# **KATA PENGANTAR**

Modul pembelajaran Python ini disusun dengan tujuan memberikan pemahaman yang komperhensif tentang bahasa pemrograman Python. Python menjadi pilihan utama bagi banyak pengembang perangkat lunak karena kemudahan sintaksisnya, fleksibilitas dan dukungan komunitas yang luas. Dalam modul ini, kita akan menjelajahi berbagai konsep dasar hingga tingkat lanjutan, membantu Anda membangun fondasi yang kuat dalam pengembangan perangkat lunak menggunakan Python.

Modul ini dirancang untuk memenuhi tugas mata kuliah Pemrograman Berorientasi Objek pada semester 3 Teknik Informatika di Universitas Muhammadiyah Cirebon, mata kuliah ini diampu oleh Bapak Freddy Wicaksono, M. Kom. Penulis percaya bahwa pengalaman langsung dalam menulis kode akan mempercepat proses pembelajaran, dan itulah mengapa modul ini dirancang sebaik mungkin untuk memudahkan pembaca dalam memahaminya

Penulis berharap modul ini tidak hanya menjadi panduan yang membantu pembaca memahami Python, tetapi juga menjadi sumber daya yang menginspirasi untuk menciptakan solusi kreatif dan efisien. Dengan pemahaman yang mendalam tentang Python, Anda akan dapat mengembangkan aplikasi, melakukan analisis data, dan mengotomatiskan tugas-tugas yang kompleks. Selamat belajar, dan semoga modul ini membawa pembaca ke tingkat keahlian baru dalam dunia pemrograman dengan Python.

Cirebon, 26 Januari 2024

Penulis

# **DAFTAR ISI**

[**KATA PENGANTAR** ii](#_Toc157355278)

[**DAFTAR ISI** iii](#_Toc157355279)

[**BAB I** 1](#_Toc157355280)

[**LUAS DAN KELILING BANGUN DATAR** 1](#_Toc157355281)

[**1.1** **Dasar Teori** 1](#_Toc157355282)

[**1.2** **Rumus** 1](#_Toc157355283)

[**1.3** **Contoh Implementasi Rumus Bangun Datar ke Dalam Python** 2](#_Toc157355284)

[**1.4** **Contoh Latihan Bangun Datar** 4](#_Toc157355285)

[**BAB II** 5](#_Toc157355286)

[**KONDISI, LOOP, NUMBER, STRING** 5](#_Toc157355287)

[**2.1** **Teori Dasar Kondisi** 5](#_Toc157355288)

[**a)** **If** 5](#_Toc157355289)

[**b)** **If Else** 5](#_Toc157355290)

[**c)** **Elif** 6](#_Toc157355291)

[**2.2** **Contoh Rumus Kondisi** 6](#_Toc157355292)

[**2.3** **Contoh Latihan Kondisi** 6](#_Toc157355293)

[**2.4** **Loop** 7](#_Toc157355294)

[**a)** **While Loop** 7](#_Toc157355295)

[**b)** **For Loop** 7](#_Toc157355296)

[**c)** **Nested Loop** 7](#_Toc157355297)

[**2.5** **Contoh Rumus Loop** 8](#_Toc157355298)

[**2.6** **Contoh Latihan Loop** 8](#_Toc157355299)

[**2.7** **Number** 9](#_Toc157355300)

[**2.8** **Contoh Latihan Number** 10](#_Toc157355301)

[**2.9** **String** 10](#_Toc157355302)

[**2.10** **Contoh Latihan** **String** 12](#_Toc157355303)

[**BAB III** 13](#_Toc157355304)

[**3.1**  **Teori Dasar** 13](#_Toc157355305)

[**3.2**  **Rumus** 13](#_Toc157355306)

[**3.3** **Contoh Implementasi Rumus Bangun Ruang ke Dalam Python** 14](#_Toc157355307)

[**3.4**  **Contoh Latihan Bangun Ruang ke Dalam Python** 16](#_Toc157355308)

[**BAB IV** 17](#_Toc157355309)

[**JADWAL MATAKULIAH** 17](#_Toc157355310)

[**4.1** **Teori Dasar** 17](#_Toc157355311)

[**4.2** **Contoh Latihan Jadwal Matakuliah ke Dalam Python** 18](#_Toc157355312)

[**Output** 19](#_Toc157355313)

[**BAB V** 20](#_Toc157355314)

[**TRANSLATOR** 20](#_Toc157355315)

[**5.1** **Dasar Teori** 20](#_Toc157355316)

[**5.3** **Contoh Latihan Translate Bahasa Jepang ke Dalam Python** 22](#_Toc157355317)

[**BAB VI** 24](#_Toc157355318)

[**OBJECT ORIENTED PROGRAMMING (OOP)** 24](#_Toc157355319)

[**6.1** **Dasar Teori** 24](#_Toc157355320)

[**6.2** **Contoh Program Berbasis OOP Murni** 24](#_Toc157355321)

[**BAB VII** 26](#_Toc157355322)

[**DATABSE MYSQL (CMD)** 26](#_Toc157355323)

**7.1 Teori Dasar** ……………………………………………………………………………………………….. 26

[**7.2** **Installasi XAMPP** 27](#_Toc157355324)

[**BAB VIII** 30](#_Toc157355325)

[**DATABSE PYTHON CONNECTION** 30](#_Toc157355326)

[**8.1** **Teori Dasar** 30](#_Toc157355327)

[**8.2** **Implementasi** 30](#_Toc157355328)

[**BAB IX** 35](#_Toc157355333)

[**LOGIN** 35](#_Toc157355334)

[**9.1** **Dasar Teori** 35](#_Toc157355335)

[**9.2** **Implementasi Program** 36](#_Toc157355336)

[**DAFTAR PUSTAKA** 40](#_Toc157355337)

# **BAB I**

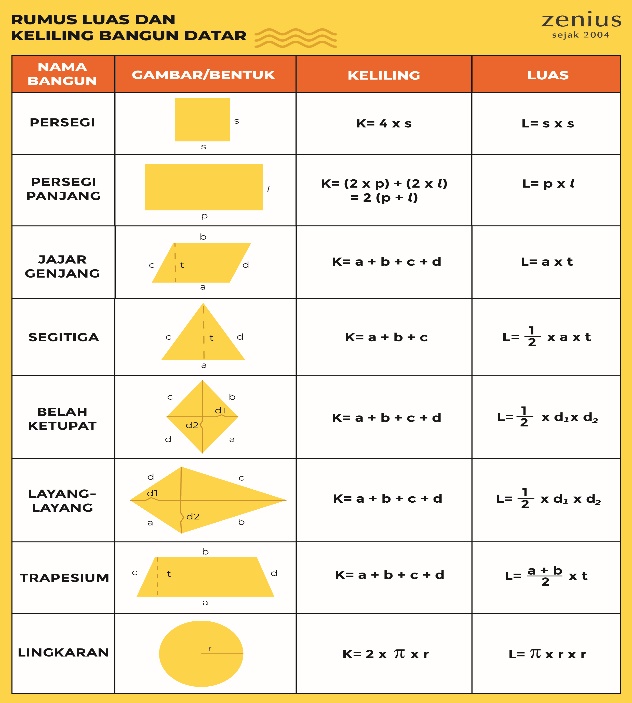
# **LUAS DAN KELILING BANGUN DATAR**

## **Dasar Teori**

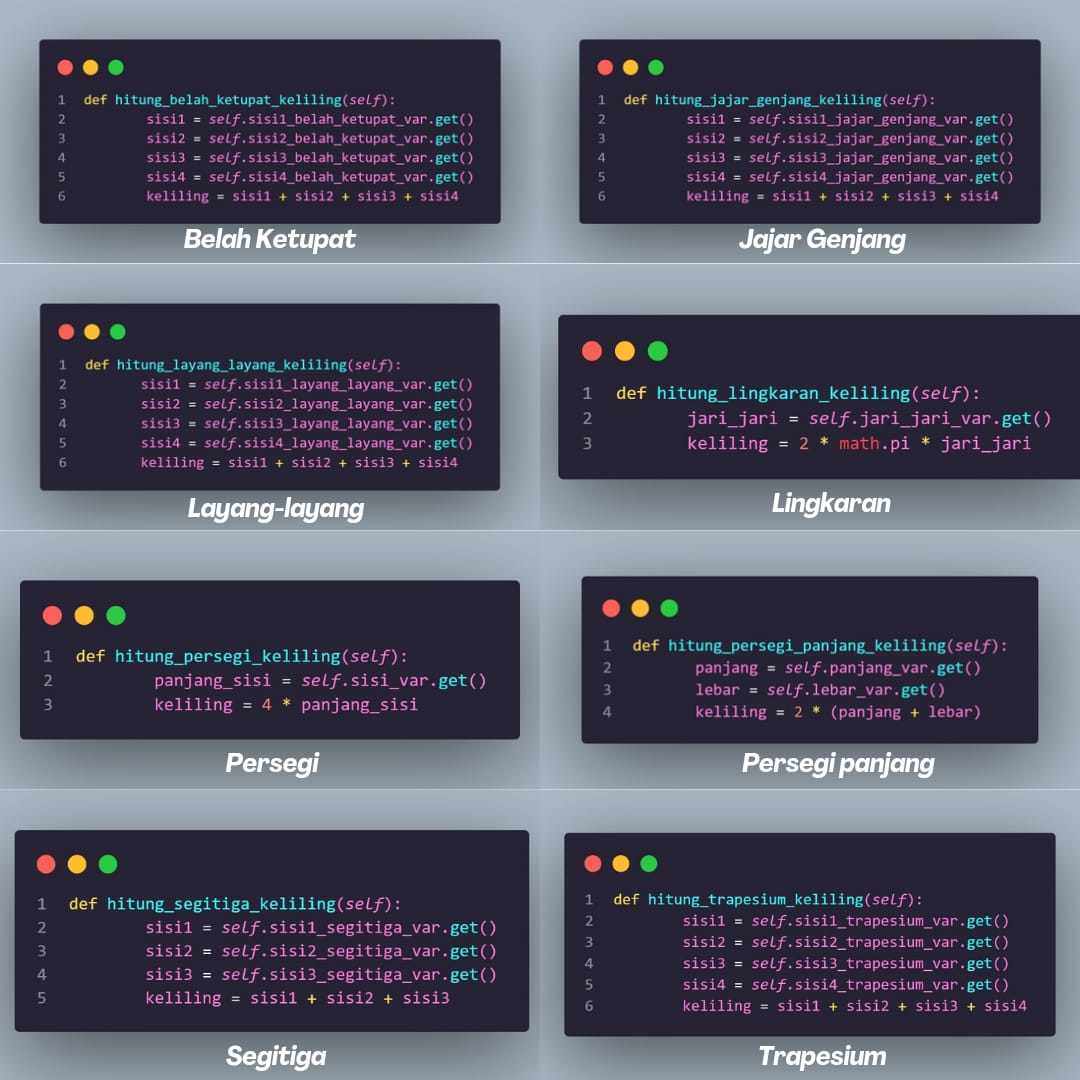
Bangun datar adalah objek geometris dua dimensi yang dibentuk oleh titik-titik yang terletak pada bidang datar. Mereka memiliki sisi-sisi yang menyusun batas-batasnya, serta sudut-sudut yang terbentuk antara sisi-sisi tersebut. Setiap jenis bangun datar memiliki karakteristik khusus yang membedakannya.

Sebagai contoh, persegi adalah bangun datar dengan keempat sisi yang sama panjang dan keempat sudut yang sama besar (sudut siku-siku 90 derajat). Sedangkan segitiga memiliki tiga sisi dan tiga sudut. Perbedaan karakteristik inilah yang membedakan satu bangun datar dari yang lain.

## **1.2 Rumus**



## **1.3 Contoh Implementasi Rumus Bangun Datar ke Dalam Python**

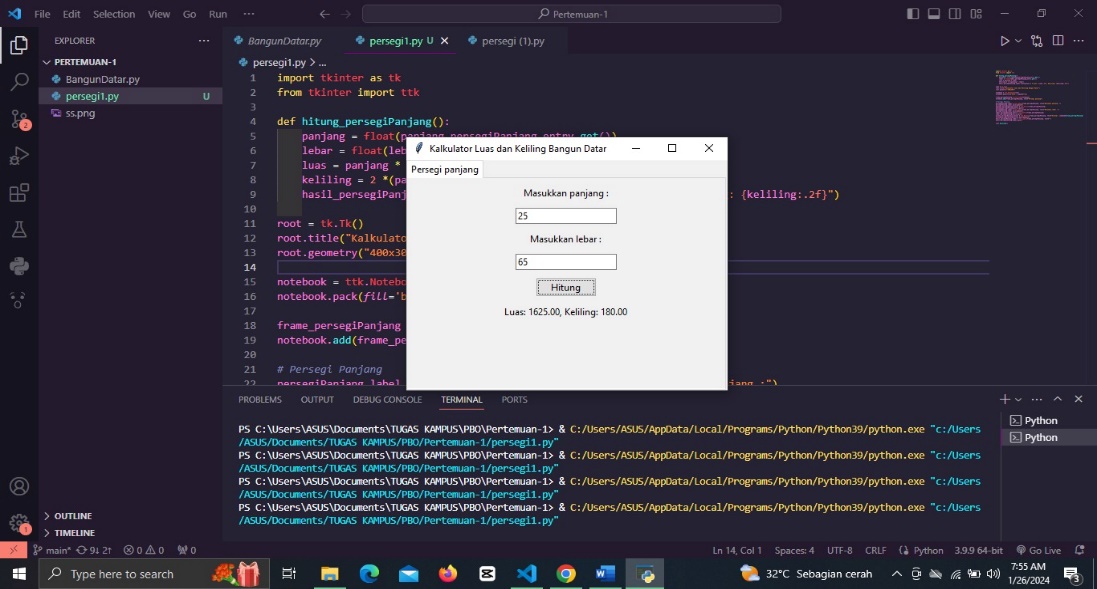
 **Keliling**

**Luas**



## **1.4 Contoh Latihan Bangun Datar**



 **Output**

# **BAB II**

# **KONDISI, LOOP, NUMBER, STRING**

## **Teori Dasar Kondisi**

### **If**

Pengambilan keputusan (kondisi if) digunakan untuk mengantisipasi kondisi yang terjadi saat jalanya program dan menentukan tindakan apa yang akan diambil sesuai dengan kondisi.

Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar True. Jika kondisi bernilai salah False maka statement/kondisi if tidak akan di-eksekusi.

### **If Else**

Pengambilan keputusan (kondisi if else) tidak hanya digunakan untuk menentukan tindakan apa yang akan diambil sesuai dengan kondisi, tetapi juga digunakan untuk menentukan tindakan apa yang akan diambil/dijalankan jika kondisi tidak sesuai.

Pada python ada beberapa statement/kondisi diantaranya adalah if, else dan elif Kondisi if digunakan untuk mengeksekusi kode jika kondisi bernilai benar.

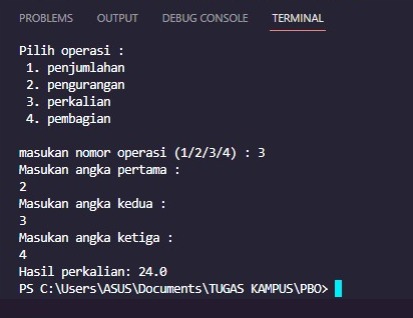
Kondisi if else adalah kondisi dimana jika pernyataan benar True maka kode dalam if akan dieksekusi, tetapi jika bernilai salah False maka akan mengeksekusi kode di dalam else.

### **Elif**

Pengambilan keputusan (kondisi if elif) merupakan lanjutan/percabangan logika dari "kondisi if". Dengan elif kita bisa membuat kode program yang akan menyeleksi beberapa kemungkinan yang bisa terjadi. Hampir sama dengan kondisi "else", bedanya kondisi "elif" bisa banyak dan tidak hanya satu.

## **Contoh Rumus Kondisi**

## **Contoh Latihan Kondisi**

****

## **Loop**

Secara umum, pernyataan pada bahasa pemrograman akan dieksekusi secara berurutan. Pernyataan pertama dalam sebuah fungsi dijalankan pertama, diikuti oleh yang kedua, dan seterusnya. Tetapi akan ada situasi dimana Anda harus menulis banyak kode, dimana kode tersebut sangat banyak. Jika dilakukan secara manual maka Anda hanya akan membuang-buang tenaga dengan menulis beratus-ratus bahkan beribu-ribu kode. Untuk itu Anda perlu menggunakan pengulangan di dalam bahasa pemrograman Python.

Di dalam bahasa pemrograman Python pengulangan dibagi menjadi 3 bagian, yaitu :

### **While Loop**

Pengulangan While Loop di dalam bahasa pemrograman Python dieksesusi statement berkali-kali selama kondisi bernilai benar atau True.

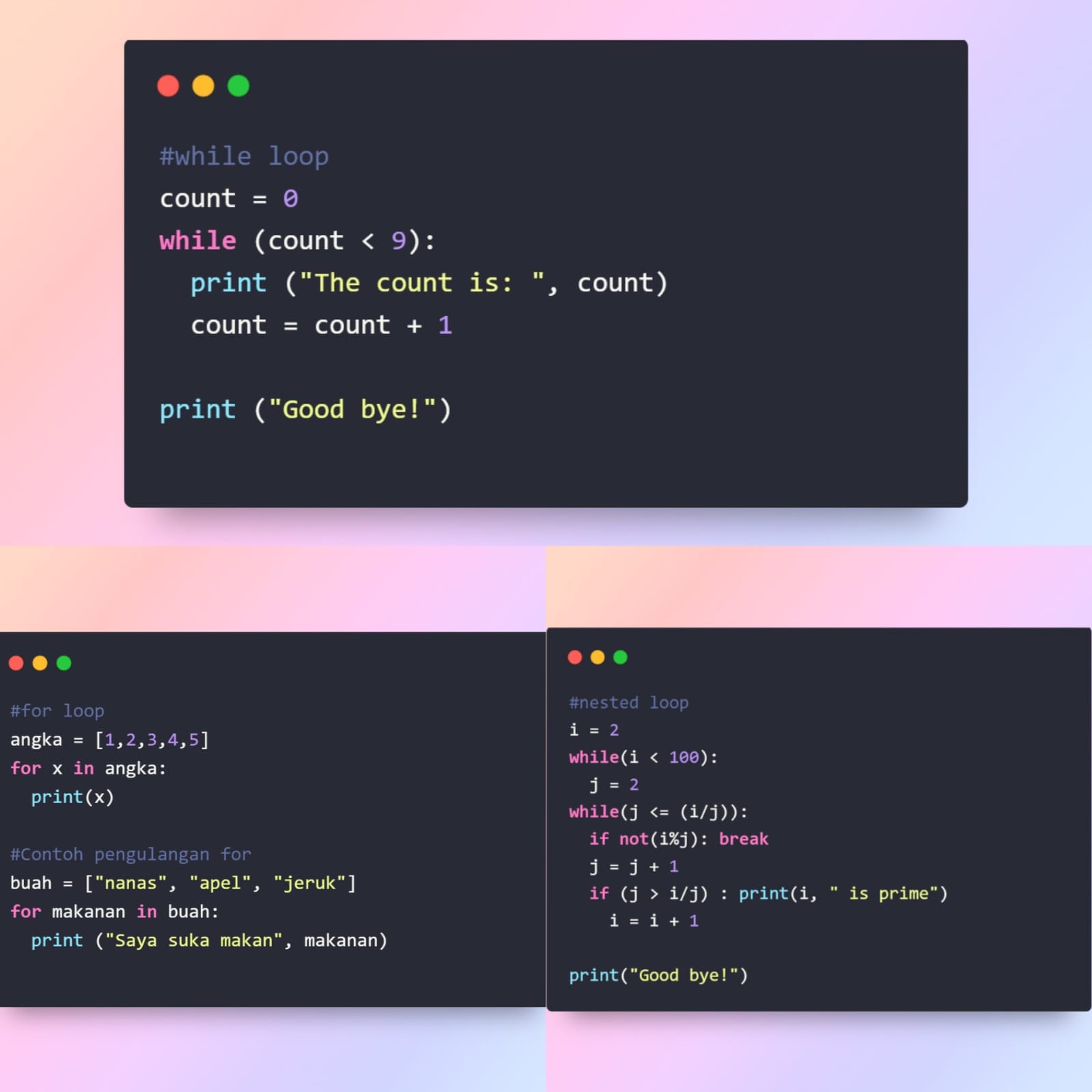
### **For Loop**

Pengulangan for pada Python memiliki kemampuan untuk mengulangi item dari urutan apapun, seperti list atau string.

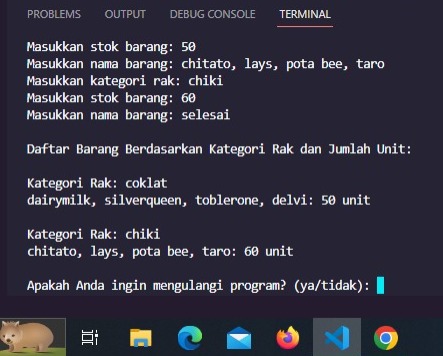
### **Nested Loop**

Bahasa pemrograman Python memungkinkan penggunaan satu lingkaran di dalam loop lain. Bagian berikut menunjukkan beberapa contoh untuk menggambarkan konsep tersebut.

## **Contoh Rumus Loop**



## **Contoh Latihan Loop**



## **Number**

Number adalah tipe data Python yang menyimpan nilai numerik. Number adalah tipe data yang tidak berubah. Ini berarti, mengubah nilai dari sejumlah tipe data akan menghasilkan objek yang baru dialokasikan.

Konversi Tipe Data Number Python

Pada Python Anda bisa mengkonversi tipe data dengan menggunakan fungsi. Dibawah ini adalah beberapa fungsi untuk mengkonversi tipe data number Python.

int(x) untuk meng-konversi x menjadi plain integer.

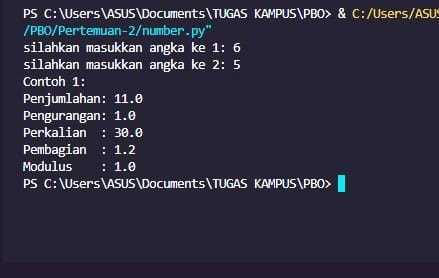
long(x) untuk meng-konversi x menjadi long integer.

float(x) untuk meng-konversi x menjadi floating point number.

complex(x) untuk meng-konversi x menjadi complex number dengna real part x dan imaginary part zero.

complex(x, y) untuk meng-konversi x dan y menjadi complex number dengan real part x dan imaginary part y. x dan numeric expressions y.

## **Contoh Latihan Number**



## **String**

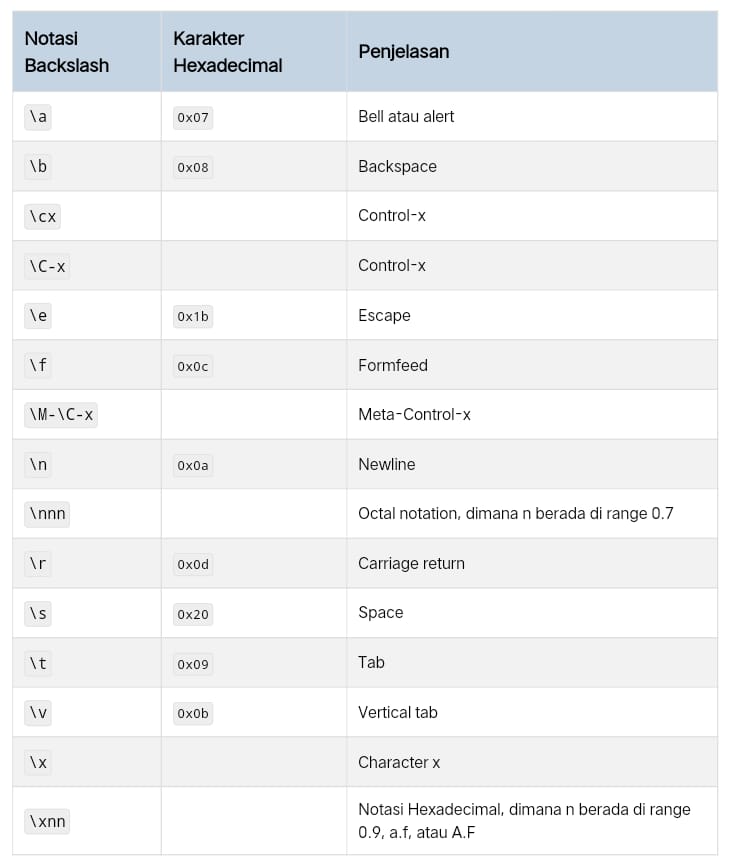
String adalah jenis yang paling populer di bahasa pemrograman. Kita bisa membuatnya hanya dengan melampirkan karakter dalam tanda kutip. Python memperlakukan tanda kutip tunggal sama dengan tanda kutip ganda. Membuat string semudah memberi nilai pada sebuah variabel.

Mengakses Nilai dalam String:

Python tidak menggunakan tipe karakter titik koma ; Ini diperlakukan sebagai string dengan panjang satu, sehingga juga dianggap sebagai substring.

Mengupdate String:

Anda dapat "memperbarui" string yang ada dengan (kembali) menugaskan variabel ke string lain. Nilai baru dapat dikaitkan dengan nilai sebelumnya atau ke string yang sama sekali berbeda sama sekali. Sebagai contoh:



Operator Format String Python

Salah satu fitur Python yang paling keren adalah format string operator %. Operator ini unik untuk string dan membuat paket memiliki fungsi dari keluarga printf C () C. berikut adalah contoh sederhananya : print ("My name is %s and weight is %d kg!" % ('Zara', 21)).

## **Contoh Latihan String**





# **BAB III**

**BANGUN RUANG**

## **3.1 Teori Dasar**

Bangun ruang adalah objek tiga dimensi yang memiliki panjang, lebar, dan tinggi. Mereka memiliki volume dan luas permukaan yang dapat diukur. Contoh bangun ruang melibatkan bentuk seperti kubus, balok, bola, tabung, kerucut, dan prisma. Setiap bangun ruang memiliki karakteristik unik yang dapat diidentifikasi berdasarkan sifat-sifatnya. Misalnya, kubus memiliki semua sisi dengan panjang yang sama, sementara tabung memiliki dua tutup lingkaran dan permukaan melengkung di sekitarnya.

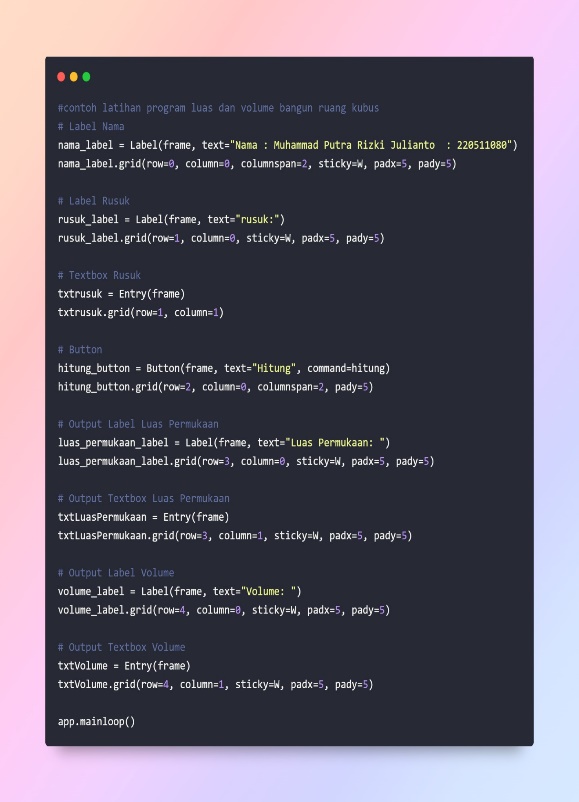
## **3.2 Rumus**

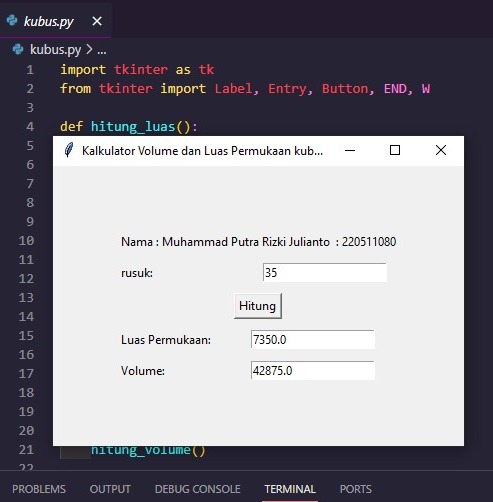
## **3.3 Contoh Implementasi Rumus Bangun Ruang ke Dalam Python**

 **Luas**

**Volume**

## **3.4 Contoh Latihan Bangun Ruang ke Dalam Python**



 **Output**

# **BAB IV**

# **JADWAL MATAKULIAH**

## **4.1 Dasar Teori**

Aplikasi ini adalah sistem jadwal mata kuliah sederhana berbasis Windows yang memungkinkan pengguna untuk menambahkan, mengedit, dan melihat jadwal mata kuliah mereka. Aplikasi ini menggunakan antarmuka grafis dengan bantuan modul Tkinter untuk memudahkan penggunaan.

Fitur utama:

1. Tambahkan Mata Kuliah: Pengguna dapat menambahkan mata kuliah beserta detailnya seperti nama mata kuliah, waktu, dan tempat.

2. Edit Jadwal: Aplikasi memungkinkan pengguna untuk mengedit informasi mata kuliah yang sudah ada.

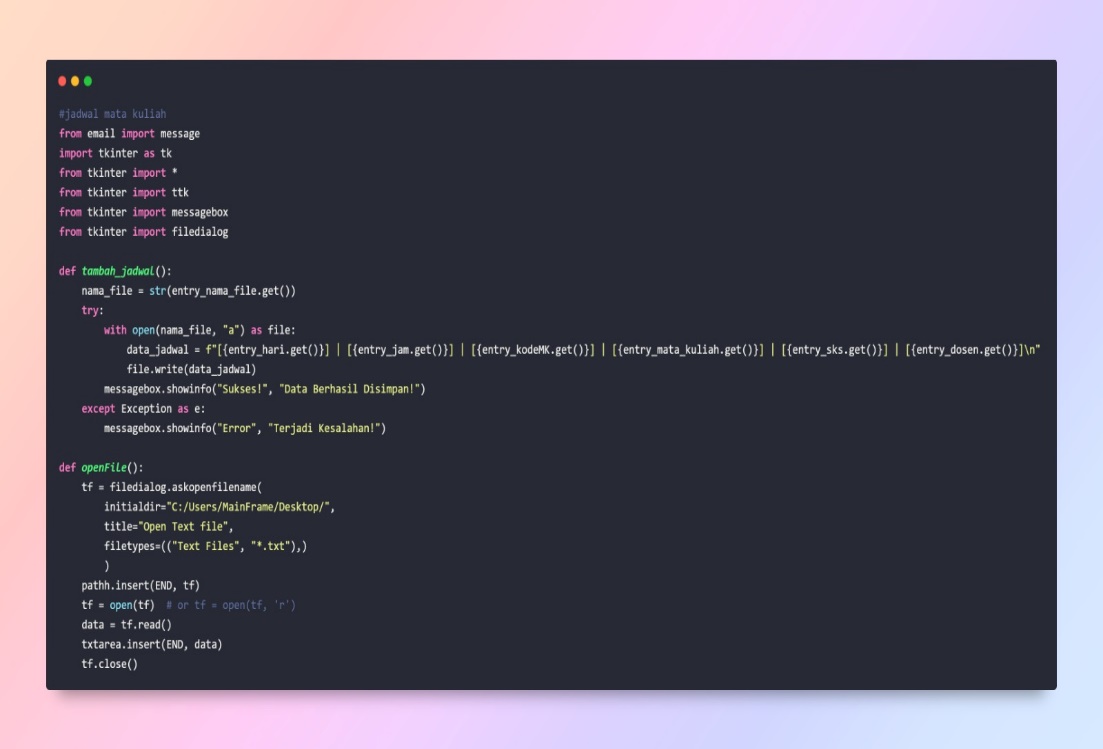
3. Tampilkan Jadwal: Pengguna dapat melihat jadwal mata kuliah mereka yang tersimpan.

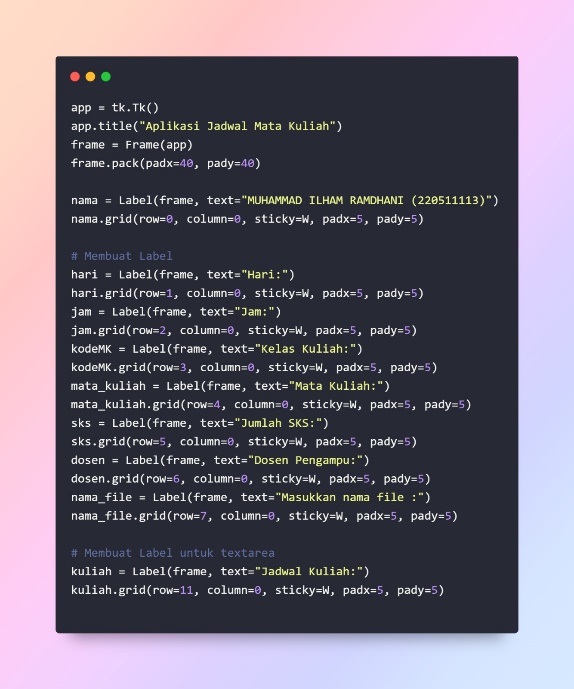
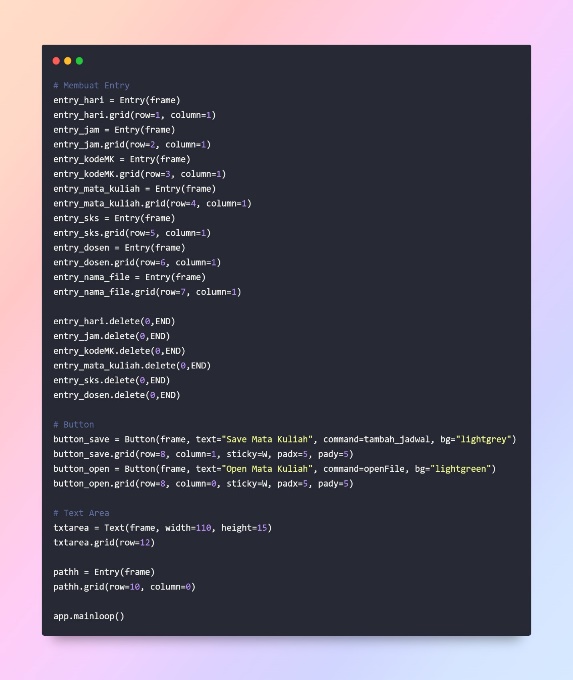
4. Simpan dan Muat: Data jadwal disimpan dalam file teks, memungkinkan pengguna untuk menyimpan jadwal mereka dan memuatnya kembali nanti.

Metode input/output file memastikan bahwa data jadwal dapat dipertahankan antar-sesi penggunaan aplikasi. Pengguna dapat menyimpan jadwal mereka dalam file dan memuatnya kembali ketika diperlukan.

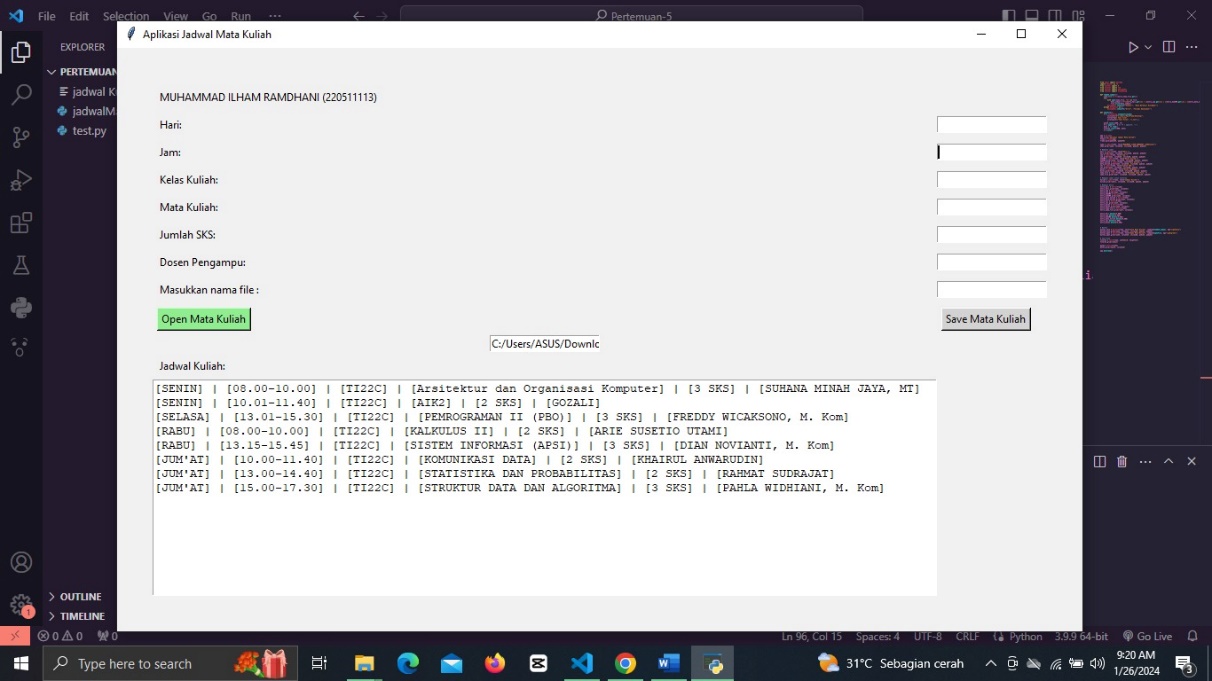
Selain itu, aplikasi ini memiliki antarmuka yang ramah pengguna dengan fungsi navigasi yang mudah dipahami. Dengan menggunakan Python dan Tkinter, aplikasi ini dapat dijalankan di lingkungan Windows tanpa memerlukan pengaturan khusus.

## **4.2 Contoh Latihan Jadwal Matakuliah ke Dalam Python**





# **Output**



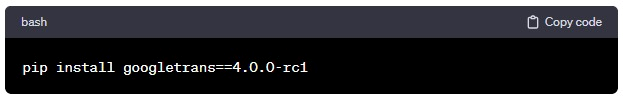
# **BAB V**

# **TRANSLATOR**

## **5.1 Dasar Teori**

Aplikasi Translator ini dirancang untuk memberikan kemudahan dalam menerjemahkan teks antar bahasa secara cepat dan efisien. Dengan antarmuka berbasis Windows yang intuitif, pengguna dapat dengan mudah memasukkan teks, memilih bahasa sumber, dan bahasa tujuan untuk proses terjemahan.

Aplikasi ini memanfaatkan pustaka penerjemahan bahasa yang didukung oleh Python, memastikan akurasi dan kelancaran dalam hasil terjemahan. Selain itu, pengguna dapat menyimpan riwayat terjemahan atau mengonversi teks terjemahan menjadi berkas teks untuk referensi lebih lanjut. Dengan dukungan berbagai bahasa dan tampilan yang bersih, Aplikasi Translator ini memberikan solusi yang user-friendly untuk kebutuhan penerjemahan sehari-hari di lingkungan Windows.

Pastikan Anda telah menginstal modul penerjemah yang dipilih sebelum menjalankan kode ini, misalnya dengan menggunakan "pip install googletrans==4.0.0-rc1"

**5.2 Kode Bahasa**

'af': 'afrikaans',

'sq': 'albanian',

'am': 'amharic',

'ar': 'arabic',

'hy': 'armenian',

'az':'azerbaijani',

'eu': 'basque',

'be':'belarusian',

'bn': 'bengali',

'bs': 'bosnian',

'bg': 'bulgarian',

'ca': 'catalan',

'ceb': 'cebuano',

'ny': 'chichewa',

'zh-cn':'chinese',

'zh-tw':'chinese',

'co': 'corsican',

'hr': 'croatian',

'cs': 'czech',

'da': 'danish',

'nl': 'dutch',

'en': 'english',

'eo': 'esperanto',

'et': 'estonian',

'tl': 'filipino'

'fi': 'finnish',

'fr': 'french',

'fy': 'frisian',

'gl': 'galician',

'ka': 'georgian',

'de': 'german',

'el': 'greek',

'gu': 'gujarati',

'ht': 'haitian’,

'ha': 'hausa',

'haw':'hawaiian',

'iw': 'hebrew',

'he': 'hebrew',

'hi': 'hindi',

'hmn': 'hmong',

'hu': 'hungarian',

'is': 'icelandic',

'ig': 'igbo',

'id':'indonesian',

'ga': 'irish',

'it': 'italian',

'ja': 'japanese',

'jw': 'javanese',

'kn': 'kannada',

'kk': 'kazakh',

'km': 'khmer',

'ko': 'korean',

'ku': 'kurdish (kurmanji)',

'ky': 'kyrgyz',

'lo': 'lao',

'la': 'latin',

'lv': 'latvian',

'lt': 'lithuanian',

'lb':'luxembour',

'mk':'macedona',

'mg': 'malagasy',

'ms': 'malay',

'ml':'malayalam',

'mt': 'maltese',

'mi': 'maori',

'mr': 'marathi',

'mn':'mongolian',

'my': 'myanmar,

'ne': 'nepali',

'ps': 'pashto',

'fa': 'persian',

'pl': 'polish',

'pt':'portuguese',

'pa': 'punjabi',

'ro': 'romanian',

'ru': 'russian',

'sm': 'samoan',

'gd': 'gaelic',

'sr': 'serbian',

'st': 'sesotho',

'sn': 'shona',

'sd': 'sindhi',

'si': 'sinhala',

'sk': 'slovak',

'sl': 'slovenian',

'so': 'somali',

'es': 'spanish',

'su': 'sundanese',

'sw': 'swahili',

'sv': 'swedish',

'tg': 'tajik',

'ta': 'tamil',

'te': 'telugu',

'th': 'thai',

'tr': 'turkish',

'uk': 'ukrainian',

'ur': 'urdu',

'ug': 'uyghur',

'uz': 'uzbek',

'vi':'vietnamese',

'cy': 'welsh',

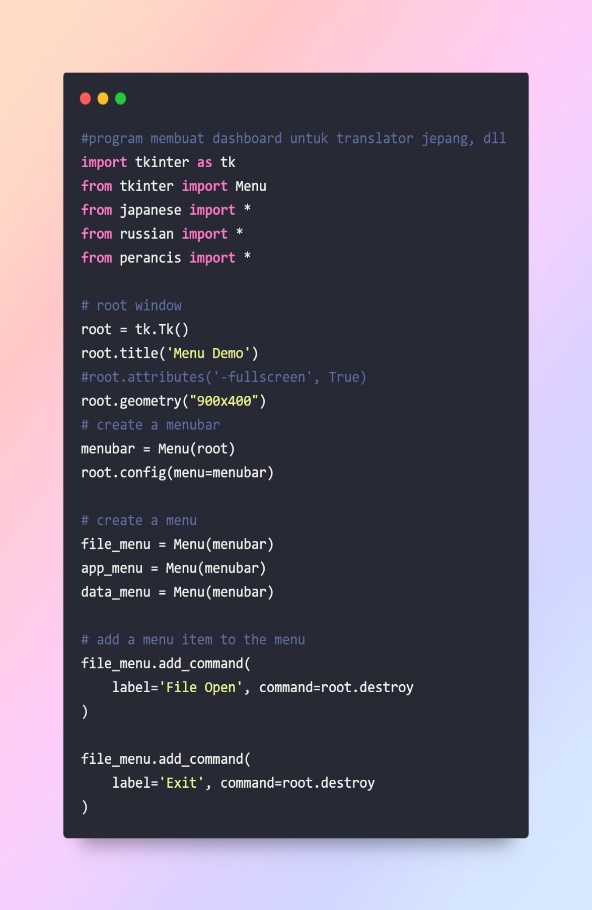
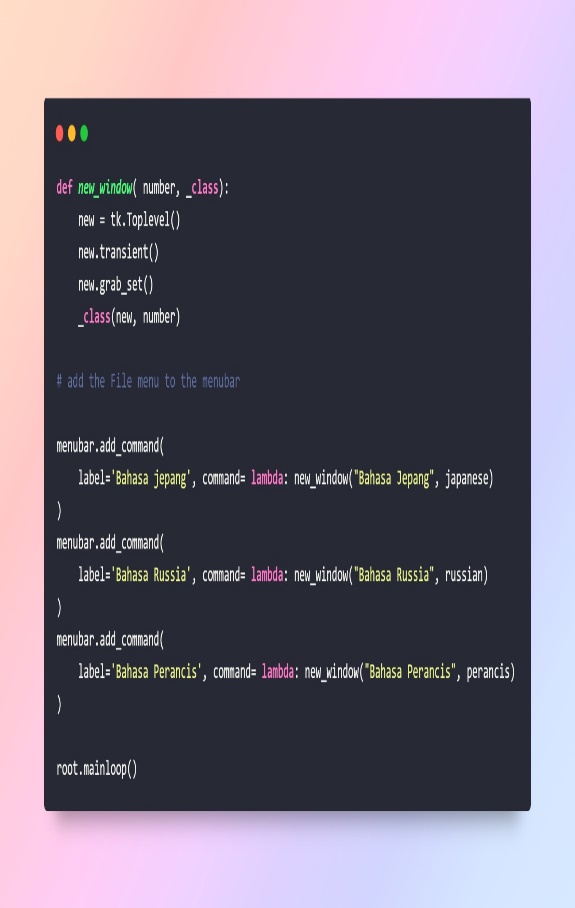
'xh': 'xhosa',

'yi': 'yiddish',

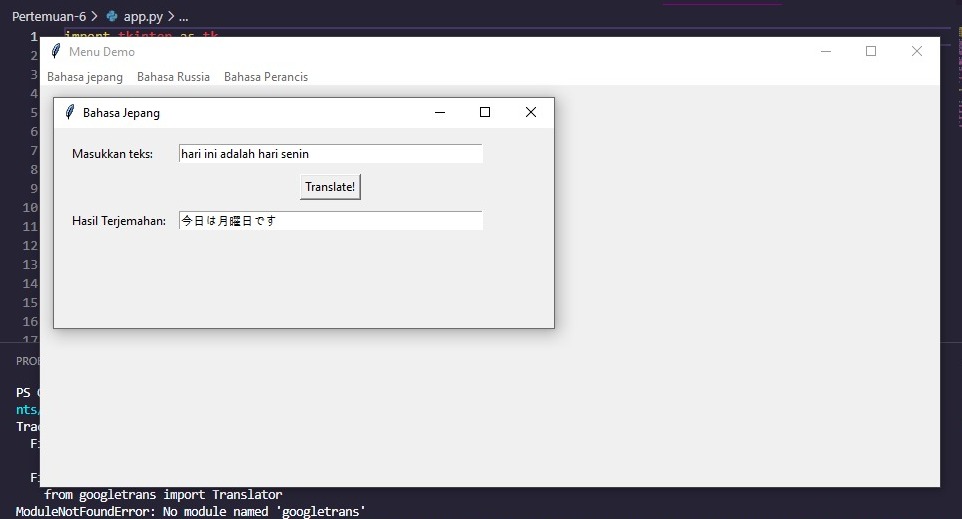
'yo': 'yoruba',

'zu': 'zulu',

## **5.3 Contoh Latihan Translate Bahasa Jepang ke Dalam Python**



**Output**



# **BAB VI**

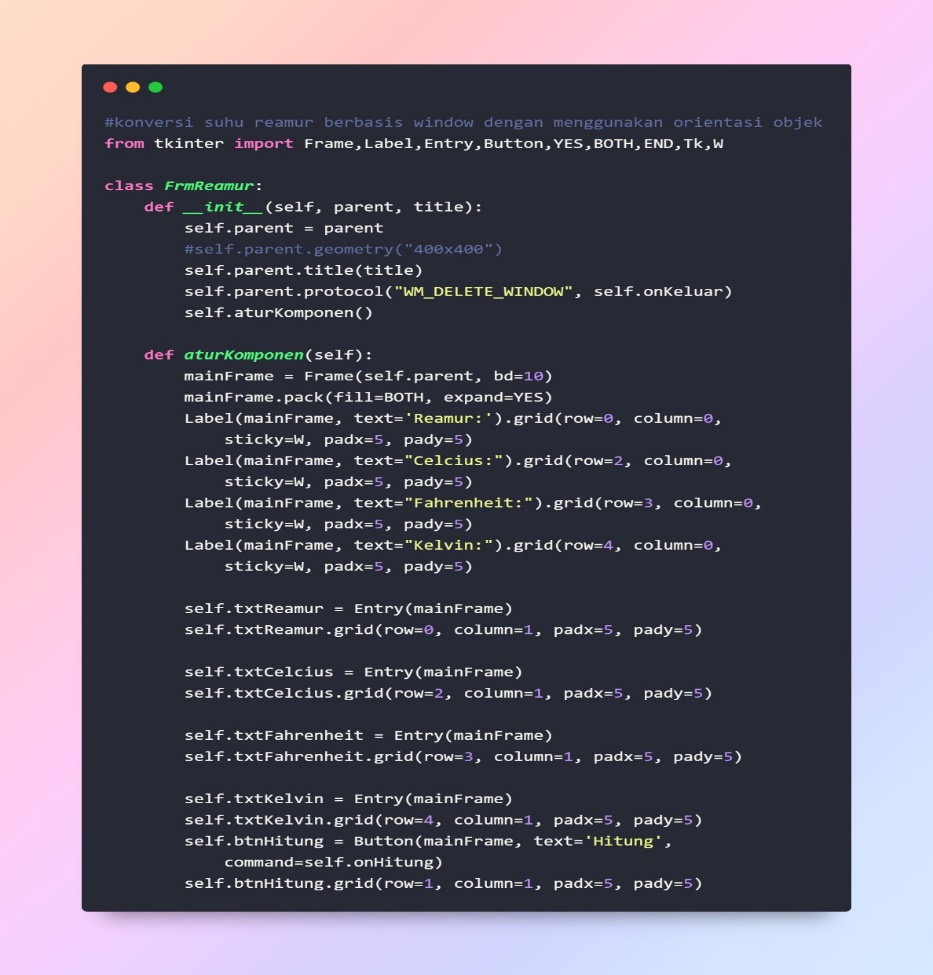
# **OBJECT ORIENTED PROGRAMMING (OOP)**

## **6.1 Dasar Teori**

Konversi Suhu Berorientasi Objek Murni (OOP Murni):

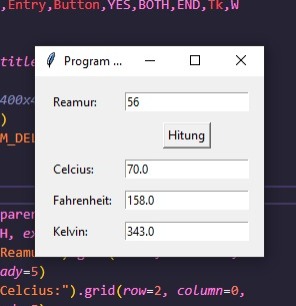
Pendekatan ini lebih ketat dalam menerapkan konsep OOP. Ada kelas-kelas terpisah untuk setiap entitas, seperti InputHandler, Converter, dan OutputHandler. Setiap kelas memiliki tanggung jawab yang terdefinisi dengan jelas, dan hubungan antara kelas-kelas tersebut dibentuk melalui konsep warisan atau komposisi.

## **6.2 Contoh Program Berbasis OOP Murni**





**Output**



# **BAB VII**

# **DATABSE MYSQL (CMD)**

## **7.1 Teori Dasar**

Database MySQL melalui Command Prompt:

Untuk mengakses MySQL melalui command prompt, pengguna perlu menggunakan klien baris perintah seperti MySQL Command-Line Client. Pengguna dapat terhubung ke server MySQL, membuat dan mengelola database, serta menjalankan query SQL langsung melalui command prompt. Ini cocok untuk pengguna yang lebih suka lingkungan teks atau membutuhkan otomatisasi melalui skrip.

**CODE:**

# Untuk terhubung ke server MySQL

mysql -u root -p

# Untuk membuat database baru

CREATE DATABASE mydatabase;

# Untuk beralih ke database yang dibuat

USE mydatabase;

# Untuk menampilkan tabel dalam database

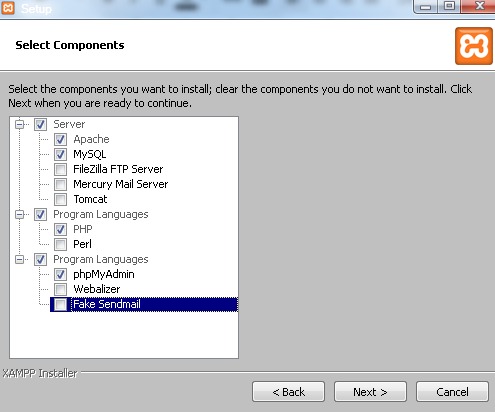
SHOW TABLES;

## **7.2 Installasi XAMPP**

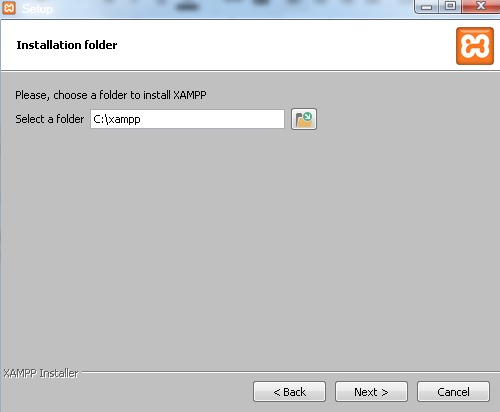
Download xampp di web resmi: [*https://www.apachefriends.org/index.html*](https://www.apachefriends.org/index.html)

****

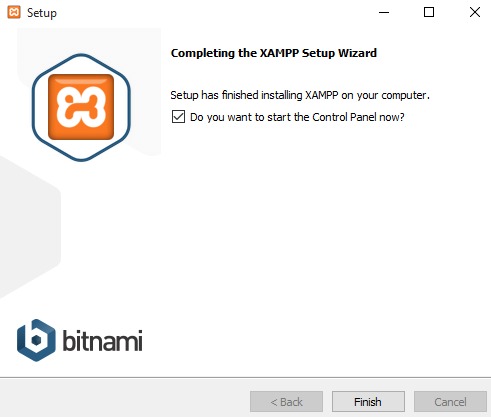
Pilih dari komponen xampp mana yang ingin diInstal dan tidak

****

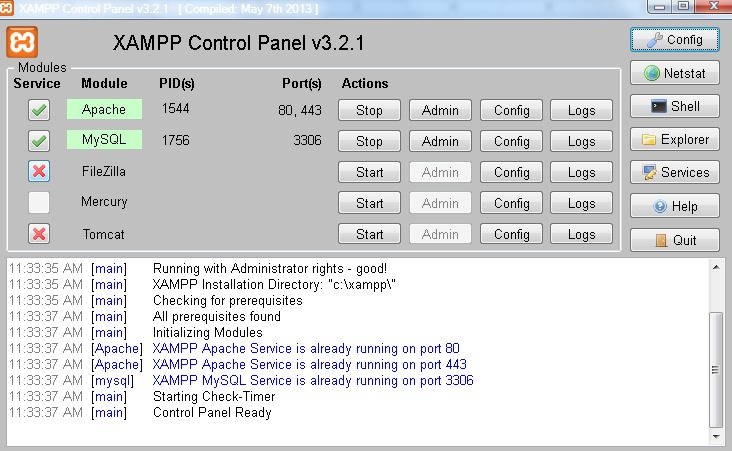
Pilih tujuan folder/direktori di mana xampp diinstall



Selesai

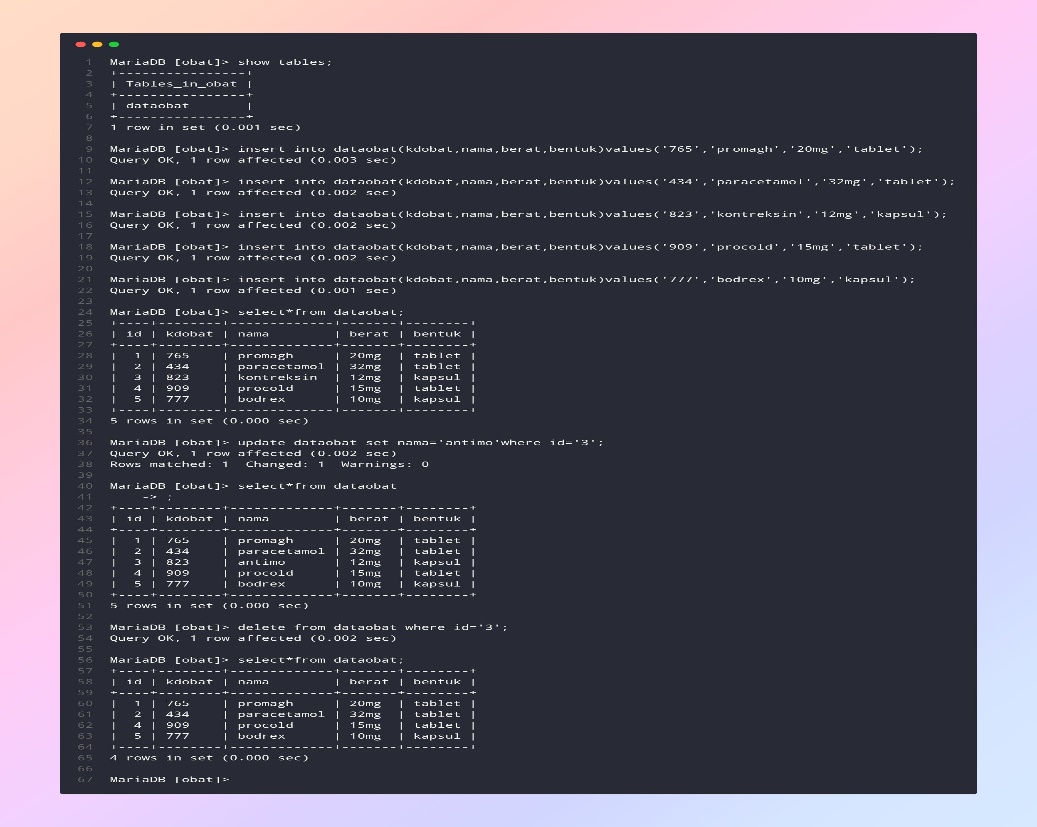


Jalankan xampp control panel di folder di mana xampp diinstal dengan membuka file xampp.exe, langkah ini dilakukan agar ketika kita membuat database di CMD maka bisa dijalankan



**7.3 Implementasi Data Base MySql melalui Command Prompt**





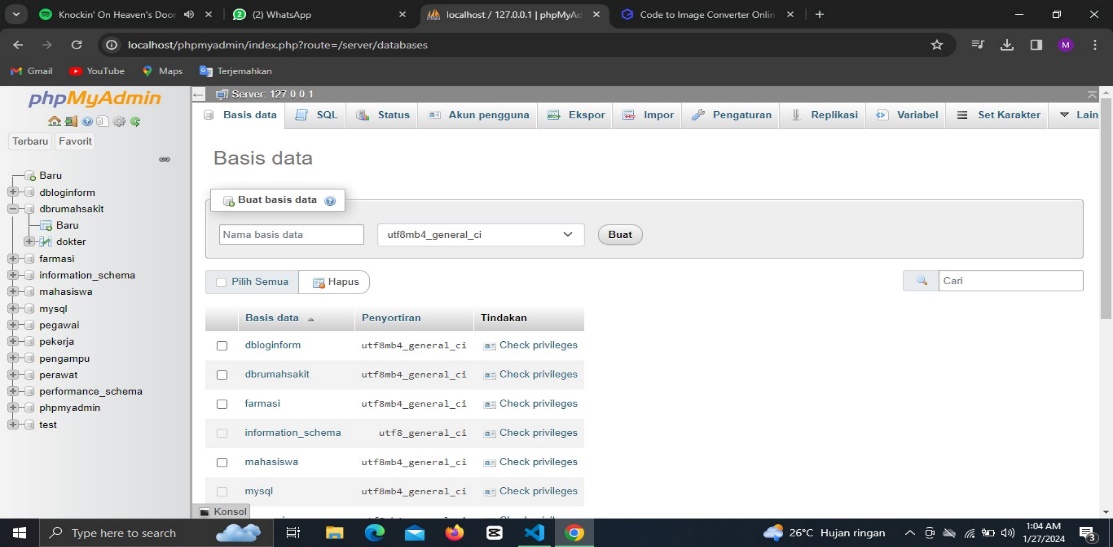
# **BAB VIII**

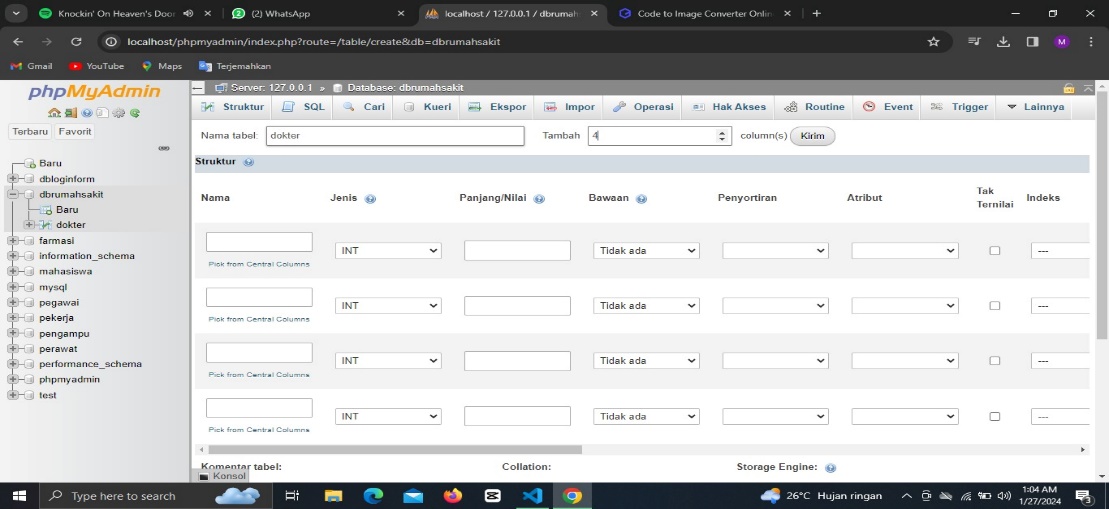
# **DATABSE PYTHON CONNECTION**

## **8.1 Teori Dasar**

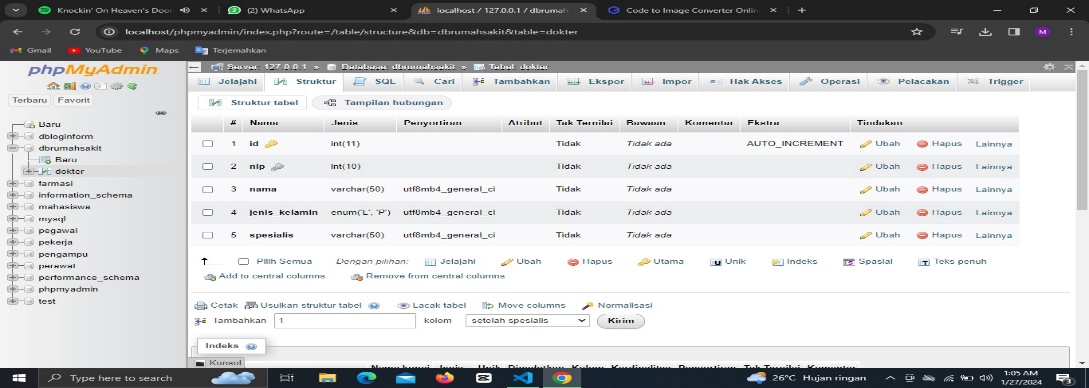
Aplikasi database menggunakan MySQL dan Tkinter dengan DB Connector adalah program yang memanfaatkan Tkinter sebagai antarmuka grafis untuk berinteraksi dengan database MySQL. Dengan menggunakan DB Connector, aplikasi ini dapat mengelola data dengan efisien.

## **8.2 Implementasi**

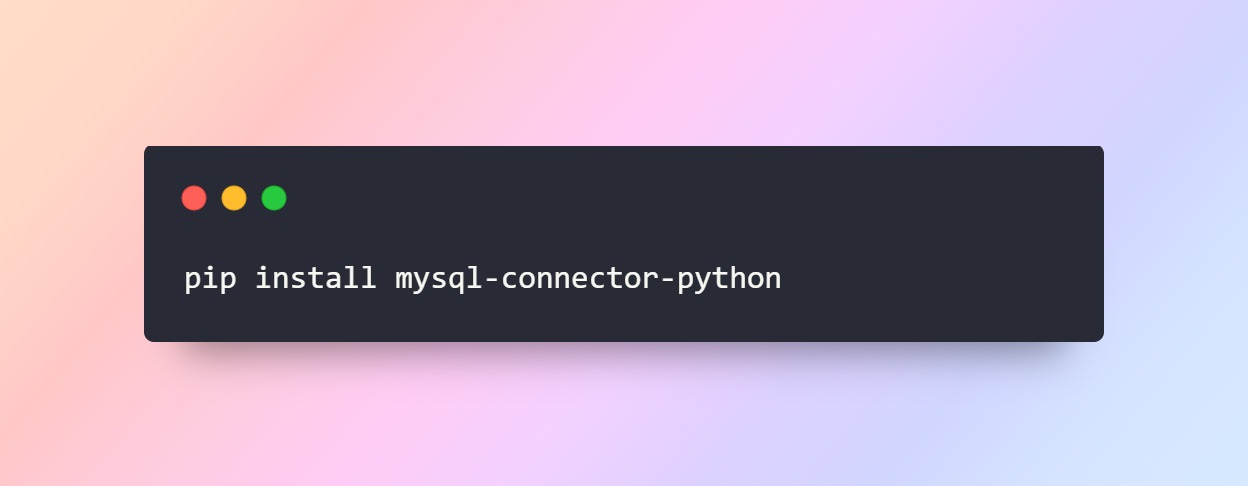
 Buka localhost pada web browser, buat nama database baru

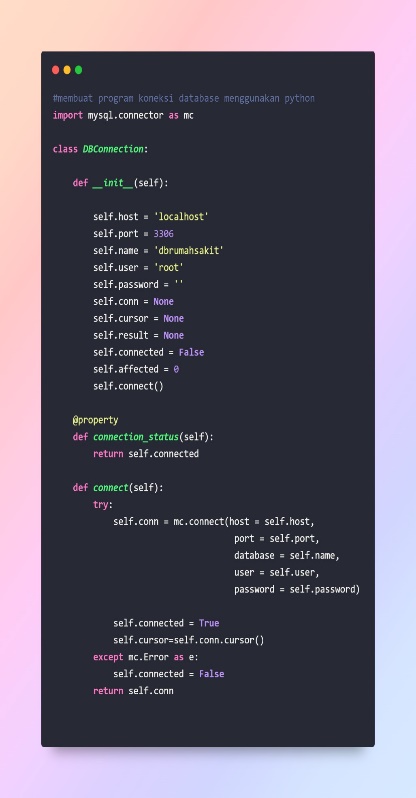
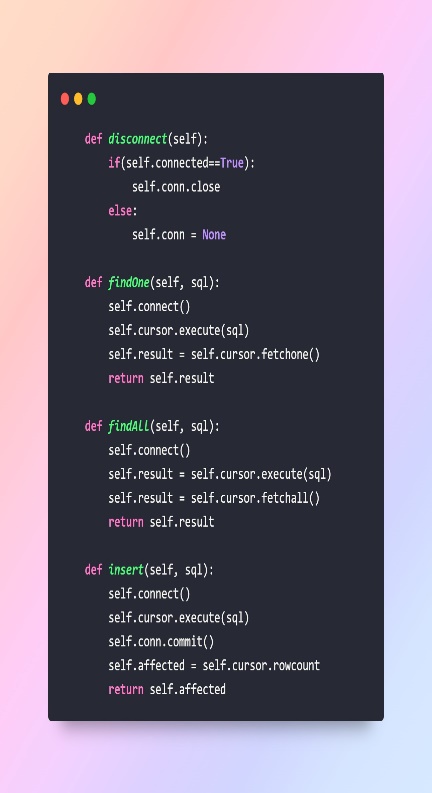
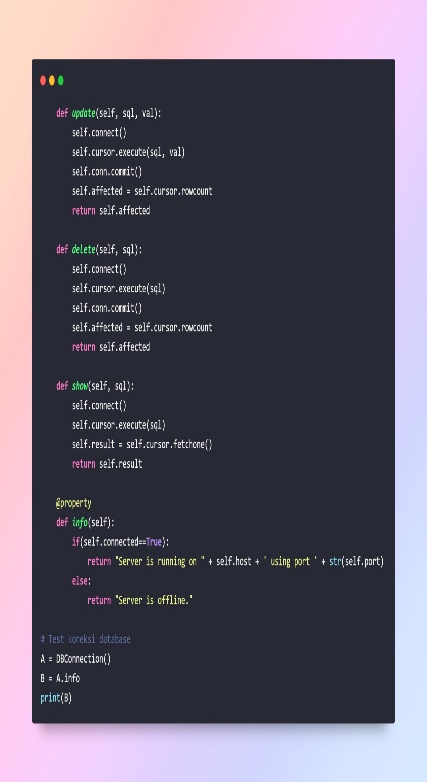
Pada database yang sdh dibuat, buat juga tabel database, nama, dan berapa kolom yang diinginkan

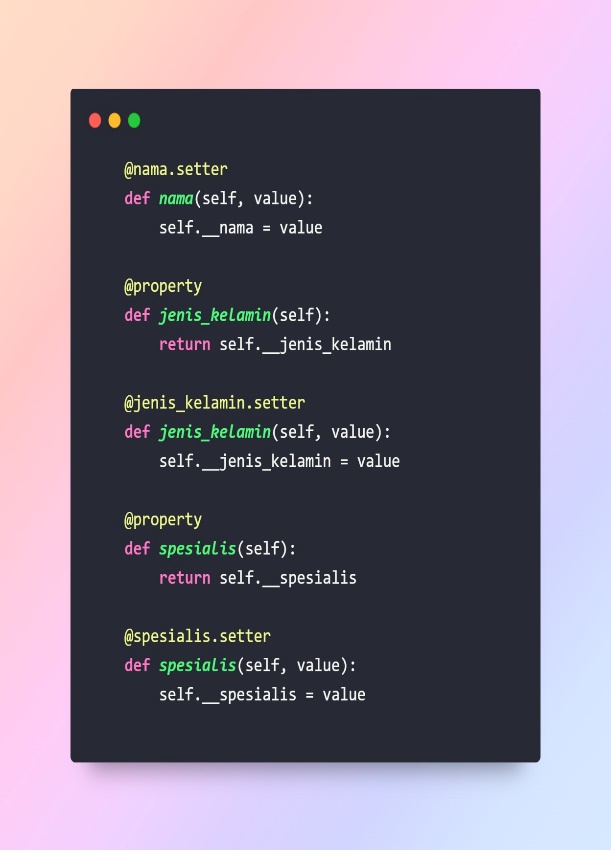
Contoh selengkapnya seperti ini membuat tabel dokter di database rumah sakit

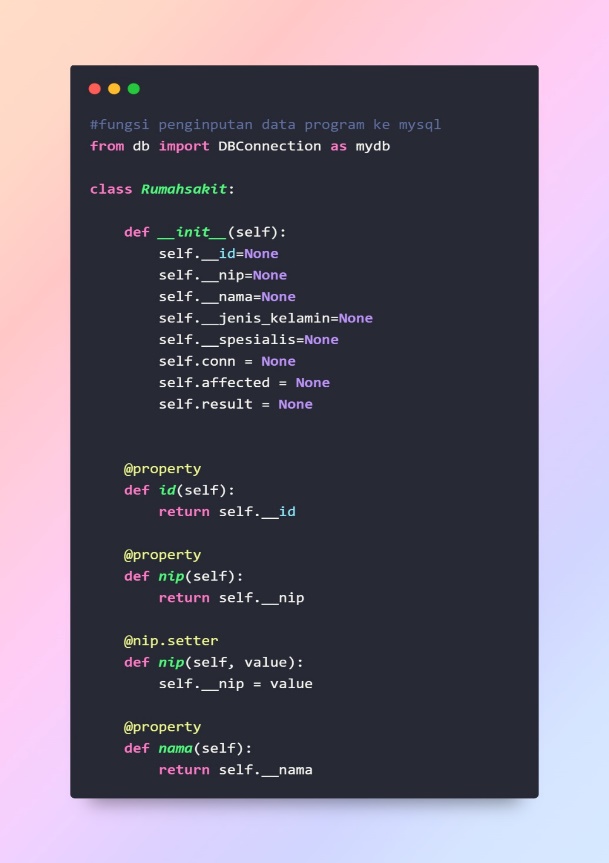
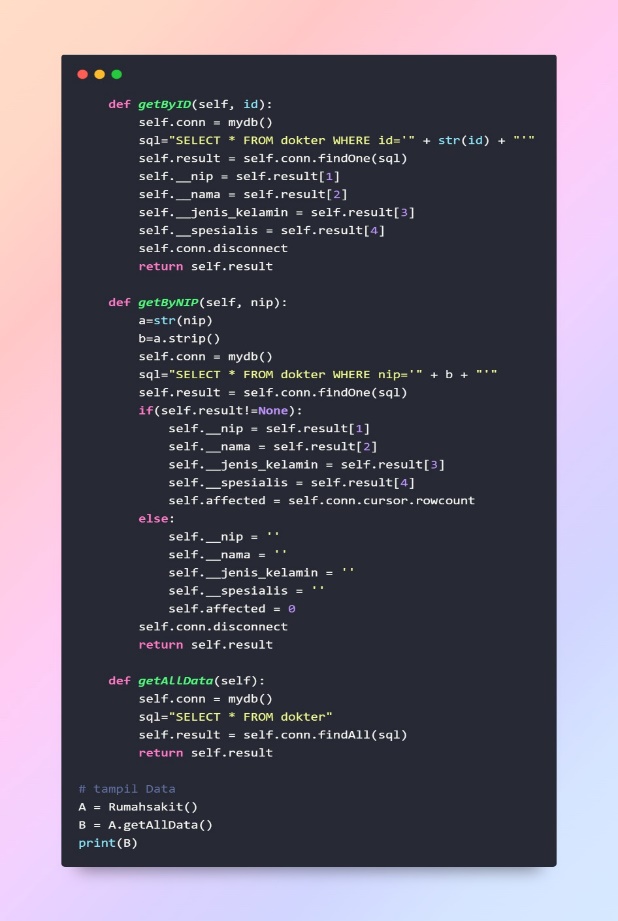


Jangan lupa menginstal mysql connector, agar program python kita dapat terkoneksi dan berinteraksi dengan mysql kita



Buat file python baru, untuk membuat program konektivitas python yg akan menghubungkan program kita dgn database

Program penginputan data kita ke mysql



# Membuat form ui basis window untuk digunakan user menginput data

# 

# Output/hasil

# 

# **BAB IX**

# **LOGIN**

## **9.1 Dasar Teori**

Login Page dan Dashboard menggunakan database MySQL adalah bagian penting dari aplikasi berbasis web yang membutuhkan otentikasi pengguna dan menampilkan data dari database. Berikut adalah deskripsi singkat keduanya:

1. Login Page:

Login Page adalah halaman awal aplikasi yang menyediakan antarmuka untuk pengguna memasukkan informasi kredensial mereka untuk mengakses aplikasi.

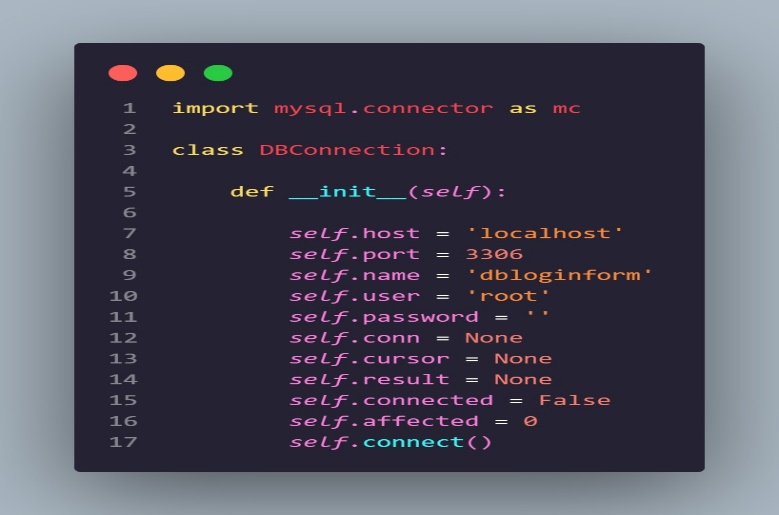
1. Dashboard:

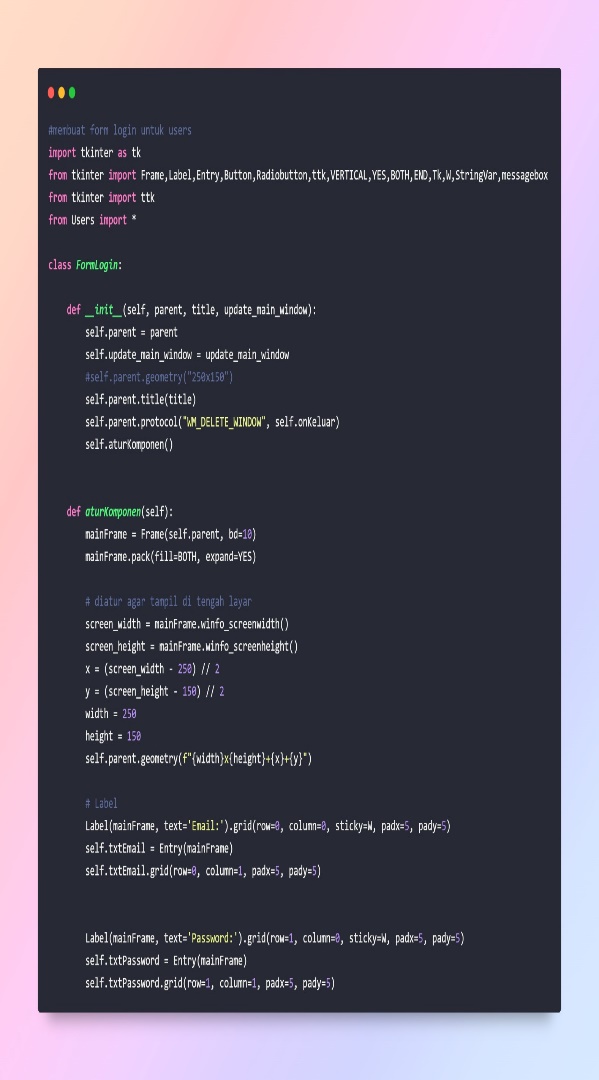
Dashboard adalah halaman utama setelah berhasil login, yang menampilkan ringkasan atau tampilan data yang relevan untuk pengguna yang terotentikasi.

Dengan menggunakan database MySQL, aplikasi ini dapat menyimpan dan mengelola data pengguna serta menyediakan pengalaman pengguna yang personal dan aman. Keseluruhan, kombinasi Login Page dan Dashboard menciptakan lingkungan yang terstruktur dan mudah digunakan bagi pengguna aplikasi berbasis web.

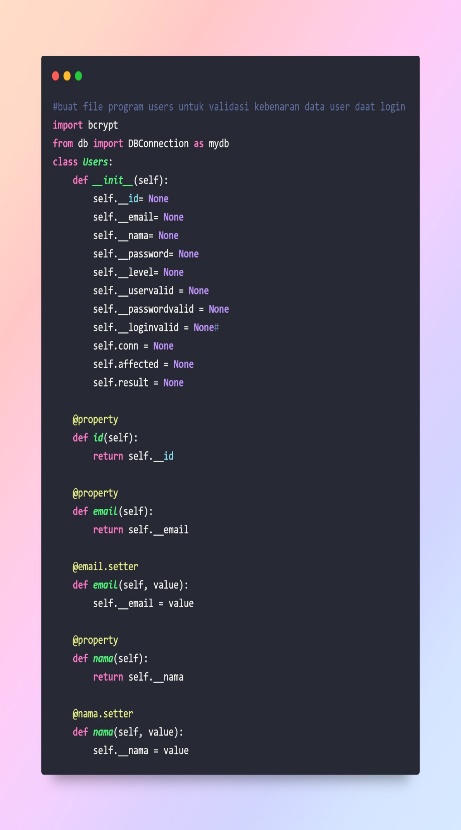
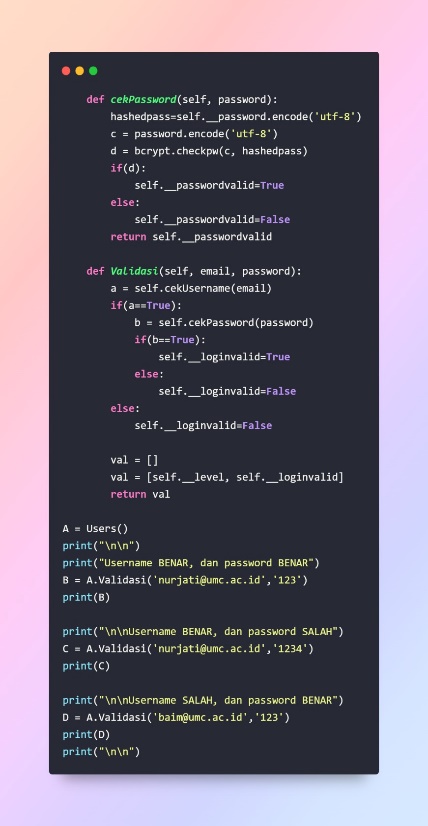
## **9.2 Implementasi Program**

Membuat dashboard multiform dengan login form, dan database. seperti biasa pada file konektivitas db dengan program python, input nama database yang telah dibuat.

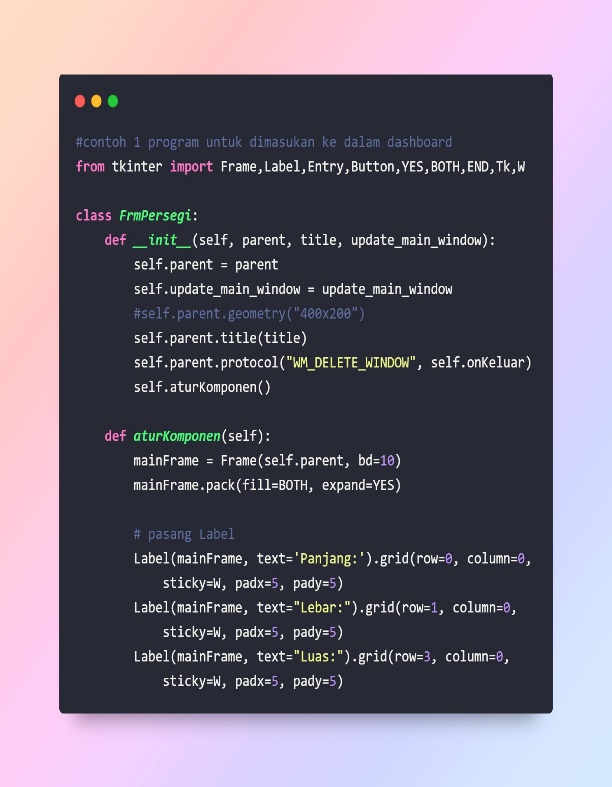
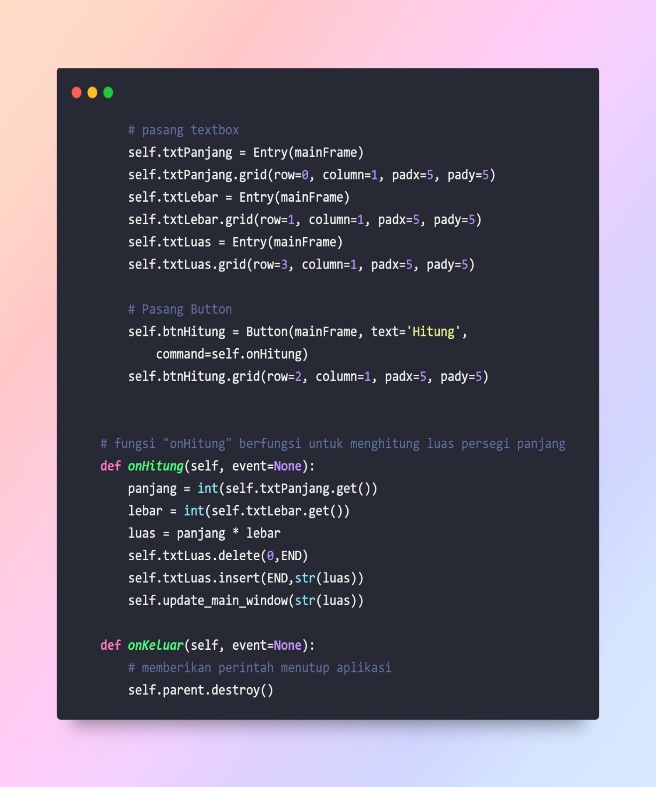


 Pertama buat program form login untuk user dahulu

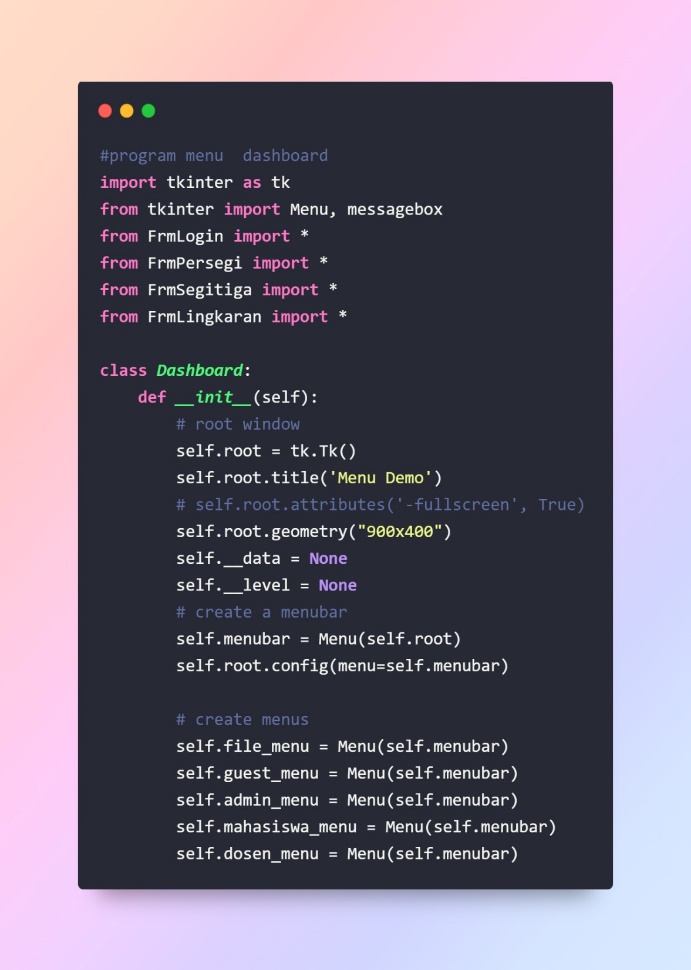
Buat file program users untuk validasi kebenaran data user saat login



Ketika sudah membuat program form login dan validasi users, siapkan 1 program saja minimal untuk dimasukan ke dalam form dashboard. di sini sebagai contoh program mencari keliling dan luas persegi

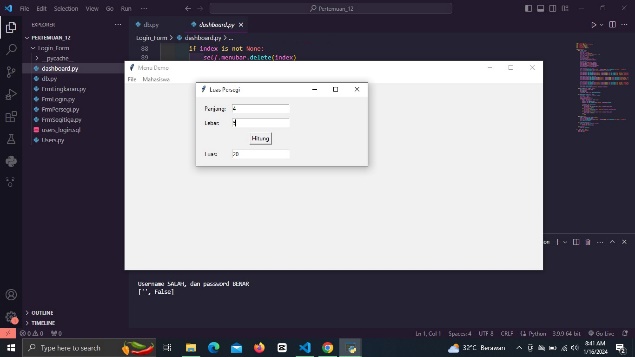
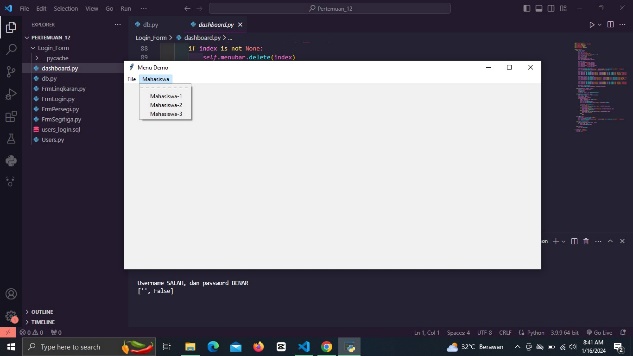
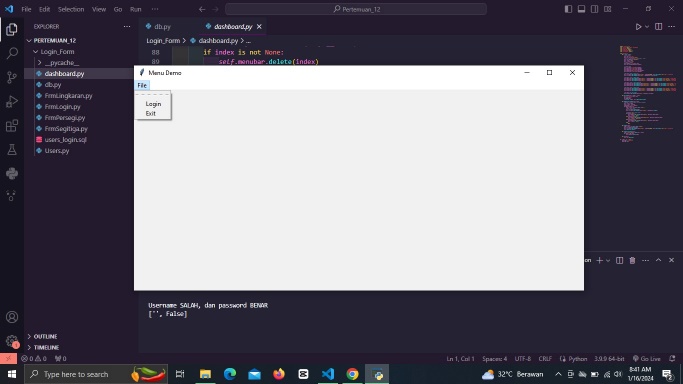
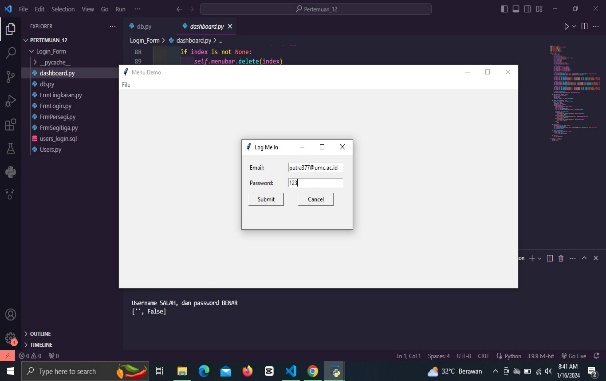


Lanjut membuat program menu dashboard yang akan menampilkan menu form dari berbagai program. dengan cara memanggil tiap nama file programnya.





Output



# **DAFTAR PUSTAKA**

[*https://belajarpython.com/tutorial/kondisi-if-else-python/*](https://belajarpython.com/tutorial/kondisi-if-else-python/)

[*https://www.imgbly.com/ib/MV6hzgsTh4*](https://www.imgbly.com/ib/MV6hzgsTh4)

[*https://github.com/Antares023/PBO*](https://github.com/Antares023/PBO)

[*https://github.com/MptrJ1807/PutraRizki-PBO.git*](https://github.com/MptrJ1807/PutraRizki-PBO.git)

[*https://github.com/hafid25/PBO-SEMESTER-3/tree/main/220511118\_PBO*](https://github.com/hafid25/PBO-SEMESTER-3/tree/main/220511118_PBO)