**TUGAS PEMROGRAMAN BERORIENTASI OBJEK**

***Aplikasi Konversi Suhu Berbasis Windows Menggunakan Python-Tkinter***

Dosen Pengampu : Freddy Wicaksono,M.Kom



Disusun Oleh:

RULASTRI

220511071

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS MUHAMMADIYAH CIREBON**

**2023**

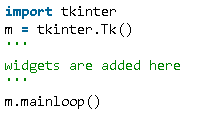
**Ada 2 metode utama yang user harus tau jika ingin membangun GUI dengan aplikasi Phyton.**

**1. Tk(screenName=None,  baseName=None,  className=’Tk’,  useTk=1) :**Untuk mengubah nama window, kamu dapat mengubah className sesuai yang diinginkan. Kode dasar yang digunakan untuk membuat window utama aplikasi adalah :

***m=tkinter.Tk() dimana m merupakan nama window yang diinginkan***

**2. Mainloop () :**Mainloop ()digunakan ketika aplikasi kamu siap dijalankan. mainloop () adalah loop tak terbatas yang digunakan untuk menjalankan aplikasi, tunggu sampai suatu peristiwa terjadi dan proseskan peristiwa selama jendela tidak ditutup.

***m.mainloop()***



tkinter juga menawarkan akses ke konfigurasi geometrik widget yang dapat mengatur widget di parent window. Ada tiga class utama dalam Geometri Class manager :

1. method pack ():  mengatur widget di blok sebelum menempatkan di parent widget.
2. method grid ():  mengatur widget di grid (struktur seperti tabel) sebelum menempatkan di parent widget.
3. method place ():  mengatur widget dengan menempatkannya pada posisi tertentu yang diarahkan oleh programmer.

**Desain GUI TKInter**

Di sini saya menggunakan beberapa widget seperti Label, Button, EntryField (Bidang masukan), dan OptionsMenu (tarik-turun).

Widget ini digunakan sebagai berikut.

* Sebuah EntryField di mana seseorang dapat memasukkan nilai apa pun dan
* OptionMenu untuk memilih unit nilai yang dimasukkan.
* A Button ketika ditekan, memanggil fungsi untuk menghitung dua satuan suhu lainnya.
* Dua Labels untuk menunjukkan hasilnya.

Pertama saya harus mengimpor dan menginisialisasi TKInter untuk menggunakannya.

import tkinter as tk

root = tk.Tk()

Kemudian saya akan menginisialisasi a Label dan `EntryField` dan juga mengatur posisinya menggunakan grid opsi tata letak.

input\_label = tk.Label(root, text="Enter temperature")

input\_entry = tk.Entry(root, textvariable=numberInput)

input\_label.grid(row=0)

input\_entry.grid(row=0, column=1)

Di sini saya perlu mengatur 'numberInput' ke sebuah string.

masukan angka = tk.StringVar()

Selanjutnya, saya akan menggunakan OptionMenu (menu drop-down) untuk menampilkan suhu Celsius, Fahrenheit, dan Kelvin dan juga mengatur nilai default dropdown ke Celsius dan mengatur posisi OptionMenu

dropDownList = ["Celsius", "Fahrenheit", "Kelvin"]

dropdown = tk.OptionMenu(root, var, \*dropDownList, command=store\_temp)

var.set(dropDownList[0])

dropdown.grid(row=0, column=3)

Saya juga perlu mengaturnya var dalam cuplikan di atas ke string.

var = tk.StringVar()

**PilihanMenu Item yang Dipilih**

Di sini saya perlu mengonversi suhu masukan ke dua format suhu lain berdasarkan format suhu masukan apakah itu satuan Celcius, Fahrenheit, atau Kelvin, dan menyimpannya di suatu tempat sehingga saya dapat menggunakannya untuk konversi.

Jika Anda memperhatikan cuplikan di atas saya menggunakan a command parameter untuk OptionMenu, ini command tidak lain hanyalah ia memanggil suatu fungsi, ini dia yang memanggil store\_temp ketika sebuah OptionMenu(dropdown) nilainya diubah.

def toko\_temp(sel\_temp):

    global tempVal

    tempVal = sel\_temp

Jadi di sini saya menyimpan nilai suhu yang dipilih tempVal variabel yang dideklarasikan secara global. Untuk mendeklarasikan variabel secara global, saya hanya menggunakan variabel dengan nilai tertentu.

tempVal = "Celsius"

Jadi sekarang setiap kali drop-down diubah, nilainya disimpan tempVal dan dapat diakses di mana saja di aplikasi kami.

**Konverter suhu**

Jadi ketika pengguna memasukkan beberapa nilai dan memilih suhu dari drop-down, hal berikutnya adalah menekan tombol yang menunjukkan hasil yang diinginkan.

Saya akan menginisialisasi tombol dengan perintah untuk memanggil fungsi konversi.

result\_button = tk.Button(root, text="Convert", command=call\_convert)

result\_button.grid(row=0, columnspan=4)

Saya akan memberikan 2 label untuk menunjukkan hasil dan nilai input yang dimasukkan di `Entri`. Di sini saya menggunakan partial dari functools untuk meneruskan parameter ke fungsi call\_convert

Jadi sebelum saya menelepon call\_convert Saya akan menempatkan baris berikut.

call\_convert = partial(call\_convert, result\_label1, result\_label2, numberInput)

dan tentu saja, saya perlu mengimpor `partial`.

dari alat fungsi impor sebagian

Jadi logikanya ada di sini call\_convert metode.

**Konversi**

Pertama saya akan mendapatkan nilai yang dimasukkan, kemudian memeriksa nilai drop-down dan kemudian melakukan konversi dan mengembalikan hasilnya ke label.

def call\_convert(rlabel1, rlabe12, inputn):

3 parameter pada baris di atas adalah 2 label untuk mengatur hasil dan nilai yang dimasukkan pada kolom Entri.

Jadi Pertama saya akan mendapatkan nilai input.

item = masukann.mendapatkan()

Sekarang saya akan memeriksa nilai dropdown, karena saya sebelumnya menyimpan nilai ini dalam variabel global tempVal Jadi sekarang kita akan memeriksa apakah Celsius, Fahrenheit, atau Kelvin dan melakukan konversi berdasarkan nilai ini.

#for Celsius

if tempVal == 'Celsius':

  ...

  ...

#for Fahrenheit

if tempVal == 'Fahrenheit':

  ...

  ...

#for Kelvin

if tempVal == 'Kelvin':

  ...

  ...

Jadi jika nilainya Celsius

Saya akan menggunakan beberapa perhitungan untuk mengonversi dan mengatur hasil tersebut ke label yang sesuai.

    if tempVal == 'Celsius':

        f = float((float(tem) \* 9 / 5) + 32)

        k = float((float(tem) + 273.15))

        rlabel1.config(text="%f Fahrenheit" % f)

        rlabe12.config(text="%f Kelvin" % k)

Berikut variabelnya fDan k menyimpan hasil konversi berdasarkan beberapa rumus yang akan dikonversi Celsius ke Fahrenheit Dan Kelvin

Demikian pula untuk Fahrenheit Dan Kelvin,

if tempVal == 'Fahrenheit':

    c = float((float(tem) - 32) \* 5 / 9)

    k = c + 273

    rlabel1.config(text="%f Celsius" % c)

    rlabe12.config(text="%f Kelvin" % k)

if tempVal == 'Kelvin':

    c = float((float(tem) - 273.15))

    f = float((float(tem) - 273.15) \* 1.8000 + 32.00)

    rlabel1.config(text="%f Celsius" % c)

    rlabe12.config(text="%f Fahrenheit" % f)

Ini mengubah suhu masukan ke dua suhu lain yang diperlukan.

**Sentuhan akhir**

Di sini saya akan menunjukkan kepada Anda Bagaimana kita mengatur ukuran jendela, mengatur warna, latar belakang, dan warna latar depan ke widget, dan juga mengatur jendela ke ukuran tetap dan semuanya.

* Atur ukuran jendela

root.geometry('400x150+100+200')

* Tetapkan judul aplikasi

root.title('Temperature Converter')

* Mengatur warna latar belakang dan latar depan

root.configure(background='#808000')

input\_label = tk.Label(root, text="Enter temperature", background='#808000', foreground="#FFFFFF")

dropdown.config(background='#808000', foreground="#FFFFFF")

dropdown["menu"].config(background='#808000', foreground="#FFFFFF")

* Atur jendela ke ukuran tetap

root.resizable(width=False, height=False)

**Program**

\_\_author\_\_ = 'Rulastri 220511071 TI22L'

import tkinter as tk

from functools import partial

# global variable

tempVal = "Celsius"

# getting drop down value

def store\_temp(sel\_temp):

global tempVal

tempVal = sel\_temp

# the main conversion

def call\_convert(rlabel1, rlabe12, inputn):

tem = inputn.get()

if tempVal == 'Celsius':

f = float((float(tem) \* 9 / 5) + 32)

k = float((float(tem) + 273.15))

rlabel1.config(text="%f Fahrenheit" % f)

rlabe12.config(text="%f Kelvin" % k)

if tempVal == 'Fahrenheit':

c = float((float(tem) - 32) \* 5 / 9)

k = c + 273

rlabel1.config(text="%f Celsius" % c)

rlabe12.config(text="%f Kelvin" % k)

if tempVal == 'Kelvin':

c = float((float(tem) - 273.15))

f = float((float(tem) - 273.15) \* 1.8000 + 32.00)

rlabel1.config(text="%f Celsius" % c)

rlabe12.config(text="%f Fahrenheit" % f)

return

# app window configuration and UI

root = tk.Tk()

root.geometry('400x150+100+200')

root.title('Temperature Converter')

root.configure(background='#808000')

root.resizable(width=False, height=False)

root.grid\_columnconfigure(1, weight=1)

root.grid\_rowconfigure(0, weight=1)

numberInput = tk.StringVar()

var = tk.StringVar()

# label and entry field

input\_label = tk.Label(root, text="Enter temperature", background='#808000', foreground="#FFFFFF")

input\_entry = tk.Entry(root, textvariable=numberInput)

input\_label.grid(row=1)

input\_entry.grid(row=1, column=1)

# result label's for showing the other two temperatures

result\_label1 = tk.Label(root, background='#808000', foreground="#FFFFFF")

result\_label1.grid(row=3, columnspan=4)

result\_label2 = tk.Label(root, background='#808000', foreground="#FFFFFF")

result\_label2.grid(row=4, columnspan=4)

# drop down initalization and setup

dropDownList = ["Celsius", "Fahrenheit", "Kelvin"]

dropdown = tk.OptionMenu(root, var, \*dropDownList, command=store\_temp)

var.set(dropDownList[0])

dropdown.grid(row=1, column=3)

dropdown.config(background='#808000', foreground="#FFFFFF")

dropdown["menu"].config(background='#808000', foreground="#FFFFFF")

# button click

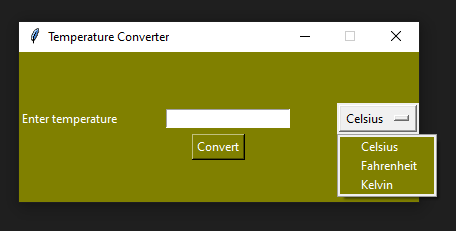
call\_convert = partial(call\_convert, result\_label1, result\_label2, numberInput)

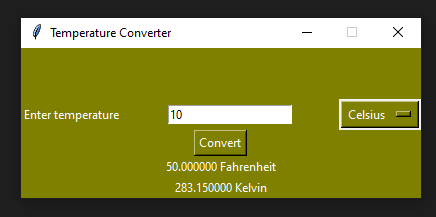
result\_button = tk.Button(root, text="Convert", command=call\_convert, background='#808000', foreground="#FFFFFF")

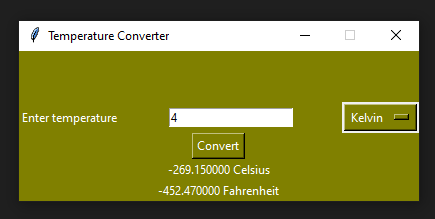
result\_button.grid(row=2, columnspan=4)

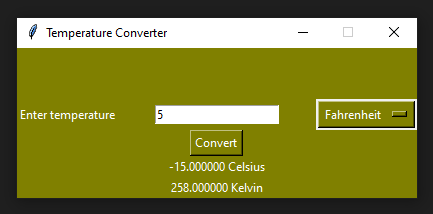
root.mainloop()

**Program Screenshots**









**Kesimpulan**

TkInter adalah pilihan yang menarik untuk kerangka GUI Python karena dibangun ke dalam pustaka standar python, dan relatif mudah untuk membuat aplikasi dengan kerangka ini. Salah satu Toolkit yang ada pada bahasa program Phyton. Phyton dengan tkinter adalah kombinasi yang tepat untuk membuat GUI dengan cepat dan mudah.