

**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**

**2409116001**

**Dwi Pebriyanto Pradana**

**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

<b>KATA PENGANTAR.....</b>	i
<b>DAFTAR ISI .....</b>	ii
<b>DAFTAR GAMBAR.....</b>	iii
<b>DAFTAR TABEL .....</b>	v
<b>DAFTAR LAMPIRAN .....</b>	vi
<b>BAB I PENDAHULUAN.....</b>	7
1.1    Deskripsi Masalah.....	7
1.2    Rumusan Masalah .....	8
1.3    Batasan Masalah.....	9
1.4    Tujuan .....	9
1.5    Manfaat .....	10
<b>BAB II PERANCANGAN.....</b>	11
2.1 Analisis Program .....	11
2.2 Flowchart Sistem Pengelolaan Kelas Online.....	12
<b>BAB III HASIL DAN PEMBAHASAN .....</b>	14
3.1.    Implementasi Program .....	14
3.2.    Alur Program.....	14
3.3.    Source Code .....	29
<b>BAB IV PENUTUP.....</b>	40
4.1 Kesimpulan .....	40
4.2 Saran .....	40
<b>DAFTAR PUSTAKA.....</b>	42
<b>LAMPIRAN .....</b>	43

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Flowchart Menu Utama .....	12
<b>Gambar 2.2</b> Flowchart Menu Teacher .....	12
<b>Gambar 2.3</b> Flowchart Menu User .....	13
<b>Gambar 3.1</b> Source Code Menu Utama .....	15
<b>Gambar 3.2</b> Output Menu Utama .....	15
<b>Gambar 3.3</b> Source Code Register .....	16
<b>Gambar 3.4</b> Output Register .....	16
<b>Gambar 3.5</b> Source Code Login .....	17
<b>Gambar 3.6</b> Output Proses Login Berhasil .....	17
<b>Gambar 3.7</b> Output Proses Login Gagal .....	18
<b>Gambar 3.8</b> Source Code Menu User .....	19
<b>Gambar 3.9</b> Output Menu User .....	19
<b>Gambar 3.10</b> Source Code Daftar Kelas .....	20
<b>Gambar 3.11</b> Output Daftar Kelas .....	20
<b>Gambar 3.12</b> Source Code Top Up Saldo dan Invoice .....	21
<b>Gambar 3.13</b> Output Proses Top Up dan Invoice .....	21
<b>Gambar 3.14</b> Source Code Beli Kelas Bagian 1 .....	22
<b>Gambar 3.15</b> Proses Pembelian Kelas .....	22
<b>Gambar 3.16</b> Source Code Akses Kelas .....	23
<b>Gambar 3.17</b> Output Akses Kelas .....	23
<b>Gambar 3.18</b> Source Code Menu Teacher .....	24
<b>Gambar 3.19</b> Output Menu Teacher .....	24
<b>Gambar 3.20</b> Source Code Lihat Kelas Saya .....	25
<b>Gambar 3.21</b> Output Lihat Kelas Saya .....	25
<b>Gambar 3.22</b> Source Code Buat Kelas .....	25
<b>Gambar 3.23</b> Output Buat Kelas .....	26
<b>Gambar 3.24</b> Source Code Perbarui Kelas .....	26
<b>Gambar 3.25</b> Output Perbarui Kelas .....	27
<b>Gambar 3.26</b> Source Code Lihat yang Punya Akses Kelas .....	27

<b>Gambar 3.27</b> Output Lihat yang Punya Akses Kelas .....	28
<b>Gambar 3.28</b> Source Code Hapus Kelas .....	28
<b>Gambar 3.29</b> Output Hapus Kelas.....	28
<b>Gambar 3.30</b> Output Code Keluar Program .....	29

## **DAFTAR TABEL**

**Tabel 3.1** Source Code Sistem Pengelolaan Kelas Online .....29

## **DAFTAR LAMPIRAN**

<b>Lampiran 1</b> Table Kontribusi .....	43
<b>Lampiran 2</b> Dokumentasi Konsultasi Terakhir Kelompok 8 .....	44

# **BAB I**

## **PENDAHULUAN**

### **1.1 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

mengakses 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.

## 1.2 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?

### **1.3 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:
  - a. **Teacher**, yang memiliki hak untuk membuat, menampilkan, memperbarui, dan menghapus data kelas, menentukan harga kelas, juga melihat daftar pengguna yang sudah membeli kelas.
  - b. **User**, yang dapat melakukan registrasi dan login, melakukan top up saldo, melihat daftar kelas, serta membeli kelas sesuai harga yang sudah ditentukan.
4. Fitur yang diimplementasikan meliputi registrasi, login, manajemen kelas (CRUD), top up saldo, dan pembelian kelas.

### **1.4 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

## 1.5 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**

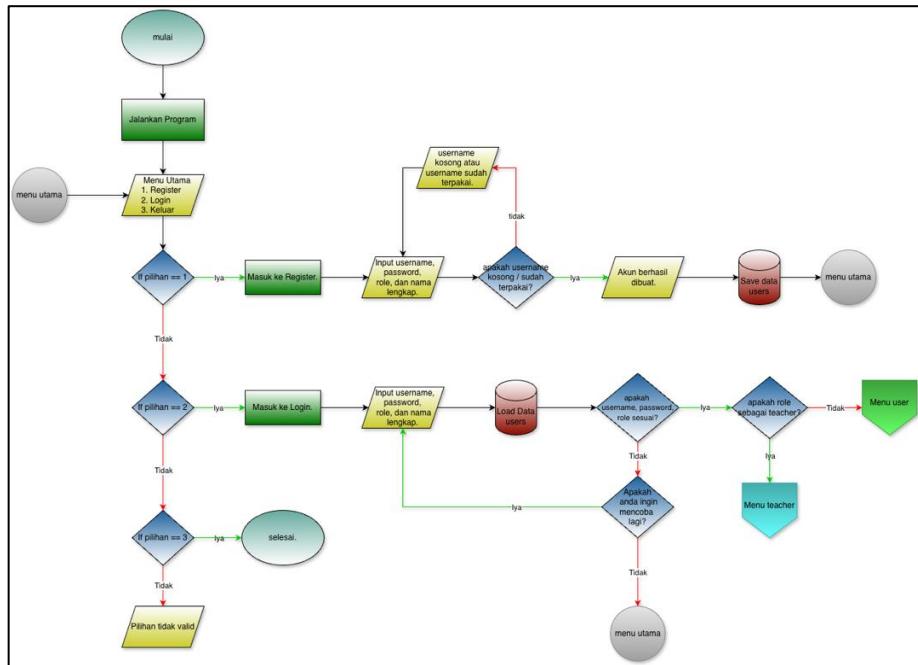
### **PERANCANGAN**

#### **2.1 Analisis Program**

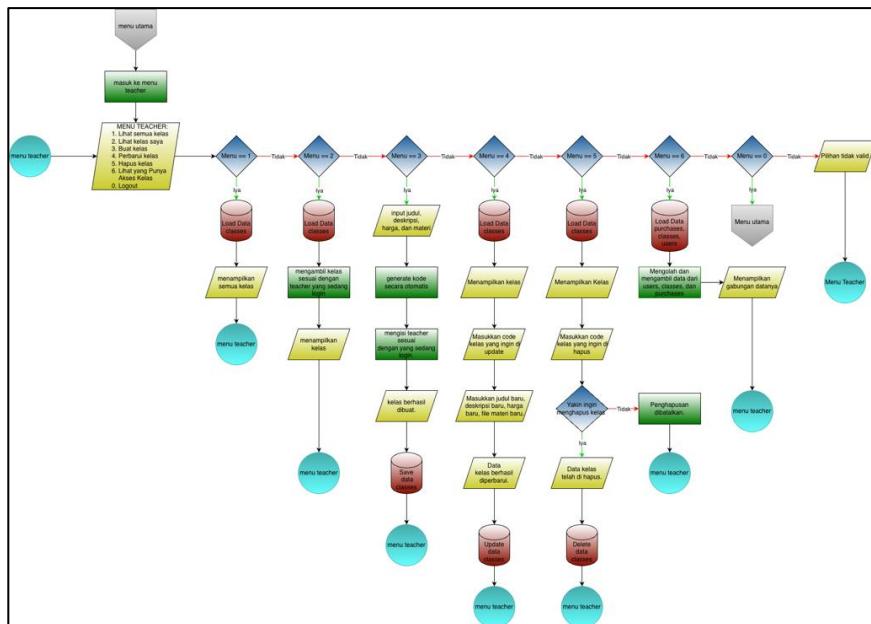
Program “Sistem Pengelolaan Kelas Online” ini dikembangkan untuk mempermudah proses pembelajaran dan pengelolaan kelas secara daring melalui sistem transaksi digital sederhana. Sistem ini dirancang agar kegiatan belajar dapat dilakukan dengan fleksibel tanpa terikat oleh waktu maupun tempat, serta memberikan 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.

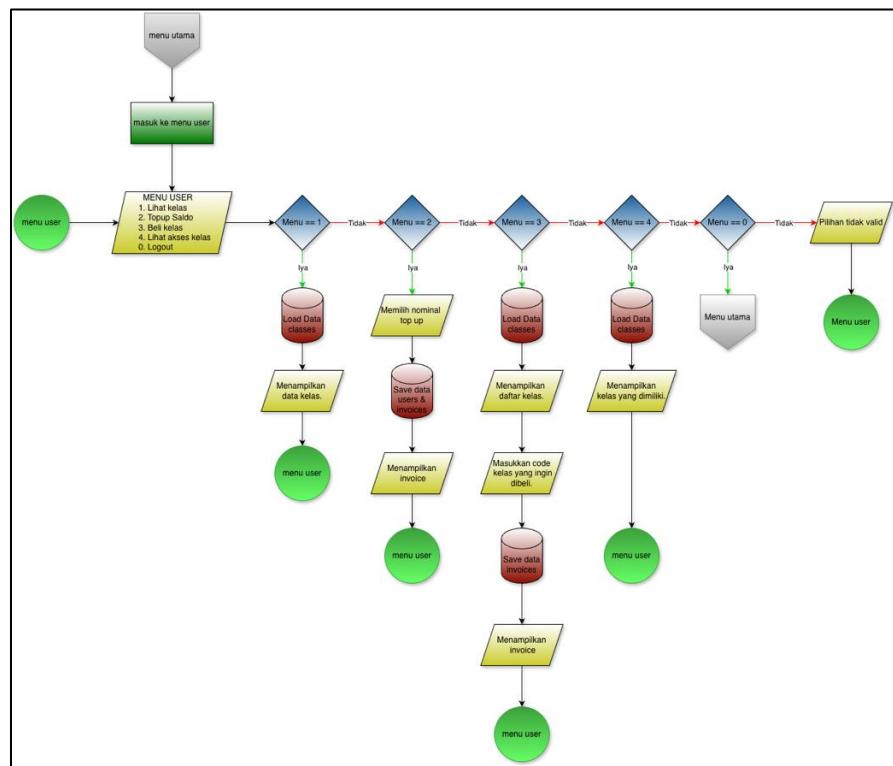
## 2.2 Flowchart Sistem Pengelolaan Kelas Online



Gambar 2.1 Flowchart Menu Utama



Gambar 2.2 Flowchart Menu Teacher



**Gambar 2.3** Flowchart Menu User

## **BAB III**

### **HASIL DAN PEMBAHASAN**

#### **3.1. Implementasi Program**

Program Sistem Pengelolaan Kelas Online dapat diimplementasikan dalam kehidupan nyata sebagai sistem pembelajaran digital sederhana yang membantu pengajar dan peserta didik dalam kegiatan belajar mengajar. Sistem ini dapat diterapkan di berbagai lingkungan, seperti lembaga kursus, sekolah, atau pelatihan daring yang ingin memiliki sistem pengelolaan kelas dan transaksi sederhana tanpa server khusus.

Beberapa contoh penerapan dalam dunia nyata:

1. Guru atau pengajar dapat membuat kelas, menentukan harga, serta memantau siapa saja yang telah membeli kelas tersebut.
2. Siswa atau peserta dapat mendaftar akun, mengisi saldo (top-up), membeli kelas sesuai kebutuhan, dan mengakses daftar kelas yang telah mereka beli.
3. Transaksi sederhana seperti pembelian kelas dan pengisian saldo dapat dijalankan langsung di terminal tanpa koneksi internet.

Selain di bidang pendidikan, sistem ini juga bisa dijadikan prototipe awal bagi pengembangan aplikasi e-learning atau marketplace kursus online yang lebih kompleks. Dengan menambahkan antarmuka grafis (GUI), sistem basis data seperti MySQL, serta integrasi pembayaran digital, sistem ini berpotensi menjadi platform pembelajaran online yang lengkap, modern, dan mudah digunakan.

#### **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

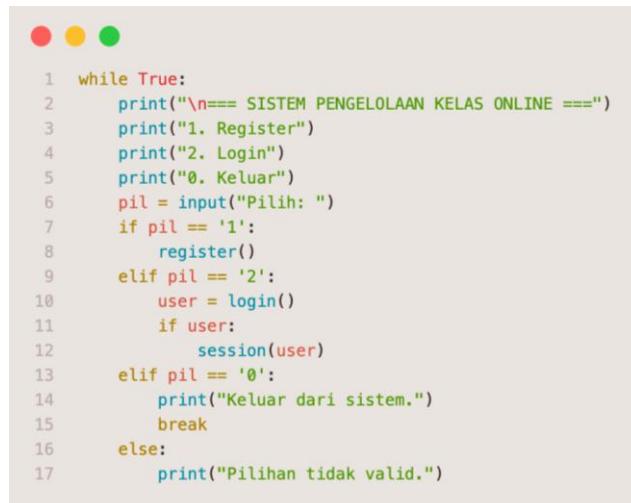
berbeda, namun keduanya saling terhubung melalui proses registrasi, login, dan pengelolaan data.

### 1. Tampilan Menu Utama

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

- 1) Register
- 2) Login
- 3) Keluar

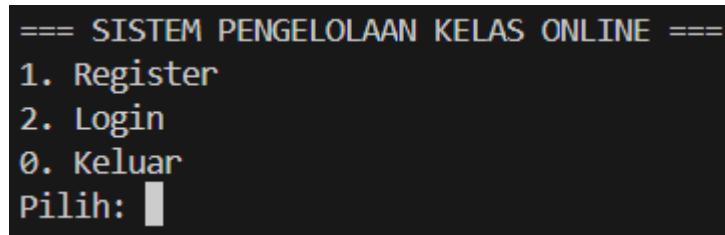
Pengguna dapat memilih 1 untuk melakukan registrasi akun baru, 2 untuk login, atau 0 untuk keluar dari sistem.



```
● ● ●

1 while True:
2     print("\n== SISTEM PENGELOLAAN KELAS ONLINE ==")
3     print("1. Register")
4     print("2. Login")
5     print("0. Keluar")
6     pil = input("Pilih: ")
7     if pil == '1':
8         register()
9     elif pil == '2':
10        user = login()
11        if user:
12            session(user)
13    elif pil == '0':
14        print("Keluar dari sistem.")
15        break
16    else:
17        print("Pilihan tidak valid.")
```

Gambar 3.1 Source Code Menu Utama



```
== SISTEM PENGELOLAAN KELAS ONLINE ==
1. Register
2. Login
0. Keluar
Pilih: |
```

Gambar 3.2 Output Menu Utama

## 2. 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.



```
● ● ●
1 def register():
2     users = read_csv("users.csv")
3     usernames = {u['username'] for u in users}
4     print("\n==== REGISTER ===")
5
6     while True:
7         username = input("Username: ").strip()
8         if not username:
9             print("Username tidak boleh kosong.")
10        elif username in usernames:
11            print("Username sudah dipakai.")
12        elif len(username) > 15:
13            print("Username tidak boleh melebihi 15 karakter!")
14        else:
15            break
16
17     password = pwinput("Password: ")
18     nama = input("Nama lengkap: ")
19     role = ""
20     while role not in ("student", "teacher"):
21         role = input("Role (student / teacher): ").strip().lower()
22
23     new = {
24         'id': str(len(users) + 1),
25         'username': username,
26         'password': password,
27         'nama': nama,
28         'role': role,
29         'saldo': '0'
30     }
31     append_csv("users.csv", new, ['id','username','password','nama','role','saldo'])
32     print("Registrasi selesai. Anda terdaftar sebagai {} dengan saldo 0.".format(role))
```

**Gambar 3.3** Source Code Register

```
==== REGISTER ===
Username: ZefriYudhaRifaa
Password: *****
Nama lengkap: ZEFRIYUDHARIFAA
Role (student / teacher): student
Registrasi selesai. Anda terdaftar sebagai student dengan saldo 0.
```

**Gambar 3.4** Output Register

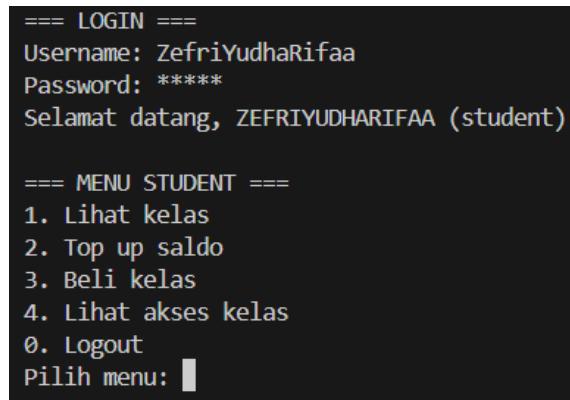
### 3. Proses Login

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.



```
1 def login():
2     while True:
3         print("\n==== LOGIN ===")
4         username = input("Username: ").strip()
5         password = pwinput("Password: ")
6
7         users = read_csv("users.csv")
8         for user in users:
9             if user['username'] == username and user['password'] == password:
10                 print(f"Selamat datang, {user['nama']} ({user['role']})")
11                 return user
12
13         print("Login gagal: username atau password salah.")
14         pilihan = input("Apakah Anda ingin mencoba lagi? (y/n): ").strip().lower()
15         if pilihan != 'y':
16             print("Kembali ke menu utama...")
17             return None
```

**Gambar 3.5** Source Code Login



```
==== LOGIN ===
Username: ZefriYudhaRifaa
Password: *****
Selamat datang, ZEFRIYUDHARIFAA (student)

==== MENU STUDENT ===
1. Lihat kelas
2. Top up saldo
3. Beli kelas
4. Lihat akses kelas
0. Logout
Pilih menu: 1
```

**Gambar 3.6** Output Proses Login Berhasil

```
==== LOGIN ====
Username: zefriyudharifaa
Password: *****
Login gagal: username atau password salah.
Apakah Anda ingin mencoba lagi? (y/n): y
```

**Gambar 3.7** Output Proses Login Gagal

#### 4. 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 while True:
2     print(f"\n==== MENU {user['role'].upper()} ===")
3     if user["role"] == "student":
4         print("1. Lihat kelas")
5         print("2. Top up saldo")
6         print("3. Beli kelas")
7         print("4. Lihat akses kelas")
8         print("0. Logout")
9         pil = input("Pilih menu: ")
10        if pil == '1':
11            list_classes()
12        elif pil == '2':
13            topup(user)
14        elif pil == '3':
15            buy_class(user)
16        elif pil == '4':
17            access_class(user)
18        elif pil == '0':
19            break
20        else:
21            print("Pilihan Anda tidak tersedia!")
```

**Gambar 3.8** Source Code Menu User

```
==== MENU STUDENT ====
1. Lihat kelas
2. Top up saldo
3. Beli kelas
4. Lihat akses kelas
0. Logout
Pilih menu: 3
```

**Gambar 3.9** Output Menu User

- Lihat daftar kelas

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

```

1  def list_classes(detail=False):
2      data = read_csv("classes.csv")
3      if not data:
4          print("Belum ada kelas.")
5          return
6      if detail:
7          headers = [
8              'kode',
9              'judul',
10             'dosen',
11             'deskripsi',
12             'harga',
13             'materi'
14         ]
15     else:
16         headers = [
17             'kode',
18             'judul',
19             'dosen',
20             'deskripsi',
21             'harga'
22         ]
23     show_table("Daftar Kelas", headers, data)

```

**Gambar 3.10** Source Code Daftar Kelas

Daftar Kelas					
kode	judul	dosen	deskripsi	harga	
95GA2	Dasar Pemrograman	maura	Ngoding ngoding	500000	
ZX2MH	Ilmu Pengetahuan Komputer	maura	Mengenai Komputer	2000000	
RMGTO	Pengantar Informasi	budiono	Informasi number one	60000	

**Gambar 3.11** Output Daftar Kelas

b. 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.



```

1 def tambah_saldo(user):
2     users = read_csv("users.csv")
3     topup_opsi = {'1':50000,'2':100000,'3':250000,'4':500000,'5':1000000}
4
5     print("\n==== TOP UP ===")
6     for k,v in topup_opsi.items():
7         print(f"{k}. Rp{v:,}")
8     pilihan = input("Pilih nominal (1-5): ").strip()
9     if pilihan not in topup_opsi:
10        print("Pilihan tidak valid.")
11        return user
12
13     jumlah = topup_opsi[pilihan]
14     for u in users:
15         if u['id'] == user['id']:
16             u['saldo'] = str(int(u['saldo']) + jumlah)
17             user['saldo'] = u['saldo']
18     write_csv("users.csv", users, ['id','username','password','nama','role','saldo'])
19     waktu = datetime.now().isoformat(sep=' ', timespec='seconds')
20
21     print("\n==== INVOICE TOP-UP ===")
22     print(f"Nama Pengguna : {user['nama']}")
23     print(f"Jumlah Top-Up : Rp{jumlah:,}")
24     print(f"Waktu Transaksi: {waktu}")
25     print(f"Saldo Sekarang : Rp{int(user['saldo']):,}")
26     print("====")
27     return user

```

**Gambar 3.12** Source Code Top Up Saldo dan Invoice

```

==== TOP UP ===
1. Rp50,000
2. Rp100,000
3. Rp250,000
4. Rp500,000
5. Rp1,000,000
Pilih nominal (1-5): 4

==== INVOICE TOP-UP ===
Nama Pengguna : ZEFRIYUDHARIFAA
Jumlah Top-Up : Rp500,000
Waktu Transaksi: 2025-10-23 12:18:20
Saldo Sekarang : Rp500,000
=====

```

**Gambar 3.13** Output Proses Top Up dan Invoice

c. 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.

```

● ● ● ● ●

1 def buy_class(user):
2     list_classes()
3     kode = input("Masukkan kode kelas yang ingin dibeli: ").strip()
4     classes = read_csv("classes.csv")
5     kelas = next((c for c in classes if c['kode'] == kode), None)
6     if not kelas:
7         print("Kelas tidak ditemukan.")
8         return
9
10    purchases = read_csv("purchases.csv")
11    sudah_beli = any(p['user_id'] == user['id'] and p['class_kode'] == kode for p in purchases)
12    if sudah_beli:
13        print("Anda sudah membeli kelas ini sebelumnya.")
14        return
15
16    harga = int(kelas['harga'])
17    if int(user['saldo']) < harga:
18        print("Saldo tidak cukup.")
19        return
20
21    user['saldo'] = str(int(user['saldo']) - harga)
22    users = read_csv("users.csv")
23    for u in users:
24        if u['id'] == user['id']:
25            u['saldo'] = user['saldo']

```

**Gambar 3.14** Source Code Beli Kelas Bagian 1

Daftar Kelas				
kode	judul	dosen	deskripsi	harga
95GA2	Dasar Pemrograman	maura	Ngoding ngoding	500000
ZX2MH	Ilmu Pengetahuan Komputer	maura	Mengenai Komputer	2000000
RMGTO	Pengantar Informasi	budiono	Informasi number one	60000

Masukkan kode kelas yang ingin dibeli: RMGTO

```

==== INVOICE PEMBELIAN ====
Nama: zefri al rizqullah
Kelas: Pengantar Informasi
Dosen: budiono
Harga: Rp60,000
Waktu: 2025-10-26 20:14:50
Sisa saldo: Rp440,000
=====
```

**Gambar 3.15** Proses Pembelian Kelas

d. Lihat akses Kelas

Menampilkan daftar kelas yang telah dibeli oleh student, termasuk nama teacher, deskripsi kelas.

```

1 def access_class(user):
2     purchases = read_csv("purchases.csv")
3     classes = read_csv("classes.csv")
4     owned_kodes = {p['class_kode'] for p in purchases if p['user_id']== user['id']}
5     owned_classes = [c for c in classes if c['kode'] in owned_kodes]
6     show_table("Kelas Dimiliki", ['kode','judul','teacher','deskripsi','materi'], owned_classes)

```

**Gambar 3.16** Source Code Akses Kelas

Kelas Dimiliki					
kode	judul	dosen	deskripsi	materi	
95GA2	Dasar Pemrograman	maura	Ngoding ngoding	ddp	
RMGTO	Pengantar Informasi	budiono	Informasi number one	Isi materi	

**Gambar 3.17** Output Akses Kelas

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

## 5. Menu Teacher

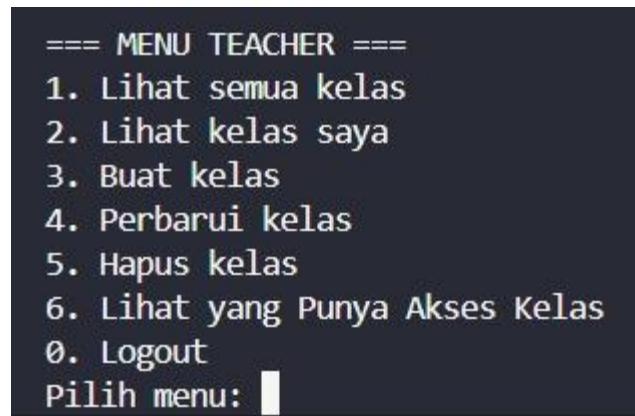
Untuk pengguna dengan peran *teacher*, sistem menampilkan menu berbeda, yaitu:

- 1) Lihat semua kelas
- 2) Lihat kelas saya
- 3) Buat kelas
- 4) Perbarui kelas
- 5) Hapus kelas
- 6) Lihat yang Punya Akses Kelas ini
- 7) Logout



```
1 elif user["role"] == "teacher":
2     print("1. Lihat semua kelas")
3     print("2. Lihat kelas saya")
4     print("3. Buat kelas")
5     print("4. Perbarui kelas")
6     print("5. Hapus kelas")
7     print("6. Lihat yang Punya Akses Kelas")
8     print("0. Logout")
9     pil = input("Pilih menu: ")
10    if pil == '1':
11        list_classes(True)
12    elif pil == '2':
13        my_classes(user)
14    elif pil == '3':
15        create_class(user)
16    elif pil == '4':
17        update_class(user)
18    elif pil == '5':
19        delete_class(user)
20    elif pil == '6':
21        lihat_akses_kelas(user)
22    elif pil == '0':
23        break
24    else:
25        print("Pilihan Anda tidak tersedia!")
26 else:
27     print("role Anda tidak valid! Terdapat kesalahan fatal dalam program. ☹")
28 break
```

**Gambar 3.18** Source Code Menu Teacher



```
==== MENU TEACHER ====
1. Lihat semua kelas
2. Lihat kelas saya
3. Buat kelas
4. Perbarui kelas
5. Hapus kelas
6. Lihat yang Punya Akses Kelas
0. Logout
Pilih menu: █
```

**Gambar 3.19** Output Menu Teacher

a. Lihat semua kelas

Menampilkan semua kelas beserta detail seperti kode, judul, deskripsi, harga, dan nama teacher (mirip dengan dengan menu student).

b. Lihat kelas saya

Menampilkan hanya kelas yang dibuat oleh teacher tersebut.

```

● ● ●

1 def my_classes(current_user):
2     classes = read_csv("classes.csv")
3     my_classes = [c for c in classes if c['teacher']==current_user['nama']]
4     if not my_classes:
5         print("Anda belum membuat kelas.")
6         return
7     show_table("Kelas saya (Teacher)", ['kode','judul','deskripsi','harga','materi'], my_classes)

```

**Gambar 3.20** Source Code Lihat Kelas Saya

Kelas saya (Teacher)				
kode	judul	deskripsi	harga	materi
95GA2	Dasar Pemrograman	Ngoding ngoding	500000	ddp
ZX2MH	Ilmu Pengetahuan Komputer	Mengenai Komputer	2000000	Sejarah Komputer tercipta di tahun 2025

**Gambar 3.21** Output Lihat Kelas Saya

### c. Buat kelas

Teacher dapat membuat kelas baru dengan mengisi form berisi judul, deskripsi, harga, dan materi. Sistem akan menghasilkan *kode kelas unik* menggunakan fungsi `generate_code()` yang terdiri dari kombinasi huruf kapital dan angka acak.

```

● ● ●

1 def create_class(user):
2     print("\n*** BUAT KELAS ***")
3     judul = input("Judul kelas: ")
4     deskripsi = input("Deskripsi: ")
5
6     while True:
7         try:
8             harga = int(input("Harga: "))
9             if harga < 50000 or harga > 5000000:
10                 print("Harga tidak boleh kurang dari Rp. 50.000 dan lebih dari Rp. 5.000.000")
11             else:
12                 break
13         except ValueError:
14             print("Harga yang Anda inputkan tidak valid!")
15
16     materi = input("Materi : ")
17     kode_kelas = generate_code()
18
19     new = {
20         'kode': kode_kelas,
21         'judul': judul,
22         'teacher': user['nama'],
23         'deskripsi': deskripsi,
24         'harga': harga,
25         'materi': materi
26     }
27     append_csv("classes.csv", new, ['kode','judul','teacher','deskripsi','harga','materi'])
28     print("Kelas berhasil dibuat dengan kode: {kode_kelas}")

```

**Gambar 3.22** Source Code Buat Kelas

```

==== BUAT KELAS ====
Judul kelas: Pengantar Informasi
Deskripsi: Informasi number one
Harga: 60000
Materi : Isi materi
Kelas berhasil dibuat dengan kode: RMGTO

```

**Gambar 3.23 Output Buat Kelas**

d. 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.

```

● ● ●

1 def update_kelas(current_user):
2     classes = read_csv("classes.csv")
3     my_classes = [c for c in classes if c['teacher'] == current_user['nama']]
4
5     if not my_classes:
6         print("Anda belum membuat kelas untuk diubah.")
7         return
8
9     show_table("Kelas Anda", ['kode','judul','deskripsi','harga','materi'], my_classes)
10    kode = input("Masukkan kode kelas yang ingin diupdate: ").strip()
11    kelas = next((c for c in classes if c['kode'] == kode and c['teacher'] == current_user['nama']), None)
12
13    if not kelas:
14        print("Kelas tidak ditemukan atau bukan milik Anda.")
15        return
16
17    print("Kosongkan jika tidak ingin mengubah data.")
18    judul_baru = input(f"Judul baru ({kelas['judul']}): ").strip() or kelas['judul']
19    deskripsi_baru = input(f"Deskripsi baru ({kelas['deskripsi']}): ").strip() or kelas['deskripsi']
20    while True:
21        try:
22            harga_baru = int(input(f"Harga baru ({kelas['harga']}): ").strip() or kelas['harga'])
23            if harga_baru < 50000 or harga_baru > 5000000:
24                print("Harga tidak boleh kurang dari Rp. 50.000 dan lebih dari Rp. 5.000.000")
25            else:
26                break
27        except ValueError:
28            print("Harga yang Anda inputkan tidak valid!")
29
30    materi_baru = input(f"Materi baru ({kelas['materi']}): ").strip() or kelas['materi']
31
32    for c in classes:
33        if c['kode'] == kode:
34            c['judul'] = judul_baru
35            c['deskripsi'] = deskripsi_baru
36            c['harga'] = harga_baru
37            c['materi'] = materi_baru
38
39    write_csv("classes.csv", classes, ['kode','judul','teacher','deskripsi','harga','materi'])
40    print("Kelas berhasil diperbarui.")

```

**Gambar 3.24 Source Code Perbarui Kelas**

```

Kelas Anda
+-----+-----+-----+-----+-----+
| kode | judul | deskripsi | harga | materi |
+-----+-----+-----+-----+-----+
| 95GA2 | Dasar Pemrograman | Ngoding ngoding | 500000 | ddp |
+-----+-----+-----+-----+-----+
| ZX2MH | nkjfjnjkdsdnfnfnsdfjk | jdfknsdklfnfnsdklnf | 2000000 | emfklnmsklf |
+-----+-----+-----+-----+-----+
Masukkan kode kelas yang ingin diupdate: ZX2MH
Kosongkan jika tidak ingin mengubah data.
Judul baru (nkjfjnjkdsdnfnfnsdfjk): Ilmu Pengetahuan Komputer
Deskripsi baru (jdfknsdklfnfnsdklnf): Mengenai Komputer
Harga baru (2000000):
Materi baru (emfklnmsklf): Sejarah Komputer tercipta di tahun 2025
Kelas berhasil diperbarui.

```

**Gambar 3.25** Output Perbarui Kelas

#### e. Lihat yang Punya Akses Kelas

Teacher dapat melihat data user yang memiliki akses kelas ia.

```

● ● ●

1 def lihat_akses_kelas(teacher):
2     purchases = read_csv("purchases.csv")
3     classes = read_csv("classes.csv")
4     users = read_csv("users.csv")
5
6     kelas_saya = [c for c in classes if c['teacher'] == teacher['nama']]
7     if not kelas_saya:
8         print("Anda belum membuat kelas, jadi belum ada pembeli.")
9         return
10
11    kode_kelas_saya = {c['kode'] for c in kelas_saya}
12
13    pembelian_terkait = [p for p in purchases if p['class_kode'] in kode_kelas_saya]
14    if not pembelian_terkait:
15        print("Belum ada user yang membeli kelas Anda.")
16        return
17
18    data_tabel = []
19    for p in pembelian_terkait:
20        user_data = next((u for u in users if u['id'] == p['user_id']), None)
21        kelas_data = next((c for c in classes if c['kode'] == p['class_kode']), None)
22        if user_data and kelas_data:
23            data_tabel.append({
24                'Nama Siswa': user_data['nama'],
25                'Username': user_data['username'],
26                'Kelas': kelas_data['judul'],
27                'Kode': kelas_data['kode'],
28                'Tanggal Beli': p['timestamp']
29            })
30
31    show_table("Daftar yang Mengakses Kelas Anda", ['Nama Siswa','Username','Kelas','Kode','Tanggal Beli'], data_tabel)

```

**Gambar 3.26** Source Code Lihat yang Punya Akses Kelas

Daftar yang Mengakses Kelas Anda				
Nama Siswa	Username	Kelas	Kode	Tanggal Beli
zefri al rizqullah	zefri	Dasar Pemrograman	95GA2	2025-10-22 20:16:52

**Gambar 3.27** Output Lihat yang Punya Akses Kelas

#### f. Hapus kelas

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

```

● ● ●
1 def delete_class(current_user):
2     classes = read_csv("classes.csv")
3     my_classes = [c for c in classes if c['teacher'] == current_user['nama']]
4
5     if not my_classes:
6         print("Anda belum membuat kelas untuk dihapus.")
7         return
8
9     show_table("Kelas Anda", ['kode','judul','deskripsi','harga','materi'], my_classes)
10    kode = input("Masukkan kode kelas yang ingin dihapus: ").strip()
11    kelas = next((c for c in classes if c['kode'] == kode and c['teacher'] == current_user['nama']), None)
12
13    if not kelas:
14        print("Kelas tidak ditemukan atau bukan milik Anda.")
15        return
16
17    konfirmasi = input(f"Yakin ingin menghapus kelas '{kelas['judul']}'? (y/n): ").strip().lower()
18    if konfirmasi != 'y':
19        print("Penghapusan dibatalkan.")
20        return
21
22    classes = [c for c in classes if c['kode'] != kode]
23    write_csv("classes.csv", classes, ['kode','judul','teacher','deskripsi','harga','materi'])
24    print(f"Kelas '{kelas['judul']}' telah dihapus.")

```

**Gambar 3.28** Source Code Hapus Kelas

Kelas Anda					
kode	judul	deskripsi	harga	materi	
95GA2	Dasar Pemrograman	Ngoding ngoding	500000	ddp	
W97BX	PPPPPLPPP	WEENWFSINFDSNFK	50000	MKEKFNEWKNFKDSNFKNSDKLNFKLDSNFKSDNFKNSDKFNDSKNFKDSNFK	
ZX2MH	nlkjfnrjksdnfsdfjk	jdfknsdklfnSDKlnf	200000	enfklnsklf	

Masukkan kode kelas yang ingin dihapus: W97BX  
Yakin ingin menghapus kelas 'PPPPPLPPP'? (y/n): y  
Kelas 'PPPPPLPPP' telah dihapus.

**Gambar 3.29** Output Hapus Kelas

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

#### 6. Transaksi dan Pembaruan Data

Pada bagian ini, sistem melakukan proses transaksi yang melibatkan perubahan data pengguna dan data kelas. Contohnya, ketika pengguna

melakukan pembelian kelas, sistem secara otomatis akan mengurangi saldo pengguna sesuai harga kelas dan menambahkan data kelas tersebut ke daftar pembelian pengguna. Selain itu, setiap perubahan yang dilakukan oleh Teacher (seperti menambah, mengubah, atau menghapus kelas) juga akan memperbarui data pada file penyimpanan (CSV). Dengan demikian, semua transaksi dan pembaruan data tersimpan secara permanen dan dapat diakses kembali ketika program dijalankan ulang.

## 7. Proses Keluar Program

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.

```
==== SISTEM PENGELOLAAN KELAS ONLINE ====
1. Register
2. Login
0. Keluar
Pilih: 0
Keluar dari sistem.

Program Selesai
```

**Gambar 3.30** Output Code Keluar Program

### 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, ALL

# 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
    table.hrules = ALL
    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.")
    elif len(username) > 15:
        print("Username tidak boleh melebihi 15 karakter!")
    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.")

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

```

```

# BAGIAN CRUD

def create_class(user):
    print("\n==== BUAT KELAS ===")
    judul = input("Judul kelas: ")
    deskripsi = input("Deskripsi: ")

    while True:
        try:
            harga = int(input("Harga: "))
            if harga < 50000 or harga > 5000000:
                print("Harga tidak boleh kurang dari Rp. 50.000 dan lebih dari Rp. 5.000.000")
            else:
                break
        except ValueError:
            print("Harga yang Anda inputkan tidak valid!")

    materi = input("Materi : ")
    kode_kelas = generate_code()

    new = {
        'kode': kode_kelas,
        'judul': judul,
        'teacher': user['nama'],
        'deskripsi': deskripsi,
        'harga': harga,
        'materi': materi
    }
    append_csv("classes.csv", new, ['kode','judul','teacher','deskripsi','harga','materi'])
    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['teacher'] == current_user['nama']]

    if not my_classes:
        print("Anda belum membuat kelas untuk diubah.")
        return

    show_table("Kelas Anda", ['kode','judul','deskripsi','harga','materi'], my_classes)
    kode = input("Masukkan kode kelas yang ingin diupdate: ").strip()
    kelas = next((c for c in classes if c['kode'] == kode and c['teacher'] == current_user['nama']), None)

    if not kelas:
        print("Kelas tidak ditemukan atau bukan milik Anda.")

```

```

        return

    print("Kosongkan jika tidak ingin mengubah data.")
    judul_baru = input(f"Judul baru ({kelas['judul']}): ").strip() or kelas['judul']
    deskripsi_baru = input(f"Deskripsi baru ({kelas['deskripsi']}): ").strip() or kelas['deskripsi']
    while True:
        try:
            harga_baru = int(input(f"Harga baru ({kelas['harga']}): ").strip() or kelas['harga'])
            if harga_baru < 50000 or harga_baru > 5000000:
                print("Harga tidak boleh kurang dari Rp. 50.000 dan lebih dari Rp. 5.000.000")
            else:
                break
        except ValueError:
            print("Harga yang Anda inputkan tidak valid!")

    materi_baru = input(f"Materi baru ({kelas['materi']}): ").strip() or kelas['materi']

    for c in classes:
        if c['kode'] == kode:
            c['judul'] = judul_baru
            c['deskripsi'] = deskripsi_baru
            c['harga'] = harga_baru
            c['materi'] = materi_baru

    write_csv("classes.csv", classes, ['kode','judul','teacher','deskripsi','harga','materi'])
    print("Kelas berhasil diperbarui.")

def delete_class(current_user):
    classes = read_csv("classes.csv")
    my_classes = [c for c in classes if c['teacher'] == current_user['nama']]

    if not my_classes:
        print("Anda belum membuat kelas untuk dihapus.")
        return

    show_table("Kelas Anda", ['kode','judul','deskripsi','harga','materi'], my_classes)
    kode = input("Masukkan kode kelas yang ingin dihapus: ").strip()
    kelas = next((c for c in classes if c['kode'] == kode and c['teacher'] == 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','teacher','deskripsi','harga','materi'])
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
    if detail:
        headers = [
            'kode',
            'judul',
            'teacher',
            'deskripsi',
            'harga',
            'materi'
        ]
    else:
        headers = [
            'kode',
            'judul',
            'teacher',
            'deskripsi',
            'harga'
        ]
    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['teacher']==current_user['nama']]
    if not my_classes:
        print("Anda belum membuat kelas.")
        return
    show_table("Kelas saya (Teacher)", ['kode','judul','deskripsi','harga','materi'], my_classes)

def topup(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['teacher'] 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"teacher: {kelas['teacher']}")
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','teacher','deskripsi','materi'], owned_classes)

def lihat_akses_kelas(teacher):
    purchases = read_csv("purchases.csv")
    classes = read_csv("classes.csv")
    users = read_csv("users.csv")

    kelas_saya = [c for c in classes if c['teacher'] == teacher['nama']]
    if not kelas_saya:
        print("Anda belum membuat kelas, jadi belum ada pembeli.")
        return

    kode_kelas_saya = {c['kode'] for c in kelas_saya}

    pembelian_terkait = [p for p in purchases if p['class_kode'] in kode_kelas_saya]
    if not pembelian_terkait:

```

```

print("Belum ada user yang membeli kelas Anda.")
return

data_tabel = []
for p in pembelian_terkait:
    user_data = next((u for u in users if u['id'] == p['user_id']), None)
    kelas_data = next((c for c in classes if c['kode'] == p['class_kode']), None)
    if user_data and kelas_data:
        data_tabel.append({
            'Nama Siswa': user_data['nama'],
            'Username': user_data['username'],
            'Kelas': kelas_data['judul'],
            'Kode': kelas_data['kode'],
            'Tanggal Beli': p['timestamp']
        })

show_table("Daftar yang Mengakses Kelas Anda", ["Nama Siswa",'Username','Kelas','Kode','Tanggal Beli'], data_tabel)

# 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 session(user):
    while True:
        print("\n==== MENU {} ====".format(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':
    topup(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("6. Lihat yang Punya Akses 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 == '6':
        lihat_akses_kelas(user)
    elif pil == '0':
        break
    else:
        print("Pilihan Anda tidak tersedia!")
else:
    print("role Anda tidak valid! Terdapat kesalahan fatal dalam program. 🤖")
    break

try:

```

```
main_menu()
except KeyboardInterrupt:
    print("\n\nProgram dihentikan!")
except EOFError:
    print("\n\nInput dihentikan, program ditutup.")
except Exception as e:
    print("\n\nTerjadi kesalahan")
finally:
    print("\nProgram Selesai")
```

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Sistem Pengelolaan Kelas Online yang dikembangkan menggunakan bahasa pemrograman Python bertujuan untuk mempermudah proses pembelajaran daring melalui fitur-fitur utama seperti registrasi, login, top up saldo, pembelian kelas, serta pengelolaan data oleh Teacher menggunakan fungsi CRUD. Program ini berhasil menunjukkan penerapan konsep dasar pemrograman, logika transaksi digital, dan pengelolaan data berbasis file CSV dalam bentuk aplikasi yang sederhana namun bermanfaat. Dengan adanya sistem ini, proses pembelajaran online dapat dilakukan secara lebih fleksibel, efisien, dan terorganisir, sehingga memberikan pengalaman belajar digital yang modern, mudah diakses, serta mendukung kegiatan belajar mengajar secara praktis.

#### **4.2 Saran**

Berdasarkan hasil perancangan dan implementasi *Sistem Pengelolaan Kelas Online* ini, kami memberikan beberapa saran pengembangan agar program dapat menjadi lebih baik ke depannya, yaitu:

1. Peningkatan Sistem Penyimpanan Data

Disarankan untuk mengganti penyimpanan berbasis file CSV ke basis data seperti MySQL atau SQLite agar pengelolaan data lebih aman, cepat, dan terstruktur.

2. Integrasi Sistem Pembayaran Digital

Fitur top up saldo dapat dikembangkan agar pengguna dapat menentukan nominal isi saldo sendiri atau bahkan terhubung dengan sistem pembayaran digital seperti e-wallet.

3. Pengembangan Fitur Notifikasi dan Laporan

Program dapat diperluas dengan fitur notifikasi pembelian kelas atau laporan transaksi otomatis agar pengguna dapat memantau aktivitas mereka dengan lebih mudah.

Dengan adanya pengembangan lebih lanjut, diharapkan sistem ini dapat menjadi platform pembelajaran digital yang lebih lengkap, aman, dan efisien dalam mendukung kegiatan belajar mengajar secara daring.

## **DAFTAR PUSTAKA**

PA-8 SISTEM PENGELOLAAN KELAS ONLINE. (2025).  
*sistem\_pengelolaan\_kelas\_online* [Kode sumber]. GitHub.  
[https://github.com/PA-8-SISTEM-PENGELOLAAN-KELAS-ONLINE/sistem\\_pengelolaan\\_kelas\\_online](https://github.com/PA-8-SISTEM-PENGELOLAAN-KELAS-ONLINE/sistem_pengelolaan_kelas_online)

## LAMPIRAN

**Lampiran 1** Table Kontribusi

Nama	Kontribusi	Bagian
Zefri Al Rizqullah (2509116084)	Konsep, Coding, Laporan, Flowchart, Pengecekan	<ol style="list-style-type: none"><li>1. Referensi konsep program</li><li>2. Pengembangan pada program (Coding)</li><li>3. <i>Finishing</i> pada program (Coding)</li><li>4. <i>Finishing</i> Laporan</li><li>5. Mencari bug dan <i>error</i></li><li>6. Pengecekan alur program &amp; flowchart</li></ol>
Muhammad Aqia Yudha Yulian Putra (2509116105)	Laporan, Flowchart	<ol style="list-style-type: none"><li>1. Logika program</li><li>2. Penyusun laporan</li><li>3. Pengembangan Alur Flowchart</li><li>4. <i>Design</i> Flowchart</li><li>5. <i>Finishing</i> Flowchart</li></ol>
Rifaa Zainul Arifin (2509116092)	Laporan, Flowchart	<ol style="list-style-type: none"><li>1. Penyusun utama dan konsep laporan</li><li>2. Pengecekan alur program &amp; flowchart</li></ol>

**Lampiran 2 Dokumentasi Konsultasi Terakhir Kelompok 8**

