**LAPORAN PEMROGRAMAN BERORIENTASI OBJEK**

## ” MEMBUAT APLIKASI KALKULATOR LUAS DAN VOLUME BALOK DENGAN KIVYMD”

Dosen Pengampuh : Freddy Wicaksono



Disusun oleh :

R.Satria Maulana A.W.P.P. 170511019

Imanda 210511089

Moh. Rifki Ramadhan 210511020

**PROGRAM STUDI TEKNIK INFORMATIKA FAKULTAS TEKNIK UNIVERSITAS MUHAMMADIYAH CIREBON 2023/2024**

DADTAR ISI

Daftar isi i

[Abstrak ii](#_Toc151229589)

[BAB I 1](#_Toc151229590)

[PENDAHULUAN 1](#_Toc151229591)

[Latar Belakang 1](#_Toc151229592)

[Rumusan dan Batasan Masalah 1](#_Toc151229593)

[Maksud dan tujuan 1](#_Toc151229594)

[Tujuan 1](#_Toc151229595)

[BAB II 2](#_Toc151229596)

[PEMBAHASAN 2](#_Toc151229597)

[Python 2](#_Toc151229598)

[KivyMD 2](#_Toc151229599)

[Pemrograman Berorientasi Objek 2](#_Toc151229600)

[BAB III 3](#_Toc151229601)

[PERANCANGAN 3](#_Toc151229602)

[BAB IV 14](#_Toc151229603)

[HASIL RANCANGAN 14](#_Toc151229604)

[1. Halaman Beranda 14](#_Toc151229605)

[2. Halaman Menghitung 15](#_Toc151229607)

[3. Halaman Error 15](#_Toc151229608)

[Kesimpulan 16](#_Toc151229609)

[Penutup 17](#_Toc151229611)

# Abstrak

Aplikasi Kalkulator Luas dan Volume Balok dengan menggunakan KivyMD dirancang sebagai sebuah implementasi sederhana untuk memberikan bantuan kepada pengguna dalam melakukan perhitungan luas permukaan dan volume balok. Dengan memanfaatkan bahasa pemrograman Python dan framework KivyMD, aplikasi ini tidak hanya menawarkan solusi praktis, tetapi juga menyajikan pengalaman interaktif yang memudahkan pengguna dalam mengatasi kebutuhan perhitungan matematika terkait bangun ruang.

Aplikasi ini memberikan kemudahan akses kepada pengguna yang mungkin membutuhkan perhitungan cepat dan akurat untuk luas permukaan dan volume balok. Kelebihan utama aplikasi ini terletak pada antarmuka yang sederhana dan ramah pengguna, serta responsifitas tinggi yang diberikan oleh framework KivyMD.

Langkah-langkah instalasi yang disediakan memastikan bahwa pengguna dapat dengan mudah mengakses dan menjalankan aplikasi ini di lingkungan Python mereka. Dengan demikian, aplikasi ini diharapkan dapat menjadi alat yang bermanfaat dalam mendukung kegiatan perhitungan matematika sehari-hari atau pun dalam konteks pendidikan.

Sebagai tambahan, aplikasi ini tidak hanya berfokus pada penyediaan hasil perhitungan, tetapi juga mengimplementasikan mekanisme penanganan error. Pesan error yang dihasilkan akan memberikan informasi yang jelas jika terjadi kesalahan, memastikan bahwa pengguna dapat dengan cepat mengidentifikasi dan memperbaiki masalah yang mungkin terjadi pada input mereka.

Dengan demikian, Aplikasi Kalkulator Luas dan Volume Balok ini diharapkan dapat memberikan nilai tambah bagi pengguna yang mencari alat bantu perhitungan sederhana dan efektif, serta dapat dijadikan sebagai referensi pembelajaran dalam pengembangan aplikasi dengan menggunakan framework KivyMD

# 

# BAB I

# PENDAHULUAN

## Latar Belakang

Dalam kehidupan sehari-hari, perhitungan luas permukaan dan volume balok seringkali menjadi kebutuhan dalam berbagai konteks, termasuk matematika, fisika, rekayasa, dan sebagainya. Dalam situasi ini, memiliki alat kalkulator yang cepat, akurat, dan mudah digunakan sangat membantu. Oleh karena itu, Aplikasi Kalkulator Luas dan Volume Balok menggunakan KivyMD dikembangkan untuk memberikan solusi praktis bagi pengguna dalam menyelesaikan perhitungan tersebut.

## Rumusan dan Batasan Masalah

Dalam pengembangan Aplikasi Kalkulator Luas dan Volume Balok menggunakan KivyMD, terdapat beberapa fokus dan pembatasan:

**1. Rumusan Masalah:**

1. Pengguna kesulitan perhitungan manual luas permukaan dan volume balok.
2. Kebutuhan akan alat kalkulator sederhana tanpa pengetahuan matematika tinggi.
3. Diperlukan penanganan error untuk input yang tidak valid.

**2. Batasan Masalah:**

* 1. Aplikasi hanya menghitung luas permukaan dan volume balok.
  2. Hanya menerima input berupa angka.
  3. Hasil perhitungan diberikan tanpa satuan pengukuran.

## Maksud dan tujuan

Aplikasi Kalkulator Luas dan Volume Balok menggunakan KivyMD dibuat untuk memberikan solusi praktis dan sederhana bagi pengguna yang membutuhkan alat kalkulator untuk perhitungan luas permukaan dan volume balok.

## Tujuan

1. Menyederhanakan perhitungan luas permukaan dan volume balok.
2. Membuat antarmuka pengguna yang sederhana dan mudah diakses.
3. Memberikan referensi pembelajaran tentang pengembangan aplikasi dengan KivyMD.

# BAB II

# PEMBAHASAN

## Python

Python adalah bahasa pemrograman tingkat tinggi yang dikenal dengan sintaksis yang bersih, mudah dibaca, dan ekspresif. Beberapa pendapat para ahli dan sumber terpercaya tentang Python melibatkan penghargaan terhadap kemudahan penggunaan, fleksibilitas, dan kekuatan bahasa ini

## KivyMD

Adalah ekstensi dari Kivy, sebuah framework open-source untuk pengembangan aplikasi multi-touch. KivyMD dirancang khusus untuk menambahkan komponen Material Design ke dalam aplikasi yang dibangun dengan Kivy

## Pemrograman Berorientasi Objek

Dalam pemrograman berorientasi objek, setiap objek apapun akan memiliki data kit (sifat, berupa variabel maupun konstanta) dan metode (perilaku atau kemampuan melakukan sesuatu, berupa fungsi). Jadi, objek dapat didefinisikan sebagai suatu entitas yang memiliki data dan metode.

Semua bahasa pemrograman berorientasi objek menyediakan mekanisme untuk dapat membantu mengimplementasikan model yang berorientasi objek. Prinsip tersebut adalah encapulation, inheritence, dan polymorphism.

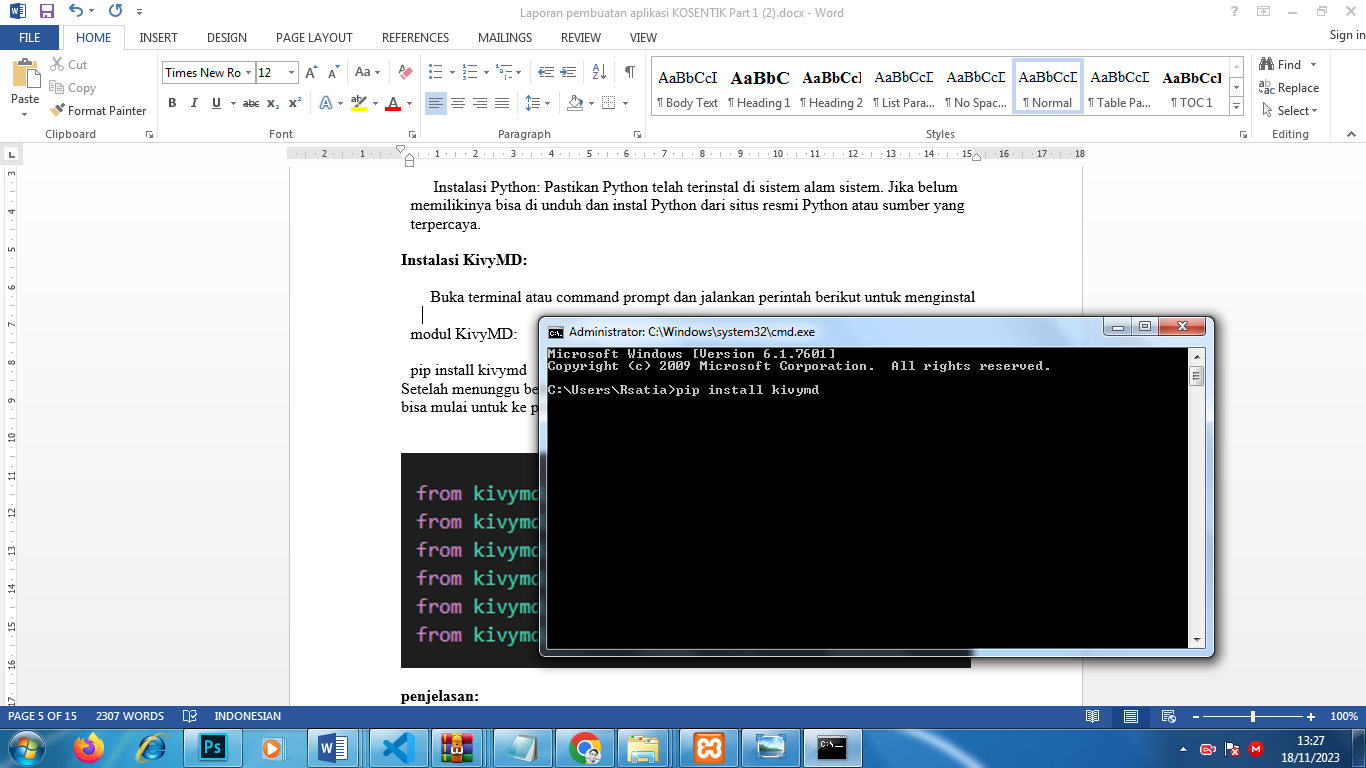
## BAB III

## PERANCANGAN

**Langkah-langkah Instalasi**

Instalasi Python: Pastikan Python telah terinstal di sistem alam sistem. Jika belum memilikinya bisa di unduh dan instal Python dari situs resmi Python atau sumber yang terpercaya.

**Instalasi KivyMD:**  
  
 Buka terminal atau command prompt dan jalankan perintah berikut untuk menginstal

  
  
 Modul KivyMD: pip install kivymd

Setelah menunggu beberapa saat hingga proses download dan instalasi selesai maka kita bisa mulai untuk ke proses selanjutnya.

**Menjalankan Aplikasi**

Untuk menjalankan aplikasi, simpan kode di atas dalam sebuah file (misalnya, calculator\_app.py). Lalu, buka terminal atau command prompt, arahkan ke direktori file tersebut, dan jalankan perintah:

python calculator\_app.py

**Penggunaan Aplikasi**

Masukkan Dimensi Balok: Isi input field untuk panjang, lebar, dan tinggi balok.

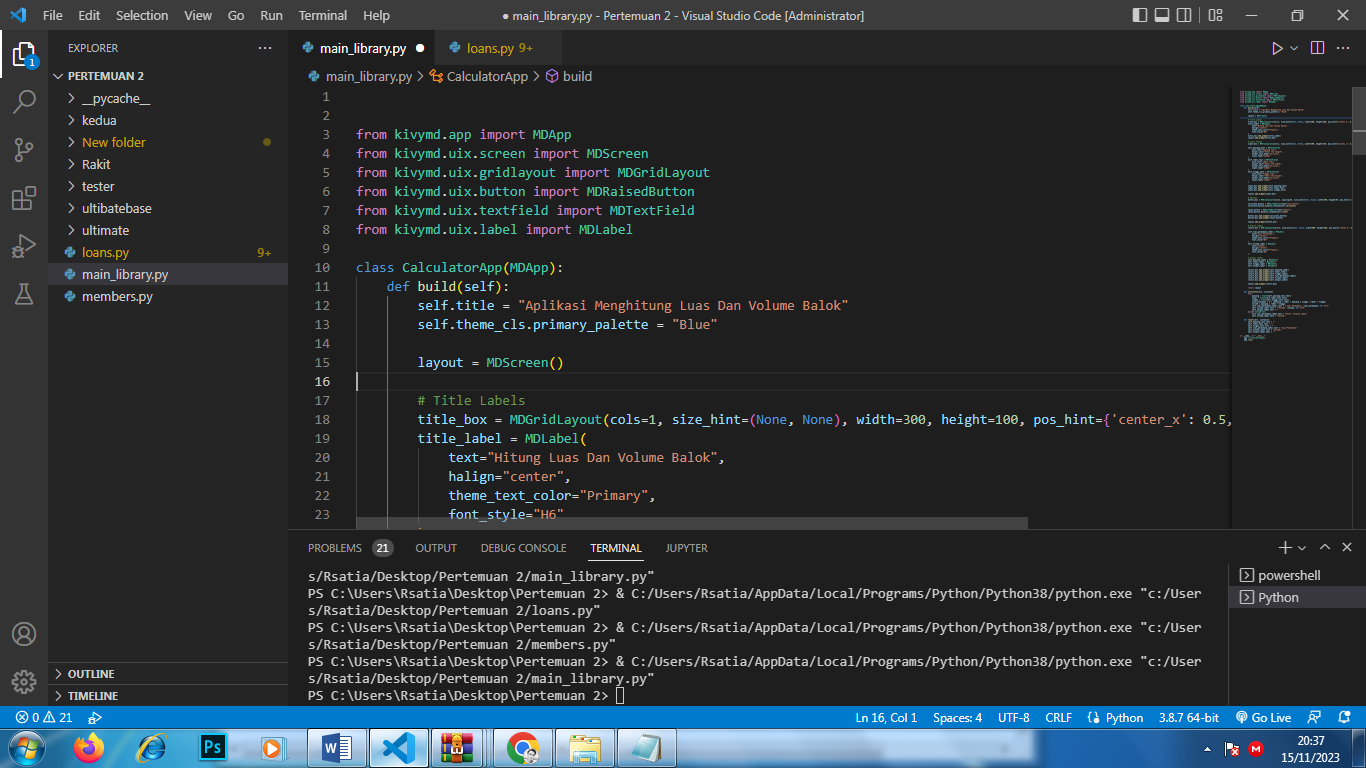
Hitung: Tekan tombol "Calculate" untuk menghitung luas permukaan dan volume balok.

Reset: Tombol "Reset" digunakan untuk menghapus input dan hasil perhitungan.

Error Handling: Aplikasi memberikan pesan error jika input tidak valid (bukan angka).

Dengan demikian, aplikasi ini memberikan pengalaman interaktif bagi pengguna untuk menghitung luas permukaan dan volume balok secara sederhana menggunakan antarmuka grafis.

**Modul import**

  
  
**Penjelasan:**

1. **from kivymd.app import MDApp**

Baris ini mengimpor kelas MDApp dari modul kivymd.app. MDApp adalah kelas utama yang harus diwarisi untuk membuat aplikasi KivyMD.

1. **From kivymd.uix.screen import MDScreen**

Baris ini mengimpor kelas MDScreen dari modul kivymd.uix.screen. MDScreen digunakan untuk membuat layar (screen) di aplikasi.

1. **from kivymd.uix.gridlayout import MDGridLayout**

Baris ini mengimpor kelas MDGridLayout dari modul kivymd.uix.gridlayout. MDGridLayout adalah tata letak (layout) berbasis grid yang memungkinkan penempatan widget dalam sel grid.

1. **from kivymd.uix.button import MDRaisedButton**

Baris ini mengimpor kelas MDRaisedButton dari modul kivymd.uix.button. MDRaisedButton adalah tombol yang memiliki efek naik saat ditekan.

1. **from kivymd.uix.textfield import MDTextField**

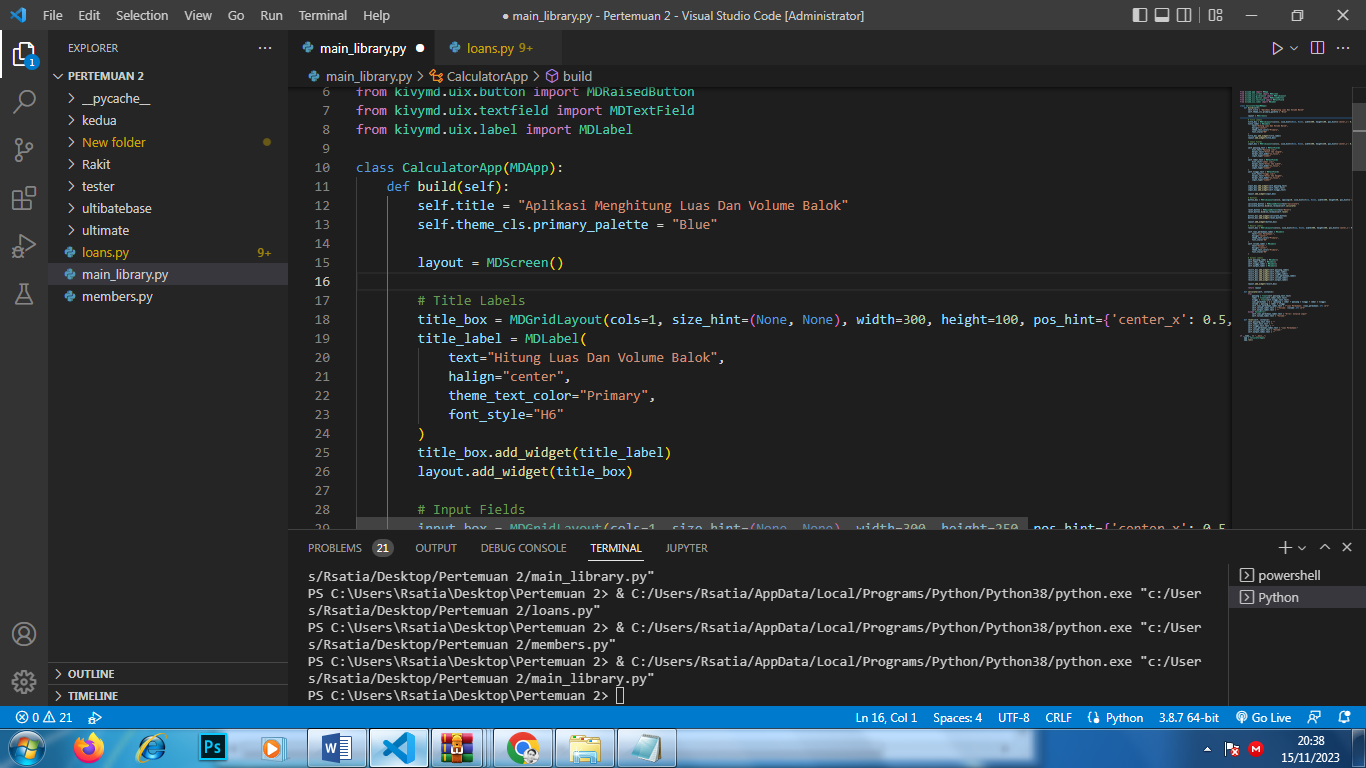
Baris ini mengimpor kelas MDTextField dari modul kivymd.uix.textfield. MDTextField adalah widget input teks yang dapat digunakan pengguna untuk memasukkan teks.

1. **from kivymd.uix.label import MDLabel**

Baris ini mengimpor kelas MDLabel dari modul kivymd.uix.label. MDLabel adalah widget label yang digunakan untuk menampilkan teks atau informasi statis di antarmuka pengguna.

Dengan menggunakan kelas-kelas ini, kita dapat membuat antarmuka pengguna dengan tombol, input teks, label, dan tata letak grid. Contohnya bisa seperti membuat layar dengan beberapa widget seperti tombol, input teks, dan label, dan mengatur tata letaknya menggunakan MDGridLayout. Saat tombol ditekan, kita dapat menangani logika atau peristiwa tertentu dalam metode yang sesuai dengan kelas aplikasi yang dibuat dengan mewarisi MDApp.

**Class Calculator App**



**Penjelasan :**

1. **Class CalculatorApp(MDApp)**

Mendefinisikan kelas CalculatorApp yang merupakan turunan dari kelas `MDApp` dari KivyMD. Ini adalah kelas utama aplikasi.

1. **def build(self)**

Merupakan metode wajib yang akan dipanggil ketika aplikasi dimulai. Metode ini digunakan untuk membuat dan mengembalikan antarmuka pengguna (UI) aplikasi.

1. **self.title = "Aplikasi Menghitung Luas Dan Volume Balok"**

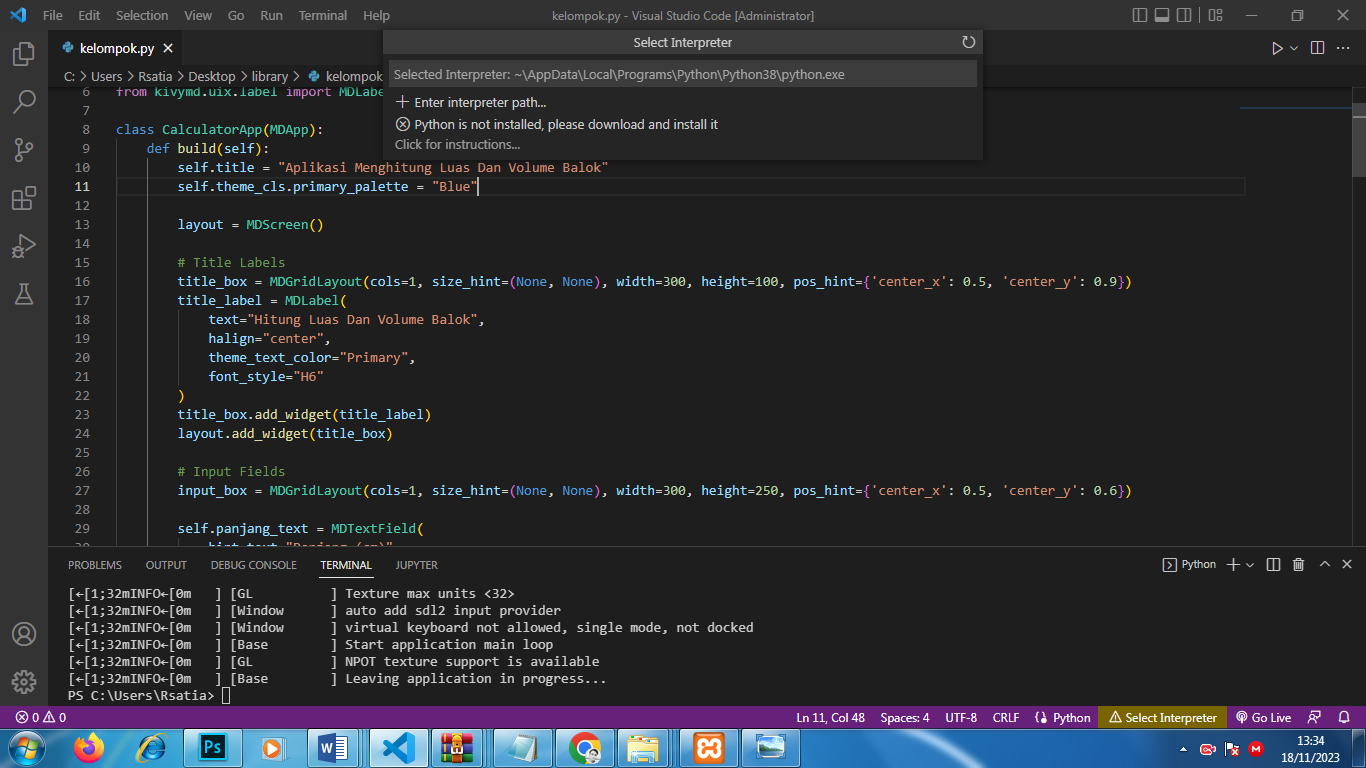
Mengatur judul aplikasi menjadi "Aplikasi Menghitung Luas Dan Volume Balok".

1. **self.theme\_cls.primary\_palette = "Blue"**

Mengatur palet warna utama aplikasi menjadi "Blue". Ini mempengaruhi warna beberapa elemen UI.

1. **layout = MDScreen()**

Membuat objek `MDScreen` sebagai layout utama aplikasi. `MDScreen` digunakan sebagai wadah untuk menempatkan elemen-elemen antarmuka pengguna.

**Title Labels** **Penjelasan**

1. **title\_box = MDGridLayout(cols=1, size\_hint=(None, None), width=300, height=100, pos\_hint={'center\_x': 0.5, 'center\_y': 0.9})** Membuat kotak layout berbasis grid (`MDGridLayout`) dengan satu kolom untuk judul. Parameter lainnya seperti `size\_hint`, `width`, `height`, dan `pos\_hint` mengatur properti tata letak dan penempatan kotak tersebut.
2. **title\_label = MDLabel(...)**

Membuat label (`MDLabel`) dengan teks "Hitung Luas Dan Volume Balok". Beberapa parameter seperti `halign`, `theme\_text\_color`, dan `font\_style` mengatur properti penampilan label.

1. **title\_box.add\_widget(title\_label)**

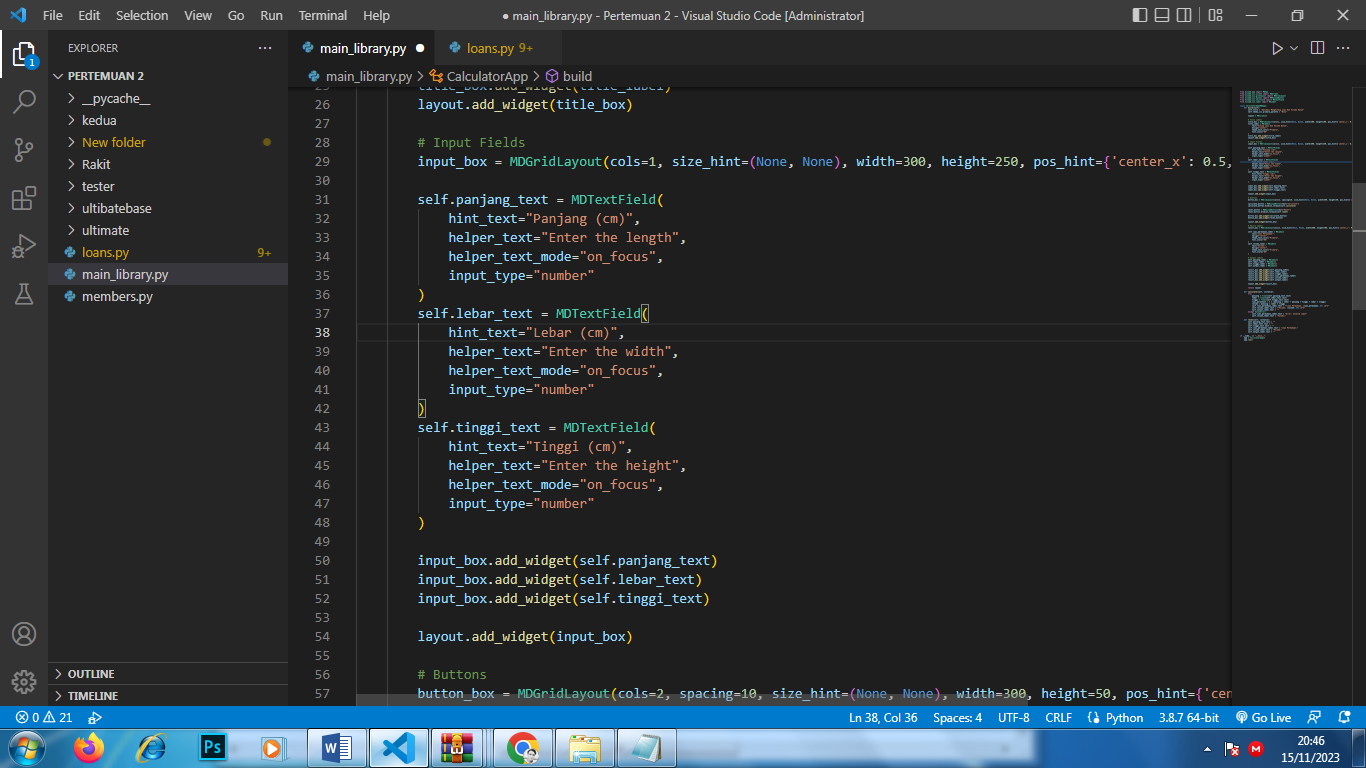
Menambahkan label ke dalam kotak judul.

1. **layout.add\_widget(title\_box)**

Menambahkan kotak judul ke dalam layout utama (`MDScreen`).

Dengan kode ini, kita telah membuat bagian pertama dari antarmuka pengguna yang berisi judul aplikasi. Bagian ini memanfaatkan elemen-elemen dari KivyMD untuk membangun tampilan yang terstruktur.

**Imput fields**



**Penjelasan:**

1. **input\_box = MDGridLayout(...)**

Membuat kotak layout berbasis grid (`MDGridLayout`) dengan satu kolom. Parameter seperti size\_hint, width, height, dan pos\_hint mengatur properti tata letak dan penempatan kotak input.

1. **self.panjang\_text = MDTextField(...)**

Membuat objek MDTextField untuk input panjang. Ini adalah elemen input teks yang dapat digunakan untuk memasukkan panjang balok. Beberapa parameter seperti hint\_text, helper\_text, helper\_text\_mode, dan input\_type mengatur properti dan perilaku input.

1. **self.lebar\_text = MDTextField(...) dan self.tinggi\_text = MDTextField(...)**

Sama seperti langkah sebelumnya, tetapi untuk input lebar dan tinggi.

1. **input\_box.add\_widget(self.panjang\_text)**

Menambahkan input panjang ke dalam kotak input.

1. **input\_box.add\_widget(self.lebar\_text) dan input\_box.add\_widget(self.tinggi\_text)**

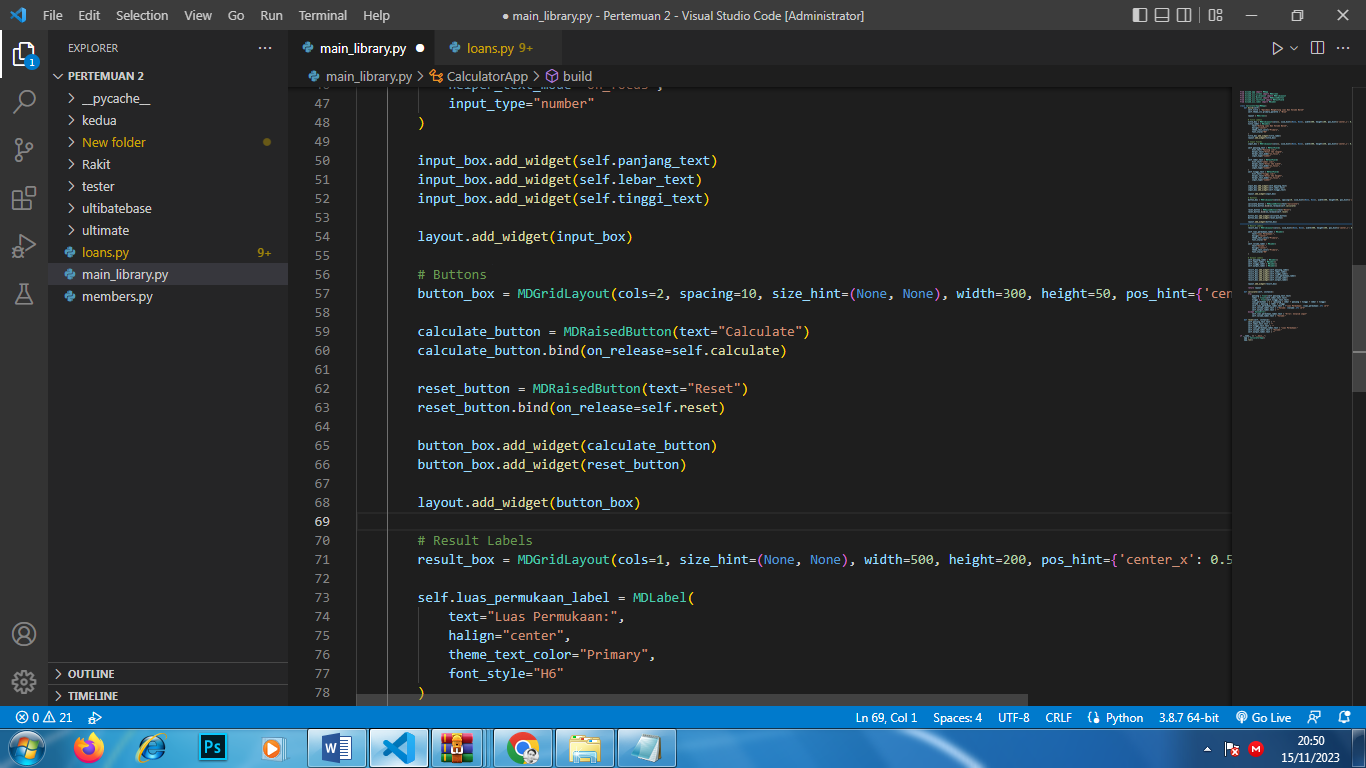
Sama seperti langkah sebelumnya, tetapi untuk input lebar dan tinggi.

1. **layout.add\_widget(input\_box)**

Menambahkan kotak input ke dalam layout utama (`MDScreen`).

Bagian ini bertanggung jawab untuk membuat dan menempatkan input fields (kolom input) pada antarmuka pengguna.

Dengan kode ini, kita telah membuat bagian input dari antarmuka pengguna yang memungkinkan pengguna memasukkan panjang, lebar, dan tinggi balok yang akan dihitung luas permukaan dan volume-nya. Ini menggunakan elemen `MDTextField` dari KivyMD untuk mendukung input teks dalam aplikasi.

**Buttons:**  
  


**Penjelasan:**

1. **button\_box = MDGridLayout(...)**

Membuat kotak layout berbasis grid (`MDGridLayout`) dengan dua kolom. Parameter seperti cols, spacing, size\_hint, width, height, dan pos\_hint mengatur properti tata letak dan penempatan kotak tombol.

1. c**alculate\_button = MDRaisedButton(text="Calculate")**

Membuat tombol "Calculate" menggunakan kelas MDRaisedButton dari KivyMD. Parameter text mengatur teks yang akan ditampilkan pada tombol.

1. **calculate\_button.bind(on\_release=self.calculate)**

Mengikat event "on\_release" dari tombol "Calculate" ke metode calculate pada objek kelas saat tombol ditekan. Artinya, ketika tombol "Calculate" ditekan, metode calculate akan dijalankan.

1. **reset\_button = MDRaisedButton(text="Reset")**

Membuat tombol "Reset" menggunakan kelas MDRaisedButton. Parameter text mengatur teks yang akan ditampilkan pada tombol.

1. **reset\_button.bind(on\_release=self.reset)**

Mengikat event "on\_release" dari tombol "Reset" ke metode `reset` pada objek kelas saat tombol ditekan. Artinya, ketika tombol "Reset" ditekan, metode `reset` akan dijalankan.

1. **button\_box.add\_widget(calculate\_button) dan button\_box.add\_widget(reset\_button)**

Menambahkan tombol "Calculate" dan "Reset" ke dalam kotak tombol.

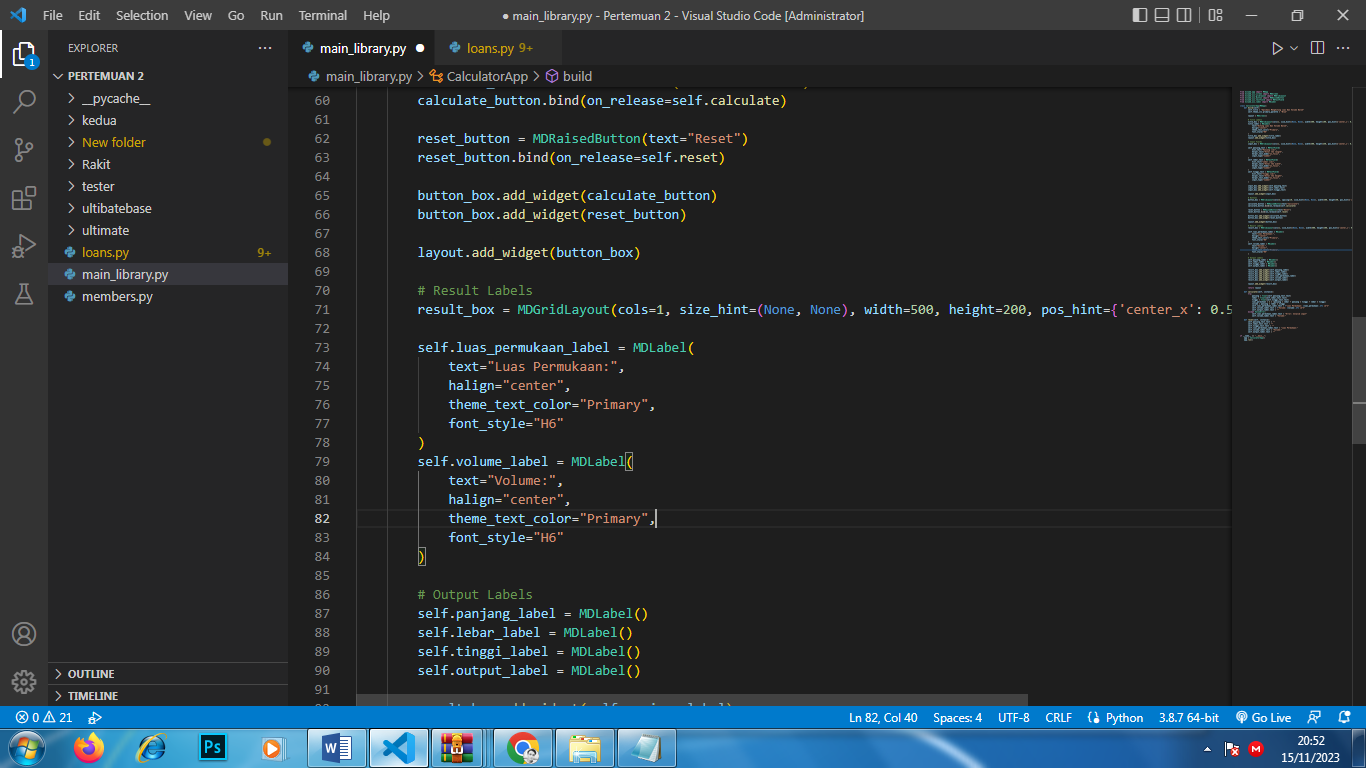
1. **layout.add\_widget(button\_box)**

Menambahkan kotak tombol ke dalam layout utama (`MDScreen`).

Bagian ini bertanggung jawab untuk membuat dan menempatkan tombol "Calculate" dan "Reset" pada antarmuka pengguna. Mari kita bahas lebih rinci:

Dengan kode ini, kita telah membuat bagian tombol pada antarmuka pengguna yang memungkinkan pengguna untuk melakukan perhitungan dan mereset input. Tombol tersebut menggunakan elemen `MDRaisedButton` dari KivyMD dan dihubungkan ke metode `calculate` dan `reset` untuk menangani peristiwa ketika tombol ditekan.

**Result Labes**



**Penjelasan**

1. **result\_box = MDGridLayout(...)**

Membuat kotak layout berbasis grid (`MDGridLayout`) dengan satu kolom. Parameter seperti cols, size\_hint, width, height, dan pos\_hint mengatur properti tata letak dan penempatan kotak label hasil.

1. **self.luas\_permukaan\_label = MDLabel(...)**

Membuat objek MDLabel untuk menampilkan label "Luas Permukaan:". Beberapa parameter seperti text, halign, theme\_text\_color, dan font\_style mengatur properti dan penampilan label.

1. **self.volume\_label = MDLabel(...)**

Membuat objek `MDLabel` untuk menampilkan label "Volume:". Sama seperti langkah sebelumnya, tetapi untuk label volume.

Kedua label tersebut kemudian ditambahkan ke dalam `result\_box` :

result\_box.add\_widget(self.luas\_permukaan\_label)

result\_box.add\_widget(self.volume\_label)

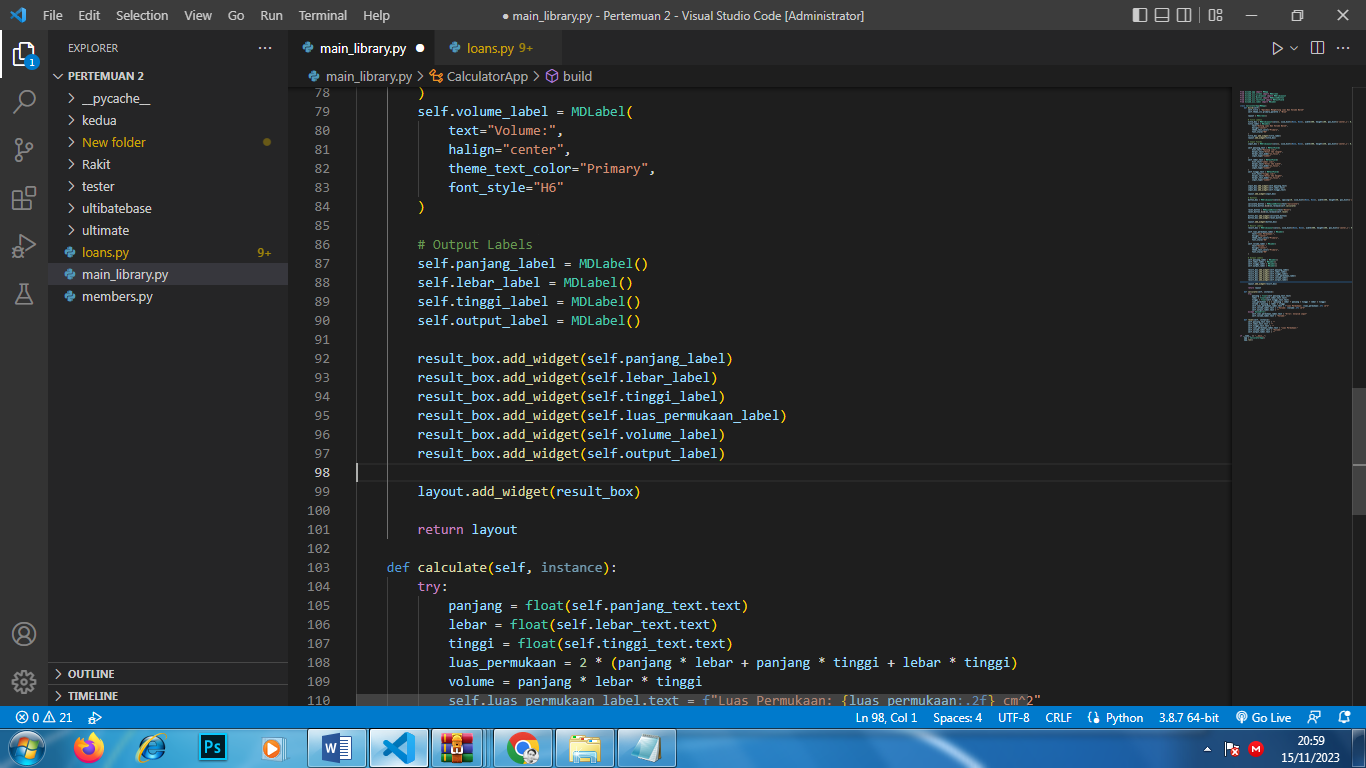
1. **layout.add\_widget(result\_box)**

Menambahkan kotak label hasil ke dalam layout utama (`MDScreen`).

Bagian ini bertanggung jawab untuk membuat dan menempatkan label hasil (label luas permukaan dan label volume) pada antarmuka pengguna. Mari kita bahas lebih rinci:

Dengan kode ini, kita telah membuat bagian label hasil pada antarmuka pengguna yang akan menampilkan informasi luas permukaan dan volume balok setelah perhitungan dilakukan. Labels ini menggunakan elemen `MDLabel` dari KivyMD untuk menampilkan teks dengan format dan gaya tertentu.

**Result Box**



**Penjelasan:**

Bagian ini bertanggung jawab untuk membuat dan menempatkan label-label output pada antarmuka pengguna. Mari kita bahas lebih rinci:

1. **Membuat Label-Label Output :**

self.panjang\_label = MDLabel()

self.lebar\_label = MDLabel()

self.tinggi\_label = MDLabel()

self.output\_label = MDLabel()

Membuat empat objek `MDLabel` untuk menampilkan output terkait panjang, lebar, tinggi, luas permukaan, volume, dan output umum.

1. **Menambahkan Label-Label Output ke dalam `result\_box` :**

result\_box.add\_widget(self.panjang\_label)

result\_box.add\_widget(self.lebar\_label)

result\_box.add\_widget(self.tinggi\_label)

result\_box.add\_widget(self.luas\_permukaan\_label)

result\_box.add\_widget(self.volume\_label)

result\_box.add\_widget(self.output\_label)

Menambahkan label-label output ke dalam kotak hasil (`result\_box`). Ini akan menampilkan nilai-nilai panjang, lebar, tinggi, luas permukaan, volume, dan output

umum setelah perhitungan.

Menambahkan `result\_box` ke dalam Layout Utama (`MDScreen`) :

1. **layout.add\_widget(result\_box)**

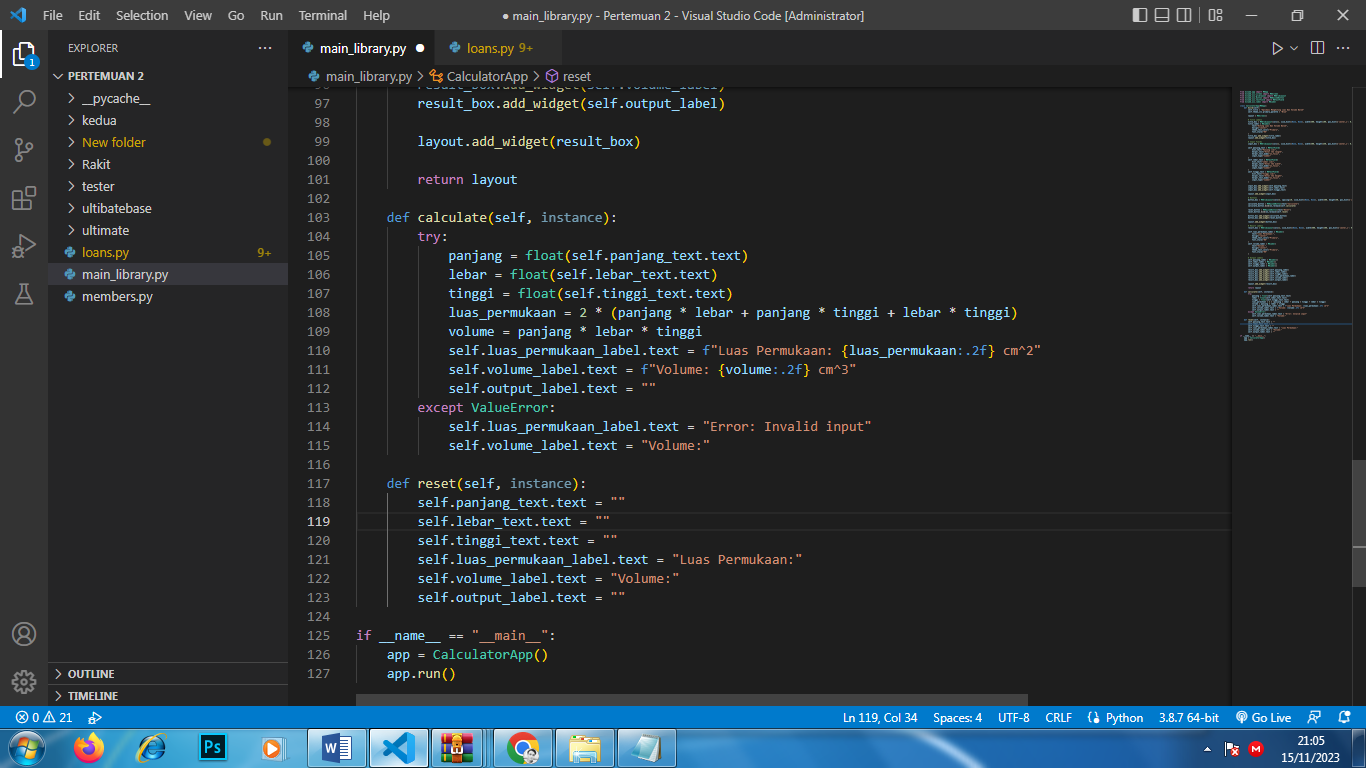
Menambahkan kotak hasil ke dalam layout utama aplikasi (`MDScreen`).

1. **`return layout`**

Mengembalikan layout utama sebagai tampilan aplikasi.

Dengan kode ini, kita telah menambahkan label-label output ke dalam antarmuka pengguna untuk menampilkan hasil perhitungan luas permukaan dan volume balok, serta pesan umum jika ada kesalahan. Label-label ini akan diperbarui setiap kali pengguna menekan tombol "Calculate".

**Deff Calculate**



**Penjelasan:**

1. **calculate(self, instance)** Metode ini dipanggil saat tombol "Calculate" ditekan.
2. **Mencoba Mendapatkan Nilai Input :**

panjang = float(self.panjang\_text.text)

lebar = float(self.lebar\_text.text)

tinggi = float(self.tinggi\_text.text)

Mengambil nilai panjang, lebar, dan tinggi dari input teks dan mengonversikannya ke tipe data float.

1. **Menghitung Luas Permukaan dan Volume**

luas\_permukaan = 2 \* (panjang \* lebar + panjang \* tinggi + lebar \* tinggi)

volume = panjang \* lebar \* tinggi

Menghitung luas permukaan dan volume berdasarkan nilai-nilai panjang, lebar, dan tinggi.

1. **Memperbarui Label-Label Output**

self.luas\_permukaan\_label.text = f"Luas Permukaan: {luas\_permukaan:.2f} cm^2"

self.volume\_label.text = f"Volume: {volume:.2f} cm^3"

self.output\_label.text = ""

Memperbarui label-label output dengan hasil perhitungan. Label luas permukaan dan volume akan menampilkan nilai dengan format tertentu, dan label output umum diatur sebagai string kosong.

1. **Exception Handling (Pengelolaan Kesalahan)**

except ValueError:

self.luas\_permukaan\_label.text = "Error: Invalid input"

self.volume\_label.text = "Volume:"

Jika ada kesalahan dalam mengonversi input ke float (misalnya, pengguna memasukkan teks bukan angka), maka blok `except` akan menangkap kesalahan tersebut dan memperbarui label-label output dengan pesan kesalahan.

1. **reset(self, instance) :** **Metode ini dipanggil saat tombol "Reset" ditekan.**

Mereset Nilai Input dan Label-Label Output :

self.panjang\_text.text = ""

self.lebar\_text.text = ""

self.tinggi\_text.text = ""

self.luas\_permukaan\_label.text = "Luas Permukaan:"

self.volume\_label.text = "Volume:"

self.output\_label.text = ""

Mengatur ulang nilai-nilai input dan label-label output menjadi nilai awal atau string kosong.

1. **Menjalankan Aplikasi**

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

app = CalculatorApp()

app.run()

Bagian ini menjalankan aplikasi saat script dijalankan. Ini membuat instance dari kelas `CalculatorApp` dan menjalankan metode `run()`, yang kemudian memanggil metode `build()` untuk membuat antarmuka pengguna dan memulai aplikasi.

# BAB IV

# HASIL RANCANGAN

# 

# Halaman Beranda

# 

# 2. Halaman Menghitung

# Halaman Error

# BAB V

# Kesimpulan

# Aplikasi ini hadir untuk menyederhanakan perhitungan luas permukaan dan volume balok, memberikan pengguna pengalaman kalkulasi yang efisien. Dengan tujuan memberikan solusi praktis dan sederhana, aplikasi Kalkulator Luas dan Volume Balok diharapkan dapat memenuhi kebutuhan pengguna dalam menangani perhitungan tersebut. BAB VI

# Penutup

# Aplikasi Kalkulator Luas dan Volume Balok menggunakan KivyMD memberikan solusi praktis untuk perhitungan luas permukaan dan volume balok. Dengan antarmuka yang sederhana dan fokus pada kebutuhan dasar, aplikasi ini dirancang untuk memberikan pengalaman kalkulasi yang cepat dan akurat. Kesederhanaan ini menjadikan aplikasi ini sebagai alat yang mudah diakses, tanpa memerlukan pengetahuan matematika tinggi.