**LAPORAN PROYEK AKHIR**

**DASAR DASAR PEMROGRAMAN**

**SISTEM PENGELOLAAN KELAS ONLINE**



**Disusun Oleh:**

**KELOMPOK 8**

|  |  |
| --- | --- |
| Zefri Al Rizqullah | 2509116084 |
| Muhammad Aqia Yudha Yulian Putra | 2509116105 |
| Rifaa Zainul Arifin | 2509116092 |

**Asisten Laboratorium:**

|  |  |
| --- | --- |
| **Taufik Ramadhani** | **Dwi Pebriyanto Pradana** |
| **2409116001** | **2409116012** |

**PROGRAM STUDI SISTEM INFORMASI**

**FAKULTAS TEKNIK**

**UNIVERSITAS MULAWARMAN**

**2025**

# KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa atas berkat dan rahmat-Nya, sehingga kami dapat menyelesaikan laporan proyek akhir praktikum Dasar-Dasar Pemrograman ini dengan baik. Laporan dengan judul "Sistem Pengelolaan Kelas Online" ini kami susun sebagai bentuk tugas dari proyek akhir yang telah kami kerjakan. Meskipun awalnya kami mengalami beberapa kesulitan dalam memahami konsep pemrograman dan pembagian tugas, namun dengan kerjasama tim yang baik dan bimbingan dari asisten laboratorium, akhirnya kami bisa menyelesaikan proyek ini.

Dalam laporan ini, kami berusaha menjelaskan secara detail proses pembuatan aplikasi, mulai dari penulisan kode program, alur kerja sistem melalui flowchart, hingga penjelasan fungsi-fungsi yang ada di dalamnya. Kami berharap penjelasan yang kami tulis bisa dipahami dengan baik oleh pembaca. Kami juga menyadari bahwa laporan ini masih jauh dari sempurna dan mungkin masih ada kekurangan di beberapa bagian. Oleh karena itu, kami sangat terbuka menerima kritik dan saran yang membangun untuk perbaikan kedepannya.

Akhir kata, kami mengucapkan terima kasih kepada asisten laboratorium yang telah sabar membimbing selama praktikum, serta teman-teman yang telah membantu dalam penyelesaian proyek ini.

Oktober, 2025

Tim Penyusun

# DAFTAR ISI

[KATA PENGANTAR i](#_Toc211476330)

[DAFTAR ISI ii](#_Toc211476331)

[DAFTAR GAMBAR iii](#_Toc211476332)

[DAFTAR TABEL vi](#_Toc211476333)

[DAFTAR LAMPIRAN vii](#_Toc211476334)

[BAB I PENDAHULUAN 1](#_Toc211476335)

[1.1 Deskripsi Masalah 1](#_Toc211476336)

[1.2 Rumusan Masalah 2](#_Toc211476337)

[1.3 Batasan Masalah 3](#_Toc211476338)

[1.4 Tujuan 3](#_Toc211476339)

[1.5 Manfaat 4](#_Toc211476340)

[BAB II PERANCANGAN 5](#_Toc211476341)

[2.1 Analisis Program 5](#_Toc211476342)

[2.2 Flowchart ( SESUAIKAN SAMA FLOW KALIAN) 6](#_Toc211476343)

[BAB III HASIL DAN PEMBAHASAN 7](#_Toc211476344)

[3.1 Implementasi Program 7](#_Toc211476345)

[3.2 Alur Program 7](#_Toc211476346)

[3.3 Source Code (KASIH KOTAK SEPERTI CONTOH 8](#_Toc211476347)

[BAB IV PENUTUP 10](#_Toc211476348)

[4.1 Kesimpulan 10](#_Toc211476349)

[4.2 Saran 10](#_Toc211476350)

[DAFTAR PUSTAKA 11](#_Toc211476351)

[LAMPIRAN 12](#_Toc211476352)

# DAFTAR GAMBAR

**Gambar 2.1** Flowchart 1.1 ........................................................................... 14

**Gambar 2.2** Flowchart 1.2 ........................................................................... 15

**Gambar 2.3** Flowchart 1.3 ........................................................................... 16

**Gambar 2.4** Flowchart 1.4 ........................................................................... 17

**Gambar 2.5** Flowchart 1.5 ........................................................................... 18

**Gambar 2.6** Flowchart 1.6 ........................................................................... 19

**Gambar 3.1** Implementasi: Python .............................................................. 20

**Gambar 3.2** Implementasi: Dictionary akun admin dan user ...................... 21

**Gambar 3.3** Implementasi: Dictionary data barang dan sewa untuk export 21

**Gambar 3.4** Implementasi: Dictionary kasir pemesanan ............................. 21

**Gambar 3.5** Implementasi: Dictionary kasir penyewaan ............................ 21

**Gambar 3.6** Implementasi: Dictionary data pelanggan sewa ...................... 21

**Gambar 3.7** Implementasi: While True loop untuk mengulang proses login

....................................................................................................................... 22

**Gambar 3.8** Implementasi: While True loop untuk mengulang input ......... 22

**Gambar 3.9** Implementasi: While dengan suatu kondisi ............................. 23

**Gambar 3.10** Implementasi: Library PrettyTable ........................................ 24

**Gambar 3.11** Implementasi: Import modul pandas ..................................... 24

**Gambar 3.12** Implementasi: Penggunaan pandas ........................................ 25

**Gambar 3.13** Implementasi: Create data 1 .................................................. 25

**Gambar 3.14** Implementasi: Create data 2 .................................................. 26

**Gambar 3.15** Implementasi: Read data 1 .................................................... 26

**Gambar 3.16** Implementasi: Read data 2 .................................................... 26

**Gambar 3.17** Implementasi: Update data 1 ................................................. 27

**Gambar 3.18** Implementasi: Update data 2 ................................................. 27

**Gambar 3.19** Implementasi: Delete data ..................................................... 28

**Gambar 3.20** Implementasi: Data akun admin dan user .............................. 28

**Gambar 3.21** Implementasi: Pilihan menu login admin dan user ............... 28

**Gambar 3.22** Implementasi: Proses login admin ......................................... 29

**Gambar 3.23** Implementasi: Proses login user dan registrasi user .............. 29

**Gambar 3.24** Implementasi: pwinput .......................................................... 29

**Gambar 3.25** Implementasi: pwinput pada input password admin ............. 29

**Gambar 3.26** Implementasi: pwinput pada input password user ................. 30

**Gambar 3.27** Implementasi: Export data ke Excel/CSV data barang-sewa 30

**Gambar 3.28** Implementasi: Export data ke Excel/CSV data pemesanan

barang ............................................................................................................ 30

**Gambar 3.29** Implementasi: Export data ke Excel/CSV data penyewaan .. 30

**Gambar 3.30** Alur Program: Menu login 1 ................................................. 31

**Gambar 3.31** Alur Program: Menu login 2 ................................................. 31

**Gambar 3.32** Alur Program: Menu login 3 ................................................. 32

**Gambar 3.33** Alur Program: Menu login 4 ................................................. 32

**Gambar 3.34** Alur Program: Menu Admin .................................................. 33

**Gambar 3.35** Alur Program: Menu User ..................................................... 34

**Gambar 3.36** Tampilan Program: Tampilan Awal ...................................... 53

**Gambar 3.37** Tampilan Program: Menu Login ........................................... 54

**Gambar 3.38** Tampilan Program: Login Admin ......................................... 54

**Gambar 3.39** Tampilan Program: Login User ............................................. 54

**Gambar 3.40** Tampilan Program: Registrasi User ...................................... 55

**Gambar 3.41** Tampilan Program: Kontak Admin ....................................... 55

**Gambar 3.42** Tampilan Program: Menu Admin .......................................... 55

**Gambar 3.43** Tampilan Program: Menu edit data barang ........................... 56

**Gambar 3.44** Tampilan Program: Tambah data barang .............................. 57

**Gambar 3.45** Tampilan Program: Ubah data barang ................................... 58

**Gambar 3.46** Tampilan Program: Hapus data barang ................................. 59

**Gambar 3.47** Tampilan Program: Tampilkan data barang .......................... 60

**Gambar 3.48** Tampilan Program: Kembali ke menu admin ........................ 60

**Gambar 3.49** Tampilan Program: Menu edit data sewa .............................. 61

**Gambar 3.50** Tampilan Program: Tambah data penyewaan ....................... 62

**Gambar 3.51** Tampilan Program: Ubah data penyewaan ............................ 63

**Gambar 3.52** Tampilan Program: Hapus data sewa .................................... 64

**Gambar 3.53** Tampilan Program: Tampilkan data sewa ............................. 64

**Gambar 3.54** Tampilan Program: Export data ............................................ 64

**Gambar 3.55** Tampilan Program: Hasil export data .................................... 65

**Gambar 3.56** Tampilan Program: Logout admin ........................................ 65

**Gambar 3.57** Tampilan Program: Menu user .............................................. 65

**Gambar 3.58** Tampilan Program: List harga barang ................................... 66

**Gambar 3.59** Tampilan Program: Pemesanan barang ................................. 67

**Gambar 3.60** Tampilan Program: Export data pemesanan .......................... 67

**Gambar 3.61** Tampilan Program: List harga sewa ...................................... 68

**Gambar 3.62** Tampilan Program: Penyewaan barang ................................. 69

**Gambar 3.63** Tampilan Program: Export data penyewaan barang .............. 69

**Gambar 3.64** Tampilan Program: Logout user ............................................ 70

# DAFTAR TABEL

[**Tabel 3. 1** Source Code 3](#_Toc210894655)

# DAFTAR LAMPIRAN

[**Lampiran 1 :** Tabel Kontribusi 6](#_Toc210894679)

# BAB I PENDAHULUAN

## Deskripsi Masalah

Perkembangan teknologi digital di masa kini memberikan pengaruh besar terhadap berbagai bidang kehidupan, termasuk pada dunia pendidikan. Proses pembelajaran tidak lagi terbatas pada ruang kelas dan tatap muka secara langsung, melainkan dapat dilakukan secara daring dengan memanfaatkan berbagai platform digital. Melalui sistem pembelajaran online, setiap individu memiliki kesempatan untuk belajar di mana saja dan kapan saja sesuai dengan minat dan kebutuhan mereka. Hal ini membuat metode belajar menjadi lebih fleksibel, dan mudah dijangkau oleh berbagai kalangan.

Namun, di balik kemudahan tersebut, masih banyak sistem pembelajaran daring yang belum memiliki pengelolaan data yang teratur dan efisien. Beberapa platform masih mengandalkan proses manual dalam pendaftaran, pembelian kelas, maupun pengelolaan data pengguna. Hal ini sering menimbulkan berbagai kendala seperti kesalahan pencatatan, keterlambatan informasi, serta kurangnya transparansi dalam proses transaksi. Selain itu, banyak sistem belum menyediakan fitur saldo atau top up yang dapat memudahkan pengguna dalam melakukan pembelian kelas secara digital.

Berangkat dari kondisi tersebut, kelompok kami mengembangkan sebuah program bernama “Sistem Pengelolaan Kelas Online”. Program ini dirancang untuk mempermudah proses pembelajaran daring sekaligus mengelola transaksi pembelian kelas secara terintegrasi. Sistem ini memiliki dua peran utama, yaitu Teacher dan User. Teacher dapat membuat, mengubah, menghapus, dan menampilkan data kelas (CRUD) dengan harga yang telah ditentukan, serta melihat daftar pengguna yang telah membeli kelas tersebut. Sementara itu, User dapat melakukan registrasi dengan saldo awal sebesar 0, melakukan login, melakukan top up saldo, dan membeli kelas sesuai keinginan mereka.

Dengan adanya sistem ini, diharapkan kegiatan pembelajaran daring dapat berlangsung lebih teratur, efisien, dan transparan. Sistem ini tidak hanya mempermudah pengguna dalam mengakses dan membeli kelas, tetapi juga membantu Teacher dalam mengelola data dan transaksi dengan lebih praktis. Selain itu, kehadiran fitur saldo dan top up memberikan nilai tambah bagi pengguna dalam melakukan transaksi digital secara aman dan fleksibel, sehingga menciptakan pengalaman belajar yang lebih modern dan menyenangkan.

## Rumusan Masalah

Berdasarkan deskripsi masalah yang telah dipaparkan pada bagian sebelumnya, adapun rumusan masalah yang terdapat dalam program “Sistem Pengelolaan Kelas Online” ini adalah sebagai berikut:

1. Bagaimana cara membangun sistem registrasi dan login yang memungkinkan pengguna untuk memiliki akun dengan saldo awal 0 serta dapat melakukan proses top up secara mandiri?
2. Bagaimana merancang sistem yang mampu menampilkan daftar kelas beserta harga secara rapi dan mudah dipahami oleh pengguna?
3. Bagaimana mengimplementasikan fitur CRUD (Create, Read, Update, Delete) bagi Teacher untuk mengelola data kelas beserta harga dan informasi pembeli?
4. Bagaimana cara sistem memproses transaksi pembelian kelas dengan menggunakan saldo pengguna secara otomatis dan aman?
5. Bagaimana sistem dapat membantu Teacher dalam memantau data pengguna, transaksi, dan pembelian kelas agar proses pengelolaan berjalan lebih efisien?

## Batasan Masalah

Agar sistem ini dapat dikembangkan dengan efektif dan sesuai dengan waktu pengerjaan tugas, maka kelompok kami memberikan beberapa batasan, antara lain:

1. Sistem dijalankan melalui terminal (console) menggunakan bahasa pemrograman Python tanpa tampilan grafis (GUI).
2. Sistem menggunakan file CSV sebagai media penyimpanan data utama untuk pengguna, kelas, dan pengajuan. Belum menggunakan sistem basis data seperti MySQL atau PostgreSQL.
3. Terdapat dua peran utama dalam sistem, yaitu:
4. **Teacher**, yang memiliki hak untuk membuat, menampilkan, memperbarui, dan menghapus data kelas, menentukan harga kelas, juga melihat daftar pengguna yang sudah membeli kelas.
5. **User**, yang dapat melakukan registrasi dan login, melakukan top up saldo, melihat daftar kelas, serta membeli kelas sesuai harga yang sudah ditentukan.
6. Fitur yang diimplementasikan meliputi registrasi, login, manajemen kelas (CRUD), top up saldo, dan pembelian kelas.

## Tujuan

Tujuan utama dari program kelompok ini adalah untuk menciptakan sistem sederhana yang dapat membantu proses pengelolaan kelas secara digital. Secara lebih rinci, tujuan pengembangan ”Sistem Pengelolaan Kelas Online” ini adalah:

1. Meningkatkan efisiensi dalam pengelolaan data dengan menyediakan fitur CRUD untuk data kelas yang dapat diakses dan dikelola oleh Teacher melalui terminal
2. Memudahkan pengguna dalam melakukan registrasi, login, top up saldo, dan pembelian kelas secara mandiri dengan sistem yang terintegrasi.
3. Menerapkan sistem autentikasi dan otorisasi yang membedakan hak akses antara Teacher dan User sesuai dengan peran dan fungsinya masing-masing
4. Menampilkan data pengguna, kelas, dan transaksi dengan tampilan yang terstruktur, rapi, dan mudah dipahami menggunakan PrettyTable agar proses pembacaan data di terminal menjadi lebih mudah
5. Memberikan sarana pembelajaran bagi anggota kelompok dalam memahami konsep pengelolaan data berbasis file CSV, penerapan logika transaksi digital sederhana, serta implementasi program yang efisien dalam Bahasa python

## Manfaat

Berdasarkan hal-hal yang telah dijelaskan sebelumnya, adapun manfaat dalam pembuatan dan penyusunan program “Sistem Pengelolaan Kelas Online” ini adalah:

1. Program ini dapat membantu menyediakan sarana pembelajaran daring yang memungkinkan User untuk membeli kelas sesuai kebutuhan mereka, serta membantu Teacher dalam mengelola data kelas dan transaksi dengan lebih mudah
2. Dengan adanya program ini, proses pembelajaran dan pengelolaan kelas dapat dilakukan secara lebih fleksibel, efisien, dan tidak terbatas oleh waktu maupun tempat
3. Program ini dirancang agar mudah diakses dan dijalankan melalui terminal, dengan tampilan data yang tersusun rapi menggunakan PrettyTable, sehingga memudahkan pengguna dalam memahami informasi yang ditampilkan

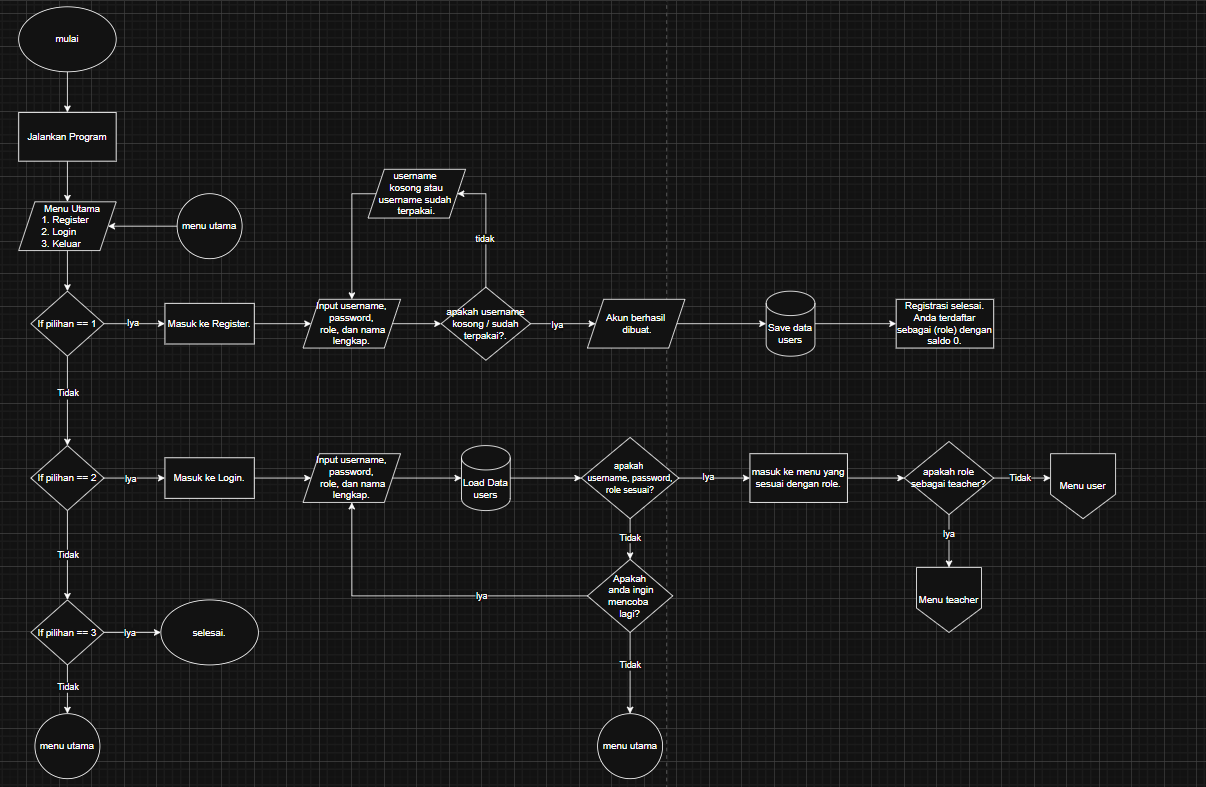
# BAB II PERANCANGAN

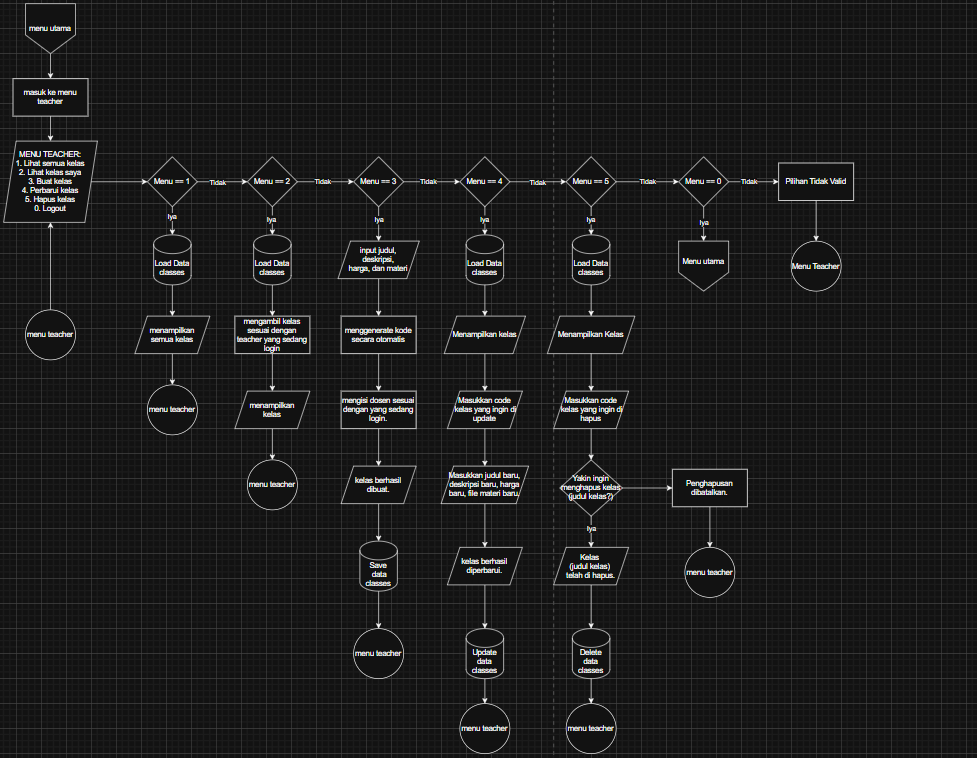
## 2.1 Analisis Program

Program “Sistem Pengelolaan Kelas Online” ini dikembangkan untuk mempermudah proses pembelajaran dan pengelolaan kelas secara daring melalui sistem transaksi digital sederhana. Sistem ini dirancang agar kegiatan belajar dapat dilakukan dengan fleksibel tanpa terikat oleh waktu maupun tempat, serta memberikan kemudahan bagi pengguna dalam mengakses kelas yang diinginkan. Dalam program ini terdapat dua peran utama, yaitu Teacher dan User, di mana Teacher memiliki hak untuk mengelola data kelas melalui fitur *Create, Read, Update,* dan *Delete (CRUD)*, menentukan harga kelas, serta melihat daftar pengguna yang telah membeli kelas, sedangkan user dapat melakukan registrasi dengan saldo awal sebesar 0, melakukan top up saldo, melihat daftar kelas yang tersedia, dan membeli kelas sesuai dengan saldo yang dimilikinya.

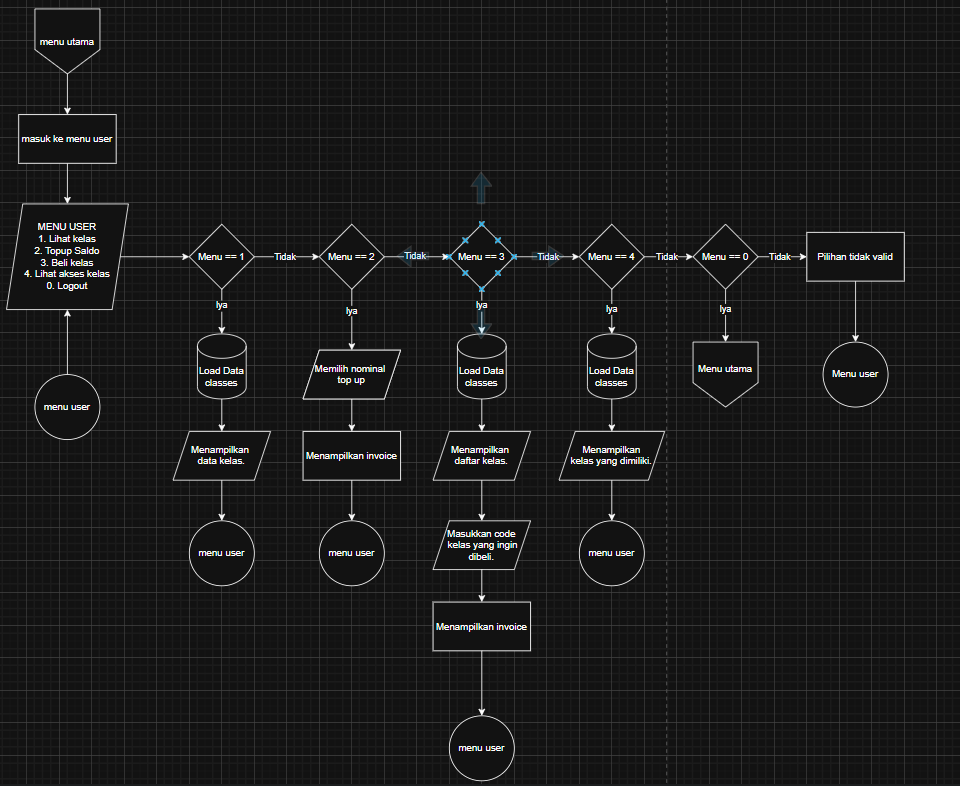
Sistem ini juga dilengkapi dengan fitur login dan logout untuk menjaga keamanan akses pengguna, serta menggunakan file CSV sebagai media penyimpanan utama. Tampilan program dibuat berbasis teks di terminal dengan bantuan modul PrettyTable, sehingga data dapat ditampilkan secara rapi dan mudah dibaca. Secara keseluruhan, program ini membantu mengoptimalkan pengelolaan kelas daring dengan sistem yang sederhana, efisien, dan terorganisir sehingga memudahkan Teacher maupun User dalam menjalankan aktivitasnya.

## 2.2 Flowchart

**Gambar 2.1** Flowchart Menu utama 1.1



**Gambar 2.2** Flowchart Menu Teacher 1.2

**Gambar 2.3** Flowchart Menu User 1.3

# BAB III HASIL DAN PEMBAHASAN

## 3.1 Implementasi Program

Program Sistem Pengelolaan Kelas Online diimplementasikan menggunakan bahasa pemrograman Python dan dijalankan pada terminal (console). Sistem ini bertujuan untuk menyediakan sarana pengelolaan kelas digital secara sederhana dan efisien, dengan memanfaatkan file CSV sebagai media penyimpanan utama untuk data pengguna, kelas, dan transaksi. Pendekatan ini memungkinkan program dijalankan tanpa memerlukan basis data eksternal, namun tetap mempertahankan struktur data yang terorganisir.

Dalam implementasinya, sistem ini menggunakan beberapa pustaka Python, yaitu:

1. **CSV** untuk membaca dan menulis data ke dalam file berformat Comma Separated Values.
2. ***Random*** dan ***string*** untuk menghasilkan kode unik bagi setiap kelas yang dibuat oleh teacher.
3. ***Datetime*** untuk mencatat waktu transaksi seperti pembelian dan top-up saldo.
4. ***Pwinput*** untuk menyembunyikan karakter sandi saat pengguna melakukan login atau registrasi, guna meningkatkan keamanan.
5. **PrettyTable** untuk menampilkan data dalam bentuk tabel yang rapi pada terminal.

Struktur program terdiri dari beberapa bagian utama, yaitu:

1. Utility Functions, yang berisi fungsi pendukung seperti read\_csv, write\_csv, dan append\_csv untuk mengelola penyimpanan data.
2. **Autentikasi (Auth),** mencakup proses registrasi dan login pengguna dengan validasi sederhana berbasis peran (student atau teacher).
3. CRUD kelas, yaitu fitur bagi teacher untuk membuat, memperbarui, menampilkan, dan menghapus data kelas yang mereka kelola.
4. Transaksi Digital, di mana student dapat melakukan top up saldo dan membeli kelas sesuai harga yang ditentukan.
5. Menu Utama dan Sesi Pengguna, yang menjadi antarmuka logika program berdasarkan peran pengguna yang sedang login.

Implementasi sistem ini memungkinkan dua jenis peran untuk berinteraksi secara dinamis:

1. Teacher, yang berfungsi sebagai pengelola kelas dan penerima saldo dari hasil transaksi pembelian kelas.
2. Student, yang dapat membeli kelas, melakukan top up, dan mengakses daftar kelas yang dimilikinya.

Dengan rancangan tersebut, sistem ini dapat diimplementasikan dalam kehidupan nyata untuk membantu proses pembelajaran daring yang terstruktur. Misalnya, dalam lingkungan pendidikan informal seperti kursus online atau pelatihan keterampilan digital. Penggunaan CSV membuat sistem mudah dijalankan di berbagai perangkat tanpa perlu server atau koneksi internet, sehingga cocok digunakan pada tahap awal pengembangan sistem e-learning sederhana.

## 3.2 Alur Program

Alur program Sistem Pengelolaan Kelas Online dimulai dari saat pengguna menjalankan program di terminal Python. Program ini berbasis menu interaktif yang memungkinkan pengguna memilih berbagai opsi untuk mengakses fitur sesuai perannya, yaitu **student** atau **teacher**. Setiap peran memiliki alur dan hak akses yang berbeda, namun keduanya saling terhubung melalui proses registrasi, login, dan pengelolaan data.

1. Tampilan Menu Utama

Saat program dijalankan, sistem akan menampilkan menu utama dengan tiga pilihan, yaitu:

1. Register
2. Login
3. Keluar

Pengguna dapat memilih **1** untuk melakukan registrasi akun baru, **2** untuk login, atau **0** untuk keluar dari sistem.

(Gambar 3.1 Menu Utama)

1. Proses Registrasi

Jika pengguna memilih opsi **Register**, sistem akan meminta input berupa:

1. Username
2. Password
3. Nama lengkap
4. Role (student / teacher)

Setelah pengguna mengisi data dengan benar, sistem menyimpannya dalam file users.csv dan memberikan saldo awal sebesar 0. Proses ini menjamin setiap pengguna memiliki identitas unik sebelum menggunakan sistem.

(Gambar 3.2 Registrasi Akun Baru)

1. Proses Login

Apabila pengguna memilih Login, maka sistem akan meminta username dan password untuk diverifikasi berdasarkan data yang ada di users.csv.

Jika data sesuai, sistem menampilkan pesan selamat datang dan mengarahkan pengguna ke menu utama sesuai dengan perannya.

(Gambar 3.3 Proses Login Berhasil)

(Gambar 3.4 Proses Login Gagal)

1. Menu Student

Bagi pengguna dengan peran student, sistem menampilkan pilihan sebagai berikut:

* 1. Lihat daftar kelas
  2. Top up saldo
  3. Beli kelas
  4. Lihat akses kelas
  5. Logout

1. Lihat kelas

Menampilkan seluruh kelas yang tersedia di classes.csv dalam bentuk tabel rapi menggunakan modul PrettyTable.

(Gambar 3.5 Daftar Kelas)

1. Top up saldo

Pengguna dapat menambah saldo dengan nominal tertentu (Rp50.000, Rp100.000, Rp250.000, Rp500.000, dan Rp1.000.000). Setelah top up berhasil, sistem menampilkan invoice transaksi yang memuat nama, jumlah top up, waktu transaksi, dan saldo terkini.

(Gambar 3.6 Proses Top Up dan Invoice)

1. Beli kelas

Student dapat membeli kelas dengan memasukkan kode kelas yang diinginkan. Sistem akan memverifikasi saldo pengguna dan mencatat transaksi pembelian ke purchases.csv Jika saldo cukup, saldo otomatis berkurang dan teacher yang bersangkutan menerima tambahan saldo sesuai harga kelas.

(Gambar 3.7 Proses Pembelian Kelas)

(Gambar 3.8 Invoice Pembelian Kelas)

1. Lihat akses Kelas

Menampilkan daftar kelas yang telah dibeli oleh student, termasuk nama dosen, deskripsi kelas, dan nama file materi dalam format PDF.

(Gambar 3.9 Daftar Kelas Dimiliki Student)

Jika student memilih **Logout**, maka sistem akan kembali ke menu utama.

1. Menu Teacher

Untuk pengguna dengan peran teacher, sistem menampilkan menu berbeda, yaitu:

1. Lihat semua kelas
2. Lihat kelas saya
3. Buat kelas
4. Perbarui kelas
5. Hapus kelas
6. Logout
7. Lihat semua kelas

Menampilkan semua kelas beserta detail seperti kode, judul, deskripsi, harga, dan nama dosen.

(Gambar 3.10 Daftar Seluruh Kelas)

1. Lihat kelas saya

Menampilkan hanya kelas yang dibuat oleh teacher tersebut.

(Gambar 3.11 Daftar Kelas Milik Teacher)

1. Buat kelas

Teacher dapat membuat kelas baru dengan mengisi form berisi judul, deskripsi, harga, dan nama file materi PDF. Sistem akan menghasilkan kode kelas unik menggunakan fungsi generate\_code() yang terdiri dari kombinasi huruf kapital dan angka acak.

(Gambar 3.12 Pembuatan Kelas Baru)

1. Perbarui kelas

Teacher dapat memperbarui data kelas berdasarkan kode kelas yang mereka miliki. Program akan menampilkan data lama dan memberi kesempatan untuk mengubah sebagian atau seluruhnya.

(Gambar 3.13 Proses Update Data Kelas)

1. Hapus kelas

Teacher dapat menghapus kelas yang sudah dibuat dengan memasukkan kode kelas dan melakukan konfirmasi penghapusan.

(Gambar 3.14 Proses Penghapusan Kelas)

Jika teacher memilih **Logout**, sistem akan kembali ke menu utama.

1. Transaksi dan Pembaruan Data
2. Proses Keluar Program

s

Dengan demikian, alur program ini mencerminkan bagaimana sistem bekerja secara terstruktur dan terintegrasi. Mulai dari registrasi, login, pengelolaan data, hingga transaksi digital sederhana — seluruh proses dilakukan secara otomatis menggunakan bahasa Python dan modul PrettyTable untuk menampilkan hasil keluaran dengan format tabel yang mudah dibaca di terminal.

## 3.3 Source Code

Berikut merupakan hasil source code Sistem Pengelolaan Kelas Online

**Tabel 3.1** Source Code Sistem Pengelolaan Kelas Online

|  |
| --- |
| import csv, random, string  from datetime import datetime  from pwinput import pwinput  from prettytable import PrettyTable  *# BAGIAN UTILITY*  def read\_csv(filename):  try:  with open(filename, mode='r', newline='', encoding='utf-8') as f:  return list(csv.DictReader(f))  except FileNotFoundError:  return []  def write\_csv(filename, rows, headers):  with open(filename, mode='w', newline='', encoding='utf-8') as f:  writer = csv.DictWriter(f, fieldnames=headers)  writer.writeheader()  writer.writerows(rows)  def append\_csv(filename, row, headers):  try:  with open(filename, mode='a', newline='', encoding='utf-8') as f:  writer = csv.DictWriter(f, fieldnames=headers)  if f.tell() == 0:  writer.writeheader()  writer.writerow(row)  except FileNotFoundError:  with open(filename, mode='w', newline='', encoding='utf-8') as f:  writer = csv.DictWriter(f, fieldnames=headers)  writer.writeheader()  writer.writerow(row)  def generate\_code(n=5):  return ''.join(random.choice(string.ascii\_uppercase + string.digits) for \_ in range(n))  def show\_table(title, headers, rows):  table = PrettyTable()  table.field\_names = headers  for r in rows:  table.add\_row([r.get(h, "") for h in headers])  print("\n" + title)  print(table)  *# BAGIAN AUTH*  def register():  users = read\_csv("users.csv")  usernames = {u['username'] for u in users}  print("\n=== REGISTER ===")  while True:  username = input("Username: ").strip()  if not username:  print("Username tidak boleh kosong.")  elif username in usernames:  print("Username sudah dipakai.")  else:  break  password = pwinput("Password: ")  nama = input("Nama lengkap: ")  role = ""  while role not in ("student", "teacher"):  role = input("Role (student / teacher): ").strip().lower()  new = {  'id': str(len(users) + 1),  'username': username,  'password': password,  'nama': nama,  'role': role,  'saldo': '0'  }  append\_csv("users.csv", new, ['id','username','password','nama','role','saldo'])  print(f"Registrasi selesai. Anda terdaftar sebagai {role} dengan saldo 0.")  *# BAGIAN CRUD*  def create\_class(user):  print("\n=== BUAT KELAS ===")  judul = input("Judul kelas: ")  deskripsi = input("Deskripsi: ")  harga = input("Harga: ")  materi = input("Nama file materi PDF: ")  kode\_kelas = generate\_code()  new = {  'kode': kode\_kelas,  'judul': judul,  'dosen': user['nama'],  'deskripsi': deskripsi,  'harga': harga,  'materi\_pdf': materi  }  append\_csv("classes.csv", new, ['kode','judul','dosen','deskripsi','harga','materi\_pdf'])  print(f"Kelas berhasil dibuat dengan kode: {kode\_kelas}")  def update\_class(current\_user):  classes = read\_csv("classes.csv")  my\_classes = [c for c in classes if c['dosen'] == current\_user['nama']]  if not my\_classes:  print("Anda belum membuat kelas untuk diubah.")  return  show\_table("Kelas Anda", ['kode','judul','deskripsi','harga','materi\_pdf'], my\_classes)  kode = input("Masukkan kode kelas yang ingin diupdate: ").strip()  kelas = next((c for c in classes if c['kode'] == kode and c['dosen'] == current\_user['nama']), None)  if not kelas:  print("Kelas tidak ditemukan atau bukan milik Anda.")  return  print("Kosongkan jika tidak ingin mengubah data.")  new\_title = input(f"Judul baru ({kelas['judul']}): ").strip() or kelas['judul']  new\_desc = input(f"Deskripsi baru ({kelas['deskripsi']}): ").strip() or kelas['deskripsi']  new\_price = input(f"Harga baru ({kelas['harga']}): ").strip() or kelas['harga']  new\_pdf = input(f"File materi baru ({kelas['materi\_pdf']}): ").strip() or kelas['materi\_pdf']  for c in classes:  if c['kode'] == kode:  c['judul'] = new\_title  c['deskripsi'] = new\_desc  c['harga'] = new\_price  c['materi\_pdf'] = new\_pdf  write\_csv("classes.csv", classes, ['kode','judul','dosen','deskripsi','harga','materi\_pdf'])  print("Kelas berhasil diperbarui.")  def delete\_class(current\_user):  classes = read\_csv("classes.csv")  my\_classes = [c for c in classes if c['dosen'] == current\_user['nama']]  if not my\_classes:  print("Anda belum membuat kelas untuk dihapus.")  return  show\_table("Kelas Anda", ['kode','judul','deskripsi','harga','materi\_pdf'], my\_classes)  kode = input("Masukkan kode kelas yang ingin dihapus: ").strip()  kelas = next((c for c in classes if c['kode'] == kode and c['dosen'] == current\_user['nama']), None)  if not kelas:  print("Kelas tidak ditemukan atau bukan milik Anda.")  return  konfirmasi = input(f"Yakin ingin menghapus kelas '{kelas['judul']}'? (y/n): ").strip().lower()  if konfirmasi != 'y':  print("Penghapusan dibatalkan.")  return  classes = [c for c in classes if c['kode'] != kode]  write\_csv("classes.csv", classes, ['kode','judul','dosen','deskripsi','harga','materi\_pdf'])  print(f"Kelas '{kelas['judul']}' telah dihapus.")  def list\_classes(detail=False):  data = read\_csv("classes.csv")  if not data:  print("Belum ada kelas.")  return  headers = ['kode','judul','dosen','deskripsi','harga'] if not detail else ['kode','judul','dosen','deskripsi','harga','materi\_pdf']  show\_table("Daftar Kelas", headers, data)  def my\_classes(current\_user):  classes = read\_csv("classes.csv")  my\_classes = [c for c in classes if c['dosen']==current\_user['nama']]  if not my\_classes:  print("Anda belum membuat kelas.")  return  show\_table("Kelas saya (Teacher)", ['kode','judul','deskripsi','harga','materi\_pdf'], my\_classes)  def tambah\_saldo(user):  users = read\_csv("users.csv")  topup\_opsi = {'1':50000,'2':100000,'3':250000,'4':500000,'5':1000000}  print("\n=== TOP UP ===")  for k,v in topup\_opsi.items():  print(f"{k}. Rp{v:,}")  pilihan = input("Pilih nominal (1-5): ").strip()  if pilihan not in topup\_opsi:  print("Pilihan tidak valid.")  return user  jumlah = topup\_opsi[pilihan]  for u in users:  if u['id'] == user['id']:  u['saldo'] = str(int(u['saldo']) + jumlah)  user['saldo'] = u['saldo']  write\_csv("users.csv", users, ['id','username','password','nama','role','saldo'])  waktu = datetime.now().isoformat(sep=' ', timespec='seconds')    print("\n=== INVOICE TOP-UP ===")  print(f"Nama Pengguna : {user['nama']}")  print(f"Jumlah Top-Up : Rp{jumlah:,}")  print(f"Waktu Transaksi: {waktu}")  print(f"Saldo Sekarang : Rp{int(user['saldo']):,}")  print("=======================")  return user  def buy\_class(user):  list\_classes()  kode = input("Masukkan kode kelas yang ingin dibeli: ").strip()  classes = read\_csv("classes.csv")  kelas = next((c for c in classes if c['kode'] == kode), None)  if not kelas:  print("Kelas tidak ditemukan.")  return  purchases = read\_csv("purchases.csv")  sudah\_beli = any(p['user\_id'] == user['id'] and p['class\_kode'] == kode for p in purchases)  if sudah\_beli:  print("Anda sudah membeli kelas ini sebelumnya.")  return  harga = int(kelas['harga'])  if int(user['saldo']) < harga:  print("Saldo tidak cukup.")  return  user['saldo'] = str(int(user['saldo']) - harga)  users = read\_csv("users.csv")  for u in users:  if u['id'] == user['id']:  u['saldo'] = user['saldo']  for u in users:  if u['nama'] == kelas['dosen'] and u['role'] == 'teacher':  u['saldo'] = str(int(u['saldo']) + harga)  break  write\_csv("users.csv", users, ['id','username','password','nama','role','saldo'])  purchase = {  'id': str(len(purchases) + 1),  'user\_id': user['id'],  'class\_kode': kode,  'timestamp': datetime.now().isoformat(sep=' ', timespec='seconds')  }  append\_csv("purchases.csv", purchase, ['id','user\_id','class\_kode','timestamp'])  print("\n=== INVOICE PEMBELIAN ===")  print(f"Nama: {user['nama']}")  print(f"Kelas: {kelas['judul']}")  print(f"Dosen: {kelas['dosen']}")  print(f"Harga: Rp{harga:,}")  print(f"Waktu: {purchase['timestamp']}")  print(f"Sisa saldo: Rp{int(user['saldo']):,}")  print("==========================")  def access\_class(user):  purchases = read\_csv("purchases.csv")  classes = read\_csv("classes.csv")  owned\_kodes = {p['class\_kode'] for p in purchases if p['user\_id']== user['id']}  owned\_classes = [c for c in classes if c['kode'] in owned\_kodes]  show\_table("Kelas Dimiliki", ['kode','judul','dosen','deskripsi','materi\_pdf'], owned\_classes)  *# BAGIAN MENU*  def main\_menu():  while True:  print("\n=== SISTEM PENGELOLAAN KELAS ONLINE ===")  print("1. Register")  print("2. Login")  print("0. Keluar")  pil = input("Pilih: ")  if pil == '1':  register()  elif pil == '2':  user = login()  if user:  session(user)  elif pil == '0':  print("Keluar dari sistem.")  break  else:  print("Pilihan tidak valid.")  def login():  while True:  print("\n=== LOGIN ===")  username = input("Username: ").strip()  password = pwinput("Password: ")  users = read\_csv("users.csv")  for user in users:  if user['username'] == username and user['password'] == password:  print(f"Selamat datang, {user['nama']} ({user['role']})")  return user  print("Login gagal: username atau password salah.")  pilihan = input("Apakah Anda ingin mencoba lagi? (y/n): ").strip().lower()  if pilihan != 'y':  print("Kembali ke menu utama...")  return None  def session(user):  while True:  print(f"\n=== MENU {user['role'].upper()} ===")  if user["role"] == "student":  print("1. Lihat kelas")  print("2. Top up saldo")  print("3. Beli kelas")  print("4. Lihat akses kelas")  print("0. Logout")  pil = input("Pilih menu: ")  if pil == '1':  list\_classes()  elif pil == '2':  tambah\_saldo(user)  elif pil == '3':  buy\_class(user)  elif pil == '4':  access\_class(user)  elif pil == '0':  break  else:  print("Pilihan Anda tidak tersedia!")  elif user["role"] == "teacher":  print("1. Lihat semua kelas")  print("2. Lihat kelas saya")  print("3. Buat kelas")  print("4. Perbarui kelas")  print("5. Hapus kelas")  print("0. Logout")  pil = input("Pilih menu: ")  if pil == '1':  list\_classes(True)  elif pil == '2':  my\_classes(user)  elif pil == '3':  create\_class(user)  elif pil == '4':  update\_class(user)  elif pil == '5':  delete\_class(user)  elif pil == '0':  break  else:  print("Pilihan Anda tidak tersedia!")  else:  print("role Anda tidak valid! Terdapat kesalahan fatal dalam program. 💁🏻")  break  main\_menu() |

# BAB IV PENUTUP

## 4.1 Kesimpulan

## 4.2 Saran

# DAFTAR PUSTAKA

# LAMPIRAN

|  |  |  |
| --- | --- | --- |
| **Nama** | **Kontribusi** | **Bagian** |
| Zefri Al Rizqullah (2509116084) | Konsep, Coding,  Laporan, Flowchart,  Pengecekan | 1. Referensi konsep program 2. Pengembangan utama pada program (Coding) 3. Finishing pada program (Coding) 4. Menjelaskan program di Laporan 5. Mencari bug dan *error* 6. Pengecekan alur program & flowchart |
| Muhammad Aqia Yudha Yulian Putra (2509116105) | Coding Support, Laporam, Flowchart | 1. Logika program 2. Penyusun laporan 3. Pengembangan Alur Flowchart 4. Support pengembangan program (Support Coding) 5. *Finishing* Flowchart |
| Rifaa Zainul Arifin  (2509116092) | Coding Support, Laporan, Flowchart | 1. Penyusun utama dan konsep laporan 2. Support pengembangan program (Support Coding) 3. Pengecekan alur program & flowchart 4. *Finishing* laporan |

**Lampiran 1 : Tabel Kontribusi**