

**SISTEM MANAJEMEN UANG KAS MAHASISWA
LAPORAN TUGAS STRUKTUR DATA**

OLEH:

HABIBURROHMAN AZZAMI (24416255201207)



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BUANA PERJUANGAN KARAWANG
2025**

DAFTAR ISI

DAFTAR ISI.....	i
BAB I PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Proyek.....	2
BAB II METODOLOGI PERANCANGAN SOLUSI.....	3
2.1 Deskripsi Aplikasi.....	3
BAB III PERANCANGAN SOLUSI.....	5
3.1 Penjelasan Teknis.....	5
3.2 Hasil Uji Coba.....	17
BAB IV KESIMPULAN DAN EVALUASI.....	23
DAFTAR PUSTAKA.....	24

BAB I

PENDAHULUAN

1.1 Latar belakang

Manajemen keuangan dalam lingkup organisasi mahasiswa, seperti pencatatan uang kas kelas, seringkali masih dilakukan secara manual menggunakan buku tulis atau spreadsheet sederhana. Metode ini rentan terhadap kesalahan pencatatan, kehilangan data, keterlambatan rekapitulasi, hingga kurangnya transparansi antara pengurus dan anggota kelas. Hal ini menjadi permasalahan nyata yang dihadapi oleh banyak mahasiswa, khususnya oleh bendahara kelas yang bertanggung jawab atas kas keuangan.

Dengan berkembangnya teknologi dan semakin luasnya penggunaan komputer dalam kehidupan akademik, dibutuhkan sebuah sistem yang mampu membantu pengelolaan uang kas secara sistematis, terstruktur, dan mudah diakses. Berdasarkan permasalahan tersebut, dibangunlah sebuah aplikasi berbasis Python yang mampu melakukan CRUD (Create, Read, Update, Delete) data kas mahasiswa serta menyajikan informasi penting seperti jumlah tunggakan, total pembayaran, riwayat transaksi, dan log aktivitas admin secara otomatis dan permanen menggunakan file CSV sebagai basis datanya.

Aplikasi ini ditujukan untuk pengurus kelas, khususnya bendahara, agar dapat memudahkan mereka dalam mencatat, menagih, dan melaporkan transaksi keuangan kelas. Selain itu, aplikasi ini juga menjadi media pembelajaran bagi mahasiswa dalam mengimplementasikan struktur data (seperti HashMap, Stack, dan List) dalam proyek nyata. Dengan antarmuka berbasis GUI (Graphical User Interface), aplikasi ini juga dirancang agar mudah digunakan oleh pengguna tanpa latar belakang teknis yang tinggi..

1.2 Rumusan Masalah

Dalam pengelolaan keuangan kas mahasiswa di tingkat kelas atau himpunan, sering dijumpai kendala seperti pencatatan yang tidak rapi, data hilang karena masih ditulis manual, serta sulitnya melacak siapa saja yang sudah atau belum membayar kas. Selain itu, kurangnya sistem pencatatan yang terpusat juga menyulitkan bendahara dalam membuat laporan keuangan secara berkala. Berdasarkan permasalahan tersebut, muncul kebutuhan akan sebuah aplikasi yang dapat mencatat pembayaran kas mahasiswa secara digital, menyimpan data secara permanen, menampilkan data tunggakan dan riwayat pembayaran, serta mudah digunakan oleh pengurus kelas. Dengan begitu, pengelolaan keuangan menjadi lebih tertata dan transparan.

1.3 Tujuan Proyek

Tujuan dari proyek akhir ini adalah untuk membangun sebuah aplikasi manajemen uang kas mahasiswa yang sederhana namun fungsional, menggunakan bahasa pemrograman Python. Aplikasi ini dirancang agar dapat mencatat dan menyimpan data pembayaran kas secara otomatis ke dalam file CSV, serta menampilkan informasi penting seperti total kas, tunggakan mahasiswa, dan riwayat pembayaran.

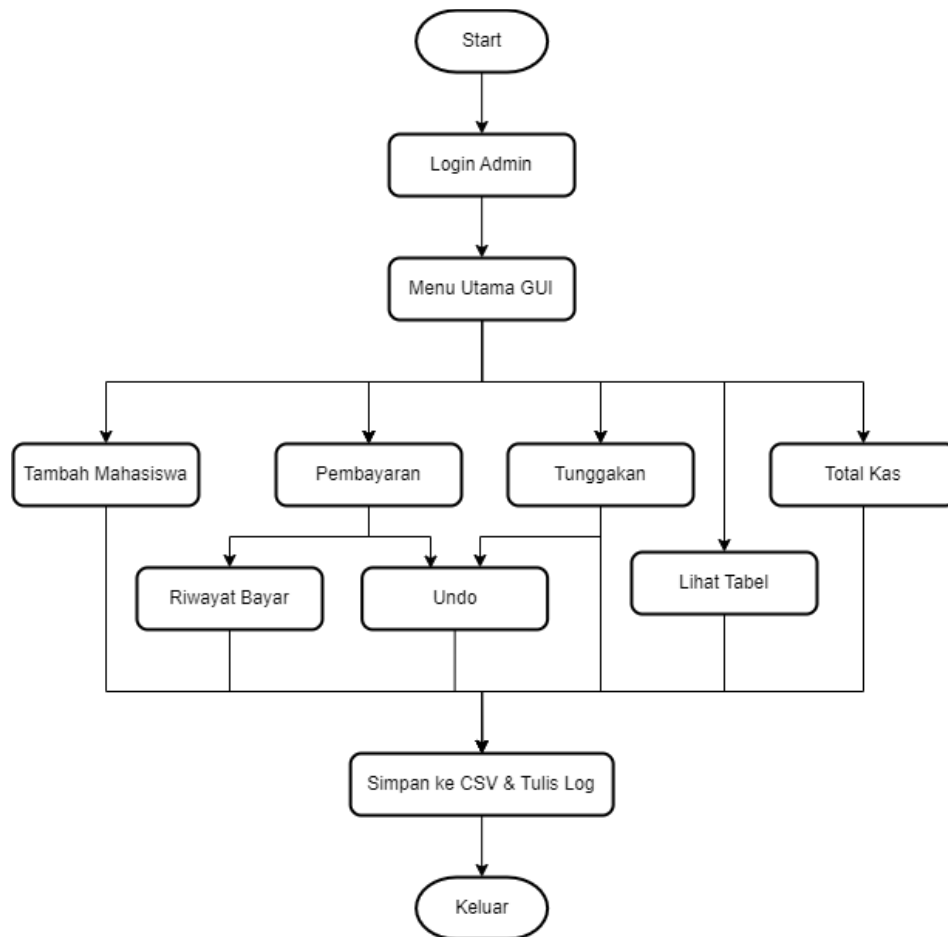
Selain itu, proyek ini bertujuan untuk melatih mahasiswa dalam menerapkan struktur data seperti HashMap dan Stack, serta meningkatkan kemampuan dalam membuat program yang modular, terstruktur, dan memiliki antarmuka grafis (GUI) sebagai nilai tambah yang mudah digunakan oleh pengguna non-teknis, seperti bendahara kelas atau pengurus organisasi.

BAB II

METODOLOGI PERANCANGAN SOLUSI

2.1 Deskripsi Aplikasi

Aplikasi yang dikembangkan adalah Sistem Manajemen Uang Kas Mahasiswa, sebuah program berbasis Python yang digunakan untuk membantu pengurus kelas, khususnya bendahara, dalam mencatat dan mengelola transaksi kas mahasiswa secara digital. Program ini memungkinkan pengguna untuk menambahkan data mahasiswa, mencatat pembayaran kas, menambahkan tunggakan secara global, melihat riwayat pembayaran, menampilkan tabel rekapitulasi kas, hingga melakukan pembatalan (*undo*) transaksi terakhir. Aplikasi ini juga menyediakan antarmuka GUI (*Graphical User Interface*) agar mudah digunakan oleh pengguna umum.



Gambar 2.1 Diagram Flowchart Umum Aplikasi

Aplikasi dimulai dengan proses login admin. Setelah login berhasil, pengguna akan masuk ke menu utama yang terdiri dari berbagai fitur seperti Tambah Mahasiswa, Pembayaran Kas, Tabel Mahasiswa, Tambah Tunggakan, Riwayat Pembayaran, dan lainnya. Setiap interaksi atau aksi yang dilakukan oleh pengguna akan disimpan dalam log aktivitas agar dapat dilacak. Data mahasiswa dan pembayaran disimpan secara permanen dalam file CSV, sehingga tetap tersedia meskipun program ditutup.

Tabel 2.1 Deskripsi Fitur

Fitur	Deskripsi
Login Admin	Login ke sistem menggunakan akun dari admin.csv.
Tambah Mahasiswa	Menambahkan mahasiswa baru ke dalam data_mahasiswa.csv.
Pembayaran Kas	Mencatat pembayaran kas oleh mahasiswa, tersimpan ke kas_mahasiswa.csv.
Undo Pembayaran	Membatalkan transaksi terakhir menggunakan struktur data Stack.
Tambah Tunggakan Global	Menambahkan beban pembayaran baru untuk semua mahasiswa.
Tabel Mahasiswa	Menampilkan rekap kas per mahasiswa (total bayar dan tunggakan).
Riwayat Pembayaran	Menampilkan riwayat pembayaran berdasarkan NIM.
Total Uang Kas	Menghitung dan menampilkan total kas yang telah dibayarkan seluruh mahasiswa.
Refresh Data	Memuat ulang data dari file CSV untuk memastikan data terbaru.

Tabel 2.2 Struktur File CSV

Nama File	Kolom yang Digunakan
data_mahasiswa.csv	NIM, Nama, Kelas
kas_mahasiswa.csv	NIM, Nama, Kelas, Jumlah, Tanggal
admin.csv	username, password
log.csv	Waktu, Username, Aksi

BAB III

PERANCANGAN SOLUSI

3.1 Penjelasan Teknis

Aplikasi dibangun menggunakan bahasa Python dengan pendekatan modular dan GUI berbasis Tkinter. Data disimpan secara permanen dalam file CSV, sehingga dapat diakses kembali saat aplikasi dijalankan ulang. Struktur data utama yang digunakan adalah:

- **HashMap**, digunakan untuk menyimpan data mahasiswa dengan NIM sebagai key.
- **Stack**, digunakan untuk menyimpan histori pembayaran terakhir sebagai dasar fitur Undo Pembayaran.
- **List**, digunakan untuk menyimpan daftar tunggakan global dan log aktivitas.

```
1 import tkinter as tk
2 from tkinter import messagebox, ttk
3 from auth import login_admin
4 from logic import *
5 from database import load_mahasiswa, load_tunggakan
6 from utils import tulis_log
7
8 class KasApp:
9     def __init__(self, root):
10         self.root = root
11         self.username = None
12         self.login_window()
13
14     def login_window(self):
15         self.root.title("Login Admin")
16         tk.Label(self.root, text="Username").grid(row=0, column=0)
17         tk.Label(self.root, text="Password").grid(row=1, column=0)
18         self.user_entry = tk.Entry(self.root)
19         self.pass_entry = tk.Entry(self.root, show="*")
20         self.user_entry.grid(row=0, column=1)
21         self.pass_entry.grid(row=1, column=1)
22         tk.Button(self.root, text="Login", command=self.login).grid(row=2, column=0, columnspan=2)
23
24     def login(self):
25         u = self.user_entry.get()
26         p = self.pass_entry.get()
27         if login_admin(u, p):
28             self.username = u
29             tulis_log(u, "Login berhasil")
30             self.root.destroy()
31             self.main_window()
32         else:
33             messagebox.showerror("Login Gagal", "Username/password salah.")
34
```

Gambar 3.1.1 Main_gui.py bagian 1

Modul main_gui.py berfungsi sebagai pusat dari seluruh jalannya aplikasi. Pada bagian awal program, sejumlah pustaka dan modul penting diimpor, yaitu tkinter sebagai pustaka GUI utama, serta messagebox untuk memunculkan pesan pop-up ke pengguna. Selanjutnya,

beberapa modul buatan sendiri juga diimpor, antara lain auth untuk login admin, logic untuk fitur logika aplikasi seperti perhitungan tunggakan dan stack, database untuk membaca dan menyimpan data ke file CSV, serta utils yang memuat fungsi bantu seperti pencatatan log.

Setelah semua modul diimpor, kelas KasApp didefinisikan sebagai jantung dari aplikasi GUI. Konstruktor `__init__` menerima parameter root, yang merupakan jendela utama Tkinter. Kemudian, aplikasi langsung memanggil metode `login_window()` untuk menampilkan form login saat pertama dijalankan.

```
def main_window(self):
    load_mahasiswa()
    load_tunggakan()

    self.root = tk.Tk()
    self.root.title("Sistem Uang Kas Mahasiswa")

    btns = [
        ("Tambah Mahasiswa", lambda: tambah_mahasiswa(self.username)),
        ("Hapus Mahasiswa", lambda: hapus_mahasiswa(self.username)),
        ("Pembayaran Kas", lambda: bayar_kas(self.username)),
        ("Undo Pembayaran Terakhir", lambda: undo_pembayaran(self.username)),
        ("Tampilkan Tabel Mahasiswa", tampil_tabel),
        ("Riwayat Pembayaran", riwayat_bayar),
        ("Tambah Tunggakan", lambda: tambah_tunggakan(self.username)),
        ("Total Uang Kas", total_kas),
        ("Refresh Data", refresh_data),
        ("Keluar", self.root.quit),
    ]

    for i, (text, cmd) in enumerate(btns):
        tk.Button(self.root, text=text, width=30, command=cmd).grid(row=i, column=0, pady=3, padx=10)

    self.root.mainloop()

if __name__ == '__main__':
    root = tk.Tk()
    app = KasApp(root)
    root.mainloop()
```

Gambar 3.1.2 Main_gui.py bagian 2

Setelah login berhasil, pengguna diarahkan ke jendela menu utama. Di sini, Tkinter digunakan kembali untuk membuat antarmuka tombol-tombol menu. Masing-masing tombol dikaitkan langsung dengan metode di dalam kelas KasApp yang sesuai dengan fungsinya. Misalnya, tombol “Tambah Mahasiswa” akan memanggil `self.tambah_mahasiswa`, dan tombol “Bayar Kas” akan memanggil `self.bayar_kas`. Seluruh tombol disusun dengan metode `grid()` dan tampilan akan dijalankan terus-menerus menggunakan `mainloop()`.


```

1 import csv
2 import os
3
4 FILE_ADMIN = "data/admin.csv"
5
6 def login_admin(username, password):
7     if not os.path.exists(FILE_ADMIN):
8         return False
9     with open(FILE_ADMIN, newline='') as f:
10         reader = csv.DictReader(f)
11         for row in reader:
12             if row['username'] == username and row['password'] == password:
13                 return True
14     return False
15

```

Gambar 3.1.3 Auth.py

Modul database.py memuat fungsi-fungsi penting yang menangani pengambilan dan penyimpanan data dari file CSV. Salah satu fungsi utamanya adalah load_mahasiswa(), yang bertugas membaca isi data_mahasiswa.csv. File ini menyimpan informasi dasar mahasiswa seperti NIM, nama, dan kelas. Setiap baris dibaca dan disimpan ke dalam struktur data mahasiswa dalam bentuk dictionary (HashMap), dengan NIM sebagai kunci utamanya. Dengan pendekatan ini, pencarian mahasiswa dapat dilakukan dengan cepat dan efisien.

```

1 import csv, os
2 from datetime import datetime
3
4 FILE_MHS = "data/data_mahasiswa.csv"
5 FILE_KAS = "data/kas_mahasiswa.csv"
6 FILE_LOG = "data/log.csv"
7
8 mahasiswa = {}
9 tunggakan_histori = []
10
11 def load_mahasiswa():
12     mahasiswa.clear()
13     if os.path.exists(FILE_MHS):
14         with open(FILE_MHS, newline='') as f:
15             reader = csv.DictReader(f)
16             for row in reader:
17                 mahasiswa[row['NIM']] = {
18                     'nama': row['Nama'],
19                     'kelas': row['Kelas'],
20                     'riwayat': []
21                 }
22     if os.path.exists(FILE_KAS):
23         with open(FILE_KAS, newline='') as f:
24             reader = csv.DictReader(f)
25             for row in reader:
26                 nim = row['NIM']
27                 if nim in mahasiswa:
28                     mahasiswa[nim]['riwayat'].append({
29                         'jumlah': int(row['Jumlah']),
30                         'tanggal': row['Tanggal']
31                     })
32

```

Gambar 3.1.4 Database.py bagian 1

Modul database.py berfungsi untuk menangani semua interaksi dengan file CSV, baik untuk data mahasiswa, data kas pembayaran, maupun log tunggakan. File ini juga menyimpan struktur data utama berupa dictionary mahasiswa dan list tunggakan_histori yang digunakan oleh seluruh aplikasi.

Gambar di atas menginisialisasi path file CSV yang digunakan (data_mahasiswa.csv, kas_mahasiswa.csv, dan log.csv). Fungsi load_mahasiswa() bertugas membaca data dari dua file tersebut. Pertama, file mahasiswa dibaca dan datanya dimasukkan ke dalam dictionary mahasiswa menggunakan NIM sebagai kunci (HashMap). Kemudian file kas_mahasiswa.csv dibaca untuk melengkapi riwayat pembayaran mahasiswa dalam bentuk list of dictionary.

```
32
33 def simpan_mahasiswa():
34     with open(FILE_MHS, 'w', newline='') as f:
35         writer = csv.writer(f)
36         writer.writerow(['NIM', 'Nama', 'Kelas'])
37         for nim, data in mahasiswa.items():
38             writer.writerow([nim, data['nama'], data['kelas']])
39
40 def simpan_pembayaran(nim, jumlah):
41     tanggal = datetime.now().strftime('%Y-%m-%d')
42     data = mahasiswa[nim]
43     with open(FILE_KAS, 'a', newline='') as f:
44         writer = csv.writer(f)
45         writer.writerow([nim, data['nama'], data['kelas'], jumlah, tanggal])
46
47 def load_tunggakan():
48     tunggakan_histori.clear()
49     if os.path.exists(FILE_LOG):
50         with open(FILE_LOG, newline='') as f:
51             reader = csv.reader(f)
52             next(reader, None)
53             for row in reader:
54                 if "Tunggakan global Rp" in row[2]:
55                     nominal = int(row[2].split("Rp")[1])
56                     tunggakan_histori.append(nominal)
57
```

Gambar 3.1.5 Database.py bagian 2

Bagian kedua modul ini berisi fungsi simpan_mahasiswa() dan simpan_pembayaran(), yang bertugas menulis data mahasiswa baru dan data pembayaran ke file CSV secara permanen. Fungsi simpan_mahasiswa() menulis ulang seluruh file, sementara simpan_pembayaran() menambahkan transaksi baru (append).

Fungsi `load_tunggakan()` membaca kembali nominal-nominal tunggakan yang pernah ditambahkan secara global dan mencatatkannya dalam list `tunggakan_histori`. Data ini diambil dari catatan di `log.csv` yang menyimpan semua aksi admin, termasuk menambahkan tagihan baru.

```
1 import csv
2 from datetime import datetime
3
4 FILE_LOG = "data/log.csv"
5
6 def tulis_log(user, aksi):
7     waktu = datetime.now().strftime('%Y-%m-%d %H:%M:%S')
8     with open(FILE_LOG, 'a', newline='') as f:
9         writer = csv.writer(f)
10        writer.writerow([waktu, user, aksi])
11
12 from database import mahasiswa, tunggakan_histori
13
14 def total_tagihan():
15     return sum(tunggakan_histori)
16
17 def total_bayar(nim):
18     return sum([r['jumlah'] for r in mahasiswa[nim]['riwayat']])
19
20 def sisa_tunggakan(nim):
21     return max(0, total_tagihan() - total_bayar(nim))
22
```

Gambar 3.1.6. Utils.py

Modul `utils.py` berfungsi sebagai pendukung untuk berbagai operasi perhitungan dan pencatatan log. Fungsi `tulis_log()` digunakan untuk mencatat setiap aksi penting yang dilakukan admin ke dalam file `log.csv`, lengkap dengan informasi waktu, nama pengguna, dan deskripsi aksi. Hal ini penting untuk keperluan audit atau pelacakan histori penggunaan aplikasi.

Bagian berikutnya dari modul ini menangani perhitungan keuangan secara otomatis. Fungsi `total_tagihan()` menjumlahkan seluruh daftar tunggakan global yang tersimpan di list `tunggakan_histori`. Fungsi `total_bayar(nim)` menghitung total uang kas yang sudah dibayar oleh seorang mahasiswa berdasarkan data riwayat pembayarannya. Sementara itu, `sisa_tunggakan(nim)` menghitung selisih antara total tagihan dan jumlah pembayaran, untuk mengetahui apakah masih ada utang kas yang belum dilunasi. Fungsi-fungsi ini sangat penting dalam proses pengecekan status kas mahasiswa dan digunakan di berbagai bagian GUI aplikasi.

```

1 import tkinter as tk
2 from tkinter import messagebox, ttk
3 from datetime import datetime
4 import os, csv
5
6 from database import mahasiswa, simpan_mahasiswa, simpan_pembayaran, load_mahasiswa, load_tunggakan
7 from utils import tulis_log, total_bayar, sisa_tunggakan, total_tagihan
8
9 FILE_KAS = "data/kas_mahasiswa.csv"
10 undo_stack = []
11
12 def tambah_mahasiswa(user):
13     win = tk.Toplevel()
14     win.title("Tambah Mahasiswa")
15
16     nim = tk.Entry(win)
17     nama = tk.Entry(win)
18     kelas = tk.Entry(win)
19
20     for i, label in enumerate(["NIM", "Nama", "Kelas"]):
21         tk.Label(win, text=label).grid(row=i, column=0)
22     for i, entry in enumerate([nim, nama, kelas]):
23         entry.grid(row=i, column=1)
24
25     def simpan():
26         n = nim.get()
27         if n in mahasiswa:
28             messagebox.showerror("Gagal", "NIM sudah ada.")
29             return
30         mahasiswa[n] = {'nama': nama.get(), 'kelas': kelas.get(), 'riwayat': []}
31         simpan_mahasiswa()
32         tulis_log(user, f"Tambah mahasiswa: {n} - {nama.get()}")
33         messagebox.showinfo("Sukses", "Mahasiswa ditambahkan.")
34         win.destroy()
35
36     tk.Button(win, text="Simpan", command=simpan).grid(row=3, column=0, columnspan=2)
37

```

Gambar 3.1.7 Logic.py bagian 1

Modul logic.py merupakan inti dari implementasi logika aplikasi. Semua fungsi interaktif dengan pengguna dan GUI Tkinter didefinisikan di sini. File ini mengimpor modul pendukung seperti database, utils, serta komponen GUI dari tkinter dan ttk.

Fungsi `tambah_mahasiswa(user)` digunakan untuk menambahkan mahasiswa baru ke dalam data permanen. GUI ini menerima input NIM, nama, dan kelas melalui entri Tkinter. Jika NIM belum ada di dalam dictionary mahasiswa, maka data baru disimpan dan dicatat dalam log.csv.

```

37
38 def hapus_mahasiswa(user):
39     win = tk.Toplevel()
40     win.title("Hapus Mahasiswa")
41
42     tk.Label(win, text="NIM:").pack()
43     nim = tk.Entry(win)
44     nim.pack()
45
46     def hapus():
47         n = nim.get()
48         if n not in mahasiswa:
49             messagebox.showerror("Error", "Tidak ditemukan.")
50             return
51         nama = mahasiswa[n]['nama']
52         del mahasiswa[n]
53         simpan_mahasiswa()
54
55         if os.path.exists(FILE_KAS):
56             rows = []
57             with open(FILE_KAS, newline='') as f:
58                 reader = csv.DictReader(f)
59                 for row in reader:
60                     if row['NIM'] != n:
61                         rows.append(row)
62             with open(FILE_KAS, 'w', newline='') as f:
63                 writer = csv.DictWriter(f, fieldnames=['NIM', 'Nama', 'Kelas', 'Jumlah', 'Tanggal'])
64                 writer.writeheader()
65                 writer.writerows(rows)
66
67         tulis_log(user, f"Hapus mahasiswa: {n} - {nama}")
68         messagebox.showinfo("Berhasil", f"{n} dihapus.")
69         win.destroy()
70
71     tk.Button(win, text="Hapus", command=hapus).pack(pady=5)

```

Gambar 3.1.8 Logic.py bagian 2

Fungsi `hapus_mahasiswa(user)` menangani proses penghapusan data mahasiswa berdasarkan NIM yang diinputkan. Selain menghapus dari dictionary, fungsi ini juga menghapus semua entri pembayaran mahasiswa tersebut dari file `kas_mahasiswa.csv`. Fungsi ini juga mencatat aksi ke file log sebagai bentuk dokumentasi perubahan data.

```

72
73 def bayar_kas(user):
74     win = tk.Toplevel()
75     win.title("Bayar Kas")
76
77     nim = tk.Entry(win)
78     jumlah = tk.Entry(win)
79
80     tk.Label(win, text="NIM").grid(row=0, column=0)
81     tk.Label(win, text="Nominal").grid(row=1, column=0)
82     nim.grid(row=0, column=1)
83     jumlah.grid(row=1, column=1)
84
85     def bayar():
86         n = nim.get()
87         if n not in mahasiswa:
88             messagebox.showerror("Gagal", "Tidak ditemukan.")
89             return
90         try:
91             jml = int(jumlah.get())
92         except:
93             messagebox.showerror("Gagal", "Nominal salah.")
94             return
95         mahasiswa[n]['riwayat'].append({'jumlah': jml, 'tanggal': datetime.now().strftime('%Y-%m-%d')})
96         undo_stack.append({'nim': n, 'jumlah': jml, 'tanggal': datetime.now().strftime('%Y-%m-%d')})
97         simpan_pembayaran(n, jml)
98         simpan_mahasiswa()
99         tulis_log(user, f"Pembayaran Rp{jml} oleh {n}")
100         messagebox.showinfo("Sukses", "Pembayaran dicatat.")
101         win.destroy()
102
103     tk.Button(win, text="Bayar", command=bayar).grid(row=2, column=0, columnspan=2)
104

```

Gambar 3.1.9. Logic.py bagian 3

Berikutnya adalah `bayar_kas(user)` yang memungkinkan admin mencatat pembayaran uang kas oleh mahasiswa. Input NIM dan nominal pembayaran akan disimpan ke dalam riwayat pembayaran dan file `kas_mahasiswa.csv`. Data pembayaran terakhir juga dimasukkan ke stack `undo_stack` untuk memungkinkan pembatalan pembayaran jika diperlukan.

```
105 def undo_pembayaran(user):
106     if not undo_stack:
107         messagebox.showinfo("Undo", "Tidak ada transaksi.")
108         return
109     last = undo_stack.pop()
110     nim, jumlah, tanggal = last['nim'], last['jumlah'], last['tanggal']
111
112     mahasiswa[nim]['riwayat'] = [r for r in mahasiswa[nim]['riwayat']
113                                   if not (r['jumlah'] == jumlah and r['tanggal'] == tanggal)]
114     simpan_mahasiswa()
115
116     rows = []
117     with open(FILE_KAS, newline='') as f:
118         reader = csv.DictReader(f)
119         for row in reader:
120             if row['NIM'] == nim and row['Jumlah'] == str(jumlah) and row['Tanggal'] == tanggal:
121                 continue
122             rows.append(row)
123     with open(FILE_KAS, 'w', newline='') as f:
124         writer = csv.DictWriter(f, fieldnames=['NIM', 'Nama', 'Kelas', 'Jumlah', 'Tanggal'])
125         writer.writeheader()
126         writer.writerows(rows)
127
128     tulis_log(user, f"Undo pembayaran Rp{jumlah} oleh {nim}")
129     messagebox.showinfo("Undo", f"Pembayaran {nim} dibatalkan.")
```

Gambar 3.1.10 Logic.py bagian 4

Fungsi `undo_pembayaran(user)` bekerja dengan mengambil entri terakhir dari `undo_stack`, kemudian menghapus pembayaran tersebut dari riwayat dan file kas. Hal ini memastikan bahwa jika ada kesalahan input, pengguna masih dapat membatalkan transaksi terakhir secara langsung.

```

def tampil_tabel():
    win = tk.Toplevel()
    win.title("Tabel Mahasiswa")
    from database import mahasiswa

    tree = ttk.Treeview(win)
    tree["columns"] = ("NIM", "Nama", "Kelas", "Bayar", "Tunggakan")
    tree.column("#0", width=0, stretch=tk.NO)
    for col in tree["columns"]:
        tree.heading(col, text=col)
        tree.column(col, anchor="w", width=150)
    tree.pack(expand=True, fill='both')

    def isi_data():
        for i in tree.get_children():
            tree.delete(i)
        load_mahasiswa()
        load_tunggakan()
        for nim, data in mahasiswa.items():
            tree.insert("", "end", values=(
                nim,
                data['nama'],
                data['kelas'],
                f"Rp{total_bayar(nim):,}",
                f"Rp{sisa_tunggakan(nim):,}"
            ))

    tk.Button(win, text="Refresh", command=isi_data).pack(pady=5)
    isi_data()

```

Gambar 3.1.11 Logic.py bagian 5

Sementara itu, `tampil_tabel()` membuat antarmuka tabel Tkinter menggunakan `ttk.Treeview`. Fungsi ini menampilkan daftar mahasiswa beserta total pembayaran dan tunggakan mereka. Tombol *Refresh* memperbarui isi tabel dari data terbaru dengan memanggil ulang `load_mahasiswa()` dan `load_tunggakan()`.

```

def riwayat_bayar():
    win = tk.Toplevel()
    win.title("Riwayat")

    tk.Label(win, text="NIM").pack()
    entry = tk.Entry(win)
    entry.pack()
    box = tk.Text(win, height=10, width=50)
    box.pack()

    def tampil():
        nim = entry.get()
        if nim not in mahasiswa:
            messagebox.showerror("Gagal", "Tidak ditemukan.")
            return
        box.delete("1.0", tk.END)
        for r in mahasiswa[nim]['riwayat']:
            box.insert(tk.END, f"- Rp{r['jumlah']:,} pada {r['tanggal']}\n")

    def refresh():
        load_mahasiswa()
        tulis_log("admin", "Refresh data riwayat")
        messagebox.showinfo("Refresh", "Data dimuat.")

    tk.Button(win, text="Lihat", command=tampil).pack(pady=2)
    tk.Button(win, text="Refresh", command=refresh).pack()

```

Gambar 3.1.12 Logic.py bagian 6

Selanjutnya fungsi `riwayat_bayar()` yang digunakan untuk menampilkan riwayat pembayaran berdasarkan NIM mahasiswa. Jika mahasiswa ditemukan, maka daftar pembayaran ditampilkan dalam Text widget berupa informasi nominal dan tanggal pembayaran. Tersedia pula tombol *Refresh* untuk memuat ulang data dari file CSV jika ada perubahan baru.


```

def tambah_tunggakan(user):
    win = tk.Toplevel()
    win.title("Tambah Tunggakan")

    tk.Label(win, text="Nominal").grid(row=0, column=0)
    nominal = tk.Entry(win)
    nominal.grid(row=0, column=1)

    def tambah():
        try:
            nilai = int(nominal.get())
        except:
            messagebox.showerror("Error", "Input harus angka.")
            return
        from database import tunggakan_histori
        tunggakan_histori.append(nilai)
        tulis_log(user, f"Tunggakan global Rp{nilai}")
        messagebox.showinfo("Sukses", f"Tunggakan Rp{nilai} ditambahkan.")
        win.destroy()

    tk.Button(win, text="Tambah", command=tambah).grid(row=1, column=0, columnspan=2)

def total_kas():
    total = 0
    if os.path.exists(FILE_KAS):
        with open(FILE_KAS, newline='') as f:
            reader = csv.DictReader(f)
            for row in reader:
                total += int(row['Jumlah'])
    messagebox.showinfo("Total Kas", f"Total kas masuk: Rp{total:,}")

```

Gambar 3.1.13 Logic.py bagian 7

Fungsi `tambah_tunggakan(user)` memungkinkan admin menambahkan nilai tunggakan baru yang akan berlaku untuk semua mahasiswa. Nilai ini disimpan dalam list `tunggakan_histori`, yang merupakan dasar perhitungan tagihan mahasiswa. Proses ini juga dicatat ke dalam log untuk keperluan pelacakan aksi.

Kemudian fungsi `total_kas()` menghitung total semua nominal pembayaran yang telah tercatat dalam file `kas_mahasiswa.csv`. Fungsi ini dibutuhkan untuk mengetahui jumlah kas yang telah terkumpul secara keseluruhan.

```

def refresh_data():
    load_mahasiswa()
    load_tunggakan()
    messagebox.showinfo("Refresh", "Data berhasil dimuat ulang.")

```

Gambar 3.1.14 Logic.py bagian 9

Terakhir, `refresh_data()` memanggil ulang fungsi `load_mahasiswa()` dan `load_tunggakan()` untuk memuat ulang semua data dari CSV. Fungsi ini memastikan bahwa perubahan dari luar GUI tetap dapat dimuat ulang saat aplikasi dijalankan tanpa perlu menutup program.

Masing-masing operasi melibatkan interaksi dengan struktur data mahasiswa dan file CSV menggunakan fungsi dari database.py. Fungsi tulis_log() dari utils.py dipanggil setiap kali terjadi perubahan penting, seperti saat menambahkan atau menghapus mahasiswa maupun mencatat pembayaran, sebagai catatan histori.

```

1  NIM,Nama,Kelas
2  24416255201213,AAN RIPANDI,IF24A
3  24416255201223,ABDULROHIMEFENDI,IF24A
4  24416255201227,ADITYA SHAID RAHMATTULLAH,IF24A
5  24416255201208,AGRY SEPTIANI ARIFIN,IF24A
6  24416255201075,AHMAD ARIFIN,IF24A
7  24416255201222,AHMAD HAFIZH,IF24A
8  24416255201212,ALFI LATIFATUL MUNAWAROH,IF24A
9  24416255201032,ALYALUNA SASMITA,IF24A
10 24416255201068,ANGGA PRATAMA PUTRA,IF24A
11 24416255201204,ANWAR RAMADHAN,IF24A
12 24416255201043,ATILA FATURRAHMAN PRATAMA,IF24A
13 24416255201110,DAFFA ANDHIKA PRATAMA,IF24A
14 24416255201123,DANNYS PANJI KURNIAWAN,IF24A
15 24416255201236,DIMAS FAJRI MUHAMMAD,IF24A

```

Gambar 3.1.15 Contoh isi data_mahasiswa.csv

File data_mahasiswa.csv menyimpan informasi pokok mahasiswa seperti NIM, nama, dan kelas. File ini dibaca dan ditulis saat proses penambahan maupun penghapusan mahasiswa.

```

1  NIM,Nama,Kelas,Jumlah,Tanggal
2  24416255201213,AAN RIPANDI,IF24A,5000,2025-07-11
3  24416255201237,DIMAZ WAHYUDY,IF24A,5000,2025-07-11
4  24416255201203,FAHRY RASYIDINANTA PUTRA,IF24A,5000,2025-07-11

```

Gambar 3.1.16 Contoh isi kas_mahasiswa.csv

File kedua, kas_mahasiswa.csv, menyimpan riwayat pembayaran dari setiap mahasiswa yang terdiri dari NIM, nama, kelas, jumlah pembayaran, dan tanggal transaksi. File ini secara otomatis diperbarui saat mahasiswa melakukan pembayaran kas melalui antarmuka GUI.

```

1  Waktu,Username,Aksi
2  2025-07-11 21:57:52,azzami,Login berhasil
3  2025-07-11 21:58:41,azzami,Pembayaran Rp5000 oleh 24416255201213
4  2025-07-11 21:59:11,azzami,Pembayaran Rp5000 oleh 244162552012372025-07-11 22:02:35,azzami,Login berhasil
5  2025-07-11 22:02:48,azzami,Tunggakan global Rp10000
6  2025-07-11 22:03:40,azzami,Tambah mahasiswa: 24416255201102 - MUHAMAD BAGAS NUR AL FARID
7  2025-07-11 22:03:58,admin,Refresh data riwayat
8  2025-07-11 22:04:14,admin,Refresh data riwayat

```

Gambar 3.1.17 Contoh isi log.csv

File ketiga adalah log.csv, yang digunakan untuk mencatat seluruh aktivitas penting yang dilakukan oleh admin dalam aplikasi, seperti login, penambahan data, pembayaran, dan penghapusan data. Catatan log ini membantu dalam pelacakan historis dan debugging apabila terjadi kesalahan dalam sistem.

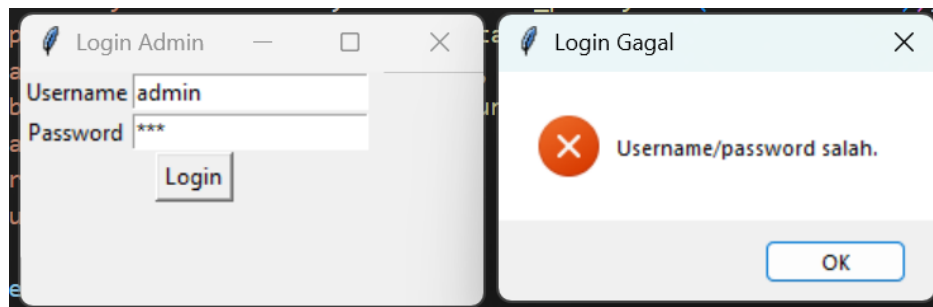
```
1 username,password
2 admin,1234
3 azzami,1207
4
5
```

Gambar 3.1.18. Contoh isi admin.csv

Terakhir, admin.csv menyimpan username dan password dari pengguna (admin) yang berwenang menjalankan sistem. Otentikasi sederhana berbasis file ini digunakan saat awal program berjalan, sebelum memasuki menu utama aplikasi.

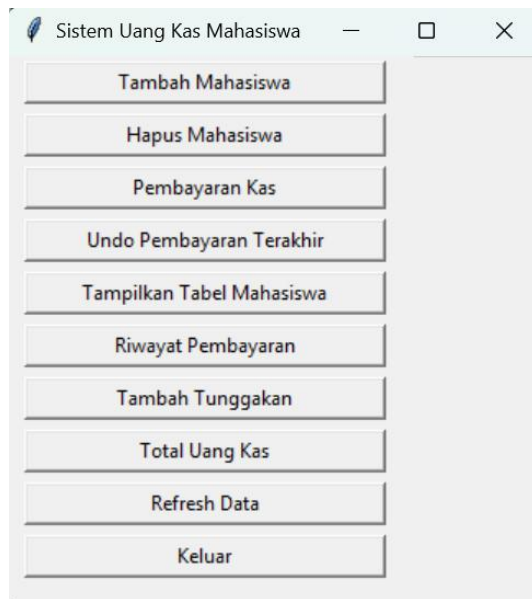
Dengan struktur file CSV yang sederhana namun efektif ini, seluruh proses penyimpanan data dilakukan secara lokal dan permanen. Hal ini membuat aplikasi tetap ringan, mudah dipindahkan, dan tidak membutuhkan instalasi basis data eksternal, namun tetap bisa menangani kebutuhan pencatatan kas mahasiswa secara sistematis.

3.2 Hasil Uji Coba



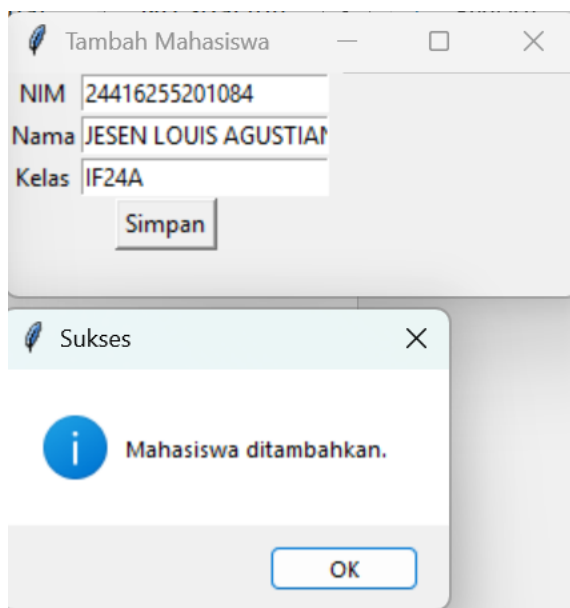
Gambar 3.2.1 Tampilan menu Login Admin

Gambar ini menunjukkan tampilan awal dari aplikasi, yaitu halaman login admin. Pengguna diminta untuk memasukkan kombinasi username dan password yang telah disimpan dalam berkas admin.csv. Jika kombinasi benar, pengguna dapat mengakses seluruh fitur aplikasi. Jika salah, akan muncul pesan kesalahan. Fitur ini merupakan pintu awal dari sistem, memastikan hanya admin yang memiliki akses untuk mengelola data mahasiswa dan kas.



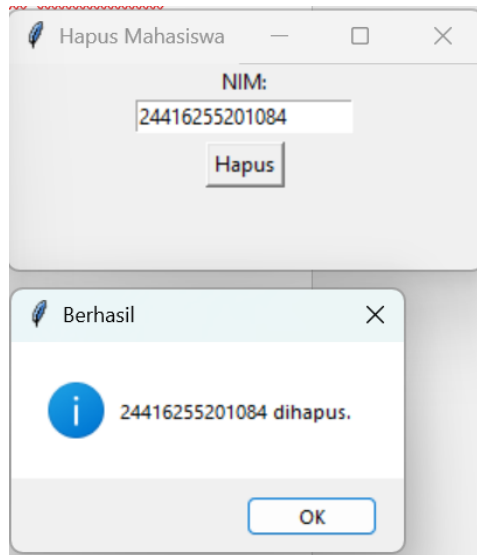
Gambar 3.2.2 Tampilan menu utama

Setelah berhasil login, pengguna diarahkan ke menu utama aplikasi. Gambar ini memperlihatkan tombol-tombol navigasi untuk mengakses berbagai fitur utama, seperti menambah mahasiswa, mencatat pembayaran, melihat riwayat, serta melakukan perhitungan total kas dan tunggakan. Tampilan ini menjadi pusat kontrol utama dalam manajemen uang kas mahasiswa.



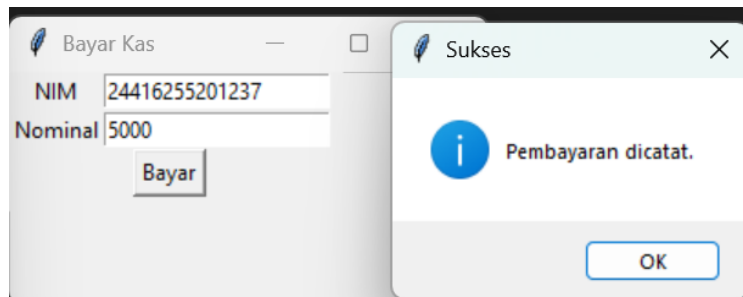
Gambar 3.2.3 Tambah Mahasiswa

Gambar ini menampilkan jendela pop-up dari fitur tambah mahasiswa. Pengguna mengisi NIM, nama, dan kelas mahasiswa. Jika NIM sudah terdaftar, sistem akan memberikan peringatan. Jika data valid, mahasiswa baru akan ditambahkan ke data_mahasiswa.csv, dan notifikasi keberhasilan akan ditampilkan. Ini membuktikan fungsi *Create* berjalan sesuai.



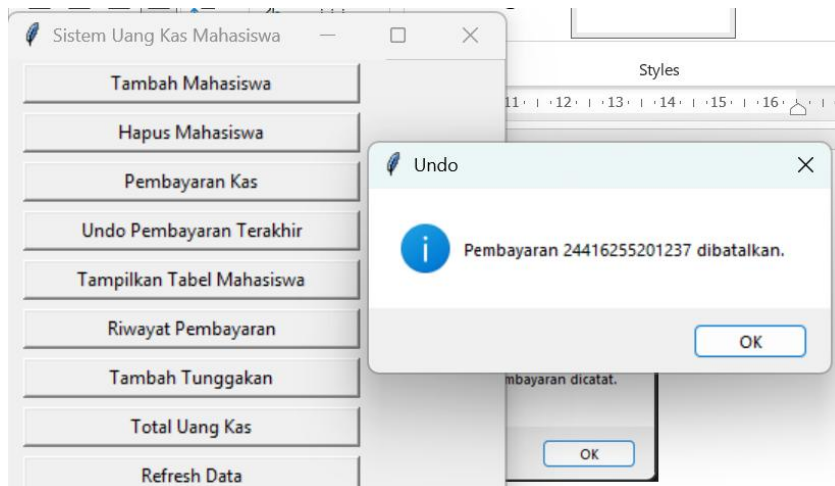
Gambar 3.2.4 Hapus Mahasiswa

Pada gambar ini terlihat fitur hapus mahasiswa yang meminta input berupa NIM. Jika NIM ditemukan, data mahasiswa akan dihapus dari memori dan juga file CSV, termasuk data pembayarannya. Gambar ini memastikan bahwa fitur *Delete* benar-benar menghapus data terkait dengan bersih dari seluruh sistem.



Gambar 3.2.5 Pembayaran Kas

Gambar ini menunjukkan form pembayaran kas mahasiswa. Admin cukup mengisi NIM dan nominal pembayaran. Sistem akan memvalidasi input, lalu menyimpan data ke kas_mahasiswa.csv dan mencatat log. Ini membuktikan fitur *Update* untuk menambahkan transaksi baru berjalan lancar, serta terjadi pencatatan di file permanen.



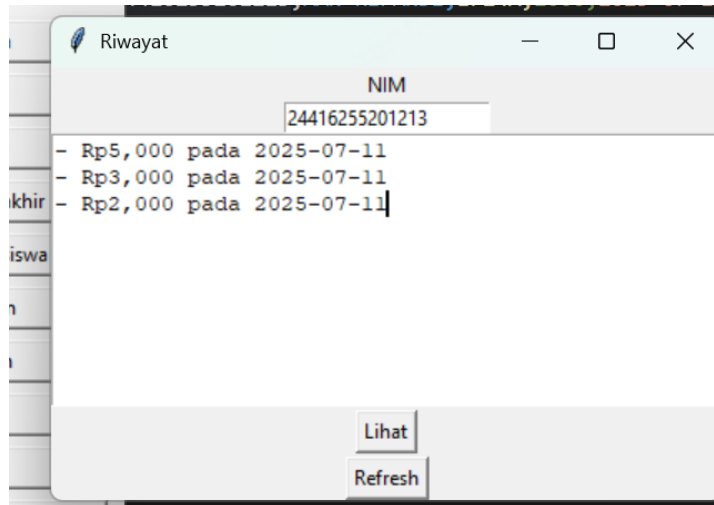
Gambar 3.2.6 Cancel Pembayaran

Fitur ini terlihat pada gambar sebagai validasi bahwa sistem menyimpan riwayat pembayaran terakhir dalam sebuah struktur data Stack (`undo_stack`). Jika admin mengaktifkan fitur Undo, maka transaksi terakhir akan dibatalkan secara otomatis, baik dari memori maupun file CSV. Ini membuktikan penerapan struktur data Stack secara nyata dalam aplikasi.

NIM	Nama	Kelas	Bayar	Tunggakan
24416255201213	AAN RIPANDI	IF24A	Rp5,000	Rp5,000
24416255201223	ABDULROHIMEFENDI	IF24A	Rp0	Rp10,000
24416255201227	ADITYA SHAID RAHMATTULLA	IF24A	Rp0	Rp10,000
24416255201208	AGRY SEPTIANI ARIFIN	IF24A	Rp0	Rp10,000
24416255201075	AHMAD ARIFIN	IF24A	Rp0	Rp10,000
24416255201222	AHMAD HAFIZH	IF24A	Rp0	Rp10,000
24416255201212	ALFI LATIFATUL MUNAWAROF	IF24A	Rp0	Rp10,000
24416255201032	ALYALUNA SASMITA	IF24A	Rp0	Rp10,000
24416255201068	ANGGA PRATAMA PUTRA	IF24A	Rp0	Rp10,000
24416255201204	ANWAR RAMADHAN	IF24A	Rp0	Rp10,000
24416255201043	ATILA FATURRAHMAN PRATA	IF24A	Rp0	Rp10,000
24416255201110	DAFFA ANDHIKA PRATAMA	IF24A	Rp0	Rp10,000
24416255201123	DANING DAMILKUDHAWARI	IF24A	Rp0	Rp10,000

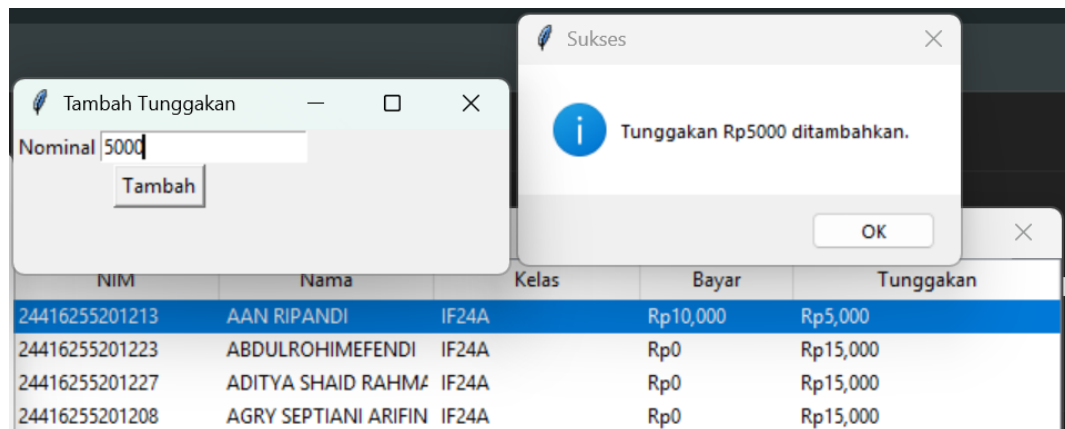
Gambar 3.2.7 Tabel Mahasiswa

Gambar ini memperlihatkan tampilan tabel mahasiswa dalam bentuk Treeview. Setiap baris menampilkan informasi lengkap: NIM, nama, kelas, total pembayaran, dan total tunggakan (jika ada). Ini merupakan fitur *Read* dari sistem, memperlihatkan data terkini dengan perhitungan otomatis berdasarkan histori pembayaran dan tunggakan yang ditetapkan.



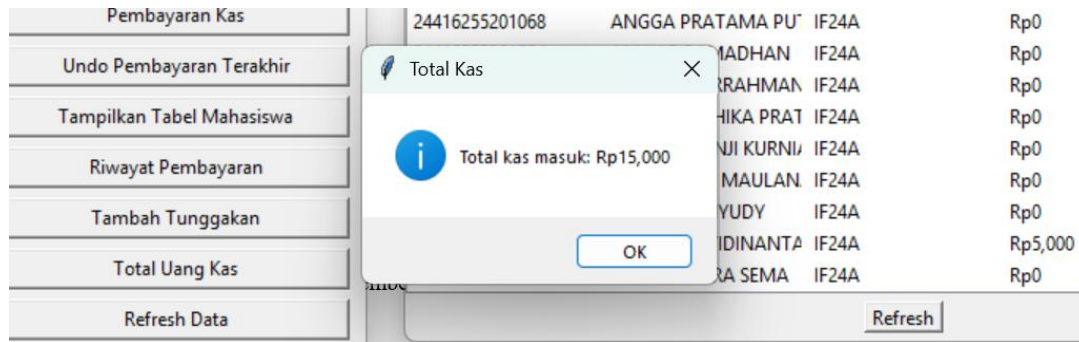
Gambar 3.2.6 Riwayat Pembayaran

Fitur ini memperlihatkan riwayat pembayaran dari satu mahasiswa berdasarkan NIM. Gambar ini menunjukkan tampilan box teks yang berisi rincian transaksi, lengkap dengan nominal dan tanggal. Ini menegaskan bahwa seluruh pembayaran tercatat dan dapat ditelusuri secara historis.



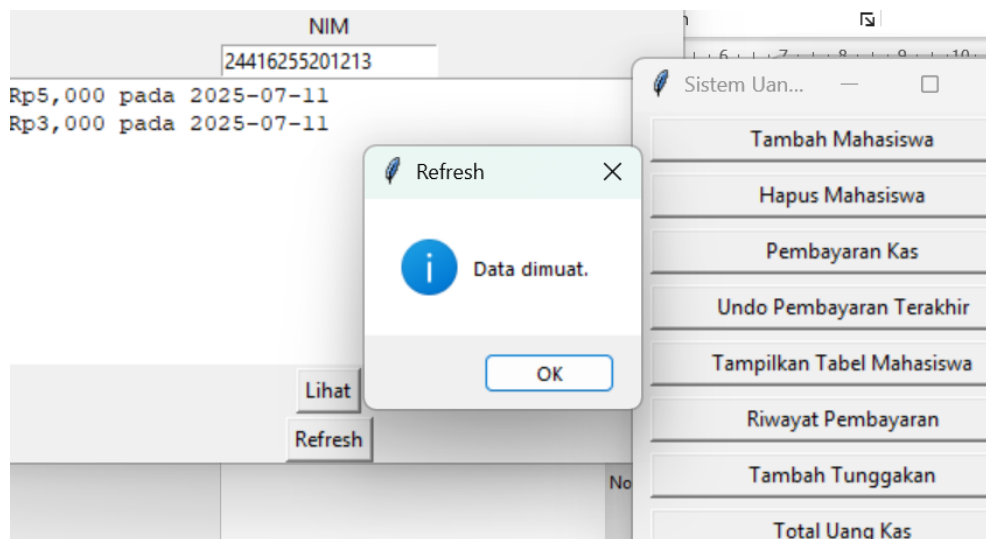
Gambar 3.2.7 Tambah Tunggakan

Gambar ini menunjukkan tampilan untuk menambahkan tunggakan global. Admin dapat menetapkan nominal tunggakan secara kolektif untuk seluruh mahasiswa. Tunggakan ini dicatat dalam log aktivitas dan digunakan sebagai acuan perhitungan sisa pembayaran masing-masing mahasiswa. Ini menunjukkan sistem dapat memperbarui tagihan secara menyeluruh.



Gambar 3.2.8 Menampilkan Total Uang Kas

Gambar ini memperlihatkan dialog informasi yang menampilkan total uang kas masuk dari seluruh mahasiswa. Sistem membaca seluruh data dari kas_mahasiswa.csv, menjumlahkan nominal pembayaran, dan menampilkannya secara otomatis. Fitur ini penting untuk memberikan gambaran total kas terkumpul tanpa harus membuka file manual.



Gambar 3.2.9 Fitur Refresh

Tampilan ini memperlihatkan fitur Refresh yang digunakan untuk memuat ulang data mahasiswa dan tunggakan dari CSV. Ini penting ketika terjadi perubahan data dari luar program, atau setelah dilakukan tindakan tambah/hapus data. Fitur ini memastikan bahwa data yang ditampilkan selalu terkini.


```
data > data_mahasiswa.csv > data
1  NIM,Nama,Kelas
34  24416255201124,SEPTIAN EKA PUTRA,IF24A
35  24416255201215,SUCI NUR SAFFRILIA DEWI,IF24A
36  24416255201205,TIARA PUTRI AULIA,IF24A
37  24416255201219,VHAREL GHERALDY. S,IF24A
38  24416255201234,VIOLA NASYABILLA LUTFIANI,IF24A
39  24416255201216,ZAIDI DZAKWAN,IF24A
40  24416255201102,MUHAMAD BAGAS NUR AL FARID,IF24C
41
```

Gambar 3.2.10 Hasil Perubahan data_mahasiswa.csv

```
data > kas_mahasiswa.csv > data
1  NIM,Nama,Kelas,Jumlah,Tanggal
2  24416255201213,AAN RIPANDI,IF24A,5000,2025-07-11
3  24416255201203,FAHRY RASYIDINANTA PUTRA,IF24A,5000,2025-07-11
4  24416255201213,AAN RIPANDI,IF24A,3000,2025-07-11
5
```

Gambar 3.2.10 Hasil Perubahan kas_mahasiswa.csv

```
data > log.csv > data
1  Waktu,Username,Aksi
2  2025-07-11 21:57:52,azzami,Login berhasil
3  2025-07-11 21:58:41,azzami,Pembayaran Rp5000 oleh 24416255201213
4  2025-07-11 21:59:11,azzami,Pembayaran Rp5000 oleh 24416255201237
5  2025-07-11 22:02:35,azzami,Login berhasil
6  2025-07-11 22:02:48,azzami,Tunggakan global Rp10000
7  2025-07-11 22:03:40,azzami,Tambah mahasiswa: 24416255201102 - MUHAMAD BAGAS NUR AL FARID
8  2025-07-11 22:03:58,admin,Refresh data riwayat
9  2025-07-11 22:04:14,admin,Refresh data riwayat
10 2025-07-11 22:25:30,azzami,Login berhasil
11 2025-07-11 22:32:39,azzami,Tambah mahasiswa: 24416255201084 - JESEN LOUIS AGUSTIAN
12 2025-07-11 22:36:18,azzami,Hapus mahasiswa: 24416255201084 - JESEN LOUIS AGUSTIAN
13
```

Gambar 3.2.10 Hasil Perubahan CSV log.csv

Gambar ini menunjukkan isi dari file data_mahasiswa.csv, kas_mahasiswa.csv, dan log.csv setelah penggunaan aplikasi. Isi file membuktikan bahwa sistem telah mencatat data secara permanen. Format CSV memudahkan integrasi ke aplikasi lain, dan menjadi bukti bahwa sistem telah menyimpan seluruh transaksi secara akurat dan terstruktur.

BAB IV

KESIMPULAN

Aplikasi Sistem Manajemen Uang Kas Mahasiswa yang telah dibangun berhasil memenuhi kebutuhan dasar pencatatan uang kas secara digital, mulai dari pencatatan data mahasiswa, transaksi pembayaran, hingga pelaporan tunggakan dan log aktivitas admin. Aplikasi ini menggunakan bahasa pemrograman Python dan menyimpan data secara permanen menggunakan file CSV, serta memanfaatkan struktur data seperti HashMap dan Stack untuk mengelola data dan fitur Undo Pembayaran. Dengan adanya antarmuka GUI, aplikasi menjadi lebih ramah bagi pengguna umum seperti bendahara kelas.

Penggunaan struktur data yang tepat telah membantu menyederhanakan proses pencarian, penyimpanan, dan pengelolaan data mahasiswa. Selain itu, penyimpanan berbasis file CSV memudahkan pengguna untuk mengakses atau memindahkan data secara fleksibel tanpa membutuhkan database kompleks.

DAFTAR PUSTAKA

Ramadhani, R. (2022). *Struktur Data dan Algoritma dengan Python*. Yogyakarta: Deepublish.

Syahputra, A. (2021). *Pemrograman Python untuk Pemula*. Jakarta: Elex Media Komputindo.

Python Software Foundation. (2024). *Python 3.12 Documentation*. Diakses dari:

<https://docs.python.org/3/>

TkDocs. (2024). *Tkinter Tutorial – Modern GUI Programming in Python*. Diakses dari:

<https://tkdocs.com/>

GeeksforGeeks. (2024). *Python CSV Module*. Diakses dari:

<https://www.geeksforgeeks.org/reading-and-writing-csv-files-in-python/>

Real Python. (2024). *Python Dictionaries (HashMap)*. Diakses dari:

<https://realpython.com/python-dicts/>

W3Schools. (2024). *Python File Handling*. Diakses dari:

https://www.w3schools.com/python/python_file_handling.asp

Stack Overflow. (2023). *Various solutions and best practices for Python GUI and CSV file management*. Diakses dari: <https://stackoverflow.com/>