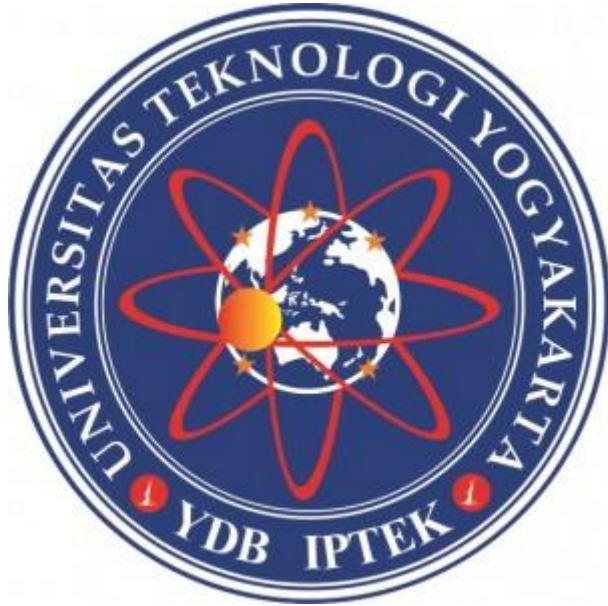


TUGAS PBOP
“IMPLEMETASI GUI PADA STUDY KASUS”



Nama :

1. Yoga rusdi hantoro_5230411266

FAKULTAS SAINS DAN TEKNOLOGI
UNIV TEKNOLOGI YOGYAKARTA
2024/2025

KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas limpahan rahmatNya sehingga penulis dapat mengatasi segala rintangan dan kesulitan sampai akhirnya dapat menyelesaikan laporan ini, penulis tidak lupa mengucapkan terimakasih atas bantuan dan kerjasama dari berbagai pihak. Ucapan terima kasih ini penulis haturkan Semoga amal dan kebaikan saudara-saudara yang mendapatkan balasan yang setimpal dari Tuhan Yang Maha Esa. Penulis menyadari segala kekurangan dan ketidaksempurnaan laporan ini, dengan segala kerendahan hati penulis dengan senang hati menerima kritik dan saran yang sifatnya membangun guna perbaikan dan kesempurnaan laporan ini. Semoga laporan ini dapat bermanfaat bagi pihak yang berkepentingan.

DAFTAR ISI

Kata pengantar

Daftar isi

Pendahuluan

Latar belakang

Tujuan

Bab II

Rangkuman

- Activity diagram
- penjelasan

penutup

PENDAHULUAN

Latar Belakang

GUI dapat membuat interface yang menarik dan memudahkan pengguna saat menggunakannya. Beberapa hal yang dapat dilakukan dengan GUI pada Python, di antaranya: Membuat jendela dan kotak dialog, Membangun GUI untuk aplikasi desktop, Menambahkan GUI ke program baris perintah, Membuat widget khusus, Membuat prototipe GUI.

Paket [tkinter](#) (“antarmuka Tk”) adalah antarmuka Python standar untuk toolkit GUI Tcl/Tk. Baik Tk maupun Tk [tkinter](#) tersedia di sebagian besar platform Unix, termasuk macOS, serta di sistem Windows.

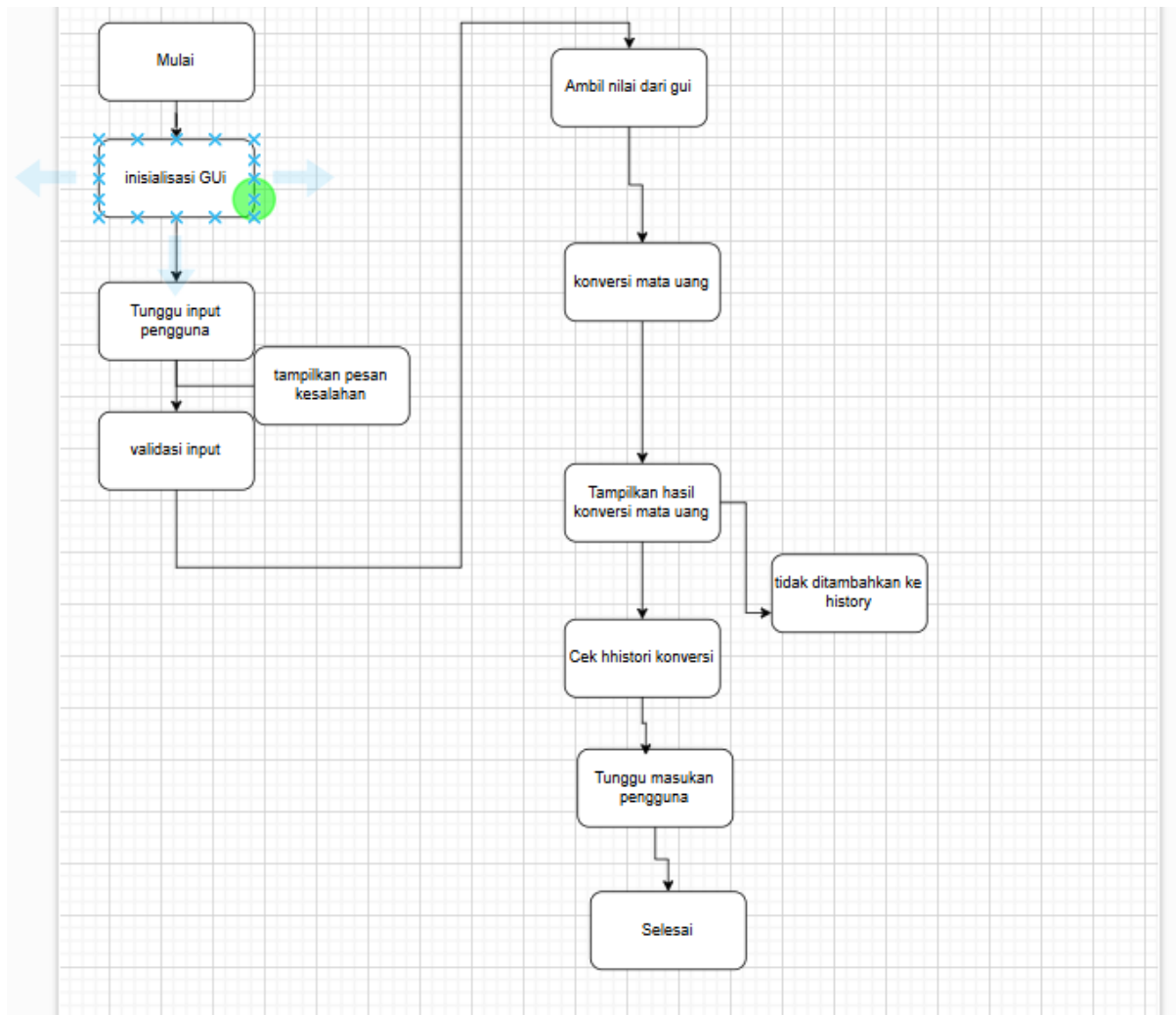
Menjalankannya dari baris perintah akan membuka jendela yang menunjukkan antarmuka Tk yang sederhana, yang akan memberi tahu Anda bahwa Tcl/Tk telah terinstal dengan benar di sistem Anda, dan juga menunjukkan versi Tcl/Tk yang terinstal, sehingga Anda dapat membaca dokumentasi Tcl/Tk yang khusus untuk versi tersebut.
`python -m tkinter`[tkinter](#)

Tujuan

Tujuan dibuatnya laporan ini adalah untuk menuntaskan tugas Pemrograman Berbasis objek pertemuan 9. dan untuk mendalami dan mempelajari implementasi GUI pada study kasus

BAB ii

Activity Diagram



Penjelasan

1. **Mulai:** Program dimulai.
2. **Inisialisasi GUI:** Membuat antarmuka pengguna dengan elemen-elemen seperti label, entry, dan combobox.
3. **Tunggu Input Pengguna:** Menunggu pengguna untuk memasukkan jumlah dan memilih mata uang input serta mata uang output.
4. **Validasi Input:**

? Jika input jumlah valid (angka), lanjut ke langkah berikutnya.

? Jika tidak valid, tampilkan pesan kesalahan dan kembali ke langkah 3.

? **Ambil Nilai dari GUI:** Mengambil nilai jumlah dan mata uang yang dipilih dari combobox.

? **Konversi Mata Uang:**

? Berdasarkan mata uang output yang dipilih, maka dilakukan konversi konversi dari IDR ke mata uang yang dipilih.

7. **Tampilkan Hasil Konversi:** Tampilkan hasil konversi di label hasil.

? **Cek Histori Konversi:**

? Periksa apakah hasil konversi sudah ada di histori.

? Jika sudah ada, tidak tambahkan ke histori.

? Jika belum ada, tambahkan hasil konversi ke tabel histori.

? **Tunggu Input Pengguna:** Kembali ke langkah 3 untuk menunggu input pengguna berikutnya.

? **Selesai:** Program berakhir jika ditutup oleh pengguna.

Program

```
import tkinter as tk
from tkinter import ttk

class CurrencyConverter:
    def __init__(self, root):
        self.root = root
        self.root.title("Konversi Mata Uang")
        self.root.geometry("500x400")

        self.label_judul = ttk.Label(self.root, text="Konversi Mata Uang",
font=("Arial", 16))
        self.label_judul.pack(pady=10)

        self.label_jumlah = ttk.Label(self.root, text="Jumlah:")
        self.label_jumlah.pack(pady=5)
        self.entry_jumlah = ttk.Entry(self.root, width=30)
        self.entry_jumlah.pack(pady=5)
```

```

        self.label_mata_uang_input = ttk.Label(self.root, text="Pilih Mata
Uang untuk Jumlah:")
        self.label_mata_uang_input.pack(pady=5)
        self.combo_mata_uang_input = ttk.Combobox(self.root, values=["IDR"],
state="readonly")
        self.combo_mata_uang_input.pack(pady=5)
        self.combo_mata_uang_input.current(0)

        self.label_mata_uang_output = ttk.Label(self.root, text="Pilih Mata
Uang yang Akan Dikeluarkan:")
        self.label_mata_uang_output.pack(pady=5)
        self.combo_mata_uang_output = ttk.Combobox(self.root, values=["USD",
"EUR", "GBP", "JPY", "AUD", "CAD", "CHF", "CNY", "SGD", "NZD"],
state="readonly")
        self.combo_mata_uang_output.pack(pady=5)
        self.combo_mata_uang_output.current(0)

        self.tombol_konversi = ttk.Button(self.root, text="Konversi",
command=self.konversi)
        self.tombol_konversi.pack(pady=10)

        self.label_hasil = ttk.Label(self.root, text="", font=("Arial", 12))
        self.label_hasil.pack(pady=10)

        self.tree = ttk.Treeview(self.root, columns=("Jumlah", "Mata Uang",
"Hasil"), show='headings')
        self.tree.heading("Jumlah", text="Jumlah")
        self.tree.heading("Mata Uang", text="Mata Uang")
        self.tree.heading("Hasil", text="Hasil")
        self.tree.pack(pady=10, fill=tk.BOTH, expand=True)

    def konversi(self):
        jumlah = self.entry_jumlah.get()
        mata_uang_output = self.combo_mata_uang_output.get()

        # Validasi input
        if not jumlah.replace('.', '', 1).isdigit(): # Memungkinkan angka
desimal
            self.label_hasil.config(text="Masukkan jumlah yang valid (hanya
angka).")
            return

        jumlah = float(jumlah)

        if mata_uang_output == "USD":
            hasil = jumlah / 15000
        elif mata_uang_output == "EUR":

```

```

        hasil = jumlah / 16000
    elif mata_uang_output == "GBP":
        hasil = jumlah / 20000
    elif mata_uang_output == "JPY":
        hasil = jumlah / 100
    elif mata_uang_output == "AUD":
        hasil = jumlah / 11000
    elif mata_uang_output == "CAD":
        hasil = jumlah / 12000
    elif mata_uang_output == "CHF":
        hasil = jumlah / 14000
    elif mata_uang_output == "CNY":
        hasil = jumlah / 2300
    elif mata_uang_output == "SGD":
        hasil = jumlah / 11000
    elif mata_uang_output == "NZD":
        hasil = jumlah / 11000

    self.label_hasil.config(text=f"{jumlah} IDR = {hasil:.2f}"
{mata_uang_output}")

    existing_items = self.tree.get_children()
    already_exists = False
    for item in existing_items:
        if self.tree.item(item)['values'] == (jumlah, "IDR", f"{hasil:.2f}"
{mata_uang_output}"):
            already_exists = True
            break

    if not already_exists:
        self.tree.insert("", "end", values=(jumlah, "IDR", f"{hasil:.2f}"
{mata_uang_output}))

def main():
    root = tk.Tk()
    app = CurrencyConverter(root)
    root.mainloop()

if __name__ == "__main__":
    main()

```