikuti petunjuk istalasi sesuai dengan system operasi yang digunakan.

#Soal Keempat

Bagaimana cara memastikan keberhasilan instalasi python?

# Jawaban: Agar kita tahu apakah kita berhasil atau tidak dalam menginstal python kita bisa buka terminal atau command prompt, kemudian ketik **‘python – version’** atau **‘python -v’** untuk melihat versi python apa yang terpasang.

# Soal Kelima

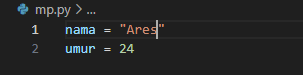
Jelaskan tipe data yang ada di python

# Jawaban: Tipe data yang ada di python meliputi **int (bilangan bulat), float (bilangan decimal), str (srting), bool (Boolean), list (daftar), tuple (tupel), set (himpunan), dan dict (kamus).**

# Soal Keenam

Jelaskan tentang variable di python dan berikan contohnya

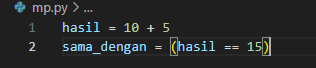
# Jawaban: Variable di python ada banyak jenisnya salah satunya adalah variable yang digunakan untuk menyimpan nilai. Contohnya:



#Soal Ketujuh

Jelaskan operator yang ada pada python dan berikan contohnya

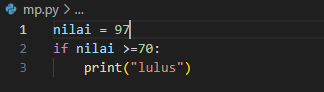
#Jawaban: Operator di python termasuk operator aritmetika **(+,-,\*,/)**, kalau operator perbandingan **(==, !=, <, >)**, dan operator logika **(and, or, not).** Contohnya:



# Soal Kedelapan

Jelaskan tantang IF (kondisi) dalam pemrograman python serta berikan contohnya

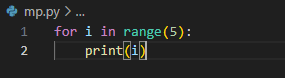
#Jawaban: Kondisi if digunakan untuk menjalankan blok kode jika suatu kondisi benar. Contohnya:



# Soal Kesembilan

Jelaskan tentang pengulangan (loop) serta berikan contohnya

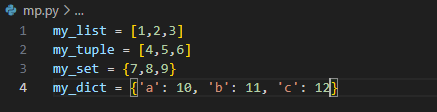
# Jawaban: Pengulangan di python dapat dilakukan dengan menggunakan ‘for’ atau ‘while’. Dan dibawah ini adalaha contoh looping dengan menggunakan ‘for’:



# Soal Kesepuluh

Jelaskan tentang List, Tuple, Set, dan Dictionary serta berikan contohnya

# Jawaban : Struktur data seperti List, Tuple, Set, Dan Dictionary digunakan untuk menyimpan list data, Contohnya :



Soal diatas adalah bentuk Latihan untuk pertemuan pertama, sebagai bentuk perkenalan Bahasa pemrograman Python, dan pada pertemuan selanjutnya bukan hanya saja teori yang akan dipelajari namun ada juga beberapa rangkaian praktek dan coding coding yang akan kita pelajari. Terima kasih dan maaf bila ada beberapa kesalahan atau kekurangan.

Pertemuan 2

( Menghitung Bangun Ruang)

Di pertemuan kali ini kita akan mencoba mnghitung bangun ruang, dengan rumus dasar yang sudah ada, kita akan memasukannya kedalam coding lalu setelah itu kitab isa menghitung luas, sisi, volume atau tinggi dari bangun ruang itu, sesuai keinginan kita.

Ada beberapa bangun ruang yang akan kita hitung yakni:

Balok, bola, kerucut, kubus, limas segiempat, limas segitiga, prisma segitiga, dan tabung.

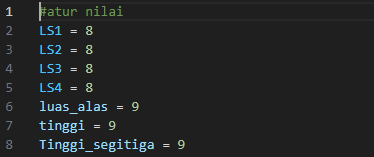
Tapi dalam contoh yang akan saya jelaskan dibawah hanya 3

(Limas segitiga, balok dan tabung)

Limas segitiga

1. Tentukan dulu nilai variable:

Disini kita akan menetapikan nilai untuk panjang sisi dan tinggi segitiga.



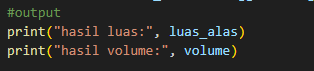
1. Menghitung luas alas dan volume

Jika kita ingin menghitung luas alas dengan menjumblahkan Panjang sisi maka kita harus menambahkan keempat sisinya dengan memasukan kata kunci #rumus dan luas\_alas.

Dan untuk menghitung volume piramida kita membutuhkan rumus 1/6 dikali dengan luas alas, tinggi dan tinggi segitiga.

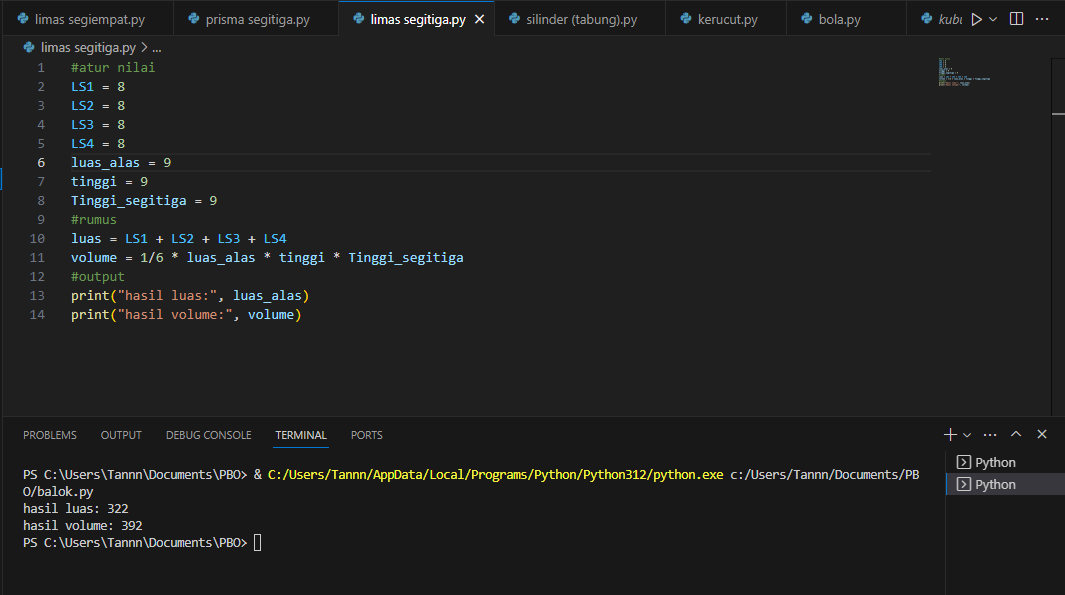


1. Menampilkan Hasil outputnya



Dari coding diatas maka hasil output yang akan keluar adalah luas dan volumenya.

Seperti ini:



Balok

Nah contoh selanjutnya kita akan mengerjakan bagian balok.

1. kita masukan dulu nilai variabelnya



Ada 3 variabel diatas yakni Panjang, lebar, dan tinggi. Ingat angka yang ada disamping itu bis akita rubah, jadi angka diatas hanya contoh.

1. Masukan rumus yang sesuai



Diatas ada 2 rumus yaitu luas & volume.

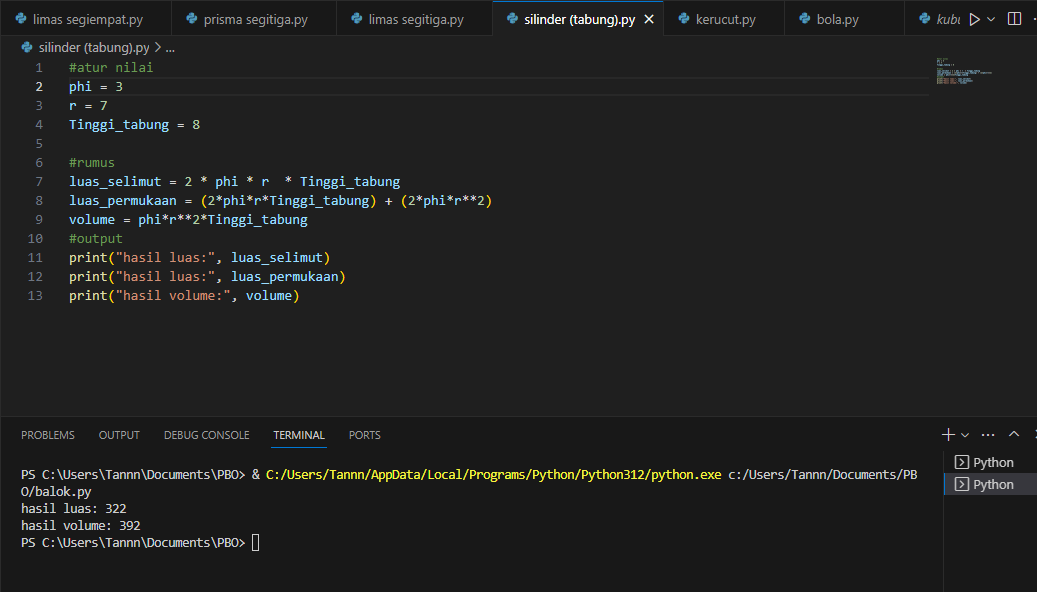
1. Dan akan keluar Hasil Output seperti ini



Tabung

Kita sudah masuk ke contoh ke-3 beda dari 2 contoh sebelumnya yang hanya mengandalkan Panjang, lebar dan tinggi kini kita akan bertemu dengan Phi dan jari-jari karena kini kita berkaitan dengan lingkaran juga.

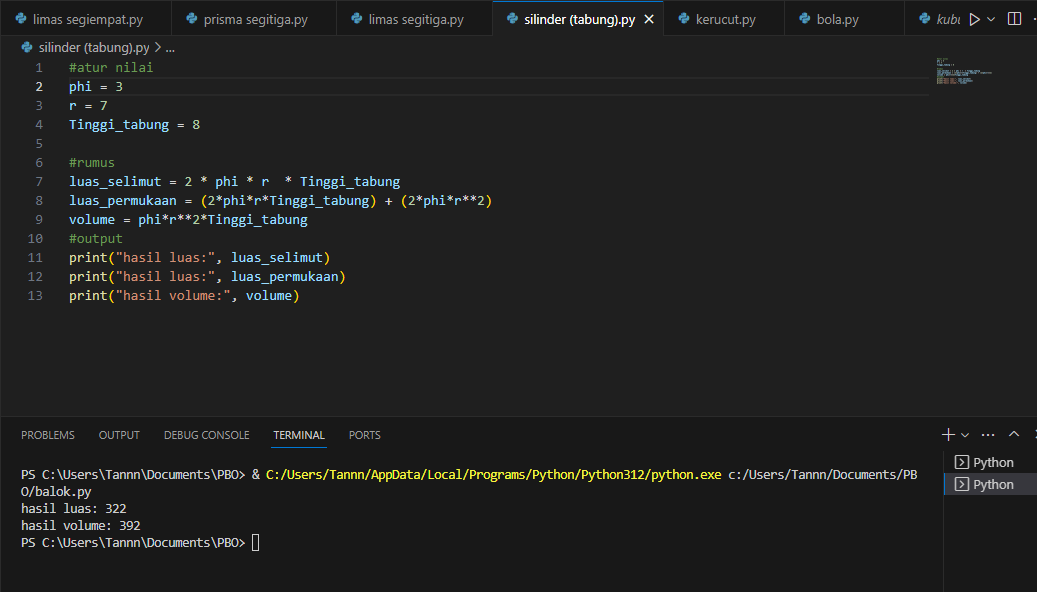
1. Kita masukan nilai dulu



Disini kita punya phi, yang kita akan set menjadi 3

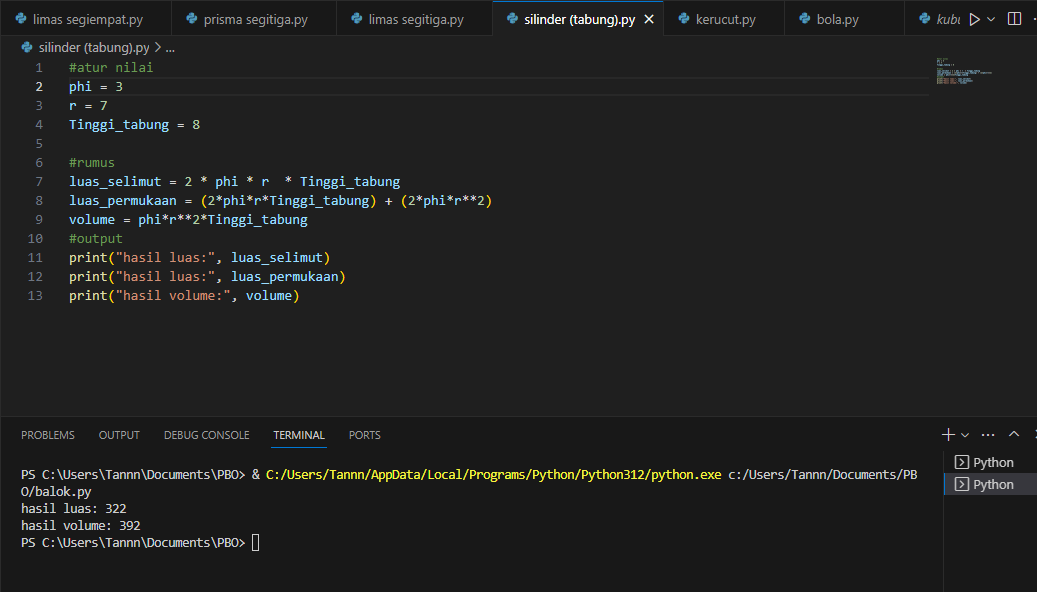
Lalu ada r (radius) yang kita atur jadi 7 dan Tinggi tabung kita akan set jadi 8

1. kita akan masukan rumusnya

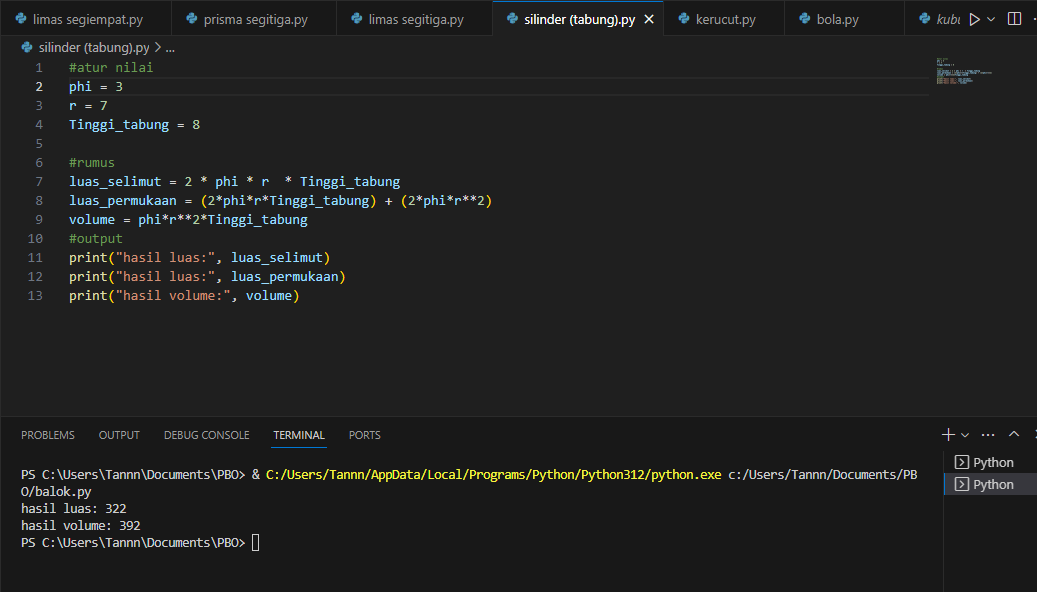


Kita akan menghitung luas selimut, luas permukaan, dan volume dengan rumus diatas.

1. Ini kode output agar kita bisa mendapatkan output yang kita mau

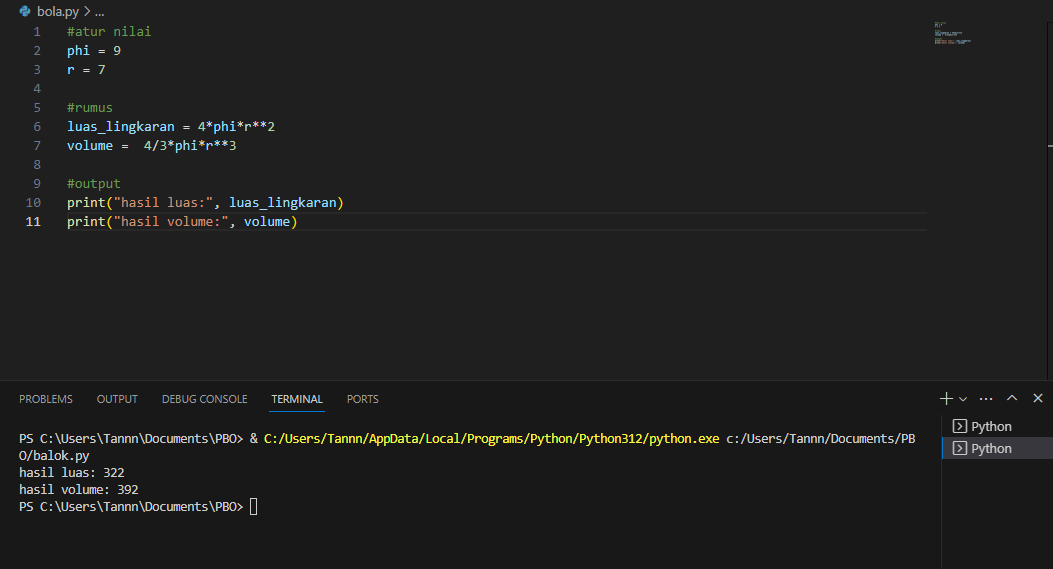
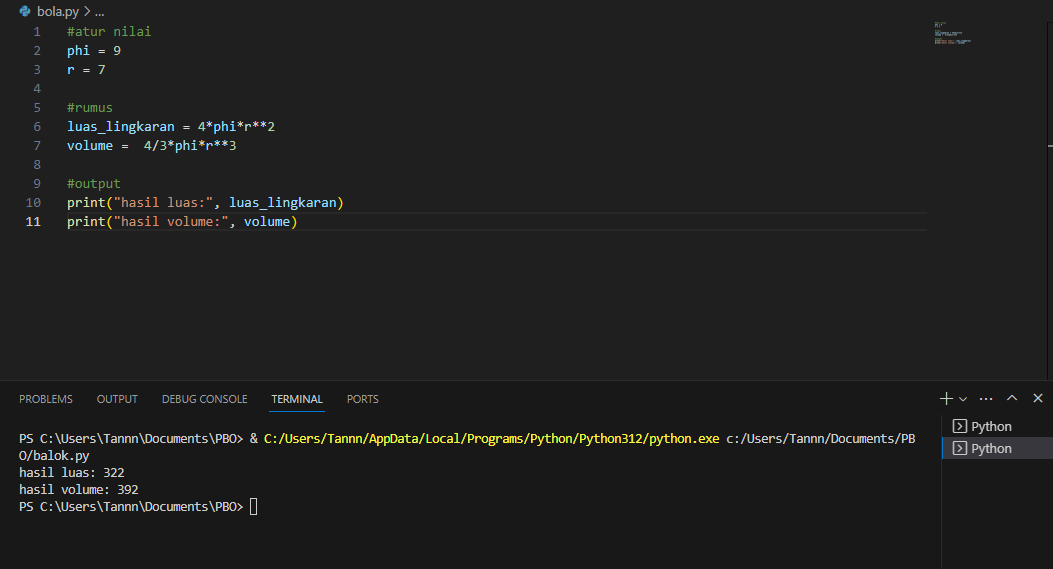


1. Hasil outputnya seperti ini



Dan Dibawah ini kita akan melihat kode dari berbagai bangun ruang yang lain.

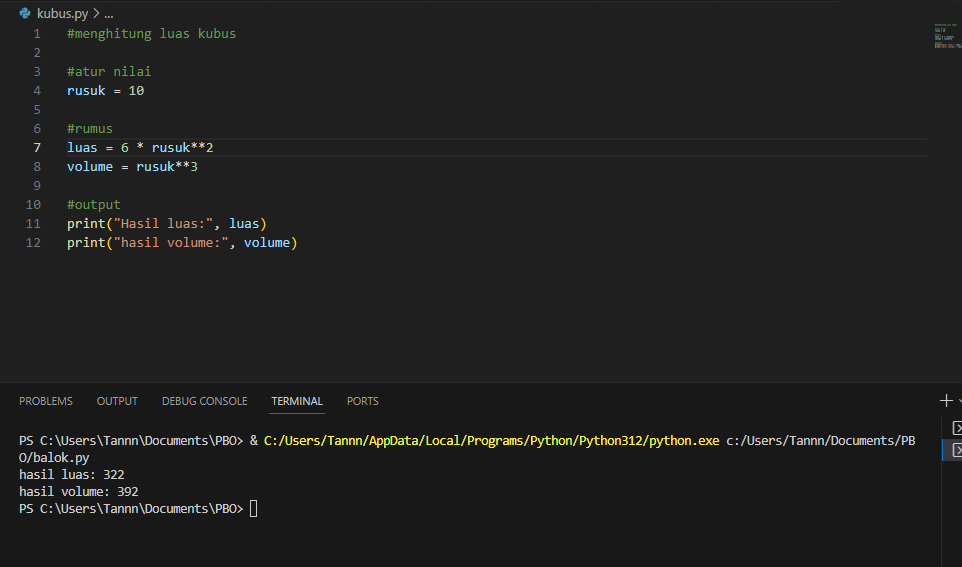
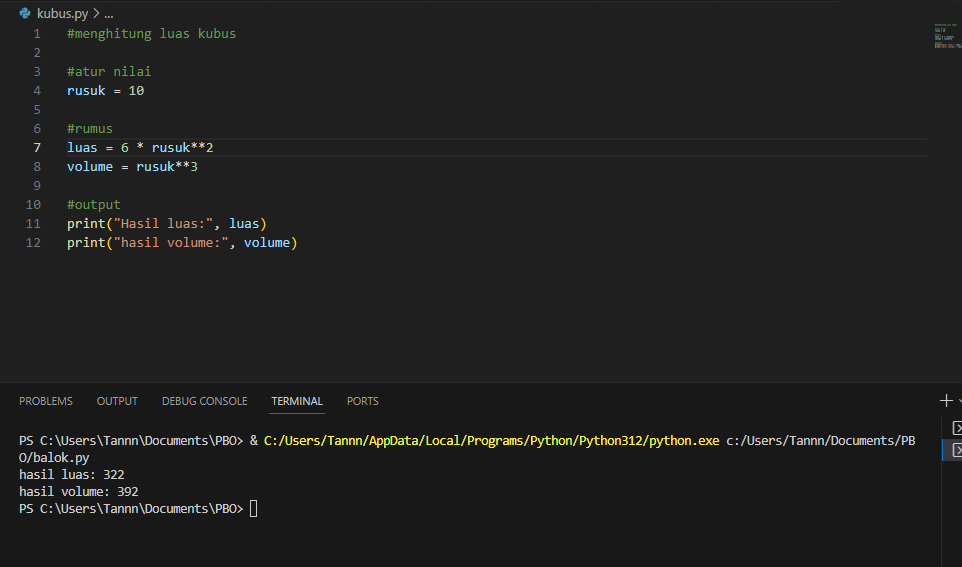
Bola input dan outputnya

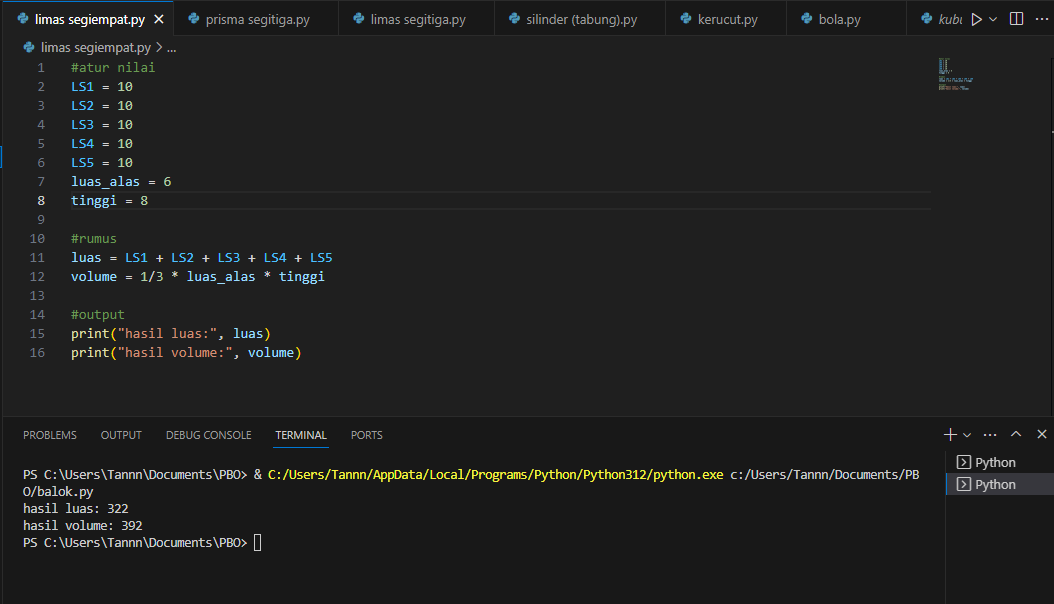
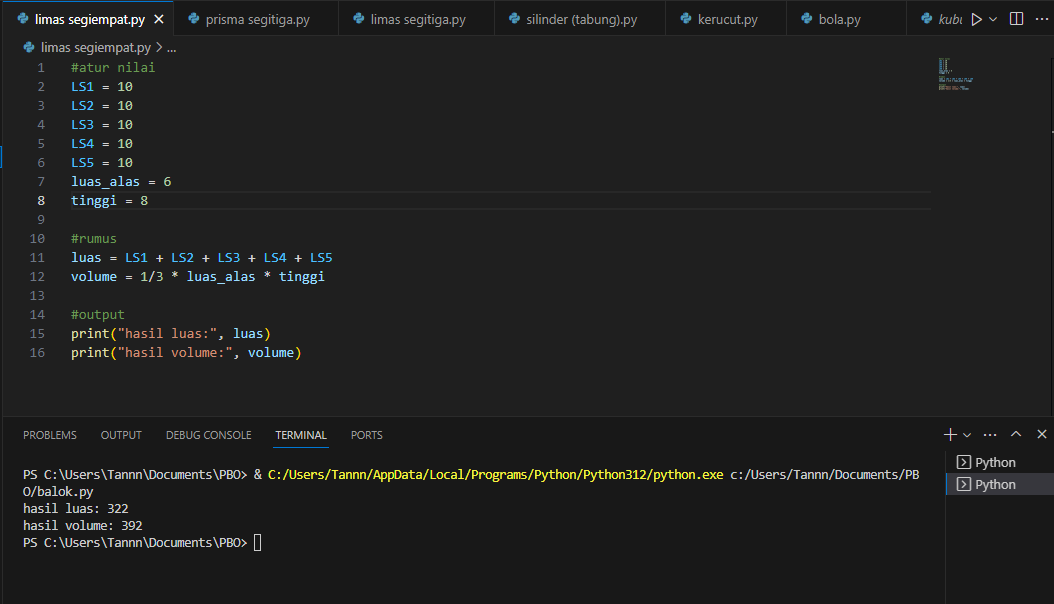
Kerucut input dan outputnya

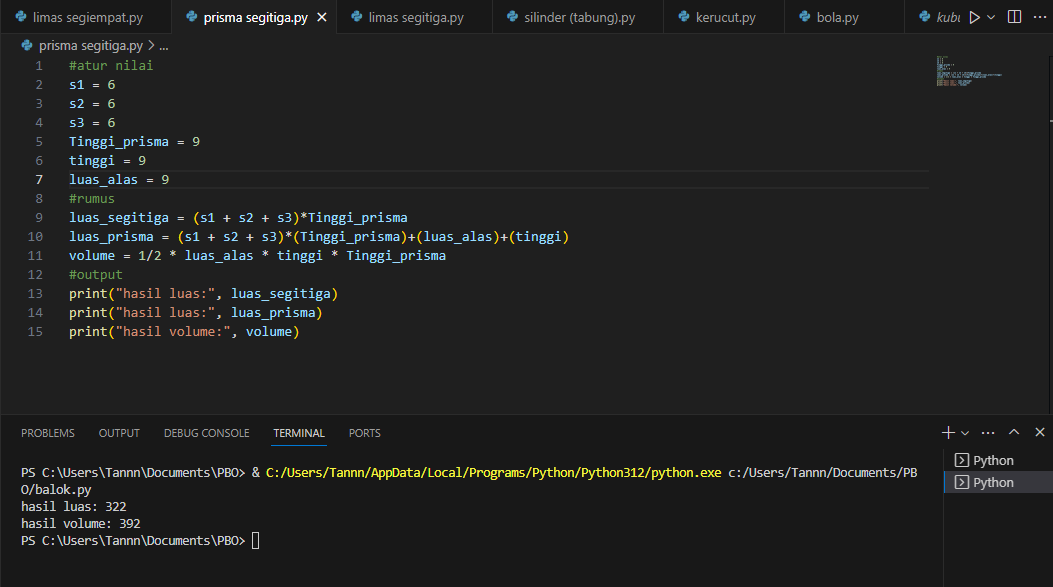
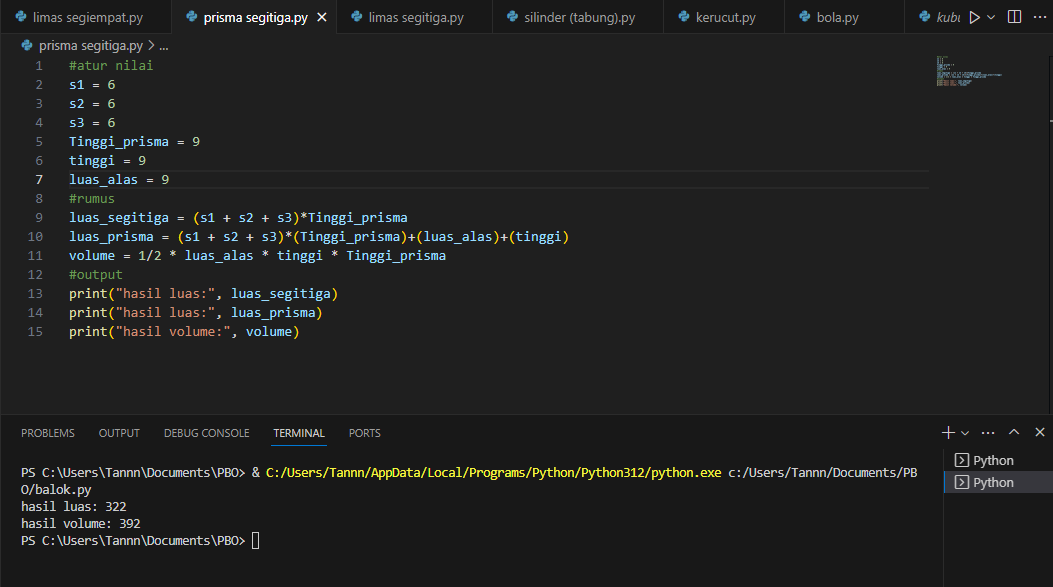
Kubus input dan outputnya

Limas segiempat input dan outputnya

Prisma segitiga input dan outputnya

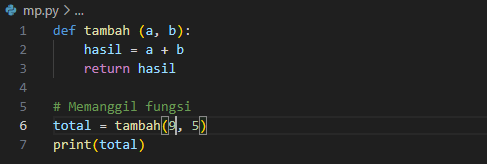
**-Selesai-**

Pertemuan 3

( Function, Tkinter)

Pada pertemuan kali ini kita akan belajar mangenai Funtion dan Tkinter, sebelumnya apa itu function?

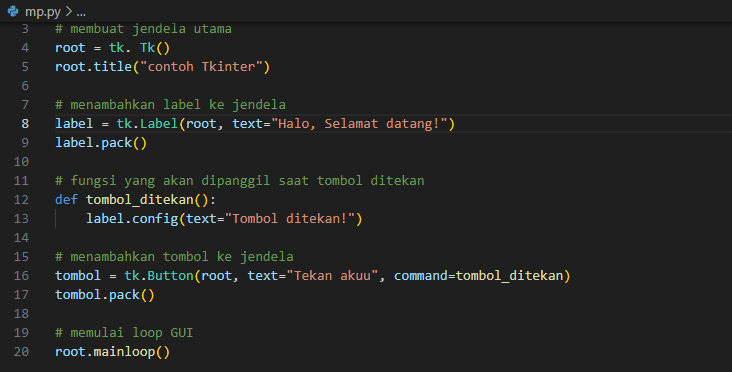
**Function** adalah blok kode yang dapat dipanggil dengan memberikan nama tertentu, function yang dapat menerima argument sebagai input, melakukan operasi tertentu dan mengembalikan hasil. Contoh sederhana function dalam python sebagai berikut:



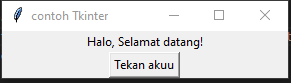
Dalam contoh ini, fungsi **‘tambah’** menrima dua argument yakni (**‘a’** dan **‘b’**), untuk menjumblahkannya, dan mengembalikan hasilnya dan kita memanggil fungsi dengan nilai 9 dan 5 maka aka nada hasil outputnya yakni 14.

Lalu apa itu Tkinter? **Tkinter** adalah modul bawaan dalam Bahasa pemrograman python yang menyediakan antarmuka grafis pengguna alias GUI. Tkinter berdasarkan toolkit GUI Tk yang dikembangkan untuk Bahasa pemrograman Tcl/Tk. Dengan Tkinter kitab isa membuat jendela, tombol, kotak teks dan elemen GUI lainnya untuk memenuhi kebutuhan kita dalam membuat aplikasi desktop dengan antarmuka pengguna yang lebih interaktif.

Berikut ini contoh sederhana penggunaan Tkinter:



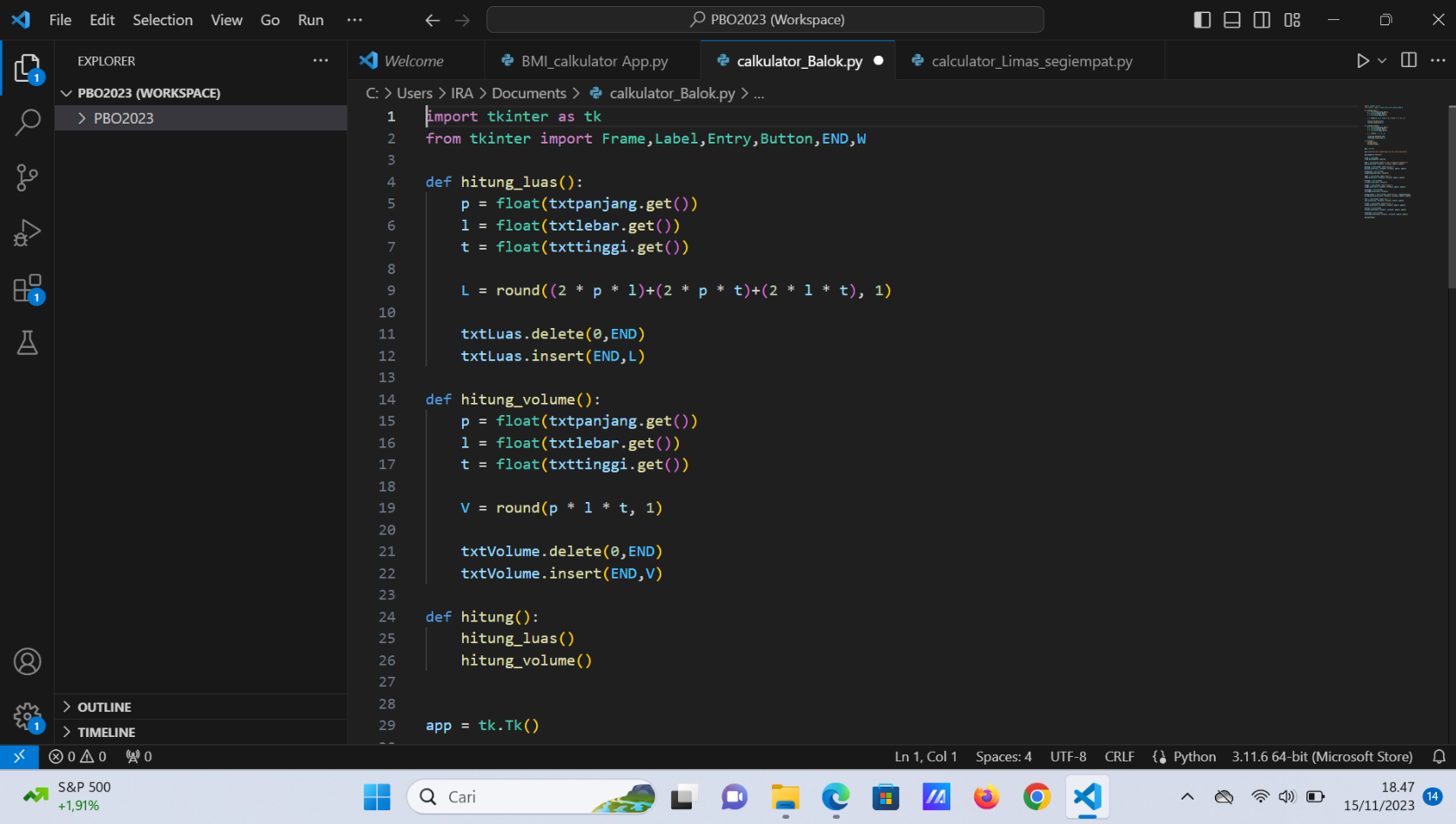
Dan hasil output dari Coding diatas sebagai berikut:



Dalam contoh ini, kita membuat jendela dengan label dan tombol menggunakan Tkinter. Ketika tombol ditekan, fungsi **‘tombol\_ditekan’** dipanggil dan mengubah teks label.

Kini ada praktek untuk mengimplementasikan contoh diatas, melanjutkan pada materi sebelumnya namun ada sedikit perbedaan yakni pada pertemuan sekarang adalah membuat calculator, maka dari itu Simak baik baik ya.

Berikut ini Langkah-langkah untuk membuat program kalkulator balok menggunakan Python:



Pertama untuk membuat Coding ini adalah

1. Mengimpor Pustaka Tkinter dan beberapa komponen yang spesifik



Disini, Pustaka ‘Tkinter’ diimpor dengan kata pengganti ‘tk’

Dan ada beberapa komponen spesifik yang penting seperti **‘Frame’, ‘Label’, ‘Entry’, ‘Button’, ‘END’,** dan **‘W’** . Semua komponen ini harus dimasukan dalam pembukaan coding pada aplikasi kalkulator yang kita buat ini.

1. Mendefinisikan fungsi-fungsi untuk perhitungan

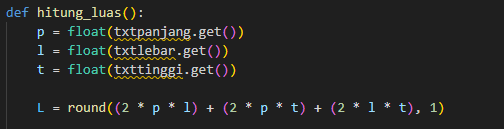


Fungsi diatas adalah untuk menghitung luas berdasarkan nilai input dari widget Entry.



Sedangkan fungsi yang ini gunanya untuk menghitung volume berdasarkan nilai input dari widget Entry.

1. Mendapatkan nilai input dan menghitung luas permukaan



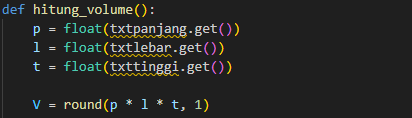
Disini, Panjang adalah **(‘p’)**, lebar **(‘l’),** dan tinggi adalah **(‘t’)** yang diperoleh dari widget Entry dan luas permukaan dihitung **(‘L’)**

1. Menghapus dan memperbaharui Hasil di GUI



Konten sebelumnya di widget entry hasil (‘txtLuas’) dihapus, dan luas permukaan yang dimasukkan.

1. mendapatkan nilai input dan menghitung Volume:



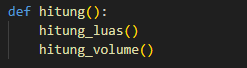
Tata cara kali ini mirip dengan luas permukaan, Panjang **(‘p’),** lebar **(‘l’)** dan tinggi **(‘t’)** diperoleh dan volume **(‘V’)** dihitung.

1. Menghapus dan memperbaharui hasil di GUI



Nah sama dengan cara pada nomer 4 kita bisa melihat hasil Entry pada konten sebelumnya **(‘txtVolume’)** dihapus, dan volume yang dihitung dimasukan.

1. Fungsi perhitungan secara keseluruhan



Untuk Fungsi ini kita memanggil dua fungsi sekaligus yakni

**‘hitung\_luas()’** dan **‘hitung\_volume()’** saat dipicu.

1. Membuat Instance aplikasi Tkinter



Jika kita ingin membuat sebuah aplikasi Tkinter maka masukan kode ini untuk instance aplikasi Tkinter, ketikkan **(‘app’)** maka aplikasi akan dibuat

1. Menjalankan Loop utama Tkinter

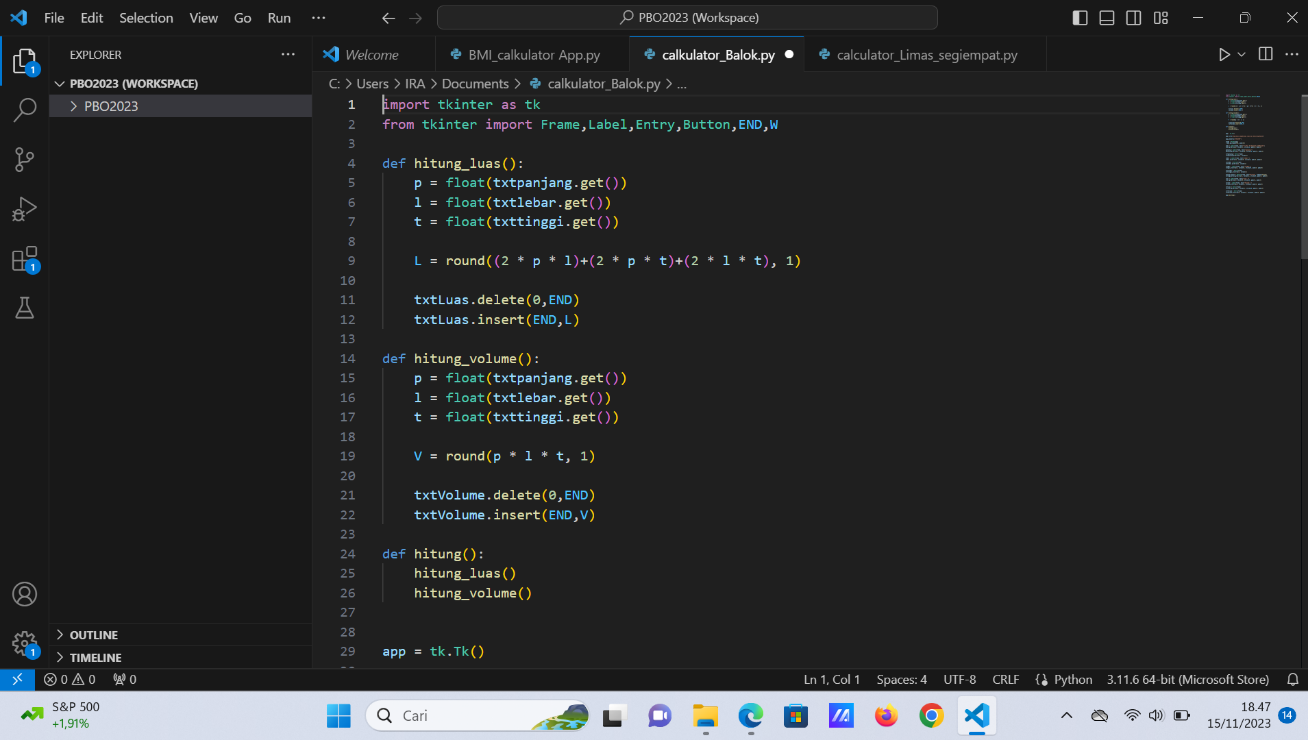
Langkah terakhir dalam membuat aplikasi kalkulator ini adalah dengan memasukan kode



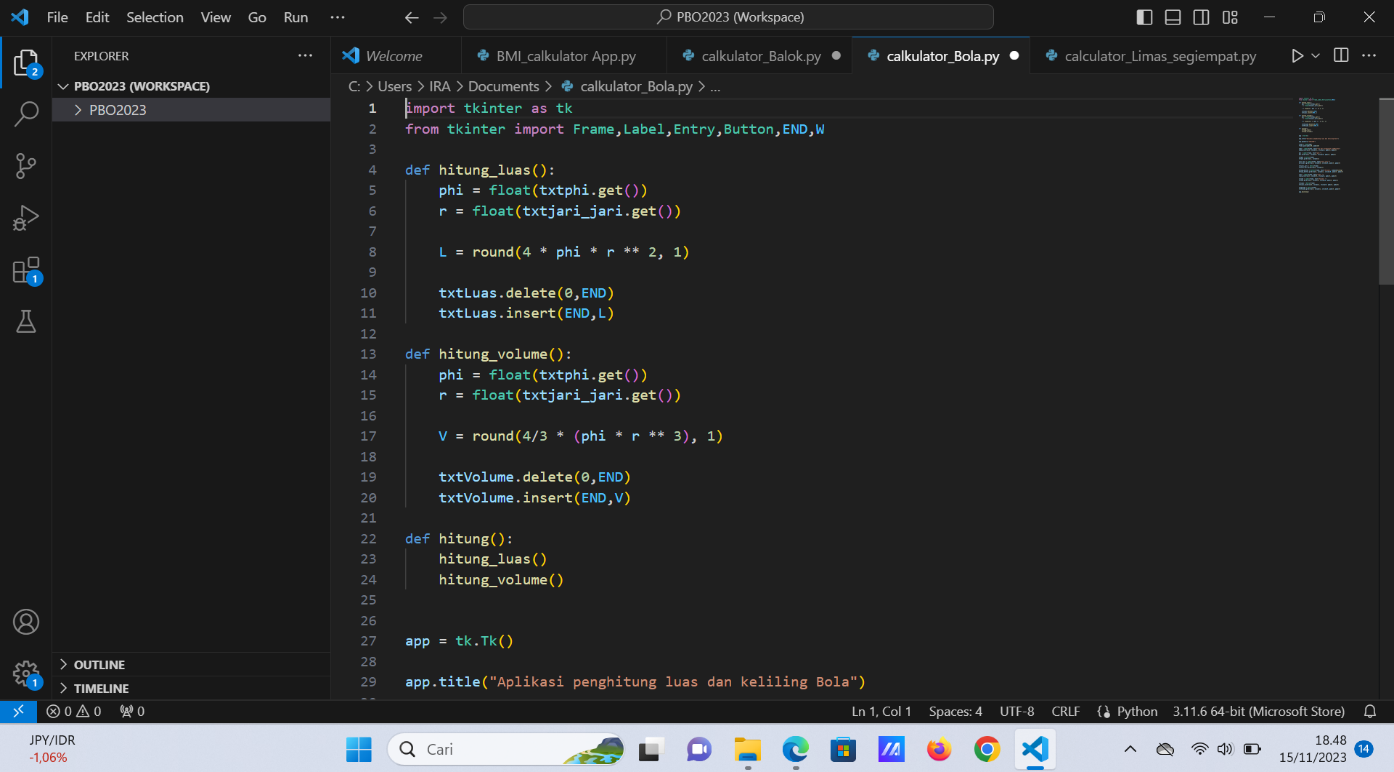
Ini akan memulai loop utama, memungkinkan aplikasi Tkinter berjalan.

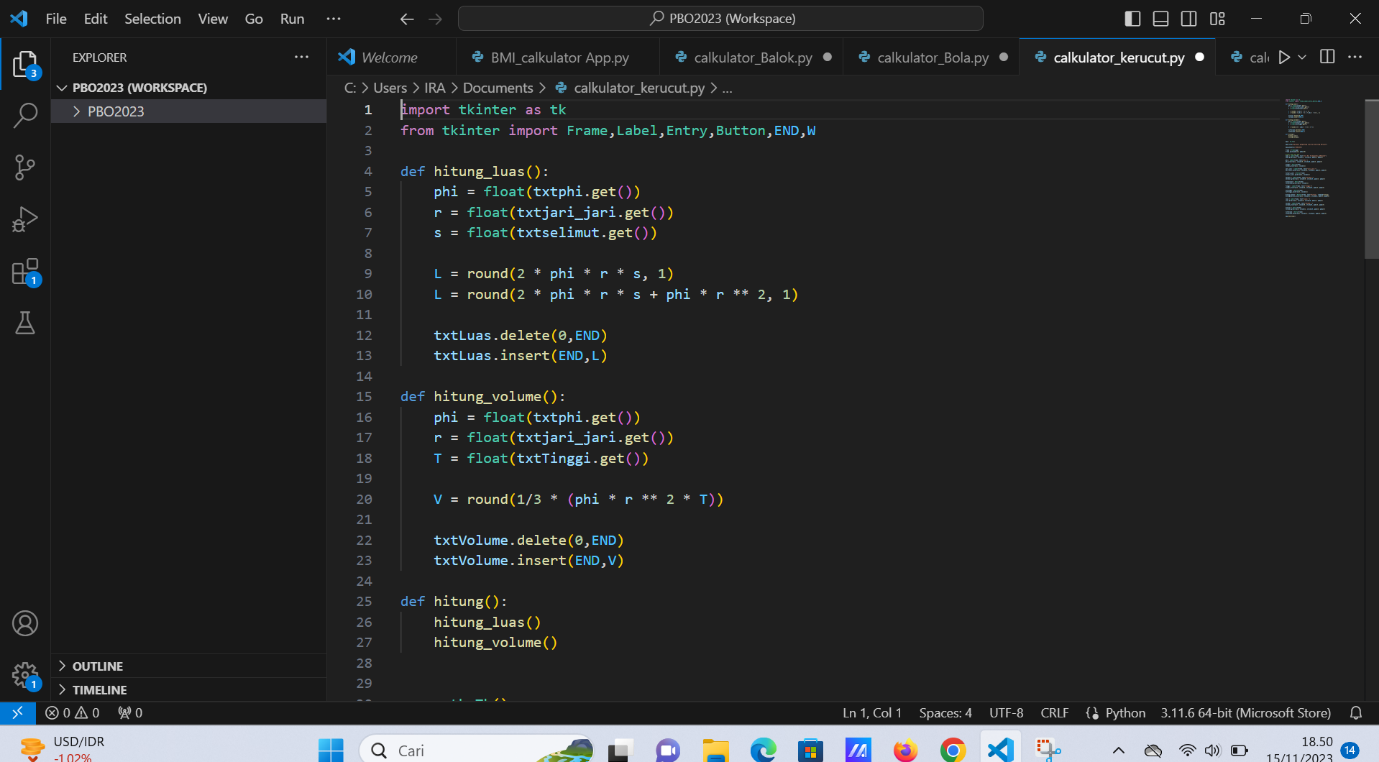
Berikut ada beberapa contoh coding untuk calculator hitungan bangun ruang

**Contoh coding kalkulator Balok**

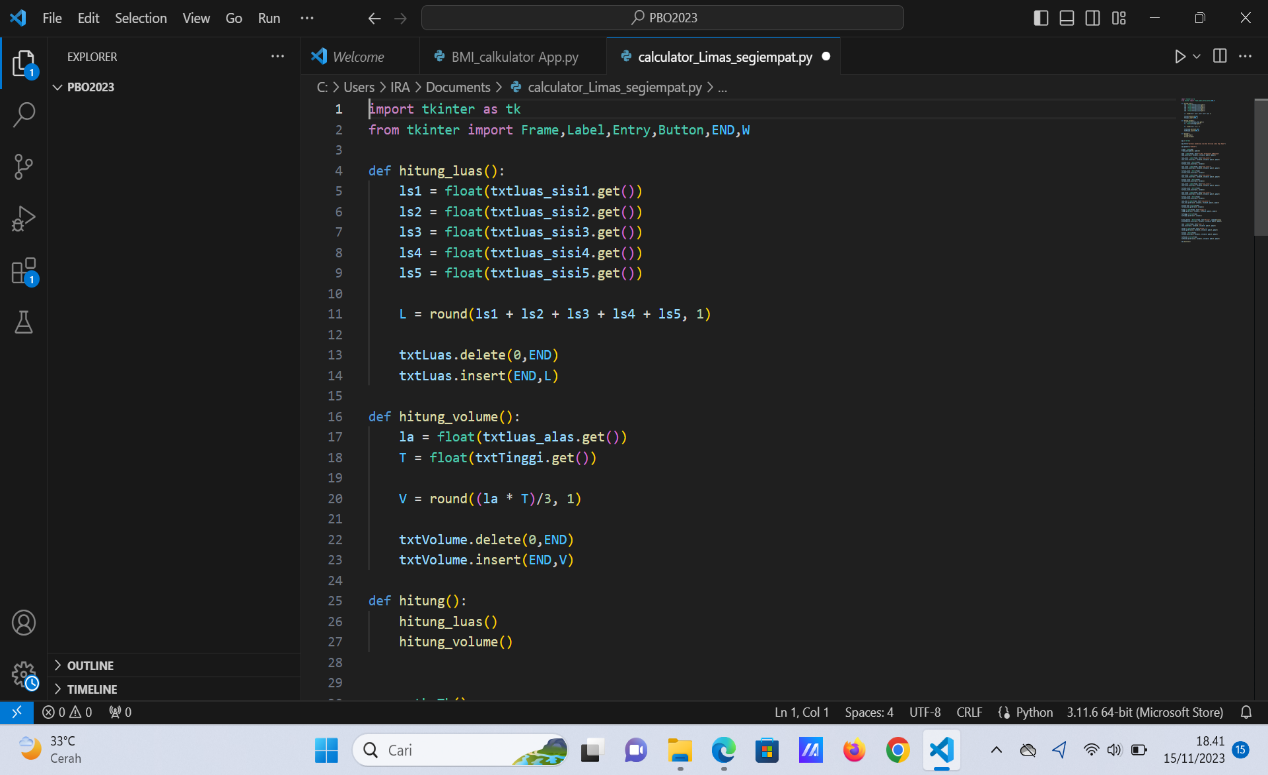


**Contoh coding Kalkulator bola**

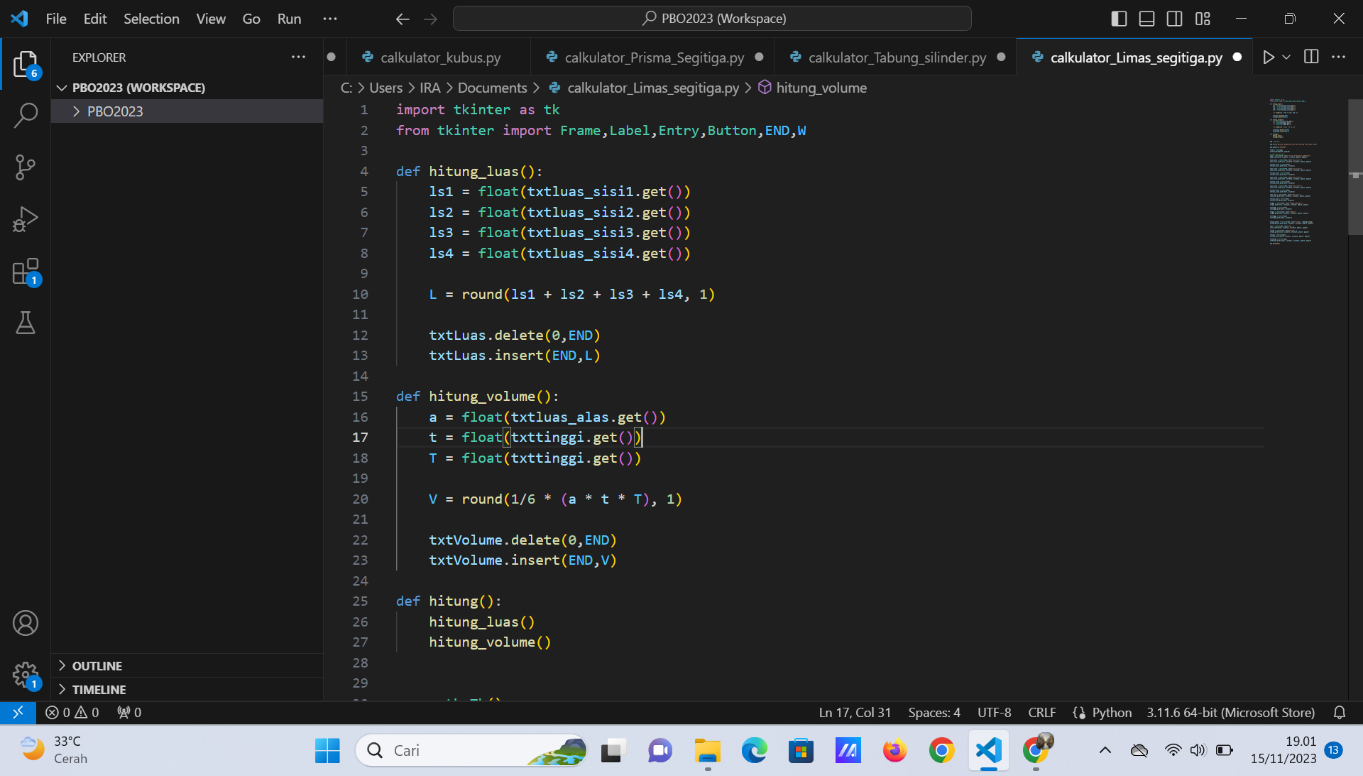


Contoh coding kalkulator Kerucut

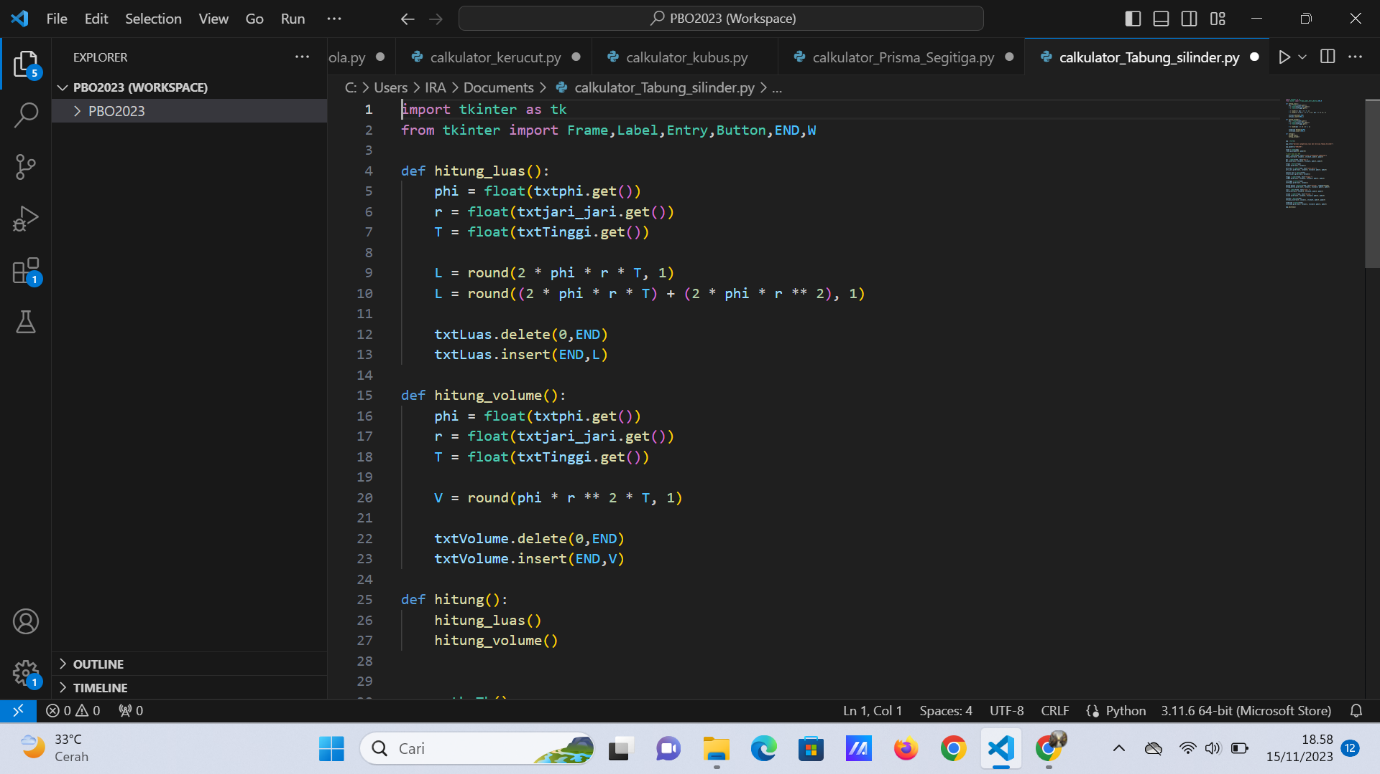
Contoh coding kalkulator Limas segiempat



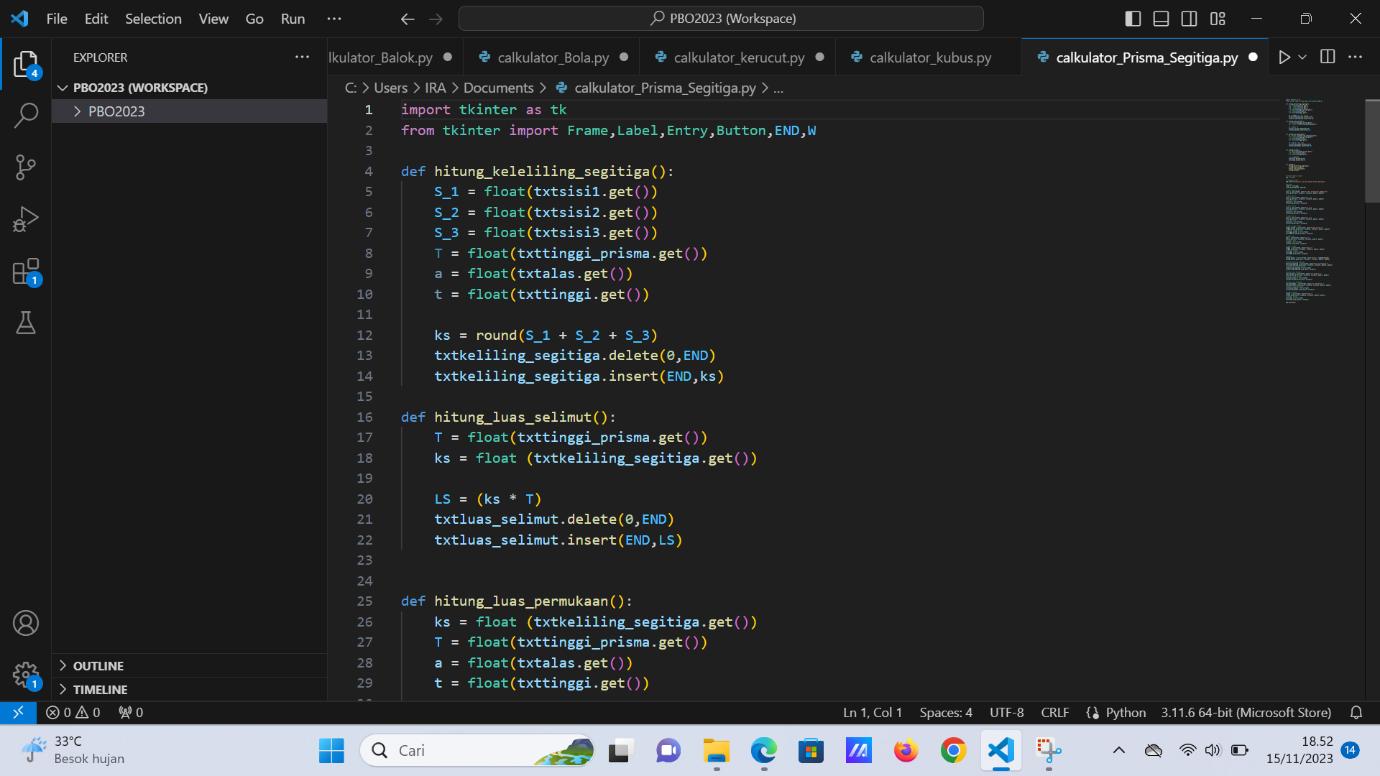
Contoh coding kalkulator Limas segitiga



Contoh coding kalkulator Silinder



Contoh coding kalkulator Prisma Segitiga



**Cara membuat Calculator BMI**

Sebelumnya apa sih BMI itu?

BMI atau dalam Bahasa Indonesia indeks masa tubuh, BMI ini jadi seperti alat ukur yang menunjukan sebandingnya berat badan dengan tinggi badan kita. Caranya, berat badan kita (Dalam kilogram) dibagi dengan tinggi badan kita (dalam meter ya) yang dipangkat 2. Nah perhitungan ini bisa jadi petunjuk buat kita apakah berat badan kita ini termasuk normal, kurang, berlebihan atau obesitas.

Makanya dipertemuan ini juga kita membuat kalkulator BMI yang nantinya mungkin bisa digunakan dalam kehidupan sehari-hari.

Berikut ini Langkah-langkah dalam membuat aplikasi kalkulator BMI

1. Import Library Tkinter



Import terlebih dahulu Pustaka tkinter dengan alias ‘tk’

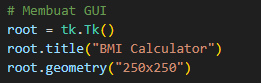
1. Membuat Fungsi

Fungsi `calculate\_bmi()` akan dijalankan saat kita menekan tombol "Hitung BMI". Fungsi ini akan mengambil tinggi dan berat badan yang dimasukkan oleh kita, lalu menghitung BMI, dan menampilkan hasilnya sesuai dengan input yang kita masukkan.



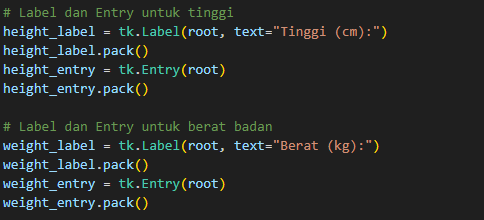
1. Membuat GUI

Kita membuat jendela aplikasi dengan memanggil `tk.Tk()`. Kemudian, kita menentukan judul jendela dengan `title()` dan ukuran jendela dengan `geometry()`



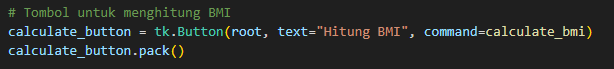
1. Menambahkan Label dan Entry untuk Tinggi dan Berat Badan

Kita membuat label dan entry untuk pengguna memasukkan tinggi dan berat badan mereka. Label digunakan untuk menampilkan teks yang menggambarkan apa yang diharapkan pengguna untuk dimasukkan di entry



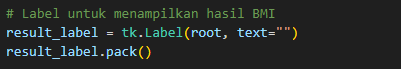
1. Menambahkan Tombol untuk Menghitung BMI

Kita menambahkan tombol dengan menggunakan `tk.Button()`. Tombol ini memiliki teks "Hitung BMI" dan akan memanggil fungsi `calculate\_bmi()` ketika ditekan



1. Menambahkan Label untuk Menampilkan Hasil BMI dan Kategori

Kita menambahkan dua label yang akan digunakan untuk menampilkan hasil BMI dan kategori BMI. Awalnya, label-label ini kosong dan akan diisi setelah pengguna menghitung BMI

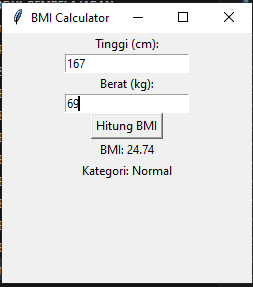


1. Menjalankan Aplikasi GUI

Terakhir, kita memanggil `root.mainloop()` untuk menjalankan aplikasi GUI kita. Ini akan membuat jendela GUI muncul dan menunggu input pengguna.



Dan akan keluar hasil Output yang seperti ini



**-Selesai-**

Pertemuan 4

(MP3 dan MP4)

Penjelasan singkat tentang Mp3 dan Mp4

MP3 adalah format audio untuk menyimpan musik digital dengan ukuran file kecil dan kualitas suara yang baik, seperti musik, podcast, dan audiobook.

MP4 adalah format file multimedia yang dapat menyimpan audio, video, teks, dan data lainnya dalam satu file. pengodean video dan audio berkualitas tinggi serta kompresi efisien, sering digunakan untuk menyimpan video digital dengan dukungan untuk fitur-fitur tambahan seperti subtitle dan metadata. Digunakan untuk pemutaran audio dan video.

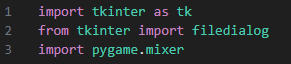
Nah kali ini kita akan belajar membuat aplikasi pemutar musik dan video versi sederhana dengan Bahasa pemrograman python.

Tanpa bas abasi lagi mari kita mulai

Cara membuat aplikasi pemutar Mp3

1. Import Modul

Pertama-tama, modul tkinter dan modul filedialog dari tkinter diimpor. Selain itu, modul pygame.mixer diimpor untuk memutar file audio.

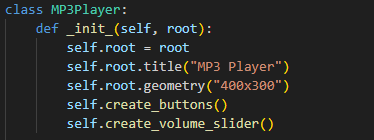


1. Membuat Kelas MP3Player

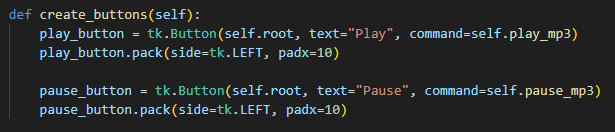
Kode kemudian mendefinisikan kelas `MP3Player` yang akan digunakan untuk membuat GUI dan mengatur pemutaran file MP3.

1. Inisialisasi Objek MP3Player

Dalam fungsi `\_\_init\_\_` dari kelas `MP3Player`, jendela tkinter diatur dengan judul "MP3 Player" dan ukuran 400x300 piksel. Metode `create\_buttons()` dan `create\_volume\_slider()` digunakanl untuk membuat tombol pemutaran dan pengaturan volume.



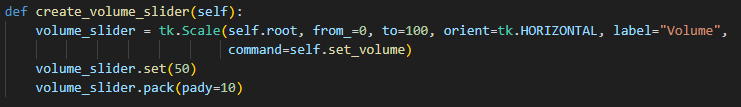
1. Membuat Tombol Pemutaran



Metode `create\_buttons()` membuat tombol untuk memutar dan menjeda file MP3. Ketika tombol "Play" ditekan, metode `play\_mp3()` dipanggil, dan ketika tombol "Pause" ditekan, metode `pause\_mp3()` dipanggil.

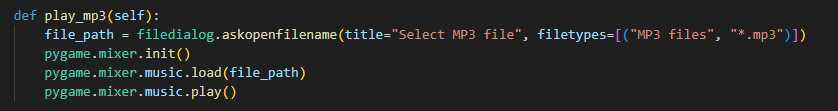
1. Membuat Slider Volume

Metode `create\_volume\_slider()` membuat slider untuk mengatur volume pemutaran. Nilai awalnya ditetapkan ke 50. Ketika slider digeser, metode `set\_volume()` dipanggil untuk menyesuaikan volume.



1. Memutar File MP3

Metode `play\_mp3()` meminta pengguna untuk memilih file MP3 yang ingin dimainkan menggunakan file dialog. Kemudian, pygame.mixer diinisialisasi dan file MP3 dimuat dan diputar menggunakan `pygame.mixer.music.load()` dan `pygame.mixer.music.play()`.



1. Menjeda Pemutaran MP3

Metode `pause\_mp3()` menjeda pemutaran file MP3 menggunakan `pygame.mixer.music.pause()`.



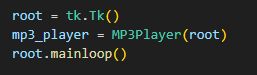
1. Mengatur Volume

Metode `set\_volume()` mengatur volume pemutaran menggunakan `pygame.mixer.music.set\_volume()`.



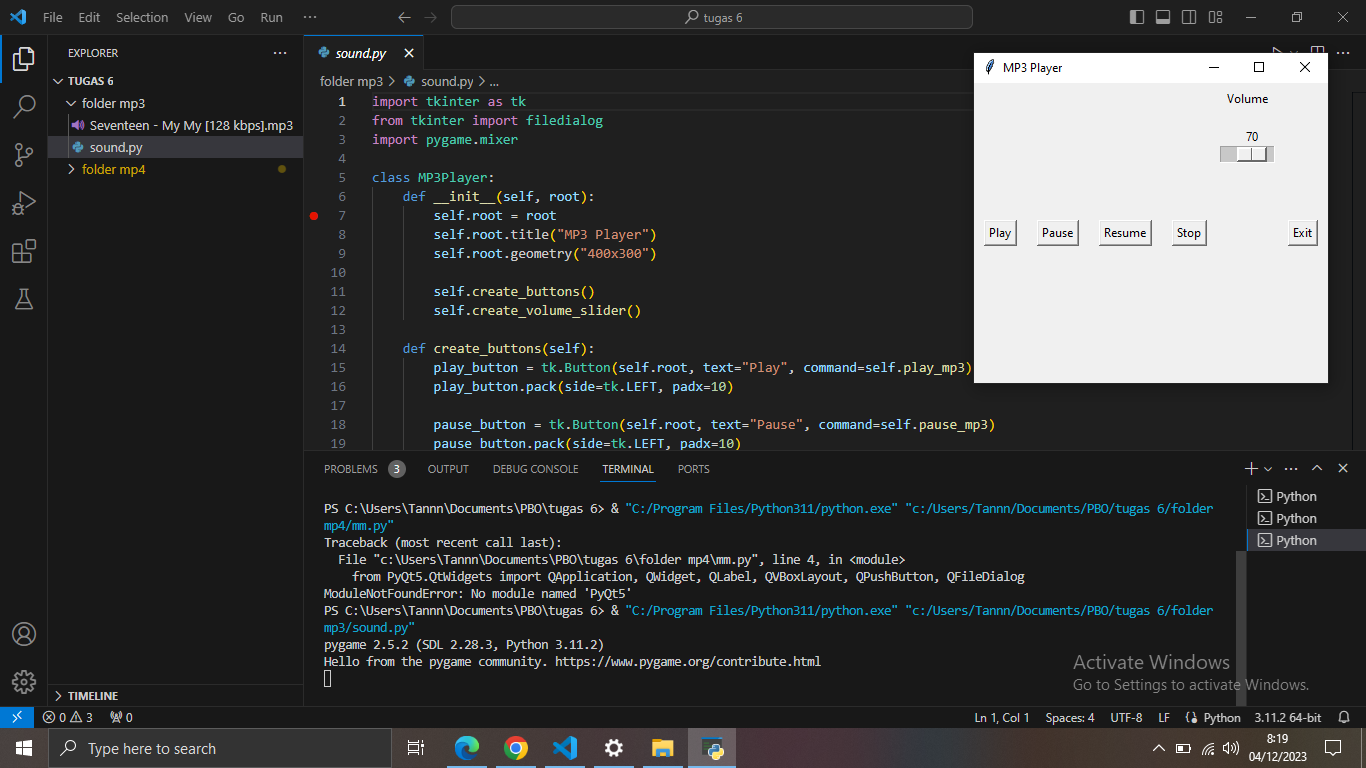
1. Membuat Jendela Tkinter dan Objek MP3Player

Objek `MP3Player` dibuat di dalam jendela tkinter, dan kemudian `root.mainloop()` dipanggil untuk menjalankan aplikasi GUI.



Setelah semua coding dimasukan, kita juga perlu memasukan file Mp3 yang akan kita putar, jangan sampai lupa ya.

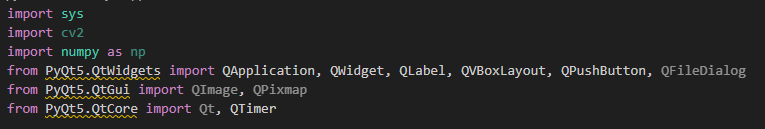
Berikut hasil output dari aplikasi MP3 yang kita buat



Cara membuat aplikasi pemutar MP4 :

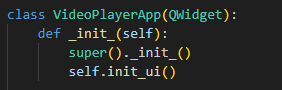
1. Import Modul

Impor modul yang diperlukan. `sys` untuk mengelola argumen baris perintah, `cv2` untuk pemrosesan video menggunakan OpenCV, dan modul PyQt5 untuk membuat antarmuka pengguna.



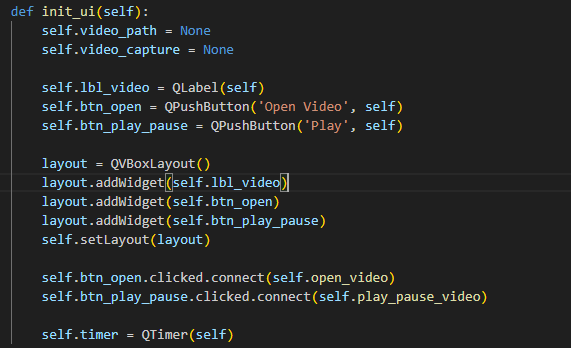
1. Definisikan Kelas VideoPlayerApp

Kelas ini membawa dari `QWidget` dan digunakan untuk membuat antarmuka pengguna video player.



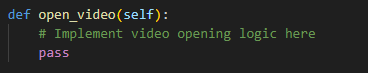
1. Inisialisasi UI

Metode `init\_ui` menginisialisasi elemen-elemen antarmuka pengguna, seperti label untuk menampilkan video, tombol untuk membuka video, dan tombol untuk memainkan atau menjeda video



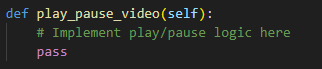
1. Metode `open\_video`

Metode ini akan dipanggil ketika tombol "Open Video" ditekan. Anda perlu mengimplementasikan logika untuk membuka video dari dialog file.



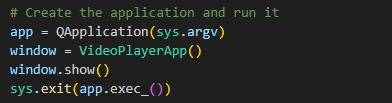
1. Metode `play\_pause\_video`

Metode ini akan dipanggil ketika tombol "Play/Pause" ditekan. Anda perlu mengimplementasikan logika untuk memainkan atau menjeda video.



1. Buat Aplikasi dan Jalankan

Buat aplikasi PyQt5, buat objek `VideoPlayerApp`, tampilkan aplikasi, dan jalankan aplikasi menggunakan `sys.exit(app.exec\_())`



Pertemuan 5

(Quiz)

Pada pertemuan kelima ini beda dari sebelum sebelumnya, jika kemarin kita diberikan tugas untuk membuat aplikasi dengan coding bahasa pemrograman python maka kali ini kita hanya akan mengerjakan Quiz.

Pada Quiz ini kita diberikan kesempatan untuk mengerjakan soal Quiznya beberapa kali dan ambil nilai yang paling besar dari hasil percobaan itu.

Ini hasil Quiz saya pada beberapa percobaan



Pertemuan 6

(Membuat Kalkulator)

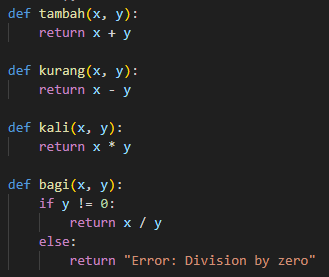
Pertemuan ini kita ditugaskan nih untuk membuat aplikasi perhitungan secara berkelompok,

Kalkulator adalah alat atau aplikasi di hp yang dipakai untuk menghitung angka. Misalnya untuk tambah, pengurangan, perkalian, atau pembagian. Selain itu, bisa juga untuk menghitung hal-hal yang lebih rumit seperti akar kuadrat, pangkat, atau trigonometri. Biasanya kalkulator ini berguna untuk keperluan sehari-hari, misalnya untuk tugas matematika, keuangan, atau teknis.

nah di modul praktikum ini saya akan menjelaskan bagaimana caranya membuat aplikasi hitung-hitungan alias Kalkulator.

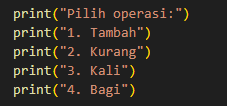
1. Definisi Fungsi Operasi Matematika

Pertama, kita punya fungsi-fungsi untuk operasi matematika dasar seperti tambah, kurang, kali, dan bagi. Ini contohnya:



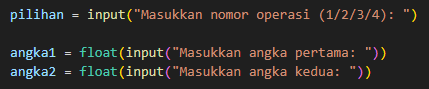
1. Tampilkan Pilihan Operasi

Terus, jika kita mau memberi opsi operasi ke user. Ini kodenya:



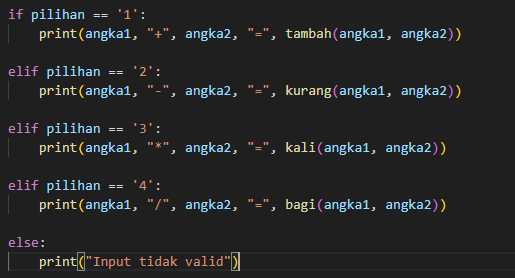
1. Terima Input dari user

Selanjutnya, kita minta user masukan nomer operasi yang mereka mau dan dua angka buat dioperasikan:

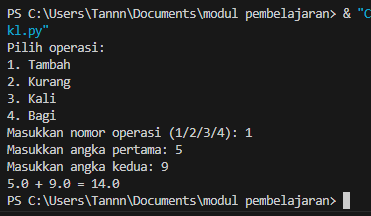


1. Olah Operasi yang Dipilih

Berdasarkan pilihan user, kita panggil fungsi yang sesuai untuk melakukan operasi matematika:



Hasil Output dari kode yang kita buat



Ini kode lengkapnya:



-Selesai-

Pertemuan 7

(Konverter Suhu)

Apa sih koverter suhu itu?

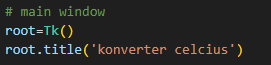
Konverter suhu adalah alat atau perangkat lunak yang digunakan untuk mengubah suhu dari satu satuan ke satuan lainnya. Misalnya, dari Celsius ke Fahrenheit, Kelvin ke Celsius, atau sebaliknya. Konverter suhu juga membantu dalam berbagai situasi di mana kamu perlu mengukur atau memahami suhu dalam berbagai skala. Koverter suhu Ini dapat digunakan dalam banyak bidang loh, seperti ilmu pengetahuan, masakan, industri, dan aplikasi sehari-hari dll.

Maka dari itu di pertemuan kali ini kita ditugaskan untuk membuat aplikasi converter suhu sederhana.

Berikut ini Langkah langkahnya

1. Setup Aplikasi

Pertama, kita setup aplikasi dengan membuat instance Tkinter pake root=Tk() dan beri judul "konverter celcius" ke window.



1. Import Tkinter

Import modul Tkinter untuk membuat GUI



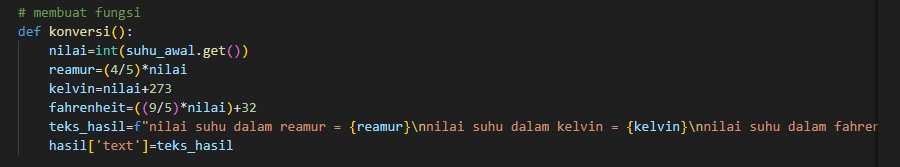
1. Buat Main Window

Ini jadi main window aplikasinya



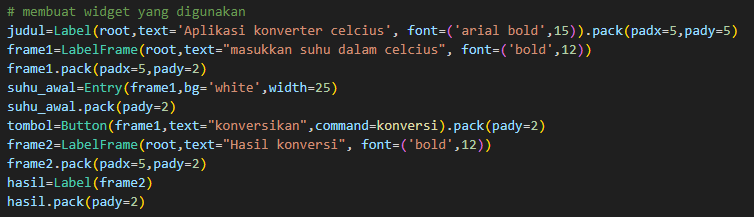
1. Definisi Fungsi Konversi

Difinisikan fungsi konversi untuk ngehitung suhu dalam Reamur, Kelvin, dan Fahrenheit.



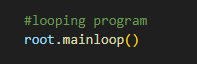
5. Buat Widget-Widget

- Buat label, frame, entry, tombol, dan label buat hasil konversi

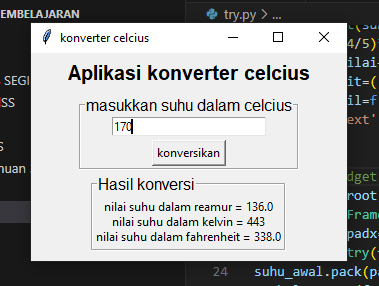


1. Looping Program

Dan yang terakhir adalah Jalankan programnya dengan root.mainloop()



Jika sudah mengikuti semua rangkaian diatas maka akan keluar Hasil output yang seperti ini.



Pertemuan 8

(UTS)

Pada pertemuan ini kita sedang melaksanakan UTS ada 3 bagian yakni

(MP3, Konversi suhu, dan Translate) Namun pada coding yang ada pada modul ini telah diperbaiki sehingga bisa berfungsi dengan benar.

Langsung saja kita ke pembahasan:

Mp3

1. Impor Modul

Kita hasur mengimpor modul yang diperlukan. kali ini, kita mengimpor `pygame` untuk pemutaran musik dan `colorama` untuk mengatur warna output konsol.



1. Fungsi `play\_music`

Ini adalah fungsi yang bertanggung jawab untuk memutar file musik. Fungsi ini menerima perintah yang merupakan nama file musik yang akan diputar.



1. Try-Except Block

Sekarang Kita menggunakan blok `try-except` untuk menangani pengecualian yang mungkin terjadi saat menjalankan kode, apalagi saat program dihentikan oleh kita.



1. Inisialisasi Pygame

Kita inisialisasi dulu modul `pygame` untuk mengoptimalkan fungsinya



1. Memuat dan Memutar Musik

Kode ini untuk memuat file musik yang diberikan dan memulai pemutaran musik.



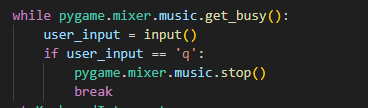
1. Pesan Output

Kita mencetak pesan ke layar menggunakan `colorama` untuk memberikan warna pada teks.



1. Pemutaran Musik dan Input Pengguna

Kita masuk ke dalam loop yang akan terus memeriksa apakah musik masih diputar dan jika bisa menekan tombol 'q', lalu musik akan dihentikan.



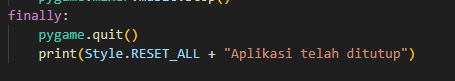
1. Tindakan Pengecualian

Kita menambahkan tindakan pengecualian untuk memastikan bahwa musik dihentikan dengan benar.



1. Penutup dan Membersihkan

Setelah musik selesai diputar atau dihentikan, kita akan keluar dari loop, menutup modul `pygame`, dan mencetak pesan penutup.

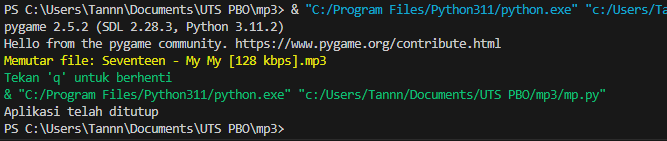


1. Fungsi Utama

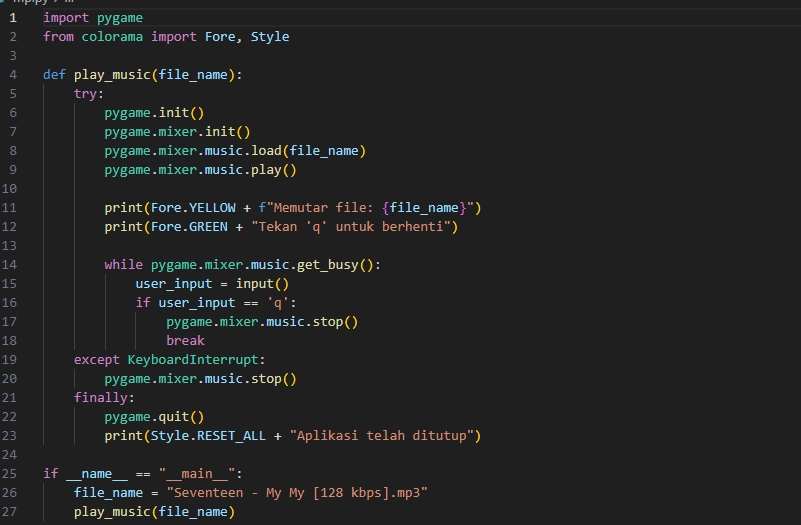
Pada bagian ini, kita memeriksa apakah file ini dijalankan sebagai skrip utama. Jika ya, kita menetapkan nama file musik yang akan diputar dan memanggil fungsi `play\_music`.



Hasil output



Ini adalah kode lengkap yang sudah diperbaiki.



Konversi Suhu

1. Impor Pustaka

Pertama, kode mengimpor pustaka yang diperlukan dari modul Colorama. Ini termasuk `Fore` (untuk warna depan), `Back` (untuk warna belakang), `Style` (untuk gaya teks), dan `init` (untuk inisialisasi).



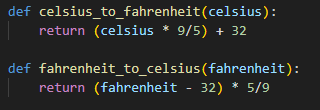
1. Inisialisasi Colorama

Kode Colorama dengan `autoreset=True`, yang berarti warna akan diatur kembali ke default setelah setiap cetakan.



1. Fungsi Konversi Suhu

Kemudian, kode mendefinisikan dua fungsi, `celsius\_to\_fahrenheit` dan `fahrenheit\_to\_celsius`, yang melakukan konversi suhu sesuai rumus yang ada.



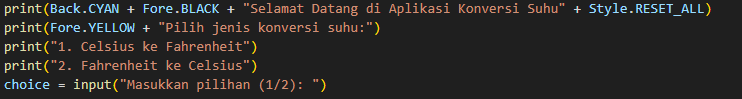
1. Fungsi Utama

Ini fungsi utama dari program yang akan dijalankan.



1. Pengaturan Teks Output

Kode mencetak pesan pembuka yang menampilkan selamat datang dan instruksi untuk memilih jenis konversi suhu. Pesan-pesan ini diberi warna menggunakan modul Colorama.



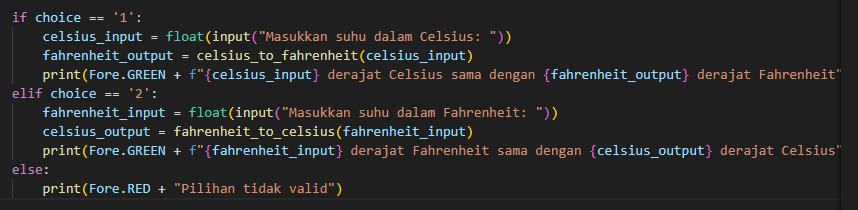
1. Meminta Input pada user

User diminta untuk memilih jenis suhu dengan memasukkan angka 1 /2 sesuai pilihan mereka.



1. Logika Konversi

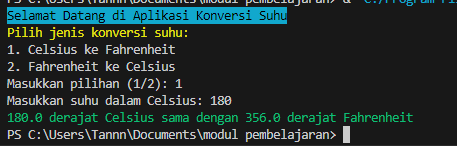
Berdasarkan pilihan pengguna, program akan menjalankan fungsi yang sesuai untuk melakukan konversi suhu.



1. Menjalankan Fungsi Utama

Terakhir, kode memeriksa apakah file ini sedang dijalankan sebagai skrip utama, dan jika ya, menjalankan fungsi `main()`.



Hasil Outputnya adalah

KODE LENGKAP YANG SUDAH DIPERBAIKI  


Translator

1. Impor Modul

Seperti biasa Langkah pertama selalu Di Mulai dengan mengimpor modul `Translator` dari `googletrans`. Modul ini akan digunakan untuk melakukan terjemahan teks.



1. Fungsi `translate\_text`

Fungsi ini tugasnya menterjemahan teks dari bahasa asal ke bahasa tujuan.



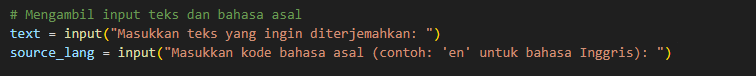
1. Inisialisasi Objek Translator

Objek `translator` itu dibuat untuk melakukan terjemahan menggunakan modul `googletrans`.



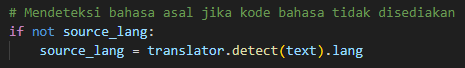
1. Input Teks dan Bahasa Asal

user nanti diminta memasukkan teks yang ingin diterjemahkan dan kode bahasa asal.



1. Mendeteksi Bahasa Asal

kalau kode bahasa asal tidak disediakan, program akan mencoba mendeteksi bahasa asal dari teks yang diberikan.



1. Input Bahasa Tujuan

User nanti akan memasukkan kode bahasa tujuan untuk terjemahan.



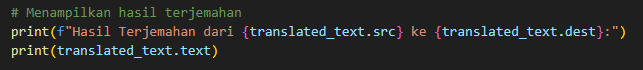
1. Melakukan Terjemahan

Karena kita menggunakan objek `translator`, program melakukan terjemahan dari bahasa asal ke bahasa tujuan.



1. Menampilkan Hasil Terjemahan

Hasil terjemahan nantinya akan ditampilkan ke layar.

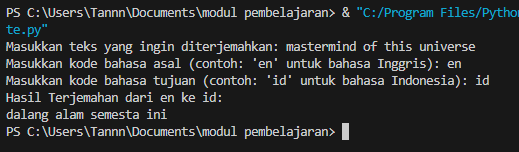


1. Fungsi Utama

Bagian ini program akan memeriksa file. lalu fungsi `translate\_text` dipanggil



Hasil Output



Kode yang sudah diperbaiki dari sebelumnya



Pertemuan 9

(Konversi suhu dengan oop terstruktur)

Dalam pertemuan di minggu ke 9 ini kita akan belajat tentang Class dalam Bahasa pmerograman Python, kelas adalah struktur yang mengelompokkan fungsi dan data bersama-sama dalam satu tempat. Kelas berisi atribut (variabel) dan metode (fungsi) yang mendefinisikan perilaku objek yang akan dibuat dari kelas itu. Objek adalah contoh dari kelas, yang mewakili satu golongan dalam program.

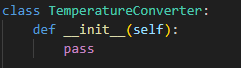
Lalu kita akan mengerjakan bagaimana caranya mengaplikasikan Class dalam pembuatan aplikasi konversi Suhu dibawah ini.

Konversi suhu dengan OOP terstruktur:

1. Definisi Kelas

Pertama mulai dengan Membuat kelas `TemperatureConverter` yang fungsinya untuk melakukan konversi suhu. Kode di dalamnya terdiri dari dua metode yaitu `celsius\_to\_fahrenheit` untuk mengonversi suhu dari Celsius ke Fahrenheit dan `fahrenheit\_to\_celsius` untuk mengonversi suhu dari Fahrenheit ke Celsius.

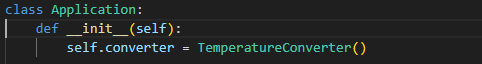
Lalu pada metode `\_\_init\_\_`, tidak ada implementasi karena kelas.



1. Definisi Kelas 2

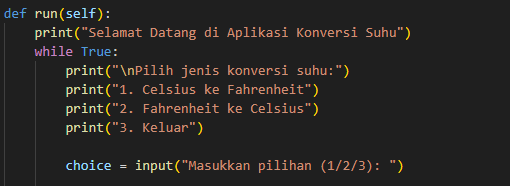
lalu Kita akan membuat kelas `Application` sebagai antarmuka pengguna. Kode yang di dalamnya menciptakan objek `TemperatureConverter` saat inisialisasi.

Lalu Metode `run` digunakan untuk menjalankan aplikasi, yang memberikan opsi kepada pengguna untuk memilih jenis konversi suhu dan memasukkan nilai suhu yang akan dikonversi.



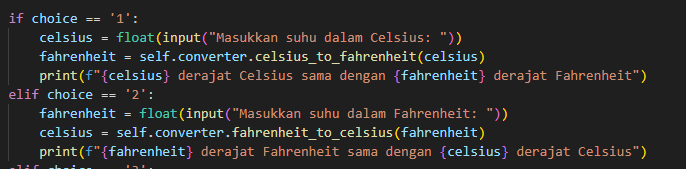
1. Pemilihan Operasi Konversi

Program akan menampilkan opsi jenis konversi suhu (Celsius ke Fahrenheit atau sebaliknya) dan meminta pengguna untuk memilih opsi tersebut.



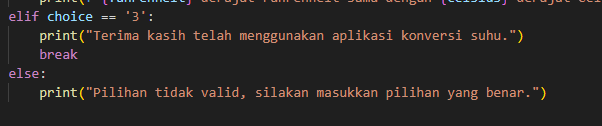
1. Pengguna Memasukkan Nilai Suhu

Setelah kita memilih jenis konversi, kita diminta untuk memasukkan nilai suhu yang akan dikonversi.

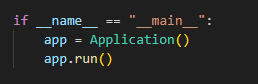


1. Menampilkan Hasil Konversi

Hasil konversi suhu ditampilkan.



1. Hasil Main



Pertemuan 10

(Pengenalan MySQL)

Pada pertemuan ini kita berganti ke materi selanjutnya yakni MySQL.

Apa sih MySQL itu?

MySQL itu semacam gudang data, tempat Dimana kita bisa nyimpen segala macam info penting, kayak daftar teman, pesanan belanja online, atau postingan di medsos. Jadi, MySQL itu membuat kita bisa simpan, atur, dan akses data dengan gampang. Kalo kita butuh tempat buat nimbun banyak info dan pengen aksesnya cepat, MySQL bisa jadi solusinya.

Ada nih beberapa hal pendukung kalua kita mau menjalankan MySQL

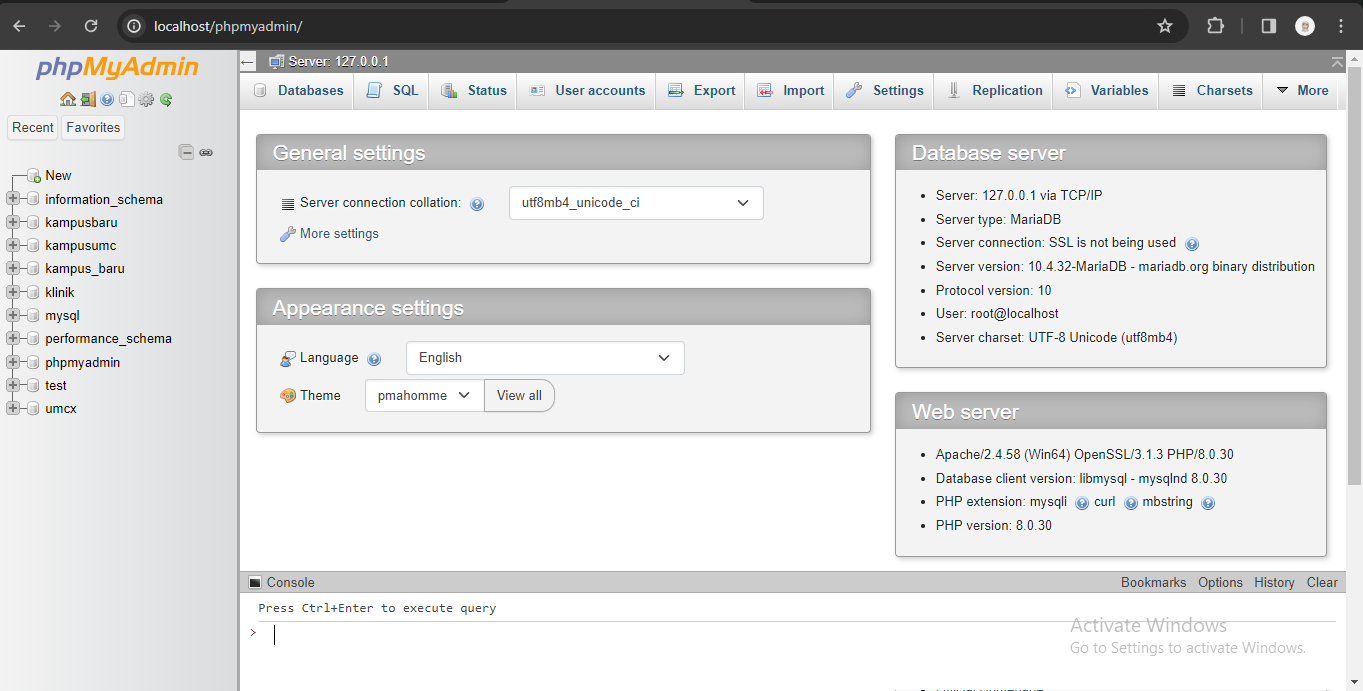
Kalau aplikasi ada XAMMP Control Panel

Kalau web Ada 2 yang pertama ada PhpMyAdmin atau sering disebut PMA

Dan satunya ada FREEDB.TECH ini biasanya digunakan untuk data Base ya.

Tapi dipertemuan ini kita hanya disuruh untuk memasukan Sreen shoot dari account PhpMyAdmin kita sebagai bentuk awal untuk pengenalan materi MySQL.

Berikut Fotonya



-Selesai-

Pertemuan 11

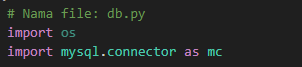
(Python database)

Python database adalah kemampuan Python untuk berinteraksi dengan berbagai jenis database, seperti MySQL, SQLite, PostgreSQL, MongoDB, dan lainnya. Dengan menggunakan modul dan library tertentu, Python dapat melakukan koneksi ke database, mengirim query SQL, mengambil dan memanipulasi data, serta melakukan operasi CRUD (Create, Read, Update, Delete) lainnya.

Setelah penjelasan singkat diatas kita akan langsung belajar cara membuat database.

1. Import Libraries

Import modul yang dibutuhkan. Dalam hal ini, kita mengimpor modul `os` dan `mysql.connector`.



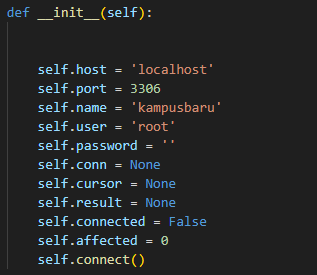
1. Buat Kelas DBConnection

kelas `DBConnection` akan bertanggung jawab untuk mengelola koneksi ke database.



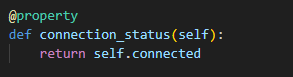
1. Inisialisasi

Pada `\_\_init\_\_`, inisialisasi parameter-parameter yang diperlukan untuk koneksi database seperti `host`, `port`, `name` (nama database), `user`, dan `password`. Kemudian, panggil metode `connect()` untuk membuat koneksi saat objek kelas dibuat.



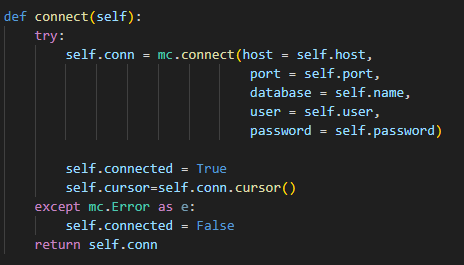
1. Metode Connection Status

Definisikan `connection\_status` sebagai properti yang mengembalikan status koneksi saat ini.



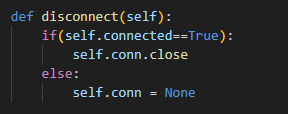
1. Metode Connect

Bagian Ini bertanggung jawab untuk membuat koneksi ke database menggunakan informasi yang telah diinisialisasi. Jika koneksi berhasil, status koneksi diatur ke True, jika tidak, status koneksi diatur ke False.



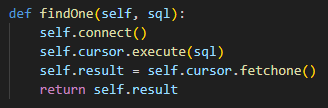
1. Metode Disconnect

Metode ini gunanya untuk memutus koneksi dari database. Jika koneksi masih aktif, maka koneksi akan ditutup.



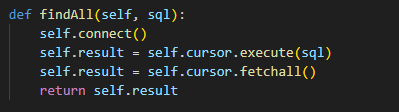
1. Metode findOne

digunakan untuk mengeksekusi query SQL dan mengembalikan satu baris kode.



1. Metode findAll

ini digunakan untuk mengeksekusi dan mengembalikan semuanya.



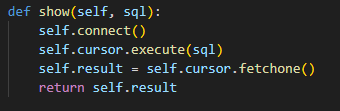
1. Insert, update, delete

Metode-metode ini digunakan untuk mengeksekusi SQL untuk operasi CRUD (Create, Read, Update, Delete) dan mengembalikan jumlah baris yang terpengaruh.



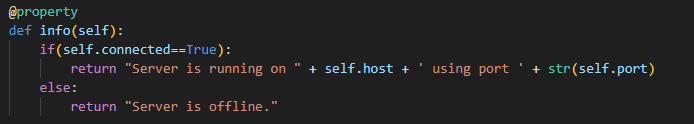
1. Show

Metode ini mirip dengan `findOne`, tapi hanya mengembalikan satu baris pertama yang ditemukan.



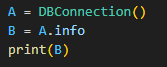
1. Info

Metode ini mengembalikan informasi status server database, apakah sedang berjalan atau tidak, jika berjalan maka aka nada tulisan “Server is running on”.



1. Instance A dan B

Membuat instance dari kelas `DBConnection` dan memanggil metode `info` untuk mendapatkan informasi status server database. Kemudian mencetak informasi tersebut.



Mata kuliah Py

1. Import Library dan Modul

Kode mengimpor kelas `DBConnection` dari modul `db`, yang gunanya untuk melakukan koneksi dan interaksi dengan database MySQL.



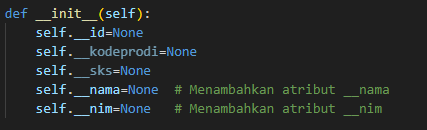
1. Kelas `Matakuliah’

Ini kelas utama yang isinya metode-metode untuk menyimpan, memperbarui, menghapus, dan mengambil data mahasiswa dari database yang kita buat di MySQL



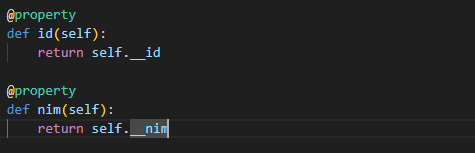
1. Atribut

Dalam `\_\_init\_\_`, beberapa atribut yang berkaitan dengan data mahasiswa seperti `\_\_id`, `\_\_kodeprodi`, `\_\_sks`, `\_\_nama`, dan `\_\_nim` diinisialisasi. double underscore (`\_\_`) menandakan bahwa atribut tersebut bersifat privat, artinya tidak bisa diakses secara langsung dari luar kelas.



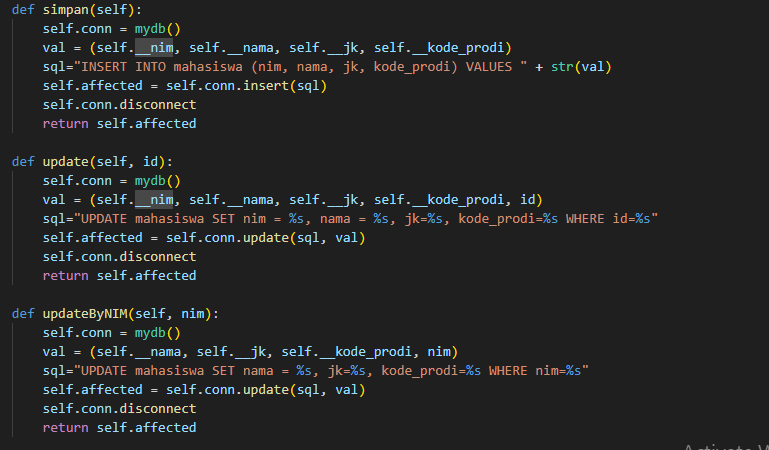
1. Property dan Setter

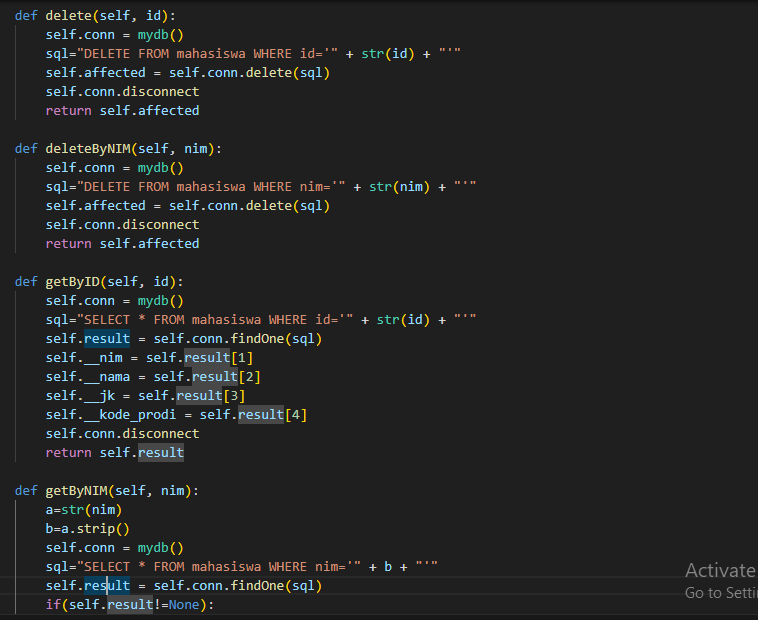
Untuk setiap atribut, ada property dan setter yang digunakan untuk mengakses dan mengubah nilai atribut tersebut dengan mode yang terkendali.

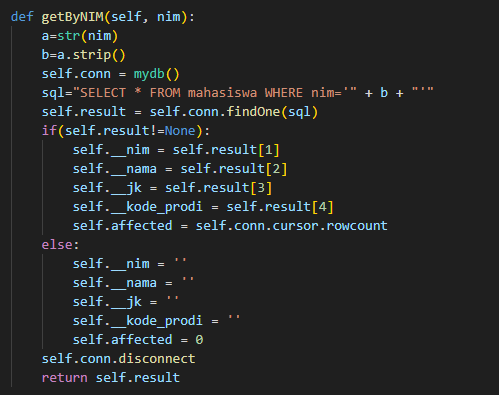


1. CRUD

Kelas `Matakuliah` memiliki metode-metode seperti `simpan`, `update`, `delete`, `getByID`, `getByNIM`, dan `getAllData` yang digunakan untuk melakukan operasi CRUD pada data mahasiswa di database.

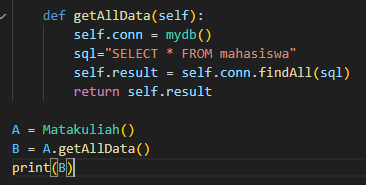






1. Pemanggilan Metode pada Objek A

Setelah kelas `Matakuliah` didefinisikan, sebuah objek `A` dibuat menggunakan kelas tersebut. Kemudian, metode `getAllData` dipanggil pada objek tersebut untuk mendapatkan semua data mahasiswa dari database, yang kemudian dicetak untuk ditampilkan.



FRM Matakuliah

1. Import Libraries

Seperti biasa Langkah pertama adalah mengimpor modul `tkinter` sebagai `tk` dan beberapa kelas dan fungsi lain yang dibutuhkan dari modul tersebut.



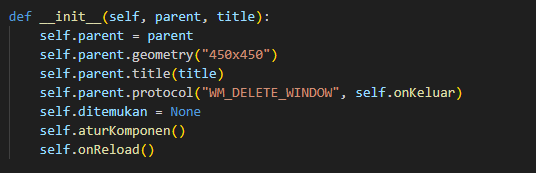
1. Kelas `FormMahasiswa’

Kelas ini bertanggung jawab untuk membuat dan mengelola GUI untuk aplikasi.



1. Metode \_\_init\_\_

Metode konstruktor yang dipanggil ketika objek kelas dibuat. Metode ini mengatur konfigurasi utama jendela aplikasi dan memanggil metode `aturKomponen()` dan `onReload()`.



1. Atur Komponen

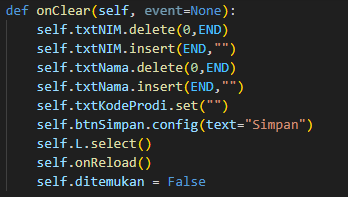
Bagian ini mengatur komponen-komponen GUI seperti label, entry fields, radio buttons, combobox, tombol, dan treeview (untuk menampilkan data dalam tabel).





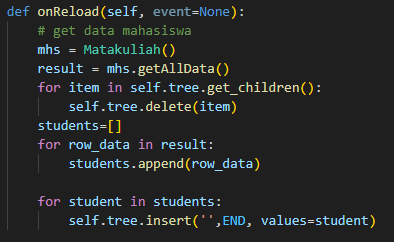
1. OnClear

OnClear digunakan untuk membersihkan atau mengosongkan field input setelah tombol "Clear" ditekan.



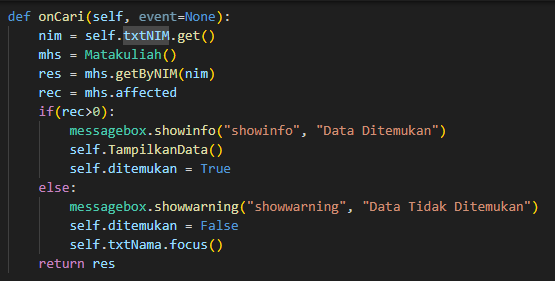
1. OnReload

Kegunaan OnReload untuk memuat kembali data mahasiswa ke dalam treeview setelah ada perubahan data atau saat aplikasi dimuat pertama kali.



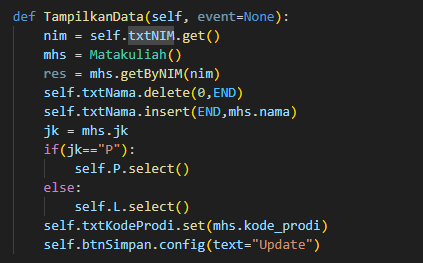
1. OnCari

Ini biasanya digunakan untuk mencari data mahasiswa berdasarkan NIM yang dimasukkan oleh pengguna.



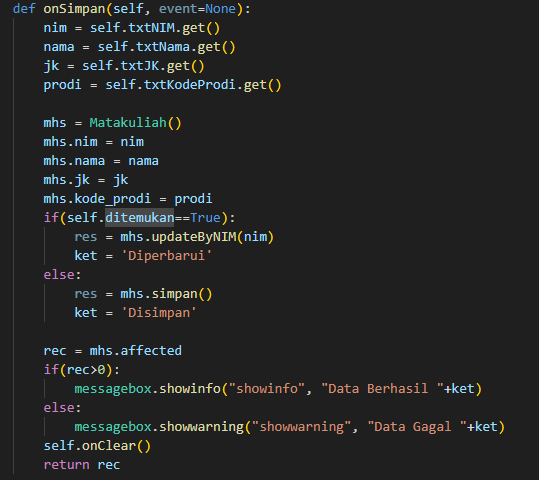
1. Metode `TampilkanData`

Metode ini digunakan untuk menampilkan data mahasiswa yang telah ditemukan berdasarkan NIM yang telah dimasukkan.



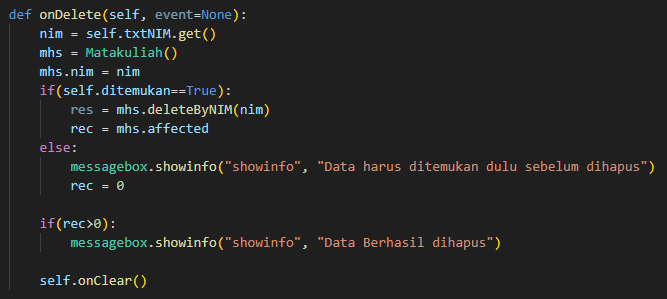
1. OnSimpan

Ini untuk menyimpan atau memperbarui data mahasiswa ke database.



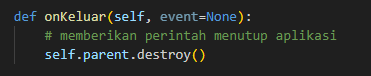
1. OnDelete

Sedangkan OnDelete gunanya untuk menghapus data mahasiswa dari database.



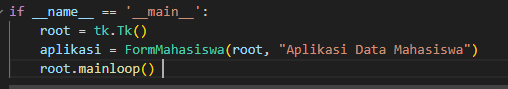
1. OnKeluar

OnKeluar untuk kode aplikasi saat tombol close di jendela aplikasi ditekan.

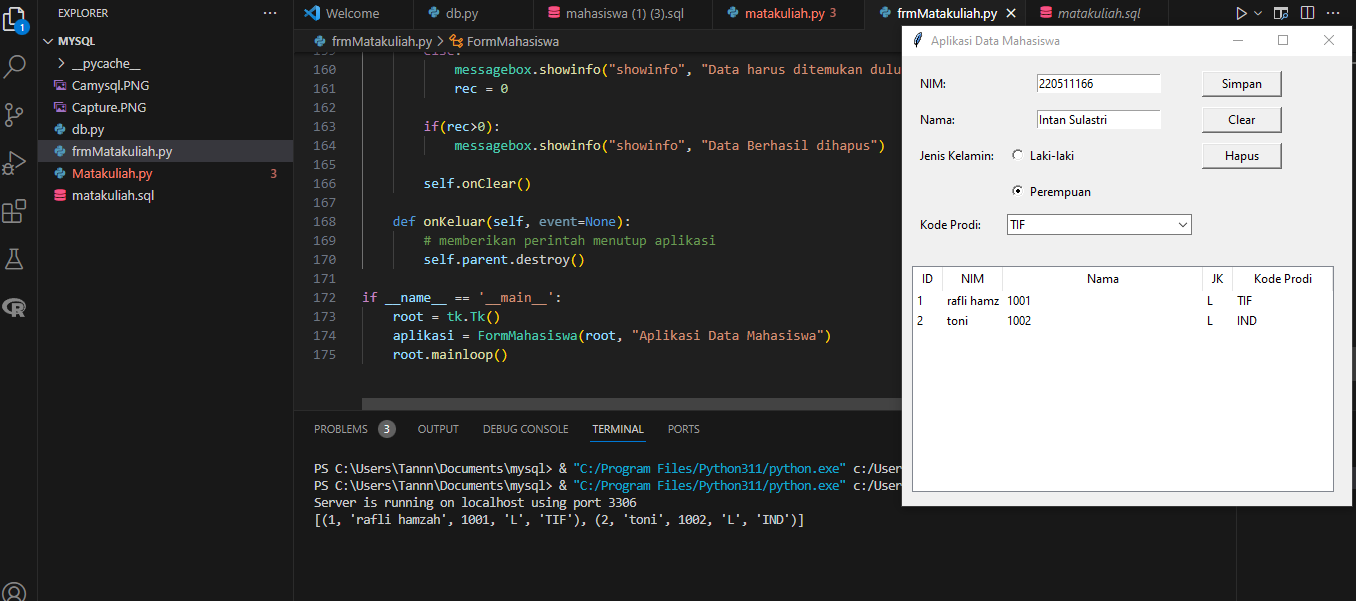


1. Main Program

Program utama yang membuat objek `Tk` sebagai root window dan objek `FormMahasiswa` sebagai aplikasi GUI utama. Kemudian, menjalankan main loop untuk menampilkan GUI dan menanggapi input pengguna.



Hasil output



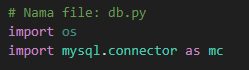
Pertemuan 12

(Aplikasi Python Database berbasis MySQL)

Database

1. Import Library

Kode mengimpor modul `os` dan modul `mysql.connector` dengan alias `mc`.



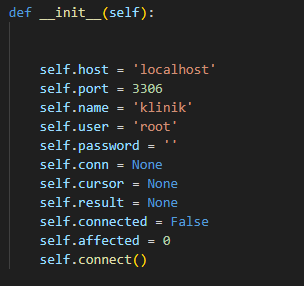
1. Kelas `DBConnection’

Kelas ini memiliki atribut untuk informasi koneksi ke database MySQL (`host`, `port`, `name`, `user`, `password`), serta atribut lain seperti `conn`, `cursor`, `result`, `connected`, dan `affected`.



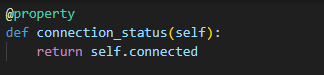
1. `\_\_init\_\_`

Konstruktor untuk inisialisasi atribut dan membuka koneksi ke database.



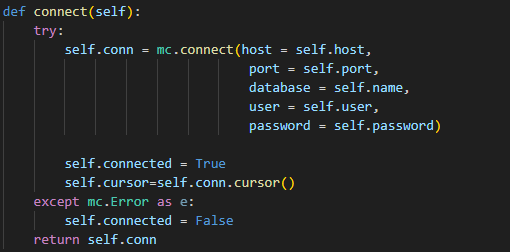
1. Property `connection\_status`

Properti untuk mendapatkan status koneksi.



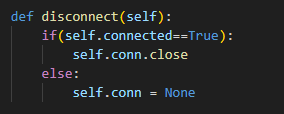
1. `connect`

Untuk membuka koneksi ke database.



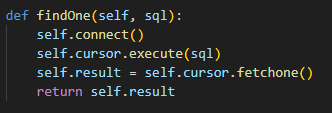
1. `disconnect`

Untuk menutup koneksi ke database.



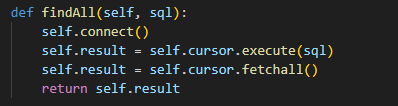
1. `findOne`

Mengeksekusi query SQL dan mengembalikan satu baris hasil.



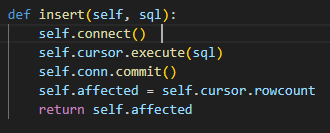
1. `findAll`

Mengeksekusi query SQL dan mengembalikan semua baris hasil.



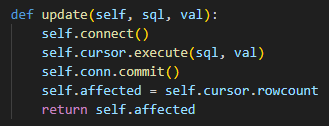
1. `insert`

Mengeksekusi perintah SQL untuk memasukkan data ke dalam database.



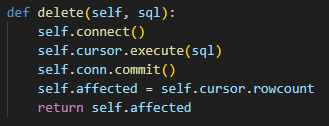
1. `update`

Mengeksekusi perintah SQL untuk memperbarui data di dalam database.



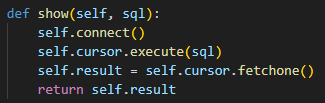
1. `delete`

Mengeksekusi perintah SQL untuk menghapus data dari database.



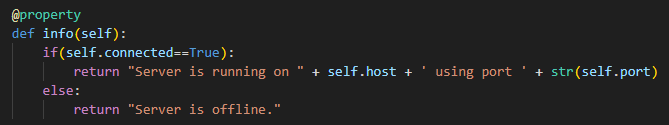
1. `show`

Mengeksekusi query SQL dan mengembalikan satu baris hasil.



1. Property `info`

Mengembalikan informasi tentang status server database.



1. Pemanggilan Metode pada Objek A

Membuat objek menggunakan kelas `DBConnection` dan memanggil properti `info` untuk mendapatkan status server database, yang kemudian dicetak.



PERAWAT.PY

1. Import Library dan Modul

Kode mengimpor kelas `DBConnection` dari modul `db`, yang digunakan untuk melakukan koneksi dan interaksi dengan database MySQL.



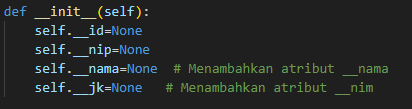
1. Kelas `perawat`

Ini adalah kelas utama yang berisi metode-metode untuk menyimpan, memperbarui, menghapus, dan mengambil data perawat dari database.



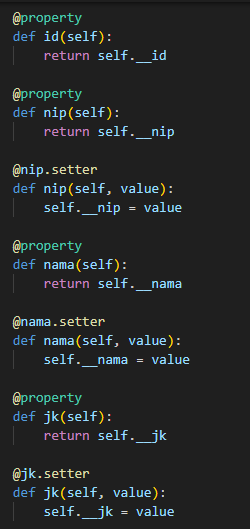
1. Inisialisasi Atribut

metode `\_\_init\_\_`, beberapa atribut yang berkaitan dengan data perawat seperti `\_\_id`, `\_\_nip`, `\_\_nama`, dan `\_\_jk` diinisialisasi. Prefix double underscore (`\_\_`) menandakan bahwa atribut tersebut bersifat privat, artinya tidak bisa diakses secara langsung dari luar kelas.



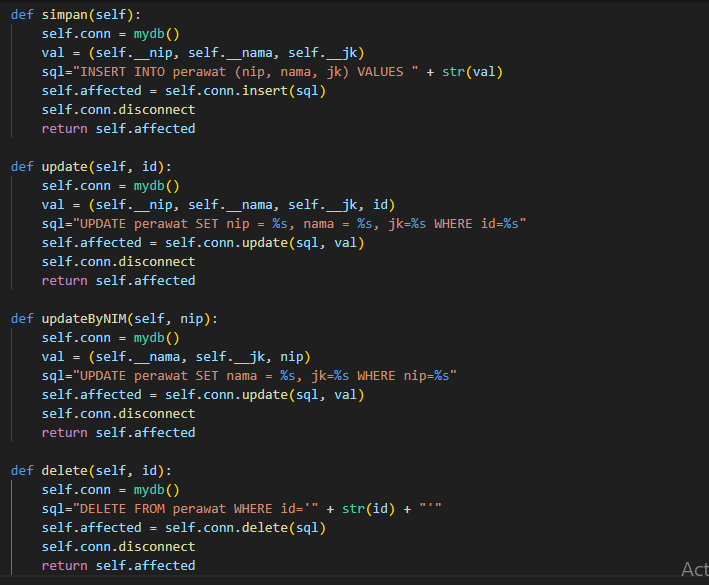
1. Property dan Setter

Untuk setiap atribut, ada property dan setter yang digunakan untuk mengakses dan mengubah nilai atribut tersebut dengan metode yang terkendali.



1. Metode CRUD

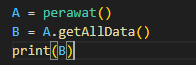
Kelas `perawat` memiliki metode-metode seperti `simpan`, `update`, `delete`, `getByID`, `getByNIM`, dan `getAllData` yang digunakan untuk melakukan operasi CRUD pada data perawat di database.





1. Pemanggilan Metode pada Objek A

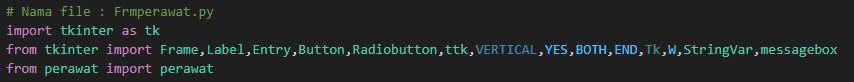
Setelah kelas `perawat` didefinisikan, sebuah objek `A` dibuat menggunakan kelas tersebut. Kemudian, metode `getAllData` dipanggil pada objek tersebut untuk mendapatkan semua data perawat dari database, yang kemudian dicetak untuk ditampilkan.



FRM PERAWAT

1. Import Libraries

Kode mengimpor modul `tkinter` sebagai `tk` dan beberapa kelas dan fungsi lain yang dibutuhkan dari modul tersebut. Selain itu, juga diimpor modul `perawat`, yang mungkin berisi definisi kelas atau fungsi terkait dengan data perawat.



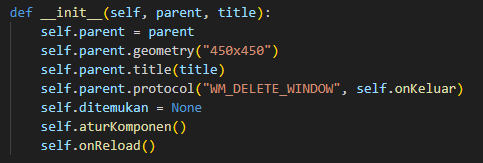
1. Kelas `Formperawat`

Kelas ini bertanggung jawab untuk membuat dan mengelola GUI untuk aplikasi pengelolaan data perawat.



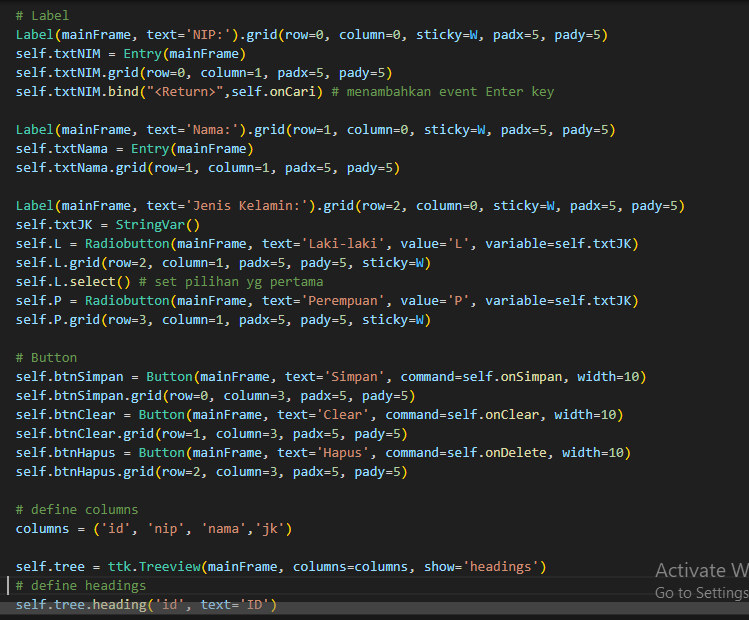
1. `\_\_init\_\_`

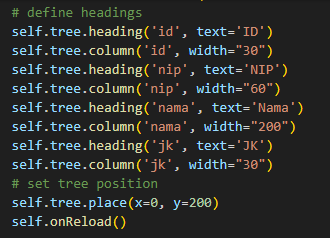
Metode konstruktor yang dipanggil ketika objek kelas dibuat. Metode ini mengatur konfigurasi utama jendela aplikasi dan memanggil metode `aturKomponen()` dan `onReload()`.



1. `aturKomponen`

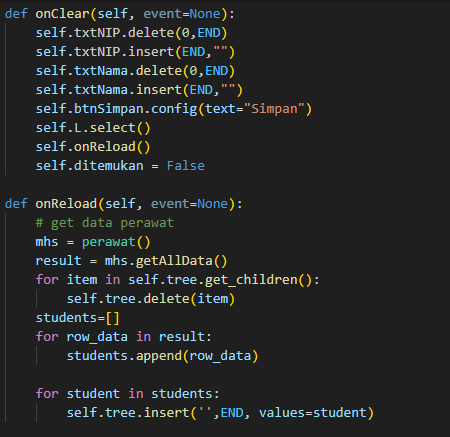
Metode ini mengatur komponen-komponen GUI seperti label, entry fields, radio buttons, tombol, dan treeview (untuk menampilkan data dalam tabel).



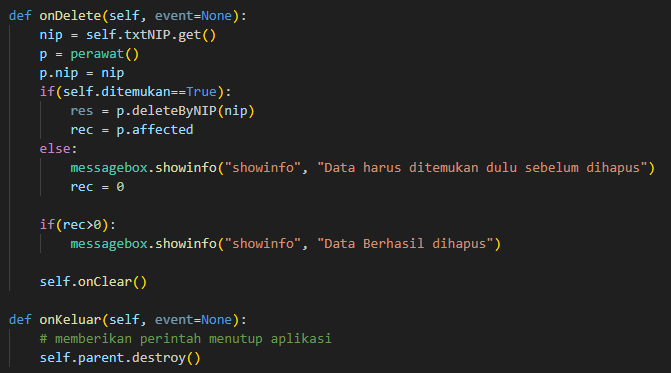


1. Metode-metode Lainnya

Terdapat beberapa metode lainnya seperti `onClear`, `onReload`, `onCari`, `TampilkanData`, `onSimpan`, `onDelete`, dan `onKeluar` yang berfungsi untuk melakukan operasi tertentu seperti membersihkan field input, memuat ulang data, mencari data, menampilkan data, menyimpan data, menghapus data, dan menutup aplikasi.

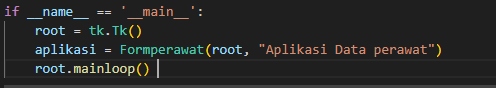






1. Main Program

Program utama yang membuat objek `Tk` sebagai root window dan objek `Formperawat` sebagai aplikasi GUI utama. Kemudian, menjalankan main loop untuk menampilkan GUI dan menanggapi input pengguna.



Pertemuan 13

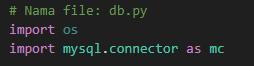
(Database menggunakan Tkinter dan MySQL)

Db.

Implementasi kelas `DBConnection` dalam Python. Kelas ini bertanggung jawab untuk mengelola koneksi ke database MySQL dan menjalankan operasi SQL seperti `findOne`, `findAll`, `insert`, `update`, dan `delete`.

1. Import Library

Kode mengimpor modul `os` dan modul `mysql.connector` dengan alias `mc`.



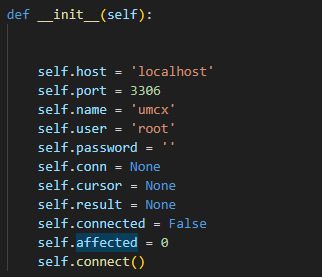
1. DBConnection

Kelas ini memiliki beberapa atribut seperti `host`, `port`, `name`, `user`, dan `password` yang digunakan untuk mengatur koneksi ke database MySQL.



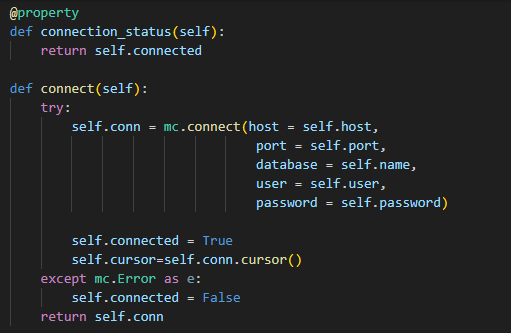
1. `\_\_init\_\_’

adalah metode konstruktor yang dipanggil saat objek kelas dibuat. ini menginisialisasi atribut-atribut kelas dan memanggil metode `connect()` untuk membuka koneksi ke database.



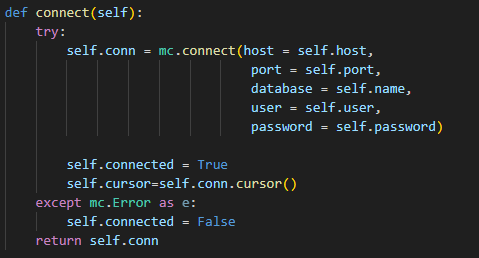
1. Property `connection\_status’

Properti ini mengembalikan status koneksi.



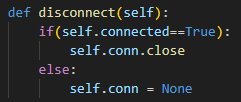
1. `connect`

Digunakan untuk membuka koneksi ke database MySQL. Jika koneksi berhasil, atribut `connected` diatur menjadi `True`, dan objek cursor akan dibuat.



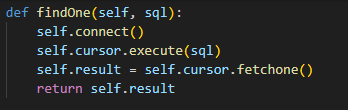
1. `disconnect`

Gunanya untuk menutup koneksi ke database jika koneksi telah terbuka sebelumnya.



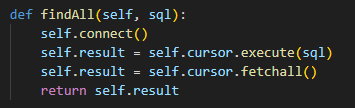
1. `findOne’

Sama seperti sebelumnya Metode ini mengeksekusi SQL dan mengembalikan satu baris hasil.



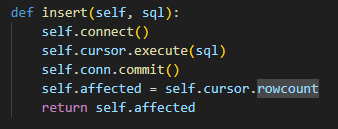
1. `findAll`

Untuk mengeksekusi query SQL dan mengembalikan semua baris hasil.



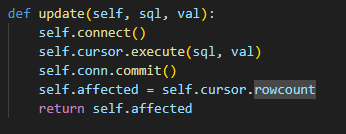
1. `insert`

Akan mengeksekusi perintah SQL untuk memasukkan data ke dalam database.



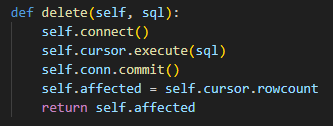
1. `update`

Menjalankan perintah SQL untuk memperbarui data.



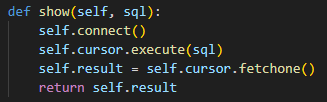
1. `delete`

Menjalankan perintah SQL untuk menghapus data.



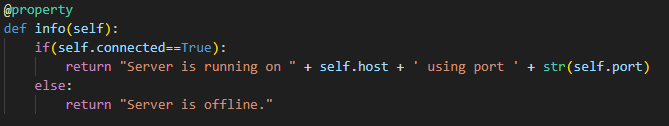
1. `show`

Metode ini mengeksekusi query SQL dan mengembalikan satu baris hasil.



1. Property `info`

Properti ini mengembalikan informasi tentang status server database.



1. yang terakhir Pemanggilan Metode pada Objek A

Setelah kelas `DBConnection` didefinisikan, sebuah objek `A` dibuat menggunakan kelas tersebut. Kemudian, properti `info` dipanggil pada objek tersebut untuk mendapatkan informasi tentang status server database, yang kemudian dicetak untuk ditampilkan.



Dashboard

1. Import Library

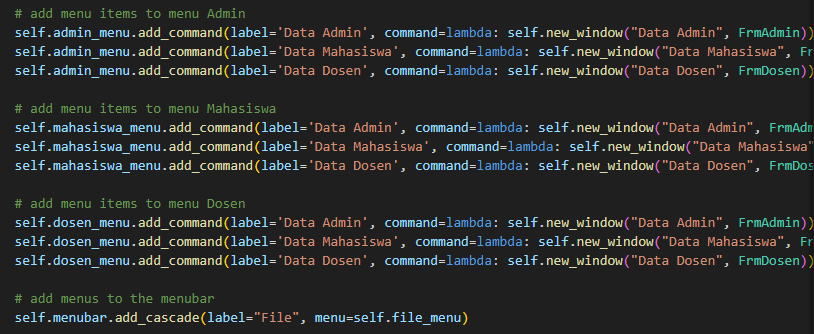
Modul `tkinter` digunakan untuk pembuatan GUI.



1. Class `Dashboard`

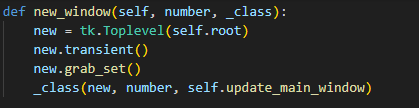
Kelas ini terdapat metode-metode untuk membuat jendela utama, menambahkan menu, membuka jendela baru, memperbarui jendela utama, dan logout.





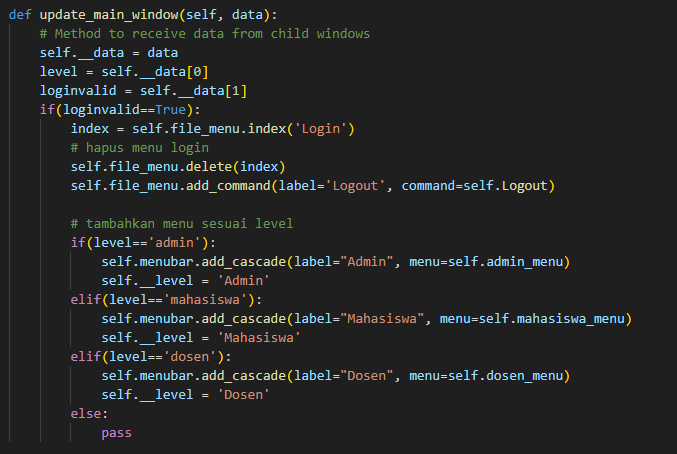
1. `new\_window`

Metode ini digunakan untuk membuat jendela baru ketika pengguna memilih suatu menu. Jendela baru dibuat sebagai child window dari jendela utama.



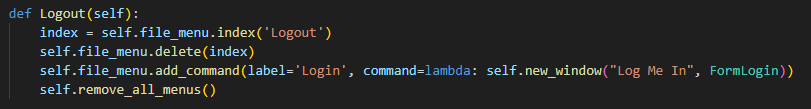
1. `update\_main\_window`

Disini akan menerima data dari jendela anak (FormLogin) setelah login berhasil. Data yang diterima mencakup level akses pengguna dan status keberhasilan login. Berdasarkan akses.



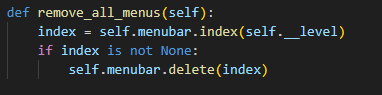
1. Metode `Logout`

ini digunakan untuk logout pengguna. Ketika logout, menu logout akan dihapus dan menu login akan ditambahkan kembali.



1. `remove\_all\_menus`

Gunanya untuk menghapus semua menu yang ditambahkan berdasarkan level akses pengguna setelah logout.



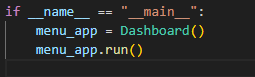
1. `run`

Ini untuk menjalankan aplikasi GUI sehingga jendela utama akan ditampilkan dan interaksi dengan pengguna dapat dilakukan.



1. Pemanggilan Objek Dashboard

Pada blok terakhir, objek `Dashboard` dibuat dan metode `run` dipanggil untuk menjalankan aplikasi.



FRM ADMIN

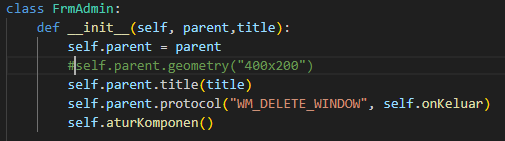
1. Import Library

Disini kita akan mengimpor kelas-kelas yang diperlukan dari modul Tkinter seperti `Frame`, `Label`, `Entry`, `Button`, dan `Tk`, serta mengimpor `font` untuk mengatur ukuran font teks.



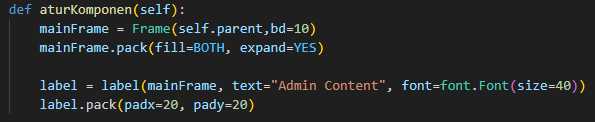
1. Class `FrmAdmin`

Kelas ini digunakan untuk membuat jendela untuk konten admin. Metode `\_\_init\_\_` digunakan untuk menginisialisasi jendela dengan judul yang diberikan dan mengatur komponen-komponen GUI.



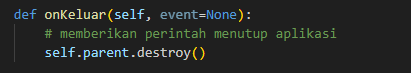
1. `aturKomponen`

Gunanya untuk mengatur komponen-komponen GUI di dalam jendela. Lalu sebuah frame utama dibuat dan diisi dengan sebuah label yang menampilkan teks "Admin Content" dengan font berukuran 40.



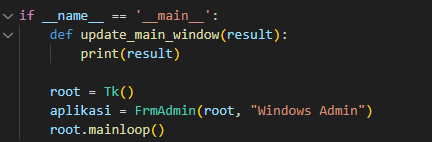
1. `onKeluar`

ini dipanggil saat jendela ditutup. Perintah `self.parent.destroy()` digunakan untuk menutup aplikasi ketika tombol close di jendela diklik.



1. Objek FrmAdmin

Di blok terakhir, objek `FrmAdmin` dibuat dan metode `run` dipanggil untuk menjalankan aplikasi.



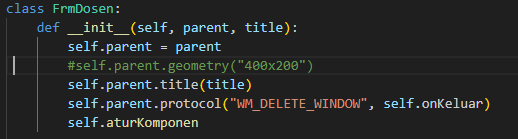
FRM DOSEN

1. mengimpor komponen yang diperlukan dari pustaka tkinter.



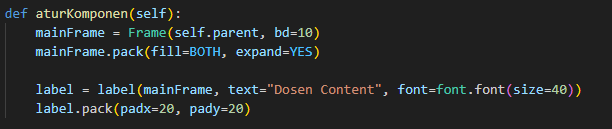
1. Mendefinisikan kelas

`FrmDosen` dengan metode inisialisator `\_\_init\_\_` yang mengatur judul jendela, protokol untuk menutup jendela, dan memanggil metode `aturKomponen` untuk mengatur komponen antarmuka pengguna (UI).



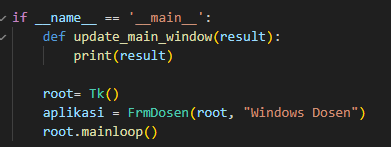
1. `aturKomponen`

membuat sebuah frame utama di dalam jendela, menambahkan sebuah label dengan teks "Konten Dosen" menggunakan ukuran font yang ditentukan, dan memasukkannya ke dalam frame utama.



1. `onKeluar`

ketika jendela ditutup, dan itu menghancurkan jendela induk, efektif menutup aplikasi.



1. `\_\_main\_\_`,

itu membuat sebuah instansi `Tk()`, menginisialisasi `FrmDosen`, dan memulai loop Tkinter dengan `root.mainloop()` untuk melihatkan jendela.

FRM MAHASISWA

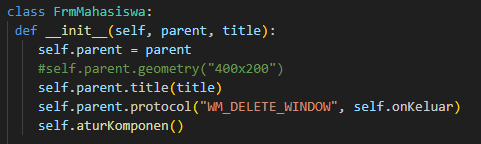
1. mengimpor

impor komponen yang diperlukan dari pustaka tkinter.



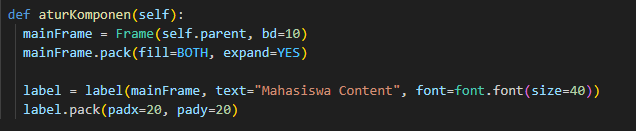
1. `FrmMahasiswa`

memiliki metode inisialisasi `\_\_init\_\_` yang mengatur judul jendela, protokol untuk menutup jendela, dan memanggil metode `aturKomponen` untuk mengatur komponen antarmuka pengguna (UI).



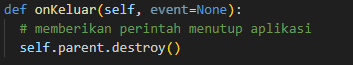
1. `aturKomponen`

membuat sebuah frame utama di dalam jendela, menambahkan sebuah label dengan teks "Konten Mahasiswa" menggunakan ukuran font yang ditentukan, dan memasukkannya ke dalam frame utama.



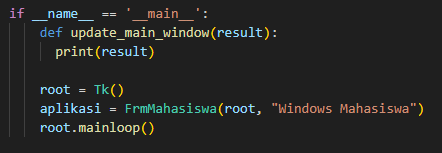
1. `onKeluar`

ketika jendela ditutup, dan itu menghancurkan jendela induk, efektif menutup aplikasi.



1. `\_\_main\_\_`,

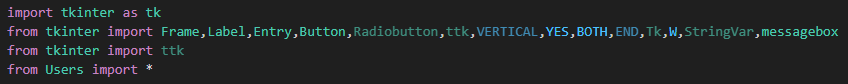
membuat sebuah instansi `Tk()`, menginisialisasi `FrmMahasiswa`, dan memulai loop Tkinter dengan `root.mainloop()` untuk menampilkan jendela.



FRM LOGIN

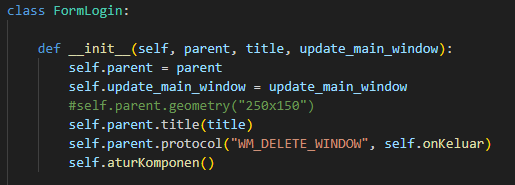
1. impor modul `tkinter`

beberapa komponen yang diperlukan seperti `Frame`, `Label`, `Entry`, `Button`, `Radiobutton`, `ttk`, `StringVar`, dan `messagebox`, serta modul `Users` yang merupakan bagian dari aplikasi untuk melakukan validasi login.



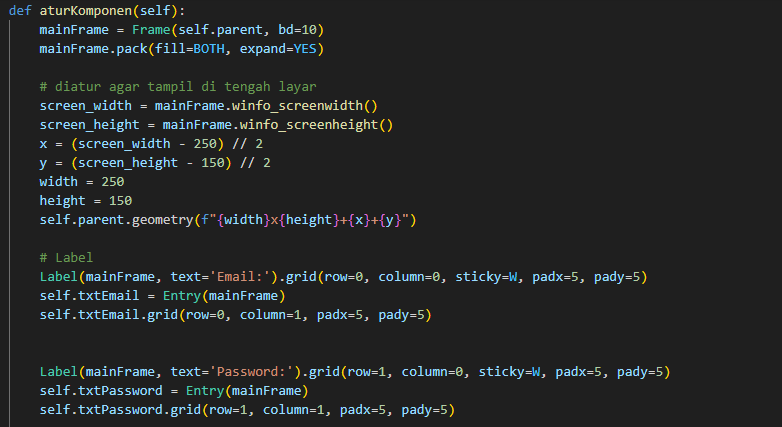
1. ‘FormLogin`

memiliki metode `\_\_init\_\_` yang digunakan untuk menginisialisasi jendela aplikasi dengan mengatur judul, ukuran, dan komponen GUI.



1. `aturKomponen`

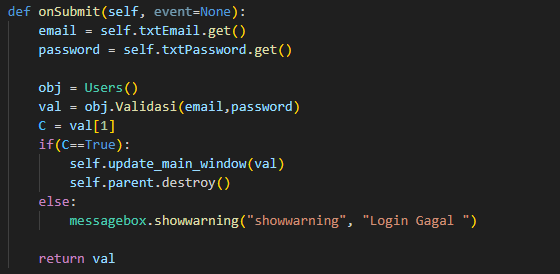
digunakan untuk menempatkan komponen-komponen GUI seperti label, entri (input), dan tombol pada frame utama.





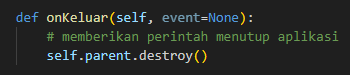
1. `onSubmit`

ketika tombol "Submit" ditekan. Ini mengambil email dan kata sandi dari entri dan kemudian memvalidasi login dengan menggunakan metode `Validasi` dari objek `Users`. Jika login berhasil, metode `update\_main\_window` dipanggil untuk memperbarui jendela utama dengan data hasil login, dan jendela aplikasi akan ditutup. Jika login gagal, akan muncul pesan peringatan.



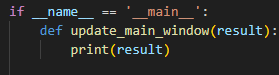
1. `onKeluar`

dipanggil ketika tombol "Cancel" ditekan atau ketika jendela ditutup, dan itu menghancurkan jendela aplikasi.



1. `\_\_main\_\_`,

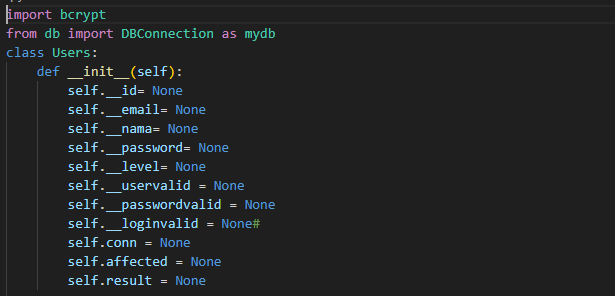
itu membuat sebuah instance `Tk()`, menginisialisasi `FormLogin`, dan memulai loop Tkinter dengan `root.mainloop()` untuk menampilkan jendela aplikasi.



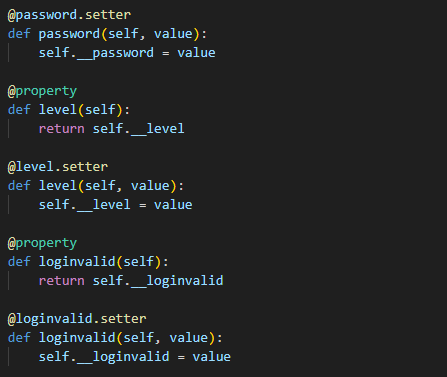
FRM USER

1. Kelas `Users`

memiliki beberapa atribut seperti `\_\_id`, `\_\_email`, `\_\_nama`, `\_\_password`, `\_\_level`, `\_\_uservalid`, `\_\_passwordvalid`, dan `\_\_loginvalid` yang digunakan untuk menyimpan informasi tentang pengguna dan hasil validasi.

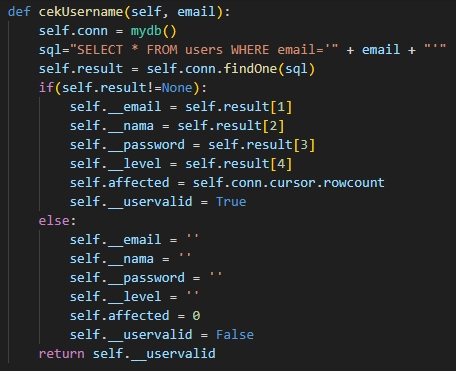






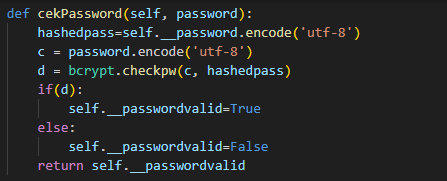
1. `cekUsername`

digunakan untuk memeriksa apakah email pengguna ada dalam database. Jika email ditemukan, atribut `\_\_uservalid` diatur menjadi `True`, dan informasi pengguna diambil dari database. Jika tidak, `\_\_uservalid` diatur menjadi `False`.



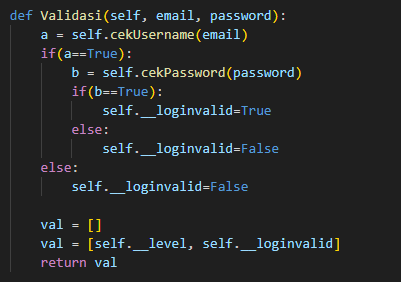
1. `cekPassword`

digunakan untuk memeriksa apakah kata sandi yang dimasukkan cocok dengan kata sandi yang disimpan dalam database. Ini menggunakan modul `bcrypt` untuk memeriksa kecocokan kata sandi yang dihash.



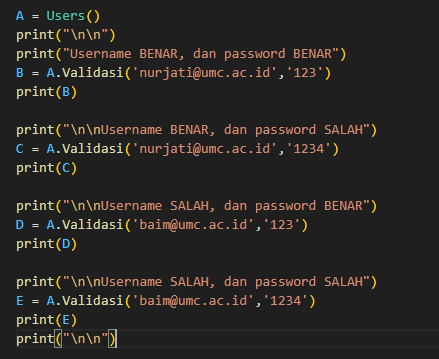
1. `Validasi`

digunakan untuk melakukan validasi lengkap dengan memanggil `cekUsername` dan `cekPassword`. Hasil validasi akhir, yaitu `\_\_loginvalid`, disertakan dalam daftar nilai yang dikembalikan bersama dengan tingkat akses pengguna



1. `\_\_main\_\_`,

beberapa skenario validasi login diprint untuk pengujian. Hasilnya menunjukkan apakah validasi dilakukan dengan benar atau salah.



Pertemuan 14

(Membuat Modul Praktikum PBO)

Pada pertemuan kali ini tugasnya adalah membuat modul praktikum selama Satu Semester tujuan dibuatnya modul ini menguji sampai mana pemahaman mahasiswa/i dalam memahami materi selama ini

Tugas ini bentuknya kelompok jadi kita diberi waktu selama 1 minggu untuk mengerjakan dan akan dikumpulkan paling lambat pada Hari minggu tanggal 28 Januari 2024.

Modul ini isinya ada banyak mulai dari pengenalan materi secara singkat, proses pembuatan aplikasi secara step by step, penjelasan dan praktek.

Dalam modul Ini Total Ada 14 pertemuan termasuk (ini) yang dikerjakan.

Saya Intan Sulastri Dari Kelas TI22L selaku penyusun pada modul ini mengucapkan terima kasih, karena berkat tugas ini saya jadi meningat materi yang sudah dijelaskan dan jadi memiliki kesempatan untuk memperbaiki Coding dari beberapa program saya yang gagal.

Semoga dengan adanya Modul Praktikum ini bisa memenuhi tugas saya

Terima kasih saya ucapkan pada Bapak Freddy Wicaksono Selaku dosen pada Mata Kuliah Pemrograman II (PBO) atas perhatiannya.

Dengan ini saya nyatakan Bahwa Modul ini telah selesai.