

**LAPORAN PROYEK AKHIR PRAKTIKUM
MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



SISTEM PERBELIAN TIKET KONSER

Oleh:

Kelompok 4 C2

IKHSAN	2409106118
ADE PASIHA TANGKE ALLO	2409106109
MARHSA AULIA FAKHRIZA	2409106116

**PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN
SAMARINDA 2024**

KATA PENGANTAR

Dengan mengucapkan puji syukur ke hadirat Allah SWT, yang telah melimpahkan rahmat, taufik, dan hidayah-Nya, sehingga kami dapat menyusun laporan ini dengan baik dan tepat waktu. Laporan ini disusun sebagai salah satu bentuk tanggung jawab dalam mendokumentasikan proses dan hasil dari proyek akhir mata kuliah Algoritma dan Pemrograman Dasar, khususnya menggunakan bahasa pemrograman Python. Semoga laporan ini bermanfaat bagi pembaca dan memberikan kontribusi positif dalam pengembangan pemahaman serta keterampilan dasar pemrograman.

Program ini adalah Aplikasi Pemesanan Tiket Konser berbasis Python, yang dirancang untuk mengelola penjualan tiket konser secara digital. Fitur utama aplikasi ini mencakup registrasi akun, login pengguna, dan akses berbeda berdasarkan peran pengguna (admin dan pengguna umum). Admin dapat menambahkan, melihat, mengedit, dan menghapus data konser, serta melihat statistik penjualan tiket. Sementara itu, pengguna umum dapat membeli tiket konser dan melihat tiket yang sudah dibeli. Program ini juga dilengkapi dengan penggunaan warna terminal untuk memperjelas antarmuka pengguna dan tabel interaktif untuk menampilkan informasi konser. Dengan pendekatan ini, aplikasi membantu meningkatkan efisiensi dan pengalaman dalam pemesanan tiket konser.

Selama proses pembuatan program penjualan tiket konser menggunakan Python, kami menghadapi beberapa kendala teknis, seperti kesulitan dalam mengimplementasikan logika pemesanan yang akurat, pengaturan database untuk penyimpanan data tiket, serta pengaturan tampilan antarmuka pengguna (UI) yang responsif. Namun, dengan kerja keras dan kolaborasi tim, kendala-kendala tersebut berhasil diatasi. Kami mengucapkan terima kasih kepada asisten yang telah memberikan arahan dan bimbingan yang sangat berarti, serta kepada rekan-rekan tim yang telah bekerja sama dengan baik untuk menyelesaikan proyek ini. Terima kasih

juga kepada pihak-pihak lain yang telah mendukung kami selama proses pembuatan program ini.

Demikianlah laporan ini kami susun sebagai bentuk tanggung jawab atas pelaksanaan proyek akhir mata kuliah Algoritma dan Pemrograman Dasar. Semoga aplikasi pemesanan tiket konser berbasis Python yang telah dikembangkan dapat memberikan manfaat dalam meningkatkan efisiensi pemesanan tiket secara digital. Kami berharap laporan ini dapat memberikan wawasan yang berguna dalam pengembangan keterampilan pemrograman lebih lanjut. Terima kasih atas segala dukungan yang telah diberikan.

Samarinda, 18 November 2024

Kelompok 4 C2

TAKARIR

<i>Database</i>	Basis Data
<i>Managemen</i>	Mengatur
<i>Input</i>	Memasukkan
<i>command line interface</i>	Antar Muka Baris Perintah
<i>dictionary</i>	Kamus
<i>VerifyOpen In EditorEditCopy</i>	Verifikasi Buka Di Editor Edit Salin
<i>If</i>	Jika
<i>Else</i>	Maka
<i>While</i>	Ketika
<i>password</i>	Kata Sandi
<i>Username</i>	Nama
<i>Output</i>	Keluaran
<i>PrettyTable</i>	Tabel Cantik
<i>None</i>	Tidak Ada
<i>Delete</i>	Menghapus
<i>pwinput</i>	modul platform
<i>try</i>	mencoba
<i>break</i>	keluar dari pengulangan
<i>exit</i>	keluar
<i>import</i>	memasukkan
<i>python</i>	bahasa pemrograman

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR	iv
DAFTAR ISI	v
DAFTAR GAMBAR	vi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional	2
1.3 Rumusan Masalah	3
1.4 Batasan Masalah	5
1.5 Tujuan	6
BAB II PERANCANGAN	9
2.1 Analisis Program	9
2.2 Flowchart	12
2.3 Konsep/Materi Praktikum yang dipakai	15
BAB III HASIL DAN PEMBAHASAN	18
3.1 Tampilan Program	18
3.2 Source Code	30
BAB IV PENUTUP	45
4.1 Kesimpulan	45
4.2 Saran	46
DAFTAR PUSTAKA	47
LAMPIRAN	48

DAFTAR GAMBAR

<u>Gambar 3.1 Tampilan awal</u>	Error! Bookmark not defined.
<u>Gambar 3.2 Tampilan Login Admin</u>	Error! Bookmark not defined.
<u>Gambar 3.3 Tampilan menu / Tambahkan Konser..</u>	Error! Bookmark not defined.
<u>Gambar 3.4 Tampilan Menu / Lihat Konser.....</u>	Error! Bookmark not defined.
<u>Gambar 3.5 Tampilan Menu / Edit Konser</u>	Error! Bookmark not defined.
<u>Gambar 3.6 Tampilan Menu / Hapus Konser</u>	Error! Bookmark not defined.
<u>Gambar 3.7 Tampilan Menu / Lihat Statistik.....</u>	Error! Bookmark not defined.
<u>Gambar 3.8 Tampilan Menu / Exit.....</u>	Error! Bookmark not defined.
<u>Gambar 3.9 Tampilan Pengguna / Beli tiket konser</u>	Error! Bookmark not defined.
<u>Gambar 3.10 Tampilan pengguna / Beli tiket konser</u>	Error! Bookmark not defined.
<u>Gambar 3.11 Tampilan Pengguna / Lihat tiket yang sudah dibeli</u>	Error! Bookmark not defined.
<u>Gambar 3.11 Tampilan Pengguna / Exit.....</u>	Error! Bookmark not defined.
<u>Gambar 4.1 Kertas Konsul</u>	Error! Bookmark not defined.
<u>Gambar 4.2 Kertas Konsul</u>	Error! Bookmark not defined.
<u>Gambar 4.3 Kertas Konsul</u>	Error! Bookmark not defined.
<u>Gambar 4.4 Kertas Konsul</u>	Error! Bookmark not defined.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Masalah dalam pembuatan program Aplikasi Pemesanan Tiket Konser ini berawal dari kebutuhan akan sistem pemesanan tiket yang praktis, efisien, dan terorganisasi. Dalam banyak kasus, pemesanan tiket secara manual sering menimbulkan beberapa permasalahan, seperti antrian panjang, kesalahan pencatatan data, sulitnya melacak penjualan tiket, hingga risiko kehilangan data tiket. Selain itu, dengan semakin tingginya minat masyarakat terhadap konser, dibutuhkan sebuah platform yang dapat mengakomodasi pembelian tiket dengan mudah dan cepat.

Secara umum, masalah yang mendasari dibuatnya program ini meliputi:

1. Kurangnya efisiensi dalam pengelolaan data konser dan tiket. Pencatatan manual membutuhkan waktu lama dan rawan kesalahan.
2. Minimnya transparansi informasi konser. Pengguna sering kali kesulitan mendapatkan informasi lengkap terkait lokasi, waktu, dan harga tiket konser.
3. Terbatasnya aksesibilitas pembelian tiket. Sistem konvensional tidak memungkinkan pembelian tiket secara mandiri oleh pengguna kapan saja.
4. Sulitnya melacak statistik penjualan tiket. Pengelola membutuhkan alat yang dapat memberikan laporan penjualan tiket secara otomatis.

Dengan dasar tersebut, aplikasi ini dirancang untuk menjadi solusi digital yang tidak hanya mempermudah proses pemesanan tiket tetapi juga mendukung pengelolaan konser dengan lebih profesional dan modern.

1.2 Kebutuhan Fungsional

Kebutuhan Fungsional Program Pemesanan Tiket Konser

1. Pengguna

Registrasi Akun: Pengguna dapat mendaftar untuk membuat akun baru dengan memasukkan username dan password.

Login: Pengguna dapat masuk ke akun menggunakan username dan password yang telah terdaftar.

Beli Tiket Konser: Pengguna dapat melihat daftar konser yang tersedia dan membeli tiket untuk konser tertentu.

Lihat Tiket yang Sudah Dibeli: Pengguna dapat melihat tiket yang telah dibeli sebelumnya.

2. Admin

Menampilkan Menu Admin: Admin dapat melihat menu khusus untuk mengelola konser.

Tambah Konser: Admin dapat menambahkan konser baru dengan informasi judul, lokasi, tanggal, dan harga tiket.

Lihat Konser: Admin dapat melihat daftar semua konser yang tersedia dalam format tabel.

Edit Konser: Admin dapat mengedit informasi konser yang sudah ada (judul, lokasi, tanggal, harga).

Hapus Konser: Admin dapat menghapus konser dari daftar konser yang tersedia.

Lihat Statistik: Admin dapat melihat statistik penjualan tiket, termasuk total konser, total tiket terjual, dan total pendapatan.

3. Interaksi Pengguna

Tampilan Menu Utama: Program harus menampilkan menu utama yang memungkinkan pengguna untuk memilih opsi registrasi, login, atau keluar.

Konfirmasi Aksi: Setelah melakukan aksi seperti mengedit atau menghapus konser, program harus meminta konfirmasi dari admin.

Pesan Kesalahan: Program harus memberikan pesan kesalahan jika pengguna memasukkan username atau password yang salah saat login.

4. Data dan Penyimpanan

Penyimpanan Data Pengguna: Program harus menyimpan data pengguna (username, password, role) dalam struktur data yang sesuai.

Penyimpanan Data Tiket Konser: Program harus menyimpan informasi tentang konser (judul, lokasi, tanggal, harga) dalam struktur data yang sesuai.

5. Statistik dan Laporan

Program harus menghitung dan menampilkan statistik terkait penjualan tiket secara otomatis setelah transaksi dilakukan.

1.3 Rumusan Masalah

1. Kesulitan dalam Akses Informasi Konser:

Banyak pengguna yang kesulitan menemukan informasi terkini mengenai konser yang akan datang, termasuk lokasi, tanggal, dan harga tiket. Hal ini

dapat menyebabkan kebingungan dan kehilangan kesempatan untuk membeli tiket konser yang diinginkan.

2. Proses Booking Tiket yang Rumit:

Proses booking tiket konser sering kali rumit dan tidak efisien, terutama jika dilakukan secara manual. Pengguna membutuhkan sistem yang memudahkan mereka untuk membeli tiket dengan cepat dan mudah.

3. Manajemen Tiket yang Tidak Efisien:

Pengguna yang telah membeli tiket sering kali kesulitan mengelola dan melacak tiket yang telah dibeli. Aplikasi ini perlu menyediakan fitur yang memungkinkan pengguna untuk melihat tiket yang telah mereka beli dengan jelas.

4. Keterbatasan dalam Pengelolaan Konser oleh Admin:

Admin konser membutuhkan alat yang efektif untuk mengelola konser, termasuk menambah, mengedit, dan menghapus konser. Tanpa sistem yang baik, proses ini menjadi lebih sulit dan memakan waktu.

5. Statistik Penjualan yang Tidak Terjangkau:

Tanpa adanya laporan atau statistik penjualan tiket, admin tidak dapat mengevaluasi kinerja konser atau mengambil keputusan strategis untuk konser mendatang. Informasi ini sangat penting untuk pengelolaan keuangan dan perencanaan acara di masa depan.

6. Keamanan Data Pengguna:

Dengan adanya registrasi akun, penting untuk memastikan bahwa data pengguna, termasuk username dan password, terlindungi dengan baik.

Masalah keamanan data ini perlu diatasi untuk memberikan rasa aman bagi pengguna saat menggunakan aplikasi.

7. Keterbatasan dalam Fitur Registrasi dan Login:

Proses registrasi dan login yang tidak intuitif dapat mengakibatkan pengguna baru merasa frustrasi. Aplikasi harus menyediakan pengalaman pengguna yang baik untuk menarik lebih banyak pengguna.

1.4 Batasan Masalah

1. Fokus pada Pengelolaan Tiket Konser:

Pembahasan akan terbatas pada pengelolaan tiket konser, termasuk pembuatan, pengeditan, penghapusan, dan pembelian tiket. Aspek lain dari manajemen acara, seperti promosi dan pemasaran konser, tidak akan dibahas.

2. Pengguna Terbatas pada Dua Peran:

Aplikasi ini hanya akan mengakomodasi dua jenis pengguna, yaitu admin dan pengguna biasa. Tidak akan ada pembahasan tentang peran tambahan atau fitur untuk pengguna lain, seperti manajer acara atau penyelenggara.

3. Data Konser yang Tersedia:

Aplikasi ini hanya akan menggunakan data konser yang telah ditentukan sebelumnya dalam kode. Tidak akan ada integrasi dengan database eksternal atau API untuk mengambil data konser secara real-time.

4. Metode Pembayaran:

Pembahasan tidak akan mencakup detail mengenai metode pembayaran atau sistem pembayaran yang beragam. Aplikasi ini hanya akan fokus pada pembelian tiket tanpa implementasi sistem pembayaran yang kompleks.

5. Platform dan Lingkungan Pengembangan:

Aplikasi ini dirancang untuk dijalankan di lingkungan terminal (*command line interface*) dan tidak akan membahas pengembangan antarmuka pengguna grafis (GUI) atau aplikasi berbasis web.

6. Keamanan dan Privasi Data:

Meskipun aplikasi ini menyimpan informasi pengguna, pembahasan tidak akan mendalami aspek keamanan dan privasi data secara mendalam, seperti enkripsi data atau perlindungan terhadap serangan siber.

7. Statistik Penjualan yang Sederhana:

Statistik yang ditampilkan dalam aplikasi akan terbatas pada total konser, total tiket terjual, dan total pendapatan. Analisis yang lebih mendalam, seperti analisis tren atau laporan penjualan per konser, tidak akan menjadi fokus

1.5 Tujuan

1. Menyediakan Platform Pemesanan Tiket yang Efisien:

Membangun aplikasi yang memudahkan pengguna dalam melakukan pemesanan tiket konser secara cepat dan efisien, sehingga mereka dapat dengan mudah menemukan dan membeli tiket yang diinginkan.

2. Mempermudah Pengelolaan Konser bagi Admin:

Mengembangkan fitur yang memungkinkan admin untuk mengelola konser dengan mudah, termasuk menambah, mengedit, dan menghapus data konser tanpa kesulitan, sehingga meningkatkan efisiensi operasional.

3. Meningkatkan Akses Informasi Konser:

Menyediakan informasi terkini mengenai konser yang akan datang, termasuk lokasi, tanggal, dan harga tiket, agar pengguna dapat membuat keputusan yang tepat saat membeli tiket.

4. Menawarkan Fitur Pelacakan Tiket yang Jelas:

Memberikan pengguna kemampuan untuk melihat tiket yang telah mereka beli dengan mudah, sehingga mereka dapat mengelola tiket konser mereka tanpa kebingungan.

5. Menyediakan Statistik Penjualan yang Berguna:

Mengimplementasikan fitur statistik yang memberikan admin informasi tentang total konser, total tiket terjual, dan total pendapatan, sehingga mereka dapat melakukan evaluasi dan perencanaan yang lebih baik untuk konser mendatang.

6. Meningkatkan Keamanan dan Privasi Data Pengguna:

Mengimplementasikan langkah-langkah dasar untuk melindungi data pengguna, termasuk username dan password, guna memberikan rasa aman kepada pengguna saat menggunakan aplikasi.

7. Meningkatkan Pengalaman Pengguna:

Mengembangkan antarmuka yang intuitif dan mudah digunakan, sehingga pengguna baru maupun yang sudah berpengalaman dapat dengan mudah menavigasi aplikasi dan menggunakan semua fitur yang tersedia.

8. Mendorong Partisipasi Pengguna Baru:

Menciptakan proses registrasi dan login yang sederhana dan cepat, sehingga lebih banyak pengguna baru yang tertarik untuk menggunakan aplikasi ini.

9. Memfasilitasi Pertumbuhan dan Pengembangan Aplikasi:

Menyusun dasar yang kuat untuk pengembangan lebih lanjut dari aplikasi, termasuk penambahan fitur baru dan peningkatan keamanan di masa depan berdasarkan umpan balik pengguna.

BAB II

PERANCANGAN

2.1 Analisis Program

Program ini adalah aplikasi berbasis teks untuk pemesanan tiket konser. Pengguna dapat mendaftar akun, login, dan melakukan berbagai tindakan berdasarkan peran mereka (admin atau pengguna biasa). Admin memiliki akses untuk mengelola konser, sedangkan pengguna biasa dapat membeli tiket konser dan melihat tiket yang telah dibeli.

2. Alur Program

1. Menampilkan Menu Utama:

Program dimulai dengan menampilkan menu utama yang menawarkan opsi untuk mendaftar akun, login, atau keluar dari aplikasi.

2. Registrasi Akun:

Jika pengguna memilih untuk mendaftar, mereka diminta untuk memasukkan *username* dan *password*. Jika *username* sudah ada, pengguna akan diminta untuk mencoba lagi. Jika berhasil, akun baru akan dibuat.

3. Login:

Pengguna dapat login dengan memasukkan *username* dan *password*. Jika berhasil, program akan menentukan peran pengguna (admin atau pengguna biasa) dan menampilkan menu yang sesuai.

4. Menu Admin:

Admin memiliki opsi untuk:

1. Menambah konser baru.
2. Melihat daftar konser yang ada.
3. Mengedit informasi konser yang sudah ada.
4. Menghapus konser dari daftar.
5. Melihat statistik penjualan tiket.
6. Keluar dari menu admin.

5. Menu Pengguna:

Pengguna biasa memiliki opsi untuk:

1. Membeli tiket konser.
2. Melihat tiket yang telah dibeli.
3. Keluar dari menu pengguna.

6. Fitur Booking Tiket:

Saat pengguna memilih untuk booking tiket, program akan menampilkan daftar konser yang tersedia. Pengguna dapat memilih konser yang diinginkan, dan setelah konfirmasi, tiket akan ditambahkan ke akun pengguna.

7. Melihat Tiket:

Pengguna dapat melihat tiket yang telah mereka beli. Jika belum ada tiket yang dibeli, program akan memberikan notifikasi bahwa pengguna belum memiliki tiket.

8. Statistik Penjualan (Admin):

Admin dapat melihat statistik seperti total konser, total tiket terjual, dan total pendapatan yang dihasilkan dari penjualan tiket.

9. Keluar dari Aplikasi:

Pengguna dapat memilih untuk keluar dari aplikasi. Program akan menanyakan konfirmasi sebelum menutup aplikasi.

3. Penjelasan Alur Program

Struktur Program:

1. Program dibangun menggunakan fungsi untuk memisahkan logika dan fungsionalitas, seperti registrasi, login, dan pengelolaan konser.
2. Penggunaan *dictionary* untuk menyimpan data pengguna dan konser memudahkan akses dan manipulasi data.

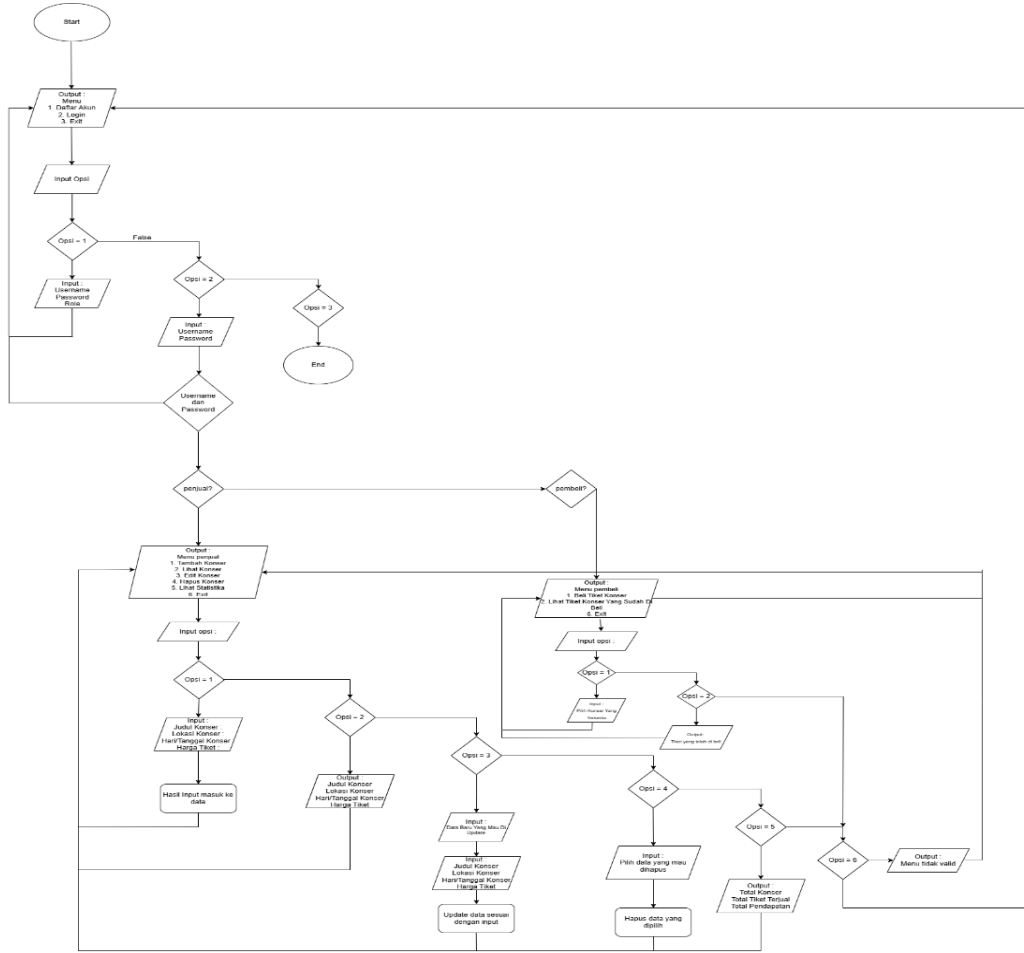
Interaksi Pengguna:

3. Program berinteraksi dengan pengguna melalui input dan output di terminal, memberikan umpan balik yang sesuai berdasarkan tindakan pengguna.
4. Warna teks ditambahkan untuk meningkatkan pengalaman pengguna dan membedakan antara pesan informasi, kesalahan, dan konfirmasi.

Pengelolaan Data:

5. Data konser dan pengguna disimpan dalam struktur data sederhana, memungkinkan program untuk melakukan pembaruan dan penghapusan dengan mudah.
6. Statistik penjualan diperbarui secara real-time setiap kali tiket dibeli.

2.2 Flowchart



Gambar 2.1 *Flowchart*

1. Start

Ini adalah titik awal dari diagram alur.

- ## 2. Output

- Di sini, sistem akan menampilkan beberapa opsi output yang dapat dipilih oleh pengguna, seperti:

1. Menu panggil
2. Ubah Konser

3. Hapus Konser

4. Input Data

5. Logout

3. Input Opsi

Pada tahap ini, pengguna akan memilih salah satu opsi yang ditampilkan pada tahap sebelumnya.

4. Opsi = 1

Jika pengguna memilih opsi 1, maka proses akan berlanjut ke tahap berikutnya.

5. Input Username Password

Pengguna akan diminta untuk memasukkan username dan password.

6. Opsi = 2

Jika pengguna memilih opsi 2, maka proses akan berlanjut ke tahap berikutnya.

7. Input Username Password

Pengguna akan diminta untuk memasukkan username dan password kembali.

8. Opsi = 3

Jika pengguna memilih opsi 3, maka proses akan berlanjut ke tahap berikutnya.

9. Input Username Password

Pengguna akan diminta untuk memasukkan username dan password sekali lagi.

10. Opsi = 4

Jika pengguna memilih opsi 4, maka proses akan berlanjut ke tahap berikutnya.

11. Input Data Sesuai Konser

Pada tahap ini, pengguna akan diminta untuk memasukkan data yang sesuai dengan konser yang dipilih.

12. Update data sesuai konser

Setelah data dimasukkan, sistem akan memperbarui data sesuai dengan konser yang dipilih.

13. Hapus data yang dipilih

Jika pengguna memilih untuk menghapus data, maka sistem akan menghapus data yang dipilih.

14. Opsi = 5

Jika pengguna memilih opsi 5, maka proses akan berakhir dengan "End".

15. End

Ini adalah titik akhir dari diagram alur.

Secara keseluruhan, diagram ini menggambarkan alur proses autentikasi pengguna dan berbagai tindakan yang dapat dilakukan setelah login berhasil, seperti melihat menu, mengubah atau menghapus konser, serta *input* data baru.

2.3 Konsep/Materi Praktikum yang dipakai

Dalam pembuatan aplikasi pemesanan tiket konser ini, kami menggunakan berbagai konsep pemrograman yang mendasari fungsionalitas dari aplikasi tersebut. Berikut adalah beberapa konsep/materi praktikum yang kami terapkan:

1. Fungsi Dasar

Fungsi dasar yang digunakan dalam aplikasi ini adalah fungsi **print()**. Fungsi ini digunakan untuk menampilkan informasi kepada pengguna, seperti menu, pesan konfirmasi, dan hasil dari berbagai operasi yang dilakukan dalam aplikasi. Contohnya:

python

[Verify](#)[Open In Editor](#)[Edit](#)[Copy code](#)

```
1 print(style.CGREEN2 + "Halo Pengguna baru! Ayo buat akun dulu")
```

2. Struktur Data

Kami menggunakan struktur data yang berbeda untuk menyimpan informasi pengguna dan tiket konser:

Dictionary: Untuk menyimpan akun pengguna dan tiket konser, kami menggunakan dictionary. Ini memungkinkan kami untuk mengakses dan memanipulasi data dengan cepat berdasarkan kunci. Contoh:

python

[Verify](#)[Open In Editor](#)[Edit](#)[Copy code](#)

```
1 akun_pengguna = {
```

```
2     "admin": {'password': '1', 'role': 'admin', 'tiket': []},
```

```
3  "pengguna": {'password': '2', 'role': 'pengguna', 'tiket': []}
```

```
4 }
```

3. Pengendalian Alur Program

Kami menggunakan struktur pengendalian alur seperti *if*, *else*, dan *while* untuk mengatur alur program. Ini memungkinkan aplikasi untuk merespons input pengguna dan menjalankan fungsi yang sesuai. Contoh:

python

VerifyOpen In EditorEditCopy code

```
1 if opsi == "1":
```

```
2     registrasi_akun()
```

4. Fungsi

Kami mendefinisikan beberapa fungsi untuk memecah program menjadi bagian-bagian yang lebih kecil dan lebih mudah dikelola. Setiap fungsi memiliki tanggung jawab tertentu, seperti **registrasi_akun()**, **login_akun()**, **tambah_konser()**, dan lain-lain. Contoh:

python

VerifyOpen In EditorEditCopy code

```
1 def registrasi_akun():
```

```
2     # Logika untuk registrasi akun
```

5. Penanganan Input dan Validasi

Aplikasi ini juga menangani input dari pengguna dan melakukan validasi untuk memastikan bahwa input tersebut valid. Misalnya, saat pengguna mencoba untuk

mendaftar dengan username yang sudah ada, aplikasi memberikan feedback yang sesuai.

6. Penggunaan Library Eksternal

Kami menggunakan library eksternal PrettyTable untuk menampilkan data dalam format tabel yang lebih terstruktur dan mudah dibaca. Ini meningkatkan pengalaman pengguna saat melihat daftar konser.

7. Penggunaan Variabel Global

Kami menggunakan variabel global untuk menyimpan total konser, total tiket terjual, dan total pendapatan. Ini memungkinkan kami untuk mengakses dan memodifikasi nilai-nilai ini di seluruh fungsi dalam program.

8. Penggunaan Kelas

Kami mendefinisikan kelas style untuk menyimpan kode warna yang digunakan untuk mempercantik tampilan output di terminal. Ini memberikan pengalaman pengguna yang lebih baik dengan memberikan visual yang menarik.

9. Modularisasi

Program ini dirancang dengan cara yang modular, sehingga setiap bagian dari aplikasi dapat dikembangkan dan diuji secara terpisah. Ini membuat pemeliharaan dan pengembangan aplikasi lebih mudah.

10. Penggunaan Loop

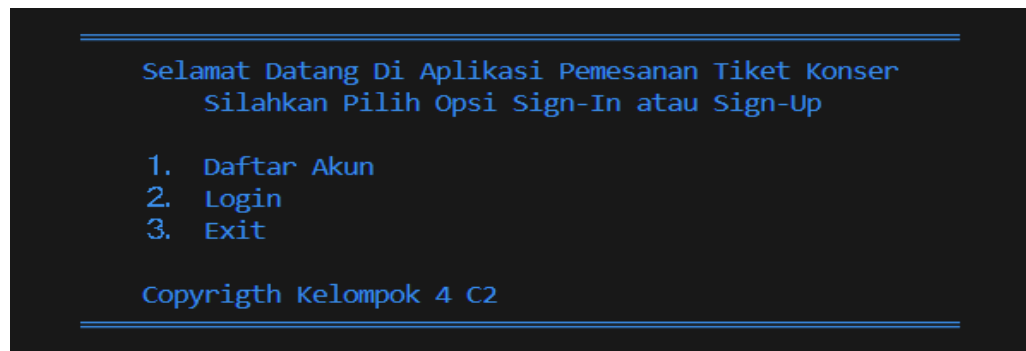
Kami menggunakan loop **while** untuk menjaga aplikasi tetap berjalan hingga pengguna memilih untuk keluar. Ini memungkinkan pengguna untuk melakukan beberapa operasi tanpa harus memulai aplikasi dari awal setiap kali.

BAB III

HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Awal Program



Gambar 3.1 Tampilan awal

Penjelasan:

Daftar akun:

Pengguna Baru: Pertama, fungsi ini menyapa pengguna baru dan meminta mereka untuk memasukkan username atau daftar akun.

Cek *Username*: Setelah itu, fungsi memeriksa apakah username yang dimasukkan sudah ada dalam akun_pengguna. Jika sudah ada, sistem akan memberikan pesan bahwa nama tersebut sudah terpakai.

Input *Password*: Jika username belum terpakai, pengguna diminta untuk memasukkan password.

Menyimpan Akun: Setelah mendapatkan username dan *password*, akun baru disimpan dalam dictionary akun_pengguna dengan role sebagai 'pengguna' dan tiket sebagai list kosong.

Konfirmasi: Fungsi ini kemudian memberikan konfirmasi bahwa akun berhasil terdaftar.

Login:

Penjelasan:

Sapa Pengguna: Fungsi ini menyapa pengguna dan meminta mereka untuk memasukkan *username* dan *password*.

Cek Kredensial: Fungsi memeriksa apakah *username* ada dalam *akun_pengguna* dan apakah *password* yang dimasukkan sesuai dengan yang terdaftar.

Login Berhasil: Jika kredensial benar, fungsi akan mengembalikan *username*, menandakan bahwa pengguna berhasil login.

Login Gagal: Jika kredensial salah, fungsi akan memberikan pesan kesalahan dan mengembalikan *None*, yang menandakan bahwa login gagal.

Exit:

Penjelasan:

Konfirmasi Keluar: Ketika pengguna memilih opsi "3" untuk keluar, aplikasi akan meminta konfirmasi apakah pengguna benar-benar ingin keluar.

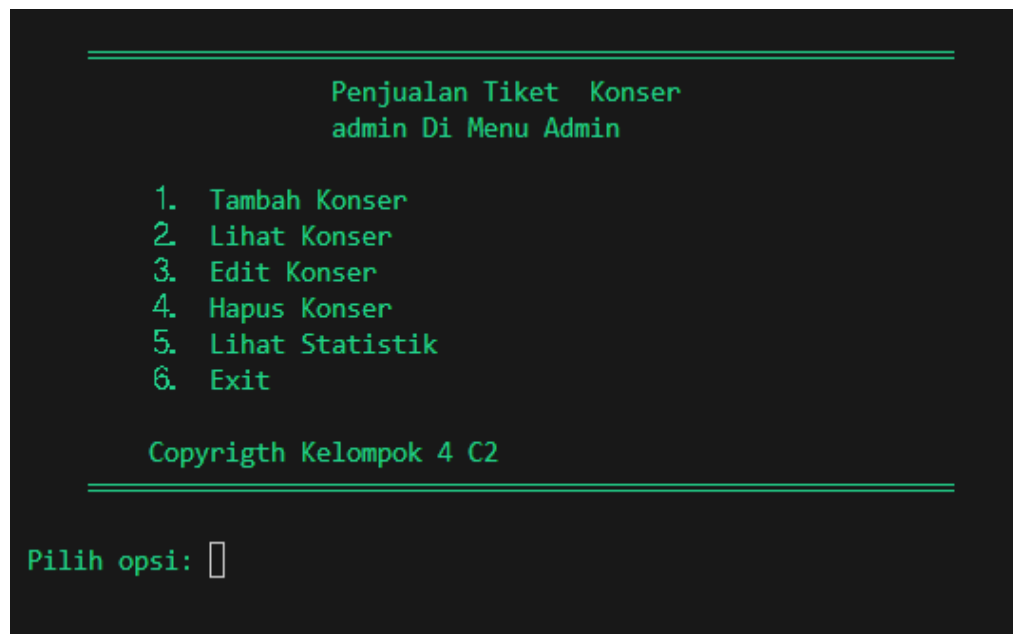
Pilihan Iya atau Tidak: Pengguna diberikan dua pilihan: "1" untuk keluar dan "2" untuk tetap berada di aplikasi.

Keluar dari Aplikasi: Jika pengguna memilih "1", aplikasi akan menampilkan pesan terima kasih dan keluar dari loop utama dengan *break*.

Tetap di Aplikasi: Jika pengguna memilih "2", aplikasi akan kembali ke menu utama.

Input Tidak Valid: Jika pengguna memberikan input selain "1" atau "2", aplikasi akan memberikan pesan kesalahan dan meminta input ulang.

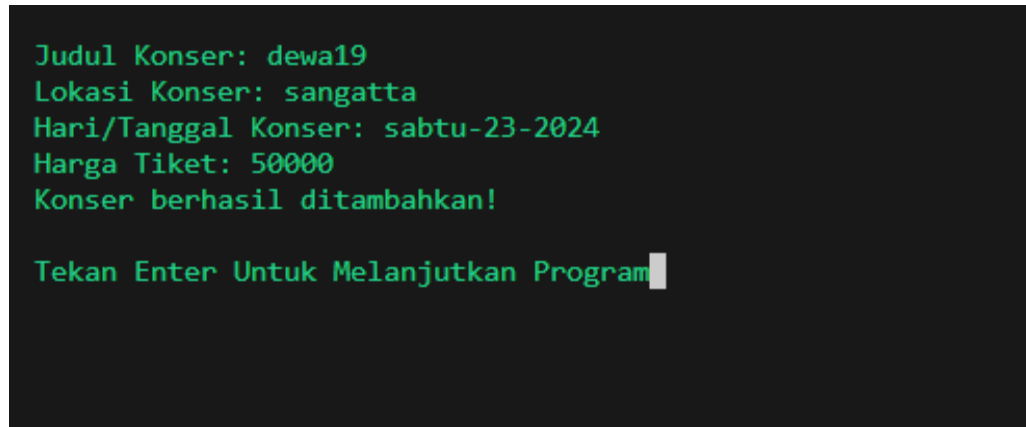
2. Tampilan Login Admin



Gambar 3.2 Tampilan Login Admin

Ketika seorang admin berhasil login, mereka akan disambut dengan sebuah menu yang dirancang untuk mengelola seluruh aspek penjualan tiket konser. Menu ini memungkinkan admin untuk melakukan berbagai fungsi penting seperti menambahkan konser baru ke dalam sistem, melihat daftar konser yang tersedia, mengedit detail konser yang sudah ada, menghapus konser tertentu, dan melihat statistik penjualan keseluruhan.

3 . Tampilan Pilihan Menu Lihat Menu Admin



Gambar 3.3 Tampilan menu / Tambahkan Konser

Saat memilih opsi 1 yaitu MenambahkanKonser kita diminta untuk memasukan judul konser yang ingin kita masukan, dan selanjutnya menambahkan Lokasi sesuai keinginan kita, dan masukan hari/ tanggal konser sesuai keinginan kita dan masukan harga tiket sesuai keinginan kita.

4. Tampilan Pilihan Menu Admin



Judul Konser	Lokasi Konser	Hari/Tanggal Konser	Harga Tiket
Lyodra	Samarinda	11-Oktober-2024	200000
Tulus	Balikpapan	12-Oktober-2024	500000
Bruno Mars	Jakarta	4-Desember-2024	3000000
Denny Caknan	Samarinda	1-Januari-2025	80000
Payung Teduh	Bandung	4-Januari-2025	200000

Tekan Enter Untuk Melanjut Pemrograman!

Gambar 3.4 Tampilan Menu / Lihat Konser

Saat memilih opsi 2 kita bisa melihat list konser tersebut yang dimana terdapat judul konser, Lokasi konser, hari/ tanggal konser, dan harga konser.

5. Tampilan Pilihan Menu Admin

```
Daftar Konser:
1. Lyodra
2. Tulus
3. Bruno Mars
4. Denny Caknan
5. Payung Teduh
Masukkan nomor konser yang ingin diedit: 5
Masukkan judul yang baru: Payung Teduh
Masukkan lokasi yang baru: Semarang
Masukkan hari/tanggal konser baru: 19-Februari-2025
Masukkan harga yang baru: 50000
Apa kamu yakin ingin mengedit konser ?
1. Iya
2. Tidak
Pilih: 1
Konser yang kamu pilih sudah di edit ya!

Tekan Enter Untuk Melanjutkan Program
```

Gambar 3.5 *Tampilan Menu / Edit Konser*

Saat memilih opsi 3 atau edit konser kita diminta untuk memasukan nomor berapa yang mau diedit, selanjutnya kita memasukan judul yang baru dan menambahkan Lokasi, hari/tanggal, dan harga yang kita inginkan. Setelah itu pilih opsi 1 atau iya untuk mengeditnya, jika tidak pilih opsi 2.

6. Tampilan Pilihan Menu Admin

```

      _____
      |  J A N U A R I 2 0 2 4  |
      |_____|

+-----+-----+-----+-----+
| Judul Konser | Lokasi Konser | Hari/Tanggal Konser | Harga Tiket |
+-----+-----+-----+-----+
| Lyodra       | Samarinda     | 11-Okttober-2024    | 200000      |
| Tulus        | Balikpapan    | 12-Okttober-2024    | 500000      |
| Bruno Mars   | Jakarta       | 4-Desember-2024     | 3000000     |
| Denny Caknan | Samarinda     | 1-Januari-2025      | 80000       |
| Payung Teduh | Semarang      | 19-Februari-2025    | 50000       |
+-----+-----+-----+-----+

Tekan Enter Untuk Melanjut Pemrograman!
Masukkan judul konser yang ingin dihapus: Payung Teduh
Apa kamu yakin ingin menghapus konser ?
1. Iya
2. Tidak
Pilih: 1
Konser yang kamu pilih sudah dihapus!

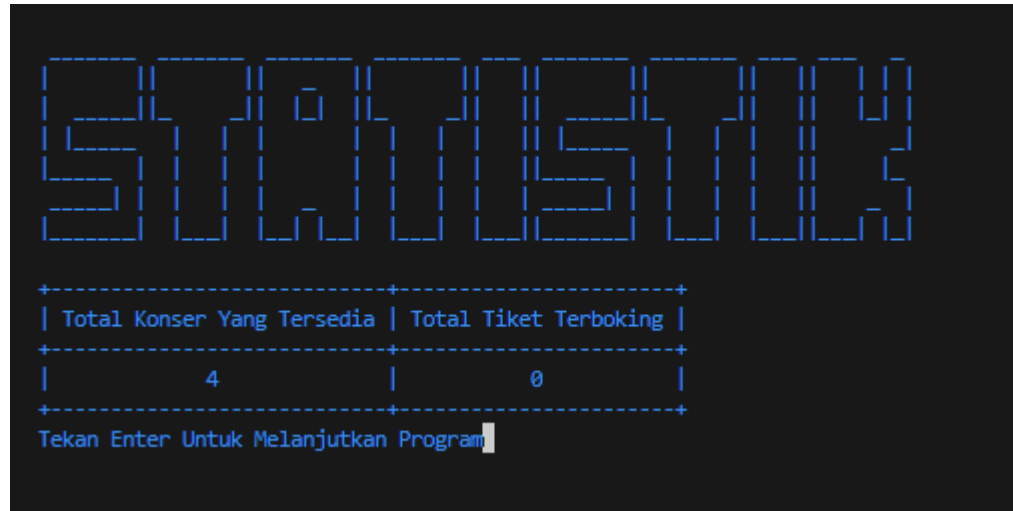
Tekan Enter Untuk Melanjutkan Program

```

Gambar 3.6 Tampilan Menu / Hapus Konser

Saat memilih opsi 4 kita diminta untuk memasukan judul konser yang ingin dihapus, setelah memasukan judul yang ingin dihapus terdapat pilihan 1 dan 2, jika memilih opsi 1 maka judul akan terhapus jika memilih opsi 2 maka judul tidak jadi dihapus.

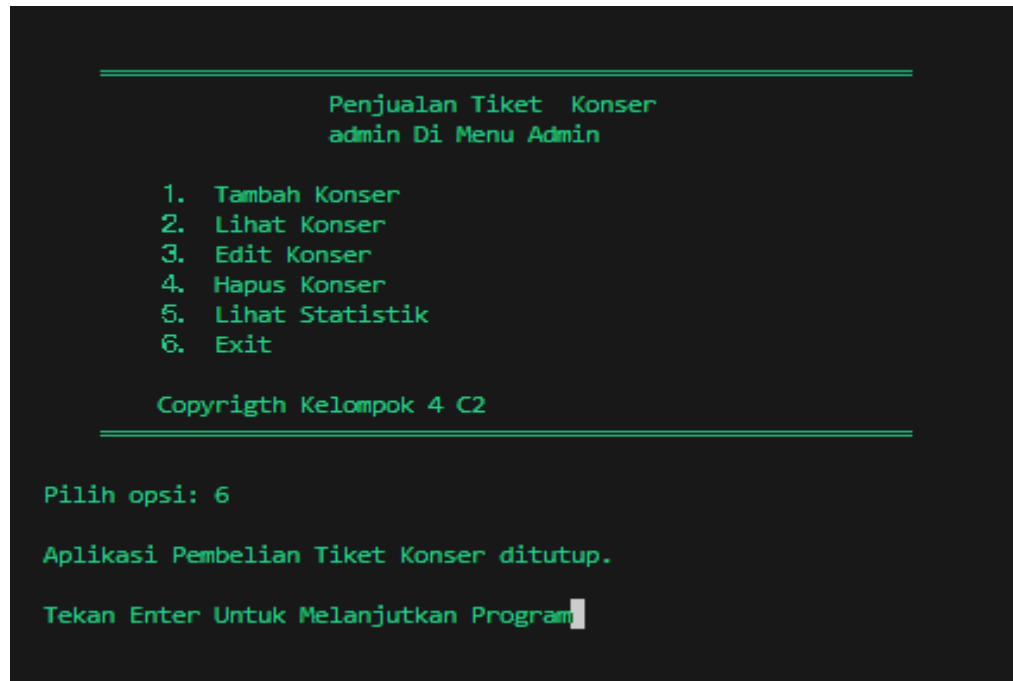
7. Tampilan Pilihan Menu Admin



Gambar 3.7 Tampilan Menu / Lihat Statistik

Saat memilih opsi 5, admin bisa melihat berapa total konser, total tiket terboking dan lanjut enter untuk melanjutkan program.

8. Tampilan Pilihan Menu Admin



```
Penjualan Tiket Konser
admin Di Menu Admin

1. Tambah Konser
2. Lihat Konser
3. Edit Konser
4. Hapus Konser
5. Lihat Statistik
6. Exit

Copyrighth Kelompok 4 C2

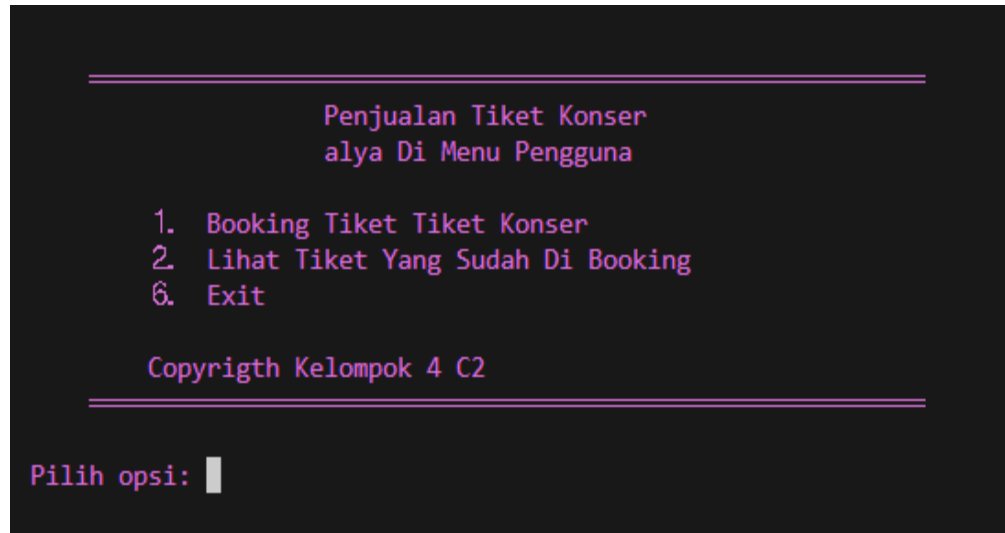
Pilih opsi: 6

Aplikasi Pembelian Tiket Konser ditutup.
Tekan Enter Untuk Melanjutkan Program
```

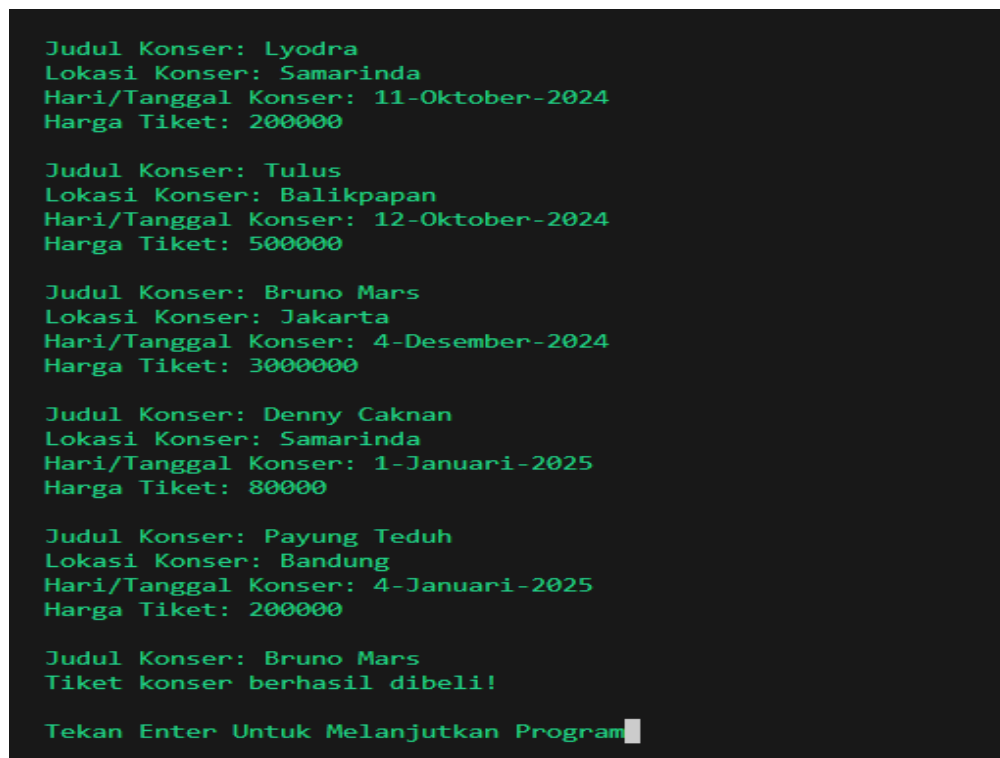
Gambar 3.8 Tampilan Menu / Exit

Saat memilih opsi 6, otomatis aplikasih pembelian tiket konser ditutup, dan langsung Kembali ke awal program.

9. Tampilan Menu Pilihan Pengguna



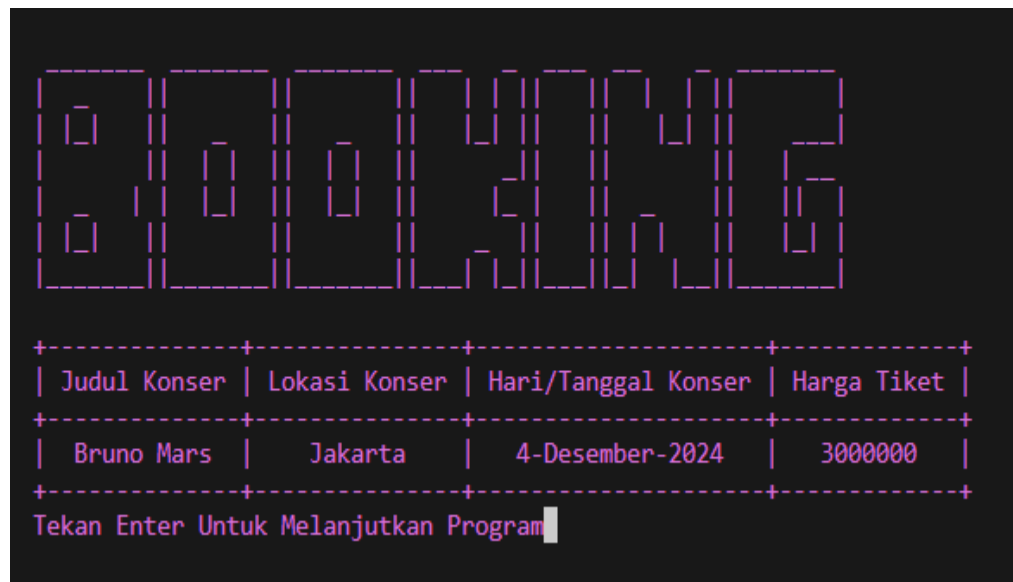
Gambar 3.9 Tampilan Pengguna / Beli tiket konser



Gambar 3.10 Tampilan pengguna / Beli tiket konser

Saat memilih opsi 1 terdapat lima judul konser yang ingin kita beli seperti Tulus, Bruno Mars, setelah memilih konser yang diinginkan tekan enter untuk melanjutkan.

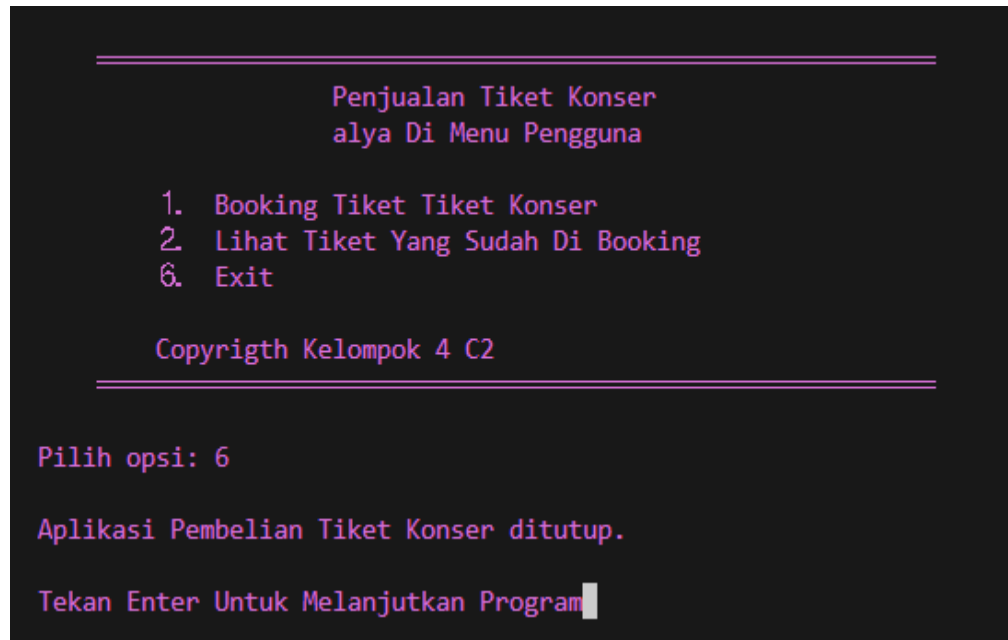
10. Tampilan Menu Lihat Pengguna



Gambar 3.11 Tampilan Pengguna / Lihat tiket yang sudah dibeli

Saat memilih opsi 2 kita dapat melihat tiket yang sudah kita beli.

11. Tampilan Menu Lihat Pengguna



Gambar 3.11 Tampilan Pengguna / Exit

Saat memilih opsi 6, otomatis aplikasih pembelian tiket konser ditutup, dan langsung Kembali ke awal program.

3.2 Source Code

```
# Memanggil Fitur Menghapus Terminal Secara Otomatis
import os

# Memanggil Fitur Privasi Password
import pwinput

# Memanggil fitur PrettyTable (Membuat table)
from prettytable import PrettyTable

# Inisialisasi data pengguna dan data tiket konser
akun_pengguna = {
    "admin": {'password': 'admin', 'role': 'admin', 'tiket': []},
}

# Data tiket konser yang tersedia
tiket_konser = {
    "Lyodra": {'lokasi': 'Samarinda', 'tanggal': '11-Oktober-2024',
'harga': '200000'},
    "Tulus": {'lokasi': 'Balikpapan', 'tanggal': '12-Oktober-2024',
'harga': '500000'},
    "Bruno Mars": {'lokasi': 'Jakarta', 'tanggal': '4-Desember-2024',
'harga': '3000000'},
    "Denny Caknan": {'lokasi': 'Samarinda', 'tanggal': '1-Januari-
2025', 'harga': '80000'},
    "Payung Teduh": {'lokasi': 'Bandung', 'tanggal': '4-Januari-
2025', 'harga': '200000'}
}

# Menambahkan Warna
class style():
```

```

# Menghapus Warna
    CEND      = '\33[0m'
# Kode Warna
    CGREY     = '\33[90m'
    CRED2     = '\33[91m'
    CGREEN2   = '\33[92m'
    CYELLOW2  = '\33[93m'
    CBLUE2    = '\33[94m'
    CVIOLET2   = '\33[95m'
    CBEIGE2    = '\33[96m'
    CWHITE2   = '\33[97m'

# Variabel global (Lihat Statistik)
total_konser = len(tiket_konser)
total_tiket_terboking = 0

# Fitur Otomatis Clear Terminal
def hapus_terminal():
    os.system('cls')

# Menu Awal Sistem Pembelian Tiket Konser
def tampilkan_menu():
    hapus_terminal()
    print(style.CBLUE2 + """
    _____

                Selamat Datang Di Pemesanan Tiket Konser
                Silahkan Pilih Opsi Sign-In atau Sign-Up

    1.   Daftar Akun
    2.   Login

```

3. Exit

Copyrighth Kelompok 4 C2

```
        """)
# Fitur Registrasi Akun
def registrasi_akun():
    hapus_terminal()
    print(style.CGREEN2 + "Halo Pengguna baru! Ayo buat akun dulu")
    Username = input("Username: ")
    # Cek apakah username sudah ada atau digunakan
    if Username in akun_pengguna:
        print(style.CRED2 + "Nama Sudah Terpakai Untuk Registrasi  
Silahkan Coba Lagi")
    else:
        # Menentukan Password Yang Ingin Digunakan
        Password = pwininput.pwininput("Password: ")
        # Ini Di gunakan apa bila password kurang dari 8 karakter
        if len>Password) < 8:
            print(style.CRED2 + "\nPassword tidak boleh kurang dari 8  
karakter." + style.CEND)
        else:
            # Apabila password lebih dari 8, maka proses lanjut untuk  
menginput ke dalam (akun_pengguna) sebagai acc baru
            akun_pengguna[Username] = {'password': Password, 'role':  
'pengguna', 'tiket': []}
            print(style.CGREEN2 + f"Akun Anda berhasil terdaftar  
dengan ID: {Username}")
            input("Tekan Enter Untuk Melanjutkan Program")

# Fitur Login Akun
```

```
def login_akun():
    hapus_terminal()
    print(style.CYELLOW2 + "Hi, Silahkan login dulu ya!")
    Username = input("Username: ")
    Password = pwininput.pwininput("Password: ")
    # Proses ini untuk mencari akun yang sudah terdata di
    (akun_pengguna)
    if Username in akun_pengguna and
    akun_pengguna[Username]['password'] == Password:
        return Username
    else:
        print(style.CRED2 + "Username dan password anda salah,
    silahkan coba lagi\n")
        input("Tekan Enter Untuk Melanjutkan Program")
        return None

# Menu Tampilan Admin
def tampilkan_menu_admin(Username):
    hapus_terminal()
    print(style.CGREEN2 + f"""

    Pemesanan Tiket Konser
    {Username} Di Menu Admin

    1. Tambah Konser
    2. Lihat Konser
    3. Edit Konser
    4. Hapus Konser
    5. Lihat Statistik
    6. Exit
    """)
```

```

Copyrighth Kelompok 4 C2

"""

# Menu Tampilan Pengguna
def tampilkan_menu_pengguna(Username):
    hapus_terminal()
    print(style.CVIOLET2 + f"""

                                Pemesanan Tiket Konser
                                {Username} Di Menu Pengguna

1.   Booking Tiket Tiket Konser
2.   Lihat Tiket Yang Sudah Di Booking
6.   Exit

Copyrighth Kelompok 4 C2

                                """)

# Fitur Create (Membuat/Menambah)
def tambah_konser():
    hapus_terminal()

    # ini menambahkan ke variable global
    global total_konser

    judul_konser = input("Judul Konser: ")
    lokasi_konser = input("Lokasi Konser: ")
    tanggal_konser = input("Hari/Tanggal Konser: ")
    harga_tiket = input("Harga Tiket: ")

```



```

        tiket_konser[judul_konser] = {'lokasi': lokasi_konser, 'tanggal':
tanggal_konser, 'harga': harga_tiket}
        total_konser += 1
        print(style.CGREEN2 + "Konser berhasil ditambahkan!\n")
        input("Tekan Enter Untuk Melanjutkan Program")

# Fitur Read (Melihat/Menampilkan)
def lihat_konser():
    hapus_terminal()
    print(style.CBLUE2 + """
    _____ _ _ _ _ _ _ _ _ _ _
    |__ _|(|)| | | | | | / /
    | | _ | | _ _ | | | ' / _ _ _ _ _ _ _ _
    | | | | | / // _ \ | _ | | < / _ \ | ' _ \ / _ | / _ \ | ' _ | | | | |
    | | | | | < | _ / | | | . \ ( ) | | | | \ _ \ | _ / | |
    | | | | | \ \ \ _ \ | \ | | | \ \ \ _ / | | | | _ / \ _ | | |
    """)

    table = PrettyTable()
    table.field_names = ["Judul Konser", "Lokasi Konser",
"Hari/Tanggal Konser", "Harga Tiket"]
    for judul, tiket in tiket_konser.items():
        table.add_row([judul, tiket['lokasi'], tiket['tanggal'],
tiket['harga']])
    print(table)
    input("Tekan Enter Untuk Melanjut Pemrograman!")

# Fitur Update (Meperbaharui/Mengedit)
def edit_konser():
    hapus_terminal()
    try:
        if not tiket_konser:

```

```

        print(style.CRED2 + "Tidak ada konser yang bisa diedit.")
    else:
        print("Daftar Konser:")
        for i, (judul, tiket) in enumerate(tiket_konser.items()):
            print(f"{i+1}. {judul}")
        edit = int(input("Masukkan nomor konser yang ingin
diedit: ")) - 1
        if 0 <= edit < len(tiket_konser):
            judul_konser = list(tiket_konser.keys())[edit]
            judul_baru = input("Masukkan judul yang baru: ")
            lokasi_baru = input("Masukkan lokasi yang baru: ")
            tanggal_baru = input("Masukkan hari/tanggal konser
baru: ")

            harga_baru = int(input("Masukkan harga yang baru: "))
            print("Apa kamu yakin ingin mengedit konser ? ")
            print("1. Iya")
            print("2. Tidak")
            memastikan_edit = input("Pilih: ")
            if memastikan_edit == "1":
                del tiket_konser[judul_konser]
                tiket_konser[judul_baru] = {'lokasi':
lokasi_baru, 'tanggal': tanggal_baru, 'harga': harga_baru}
                print(style.CGREEN2 + "Konser yang kamu pilih
sudah di edit ya!\n")
                input("Tekan Enter Untuk Melanjutkan Program")
            elif memastikan_edit == "2":
                print(style.CRED2 + "Aksi untuk mengedit konser
dibatalkan")
                input("Tekan Enter Untuk Melanjutkan Program")
            else:
                print(style.CRED2 + "Mohon pilih '1' atau '2'")

```



```

table.field_names = ["Judul Konser", "Lokasi Konser",
"Hari/Tanggal Konser", "Harga Tiket"]
for judul, tiket in tiket_konser.items():
    table.add_row([judul, tiket['lokasi'], tiket['tanggal'],
tiket['harga']])
print(style.CGREEN2)
print(table)
input("Tekan Enter Untuk Melanjut Pemrograman!")

global total_konser
if not tiket_konser:
    print("Tidak ada konser yang bisa dihapus.")
    input("Tekan Enter Untuk Melanjutkan Program")
else:
    hapus = input("Masukkan judul konser yang ingin dihapus: ")
    if hapus in tiket_konser:
        print(f"Apa kamu yakin ingin menghapus konser ? ")
        print("1. Iya")
        print("2. Tidak")
        memastikan_hapus = input("Pilih: ")
        if memastikan_hapus == "1":
            del tiket_konser[hapus]
            total_konser -= 1
            print("Konser yang kamu pilih sudah dihapus!\n")
            input("Tekan Enter Untuk Melanjutkan Program")
        elif memastikan_hapus == "2":
            print("Aksi untuk menghapus konser dibatalkan")
            input("Tekan Enter Untuk Melanjutkan Program")
        else:
            print("Mohon pilih '1' atau '2'")
            input("Tekan Enter Untuk Melanjutkan Program")

```

```

        return

    else:
        print("Tidak ada konser yang kamu maksud, silahkan input
ulang.\n")
        input("Tekan Enter Untuk Melanjutkan Program")
        return

# Fitur Booking / Create (Membuat/Menambahkan)
def booking_tiket(Username):
    hapus_terminal()
    # Menambah data ke variable global
    global total_tiket_terboking
    # Untuk Memanggil data di variable lokal (tiket_konser) lalu di
print
    for judul, tiket in tiket_konser.items():
        print(style.CGREEN2 + f"Judul Konser: {judul}\nLokasi Konser:
{tiket['lokasi']}\nHari/Tanggal Konser: {tiket['tanggal']}\nHarga
Tiket: {tiket['harga']}\n")
        judul_konser = input("Judul Konser: ")
        # Jika Sudah memilih tiket yang di booking, maka akan masuk ke
data pengguna yang booking
        if judul_konser in tiket_konser:
            akun_pengguna[Username]['tiket'].append({'judul':
judul_konser, 'lokasi': tiket_konser[judul_konser]['lokasi'],
'tanggal': tiket_konser[judul_konser]['tanggal'], 'harga':
tiket_konser[judul_konser]['harga']})
            total_tiket_terboking += 1
            print(style.CGREEN2 + "Tiket konser berhasil dibeli!\n")
            input("Tekan Enter Untuk Melanjutkan Program")
        else:
            print(style.CRED2 + "Konser tidak tersedia.\n")

```

```

        input("Tekan Enter Untuk Melanjutkan Program")

# Fitur Read Untuk Pengguna (Melihat/Menampilkan)
def lihat_tiket(Username):
    hapus_terminal()
    print(style.CVIOLET2 + """
    _____
    |  _  ||   ||   ||   ||  ||  ||   ||  ||  ||   ||
    | |_||  ||  _ ||  _ ||  ||  |_||  ||  ||  |_||  ||
    |    ||  ||  ||  ||  ||  ||   _||  ||   ||  ||  _
    |  _ ||  ||  |_||  ||  |_||  ||   _||  ||  _  ||  ||
    | |_||  ||   ||   ||   ||   _||  ||  ||  ||  |_||
    |____||____||____||____||_||_||_||_||_||_||
    """)

    # Untuk Menggunakan Fitur PrettyTable
    table = PrettyTable()
    table.field_names = ["Judul Konser", "Lokasi Konser",
"Hari/Tanggal Konser", "Harga Tiket"]
    for tiket in akun_pengguna[Username]['tiket']:
        table.add_row([tiket['judul'], tiket['lokasi'],
tiket['tanggal'], tiket['harga']])
    print(table)
    input("Tekan Enter Untuk Melanjutkan Program")

# Fitur Read KHUSUS ADMIN (Melihat/Menampilkan)
def lihat_statistik():
    hapus_terminal()
    print(style.CBLUE2 + """
    _____
    _

```

```

|      ||      || _ ||      ||      ||      ||      ||
| |
| ____||_      _|| | | ||_      _|| || ____||_      _|| || | | |
|_ | |
| |____ | | | | | | | | | | | | | | ____ | | | | | |
|_ |
| ____ | | | | | | | | | | | | ____ | | | | | | | |
|_
| ____| | | | | | _ | | | | | | | ____| | | | | | | |
_ |
| ____| | ____| | | | | | | | | | ____| | ____| | | | | |
|_ |
        """)
    table = PrettyTable()
    table.field_names = ["Total Konser Yang Tersedia", "Total Tiket
Terboking"]
    table.add_row([total_konser,total_tiket_terboking])
    print(table)
    input("Tekan Enter Untuk Melanjutkan Program")

# Menjalankan Program
while True:
    tampilkan_menu()
    opsi = input("Pilih opsi: ")
    print(" ")

    if opsi == "1":
        registrasi_akun()
    elif opsi == "2":
        Username = login_akun()
        if Username:

```

```

while True:
    if akun_pengguna[Username]['role'] == 'admin':
        tampilkan_menu_admin(Username)
    else:
        tampilkan_menu_pengguna(Username)
    status = input("Pilih opsi: ")
    print(" ")

    # Opsi Menambah (Create)
    if status == "1":
        if akun_pengguna[Username]['role'] == 'admin':
            tambah_konser()
        else:
            booking_tiket(Username)
    # Opsi Melihat (Read)
    elif status == "2":
        if akun_pengguna[Username]['role'] == 'admin':
            lihat_konser()
        else:
            lihat_tiket(Username)
    # Opsi Editing (Update) Hanya Untuk Admin
    elif status == "3":
        if akun_pengguna[Username]['role'] == 'admin':
            edit_konser()
        else:
            print("Anda tidak memiliki akses untuk
mengedit konser.\n")
            input("Tekan Enter Untuk Melanjutkan
Program")
    # Opsi Menghapus (Delete) Hanya Untuk Admin
    elif status == "4":

```



```

        if akun_pengguna[Username]['role'] == 'admin':
            hapus_konser()
        else:
            print("Anda tidak memiliki akses untuk
menghapus konser.\n")
            input("Tekan Enter Untuk Melanjutkan
Program")

        # Opsi Statistik (Hanya Untuk Admin)
        elif status == "5":
            if akun_pengguna[Username]['role'] == 'admin':
                lihat_statistik()
            else:
                print("Anda tidak memiliki akses untuk
melihat statistik.\n")
                input("Tekan Enter Untuk Melanjutkan
Program")

        # Opsi Keluar Dri Program
        elif status == "6":
            print("Aplikasi Pembelian Tiket Konser
ditutup.\n")
            input("Tekan Enter Untuk Melanjutkan
Program")
            break
        else:
            print("Input tidak valid, silahkan pilih 1, 2, 3,
4, 5, atau 6.\n")
            input("Tekan Enter Untuk Melanjutkan Program")
    else:
        continue

    # Opsi ini untuk memberi pengguna pilihan, memberhentikan program
atau lanjut

```

```
elif opsi == "3":
    print("Apakah kamu yakin ingin keluar dari aplikasi? ")
    print("1. Iya")
    print("2. Tidak")
    pilih = input("Input pilihan: ")
    print(" ")
    if pilih == "1":
        print("Terimakasih sudah menggunakan aplikasi, semoga  
harimu menyenangkan!")
        break
    elif pilih == "2":
        continue
    else:
        print("Input tidak valid, silahkan pilih '1' atau '2'\n")
        input("Tekan Enter Untuk Melanjutkan Program")
else:
    print("Input tidak valid, silahkan pilih 1, 2, atau 3")
    input("Tekan Enter Untuk Melanjutkan Program")
```

BAB IV

PENUTUP

4.1 Kesimpulan

Program ini adalah aplikasi pemesanan tiket konser yang dirancang untuk memudahkan pengguna dalam mendaftar, login, membeli tiket, dan mengelola konser. Aplikasi ini memiliki dua jenis akun, yaitu admin dan pengguna, yang masing-masing memiliki fungsi berbeda. Admin dapat menambah, mengedit, dan menghapus konser, sementara pengguna dapat membeli tiket dan melihat tiket yang sudah dibeli.

Aplikasi ini memungkinkan pengguna baru untuk mendaftar dan pengguna yang sudah terdaftar untuk login dengan aman. Setelah login, pengguna dapat dengan mudah melihat daftar konser yang tersedia dan melakukan pembelian tiket. Semua informasi tiket yang dibeli akan tersimpan di akun pengguna.

Admin juga memiliki akses untuk melihat statistik penjualan, termasuk total konser, jumlah tiket terjual, dan pendapatan dari penjualan tiket. Antarmuka aplikasi dirancang sederhana dan intuitif, dengan penggunaan warna yang menarik dan tabel untuk menyajikan informasi secara terstruktur.

Secara keseluruhan, aplikasi ini memberikan kemudahan dalam manajemen tiket konser dan pengalaman pengguna yang baik. Pengembangan lebih lanjut dapat mencakup fitur pembayaran, notifikasi, pencarian konser, dan antarmuka responsif untuk perangkat mobile.

4.2 Saran

Untuk pengembangan lebih lanjut dari aplikasi pemesanan tiket konser, ada beberapa saran yang dapat dipertimbangkan. Pertama, integrasi sistem pembayaran yang memudahkan pengguna untuk melakukan transaksi langsung dalam aplikasi akan sangat bermanfaat. Selain itu, fitur notifikasi untuk menginformasikan pengguna tentang konser mendatang atau penawaran khusus dapat meningkatkan keterlibatan pengguna.

Menambahkan kemampuan pencarian dan filter akan membantu pengguna menemukan konser yang sesuai dengan preferensi mereka dengan lebih cepat. Fitur ulasan dan rating konser juga dapat memberikan umpan balik yang berguna bagi penyelenggara dan membantu pengguna lain dalam memilih konser.

Pengembangan antarmuka responsif agar aplikasi dapat diakses dengan baik di perangkat mobile sangat penting, mengingat banyak pengguna lebih suka menggunakan smartphone. Selain itu, menyediakan opsi pembatalan dan pengembalian dana akan memberikan rasa aman bagi pengguna yang mungkin tidak dapat menghadiri konser.

Implementasi program loyalitas untuk pengguna yang sering membeli tiket dapat meningkatkan retensi pengguna, sementara analisis data tentang perilaku pengguna dan tren penjualan tiket dapat membantu dalam pengambilan keputusan strategis. Dengan langkah-langkah ini, aplikasi dapat memberikan pengalaman yang lebih baik dan menarik bagi semua pengguna.

DAFTAR PUSTAKA

Zainuddin, M. (2021). *Pemrograman Python untuk Pemula*. Yogyakarta: Andi Offset.

Kusnadi, R. (2020). *Dasar-Dasar Pemrograman Python*. Jakarta: Elex Media Komputindo.

Wahyu, S. (2022). *Algoritma dan Struktur Data dengan Python*. Bandung: Informatika.

Sumber Online/Referensi Web:

Dokumentasi Resmi Python

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

Tutorial Python PrettyTable

<https://pypi.org/project/prettytable/>

Real Python - Python Tutorials

<https://realpython.com/>

GeeksforGeeks - Python Programming

<https://www.geeksforgeeks.org/python-programming-language/>

AI dan Chatbot Referensi:

ChatGPT (OpenAI)

<https://chat.openai.com/>

LAMPIRAN

Pembelian Tiket Konser Algoritma dan Pemrograman Dasar

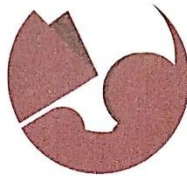
KELAS Informatika C2'24:

KELOMPOK 4:

Ikhsan / 2409106118(ketua)

Ade Pasiha Tangke Allo / 2409106109

Marsha Aulia Fakhriza / 2409106116



**LABORATORIUM FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN**

2024

CS Dipindai dengan CamScanner


Gambar 4.1 Kertas Konsul

Tanggal Konsultasi : 02, November, 2024


Uraian / Pembahasan :

1. Input harus diberi Panah
2. Untuk Decision 'pembeli' di hapus
3. 'Input masuk data' Pake Simbol Proses
4. Output diperseias



Asisten Lab


Nama: Muh. Shandy A.

Ketua Kelompok


Nama: Ikhsan

Gambar 4.2 Kertas Konsul

Tanggal Konsultasi : 13. November. 2024	
Uraian / Pembahasan : 1. Error Handling yang kosong 2. Fitur Pembayaran 3. Read di Fitur Hopos 4. Memperbaiki Atur	
Asisten Lab  Nama: Muh-Shandy A.	Ketua Kelompok  Nama: Ikhsan

CS Dipindai dengan CamScanner

Gambar 4.3 Kertas Konsul

Aturan Konsultasi :

- A. Kartu** Konsul wajib dibawa saat dilakukan konsultasi
- B. Ketua** Kelompok dan Anggota diwajibkan untuk hadir tiap konsultasi
- C. Konsul** dilaksanakan minimal 2 kali dengan ketentuan sebagai berikut :
 - = **Konsul 1** : Konsep Program
 - = **Konsul 2** : Penyelesaian Program

N.B : Batas waktu konsultasi ialah H+7 untuk konsul 1 dan H+14 untuk konsul 2 (Dimulai sejak pengumuman dan pembentuk kelompok PA)

Gambar 4.4 Kertas Konsul