

**TUGAS STUDI KASUS GUI
MENGUNAKAN PYTHON
“MANAJEMEN PERPUSTAKAAN SEDERHANA”
PEMROGRAMAN BERORIENTASI OBJEK PRAKTIK**

Kelas VIII

T.A. Semester Genap 2024/2025



Disusun Oleh :

Raihan Ramadhan Indratmo

5230411282

**PROGRAM STUDI INFORMATIKA
FAKULTAS SAINS & TEKNOLOGI
UNIVERSITAS TEKNOLOGI YOGYAKARTA
YOGYAKARTA**

2024

A. Deskripsi Studi Kasus

Perpustakaan pada umumnya menghadapi tantangan besar dalam mengelola koleksi buku, penelusuran, dan pencatatan data. Metode konvensional menggunakan kartu katalog dan pencatatan manual rentan terhadap kesalahan. Maka dari itu SiDaPus hadir untuk memberikan solusi. SiDaPus (Sistem Data Perpustakaan) merupakan aplikasi GUI sederhana berbasis Python dan Tkinter, yang dirancang untuk membantu mengelola data buku di sebuah perpustakaan. Aplikasi ini memungkinkan petugas perpustakaan untuk melakukan manajemen dasar seperti menambah, menghapus, mencari, dan meminjam buku dengan antarmuka yang sederhana serta mudah digunakan. SiDaPus memiliki beberapa fitur diantaranya :

1. Manajemen Data Buku
 - Menambahkan buku baru dalam sistem
 - Menghapus buku dari sistem
 - Menampilkan daftar buku dalam tabel
2. Validasi Data
 - Memastikan semua field dalam data buku terisi
 - Mengharuskan tahun terbit berupa angka
 - Memastikan stok buku tidak negatif
3. Pencarian Buku
 - Mencari buku berdasarkan judul
 - Fitur clear
4. Peminjaman Buku
 - Mengurangi stok buku saat dipinjam
 - Mencegah peminjaman ketika permintaan > stok

B. Pembahasan Kode Program

a.

```
# Import semua fungsi, modul, dan kelas dari library tkinter (tidak mencakup submodul)
import tkinter as tk
# Merupakan submodul maka harus di import lagi (tidak include dengan *), agar tersedia secara eksplisit dalam kode
from tkinter import messagebox, ttk
```

b. Class SistemSidapus:

Class ini adalah inti program, di mana seluruh komponen GUI dan logic program didefinisikan.

```
class SistemSidapus:
    def __init__(self, root):
        self.root = root
        # Menamai program
        self.root.title("SiDaPus | Sistem Data Perpustakaan")
        self.root.geometry("700x600") # Ukuran
        self.root.resizable(False, False) # Tidak dapat dirubah (tetap)

        # Data Penyimpanan Buku
        self.book_data = [] # List untuk simpan data buku
```

c. Membuat User Interface

Disini saya menggunakan 3 wrapper (frame) untuk memudahkan letak komponen.

```
# Membuat User Interface
self.create_ui()

def create_ui(self):
    # Wrapper (frames yang akan saya gunakan 3 frame)
    # Komponen tabel ada di wrapper 1
    self.wrapper1 = tk.LabelFrame(self.root, text="Daftar Buku")
    # Komponen search ada di dalam wrapper 2 (include dengan 2 button)
    self.wrapper2 = tk.LabelFrame(self.root, text="Pencarian")
    # Komponen data buku ada di dalam wrapper 3 (include dengan 3 button)
    self.wrapper3 = tk.LabelFrame(self.root, text="Data Buku")

    self.wrapper1.pack(fill="both", expand="yes", padx=20, pady=20)
    self.wrapper2.pack(fill="both", padx=20, pady=20)
    self.wrapper3.pack(fill="both", padx=20, pady=20)
```

d. Variabel Form (Variabel Penghubung)

Variabel penghubung digunakan untuk menampung isi form

```
# Membuat Variabel Penghubung
self.setup_form_variables()

# Buat label dan input
self.create_form_fields()

# Buat Tabel
self.create_table()

# Buat search
self.create_search_components()

def setup_form_variables(self):
    # variabel penghubung antara input dan data
    self.v_judul = tk.StringVar()
    self.v_kategori = tk.StringVar()
    self.v_no_rak = tk.StringVar()
    self.v_penulis = tk.StringVar()
    self.v_penerbit = tk.StringVar()
    self.v_tahun_terbit = tk.StringVar()
    self.v_stok = tk.IntVar()
    self.v_stok.set(0)
    self.q = tk.StringVar()
```

e. Form Input Buku

Dalam form input buku akan ada 2 kolom. Kolom 1 berisi judul, kategori (10 pilihan kategori), no rak (5 pilihan no rak), dan penulis. Sedangkan di kolom 2 terdapat penerbit, tahun terbit, dan stock. Selain itu juga akan ada 3 button (tambah buku, hapus, dan pinjam buku). Ini semua tersimpan pada wrapper 3 (frame 3).

```
def create_form_fields(self):
    # kolom 1
    fields = [
        ("Judul", self.v_judul, tk.Entry),
        ("Kategori", self.v_kategori, ttk.Combobox,
         {'values': ('000 - Umum', '100 - Filsafat dan Psikologi',
                    '200 - Agama', '300 - Ilmu - Ilmu Sosial',
                    '400 - Bahasa', '500 - Ilmu Pengetahuan Alam dan Matematika',
                    '600 - Teknologi dan Ilmu - Ilmu Terapan',
                    '700 - Seni, Hiburan, dan Olahraga',
                    '800 - Sastra', '900 - Geografi dan Sejarah')})),
        ("No Rak", self.v_no_rak, ttk.Combobox, {
         'values': ('001', '002', '003', '004', '005')})),
        ("Penulis", self.v_penulis, tk.Entry)
    ]

    # kolom 2
    second_column_fields = [
        ("Penerbit", self.v_penerbit, tk.Entry),
        ("Tahun Terbit", self.v_tahun_terbit, tk.Entry),
        ("Stok", self.v_stok, tk.Entry)
    ]

    for i, (label_text, var, field_type, *extra_args) in enumerate(fields):
        label = tk.Label(self.wrapper3, text=label_text)
        label.grid(row=i, column=0, sticky="w", pady=4)

        if extra_args:
            field = field_type(self.wrapper3, width=17,
                               textvariable=var, **extra_args[0])
        else:
            field = field_type(self.wrapper3, textvariable=var)

        field.grid(row=i, column=1, columnspan=2,
                  sticky="w", pady=4, padx=10)

    for i, (label_text, var, field_type, *extra_args) in enumerate(second_column_fields):
        label = tk.Label(self.wrapper3, text=label_text)
        label.grid(row=i, column=3, sticky="w", pady=4, padx=10)

        if extra_args:
            field = field_type(
                self.wrapper3, textvariable=var, **extra_args[0])
        else:
            field = field_type(self.wrapper3, textvariable=var)

        field.grid(row=i, column=4, columnspan=2, sticky="w", pady=4)
```

```

frame_btn = tk.Frame(self.wrapper3)
frame_btn.grid(row=4, column=0, columnspan=5, sticky="w", pady=10)

add_btn = tk.Button(frame_btn, text="Tambah Baru",
                    command=self.add_new)
delete_btn = tk.Button(frame_btn, text="Hapus",
                       command=self.delete_book)
borrow_btn = tk.Button(
    frame_btn, text="Pinjam Buku", command=self.borrow_book)

add_btn.pack(side=tk.LEFT, padx=5)
delete_btn.pack(side=tk.LEFT, padx=5)
borrow_btn.pack(side=tk.LEFT, padx=5)

```

f. Buat Tabel

Tabel yang akan dibuat berada di wrapper 1 (frame 1), yang terdiri dari 7 kolom (judul, kategori, no rak buku, penulis, penerbit, tahun terbit, dan stok). Selain itu juga disediakan scrollbar vertical dan horizontal untuk menunjang kenyamanan dalam membaca data.

```

def create_table(self):
    # Tabel Data Buku
    self.trv = ttk.Treeview(self.wrapper1, column=(1, 2, 3, 4, 5, 6, 7),
                           show="headings", height=5)
    self.trv.bind("<<TreeviewSelect>>", self.on_table_select)

    # Headings Tabel
    headings = ["Judul", "Kategori", "No Rak Buku",
               "Penulis", "Penerbit", "Tahun Terbit", "Stok"]
    for i, heading in enumerate(headings, 1):
        self.trv.heading(i, text=heading)

    column_configs = [
        (1, 95, 135, 'center'),
        (2, 95, 135, 'center'),
        (3, 95, 135, 'center'),
        (4, 95, 135, 'center'),
        (5, 95, 135, 'center'),
        (6, 95, 135, 'center'),
        (7, 65, 105, 'center')
    ]

    for col, width, minwidth, anchor in column_configs:
        self.trv.column(col, width=width, minwidth=minwidth,
                        anchor=anchor)

```

```
# Scrollbars
y scrollbar = tk.Scrollbar(
    self.wrapper1, orient="vertical", command=self.trv.yview)
y scrollbar.pack(side=tk.RIGHT, fill="y")

x scrollbar = tk.Scrollbar(
    self.wrapper1, orient="horizontal", command=self.trv.xview)
x scrollbar.pack(side=tk.BOTTOM, fill="x")

self.trv.configure(yscrollcommand=y scrollbar.set,
                   xscrollcommand=x scrollbar.set)
self.trv.pack(fill=tk.BOTH, expand=True)
```

g. Pencarian Buku

Buku dapat dicari dengan pencarian buku yang ada di wrapper 2 (frame 2). Hal ini bertujuan untuk meningkatkan efisiensi waktu saat mencari data yang diperlukan.

```
def create_search_components(self):

    lbl = tk.Label(self.wrapper2, text="Search")
    lbl.pack(side=tk.LEFT, padx=10, pady=15)

    ent = tk.Entry(self.wrapper2, textvariable=self.q)
    ent.pack(side=tk.LEFT, padx=6, pady=15)

    btn = tk.Button(self.wrapper2, text="Search", command=self.search)
    btn.pack(side=tk.LEFT, padx=6, pady=15)

    cbtn = tk.Button(self.wrapper2, text="Clear", command=self.clear)
    cbtn.pack(side=tk.LEFT, padx=6, pady=15)
```

h. Tambah Buku

Dalam menambahkan buku, semua data dalam wrapper 3 (frame 3) wajib diisi agar lolos validasi. Selain itu tahun terbit juga harus diisi hanya dengan angka.

```
def add_new(self):
    # Validasi data harus diisi semua
    if not (self.v_judul.get() and self.v_kategori.get() and
            self.v_no_rak.get() and self.v_penulis.get() and
            self.v_penerbit.get() and self.v_tahun_terbit.get() and
            self.v_stok.get()):
        messagebox.showwarning("Peringatan", "Semua data harus diisi!")
        return

    # Validasi tahun hanya angka
    if not self.v_tahun_terbit.get().isdigit():
        messagebox.showerror("Error", "Tahun terbit harus berupa angka!")
        return

    # Validasi stok harus berupa angka
    try:
        stok = int(self.v_stok.get())
        if stok < 0:
            messagebox.showerror("Error", "Stok tidak boleh negatif!")
            return
    except ValueError:
        messagebox.showerror("Error", "Stok harus berupa angka!")
        return
```

```
# Jika validasi lolos, data disimpan
new_book = {
    "judul": self.v_judul.get(),
    "kategori": self.v_kategori.get(),
    "no_rak": self.v_no_rak.get(),
    "penulis": self.v_penulis.get(),
    "penerbit": self.v_penerbit.get(),
    "tahun_terbit": self.v_tahun_terbit.get(),
    "stok": stok
}
self.book_data.append(new_book)
self.refresh_table()
self.clear_form()
messagebox.showinfo("Sukses", "Buku berhasil ditambahkan.")
```

i. Hapus Buku

```
def delete_book(self):
    selected_item = self.trv.selection()
    if not selected_item:
        messagebox.showwarning(
            "Peringatan", "Pilih buku yang ingin dihapus!")
        return
    item_index = self.trv.index(selected_item[0])
    del self.book_data[item_index]
    self.refresh_table()
    self.clear_form()
    messagebox.showinfo("Sukses", "Buku berhasil dihapus.")
```

j. Pinjam Buku

Dalam meminjam buku pastikan stok yang ingin dipinjam < stok yang tersedia. Jika tidak akan muncul sebuah message box (error).

```
def borrow_book(self):
    selected_item = self.trv.selection()
    if not selected_item:
        messagebox.showwarning(
            "Peringatan", "Pilih buku yang ingin dipinjam!")
        return

    try:
        jumlah_pinjam = int(self.v_stok.get())
        if jumlah_pinjam <= 0:
            messagebox.showerror(
                "Error", "Jumlah pinjam harus lebih dari nol!")
            return

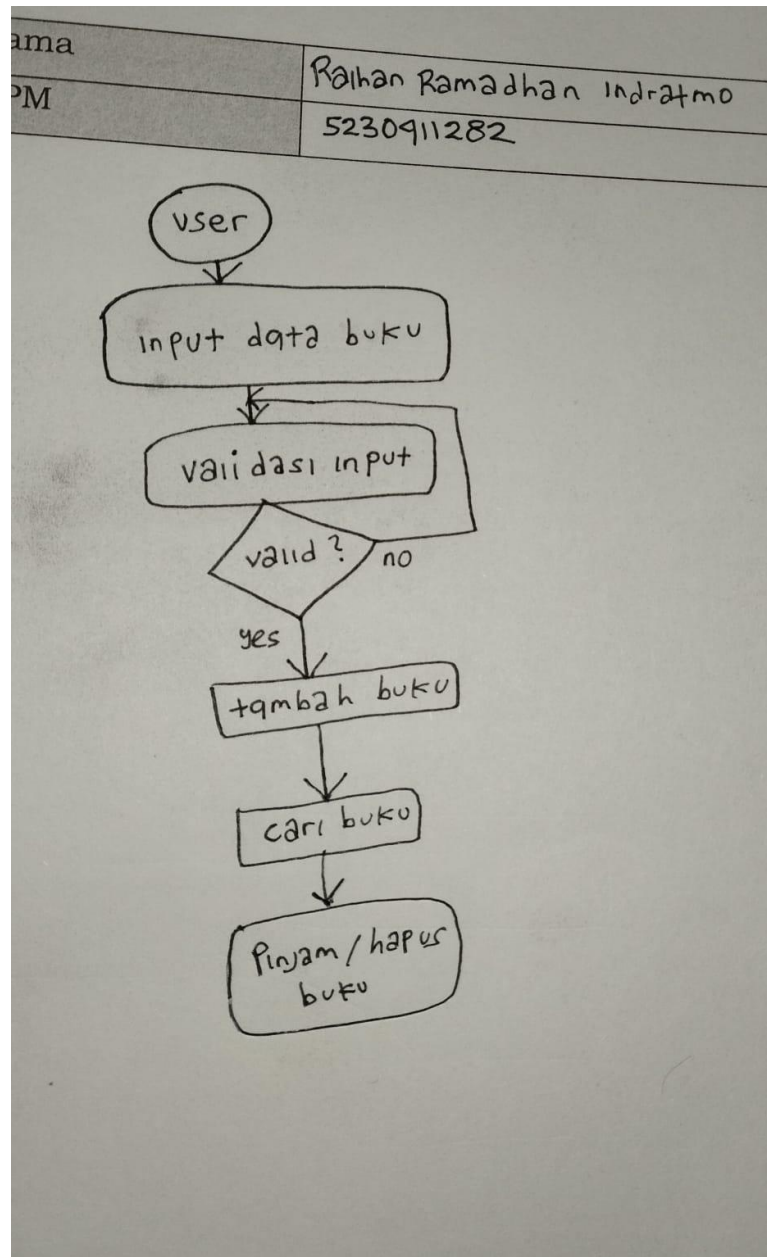
        item_index = self.trv.index(selected_item[0])
        if self.book_data[item_index]["stok"] >= jumlah_pinjam:
            self.book_data[item_index]["stok"] -= jumlah_pinjam
            self.refresh_table()
            self.clear_form()
            messagebox.showinfo("Sukses", "Buku berhasil dipinjam.")
        else:
            messagebox.showerror("Error", "Stok buku tidak mencukupi.")
    except ValueError:
        messagebox.showerror(
            "Error", "Masukkan jumlah pinjam sebagai angka!")
```

k. Jalan Program

```
def main():  
    root = tk.Tk()  
    app = SistemSidapus(root)  
    root.mainloop()  
  
if __name__ == "__main__":  
    main()
```


C. Activity Diagram

Berikut merupakan activity diagram dari program GUI python yang saya buat :



D. Implementasi

1. Running Program

The screenshot shows the main window of the SiDaPus application. It features a title bar with the text "SiDaPus | Sistem Data Perpustakaan". Below the title bar, there is a section titled "Daftar Buku" which contains a table with columns: "Judul", "Kategori", "No Rak Buku", "Penulis", and "Penerbit". The table is currently empty. Below the table, there is a "Pencarian" (Search) section with a text input field, a "Search" button, and a "Clear" button. At the bottom, there is a "Data Buku" section with form fields for "Judul", "Penerbit", "Kategori", "Tahun Terbit", "No Rak", "Stok", and "Penulis". The "Stok" field has a value of "0". Below these fields are three buttons: "Tambah Baru", "Hapus", and "Pinjam Buku".

2. Inputkan Data Buku

This screenshot shows the "Data Buku" form with the following input data:

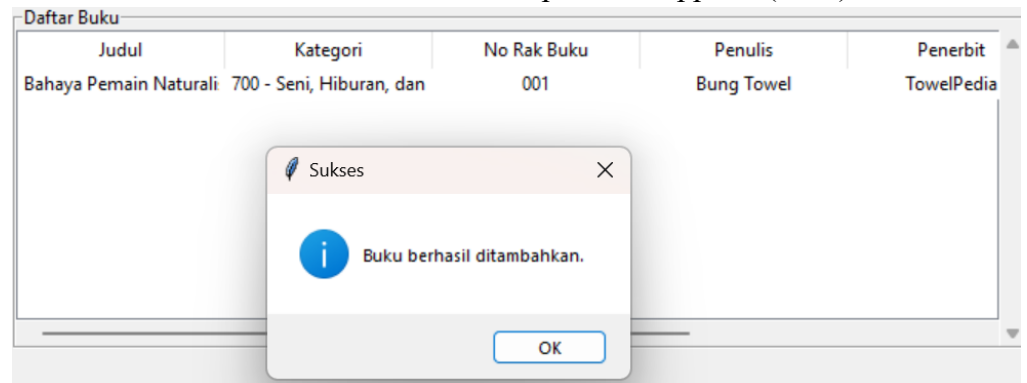
Field	Value
Judul	ya Pemain Naturalisasi
Penerbit	TowelPedia
Kategori	700 - Seni, Hiburan, ▾
Tahun Terbit	2024
No Rak	001 ▾
Stok	99
Penulis	Bung Towel

Jika ada salah satu data yang belum diisi atau mengisi tahun terbit dengan selain angka, maka akan muncul validasi berupa message box.

The first screenshot shows an "Error" message box with the text "Tahun terbit harus berupa angka!" (Year must be a number!). The second screenshot shows a "Peringatan" (Warning) message box with the text "Semua data harus diisi!" (All data must be filled!). Both message boxes have an "OK" button.

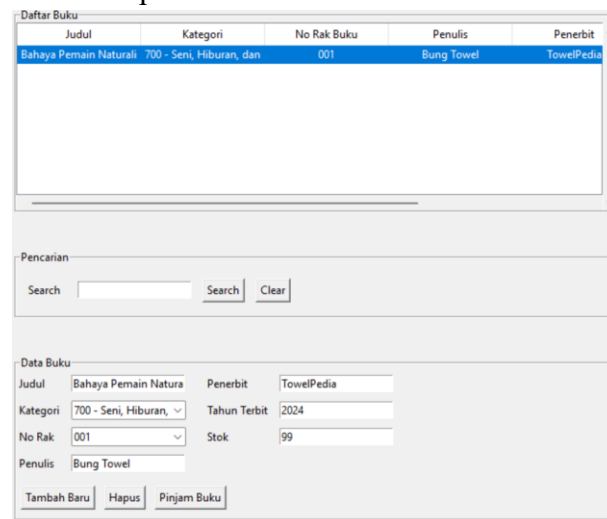
3. Pencet Tombol Tambah Baru

Jika sudah mengisi semua tabel dan lolos validasi maka pencet tombol tambah baru dan data akan tersimpan di wrapper 1 (tabel).

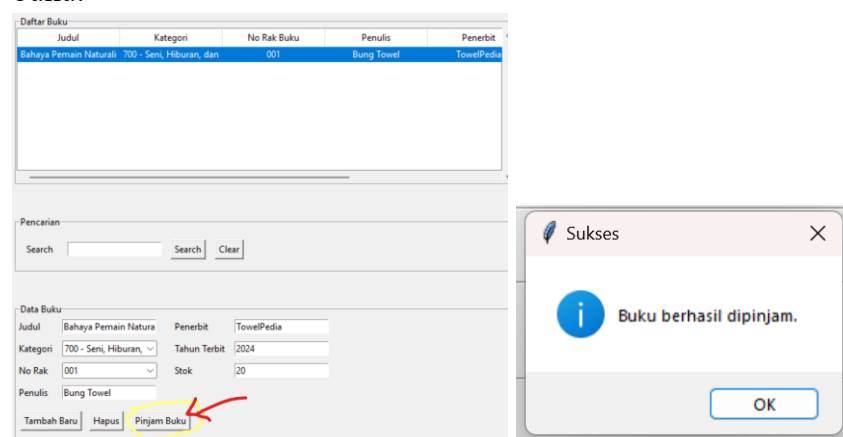


4. Meminjam Barang

Jika ingin memilih buku untuk dipinjam, cukup pencet saja buku yg berkaitan pada tabel. Maka data buku akan terisi sendiri.



Setelah itu pilih stok yang akan dipinjam kemudian klik pinjam buku.



Maka stok yang ada di tabel akan berkurang sesuai dengan pinjaman yang diinputkan tadi ($99 - 20 = 79$).

Daftar Buku				
No Rak Buku	Penulis	Penerbit	Tahun Terbit	Stok
001	Bung Towel	TowelPedia	2024	79

5. Mencari Data

Jika data banyak dan kita ingin mencari sebuah buku, ketikkan saja judul buku pada search. Misal ada 3 buku :

Daftar Buku				
Judul	Kategori	No Rak Buku	Penulis	Penerbit
shaya Pemain Naturali	700 - Seni, Hiburan, dan	001	Bung Towel	TowelPedia
Towel	100 - Filsafat dan Psikolo	001	Justin	Ajaks
bah Kak Gem Jadi Sald	600 - Teknologi dan Ilmu	002	Agus	Agus Lapar Bu

Saya ingin mencari buku Towel, maka saya hanya perlu menuliskan Towel pada search bar.

Daftar Buku				
Judul	Kategori	No Rak Buku	Penulis	Penerbit
Towel	100 - Filsafat dan Psikolo	001	Justin	Ajaks

Pencarian	
Search	<input type="text" value="Towel"/> <input type="button" value="Search"/> <input type="button" value="Clear"/>

6. Hapus Buku

Jika dirasa sebuah kejayaan dari buku tersebut telah usai, rusak, atau hal lain yang mendasari buku tersebut untuk tidak dipakai lagi, maka pengguna dapat menghapusnya dari sistem melalui tombol hapus.

E. Evaluasi

Mungkin program GUI sederhana yang saya buat masih jauh dari kata layak untuk digunakan. Maka beberapa point mungkin akan menjadi evaluasi saya, diantaranya :

- Penyimpanan data masih bersifat sementara (tidak ada database)
- Keterbatasan fitur (tidak ada nama peminjam, edit data buku, dll)
- Program yang saya buat sangat sederhana