

**LAPORAN PROJECT SIPUDI**  
**“Aplikasi Peminjaman Buku Digital”**

**Mata Kuliah**  
Pemrograman Berbasis Objek

**Dosen Pengampu**  
Rosita, S.Pd., M.Pd.



Disusun Oleh:

Aftita Choirunnisa

24091397083

2024C

**PROGRAM STUDI MANAJEMEN INFORMATIKA**  
**FAKULTAS VOKASI**  
**UNIVERSITAS NEGERI SURABAYA**  
**2025**

## DAFTAR ISI

<b>DAFTAR ISI .....</b>	2
<b>BAB I PENDAHULUAN.....</b>	3
1.1    Latar Belakang .....	3
1.2    Tujuan .....	3
1.3    Ruang Lingkup.....	3
<b>BAB II KONSEP OBJECT-ORIENTED PROGRAMMING.....</b>	4
2.1    Encapsulation .....	4
2.2    Inheritance .....	4
2.1    Polymorphism .....	4
<b>BAB III PERANCANGAN SISTEM .....</b>	5
3.1    Deskripsi Umum .....	5
3.2    Class Diagram .....	5
3.3    Penjelasan Class Diagram.....	6
<b>BAB IV IMPLEMENTASI KODE .....</b>	8
4.1    Kode program .....	8
4.2    Penjelasan kode (inline atau poin-poin) .....	9
4.3    Kerja Aplikasi/Sistem.....	17
<b>BAB V IMPLEMENTASI DAN PENGUJIAN APLIKASI.....</b>	18
5.1    Fitur .....	18
5.2    Cara Penggunaan .....	18
5.3    Tampilan Aplikasi .....	18
<b>BAB VI PENUTUP .....</b>	20
6.1    Kesimpulan .....	20
6.2    Saran .....	20

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Pengelolaan peminjaman buku secara manual sering menimbulkan berbagai masalah, seperti data yang tidak rapi, kesalahan pencatatan, dan kesulitan dalam membuat laporan peminjaman. Proses pencatatan yang masih dilakukan secara tulis tangan atau menggunakan file sederhana juga membuat data mudah hilang dan sulit ditelusuri kembali.

Untuk mengatasi permasalahan tersebut, dibuatlah Aplikasi Peminjaman Buku Digital yang bertujuan membantu proses peminjaman dan pengembalian buku agar lebih terstruktur dan efisien. Aplikasi ini dikembangkan menggunakan bahasa pemrograman Python dengan konsep *Object-Oriented Programming* (OOP) dan dilengkapi dengan antarmuka grafis (GUI) berbasis Tkinter agar mudah digunakan. Selain itu, sistem ini menyediakan fitur ekspor data peminjaman ke dalam file CSV yang dapat dimanfaatkan sebagai bahan laporan dan dokumentasi, sehingga pengelolaan data buku menjadi lebih rapi dan terorganisir.

## 1.2 Tujuan

1. Menerapkan konsep *Object-Oriented Programming* (OOP) dalam pembuatan aplikasi menggunakan bahasa pemrograman Python.
2. Membuat proses peminjaman dan pengembalian buku agar lebih terstruktur, rapi, dan efisien.
3. Memudahkan pengguna dalam mengelola data buku dan data peminjaman melalui tampilan antarmuka grafis (GUI) berbasis Tkinter.

## 1.3 Ruang Lingkup

1. Aplikasi hanya menyediakan fitur peminjaman dan pengembalian buku pada Aplikasi peminjaman buku digital.
2. Data buku yang dikelola terdiri dari buku fiksi dan non-fiksi.
3. Aplikasi dikembangkan menggunakan bahasa pemrograman Python dengan konsep OOP.
4. Antarmuka aplikasi dibuat menggunakan library Tkinter.
5. Penyimpanan laporan peminjaman dilakukan dalam bentuk file CSV, tanpa menggunakan database.
6. Aplikasi digunakan oleh satu pengguna dalam satu sesi (belum mendukung multi-user).

## BAB II KONSEP OBJECT-ORIENTED PROGRAMMING

Aplikasi Peminjaman Buku Digital ini dibangun dengan menggunakan konsep *Object-Oriented Programming* (OOP) agar struktur program lebih rapi dan mudah dipahami. Dengan pendekatan OOP, setiap bagian penting dalam aplikasi, seperti pengelolaan buku dan anggota, dibuat dalam bentuk *class* yang memiliki fungsi masing-masing. Cara ini memudahkan pengembang dalam mengatur alur program, mengurangi penulisan kode yang berulang, serta mempermudah pengembangan sistem jika ke depannya ingin menambahkan fitur baru. Pada bab ini akan dibahas tiga konsep utama OOP yang digunakan dalam aplikasi, yaitu Encapsulation, Inheritance, dan Polymorphism beserta contoh penerapannya dalam sistem.

### 2.1 Encapsulation

Encapsulation diterapkan dengan cara membatasi akses langsung ke data penting yang ada di dalam sebuah kelas. Artinya, data tidak bisa sembarangan diubah dari luar kelas, tetapi harus melalui metode atau fungsi yang sudah disediakan.

Pada aplikasi ini, Encapsulation diterapkan pada *class Buku*. Atribut seperti judul, penulis, tahun terbit, dan status buku dibuat sebagai atribut private dengan tanda \_\_. Dengan begitu, data buku tidak bisa langsung diubah dari luar class, melainkan harus melalui method yang sudah disediakan. Selain itu, Encapsulation juga diterapkan pada *class Anggota*. Daftar buku yang sedang dipinjam disimpan dalam atribut private. Untuk menambah atau mengurangi daftar buku tersebut, sistem hanya memperbolehkan perubahan melalui method pinjam\_buku() dan kembalikan\_buku(). Cara ini membuat proses peminjaman dan pengembalian buku menjadi lebih terkontrol dan sesuai dengan aturan aplikasi.

### 2.2 Inheritance

Inheritance digunakan untuk menghindari penulisan kode yang berulang. Pada aplikasi ini, *class Buku* berperan sebagai *class induk*, sedangkan *BukuFiksi* dan *BukuNonFiksi* sebagai *class turunan*.

*Class BukuFiksi* dan *BukuNonFiksi* mewarisi atribut dan method dari *class Buku*, seperti judul, penulis, tahun terbit, dan status peminjaman. Setiap *class turunan* juga memiliki atribut tambahan, yaitu genre untuk buku fiksi dan bidang untuk buku non-fiksi. Dengan inheritance, kode menjadi lebih ringkas dan mudah dikembangkan jika ingin menambahkan jenis buku baru.

### 2.1 Polymorphism

Polymorphism diterapkan dengan cara method overriding, yaitu ketika *class turunan* mengubah cara kerja method dari *class induk*.

Pada aplikasi ini, method getInfo() yang ada di *class Buku* dioverride di *class BukuFiksi* dan *BukuNonFiksi*. Walaupun nama method-nya sama, informasi yang ditampilkan berbeda sesuai dengan jenis bukunya. Satu method bisa digunakan untuk berbagai jenis objek, sehingga kode menjadi lebih rapi, fleksibel, dan mudah dikelola.

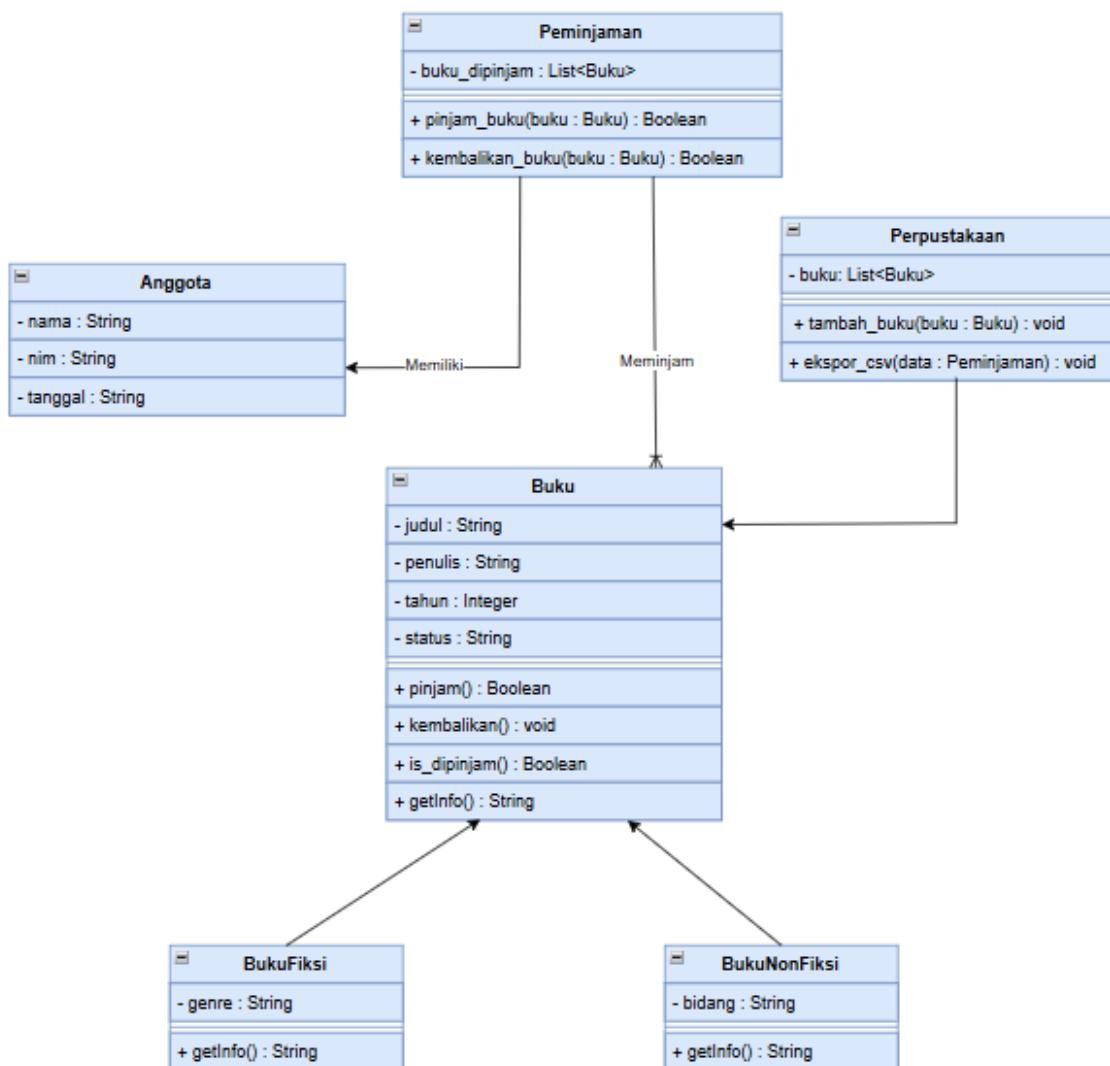
## BAB III PERANCANGAN SISTEM

### 3.1 Deskripsi Umum

Aplikasi Peminjaman Buku Digital adalah aplikasi desktop sederhana yang dibuat menggunakan bahasa pemrograman Python dengan konsep Object-Oriented Programming (OOP). Aplikasi ini bertujuan untuk membantu proses peminjaman dan pengembalian buku agar lebih rapi dan mudah dikelola. Dengan antarmuka grafis (GUI) berbasis Tkinter, pengguna dapat mengoperasikan aplikasi secara visual tanpa harus berinteraksi langsung dengan kode. Aplikasi ini menyediakan fitur peminjaman buku, pengembalian buku, filter kategori buku, serta ekspor data peminjaman ke dalam file CSV untuk keperluan laporan.

### 3.2 Class Diagram

Aplikasi Peminjaman Buku Digital menggunakan konsep Object-Oriented Programming (OOP), sehingga setiap bagian utama seperti buku dan anggota dibagi ke dalam class masing-masing. Berikut gambar class diagramnya.



### 3.3 Penjelasan Class Diagram

Diagram kelas ini menggambarkan hubungan antara beberapa entitas dalam sistem perpustakaan sederhana. Berikut penjelasan untuk tiap kelas dan relasi yang ada:

1. Class Perpustakaan ini adalah kelas entitas utama yang memegang daftar buku. Perpustakaan memiliki metode untuk menambah buku ke dalam sistem dan juga untuk mengekspor data peminjaman buku ke format CSV.

Atribut:

- a. buku: List<Buku>: Daftar buku yang ada di perpustakaan.

Metode:

- a. tambah\_buku(buku: Buku): void: Menambahkan buku baru ke dalam perpustakaan.
- b. eksport\_csv(data: Peminjaman): void: Mengekspor data peminjaman buku dalam format CSV.

2. Class Buku adalah kelas yang memiliki informasi seperti judul, penulis, tahun terbit, dan status (apakah sedang dipinjam atau tersedia). Buku ini juga memiliki metode untuk dipinjam dan dikembalikan. Ada juga atribut is\_dipinjam yang menandakan apakah buku sedang dipinjam atau tidak.

Atribut:

- a. judul: String: Judul buku.
- b. penulis: String: Penulis buku.
- c. tahun: Integer: Tahun terbit buku.
- d. status: String: Status buku (apakah dipinjam atau tersedia).

Metode:

- a. pinjam(): Boolean: Menandakan bahwa buku sedang dipinjam.
- b. kembalikan(): void: Mengembalikan buku yang dipinjam.
- c. is\_dipinjam(): Boolean: Menandakan status buku, apakah sedang dipinjam.
- d. getInfo(): String: Mendapatkan informasi detail tentang buku.

3. Class Anggota adalah kelas pengguna yang bisa meminjam buku dari perpustakaan. Setiap anggota memiliki nama, nomor induk mahasiswa (NIM), dan tanggal bergabung dengan perpustakaan. Anggota ini dapat memiliki banyak buku yang dipinjam.

Atribut:

- a. nama: String: Nama anggota.
- b. nim: String: Nomor Induk Mahasiswa anggota.
- c. tanggal: String: Tanggal bergabung dengan perpustakaan.

4. Class Peminjaman ini mengelola peminjaman buku oleh anggota. Di dalamnya, terdapat metode untuk meminjam dan mengembalikan buku. Setiap peminjaman mencatat buku yang dipinjam oleh anggota.

Atribut:

- a. buku\_dipinjam: List<Buku>: Daftar buku yang sedang dipinjam oleh anggota.

Metode:

- a. pinjam\_buku(buku: Buku): Boolean: Memproses peminjaman buku oleh anggota.
  - b. kembalikan\_buku(buku: Buku): Boolean: Memproses pengembalian buku oleh anggota.
5. Class BukuFiksi dan BukuNonFiksi merupakan subclass/ kelas turunan dari kelas Buku yang digunakan untuk membedakan jenis buku. BukuFiksi memiliki atribut tambahan berupa genre, sementara BukuNonFiksi memiliki atribut bidang yang menjelaskan kategori buku non-fiksi tersebut.

Buku Fiksi

Atribut:

- a. genre: String: Genre buku fiksi, seperti fiksi ilmiah, fantasi, dll.

Metode:

- a. getInfo(): String: Mengambil informasi tentang buku fiksi.

BukuNonFiksi: Kelas ini juga subclass dari Buku, tetapi digunakan untuk buku non-fiksi.

Atribut:

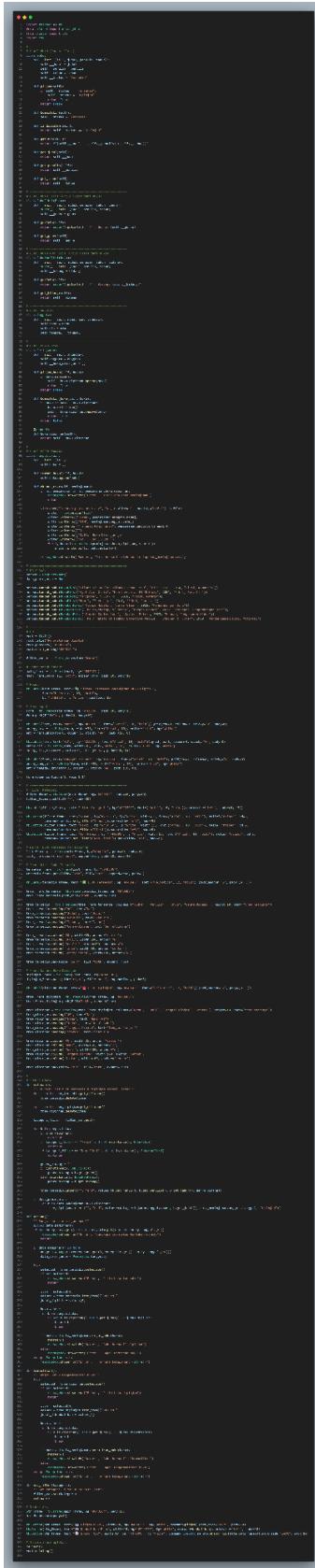
- a. bidang: String: Bidang dari buku non-fiksi, seperti sejarah, teknologi, dll.

Metode:

- a. getInfo(): String: Mengambil informasi tentang buku non-fiksi.

## BAB IV IMPLEMENTASI KODE

### 4.1 Kode program



The image shows a terminal window with a black background and white text, displaying a large block of Java code. The code is organized into several classes and methods, with various annotations and imports visible. The text is too small to be read individually but represents a substantial portion of a Java application's source code.

## 4.2 Penjelasan kode (inline atau poin-poin)



```
1 import tkinter as tk
2 from tkinter import messagebox
3 from tkinter import ttk
4 import csv
```

- Tkinter: library utama untuk membuat GUI.
- Messagebox: menampilkan pesan dialog (error & info).
- Ttk: widget modern seperti Treeview.
- Csv: mengekspor data peminjaman ke file CSV.

```
7 # CLASS BUKU (Parent Class)
8 class Buku:
9     def __init__(self, judul, penulis, tahun):
10         self.__judul = judul
11         self.__penulis = penulis
12         self.__tahun = tahun
13         self.__status = "Tersedia"
14
15     def pinjam(self):
16         if self.__status == "Tersedia":
17             self.__status = "Dipinjam"
18             return True
19         return False
20
21     def kembalikan(self):
22         self.__status = "Tersedia"
23
24     def is_dipinjam(self):
25         return self.__status == "Dipinjam"
26
27     def getInfo(self):
28         return f"{self.__judul} - {self.__penulis} ({self.__tahun})"
29
30     def get_judul(self):
31         return self.__judul
32
33     def get_penulis(self):
34         return self.__penulis
35
36     def get_tahun(self):
37         return self.__tahun
```

- Class ini merupakan class induk yang merepresentasikan objek buku.
- Atribut di class ini menggunakan akses private (\_\_) sebagai penerapan encapsulation.
- Memiliki method utama:
  - a. pinjam(): Method ini digunakan untuk memproses peminjaman buku.
  - b. kembalikan(): Method ini digunakan untuk mengembalikan buku. Status buku diubah kembali menjadi "Tersedia" sehingga dapat dipinjam ulang.
  - c. is\_dipinjam(): mengecek status buku.
  - d. getInfo(): menampilkan informasi buku.
  - e. Getter (get\_judul(), get\_penulis(), get\_tahun()): Digunakan untuk mengakses atribut private secara aman tanpa melanggar encapsulation.

```

48 # CLASS BUKU FIksi (Child Class dari Buku)
49 class BukuFiksi(Buku):
50     def __init__(self, judul, penulis, tahun, genre):
51         super().__init__(judul, penulis, tahun)
52         self.__genre = genre
53
54     def getInfo(self):
55         return super().getInfo() + f" | Genre: {self.__genre}"
56
57     def get_genre(self):
58         return self.__genre

```

- Class ini merupakan turunan dari class Buku dan digunakan untuk buku kategori fiksi.
- Method getInfo(): Method ini mengoverride method getInfo() pada class induk untuk menambahkan informasi genre.

```

53 # CLASS BUKU NON FIksi (Child Class dari Buku)
54 class BukuNonFiksi(Buku):
55     def __init__(self, judul, penulis, tahun, bidang):
56         super().__init__(judul, penulis, tahun)
57         self.__bidang = bidang
58
59     def getInfo(self):
60         return super().getInfo() + f" | Bidang: {self.__bidang}"
61
62     def get_bidang(self):
63         return self.__bidang
64

```

- Class ini merupakan turunan dari class Buku dan digunakan untuk buku kategori non fiksi.
- Tambahan Atribut \_\_bidang Override getInfo() Menambahkan informasi bidang buku non-fiksi.

```

66 # CLASS ANGGOTA
67 class Anggota:
68     def __init__(self, nama, nim, tanggal):
69         self.nama = nama
70         self.nim = nim
71         self.tanggal = tanggal
72

```

- Class ini merepresentasikan identitas anggota perpustakaan.
- Class ini menyimpan data peminjam: Nama, NIM dan Tanggal peminjaman.

```

74 # CLASS PEMINJAMAN
75 class Peminjaman:
76     def __init__(self, anggota):
77         self.anggota = anggota
78         self.__buku_dipinjam = []
79
80     def pinjam_buku(self, buku):
81         if buku.pinjam():
82             self.__buku_dipinjam.append(buku)
83             return True
84         return False
85
86     def kembalikan_buku(self, buku):
87         if buku in self.__buku_dipinjam:
88             buku.kembalikan()
89             self.__buku_dipinjam.remove(buku)
90             return True
91         return False
92
93     @property
94     def buku_dipinjam(self):
95         return self.__buku_dipinjam

```

- Class ini mengelola proses peminjaman buku oleh anggota.
- Atribut:
  - a. anggota: objek Anggota
  - b. \_\_buku\_dipinjam: daftar buku yang sedang dipinjam
- Method pinjam\_buku: Memanggil method pinjam() pada objek buku. Jika berhasil, buku ditambahkan ke daftar peminjaman.
- Method kembalikan\_buku: Mengembalikan buku dan menghapusnya dari daftar peminjaman.
- Property buku\_dipinjam: Digunakan untuk mengakses daftar buku dipinjam secara aman.

```

98 # CLASS PERPUSTAKAAN
99 class Perpustakaan:
100     def __init__(self):
101         self.buku = []
102
103     def tambah_buku(self, buku):
104         self.buku.append(buku)
105
106     def ekspor_csv(self, peminjaman):
107         if not peminjaman or not peminjaman.buku_dipinjam:
108             messagebox.showerror("Error", "Tidak ada data peminjaman")
109             return
110
111         with open("laporan_peminjaman.csv", "w", newline="", encoding="utf-8") as file:
112             writer = csv.writer(file)
113             writer.writerow(["Nama", peminjaman.anggota.nama])
114             writer.writerow(["NIM", peminjaman.anggota.nim])
115             writer.writerow(["Tanggal Peminjaman", peminjaman.anggota.tanggal])
116             writer.writerow([])
117             writer.writerow(["Daftar Buku Dipinjam"])
118             writer.writerow(["No", "Judul Buku"])
119             for i, buku in enumerate(peminjaman.buku_dipinjam, start=1):
120                 writer.writerow([i, buku.getInfo()])
121
122         messagebox.showinfo("Sukses", "Data berhasil diekspor ke laporan_peminjaman.csv")
123

```

- Class ini berfungsi sebagai pengelola seluruh buku.
- Method tambah\_buku: Menambahkan buku ke dalam koleksi perpustakaan.
- Method ekspor\_csv: Mengekspor data peminjaman ke file laporan\_peminjaman.csv. Berisi data anggota dan daftar buku dipinjam.

```

125 # DATA AWAL
126 perpus = Perpustakaan()
127 data_peminjaman = None
128
129 perpus.tambah_buku(BukuFiksi("Dilan: Dia adalah Dilanku Tahun 1990", "Pidi Baiq", 2014, "Fiksi, Romantis"))
130 perpus.tambah_buku(BukuFiksi("Ayat-Ayat Cinta", "Habiburrahman El Shirazy", 2004, "Fiksi, Romantis"))
131 perpus.tambah_buku(BukuFiksi("Mariposa", "Luluk HF", 2018, "Fiksi, Remaja"))
132 perpus.tambah_buku(BukuFiksi("Bumi", "Tere Liye", 2014, "Fiksi, Fantasi"))
133 perpus.tambah_buku(BukuNonFiksi("Atomic Habits", "James Clear", 2018, "Pengembangan Diri"))
134 perpus.tambah_buku(BukuNonFiksi("The Psychology of Money", "Morgan Housel", 2020, "Keuangan, Pengembangan Diri"))
135 perpus.tambah_buku(BukuNonFiksi("Kimchi Confessions", "Xaviera Putri", 2023, "Memoar, Pengalaman Hidup"))
136 perpus.tambah_buku(BukuNonFiksi("The 7 Habits of Highly Effective People", "Stephen R. Covey", 1989, "Pengembangan Diri, Motivasi"))

```

- perpus merupakan objek dari class Perpustakaan yang berfungsi sebagai pengelola seluruh koleksi buku.
- data\_peminjaman diinisialisasi dengan nilai None karena pada awal aplikasi belum ada anggota yang melakukan peminjaman.
- Variabel ini nantinya akan menyimpan objek dari class Peminjaman ketika pengguna mulai meminjam buku.
- Perpus.tambah\_buku: adalah bagian yang digunakan untuk menambahkan data buku awal ke dalam sistem. Buku dibedakan menjadi dua jenis yaitu buku fiksi dan buku non fiksi.

```

137 # GUI
138 root = tk.Tk()
139 root.title("Perpustakaan Digital")
140 root.geometry("1000x700")
141 root.configure(bg="#84B1C4")
142

```

- Root: merupakan window utama aplikasi.
- title(): menetapkan judul aplikasi.
- geometry(): mengatur ukuran window.
- configure(): mengatur warna latar belakang aplikasi.

```

143 filter_var = tk.StringVar(value="Semua")
144

```

- Digunakan untuk menyimpan kategori filter buku.
- Nilai awal "Semua" menampilkan seluruh buku. Nilai ini akan berubah ketika tombol filter ditekan.

```

145 # Frame untuk layout
146 main_frame = tk.Frame(root, bg="#84B1C4")
147 main_frame.pack(fill="both", expand=True, padx=10, pady=10)
148

```

- main\_frame berfungsi sebagai wadah utama semua komponen GUI.
- Menggunakan pack() agar frame menyesuaikan ukuran window.

```

149 # Header
150 tk.Label(main_frame, text="Sistem Informasi Peminjaman Buku Digital",
151         font=("Helvetica", 18, "bold"),
152         bg="#84B1C4", fg="white").pack(pady=10)
153

```

- Label ini berfungsi sebagai judul aplikasi yang tampil di bagian atas.
- Menggunakan font besar dan bold agar mudah dibaca dan mudah di ingat oleh user.

```

154 # Form input
155 form = tk.Frame(main_frame, bg="#CCD7DC", padx=20, pady=15)
156 form.pack(fill="x", padx=20, pady=10)
157
158 tk.Label(form, text="Nama:", bg="#CCD7DC", font=("Arial", 10, "bold")).grid(row=0, column=0, sticky="w", pady=8)
159 entry_nama = tk.Entry(form, width=35, font=("Arial", 10), relief="flat", bg="white")
160 entry_nama.grid(row=0, column=1, sticky="ew", padx=(10, 0))
161
162 tk.Label(form, text="NIM:", bg="#CCD7DC", font=("Arial", 10, "bold")).grid(row=1, column=0, sticky="w", pady=8)
163 entry_nim = tk.Entry(form, width=35, font=("Arial", 10), relief="flat", bg="white")
164 entry_nim.grid(row=1, column=1, sticky="ew", padx=(10, 0))
165
166 tk.Label(form, text="Tanggal Pinjam:", bg="#CCD7DC", font=("Arial", 10, "bold")).grid(row=2, column=0, sticky="w", pady=8)
167 entry_tanggal = tk.Entry(form, width=35, font=("Arial", 10), relief="flat", bg="white")
168 entry_tanggal.grid(row=2, column=1, sticky="ew", padx=(10, 0))
169
170 form.columnconfigure(1, weight=1)

```

- Digunakan untuk memasukkan Nama anggota, NIM dan Tanggal peminjaman.
- Data ini akan digunakan untuk membuat objek Anggota saat peminjaman pertama kali dilakukan.
- Menggunakan grid() agar tata letak rapi.
- row dan column berfungsi untuk menentukan posisi widget dalam grid, di mana label diletakkan pada kolom pertama dan field input (Entry) pada kolom kedua dengan baris yang sama sehingga sejajar.
- sticky digunakan untuk mengatur posisi widget di dalam sel, seperti sticky="w" dan sticky="ew" agar field input dapat melebar ke kiri dan kanan.
- parameter padx dan pady digunakan untuk memberi jarak horizontal dan vertikal antar komponen sehingga tampilan tidak terlalu rapat.
- Penggunaan form.columnconfigure(1, weight=1) bertujuan agar kolom input bersifat fleksibel dan menyesuaikan ukuran jendela ketika diperbesar.

```

173 # Filter Kategori
174 filter_frame = tk.Frame(main_frame, bg="#CCD7DC", padx=20, pady=10)
175 filter_frame.pack(fill="x", padx=20)
176
177 tk.Label(filter_frame, text="Filter Kategori:", bg="#CCD7DC", font=("Arial", 10, "bold")).pack(side="left", padx=(0, 15))
178
179 tk.Button(filter_frame, text="Semua", bg="#2196F3", fg="white", width=15, font=("Arial", 10, "bold"), relief="raised", bd=2,
180     command=lambda: set_filter("Semua")).pack(side="left", padx=5)
181 tk.Button(filter_frame, text="Fiksi", bg="#4CAF50", fg="white", width=15, font=("Arial", 10, "bold"), relief="raised", bd=2,
182     command=lambda: set_filter("Fiksi")).pack(side="left", padx=5)
183 tk.Button(filter_frame, text="Non-Fiksi", bg="#FF9800", fg="white", width=15, font=("Arial", 10, "bold"), relief="raised", bd=2,
184     command=lambda: set_filter("Non-Fiksi")).pack(side="left", padx=5)
185

```

- Masing-masing tombol dihubungkan dengan fungsi set\_filter() melalui parameter command.
- parameter command yang menggunakan ekspresi lambda, sehingga ketika tombol ditekan, nilai kategori filter akan diperbarui sesuai pilihan pengguna.
- Nilai kategori tersebut disimpan dalam variabel filter\_var bertipe StringVar, yang berfungsi sebagai penampung status filter saat ini.
- Setelah kategori diubah, fungsi refresh() dipanggil untuk memperbarui tampilan daftar buku agar hanya menampilkan buku yang sesuai dengan kategori yang dipilih.

```

186 # Daftar Buku Tersedia dan Dipinjam
187 list_frame = tk.Frame(main_frame, bg="#D4D5D6", padx=10, pady=10)
188 list_frame.pack(fill="both", expand=True, padx=20, pady=10)
189

```

- Daftar Buku Tersedia dan Buku Dipinjam, menggunakan komponen Treeview dari ttk untuk menampilkan data buku dalam bentuk tabel yang terstruktur.

- Tampilan ini dibagi menjadi dua bagian menggunakan Frame, yaitu sisi kiri untuk buku yang masih tersedia dan sisi kanan untuk buku yang sedang dipinjam.

```

190 # Frame kiri: Buku Tersedia
191 tersedia_frame = tk.Frame(list_frame, bg="#D4D5D6")
192 tersedia_frame.pack(side="left", fill="both", expand=True, padx=5)
193
194 tk.Label(tersedia_frame, text="■ Buku Tersedia", bg="#D4D5D6", font=("Helvetica", 12, "bold")).pack(anchor="w", pady=(0, 5))
195
196 tree_frame_tersedia = tk.Frame(tersedia_frame, bg="#D4D5D6")
197 tree_frame_tersedia.pack(fill="both", expand=True)
198
199 tree_tersedia = ttk.Treeview(tree_frame_tersedia, columns=("Judul", "Penulis", "Tahun", "Genre/Bidang"), height=10, show="tree headings")
200 tree_tersedia.heading("#0", text="No")
201 tree_tersedia.heading("Judul", text="Judul")
202 tree_tersedia.heading("Penulis", text="Penulis")
203 tree_tersedia.heading("Tahun", text="Tahun")
204 tree_tersedia.heading("Genre/Bidang", text="Genre/Bidang")
205
206 tree_tersedia.column("#0", width=35, anchor="center")
207 tree_tersedia.column("Judul", width=200, anchor="w")
208 tree_tersedia.column("Penulis", width=150, anchor="w")
209 tree_tersedia.column("Tahun", width=60, anchor="center")
210 tree_tersedia.column("Genre/Bidang", width=150, anchor="w")
211
212 tree_tersedia.pack(side="left", fill="both", expand=True)
213

```

- tersedia\_frame yang diletakkan di sisi kiri dari list\_frame menggunakan metode pack(side="left")
- Properti fill="both" dan expand=True digunakan agar frame menyesuaikan ukuran jendela, sementara padx=5 memberi jarak antar frame.
- Di dalam frame ini ditambahkan Label sebagai judul “Buku Tersedia” untuk memberi penanda visual kepada pengguna.
- tree\_frame\_tersedia sebagai wadah khusus untuk komponen Treeview.
- Komponen Treeview (tree\_tersedia) digunakan untuk menampilkan data buku tersedia dalam bentuk tabel dengan kolom Judul, Penulis, Tahun, dan Genre/Bidang.

```

214 # Frame kanan: Buku Dipinjam
215 dipinjam_frame = tk.Frame(list_frame, bg="#D4D5D6")
216 dipinjam_frame.pack(side="right", fill="both", expand=True, padx=5)
217
218 tk.Label(dipinjam_frame, text="■ Buku Dipinjam", bg="#D4D5D6", font=("Helvetica", 12, "bold")).pack(anchor="w", pady=(0, 5))
219
220 tree_frame_dipinjam = tk.Frame(dipinjam_frame, bg="#D4D5D6")
221 tree_frame_dipinjam.pack(fill="both", expand=True)
222
223 tree_dipinjam = ttk.Treeview(tree_frame_dipinjam, columns=("Nama", "Judul", "Tanggal Pinjam", "Status"), height=10, show="tree headings")
224 tree_dipinjam.heading("#0", text="No")
225 tree_dipinjam.heading("Nama", text="Nama (NIM)")
226 tree_dipinjam.heading("Judul", text="Judul Buku")
227 tree_dipinjam.heading("Tanggal Pinjam", text="Tanggal Pinjam")
228 tree_dipinjam.heading("Status", text="Status")
229
230 tree_dipinjam.column("#0", width=35, anchor="center")
231 tree_dipinjam.column("Nama", width=150, anchor="w")
232 tree_dipinjam.column("Judul", width=200, anchor="w")
233 tree_dipinjam.column("Tanggal Pinjam", width=120, anchor="center")
234 tree_dipinjam.column("Status", width=100, anchor="center")
235
236 tree_dipinjam.pack(side="left", fill="both", expand=True)
237

```

- frame Buku Dipinjam, yang dibuat dengan konsep yang sama namun diletakkan di sisi kanan menggunakan pack(side="right").
- Frame ini juga dilengkapi label judul “Buku Dipinjam” sebagai penanda.
- Treeview (tree\_dipinjam) yang menampilkan data buku yang sedang dipinjam.
- Kolom pada tree\_dipinjam disesuaikan dengan kebutuhan informasi peminjaman, yaitu Nama (NIM), Judul Buku, Tanggal Pinjam, dan Status.

Secara keseluruhan, penggunaan dua Treeview ini memungkinkan aplikasi membedakan secara jelas antara buku yang masih tersedia dan buku yang sedang dipinjam.

```

241 # FUNGSI UTAMA
242 def refresh():
243     """Refresh list buku tersedia & dipinjam sesuai filter"""
244     for item in tree_tersedia.get_children():
245         tree_tersedia.delete(item)
246
247     for item in tree_dipinjam.get_children():
248         tree_dipinjam.delete(item)
249
250     kategori_filter = filter_var.get()
251
252     for b in perpus.buku:
253         if b.is_dipinjam():
254             continue
255         if kategori_filter == "Fiksi" and not isinstance(b, BukuFiksi):
256             continue
257         if kategori_filter == "Non-Fiksi" and not isinstance(b, BukuNonFiksi):
258             continue
259
260         genre_bidang = ""
261         if isinstance(b, BukuFiksi):
262             genre_bidang = b.get_genre()
263         elif isinstance(b, BukuNonFiksi):
264             genre_bidang = b.get_bidang()
265
266         tree_tersedia.insert("", "end", values=(b.get_judul(), b.get_penulis(), b.get_tahun(), genre_bidang))
267
268     if data_peminjaman:
269         for b in data_peminjaman.buku_dipinjam:
270             tree_dipinjam.insert("", "end", values=(data_peminjaman.anggota.nama, b.get_judul(), data_peminjaman.anggota.tanggal, "Dipinjam"))
271

```

- Fungsi refresh() digunakan untuk memperbarui tampilan daftar buku tersedia dan buku dipinjam sesuai dengan kondisi data terbaru dan kategori filter yang dipilih.
- Menghapus seluruh data lama yang ada pada Treeview buku tersedia dan buku dipinjam agar tidak terjadi duplikasi data.
- filter\_var untuk menentukan apakah yang ditampilkan adalah semua buku, buku fiksi, atau buku non-fiksi.
- Fungsi kemudian melakukan iterasi terhadap seluruh objek buku yang tersimpan dalam perpus.buku.
- Pengecekan tipe objek dilakukan menggunakan isinstance() untuk memastikan buku yang ditampilkan sesuai dengan kategori filter.
- Jika terdapat data peminjaman, fungsi ini juga menampilkan daftar buku yang sedang dipinjam beserta nama peminjam dan tanggal peminjaman pada tabel buku dipinjam.

```

272 def pinjam():
273     """Fungsi untuk meminjam buku"""
274     global data_peminjaman
275     if not entry_nama.get() or not entry_nim.get() or not entry_tanggal.get():
276         messagebox.showerror("Error", "Lengkapi identitas terlebih dahulu")
277         return
278
279     if data_peminjaman is None:
280         anggota = Anggota(entry_nama.get(), entry_nim.get(), entry_tanggal.get())
281         data_peminjaman = Peminjaman(anggota)
282
283     try:
284         selected = tree_tersedia.selection()
285         if not selected:
286             messagebox.showerror("Error", "Pilih buku tersedia")
287             return
288
289         item = selected[0]
290         values = tree_tersedia.item(item)['values']
291         judul_dipilih = values[0]
292
293         buku = None
294         for b in perpus.buku:
295             if not b.is_dipinjam() and b.get_judul() == judul_dipilih:
296                 buku = b
297                 break
298
299         if buku and data_peminjaman.pinjam_buku(buku):
300             refresh()
301             messagebox.showinfo("Sukses", "Buku berhasil dipinjam")
302         else:
303             messagebox.showerror("Error", "Gagal meminjam buku")
304     except Exception as e:
305         messagebox.showerror("Error", f"Terjadi kesalahan: {str(e)}")
306

```

- Fungsi pinjam() bertugas untuk mengelola proses peminjaman buku. Fungsi ini diawali dengan validasi input identitas peminjam, yaitu nama, NIM, dan tanggal pinjam.
- Jika salah satu data belum diisi, sistem akan menampilkan pesan kesalahan.
- Apabila belum ada data peminjaman sebelumnya, fungsi akan membuat objek Anggota dan Peminjaman.
- Jika proses peminjaman berhasil dilakukan melalui method pinjam\_buku(), maka tampilan akan diperbarui dan pesan keberhasilan ditampilkan kepada pengguna.
- Jika gagal, sistem akan memberikan notifikasi kesalahan.

```

307 def kembalikan():
308     """Fungsi untuk mengembalikan buku"""
309     try:
310         selected = tree_dipinjam.selection()
311         if not selected:
312             messagebox.showerror("Error", "Pilih buku dipinjam")
313             return
314
315         item = selected[0]
316         values = tree_dipinjam.item(item)['values']
317         judul_dikembalikan = values[1]
318
319         buku = None
320         for b in perpus.buku:
321             if b.is_dipinjam() and b.get_judul() == judul_dikembalikan:
322                 buku = b
323                 break
324
325         if buku and data_peminjaman.kembalikan_buku(buku):
326             refresh()
327             messagebox.showinfo("Sukses", "Buku berhasil dikembalikan")
328         else:
329             messagebox.showerror("Error", "Gagal mengembalikan buku")
330     except Exception as e:
331         messagebox.showerror("Error", f"Terjadi kesalahan: {str(e)}")
332

```

- Fungsi kembalikan() digunakan untuk mengelola proses pengembalian buku. Fungsi ini memastikan bahwa pengguna telah memilih buku dari daftar buku dipinjam.
- Jika buku ditemukan dan proses pengembalian berhasil melalui method kembalikan\_buku(), maka status buku akan dikembalikan menjadi tersedia.

```

333 def set_filter(kategori):
334     """Set kategori filter & refresh list"""
335     filter_var.set(kategori)
336     refresh()
337

```

- Fungsi set\_filter() berfungsi untuk mengatur kategori filter buku yang akan ditampilkan pada daftar buku tersedia.
- Fungsi ini menerima parameter kategori, kemudian menyimpannya ke dalam variabel filter\_var.
- Setelah itu, fungsi refresh() dipanggil untuk memperbarui tampilan sesuai dengan filter yang dipilih.

```

338 # Tombol Aksi
339 btn_frame = tk.Frame(main_frame, bg="#84B1C4", pady=10)
340 btn_frame.pack(pady=10)
341
342 tk.Button(btn_frame, text="Pinjam Buku", width=18, bg="#4CAF50", fg="white", command=pinjam).pack(side="left", padx=10)
343 tk.Button(btn_frame, text="Kembalikan Buku", width=18, bg="#F44336", fg="white", command=kembalikan).pack(side="left", padx=10)
344 tk.Button(btn_frame, text="Eksport CSV", width=18, bg="#FF9800", fg="white", command=lambda: perpus.ekspor_csv(data_peminjaman)).pack(side="left", padx=10)

```

- btn\_frame: Kode ini membuat frame khusus untuk menampung tombol aksi pada aplikasi.
- btn\_frame ditempatkan di dalam main\_frame dengan warna latar #84B1C4.
- Properti pady=10 pada Frame dan pack() memberi jarak vertikal agar tombol tidak menempel dengan komponen lain, sehingga tampilan lebih rapi dan nyaman dilihat
- Pinjam Buku: memanggil fungsi pinjam()
- Kembalikan Buku: memanggil fungsi kembalikan()
- Eksport CSV: menyimpan laporan peminjaman ke file CSV

```

346 # Inisialisasi aplikasi
347 refresh()
348 root.mainloop()

```

- refresh(): memastikan data tampil sejak awal aplikasi dijalankan.
- mainloop(): menjalankan aplikasi GUI hingga ditutup pengguna.

### 4.3 Kerja Aplikasi/Sistem

Saat aplikasi dijalankan, sistem akan menginisialisasi objek Perpustakaan yang berisi data buku, serta menampilkan tampilan utama aplikasi berupa form input peminjam, daftar buku tersedia, dan daftar buku dipinjam.

Pengguna terlebih dahulu mengisi data peminjam (nama, NIM, dan tanggal peminjaman). Data ini kemudian disimpan dalam objek Anggota dan dikelola oleh objek Peminjaman. Ketika pengguna memilih buku dan menekan tombol Pinjam Buku, sistem akan memanggil method pinjam\_buku() untuk mengubah status buku menjadi Dipinjam dan memindahkannya ke daftar buku dipinjam.

Sistem secara otomatis memperbarui tampilan melalui fungsi refresh() agar daftar buku tersedia dan dipinjam selalu sesuai dengan kondisi terkini. Pada saat proses pengembalian, pengguna memilih buku yang dipinjam dan menekan tombol Kembalikan Buku, lalu sistem akan memanggil method kembalikan\_buku() untuk mengubah status buku menjadi Tersedia.

Selain itu, aplikasi menyediakan fitur ekspor data peminjaman ke file CSV melalui method ekspor\_csv(), sehingga data peminjaman dapat digunakan sebagai laporan. Seluruh proses berjalan secara terintegrasi antara logika program dan tampilan GUI, sehingga aplikasi dapat digunakan dengan mudah dan efisien.

## BAB V IMPLEMENTASI DAN PENGUJIAN APLIKASI

### 5.1 Fitur

Aplikasi Peminjaman Buku Digital memiliki beberapa fitur utama yang mendukung proses peminjaman dan pengembalian buku. Meliputi:

1. Input Data Peminjam: Pengguna dapat mengisi nama, NIM, dan tanggal peminjaman.
2. Daftar Buku Tersedia: Menampilkan daftar buku yang masih tersedia untuk dipinjam.
3. Daftar Buku Dipinjam: Menampilkan buku yang sedang dipinjam beserta informasi peminjam.
4. Peminjaman Buku: Pengguna dapat meminjam buku yang dipilih dengan satu klik.
5. Pengembalian Buku: Buku yang telah dipinjam dapat dikembalikan melalui sistem.
6. Filter Kategori Buku: Menyaring buku berdasarkan kategori Fiksi dan Non-Fiksi.
7. Ekspor Data ke CSV: Menyimpan data peminjaman ke file CSV sebagai laporan.

### 5.2 Cara Penggunaan

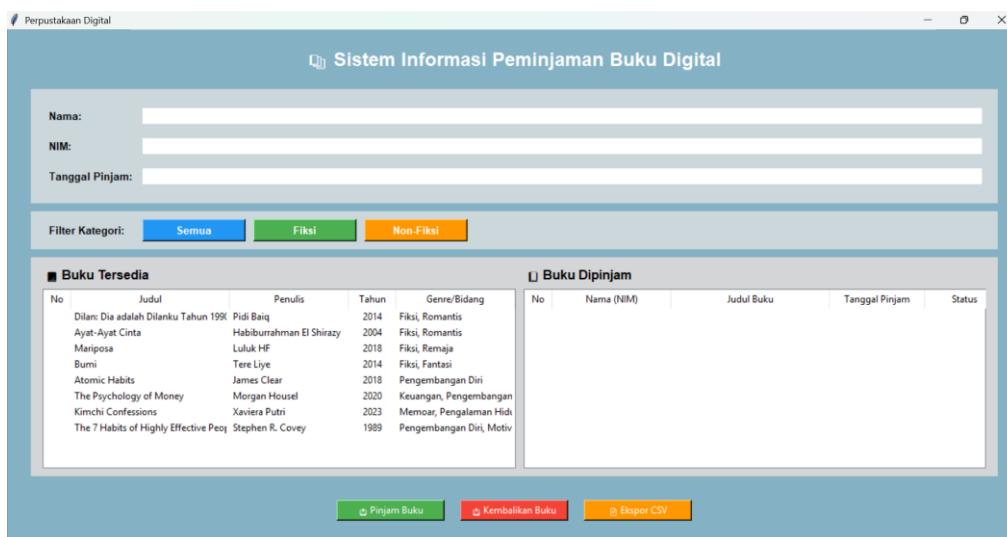
Tata cara penggunaan aplikasi ini cukup sederhana:

1. Masukkan Nama, NIM peminjam dan Tanggal Pinjam.
2. Gunakan filter untuk melihat buku Fiksi atau Non-Fiksi.
3. Pilih buku pada daftar Buku Tersedia.
4. Klik tombol 'Pinjam Buku' untuk meminjam buku.
5. Buku yang dipinjam akan muncul di daftar Buku Dipinjam.
6. Untuk mengembalikan Buku, Pilih buku di daftar Buku Dipinjam lalu klik 'Kembalikan Buku'.
7. Klik 'Ekspor CSV' untuk menyimpan laporan peminjaman.

### 5.3 Tampilan Aplikasi

Tampilan aplikasi dirancang sederhana dan mudah digunakan dengan antarmuka grafis berbasis Tkinter.

#### a. Tampilan Awal Aplikasi



## b. Hasil Peminjaman

The screenshot shows a Windows application window titled "Sistem Informasi Peminjaman Buku Digital". At the top, there are input fields for "Nama" (Nisaa), "NIM" (240981), and "Tanggal Pinjam" (14 Desember 2025). Below these are filter buttons for "Semua", "Fiksi", and "Non-Fiksi". The main area is divided into two tables: "Buku Tersedia" (Available Books) and "Buku Dipinjam" (Borrowed Books).

**Buku Tersedia**

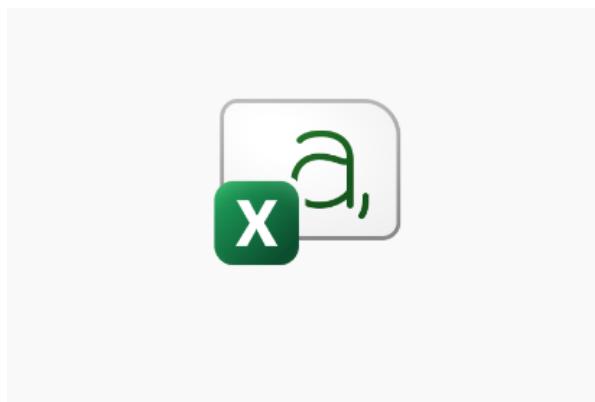
No	Judul	Penulis	Tahun	Genre/Bidang
1	Dilan: Dia adalah Dilanku Tahun 1998	Pidi Baiq	2014	Fiksi, Romantis
2	Ayat-Ayat Cinta	Habiburrahman El Shirazy	2004	Fiksi, Romantis
3	Bumi	Tere Liye	2014	Fiksi, Fantasi
4	Atomic Habits	James Clear	2018	Pengembangan Diri
5	The Psychology of Money	Morgan Housel	2020	Keuangan, Pengembangan
6	The 7 Habits of Highly Effective People	Stephen R. Covey	1989	Pengembangan Diri, Motiv

**Buku Dipinjam**

No	Nama (NIM)	Judul Buku	Tanggal Pinjam	Status
1	Nisaa	Mariposa	14 Desember 2025	Dipinjam
2	Nisaa	Kimchi Confessions	14 Desember 2025	Dipinjam

At the bottom are buttons for "Pinjam Buku", "Kembalikan Buku", and "Ekspor CSV".

## c. Ekspor Data



### Details

Type	Microsoft Excel Comma Sepa...
Size	237 bytes
File location	C:\Users\Lenovo\Downloads...
Date modified	14/12/2025 23:09

## **BAB VI PENUTUP**

### **6.1 Kesimpulan**

Berdasarkan hasil perancangan dan implementasi yang telah dilakukan, dapat disimpulkan bahwa Aplikasi Peminjaman Buku Digital berhasil dibuat dan berjalan dengan baik. Aplikasi ini mampu mengelola proses peminjaman dan pengembalian buku secara sederhana dan terstruktur dengan menerapkan konsep Object-Oriented Programming (OOP). Penggunaan antarmuka grafis berbasis Tkinter memudahkan pengguna dalam mengoperasikan aplikasi, serta fitur ekspor data ke file CSV membantu dalam pembuatan laporan peminjaman.

### **6.2 Saran**

Untuk pengembangan ke depannya, aplikasi ini masih dapat ditingkatkan dengan menambahkan sistem penyimpanan data berbasis database agar data lebih aman dan tersimpan secara permanen. Selain itu, fitur ekspor laporan ke format PDF juga dapat ditambahkan sehingga laporan peminjaman bisa langsung dicetak. Pengembangan tampilan antarmuka agar terlihat lebih modern serta penambahan sistem login pengguna juga disarankan untuk meningkatkan keamanan dan kenyamanan dalam penggunaan aplikasi.