**Tugas Pembuatan Aplikasi Perpustakaan dengan OOP Python dan Streamlit**

**Dosen Pengampu: budiman SMP**

****

**Disusun oleh :**

**Muhammad Faaiz Al Faqih (22010058)**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER**

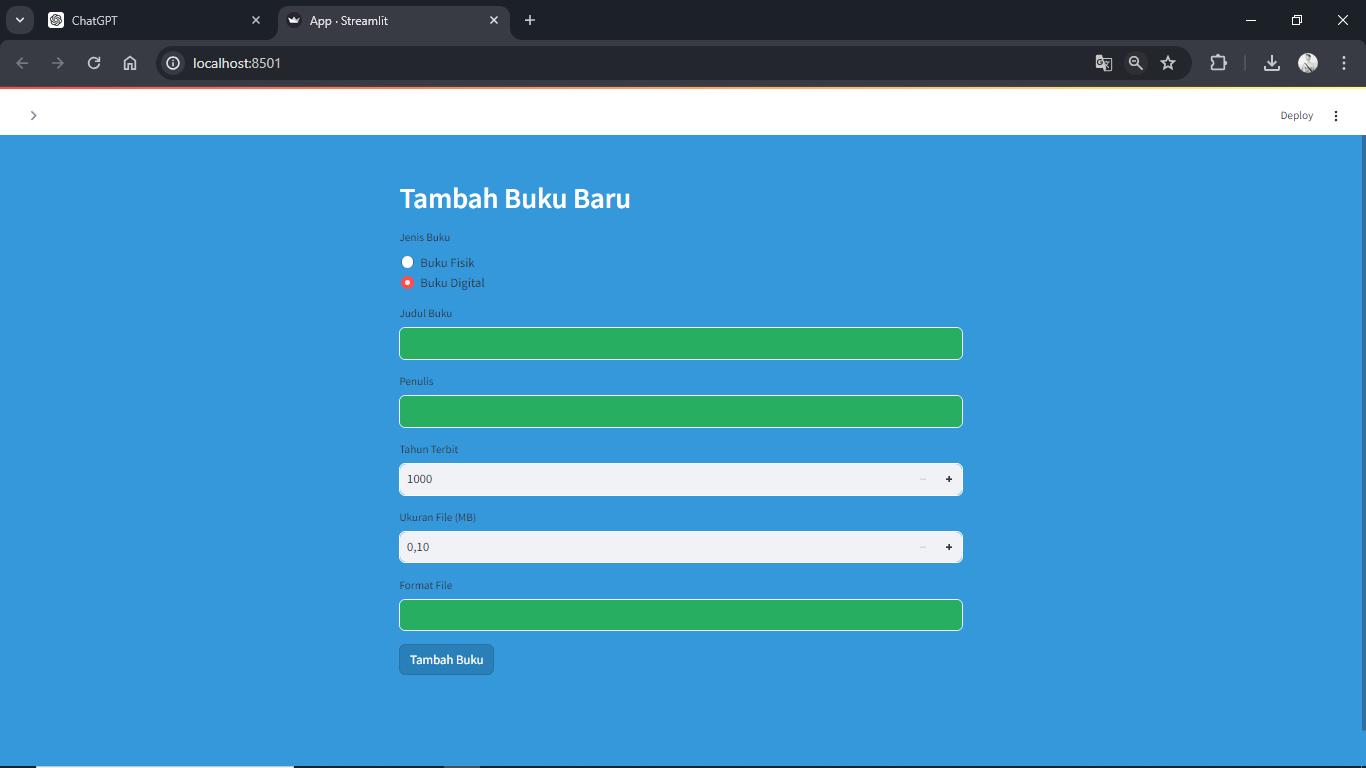
**DHARMA WACANA METRO**

**2023/2024**

## 1. Penjelasan Aplikasi Perpustakaan Digital

Aplikasi ini adalah sistem manajemen perpustakaan berbasis web yang dibangun menggunakan Streamlit. Aplikasi ini memungkinkan pengguna untuk menambah, mencari, menampilkan, meminjam, mengembalikan, menghapus, dan memperbarui informasi buku, baik buku fisik maupun digital. Berikut adalah penjelasan rinci tentang fitur-fitur yang tersedia :

1. Tambah Buku



 Jenis Buku

Pilihan antara Buku Fisik atau Buku Digital.

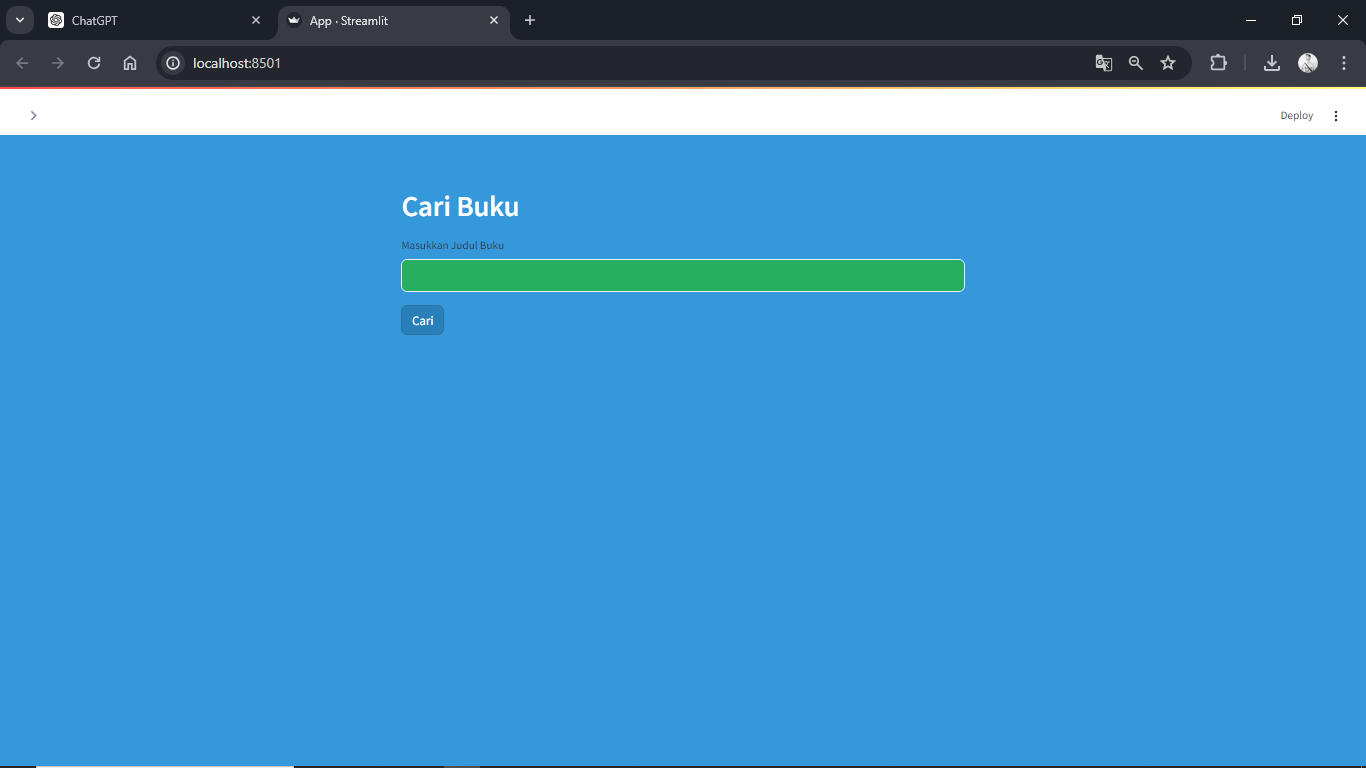
**** Input Data

Judul, penulis, tahun terbit, dan atribut spesifik (jumlah halaman & berat untuk buku fisik; ukuran file & format untuk buku digital).

 Proses

Setelah data diinput, buku akan ditambahkan ke perpustakaan dan data akan disimpan dalam file Excel.\

#### 2. **Cari Buku**



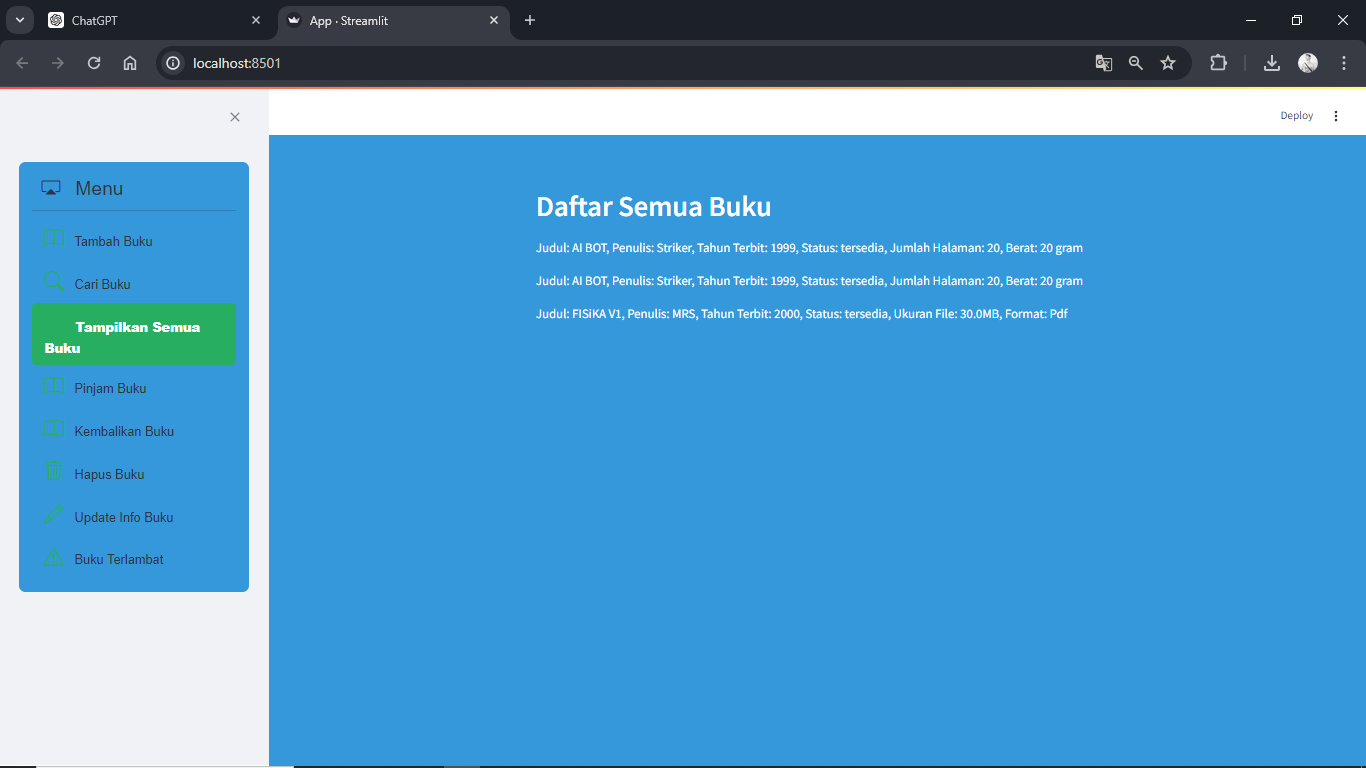
* **Input**

Judul buku.

* **Proses**

Sistem akan mencari buku berdasarkan judul yang dimasukkan dan menampilkan informasi buku jika ditemukan.

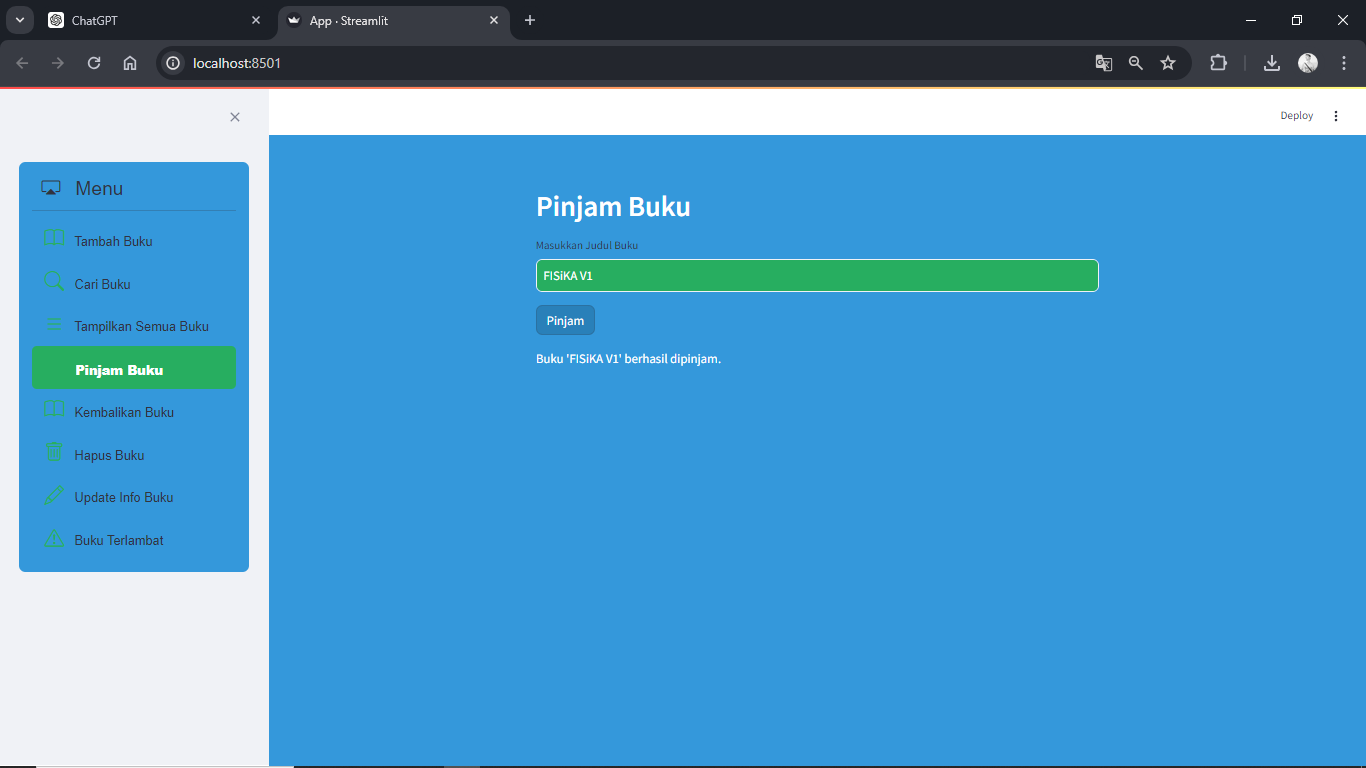
#### 3. **Tampilkan Semua Buku**



* **Proses**

Menampilkan daftar semua buku yang tersedia di perpustakaan dengan informasi lengkap.

#### 4. **Pinjam Buku**



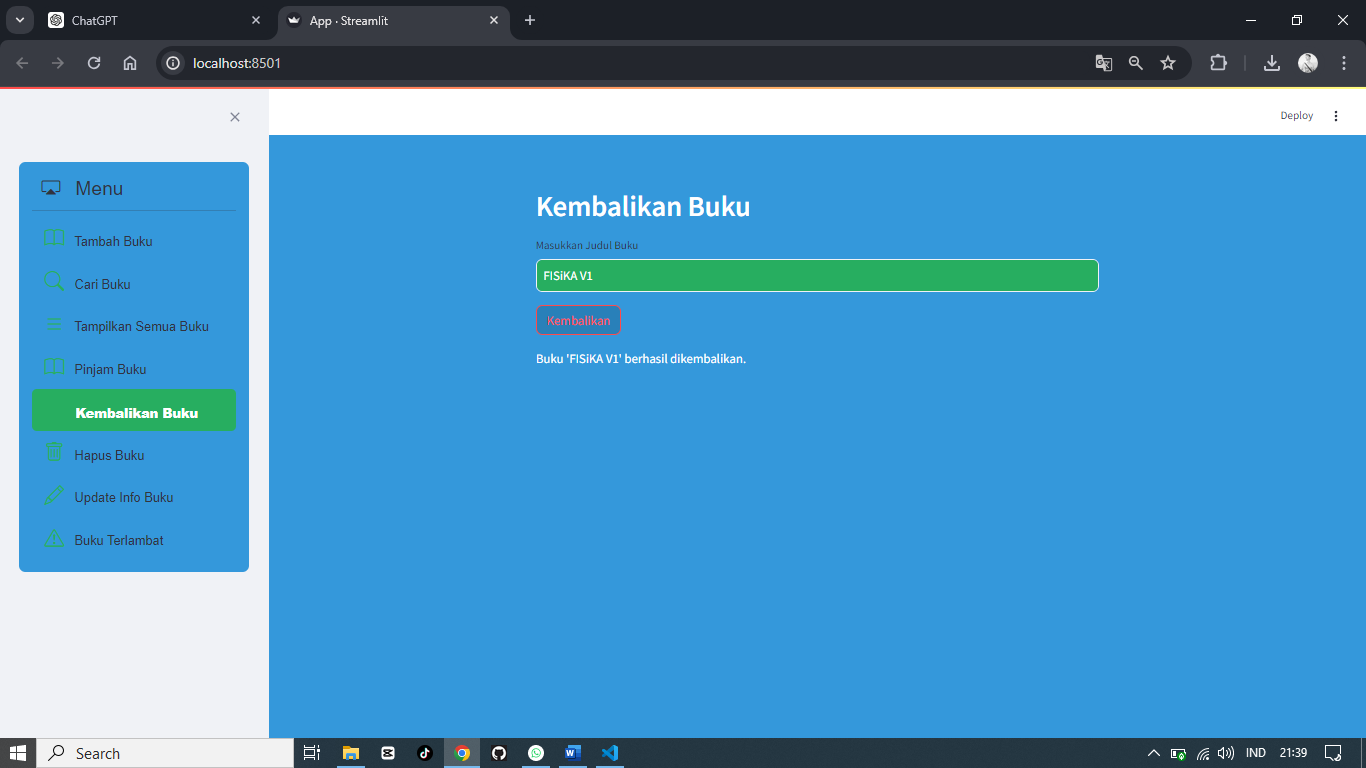
* **Input**

Judul buku.

* **Proses**

Sistem akan memeriksa ketersediaan buku dan mengubah statusnya menjadi dipinjam jika tersedia.

#### 5. **Kembalikan Buku**



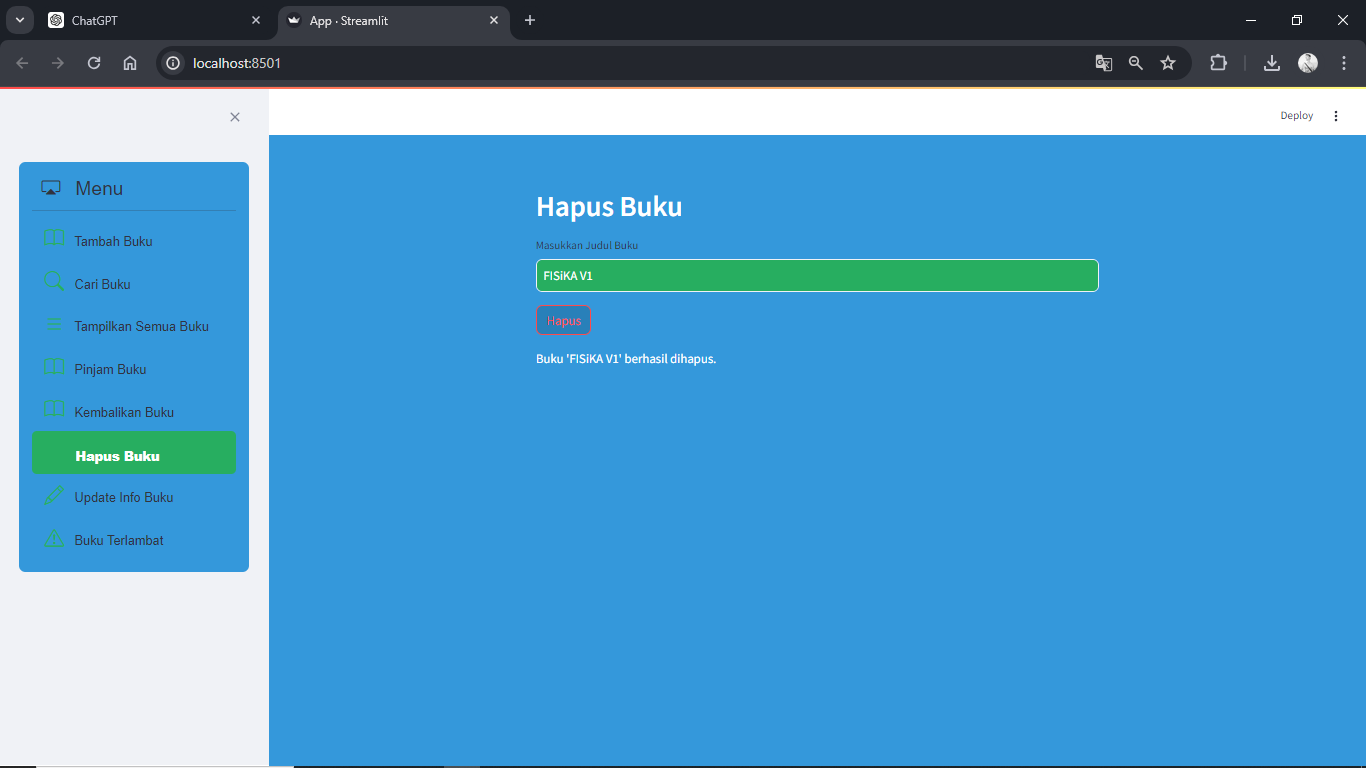
* **Input**

Judul buku.

* **Proses**

Sistem akan mengubah status buku menjadi tersedia kembali jika buku tersebut sedang dipinjam.

#### 6. **Hapus Buku**



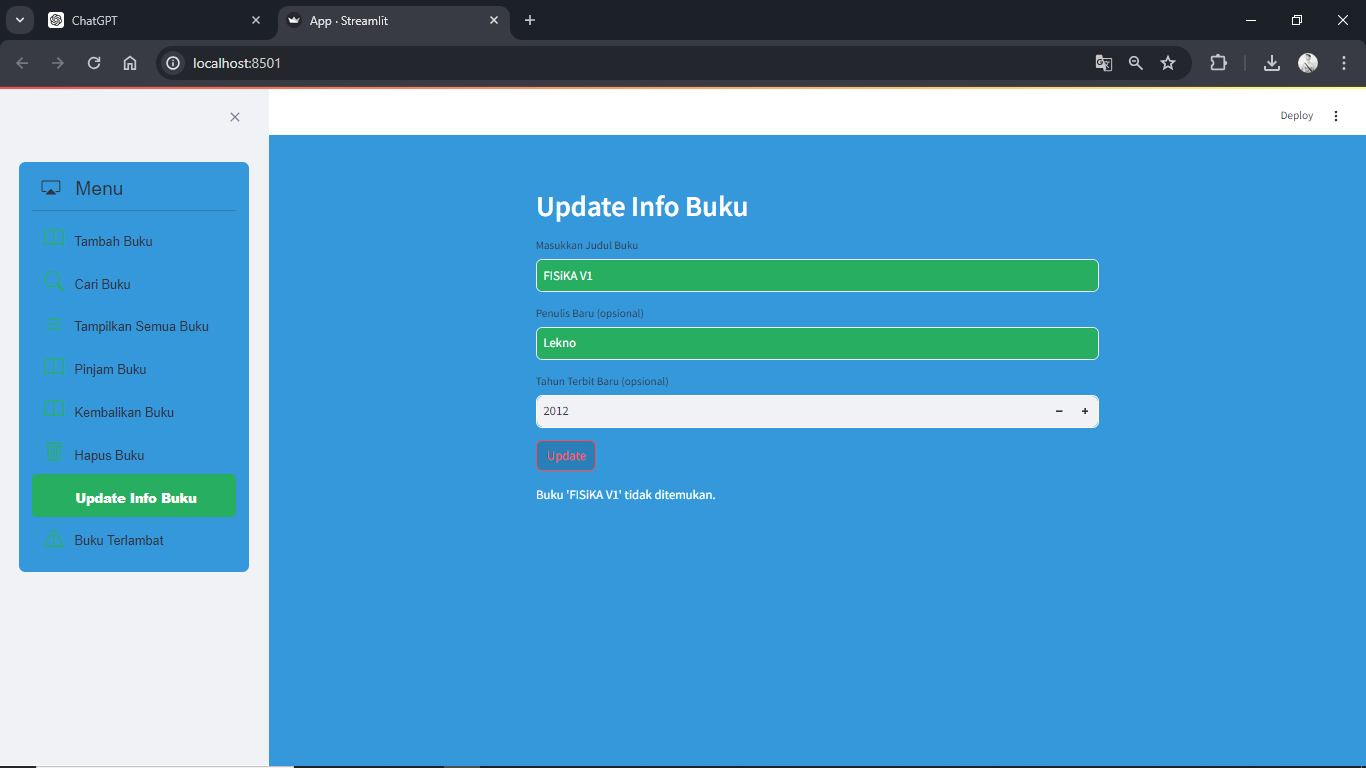
* **Input**

Judul buku.

* **Proses**

Sistem akan menghapus buku dari daftar perpustakaan jika buku ditemukan.

#### 7. **Update Info Buku**



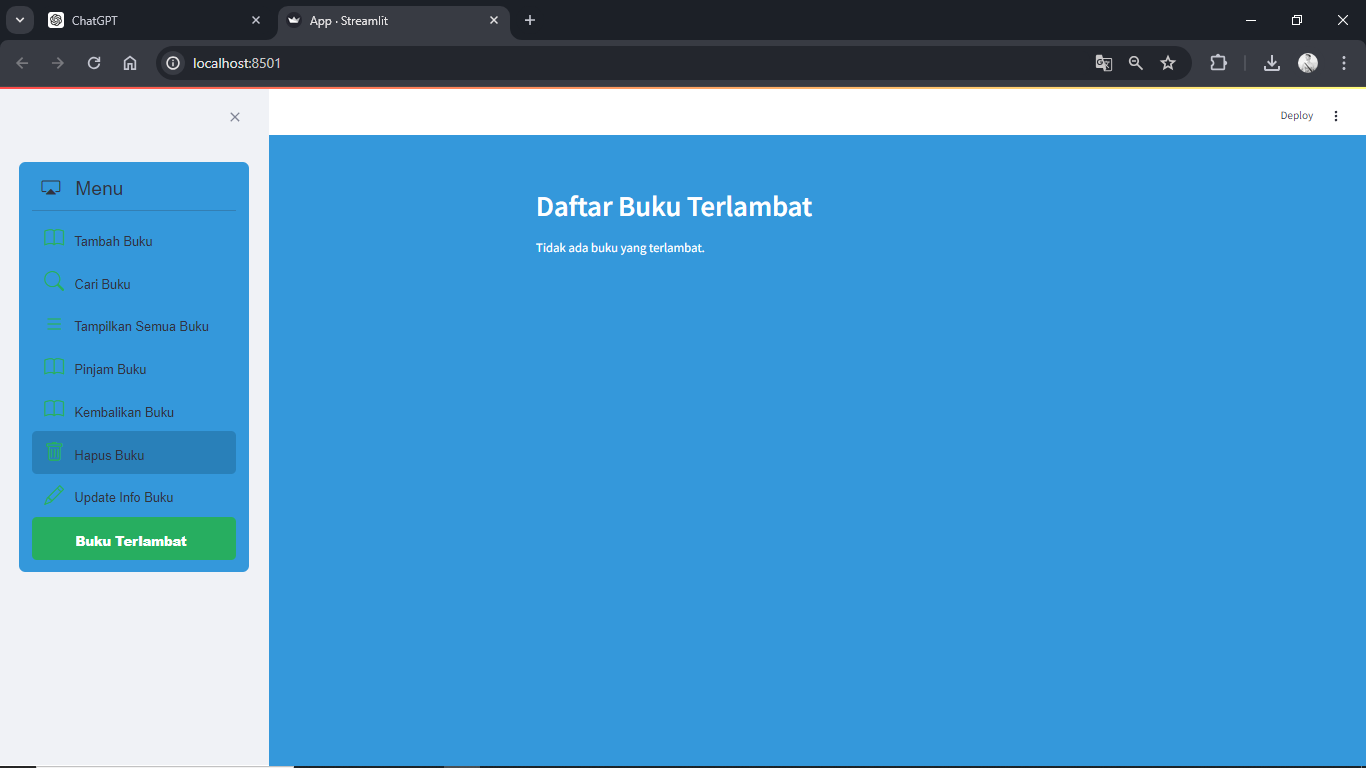
* **Input**

Judul buku, penulis baru, dan tahun terbit baru (opsional).

* **Proses**

Sistem akan memperbarui informasi buku sesuai dengan input yang diberikan.

#### 8. **Buku Terlambat**



* **Proses**

Menampilkan daftar buku yang terlambat dikembalikan (lebih dari 7 hari sejak tanggal peminjaman) beserta informasi keterlambatan.

# A. Struktur Kode

* **Kelas dan Turunannya:**
  + Buku: Kelas dasar untuk buku dengan atribut umum.
  + BukuDigital: Kelas turunan untuk buku digital dengan atribut tambahan ukuran file dan format.
  + BukuFisik: Kelas turunan untuk buku fisik dengan atribut tambahan jumlah halaman dan berat.
* **Kelas Perpustakaan:**
  + Mengelola daftar buku dengan metode untuk menambah, mencari, menampilkan, meminjam, mengembalikan, menghapus, dan memperbarui buku.
* **Fungsi Simpan Data:**
  + simpan\_data\_excel(perpustakaan): Menyimpan data buku dalam format Excel.

## B. Tampilan dan Antarmuka

* **Streamlit Option Menu:**
  + Sidebar dengan pilihan menu untuk navigasi fitur.
  + Gaya khusus untuk antarmuka dengan penyesuaian warna dan elemen UI.
* **Interaksi Pengguna:**
  + Input melalui text input, radio buttons, dan tombol.
  + Pesan konfirmasi dan peringatan untuk setiap aksi yang dilakukan pengguna.

**Code Program**

import streamlit as st

from streamlit\_option\_menu import option\_menu

import datetime

import pandas as pd

# Kelas Buku dan Turunannya

class Buku:

def \_\_init\_\_(self, judul, penulis, tahun\_terbit):

self.judul = judul

self.penulis = penulis

self.tahun\_terbit = tahun\_terbit

self.status = "tersedia"

self.tanggal\_peminjaman = None

def info\_buku(self):

return f"Judul: {self.judul}, Penulis: {self.penulis}, Tahun Terbit: {self.tahun\_terbit}, Status: {self.status}"

class BukuDigital(Buku):

def \_\_init\_\_(self, judul, penulis, tahun\_terbit, ukuran\_file, format\_file):

super().\_\_init\_\_(judul, penulis, tahun\_terbit)

self.ukuran\_file = ukuran\_file

self.format\_file = format\_file

def info\_buku(self):

info = super().info\_buku()

return f"{info}, Ukuran File: {self.ukuran\_file}MB, Format: {self.format\_file}"

class BukuFisik(Buku):

def \_\_init\_\_(self, judul, penulis, tahun\_terbit, jumlah\_halaman, berat):

super().\_\_init\_\_(judul, penulis, tahun\_terbit)

self.jumlah\_halaman = jumlah\_halaman

self.berat = berat

def info\_buku(self):

info = super().info\_buku()

return f"{info}, Jumlah Halaman: {self.jumlah\_halaman}, Berat: {self.berat} gram"

class Perpustakaan:

def \_\_init\_\_(self):

self.daftar\_buku = []

def tambah\_buku(self, buku):

self.daftar\_buku.append(buku)

def cari\_buku(self, judul):

for buku in self.daftar\_buku:

if buku.judul.lower() == judul.lower():

return buku

return None

def tampilkan\_semua\_buku(self):

return [buku.info\_buku() for buku in self.daftar\_buku]

def pinjam\_buku(self, judul):

buku = self.cari\_buku(judul)

if buku and buku.status == "tersedia":

buku.status = "dipinjam"

buku.tanggal\_peminjaman = datetime.datetime.now()

return f"Buku '{judul}' berhasil dipinjam."

else:

return f"Buku '{judul}' tidak tersedia untuk dipinjam."

def kembalikan\_buku(self, judul):

buku = self.cari\_buku(judul)

if buku and buku.status == "dipinjam":

buku.status = "tersedia"

buku.tanggal\_peminjaman = None

return f"Buku '{judul}' berhasil dikembalikan."

else:

return f"Buku '{judul}' tidak sedang dipinjam."

def hapus\_buku(self, judul):

buku = self.cari\_buku(judul)

if buku:

self.daftar\_buku.remove(buku)

return f"Buku '{judul}' berhasil dihapus."

else:

return f"Buku '{judul}' tidak ditemukan."

def update\_info\_buku(self, judul, penulis=None, tahun\_terbit=None):

buku = self.cari\_buku(judul)

if buku:

if penulis:

buku.penulis = penulis

if tahun\_terbit:

buku.tahun\_terbit = tahun\_terbit

return f"Informasi buku '{judul}' berhasil diperbarui."

else:

return f"Buku '{judul}' tidak ditemukan."

def buku\_terlambat(self):

buku\_terlambat = []

for buku in self.daftar\_buku:

if buku.status == "dipinjam" and buku.tanggal\_peminjaman:

if (datetime.datetime.now() - buku.tanggal\_peminjaman).days > 7:

buku\_terlambat.append(buku)

return buku\_terlambat

# Fungsi untuk menyimpan data ke file Excel

def simpan\_data\_excel(perpustakaan):

data = []

for buku in perpustakaan.daftar\_buku:

data.append([buku.judul, buku.penulis, buku.tahun\_terbit, buku.status])

df = pd.DataFrame(data, columns=["Judul", "Penulis", "Tahun Terbit", "Status"])

df.to\_excel("daftar\_buku.xlsx", index=False)

# Inisialisasi perpustakaan di session state

if 'perpustakaan' not in st.session\_state:

st.session\_state.perpustakaan = Perpustakaan()

perpustakaan = st.session\_state.perpustakaan

# Pilihan menu

with st.sidebar:

selected = option\_menu(

menu\_title="Menu",

options=["Tambah Buku", "Cari Buku", "Tampilkan Semua Buku", "Pinjam Buku", "Kembalikan Buku", "Hapus Buku", "Update Info Buku", "Buku Terlambat"],

icons=["book", "search", "list", "book", "book", "trash", "pencil", "exclamation-triangle"],

menu\_icon="cast",

default\_index=0,

styles={

"container": {"padding": "5px", "background-color": "#3498db"},

"icon": {"color": "#27ae60", "font-size": "25px"},

"nav-link": {"font-size": "16px", "text-align": "left", "margin": "0px", "--hover-color": "#2980b9"},

"nav-link-selected": {"background-color": "#27ae60"},

}

)

st.markdown(

"""

<style>

.main {

background-color: #3498db;

color: #fff;

}

h1, h2, h3, h4, h5, h6 {

color: #fff;

}

.stTextInput > div > div > input {

color: #fff;

background-color: #27ae60;

}

.stNumberInput > div > div > input {

color: #fff;

background-color: #27ae60;

}

.stButton > button {

color: #fff;

background-color: #2980b9;

}

.stRadio > div > div {

color: #fff;

}

.css-1d391kg {

background-color: #3498db;

color: #fff;

}

.css-1aumxhk {

background-color: #3498db;

color: #fff;

}

</style>

""",

unsafe\_allow\_html=True

)

if selected == "Tambah Buku":

st.header("Tambah Buku Baru")

jenis\_buku = st.radio("Jenis Buku", ("Buku Fisik", "Buku Digital"))

judul = st.text\_input("Judul Buku")

penulis = st.text\_input("Penulis")

tahun\_terbit = st.number\_input("Tahun Terbit", min\_value=1000, max\_value=datetime.datetime.now().year, step=1)

if jenis\_buku == "Buku Fisik":

jumlah\_halaman = st.number\_input("Jumlah Halaman", min\_value=1, step=1)

berat = st.number\_input("Berat (gram)", min\_value=1, step=1)

if st.button("Tambah Buku"):

buku = BukuFisik(judul, penulis, tahun\_terbit, jumlah\_halaman, berat)

perpustakaan.tambah\_buku(buku)

simpan\_data\_excel(perpustakaan)

st.success("Buku fisik berhasil ditambahkan.")

else:

ukuran\_file = st.number\_input("Ukuran File (MB)", min\_value=0.1, step=0.1)

format\_file = st.text\_input("Format File")

if st.button("Tambah Buku"):

buku = BukuDigital(judul, penulis, tahun\_terbit, ukuran\_file, format\_file)

perpustakaan.tambah\_buku(buku)

simpan\_data\_excel(perpustakaan)

st.success("Buku digital berhasil ditambahkan.")

elif selected == "Cari Buku":

st.header("Cari Buku")

judul = st.text\_input("Masukkan Judul Buku")

if st.button("Cari"):

buku = perpustakaan.cari\_buku(judul)

if buku:

st.write(buku.info\_buku())

else:

st.warning("Buku tidak ditemukan.")

elif selected == "Tampilkan Semua Buku":

st.header("Daftar Semua Buku")

semua\_buku = perpustakaan.tampilkan\_semua\_buku()

if semua\_buku:

for info\_buku in semua\_buku:

st.write(info\_buku)

else:

st.write("Tidak ada buku yang ditemukan.")

elif selected == "Pinjam Buku":

st.header("Pinjam Buku")

judul = st.text\_input("Masukkan Judul Buku")

if st.button("Pinjam"):

pesan = perpustakaan.pinjam\_buku(judul)

st.write(pesan)

elif selected == "Kembalikan Buku":

st.header("Kembalikan Buku")

judul = st.text\_input("Masukkan Judul Buku")

if st.button("Kembalikan"):

pesan = perpustakaan.kembalikan\_buku(judul)

st.write(pesan)

elif selected == "Hapus Buku":

st.header("Hapus Buku")

judul = st.text\_input("Masukkan Judul Buku")

if st.button("Hapus"):

pesan = perpustakaan.hapus\_buku(judul)

st.write(pesan)

elif selected == "Update Info Buku":

st.header("Update Info Buku")

judul = st.text\_input("Masukkan Judul Buku")

penulis\_baru = st.text\_input("Penulis Baru (opsional)")

tahun\_terbit\_baru = st.number\_input("Tahun Terbit Baru (opsional)", min\_value=1000, max\_value=datetime.datetime.now().year, step=1)

if st.button("Update"):

pesan = perpustakaan.update\_info\_buku(judul, penulis\_baru if penulis\_baru else None, tahun\_terbit\_baru if tahun\_terbit\_baru else None)

st.write(pesan)

elif selected == "Buku Terlambat":

st.header("Daftar Buku Terlambat")

buku\_terlambat = perpustakaan.buku\_terlambat()

if buku\_terlambat:

data = []

for buku in buku\_terlambat:

data.append([buku.judul, buku.penulis, buku.tahun\_terbit, buku.tanggal\_peminjaman, (datetime.datetime.now() - buku.tanggal\_peminjaman).days])

df = pd.DataFrame(data, columns=["Judul", "Penulis", "Tahun Terbit", "Tanggal Peminjaman", "Hari Terlambat"])

st.table(df)

else:

st.write("Tidak ada buku yang terlambat.")