PYTHON



MODUL

PRAKTIKUM

Pemrograman Berorientasi Objek

2023/2024

Disusun Oleh:

1. Dicky Franssetiaji (231511010)

2. Rulastri (220511071)

3. Baruna Hadi Maulana (220511190)

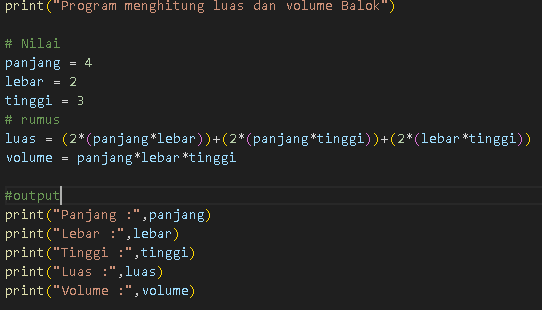
**MODUL 1**

**PENERAPAN PYTHON PADA PROGRAM PENGHITUNGAN LUAS DAN VOLUME BANGUN RUANG**

Python adalah bahasa pemrograman interpretatif multiguna. Tidak seperti bahasa lain yang  
susah untuk dibaca dan dipahami, python lebih menekankan pada keterbacaan kode agar lebih  
mudah untuk memahami sintaks. Hal ini membuat Python sangat mudah dipelajari baik untuk  
pemula maupun untuk yang sudah menguasai bahasa pemrograman lain.

Pada modul 1 ini kita akan mencoba mengimplementasikan *python* pada program menghitung luas dan volume bangun ruang. Kita akan ambil contoh bangun ruang balok untuk dipakai pada program. Ada beberapa hal yang perlu diperhatikan, diantaranya adalah kita harus mengetahui rumus-rumus penghitungan luas dan volume bangun ruang. Untuk bangun ruang balok rumus penghitungan luas adalah “(2 X (Panjang X Lebar)) + (2 X (Panjang X Tinggi)) + (2 X (Lebar X Tinggi))”. Sedangkan rumus penghitungan volumenya adalah “Panjang X Lebar X Tinggi”.

Berikut terlampir contoh source code dari program tersebut :



Pada *code* diatas kita langsung memberikan nilai pada variabel-variabel yang kita buat tanpa inputan *user* sehingga program kita bisa lebih sederhana. Untuk lebih memudahkan kita memahami *code* diatas, maka berikut penjelasan langkah-langkahnya:

1. Perintah untuk menampilkan nama program

1

Perintah ini merupakan perintah untuk menampilkan nama program yang akan kita buat.sebagai contoh kita beri nama program tersebut “Program menghitung luas dan volume Balok”. Perintah ini terdapat pada awal.

1. Mendeklarasikan variabel dan pemberian nilai pada variabel



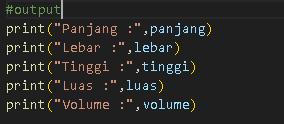
Variabel yang kita gunakan pada program ini akan dipakai pada rumus penghitungan. Buatlah variabel sesuai kebutuhan program. Pada contoh diatas, variabelnya adalah “panjang”, “lebar”, dan “tinggi”. Kita bisa langsung memberikan nilai pada variabel ini. Sebagai contoh nilai untuk panjang adalah 4, untuk lebar adalah 2, dan untuk tinggi adalah 3.

1. Membuat rumus luas dan volume balok

1

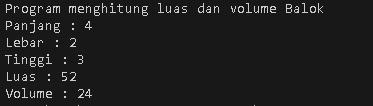
Fungsi rumus tersebut adalah untuk mengolah variabel yang sudah kita beri nilai sebelumnya. Pada contoh diatas variabel luas akan berisi hasil penghitungan dari rumus luas balok. Sedangkan untuk variabel volume berisi hasil penghitungan dari rumus volume balok.

1. Membuat perintah *output*



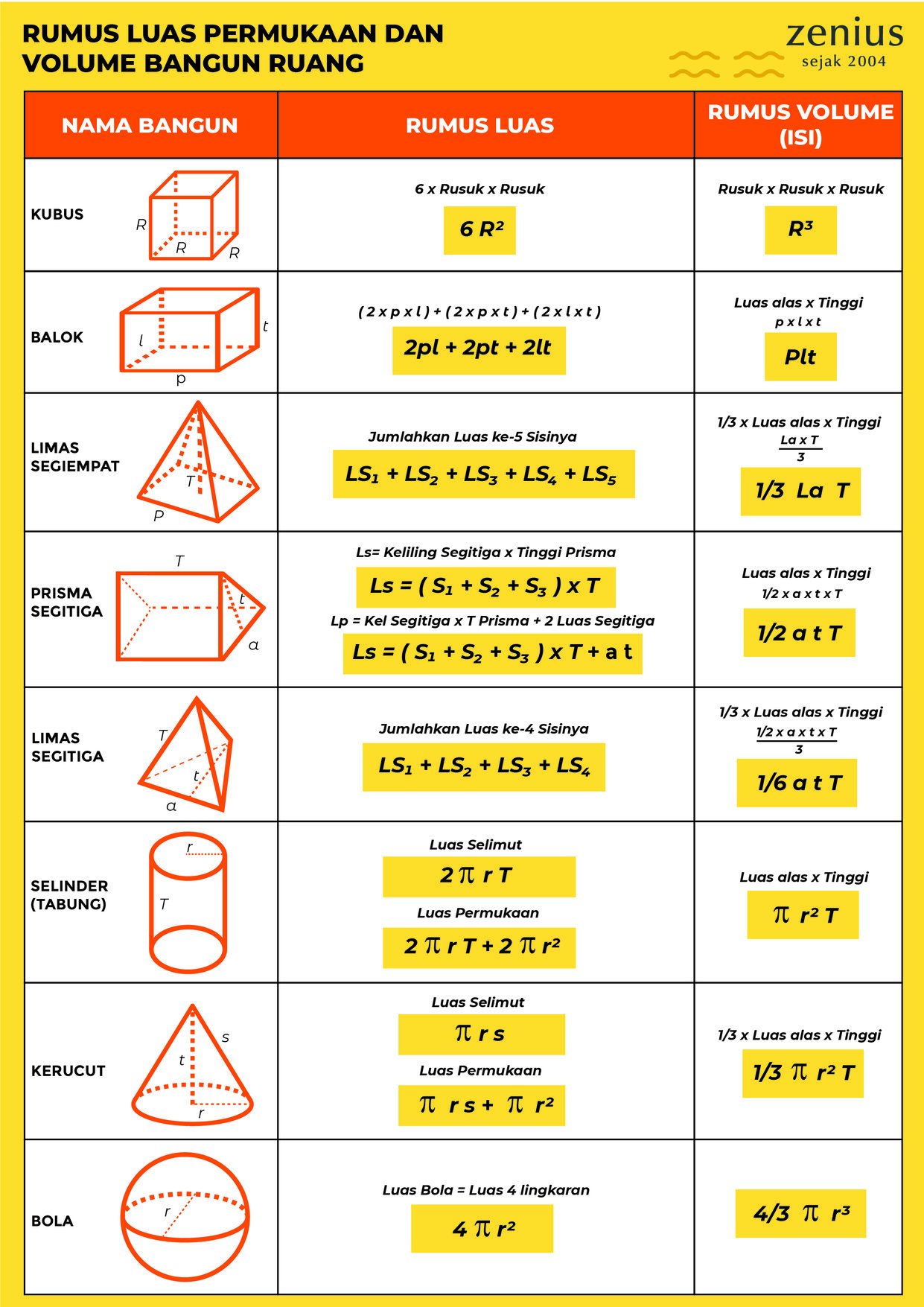
Perintah diatas merupakan perintah untuk memberika *output* yang akan kita tampilkan Disini kita akan menampilkan Panjang yang diikuti nilai dari variabel panjang, Lebar yang diikuti nilai dari variabel lebar, Tinggi yang diikuti nilai dari variabel tinggi, Luas yang diikuti nilai dari variabel luas yang didapatkan dari hasil penghitungan rumus luas, dan Volume yang diikuti nilai dari variabel volume yang didapatkan dari hasil penghitungan rumus volume.

1. *Outuput* program



*Output* tersebut adalah hasil dari pemrosessan perintah-perintah yang kita berikan.

Sebagai catatan nilai yang diberikan tidak terpaku seperti contoh saja. Kita bebas memberikan nilai pada variabel-variabel yang sudah kita buat. Dengan penjelasan diatas juga kita bisa membuat program menghitung luas dan volume pada bangun ruang yang lainnya sebagai latihan. Berikut adalah rumus-rumus Luas dan Volume Bangun Ruang yang bisa kita implementasikan dengan cara memodifikasi dengan memakai *code* diatas sebagai dasarnya :

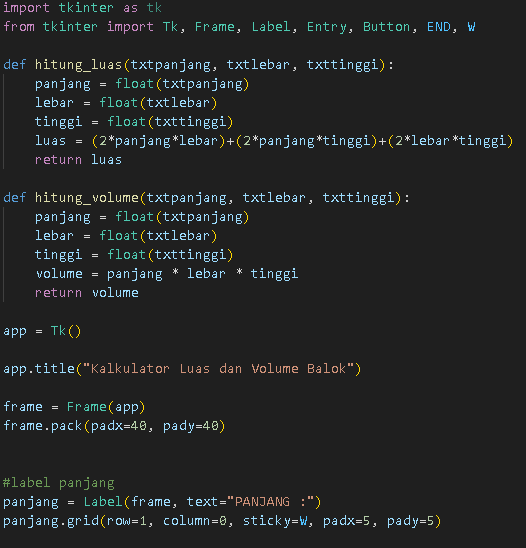


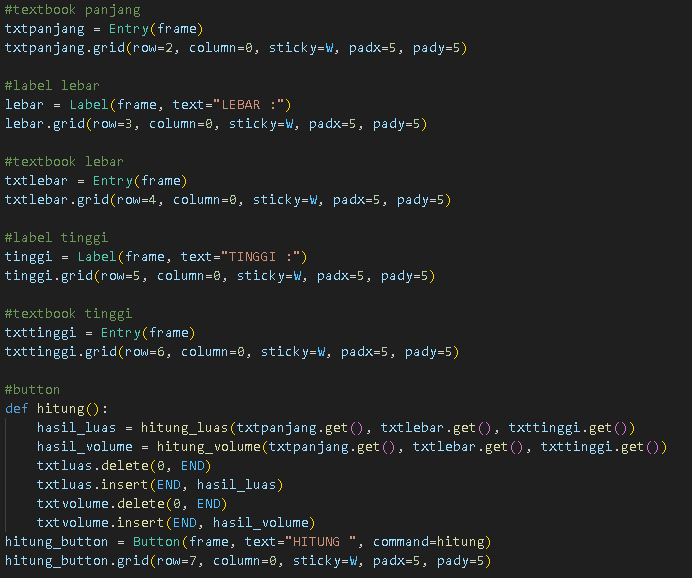
**MODUL 2**

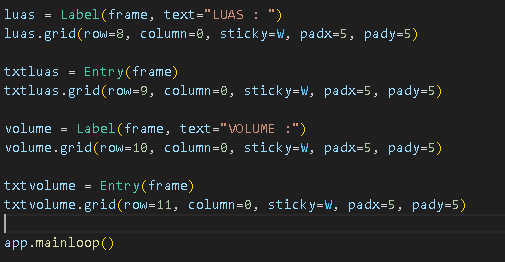
**PENERAPAN PYTHON PADA PROGRAM PENGHITUNGAN LUAS DAN VOLUME BANGUN RUANG BERBASIS WINDOWS**

Modul 2 ini adalah untuk membuat program pada modul 1 berbasis windows. Pada program berbasis windows ini kita akan membuat kolom *input* untuk *user.* Sehingga kita bisa mengubah nilai dari setiap variabelnya dengan leluasa pada program. Pada modul ini kita akan memakai modul *tkinter* yang akan membantu kita membuat program berbasis *windows.*

Pada contoh kali ini kita masih akan menggunakan bangun ruang balok. Berikut terlampir *code* untuk programnya :







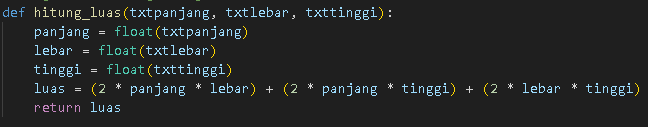
Pada *code* diatas *user* bisa memasukan nilai ke variabel yang ada. *User* juga bisa mengubah sendiri dengan leluasa nilai-nilai tersebut. Berikut adalah penjelasan langkah-langkah pada *code* diatas :

1. Melakukan *import* modul tkinter

2

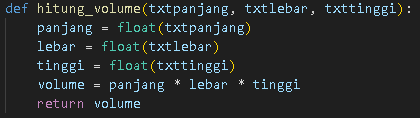
Modul tkinter adalah modul yang akan kita pakai untuk membuat program berbasis *windows.* Pada langkah ini kita akan meng-*import* modul tkinter yang akan dipakai pada *code* diatas.

1. Membuat fungsi hitung\_luas



Melakukan definisi fungsi kita lakukan agar *code* yang kita kerjakan bisa lebih terstruktur. Pada fungsi ini variabel panjang nilainya diambil dari *input* panjang, variabel lebar nilainya diambil dari *input* lebar, dan variabel tinggi nilainya didapat dari *input* tinggi. Disini kita juga membuat variabel luas yang isinya rumus luas balok dengan nilai yang didapatkan dengan mengolah variabel panjang, lebar, tinggi.

1. Membuat fungsi hitung\_volume



Fungsi ini adalah fungsi untuk menghitung volume balok. Sama seperti langkah sebelumnya variabel panjang nilainya diambil dari *input* panjang, variabel lebar nilainya diambil dari *input* lebar, dan variabel tinggi nilainya didapat dari *input* tinggi. Disini kita juga membuat variabel volume yang isinya rumus volume balok dengan nilai yang didapatkan dengan mengolah variabel panjang, lebar, tinggi.

1. Membuat objek tkinter

2

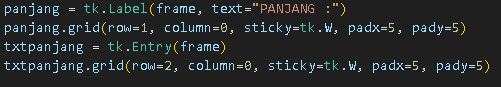
1. Memberi judul pada jendela aplikasi

2

1. Memberi frame pada jendela aplikasi

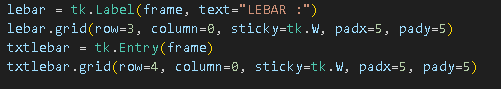
2

1. Membuat label dan entry untuk panjang



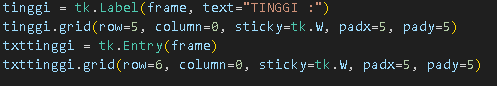
Fungsi dari label adalah untuk memberikan informasi pada *user* untuk memasukan panjang balok pada bagian entry. Fungsi dari entry adalah untuk tempat user memasukan input panjang balok.

1. Membuat label dan entry untuk lebar



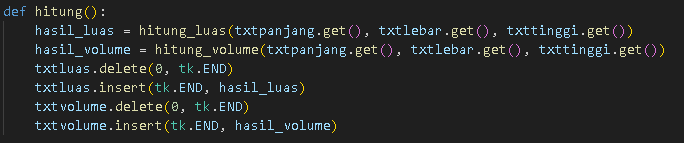
Fungsi dari label adalah untuk memberikan informasi pada *user* untuk memasukan lebar balok pada bagian entry. Fungsi dari entry adalah untuk tempat user memasukan input lebar balok

1. Membuat label dan entry untuk tinggi



Fungsi dari label adalah untuk memberikan informasi pada *user* untuk memasukan tinggi balok pada bagian entry. Fungsi dari entry adalah untuk tempat user memasukan input tinggi balok

1. Membuat fungsi hitung



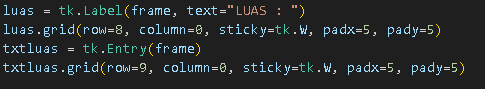
Fungsi hitung ini digunakan untuk mengambil nilai pada variabel-variabel yang nilainya sudah diinput oleh user. Pada fungsi hitung ini berguna untuk menampung semua nilai panjang, lebar, dan tinggi. Pada fungsi ini juga terdapat fungsi hitung\_luas dan hitung\_volume dan juga memberikan nilai dari hasil pengolahan nilai pada fungsi hitung\_luas dan hitung\_volume.

1. Membuat tombol hitung

2

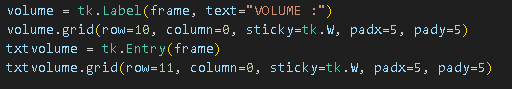
Tombol hitung akan melaksanakan fungsi hitung.

1. Membuat label dan Entry untuk menampilkan hasil luas



Fungsi dari label adalah untuk memberikan informasi pada *user* bahwa hasil penghitungan luas balok terdapat pada bagian entry. Fungsi dari entry adalah untuk tempat menampilkan hasil penghitungan luas balok.

1. Membuat label dan entry untuk menampilkan hasil volume

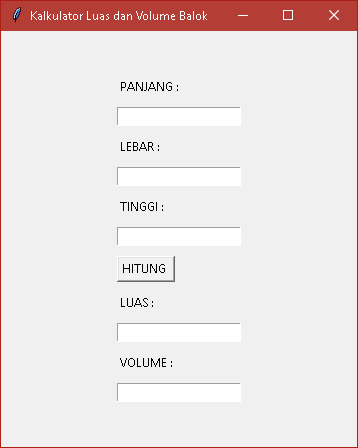


Fungsi dari label adalah untuk memberikan informasi pada *user* bahwa hasil penghitungan volume balok terdapat pada bagian entry. Fungsi dari entry adalah untuk tempat menampilkan hasil penghitungan volume balok.

1. Menjalankan aplikasi

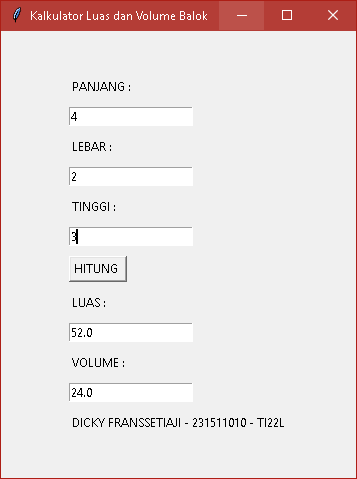
2

1. *Output* program



Berikut tampilan program ketika pertama kali dibuka oleh user.

1. Output program setelah user melakukan input



Berikut tampilan program ketika user sudah melakukan input panjang, lebar, dan tinggi, kemudian memencet tombol hitung.

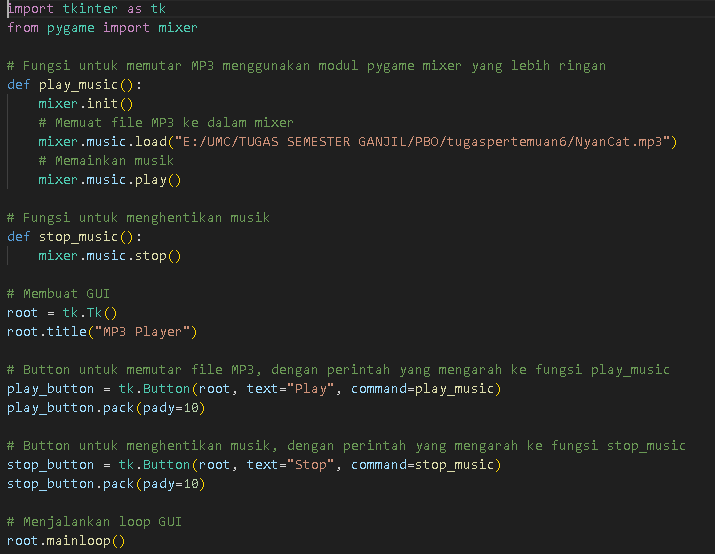
Dengan menggunakan program berbasis windows, user bisa dengan mudah memahami maksud dari program tersebut. Sehingga user dengan leluasa bisa mengubah-ubah nilai variabelnya sesuai kebutuhan. Seperti modul sebelumnya, kini kita bisa memodifikasi code diatas untuk bangun ruang lainnya. Jadikan code diatas sebagai dasar membuat aplikasi penghitungan bangun ruang yang lainnya.

**MODUL 3**

**MEMBUAT APLIKASI UNTUK MEMAINKAN FILE MP3 DAN MP4**

Kita pasti sudah sangat familiar dengan file mp3 dan mp4. File mp3 adalah file yang berisi lagu atau suara atau audio. Sedangkan file mp4 adalah file yang berisi video. Pada modul kali ini kita akan membuat sebuah program berbasis windows untuk memainkan kedua file tersebut. Ini sangat bermanfaat untuk kita agar bisa memainkan lagu atau video yang kita inginkan dengan program yang kita buat sendiri.

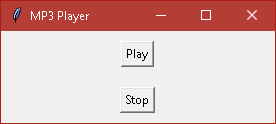
Pada contoh kali ini saya akan memakai file mp3 dari sebuah audio yang terkenal atau viral di internet beberapa tahun yang lalu, yaitu lagu “Nyan Cat”. Dan untuk file mp4 saya juga memakai file yang sama yaitu video “Nyan Cat”. Berikut terlampir code mp3 terlebih dahulu :



Code tersebut sudah saya sertakan komentar pada setiap barisnya agar lebih memudahkan kita untuk memahami maksud code tersebut. Disini kita akan memakai modul dari pygame agar bisa memudahkan kita memainkan file mp3 dengan mudah dan lancar.

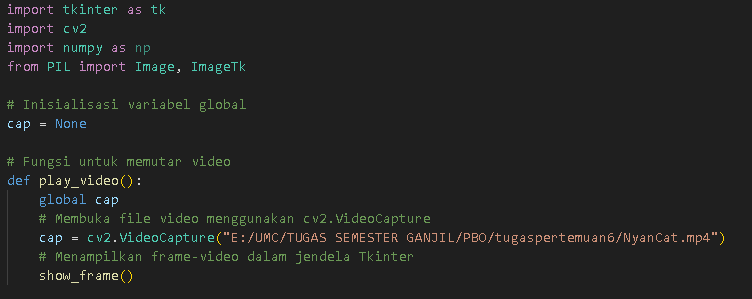
Ada 2 fungsi pada code ini yaitu fungsi untuk memainkan dan menghentikan musik. Untuk memuat file mp3 kita harus memasukan directory file dimana kita menyimpan file mp3nya. Ketika kita salah memasukkan directory akan mengakibatkan program tidak berjalan sebagaimana mestinya. Kita bisa dengan bebas memakai file audio apapun yang kita mau tidak terpaku pada contoh.

Tampilan program tersebut adalah seperti di bawah :

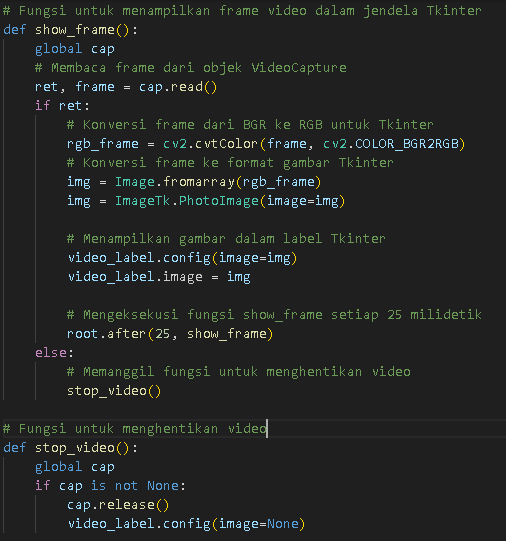


Program ini kita beri nama MP3 Player sesuai dengan fungsinya. Didalamnya terdapat juga tombol “play” untuk memainkan musik, dan tombol stop untuk menghentikan musik.

Selanjutnya adalah code untuk memainkan file mp4 atau video:



Kita akan memakai modul tkinter, cv2 untuk mengcapture video, dan PIL untuk mengkonversi tiap frame video menjadi gambar. Pertama kita akan membuat fungsi play\_video yang isinya adalah untuk mengcapture atau mengambil video dari file yang terdapat pada directory yang sudah kita masukkan. Didalam fungsi play\_video juga kita akan memanggil fungsi selanjutnya yaitu fungsi show\_frame.

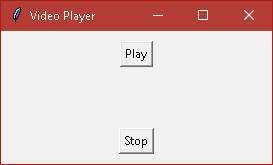


Untuk menampilkan video pada tkinter kita akan menampilkan frame dari vide yang sudah dicapture, Sehingga kita harus membuat fungsi show\_frame terlebih dahulu. Isi dari fungsi tersebut adalah untuk mengambil frame dari video yang sudah di capture oleh modul cv2. Pada fungsi ini terdapat 2 kondisi. Kondisi pertama adalah tkinter akan melakukan konversi tiap frame dari video tersebut. Selanjutnya frame tersebut akan dikonversi lagi menjadi gambar yang akan dimasukan pada label dari modul tkinter. Terakhir fungsi show\_frame akan dieksekusi setiap 25 milidetik. Kondisi kedua adalah kondisi untuk memanggil fungsi menghentikan video.



Langkah selanjutnya kita akan membuat jendela aplikasi menggunakan modul tkinter. Kemudian kita akan membuat button play\_video untuk mengeksekusi fungsi memainkan video. Selanjutnya kita akan membuat label untuk menampilkan video tersebut. Terakhir kita membuat tombol untuk mengeksekusi fungsi menghentikan video.

Ini adalah jendela aplikasi ketika aplikasi pertama dibuka:



Dan ini adalah jendela ketika tombol play ditekan:



**MODUL 4**

**MEMBUAT APLIKASI KONVERSI SUHU BERBASIS WINDOWS**

Aplikasi konversi suhu berbasis windows menggunakan python-tkinter merupakan aplikasi yang dibuat agar kita bisa mengetahui hasil konversi dari 3 satuan suhu yaitu Celcius, Fahrenheit, dan Kelvin, menggunakan Python-Tkinter. *Source code* aplikasi ini berisi beberapa elemen tkinter seperti label, entry, combobox, dan button. Modul 4 ini akan menjelaskan langkah-langkah membuat aplikasi tersebut.

Berikut langkah-langkahnya :

|  |
| --- |
| # Import modul tkinter dengan alias 'tk'  import tkinter as tk  # Import modul ttk dari tkinter  from tkinter import ttk |

Pertama kita akan memasukkan modul tkinter. Modul ini mempunyai beberapa elemen yang akan kita pakai pada pembuatan aplikasi ini.

|  |
| --- |
| # Fungsi untuk melakukan konversi suhu  def konversi():      try:          # Mengambil nilai dari suhu dan mengubahnya menjadi float          nilai\_suhu= float(suhu.get())          # Mengambil satuan suhu yang dipilih dari combobox satuan suhu          satuan\_suhu = combobox\_satuan\_suhu.get()          # Mengambil satuan suhu yang dipilih dari combobox output          konversi\_satuan\_suhu = combobox\_konversi\_suhu.get() |

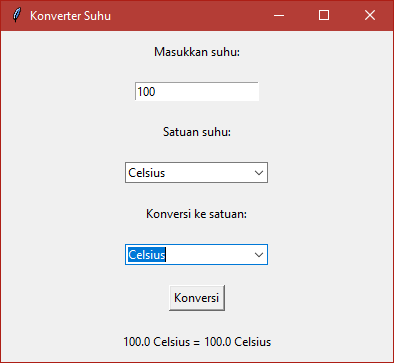
Langkah kedua adalah melakukan deklarasi fungsi untuk melakukan konversi suhu.

Dibawah ini ada beberapa logika pada fungsi konversi:

|  |
| --- |
| # Jika pilihan satuan suhu dan konversi satuan adalah jenis suhu yang sama :          if satuan\_suhu == konversi\_satuan\_suhu:              variabel\_hasil\_konversi.set(f"{nilai\_suhu} {satuan\_suhu} = {nilai\_suhu} {konversi\_satuan\_suhu}") |

Logika yang dilakukan jika ‘satuan\_suhu’ yang dipilih sama dengan ‘konversi\_satuan\_suhu’ yang dipilih. Kedua pilihan ini dipilih user melalui combobox yang disediakan. Jika nilai keduanya sama, maka program akan langsung menampilkan hasil konversu yang sama dengan hasil inputan tanpa dilakukan penghitungan atau operasi matematika.

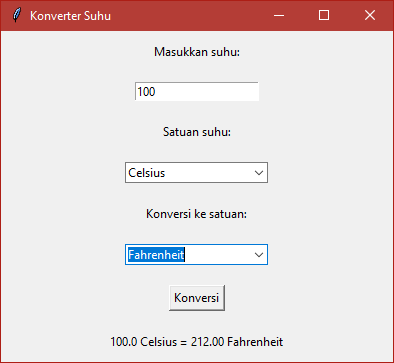
Berikut tampilannya :



|  |
| --- |
| # Jika pilihan satuan suhu adalah celcius dan konversi satuan adalah fahrenheit :          elif satuan\_suhu == "Celsius" and konversi\_satuan\_suhu == "Fahrenheit":              hasil\_konversi = (nilai\_suhu\* 9/5) + 32              variabel\_hasil\_konversi.set(f"{nilai\_suhu} Celsius = {hasil\_konversi:.2f} Fahrenheit") |

Logika jika user memilih ‘satuan\_suhu’ Celcius dan memilih ‘konversi\_satuan\_suhu’ Fahrenheit. Operasi yang dilakukan adalah dengan formula (nilai\_suhu \* 9/5) + 32, dimana nilai suhu adalah hasil inputan user. Hasil dari operasi tersebut akan disimpan di variabel ‘hasil\_konversi’, dan program akan menampilkan hasil inputan dan hasil konversi.

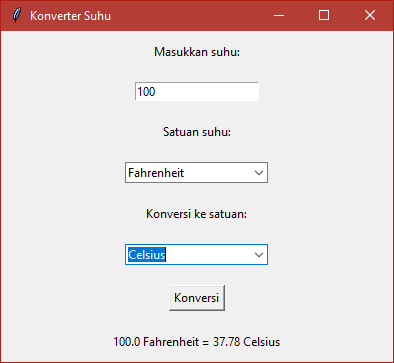
Berikut tampilannya :



|  |
| --- |
| # Jika pilihan satuan suhu adalah fahrenheit dan konversi satuan adalah celcius :          elif satuan\_suhu == "Fahrenheit" and konversi\_satuan\_suhu == "Celsius":              hasil\_konversi = (nilai\_suhu- 32) \* 5/9              variabel\_hasil\_konversi.set(f"{nilai\_suhu} Fahrenheit = {hasil\_konversi:.2f} Celsius") |

Logika jika user memilih ‘satuan\_suhu’ Fahrenheit dan memilih ‘konversi\_satuan\_suhu’ Celcius. Operasi yang dilakukan adalah dengan formula “(nilai\_suhu – 32)\* 5/9”, dimana nilai suhu adalah hasil inputan user. Hasil dari operasi tersebut akan disimpan di variabel ‘hasil\_konversi’, dan program akan menampilkan hasil inputan dan hasil konversi

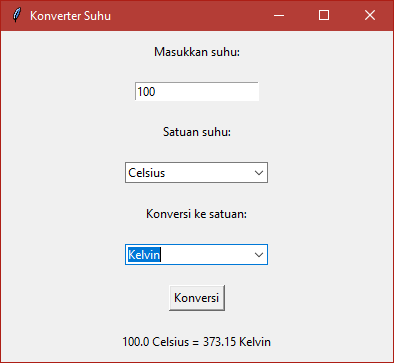
Berikut tampilannya :



|  |
| --- |
| # Jika pilihan satuan suhu adalah celcius dan konversi satuan adalah kelvin :          elif satuan\_suhu == "Celsius" and konversi\_satuan\_suhu == "Kelvin":              hasil\_konversi = nilai\_suhu+ 273.15              variabel\_hasil\_konversi.set(f"{nilai\_suhu} Celsius = {hasil\_konversi:.2f} Kelvin") |

Logika jika user memilih ‘satuan\_suhu’ Celcius dan memilih ‘konversi\_satuan\_suhu’ Kelvin. Operasi yang dilakukan adalah dengan formula “nilai\_suhu + 273.15”, dimana nilai suhu adalah hasil inputan user. Hasil dari operasi tersebut akan disimpan di variabel ‘hasil\_konversi’, dan program akan menampilkan hasil inputan dan hasil konversi.

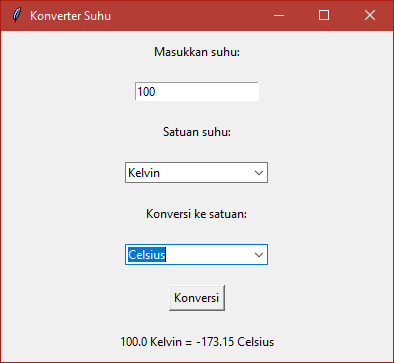
Berikut tampilannya :



|  |
| --- |
| # Jika pilihan satuan suhu adalah kelvin dan konversi satuan adalah celcius :          elif satuan\_suhu == "Kelvin" and konversi\_satuan\_suhu == "Celsius":              hasil\_konversi = nilai\_suhu- 273.15              variabel\_hasil\_konversi.set(f"{nilai\_suhu} Kelvin = {hasil\_konversi:.2f} Celsius") |

Logika jika user memilih ‘satuan\_suhu’ Kelvin dan memilih ‘konversi\_satuan\_suhu’ Celcius. Operasi yang dilakukan adalah dengan formula “nilai\_suhu – 273.15”, dimana nilai suhu adalah hasil inputan user. Hasil dari operasi tersebut akan disimpan di variabel ‘hasi;\_konversi’, dan program akan menampilkan hasil inputan dan hasil konversi.

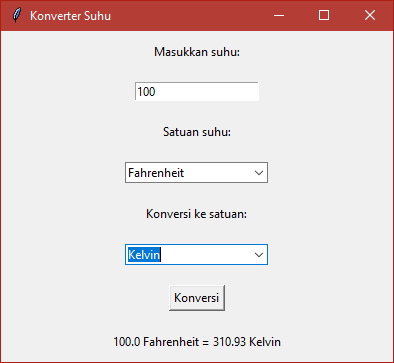
Berikut tampilannya :



|  |
| --- |
| # Jika pilihan satuan suhu adalah fahrenheit dan konversi satuan adalah kelvin :          elif satuan\_suhu == "Fahrenheit" and konversi\_satuan\_suhu == "Kelvin":              celsius = (nilai\_suhu- 32) \* 5/9              hasil\_konversi = celsius + 273.15              variabel\_hasil\_konversi.set(f"{nilai\_suhu} Fahrenheit = {hasil\_konversi:.2f} Kelvin") |

Logika jika user memilih ‘satuan\_suhu’ Fahrenheit dan memilih ‘konversi\_satuan\_suhu’ Kelvin. Disini ada dua operasi dan dua formula yang digunakan penulis. Operasi pertama dengan formula mengubah Fahrenheit ke Celcius yaitu “(nilai\_suhu – 32)\* 5/9”, dimana ‘nilai\_suhu’ adalah hasil inputan user. Hasil dari operasi pertama akan disimpan dalam variabel ‘celcius’. Operasi kedua dengan formula “celcius + 273.15”, dimana ‘celcius’ merupakan variabel hasil dari operasi pertama, dan hasil operasi kedua ini akan disimpan di variabel ‘hasil\_konversi’, dan program akan menampilkan hasil inputan dan hasil konversi.

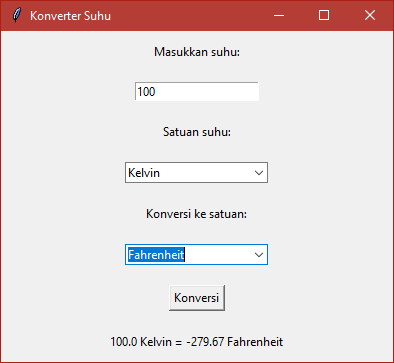
Berikut tampilannya :



|  |
| --- |
| # Jika pilihan satuan suhu adalah kelvin dan konversi satuan adalah fahrenheit :          elif satuan\_suhu == "Kelvin" and konversi\_satuan\_suhu == "Fahrenheit":              celsius = nilai\_suhu- 273.15              hasil\_konversi = (celsius \* 9/5) + 32              variabel\_hasil\_konversi.set(f"{nilai\_suhu} Kelvin = {hasil\_konversi:.2f} Fahrenheit") |

Logika jika user memilih ‘ssatuan\_suhu’ Kelvin dan memilih ‘konversi\_satuan\_suhu’ Fahrenheit. Disini juga ada dua operasi dan dua formula yang digunakan penulis. Operasi pertama dengan formulan mengubah Kelvin ke Celcius yaitu “nilai\_suhu – 273.15”, dimana ‘nilai\_suhu’ adalah hasil inputan user. Hasil dari operasi pertama akan disimpan dalam variabel ‘celcius’. Operasi kedua dengan formula “(celcius \* 9/5) + 32”, dimana ‘celcius’ merupakan variabel hasil dari operasi pertama, dan hasil operasi kedua ini akan disimpan di variabel ‘hasil\_konversi’, dan program akan menampilkan hasil inputan dan hasil konversi.

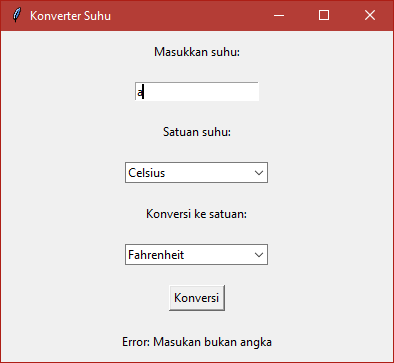
Berikut tampilannya:



|  |
| --- |
| # Menangani exception jika nilai yang dimasukkan tidak valid      except ValueError :          variabel\_hasil\_konversi.set("Error: Masukan bukan angka") |

Jika input yang dilakukan user adalah huruf bukan angka, maka program akan menampilkan pesan “Error: Masukan bukan angka”.

Berikut tampilannya:



Selanjutnya adalah langkah-langkah untuk membuat tampilan program:

|  |
| --- |
| # Membuat instance dari Tkinter  root = tk.Tk()  # Memberi judul pada windows  root.title("Konverter Suhu") |

Selanjutnya adalah langkah untuk membuat label dan entry untuk input suhu:

|  |
| --- |
| # Membuat label untuk input suhu  label\_input = tk.Label(root, text="Masukkan suhu:")  label\_input.pack(pady=10)  # Membuat entry widget untuk memasukkan suhu  suhu = tk.Entry(root)  suhu.pack(pady=10) |

Selanjutnya adalah langkah untuk membuat label untutk satuan suhu dan combobox untuk memilih satuan suhu pada input:

|  |
| --- |
| # Membuat label untuk satuan suhu pada input  label\_input\_unit = tk.Label(root, text="Satuan suhu:")  label\_input\_unit.pack(pady=10)  # Membuat combobox untuk memilih satuan suhu pada input  combobox\_satuan\_suhu = ttk.Combobox(root, values=["Celsius", "Fahrenheit", "Kelvin"])  combobox\_satuan\_suhu.pack(pady=10)  combobox\_satuan\_suhu.set("Celsius") |

Selanjutnya adalah langkah untuk membuat label untutk satuan suhu dan combobox untuk memilih satuan suhu pada output:

|  |
| --- |
| # Membuat label untuk satuan suhu pada output  label\_output\_unit = tk.Label(root, text="Konversi ke satuan:")  label\_output\_unit.pack(pady=10)  # Membuat combobox untuk memilih satuan suhu pada output  combobox\_konversi\_suhu = ttk.Combobox(root, values=["Celsius", "Fahrenheit", "Kelvin"])  combobox\_konversi\_suhu.pack(pady=10)  combobox\_konversi\_suhu.set("Fahrenheit") |

Selanjutnya adalah langkah untuk membuat tombol konversi suhu:

|  |
| --- |
| # Membuat tombol untuk melakukan konversi  tombol\_konversi = tk.Button(root, text="Konversi", command=konversi)  tombol\_konversi.pack(pady=10) |

Selanjutnya adalah langkah untuk membuat variabelstringvar untuk menampilkan hasil konversi pada label:

|  |
| --- |
| # Variabel StringVar untuk menampilkan hasil konversi pada label  variabel\_hasil\_konversi = tk.StringVar()  # Membuat label untuk menampilkan hasil konversi  label\_hasil\_konversi = tk.Label(root, textvariable = variabel\_hasil\_konversi)  label\_hasil\_konversi.pack(pady=10) |

Terakhir adalah menjalankan program dan memainkan loop tkinter:

|  |
| --- |
| root.mainloop() |

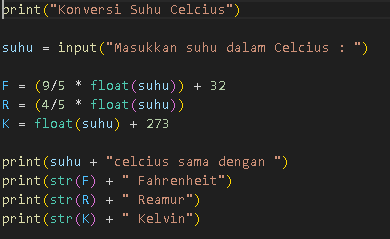
**MODUL 5**

**PERBEDAAN PEMROGRAMAN TIDAK TERSTRUKTUR, TERSTRUKTUR, DAN OBJECT ORIENTED PROGRAMMING**

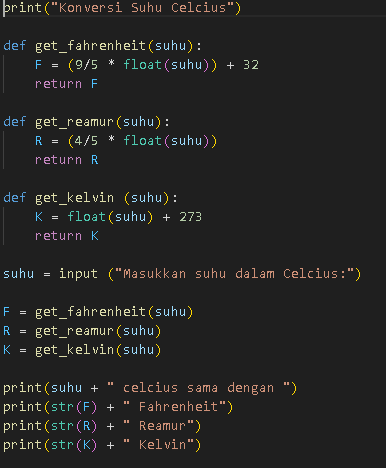
Modul 5 ini akan membahas mengenai perbedaan-perbedaan yang sering kita temui saat membuat program. Ada 3 perbedaan yang mendasar dari gaya pemrograman ini, yaitu pemrograman tidak terstruktur, pemrograman terstruktur, dan object oriented programming(OOP).

Untuk lebih mudah memahaminya berikut akan ditampilkan 3 code yang berbeda dengan tujuan yang sama yaitu melakukan konversi suhu:

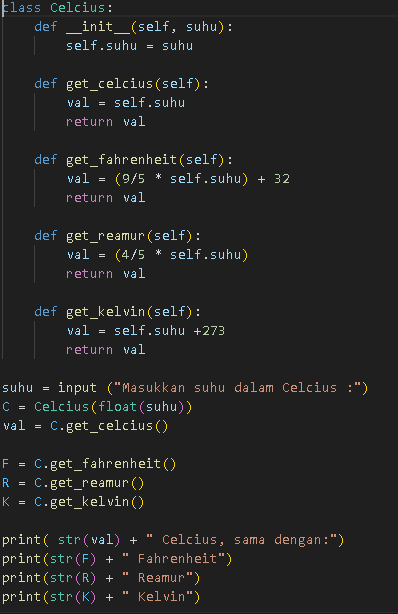
1. Tidak terstruktur



1. Terstruktur



1. OOP



Perbedaan mendasar adalah pada pemrograman tidak terstruktur seperti pada modul 1, semua variabel langsung kita berikan nilainya dan tidak memakai fungsi untuk melakukan konversinya. Sedangkan untuk yang terstruktur dan OOP terdapat fungsi untuk melakukan konversi.

Perbedaan kedua untuk yang terstruktur dan oop adalah pada oop kita membuat sebuah class yang terdapat fungsi \_\_init\_\_ didalamnya. Fungsi init adalah fungsi yang akan otomatis dijalankan ketika kita mulai membuka program. Sedangkan pada terstruktur tidak ada fungsi tersebut.

**MODUL 6**

**PENGENALAN MySQL & SQL**

**MySQL** adalah sistem manajemen basis data relasional open-source yang sangat populer. Istilah "MySQL" merupakan gabungan dari "My", nama putrinya dari salah satu pendiri, Michael Widenius, dan "SQL", singkatan untuk Structured Query Language.

MySQL memungkinkan pengguna untuk menyimpan, mengelola, dan mengambil data struktural secara efisien. Ini digunakan secara luas untuk berbagai aplikasi, mulai dari proyek skala kecil hingga situs web skala besar dan solusi tingkat perusahaan. Adapun kelebihan dari MySQL, diantaranya:

1. Open Source: MySQL adalah aplikasi open source yang memiliki lisensi GPL gratis di bawah pengelolaan Oracle. Hal ini memastikan kinerja aplikasi ini dan memudahkan pencarian solusi jika ada masalah.
2. Tipe Data Bervariasi: MySQL menyediakan berbagai tipe data, seperti integer, float, double, dan date. Hal ini memudahkan pengelolaan database server saat mengembangkan software yang kompleks dan terstruktur.
3. Tidak Memerlukan RAM Besar: MySQL dapat digunakan meski kapasitas server rendah atau terbatas. Ini membuat MySQL menjadi pilihan yang baik untuk pengguna dengan spesifikasi hardware yang rendah.
4. Keamanan Terjamin: Meskipun MySQL adalah open source, software ini menyediakan fitur keamanan yang terjamin, seperti level pada host dan subnetmask, serta enkripsi password yang tinggi.
5. Mendukung Multi User: MySQL dapat digunakan bersama-sama secara real time, memudahkan pengerjaan proyek bersama tim dalam satu waktu.
6. Terintegrasi dengan Bahasa Pemrograman Lain: MySQL dapat diintegrasikan dengan berbagai bahasa pemrograman seperti R, Python, C++, dan lainnya.

Selain kelebihan dari MySQL di atas, berikut adalah beberapa kekurangan MySQL:

1. Sulit Mengelola Data yang Besar: MySQL adalah software yang ramah dengan spesifikasi rendah. Jika Anda ingin menggunakan database yang cukup besar, ada baiknya untuk menggunakan sistem manajemen database selain MySQL.
2. Kurangnya Technical Support: Karena bersifat open source, MySQL tidak memiliki technical support yang memadai. Hal ini mempersulit pengguna untuk menghubungi technical support.
3. Kurang Baik untuk Aplikasi Berbasis Game dan Mobile: Banyak developer game dan mobile tidak menggunakan MySQL karena manajemen database MySQL kurang cocok untuk mengelola aplikasi tersebut.

**SQL** merupakan bahasa pemrograman yang perlu Anda pahami karena dapat merelasikan antara beberapa tabel dengan database maupun antar database. Ada tiga bentuk SQL yang perlu Anda ketahui, yaitu Data Definition Language (DDL), Data Manipulation Language(DML), dan Data Control Language (DCL).

|  |  |
| --- | --- |
| **CREATE** | **Dipakai untuk membuat tabel dan database.** |
| **DROP** | **Dipakai untuk menghapus database dan tabel.** |
| **ALTER** | **Dipakai untuk mengubah struktur tabel yang sudah ada.  Alter dapat mengganti field menggunakan perintah “Change”, menambahkan field menggunakan perintah “Add”, atau menghapus field menggunakan perintah “drop”, dan mengubah namanya menggunakan perintah “Rename”.** |

**Data Definition Language (DDL)**

DDL berguna pada saat Anda ingin mendefinisikan data di dalam database. Terdapat beberapa query yang dikelompokkan ke dalam DDL, yaitu:

Argumen DDL di atas perlu Anda pahami karena merupakan dasar penggunaan SQL di bagian awal pembuatan database. Contohnya saja jika belum menjalankan perintah “CREATE”, Anda belum bisa melanjutkan penggunaan argumen yang lainnya.

**Data Manipulation Language (DML)**

DML dapat Anda pakai setelah menjalankan perintah DDL. DML berfungsi untuk memanipulasi, mengubah, atau mengganti isi dari database (tabel) yang sudah ada.

Terdapat beberapa perintah DML yang perlu Anda ketahui, yaitu:

|  |  |
| --- | --- |
| **INSERT** | **Dipakai untuk memasukkan data ke dalam tabel pada database.** |
| **UPDATE** | **Dipakai untuk mengubah data yang ada di dalam tabel pada database.** |
| **DELETE** | **Dipakai untuk menghapus data di dalam tabel pada database.** |

**Data Control Language (DCL)**

Jika Anda sudah mempunyai user dan ingin mengatur hak akses masing-masing user, Anda sebaiknya memahami berbagai macam jenis DCL dan cara penggunaannya. DCL berguna untuk memberikan hak akses database, mendefinisikan space, mengalokasikan space, dan melakukan audit penggunaan database.

Terdapat beberapa perintah DCL yang perlu Anda ketahui, yaitu:

|  |  |
| --- | --- |
| **GRANT** | **Dipakai untuk memberikan izin kepada user untuk mengakses database.** |
| **REVOKE** | **Dipakai untuk membatalkan izin user untuk mengakses database.** |
| **COMMIT** | **Dipakai untuk menetapkan penyimpanan pada database.** |
| **ROOLBACK** | **Dipakai untuk membatalkan penyimpanan pada database.** |

MySQL termasuk kelompok database RDBMS (Relationship Database Management System) merupakan database yang terdiri atas tabel-tabel. Database mana saja yang termasuk ke dalam RDBMS:

1. ORACLE

2. SQL SERVER

3. MYSQL

4. POSTGRESQL

5. SQLite

Selain kelompok RDBMS ada kelompok database yang lain yaitu: No SQL

1. MongoDB

2. Apache HBase

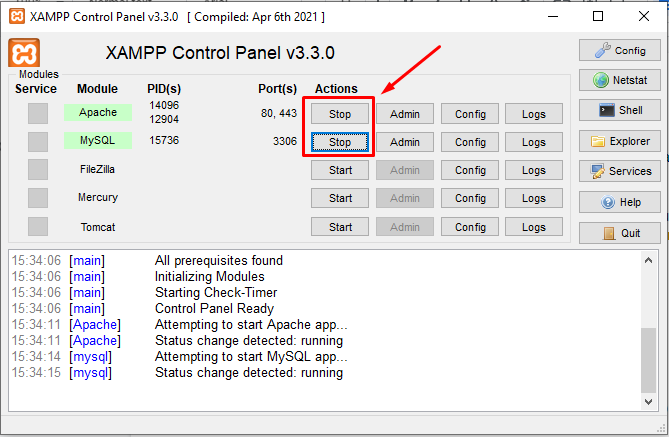
3. Cassandra

4. couchdb

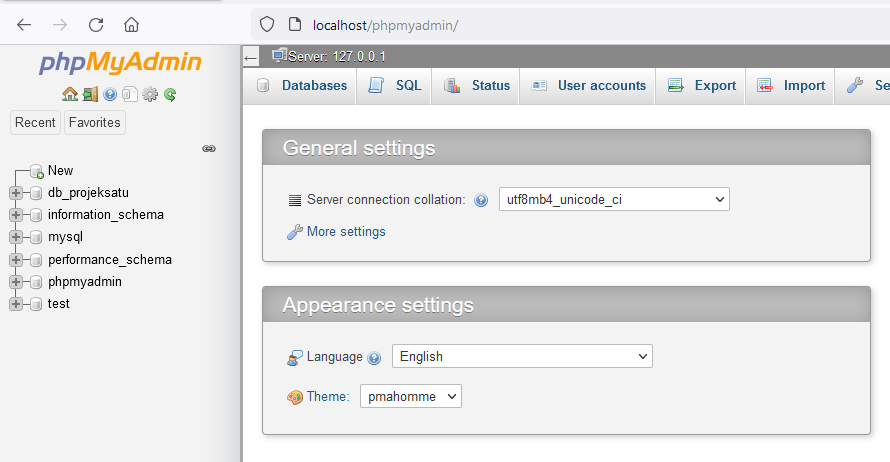
**Cara mengakses database di localhost phpMyAdmin** sangat mudah dilakukan. Anda cukup mengetikkan localhost/phpmyadmin di address bar browser. Namun, pastikan Anda sudah menginstall aplikasi XAMPP pada komputer yang digunakan. Jika Anda belum menginstall aplikasi XAMPP pada komputer, Anda dapat mengikuti langkah berikut ini:

Pertama, download installer XAMPP melalui halaman <https://www.apachefriends.org/> . Setelah selesai, cari file installer yang biasanya tersimpan pada folder Download di komputer. Selanjutnya klik install dan ikuti instruksi yang diberikan.

Setelah XAMPP berhasil terinstall, jalankan service Apache dan database MySQL.



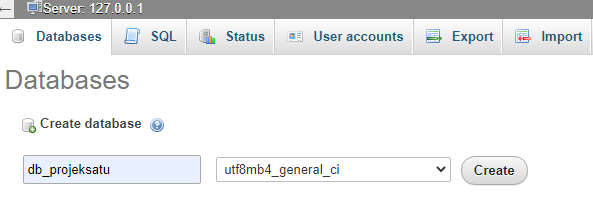
Selanjutnya, ketikkan URL localhost/phpmyadmin di browser yang digunakan. Akan muncul halaman phpMyAdmin seperti tampilan berikut.



**Cara Membuat Database di Localhost phpMyAdmin XAMPP**

Langkah pertama yang perlu Anda lakukan adalah membuat nama database, dengan panduan berikut:

1. Klik menu **Databases** > Isi nama database yang akan digunakan > Klik tombol **Create**.



Setelah database selesai dibuat, lanjutkan dengan membuat user untuk mengakses database tersebut.

2. Klik menu **Privileges > Add user account >** centang hak akses yang akan diberikan pada user tersebut **>** klik **Add user.**

Jika Anda sudah memiliki file database, Anda dapat mengupload database tersebut ke localhost XAMPP melalui menu **Import**. File yang dapat diimport ke phpMyAdmin ini antara lain file dengan format .sql, gzip, bzip, atau zip dengan ukuran file hingga 40 MB. Jika Anda ingin mengupload file dengan ukuran lebih besar, Anda dapat melakukan import database melalui cmd (Command Prompt).



Selain menu import, ada juga menu **export**. Menu Export ini digunakan apabila Anda ingin memiliki file backup database, atau ingin memindahkan database ke server yang lain. Untuk melakukan export, silakan klik nama databasenya, kemudian klik tombol export. Selanjutnya ada pilihan format export yang akan digunakan. Beberapa format yang didukung antara lain CSV, SQL, PDF, Ms Excel, Ms Word dan lainnya.

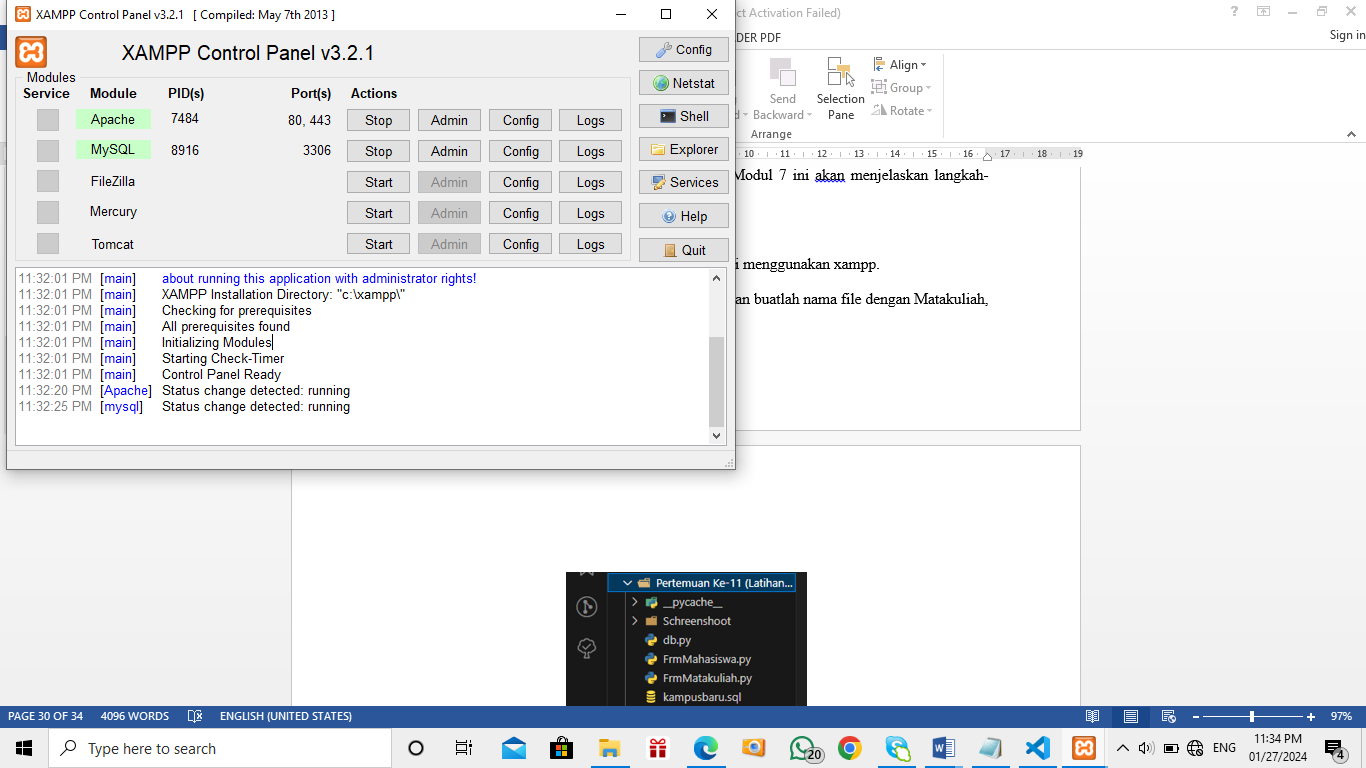
**MODUL 7**

**Membuat Aplikasi Sederhana Berbasis Windows Python-Xampp**

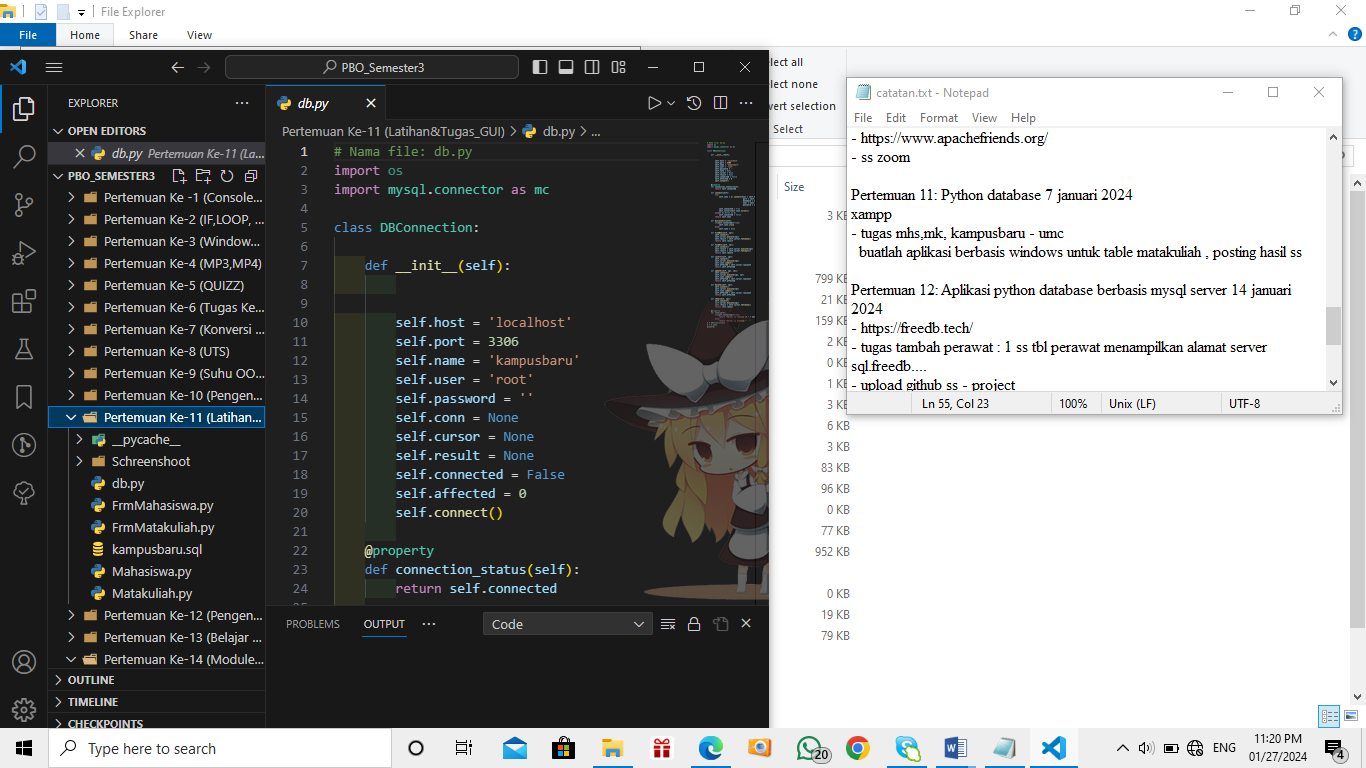
Aplikasi sederhana berbasis windows menggunakan python-xampp merupakan aplikasi yang dibuat agar kita bisa mengetahui cara mengkoneksikan database kedalam sebuah program yang kita buat, menggunakan Python-Xampp. *Source code* aplikasi ini berisi beberapa elemen tkinter seperti label, entry, combobox. Sehingga Modul 7 ini akan menjelaskan langkah-langkah membuat aplikasi tersebut.

Berikut langkah-langkahnya:

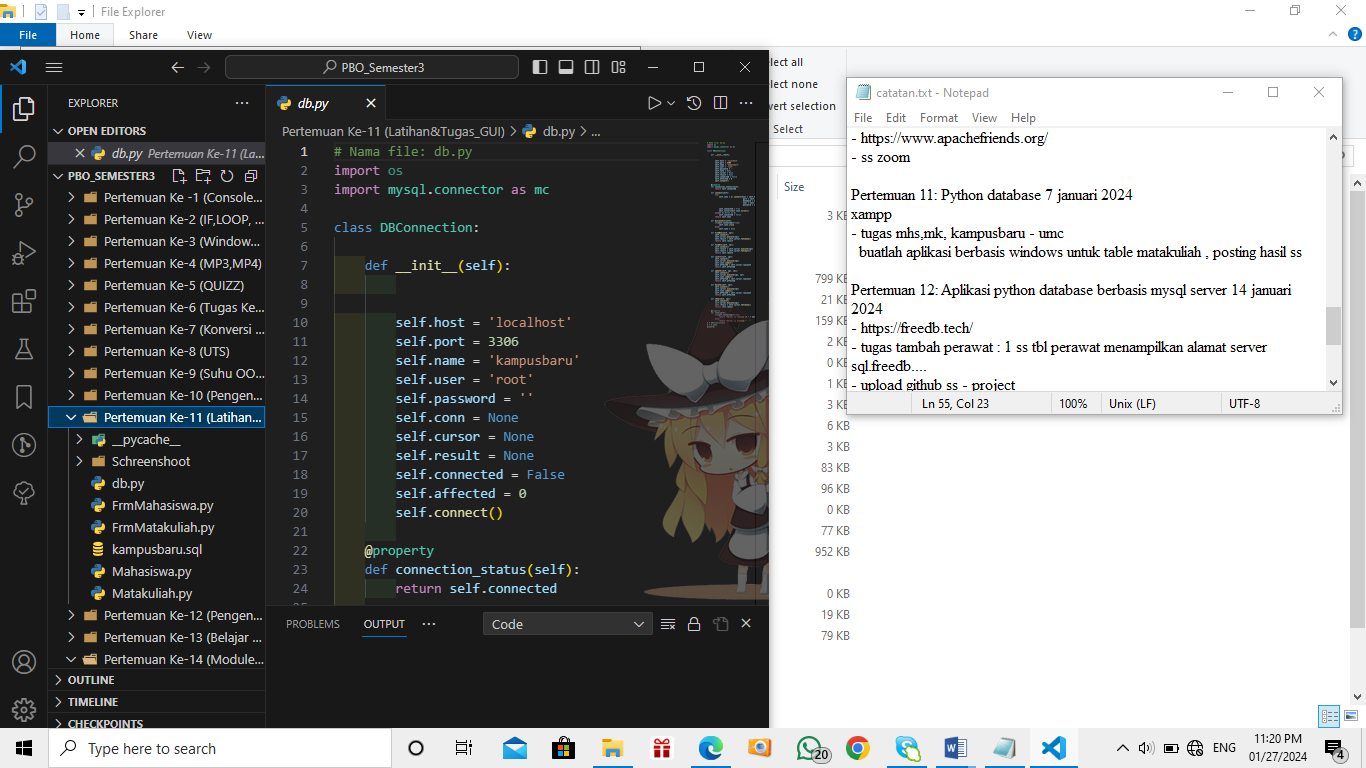
Pertama aktifkanlah server nya terlebih dahulu kita disini menggunakan xampp.



Kedua buatlah database dengan nama file db.py, kemudian buatlah nama file dengan Matakuliah, Mahasiswa, FrmMatakuliah, FrmMahasiswa.



Berikut script untuk mengkoneksikan ke database yang ada pada file db.py.



Setelah itu baru kita masukan script codingan pada salah satu file Matakuliah.py.

# Nama file: Matakuliah.py

from db import DBConnection as mydb

class Matakuliah:

    def \_\_init\_\_(self):

        self.\_\_id=None

        self.\_\_kodemk=None

        self.\_\_namamk=None

        self.\_\_sks=None

        self.\_\_info = None

        self.conn = None

        self.affected = None

        self.result = None

    @property

    def info(self):

        if(self.\_\_info==None):

            return "KODEMK:" + self.\_\_kodemk + "\n" + "Nama MK:" + self.\_\_namamk + "\n" + "Jumlah SKS:" + self.\_\_sks

        else:

            return self.\_\_info

    @info.setter

    def info(self, value):

        self.\_\_info = value

    @property

    def id(self):

        return self.\_\_id

    @property

    def kodemk(self):

        return self.\_\_kodemk

    @kodemk.setter

    def kodemk(self, value):

        self.\_\_kodemk = value

    @property

    def namamk(self):

        return self.\_\_namamk

    @namamk.setter

    def namamk(self, value):

        self.\_\_namamk = value

    @property

    def sks(self):

        return self.\_\_sks

    @sks.setter

    def sks(self, value):

        self.\_\_sks = value

    def simpan(self):

        self.conn = mydb()

        val = (self.\_\_kodemk, self.\_\_namamk, self.\_\_sks)

        sql="INSERT INTO matakuliah (kodemk, namamk, sks) VALUES " + str(val)

        self.affected = self.conn.insert(sql)

        self.conn.disconnect

        return self.affected

    def update(self, id):

        self.conn = mydb()

        val = (self.\_\_kodemk, self.\_\_namamk, self.\_\_sks, id)

        sql="UPDATE matakuliah SET kodemk = %s, namamk = %s, sks=%s WHERE id=%s"

        self.affected = self.conn.update(sql, val)

        self.conn.disconnect

        return self.affected

    def updateByNIM(self, kodemk):

        self.conn = mydb()

        val = (self.\_\_namamk, self.\_\_sks, kodemk)

        sql="UPDATE matakuliah SET namamk = %s, sks=%s WHERE kodemk=%s"

        self.affected = self.conn.update(sql, val)

        self.conn.disconnect

        return self.affected

    def delete(self, id):

        self.conn = mydb()

        sql="DELETE FROM matakuliah WHERE id='" + str(id) + "'"

        self.affected = self.conn.delete(sql)

        self.conn.disconnect

        return self.affected

    def deleteByNIM(self, kodemk):

        self.conn = mydb()

        sql="DELETE FROM matakuliah WHERE kodemk='" + str(kodemk) + "'"

        self.affected = self.conn.delete(sql)

        self.conn.disconnect

        return self.affected

    def getByID(self, id):

        self.conn = mydb()

        sql="SELECT \* FROM matakuliah WHERE id='" + str(id) + "'"

        self.result = self.conn.findOne(sql)

        self.\_\_kodemk = self.result[1]

        self.\_\_namamk = self.result[2]

        self.\_\_sks = self.result[3]

        self.conn.disconnect

        return self.result

    def getByNIM(self, kodemk):

        a=str(kodemk)

        b=a.strip()

        self.conn = mydb()

        sql="SELECT \* FROM matakuliah WHERE kodemk='" + b + "'"

        self.result = self.conn.findOne(sql)

        if(self.result!=None):

            self.\_\_kodemk = self.result[1]

            self.\_\_namamk = self.result[2]

            self.\_\_sks = self.result[3]

            self.affected = self.conn.cursor.rowcount

        else:

            self.\_\_kodemk = ''

            self.\_\_namamk = ''

            self.\_\_sks = ''

            self.affected = 0

        self.conn.disconnect

        return self.result

    def getAllData(self):

        self.conn = mydb()

        sql="SELECT \* FROM matakuliah"

        self.result = self.conn.findAll(sql)

        return self.result

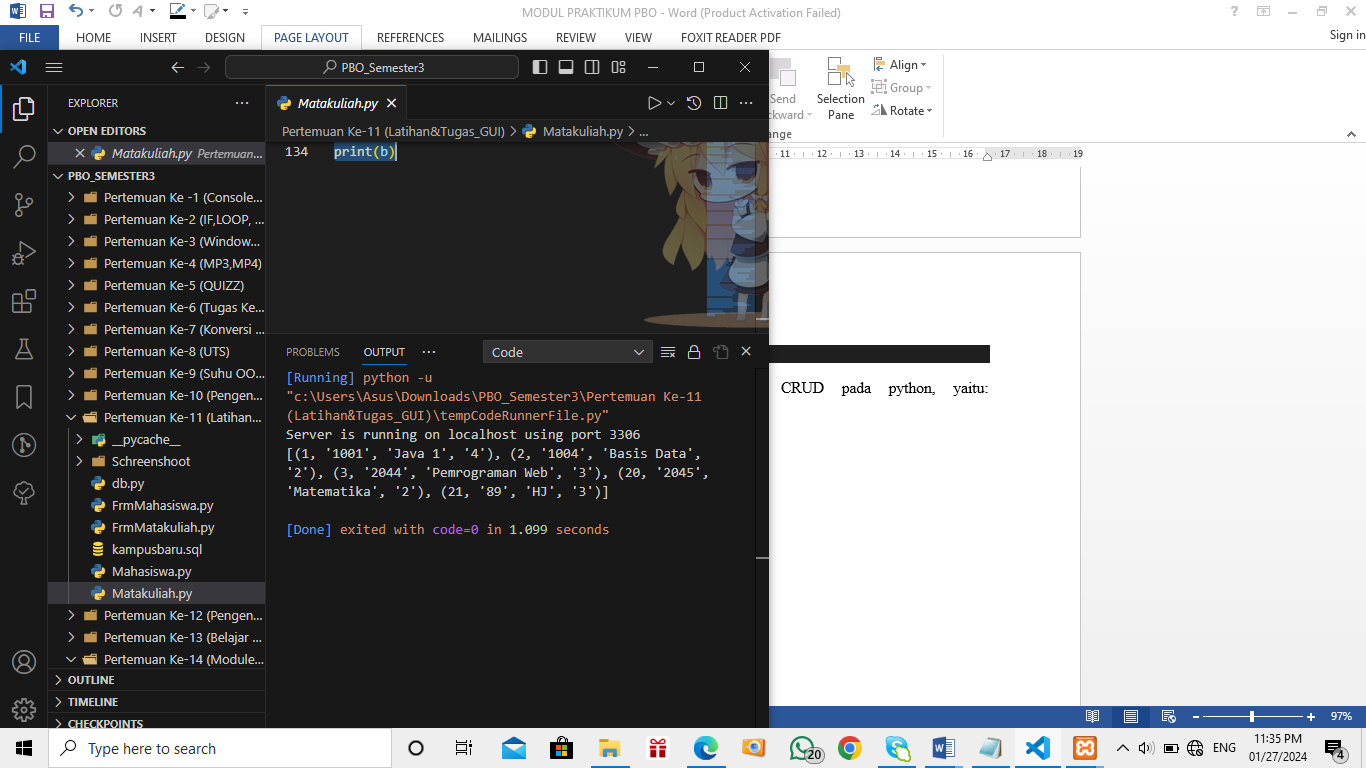
a = Matakuliah()

b = a.getAllData()

print(b)

Pada source code diatas sudah terdapat perintah CRUD pada python, yaitu: Creat,Read,Update,Delete.

Output:



Kemudian masukan source code berikut pada file FrmMatakuliah.py yang akan memunculkan layar Windows dengan TkInter.

# Nama file: FrmMatakuliah.py

import tkinter as tk

from tkinter import Frame,Label,Entry,Button,Radiobutton,ttk,VERTICAL,YES,BOTH,END,Tk,W,StringVar,messagebox

from Matakuliah import Matakuliah

import os

os.system("cls")

class FormMatakuliah:

    def \_\_init\_\_(self, parent, title):

        self.parent = parent

        self.parent.geometry("550x450")

        self.parent.title(title)

        self.parent.protocol("WM\_DELETE\_WINDOW", self.onKeluar)

        self.ditemukan = None

        self.aturKomponen()

        self.onReload()

    def aturKomponen(self):

        mainFrame = Frame(self.parent, bd=10)

        mainFrame.pack(fill=BOTH, expand=YES)

        # Label

        Label(mainFrame, text='Data Matakuliah Universitas Muhammadiyah Cirebon   ').grid(row=15, column=1,

            sticky=W, padx=5, pady=5)

        Label(mainFrame, text='KODE MK                   :').grid(row=1, column=0,

            sticky=W, padx=5, pady=5)

        Label(mainFrame, text='Nama MK                  :').grid(row=2, column=0,

            sticky=W, padx=5, pady=5)

        Label(mainFrame, text='Jumlah SKS                :').grid(row=3, column=0,

            sticky=W, padx=5, pady=5)

        # Textbox

        self.txtKODEMK = Entry(mainFrame)

        self.txtKODEMK.grid(row=1, column=1, padx=5, pady=5)

        self.txtKODEMK.bind("<Return>",self.onCari) # menambahkan event Enter key

        self.txtNamaMK = Entry(mainFrame)

        self.txtNamaMK.grid(row=2, column=1, padx=5, pady=5)

        # Radio Button

        # self.txtJK = StringVar()

        # self.L = Radiobutton(mainFrame, text='Laki-laki', value='L', variable=self.txtJK)

        # self.L.grid(row=3, column=1, padx=5, pady=5, sticky=W)

        # self.L.select() # set pilihan yg pertama

        # self.P = Radiobutton(mainFrame, text='Perempuan', value='P', variable=self.txtJK)

        # self.P.grid(row=4, column=1, padx=5, pady=5, sticky=W)

        # Combo Box

        self.txtSKS = StringVar()

        Cbo = ttk.Combobox(mainFrame, width = 17, textvariable = self.txtSKS)

        Cbo.grid(row=3, column=1, padx=5, pady=5)

        # Adding combobox drop down list

        Cbo['values'] = ('1','2', '3', '4')

        Cbo.current()

        # Button

        self.btnSimpan = Button(mainFrame, text='Save', command=self.onSimpan, width=10,  fg= "white", bg="blue")

        self.btnSimpan.grid(row=6, column=0, padx=5, pady=5)

        self.btnClear = Button(mainFrame, text='Clear', command=self.onClear, width=10,  fg= "black", bg="yellow")

        self.btnClear.grid(row=6, column=1, padx=5, pady=5)

        self.btnHapus = Button(mainFrame, text='Delete', command=self.onDelete, width=10,  fg= "white", bg="red")

        self.btnHapus.grid(row=6, column=2, padx=5, pady=5)

        self.btnCari = Button(mainFrame, text='Cari KODE MK', command=self.onCari, width=10,  fg= "white", bg="green")

        self.btnCari.grid(row=1, column=2, padx=5, pady=5)

        # define columns

        columns = ('id', 'kodemk', 'namamk','sks')

        self.tree = ttk.Treeview(mainFrame, columns=columns, show='headings')

        # define headings

        self.tree.heading('id', text='No')

        self.tree.column('id', width="30")

        self.tree.heading('kodemk', text='KODE MK')

        self.tree.column('kodemk', width="60")

        self.tree.heading('namamk', text='Nama MK')

        self.tree.column('namamk', width="200")

        self.tree.heading('sks', text='Jumlah SKS')

        self.tree.column('sks', width="100")

        # set tree position

        self.tree.place(x=80, y=165)

        self.onReload()

    def onClear(self, event=None):

        self.txtKODEMK.delete(0,END)

        self.txtKODEMK.insert(END,"")

        self.txtNamaMK.delete(0,END)

        self.txtNamaMK.insert(END,"")

        self.txtSKS.set("")

        self.btnSimpan.config(text="Simpan")

        # self.L.select()

        self.onReload()

        self.ditemukan = False

    def onReload(self, event=None):

        # get data Matakuliah

        mk = Matakuliah()

        result = mk.getAllData()

        for item in self.tree.get\_children():

            self.tree.delete(item)

        students=[]

        for row\_data in result:

            students.append(row\_data)

        for student in students:

            self.tree.insert('',END, values=student)

    def onCari(self, event=None):

        kodemk = self.txtKODEMK.get()

        mk = Matakuliah()

        res = mk.getByNIM(kodemk)

        rec = mk.affected

        if(rec>0):

            messagebox.showinfo("showinfo", "Data Ditemukan")

            self.TampilkanData()

            self.ditemukan = True

        else:

            messagebox.showwarning("showwarning", "Data Tidak Ditemukan")

            self.ditemukan = False

            self.txtNamaMK.focus()

        return res

    def TampilkanData(self, event=None):

        kodemk = self.txtKODEMK.get()

        mk = Matakuliah()

        res = mk.getByNIM(kodemk)

        self.txtNamaMK.delete(0,END)

        self.txtNamaMK.insert(END,mk.namamk)

        # jk = mk.jk

        # if(jk=="P"):

        #    self.P.select()

        # else:

        #    self.L.select()

        self.txtSKS.set(mk.sks)

        self.btnSimpan.config(text="Update")

    def onSimpan(self, event=None):

        kodemk = self.txtKODEMK.get()

        namamk = self.txtNamaMK.get()

        # jk = self.txtJK.get()

        sks = self.txtSKS.get()

        mk = Matakuliah()

        mk.kodemk = kodemk

        mk.namamk = namamk

        # mk.jk = jk

        mk.sks = sks

        if(self.ditemukan==True):

            res = mk.updateByNIM(kodemk)

            ket = 'Diperbarui'

        else:

            res = mk.simpan()

            ket = 'Disimpan'

        rec = mk.affected

        if(rec>0):

            messagebox.showinfo("showinfo", "Data Berhasil "+ket)

        else:

            messagebox.showwarning("showwarning", "Data Gagal "+ket)

        self.onClear()

        return rec

    def onDelete(self, event=None):

        kodemk = self.txtKODEMK.get()

        mk = Matakuliah()

        mk.kodemk = kodemk

        if(self.ditemukan==True):

            res = mk.deleteByNIM(kodemk)

            rec = mk.affected

        else:

            messagebox.showinfo("showinfo", "Data harus ditemukan dulu sebelum dihapus")

            rec = 0

        if(rec>0):

            messagebox.showinfo("showinfo", "Data Berhasil dihapus")

        self.onClear()

    def onKeluar(self, event=None):

        # memberikan perintah menutup aplikasi

        self.parent.destroy()

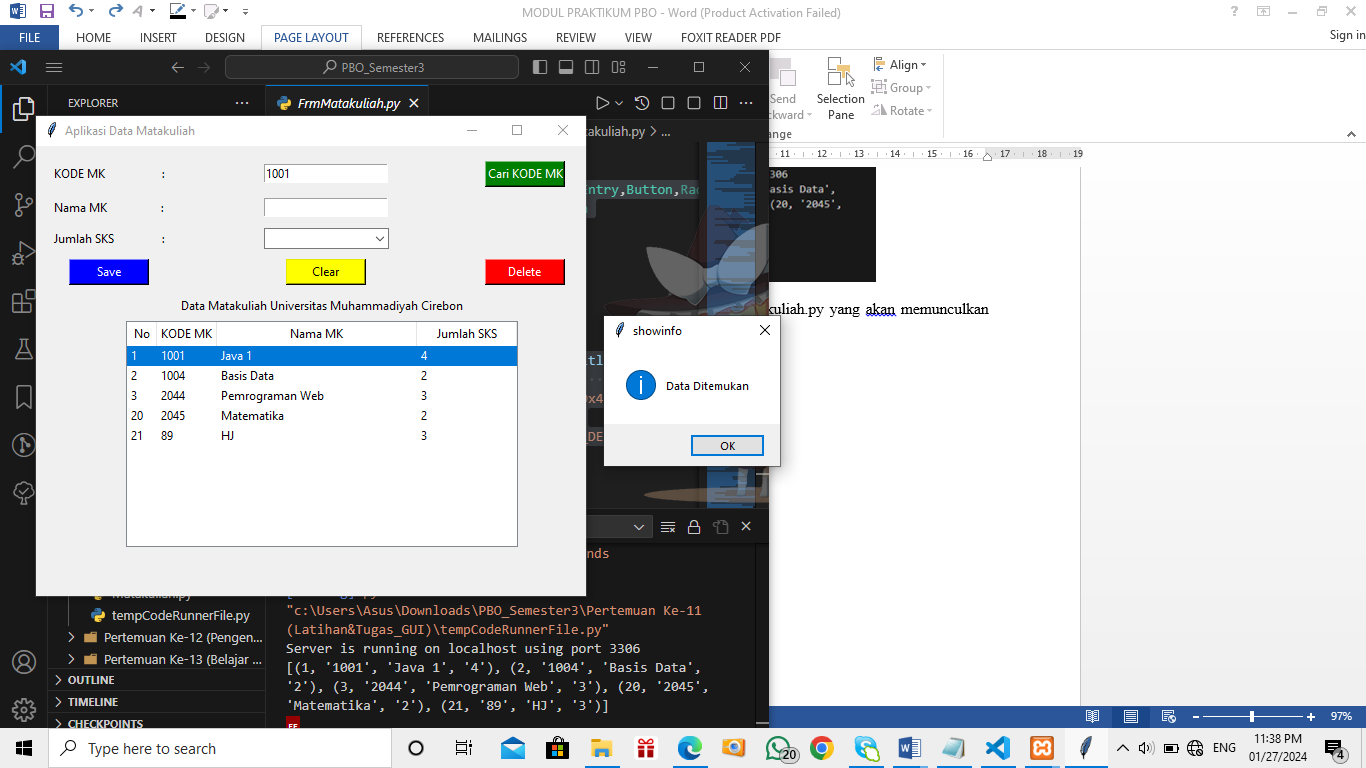
if \_\_name\_\_ == '\_\_main\_\_':

    root2 = tk.Tk()

    aplikasi = FormMatakuliah(root2, "Aplikasi Data Matakuliah")

    root2.mainloop()

Next, maka akan muncul seperti layar dibawah ini ketika program dijalankan.



Maka program tersebut sudah berhasil dijalankan.

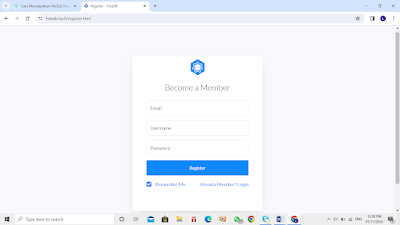
**MODUL 8**

**MEMBUAT APLIKASI BERBASIS WINDOWS**

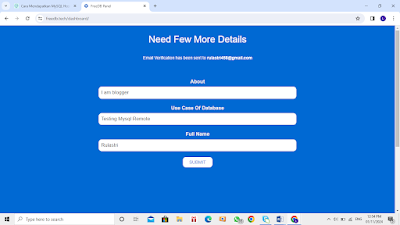
**DENGAN MYSQL SERVER ONLINE (FREEDB.TECH)**

Dalam modul 8 ini membahas tentang bagaimana cara membuat aplikasi berbasis windows dengan mysql server online yang biasa disebut remote mysql (freedb.tech).

Pertama, Isikan formulir yang berupa email, username, dan password dengan tepat dan benar, klik Register.

[](https://blogger.googleusercontent.com/img/a/AVvXsEgleaPFX64x5CwL-maCoHc9AW6oeeozSrdUg2o4b3RvqYsOvL-13HE2B4WKntE3Om0D55CZZjF12CfQ5_31WG5eEDb65Kkw3rg5kgSGcM0ikB_x2mFk1PNJ0XnOhR7T9XamLx8GSq_GsABeafGHHQHeJhVsOzu07NVQr1VZFSz23heVwJ7wc_mhZuxBfTqV)

Kedua, Isikan formulir untuk survei akun dengan sesuai jawaban kamu, klik Submit.

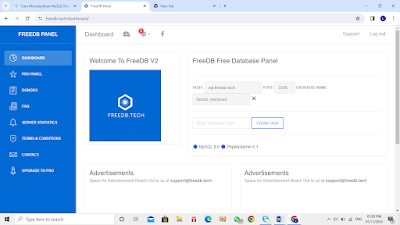
[](https://blogger.googleusercontent.com/img/a/AVvXsEjNUzUvD2qD_nf-ZmyWpj_NkzF9889CF64qy1S6WCqomSvAtGLgxgwSIJZ6TdcMkPqx7sorPULnu1ZPINxsM_raKHJHjPnc85ShGyVYJCnTAYUDeepAZArnwn7Y-yx2uhNvJQnglDVLeNnum3gOpz5RMFctyf_CBBo2Ins-k8KkPCEDoa5Nd7FW4WB5EtUN)

Ketiga, Kunjungi situs Mail yang didaftarkan akun [FreeDB.tech](https://freedb.tech/register.html) tadi untuk melakukan verifikasi akun.  
  
Empat, Buka isi kotak masuk yang pengirim dari [FreeDB.tech](https://freedb.tech/register.html)

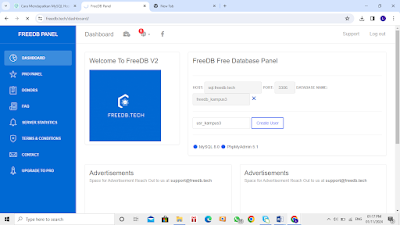
Lima, Klik link “Verify Your Account”.

Enam, Buatlah database baru dengan sesuai kebutuhan kamu, klik Create DB.

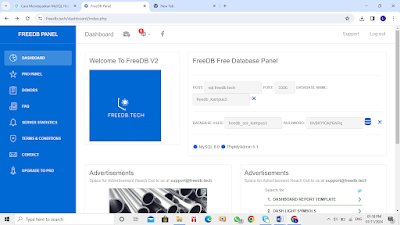
    Berikut hasil dalam pembuatan database baru

[](https://blogger.googleusercontent.com/img/a/AVvXsEhAe6bTf0syHj3CLaKxSWQqAvJ8ppUcRKowRGK1mMhDfBOaXz6Ck1zWcveVnGexeXxcXjiWL5uETsLrV9v-I_rfqssXvkP8tDOEan-QaUj5BLTCPkxwcnTHxZB75s19MqVvCNjbnWHUubmgE_OlZ-1VaST2JTH2x0AsRn60B1HPrKhC798gGyJk7qEwev1c)

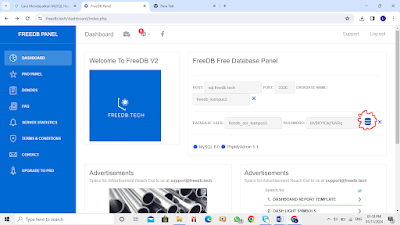
8. Buatlah nama pengguna baru dengan sesuai kebutuhan kamu, klik Create User.

[](https://blogger.googleusercontent.com/img/a/AVvXsEjNsR66tHpW9lrzbFrINW0-Vo4lORx3y6u77EHZz_P9TzrTM3I0SYXQPE8Lkvw07sNvL10CacY8sQzqjJpCyWT_FIP-zZnIOSF6zx42XKJaRCCcLpjJtKpS305Vm52rX4tX-QdF1ZPz79pmuhnWrzext3meuXxEXil234lPMLRevcmfm17HXTNIHVMyfT6R)

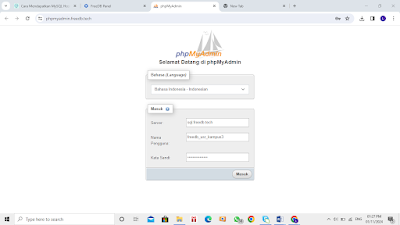
    Berikut hasil dalam pembuatan nama pengguna baru.



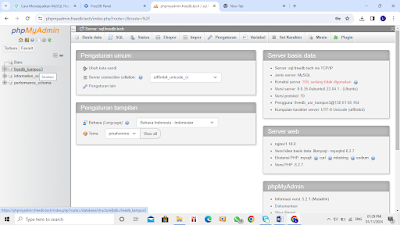
9. Apabila kamu ingin mengakses [PHPMyAdmin](https://phpmyadmin.freedb.tech/), bisa klik icon database yang terdapat pada gambar di bawah ini.



10. Masukkan pengisian data nama server (host), nama pengguna, beserta kata sandi yang didapatkan dari dashboard [FreeDB.tech](https://freedb.tech/register.html) tadi, klik Kirim.



11. Berikut hasilnya.



    Selesai.

    Kini, kamu bisa mendapatkan MySQL Hosting gratis dengan tujuan untuk edukasi atau belajar dalam melakukan koneksi database dan aplikasi secara remote seperti aplikasi laporan online (dalam bentuk simulasi atau devlopment).

**MODUL 9**

**MEMBUAT APLIKASI LOGIN BERBASIS WINDOWS-TKINTER,PYTHON**

Berikut adalah langkah-langkahnya:

Pertama seperti biasa aktifkan server offline terlebih dahulu, xampp.

Kedua buatlah file dengan nama file db.py dengan database ‘kampus\_baru’.

        self.host = 'localhost'

        self.port = 3306

        self.name = 'kampusbaru'

        self.user = 'root'

        self.password = ''

        self.conn = None

        self.cursor = None

        self.result = None

        self.connected = False

        self.affected = 0

        self.connect()

Kemudian buatlah file Users.py dan ketikan source code berikut ini:

import bcrypt

from db import DBConnection as mydb

class Users:

    def \_\_init\_\_(self):

        self.\_\_id= None

        self.\_\_email= None

        self.\_\_nama= None

        self.\_\_password= None

        self.\_\_level= None

        self.\_\_uservalid = None

        self.\_\_passwordvalid = None

        self.\_\_loginvalid = None#

        self.conn = None

        self.affected = None

        self.result = None

    @property

    def id(self):

        return self.\_\_id

    @property

    def email(self):

        return self.\_\_email

    @email.setter

    def email(self, value):

        self.\_\_email = value

    @property

    def nama(self):

        return self.\_\_nama

    @nama.setter

    def nama(self, value):

        self.\_\_nama = value

    @property

    def password(self):

        return self.\_\_password

    @password.setter

    def password(self, value):

        self.\_\_password = value

    @property

    def level(self):

        return self.\_\_level

    @level.setter

    def level(self, value):

        self.\_\_level = value

    @property

    def loginvalid(self):

        return self.\_\_loginvalid

    @loginvalid.setter

    def loginvalid(self, value):

        self.\_\_loginvalid = value

    def cekUsername(self, email):

        self.conn = mydb()

        sql="SELECT \* FROM users WHERE email='" + email + "'"

        self.result = self.conn.findOne(sql)

        if(self.result!=None):

            self.\_\_email = self.result[1]

            self.\_\_nama = self.result[2]

            self.\_\_password = self.result[3]

            self.\_\_level = self.result[4]

            self.affected = self.conn.cursor.rowcount

            self.\_\_uservalid = True

        else:

            self.\_\_email = ''

            self.\_\_nama = ''

            self.\_\_password = ''

            self.\_\_level = ''

            self.affected = 0

            self.\_\_uservalid = False

        return self.\_\_uservalid

    def cekPassword(self, password):

        hashedpass=self.\_\_password.encode('utf-8')

        c = password.encode('utf-8')

        d = bcrypt.checkpw(c, hashedpass)

        if(d):

            self.\_\_passwordvalid=True

        else:

            self.\_\_passwordvalid=False

        return self.\_\_passwordvalid

    def Validasi(self, email, password):

        a = self.cekUsername(email)

        if(a==True):

            b = self.cekPassword(password)

            if(b==True):

                self.\_\_loginvalid=True

            else:

                self.\_\_loginvalid=False

        else:

            self.\_\_loginvalid=False

        val = []

        val = [self.\_\_level, self.\_\_loginvalid]

        return val

A = Users()

print("\n\n")

print("Username BENAR, dan password BENAR")

B = A.Validasi('nurjati@umc.ac.id','123')

print(B)

print("\n\nUsername BENAR, dan password SALAH")

C = A.Validasi('nurjati@umc.ac.id','1234')

print(C)

print("\n\nUsername SALAH, dan password BENAR")

D = A.Validasi('baim@umc.ac.id','123')

print(D)

print("\n\nUsername SALAH, dan password SALAH")

E = A.Validasi('baim@umc.ac.id','1234')

print(E)

print("\n\n")

Selanjutnya yaitu buatlah file dengan nama file Dashboard.py, ketikan source code berikut ini:

import tkinter as tk

from tkinter import Menu, messagebox

from FrmLogin import \*

from FrmAdmin import \*

from FrmMahasiswa import \*

from FrmDosen import \*

class Dashboard:

    def \_\_init\_\_(self):

        # root window

        self.root = tk.Tk()

        self.root.title('Menu Demo')

        # self.root.attributes('-fullscreen', True)

        self.root.geometry("900x400")

        self.\_\_data = None

        self.\_\_level = None

        # create a menubar

        self.menubar = Menu(self.root)

        self.root.config(menu=self.menubar)

        # create menus

        self.file\_menu = Menu(self.menubar)

        self.admin\_menu = Menu(self.menubar)

        self.mahasiswa\_menu = Menu(self.menubar)

        self.dosen\_menu = Menu(self.menubar)

        # add menu items to File menu

        self.file\_menu.add\_command(label='Login', command=lambda: self.new\_window("Log Me In", FormLogin))

        self.file\_menu.add\_command(label='Exit', command=self.root.destroy)

        # add menu items to menu Admin

        self.admin\_menu.add\_command(label='Data Admin', command=lambda: self.new\_window("Data Admin", FrmAdmin))

        self.admin\_menu.add\_command(label='Data Mahasiswa', command=lambda: self.new\_window("Data Mahasiswa", FrmMahasiswa))

        self.admin\_menu.add\_command(label='Data Dosen', command=lambda: self.new\_window("Data Dosen", FrmDosen))

        # add menu items to menu Mahasiswa

        self.mahasiswa\_menu.add\_command(label='Data Admin', command=lambda: self.new\_window("Data Admin", FrmAdmin))

        self.mahasiswa\_menu.add\_command(label='Data Mahasiswa', command=lambda: self.new\_window("Data Mahasiswa", FrmMahasiswa))

        self.mahasiswa\_menu.add\_command(label='Data Dosen', command=lambda: self.new\_window("Data Dosen", FrmDosen))

        # add menu items to menu Dosen

        self.dosen\_menu.add\_command(label='Data Admin', command=lambda: self.new\_window("Data Admin", FrmAdmin))

        self.dosen\_menu.add\_command(label='Data Mahasiswa', command=lambda: self.new\_window("Data Mahasiswa", FrmMahasiswa))

        self.dosen\_menu.add\_command(label='Data Dosen', command=lambda: self.new\_window("Data Dosen", FrmDosen))

        # add menus to the menubar

        self.menubar.add\_cascade(label="File", menu=self.file\_menu)

    def new\_window(self, number, \_class):

        new = tk.Toplevel(self.root)

        new.transient()

        new.grab\_set()

        \_class(new, number, self.update\_main\_window)

    def update\_main\_window(self, data):

        # Method to receive data from child windows

        self.\_\_data = data

        level = self.\_\_data[0]

        loginvalid = self.\_\_data[1]

        if(loginvalid==True):

            index = self.file\_menu.index('Login')

            # hapus menu login

            self.file\_menu.delete(index)

            self.file\_menu.add\_command(label='Logout', command=self.Logout)

            # tambahkan menu sesuai level

            if(level=='admin'):

                self.menubar.add\_cascade(label="Admin", menu=self.admin\_menu)

                self.\_\_level = 'Admin'

            elif(level=='mahasiswa'):

                self.menubar.add\_cascade(label="Mahasiswa", menu=self.mahasiswa\_menu)

                self.\_\_level = 'Mahasiswa'

            elif(level=='dosen'):

                self.menubar.add\_cascade(label="Dosen", menu=self.dosen\_menu)

                self.\_\_level = 'Dosen'

            else:

                pass

    def Logout(self):

        index = self.file\_menu.index('Logout')

        self.file\_menu.delete(index)

        self.file\_menu.add\_command(label='Login', command=lambda: self.new\_window("Log Me In", FormLogin))

        self.remove\_all\_menus()

    def remove\_all\_menus(self):

        index = self.menubar.index(self.\_\_level)

        if index is not None:

            self.menubar.delete(index)

    def run(self):

        self.root.mainloop()

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

    menu\_app = Dashboard()

    menu\_app.run()

Tahapan selanjutnya buatlah file dengan nama file FrmLogin.py. Yang dimana disini merupakan bagian penting untuk bisa mengakses sebuah program sebagai keamanan.

import tkinter as tk

from tkinter import Frame,Label,Entry,Button,Radiobutton,ttk,VERTICAL,YES,BOTH,END,Tk,W,StringVar,messagebox

from tkinter import ttk

from Users import \*

class FormLogin:

    def \_\_init\_\_(self, parent, title, update\_main\_window):

        self.parent = parent

        self.update\_main\_window = update\_main\_window

        #self.parent.geometry("250x150")

        self.parent.title(title)

        self.parent.protocol("WM\_DELETE\_WINDOW", self.onKeluar)

        self.aturKomponen()

    def aturKomponen(self):

        mainFrame = Frame(self.parent, bd=10)

        mainFrame.pack(fill=BOTH, expand=YES)

        # diatur agar tampil di tengah layar

        screen\_width = mainFrame.winfo\_screenwidth()

        screen\_height = mainFrame.winfo\_screenheight()

        x = (screen\_width - 250) // 2

        y = (screen\_height - 150) // 2

        width = 250

        height = 150

        self.parent.geometry(f"{width}x{height}+{x}+{y}")

        # Label

        Label(mainFrame, text='Email:').grid(row=0, column=0, sticky=W, padx=5, pady=5)

        self.txtEmail = Entry(mainFrame)

        self.txtEmail.grid(row=0, column=1, padx=5, pady=5)

        Label(mainFrame, text='Password:').grid(row=1, column=0, sticky=W, padx=5, pady=5)

        self.txtPassword = Entry(mainFrame)

        self.txtPassword.grid(row=1, column=1, padx=5, pady=5)

        # Button

        self.btnSubmit = Button(mainFrame, text='Submit', command=self.onSubmit, width=10)

        self.btnSubmit.grid(row=2, column=0, padx=5, pady=5)

        self.btnCancel = Button(mainFrame, text='Cancel', command=self.onKeluar, width=10)

        self.btnCancel.grid(row=2, column=1, padx=5, pady=5)

    def onSubmit(self, event=None):

        email = self.txtEmail.get()

        password = self.txtPassword.get()

        obj = Users()

        val = obj.Validasi(email,password)

        C = val[1]

        if(C==True):

            self.update\_main\_window(val)

            self.parent.destroy()

            messagebox.showinfo("showinfo", "Data Berhasil ")

        else:

            messagebox.showwarning("showwarning", "Login Gagal ")

        return val

    def onKeluar(self, event=None):

        # memberikan perintah menutup aplikasi

        self.parent.destroy()

if \_\_name\_\_ == '\_\_main\_\_':

    def update\_main\_window(result):

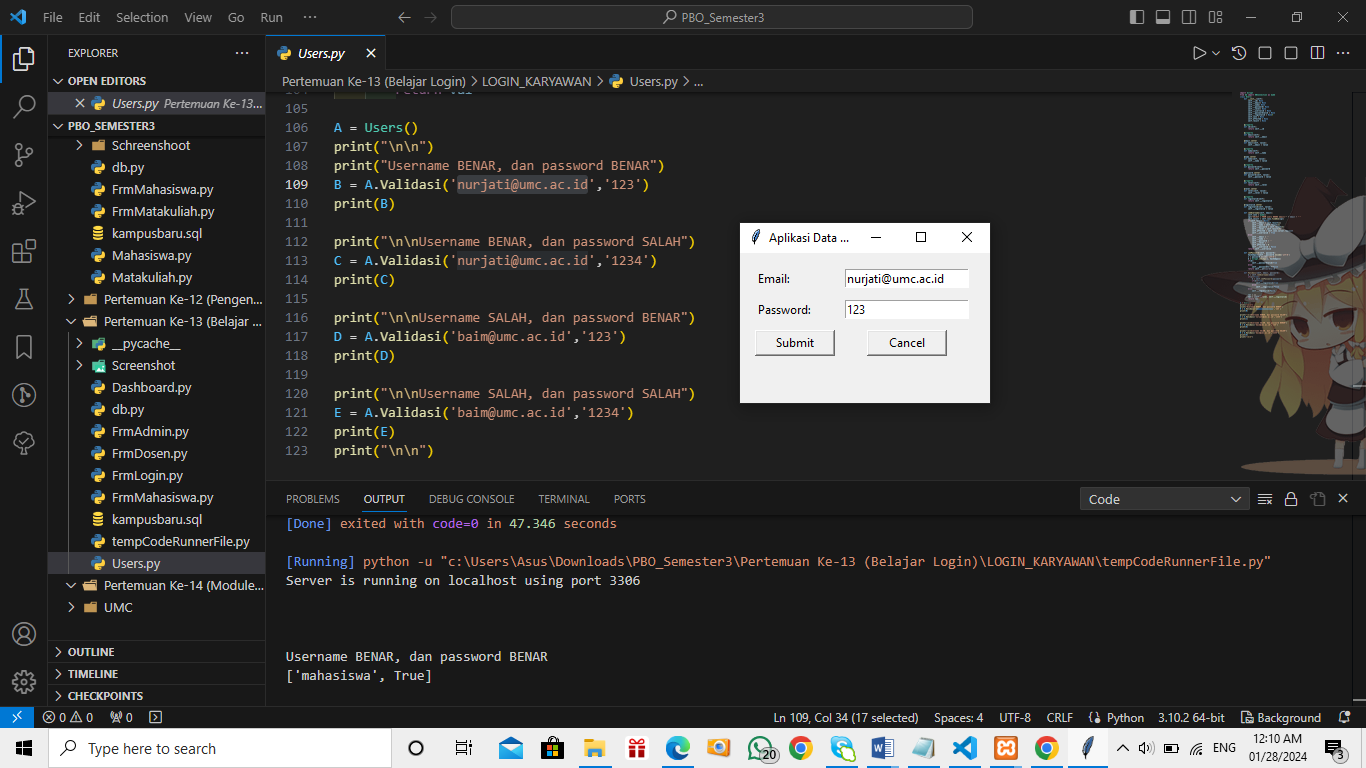
        print(result)

    root = tk.Tk()

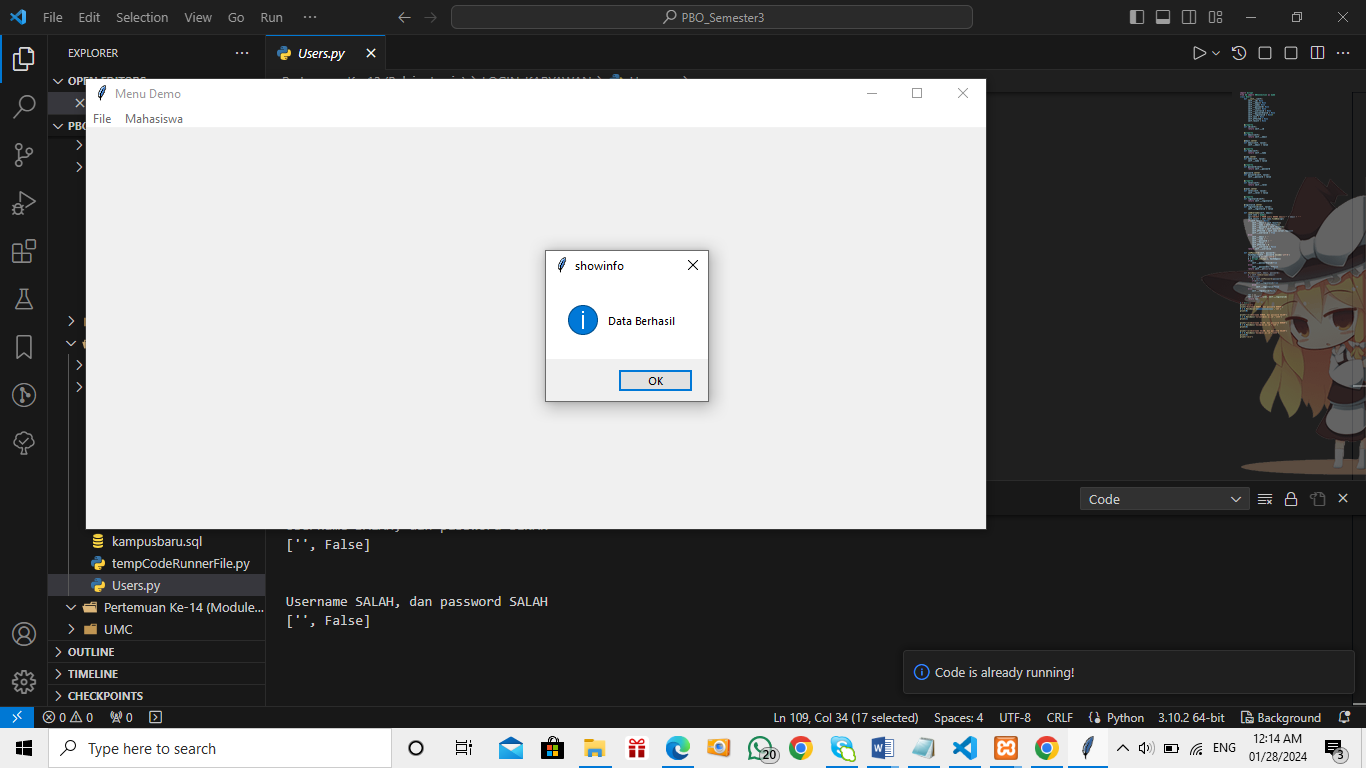
    aplikasi = FormLogin(root, "Aplikasi Data Login",update\_main\_window)

    root.mainloop()

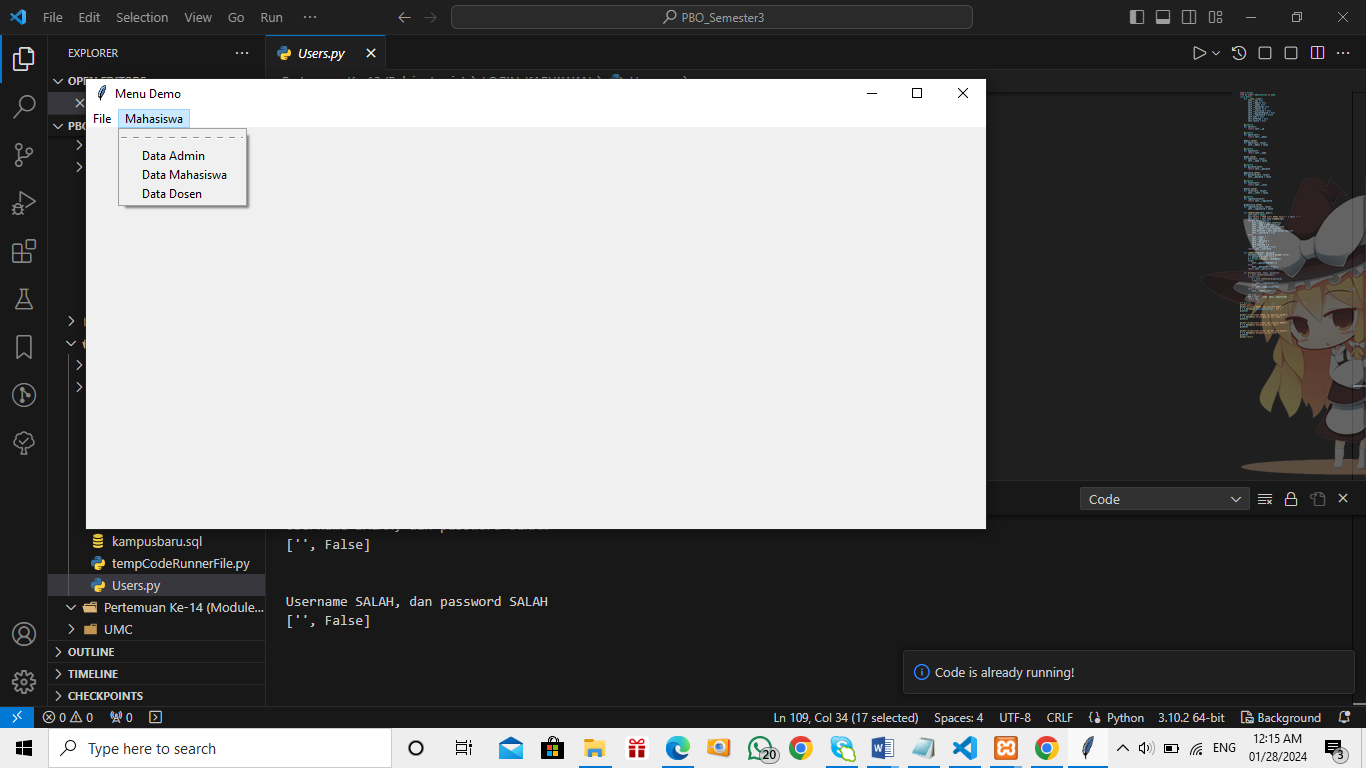
Berikut tampilan saat login username dan password.



Dan buatlah file FrmMatakuliah dan FrmMahasiswa, beginilah ketika di Run akan menghasilkan tampilan seperti ini:



Berikut ini merupakan tampilan dari Dashboard ketika kita sudah memasukan username dan password pada saat Login.



Selesai.

**Kesimpulan**

Dari pembahasan pada setiap halamn dalam modul ini, penulis dapat menarik kesimpulan bahwa :

1). Bahasa pemograman Python merupakan bahasa pemograman yang cukup mudah digunakan bagi seorang programmer dalam merancang sebuah program. Bahasa pemograman ini juga dapat berjalan pada berbagai sistem operasi seperti sistem operasi windows, Linux, dan Mac OS.

2). Program ini sescara visual cukup memudahkan pengguna dalam hal pengoprasian.

3). Program ini cukup valid terhadap perhitungan manual, sehingga dapat digunakan sebagai alat bantu perancangan.

4). Hasil output program ini mencakup proses perhitungan dan gambar hasil perhitungan, yang dapat disimpan oleh pengguna apabila sewaktu-waktu akan direvisi kembali dan dapat dicetak.