

# **LIST STUDI KASUS GUI MANAJEMEN PROKER ORGANISASI**

T.A Semester Ganjil 2024/2025



DISUSUN OLEH :  
NAJWA SHAFIRA ZAHRA (5230411293)

**PROGRAM STUDI INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS TEKNOLOGI YOGYAKARTA  
YOGYAKARTA  
2024**

# DAFTAR ISI

I.	STUDI KASUS: Sistem Manajemen Program Kerja Himpunan Mahasiswa Informatika ..	3
II.	TUJUAN.....	3
III.	KODE PROGRAM.....	3
IV.	HASIL KODE PROGRAM .....	7
V.	PEMBAHASAN .....	8
1.	Import Pustaka .....	8
2.	Fungsi untuk Menambah Proker .....	8
3.	Fungsi untuk Menghapus Proker .....	9
4.	Fungsi untuk Melihat Detail Proker .....	10
5.	Fungsi untuk mengekspor ke Excel .....	11
6.	Data Awal.....	12
7.	Antarmuka Pengguna (UI) .....	12
8.	Menampilkan data awal.....	13
9.	Menjalankan Aplikasi.....	13
VI.	Activity diagram.....	14
VII.	KESIMPULAN .....	15

## I. STUDI KASUS: Sistem Manajemen Program Kerja Himpunan Mahasiswa Informatika

Himpunan Mahasiswa Informatika sering kali menghadapi tantangan dalam mengelola berbagai program kerja (proker) yang dirancang untuk mendukung kegiatan akademik, pengembangan teknologi, dan pemberdayaan mahasiswa. Program kerja ini perlu dikelola dengan baik agar informasi seperti nama program, detail, kategori (bulanan/tahunan), dan tanggal pelaksanaan dapat dicatat dan diakses dengan mudah oleh anggota organisasi.

## II. TUJUAN

1. **Pencatatan Program Kerja:** Pengguna dapat menambahkan nama, detail, tanggal pelaksanaan, dan kategori program kerja.
2. **Pengelompokan Berdasarkan Kategori:** Memungkinkan pengguna untuk memfilter program kerja berdasarkan kategori bulanan atau tahunan.
3. **Detail Program Kerja:** Menampilkan informasi rinci dari setiap program kerja yang dipilih.
4. **Penghapusan Data:** Menghapus program kerja yang sudah tidak relevan.
5. **Ekspor Data:** Menyimpan daftar program kerja ke file teks sebagai dokumentasi.

## III. KODE PROGRAM

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

# Fungsi untuk menambah program kerja
def tambah_proker():
    nama_proker = entry_nama.get()
    detail_proker = entry_detail.get()
    kategori_proker = kategori_var.get()
    tanggal_proker = entry_tanggal.get()

    # Validasi input
    if nama_proker == "" or detail_proker == "" or tanggal_proker == "":
        messagebox.showwarning("Input Error", "Semua kolom harus diisi!")
        return

    # Validasi format tanggal
    try:
        datetime.strptime(tanggal_proker, "%d/%m/%Y")
    except ValueError:
        messagebox.showwarning("Format Tanggal Salah", "Gunakan format DD/MM/YYYY.")
        return
```

```

# Menambahkan ke daftar
proker_list.append({"nama": nama_proker, "detail": detail_proker,
"kategori": kategori_proker, "tanggal": tanggal_proker})
listbox_proker.insert(tk.END, f"{tanggal_proker} - {nama_proker} -
{kategori_proker}")
entry_nama.delete(0, tk.END)
entry_detail.delete(0, tk.END)
entry_tanggal.delete(0, tk.END)

# Fungsi untuk menghapus program kerja
def hapus_proker():
    try:
        selected_index = listbox_proker.curselection()[0]
        listbox_proker.delete(selected_index)
        del proker_list[selected_index]
    except IndexError:
        messagebox.showwarning("Peringatan", "Pilih proker yang ingin
dihapus!")

# Fungsi untuk menampilkan detail proker
def lihat_detail():
    try:
        selected_index = listbox_proker.curselection()[0]
        proker = proker_list[selected_index]
        messagebox.showinfo(
            "Detail Proker",
            f>Nama: {proker['nama']}\nTanggal:
{proker['tanggal']}\nDetail: {proker['detail']}\nKategori:
{proker['kategori']}"
        )
    except IndexError:
        messagebox.showwarning("Peringatan", "Pilih proker yang ingin
dilihat detailnya!")

# Fungsi untuk mengekspor data ke file
def export_data():
    try:
        file = filedialog.asksaveasfilename(defaultextension=".txt",
filetypes=[("Text files", "*.txt")])
        if file:
            with open(file, "w") as f:
                for proker in proker_list:
                    f.write(f"Tanggal: {proker['tanggal']}\nNama:
{proker['nama']}\nDetail: {proker['detail']}\nKategori:
{proker['kategori']}\n\n")
            messagebox.showinfo("Sukses", "Data berhasil diekspor!")
    except Exception as e:
        messagebox.showerror("Error", f"Gagal mengekspor data: {e}")

```

```

# Fungsi untuk filter proker berdasarkan kategori
def filter_proker():
    kategori = kategori_filter_var.get()
    listbox_proker.delete(0, tk.END)
    for proker in proker_list:
        if kategori == "Semua" or proker["kategori"] == kategori:
            listbox_proker.insert(tk.END, f"{proker['tanggal']} - {proker['nama']} - {proker['kategori']}")

# Data awal proker
proker_list = [
    {"nama": "Workshop Pemrograman", "detail": "Workshop tentang pemrograman Python.", "kategori": "Bulanan", "tanggal": "15/12/2024"},
    {"nama": "Seminar Teknologi", "detail": "Seminar tentang tren teknologi terbaru.", "kategori": "Tahunan", "tanggal": "25/12/2024"}
]

# Setup UI
root = tk.Tk()
root.title("Manajemen Proker Organisasi Himpunan Mahasiswa Informatika")
root.geometry("550x450")
root.config(bg="#E0E0E0")

# Frame Judul
frame_judul = tk.Frame(root, bg="#E0E0E0")
frame_judul.pack(pady=10)
judul = tk.Label(frame_judul, text="Manajemen Proker Organisasi Himpunan Mahasiswa Informatika", font=("Arial", 18, "bold"), bg="#E0E0E0", fg="#F5A623")
judul.pack()

# Frame Input
frame_input = tk.Frame(root, bg="#E0E0E0")
frame_input.pack(pady=10)
label_nama = tk.Label(frame_input, text="Nama Proker:", bg="#E0E0E0")
label_nama.grid(row=0, column=0)
entry_nama = tk.Entry(frame_input, width=30)
entry_nama.grid(row=0, column=1)

label_detail = tk.Label(frame_input, text="Detail Proker:", bg="#E0E0E0")
label_detail.grid(row=1, column=0)
entry_detail = tk.Entry(frame_input, width=30)
entry_detail.grid(row=1, column=1)

```

```

label_tanggal = tk.Label(frame_input, text="Tanggal (DD/MM/YYYY):",
bg="#E0E0E0")
label_tanggal.grid(row=2, column=0)
entry_tanggal = tk.Entry(frame_input, width=30)
entry_tanggal.grid(row=2, column=1)

label_kategori = tk.Label(frame_input, text="Kategori Proker:",
bg="#E0E0E0")
label_kategori.grid(row=3, column=0)
kategori_var = tk.StringVar(value="Bulanan")
kategori_bulanan = tk.Radiobutton(frame_input, text="Bulanan",
variable=kategori_var, value="Bulanan", bg="#E0E0E0")
kategori_bulanan.grid(row=3, column=1, sticky="w")
kategori_tahunan = tk.Radiobutton(frame_input, text="Tahunan",
variable=kategori_var, value="Tahunan", bg="#E0E0E0")
kategori_tahunan.grid(row=3, column=1, sticky="e")

# Frame Listbox dan Tombol
frame_listbox = tk.Frame(root, bg="#E0E0E0")
frame_listbox.pack(pady=10)
listbox_proker = tk.Listbox(frame_listbox, width=60, height=10)
listbox_proker.pack()

# Frame Tombol
frame_tombol = tk.Frame(root, bg="#E0E0E0")
frame_tombol.pack(pady=10)
tambah_button = tk.Button(frame_tombol, text="Tambah Proker",
command=tambah_proker, bg="#F5A623")
tambah_button.pack(side=tk.LEFT, padx=5)

hapus_button = tk.Button(frame_tombol, text="Hapus Proker",
command=hapus_proker, bg="#F5A623")
hapus_button.pack(side=tk.LEFT, padx=5)

detail_button = tk.Button(frame_tombol, text="Lihat Detail",
command=lihat_detail, bg="#F5A623")
detail_button.pack(side=tk.LEFT, padx=5)

export_button = tk.Button(frame_tombol, text="Export Data",
command=export_data, bg="#F5A623")
export_button.pack(side=tk.LEFT, padx=5)

# Frame Filter
frame_filter = tk.Frame(root, bg="#E0E0E0")
frame_filter.pack(pady=10)
kategori_filter_var = tk.StringVar(value="Semua")

```

```

kategori_filter_bulanan = tk.Radiobutton(frame_filter, text="Bulanan",
variable=kategori_filter_var, value="Bulanan", bg="#E0E0E0",
command=filter_proker)
kategori_filter_bulanan.pack(side=tk.LEFT, padx=5)
kategori_filter_tahunan = tk.Radiobutton(frame_filter, text="Tahunan",
variable=kategori_filter_var, value="Tahunan", bg="#E0E0E0",
command=filter_proker)
kategori_filter_tahunan.pack(side=tk.LEFT, padx=5)
kategori_filter_semua = tk.Radiobutton(frame_filter, text="Semua",
variable=kategori_filter_var, value="Semua", bg="#E0E0E0",
command=filter_proker)
kategori_filter_semua.pack(side=tk.LEFT, padx=5)

# Menampilkan Data Awal
for proker in proker_list:
    listbox_proker.insert(tk.END, f"{proker['tanggal']} -
{proker['nama']} - {proker['kategori']}")

# Menjalankan aplikasi
root.mainloop()

```

#### IV. HASIL KODE PROGRAM

**Manajemen Proker Organisasi Himpunan Mahasiswa Informatika**

Nama Proker:

Detail Proker:

Tanggal (DD/MM/YYYY):

Kategori Proker: ☒ Bulanan ☐ Tahunan

15/12/2024 - Workshop Pemrograman - Bulanan  
25/12/2024 - Seminar Teknologi - Tahunan

## V. PEMBAHASAN

### 1. Import Pustaka

```
1 import tkinter as tk
2 from tkinter import messagebox, filedialog
3 from datetime import datetime
4 from openpyxl import Workbook
5
```

- **tkinter**: Digunakan untuk membuat antarmuka pengguna (GUI).
- **messagebox**: Untuk menampilkan dialog pesan (peringatan, error, atau informasi).
- **filedialog**: Untuk membuka atau menyimpan file melalui dialog file.
- **datetime**: Untuk memvalidasi dan memformat tanggal.
- **openpyxl**: Digunakan untuk membuat dan mengelola file Excel (.xlsx).

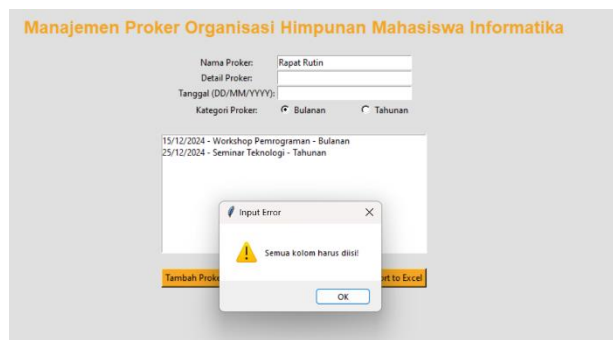
### 2. Fungsi untuk Menambah Proker

```
5 # Fungsi untuk menambah program kerja
7 def tambah_proker():
8     nama_proker = entry_nama.get()
9     detail_proker = entry_detail.get()
10    kategori_proker = kategori_var.get()
11    tanggal_proker = entry_tanggal.get()
12
```

- entry\_nama, entry\_detail, dan entry\_tanggal untuk nama, detail, dan tanggal proker.
- kategori\_var.get() mengambil nilai kategori yang dipilih (Bulanan/Tahunan).

```
if nama_proker == "" or detail_proker == "" or tanggal_proker == "":
    messagebox.showwarning("Input Error", "Semua kolom harus diisi!")
    return
```

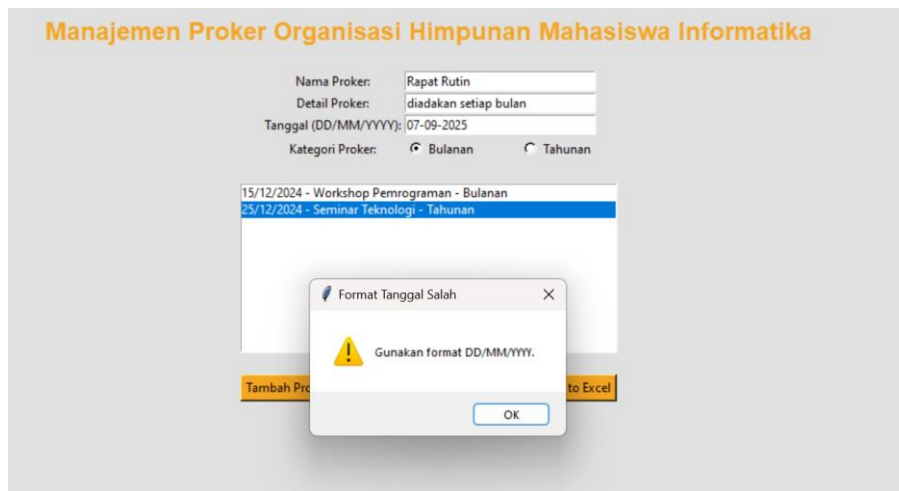
- Memvalidasi apakah semua input telah diisi.
- Jika ada yang kosong, dialog peringatan akan muncul, dan fungsi dihentikan.





```
try:
    datetime.strptime(tanggal_proker, "%d/%m/%Y")
except ValueError:
    messagebox.showwarning("Format Tanggal Salah", "Gunakan format DD/MM/YYYY.")
    return
```

- Mengecek apakah tanggal yang dimasukkan sesuai format **DD/MM/YYYY**.
- Jika salah, dialog peringatan muncul.



```
# Menambahkan ke daftar
proker_list.append({"nama": nama_proker, "detail": detail_proker, "kategori": kategori_proker, "tanggal": tanggal})
listbox_proker.insert(tk.END, f"{tanggal_proker} - {nama_proker} - {kategori_proker}")
entry_nama.delete(0, tk.END)
entry_detail.delete(0, tk.END)
entry_tanggal.delete(0, tk.END)
```

- Menambahkan data baru ke dalam `proker_list`.
- Menampilkan data baru di `listbox_proker`.

### 3. Fungsi untuk Menghapus Proker

```
# Fungsi untuk menghapus program kerja
def hapus_proker():
    try:
        selected_index = listbox_proker.curselection()[0]
        listbox_proker.delete(selected_index)
        del proker_list[selected_index]
    except IndexError:
        messagebox.showwarning("Peringatan", "Pilih proker yang ingin dihapus!")
```

- Menghapus item yang dipilih di `listbox_proker`:
- `curselection()` mengambil indeks item yang dipilih.
- `listbox_proker.delete(selected_index)` menghapus dari listbox.

- `del proker_list[selected_index]` menghapus dari daftar.
- Jika tidak ada item yang dipilih, dialog peringatan muncul.

### Before

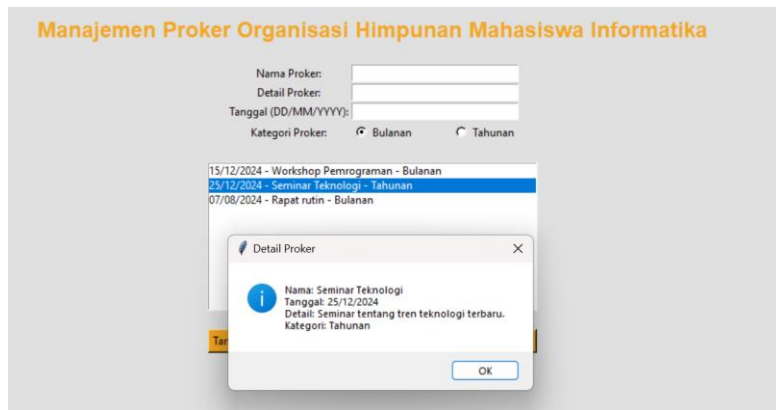
### After

Pilih proker yang mau dihapus lalu klik hapus proker maka proker akan otomatis kehapus

## 4. Fungsi untuk Melihat Detail Proker

```
# Fungsi untuk menampilkan detail proker
def lihat_detail():
    try:
        selected_index = listbox_proker.curselection()[0]
        proker = proker_list[selected_index]
        messagebox.showinfo(
            "Detail Proker",
            f"Nama: {proker['nama']}\nTanggal: {proker['tanggal']}\nDetail: {proker['detail']}\nKategori: {proker['ka']}"
        )
    except IndexError:
        messagebox.showwarning("Peringatan", "Pilih proker yang ingin dilihat detailnya!")
```

- Menampilkan detail proker yang dipilih di dalam dialog **messagebox**:
  - Data diambil dari `proker_list[selected_index]`.
  - Jika tidak ada yang dipilih, dialog peringatan muncul.



## 5. Fungsi untuk mengekspor ke Excel

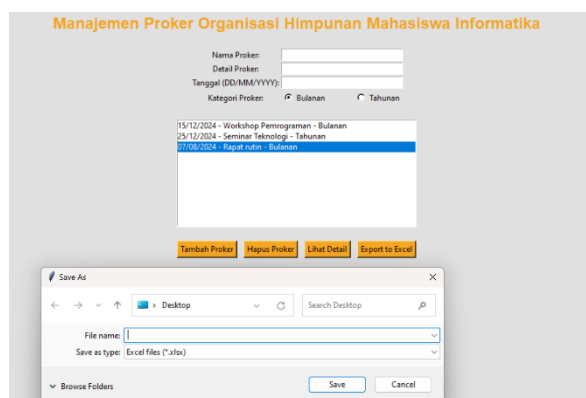
```
# Fungsi untuk mengekspor data ke file Excel
def export_to_excel():
    try:
        file = filedialog.asksaveasfilename(defaultextension=".xlsx", filetypes=[("Excel files", "*.xlsx")])
        if file:
            wb = Workbook()
            ws = wb.active
            ws.title = "Data Proker"

            # Header
            ws.append(["Tanggal", "Nama Proker", "Detail", "Kategori"])

            # Isi data
            for proker in proker_list:
                ws.append([proker["tanggal"], proker["nama"], proker["detail"], proker["kategori"]])

            # Simpan file
            wb.save(file)
            messagebox.showinfo("Sukses", "Data berhasil diekspor ke Excel!")
    except Exception as e:
        messagebox.showerror("Error", f"Gagal mengekspor data: {e}")
```

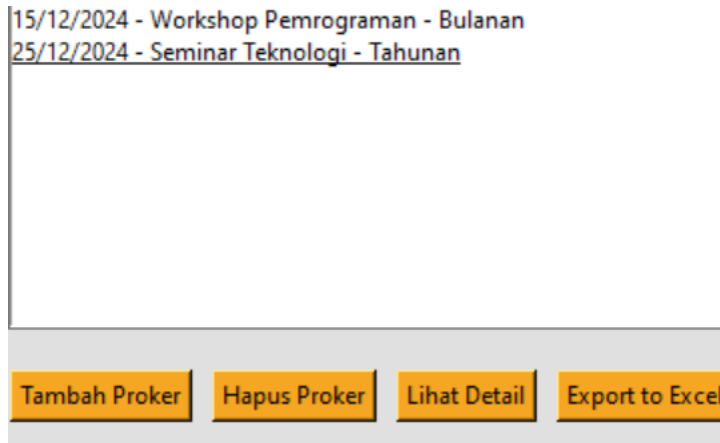
- Membuka dialog penyimpanan file menggunakan `filedialog.asksaveasfilename`.
- Membuat file Excel baru:
- `Workbook()` membuat workbook baru.
- Menambahkan header kolom dengan `ws.append`.
- Menambahkan isi data dari `proker_list` ke Excel.
- Menyimpan file dan menampilkan dialog sukses.
- Jika terjadi kesalahan, dialog error akan ditampilkan.



## 6. Data Awal

```
# Data awal proker
proker_list = [
    {"nama": "Workshop Pemrograman", "detail": "Workshop tentang pemrograman Python.", "kategori": "Bulanan", "tanggal": "15/12/2024"},
    {"nama": "Seminar Teknologi", "detail": "Seminar tentang tren teknologi terbaru.", "kategori": "Tahunan", "tanggal": "25/12/2024"}
]
```

Data awal program kerja. Ditampilkan di listbox\_proker saat aplikasi dijalankan



## 7. Antarmuka Pengguna (UI)

- Membuat judul utama aplikasi

```
root = tk.Tk()
root.title("Manajemen Proker Organisasi Himpunan Mahasiswa Informatika")
root.geometry("550x450")
root.config(bg="#E0E0E0")
```

- Input data
  - Membuat input field untuk nama, detail, dan tanggal.
  - kategori\_var digunakan untuk menyimpan pilihan kategori (Bulanan/Tahunan).

```
# Frame Judul
frame_judul = tk.Frame(root, bg="#E0E0E0")
frame_judul.pack(pady=10)
judul = tk.Label(frame_judul, text="Manajemen Proker Organisasi Himpunan Mahasiswa Informatika", font=("Arial", 18, "bold"))
judul.pack()
```

- Listbox dan Tombol
  - **Listbox** untuk menampilkan daftar program kerja.
  - **Tombol** untuk menjalankan fungsi-fungsi aplikasi.

```

# Frame Listbox dan Tombol
frame_listbox = tk.Frame(root, bg="#E0E0E0")
frame_listbox.pack(pady=10)
listbox_proker = tk.Listbox(frame_listbox, width=60, height=10)
listbox_proker.pack()

# Frame Tombol
frame_tombol = tk.Frame(root, bg="#E0E0E0")
frame_tombol.pack(pady=10)
tambah_button = tk.Button(frame_tombol, text="Tambah Proker", command=tambah_proker, bg="#F5A623")
tambah_button.pack(side=tk.LEFT, padx=5)

hapus_button = tk.Button(frame_tombol, text="Hapus Proker", command=hapus_proker, bg="#F5A623")
hapus_button.pack(side=tk.LEFT, padx=5)

detail_button = tk.Button(frame_tombol, text="Lihat Detail", command=lihat_detail, bg="#F5A623")
detail_button.pack(side=tk.LEFT, padx=5)

```

## 8. Menampilkan data awal

Mengisi listbox dengan data awal dari proker\_list.

```

# Menampilkan Data Awal
for proker in proker_list:
    listbox_proker.insert(tk.END, f"{proker['tanggal']} - {proker['nama']} - {proker['kategori']}")

```

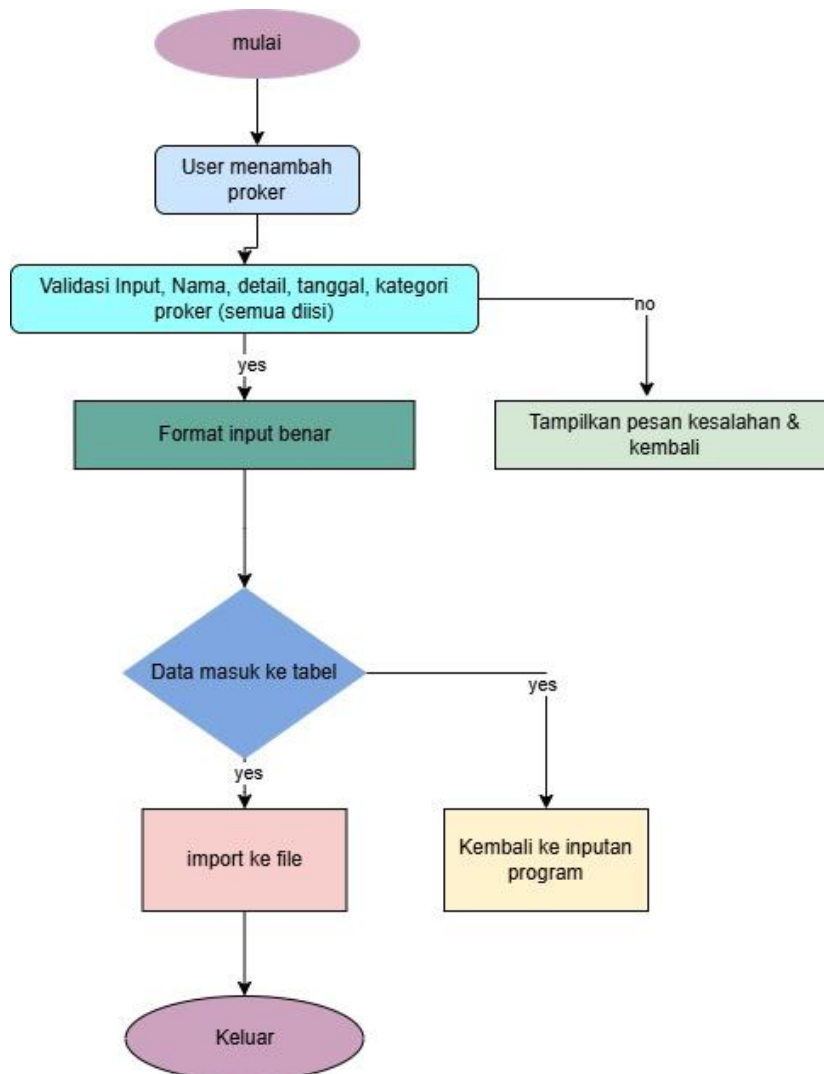
## 9. Menjalankan Aplikasi

```

# Menjalankan aplikasi
root.mainloop()

```

## VI. Activity diagram



- **Mulai**
- **Input Data Proker (Nama, Detail, Tanggal, Kategori)**
- **Validasi Input (Tidak Boleh Kosong):**

Jika input kosong, tampilkan peringatan dan kembali ke input data.

Jika input valid, lanjutkan ke langkah berikutnya.
- **Format input**

Benar = data masuk ke tabel

Salah = Tampilkan pesan kesalahan dan Kembali
- **Data masuk ke tabel**

Jika Input import ke file, maka file akan tersimpan di file laptop dengan format.xlsx

Jika input Kembali ke inputan program, maka inputan akan Kembali ke semula
- **Selesai**

## VII. KESIMPULAN

Program ini cukup lengkap untuk kebutuhan pengelolaan program kerja organisasi. Dengan fungsi ekspor ke Excel, program ini memberikan kemudahan dalam menyimpan dan membagikan data. Kelebihannya:

1. Memiliki validasi input yang baik.
2. Antarmuka pengguna yang sederhana.
3. Memanfaatkan pustaka **openpyxl** untuk fitur ekspor data.