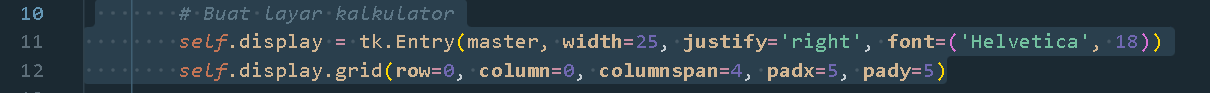
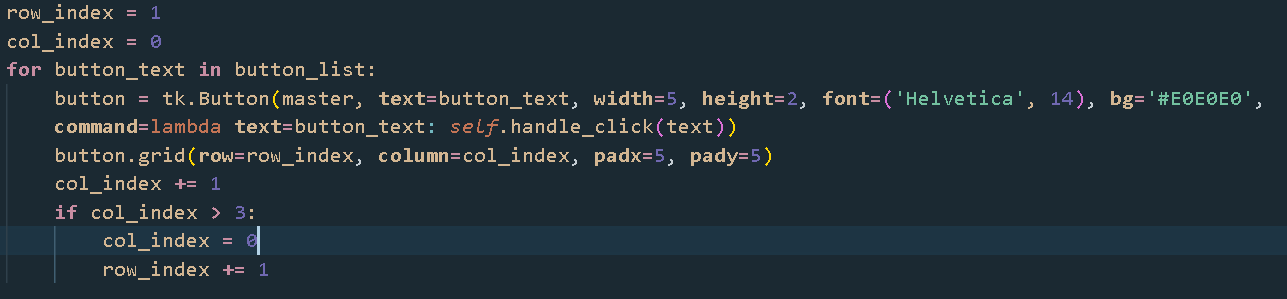
Kode tersebut merupakan program kalkulator sederhana yang ditulis menggunakan library Tkinter pada Python.

* import tkinter as tk digunakan untuk mengimpor library Tkinter dan memberikan alias "tk" agar mudah digunakan dalam kode.
* from tkinter import ttk digunakan untuk mengimpor sub-module ttk (Themed Tkinter) yang merupakan sebuah paket yang menyediakan kelas-kelas widget Tkinter dengan gaya tematik.
* class Calculator: adalah definisi sebuah class Calculator yang akan digunakan untuk membuat objek kalkulator.
* def \_\_init\_\_(self, master): adalah constructor (metode khusus yang dipanggil saat objek diinisialisasi) untuk class Calculator. master adalah argumen yang mewakili jendela utama Tkinter untuk menempatkan widget-widget kalkulator.
* self.master = master digunakan untuk menginisialisasi atribut master dengan argumen master yang diterima pada konstruktor.
* master.title("Hitung-Hitung") digunakan untuk memberikan judul jendela utama menjadi "Hitung-Hitung".
* master.configure(bg='#25282C') digunakan untuk memberikan warna background jendela utama menjadi '#25282C', yaitu warna abu-abu tua.
* Berikutnya, constructor akan membuat objek Entry yang akan digunakan sebagai layar kalkulator. Objek ini akan menampilkan hasil perhitungan kalkulator dan ditempatkan pada baris 0 dan kolom 0 dari grid.
* Selanjutnya, kode akan membuat tombol-tombol untuk angka, operator, dan tombol penghapusan. Tombol-tombol ini akan diatur pada posisi yang diinginkan pada grid menggunakan metode grid().
* Terakhir, constructor akan mendefinisikan tiga metode yaitu handle\_click(), handle\_delete(), dan handle\_equal(), yang masing-masing digunakan untuk mengatur aksi ketika tombol di klik.



Kode tersebut membuat sebuah Entry widget dari tkinter dengan lebar sebesar 25 karakter, tulisan akan diatur rata kanan dengan jenis huruf Helvetica dan ukuran 18. Widget ini kemudian ditempatkan pada baris pertama (row=0) dan kolom pertama (column=0) dari grid layout pada master window. columnspan=4 mengatur widget ini akan memenuhi empat kolom grid, dan padx=5 dan pady=5 mengatur jarak antara widget dengan tepi grid layout. Widget ini akan menampilkan dan memungkinkan pengguna untuk memasukkan masukan yang akan dihitung.

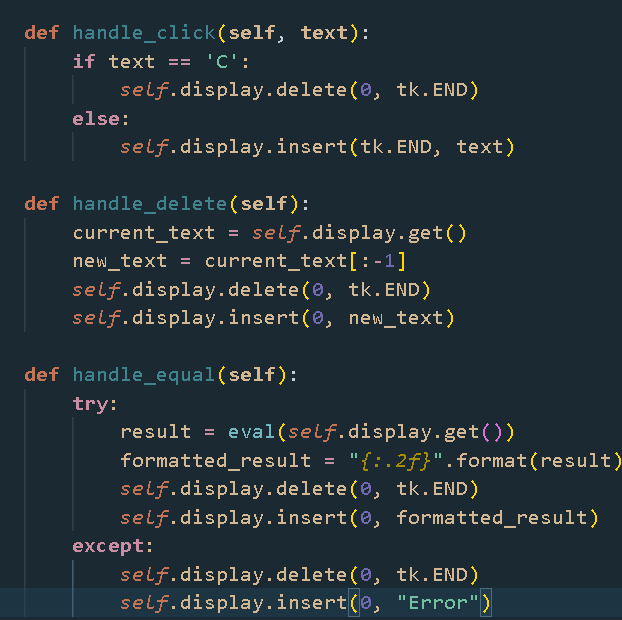


Kode ini digunakan untuk membuat tombol-tombol angka dan operator matematika pada kalkulator. Pertama, kode menentukan nilai awal row\_index dan col\_index sebagai 1 dan 0. Kemudian, untuk setiap teks tombol dalam daftar button\_list, kode membuat tombol menggunakan teks tersebut dengan menggunakan tk.Button. Ukuran tombol ditentukan dengan width=5 dan height=2, sedangkan font dan warna latar belakang tombol ditentukan dengan font dan bg. Command tombol ditentukan dengan menggunakan lambda function agar setiap tombol dapat memanggil method handle\_click() dengan argumen yang sesuai.

Setiap tombol kemudian ditempatkan pada grid dengan menggunakan grid() method dengan menentukan baris (row\_index) dan kolom (col\_index) untuk menentukan posisi tombol pada kalkulator. Setelah tombol ditambahkan ke grid, kolom saat ini (col\_index) ditingkatkan dengan 1. Jika kolom saat ini lebih besar dari 3, artinya telah mencapai kolom terakhir, maka kolom saat ini diubah kembali menjadi 0 dan baris saat ini (row\_index) ditingkatkan dengan 1, sehingga tombol-tombol selanjutnya akan ditempatkan pada baris selanjutnya.



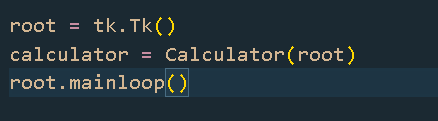
Kode tersebut membuat 2 tombol di bawah semua tombol angka pada kalkulator. Tombol pertama adalah tombol 'Del' untuk menghapus satu digit terakhir dari tampilan kalkulator dan memiliki warna latar belakang merah tua dan warna teks putih. Tombol kedua adalah tombol sama dengan '=' untuk mengevaluasi persamaan pada tampilan kalkulator dan memiliki warna latar belakang hijau dan warna teks putih. Keduanya memiliki fungsi yang berbeda, yaitu 'handle\_delete' dan 'handle\_equal', masing-masing digunakan saat tombol ditekan.



Dalam fungsi handle\_click(self, text), program akan memeriksa apakah teks yang diterima sama dengan C. Jika iya, maka fungsi akan menghapus semua karakter pada self.display dengan menggunakan metode delete yang dimulai dari indeks 0 hingga indeks terakhir (tk.END). Jika teks yang diterima bukan C, maka fungsi akan menambahkan teks ke dalam self.display di akhir teks yang sudah ada dengan menggunakan metode insert yang dimulai dari indeks terakhir (tk.END).

Dalam fungsi handle\_delete(self), program akan mengambil teks yang ada di dalam self.display menggunakan metode get(), lalu menghapus karakter terakhir dari teks tersebut dengan mengambil seluruh karakter teks kecuali karakter terakhir menggunakan slicing ([:-1]). Kemudian, teks yang sudah dimodifikasi akan dimasukkan kembali ke dalam self.display dengan menggunakan metode delete dan insert seperti pada fungsi handle\_click(self, text).

Dalam fungsi handle\_equal(self), program akan mencoba melakukan evaluasi ekspresi matematika yang ada pada self.display menggunakan fungsi bawaan Python eval(). Jika evaluasi berhasil dilakukan, maka hasil evaluasi akan diformat dengan 2 angka di belakang koma menggunakan fungsi format() dan kemudian dimasukkan ke dalam self.display dengan menggunakan metode delete dan insert seperti pada fungsi handle\_click(self, text). Jika evaluasi gagal dilakukan, maka teks "Error" akan dimasukkan ke dalam self.display dengan menggunakan metode delete dan insert.



Kode ini membuat objek Tk() dari modul tkinter yang merepresentasikan main window dari aplikasi. Kemudian objek Calculator dibuat dengan parameter root (yaitu main window tersebut), yang akan menempatkan dan mengatur seluruh elemen yang dibutuhkan dalam kalkulator. Selanjutnya, root.mainloop() dipanggil untuk menampilkan main window dalam mode event loop, sehingga aplikasi dapat menerima input dari pengguna dan memberikan respons terhadap input tersebut. Proses ini akan terus berjalan sampai pengguna menutup window aplikasi.