

APLIKASI BOOKING PENYEWAAN STUDIO FOTOGRAFI



**Disusun Oleh:
KELOMPOK 5**

Zahra Aurelly Herdiansyah	(2409116062)
Satria Rajawali Ektya Antara	(2409116067)
Irvan Alif	(2409116071)

Asisten Laboratorium:

<u>MUHAMMAD ARIFIN DAVA</u>	<u>MUHAMMAD AIDIL MIRZA</u>
(2309116059)	(2409116040)

**PROGRAM STUDI SISTEM INFORMASI
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN**

2024

Kata Pengantar

Puji syukur kehadiran Tuhan Yang Maha Esa atas limpahan rahmat, berkah, dan karunia-Nya sehingga kami dapat menyelesaikan laporan proyek akhir berjudul “Aplikasi Booking Penyewaan Studio Fotografi” dengan tepat waktu. Pembuatan laporan ini bertujuan untuk memenuhi proyek akhir dari praktikum yang dilaksanakan, serta diharapkan mampu untuk membantu kita dalam menguasai bahasa pemrograman python, sehingga nantinya kita dapat mengikuti perkembangan teknologi dan membuat suatu program yang memberikan manfaat bagi masyarakat luas. Dengan disusunnya laporan ini, tidak lupa kami ucapkan rasa terimakasih sebesar-besarnya kepada:

1. Bapak Putut Widagdo, S.Kom., M.Kom. Dan Bapak Amin Padmo Azam Masa, S.Kom., M.Cs. selaku dosen pengampu mata kuliah Dasar-Dasar Pemrograman
2. Abang Muhammad Arifin Dava selaku pembimbing proyek akhir praktikum Dasar-Dasar Pemrograman.
3. Seluruh Asisten Laboratorium praktikum Dasar-Dasar Pemrograman 2023
4. Dan teman-teman yang berkontribusi dalam membantu kami menyelesaikan program ini.

Kami menyadari bahwa laporan yang kami buat masih jauh dari kata sempurna dan masih banyak terdapat kesalahan. Oleh karena itu, kami mengharapkan dapat menerima kritik dan saran yang bersifat membangun agar kami dapat mengetahui dimana letak kesalahan kami dan tidak mengulangnya lagi di laporan selanjutnya yang akan datang.

Samarinda, 7 November 2024

Kelompok 5 B 2024

Daftar Isi

Kata Pengantar	ii
Daftar Isi	iii
Daftar Gambar	iii
Daftar Tabel	vii
BAB I.....	8
1.1 Deskripsi Masalah.....	8
1.2 Rumusan Masalah.....	8
1.3 Batasan Masalah	8
1.4 Tujuan	8
1.5 Manfaat	8
BAB II	9
2.1 Analisis Program.....	9
2.2 Flowchart	9
BAB III.....	10
3.1 Implementasi Program	10
3.2 Alur Program	10
BAB IV	11
4.1 Kesimpulan	12
4.2 Saran	12
Daftar Pustaka	13
Lampiran	14

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

Daftar Tabel

Tabel 1.1 Tabel Kontribusi	74
---	----

BAB I

PANDAHULUAN

1.1 Deskripsi Masalah

Selama praktikum mata kuliah dasar-dasar pemrograman berlangsung mahasiswa diajarkan menggunakan bahasa pemrograman python. Seperti yang kita ketahui Python merupakan bahasa pemrograman dinamis yang mendukung pemrograman berbasis objek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan bahasa pemrograman tingkat tinggi yang diracik oleh Guido van Rossum.

Python banyak digunakan untuk membuat berbagai macam program, seperti: program CLI, Program GUI (desktop), Aplikasi Mobile, Web, IoT, Game, Program untuk Hacking, dsb. Python juga dikenal dengan bahasa pemrograman yang mudah dipelajari, karena struktur sintaknya rapi dan mudah dipahami. Dengan bahasa python ini kami kaitkan dengan kemajuan teknologi yang turut mempengaruhi semua bidang kehidupan untuk menjadi semakin lebih mudah dan cepat dalam melakukan sebuah pekerjaan.

Berdasarkan fakta yang beredar bahwa, masih banyak toko yang menggunakan sistem pelayanan dengan cara yang manual. Dengan sistem tersebut terkadang menyebabkan masalah yang dapat terjadi seperti pada saat melakukan pemesanan atau transaksi pembayaran dan hal tersebut menyebabkan ketidakpuasan pelanggan terhadap sistem yang masih dilakukan dengan cara manual.

Maka berdasarkan latar belakang tersebut kami merancang sebuah proyek dengan judul " Aplikasi Booking Penyewaan Studio Fotografi " yang disusun menggunakan bahasa dan struktur dari pemrograman python yang akan mempermudah suatu transaksi serta pelayanan dalam penjualan. Dengan sistem

ini, diharapkan penyewaan studio fotografi dapat menjadi lebih efisien, terorganisir, dan memberikan kenyamanan bagi pengguna serta pemilik studio.

1.2 Rumusan Masalah

1. Bagaimana cara menyediakan platform yang memudahkan pengguna untuk mencari dan memesan studio fotografi secara online?
2. Bagaimana mengatur ketersediaan jadwal studio secara efisien?
3. Bagaimana cara membuat aplikasi yang mudah digunakan, baik untuk pengguna umum maupun admin pemilik studio?

1.3 Batasan Masalah

1. Menggunakan bahasa pemrograman python
2. Program ini membutuhkan database

1.4 Tujuan

1. Mempermudah proses pencarian dan pemesanan studio
2. Mengoptimalkan pengelolaan ketersediaan jadwal
3. Meningkatkan efisiensi operasional studio fotografi

1.5 Manfaat

1. Proses pemesanan yang cepat dan efisien
2. Kemudahan pembayaran online
3. Pengalaman pengguna yang user-friendly

BAB II sss

Perancangan

2.1 Analisis Program

Library yang digunakan pada program ini adalah:

1. `import csv`:

untuk menginput csv yang digunakan untuk penyimpanan dinamis yaitu untuk menyimpan data user menu dan stock produk dan harga

2. `pwinput`:

Fungsi: Library ini digunakan untuk memudahkan penggunaan masukan berbentuk kata sandi (password) dengan tampilan yang lebih aman. Cara kerja: `pwinput` akan menyembunyikan masukan password yang diketik oleh pengguna sehingga tidak terlihat dalam teks biasa di layar. Ini meningkatkan keamanan karena kata sandi tidak akan terbaca oleh mata orang lain yang melihat layar. Pengguna akan diminta untuk memasukkan kata sandi, dan nilai yang dimasukkan akan disaring untuk keamanan. Contoh penggunaan: `password = pwinput("Masukkan kata sandi: ")`

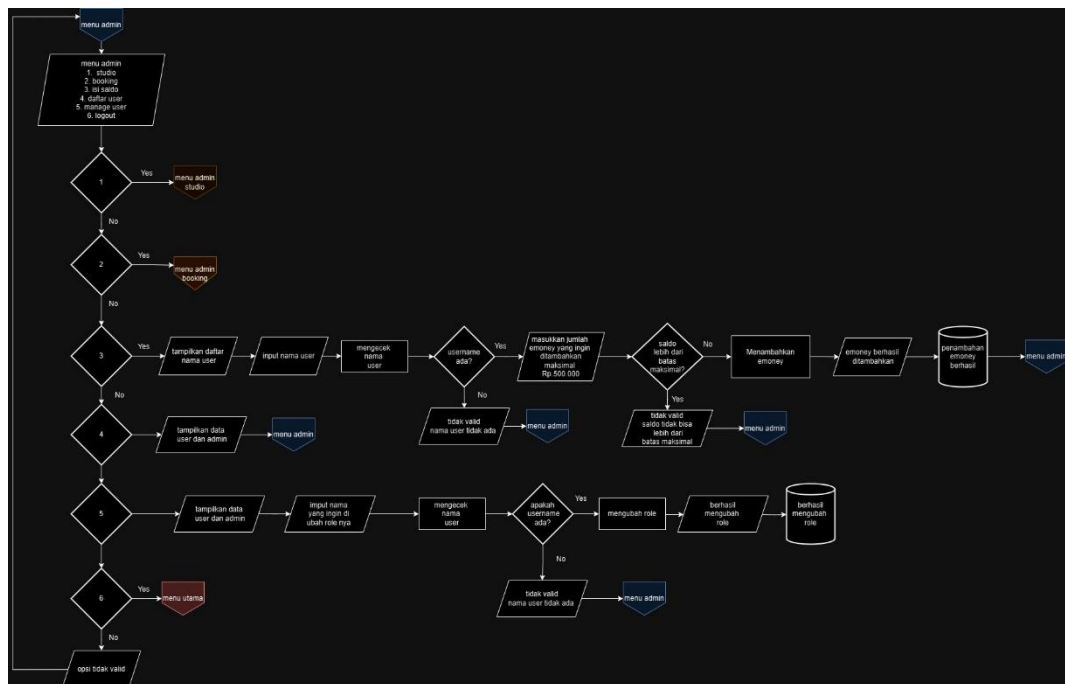
3. `PrettyTable`:

Fungsi: Library ini digunakan untuk membuat tabel yang diformat dengan baik dalam teks untuk keperluan tampilan data dalam bentuk tabel. Cara kerja: `PrettyTable` memungkinkan Anda untuk membuat tabel dengan kolom dan baris, dan Anda dapat dengan mudah menambahkan data ke dalamnya. Anda dapat mengatur tampilan tabel, termasuk mengubah tampilan header dan penyesuaian format kolom.

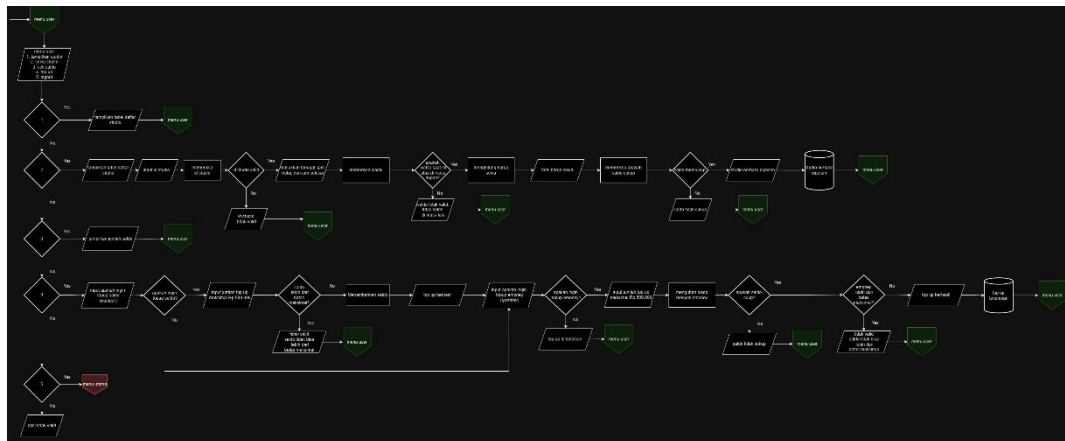

```

graph TD
    Start([mulai]) --> Menu[/menu utama  
1. registrasi  
2. login  
3. exit/]
    Menu --> D1{1}
    D1 -- Yes --> InputU[/masukkan username/]
    InputU --> CheckU[memeriksa apakah username sudah ada]
    CheckU -- No --> InputP[/masukkan password/]
    InputP --> AddUP[/Menambahkan username dan password baru/]
    AddUP --> SuccessUP[/berhasil menambahkan username dan password baru/]
    SuccessUP --> DB[(penambahan username dan password)]
    DB --> Menu
    CheckU -- Yes --> MenuU[/menu utama/]
    D1 -- No --> D2{2}
    D2 -- Yes --> InputUP[/masukkan username dan password/]
    InputUP --> CheckL{apakah login sudah 3 kali}
    CheckL -- No --> CheckDB[memeriksa apakah username ada di dalam data base]
    CheckDB --> D3{apakah sebagai admin}
    D3 -- Yes --> MenuAdmin[/menu admin/]
    D3 -- No --> D4{apakah sebagai user}
    D4 -- Yes --> MenuUser[/menu user/]
    D4 -- No --> CheckL
    CheckL -- Yes --> Timeout[/kesempatan habis/]
    Timeout --> End([Selesai])
    D2 -- No --> D3
    D2 -- No --> D4
    D2 -- No --> Invalid[/opsi tidak valid/]
    Invalid --> Menu
    D3 -- Yes --> Welcome[/Terimakasih telah Menggunakan Program ini/]
    Welcome --> End
  
```

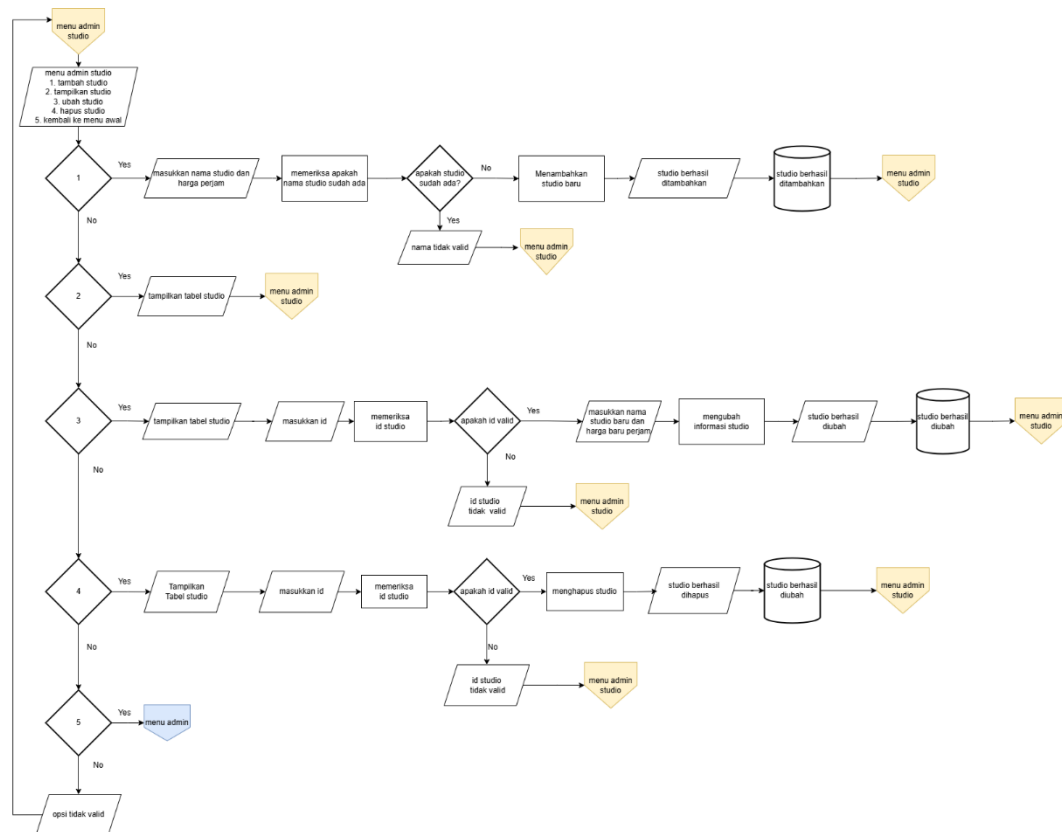
Gambar 2.1 Flowchart 1.1



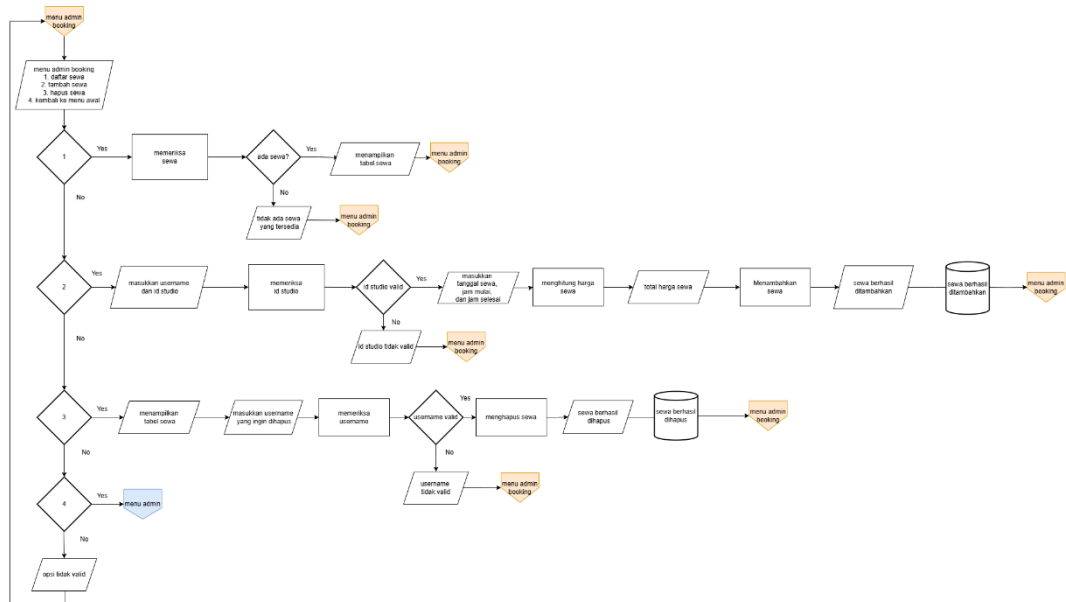
Gambar 2.2 Flowchart 1.2



Gambar 2.3 Flowchart 1.3



Gambar 2.4 Flowchart 1.4



Gambar 2.5 Flowchart 1.5

BAB III

Hasil dan Pembahasan

3.1 Implementasi Program

a. fungsi

```
1 import csv
2 from prettytable import PrettyTable
3 import pwinput
4 from datetime import datetime
```

Gambar 3.1.1

- import csv: digunakan untuk membaca dan menulis file csv di python, biasa digunakan untuk menyimpan data dalam format
- from prettytable import PrettyTable: digunakan untuk menampilkan data dalam bentuk table yang rapih di terminal
- import pwinput: digunakan untuk mengambil input password dari pengguna dengan cara yang lebih aman, akan menyembunyikan input yang dimasukkan
- from datetime import datetime: digunakan untuk bekerja dengan tanggal dan waktu di python

b. dictionary

```

7  studios = {
8
9  }
10 users = {}
11 booking = {
12     1: [], 2: [], 3: [], 4: []
13 }
14 transactions = {
15
16 }

```

Gambar 3.1.2

- studios: menampung informasi tentang studio yang tersedia
- users: menampung data pengguna yang terdaftar
- booking: menampung data pemesanan untuk setiap studio, setiap studio memiliki daftar pemesanan yang dilakukan oleh pengguna
- transactions: menampung data transaksi yang telah terjadi

```

21 def load_studios():
22     try:
23         with open('studios.csv', mode='r', newline='') as file:
24             csv_reader = csv.reader(file)
25             for row in csv_reader:
26                 if len(row) >= 3:
27                     studio_id, name, price, = row
28                     studios[studio_id] = {'name': name, 'price': int(price)}
29     except FileNotFoundError:
30         print("File 'studios.csv' tidak ditemukan, membuat file baru.")
31         with open('username_user.csv', mode='w', newline='') as file:
32             pass

```

Gambar 3.1.3

- bertujuan untuk memuat data studio dari file csv dan menyimpannya dalam dictionary studios, fungsi ini juga menangani kasus Ketika file csv yang dimaksud tidak ditemukan dengan membuat file baru

```

34 def load_users():
35     try:
36         with open('username_user.csv', mode='r', newline='') as file:
37             csv_reader = csv.reader(file)
38             for row in csv_reader:
39                 if len(row) >= 4:
40                     username, password, saldo, emoney, role = row
41                     users[username] = {'password': password, 'saldo': int(saldo), 'emoney': int(emoney), 'role': role}
42     except FileNotFoundError:
43         print("File 'username_user.csv' tidak ditemukan, membuat file baru.")
44         with open('username_user.csv', mode='w', newline='') as file:
45             pass

```

Gambar 3.1.4

- berfungsi untuk membaca data pengguna dari file csv dan jika tidak ditemukan, ia membuat file kosong baru. Data yang berhasil dimuat disimpan dalam dictionary users, yang mempermudah aplikasi dalam mengelola informasi pengguna.

```

46
47 def load_transactions():
48     try:
49         with open('invoice.csv', mode='r', newline='') as file:
50             csv_reader = csv.DictReader(file)
51             for row in csv_reader:
52                 studio_id = row['studio_id']
53                 if studio_id not in transactions:
54                     transactions[studio_id] = []
55
56                 transactions[studio_id].append({
57                     "username": row['username'],
58                     "nama_studio": row['nama_studio'],
59                     "durasi": float(row['durasi']),
60                     "tanggal": row['tanggal'],
61                     "jam_mulai": row['jam_mulai'],
62                     "jam_selesai": row['jam_selesai'],
63                     "harga_total": float(row['harga_total'])
64                 })
65
66     except FileNotFoundError:
67         print("File 'invoice.csv' tidak ditemukan.")
68     return transactions

```

Gambar 3.1.5

- bertujuan untuk memuat data transaksi dari file csv(invoice.csv) dan menyimpannya dalam dictionary transactions. Fungsi ini menangani kasus di mana file tidak ditemukan dengan mencetak pesan kesalahan. Fungsi load_transactions() memungkinkan Anda untuk memuat dan mengelompokkan data transaksi yang ada dalam file CSV berdasarkan studio_id. Jika file CSV tidak ditemukan, fungsi ini akan menangani error dengan mencetak pesan dan tidak menyebabkan program crash. Fungsi ini sangat berguna untuk aplikasi yang memerlukan pemrosesan transaksi dan laporan terkait studio, seperti aplikasi pemesanan ruang studio atau penyewaan fasilitas lainnya.

```

69
70 def save_studios(studios):
71     try:
72         with open('studios.csv', mode='w', newline='') as file:
73             csv_writer = csv.writer(file)
74             for id, studio_data in studios.items():
75                 row = [
76                     id,
77                     studio_data['name'],
78                     studio_data['price']
79                 ]
80                 csv_writer.writerow(row)
81     except Exception as e:
82         print(f"Terjadi kesalahan saat menyimpan data pengguna: {e}")
83
84

```

Gambar 3.1.6

- Fungsi `save_studios()` digunakan untuk menyimpan data studio ke dalam file CSV. Fungsi ini membuka file dalam mode tulis ('w'), menulis data studio dalam format CSV, dan menangani kemungkinan kesalahan yang terjadi selama proses penyimpanan. Fungsi ini sangat berguna untuk menyimpan dan memperbarui data studio dalam aplikasi yang membutuhkan penyimpanan data di luar program, misalnya untuk aplikasi manajemen penyewaan studio atau fasilitas serupa.

```

85 def save_transactions(transactions):
86     try:
87         with open('invoice.csv', mode='w', newline='') as file:
88             csv_writer = csv.DictWriter(file, fieldnames=['studio_id', 'username', 'nama_studio', 'durasi', 'tanggal', 'jam_mulai', 'jam_selesai', 'harga_total'])
89             csv_writer.writeheader()
90
91             for studio_id, transaksi in transactions.items():
92                 for transaksi in transaksi:
93                     row = {
94                         'studio_id': studio_id,
95                         'username': transaksi['username'],
96                         'nama_studio': transaksi['nama_studio'],
97                         'durasi': transaksi['durasi'],
98                         'tanggal': transaksi['tanggal'],
99                         'jam_mulai': transaksi['jam_mulai'],
100                        'jam_selesai': transaksi['jam_selesai'],
101                        'harga_total': transaksi['harga_total']
102                    }
103                    csv_writer.writerow(row)
104                    print("Transaksi berhasil disimpan.")
105            except Exception as e:
106                print(f"Terjadi kesalahan saat menyimpan transaksi: {e}")
107

```

Gambar 3.1.7

- Fungsi `save_transactions()` bertujuan untuk menyimpan data transaksi yang terdapat dalam parameter `transactions` kedalam file csv.

```

107
108 def save_users(users):
109     try:
110         with open('username_user.csv', mode='w', newline='') as file:
111             csv_writer = csv.writer(file)
112             csv_writer.writerow(['username', 'password', 'saldo', 'emoney', 'role'])
113
114             for username, user_data in users.items():
115                 row = [
116                     username,
117                     user_data['password'],
118                     user_data['saldo'],
119                     user_data['emoney'],
120                     user_data['role']
121                 ]
122                 csv_writer.writerow(row)
123
124     print("Pengguna berhasil disimpan.")
125 except Exception as e:
126     print(f"Terjadi kesalahan saat menyimpan data pengguna: {e}")
127
128

```

Gambar 3.1.8

- Fungsi `save_users()` bertujuan untuk menyimpan data pengguna ke dalam sebuah file csv.

```

128
129 def main_menu():
130     while True:
131         print("\n=====")
132         print("\n===Menu Utama===")
133         print("1. Registrasi")
134         print("2. Login")
135         print("3. Exit")
136         print("\n=====")
137         choice = input("Pilih opsi: ")
138
139         if choice == '1':
140             register()
141         elif choice == '2':
142             username = login()
143             if username:
144                 user_menu(username)
145         elif choice == '3':
146             print("Terima kasih telah menggunakan aplikasi.")
147             exit()
148         else:
149             print("Opsi tidak valid. Silahkan dicoba kembali!")
150

```

Gambar 3.1.9

- Fungsi `main_menu()` merupakan bagian dari system menu utama aplikasi yang memungkinkan pengguna untuk memilih antara tiga opsi.


```

150
151 def register():
152     username = input("Masukkan username: ")
153
154     if username in users:
155         print("Username sudah ada. Silakan pilih username lain.")
156         return
157
158     password = pwinput.pwinput("Masukkan password: ")
159     saldo = 0
160     emoney = 0
161     role = 'user'
162     users[username] = {'password': password, 'saldo': saldo, 'emoney' : emoney, 'role': 'user'}
163     print("Akun berhasil dibuat!")
164
165     with open('username_user.csv', mode='a', newline='') as file:
166         csv_writer = csv.writer(file)
167         csv_writer.writerow([username, password, saldo, emoney, role])
168     print(f>Data {username} berhasil ditambahkan.")

```

Gambar 3.1.10

- Fungsi ini digunakan untuk mendaftarkan (registrasi) pengguna baru ke dalam system. Fungsi ini memastikan bahwa pengguna baru bisa mendaftar dan datanya disimpan baik.

```

169
170 def login():
171     load_users()
172     x = 3
173     while x > 0:
174         username = input("Masukkan username: ")
175         password = pwinput.pwinput("Masukkan password: ")
176
177         if username in users and users[username]['password'] == password and users[username]['role'] == 'admin':
178             print("Login admin berhasil!")
179             admin_menu()
180             return
181         elif username in users and users[username]['password'] == password and users[username]['role'] == 'user':
182             print("Login berhasil!")
183             return username
184         else:
185             x -= 1
186             if x == 0:
187                 print("Anda telah mencoba login 3 kali. Program akan keluar.")
188                 exit()
189             else:
190                 print(f"Username atau password salah. Sisa percobaan: {x}")
191
192     return None
193

```

Gambar 3.1.11

- Fungsi login() bertujuan untuk memverifikasi kredensial pengguna dan memberikan akses ke menu berdasarkan role pengguna.

```

92 #menu user
93 def user_menu(username):
94     while True:
95         print("\n-----")
96         print("\n---Menu User---")
97         print("1. Tampilkan Studio")
98         print("2. Sewa Studio")
99         print("3. Cek Saldo")
100        print("4. Top up")
101        print("5. Logout")
102        print("\n-----")
103        choice = input("Pilih opsi: ")
104
105        if choice == '1':
106            tampil_studios()
107        elif choice == '2':
108            sewa_studio(username)
109        elif choice == '3':
110            if users[username]['emoney'] == 0:
111                print("Saldo E-Money Anda masih kosong.")
112            else:
113                print(f"Saldo E-Money Anda: {users[username].get('emoney', 0)}")
114        elif choice == "4":
115            print(f"Saldo anda saat ini: {users[username]['saldo']}")
116            saldo = input("Apakah ingin top up saldo anda? \nTekan (iya/tidak): ")
117            if saldo == 'iya':
118                try:
119                    top_up = int(input("Masukkan jumlah top up saldo anda: "))
120                    if top_up <= 0:
121                        print("Jumlah top up tidak valid.")
122                        continue
123                    elif top_up > 500000:
124                        print("Top up saldo melebihi batas maksimal yaitu 500000")
125                        continue
126                except ValueError:
127                    print("Input harus berupa angka.")
128                    continue
129                users[username]['saldo'] += top_up
130                print(f"Saldo utama anda telah bertambah sebesar {top_up}. Saldo saat ini: {users[username]['saldo']}")
131            elif saldo == 'tidak':
132                print("Top up saldo dibatalkan.")
133            else:
134                print("Pilihan tidak valid.")
135                print("")
136
137            money = input("Apakah ingin top up E-Money anda? \nTekan (iya/tidak): ").lower()
138            if money == 'iya':
139                try:
140                    top_up_emoney = int(input("Masukkan jumlah top up E-Money: "))
141                    if top_up_emoney <= 0:
142                        print("Jumlah top up E-Money tidak valid.")
143                        continue
144                except ValueError:
145                    print("Input harus berupa angka.")
146                    continue
147
148                if top_up_emoney <= users[username]['saldo']:
149                    users[username]['saldo'] -= top_up_emoney
150                    users[username]['emoney'] = users[username].get("emoney", 0) + top_up_emoney
151                    print(f"E-Money berhasil ditambahkan. Saldo E-Money anda: {users[username]['emoney']}")
152                else:
153                    print("Saldo utama anda tidak mencukupi untuk top up E-Money.")
154            elif money == 'tidak':
155                print("Top up E-Money dibatalkan.")
156            else:
157                print("Pilihan tidak valid.")
158        elif choice == '5':
159            main_menu()
160            print("Logout berhasil.")
161            break
162        else:
163            print("Opsi tidak valid.")
164

```

Gambar 3.1.12

- Fungsi `user_menu(username)` merupakan menu interaktif untuk pengguna biasa(user) setelah login, menu ini memungkinkan pengguna untuk memilih berbagai opsi terkait studio dan keuangan mereka.

```

265
266 def tampil_studios():
267     table = PrettyTable(["ID", "Nama Studio", "Harga"])
268     for studio_id, info in studios.items():
269         table.add_row([studio_id, info['name'], info['price']])
270     print(table)
271

```

Gambar 3.1.13

- Fungsi `tampil_studios()` berfungsi untuk menampilkan daftar studio yang tersedia dalam bentuk table yang rapi.

```

271 def sewa_studio(username):
272     tampil_studios()
273     waktu_sekarang = datetime.now().strftime("%Y/%m/%d %H:%M")
274     try:
275         studio_id = input("Masukkan ID studio yang ingin disewa: ")
276
277         if studio_id not in studios:
278             print("Studio tidak ditemukan.")
279             return
280
281         if studio_id not in transactions:
282             transactions[studio_id] = []
283
284         tahun = int(input("masukan tahun: "))
285         bulan = int(input("masukan bulan: "))
286         if bulan < 1 or bulan > 12:
287             print("Bulan tidak valid. Masukkan antara 1 hingga 12.")
288             return
289         hari = int(input("masukan hari tanggal: "))
290         if hari < 1 or hari > 30:
291             print("Hari tidak valid. Masukkan antara 1 hingga 30.")
292             return
293
294         jam_mulai = int(input("Masukkan jam mulai (misal: 18): "))
295         jam_selesai = int(input("Masukkan jam selesai (misal: 20): "))
296
297         tanggal = f"{tahun}-{bulan:02d}-{hari:02d}"
298         tanggal_cek = datetime.strptime(f"{tanggal} {jam_mulai:02d}:00", "%Y-%m-%d %H:%M")
299         if tanggal_cek < datetime.now():
300             print("Tanggal tidak valid. Tidak boleh di masa lalu.")
301             return
302
303         if jam_mulai <= 0 or jam_selesai <= 0:
304             print("Jam tidak boleh 0 atau kurang.")
305             return
306
307         if jam_mulai >= 24 or jam_selesai > 24:
308             print("Jam hanya valid antara 0 hingga 24.")
309             return
310
311         if jam_mulai >= jam_selesai:
312             print("Jam selesai harus lebih besar dari jam mulai.")
313             return
314
315     except ValueError:
316         print("Input jam harus berupa angka bulat.")
317         return
318

```

Gambar 3.1.14

```

319: try:
320:     with open('invoice.csv', mode='r', newline='') as file:
321:         reader = csv.DictReader(file)
322:         for row in reader:
323:             if row['studio_id'] == str(studio_id) and row['tanggal'] == tanggal:
324:                 existing_jam_mulai = int(row['jam_mulai'])
325:                 existing_jam_selesai = int(row['jam_selesai'])
326:
327:                 if (jam_mulai < existing_jam_selesai and jam_selesai > existing_jam_mulai):
328:                     print(f'Studio {row['nama_studio']} sudah dipesan pada tanggal {tanggal} dari jam {existing_jam_mulai} sampai {existing_jam_selesai}. Silakan pilih waktu lain.")
329:                 return
330: except FileNotFoundError:
331:     print("File 'invoice.csv' tidak ditemukan.")
332:
333: durasi = jam_selesai - jam_mulai
334: total_price = studios[studio_id]['price'] * durasi
335:
336: if users[username]['emoney'] < total_price:
337:     print("Saldo E-Money tidak cukup.")
338:     return
339:
340: users[username]['emoney'] -= total_price
341:
342: transactions[studio_id].append({
343:     'studio': studio_id,
344:     'username': username,
345:     'nama_studio': studios[studio_id]['name'],
346:     'durasi': durasi,
347:     'tanggal': tanggal,
348:     'jam_mulai': jam_mulai,
349:     'jam_selesai': jam_selesai,
350:     'harga_total': total_price,
351:     'tanggal_transaksi': waktu_sekarang
352: })
353:
354: print(f'Studio {studios[studio_id]['name']} berhasil dipesan oleh {username} pada tanggal {tanggal} dari jam {jam_mulai:.2f} sampai {jam_selesai:.2f}. Total harga: Rp{total_price:,.0f}.")
355: save_transactions(transactions)
356: save_users(users)
357: print_invoice(studio_id, username, durasi, tanggal, jam_mulai, jam_selesai, total_price, waktu_sekarang)
358:

```

Gambar 3.1.15

- Fungsi sewa_studio(username) untuk memproses proses penyewaan studio oleh pengguna. Fungsi ini mengharuskan pengguna untuk memilih studio, menentukan waktu sewa, serta memverifikasi beberapa kondisi terkait ketersediaan studio dan saldo pengguna.

```

58:
59: def print_invoice(studio_id, username, hours, tanggal, jam_mulai, jam_selesai, total_price, tanggal_transaksi):
60:     table = PrettyTable()
61:     table.field_names = ["Deskripsi", "Detail"]
62:
63:     table.add_row(["Tanggal Transaksi", tanggal_transaksi])
64:     table.add_row(["Studio ID", studio_id])
65:     table.add_row(["Nama Studio", studios[studio_id]['name']])
66:     table.add_row(["Penyewa", username])
67:     table.add_row(["Durasi (jam)", hours])
68:     table.add_row(["Tanggal Penyewaan", tanggal])
69:     table.add_row(["Jam Mulai", f"{jam_mulai:.2f}"])
70:     table.add_row(["Jam Selesai", f"{jam_selesai:.2f}"])
71:     table.add_row(["Total Harga", f"Rp{total_price:,.0f}"])
72:
73:     print("\n==== INVOICE =====")
74:     print(table)
75:     print("=====\n")
76:

```

Gambar 3.1.16

- Fungsi ini berfungsi untuk mencetak invoice atau bukti transaksi dalam format table yang rapih menggunakan library prettytable. Fungsi ini mengambil berbagai informasi terkait penyewaan studio dan menyusunnya dalam sebuah table yang mudah dibaca oleh pengguna.

```

377
378 def admin_menu():
379     while True:
380         print("\n=====")
381         print("\n==Menu Admin==")
382         print("1. Studio")
383         print("2. Booking")
384         print("3. Isi Saldo")
385         print("4. Daftar User")
386         print("5. Manage User")
387         print("6. Log Out")
388         print("\n=====")
389         choice = input("Pilih opsi: ")
390
391         if choice == '1':
392             admin_menu_studio()
393         elif choice == '2':
394             admin_menu_booking()
395         elif choice == '3':
396             isi_saldo_user()
397         elif choice == '4':
398             tampilkan_data_users()
399         elif choice == '5':
400             manage_user()
401         elif choice == '6':
402             main_menu()
403             print("Logout berhasil.")
404             break
405         else:
406             print("Opsi tidak valid.")
407

```

Gambar 3.1.17

- Fungsi ini sebagai menu utama untuk pengguna dengan role admin. Dalam menu ini, admin dapat memilih berbagai opsi untuk mengelola aplikasi.

```

407
408 def isi_saldo_user():
409     tampilkan_data_users()
410     username = input("Masukkan username pengguna: ")
411     if username not in users:
412         print("Username tidak ditemukan.")
413         return
414     opsi = input(f"apakah anda yakin ingin top up e-money untuk akun {username}? 1/any :")
415     if opsi == '1':
416         try:
417             saldo_add = int(input("Masukkan jumlah saldo e-money yang ingin ditambahkan: "))
418             if saldo_add > 0:
419                 users[username]['emoney'] += saldo_add
420                 print(f"Saldo e-money untuk {username} berhasil ditambahkan. Saldo e-money sekarang: {users[username]['emoney']}")
421             else:
422                 print("Jumlah saldo harus positif.")
423         except ValueError:
424             print("Input tidak valid. Harap masukkan angka.")
425     else:
426         print("adidaw kacaw")
427     return
428

```

Gambar 3.1.18

- Berfungsi untuk menambahkan saldo e-money untuk pengguna tertentu. Fungsi ini meminta input dari admin untuk menentukan pengguna mana

yang akan mendapatkan top up saldo dan berapa jumlah yang akan ditambahkan.

```
429
430 def admin_menu_studio():
431     while True:
432         print("\n=====")
433         print("\n==Menu Admin Studio==")
434         print("1. Tambah Studio")
435         print("2. Tampilkan Studio")
436         print("3. Ubah Studio")
437         print("4. Hapus Studio")
438         print("5. Kembali ke menu awal")
439         print("\n=====")
440         choice = input("Pilih opsi: ")
441
442         if choice == '1':
443             tambah_studio()
444         elif choice == '2':
445             tampil_studios()
446         elif choice == '3':
447             ubah_studio()
448         elif choice == '4':
449             del_studio()
450         elif choice == '5':
451             admin_menu()
452             break
453         else:
454             print("Ops! tidak valid.")
455
```

Gambar 3.1.19

- Fungsi `admin_menu_studio()` ini berfungsi sebagai menu untuk admin yang memungkinkan admin untuk melakukan berbagai operasi terkait studio, seperti menambah, menampilkan, mengubah, atau menghapus studio. Menu ini akan memanggil fungsi-fungsi yang lebih spesifik berdasarkan pilihan admin.

```
455
456 def tambah_studio():
457     name = input("Masukkan nama studio: ")
458     if any(studio['name'].lower() == name.lower() for studio in studios.values()):
459         print("Nama studio sudah ada. Silakan pilih nama studio lain.")
460         return
461
462     try:
463         price = float(input("Masukkan harga sewa per jam: "))
464         if price <= 0:
465             print("Harga sewa tidak valid. Harap masukkan harga yang lebih besar dari 0.")
466             return
467     except ValueError:
468         print("Harga sewa harus berupa angka.")
469         return
470
471     studio_id = len(studios) + 1
472     studios[studio_id] = {'name': name, 'price': price}
473     booking[studio_id] = []
474     print(f"Studio '{name}' berhasil ditambahkan dengan harga sewa Rp{price:,.0f} per jam.")
475     save_studios(studios)
476
```

Gambar 3.1.20

- Fungsi `tambah_studio()` bertujuan untuk memungkinkan admin menambahkan studio baru ke dalam sistem. Fungsi ini bisa digunakan dengan baik dalam aplikasi untuk mempermudah pengelolaan data studio oleh admin.

```

476
477 def ubah_studio():
478     tampil_studios()
479     try:
480         studio_id = int(input("Masukkan ID studio yang ingin diubah: "))
481     except ValueError:
482         print("ID studio harus berupa angka.")
483         return
484
485     if studio_id not in studios:
486         print("ID studio tidak valid.")
487         return
488
489     studio = studios[studio_id]
490     print(f"Studio yang dipilih: {studio['name']} (Harga: Rp{studio['price']:,.0f})")
491     new_name = input("Masukkan nama studio baru: ")
492
493     if any(studio['name'].lower() == new_name.lower() for studio in studios.values()):
494         print("Nama studio sudah ada. Silakan pilih nama studio lain.")
495         return
496
497     try:
498         new_price = float(input("Masukkan harga sewa per jam baru: "))
499         if new_price <= 0:
500             print("Harga sewa tidak valid. Harap masukkan harga yang lebih besar dari 0.")
501             return
502     except ValueError:
503         print("Harga sewa harus berupa angka.")
504         return
505
506     studios[studio_id] = {'name': new_name, 'price': new_price}
507     print(f"Studio berhasil diubah menjadi '{new_name}' dengan harga sewa Rp{new_price:,.0f} per jam.")
508     save_studios(studios)
509

```

Gambar 3.1.21

- Fungsi `ubah_studio()` bertujuan untuk memungkinkan admin mengubah informasi studio yang sudah ada di dalam sistem, seperti nama studio dan harga sewa per jam.

```

509
510 def del_studio():
511     tampil_studios()
512     try:
513         studio_id = int(input("Masukkan ID studio yang ingin dihapus: "))
514     except ValueError:
515         print("ID studio harus berupa angka.")
516         return
517
518     if studio_id not in studios:
519         print("ID studio tidak valid.")
520         return
521
522     if transactions.get(studio_id):
523         print(f"Studio {studios[studio_id]['name']} memiliki transaksi yang belum dibatalkan!")
524         return
525     del studios[studio_id]
526     del booking[studio_id]
527     print("Studio berhasil dihapus!")
528     save_studios(studios)

```

Gambar 3.1.22

- Fungsi `del_studio()` digunakan untuk menghapus studio dari sistem. Fungsi `del_studio()` memberikan cara untuk menghapus studio dari sistem, namun dengan beberapa pengecekan untuk memastikan bahwa studio yang akan dihapus tidak memiliki transaksi yang belum dibatalkan. Fungsi ini memberikan kontrol kepada admin untuk mengelola studio dengan hati-hati, mencegah penghapusan data yang dapat merusak integritas sistem transaksi.


```

#menu admin booking
def admin_menu_booking():
    while True:
        print("\n=====")
        print("\n==Menu Admin Booking==")
        print("1. Daftar Sewa")
        print("2. Tambah Sewa")
        print("3. Hapus Sewa")
        print("4. Kembali ke menu awal")
        print("\n=====")
        choice = input("Pilih opsi: ")

        if choice == '1':
            daftar_sewa()
        elif choice == '2':
            tambah_sewa()
        elif choice == '3':
            hapus_sewa()
        elif choice == '4':
            admin_menu()
            break
        else:
            print("Opsi tidak valid.")

```

Gambar 3.1.23

- Fungsi `admin_menu_booking()` adalah bagian dari menu admin yang memungkinkan admin untuk mengelola pemesanan studio. Fungsi ini memberikan beberapa opsi terkait dengan manajemen sewa studio, seperti melihat daftar sewa, menambah sewa baru, atau menghapus sewa yang sudah ada.

```

def daftar_sewa():
    if not any(transactions.values()):
        print("Tidak ada booking yang tersedia.")
        return

    table = PrettyTable()
    table.field_names = ["Studio ID", "Penyewa", "Nama Studio", "Durasi", "Tanggal", "Jam Mulai", "Jam Selesai", "Harga Total"]

    for studio_id, transaksi in transactions.items():
        for trans in transaksi:
            table.add_row([
                studio_id,
                trans["username"],
                trans["nama_studio"],
                trans["durasi"],
                trans["tanggal"],
                trans["jam_mulai"],
                trans["jam_selesai"],
                trans["harga_total"]
            ])

    print(table)

```

Gambar 3.1.24

- Fungsi `daftar_sewa()` berfungsi untuk menampilkan daftar semua pemesanan studio (booking) yang ada, berdasarkan data yang ada dalam

transactions. Fungsi ini memanfaatkan pustaka PrettyTable untuk menampilkan data dalam format tabel yang lebih rapi dan mudah dibaca.

```
def tambah_sewa():
    tampil_studios()
    username = input("Masukkan Penyewa: ")
    waktu_sekarang = datetime.now().strftime("%Y/%m/%d %H:%M")
    try:
        studio_id = input("Masukkan ID studio yang ingin disewa: ")
        if studio_id not in studios:
            print("Studio tidak ditemukan.")
            return
        if studio_id not in transactions:
            transactions[studio_id] = []
        tahun = int(input("masukan tahun: "))
        bulan = int(input("masukan bulan: "))
        if bulan < 1 or bulan > 12:
            print("Bulan tidak valid. Masukkan antara 1 hingga 12.")
            return
        hari = int(input("masukan hari tanggal: "))
        if hari < 1 or hari > 30:
            print("Hari tidak valid. Masukkan antara 1 hingga 30.")
            return
        jam_mulai = int(input("Masukkan jam mulai (misal: 18): "))
        jam_selesai = int(input("Masukkan jam selesai (misal: 20): "))
        tanggal = f"{tahun}-{bulan:02d}-{hari:02d}"
        tanggal_cek = datetime.strptime(f"{tanggal} {jam_mulai:02d}:00", "%Y-%m-%d %H:%M")
        if tanggal_cek < datetime.now():
            print("Tanggal tidak valid. Tidak boleh di masa lalu.")
            return
        if jam_mulai <= 0 or jam_selesai <= 0:
            print("Jam tidak boleh 0 atau kurang.")
            return
        if jam_mulai >= 24 or jam_selesai > 24:
            print("Jam hanya valid antara 0 hingga 24.")
            return
        if jam_mulai >= jam_selesai:
            print("Jam selesai harus lebih besar dari jam mulai.")
            return
    except ValueError:
        print("Invalid Input.")
    return
```

Gambar 3.1.25

```
try:
    with open('invoice.csv', mode='r', newline='') as file:
        reader = csv.DictReader(file)
        for row in reader:
            if row['studio_id'] == str(studio_id) and row['tanggal'] == tanggal:
                existing_jam_mulai = int(row['jam_mulai'])
                existing_jam_selesai = int(row['jam_selesai'])
                if (jam_mulai < existing_jam_selesai and jam_selesai > existing_jam_mulai):
                    print(f"Studio {row['nama_studio']} sudah dipesan pada tanggal {tanggal} dari jam {existing_jam_mulai} sampai {existing_jam_selesai}. Silakan pilih waktu lain.")
                    return
except FileNotFoundError:
    print("File 'invoice.csv' tidak ditemukan.")

durasi = jam_selesai - jam_mulai
total_price = studios[studio_id]['price'] * durasi

transactions[studio_id].append({
    'studio': studio_id,
    'username': username,
    'nama_studio': studios[studio_id]['name'],
    'durasi': durasi,
    'tanggal': tanggal,
    'jam_mulai': jam_mulai,
    'jam_selesai': jam_selesai,
    'harga_total': total_price,
    'tanggal_transaksi': waktu_sekarang
})

print(f"Studio {studios[studio_id]['name']} berhasil dipesan oleh {username} pada tanggal {tanggal} dari jam {jam_mulai:2f} sampai {jam_selesai:2f}. Total harga: Rp{total_price:,0f}.")
save_transactions(transactions)
print_invoice(studio_id, username, durasi, tanggal, jam_mulai, jam_selesai, total_price, waktu_sekarang)
```

Gambar 3.1.26

- Fungsi tambah_sewa() secara keseluruhan menangani proses pemesanan studio, mulai dari memilih studio, memvalidasi waktu, menghitung harga sewa, memeriksa konflik jadwal, hingga mencetak invoice. Fungsi ini

memastikan bahwa hanya pemesanan yang valid dan tidak bertabrakan yang akan diproses, serta transaksi yang valid akan disimpan ke dalam sistem dan dicetak invoicenya.

```
660
661 def hapus_sewa():
662     username = input("Masukkan username yang ingin dihapus sewanya: ")
663     transaksi_ditemukan = False
664
665     transaksi_terpilih = []
666
667     for studio_id, transaksi in transactions.items():
668         for transaksi in transaksi:
669             if transaksi['username'] == username:
670                 transaksi_terpilih.append((studio_id, transaksi))
671                 transaksi_ditemukan = True
672
673     if transaksi_ditemukan:
674         for studio_id, transaksi in transaksi_terpilih:
675             print("\nTransaksi ditemukan:")
676             print(f"Studio: {transaksi['nama_studio']}")
677             print(f"Tanggal: {transaksi['tanggal']}")
678             print(f"Jam Mulai: {transaksi['jam_mulai']}")
679             print(f"Jam Selesai: {transaksi['jam_selesai']}")
680             print(f"Harga Total: {transaksi['harga_total']}")
681
682             konfirmasi = input("Apakah Anda ingin menghapus transaksi ini? (y/n): ")
683             if konfirmasi.lower() == 'y':
684                 transactions[studio_id].remove(transaksi)
685                 print(f"Transaksi untuk studio {transaksi['nama_studio']} berhasil dihapus.")
686                 save_transactions(transactions)
687                 return
688             else:
689                 print("Transaksi tidak dihapus.")
690     else:
691         print(f"Tidak ditemukan transaksi dengan username {username}.")
692
```

Gambar 3.1.27

- Fungsi `hapus_sewa()` memungkinkan admin untuk menghapus transaksi sewa studio yang telah dilakukan oleh pengguna. Fungsi ini menyediakan antarmuka untuk mencari transaksi berdasarkan username, menampilkan detail transaksi yang ditemukan, serta meminta konfirmasi sebelum menghapus transaksi tersebut.

```
692
693 def tampilkan_data_users():
694     table = PrettyTable(["Username", "Saldo", "emoney", "Role"])
695
696     for username, info in users.items():
697         table.add_row([username, info['saldo'], info['emoney'], info['role']])
698     print(table)
699
```

Gambar 3.1.28

- Fungsi `tampilkan_data_users()` digunakan untuk menampilkan informasi pengguna yang tersimpan dalam sistem, yang termasuk nama pengguna,

saldo, saldo e-money, dan peran mereka. Dengan menggunakan PrettyTable, data pengguna ditampilkan dalam format tabel yang lebih terstruktur dan mudah dibaca, baik oleh admin maupun pengguna yang memiliki akses.

```
700 def manage_user():
701     tampilkan_data_users()
702     username = input("masukan username yang ingin di angkat menjadi admin: ")
703     if username not in users:
704         print("Username tidak ditemukan.")
705         return
706     if users[username]['role'] == 'admin':
707         print(f"{username} sudah merupakan admin")
708     else:
709         users[username]['role'] = 'admin'
710         print(f"Berhasil mengubah status {username} ke admin.")
711         save_users(users)
712
```

Gambar 3.1.29

- Fungsi `manage_user()` digunakan oleh admin untuk mengubah status seorang pengguna menjadi admin. Fungsi ini menampilkan daftar pengguna, memungkinkan admin untuk memilih seorang pengguna, dan kemudian mengubah peran pengguna tersebut jika memenuhi syarat.

```
713
714 if __name__ == "__main__":
715     load_studios()
716     transactions = load_transactions()
717     load_users()
718     main_menu()
```

Gambar 3.1.30

- Bagian kode ini adalah titik awal dari aplikasi yang akan memulai alur eksekusi setelah semua data yang dibutuhkan dimuat. Dengan menggunakan `if __name__ == "__main__":`, program ini dapat dijalankan langsung atau diimpor ke dalam program lain tanpa mengeksekusi kode yang tidak diinginkan.

3.2 Alur Program

```
=====
===Menu Utama===
1. Registrasi
2. Login
3. Exit

=====
Pilih opsi: 1
Masukkan username: sifa
Masukkan password: ***
Akun berhasil dibuat!
Data sifa berhasil ditambahkan.
```

Gambar 3.2.1

```
=====
===Menu Utama===
1. Registrasi
2. Login
3. Exit

=====
Pilih opsi: 1
Masukkan username: sifa
Username sudah ada. Silakan pilih username lain.
```

Gambar 3.2.2

```
===Menu Utama===
1. Registrasi
2. Login
3. Exit

=====
Pilih opsi: 2
Masukkan username: sifa
Masukkan password: *
Username atau password salah. Sisa percobaan: 2
Masukkan username: sif
Masukkan password: ***
Username atau password salah. Sisa percobaan: 1
Masukkan username:
Masukkan password: *
Anda telah mencoba login 3 kali. Program akan keluar.
PS C:\Users\ASUS\latihan>
```

Gambar 3.2.3

```

=====

====Menu User====
1. Tampilkan Studio
2. Sewa Studio
3. Cek Saldo
4. Top up
5. Logout

=====
Pilih opsi: 4
Saldo anda saat ini: 0
Apakah ingin top up saldo anda?
Tekan (iya/tidak): iya
Masukkan jumlah top up saldo anda: 700000
Top up saldo melebihi batas maksimal yaitu 500000

```

Gambar 3.2.4

```

=====

====Menu User====
1. Tampilkan Studio
2. Sewa Studio
3. Cek Saldo
4. Top up
5. Logout

=====
Pilih opsi: 4
Saldo anda saat ini: 0
Apakah ingin top up saldo anda?
Tekan (iya/tidak): iya
Masukkan jumlah top up saldo anda: 200000
Saldo utama anda telah bertambah sebesar 200000. Saldo saat ini: 200000
Apakah ingin top up E-Money anda?
Tekan (iya/tidak): iya
Masukkan jumlah top up E-Money: 250000
Saldo utama anda tidak mencukupi untuk top up E-Money.

```

Gambar 3.2.5

```

=====

====Menu User====
1. Tampilkan Studio
2. Sewa Studio
3. Cek Saldo
4. Top up
5. Logout

=====
Pilih opsi: 4
Saldo anda saat ini: 200000
Apakah ingin top up saldo anda?
Tekan (iya/tidak): tidak
Top up saldo dibatalkan.
Apakah ingin top up E-Money anda?
Tekan (iya/tidak): iya
Masukkan jumlah top up E-Money: 150000
E-Money berhasil ditambahkan. Saldo E-Money anda: 150000

```

Gambar 3.2.6

```

=====

====Menu User====
1. Tampilkan Studio
2. Sewa Studio
3. Cek Saldo
4. Top up
5. Logout

=====
Pilih opsi: 3
Saldo E-Money Anda: 150000

```

Gambar 3.2.7

```
=====
===Menu Utama===
1. Registrasi
2. Login
3. Exit

=====
Pilih opsi: 2
Masukkan username: admin
Masukkan password: *****
Login admin berhasil!
```

Gambar 3.2.8

```
=====
==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Log Out

=====
Pilih opsi: 1
```

Gambar 3.2.9


```
=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 2
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1  | Studio cihuy | 65000 |
| 2  | Studio sigma | 90000 |
| 3  | Studio lope  | 115000|
| 4  | Studio genzy | 135000|
+---+-----+-----+
```

Gambar 3.2.10

```
=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 1
Masukkan nama studio: Studio cihuy
Nama studio sudah ada. Silakan pilih nama studio lain.
```

Gambar 3.2.11

```

=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 1
Masukkan nama studio: Studio kiw
Masukkan harga sewa per jam: 20000
Studio 'Studio kiw' berhasil ditambahkan dengan harga sewa Rp20,000 per jam.

```

Gambar 3.2.12

```

=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 2
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1  | Studio cihuy | 65000 |
| 2  | Studio sigma | 90000 |
| 3  | Studio lope  | 115000 |
| 4  | Studio genzy | 135000 |
| 5  | Studio kiw   | 20000.0 |
+---+-----+-----+

```

Gambar 3.2.13

```
=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 3
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1  | Studio cihuy | 65000 |
| 2  | Studio sigma | 90000 |
| 3  | Studio lope  | 115000|
| 4  | Studio genzy | 135000|
| 5  | Studio kiw   | 20000.0|
+---+-----+-----+
Masukkan ID studio yang ingin diubah: 5
Studio yang dipilih: Studio kiw (Harga: Rp20,000)
Masukkan nama studio baru: Studio gg
Masukkan harga sewa per jam baru: 40000
Studio berhasil diubah menjadi 'Studio gg' dengan harga sewa Rp40,000 per jam.
```

Gambar 3.2.14

```
=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 2
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1  | Studio cihuy | 65000 |
| 2  | Studio sigma | 90000 |
| 3  | Studio lope  | 115000|
| 4  | Studio genzy | 135000|
| 5  | Studio gg    | 40000.0|
+---+-----+-----+
```

Gambar 3.2.15

```

=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 4
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1 | Studio cihuy | 65000 |
| 2 | Studio sigma | 90000 |
| 3 | Studio lope | 115000 |
| 4 | Studio genzy | 135000 |
| 5 | Studio gg | 40000.0 |
+---+-----+-----+
Masukkan ID studio yang ingin dihapus: 5
Studio berhasil dihapus!

=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 2
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1 | Studio cihuy | 65000 |
| 2 | Studio sigma | 90000 |
| 3 | Studio lope | 115000 |
| 4 | Studio genzy | 135000 |
+---+-----+-----+

```

Gambar 3.2.16

```
=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 5

=====

==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Log Out

=====
Pilih opsi: █
```

Gambar 3.2.17

```
=====
====Menu User====
1. Tampilkan Studio
2. Sewa Studio
3. Cek Saldo
4. Top up
5. Logout

=====
Pilih opsi: 5

=====

===Menu Utama===
1. Registrasi
2. Login
3. Exit

=====
Pilih opsi: █
```

Gambar 3.2.18

```
==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: 1

=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
```

Gambar 3.2.19

```
=====
==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 1
Masukkan nama studio: kacaw
Masukkan harga sewa per jam: 5000
Studio 'kacaw' berhasil ditambahkan dengan harga sewa Rp5,000 per jam.

=====
==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: █
```

Gambar 3.2.20

```
=====
==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 2

+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1 | Studio cihuy | 65000 |
| 2 | Studio sigma | 90000 |
| 3 | Studio lope | 115000 |
| 4 | Studio genzy | 135000 |
| 5 | kacaw | 5000 |
+---+-----+-----+

=====
```

Gambar 3.2.21

```
=====
Pilih opsi: 3

+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1 | Studio cihuy | 65000 |
| 2 | Studio sigma | 90000 |
| 3 | Studio lope | 115000 |
| 4 | Studio genzy | 135000 |
| 5 | kacaw | 5000 |
+---+-----+-----+

Masukkan ID studio yang ingin diubah: 5
Studio yang dipilih: kacaw (Harga: Rp5,000)
Masukkan nama studio baru: kuul
Masukkan harga sewa per jam baru: 10000
Studio berhasil diubah menjadi 'kuul' dengan harga sewa Rp10,000 per jam.

=====
```

Gambar 3.2.22


```
=====
Pilih opsi: 4
+---+-----+-----+
| ID | Nama Studio | Harga |
+---+-----+-----+
| 1  | Studio cihuy | 65000 |
| 2  | Studio sigma | 90000 |
| 3  | Studio lope  | 115000|
| 4  | Studio genzy | 135000|
| 5  |      kuul   | 10000 |
+---+-----+-----+
Masukkan ID studio yang ingin dihapus: 5
Studio berhasil dihapus!

=====

==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal
```

Gambar 3.2.23

```
=====
==Menu Admin Studio==
1. Tambah Studio
2. Tampilkan Studio
3. Ubah Studio
4. Hapus Studio
5. Kembali ke menu awal

=====
Pilih opsi: 5

=====

==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: █
```

Gambar 3.2.24

```
==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: 2

=====

==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
```

Gambar 3.2.25

```
=====

==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
Pilih opsi: 1

+-----+-----+-----+-----+-----+-----+-----+-----+
| Studio ID | Penyewa | Nama Studio | Durasi | Tanggal | Jam Mulai | Jam Selesai | Harga Total |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | gg | Studio genzy | 1.0 | 2024-12-01 | 1 | 2 | 135000 |
| 4 | 4 | Studio genzy | 1.0 | 2024-12-02 | 1 | 2 | 135000 |
| 2 | skibidi | Studio sigma | 2.0 | 2024-12-08 | 12 | 14 | 180000 |
+-----+-----+-----+-----+-----+-----+-----+-----+

=====
```

Gambar 3.2.26

```

==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
Pilih opsi: 2
+-----+-----+
| ID | Nama Studio | Harga |
+-----+-----+
| 1 | Studio cihuy | 65000 |
| 2 | Studio sigma | 90000 |
| 3 | Studio lope | 115000 |
| 4 | Studio genzy | 135000 |
+-----+-----+
Masukkan Penyewa: pro
Masukkan ID studio yang ingin disewa: 4
masukan tahun: 2025
masukan bulan: 1
masukan hari tanggal: 12
Masukkan jam mulai (misal: 18): 9
Masukkan jam selesai (misal: 20): 12
Studio Studio genzy berhasil dipesan oleh pro pada tanggal 2025-01-12 dari jam 9.00 sampai 12.00. Total harga: Rp405,000.

===== INVOICE =====
+-----+-----+
| Deskripsi | Detail |
+-----+-----+
| Tanggal Transaksi | 2024/11/09 22:46 |
| Studio ID | 4 |
| Nama Studio | Studio genzy |
| Penyewa | pro |
| Durasi (jam) | 3 |
| Tanggal Penyewaan | 2025-01-12 |
| Jam Mulai | 9.00 |
| Jam Selesai | 12.00 |
| Total Harga | Rp405,000 |
+-----+-----+
=====

```

Gambar 3.2.27

```

==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
Pilih opsi: 3

+-----+-----+-----+-----+-----+-----+-----+-----+
| Studio ID | Penyewa | Nama Studio | Durasi | Tanggal | Jam Mulai | Jam Selesai | Harga Total |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | gg | Studio genzy | 1.0 | 2024-12-01 | 1 | 2 | 135000 |
| 4 | 4 | Studio genzy | 1.0 | 2024-12-02 | 1 | 2 | 135000 |
| 4 | pro | Studio genzy | 3.0 | 2025-01-12 | 9 | 12 | 405000 |
| 2 | skibidi | Studio sigma | 2.0 | 2024-12-08 | 12 | 14 | 180000 |
+-----+-----+-----+-----+-----+-----+-----+-----+

Masukkan username yang ingin dihapus sewanya: skibidi

Transaksi ditemukan:
Studio: Studio sigma
Tanggal: 2024-12-08
Jam Mulai: 12
Jam Selesai: 14
Harga Total: 180000
Apakah Anda ingin menghapus transaksi ini? (y/n): y
Transaksi untuk studio Studio sigma berhasil dihapus.

=====

==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
Pilih opsi: 1

+-----+-----+-----+-----+-----+-----+-----+-----+
| Studio ID | Penyewa | Nama Studio | Durasi | Tanggal | Jam Mulai | Jam Selesai | Harga Total |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 4 | gg | Studio genzy | 1.0 | 2024-12-01 | 1 | 2 | 135000 |
| 4 | 4 | Studio genzy | 1.0 | 2024-12-02 | 1 | 2 | 135000 |
| 4 | pro | Studio genzy | 3.0 | 2025-01-12 | 9 | 12 | 405000 |
+-----+-----+-----+-----+-----+-----+-----+-----+

=====

==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
Pilih opsi: 

```

Gambar 3.2.28

```
=====
==Menu Admin Booking==
1. Daftar Sewa
2. Tambah Sewa
3. Hapus Sewa
4. Kembali ke menu awal

=====
Pilih opsi: 4

=====

==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: █
```

Gambar 3.2.29

```
=====
Pilih opsi: 4
+-----+-----+-----+-----+
| Username | Saldo | emoney | Role |
+-----+-----+-----+-----+
| gg       | 0     | 1000000000000000 | user |
| admin    | 0     | 0         | admin |
| sigma    | 0     | 0         | user  |
| skibidi  | 0     | 320000    | user  |
+-----+-----+-----+-----+

=====

==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: █
```

Gambar 3.2.30

```

5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: 5
+-----+-----+-----+-----+
| Username | Saldo | emoney | Role |
+-----+-----+-----+-----+
| gg       | 0     | 1000000000000000 | user |
| admin    | 0     | 0         | admin |
| sigma    | 0     | 0         | user  |
| skibidi  | 0     | 320000    | user  |
+-----+-----+-----+-----+

masukan username yang ingin di angkat menjadi admin: skibidi
Berhasil mengubah status skibidi ke admin.

=====

==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: 4
+-----+-----+-----+-----+
| Username | Saldo | emoney | Role |
+-----+-----+-----+-----+
| gg       | 0     | 1000000000000000 | user |
| admin    | 0     | 0         | admin |
| sigma    | 0     | 0         | user  |
| skibidi  | 0     | 320000    | admin |
+-----+-----+-----+-----+

=====

```

Gambar 3.2.31


```
=====
==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: 6
[22, 34, 54, 62, 67, 71]
cari indeks: 54
element ditemukan pada index 2

=====
```

Gambar 3.2.32

```
=====
==Menu Admin==
1. Studio
2. Booking
3. Isi Saldo
4. Daftar User
5. Manage User
6. Search (bonus)
7. Log Out

=====
Pilih opsi: 7

=====
===Menu Utama===
1. Registrasi
2. Login
3. Exit

=====
Pilih opsi: 3
Terima kasih telah menggunakan aplikasi.
PS D:\code\codengab\ini-projek-1>
```

Gambar 3.2.33

BAB IV

Penutup

4.1 Kesimpulan

Dari laporan yang telah disusun, dapat disimpulkan bahwa penggunaan bahasa pemrograman Python dalam pengembangan aplikasi menawarkan berbagai keuntungan, seperti kemudahan dalam pemrograman, sintaks yang mudah dipahami, dan fleksibilitas dalam pengembangan berbagai jenis aplikasi. Dalam konteks praktikum ini, mahasiswa telah diberikan pemahaman tentang bagaimana Python dapat digunakan untuk merancang aplikasi yang memiliki nilai praktis, salah satunya adalah aplikasi *Booking Penyewaan Studio Fotografi*. Aplikasi ini bertujuan untuk mengatasi masalah yang timbul dari sistem manual yang masih digunakan oleh beberapa toko atau penyedia jasa, khususnya dalam bidang penyewaan studio fotografi.

Melalui aplikasi ini, diharapkan proses transaksi, pemesanan, dan pelayanan dapat berjalan lebih efisien, terorganisir, serta memberikan kenyamanan baik bagi pelanggan maupun pemilik studio. Aplikasi yang dibangun menggunakan Python ini juga memanfaatkan potensi kemajuan teknologi untuk meningkatkan kualitas layanan dan mengurangi risiko kesalahan manusia yang sering terjadi dalam sistem manual.

4.2 Saran

Sebagai langkah pengembangan lebih lanjut, disarankan untuk menambahkan fitur-fitur lain yang dapat meningkatkan pengalaman pengguna, seperti integrasi dengan sistem pembayaran online, pengingat otomatis untuk pelanggan, atau sistem ulasan dan rating untuk kualitas studio.

Pastikan aplikasi dilengkapi dengan fitur keamanan yang memadai, seperti enkripsi data pelanggan dan sistem otentikasi yang kuat, untuk menjaga kerahasiaan dan integritas data pengguna.

Untuk meningkatkan penerimaan aplikasi, penting untuk memberikan pelatihan atau tutorial penggunaan aplikasi kepada pemilik studio dan pelanggan agar mereka dapat memanfaatkan sistem ini secara maksimal.

Lampiran

Tabel Kontribusi

Nama	Kontribusi	Bagian
Zahra Aurellya Herdiansyah (2409116062)	Konsep, Coding, Laporan, Pengecekan, flowchart	1. Referensi konsep program 2. Coding 3. Menjelaskan program di Laporan 4. Mencari bug dan <i>error</i>
Satria Rajawali Ektya Antara (2409116067)	Coding, konsep, pengembangan	1. Logika program 2. Mengecek alur program Coding 3. <i>Finishing</i> Laporan
Irvan Alif (2409116071)	Konsep, Pengembangan, flowchart	1. Konsep program 2. Flowchart program 3. Pengembangan program 4. Mengecek alur program

Tabel 1.1 Tabel Kontribusi