

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



MANAJEMEN *ROOM* WARNET

Oleh:

Kelompok 3

M. Ilma Yusrian Fahmi	2409106079
Ega Clearesta Hananta	2409106088
Keysha Khoirunnisa Aulia K.	2409106077

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

KATA PENGANTAR

Puji syukur kami haturkan kehadiran Allah Swt. yang telah melimpahkan rahmat dan hidayah-Nya sehingga kami bisa menyelesaikan laporan proyek akhir praktikum Algoritma Pemrograman Dasar yang bertema “Manajemen *Room* Warnet”. Tujuan penyusunan makalah ini adalah untuk melengkapi rangkaian tugas proyek akhir praktikum Algoritma Pemrograman Dasar.

Program terdiri dari dua *user*, yaitu pelanggan dan admin. Jika anda sebagai admin maka anda akan mempunyai beberapa pilihan menu berikut:

1. Hapus akun, menu ini digunakan admin untuk menghapus akun pelanggan yang sudah tidak dipakai atau sudah tidak berlangganan.
2. *Booking room*, menu ini digunakan admin untuk melakukan reservasi bagi pelanggan yang belum reservasi sebelumnya.
3. *Clear room*, menu ini digunakan admin untuk memperbarui status *room* yang awalnya digunakan menjadi tersedia kembali ketika sudah selesai digunakan oleh pelanggan.
4. Manajemen *room*, menu ini digunakan admin untuk menambah *room* baru atau memperbarui (menambah/menghapus) fasilitas dalam *room*.
5. *List room*, menu ini digunakan admin untuk menampilkan *list room* beserta status *room* (digunakan/tersedia), fasilitas *room*, dan harga/jam *room*.
6. *Log out*, menu ini digunakan admin untuk keluar dari akun admin.

Jika anda sebagai pelanggan maka anda akan mempunyai beberapa pilihan menu berikut:

1. Reservasi *room*, menu ini digunakan pelanggan untuk reservasi *room* secara online melalui *device* pribadi pelanggan.

2. *List room*, menu ini digunakan pelanggan untuk menampilkan *list room* beserta status *room* (digunakan/tersedia), fasilitas *room*, dan harga/jam *room*.
3. *Log out*, menu ini digunakan pelanggan untuk keluar dari akun miliknya.

Kendala yang kami alami selama penyusunan dan pembuatan program yaitu dalam membuat *output* terlihat rapi serta tidak lupa kami ucapkan terima kasih kepada pihak-pihak yang membantu selama pembuatan program dan penyusunan laporan yaitu Bapak Awang Harsa Kridalaksana, S.Kom, M.Kom selaku dosen pengampu mata kuliah Algoritma Pemrograman Dasar, Bapak Muhammad Bambang Firdaus, S.Kom, M.Kom selaku dosen pengampu mata kuliah Logika Informatika, Bang Muhammad Afrizal Kesuma selaku asisten laboratorium dan pembimbing kami.

Sebagai penyusun, kami menyadari bahwa masih terdapat kekurangan, baik dari penyusunan maupun tata bahasa penyampaian dalam laporan ini. Oleh karena itu, kami dengan rendah hati menerima saran dan kritik dari pembaca agar kami dapat memperbaiki makalah ini.

Samarinda, 20 November 2024

Kelompok 3

TAKARIR

<i>Managemen</i>	Mengatur
<i>Input</i>	Memasukkan
<i>Output</i>	Keluaran
<i>List room</i>	Daftar ruangan
<i>Sign In</i>	Masuk ke akun
<i>Sign Up</i>	Membuat akun
<i>User</i>	Pengguna
<i>Clear room</i>	Membersihkan ruangan
<i>Sim Racing</i>	Simulator balap
<i>PC Gaming</i>	Komputer khusus game
<i>Flowchart</i>	Bagan alur
<i>Exit</i>	Keluar
<i>Booking room</i>	Memesan ruangan
<i>Log Out</i>	Keluar akun
<i>Update</i>	Memperbarui
<i>Username</i>	Nama akun
<i>Password</i>	Sandi
<i>Print</i>	Mencetak
<i>Key</i>	Kunci
<i>Value</i>	Nilai
<i>Dictionary</i>	Tipe data koleksi
<i>Create</i>	Membuat
<i>Read</i>	Membaca
<i>Delete</i>	Menghapus
<i>Immutable</i>	Kekal
<i>Return</i>	Kembali

Error handling

Penanganan kesalahan

try-Except

Menangani kesalahan

DAFTAR ISI

KATA PENGANTAR	ii
TAKARIR.....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR	viii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Kebutuhan Fungsional.....	1
1.3 Rumusan Masalah	2
1.4 Batasan Masalah.....	2
1.5 Tujuan.....	3
BAB II PERANCANGAN.....	4
2.1 Analisis Program	4
2.2 Flowchart.....	7
2.3 Konsep/Materi Praktikum yang Dipakai	12
BAB III HASIL DAN PEMBAHASAN.....	14
3.1 Tampilan Program	14
1. Tampilan Menu Log <i>In</i>	14
2.Tampilan Menu <i>User</i> Admin.....	15
3.Tampilan Menu <i>User</i> Pelanggan	20
3.2 Source Code	22
BAB IV PENUTUP	41
4.1 Kesimpulan.....	41

4.2 Saran	41
DAFTAR PUSTAKA	43
LAMPIRAN	44

DAFTAR GAMBAR

Gambar 2. 1 Flowchart menu login	7
Gambar 2. 2 Flowchart menu admin.....	8
Gambar 2. 3 Flowchart menu pelanggan	11
Gambar 3. 1 Tampilan menu awal program	14
Gambar 3. 2 Tampilan menu sign in untuk <i>user</i> admin.....	14
Gambar 3. 3 Tampilan menu sign in untuk <i>user</i> pelanggan	15
Gambar 3. 4 Tampilan menu sign up untuk <i>user</i> pelanggan.....	15
Gambar 3. 5 Tampilan menu pilihan untuk <i>user</i> admin	15
Gambar 3. 6 Tampilan <i>user</i> admin dalam pilihan hapus akun.....	16
Gambar 3. 7 Tampilan <i>user</i> admin dalam pilihan booking room	16
Gambar 3. 8 Tampilan <i>user</i> admin dalam pilihan hapus room	17
Gambar 3. 9 Tampilan <i>user</i> dalam pilihan manajemen room	17
Gambar 3. 10 Tampilan <i>user</i> admin dalam pilihan manajemen room dan tambah room baru	17
Gambar 3. 11 Tampilan <i>user</i> admin dalam pilihan manajemen room dan hapus room	18
Gambar 3. 12 Tampilan <i>user</i> admin dalam pilihan manajemen room dan update fasilitas room.....	18
Gambar 3. 13 Tampilan <i>user</i> admin dalam pilihan manajemen room, update fasilitas room, dan tambah fasilitas	19
Gambar 3. 14 Tampilan <i>user</i> admin dalam pilihan manajemen room, update fasilitas room, dan hapus fasilitas.....	19
Gambar 3. 15 Tampilan <i>user</i> admin dalam pilihan info room.....	19
Gambar 3. 16 Tampilan <i>user</i> admin dalam pilihan list room	20
Gambar 3. 17 Tampilan menu pilihan untuk <i>user</i> pelanggan	20
Gambar 3. 18 Tampilan <i>user</i> pelanggan dalam pilihan reservasi room.....	21

Gambar 3. 19 Tampilan <i>user</i> pelanggan dalam pilihan list room	22
Gambar 3. 20 Modul yang digunakan.....	22
Gambar 3. 21 Fungsi membaca file akun.json.....	23
Gambar 3. 22 Fungsi untuk menulis file akun.json	23
Gambar 3. 23 Fungsi untuk membaca file vvip_room.json	23
Gambar 3. 24 Fungsi untuk menulis file vvip_room.json.....	24
Gambar 3. 25 Fungsi untuk menampilkan menu admin	24
Gambar 3. 26 Fungsi untuk menampilkan menu <i>user</i>	25
Gambar 3. 27 Fungsi untuk melakukan log in	25
Gambar 3. 28 Fungsi untuk membuat akun	27
Gambar 3. 29 Fungsi untuk memesan/booking room yang tersedia.....	29
Gambar 3. 30 Fungsi untuk menghapus room yang ada.....	30
Gambar 3. 31 Fungsi untuk menampilkan list akun yang telah dibuat.....	31
Gambar 3. 32 Fungsi untuk informasi semua room.....	31
Gambar 3. 33 Fungsi untuk menampilkan informasi room yang digunakan.....	32
Gambar 3. 34 Fungsi untuk menghapus terminal	32
Gambar 3. 35 Fungsi untuk menampilkan main menu	33
Gambar 3. 36 Fungsi untuk menghapus room yang digunakan.....	33
Gambar 3. 37 Fungsi untuk memperbaiki (menambah/menghapus) fasilitas room ...	36
Gambar 3. 38 Fungsi untuk menambah room baru kedalam list	37
Gambar 3. 39 Fungsi untuk menampilkan menu manajemen room	38
Gambar 3. 40 Main Program.....	40
Lampiran 1.....	44
Lampiran 2	45

BAB I

PENDAHULUAN

1.1 Latar Belakang

Warnet modern tidak hanya menyediakan akses internet, tetapi juga berbagai fasilitas seperti *Sim Racing*, *PC gaming*, dan konsol *PS5*. Dengan berbagai fasilitas yang tersedia, manajemen ruangan menjadi salah satu tantangan utama bagi pemilik warnet. Kesulitan dalam mengelola pemesanan, penggunaan ruangan, serta penjadwalan pengguna, dapat mengurangi efisiensi operasional dan kualitas layanan yang diberikan kepada pelanggan. Oleh karena itu, diperlukan sebuah sistem manajemen ruangan yang efektif untuk memastikan bahwa semua fasilitas dapat dimanfaatkan secara optimal dan terorganisir.

1.2 Kebutuhan Fungsional

Pada awal program terdapat tiga pilihan menu, yaitu:

1. *Sign in*, fungsi ini digunakan untuk login dan masuk ke dalam program, baik sebagai admin maupun pelanggan.
2. *Sign up*, fungsi ini digunakan untuk membuat akun bagi pelanggan yang belum memiliki akun.
3. *Exit*, fungsi ini digunakan untuk keluar dari program.

Jika login sebagai admin maka akan menampilkan menu berikut:

1. Hapus akun, fungsi ini digunakan admin untuk menghapus akun pelanggan yang sudah tidak dipakai atau sudah tidak berlangganan.
2. *Booking room*, fungsi ini digunakan admin untuk melakukan reservasi bagi pelanggan yang belum reservasi sebelumnya.
3. *Clear room*, fungsi ini digunakan admin untuk memperbarui status *room* yang awalnya digunakan menjadi tersedia kembali ketika sudah selesai digunakan oleh pelanggan.
4. Manajemen *room*, fungsi ini digunakan admin untuk menambah *room* baru atau memperbarui (menambah/menghapus) fasilitas dalam *room*.

5. *List room*, fungsi ini digunakan admin untuk menampilkan *list room* beserta status *room* (digunakan/tersedia), fasilitas *room*, dan harga/jam *room*.
6. *Log out*, fungsi ini digunakan admin untuk keluar dari akun admin.

Jika login sebagai pelanggan maka akan menampilkan menu berikut:

1. Reservasi *room*, menu ini digunakan pelanggan untuk reservasi *room* secara online melalui device pribadi pelanggan.
2. *List room*, menu ini digunakan pelanggan untuk menampilkan *list room* beserta status *room* (digunakan/tersedia), fasilitas *room*, dan harga/jam *room*.
3. *Log out*, menu ini digunakan pelanggan untuk keluar dari akun miliknya.

1.3 Rumusan Masalah

Bagaimana mengelola pemesanan ruangan dan fasilitas seperti *Sim Racing*, *PC gaming*, dan konsol *PS5* secara efisien untuk menghindari konflik jadwal dan memaksimalkan penggunaan ?

Pelanggan dan staf mungkin lupa mengenai jadwal pemesanan yang telah dibuat, yang bisa menyebabkan ketidaknyamanan. Sistem dapat mengirim pengingat otomatis dan notifikasi kepada pelanggan dan staf mengenai pemesanan yang akan datang. Ini membantu memastikan semua pihak selalu terinformasi dan mempersiapkan diri sesuai jadwal.

Ketika fasilitas tidak dikelola dengan baik, ada kemungkinan beberapa fasilitas tidak digunakan secara optimal. Sistem harus memiliki fitur pemantauan ketersediaan yang memberikan informasi terkini tentang penggunaan ruangan dan peralatan.

1.4 Batasan Masalah

Batasan masalahnya adalah program ini hanya bisa digunakan untuk melakukan reservasi *room*, melihat *list room*, memperbarui status *room* dan belum bisa menambahkan jam pada satu ruangan yang telah dilakukan reservasi.

1.5 Tujuan

Tujuan kami adalah menciptakan program yang dapat mempermudah dalam proses pengelolaan pemesanan ruangan dan fasilitas seperti *Sim Racing*, *PC gaming*, dan konsol *PS5* secara efisien untuk menghindari konflik jadwal dan memaksimalkan penggunaan.

BAB II

PERANCANGAN

2.1 Analisis Program

Pada awal program *user* akan di beri 3 pilihan, yaitu *sign in*, *sign up*, dan *exit*. Jika sebagai pelanggan maka program akan meminta untuk langsung *sign in*, namun jika belum memiliki akun akan program akan meminta pelanggan untuk *sign up* terlebih dahulu. Kemudian setelah *sign in*, pelanggan akan di beri 3 pilihan kembali, yaitu:

1. Reservasi *room*

Jika pelanggan memilih menu reservasi *room*, maka program akan menampilkan *list room* yang berisi *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*. Kemudian program akan meminta pelanggan untuk menginput nama pemesan, *ID room* yang akan di reservasi, dan lamanya waktu reservasi. Setelahnya, program akan mengkalkulasi total harga yang harus dibayar oleh pelanggan sesuai dengan harga/jam dari *room* yang dipilih dan menampilkan nota reservasi yang berisi nama pemesan, *ID room*, waktu pemesanan, dan total harga yang harus di bayar.

2. *List room*

Jika pelanggan memilih menu *list room*, maka program akan menampilkan *list room* yang berisi *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*.

3. *Log out*

Jika pelanggan memilih menu *log out*, maka program akan *log out* dari akun yang sebelumnya dipakai.

Jika sebagai admin maka program akan meminta untuk langsung *sign in* sebagai admin. Kemudian setelah *sign in*, admin akan diberikan 7 pilihan, yaitu:

1. Hapus akun

Jika admin memilih menu ini, maka program akan menampilkan daftar akun yang sudah ada didalam data warnet, kemudian program akan meminta admin untuk menginput nomor akun yang ingin dihapus.

2. *Booking room*

Jika admin memilih menu ini, maka program akan menampilkan *list room* yang berisi *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*. Kemudian program akan meminta pelanggan untuk menginput nama pemesan, *ID room* yang akan di reservasi, dan lamanya waktu reservasi. Setelahnya, program akan mengkalkulasi total harga yang harus dibayar oleh pelanggan sesuai dengan harga/jam dari *room* yang dipilih.

3. *Clear room*

Jika admin memilih menu ini, maka program akan menampilkan *list room* yang sedang digunakan, kemudian program akan meminta admin untuk menginput nomor *room* yang akan diganti statusnya.

4. Manajemen *room*

Jika admin memilih menu ini, maka program akan kembali memberikan pilihan menu, yaitu:

1. Tambah *room* baru

Jika admin memilih menu ini, maka program akan menampilkan tipe-tipe *room* dan program akan meminta admin untuk menginput tipe *room* apa yang ingin ditambahkan.

2. Hapus *room*

Jika admin memilih menu ini, maka program akan menampilkan *list room* yang ada didalam program dan program akan meminta admin untuk menginput nomor *room* yang ingin dihapus.

3. *Update fasilitas room*

Jika admin memilih menu ini, maka program akan menampilkan *list room* yang berisi fasilitas *room*. Kemudian program akan meminta admin untuk menginput nomor *room* yang ingin diganti, setelah menginput nomor *room*, program akan menampilkan menu kembali, yaitu:

1. Tambah fasilitas

Jika admin memilih menu ini, maka program akan meminta admin untuk menginput nomor fasilitas yang ingin ditambahkan.

2. Hapus fasilitas

Jika admin memilih menu ini, maka program akan meminta admin untuk menginput nomor fasilitas yang ingin dihapus.

5. *Info room*

Jika admin memilih menu ini, maka program akan menampilkan *room* apa saja yang sedang digunakan, siapa yang reservasi, dan lamanya waktu reservasi.

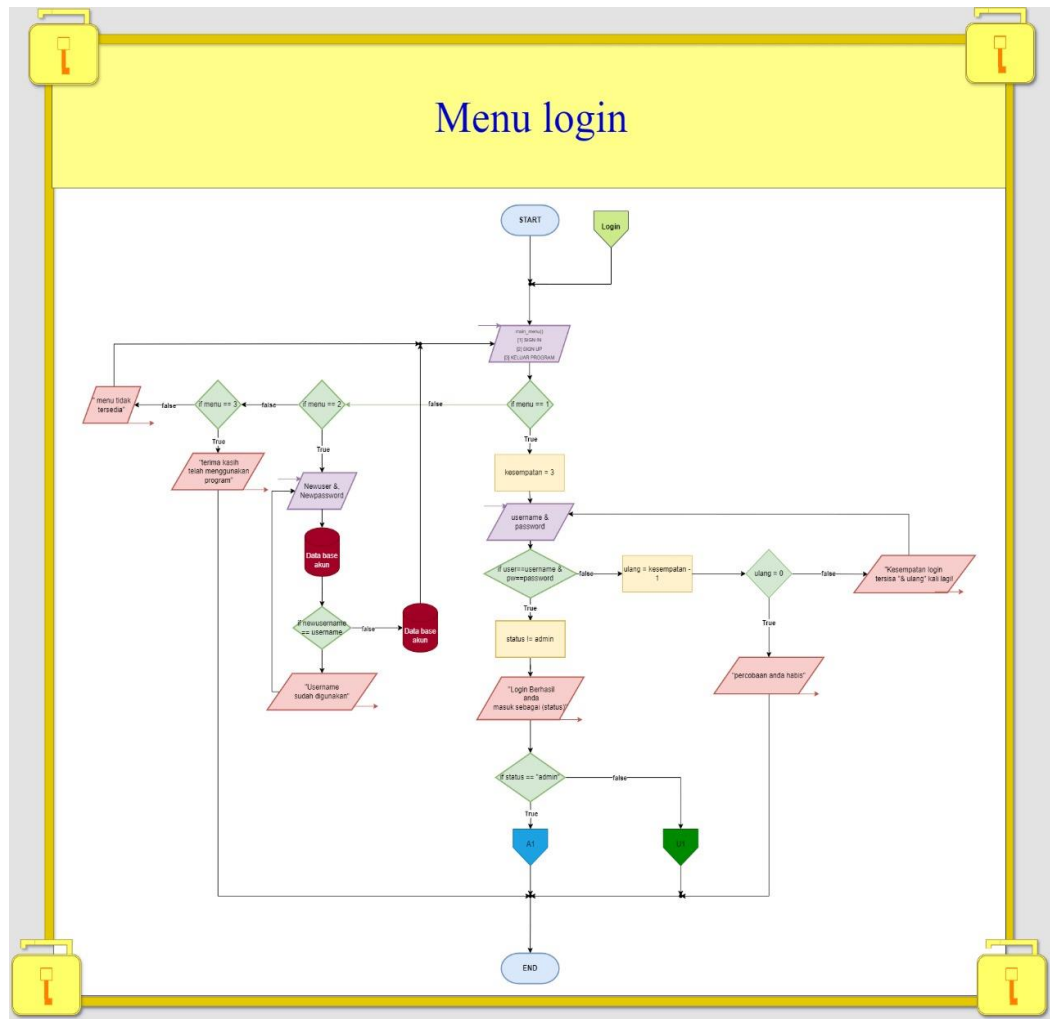
6. *List room*

Jika admin memilih menu ini, maka program akan menampilkan *list room* warnet yang memuat informasi berupa, *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*.

7. *Log out*

Jika admin memilih menu ini, maka program akan *log out* dari akun sebagai admin.

2.2 Flowchart

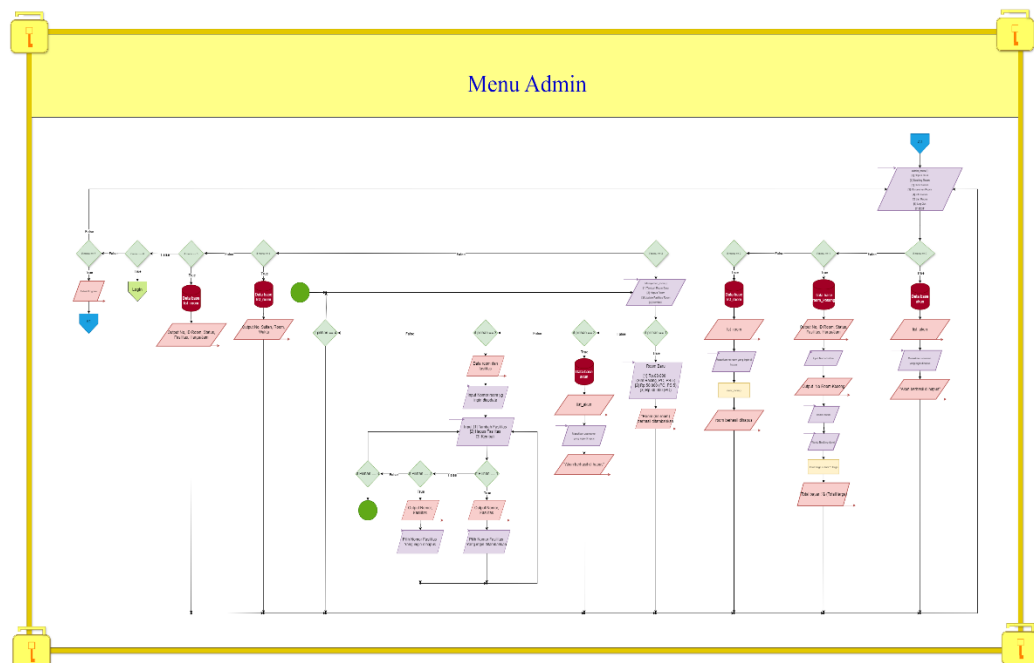


Gambar 2. 1 Flowchart menu login

Pada awal program akan masuk kepada menu login, program akan menampilkan pilihan *sign in*, *sign up*, dan keluar program. Jika *user* sebagai admin maka *user* bisa langsung memilih menu *sign in*, program akan meminta *user* untuk menginput *username* dan *password*, jika kombinasi *username* dan *password* salah maka kesempatan *user* akan berkurang hingga kesempatannya habis dan program akan berhenti. Jika kombinasi *username* dan *password* benar,

maka program akan mengecek status *user*, dan program akan masuk kedalam program admin.

Jika *user* sebagai pelanggan dan belum memiliki akun, *user* akan diminta untuk melakukan registrasi terlebih dahulu di menu *sign up*, setelah *user* melakukan register, maka program akan meminta *user* untuk kembali melakukan *sign in* dengan akun yang sudah dibuat. Jika kombinasi *username* dan password salah maka kesempatan *user* akan berkurang hingga kesempatannya habis dan program akan berhenti. Jika kombinasi *username* dan *password* benar, maka program akan mengecek status *user*, dan program akan masuk kedalam program pelanggan.



Gambar 2. 2 Flowchart menu admin

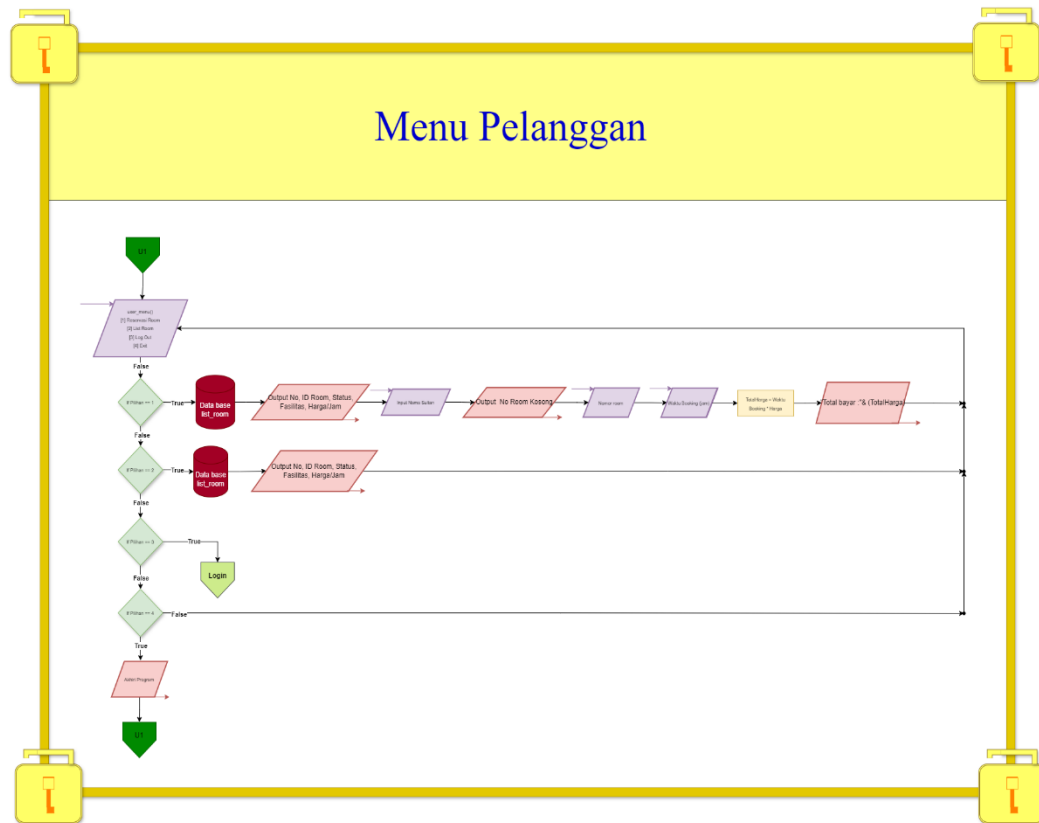
Pada program admin akan ditampilkan beberapa menu, yaitu:

1. Hapus akun, jika admin memilih menu ini maka program akan menampilkan seluruh list akun yang sudah ada dan program akan meminta admin untuk

memasukkan nomor *username* yang ingin dihapus, kemudian program akan memberikan *output* “Akun berhasil dihapus”.

2. *Booking room*, jika admin memilih menu ini maka program akan menampilkan seluruh *list room* yang berisi informasi berupa nomor *room*, *ID room*, status *room*, fasilitas *room*, harga/jam dari *room*. Kemudian program akan meminta admin untuk menginput nama pemesan, setelahnya program akan menampilkan *list room* kosong. Setelah itu, program akan meminta admin untuk menginput nomor *room* yang tersedia, lamanya waktu *booking*, dan program akan mengkalkulasi harga yang harus dibayar berdasarkan *room* yang dipilih dan lamanya waktu *booking*. Setelah itu program akan menampilkan total harga yang harus dibayar.
3. *Clear room*, jika admin memilih menu ini maka program akan menampilkan *list room* yang sedang digunakan beserta nama pemesannya. Kemudian program akan meminta admin untuk menginput nomor *room* yang ingin dihapus, setelahnya program akan memanggil fungsi *def clear_room()* dan memberikan *output* “*room* berhasil dihapus”
4. Manajemen *room*, jika admin memilih menu ini maka program akan menampilkan beberapa menu pilihan, yaitu:
 1. Tambah *room* baru, jika admin memilih menu ini maka program akan menampilkan list tipe-tipe *room* yang tersedia dan akan meminta admin untuk menginput nomor tipe *room* yang ingin ditambah.
 2. Hapus *room*, jika admin memilih menu ini maka program akan menampilkan *list room* yang ada dan program akan meminta admin untuk menginput nomor *room* yang ingin dihapus.
 3. *Update fasilitas room*, jika admin memilih menu ini maka program akan menampilkan *list room* beserta informasi berupa fasilitas *room* dan program akan meminta admin untuk menginput nomor *room* yang ingin diupdate, kemudian program akan menampilkan beberapa menu, yaitu:

1. Tambah fasilitas, jika admin memilih menu ini maka program akan menampilkan fasilitas *room* yang tersedia setelah itu meminta admin untuk menginput nomor fasilitas yang ingin ditambahkan.
2. Hapus fasilitas, jika admin memilih menu ini maka program akan menampilkan fasilitas *room* yang tersedia setelah itu meminta admin untuk menginput nomor fasilitas yang ingin dihapus.
3. Kembali, jika admin memilih menu ini maka program akan kembali pada menu manajemen *room*.
5. Info *room*, jika admin memilih menu ini maka program akan menampilkan *list room* yang telah di *booking*.
6. *List room*, jika admin memilih menu ini maka program akan menampilkan seluruh *list room* beserta informasi *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*.
7. *Log out*, jika admin memilih menu ini maka program akan kembali ke menu *log in*.
8. *Exit*, jika admin memilih menu ini maka program akan mengeluarkan *output* “akhir program” dan program berhenti.



Gambar 2. 3 Flowchart menu pelanggan

Pada program pelanggan akan ditampilkan beberapa menu, yaitu:

1. Reservasi *room*, jika pelanggan memilih menu reservasi *room*, maka program akan menampilkan *list room* yang berisi *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*. Kemudian program akan meminta pelanggan untuk menginput nama pemesan, *ID room* yang akan di reservasi, dan lamanya waktu reservasi. Setelahnya, program akan mengkalkulasi total harga yang harus dibayar oleh pelanggan sesuai dengan harga/jam dari *room* yang dipilih dan menampilkan nota reservasi yang berisi nama pemesan, *ID room*, waktu pemesanan, dan total harga yang harus di bayar.
2. *List room*, jika pelanggan memilih menu *list room*, maka program akan menampilkan *list room* yang berisi informasi berupa *ID room*, status *room*, fasilitas *room*, dan harga/jam dari *room*.

3. *Log out*, jika pelanggan memilih menu *log out* maka program akan kembali ke menu *log in*.

2.3 Konsep/Materi Praktikum yang Dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan

persyaratan yang terdapat pada modul diantaranya:

1. Fungsi dasar: fungsi yang kami gunakan adalah fungsi *print*.
Fungsi *print* pada program merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga.
2. Fungsi list & tuple: fungsi yang kami gunakan adalah fungsi list dan tuple.
Fungsi tipe data list pada program merupakan fungsi tipe data yang menggunakan *key* dan *value* untuk memanggilnya. Fungsi tipe data tuple merupakan struktur data yang digunakan untuk menyimpan sekumpulan data. Tuple bersifat *immutable*, artinya isi tuple tidak bisa kita ubah dan hapus. Namun, dapat kita isi dengan berbagai macam nilai objek.
3. Fungsi dictionary: fungsi yang kami gunakan adalah fungsi *dictionary*.
Dictionary adalah suatu tipe data pada python yang berfungsi untuk menyimpan kumpulan data/nilai. Pendeklarasian pada *dictionary* akan menggunakan tanda kurung kurawal { } lalu isi dari *dictionary* tersebut akan berada didalamnya.
4. Fungsi perulangan: fungsi yang kami gunakan adalah fungsi *while True* dan *for*.
Fungsi tersebut merupakan instruksi yang digunakan untuk memerintahkan komputer melakukan sesuatu secara berulang-ulang dengan jumlah yang ditentukan atau selama masih memenuhi suatu kondisi yang telah ditentukan.
5. Fungsi penambahan: fungsi yang kami gunakan adalah fungsi *.append*.
Fungsi tersebut merupakan fungsi yang digunakan untuk menambahkan data kedalam list.
6. Fungsi *def*: fungsi yang kami gunakan adalah fungsi *def*.

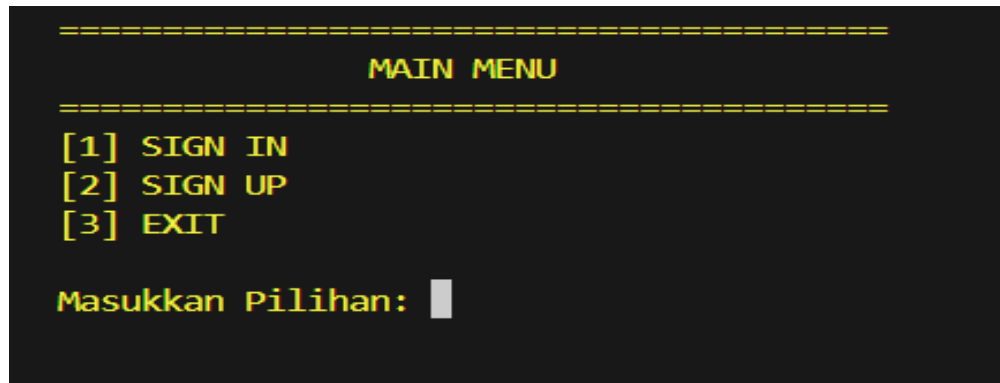
Fungsi ini digunakan untuk memecah program besar menjadi sub program yang lebih sederhana.

7. Fungsi pengembalian nilai: fungsi yang kami gunakan adalah fungsi *return*.
Fungsi tersebut merupakan fungsi yang digunakan mengembalikan nilai dari hasil pemrosesannya.
8. Fungsi percabangan: fungsi yang kami gunakan adalah fungsi *if*, *elif*, dan *else*.
Fungsi ini digunakan untuk menentukan tindakan dan perintah sesuai kondisi yang digunakan.
9. Fungsi CRUD: fungsi yang kami gunakan adalah fungsi CRUD (*Create*, *Read*, *Update*, *Delete*).
Fungsi ini digunakan untuk mempermudah pengelolaan data dalam suatu sistem dimana kita bisa melihat, ,menambah, mengedit, dan menghapus data yang ada.
10. Fungsi *error handling* & file eksternal: fungsi yang kami gunakan dalam error handling adalah *try-Except* dan untuk file eksternalnya adalah *json*.

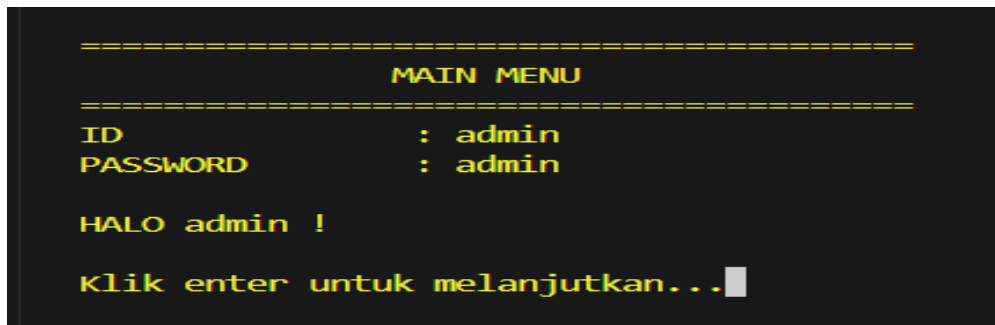
BAB III HASIL DAN PEMBAHASAN

3.1 Tampilan Program

1. Tampilan Menu Log In



Gambar 3. 1 Tampilan menu awal program



Gambar 3. 2 Tampilan menu sign in untuk user admin

```
=====
                        MAIN MENU
=====
ID                      : kysh
PASSWORD                : skskkysh

HALO kysh !

Klik enter untuk melanjutkan...█
```

Gambar 3. 3 Tampilan menu sign in untuk user pelanggan

```
=====
                        BIKIN AKUN DULU GAN!
=====
ID BARU                 : pahmi
PASSWORD Baru          : ega
AKUN BERHASIL DIBUAT SILAHKAN LOGIN

Klik enter untuk melanjutkan...█
```

Gambar 3. 4 Tampilan menu sign up untuk user pelanggan

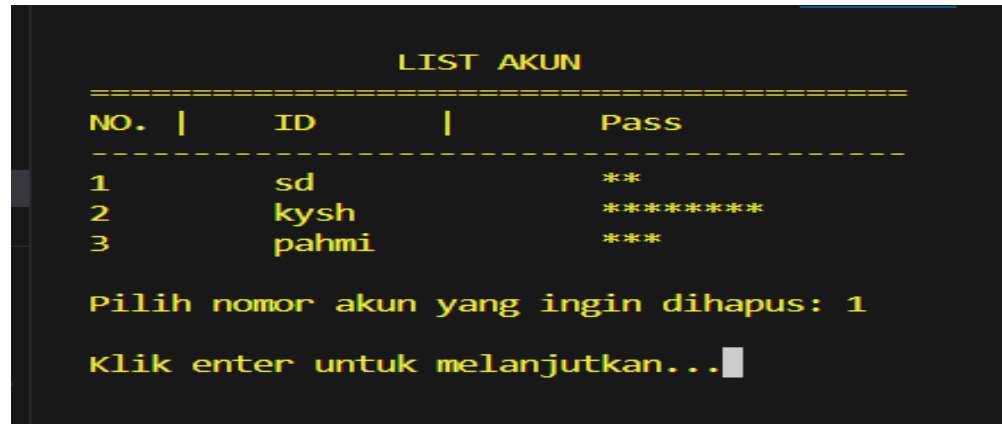
2.Tampilan Menu User Admin

```
=====
                        SULTAN GAMING CENTRE
=====

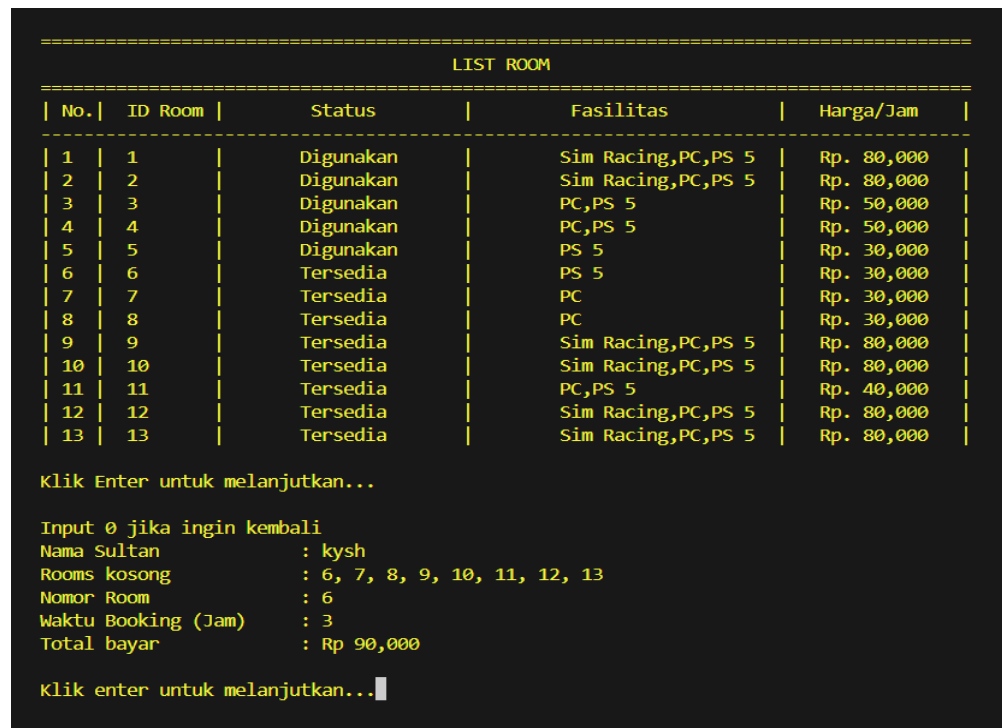
[0] Hapus Akun
[1] Booking Room
[2] Hapus Room
[3] Manajemen Room
[4] Info Room
[5] List Room
[6] Log Out
[7] Exit

Masukkan pilihan: █
```

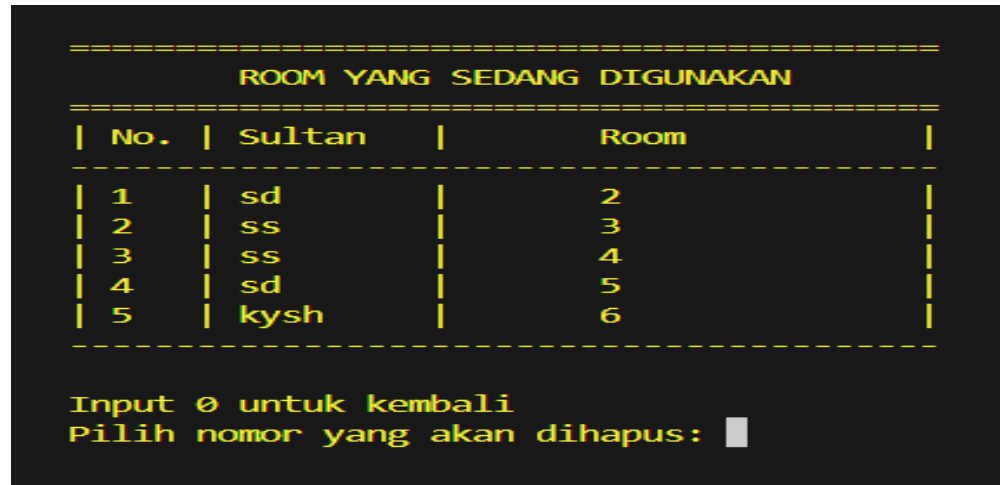
Gambar 3. 5 Tampilan menu pilihan untuk user admin



Gambar 3. 6 Tampilan user admin dalam pilihan hapus akun



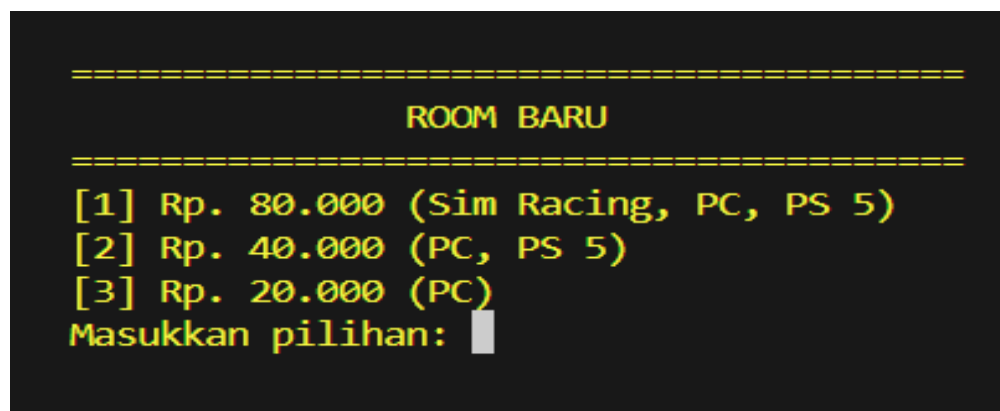
Gambar 3. 7 Tampilan user admin dalam pilihan booking room



Gambar 3. 8 Tampilan user admin dalam pilihan hapus room



Gambar 3. 9 Tampilan user dalam pilihan manajemen room



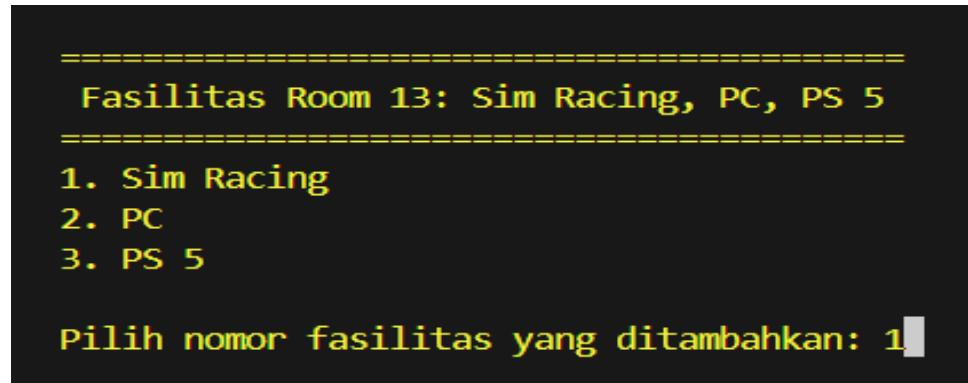
Gambar 3. 10 Tampilan user admin dalam pilihan manajemen room dan tambah room baru



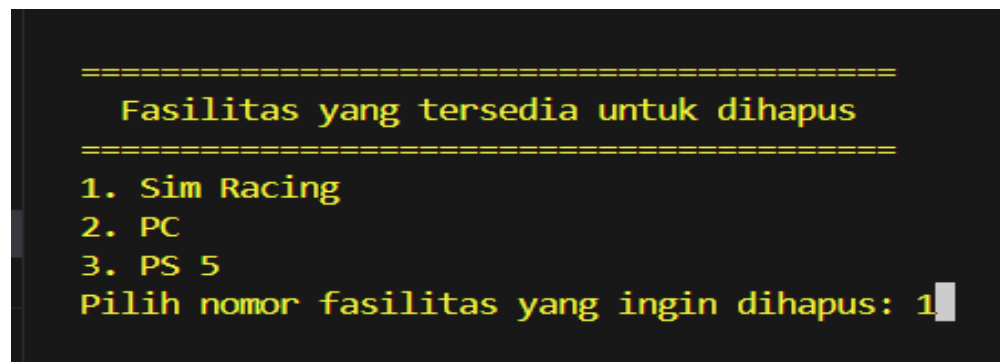
Gambar 3. 11 Tampilan user admin dalam pilihan manajemen room dan hapus room



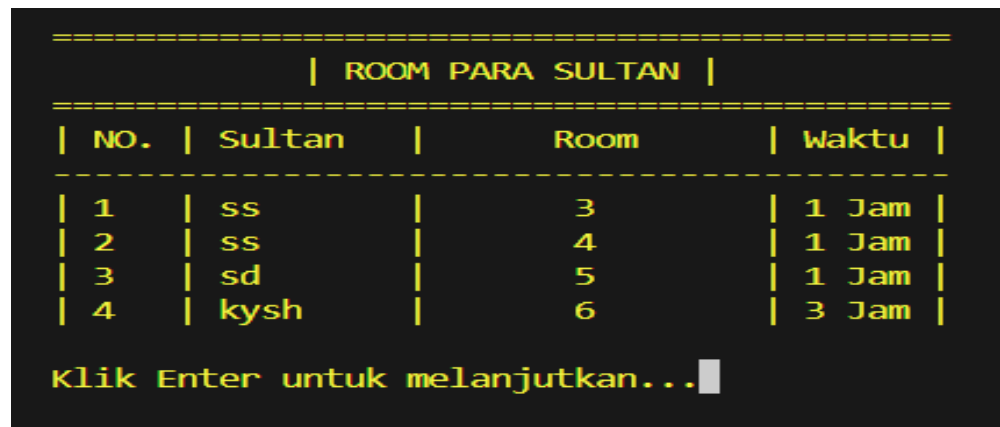
Gambar 3. 12 Tampilan user admin dalam pilihan manajemen room dan update fasilitas room



Gambar 3. 13 Tampilan user admin dalam pilihan manajemen room, update fasilitas room, dan tambah fasilitas



Gambar 3. 14 Tampilan user admin dalam pilihan manajemen room, update fasilitas room, dan hapus fasilitas



Gambar 3. 15 Tampilan user admin dalam pilihan info room

LIST ROOM				
No.	ID Room	Status	Fasilitas	Harga/Jam
1	1	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000
2	2	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000
3	3	Digunakan	PC,PS 5	Rp. 50,000
4	4	Digunakan	PC,PS 5	Rp. 50,000
5	5	Digunakan	PS 5	Rp. 30,000
6	6	Digunakan	PS 5	Rp. 30,000
7	7	Tersedia	PC	Rp. 30,000
8	8	Tersedia	PC	Rp. 30,000
9	9	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000
10	10	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000
11	11	Tersedia	PC,PS 5	Rp. 40,000
12	12	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000
13	13	Tersedia	PC,PS 5	Rp. 40,000

Klik Enter untuk melanjutkan...

Gambar 3. 16 Tampilan user admin dalam pilihan list room

3.Tampilan Menu *User Pelanggan*

```

=====
                        SULTAN GAMING CENTRE
=====

[1] Reservasi Room
[2] List Room
[3] Log Out
[4] Exit

Masukkan pilihan: 1

```

Gambar 3. 17 Tampilan menu pilihan untuk user pelanggan

LIST ROOM					
No.	ID Room	Status	Fasilitas	Harga/Jam	
1	1	Digunakan	Sim Racing,PC,PS 5	Rp. 80,000	
2	2	Digunakan	Sim Racing,PC,PS 5	Rp. 80,000	
3	3	Digunakan	PC,PS 5	Rp. 50,000	
4	4	Digunakan	PC,PS 5	Rp. 50,000	
5	5	Digunakan	PS 5	Rp. 30,000	
6	6	Digunakan	PS 5	Rp. 30,000	
7	7	Tersedia	PC	Rp. 30,000	
8	8	Tersedia	PC	Rp. 30,000	
9	9	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000	
10	10	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000	
11	11	Tersedia	PC,PS 5	Rp. 40,000	
12	12	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000	
13	13	Tersedia	PC,PS 5	Rp. 40,000	

Klik Enter untuk melanjutkan...

Input 0 jika ingin kembali

Nama Sultan : kysh

Rooms kosong : 7, 8, 9, 10, 11, 12, 13

Nomor Room : 7

Waktu Booking (Jam) : 2

NOTA RESERVASI	
NAMA	: kysh
ROOM	: 7
WAKTU (Jam)	: 2
TOTAL BAYAR	: Rp 60,000
Silahkan hubungi @Admin untuk konfirmasi	
Pembayaran telah dikonfirmasi, Terimakasih	
Klik enter untuk melanjutkan...	

Gambar 3. 18 Tampilan user pelanggan dalam pilihan reservasi room

LIST ROOM					
No.	ID Room	Status	Fasilitas	Harga/Jam	
1	1	Digunakan	Sim Racing,PC,PS 5	Rp. 80,000	
2	2	Digunakan	Sim Racing,PC,PS 5	Rp. 80,000	
3	3	Digunakan	PC,PS 5	Rp. 50,000	
4	4	Digunakan	PC,PS 5	Rp. 50,000	
5	5	Digunakan	PS 5	Rp. 30,000	
6	6	Digunakan	PS 5	Rp. 30,000	
7	7	Digunakan	PC	Rp. 30,000	
8	8	Tersedia	PC	Rp. 30,000	
9	9	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000	
10	10	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000	
11	11	Tersedia	PC,PS 5	Rp. 40,000	
12	12	Tersedia	Sim Racing,PC,PS 5	Rp. 80,000	
13	13	Tersedia	PC,PS 5	Rp. 40,000	

Klik Enter untuk melanjutkan...

Gambar 3. 19 Tampilan user pelanggan dalam pilihan list room


3.2 Source Code

```

1 import os # Menyediakan macam-macam fungsi untuk interaksi dengan sistem operasi
2 import time # Digunakan untuk jeda time sleep
3 import json # ddGunakan untuk bekerja dengan data dalam format JSON (JavaScript Object Notation)
4 from colorama import Fore
5 # Modul colorama digunakan untuk memberikan warna pada teks yang ditampilkan di terminal
6 # Fore adalah salah satu bagian dari colorama digunakan untuk mengubah warna teks (foreground) yang ditampilkan di terminal
7 print(Fore.LIGHTYELLOW_EX)


```

Gambar 3. 20 Modul yang digunakan




```
1 # Fungsi untuk membaca file akun.json
2 def muat_akun():
3     """encoding utf-8"""
4     with open("akun.json", "r", encoding="utf-8") as file:
5         return json.load(file)
```

Gambar 3. 21 Fungsi membaca file akun.json



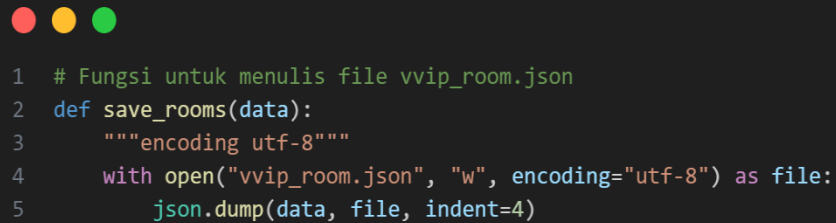
```
1 # Fungsi untuk menulis file akun.json
2 def save_akun(data):
3     """encoding utf-8"""
4     with open("akun.json", "w", encoding="utf-8") as file:
5         json.dump(data, file, indent=4)
```

Gambar 3. 22 Fungsi untuk menulis file akun.json



```
1 # Fungsi untuk membaca file vvip_room.json
2 def muat_rooms():
3     """encoding utf-8"""
4     with open("vvip_room.json", "r", encoding="utf-8") as file:
5         return json.load(file)
```

Gambar 3. 23 Fungsi untuk membaca file vvip_room.json

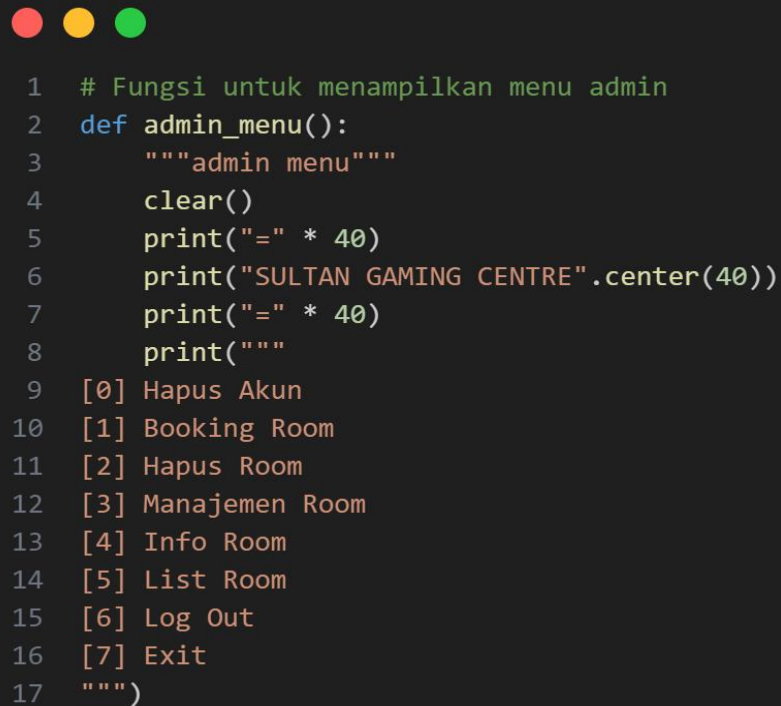


```

1  # Fungsi untuk menulis file vvip_room.json
2  def save_rooms(data):
3      """encoding utf-8"""
4      with open("vvip_room.json", "w", encoding="utf-8") as file:
5          json.dump(data, file, indent=4)

```

Gambar 3. 24 Fungsi untuk menulis file vvip_room.json



```

1  # Fungsi untuk menampilkan menu admin
2  def admin_menu():
3      """admin menu"""
4      clear()
5      print("=" * 40)
6      print("SULTAN GAMING CENTRE".center(40))
7      print("=" * 40)
8      print("""
9  [0] Hapus Akun
10 [1] Booking Room
11 [2] Hapus Room
12 [3] Manajemen Room
13 [4] Info Room
14 [5] List Room
15 [6] Log Out
16 [7] Exit
17 """)

```

Gambar 3. 25 Fungsi untuk menampilkan menu admin

```

1  # Fungsi untuk menampilkan menu user
2  def user_menu():
3      """user menu"""
4      clear()
5      print("=" * 40)
6      print("SULTAN GAMING CENTRE".center(40))
7      print("=" * 40)
8      print("""
9      [1] Reservasi Room
10     [2] List Room
11     [3] Log Out
12     [4] Exit
13     """)

```


Gambar 3. 26 Fungsi untuk menampilkan menu user

```

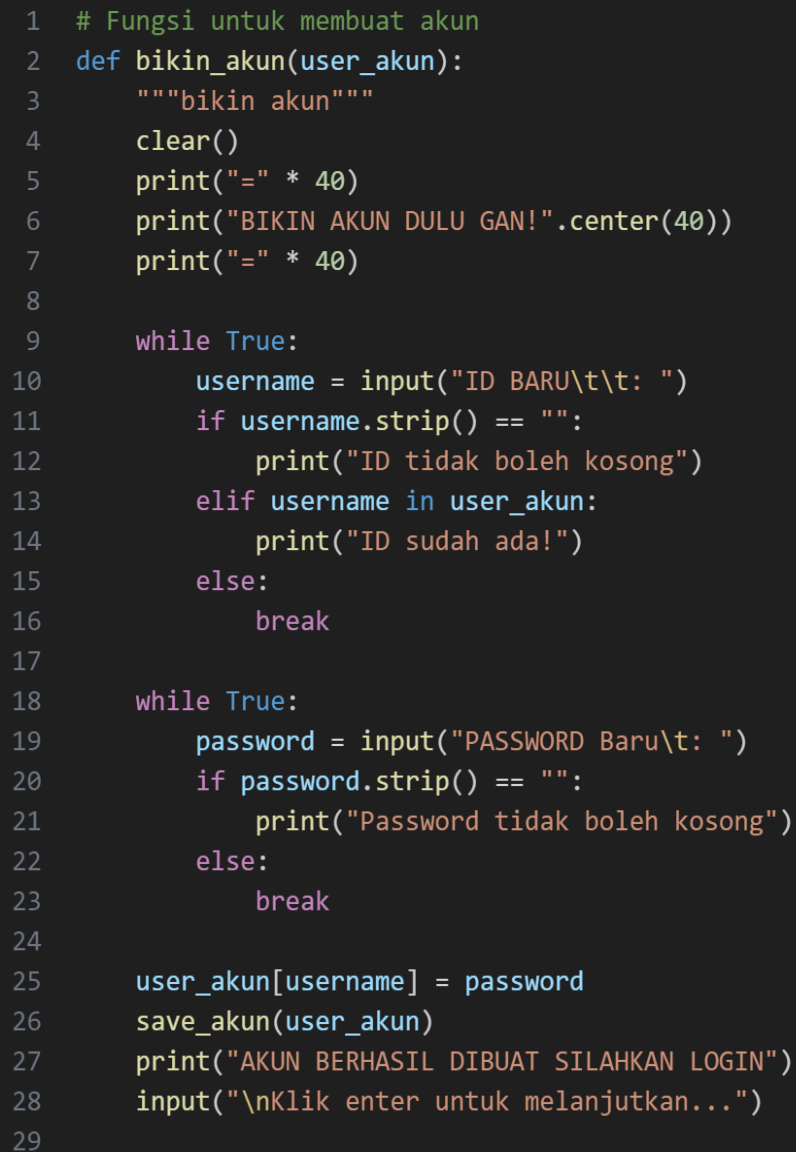
1  # Fungsi untuk melakukan login
2  def login(user_akun):
3      """login"""
4      clear()
5      batas_login = 3
6      while batas_login > 0:
7          clear()
8          print("=" * 40)
9          print("MAIN MENU".center(40))
10         print("=" * 40)
11         username = input("ID\t\t: ")
12         password = input("PASSWORD\t: ")
13         if username in user_akun and user_akun[username] == password:
14             print(f"\nHALO {username} !")
15             input("\nKlik enter untuk melanjutkan...")
16             return True, username
17         batas_login -= 1
18         print("\nID atau PASSWORD salah!")
19         print(f"\nKesempatan login tersisa {batas_login} kali lagi!")
20         input("\nKlik enter untuk melanjutkan...")
21         clear()
22     print("\nANDA GAGAL LOGIN 3 KALI. HUBUNGI ADMIN")
23     exit()
24

```

Gambar 3. 27 Fungsi untuk melakukan log in



```
1  # Fungsi untuk membuat akun
2  def bikin_akun(user_akun):
3      """bikin akun"""
4      clear()
5      print("=" * 40)
6      print("BIKIN AKUN DULU GAN!".center(40))
7      print("=" * 40)
8
9      while True:
10         username = input("ID BARU\t\t: ")
11         if username.strip() == "":
12             print("ID tidak boleh kosong")
13         elif username in user_akun:
14             print("ID sudah ada!")
15         else:
16             break
17
18     while True:
19         password = input("PASSWORD Baru\t: ")
20         if password.strip() == "":
21             print("Password tidak boleh kosong")
22         else:
23             break
24
25     user_akun[username] = password
26     save_akun(user_akun)
27     print("AKUN BERHASIL DIBUAT SILAHKAN LOGIN")
28     input("\nKlik enter untuk melanjutkan...")
29
```



```

1  # Fungsi untuk membuat akun
2  def bikin_akun(user_akun):
3      """bikin akun"""
4      clear()
5      print("=" * 40)
6      print("BIKIN AKUN DULU GAN!".center(40))
7      print("=" * 40)
8
9      while True:
10         username = input("ID BARU\t\t: ")
11         if username.strip() == "":
12             print("ID tidak boleh kosong")
13         elif username in user_akun:
14             print("ID sudah ada!")
15         else:
16             break
17
18         while True:
19             password = input("PASSWORD Baru\t: ")
20             if password.strip() == "":
21                 print("Password tidak boleh kosong")
22             else:
23                 break
24
25         user_akun[username] = password
26         save_akun(user_akun)
27         print("AKUN BERHASIL DIBUAT SILAHKAN LOGIN")
28         input("\nKlik enter untuk melanjutkan...")
29

```

Gambar 3. 28 Fungsi untuk membuat akun

```

1  # Fungsi untuk memesan/booking room yang tersedia
2  def booking_room():
3      """Booking room"""
4      clear()
5      print("=" * 40)
6      print("BOOKING ROOM".center(40))
7      print("=" * 40)
8      room_kosong()
9
10     while True:
11         print("\nInput 0 jika ingin kembali")
12         sultan = input("Nama Sultan\t\t: ")
13         if sultan == "0":
14             return
15         elif sultan.strip() == "" or not sultan.isalpha():
16             clear()
17             print("Pastikan input hanya huruf!")
18             input("\nKlik enter untuk melanjutkan...")
19             clear()
20             room_kosong()
21         else:
22             break
23
24     rooms = muat_rooms()
25     rooms_kosong = []
26     for r, room in rooms.items():
27         if room["status"] == "Tersedia":
28             rooms_kosong.append(str(r))
29
30     print("Rooms kosong\t\t:", ", ".join(rooms_kosong))
31     nomor_room = input("Nomor Room\t\t: ")

```

```

1  if nomor_room in rooms_kosong:
2      waktu_booking = input("Waktu Booking (Jam)\t: ")
3      try:
4          waktu_booking = int(waktu_booking)
5          harga = rooms[nomor_room]["harga"] * waktu_booking
6          rooms[nomor_room]["status"] = "Digunakan"
7          rooms[nomor_room]["waktu_bermain"] = waktu_booking
8          rooms[nomor_room]["sultan"] = sultan
9          save_rooms(rooms)
10
11         if usersekarang == "admin":
12             print(f"Total bayar\t\t: Rp {harga:,.0f}")
13             input("\nKlik enter untuk melanjutkan...")
14
15         else:
16             clear()
17             print("=" * 43)
18             print("NOTA RESERVASI".center(43))
19             print("=" * 43)
20             print(f"| NAMA\t\t: {sultan:<24}|" )
21             print(f"| ROOM\t\t: {nomor_room:<24}|" )
22             print(f"| WAKTU (Jam)\t: {waktu_booking:<24}|" )
23             print(f"| TOTAL BAYAR\t: Rp {harga:,.0f}".ljust(40) + "|" )
24             print("="*43)
25             print("Silahkan hubungi @Admin untuk konfirmasi".center(43))
26             print("=" * 43)
27             time.sleep(5)
28             print("Pembayaran telah dikonfirmasi, Terimakasih")
29             input("\nKlik enter untuk melanjutkan...")
30
31     except ValueError:
32         clear()
33         print("Input waktu tidak valid!")
34         input("\nKlik enter untuk melanjutkan...")
35
36 else:
37     clear()
38     print("Room tidak tersedia.")
39     input("\nKlik enter untuk melanjutkan...")

```

Gambar 3. 29 Fungsi untuk memesan/booking room yang tersedia

```

1 # Fungsi untuk menghapus room yang ada
2 def clear_room():
3     """clear room"""
4     clear()
5     rooms = muat_rooms()
6     room_terpakai = []
7     for r, room in rooms.items():
8         if room["status"] == "Digunakan":
9             room_terpakai.append((r, room))
10
11     print("=" * 41)
12     print("ROOM YANG SEDANG DIGUNAKAN".center(41))
13     print("=" * 41)
14     print("| No. | \tSultan\t | \t Room\t |")
15     print("-" * 41)
16
17     for index, (nomor_room, room) in enumerate(room_terpakai, start=1):
18         print(f"| {index: <3} | {room.get('sultan'):<8} | \t {nomor_room:<15} |")
19

```

```

1 try:
2     print("-"*41)
3     print("\nInput 0 untuk kembali")
4     pilihan = int(input("Pilih nomor yang akan dihapus: "))
5
6     if pilihan == 0:
7         return
8     elif pilihan < 1 or pilihan > len(room_terpakai):
9         clear()
10        print("Nomor room tidak valid.")
11        input("\nKlik Enter untuk melanjutkan...")
12        return
13
14        hapus_room = room_terpakai[pilihan - 1][0]
15
16        rooms[hapus_room]["status"] = "Tersedia"
17        rooms[hapus_room]["waktu_bermain"] = 0
18        rooms[hapus_room]["sultan"] = ""
19
20        save_rooms(rooms)
21        clear()
22        print("DONE")
23    except ValueError:
24        clear()
25        print("Input tidak valid.")
26        input("\nKlik Enter untuk melanjutkan...")

```

Gambar 3. 30 Fungsi untuk menghapus room yang ada

```

1 akun = muat_akun() # Variabel akun menjadi tempat untuk menyimpan data akun yang ada
2 # Fungsi untuk menampilkan list akun yang telah dibuat
3 def list_akun():
4     """list akun"""
5     clear()
6     print("LIST AKUN".center(40))
7     print("=" * 40)
8     print("NO. | \t ID\t | \t Pass")
9     print("-" * 40)
10    akun_tampil = {username: password for username, password in akun.items() if username != "admin"}
11
12    for u, (username, password) in enumerate(akun_tampil.items(), start=1):
13        print(f"{u}\t {username}\t\t {'*' * len(password)}")
14
15    hapus = input("\nPilih nomor akun yang ingin dihapus: ")
16
17    if hapus.isdigit() and 0 < int(hapus) <= len(akun_tampil):
18        akun_hapus = list(akun_tampil.keys())[int(hapus) - 1]
19        del akun[akun_hapus]
20        save_akun(akun)
21    else:
22        print("\nPilihan tidak valid")
23    input("\nKlik enter untuk melanjutkan...")
24

```

Gambar 3. 31 Fungsi untuk menampilkan list akun yang telah dibuat

```

1 # Fungsi untuk informasi semua room
2 def room_kosong():
3     """daftar room tersedia"""
4     clear()
5     rooms = muat_rooms()
6     print("=" * 86)
7     print("LIST ROOM".center(86))
8     print("=" * 86)
9     print("| No. | \t ID Room\t | \t Status\t      | \t Fasilitas\t      | \t Harga/Jam      |")
10    print("-" * 86)
11
12    room_tersedia = []
13    for r, room in rooms.items():
14        room_tersedia.append((r, room))
15
16    if room_tersedia:
17        for index, (nomor_room, room) in enumerate(room_tersedia, start=1):
18            fasilitas = ", ".join(room["fasilitas"])
19            harga = room["harga"]
20            print(f"| {index:<3} | \t {nomor_room}\t | \t {room['status']:<15} | \t {fasilitas:<20} | \t Rp. {harga:,.0f} |")
21    else:
22        print("Tidak ada room yang kosong.")
23    input("\nKlik Enter untuk melanjutkan...")
24
25
26

```

Gambar 3. 32 Fungsi untuk informasi semua room


```

1  # Fungsi untuk menampilkan informasi room yang digunakan
2  def list_room():
3      """List room yang digunakan"""
4      clear()
5      rooms = muat_rooms()
6      print("=" * 43)
7      print("| ROOM PARA SULTAN |".center(43))
8      print("=" * 43)
9      print("| NO. | Sultan\t |\tRoom\t | Waktu |")
10     print("-" * 43)
11
12     rooms_digunakan = []
13     for r, room in rooms.items():
14         if room["status"] == "Digunakan":
15             rooms_digunakan.append((r, room))
16
17     for i, (nomor_room, room) in enumerate(rooms_digunakan, start=1):
18         print(f"| {i:<4} | {room.get("sultan"):<8} |\t {nomor_room:<6} | {room["waktu_bermain"]} Jam |")
19
20     input("\nKlik Enter untuk melanjutkan...")
21

```

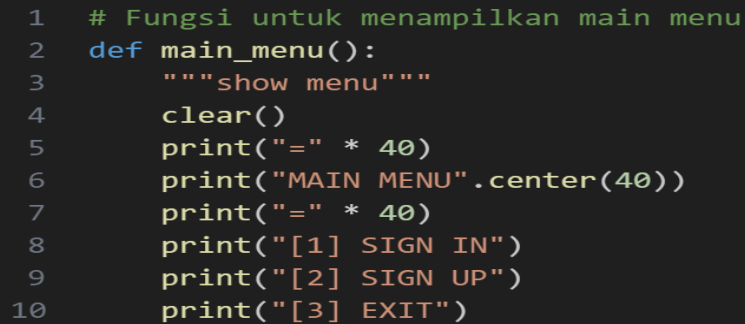
Gambar 3. 33 Fungsi untuk menampilkan informasi room yang digunakan

```

1  # Fungsi untuk menghapus terminal
2  def clear():
3      """clear terminal"""
4      os.system("cls || clear")
5

```

Gambar 3. 34 Fungsi untuk menghapus terminal

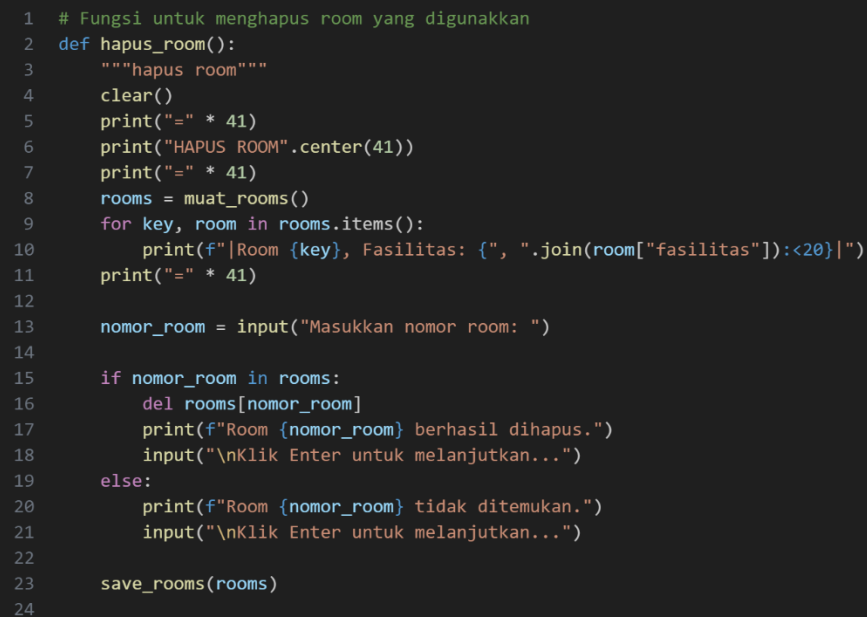


```

1  # Fungsi untuk menampilkan main menu
2  def main_menu():
3      """show menu"""
4      clear()
5      print("=" * 40)
6      print("MAIN MENU".center(40))
7      print("=" * 40)
8      print("[1] SIGN IN")
9      print("[2] SIGN UP")
10     print("[3] EXIT")

```

Gambar 3. 35 Fungsi untuk menampilkan main menu



```

1  # Fungsi untuk menghapus room yang digunakan
2  def hapus_room():
3      """hapus room"""
4      clear()
5      print("=" * 41)
6      print("HAPUS ROOM".center(41))
7      print("=" * 41)
8      rooms = muat_rooms()
9      for key, room in rooms.items():
10         print(f"|Room {key}, Fasilitas: {", ".join(room["fasilitas"]):<20}|")
11     print("=" * 41)
12
13     nomor_room = input("Masukkan nomor room: ")
14
15     if nomor_room in rooms:
16         del rooms[nomor_room]
17         print(f"Room {nomor_room} berhasil dihapus.")
18         input("\nKlik Enter untuk melanjutkan...")
19     else:
20         print(f"Room {nomor_room} tidak ditemukan.")
21         input("\nKlik Enter untuk melanjutkan...")
22
23     save_rooms(rooms)
24

```

Gambar 3. 36 Fungsi untuk menghapus room yang digunakan

```

1  # Fungsi untuk memperbarui (menambah/menghapus) fasilitas room
2  def update_room():
3      """update room"""
4      rooms = muat_rooms()
5      while True:
6          clear()
7          print("=" * 41)
8          print("UPDATE ROOM".center(41))
9          print("=" * 41)
10         # menampilkan ID Room dan fasilitas
11         for key, room in rooms.items():
12             print(f"|Room {key}, Fasilitas: {' '.join(room['fasilitas']):<20}|")
13         print("=" * 41)
14         print("Input 0 untuk kembali")
15         nomor_room = input("Masukkan Nomor Room: ")
16         if nomor_room == "0":
17             return
18         if nomor_room in rooms:
19             break
20         else:
21             clear()
22             print("Input tidak valid!")
23             input("\nKlik enter untuk melanjutkan...")
24
25     clear()
26     fasilitas_sekarang = rooms[nomor_room]["fasilitas"]
27     harga_sekarang = rooms[nomor_room]["harga"]

```

```

1  while True:
2      print("=" * 41)
3      print(f"Fasilitas Room {nomor_room}: {' '.join(fasilitas_sekarang)}.center(41))
4      print("=" * 41)
5      print("[1] Tambah fasilitas")
6      print("[2] Hapus fasilitas")
7      print("[3] Kembali")
8      pilihan = input("Masukkan pilihan: ")
9
10     if pilihan == "1":
11         list_fasilitas = ["Sim Racing", "PC", "PS 5"]
12         clear()
13         print("=" * 41)
14         print(f"Fasilitas Room {nomor_room}: {' '.join(fasilitas_sekarang)}.center(41))
15         print("=" * 41)
16
17         for f, fasilitas in enumerate(list_fasilitas, start=1):
18             print(f"{f}. {fasilitas}")
19
20         try:
21             pilih_fasilitas = int(input("\nPilih nomor fasilitas yang ditambahkan: "))
22             if pilih_fasilitas in range(1, len(list_fasilitas) + 1):
23                 fasilitas_baru = list_fasilitas[pilih_fasilitas - 1]
24
25                 if fasilitas_baru not in fasilitas_sekarang:
26                     fasilitas_sekarang.append(fasilitas_baru)
27                     clear()
28                     print(f"{fasilitas_baru} berhasil ditambahkan!")
29                     if fasilitas_baru == "Sim Racing":
30                         harga_sekarang += 40000
31                     elif fasilitas_baru == "PC" or fasilitas_baru == "PS 5":
32                         harga_sekarang += 20000
33                 else:
34                     clear()
35                     print(f"{fasilitas_baru} sudah ada di Room {nomor_room}")
36             else:
37                 clear()
38                 print("Nomor fasilitas tidak valid!")
39                 input("\nKlik enter untuk melanjutkan...")
40
41         except ValueError:
42             clear()
43             print("Input tidak valid!")
44             input("\nKlik enter untuk melanjutkan...")

```

```

1 elif pilihan == "2":
2     clear()
3     print("=" * 41)
4     print("Fasilitas yang tersedia untuk dihapus".center(41))
5     print("=" * 41)
6
7     for i, fasilitas in enumerate(fasilitas_sekarang, 1):
8         print(f"{i}. {fasilitas}")
9
10    try:
11        pilih_fasilitas = int(input("Pilih nomor fasilitas yang ingin dihapus: "))
12        if 1 <= pilih_fasilitas <= len(fasilitas_sekarang):
13            fasilitas_dihapus = fasilitas_sekarang.pop(pilih_fasilitas - 1)
14
15            if fasilitas_dihapus == "Sim Racing":
16                harga_sekarang -= 40000
17            elif fasilitas_dihapus == "PC" or fasilitas_dihapus == "PS 5":
18                harga_sekarang -= 20000
19            else:
20                clear()
21                print("Nomor fasilitas tidak valid!")
22                input("\nKlik enter untuk melanjutkan...")
23        except ValueError:
24            clear()
25            print("Input tidak valid!")
26            input("\nKlik enter untuk melanjutkan...")
27
28    elif pilihan == "3":
29        break
30

```

```

1 else:
2     clear()
3     print("Pilihan tidak valid!")
4     input("\nKlik Enter untuk melanjutkan...")
5
6     rooms[nomor_room]["fasilitas"] = fasilitas_sekarang
7     rooms[nomor_room]["harga"] = harga_sekarang
8     save_rooms(rooms)
9     input("\nKlik enter untuk melanjutkan...")
10    clear()
11

```

Gambar 3. 37 Fungsi untuk memperbarui (menambah/menghapus) fasilitas room

```

1  # Fungsi untuk menambah room baru kedalam list
2  def tambah_room():
3      """Tambah room baru"""
4      clear()
5      print("=" * 40)
6      print("ROOM BARU".center(40))
7      print("=" * 40)
8      rooms = muat_rooms()
9      nomor_room_baru = str(len(rooms) + 1)
10     harga = 0
11     fasilitas = []
12
13     print("[1] Rp. 80.000 (Sim Racing, PC, PS 5)")
14     print("[2] Rp. 40.000 (PC, PS 5)")
15     print("[3] Rp. 20.000 (PC)")
16
17     try:
18         pilih = int(input("Masukkan pilihan: "))
19
20         if pilih == 1:
21             harga = 80000
22             fasilitas = ["Sim Racing", "PC", "PS 5"]
23
24         elif pilih == 2:
25             harga = 40000
26             fasilitas = ["PC", "PS 5"]
27
28         elif pilih == 3:
29             harga = 20000
30             fasilitas = ["PC"]
31
32         else:
33             print("Pilihan tidak valid!")
34             input("\nKlik Enter untuk melanjutkan...")
35             return
36
37     except ValueError:
38         print("Input tidak valid!")
39         input("\nKlik Enter untuk melanjutkan...")
40         return
41
42     room_baru = {
43         "status" : "Tersedia",
44         "waktu_bermain" : 0,
45         "sultan" : "",
46         "harga" : harga,
47         "fasilitas" : fasilitas
48     }
49
50     rooms[nomor_room_baru] = room_baru
51     save_rooms(rooms)
52     print("=" * 40)
53     print(f"Room {nomor_room_baru} Ditambahkan".center(40))
54     print("=" * 40)
55

```

Gambar 3. 38 Fungsi untuk menambah room baru kedalam list

```

1  # Fungsi untuk menampilkan menu manajemen room
2  def atur_room():
3      """atur room"""
4      while True:
5          clear()
6          print("=" * 40)
7          print("MANAJEMEN ROOM".center(40))
8          print("=" * 40)
9          print("[1] Tambah Room Baru")
10         print("[2] Hapus Room")
11         print("[3] Update Fasilitas Room")
12         print("[4] Kembali")
13
14         try:
15             pilih = int(input("Masukkan Pilihan: "))
16             if pilih == 1:
17                 tambah_room()
18
19             elif pilih == 2:
20                 hapus_room()
21
22             elif pilih == 3:
23                 update_room()
24
25             elif pilih == 4:
26                 break
27             else:
28                 print("Input tidak valid!")
29                 input("\nKlik enter untuk melanjutkan...")
30                 clear()
31
32         except ValueError:
33             print("Input tidak valid!")
34             input("\nKlik enter untuk melanjutkan...")
35             clear()

```

Gambar 3. 39 Fungsi untuk menampilkan menu manajemen room

```

1  #MAIN PROGRAM
2  try:
3      while True:
4          clear()
5          main_menu()
6          pilihan = input("\nMasukkan Pilihan: ")
7          if pilihan == "1":
8              IS_LOGIN, usersekarang = login(akun)
9              if IS_LOGIN:
10                 while IS_LOGIN:
11                     clear()
12                     if usersekarang == "admin":
13                         admin_menu()
14                         pilihan = input("Masukkan pilihan: ")
15
16                     if pilihan == "0":
17                         list_akun()
18
19                     elif pilihan == "1":
20                         booking_room()
21
22                     elif pilihan == "2":
23                         clear_room()
24
25                     elif pilihan == "3":
26                         atur_room()
27
28                     elif pilihan == "4":
29                         list_room()
30
31                     elif pilihan == "5":
32                         room_kosong()
33
34                     elif pilihan == "6":
35                         IS_LOGIN = False
36
37                     elif pilihan == "7":
38                         clear()
39                         exit()
40
41                 else:
42                     print("\nInput Tidak Valid!")
43                     input("\nKlik enter untuk melanjutkan...")
44

```



```
1         else:
2             user_menu()
3             pilihan = input("Masukkan pilihan: ")
4             if pilihan == "1":
5                 booking_room()
6
7             elif pilihan == "2":
8                 room_kosong()
9
10            elif pilihan == "3":
11                IS_LOGIN = False
12
13            elif pilihan == "4":
14                clear()
15                exit()
16
17            else:
18                print("\nInput tidak valid!")
19                input("\nKlik enter untuk melanjutkan...")
20
21        elif pilihan == "2":
22            bikin_akun(akun)
23
24        elif pilihan == "3":
25            IS_LOGIN = False
26            clear()
27            break
28
29        else:
30            print("Input Tidak Valid!")
31            input("\nKlik enter untuk melanjutkan...")
32            continue
33    except KeyboardInterrupt:
34        print("\nProgram dihentikan.")
```

Gambar 3. 40 Main Program

BAB IV PENUTUP

4.1 Kesimpulan

Program manajemen ruangan yang telah dibuat bertujuan untuk mempermudah pengelolaan pemesanan dan penggunaan fasilitas seperti *Sim Racing*, *PC gaming*, dan *PS5* di sebuah warnet. Dengan adanya sistem ini, pemilik warnet dapat mengelola jadwal pemesanan dan penggunaan ruangan secara lebih efisien, serta meminimalisir konflik jadwal yang mungkin ada. Fitur-fitur seperti reservasi *room*, manajemen akun, pemantauan ketersediaan ruangan, dan pembaruan status ruangan memberikan kontrol yang lebih baik terhadap operasional warnet.

Selain itu, program ini juga memudahkan pelanggan untuk melakukan reservasi secara online dan melihat status ruangan yang tersedia, sehingga pelanggan menjadi lebih praktis dan terorganisir. Meskipun ada beberapa batasan seperti belum dapat menambahkan jam pada ruangan yang telah terpesan, sistem ini sudah memberikan solusi yang efektif bagi pengelolaan warnet modern yang menyediakan berbagai fasilitas.

Secara keseluruhan, program ini diharapkan dapat meningkatkan efisiensi operasional dan kualitas layanan, serta memberikan kenyamanan baik bagi pemilik warnet maupun pelanggan. Ke depannya, pengembangan fitur-fitur tambahan dapat dilakukan untuk lebih meningkatkan fungsionalitas program ini

4.2 Saran

Menambahkan opsi untuk memperbarui durasi reservasi atau menambah jam, dengan mempertimbangkan ketersediaan ruangan.

Menambahkan fitur untuk pelanggan memberikan ulasan atau rating setelah menggunakan fasilitas, serta memberikan feedback mengenai kondisi fasilitas dan pelayanan.

Menambahkan fitur pelaporan yang memberikan data tentang jumlah pemesanan, fasilitas yang paling banyak digunakan, dan penghasilan yang dihasilkan dari pemesanan untuk membantu pemilik mengambil keputusan bisnis yang lebih baik.

DAFTAR PUSTAKA

ASCII. (2024). Modul Praktikum Algoritma Pemrograman Dasar, *List & Tuple*, 1-14.

ASCII. (2024). Modul Praktikum Algoritma Pemrograman Dasar, *Fungsi dan Prosedur*, 1-2.

ASCII. (2024). Modul Praktikum Algoritma Pemrograman Dasar, *Dictionary*, 1.

ASCII. (2024). Modul Praktikum Algoritma Pemrograman Dasar, *Perulangan*, 1.

ASCII. (2024). Modul Praktikum Algoritma Pemrograman Dasar, *Percabangan*, 1-6.

LAMPIRAN

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)

Manajemen Room Warnet

Algoritma dan Pemrograman Dasar

KELAS IF B2 24 :

KELOMPOK 3:

Muhammad Ilma Yusrian Fahmi 2409106079



Ega Clearesta Hananta 2409106088



Keysha Khoirunnisa Aulia Khotim 2409106077



LABORATORIUM FAKULTAS TEKNIK
UNIVERSITAS MULAWARMAN 2024

Lampiran 1

Tanggal Konsultasi : 31 Oktober 2024	
Uraian / Pembahasan : <ul style="list-style-type: none"> • Sruve harga • Fitur update room • fitur lihat akun diganti jadi hapus akun • Password dibuat bintang (*) karena privasi user • User bisa reservasi room • Perjelas isi flowchart 	
Asisten Lab	Ketua Kelompok
	
Nama: M. AFRIZAL KESUMA	Nama: M. Ilma Yucian F.

Tanggal Konsultasi : 13 November 2024	
Uraian / Pembahasan : <ul style="list-style-type: none"> • Tampilan booking room sebelum booking ditampilkan list room yang tersedia beserta fasilitas dan harga. • Tambahkan fitur update room buat ganti fasilitas atau harga • Nama booking tidak bisa kosong. • Menu room Sultan hanya bisa di buat admin • Menu room Sultan diganti jadi info room • Menu nomornya urut jangan loncat - loncat (1, 2, 3, 4) 	
Asisten Lab	Ketua Kelompok
	
Nama: M. AFRIZAL KESUMA	Nama: M. Ilma Yucian F.

Lampiran 2