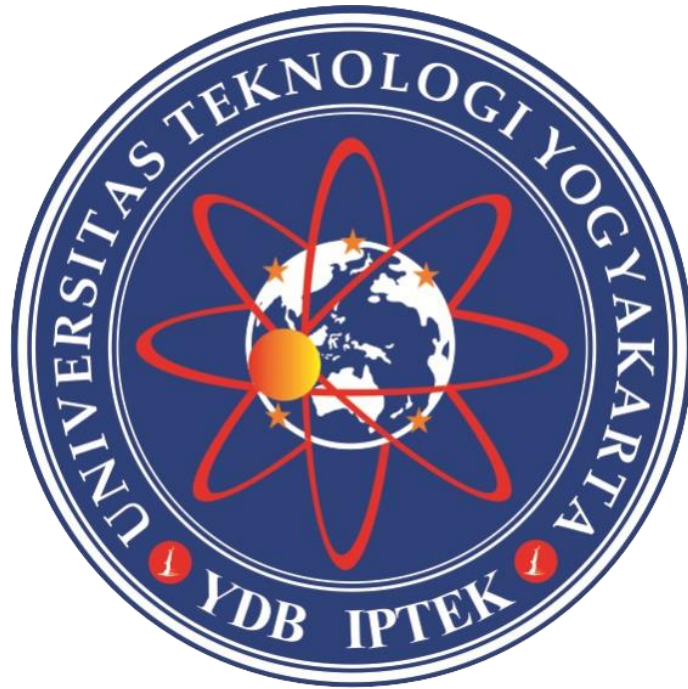


Laporan Implementasi GUI pada Stopwatch



Disusun oleh:

Nama : Taufik Yoga Pratama
NPM : 5230411269
Kelas : VIII

PROGRAM STUDI INFORMATIKA
UNIVERSITAS TEKNOLOGI YOGYAKARTA

2023/2024

Daftar Isi

HALAMAN JUDUL	i
DAFTAR ISI	ii
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang.....	1
1.2. Rumusan Masalah.....	1
1.3. Tujuan.....	1
BAB II PEMBAHASAN	2
2.1. Activity Diagram	2
2.2. Codingan	2
2.3. Penjelasan.....	3
2.4. Dokumentasi.....	4
BAB III PENUTUP	5
3.1. Kesimpulan.....	5

BAB I

PENDAHULUAN

1.1. Latar Belakang

Stopwatch adalah alat yang digunakan untuk mengukur waktu secara tepat dalam hitungan detik. Aplikasi stopwatch ini dikembangkan untuk memahami cara membuat aplikasi berbasis Graphical User Interface (GUI) menggunakan Python, khususnya dengan library Tkinter. Tkinter adalah salah satu library Python yang sering digunakan untuk membuat aplikasi desktop dengan antarmuka grafis yang sederhana namun fungsional. Dalam dunia pemrograman, membuat aplikasi sederhana seperti stopwatch dapat menjadi langkah awal yang baik dalam mempelajari pembuatan aplikasi GUI dan interaksi pengguna dengan aplikasi.

1.2. Rumusan Masalah

1. Bagaimana cara membuat aplikasi stopwatch sederhana menggunakan Tkinter?
2. Bagaimana cara mengatur waktu yang berjalan dan menghentikan serta mereset waktu pada aplikasi?
3. Bagaimana cara menangani interaksi pengguna melalui tombol pada aplikasi GUI?

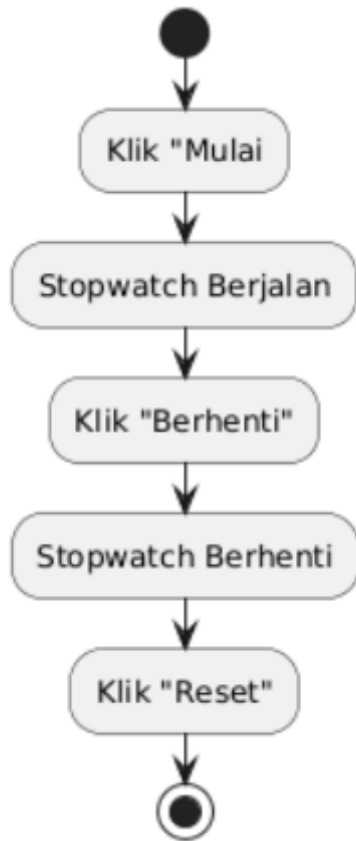
1.3. Tujuan

1. Membuat aplikasi stopwatch sederhana dengan menggunakan Python dan Tkinter.
2. Memahami dasar-dasar pembuatan antarmuka grafis dan pengelolaan event dalam Tkinter.
3. Menyediakan aplikasi yang memungkinkan pengguna untuk mengatur, memulai, menghentikan, dan mereset waktu dengan mudah.

BAB II

PEMBAHASAN

2.1. Activity Diagram



2.2. Codingan

```
import tkinter as tk
from tkinter import ttk

class AplikasiStopwatch:
    def __init__(self, root):

        self.root = root
        self.root.title("Stopwatch")
        self.root.geometry("350x200")

        self.waktu_berjalan = 0
        self.sedang_berjalan = False

        self.label_waktu = tk.Label(self.root, text="0:00:00", font=("Helvetica", 30))
        self.label_waktu.pack(pady=20)

        bingkai_tombol = tk.Frame(self.root)
        bingkai_tombol.pack(pady=20)

        self.tombol_mulai = ttk.Button(bingkai_tombol, text="Mulai", command=self.mulai)
```

```

self.tombol_mulai.grid(row=0, column=0, padx=10)

self.tombol_berhenti = ttk.Button(bingkai_tombol, text="Berhenti",
command=self.berhenti)
self.tombol_berhenti.grid(row=0, column=1, padx=10)

self.tombol_reset = ttk.Button(bingkai_tombol, text="Reset", command=self.reset)
self.tombol_reset.grid(row=0, column=2, padx=10)

def perbarui_waktu(self):
    if self.sedang_berjalan:
        self.waktu_berjalan += 1
        menit, detik = divmod(self.waktu_berjalan, 60)
        jam, menit = divmod(menit, 60)
        self.label_waktu.config(text=f"{jam}:{menit:02}:{detik:02}")
        self.root.after(1000, self.perbarui_waktu)

def mulai(self):
    if not self.sedang_berjalan:
        self.sedang_berjalan = True
        self.perbarui_waktu()

def berhenti(self):
    self.sedang_berjalan = False

def reset(self):
    self.sedang_berjalan = False
    self.waktu_berjalan = 0
    self.label_waktu.config(text="0:00:00")

if __name__ == "__main__":
    root = tk.Tk()
    aplikasi = AplikasiStopwatch(root)
    root.mainloop()

```

2.3. Penjelasan

2.3.1. Inisialisasi Program (__init__):

- self.root: Membuat jendela utama dengan ukuran 350x200 piksel.
- self.waktu_berjalan: Variabel untuk menyimpan waktu yang telah berjalan (dalam detik).
- self.sedang_berjalan: Status apakah stopwatch sedang berjalan (True/False).
- label_waktu: Label untuk menampilkan waktu dalam format HH:MM:SS.
- Tombol: Tiga tombol untuk Mulai, Berhenti, dan Reset diletakkan dalam bingkai_tombol.

2.3.2. Fungsi perbarui_waktu():

- Menambahkan waktu setiap 1 detik saat stopwatch berjalan.
- Memformat waktu dari detik menjadi jam, menit, dan detik.
- Memperbarui tampilan label waktu setiap detik menggunakan `after`.

2.3.3. Fungsi `mulai()`:

- Mengatur `self.sedang_berjalan` menjadi `True`.
- Memulai stopwatch dengan memanggil `perbarui_waktu()`.

2.3.4. Fungsi `berhenti()`:

- Menghentikan stopwatch dengan mengatur `self.sedang_berjalan` menjadi `False`.

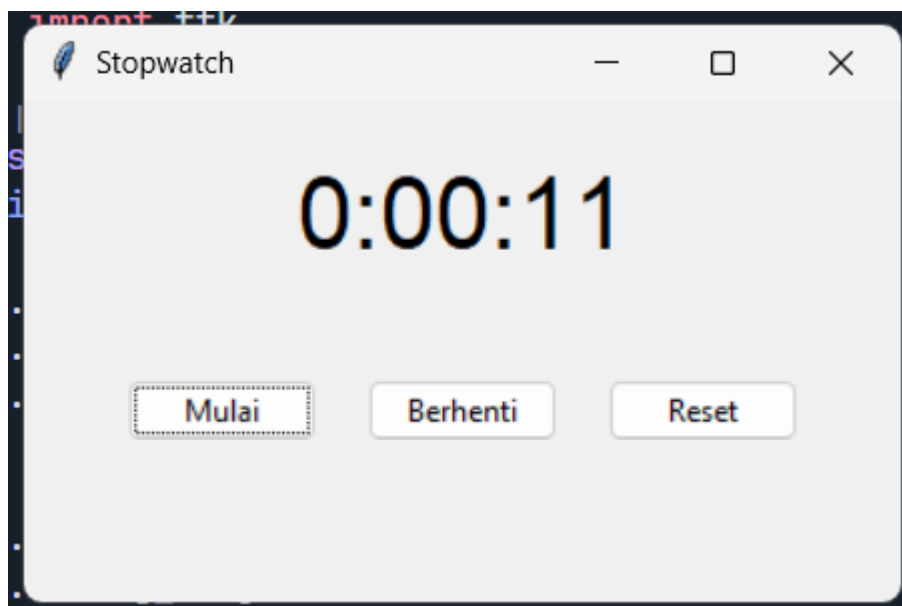
2.3.5. Fungsi `reset()`:

- Menghentikan stopwatch.
- Mengatur ulang `self.waktu_berjalan` ke 0 dan menampilkan waktu awal di `label_waktu`.

2.3.6. Bagian Utama (`if __name__ == "__main__":`):

- Membuat jendela utama (`root`) dan menjalankan aplikasi stopwatch dengan `root.mainloop()`.

2.4. Dokumentasi



BAB III

PENUTUP

3.1. Kesimpulan

Aplikasi stopwatch yang dibuat menggunakan Tkinter ini berhasil mengimplementasikan fungsi dasar stopwatch seperti memulai, menghentikan, dan mereset waktu. Program ini mengajarkan dasar-dasar pemrograman GUI dengan Tkinter, termasuk penggunaan widget seperti label dan tombol, serta pengelolaan event dengan metode `after`. Aplikasi ini juga menunjukkan bagaimana mengatur dan memperbarui waktu dalam aplikasi berbasis grafis, memberikan dasar yang solid untuk pengembangan aplikasi GUI lebih lanjut.

Secara keseluruhan, aplikasi stopwatch ini merupakan contoh aplikasi sederhana yang efektif untuk memahami konsep dasar dalam pembuatan aplikasi desktop menggunakan Tkinter.