

LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK (PBO) – Tugas 5



Disusun Oleh

Ahmad Ali Mukti 123140155

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SUMATERA

2025

Soal nomor 1

Deskripsi Tantangan

Buatlah aplikasi GUI dengan Tkinter yang memiliki fitur berikut:

1. Input Catatan:

- Pengguna dapat memasukkan judul catatan di **Entry**.
- Pengguna dapat menulis isi catatan di **Text** (area teks multi-baris).
- Tombol "Tambah Catatan" untuk menyimpan catatan ke daftar.

2. Tampilan Daftar Catatan:

- Tampilkan judul catatan dalam **Listbox**.
- Ketika pengguna mengklik judul di **Listbox**, isi catatan ditampilkan di area **Text** (read-only mode).

8 / 10

GUI dan TKINTER

- Sertakan **Scrollbar** untuk **Listbox** agar dapat menampung banyak catatan.

3. Hapus Catatan:

- Tombol "Hapus Catatan" untuk menghapus catatan yang dipilih di **Listbox**.
- Tampilkan konfirmasi menggunakan **messagebox** sebelum menghapus.

4. Validasi Input:

- Pastikan judul dan isi catatan tidak kosong sebelum menyimpan.
- Tampilkan pesan error menggunakan **messagebox** jika input tidak valid.

5. Menu Bar:

- Tambahkan **Menu** dengan opsi:
 - **File:** Keluar (menutup aplikasi).
 - **Bantuan:** Tentang (tampilkan dialog dengan nama dan versi aplikasi).

Penjelasan :

1. Impor Modul

```
import tkinter as tk
from tkinter import messagebox, scrolledtext
from datetime import datetime
```

- tkinter: Modul utama untuk membuat GUI
- messagebox: Untuk menampilkan popup dialog
- scrolledtext: Untuk membuat text area dengan scrollbar
- datetime: Untuk mendapatkan waktu pembuatan catatan

2. Kelas Utama

```
class AplikasiCatatan:
    def __init__(self, root):
        self.root = root
        self.root.title("Catatan Harian")
        self.root.geometry("800x600")
        self.catatan = [] # Penyimpanan data catatan
```

- Inisialisasi window utama dengan ukuran 800x600
- self.catatan sebagai penyimpanan data sementara (list of dictionaries)

3. Menu Bar

```
def buat_menu(self):
    menubar = tk.Menu(self.root)

    # Menu File
    file_menu = tk.Menu(menubar, tearoff=0)
    file_menu.add_command(label="Keluar", command=self.root.quit)
    menubar.add_cascade(label="File", menu=file_menu)

    # Menu Bantuan
    help_menu.add_command(label="Tentang", command=self.tampilkan_tentang)
    self.root.config(menu=menubar)
```

- Membuat menu bar dengan 2 dropdown:
 - **File:** Berisi opsi Keluar
 - **Bantuan:** Berisi opsi Tentang

4. Tata letak dan widget

```
def buat_widget(self):
    # Warna tema
    bg_color = '#D2E0FB' # Warna background utama (biru muda)
    widget_bg = '#C4D9FF' # Warna input field
    text_bg = '#8EACCD' # Warna area tampilan catatan
    button_bg = '#e1e1e1' # Warna tombol
```

- **Input Area:**
 - Entry field untuk judul catatan
 - ScrolledText untuk isi catatan
 - Tombol "Tambah" dan "Hapus"
- **Display Area:**
 - Listbox untuk daftar judul catatan
 - ScrolledText untuk menampilkan isi catatan (read-only)

5. Fungsi Utama

A, Tambahan catatan

```
def tambah_catatan(self):
    # Validasi input
    if not judul or not isi:
        messagebox.showerror("Error", "Judul dan isi tidak boleh k

    # Menyimpan dengan timestamp
    catatan_baru = {
        'judul': judul,
        'isi': isi,
        'waktu': datetime.now().strftime("%d/%m/%Y %H:%M")
    }
    self.catatan.append(catatan_baru)
```

- Validasi input kosong
- Menyimpan data dalam format dictionary dengan timestamp

B. Tampilkan Catatan

```
def tampilkan_catatan(self, event):
    # Mengambil data dari list berdasarkan indeks yang dipilih
    catatan = self.catatan[index]
    self.tampil_text.insert(tk.END, f"Judul: {catatan['judul']}\nWaktu: {catatan['waktu']}\n\n{c
atatan['isi']}")
```

Menampilkan detail catatan

C. Hapus catatan

```
def hapus_catatan(self):
    if messagebox.askyesno("Konfirmasi", "Yakin ingin menghapus?"):
        self.daftar_listbox.delete(index)
        self.catatan.pop(index)
```

Source code :

```

import tkinter as tk
from tkinter import messagebox, scrolledtext
from datetime import datetime

class AplikasiCatatan:
    def __init__(self, root):
        self.root = root
        self.root.title("Catatan Harian")
        self.root.geometry("800x600")

        # Struktur data untuk menyimpan catatan
        self.catatan = []

        # Membuat menu bar
        self.buat_menu()

        # Membuat frame utama
        self.buat_widget()

    def buat_menu(self):
        # Membuat menu bar
        menubar = tk.Menu(self.root)

        # Menu File
        file_menu = tk.Menu(menubar, tearoff=0)
        file_menu.add_command(label="Keluar", command=self.root.quit)
        menubar.add_cascade(label="File", menu=file_menu)

        # Menu Bantuan
        help_menu = tk.Menu(menubar, tearoff=0)
        help_menu.add_command(label="tentang", command=self.tampilkan_tentang)
        menubar.add_cascade(label="Bantuan", menu=help_menu)

        self.root.config(menu=menubar)

    def buat_widget(self):
        # Warna Tema
        bg_color = "#D2B48C"
        widget_bg = "#C4A88F"
        text_bg = "#B8ACCD"
        button_bg = "#E1E1E1"

        # Configure root window background
        self.root.config(bg=bg_color)

        # Frame untuk input catatan
        input_frame = tk.Frame(self.root, bg=bg_color)
        input_frame.pack(pady=10, padx=10, fill=tk.X)

        # Label dan Entry untuk judul
        tk.Label(input_frame, text="Judul:", bg=bg_color).grid(row=0, column=0, sticky=tk.W)
        self.judul_entry = tk.Entry(input_frame, width=50, bg=widget_bg)
        self.judul_entry.grid(row=0, column=1, padx=5)

        # Label dan Text untuk isi catatan
        tk.Label(input_frame, text="Isi Catatan:", bg=bg_color).grid(row=1, column=0, sticky=tk.NW,
pady=5)
        self.isi_text = scrolledtext.ScrolledText(input_frame, width=50, height=10, wrap=tk.WORD,
bg=widget_bg)
        self.isi_text.grid(row=1, column=1, padx=5, pady=5)

        # Frame untuk tombol
        button_frame = tk.Frame(self.root, bg=bg_color)
        button_frame.pack(pady=5)

        # Tombol dengan warna
        self.tambah_btn = tk.Button(button_frame, text="Tambah Catatan",
command=self.tambah_catatan, bg=button_bg)
        self.tambah_btn.pack(side=tk.LEFT, padx=5)

        self.hapus_btn = tk.Button(button_frame, text="Hapus Catatan",
command=self.hapus_catatan, bg=button_bg)
        self.hapus_btn.pack(side=tk.LEFT, padx=5)

        # Frame untuk menampilkan catatan
        display_frame = tk.Frame(self.root, bg=bg_color)
        display_frame.pack(pady=10, padx=10, fill=tk.BOTH, expand=True)

        # Listbox dengan warna
        self.daftar_listbox = tk.Listbox(display_frame, width=30, bg=widget_bg)
        self.daftar_listbox.pack(side=tk.LEFT, fill=tk.Y, padx=(0, 5))

        # Text widget dengan warna
        self.tampil_text = scrolledtext.ScrolledText(
display_frame,
width=50,
height=20,
wrap=tk.WORD,
state=tk.DISABLED,
bg=text_bg
)
        self.tampil_text.pack(side=tk.LEFT, fill=tk.BOTH, expand=True)

        # Event binding untuk Listbox
        self.daftar_listbox.bind('<<ListboxSelect>>', self.tampilkan_catatan)

    def tambah_catatan(self):
        # Mendapatkan input dari pengguna
        judul = self.judul_entry.get().strip()
        isi = self.isi_text.get("1.0", tk.END).strip()

        # Validasi input
        if not judul or not isi:
            messagebox.showerror("Error", "Judul dan isi catatan tidak boleh kosong! Isi dulu yaa sebelum tambah notes")
            return

        # Menambahkan timestamp
        waktu = datetime.now().strftime("%d/%m/%Y %H:%M")

        # Menyimpan catatan
        catatan_baru = {
'judul': judul,
'isi': isi,
'waktu': waktu
}
        self.catatan.append(catatan_baru)

        # Menambahkan judul ke Listbox
        self.daftar_listbox.insert(tk.END, f"{judul} ({waktu})")

        # Mengosongkan input fields
        self.judul_entry.delete(0, tk.END)
        self.isi_text.delete("1.0", tk.END)

        messagebox.showinfo("Sukses", "Catatan berhasil ditambahkan!")

    def tampilkan_catatan(self, event):
        # Mendapatkan indeks catatan yang dipilih
        selection = self.daftar_listbox.curselection()

        if selection:
            index = selection[0]

            # Mengaktifkan text widget sementara untuk mengedit
            self.tampil_text.config(state=tk.NORMAL)
            self.tampil_text.delete("1.0", tk.END)

            # Menampilkan isi catatan
            catatan = self.catatan[index]
            self.tampil_text.insert(tk.END, f"Judul: {catatan['judul']}\n")
            self.tampil_text.insert(tk.END, f"Waktu: {catatan['waktu']}\n")
            self.tampil_text.insert(tk.END, "\n" + catatan['isi'])

            # Mengembalikan ke mode read-only
            self.tampil_text.config(state=tk.DISABLED)

    def hapus_catatan(self):
        # Mendapatkan indeks catatan yang dipilih
        selection = self.daftar_listbox.curselection()

        if not selection:
            messagebox.showerror("Error", "Mau hapus apa nih? Pilih dulu catatan yang mau dihapus!")
            return

        # Konfirmasi penghapusan
        if messagebox.askyesno("Konfirmasi", "Apakah Anda yakin ingin menghapus catatan ini?"):
            index = selection[0]

            # Menghapus dari Listbox dan daftar catatan
            self.daftar_listbox.delete(index)
            self.catatan.pop(index)

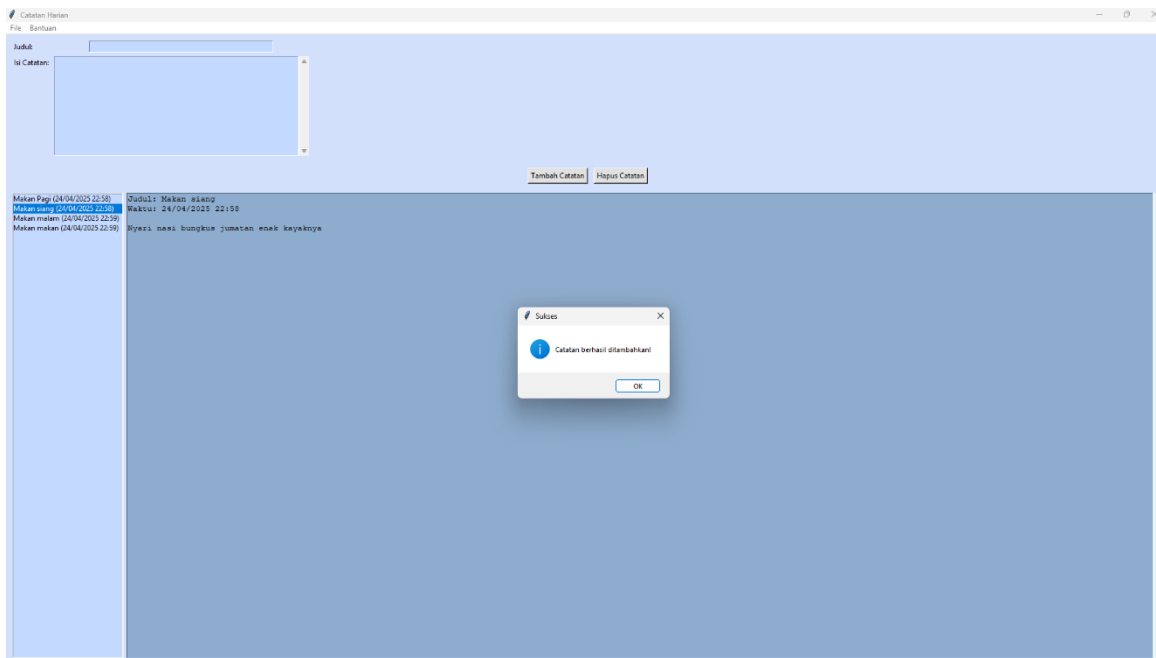
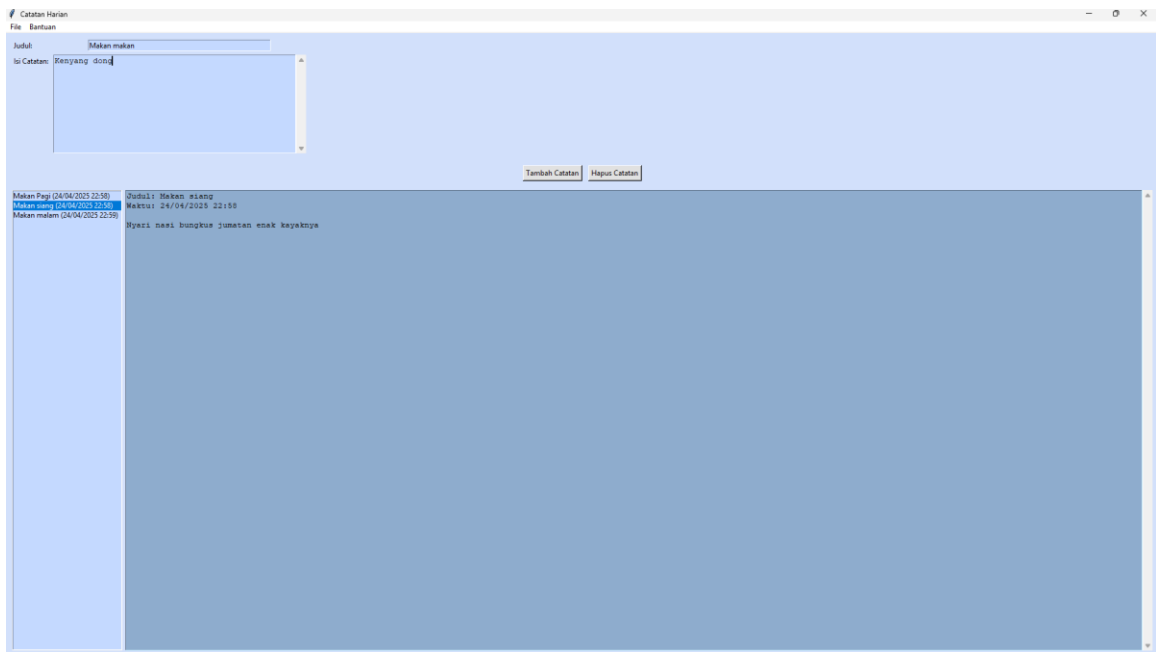
            # Mengosongkan tampilan isi catatan
            self.tampil_text.config(state=tk.NORMAL)
            self.tampil_text.delete("1.0", tk.END)
            self.tampil_text.config(state=tk.DISABLED)

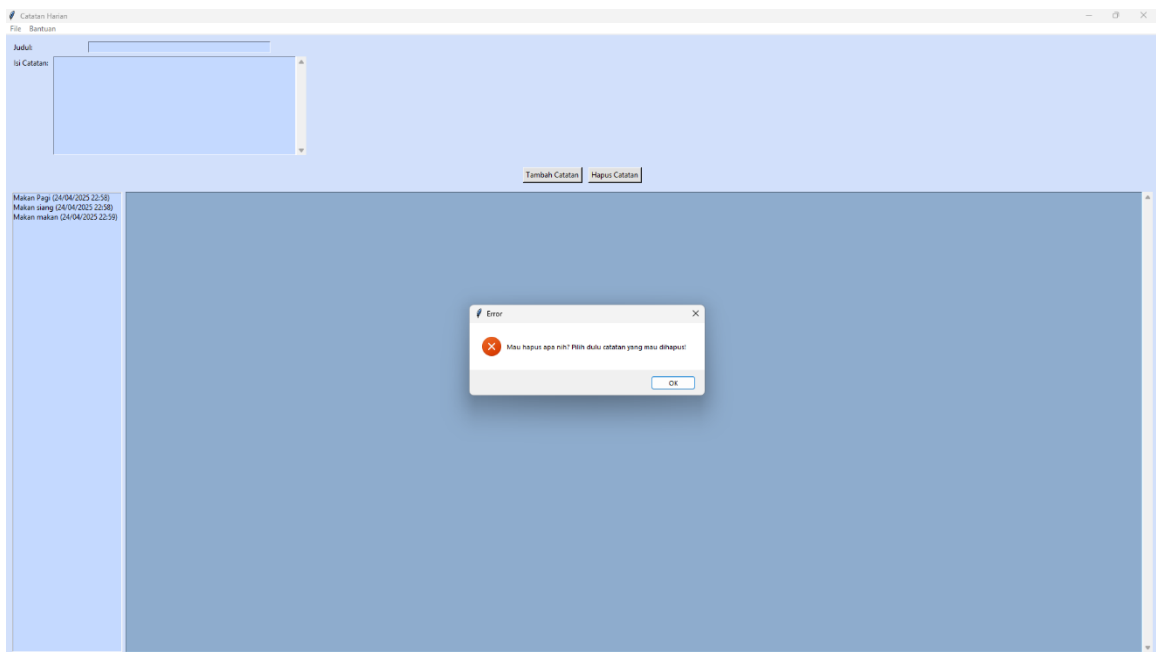
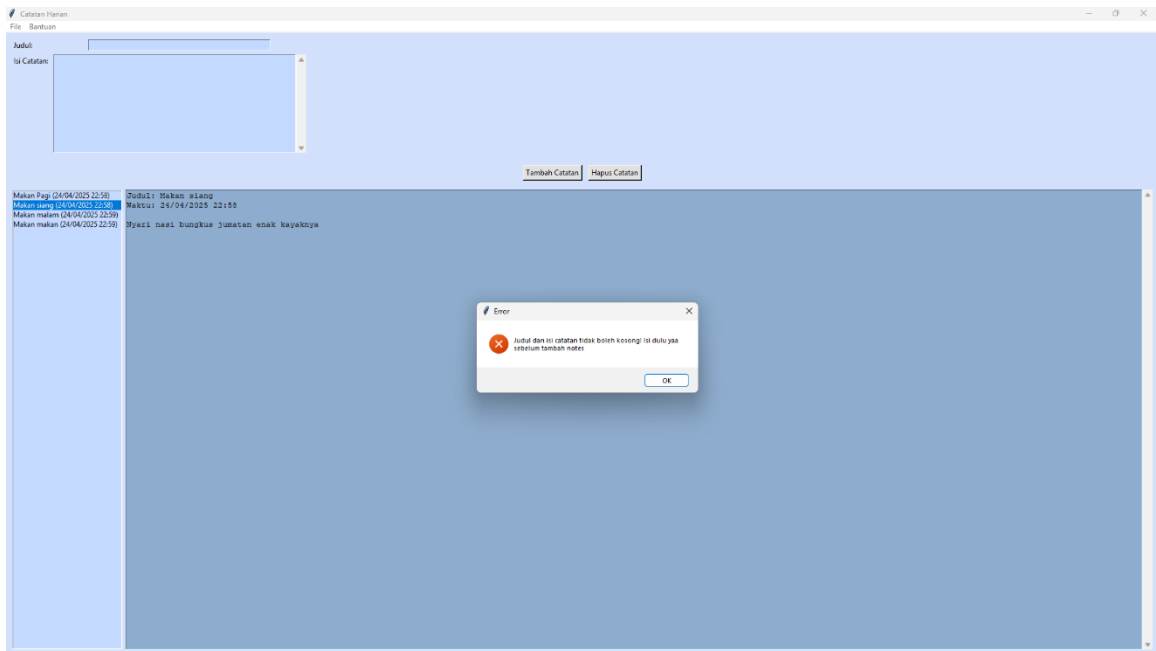
    def tampilkan_tentang(self):
        about_text = "Aplikasi Catatan Harian\nVersi 1.0\n\nAplikasi sederhana untuk mencatat kegiatan harian."
        messagebox.showinfo("Tentang", about_text)

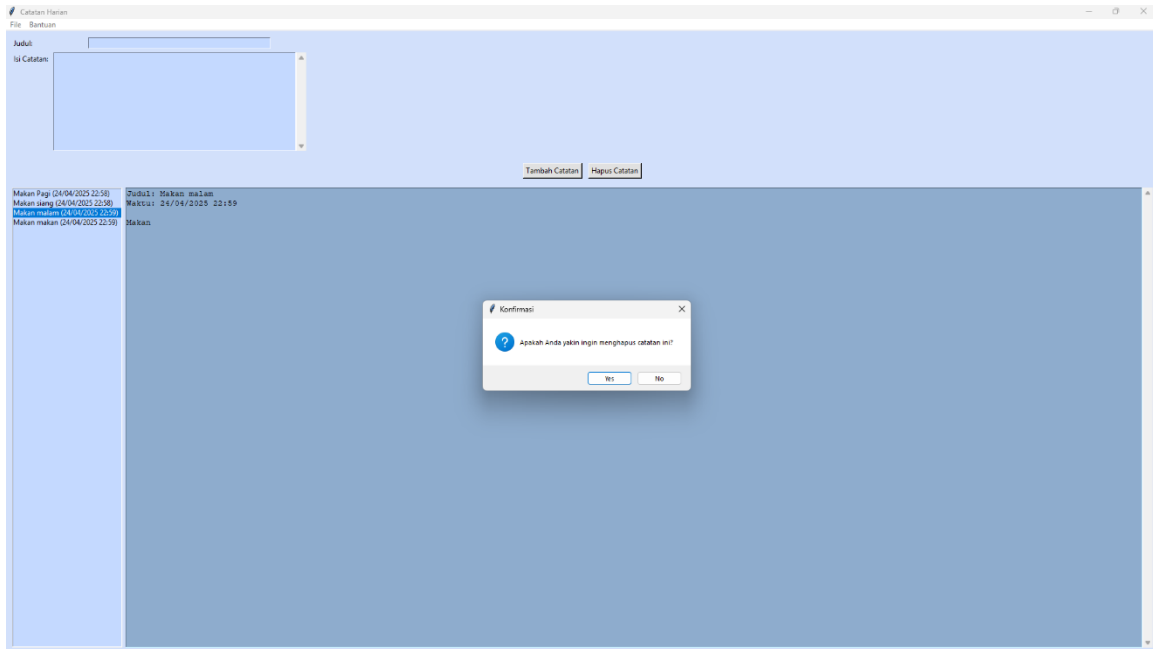
if __name__ == "__main__":
    root = tk.Tk()
    app = AplikasiCatatan(root)
    root.mainloop()

```

Output Hasil (Screenshot)







Lampiran :

<https://chat.deepseek.com/a/chat/s/43e35156-9040-4878-85a5-b365b30c7e3e>

+copilot