

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK (PBO) – 5**



Disusun Oleh

Abel Fortino 123140111

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SUMATERA**

2025

Soal

Challenge Praktikum: Membuat Aplikasi Catatan Harian dengan Tkinter

Latar Belakang

Dalam praktikum ini, Anda akan ditantang untuk membuat **Aplikasi Catatan Harian** sederhana menggunakan Tkinter. Aplikasi ini memungkinkan pengguna untuk menulis, menyimpan, dan mengelola catatan harian dengan antarmuka grafis yang intuitif. Tantangan ini dirancang untuk menguji pemahaman Anda tentang widget Tkinter, manajemen tata letak, penanganan event, dan interaksi dasar dengan file.

Deskripsi Tantangan

Buatlah aplikasi GUI dengan Tkinter yang memiliki fitur berikut:

- 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.
- Tampilan Daftar Catatan:**
 - Tampilkan judul catatan dalam `ListBox`.
 - Ketika pengguna mengklik judul di `ListBox`, isi catatan ditampilkan di area `Text` (read-only mode).
 - Sertakan `Scrollbar` untuk `ListBox` agar dapat menampung banyak catatan.
- Hapus Catatan:**
 - Tombol "Hapus Catatan" untuk menghapus catatan yang dipilih di `ListBox`.
 - Tampilkan konfirmasi menggunakan `messagebox` sebelum menghapus.
- Validasi Input:**
 - Pastikan judul dan isi catatan tidak kosong sebelum menyimpan.
 - Tampilkan pesan error menggunakan `messagebox` jika input tidak valid.
- Menu Bar:**
 - Tambahkan Menu dengan opsi:
 - File:** Keluar (menutup aplikasi).
 - Bantuan:** Tentang (tampilkan dialog dengan nama dan versi aplikasi).

Persyaratan Teknis

- Gunakan **Python 3** dan **Tkinter**.
- Gunakan **Grid** untuk tata letak utama agar rapi, atau **Pack** untuk bagian sederhana.
- Simpan catatan dalam struktur data sederhana seperti daftar (`list`) atau kamus (`dict`) di memori (tidak perlu penyimpanan file untuk menyederhanakan).
- Gunakan `messagebox` untuk konfirmasi dan pesan error.
- Aplikasi harus memiliki antarmuka yang jelas dan mudah digunakan.
- Kode harus terstruktur dengan fungsi untuk modularitas (misalnya, fungsi untuk menambah, menghapus, dan menampilkan catatan).
- Tambahkan komentar untuk menjelaskan logika utama.

Bonus (Opsional)

- Penyimpanan File:**
 - Simpan catatan ke file teks atau JSON saat aplikasi ditutup dan muat kembali saat dibuka.
- Waktu Catatan:**
 - Tambahkan tanggal/waktu pembuatan catatan (gunakan modul `datetime`) dan tampilkan di samping judul di `ListBox`.
- Edit Catatan:**
 - Tambahkan tombol "Edit Catatan" untuk memperbarui judul atau isi catatan yang dipilih.

Panduan Implementasi

1. **Inisialisasi Jendela:**
 - Buat jendela utama dengan Tk() dan atur judul (misalnya, "Catatan Harian").
 - Tambahkan Menu dengan opsi File dan Bantuan.
2. **Widget Utama:**
 - Gunakan Entry untuk input judul.
 - Gunakan Text untuk input isi catatan.
 - Gunakan Listbox dengan Scrollbar untuk daftar judul.
 - Tambahkan dua Button untuk "Tambah Catatan" dan "Hapus Catatan".
3. **Fungsi Inti:**
 - **Tambah Catatan:** Validasi input, simpan judul dan isi ke daftar/kamus, tambahkan judul ke Listbox.
 - **Tampilkan Catatan:** Saat judul di Listbox diklik (gunakan event <listboxSelect>), tampilkan isi di Text (set state="disabled" untuk read-only).
 - **Hapus Catatan:** Hapus catatan yang dipilih dari daftar dan Listbox setelah konfirmasi.
4. **Validasi:**
 - Gunakan messagebox.showerror untuk input kosong.
 - Gunakan messagebox.askyesno untuk konfirmasi penghapusan.

Penjelasan :

import tkinter as tk : Untuk membuat antarmuka GUI.

import json : Untuk menyimpan file catatan.json dalam format JSON.

import os : Untuk mengecek apakah file catatan.json sudah ada.

from tkinter import messagebox : Untuk menampilkan pesan popup seperti error.

from datetime import datetime : Untuk mengambil waktu sekarang saat catatan ditambahkan.

tambah_catatan()

- Mengambil isi dari entry_judul (judul) dan text_isi (isi catatan).
- Cek kalau dua-duanya gak kosong. Kalau kosong, munculin error pakai messagebox.showerror.
- Tambahin cap waktu (datetime.now().strftime(...)) ke judul.
- Menyimpan catatan sebagai dictionary ke list catatan.
- Masukin judulnya ke listbox supaya muncul di tampilan.
- Kosongin kembali input judul dan isi setelah catatan disimpan.

tampilkan_catatan(event)

- Ambil index yang dipilih, terus ambil isi dari list catatan.
- Tampilkan isi catatan ke text_isi tapi dalam mode **read-only** (state "disabled").

hapus_catatan()

- Cek kalau gak ada yang dipilih di listbox, kasih error pakai messagebox.
- Kalau ada, munculin konfirmasi pakai askyesno.
- Kalau user setuju, hapus catatan dari listbox dan juga dari list catatan.

- Kosongin area text_isi juga.

simpan_ke_file()

- Fungsi ini dipanggil saat aplikasi ditutup.
- Menyimpan semua data catatan ke file catatan.json dalam format JSON.

muat_dari_file()

- Fungsi ini dipanggil saat aplikasi baru dibuka, kalau file catatan.json ada, baca isinya dan tampilkan semua judul ke listbox.

keluar()

- Untuk nutup aplikasi.

```
import tkinter as tk
import json
import os
from tkinter import messagebox
from datetime import datetime

root = tk.Tk()
root.title("Catatan Harian")

catatan = []

def tambah_catatan():
    judul = entry_judul.get().strip()
    isi = text_isi.get("1.0", "end").strip()
    if not judul or not isi:
        return messagebox.showerror("Error", "Judul dan isi tidak boleh kosong.")
    waktu = datetime.now().strftime("%Y-%m-%d %H:%M")
    catatan.append({"judul": f"{waktu} - {judul}", "isi": isi})
    listbox.insert("end", catatan[-1]["judul"])
    entry_judul.delete(0, "end")
    text_isi.delete("1.0", "end")

def tampilkan_catatan(event):
    if not listbox.curselection(): return
    idx = listbox.curselection()[0]
    text_isi.config(state="normal")
    text_isi.delete("1.0", "end")
    text_isi.insert("end", catatan[idx]["isi"])
    text_isi.config(state="disabled")
```

```

def hapus_catatan():
    if not listbox.curselection():
        return messagebox.showerror("Error", "Pilih catatan yang akan
dihapus.")
    if messagebox.askyesno("Konfirmasi", "Yakin ingin menghapus catatan
ini?"):
        idx = listbox.curselection()[0]
        listbox.delete(idx)
        del catatan[idx]
        text_isi.config(state="normal")
        text_isi.delete("1.0", "end")

def simpan_ke_file():
    with open("catatan.json", "w") as f:
        json.dump(catatan, f)

def muat_dari_file():
    if os.path.exists("catatan.json"):
        with open("catatan.json", "r") as f:
            data = json.load(f)
            catatan.extend(data)
            for item in data:
                listbox.insert("end", item["judul"])

def keluar():
    simpan_ke_file()
    root.destroy()

def keluar(): root.destroy()
def tentang(): messagebox.showinfo("Tentang", "Catatan Harian v1.0\nDibuat
dengan Tkinter")

menu = tk.Menu(root)
file_menu = tk.Menu(menu, tearoff=0)
file_menu.add_command(label="Keluar", command=keluar)
menu.add_cascade(label="File", menu=file_menu)
help_menu = tk.Menu(menu, tearoff=0)
help_menu.add_command(label="Tentang", command=tentang)
menu.add_cascade(label="Bantuan", menu=help_menu)
root.config(menu=menu)

entry_judul = tk.Entry(root, width=40)
entry_judul.grid(row=0, column=0, columnspan=2, sticky="we", padx=5,
pady=5)

```

```
tk.Button(root, text="Tambah Catatan", command=tambah_catatan).grid(row=1,
column=0, sticky="we", padx=5)
tk.Button(root, text="Hapus Catatan", command=hapus_catatan).grid(row=1,
column=1, sticky="we", padx=5)

frame_kiri = tk.Frame(root)
frame_kanan = tk.Frame(root)
root.columnconfigure(0, weight=2)
root.columnconfigure(1, weight=3)

frame_kiri.grid(row=2, column=0, sticky="nsew", padx=5, pady=5)
frame_kanan.grid(row=2, column=1, sticky="nsew", padx=5, pady=5)

listbox = tk.Listbox(frame_kiri)
listbox.pack(side="left", fill="both", expand=True)
scrollbar = tk.Scrollbar(frame_kiri, command=listbox.yview)
scrollbar.pack(side="right", fill="y")
listbox.config(yscrollcommand=scrollbar.set)
listbox.bind("<<ListboxSelect>>", tampilkan_catatan)

text_isi = tk.Text(frame_kanan, wrap="word")
text_isi.pack(fill="both", expand=True)

root.mainloop()
```

Lampiran

1. <https://www.w3schools.in/python/gui-programming>
2. <https://chatgpt.com/share/680a1b60-16f8-8005-8fbe-f5ef65710637>

Masalahnya terletak di bagian `Text` widget (`text_isi`) yang diatur ke `state="disabled"` saat ditampilkan dan tidak diubah kembali ke `normal` ketika pengguna ingin menulis catatan baru.

Penyebab:

Setelah memilih catatan dari `Listbox`, fungsi `tampilkan_catatan()` mengatur `text_isi` menjadi `state="disabled"`, agar tidak bisa diedit. Tapi saat menambahkan catatan baru, `text_isi` tidak diatur kembali menjadi `normal`, sehingga kamu tidak bisa mengetik apa-apa.

Solusi:

Tambahkan baris ini di awal fungsi `tambah_catatan()` sebelum mengambil isi dari `text_isi`:

```
python Copy Edit

text_isi.config(state="normal")
```

Dan jika kamu ingin pengguna bisa mengetik catatan sebelum diklik `Listbox`, sebaiknya atur `text_isi` ke `"normal"` sejak awal dan hanya `disabled` saat menampilkan isi catatan.

Perbaiki kode:

Modifikasi bagian `text_isi` jadi seperti ini:

```
python Copy Edit

text_isi = tk.Text(frame_kanan, wrap="word") # jangan set state="disabled"
text_isi.pack(fill="both", expand=True)
```

Dan di fungsi `tambah_catatan()`:

```
python Copy Edit

def tambah_catatan():
    text_isi.config(state="normal") # Tambahkan ini
    judul = entry_judul.get().strip()
    isi = text_isi.get("1.0", "end").strip()
    if not judul or not isi:
        return messagebox.showerror("Error", "Judul dan isi tidak boleh kosong.")
    waktu = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    catatan = f"{waktu} | {judul} | {isi}"
    listbox.insert(END, catatan)
    text_isi.delete("1.0", "end")
    entry_judul.delete(0, "end")
    entry_isi.delete(0, "end")
    entry_isi.focus()
```