

**LAPORAN AKHIR**  
**DASAR DASAR PEMROGRAMAN**  
**SISTEM PENJUALAN TIKET EVENT OLAHRAGA**



**Di Susun Oleh:**

**Kelompok 1 / C /2024**

**Muhammad Fakhri Al-Kautsar: (2409116081)**

**Raihan Fariz Novanto: (2409116083)**

**Moreno Ferdinand Farhantino: (2409116097)**

**Ahmad Samsul Arifin: (2409116113)**

**Asisten Laboratorium**

**Muhammad Aryaputra Wirawan**  
**(2309116027)**

**Athira Fahmi**  
**(2309116010)**

**PROGRAM STUDI SISTEM INFORMASI**  
**FAKULTAS TEKNIK**  
**UNIVERSITAS MULAWARMAN**  
**2024**

## **KATA PENGANTAR**

Puji syukur kami panjatkan ke hadirat Tuhan Yang Maha Esa, yang telah memberikan rahmat dan karunia-Nya sehingga kami dapat menyusun dan menyelesaikan laporan akhir yang berjudul “ sistem penjualan tiket event olahraga ” ini dapat kami selesaikan tepat waktu.

Pembuatan laporan ini bertujuan untuk memenuhi projek akhir dari praktikum yang dilaksanakan ,serta diharapkan mampu untuk membantu kita dalam menguasai bahasa pemrograman python.Dengan disusunnya laporan ini ,tidak lupa kami ucapkan terimakasih sebesar besarnya kepada :

1. Mba Athira Fahmi selaku pembimbing
2. Seluruh asisten laboratorium praktikum dasar – dasar pemrograman 2024
3. Orang tua yang selalu support
4. Dan teman – teman yang berkontribusi dalam membantu kami dalam menyelesaikan program ini

Kami menyadari bahwa laporan ini masih jauh dari kesempurnaan. Oleh karena itu, kami terbuka terhadap kritik dan saran yang membangun dari pembaca demi perbaikan penulisan di masa mendatang. Semoga makalah ini dapat bermanfaat bagi pembaca.

Samarinda, 06 November 2024

Kelompok 1

## DAFTAR ISI

<b>KATA PENGANTAR.....</b>	<b>i</b>
<b>DAFTAR ISI.....</b>	<b>ii</b>
<b>DAFTAR GAMBAR.....</b>	<b>iii</b>
<b>DAFTAR TABEL .....</b>	<b>vi</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1. Deskripsi Masalah.....	1
1.2. Rumusan Masalah.....	1
1.3. Batasan Masalah .....	1
1.4. Tujuan .....	2
1.5. Manfaat .....	2
<b>BAB II PERANCANGAN .....</b>	<b>3</b>
2.1. Analisis Program.....	3
2.2. Flowchart .....	3
<b>BAB III HASIL DAN PEMBAHASAN .....</b>	<b>9</b>
3.1. Implementasi program.....	9
3.2. Alur Program.....	13
3.3. Source Kode .....	45
<b>BAB IV PENUTUP .....</b>	<b>87</b>
4.1. Kesimpulan .....	87
4.2. Saran.....	87
<b>DAFTAR PUSTAKA .....</b>	<b>89</b>
<b>LAMPIRAN.....</b>	<b>90</b>

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Flowchart 1.1 .....	4
<b>Gambar 2.2</b> Flowchart 1.2 .....	5
<b>Gambar 2.3</b> Flowchart 1.3 .....	6
<b>Gambar 2.4</b> Flowchart 1.4 .....	6
<b>Gambar 2.5</b> flowchart 1.5 .....	7
<b>Gambar 2.6</b> Flowchart 1.6 .....	7
<b>Gambar 2.7</b> Flowchart 1.7 .....	7
<b>Gambar 2.8</b> Flowchart 1.8 .....	8
<b>Gambar 3.1</b> Implementasi Json .....	9
<b>Gambar 3.2</b> Implementasi Json .....	9
<b>Gambar 3.3</b> Implementasi Function(def) .....	9
<b>Gambar 3.4</b> Implementasi While Loop .....	10
<b>Gambar 3.5</b> Implementasi If Else .....	11
<b>Gambar 3.6</b> Implementasi Library dan Pwinput .....	11
<b>Gambar 3.7</b> Implementasi Try Except dan Crud.....	12
<b>Gambar 3.8</b> Alur Program Json .....	13
<b>Gambar 3.9</b> Alur Program Daftar Pengguna 1 .....	14
<b>Gambar 3.10</b> Alur Program Daftar Pengguna 2 .....	14
<b>Gambar 3.11</b> Alur Program Login Pengguna.....	15
<b>Gambar 3.12</b> Alur Program Login Admin .....	15
<b>Gambar 3.13</b> Alur Program Bola 1.....	16
<b>Gambar 3 14</b> Alur Program Bola 2.....	16
<b>Gambar 3.15</b> Alur Program Badminton 1 .....	16
<b>Gambar 3.16</b> Alur Program Badminton 2 .....	16
<b>Gambar 3.17</b> Alur Program Basket 1 .....	17
<b>Gambar 3.18</b> Alur Program Basket 2 .....	17
<b>Gambar 3.19</b> Alur Program Voli 1 .....	17
<b>Gambar 3.20</b> Alur Program Voli 2 .....	17
<b>Gambar 3.21</b> Alur Program Futsal 1 .....	18

<b>Gambar 3.22</b> Alur Program Futsal 2 .....	18
<b>Gambar 3.23</b> Alur Program Baris.....	18
<b>Gambar 3.24</b> Alur Program Pilihan Admin.....	19
<b>Gambar 3.25</b> Alur Program Menu Cari Tiket 1 .....	20
<b>Gambar 3.26</b> Alur Program Menu Cari Tiket 2 .....	20
<b>Gambar 3.27</b> Alur Program Sorting Tiket.....	21
<b>Gambar 3.28</b> Alur Program After Menu Admin 1 .....	21
<b>Gambar 3.29</b> Alur Program After Menu Admin 2 .....	22
<b>Gambar 3.30</b> Alur Program Menu After User 1 .....	22
<b>Gambar 3.31</b> Alur Program After Menu User 2.....	23
<b>Gambar 3.32</b> Alur Program Menu.....	23
<b>Gambar 3.33</b> Alur Program Menu User .....	24
<b>Gambar 3.34</b> Alur Program Tampilan Menu .....	25
<b>Gambar 3.35</b> Alur Program Menampilkan Menu Transaksi .....	25
<b>Gambar 3.36</b> Alur Program Menu Create 1 .....	26
<b>Gambar 3.37</b> Alur Program Menu Create 2 .....	27
<b>Gambar 3.38</b> Alur Program Create 3.....	28
<b>Gambar 3.39</b> Alur Program Menu Update 1 .....	29
<b>Gambar 3.40</b> Alur Program Menu Update 2 .....	30
<b>Gambar 3.41</b> Alur Program Menu Update 3 .....	30
<b>Gambar 3.42</b> Alur Program Menu Update 4 .....	31
<b>Gambar 3.43</b> Alur Program Delete 1 .....	31
<b>Gambar 3.44</b> Alur Program Delete 2.....	32
<b>Gambar 3.45</b> Alur Program Delete 3.....	32
<b>Gambar 3.46</b> Alur Program Menu Delete 4 .....	33
<b>Gambar 3.47</b> Alur Program Menu Pengguna .....	33
<b>Gambar 3.48</b> Alur Program Menu Pelanggan 1 .....	34
<b>Gambar 3.49</b> Alur Program Menu Pelanggan 2 .....	34
<b>Gambar 3.50</b> Alur Program Struk Pembelian 1.....	35
<b>Gambar 3.51</b> Alur Program Struk Pembelian 2.....	35
<b>Gambar 3.52</b> Alur Program Proses Pembelian 1 .....	36
<b>Gambar 3.53</b> Alur Program Proses Pembelian 2.....	37

<b>Gambar 3.54</b> Alur Program Proses Pembelian 3.....	38
<b>Gambar 3.55</b> Alur Program Transaksi 1.....	39
<b>Gambar 3.56</b> Alur Program Transaksi 2.....	39
<b>Gambar 3.57</b> Alur Program Transaksi 3.....	40
<b>Gambar 3.58</b> Alur Program Melihat Saldo 1.....	41
<b>Gambar 3.59</b> Alur Program Melihat Saldo 2.....	41
<b>Gambar 3.60</b> Alur Program Top-Up 1 .....	42
<b>Gambar 3.61</b> Alur Program Top-Up 2 .....	43
<b>Gambar 3.62</b> Alur Program Top-Up 3 .....	43
<b>Gambar 3.63</b> Alur Program Menu Cari Pelanggan 1 .....	44
<b>Gambar 3.64</b> Alur Program Menu Cari Pelanggan 2 .....	44
<b>Gambar 3.65</b> Alur Program Menu Main 1 .....	45
<b>Gambar 3.66</b> Alur Program Menu Main 2 .....	45

## DAFTAR TABEL

<b>Tabel 3.1</b> Source Kode .....	86
<b>Tabel 3.2</b> Tabel Kontribusi.....	90

# **BAB I**

## **PENDAHULUAN**

### **1.1. Deskripsi Masalah**

Saat ini, banyak organisasi dan penyelenggara event olahraga mengalami kesulitan dalam mengelola penjualan tiket secara efektif. Sistem penjualan tiket konvensional sering kali terbatas dalam hal efisiensi, skalabilitas, dan kenyamanan, baik bagi penyelenggara maupun pelanggan. Contohnya Ketika tiket dijual melalui loket fisik, banyak calon penonton harus menghadapi antrian panjang. Selain itu, platform yang tidak bisa menangani volume permintaan tinggi dapat mengalami penurunan performa atau bahkan crash, yang menyebabkan keterlambatan dan ketidakpuasan pelanggan. Sistem yang kurang aman rentan terhadap penjualan tiket palsu atau duplikasi tiket. Hal ini dapat merugikan pelanggan dan mengurangi kepercayaan terhadap penyelenggara acara. Selain itu, kurangnya sistem verifikasi tiket digital dapat menyebabkan tiket ilegal dijual di pasar gelap.

Dari permasalahan berikut, kelompok kami telah merancang sebuah program penjualan tiket event olahraga yang dapat membantu para pelanggan agar lebih mudah dalam pembelian tiket ini. Dengan adanya program penjualan tiket event olahraga ini, para penyelenggara juga dapat mencapai penjualan yang maksimal.

### **1.2. Rumusan Masalah**

Berdasarkan latar belakang di atas, maka dapat diketahui rumusan masalah sebagai berikut:

1. Bagaimana merancang sebuah program penjualan tiket event olahraga berbasis python?
2. Apakah program dapat menciptakan sistem kerja yang lebih efisien ?
3. Apakah program mampu dioperasikan dengan mudah oleh penyelenggara maupun pelanggan ?

### **1.3. Batasan Masalah**



Dari permasalahan yang telah dijabarkan diatas, terdapat batasan masalah dalam penyusunan laporan program ini:

1. Program aplikasi ini masih berupa program saja dan belum dapat diterapkan sepenuhnya dalam kehidupan nyata.
2. Program ini hanya merupakan program sederhana.
3. Program ini dibuat menggunakan bahasa pemrograman python.
4. Program ini membutuhkan database

#### **1.4. Tujuan**

Adapun tujuan dari pembuatan laporan ini, yaitu :

1. Untuk menjelaskan alur dan tampilan program system penjualan tiket event olahraga
2. Untuk mempermudah pelanggan dalam melakukan transaksi pembelian tiket event olahraga secara online
3. Untuk Meningkatkan sistem keamanan kepada penyelenggara dan pelanggan
4. Untuk mengasah kemampuan mahasiswa dalam menyusun dan merancang program

#### **1.5. Manfaat**

Adapun manfaat dari pembuatan laporan ini, yaitu :

1. Untuk meningkatkan kemampuan menganalisis suatu program bagi mahasiswa untuk kasus yang berbeda nantinya
2. Meningkatkan keamanan dalam transaksi pembelian
3. Mempermudah sistem layanan dalam proses transaksi pembelian
4. Dapat mengetahui keefektifan penggunaan suatu sistem kerja dengan menggunakan bahasa pemrograman pyhton

## **BAB II**

### **PERANCANGAN**

#### **2.1. Analisis Program**

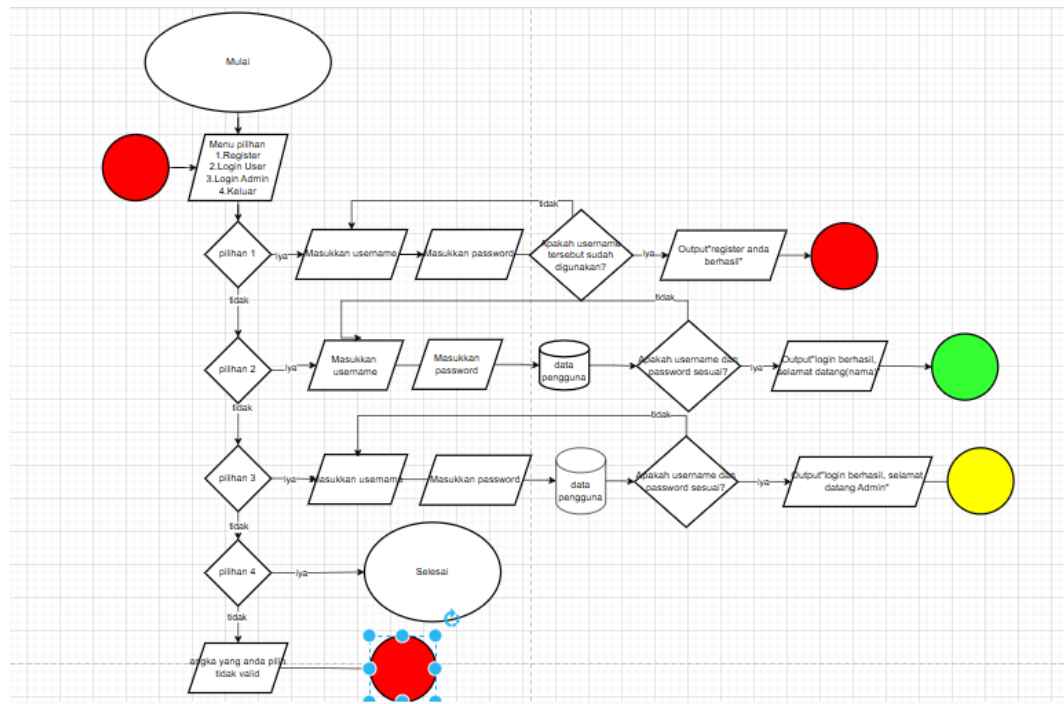
Program ini adalah program sederhana yang di rancang dalam bahasa pemrograman pyhton.Dibawah ini analisis program ini beserta library yang di gunakan :

##### **1. Import Library:**

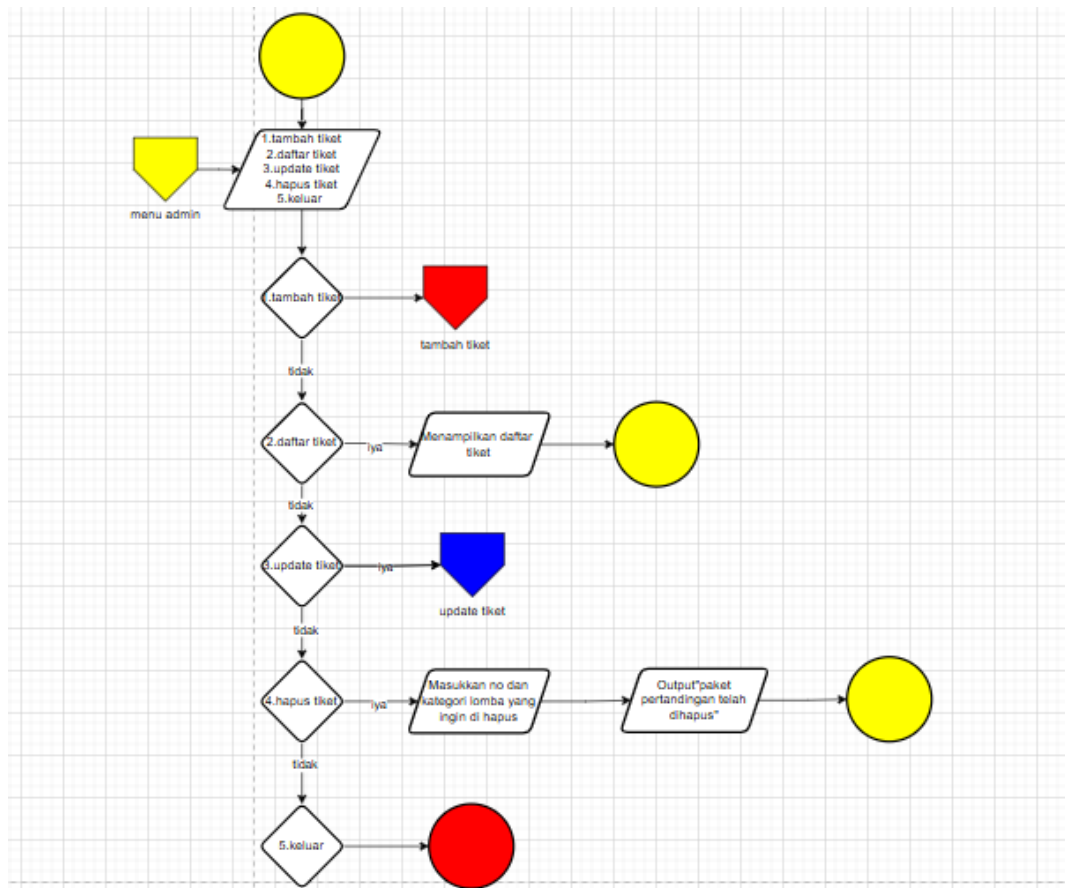
- Prettytable : Digunakan untuk membuat tampilan tabel yang rapi dalam konsol.
- Os : Digunakan untuk membersihkan layar konsol dengan perintah "cls" atau "clear" tergantung pada sistem operasi.
- Pwinput : Menggunakan library ini untuk mengambil kata sandi dari pengguna tanpa menampilkan teks yang dimasukkan (untuk keamanan).
- Json : Digunakan untuk membaca dan menulis data dalam format JSON.
- Datetime : Library ini digunakan untuk bekerja dengan tanggal dan waktu dalam Python.
- Itemgetter : Digunakan dalam pengurutan data, misalnya mengurutkan data transaksi berdasarkan waktu atau jumlah transaksi.
- Uuid : Digunakan untuk menghasilkan ID unik, misalnya untuk ID transaksi atau ID tiket, yang memastikan setiap entri adalah unik.
- Time : Di gunakan untuk memberikan jeda atau penundaan dalam proses tertentu, serta mencatat waktu dalam format detik.

#### **2.2. Flowchart**

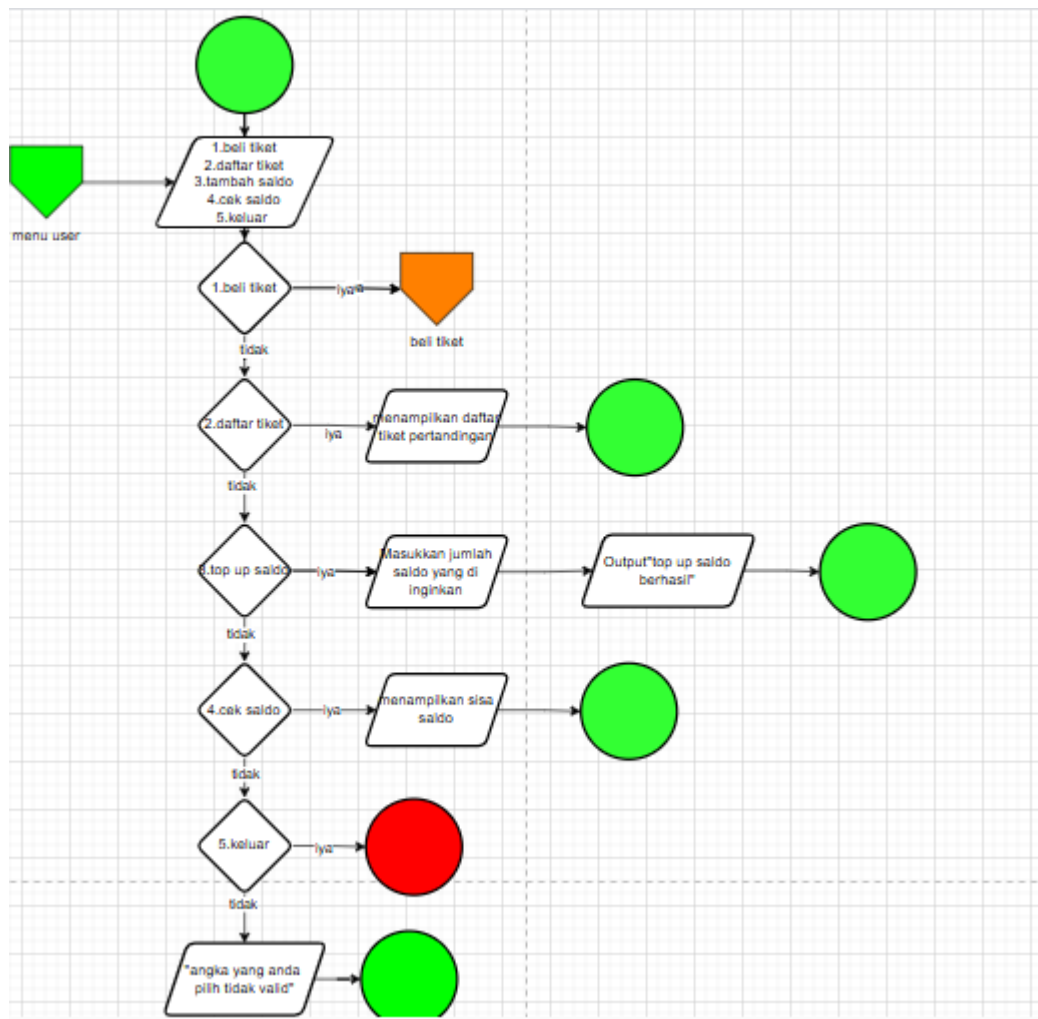
Berikut flowchart dari program “sistem penjualan tiket event olahraga” yang telah di buat :



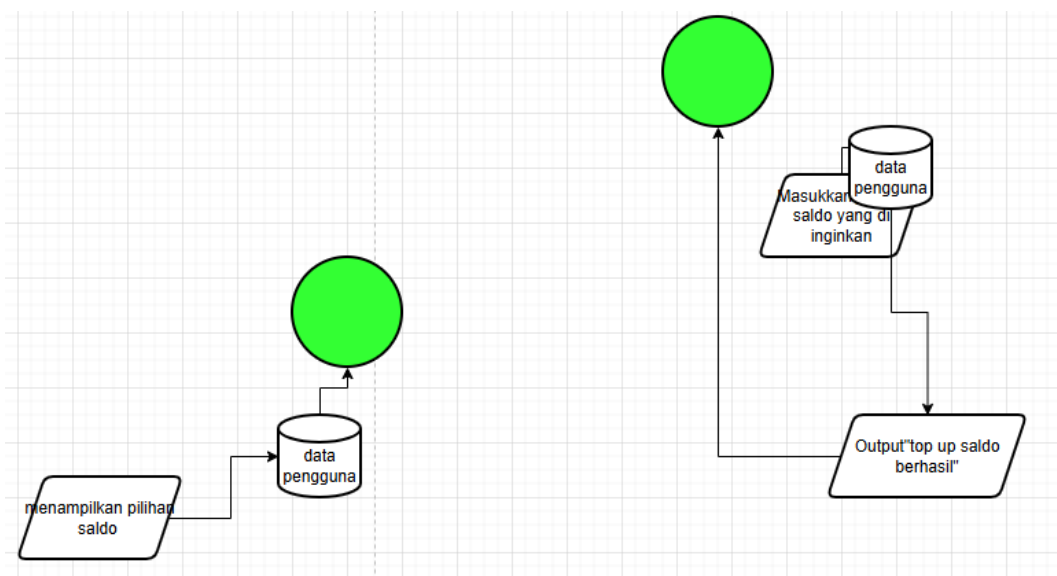
**Gambar 2.1** Flowchart 1.1



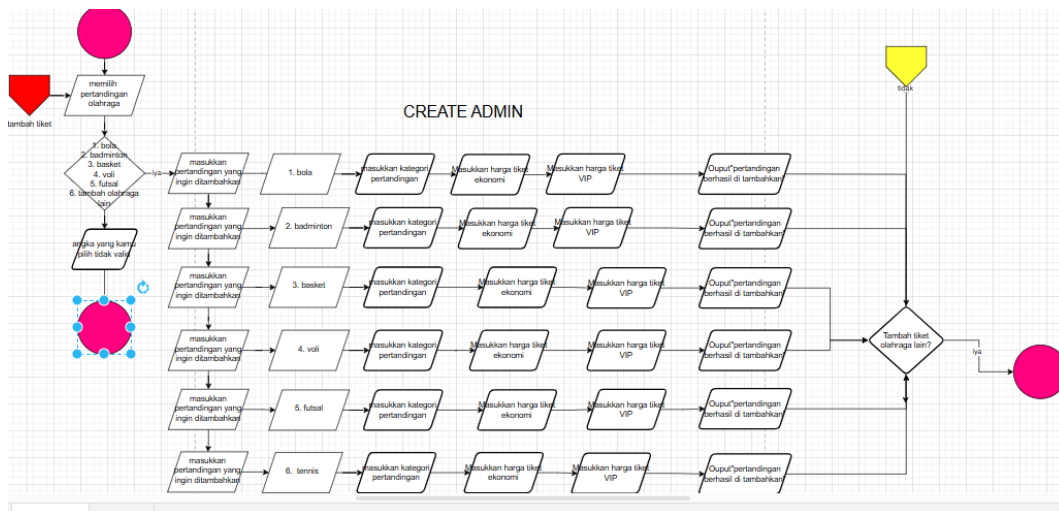
**Gambar 2.2** Flowchart 1.2



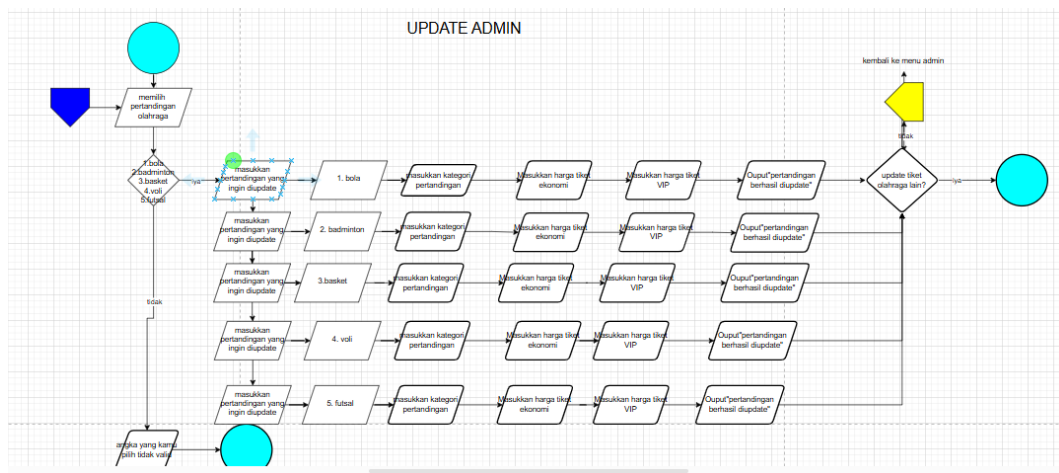
**Gambar 2.3** Flowchart 1.3



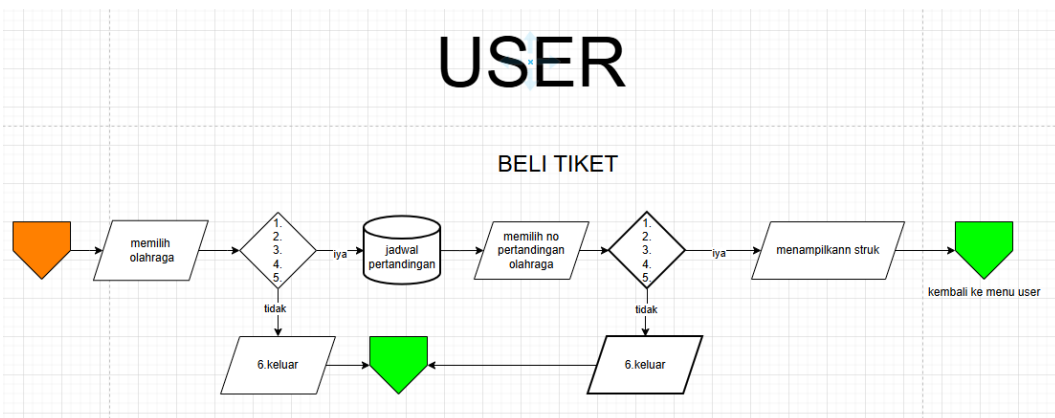
**Gambar 2.4** Flowchart 1.4



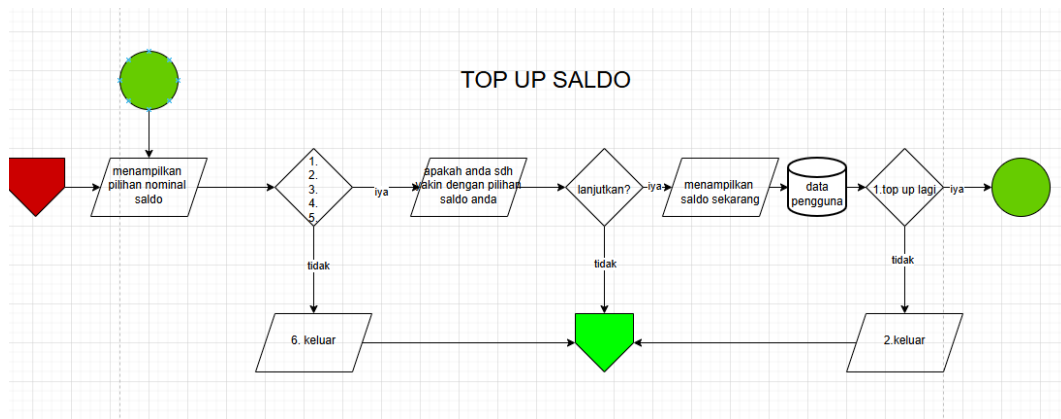
Gambar 2.5 flowchart 1.5



Gambar 2.6 Flowchart 1.6



Gambar 2.7 Flowchart 1.7



**Gambar 2.8** Flowchart 1.8

## BAB III

### HASIL DAN PEMBAHASAN

#### 3.1. Implementasi program

##### a) Json Dictionary

```
PathjsonUser = r"C:\Koding\TestCode\regist.json"

# =====
terserah = os.path.abspath(os.curdir)

json_s = "C:\\Koding\\TestCode\\schedule.json"
```

**Gambar 3.1** Implementasi Json

```
{
  {
    "Nama User": "halo",
    "Pw User": "halo",
    "Saldo": 50200000
  },
  {
    "Nama User": "ikan",
    "Pw User": "ikan",
    "Saldo": 50000
  },
}
```

**Gambar 3.2** Implementasi Json

Digunakan untuk menyimpan data secara dinamis

##### b) Function (def)

```
def futsal():
    tabel = PrettyTable()
    tabel.title = "PERTANDINGAN FUTSAL"
    tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga Tiket VIP"]
    nomor = 1
    for kategori in data['Kategori']:
        if kategori['Nama Kategori'] == 'Futsal':
            for match in kategori['Jadwal']:
                tabel.add_row([f"{nomor}", match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'],
                                match['Harga Tiket Ekonomi'], match['Harga Tiket VIP']])
                nomor += 1
```

**Gambar 3.3** Implementasi Function(def)



Def digunakan untuk melakukan untuk memanggil variable.

c) While Loop

```
def delete():
    print("====HAPUS TIKET====")
    print("")
    while True:
        try:
            show_menu_transaksi()
            kategorilomba = input("Masukkan cabang lomba: ").strip()

            if kategorilomba == "1":
                while True:
                    try:
                        bola()
                        nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
                        for kategori in data['Kategori']:
                            if kategori['Nama Kategori'] == 'Bola':
                                if 0 <= nomor < len(kategori['Jadwal']):
                                    tiketlama = kategori['Jadwal'].pop(nomor)
                                    save_schedule(data)
                                    print(f"Paket {tiketlama['Pertandingan']} telah dihapus")
                                    input("TEKAN ENTER UNTUK MELANJUTKAN...")
                                    khususAtmint()
                                    break
                                else:
                                    print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
                            else:
                                continue
                        except (ValueError, KeyboardInterrupt):
                            print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
```

**Gambar 3.4** Implementasi While Loop

Menggunakan while loop agar program dapat di eksekusi berkali kali

d) If Else

```

def afterMenuUser():
    while True:
        try:
            tabel = PrettyTable()
            tabel.clear_rows()
            tabel.title = "SILAHKAN PILIH"
            tabel.field_names = ["NO", "PILIHAN"]
            tabel.add_row(["[1]", "Cari Tiket"])
            tabel.add_row(["[2]", "Sorting tiket (Berdasarkan abjad)"])
            tabel.add_row(["[3]", "Keluar"])
            print(tabel)
            pilihan = int(input("Masukkan pilihan 1/2/3: "))
            if pilihan == 1:
                cari_tiket()
                print("=====")
                print("  1. Cari Tiket Lain")
                print("  2. Kembali")
                print("=====")
                pilihan2 = int(input("Masukkan pilihan 1/2: "))
                if pilihan2 == 1:
                    cari_tiket()
                elif pilihan2 == 2:
                    menu_pelanggan()
                else:
                    print("PENCARIAN ANDA TIDAK VALID, MASUKKAN PILIHAN YANG VALID")

```

**Gambar 3.5** Implementasi If Else

Dalam program ini kami menggunakan If Else untuk pengambilan Keputusan

e) Library PrettyTabel dan PwInput

```

def show_menu(is_admin=True):
    os.system("cls")
    tabel = PrettyTable()
    tabel.clear_rows()
    tabel.title = "====CABANG LOMBA===="
    tabel.field_names = ["NO", "PILIHAN"]
    tabel.add_row(["[1]", "BOLA"])
    tabel.add_row(["[2]", "BADMINTON"])
    tabel.add_row(["[3]", "BASKET"])
    tabel.add_row(["[4]", "VOLI"])
    tabel.add_row(["[5]", "FUTSAL"])
    tabel.add_row(["[6]", "KEMBALI"])
    print(tabel)

```

**Gambar 3.6** Implementasi Library dan Pwinput

PrettyTable berfungsi untuk membuat tabel yang menjadi lebih rapi dan pw input untuk menyamarkan password yang diinput.

f) Try Except dan Crud

```
def delete():
    print("====HAPUS TIKET====")
    print("")
    while True:
        try:
            show_menu_transaksi()
            kategoriLomba = input("Masukkan cabang lomba: ").strip()

            if kategoriLomba == "1":
                while True:
                    try:
                        bola()
                        nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
                        for kategori in data['Kategori']:
                            if kategori['Nama Kategori'] == 'Bola':
                                if 0 <= nomor < len(kategori['Jadwal']):
                                    tiketLama = kategori['Jadwal'].pop(nomor)
                                    save_schedule(data)
                                    print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
                                    input("TEKAN ENTER UNTUK MELANJUTKAN...")
                                    khususAtmint()
                                    break
                                else:
                                    print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
                            else:
                                continue
                        except (ValueError, KeyboardInterrupt):
                            print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
                    except (ValueError, KeyboardInterrupt):
                        print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
```

**Gambar 3.7** Implementasi Try Except dan Crud

Kami menggunakan Try Except untuk handling error yang terjadi pada program, dan Create, Read, Update dan Delete merupakan fungsi utama yang terdapat dalam program ini untuk menu admin.

### 3.2. Alur Program

```
10
11 PathjsonUser = r"C:\Koding\TestCode\regist.json"
12
13 # =====
14 terserah = os.path.abspath(os.curdir)
15
16 json_s = "C:\\Koding\\TestCode\\schedule.json"
17
18 def load_schedule():
19     with open(json_s, 'r') as data_schedule:
20         return json.load(data_schedule) #load => loads,
21         # return data
22
23 def save_schedule(data):
24     with open(json_s, 'w') as data_schedule:
25         json.dump(data, data_schedule, indent=4)
26
27 data = load_schedule()
28 # =====
29
30 def loadDataUser():
31     try:
32         with open(PathjsonUser, "r") as jsonUser:
33             return json.load(jsonUser)
34     except FileNotFoundError:
35         return {"Nama": [], "Password": [], "saldo": []}
36
37 def savedataUser(data):
38     with open(PathjsonUser, "w") as jsonUser:
39         json.dump(data, jsonUser, indent=4)
40
```

**Gambar 3.8** Alur Program Json

Pada baris ke 11-16 kami membuat penyimpanan regist di dalam json.

Pada baris ke 30-35 kami membuat fungsi def memuat data pengguna untuk memuat data ke json.

Pada baris ke 37-39 kami membuat fungsi def menyimpan data pengguna untuk menyimpan data ke json.

```

43 def daftaruser():
44     users = loadDataUser ()
45     while True:
46         try:
47             while True:
48                 namaUser = input("Masukkan nama (min 3 karakter): ").strip()
49                 if len(namaUser) < 3:
50                     print("Nama pengguna harus minimal 3 karakter!")
51                 elif len(namaUser) > 50:
52                     print("Nama pengguna tidak boleh lebih dari 50 karakter!")
53                 elif any(user["Nama User"] == namaUser for user in users):
54                     print(f"{namaUser} sudah terdaftar.")
55                 else:
56                     break
57             while True:
58                 passwordUser = pwininput.pwininput("Password (min 6 karakter): ", "**")
59                 if len(passwordUser) < 6:
60                     print("Password harus minimal 6 karakter!")
61                 else:
62                     break
63             new_user = {
64                 "Nama User": namaUser ,
65                 "Pw User": passwordUser ,
66                 "Saldo": 0
67             }
68             users.append(new_user)
69             savedataUser (users)
70
71

```

**Gambar 3.9** Alur Program Daftar Pengguna 1

```

71     savedataUser (users)
72     print('
73 +=====+
74 |          REGISTER ANDA BERHASIL          |
75 +=====+')
76     break
77 except ValueError:
78     print("Terdapat kesalahan, silahkan coba lagi!")
79 except KeyboardInterrupt:
80     print("Tolong untuk tidak menekan Ctrl dan C secara bersamaan!")
81

```

**Gambar 3.10** Alur Program Daftar Pengguna 2

Pada baris ke 43-80 kami membuat fungsi def daftar pengguna untuk pengguna mendaftar

```

82 def loginUser():
83     global username
84     users_data = loadDataUser()
85     while True:
86         try:
87             if not users_data:
88                 print("No user data found.")
89                 return False
90
91                 print("==== LOGIN USER ====")
92                 username = input("Masukkan username: ")
93                 password = pwinput.pwinput("Masukkan password: ", "*")
94
95                 for user in users_data:
96                     if user["Nama User"] == username:
97                         if user["Pw User"] == password:
98                             print("Login berhasil!")
99                             return True
100                     else:
101                         print("Password salah!")
102                         return False
103                         break
104                 print("Username tidak ditemukan!")
105                 return False
106         except(ValueError, KeyboardInterrupt):
107             print("Tolong untuk tidak menekan Ctrl dan C secara bersamaan!")

```

**Gambar 3.11** Alur Program Login Pengguna

Pada baris ke 82-107 kami membuat fungsi def login pengguna untuk pengguna masuk ke program

```

109 def loginAdmin():
110     admin = {
111         "RehanGokil": "Admin#1234",
112         "SamsulBetawi": "SayaJawa",
113         "MorenoPakPak": "PakPak123",
114         "FakriAyamPramuka": "MintaSambelnyaMas"
115     }
116     username = input("Masukkan username Anda: ")
117     password = pwinput.pwinput("Masukkan Password Anda: ", '*')
118     if username in admin and password == admin[username]:
119         print("+=====+")
120         print("|                SELAMAT DATANG ADMIN                |")
121         print("+=====+")
122         return True
123     else:
124         print("Username atau Password salah, Masukkan dengan Benar!!")
125         return False

```

**Gambar 3.12** Alur Program Login Admin

Pada baris ke 109-125 kami membuat fungsi def login admin untuk admin masuk ke program

```
128 def bola():
129     tabel = PrettyTable()
130     tabel.title = "PERTANDINGAN BOLA"
131     tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga Tiket VIP"]
132     nomor = 1
133     for kategori in data['Kategori']:
134         if kategori['Nama Kategori'] == 'Bola':
135             for match in kategori['Jadwal']:
136                 tabel.add_row([f"({nomor})", match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket']}", f"Rp{match['Harga Tiket VIP']}")])
137                 nomor += 1
138     print(tabel)
139
```

**Gambar 3.13** Alur Program Bola 1

```
136 match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi']:,}", f"Rp{match['Harga Tiket VIP']:,}")])
137
```

**Gambar 3.14** Alur Program Bola 2

Pada baris ke 128-136 kami membuat fungsi def bola untuk pelanggan dapat memilih pertandingan bola yang akan di beli dengan format yang tersedia yaitu: kategori,tanggal,waktu,dan jenis harga tiket

```
140 def badminton():
141     tabel = PrettyTable()
142     tabel.title = "PERTANDINGAN BADMINTON"
143     tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga Tiket VIP"]
144     nomor = 1
145     for kategori in data['Kategori']:
146         if kategori['Nama Kategori'] == 'Badminton':
147             for match in kategori['Jadwal']:
148                 tabel.add_row([f"({nomor})", match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket']}", f"Rp{match['Harga Tiket VIP']}")])
149                 nomor += 1
150     print(tabel)

```

**Gambar 3.15** Alur Program Badminton 1

```
148 match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi']:,}", f"Rp{match['Harga Tiket VIP']:,}")])
149
```

**Gambar 3.16** Alur Program Badminton 2

Pada baris ke 140-148 kami membuat fungsi def badminton untuk pelanggan dapat memilih pertandingan badminton yang akan di beli dengan format yang tersedia yaitu: kategori,tanggal,waktu,dan jenis harga tiket

```

152 def basket():
153     tabel = PrettyTable()
154     tabel.title = "PERTANDINGAN BASKET"
155     tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga Tiket VIP"]
156     nomor = 1
157     for kategori in data['Kategori']:
158         if kategori['Nama Kategori'] == 'Basket':
159             for match in kategori['Jadwal']:
160                 tabel.add_row([f"{{nomor}}", match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket
161                 nomor += 1
162                 print(tabel)

```

**Gambar 3.17** Alur Program Basket 1

```

159     match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi':.}]", f"Rp{match['Harga Tiket VIP':.}]]
161

```

**Gambar 3.18** Alur Program Basket 2

Pada baris ke 152-160 kami membuat fungsi def basket untuk pelanggan dapat memilih pertandingan basket yang akan di beli dengan format yang tersedia yaitu: kategori,tanggal,waktu,dan jenis harga tiket

```

164 def voli():
165     tabel = PrettyTable()
166     tabel.title = "PERTANDINGAN VOLI"
167     tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga Tiket VIP"]
168     nomor = 1
169     for kategori in data['Kategori']:
170         if kategori['Nama Kategori'] == 'Voli':
171             for match in kategori['Jadwal']:
172                 tabel.add_row([f"{{nomor}}", match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket
173                 nomor += 1
174                 print(tabel)
175

```

**Gambar 3.19** Alur Program Voli 1

```

172     match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi':.}]", f"Rp{match['Harga Tiket VIP':.}]]
173

```

**Gambar 3.20** Alur Program Voli 2

Pada baris ke 164-174 kami membuat fungsi def voli untuk pelanggan dapat memilih pertandingan voli yang akan di beli dengan format yang tersedia yaitu: kategori,tanggal,waktu,dan jenis harga tiket



```

176 def futsal():
177     tabel = PrettyTable()
178     tabel.title = "PERTANDINGAN FUTSAL"
179     tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga Tiket VIP"]
180     nomor = 1
181     for kategori in data["Kategori"]:
182         if kategori["Nama Kategori"] == 'Futsal':
183             for match in kategori["Jadwal"]:
184                 tabel.add_row([f"{nomor}", match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket
185                 nomor += 1
186                 print(tabel)

```

**Gambar 3.21** Alur Program Futsal 1

```

184 match['Pertandingan'], match['Kategori'], match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi':,}]", f"Rp{match['Harga Tiket VIP':,}]")

```

**Gambar 3.22** Alur Program Futsal 2

Pada baris ke 176-186 kami membuat fungsi def futsal untuk pelanggan dapat memilih pertandingan futsal yang akan di beli dengan format yang tersedia yaitu: kategori,tanggal,waktu,dan jenis harga tiket

```

188 def baris():
189     print(" ")
190     print("=====+")
191     print(" ")

```

**Gambar 3.23** Alur Program Baris

Pada baris ke 188-191 kami membuat fungsi baris untuk mempercantik program

```

193 def khususAtmint():
194     while True:
195         try:
196             os.system("cls")
197             tabel = PrettyTable()
198             tabel.clear_rows()
199             tabel.title = "PILIHAN ADMIN"
200             tabel.field_names = ["NO", "PILIHAN"]
201             tabel.add_row(["1", "Menu Tiket"])
202             tabel.add_row(["2", "Tambah Tiket"])
203             tabel.add_row(["3", "Update Tiket"])
204             tabel.add_row(["4", "Hapus Tiket"])
205             tabel.add_row(["5", "Keluar"])
206             print(tabel)
207             pilihan = input("Masukkan pilihan 1/2/3/4: ")
208
209             if pilihan == '1':
210                 menu()
211             elif pilihan == '2':
212                 create()
213             elif pilihan == '3':
214                 update()
215             elif pilihan == '4':
216                 delete()
217             elif pilihan == '5':
218                 main()
219             else:
220                 print("Maaf, pilihan tidak ada di menu, masukkan pilihan yang benar")
221                 break
222         except (ValueError, KeyboardInterrupt):
223             print("MOHON MASUKKAN DATA YANG VALID DAN JANGAN MENEKAN CTRL + C!!")

```

**Gambar 3.24** Alur Program Pilihan Admin

Pada baris 193-223 kami membuat fungsi khusus admin untuk memilih pilihan yang tersedia yaitu crud

```

225 def cari_tiket():
226     os.system("cls")
227     print("\n+=====+")
228     print("|          CARI TIKET          |")
229     print("+=====+")
230
231     try:
232         keyword = input("Masukkan kata kunci pencarian: ").lower()
233
234         found = False
235         table = PrettyTable()
236         table.field_names = ["Kategori", "Pertandingan", "Jenis", "Waktu", "Tiket Ekonomi", "Tiket VIP"]
237
238         for kategori in data['Kategori']:
239             kategori_nama = kategori['Nama Kategori']
240             for jadwal in kategori['Jadwal']:
241                 if (keyword in kategori_nama.lower() or
242                     keyword in jadwal['Pertandingan'].lower()):
243                     table.add_row([
244                         kategori_nama,
245                         jadwal['Pertandingan'],
246                         jadwal['Kategori'],
247                         jadwal['Tanggal/Waktu'],
248                         f"Rp {jadwal['Harga Tiket Ekonomi']:,}",
249                         f"Rp {jadwal['Harga Tiket VIP']:,}"
250                     ])
251             found = True
252
253         if found:
254             print("\nHasil Pencarian:")
255             print(table)
256         else:
257             print("\nTidak ditemukan hasil yang cocok dengan kata kunci pencarian.")

```

**Gambar 3.25** Alur Program Menu Cari Tiket 1

```

257
258     except KeyboardInterrupt:
259         print("\nPencarian dibatalkan.")
260     except Exception as e:
261         print(f"\nTerjadi kesalahan: {str(e)}")

```

**Gambar 3.26** Alur Program Menu Cari Tiket 2

Pada baris 225-261 kami membuat fungsi cari tiket yaitu untuk memilih pilihan tiket yang tersedia dengan beberapa kategori pertandingan

```

263 def sorting_tiket():
264     semua_pertandingan = []
265     for kategori in data['Kategori']:
266         for jadwal in kategori['Jadwal']:
267             semua_pertandingan.append({
268                 'Kategori': kategori['Nama Kategori'],
269                 'Pertandingan': jadwal['Pertandingan'],
270                 'Kategori_Pertandingan': jadwal['Kategori'],
271                 'Tanggal/Waktu': jadwal['Tanggal/Waktu'],
272                 'Harga Tiket Ekonomi': jadwal['Harga Tiket Ekonomi'],
273                 'Harga Tiket VIP': jadwal['Harga Tiket VIP']
274             })
275
276     semua_pertandingan.sort(key=itemgetter('Pertandingan'))
277
278     tabel = PrettyTable()
279     tabel.field_names = ["Kategori", "Pertandingan", "Kategori Pertandingan", "Tanggal/Waktu", "Harga Ekonomi", "Harga VIP"]
280     for pertandingan in semua_pertandingan:
281         tabel.add_row([
282             pertandingan['Kategori'],
283             pertandingan['Pertandingan'],
284             pertandingan['Kategori_Pertandingan'],
285             pertandingan['Tanggal/Waktu'],
286             f"Rp{pertandingan['Harga Tiket Ekonomi']:,}",
287             f"Rp{pertandingan['Harga Tiket VIP']:,}"
288         ])
289     print(tabel)

```

**Gambar 3.27** Alur Program Sorting Tiket

Pada baris 263-289 kami membuat fungsi sorting tiket yaitu untuk mengurutkan pertandingan berdasarkan urutan huruf abjad

```

291 def afterMenuAdmin():
292     while True:
293         try:
294             tabel = PrettyTable()
295             tabel.clear_rows()
296             tabel.title = "SILAHKAN PILIH"
297             tabel.field_names = ["NO", "PILIHAN"]
298             tabel.add_row(["1", "Cari Tiket"])
299             tabel.add_row(["2", "Sorting tiket (Berdasarkan abjad)"])
300             tabel.add_row(["3", "Keluar"])
301             print(tabel)
302             pilihan = int(input("Masukkan pilihan 1/2/3: "))
303             if pilihan == 1:
304                 cari_tiket()
305                 print("=====")
306                 print("    1. Cari Tiket Lain" )
307                 print("    2. Kembali" )
308                 print("=====")
309                 pilihan2 = int(input("Masukkan pilihan 1/2: "))
310                 if pilihan2 == 1:
311                     cari_tiket()
312                 elif pilihan2 == 2:
313                     khususAtmint()
314                 else:
315                     print("PENCARIAN ANDA TIDAK VALID, MASUKKAN PILIHAN YANG VALID")
316
317             elif pilihan == 2:
318                 sorting_tiket()
319                 print("=====")
320                 print( input("Tekan enter untuk kembali..."))
321                 print("=====")
322                 khususAtmint()

```

**Gambar 3.28** Alur Program After Menu Admin 1

```

324 elif pilihan == 3:
325     khususAtmint()
326     break
327 else:
328     print("Pilihan anda tidak ada, mohon masukkan pilihan dengan benar!!")
329 except (ValueError, KeyboardInterrupt):
330     print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

```

**Gambar 3.29** Alur Program After Menu Admin 2

Pada baris 291-330 kami membuat fungsi after menu admin yaitu untuk menyediakan menu interaktif untuk admin yaitu dengan pilihan : cari tiket, sorting tiket, dan keluar

```

332 def afterMenuUser():
333     while True:
334         try:
335             tabel = PrettyTable()
336             tabel.clear_rows()
337             tabel.title = "SILAHKAN PILIH"
338             tabel.field_names = ["NO", "PILIHAN"]
339             tabel.add_row (["1", "Cari Tiket"])
340             tabel.add_row (["2", "Sorting tiket (Berdasarkan abjad)"])
341             tabel.add_row (["3", "Keluar"])
342             print(tabel)
343             pilihan = int(input("Masukkan pilihan 1/2/3: "))
344             if pilihan == 1:
345                 cari_tiket()
346                 print("=====")
347                 print("  1. Cari Tiket Lain" )
348                 print("    2. Kembali" )
349                 print("=====")
350                 pilihan2 = int(input("Masukkan pilihan 1/2: "))
351                 if pilihan2 == 1:
352                     cari_tiket()
353                 elif pilihan2 == 2:
354                     menu_pelanggan()
355                 else:
356                     print("PENCARIAN ANDA TIDAK VALID, MASUKKAN PILIHAN YANG VALID")
357             elif pilihan == 2:
358                 sorting_tiket()
359                 print("=====")
360                 print( input("Tekan enter untuk kembali..."))
361                 print("=====")
362                 menu_pelanggan()
363

```

**Gambar 3.30** Alur Program Menu After User 1

```

365 elif pilihan == 3:
366     menu_pelanggan()
367     break
368 else:
369     print("Pilihan anda tidak ada, mohon masukkan pilihan dengan benar!!")
370 except (ValueError, KeyboardInterrupt):
371     print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

```

**Gambar 3.31** Alur Program After Menu User 2

Pada baris 332-371 kami membuat fungsi after menu pengguna yaitu untuk menyediakan menu interaktif untuk pengguna yaitu dengan pilihan : cari tiket, sorting tiket, dan keluar

```

373 def menu():
374     os.system("cls")
375
376     bola()
377
378     baris()
379
380     badminton()
381
382     baris()
383
384     basket()
385
386     baris()
387
388     voli()
389
390     baris()
391
392     futsal()
393
394     afterMenuAdmin()

```

**Gambar 3.32** Alur Program Menu

Pada baris 373-394 kami membuat fungsi menu yaitu untuk menampilkan tampilan utama program

```

396 def menuUser():
397     os.system("cls")
398
399     bola()
400
401     baris()
402
403     badminton()
404
405     baris()
406
407     basket()
408
409     baris()
410
411     voli()
412
413     baris()
414
415     futsal()
416
417     afterMenuUser()
418

```

**Gambar 3.33** Alur Program Menu User

Pada baris 396-417 kami membuat fungsi menu user yaitu untuk menampilkan tampilan utama program

```

419 def show_menu(is_admin=True):
420     os.system("cls")
421     tabel = PrettyTable()
422     tabel.clear_rows()
423     tabel.title = "====CABANG LOMBA===="
424     tabel.field_names = ["NO", "PILIHAN"]
425     tabel.add_row(["1", "BOLA"])
426     tabel.add_row(["2", "BADMINTON"])
427     tabel.add_row(["3", "BASKET"])
428     tabel.add_row(["4", "VOLI"])
429     tabel.add_row(["5", "FUTSAL"])
430     tabel.add_row(["6", "KEMBALI"])
431     print(tabel)
432
433     pilihan = input("Masukkan pilihan (1-6): ")
434     if pilihan == "6":
435         if is_admin:
436             khususAtmint()
437         else:
438             menu_pelanggan()
439         return True
440     return False

```

**Gambar 3.34** Alur Program Tampilan Menu

Pada baris 419-440 kami membuat fungsi tampilan menu yaitu untuk menampilkan beberapa jenis pertandingan olahraga

```

442 def show_menu_transaksi():
443     tabel = PrettyTable()
444     tabel.clear_rows()
445     tabel.title = "====CABANG LOMBA===="
446     tabel.field_names = ["NO", "PILIHAN"]
447     tabel.add_row(["1", "BOLA"])
448     tabel.add_row(["2", "BADMINTON"])
449     tabel.add_row(["3", "BASKET"])
450     tabel.add_row(["4", "VOLI"])
451     tabel.add_row(["5", "FUTSAL"])
452     tabel.add_row(["6", "KEMBALI"])
453     print(tabel)

```

**Gambar 3.35** Alur Program Menampilkan Menu Transaksi



Pada baris 442-453 kami membuat fungsi tampilan menu transaksi yaitu untuk menampilkan beberapa jenis pertandingan olahraga yang tersedia untuk melakukan transaksi

```
456 def create():
457     os.system("cls")
458     while True:
459         try:
460             print("====TAMBAH TIKET====")
461             tabel = PrettyTable()
462             tabel.clear_rows()
463             tabel.title = "MAU NAMBAH TIKET NONTON APA NIH?"
464             tabel.field_names = ["NO", "PILIHAN"]
465             tabel.add_row(["1", "BOLA"])
466             tabel.add_row(["2", "BADMINTON"])
467             tabel.add_row(["3", "BASKET"])
468             tabel.add_row(["4", "VOLI"])
469             tabel.add_row(["5", "FUTSAL"])
470             tabel.add_row(["6", "KEMBALI"])
471             print(tabel)
472             pilihanCreate = int(input("Masukkan pilihan 1/2/3/4/5/6: "))
473
474             if pilihanCreate in [1, 2, 3, 4, 5]:
475                 kategori_map = {1: 'Bola', 2: 'Badminton', 3: 'Basket', 4: 'Voli', 5: 'Futsal'}
476                 kategori_nama = kategori_map[pilihanCreate]
477
478                 print(f"====TAMBAH TIKET {kategori_nama.upper()}====")
479
480                 # Validasi Pertandingan
481                 while True:
482                     pertandingan = input("Masukkan pertandingan yang ingin ditambahkan: ").strip()
483                     if pertandingan and len(pertandingan) >= 3:
484                         break
485                     else:
486                         print("Nama pertandingan tidak boleh kosong dan minimal 3 karakter!")
487
```

**Gambar 3.36** Alur Program Menu Create 1

```

488 # Validasi Kategori
489 while True:
490     kategori = input("Masukkan kategori pertandingan: ").strip()
491     if kategori and len(kategori) >= 2:
492         break
493     else:
494         print("Kategori tidak boleh kosong dan minimal 2 karakter!")
495
496 # Validasi Tanggal
497 while True:
498     tanggal = input("Masukkan tanggal pertandingan: ").strip()
499     if tanggal and len(tanggal) >= 5:
500         break
501     else:
502         print("Tanggal tidak boleh kosong dan minimal 5 karakter!")
503
504 # Validasi Harga Tiket Ekonomi
505 while True:
506     try:
507         hargaEkonomi = int(input("Masukkan Harga Tiket Ekonomi: "))
508         if hargaEkonomi > 0:
509             break
510         else:
511             print("Harga tiket harus lebih dari 0!")
512     except ValueError:
513         print("Masukkan angka yang valid!")
514
515 # Validasi Harga Tiket VIP
516 while True:
517     try:
518         hargaVIP = int(input("Masukkan Harga Tiket VIP: "))

```

**Gambar 3.37** Alur Program Menu Create 2

```

519         if hargaVIP > 0:
520             break
521         else:
522             print("Harga tiket harus lebih dari 0!")
523     except ValueError:
524         print("Masukkan angka yang valid!")
525
526     for kategori_data in data['Kategori']:
527         if kategori_data['Nama Kategori'] == kategori_nama:
528             kategori_data['Jadwal'].append({
529                 "Pertandingan": pertandingan,
530                 "Kategori": kategori,
531                 "Tanggal/Waktu": tanggal,
532                 "Harga Tiket Ekonomi": hargaEkonomi,
533                 "Harga Tiket VIP": hargaVIP
534             })
535             save_schedule(data)
536             print(f"Pertandingan {pertandingan} berhasil ditambahkan")
537             break
538         else:
539             print(f"Kategori {kategori_nama} tidak ditemukan")
540
541     khususAtmint()
542 elif pilihanCreate == 6:
543     khususAtmint()
544     break
545 else:
546     print("Input anda tidak valid, silahkan masukkan pilihan yang tersedia!")
547 except (ValueError, KeyboardInterrupt):
548     print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

```

**Gambar 3.38** Alur Program Create 3

Pada baris 456-548 kami membuat fungsi membuat yaitu untuk membuat beberapa jenis pertandingan olahraga yang baru dengan pilihan :jenis pertandingan, kategori,tanggal,dan harga

```

550 def update():
551     os.system("cls")
552     print("====UPDATE TIKET====")
553     print("")
554     while True:
555         try:
556             if show_menu(is_admin=True):
557                 return
558             kategoriLomba = input("Masukkan cabang lomba (1-5): ").strip()
559
560             kategori_map = {
561                 "1": "Bola",
562                 "2": "Badminton",
563                 "3": "Basket",
564                 "4": "Voli",
565                 "5": "Futsal"
566             }
567
568             if kategoriLomba in kategori_map:
569                 kategori_nama = kategori_map[kategoriLomba]
570
571                 if kategoriLomba == "1":
572                     bola()
573                 elif kategoriLomba == "2":
574                     badminton()
575                 elif kategoriLomba == "3":
576                     basket()
577                 elif kategoriLomba == "4":
578                     voli()
579                 elif kategoriLomba == "5":
580                     futsal()
581

```

**Gambar 3.39** Alur Program Menu Update 1

```

582     nomor = int(input("Masukkan nomor yang ingin diubah: ")) - 1
583     for kategori in data['Kategori']:
584         if kategori['Nama Kategori'] == kategori_nama:
585             if 0 <= nomor < len(kategori['Jadwal']):
586                 match = kategori['Jadwal'][nomor]
587
588                 while True:
589                     pertandingan = input("Pertandingan baru: ").strip()
590                     if pertandingan and len(pertandingan) >= 3:
591                         break
592                     else:
593                         print("Nama pertandingan tidak boleh kosong dan minimal 3 karakter!")
594
595                 while True:
596                     kategori_ = input("Kategori baru: ").strip()
597                     if kategori_ and len(kategori_) >= 2:
598                         break
599                     else:
600                         print("Kategori tidak boleh kosong dan minimal 2 karakter!")
601
602                 while True:
603                     tanggal = input("Tanggal/Waktu baru: ").strip()
604                     if tanggal and len(tanggal) >= 5:
605                         break
606                     else:
607                         print("Tanggal tidak boleh kosong dan minimal 5 karakter!")
608
609                 while True:
610                     try:
611                         hargaekonomi = int(input("Harga Tiket Ekonomi baru: "))
612                         if hargaekonomi > 0:

```

**Gambar 3.40** Alur Program Menu Update 2

```

613         break
614     else:
615         print("Harga tiket harus lebih dari 0!")
616     except ValueError:
617         print("Masukkan angka yang valid!")
618
619     while True:
620         try:
621             hargavip = int(input("Harga Tiket VIP baru: "))
622             if hargavip > 0:
623                 break
624             else:
625                 print("Harga tiket harus lebih dari 0!")
626         except ValueError:
627             print("Masukkan angka yang valid!")
628
629         match['Pertandingan'] = pertandingan
630         match['Kategori'] = kategori_
631         match['Tanggal/Waktu'] = tanggal
632         match['Harga Tiket Ekonomi'] = hargaekonomi
633         match['Harga Tiket VIP'] = hargavip
634
635         save_schedule(data)
636         print(f"Data ke-{nomor+1} telah diperbarui")
637     else:
638         print("Nomor pertandingan tidak valid")
639         break
640     else:
641         print("Kategori tidak ditemukan")
642
643     khususAtmint()

```

**Gambar 3.41** Alur Program Menu Update 3

```

644         break
645     else:
646         print("Input tidak valid. Masukkan angka 1-5.")
647     except(ValueError, KeyboardInterrupt):
648         print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

```

**Gambar 3.42** Alur Program Menu Update 4

Pada baris 550-648 kami membuat fungsi update atau menambahkan yaitu untuk menambahkan beberapa jenis pertandingan olahraga yang baru dengan pilihan :jenis pertandingan, kategori,tanggal,dan harga

```

650 def delete():
651     print("====HAPUS TIKET====")
652     print("")
653     while True:
654         try:
655             show_menu_transaksi()
656             kategoriLomba = input("Masukkan cabang lomba: ").strip()
657
658             if kategoriLomba == "1":
659                 while True:
660                     try:
661                         bola()
662                         nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
663                         for kategori in data['Kategori']:
664                             if kategori['Nama Kategori'] == 'Bola':
665                                 if 0 <= nomor < len(kategori['Jadwal']):
666                                     tiketLama = kategori['Jadwal'].pop(nomor)
667                                     save_schedule(data)
668                                     print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
669                                     input("TEKAN ENTER UNTUK MELANJUTKAN...")
670                                     khususAtmint()
671                                     break
672                         except(ValueError, KeyboardInterrupt):
673                             print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
674
675             elif kategoriLomba == "2":
676                 while True:
677                     try:
678                         badminton()
679                         nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
680                         for kategori in data['Kategori']:
681                             if kategori['Nama Kategori'] == 'Badminton':

```

**Gambar 3.43** Alur Program Delete 1

```

682         if 0 <= nomor < len(kategori['Jadwal']):
683             tiketLama = kategori['Jadwal'].pop(nomor)
684             save_schedule(data)
685             print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
686             input("TEKAN ENTER UNTUK MELANJUTKAN...")
687             khususAtmint()
688             break
689     except(ValueError, KeyboardInterrupt):
690         print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
691
692     elif kategorilomba == "3":
693         while True:
694             try:
695                 basket()
696                 nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
697                 for kategori in data['Kategori']:
698                     if kategori['Nama Kategori'] == 'Basket':
699                         if 0 <= nomor < len(kategori['Jadwal']):
700                             tiketLama = kategori['Jadwal'].pop(nomor)
701                             save_schedule(data)
702                             print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
703                             input("TEKAN ENTER UNTUK MELANJUTKAN...")
704                             khususAtmint()
705                             break
706             except(ValueError, KeyboardInterrupt):
707                 print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
708
709     elif kategorilomba == "4":
710         while True:
711             try:
712                 voli()

```

**Gambar 3.44** Alur Program Delete 2

```

713         nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
714         for kategori in data['Kategori']:
715             if kategori['Nama Kategori'] == 'Voli':
716                 if 0 <= nomor < len(kategori['Jadwal']):
717                     tiketLama = kategori['Jadwal'].pop(nomor)
718                     save_schedule(data)
719                     print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
720                     input("TEKAN ENTER UNTUK MELANJUTKAN...")
721                     khususAtmint()
722                     break
723             except(ValueError, KeyboardInterrupt):
724                 print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
725
726     elif kategorilomba == "5":
727         while True:
728             try:
729                 futsal()
730                 nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
731                 for kategori in data['Kategori']:
732                     if kategori['Nama Kategori'] == 'Futsal':
733                         if 0 <= nomor < len(kategori['Jadwal']):
734                             tiketLama = kategori['Jadwal'].pop(nomor)
735                             save_schedule(data)
736                             print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
737                             input("TEKAN ENTER UNTUK MELANJUTKAN...")
738                             khususAtmint()
739             except(ValueError, KeyboardInterrupt):
740                 print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")
741
742     else:
743         print("Input anda tidak valid, silahkan masukkan cabang lomba yang tersedia!")

```

**Gambar 3.45** Alur Program Delete 3

```

745     except(ValueError, KeyboardInterrupt):
746         print("TOLONG MASUKKAN INPUT YANG VALID DAN JANGAN MEMENCET CTRL + C!")
747         break
748

```

**Gambar 3.46** Alur Program Menu Delete 4

Pada baris 650-747 kami membuat fungsi delete atau menghapus yaitu untuk menghapus beberapa jenis pertandingan olahraga yang ada dengan pilihan : Nomor,jenis pertandingan, kategori,dan jadwal

```

751     def tampilkanTiket():
752         os.system("cls")
753         menuUser()
754         afterMenuUser()
755
756     def transaksi():
757         os.system("cls")
758         transaksi2()
759
760     def lihatSaldo():
761         os.system("cls")
762         lihatSaldo3()
763
764     def topUp():
765         os.system("cls")
766         topUp4()
767
768     def cari():
769         os.system("cls")
770         cari5()
771

```

**Gambar 3.47** Alur Program Menu Pengguna

Pada baris 751-770 kami membuat fungsi menampilkan tiket,transaksi,lihat saldo,top-up,cari yaitu menampilkan,membayar,dan mencari beberapa jenis pertandingan olahraga yang ada dengan pilihan : Nomor,jenis pertandingan, kategori,dan jadwal. Kemudian untuk menu Topu-up yaitu untuk kita mengisi saldo kita dan untuk yang melihat saldo itu berguna untuk melihat saldo kita sebelum membayar apakah cukup untuk membeli tiket tersebut



```

773 def menu_pelanggan():
774     while True:
775         os.system("cls" if os.name == "nt" else "clear")
776         print("+=====+")
777         print("|          SELAMAT DATANG DI SPORTFEST          |")
778         print("+-----+")
779         print("| TEMPAT BELI TIKET NONTON OLAHRAGA FAVORIT MU :* |")
780         print("+=====+")
781
782         print("\n")
783         print("+-----+")
784         print("|          Menu Pelanggan          |")
785         print("+-----+")
786         print("|      1. Tampilkan tiket          |")
787         print("|      2. Beli tiket nonton        |")
788         print("|      3. Lihat Saldo E-Money      |")
789         print("|      4. Top Up Saldo E-Money     |")
790         print("|      5. Cari                    |")
791         print("|      6. Keluar                  |")
792         print("+-----+")
793
794         try:
795             pilih_menu = int(input("Masukkan menu yang anda inginkan (1/2/3/4/5/6): "))
796             if pilih_menu == 1:
797                 tampilkanTiket()
798             elif pilih_menu == 2:
799                 transaksi()
800             elif pilih_menu == 3:
801                 lihatSaldo()
802             elif pilih_menu == 4:
803                 topUp()
804             elif pilih_menu == 5:
805                 cari()
806             elif pilih_menu == 6:
807                 print("Terima kasih telah menggunakan layanan kami!")
808                 main()
809             else:
810                 print("+=====+")
811                 print("| Tolong masukkan angka 1/2/3/4/5/6 |")
812                 print("+=====+\n")
813         except (ValueError, KeyboardInterrupt):
814             print("\n+=====+")
815             print("| Mohon masukkan data yang valid dan jangan tekan ctrl + C! |")
816             print("+=====+\n")
817         except Exception as e:
818             print(f"Terjadi kesalahan: {e}")
819

```

**Gambar 3.48** Alur Program Menu Pelanggan 1

```

804         elif pilih_menu == 5:
805             cari()
806         elif pilih_menu == 6:
807             print("Terima kasih telah menggunakan layanan kami!")
808             main()
809         else:
810             print("+=====+")
811             print("| Tolong masukkan angka 1/2/3/4/5/6 |")
812             print("+=====+\n")
813     except (ValueError, KeyboardInterrupt):
814         print("\n+=====+")
815         print("| Mohon masukkan data yang valid dan jangan tekan ctrl + C! |")
816         print("+=====+\n")
817     except Exception as e:
818         print(f"Terjadi kesalahan: {e}")
819

```

**Gambar 3.49** Alur Program Menu Pelanggan 2

Pada baris 773-818 kami membuat fungsi menu pelanggan yaitu untuk memilih menu pelanggan sendiri ada 6 pilihan yaitu: Menampilkan tiket, melakukan transaksi, lihat saldo, top-up, cari dan keluar

```

821 def generate_invoice(username, pertandingan, jenis_tiket, jumlah_tiket, total_harga):
822     try:
823         # Validasi input
824         if not all([username, pertandingan, jenis_tiket, jumlah_tiket, total_harga]):
825             print("Error: Data tidak lengkap")
826             return False
827
828         # Validasi tipe data
829         if not isinstance(jumlah_tiket, int) or jumlah_tiket <= 0:
830             print("Error: Jumlah tiket harus bilangan bulat positif")
831             return False
832
833         if not isinstance(total_harga, (int, float)) or total_harga <= 0:
834             print("Error: Total harga harus bilangan positif")
835             return False
836
837         invoice_id = str(uuid.uuid4())[:8]
838         current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
839
840         invoice_content = f"""
841         {' '*40}
842         STRUK PEMBELIAN TIKET SPORTFEST
843         {' '*40}
844         No. Invoice      : {invoice_id}
845         Tanggal         : {current_time}
846         Nama Pembeli    : {username}
847         {' '*40}
848         Pertandingan    : {pertandingan.get('Pertandingan', 'Tidak Diketahui')}
849         Kategori        : {pertandingan.get('Kategori', 'Tidak Diketahui')}
850         Tanggal/Waktu   : {pertandingan.get('Tanggal/Waktu', 'Tidak Diketahui')}
851         {' '*40}
852         Jenis Tiket     : {jenis_tiket}

```

**Gambar 3.50** Alur Program Struk Pembelian 1

```

852     Jenis Tiket      : {jenis_tiket}
853     Jumlah Tiket    : {jumlah_tiket}
854     Harga Satuan    : Rp {pertandingan.get(f'Harga Tiket {jenis_tiket.replace(" ", "")}', 0):,}
855     Total Harga     : Rp {total_harga:,}
856     {' '*40}
857     Terima kasih atas pembelian Anda!
858     {' '*40}
859     """
860     print(invoice_content)
861
862     struk_path = os.path.join(os.path.dirname(__file__), 'struk.txt')
863
864     with open(struk_path, 'a', encoding='utf-8') as file:
865         file.write(invoice_content + "\n")
866
867     print(f"\nStruk telah disimpan di {struk_path}")
868     return True
869
870     except Exception as e:
871         print(f"Gagal membuat struk: {e}")
872         import traceback
873         traceback.print_exc()
874         return False
875

```

**Gambar 3.51** Alur Program Struk Pembelian 2

Pada baris 821-874 kami membuat fungsi menu struk pembelian yaitu untuk membuat dan menampilkan struk pembelian tiket untuk suatu pertandingan, lengkap dengan detail pembelian

```
877 def proses_pembelian_tiket(pertandingan, username):
878     try:
879         # Validasi input
880         if not pertandingan or not username:
881             print("Error: Data tidak lengkap")
882             return False
883
884         print("LIMIT PEMBELIAN TIKET HANYA 5 TIKET")
885         print("=====")
886         print("          1. Tiket Ekonomi          ")
887         print("          2. Tiket VIP              ")
888         print("=====")
889
890         while True:
891             try:
892                 nanyaJenis = int(input("Tiket yang ingin dibeli (1/2): "))
893                 if nanyaJenis not in [1, 2]:
894                     print("Pilihan tidak valid. Silakan pilih 1 atau 2.")
895                     continue
896
897                 nanya = int(input("Berapa banyak tiket yang ingin anda beli: "))
898
899                 # Validasi jumlah tiket
900                 if nanya > 5:
901                     print("LIMIT TIKET HANYA 5 TIKET")
902                     continue
903                 elif nanya <= 0:
904                     print("Tolong masukkan jumlah tiket yang valid!")
905                     continue
906
907                 # Menghitung total harga dengan validasi
908                 if nanyaJenis == 1:
```

**Gambar 3.52** Alur Program Proses Pembelian 1

```

909     harga_tiket = pertandingan.get('Harga Tiket Ekonomi', 0)
910     jenis_tiket = "Ekonomi"
911     else:
912         harga_tiket = pertandingan.get('Harga Tiket VIP', 0)
913         jenis_tiket = "VIP"
914
915     # Validasi harga tiket
916     if harga_tiket <= 0:
917         print("Error: Harga tiket tidak valid")
918         return False
919
920     total_harga = harga_tiket * nanya
921
922     users = loadDataUser()
923     for user in users:
924         if user["Nama User"] == username:
925             if user.get("Saldo", 0) >= total_harga:
926                 user["Saldo"] -= total_harga
927                 savedataUser(users)
928
929                 invoice_success = generate_invoice([
930                     username,
931                     pertandingan,
932                     jenis_tiket,
933                     nanya,
934                     total_harga
935                 ])
936
937             if invoice_success:
938                 print(f"\nSisa saldo: Rp {user['Saldo']:,}")
939                 return True

```

**Gambar 3.53** Alur Program Proses Pembelian 2

```

939         return True
940     else:
941         print("Gagal membuat invoice")
942         return False
943     else:
944         print("=====")
945         print("\nSaldo tidak mencukupi")
946         print(f"Total harga: Rp {total_harga:,}")
947         print(f"Saldo Anda: Rp {user.get('Saldo', 0):,}")
948         print("=====")
949         print("          1. Isi Saldo")
950         print("          2. Batal")
951         print("=====")
952         pilihan = int(input("Masukkan pilihan: "))
953         if pilihan == 1:
954             topUp4()
955         elif pilihan == 2:
956             print("Pembelian tiket dibatalkan")
957             menu_pelanggan()
958         return False
959
960     print("User tidak ditemukan")
961     return False
962
963 except ValueError:
964     print(["Masukkan angka yang valid!"])
965
966 except Exception as e:
967     print(f"Terjadi kesalahan dalam proses pembelian: {e}")
968     return False

```

**Gambar 3.54** Alur Program Proses Pembelian 3

Pada baris 877-968 kami membuat fungsi proses pembelian yaitu untuk membeli jenis tiket ada dua yaitu ekonomi dan vip dan untuk pembayarannya apabila tidak mencukupi maka di sarankan untuk isi saldo terlebih dahulu

```

970 def transaksi2():
971     os.system("cls")
972     print("=====")
973     print("          BELI TIKET NONTON          ")
974     print("=====")
975     if 'username' not in globals():
976         print("Silakan login terlebih dahulu!")
977         return
978
979     while True:
980         try:
981             show_menu_transaksi()
982             pilihan = input("Masukkan nomor tiket olahraga yang ingin anda beli (1-6): ")
983
984             if pilihan == "6":
985                 return
986
987             pilihan = int(pilihan)
988             kategori_map = {
989                 1: 'Bola',
990                 2: 'Badminton',
991                 3: 'Basket',
992                 4: 'Voli',
993                 5: 'Futsal'
994             }
995
996             if pilihan in kategori_map:
997                 kategori_nama = kategori_map[pilihan]
998
999                 if pilihan == 1:
1000                     bola()
1001                 elif pilihan == 2:

```

**Gambar 3.55** Alur Program Transaksi 1

```

1002                     badminton()
1003                 elif pilihan == 3:
1004                     basket()
1005                 elif pilihan == 4:
1006                     voli()
1007                 elif pilihan == 5:
1008                     futsal()
1009
1010             pilihan_pertandingan = int(input(f"Masukkan nomor pertandingan yang ingin anda beli: "))
1011
1012             for kategori in data['Kategori']:
1013                 if kategori['Nama Kategori'] == kategori_nama:
1014                     if 0 < pilihan_pertandingan <= len(kategori['Jadwal']):
1015                         pertandingan = kategori['Jadwal'][pilihan_pertandingan-1]
1016                         print(f"\nAnda memilih pertandingan: {pertandingan['Pertandingan']}")
1017                         print(f"Kategori: {pertandingan['Kategori']}")
1018                         print(f"Waktu: {pertandingan['Tanggal/Waktu']}")
1019                         print(f"Harga Tiket Ekonomi: Rp {pertandingan['Harga Tiket Ekonomi':,]}")
1020                         print(f"Harga Tiket VIP: Rp {pertandingan['Harga Tiket VIP':,]}\n")
1021                         while True:
1022                             try:
1023                                 print("TERIMA KASIH SUDAH BERBELANJA!!")
1024                                 input("Tekan Enter Untuk Menanjutkan...")
1025                                 menu_pelanggan()
1026                                 break
1027                             except (ValueError, KeyboardInterrupt):
1028                                 print("Mohon masukkan huruf y atau n!!")
1029                         else:
1030                             print("Nomor pertandingan tidak valid")
1031                             break
1032             else:

```

**Gambar 3.56** Alur Program Transaksi 2

```

1032         else:
1033             print("TOLONG MASUKKAN ANGKA 1 - 5 UNTUK MEMILIH PERTANDINGAN")
1034
1035     except ValueError:
1036         print("Masukkan angka yang valid!")
1037     except KeyboardInterrupt:
1038         print("\n+=====+")
1039         print("| Mohon masukkan data yang valid dan jangan tekan ctrl + C! |")
1040         print("+=====+\n")
1041     except Exception as e:
1042         print(f"Terjadi kesalahan: {e}")

```

**Gambar 3.57** Alur Program Transaksi 3

Pada baris 970-1042 kami membuat fungsi transaksi yaitu untuk melakukan transaksi harus login terlebih dahulu, kemudian masukkan nomor pertandingan yang ingin di beli dan kemudian akan di proses untuk pembayarannya

```

def lihatSaldo3():
    os.system("cls")
    if 'username' not in globals():
        print("Silakan login terlebih dahulu!")
        return
    try:
        users = loadDataUser()
        user_found = False
        for user in users:
            if user["Nama User"] == username:
                user_found = True
                saldo = user["Saldo"]
                print("=====")
                print(f"          SALDO ANDA ADALAH Rp {saldo:,} ")
                print("=====")
                print("\n=====")
                print("          1. Ingin menambahkan saldo?")
                print("          2. Kembali")
                print("=====")
                pilihan = int(input("Masukkan pilihan 1/2: "))
                if pilihan == 1:
                    topUp4()
                elif pilihan == 2:
                    menu_pelanggan()
                else:
                    print("Nomor pilihan tidak valid")
                    break

```

**Gambar 3.58** Alur Program Melihat Saldo 1

```

1076         if not user_found:
1077             print("Error: User tidak ditemukan")
1078
1079     except Exception as e:
1080         print(f"Terjadi kesalahan: {str(e)}")

```

**Gambar 3.59** Alur Program Melihat Saldo 2



Pada baris 1044-1080 kami membuat fungsi melihat saldo yaitu untuk melihat saldo harus login terlebih dahulu, kemudian akan muncul saldo kita selanjutnya ada pilihan apakah kita mau top-up atau Kembali ke menu pelanggan

```
1083 def topUp4():
1084     os.system("cls")
1085     if 'username' not in globals():
1086         print("Silakan login terlebih dahulu!")
1087         return
1088
1089     print("+=====+")
1090     print("|          TOP UP SALDO          |")
1091     print("+=====+")
1092     users = loadDataUser()
1093     tabel = PrettyTable()
1094     tabel.clear_rows()
1095     tabel.title = "====TOP UP SALDO===="
1096     tabel.field_names = ["NO", "PILIHAN"]
1097     tabel.add_row(["1", "Rp 50.000"])
1098     tabel.add_row(["2", "Rp 100.000"])
1099     tabel.add_row(["3", "Rp 500.000"])
1100     tabel.add_row(["4", "Rp 1.000.000"])
1101     tabel.add_row(["5", "Rp 10.000.000"])
1102     tabel.add_row(["6", "Kembali"])
1103     print(tabel)
1104
1105     try:
1106         pilihan = int(input("Masukkan pilihan nominal (1-6): "))
1107
1108         nominal = [50000, 100000, 500000, 1000000, 10000000]
1109
1110         if 1 <= pilihan <= 5:
1111             user_found = False
1112             for user in users:
1113                 if user["Nama User"] == username:
1114                     user_found = True
```

**Gambar 3.60** Alur Program Top-Up 1

```

1115     konfirmasi = input(f"Anda akan top up sebesar Rp {nominal[pilihan-1]:,}. Lanjutkan? (y/n): ")
1116     if konfirmasi.lower() == 'y':
1117         user["Saldo"] += nominal[pilihan - 1]
1118         savedataUser(users)
1119         print(f"Top up berhasil!")
1120         print("=====")
1121         print(f"Saldo anda sekarang adalah Rp {user['Saldo']:,}")
1122         print("-----")
1123         print("                        1. Top Up lagi                ")
1124         print("                        2. Kembali                        ")
1125         print("=====")
1126         pilihan2 = int(input("Masukkan pilihan: "))
1127         if pilihan2 == 1:
1128             topUp4()
1129         elif pilihan2 == 2:
1130             menu_pelanggan()
1131         else:
1132             print("pilihan anda tidak valid atau tidak ada di menu")
1133     else:
1134         print("Top up dibatalkan")
1135         break
1136
1137     if not user_found:
1138         print("Error: User tidak ditemukan")
1139
1140 elif pilihan == 6:
1141     menu_pelanggan()
1142 else:
1143     print("Pilihan anda tidak tersedia")
1144
1145 except (ValueError, KeyboardInterrupt):

```

**Gambar 3.61** Alur Program Top-Up 2

```

except (ValueError, KeyboardInterrupt):
    print("Masukkan pilihan yang valid (angka 1-6) dan jangan menekan ctrl + c!")
except Exception as e:
    print(f"Terjadi kesalahan: {str(e)}")

```

**Gambar 3.62** Alur Program Top-Up 3

Pada baris 1083-1148 kami membuat fungsi top-up yaitu untuk mengisi saldo harus login terlebih dahulu, kemudian akan ada 5 pilihan nominal untuk top-up, kemudian setelah berhasil akan ada 2 pilihan yaitu Kembali top-up atau Kembali

```

1150 def cari5():
1151     os.system("cls")
1152     print("\n+=====+")
1153     print("|          CARI TIKET          |")
1154     print("+=====+")
1155
1156     try:
1157         keyword = input("Masukkan kata kunci pencarian: ").lower()
1158
1159         found = False
1160         table = PrettyTable()
1161         table.field_names = ["Kategori", "Pertandingan", "Jenis", "Waktu", "Tiket Ekonomi", "Tiket VIP"]
1162
1163         for kategori in data['Kategori']:
1164             kategori_nama = kategori['Nama Kategori']
1165             for jadwal in kategori['Jadwal']:
1166                 # Mencari berdasarkan nama kategori atau nama pertandingan
1167                 if (keyword in kategori_nama.lower() or
1168                     keyword in jadwal['Pertandingan'].lower()):
1169                     table.add_row([
1170                         kategori_nama,
1171                         jadwal['Pertandingan'],
1172                         jadwal['Kategori'],
1173                         jadwal['Tanggal/Waktu'],
1174                         f"Rp {jadwal['Harga Tiket Ekonomi']:,}",
1175                         f"Rp {jadwal['Harga Tiket VIP']:,}"
1176                     ])
1177             found = True
1178
1179         if found:
1180             print("\nHasil Pencarian:")
1181             print(table)

```

**Gambar 3.63** Alur Program Menu Cari Pelanggan 1

```

1182     else:
1183         print("\nTidak ditemukan hasil yang cocok dengan kata kunci pencarian.")
1184
1185         input("\nTekan Enter untuk kembali ke menu...")
1186
1187     except KeyboardInterrupt:
1188         print("\nPencarian dibatalkan.")
1189     except Exception as e:
1190         print(f"\nTerjadi kesalahan: {str(e)}")
1191

```

**Gambar 3.64** Alur Program Menu Cari Pelanggan 2

Pada baris 1150-1191 kami membuat fungsi cari yaitu untuk mencari maka masukkan kata kunci yang ingin di cari di programnya

```

1192 def main():
1193     os.system("cls")
1194     while True:
1195         try:
1196             print("\n+===== MENU =====+")
1197             print("|          1. Register          |")
1198             print("|          2. Login user        |")
1199             print("|          3. Login admin       |")
1200             print("|          4. Keluar            |")
1201             print("+=====+")
1202             pilihan = int(input("Pilih opsi (1/2/3/4): "))
1203             if pilihan == 1:
1204                 daftaruser()
1205             elif pilihan == 2:
1206                 if loginUser():
1207                     menu_pelanggan()
1208                 else:
1209                     print("Username atau Password salah, silahkan coba lagi")
1210             elif pilihan == 3:
1211                 if loginAdmin() == True:
1212                     khususAtmint()
1213                     break
1214                 else:
1215                     loginAdmin()
1216             elif pilihan == 4:
1217                 print("Program berakhir.")
1218                 exit()
1219                 break
1220             else:
1221                 print("Pilihan tidak valid, coba lagi.")
1222         except (ValueError, KeyboardInterrupt):
1223             print("Pilihan anda tidak valid, dan jangan menekan CTRL + C")

```

**Gambar 3.65 Alur Program Menu Main 1**

```

1221         print("Pilihan tidak valid, coba lagi.")
1222     except (ValueError, KeyboardInterrupt):
1223         print("Pilihan anda tidak valid, dan jangan menekan CTRL + C")
1224
1225 if __name__ == "__main__":
1226     main()

```

**Gambar 3.66 Alur Program Menu Main 2**

Pada baris 1192-1226 kami membuat fungsi main yaitu untuk menjalankan program ini maka harus ada fungsi ini, kemudian di fungsi main ini ada empat pilihan yaitu register, login pengguna, login admin, dan keluar

### 3.3. Source Code

```

import json
from prettytable import PrettyTable
from operator import itemgetter
from datetime import datetime
import pwininput
import getpass
import time
import uuid
import os

PathjsonUser = r"C:\Koding\TestCode\regist.json"

# =====
terserah = os.path.abspath(os.curdir)

json_s = "C:\\Koding\\TestCode\\schedule.json"

def load_schedule():
    with open(json_s, 'r') as data_schedule:
        return json.load(data_schedule) #load => loads, biar sekalian diubah jadi string dan jadi format
        dictionary
    # return data

def save_schedule(data):
    with open(json_s, 'w') as data_schedule:
        json.dump(data, data_schedule, indent=4)

data = load_schedule()
# =====

def loadDataUser():

```

```

try:
    with open(PathjsonUser, "r") as jsonUser:
        return json.load(jsonUser)
except FileNotFoundError:
    return {"Nama": [], "Password": [], "saldo": []}

def savedataUser(data):
    with open(PathjsonUser, "w") as jsonUser:
        json.dump(data, jsonUser, indent=4)

#=====REGISTER=====
=====#

def daftaruser():
    users = loadDataUser ()
    while True:
        try:
            while True:
                namaUser = input("Masukkan nama (min 3 karakter): ").strip()
                if len(namaUser ) < 3:
                    print("Nama pengguna harus minimal 3 karakter!")
                elif len(namaUser) > 50:
                    print("Nama pengguna tidak boleh lebih dari 50 karakter!")
                elif any(user["Nama User"] == namaUser for user in users):
                    print(f"{namaUser } sudah terdaftar.")
                else:
                    break

            while True:
                passwordUser = pwininput.pwininput("Password (min 6 karakter): ", "*")
                if len(passwordUser ) < 6:

```

```

        print("Password harus minimal 6 karakter!")
    else:
        break

    new_user = {
        "Nama User": namaUser ,
        "Pw User": passwordUser ,
        "Saldo": 0
    }
    users.append(new_user)
    savedataUser (users)
    print("""
=====+
|      REGISTER ANDA BERHASIL      |
=====+""")

    break
except ValueError:
    print("Terdapat kesalahan, silahkan coba lagi!")
except KeyboardInterrupt:
    print("Tolong untuk tidak menekan Ctrl dan C secara bersamaan!")

def loginUser():
    global username
    users_data = loadDataUser()
    while True:
        try:
            if not users_data:
                print("No user data found.")
                return False

```

```

print("==== LOGIN USER ====")

username = input("Masukkan username: ")
password = pwinput.pwinput("Masukkan password: ", "*")

for user in users_data:
    if user["Nama User"] == username:
        if user["Pw User"] == password:
            print("Login berhasil!")
            return True
        else:
            print("Password salah!")
            return False
            break
    print("Username tidak ditemukan!")
    return False
except(ValueError, KeyboardInterrupt):
    print("Tolong untuk tidak menekan Ctrl dan C secara bersamaan!")

def loginAdmin():
    admin = {
        "RehanGokil": "Admin#1234",
        "SamsulBetawi": "SayaJawa",
        "MorenoPakPak": "PakPak123",
        "FakriAyamPramuka": "MintaSambelnyaMas"
    }
    username = input("Masukkan username Anda: ")
    password = pwinput.pwinput("Masukkan Password Anda: ", '*')
    if username in admin and password == admin[username]:
        print("+=====+")
        print("|          SELAMAT DATANG ADMIN          |")
        print("+=====+")

```



```

        return True
    else:
        print("Username atau Password salah, Masukkan dengan Benar!!")
        return False

#=====D
EF-
ADMINCRUD=====
=====#

def bola():
    tabel = PrettyTable()
    tabel.title = "PERTANDINGAN BOLA"
    tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi",
"Harga Tiket VIP"]
    nomor = 1
    for kategori in data['Kategori']:
        if kategori['Nama Kategori'] == 'Bola':
            for match in kategori['Jadwal']:
                tabel.add_row([f"[{nomor}]", match['Pertandingan'], match['Kategori'],
match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi':,}]", f"Rp{match['Harga Tiket
VIP':,}]")
                nomor += 1
            print(tabel)

def badminton():
    tabel = PrettyTable()
    tabel.title = "PERTANDINGAN BADMINTON"
    tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi",
"Harga Tiket VIP"]
    nomor = 1

```

```

for kategori in data['Kategori']:
    if kategori['Nama Kategori'] == 'Badminton':
        for match in kategori['Jadwal']:
            tabel.add_row([f"[{nomor}]", match['Pertandingan'], match['Kategori'],
match['Tanggal/Waktu'], f"Rp{match['Harga Tiket Ekonomi':,}]", f"Rp{match['Harga Tiket
VIP':,}]")
            nomor += 1
        print(tabel)

def basket():
    tabel = PrettyTable()
    tabel.title = "PERTANDINGAN BASKET"
    tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga
Tiket VIP"]
    nomor = 1
    for kategori in data['Kategori']:
        if kategori['Nama Kategori'] == 'Basket':
            for match in kategori['Jadwal']:
                tabel.add_row([f"[{nomor}]", match['Pertandingan'], match['Kategori'],
match["Tanggal/Waktu"], f"Rp{match['Harga Tiket Ekonomi':,}]", f"Rp{match['Harga Tiket
VIP':,}]")
                nomor += 1
            print(tabel)

def voli():
    tabel = PrettyTable()
    tabel.title = "PERTANDINGAN VOLI"
    tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga
Tiket VIP"]
    nomor = 1

```

```

for kategori in data['Kategori']:
    if kategori['Nama Kategori'] == 'Voli':
        for match in kategori['Jadwal']:
            tabel.add_row([f"[{nomor}]", match['Pertandingan'], match['Kategori'],
match["Tanggal/Waktu"], f"Rp{match['Harga Tiket Ekonomi']:,}", f"Rp{match['Harga Tiket VIP']:,}"])
            nomor += 1
        print(tabel)

def futsal():
    tabel = PrettyTable()
    tabel.title = "PERTANDINGAN FUTSAL"
    tabel.field_names = ["No", "Pertandingan", "Kategori", "Waktu", "Harga Tiket Ekonomi", "Harga
Tiket VIP"]
    nomor = 1
    for kategori in data['Kategori']:
        if kategori['Nama Kategori'] == 'Futsal':
            for match in kategori['Jadwal']:
                tabel.add_row([f"[{nomor}]", match['Pertandingan'], match['Kategori'],
match["Tanggal/Waktu"], f"Rp{match['Harga Tiket Ekonomi']:,}", f"Rp{match['Harga Tiket VIP']:,}"])
                nomor += 1
            print(tabel)

def baris():
    print(" ")
    print("+=
=====
+=")
    print(" ")

def khususAtmint():
    while True:
        try:

```

```

os.system("cls")
tabel = PrettyTable()
tabel.clear_rows()
tabel.title = "PILIHAN ADMIN"
tabel.field_names = ["NO", "PILIHAN"]
tabel.add_row (["1", "Menu Tiket"])
tabel.add_row (["2", "Tambah Tiket"])
tabel.add_row (["3", "Update Tiket"])
tabel.add_row (["4", "Hapus Tiket"])
tabel.add_row (["5", "Keluar"])
print(tabel)
pilihan = input("Masukkan pilihan 1/2/3/4: ")

if pilihan == '1':
    menu()
elif pilihan == '2':
    create()
elif pilihan == '3':
    update()
elif pilihan == '4':
    delete()
elif pilihan == '5':
    main()
else:
    print("Maaf, pilihan tidak ada di menu, masukkan pilihan yang benar")
    break
except(ValueError, KeyboardInterrupt):
    print("MOHON MASUKKAN DATA YANG VALID DAN JANGAN MENEKAN CTRL +
C!!")

def cari_tiket():

```

```

os.system("cls")

print("\n+=====+")
print("|      CARI TIKET      |")
print("+=====+")

try:
    keyword = input("Masukkan kata kunci pencarian: ").lower()

    found = False
    table = PrettyTable()
    table.field_names = ["Kategori", "Pertandingan", "Jenis", "Waktu", "Tiket Ekonomi", "Tiket
VIP"]

    for kategori in data['Kategori']:
        kategori_nama = kategori['Nama Kategori']
        for jadwal in kategori['Jadwal']:
            if (keyword in kategori_nama.lower() or
                keyword in jadwal['Pertandingan'].lower()):
                table.add_row([
                    kategori_nama,
                    jadwal['Pertandingan'],
                    jadwal['Kategori'],
                    jadwal['Tanggal/Waktu'],
                    f"Rp {jadwal['Harga Tiket Ekonomi']:,}",
                    f"Rp {jadwal['Harga Tiket VIP']:,}"
                ])
            found = True

    if found:
        print("\nHasil Pencarian:")
        print(table)
    else:

```

```

print("\nTidak ditemukan hasil yang cocok dengan kata kunci pencarian.")

except KeyboardInterrupt:
    print("\nPencarian dibatalkan.")
except Exception as e:
    print(f"\nTerjadi kesalahan: {str(e)}")

def sorting_tiket():
    semua_pertandingan = []
    for kategori in data['Kategori']:
        for jadwal in kategori['Jadwal']:
            semua_pertandingan.append({
                'Kategori': kategori['Nama Kategori'],
                'Pertandingan': jadwal['Pertandingan'],
                'Kategori_Pertandingan': jadwal['Kategori'],
                'Tanggal/Waktu': jadwal['Tanggal/Waktu'],
                'Harga Tiket Ekonomi': jadwal['Harga Tiket Ekonomi'],
                'Harga Tiket VIP': jadwal['Harga Tiket VIP']
            })

    semua_pertandingan.sort(key=itemgetter('Pertandingan'))

    tabel = PrettyTable()
    tabel.field_names = ["Kategori", "Pertandingan", "Kategori Pertandingan", "Tanggal/Waktu",
                        "Harga Ekonomi", "Harga VIP"]
    for pertandingan in semua_pertandingan:
        tabel.add_row([
            pertandingan['Kategori'],
            pertandingan['Pertandingan'],
            pertandingan['Kategori_Pertandingan'],
            pertandingan['Tanggal/Waktu'],

```

```

        f"Rp{pertandingan['Harga Tiket Ekonomi']:,}",
        f"Rp{pertandingan['Harga Tiket VIP']:,}"
    ])
    print(tabel)

def afterMenuAdmin():
    while True:
        try:
            tabel = PrettyTable()
            tabel.clear_rows()
            tabel.title = "SILAHKAN PILIH"
            tabel.field_names = ["NO", "PILIHAN"]
            tabel.add_row (["1", "Cari Tiket"])
            tabel.add_row (["2", "Sorting tiket (Berdasarkan abjad)"])
            tabel.add_row (["3", "Keluar"])
            print(tabel)
            pilihan = int(input("Masukkan pilihan 1/2/3: "))
            if pilihan == 1:
                cari_tiket()
                print("=====")
                print(" 1. Cari Tiket Lain" )
                print(" 2. Kembali" )
                print("=====")
                pilihan2 = int(input("Masukkan pilihan 1/2: "))
                if pilihan2 == 1:
                    cari_tiket()
                elif pilihan2 == 2:
                    khususAtmint()
                else:
                    print("PENCARIAN ANDA TIDAK VALID, MASUKKAN PILIHAN YANG VALID")

```

```

elif pilihan == 2:
    sorting_tiket()
    print("=====")
    print( input("Tekan enter untuk kembali..."))
    print("=====")
    khususAtmint()

elif pilihan == 3:
    khususAtmint()
    break
else:
    print("Pilihan anda tidak ada, mohon masukkan pilihan dengan benar!!")
except (ValueError, KeyboardInterrupt):
    print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

def afterMenuUser():
    while True:
        try:
            tabel = PrettyTable()
            tabel.clear_rows()
            tabel.title = "SILAHKAN PILIH"
            tabel.field_names = ["NO", "PILIHAN"]
            tabel.add_row (["1]", "Cari Tiket"])
            tabel.add_row (["2]", "Sorting tiket (Berdasarkan abjad)"])
            tabel.add_row (["3]", "Keluar"])
            print(tabel)
            pilihan = int(input("Masukkan pilihan 1/2/3: "))
            if pilihan == 1:
                cari_tiket()

```



```

print("=====")
print(" 1. Cari Tiket Lain" )
print(" 2. Kembali" )
print("=====")
pilihan2 = int(input("Masukkan pilihan 1/2: "))
if pilihan2 == 1:
    cari_tiket()
elif pilihan2 == 2:
    menu_pelanggan()
else:
    print("PENCARIAN ANDA TIDAK VALID, MASUKKAN PILIHAN YANG
VALID")

elif pilihan == 2:
    sorting_tiket()
    print("=====")
    print( input("Tekan enter untuk kembali..."))
    print("=====")
    menu_pelanggan()

elif pilihan == 3:
    menu_pelanggan()
    break
else:
    print("Pilihan anda tidak ada, mohon masukkan pilihan dengan benar!!")
except (ValueError, KeyboardInterrupt):
    print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

def menu():
    os.system("cls")

```

bola()

baris()

badminton()

baris()

basket()

baris()

voli()

baris()

futsal()

afterMenuAdmin()

def menuUser():

os.system("cls")

bola()

baris()

badminton()

baris()

basket()

baris()

voli()

baris()

futsal()

afterMenuUser()

```
def show_menu(is_admin=True):
    os.system("cls")
    tabel = PrettyTable()
    tabel.clear_rows()
    tabel.title = "====CABANG LOMBA===="
    tabel.field_names = ["NO", "PILIHAN"]
    tabel.add_row (["1", "BOLA"])
    tabel.add_row (["2", "BADMINTON"])
    tabel.add_row (["3", "BASKET"])
    tabel.add_row (["4", "VOLI"])
    tabel.add_row (["5", "FUTSAL"])
    tabel.add_row (["6", "KEMBALI"])
    print(tabel)

    pilihan = input("Masukkan pilihan (1-6): ")
    if pilihan == "6":
        if is_admin:
```

```

        khususAtmint()
    else:
        menu_pelanggan()
    return True
return False

def show_menu_transaksi():
    tabel = PrettyTable()
    tabel.clear_rows()
    tabel.title = "====CABANG LOMBA===="
    tabel.field_names = ["NO", "PILIHAN"]
    tabel.add_row(["1", "BOLA"])
    tabel.add_row(["2", "BADMINTON"])
    tabel.add_row(["3", "BASKET"])
    tabel.add_row(["4", "VOLI"])
    tabel.add_row(["5", "FUTSAL"])
    tabel.add_row(["6", "KEMBALI"])
    print(tabel)

def create():
    os.system("cls")
    while True:
        try:
            print("====TAMBAH TIKET====")
            tabel = PrettyTable()
            tabel.clear_rows()
            tabel.title = "MAU NAMBAH TIKET NONTON APA NIH?"
            tabel.field_names = ["NO", "PILIHAN"]
            tabel.add_row(["1", "BOLA"])

```

```

tabel.add_row(["[2]", "BADMINTON"])
tabel.add_row(["[3]", "BASKET"])
tabel.add_row(["[4]", "VOLI"])
tabel.add_row(["[5]", "FUTSAL"])
tabel.add_row(["[6]", "KEMBALI"])
print(tabel)
pilihanCreate = int(input("Masukkan pilihan 1/2/3/4/5/6: "))

if pilihanCreate in [1, 2, 3, 4, 5]:
    kategori_map = {1: 'Bola', 2: 'Badminton', 3: 'Basket', 4: 'Voli', 5: 'Futsal'}
    kategori_nama = kategori_map[pilihanCreate]

    print(f"=====TAMBAH TIKET {kategori_nama.upper()}=====")

    # Validasi Pertandingan
    while True:
        pertandingan = input("Masukkan pertandingan yang ingin ditambahkan: ").strip()
        if pertandingan and len(pertandingan) >= 3:
            break
        else:
            print("Nama pertandingan tidak boleh kosong dan minimal 3 karakter!")

    # Validasi Kategori
    while True:
        kategori = input("Masukkan kategori pertandingan: ").strip()
        if kategori and len(kategori) >= 2:
            break
        else:
            print("Kategori tidak boleh kosong dan minimal 2 karakter!")

```

```

# Validasi Tanggal
while True:
    tanggal = input("Masukkan tanggal pertandingan: ").strip()
    if tanggal and len(tanggal) >= 5:
        break
    else:
        print("Tanggal tidak boleh kosong dan minimal 5 karakter!")

# Validasi Harga Tiket Ekonomi
while True:
    try:
        hargaEkonomi = int(input("Masukkan Harga Tiket Ekonomi: "))
        if hargaEkonomi > 0:
            break
    except ValueError:
        print("Masukkan angka yang valid!")

# Validasi Harga Tiket VIP
while True:
    try:
        hargaVIP = int(input("Masukkan Harga Tiket VIP: "))
        if hargaVIP > 0:
            break
    except ValueError:
        print("Masukkan angka yang valid!")

```

```

for kategori_data in data['Kategori']:
    if kategori_data['Nama Kategori'] == kategori_nama:
        kategori_data['Jadwal'].append({
            "Pertandingan": pertandingan,
            "Kategori": kategori,
            "Tanggal/Waktu": tanggal,
            "Harga Tiket Ekonomi": hargaEkonomi,
            "Harga Tiket VIP": hargaVIP
        })
        save_schedule(data)
        print(f"Pertandingan {pertandingan} berhasil ditambahkan")
        break
    else:
        print(f"Kategori {kategori_nama} tidak ditemukan")

    khususAtmint()
elif pilihanCreate == 6:
    khususAtmint()
    break
else:
    print("Input anda tidak valid, silahkan masukkan pilihan yang tersedia!")
except(ValueError, KeyboardInterrupt):
    print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

def update():
    os.system("cls")
    print("=====UPDATE TIKET=====")
    print("")
    while True:
        try:

```

```

if show_menu(is_admin=True):
    return
kategoriLomba = input("Masukkan cabang lomba (1-5): ").strip()

kategori_map = {
    "1": "Bola",
    "2": "Badminton",
    "3": "Basket",
    "4": "Voli",
    "5": "Futsal"
}

if kategoriLomba in kategori_map:
    kategori_nama = kategori_map[kategoriLomba]

    if kategoriLomba == "1":
        bola()
    elif kategoriLomba == "2":
        badminton()
    elif kategoriLomba == "3":
        basket()
    elif kategoriLomba == "4":
        voli()
    elif kategoriLomba == "5":
        futsal()

    nomor = int(input("Masukkan nomor yang ingin diubah: ")) - 1
    for kategori in data['Kategori']:
        if kategori['Nama Kategori'] == kategori_nama:
            if 0 <= nomor < len(kategori['Jadwal']):

```



```
match = kategori['Jadwal'][nomor]
```

```
while True:
```

```
    pertandingan = input("Pertandingan baru: ").strip()
```

```
    if pertandingan and len(pertandingan) >= 3:
```

```
        break
```

```
    else:
```

```
        print("Nama pertandingan tidak boleh kosong dan minimal 3 karakter!")
```

```
while True:
```

```
    kategori_ = input("Kategori baru: ").strip()
```

```
    if kategori_ and len(kategori_) >= 2:
```

```
        break
```

```
    else:
```

```
        print("Kategori tidak boleh kosong dan minimal 2 karakter!")
```

```
while True:
```

```
    tanggal = input("Tanggal/Waktu baru: ").strip()
```

```
    if tanggal and len(tanggal) >= 5:
```

```
        break
```

```
    else:
```

```
        print("Tanggal tidak boleh kosong dan minimal 5 karakter!")
```

```
while True:
```

```
    try:
```

```
        hargaekonomi = int(input("Harga Tiket Ekonomi baru: "))
```

```
        if hargaekonomi > 0:
```

```
            break
```

```
        else:
```

```
            print("Harga tiket harus lebih dari 0!")
```

```

        except ValueError:
            print("Masukkan angka yang valid!")

    while True:
        try:
            hargavip = int(input("Harga Tiket VIP baru: "))
            if hargavip > 0:
                break
            else:
                print("Harga tiket harus lebih dari 0!")
        except ValueError:
            print("Masukkan angka yang valid!")

    match['Pertandingan'] = pertandingan
    match['Kategori'] = kategori_
    match['Tanggal/Waktu'] = tanggal
    match['Harga Tiket Ekonomi'] = hargaekonomi
    match['Harga Tiket VIP'] = hargavip

    save_schedule(data)
    print(f"Data ke- {nomor+1 } telah diperbarui")
else:
    print("Nomor pertandingan tidak valid")
    break
else:
    print("Kategori tidak ditemukan")

    khususAtmint()
    break
else:

```

```

        print("Input tidak valid. Masukkan angka 1-5.")
    except(ValueError, KeyboardInterrupt):
        print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

def delete():
    print("=====HAPUS TIKET=====")
    print("")
    while True:
        try:
            show_menu_transaksi()
            kategoriLomba = input("Masukkan cabang lomba: ").strip()

            if kategoriLomba == "1":
                while True:
                    try:
                        bola()
                        nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
                        for kategori in data['Kategori']:
                            if kategori['Nama Kategori'] == 'Bola':
                                if 0 <= nomor < len(kategori['Jadwal']):
                                    tiketLama = kategori['Jadwal'].pop(nomor)
                                    save_schedule(data)
                                    print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
                                    input("TEKAN ENTER UNTUK MELANJUTKAN...")
                                    khususAtmint()
                                    break
                                else:
                                    print("Nomor tidak valid")
                            else:
                                print("Kategori tidak valid")
                        except(ValueError, KeyboardInterrupt):
                            print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL + C!!")

                    elif kategoriLomba == "2":

```

```

while True:
    try:
        badminton()
        nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
        for kategori in data['Kategori']:
            if kategori['Nama Kategori'] == 'Badminton':
                if 0 <= nomor < len(kategori['Jadwal']):
                    tiketLama = kategori['Jadwal'].pop(nomor)
                    save_schedule(data)
                    print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
                    input("TEKAN ENTER UNTUK MELANJUTKAN...")
                    khususAtmint()
                    break
        except(ValueError, KeyboardInterrupt):
            print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL
+ C!!")

elif kategoriLomba == "3":
    while True:
        try:
            basket()
            nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
            for kategori in data['Kategori']:
                if kategori['Nama Kategori'] == 'Basket':
                    if 0 <= nomor < len(kategori['Jadwal']):
                        tiketLama = kategori['Jadwal'].pop(nomor)
                        save_schedule(data)
                        print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
                        input("TEKAN ENTER UNTUK MELANJUTKAN...")
                        khususAtmint()
                        break

```

```

except(ValueError, KeyboardInterrupt):
    print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL
+ C!!")

elif kategoriLomba == "4":
    while True:
        try:
            voli()
            nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
            for kategori in data['Kategori']:
                if kategori['Nama Kategori'] == 'Voli':
                    if 0 <= nomor < len(kategori['Jadwal']):
                        tiketLama = kategori['Jadwal'].pop(nomor)
                        save_schedule(data)
                        print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
                        input("TEKAN ENTER UNTUK MELANJUTKAN...")
                        khususAtmint()
                        break
        except(ValueError, KeyboardInterrupt):
            print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN CTRL
+ C!!")

elif kategoriLomba == "5":
    while True:
        try:
            futsal()
            nomor = int(input("Masukkan nomor yang mau dihapus: ")) - 1
            for kategori in data['Kategori']:
                if kategori['Nama Kategori'] == 'Futsal':
                    if 0 <= nomor < len(kategori['Jadwal']):
                        tiketLama = kategori['Jadwal'].pop(nomor)
                        save_schedule(data)

```

```

        print(f"Paket {tiketLama['Pertandingan']} telah dihapus")
        input("TEKAN ENTER UNTUK MELANJUTKAN...")
        khususAtmint()
    except(ValueError, KeyboardInterrupt):
        print("PILIHAN ANDA TIDAK VALID DAN JANGAN MENEKAN
CTRL + C!!")

    else:
        print("Input anda tidak valid, silahkan masukkan cabang lomba yang tersedia!")

    except(ValueError, KeyboardInterrupt):
        print("TOLONG MASUKKAN INPUT YANG VALID DAN JANGAN
MEMENCET CTRL + C!")
        break

#=====A
DMIN-
END=====
=====#

def tampilkanTiket():
    os.system("cls")
    menuUser()
    afterMenuUser()

def transaksi():
    os.system("cls")
    transaksi2()

```

```

def lihatSaldo():
    os.system("cls")
    lihatSaldo3()

def topUp():
    os.system("cls")
    topUp4()

def cari():
    os.system("cls")
    cari5()

def menu_pelanggan():
    while True:
        os.system("cls" if os.name == "nt" else "clear")
        print("+=====+")
        print("|          SELAMAT DATANG DI SPORTFEST          |")
        print("+-----+")
        print("| TEMPAT BELI TIKET NONTON OLAHRAGA FAVORIT MU :* |")
        print("+=====+")

        print("\n")
        print("+-----+")
        print("|          Menu Pelanggan          |")
        print("+-----+")
        print("|  1. Tampilkan tiket          |")
        print("|  2. Beli tiket nonton        |")
        print("|  3. Lihat Saldo E-Money      |")
        print("|  4. Top Up Saldo E-Money     |")
        print("|  5. Cari                      |")
        print("|  6. Keluar                    |")
        print("+-----+")

```

```

try:
    pilih_menu = int(input("Masukkan menu yang anda inginkan (1/2/3/4/5/6): "))
    if pilih_menu == 1:
        tampilkanTiket()
    elif pilih_menu == 2:
        transaksi()
    elif pilih_menu == 3:
        lihatSaldo()
    elif pilih_menu == 4:
        topUp()
    elif pilih_menu == 5:
        cari()
    elif pilih_menu == 6:
        print("Terima kasih telah menggunakan layanan kami!")
        main()
    else:
        print("+=====+")
        print("| Tolong masukkan angka 1/2/3/4/5/6   |")
        print("+=====+\n")
except (ValueError, KeyboardInterrupt):
    print("\n+=====+
=====+")
    print("| Mohon masukkan data yang valid dan jangan tekan ctrl + C! |")
    print("+=====+
=====+\n")
except Exception as e:
    print(f"Terjadi kesalahan: {e}")

def generate_invoice(username, pertandingan, jenis_tiket, jumlah_tiket, total_harga):
    try:
        # Validasi input
        if not all([username, pertandingan, jenis_tiket, jumlah_tiket, total_harga]):

```



```

if not all([username, pertandingan, jenis_tiket, jumlah_tiket, total_harga]):
    print("Error: Data tidak lengkap")
    return False

# Validasi tipe data
if not isinstance(jumlah_tiket, int) or jumlah_tiket <= 0:
    print("Error: Jumlah tiket harus bilangan bulat positif")
    return False

if not isinstance(total_harga, (int, float)) or total_harga <= 0:
    print("Error: Total harga harus bilangan positif")
    return False

invoice_id = str(uuid.uuid4())[:8]
current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")

invoice_content = f"""
{'='*40}

STRUK PEMBELIAN TIKET SPORTFEST

{'='*40}
No. Invoice : {invoice_id}
Tanggal    : {current_time}
Nama Pembeli : {username}

{'='*40}
Pertandingan : {pertandingan.get('Pertandingan', 'Tidak Diketahui')}
Kategori     : {pertandingan.get('Kategori', 'Tidak Diketahui')}
Tanggal/Waktu : {pertandingan.get('Tanggal/Waktu', 'Tidak Diketahui')}

{'='*40}
Jenis Tiket  : {jenis_tiket}
Jumlah Tiket : {jumlah_tiket}
Harga Satuan : Rp {pertandingan.get(f'Harga Tiket {jenis_tiket.replace(" ", "")}', 0):,}
Total Harga  : Rp {total_harga:,}

```

```

{'='*40}
    Terima kasih atas pembelian Anda!
{'='*40}
"""

    print(invoice_content)

    struk_path = os.path.join(os.path.dirname(__file__), 'struk.txt')

    with open(struk_path, 'a', encoding='utf-8') as file:
        file.write(invoice_content + "\n")

    print(f"\nStruk telah disimpan di {struk_path}")
    return True

except Exception as e:
    print(f"Gagal membuat struk: {e}")
    import traceback
    traceback.print_exc()
    return False

def proses_pembelian_tiket(pertandingan, username):
    try:
        # Validasi input
        if not pertandingan or not username:
            print("Error: Data tidak lengkap")
            return False

        print("LIMIT PEMBELIAN TIKET HANYA 5 TIKET")
        print("=====")
        print("        1. Tiket Ekonomi        ")
        print("        2. Tiket VIP            ")
        print("=====")

```

```

while True:
    try:
        nanyaJenis = int(input("Tiket yang ingin dibeli (1/2): "))
        if nanyaJenis not in [1, 2]:
            print("Pilihan tidak valid. Silakan pilih 1 atau 2.")
            continue

        nanya = int(input("Berapa banyak tiket yang ingin anda beli: "))

        # Validasi jumlah tiket
        if nanya > 5:
            print("LIMIT TIKET HANYA 5 TIKET")
            continue
        elif nanya <= 0:
            print("Tolong masukkan jumlah tiket yang valid!")
            continue

        # Menghitung total harga dengan validasi
        if nanyaJenis == 1:
            harga_tiket = pertandingan.get('Harga Tiket Ekonomi', 0)
            jenis_tiket = "Ekonomi"
        else:
            harga_tiket = pertandingan.get('Harga Tiket VIP', 0)
            jenis_tiket = "VIP"

        # Validasi harga tiket
        if harga_tiket <= 0:
            print("Error: Harga tiket tidak valid")
            return False

        total_harga = harga_tiket * nanya

```

```

users = loadDataUser()
for user in users:
    if user["Nama User"] == username:
        if user.get("Saldo", 0) >= total_harga:
            user["Saldo"] -= total_harga
            savedataUser(users)

        invoice_success = generate_invoice(
            username,
            pertandingan,
            jenis_tiket,
            nanya,
            total_harga
        )

        if invoice_success:
            print(f"\nSisa saldo: Rp {user['Saldo']:,.}")
            return True
        else:
            print("Gagal membuat invoice")
            return False
    else:
        print("=====")
        print("\nSaldo tidak mencukupi")
        print(f"Total harga: Rp {total_harga:,.}")
        print(f"Saldo Anda: Rp {user.get('Saldo', 0):,.}")
        print("-----")
        print("      1. Isi Saldo")
        print("      2. Batal")
        print("=====")

```

```

        pilihan = int(input("Masukkan pilihan: "))
        if pilihan == 1:
            topUp4()
        elif pilihan == 2:
            print("Pembelian tiket dibatalkan")
            menu_pelanggan()
        return False

    print("User tidak ditemukan")
    return False

except ValueError:
    print("Masukkan angka yang valid!")

except Exception as e:
    print(f"Terjadi kesalahan dalam proses pembelian: {e}")
    return False

def transaksi2():
    os.system("cls")
    print("=====")
    print("          BELI TIKET NONTON          ")
    print("=====")
    if 'username' not in globals():
        print("Silakan login terlebih dahulu!")
        return

    while True:
        try:
            show_menu_transaksi()
            pilihan = input("Masukkan nomor tiket olahraga yang ingin anda beli (1-6): ")

```

```

if pilihan == "6":
    return

pilihan = int(pilihan)
kategori_map = {
    1: 'Bola',
    2: 'Badminton',
    3: 'Basket',
    4: 'Voli',
    5: 'Futsal'
}

if pilihan in kategori_map:
    kategori_nama = kategori_map[pilihan]

    if pilihan == 1:
        bola()
    elif pilihan == 2:
        badminton()
    elif pilihan == 3:
        basket()
    elif pilihan == 4:
        voli()
    elif pilihan == 5:
        futsal()

    pilihan_pertandingan = int(input(f"Masukkan nomor pertandingan yang ingin anda beli: "))

    for kategori in data['Kategori']:
        if kategori['Nama Kategori'] == kategori_nama:
            if 0 < pilihan_pertandingan <= len(kategori['Jadwal']):

```

```

    pertandingan = kategori['Jadwal'][pilihan_pertandingan-1]
    print(f"\nAnda memilih pertandingan: {pertandingan['Pertandingan']}")
    print(f"Kategori: {pertandingan['Kategori']}")
    print(f"Waktu: {pertandingan['Tanggal/Waktu']}")
    print(f"Harga Tiket Ekonomi: Rp {pertandingan['Harga Tiket Ekonomi'];,}")
    print(f"Harga Tiket VIP: Rp {pertandingan['Harga Tiket VIP'];,}\n")
    while True:
        try:
            print("TERIMA KASIH SUDAH BERBELANJA!!!")
            input("Tekan Enter Untuk Menanjutkan...")
            menu_pelanggan()
            break
        except(ValueError, KeyboardInterrupt):
            print("Mohon masukkan huruf y atau n!!")
    else:
        print("Nomor pertandingan tidak valid")
        break
else:
    print("TOLONG MASUKKAN ANGKA 1 - 5 UNTUK MEMILIH PERTANDINGAN")

except ValueError:
    print("Masukkan angka yang valid!")
except KeyboardInterrupt:
    print("\n+=====+
+")
    print("| Mohon masukkan data yang valid dan jangan tekan ctrl + C! |")
    print("+=====+\n")
except Exception as e:
    print(f"Terjadi kesalahan: {e}")

def lihatSaldo3():

```

```

os.system("cls")

if 'username' not in globals():
    print("Silakan login terlebih dahulu!")
    return

try:
    users = loadDataUser()
    user_found = False

    for user in users:
        if user["Nama User"] == username:
            user_found = True
            saldo = user["Saldo"]
            print("=====")
            print(f"    SALDO ANDA ADALAH Rp {saldo:,} ")
            print("=====")

            print("\n=====")
            print("    1. Ingin menambahkan saldo?")
            print("    2. Kembali")
            print("=====")

            pilihan = int(input("Masukkan pilihan 1/2: "))
            if pilihan == 1:
                topUp4()
            elif pilihan == 2:
                menu_pelanggan()
            else:
                print("Nomor pilihan tidak valid")
            break

```



```

    if not user_found:
        print("Error: User tidak ditemukan")

except Exception as e:
    print(f"Terjadi kesalahan: {str(e)}")

def topUp4():
    os.system("cls")
    if 'username' not in globals():
        print("Silakan login terlebih dahulu!")
        return

    print("+=====+")
    print("|      TOP UP SALDO      |")
    print("+=====+")
    users = loadDataUser()
    tabel = PrettyTable()
    tabel.clear_rows()
    tabel.title = "====TOP UP SALDO===="
    tabel.field_names = ["NO", "PILIHAN"]
    tabel.add_row(["1", "Rp 50.000"])
    tabel.add_row(["2", "Rp 100.000"])
    tabel.add_row(["3", "Rp 500.000"])
    tabel.add_row(["4", "Rp 1.000.000"])
    tabel.add_row(["5", "Rp 10.000.000"])
    tabel.add_row(["6", "Kembali"])
    print(tabel)

    try:
        pilihan = int(input("Masukkan pilihan nominal (1-6): "))

        nominal = [50000, 100000, 500000, 1000000, 10000000]

```

```

if 1 <= pilihan <= 5:
    user_found = False
    for user in users:
        if user["Nama User"] == username:
            user_found = True
            konfirmasi = input(f"Anda akan top up sebesar Rp {nominal[pilihan-1]:,}. Lanjutkan? (y/n):
")
            if konfirmasi.lower() == 'y':
                user["Saldo"] += nominal[pilihan - 1]
                savedataUser(users)
                print(f"Top up berhasil!")
                print("=====")
                print(f"Saldo anda sekarang adalah Rp {user['Saldo']:,}")
                print("-----")
                print("          1. Top Up lagi          ")
                print("          2. Kembali              ")
                print("=====")
                pilihan2 = int(input("Masukkan pilihan: "))
                if pilihan2 == 1:
                    topUp4()
                elif pilihan2 == 2:
                    menu_pelanggan()
                else:
                    print("pilihan anda tidak valid atau tidak ada di menu")
                else:
                    print("Top up dibatalkan")
                break

    if not user_found:
        print("Error: User tidak ditemukan")

```

```

def cari5():
    os.system("cls")
    print("\n+=====+")
    print("|          CARI TIKET          |")
    print("+=====+")

    try:
        keyword = input("Masukkan kata kunci pencarian: ").lower()

        found = False
        table = PrettyTable()
        table.field_names = ["Kategori", "Pertandingan", "Jenis", "Waktu", "Tiket Ekonomi", "Tiket VIP"]

        for kategori in data['Kategori']:
            kategori_nama = kategori['Nama Kategori']
            for jadwal in kategori['Jadwal']:
                # Mencari berdasarkan nama kategori atau nama pertandingan
                if (keyword in kategori_nama.lower() or
                    keyword in jadwal['Pertandingan'].lower()):
                    table.add_row([
                        kategori_nama,
                        jadwal['Pertandingan'],
                        jadwal['Kategori'],
                        jadwal['Tanggal/Waktu'],
                        f"Rp {jadwal['Harga Tiket Ekonomi']:,}",
                        f"Rp {jadwal['Harga Tiket VIP']:,}"
                    ])
            found = True

        if found:
            print("\nHasil Pencarian:")

```

```

        print(table)
    else:
        print("\nTidak ditemukan hasil yang cocok dengan kata kunci pencarian.")

        input("\nTekan Enter untuk kembali ke menu...")

except KeyboardInterrupt:
    print("\nPencarian dibatalkan.")
except Exception as e:
    print(f"\nTerjadi kesalahan: {str(e)}")

def main():
    os.system("cls")
    while True:
        try:
            print("\n+===== MENU =====+")
            print("|          1. Register          |")
            print("|          2. Login user        |")
            print("|          3. Login admin       |")
            print("|          4. Keluar            |")
            print("+=====+")
            pilihan = int(input("Pilih opsi (1/2/3/4): "))
            if pilihan == 1:
                daftaruser()
            elif pilihan == 2:
                if loginUser():
                    menu_pelanggan()
                else:
                    print("Username atau Password salah, silahkan coba lagi")
            elif pilihan == 3:

```

```
        if loginAdmin() == True:
            khususAtmint()
            break
        else:
            loginAdmin()
    elif pilihan == 4:
        print("Program berakhir.")
        exit()
        break
    else:
        print("Pilihan tidak valid, coba lagi.")
except (ValueError, KeyboardInterrupt):
    print("Pilihan anda tidak valid, dan jangan menekan CTRL + C")

if __name__ == "__main__":
    main()
```

**Tabel 3.1** Source Kode

## **BAB IV PENUTUP**

### **4.1. Kesimpulan**

Dalam pengembangan program "Tiket Penjualan Event Olahraga" ini, kami berhasil membuat sistem sederhana yang mampu mengelola penjualan tiket tersebut. Program ini memiliki dua peran utama: admin dan pelanggan. Admin dapat mengelola tiket, termasuk menambahkan, memperbarui, dan menghapus tiket. Sementara itu, pelanggan dapat menggunakan fitur otentikasi, mengisi saldo E-Money, dan melakukan proses checkout.

Proses pembuatan program ini memberikan pemahaman dasar dalam beberapa konsep pemrograman, seperti penggunaan struktur data, pengolahan input, dan penyimpanan data dengan menggunakan file JSON. Program ini juga memanfaatkan modul eksternal, seperti `PrettyTable`, untuk membuat tampilan tabel yang informatif.

Secara keseluruhan, pengembangan program "Tiket Penjualan Event Olahraga" ini merupakan langkah awal yang baik untuk memahami dasar-dasar pemrograman dan membangun dasar untuk proyek-proyek yang lebih kompleks di masa depan. Program ini dapat diperluas dan ditingkatkan sesuai kebutuhan untuk menciptakan pengalaman yang lebih baik bagi pelanggan.

### **4.2. Saran**

Dalam pembuatan laporan ini, kami menyadari bahwa masih ada banyak ruang untuk perbaikan. Kami mengakui bahwa laporan ini belum sempurna, terutama dalam hal penulisan, tata letak, dan penggunaan kosa kata yang perlu ditingkatkan. Kami berharap para pembaca laporan ini untuk memberikan kritik, saran, dan masukan yang konstruktif agar kami dapat terus meningkatkan kualitas laporan ini dan laporan-laporan kami di masa depan. Harapan kami

adalah agar laporan ini bisa memberikan manfaat kepada siapa pun yang membacanya.

## DAFTAR PUSTAKA

Registrar, A. (2023, Oktober, 27). Python Dictionaries.

Diakses dari [https://www.w3schools.com/python/python\\_dictionaries.asp](https://www.w3schools.com/python/python_dictionaries.asp)

Registrar, A. (2023, Oktober, 27). Python JSON.

Diakses dari [https://www.w3schools.com/python/python\\_json.asp](https://www.w3schools.com/python/python_json.asp) Python string isalnum() method. (2023, Oktober 27).

Diakses dari [https://www.w3schools.com/python/ref\\_string\\_isalnum.asp](https://www.w3schools.com/python/ref_string_isalnum.asp)

W3Schools. (2023, Oktober 28). Python Try... Except.

Diakses dari [https://www.w3schools.com/python/python\\_try\\_except.asp](https://www.w3schools.com/python/python_try_except.asp)

[https://www.w3schools.com/python/python\\_file\\_write.asp](https://www.w3schools.com/python/python_file_write.asp)

[https://www.w3schools.com/python/python\\_datetime.asp](https://www.w3schools.com/python/python_datetime.asp)

<https://www.geeksforgeeks.org/mongodb-and-python/>

[https://www.w3schools.com/python/python\\_mongodb\\_getstarted.asp](https://www.w3schools.com/python/python_mongodb_getstarted.asp)

<https://www.mongodb.com/community/forums/t/issue-with-passing-a-variable-to-the-find-one-query/2505/2>



## LAMPIRAN

Tabel Kontribusi

Nama	Kontribusi	Bagian
Muhammad Fakhri Al-Kautsar (2409116081)	Membuat Flowchart	Flowchart
Raihan Fariz Novanto (2409116083)	Membuat Program	Program
Moreno Ferdinand Farhantino (2409116097)	Membuat Flowchart	Flowchart
Ahmad Samsul Arifin (2409116113)	Membuat Laporan	Laporan

**Tabel 3.2** Tabel Kontribusi