

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



**Manejemen kamar kos-kosan**

**Oleh:**

**Kelompok 1 A2**

<b>RAFFI ZIDANE WAHYUDIN</b>	<b>2409106037</b>
<b>MUHAMMAD RIZAL ALFATH</b>	<b>2409106039</b>
<b>MUHAMMAD RASYID</b>	<b>2409106042</b>
<b>RANGGA LAWE</b>	<b>2409106043</b>

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

## KATA PENGANTAR

Puji syukur kami panjatkan kepada Tuhan Yang Maha Esa, karena atas limpahan rahmatnya penyusun dapat menyelesaikan laporan proyek akhir ini tepat waktu tanpa ada halangan yang berarti dan sesuai dengan harapan. Laporan praktikum ini disusun sebagai salah satu bentuk pertanggungjawaban kegiatan praktikum sekaligus untuk mendalami konsep yang telah dipelajari di kelas selama 1 semester.

Program yang kami buat sendiri yaitu tentang manajemen kos kos an yang mana terdapat menu untuk melihat kamar kos, penghuni, mengatur kamar kos, serta bisa melakukan pemesanan dan memberikan rating untuk kamar dengan menerapkan semua materi atau pedoman yang ada di modul praktikum algoritma pemrograman dasar

Selama penyusunan laporan dan pembuatan program, kami menghadapi beberapa kendala, seperti kurang nya pemahaman akan materi di luar modul praktikum dan kurangnya komunikasi antar tim Namun, berkat bimbingan dari asisten praktikum, serta bantuan dan dukungan dari rekan-rekan, kendala tersebut dapat teratasi dengan baik. Untuk itu, kami mengucapkan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu

Kami menyadari bahwa laporan ini masih memiliki banyak kekurangan. Oleh karena itu, kami sangat mengharapkan kritik dan saran yang membangun agar selanjutnya dapat memberikan hasil yang lebih baik.

Samarinda, 18 november 2024

Kelompok 1

Kelompok 01A2 2024 | ii

## TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah sebagai berikut:

<i>Database</i>	Basis Data
<i>Managemen</i>	Mengatur
<i>Input</i>	Memasukkan
<i>Python</i>	bahasa program
<i>User</i>	penguna
<i>Password</i>	kata sandi
<i>Akun</i>	tempat penyimpanan
<i>Login</i>	proses masuk akun
<i>Rating</i>	penilaian
<i>CRUD</i>	program
<i>If/elif/else</i>	fungsi percabangan
<i>Cctv</i>	kamera pengawas
<i>Error</i>	kerusakan pada fungsi

## DAFTAR ISI

KATA PENGANTAR .....	ii
TAKARIR.....	iii
DAFTAR ISI.....	iv
DAFTAR GAMBAR.....	vi
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.....	2
BAB II PERANCANGAN.....	3
2.1 Analisis Program.....	3
2.2 Flowchart.....	3
2.3 Konsep/Materi Praktikum yang dipakai.....	4
BAB III HASIL DAN PEMBAHASAN.....	8
3.1 Tampilan Program.....	8
1. Tampilan Menu.....	8
2. Menu login.....	9
3. Menu admin (pemilik kos).....	9

4. Menu manage kamar.....	9
5. Menu manage penghuni.....	10
6. Menu lihat pemesanan.....	11
7. Menu lihat rating.....	12
8. Menu manage account.....	12
9. Menu user(penghuni kos).....	12
3.2 Source Code.....	13
BAB IV PENUTUP.....	50
4.1 Kesimpulan.....	50
4.2 Saran.....	50
DAFTAR PUSTAKA.....	51
LAMPIRAN.....	52

## DAFTAR GAMBAR

Gambar 1.1 flowchart.....	Error! Bookmark not defined.
Gambar 2.1 awal program.....	Error! Bookmark not defined.
Gambar 2.2 tampilan menu login .....	Error! Bookmark not defined.
Gambar 2.3 tampilan menu admin.....	Error! Bookmark not defined.
Gambar 2.4 manage kamar dan lihat kamar.....	Error! Bookmark not defined.
Gambar 2.5 tambah kamar.....	Error! Bookmark not defined.
Gambar 2.6 list kamar setelah ditambah.....	Error! Bookmark not defined.
Gambar 2.7 hapus kamar.....	Error! Bookmark not defined.
Gambar 2.8 list kamar setelah dihapus.....	Error! Bookmark not defined.
Gambar 2.9 update kamar.....	Error! Bookmark not defined.
Gambar 2.10 list kamar setelah diupdate .....	Error! Bookmark not defined.
Gambar 2.11 manage dan lihat kamar .....	Error! Bookmark not defined.
Gambar 2.12 tambah penghuni.....	Error! Bookmark not defined.
Gambar 2.13 hapus penghuni.....	Error! Bookmark not defined.
Gambar 2.14 jika tidak ada pemesanan.....	Error! Bookmark not defined.
Gambar 2.15 ada pemesanan.....	Error! Bookmark not defined.
Gambar 2.16 rating kos-kosan.....	Error! Bookmark not defined.
Gambar 2.17 untuk manage akun user.....	Error! Bookmark not defined.
Gambar 2.18 untuk akun user dan lihat kamar.....	Error! Bookmark not defined.
Gambar 2.19 user rating.....	Error! Bookmark not defined.
Gambar 2.20 menu akun user yang terkonfirmasi.....	Error! Bookmark not defined.
Gambar 2.21 akun user yang belum terkonfirmasi.....	Error! Bookmark not defined.
Gambar 3.1 lampiran konsul.....	Error! Bookmark not defined.

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Berkembangnya zaman membuat pengelolaan data yang efektif dan efisien adalah kebutuhan utama dalam segala bidang, termasuk dalam manajemen kos kosan, banyaknya pengelola kos kosan yang menggunakan pengelolaan data secara manual untuk mencatat informasi dan data penting lainnya ,yang mana hal itu sangat tidak efisien dan tidak efektif, oleh karena itu kami membuat program untuk membantu dalam mengelola data kos kosan agar lebih efisien dan efektif menggunakan python yang dirancang untuk kebutuhan manajemen kos kosan

### **1.2 Kebutuhan Fungsional**

1. Sistem mampu melakukan proses pembuatan akun untuk user dengan memasukkan nama dan password
2. Sistem mampu melakukan proses login melalui akun yang sudah dibuat sebelumnya dengan memasukkan nama dan password
3. Sistem mampu menampilkan data kos kosan ke user untuk melihat kamar kos kosong
4. Sistem mampu melakukan proses pemberian rating dari user untuk rating kos kosan
5. Sistem mampu menampilkan user untuk dilihat di menu admin
6. Sistem mampu memberi admin untuk manage kamar
7. Sistem mampu memberi admin untuk manage penghuni
8. Sistem mampu melihat rating untuk admin
9. Sistem mampu melihat nama user dan password user

### **1.3 Rumusan Masalah**

Rumusan masalah yang diangkat dalam pengembangan program ini adalah:

1. Bagaimana merancang sistem berbasis CRUD menggunakan Python untuk membantu pengelolaan kos-kosan?
2. Bagaimana program dapat memberikan kemudahan dalam pengelolaan data penyewa dan pembayaran secara efisien?
3. Bagaimana memastikan sistem ini memiliki antarmuka yang sederhana namun efektif untuk digunakan oleh pengelola kos-kosan?

### **1.4 Batasan Masalah**

1. Tidak mencakup manajemen kerusakan, isi, dan penambahan fasilitas seperti listrik dan air
2. Hanya ada admin sebagai pemilik kos dan user sebagai penyewa kos tanpa ada pihak ke-3 seperti agen
3. Tidak mencakup data pribadi seperti ktp dan ktm
4. Tidak mencakup manajemen proses pembayaran
5. Tidak mencakup koneksi dengan perangkat keras seperti cctv
6. Tidak mencakup sistem keamanan tingkat tinggi untuk password user

### **1.5 Tujuan**

Dengan demikian tujuan dari dibuatnya program ini untuk membantu efisiensi dan ke efektifitasan dari manajemen kos-kosan dengan program sederhana dan user interface sederhana



## **BAB II**

### **PERANCANGAN**

#### **2.1 Analisis Program**

Alurnya sendiri dimulai dengan menu untuk login yang mana ada pilihan untuk login atau registrasi

Lalu untuk login admin sudah ditetapkan nama dan password adminnya, setelah masuk menu admin terdapat menu untuk manage kamar, manage penghuni, lihat rating dan manage akun

Di menu manage sendiri, seperti namanya, untuk mengatur dari akun dan kamar

Lalu untuk login user, mereka harus registrasi terlebih dahulu lalu login menggunakan registrasi mereka, menu user sendiri terdapat menu untuk lihat kamar, pesan kamar, dan berikan rating

Ketika user memesan kamar, admin akan mengkonfirmasi untuk pesanan kamar mereka

#### **2.2 Flowchart**

Untuk alur dari flowchart dibawah ialah, dimulai dengan memilih apakah login admin, user, registrasi atau keluar jika memilih login admin maka akan masuk ke input username dan password, jika username dan password benar maka akan dilanjutkan ke fungsi fungsi admin, begitu juga jika login user, jika username dan password benar maka akan diarahkan ke fungsi fungsi penghuni, jika registrasi maka akan diarahkan ke input untuk username dan password



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. Tipe data: Seperti namanya, tipe data merupakan jenis dari suatu data yang akan kita gunakan. Setiap data tersebut memiliki nilainya, dan setiap nilai memiliki jenisnya. Ada yang bertipe angka, huruf/karakter, ada juga yang bertipe benar/salah.
3. percabangan: fungsi percabangan ada if/elif/else yang mana digunakan dalam menentukan pilihan dari tindakan yang diinput oleh pengguna baik admin maupun user
4. perulangan: fungsi perulangan yaitu for/while/break/continue, dengan penjelasan sebagai berikut:
  1. Perulangan for disebut juga sebagai counted loop (perulangan yang terhitung). Perulangan for biasanya digunakan untuk mengulangi kode yang sudah diketahui banyak perulangannya
  2. Perulangan while disebut juga sebagai uncounted loop (perulangan yang tak terhitung). Perulangan while memiliki syarat dan akan terus mengulang apabila memenuhi syarat tersebut. Oleh karena itu, perulangan while tidak tentu berapa banyak perulangannya.
  3. Fungsi break adalah perintah khusus yang dipakai untuk memaksa sebuah perulangan berhenti sebelum waktunya. Perintah break bisa digunakan untuk jenis perulangan baik while maupun perulangan for pada bahasa pemrograman Python.
  4. Fungsi continue adalah perintah yang digunakan di dalam loop untuk melewati iterasi saat ini dan langsung lanjut ke iterasi berikutnya,

tanpa menjalankan kode yang ada setelah pernyataan continue dalam iterasi tersebut.

5. List dan tuple: fungsi untuk menyimpan data, dengan penjelasan sebagai berikut:

1. List dapat menyimpan berbagai data sekaligus seperti string, integer, float, dan sebagainya. Kita bahkan dapat menyimpan list di dalam sebuah list yang disebut dengan nested list atau list multi-dimensi. List didefinisikan menggunakan tanda kurung siku [ ] dan di setiap anggota dari data tersebut dipisahkan oleh tanda koma.

2. Tuple pada python adalah 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. Tuple adalah salah satu struktur data di Python yang mampu menyimpan sekumpulan nilai dalam satu Variabel.

6. Dictionary: Dictionary adalah suatu tipe data pada python yang berfungsi untuk menyimpan kumpulan data/nilai. Berbeda dengan list. Dimana list bisa dibidang terbatas untuk cara pemanggilannya. Yaitu hanya dapat dipanggil menggunakan index-nya saja. Dictionary ini sendiri sesuai kalimat asalnya yaitu kamus. Dimana dalam kamus ada sebuah Kunci/Atribut dan Nilai/Informasinya. Sehingga kita cukup menggunakan kata kunci yang dimiliki oleh suatu informasi yang diinginkan untuk memanggilnya.

7. Fungsi dan prosedur: Program yang kompleks dan memiliki banyak fitur, diharuskan menggunakan fungsi. Mengapa harus menggunakan fungsi? Karena jika tidak menggunakannya kita akan kerepotan menulis kode programnya, coding yang banyak yang harus ditulis dan kode akan menjadi sulit dibaca dan dirawat (maintenance). Dengan fungsi, kita dapat memecah

program besar menjadi sub program yang lebih sederhana. Masing-masing fitur pada program dapat kita buat dalam satu fungsi. Pada saat kita membutuhkan fitur tersebut, kita tinggal panggil fungsinya saja. Hal ini akan kita coba pada contoh program yang sudah disediakan di bawah. Teori dasar dan hal apa saja yang harus kita ketahui tentang fungsi di Python.

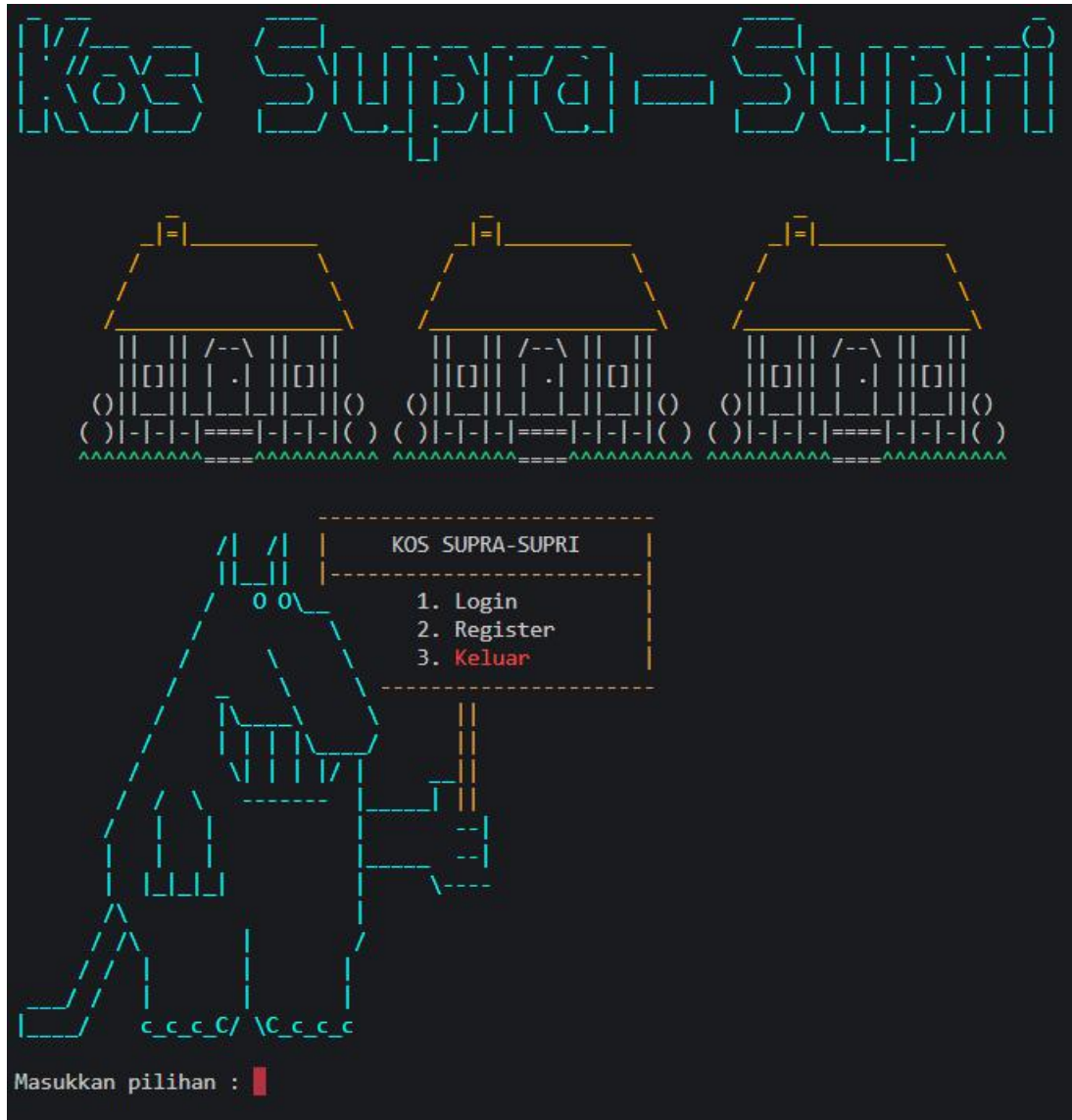
8. Error handling: cara mengatasi serta memanfaatkan apa yang dinamakan error dan membuat file yang dapat menampung data baik itu input maupun output pada suatu program. Hal yang pertama dilakukan ketika mendapat error, perhatikan error terjadi di file apa dan di line berapa, selanjutnya identifikasi error apa yang didapat. Pada kasus di atas, terjadi error di file main.py pada baris ke 1, dan tipe error berupa ValueError.
9. CRUD: create, read, update, delete:
  1. Create: Fungsi ini memungkinkan pengguna untuk menambahkan data
  2. Read: Fungsi ini digunakan untuk membaca dan menampilkan seluruh data
  3. Update: Fungsi ini memungkinkan pengguna memperbarui data yang telah tersimpan
  4. Delete: Fungsi ini digunakan untuk menghapus data yang tidak lagi diperlukan.

## BAB III

### HASIL DAN PEMBAHASAN

#### 3.1 Tampilan Program

##### 1. Tampilan Menu



Gambar 2.1 awal program.

Menu dimana program dimulai

## 2. Menu login

```
===== Login =====  
  
masukan username : admin  
masukan password : admin
```

Gambar 2.2 Tampilan menu login

Menu dimana admin atau user memasukkan username dan password

## 3. Menu admin (pemilik kos)

```
Berhasil login sebagai admin  
  
===== Menu Ibu Kos =====  
| 1. Manage Kamar |  
| 2. Manage Penghuni |  
| 3. Lihat Pemesan |  
| 4. Lihat Rating |  
| 5. Manage Account |  
| 6. Keluar |  
===== Menu Ibu Kos =====  
  
Pilih Menu : 1
```

Gambar 2.3 tampilan menu admin

Menu untuk admin (pemilik kos) untuk mengelola kos

## 4. Menu manage kamar

```
===== Manage kamar =====  
| ID | Fasilitas | Harga |  
===== Manage kamar =====  
| 1 | kamar berAC | 1,500,000 |  
| 2 | kamar biasa | 1,000,000 |  
| 3 | kamar VVIP | 5,000,000 |  
===== Manage kamar =====  
  
1. Lihat Kamar  
2. Tambah Kamar  
3. Hapus Kamar  
4. Kembali  
===== Manage kamar =====  
  
Pilih Menu: 1
```

Gambar 2.4 manage kamar dan lihat kamar

```
===== Tambah Kamar =====  
  
Masukkan nama kamar : kamarudin  
Masukkan harga kamar : 500000  
  
Kamar berhasil ditambahkan!  
  
Tambah kamar lain? (y/n):
```

Gambar 2.5 tambah kamar

ID	Fasilitas	Harga
1	kamar berAC	1,500,000
2	kamar biasa	1,000,000
3	kamar VVIP	5,000,000
4	kamarudin	500,000

Gambar 2.6 list kamar setelah ditambah

ID	Fasilitas	Harga
1	kamar berAC	1,500,000
2	kamar biasa	1,000,000
3	kamar VVIP	5,000,000
4	kamarudin	500,000

Masukkan ID kamar yang ingin dihapus : 4

Gambar 2.7 hapus kamar

ID	Fasilitas	Harga
1	kamar berAC	1,500,000
2	kamar biasa	1,000,000
3	kamar VVIP	5,000,000

Gambar 2.8 list kamar setelah dihapus

ID	Fasilitas	Harga
1	kamar berAC	1,500,000
2	kamar biasa	1,000,000
3	kamar VVIP	5,000,000

Masukan id yang ingin di edit : 1  
 Masukkan nama kamar baru: kamarudin  
 Masukkan harga kamar baru : 500000  
 Kamar berhasil diedit

Gambar 2.9 update kamar

ID	Fasilitas	Harga
1	kamarudin	500,000
2	kamar biasa	1,000,000
3	kamar VVIP	5,000,000

Gambar 2.10 list kamar setelah diupdate

Menu admin untuk manage kamar dengan fungsi untuk melihat,menambah,mengupdate dan menghapus kamar

## 5. Menu manage penghuni

Tidak ada Penghuni yang tersedia. Tidak ada pengguna yang memesan kamar.

Manage Penghuni

1. Lihat Penghuni  
2. Tambah Penghuni  
3. Hapus Penghuni  
4. Kembali

Manage Penghuni

1. Lihat Penghuni  
2. Tambah Penghuni  
3. Hapus Penghuni  
4. Kembali

Pilih Menu:

Pilih Menu:

Gambar 2.11 manage dan lihat penghuni

Gambar 2.12 tambah penghuni



```
Tidak ada penghuni yang terdaftar
=====
|      Manage Penghuni      |
=====
|  1. Lihat Penghuni  |
|  2. Tambah Penghuni |
|  3. Hapus Penghuni  |
|  4. Kembali         |
=====
Pilih Menu: 
```

Gambar 2.13 hapus penghuni

Menu manage penghuni dengan fungsi untuk melihat, menambah, dan menghapus penghuni, belum bisa/saat ini data penghuni kosong

## 6. Menu lihat pemesanan

```
Tidak ada pengguna yang memesan kamar.
=====
|      Menu Ibu Kos      |
=====
|  1. Manage Kamar  |
|  2. Manage Penghuni |
|  3. Lihat Pemesan  |
|  4. Lihat Rating   |
|  5. Manage Account |
|  6. Keluar         |
=====
Pilih Menu : 
```

Gambar 2.14 jika tidak ada pemesanan

```
=====
| ID | Nama | Fasilitas | Status |
=====
| 1 | user1 | kamarudin | Belum Dikonfirmasi |
=====

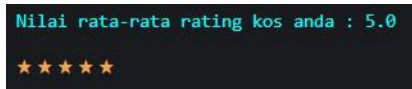
Menerima penghuni
1. Konfirmasi
2. Batalkan
3. kembali

Masukan Pilihan : 
```

Gambar 2.15 ada pemesanan

Menu untuk admin melihat pemesanan dan konfirmasi pemesanan kamar kos

## 7. Menu lihat rating



Nilai rata-rata rating kos anda : 5.0  
★★★★★

Gambar 2.16 rating kos-kosan

Menu untuk admin mengetahui rating dari kos-kosan mereka

## 8. Menu manage account



```
=====
| ID | Nama | password |
=====
| 1 | user1 | 1 |
=====

=====
| Manage Account |
=====
| 1. Lihat Akun |
| 2. Tambah Akun |
| 3. Hapus Akun |
| 4. Kembali |
=====

Masukan pilihan : █
```

Gambar 2.17 untuk manage akun user

Menu untuk admin melihat user dan menambah serta menghapus user

## 9. Menu user(penghuni kos)

```

=====
| ID | Fasilitas | Harga |
=====
| 1 | kamarudin | 500,000 |
| 2 | kamar biasa | 1,000,000 |
| 3 | kamar VVIP | 5,000,000 |
=====

Menu Penghuni
=====
1. Lihat Kamar
2. Pesan Kamar
3. Berikan Rating
4. Keluar
=====

Pilih Menu : 

```

Gambar 2.18 untuk akun user dan lihat kamar

```

===== Rating =====

Berikan rating Untuk Kos-kosan
Rating (1-5) : 5

```

Gambar 2.19 user rating

```

Berhasil Login sebagai user
Hallo user1 Selamat datang

Menu Penghuni
=====
1. Lihat Kamar
2. Pesan Kamar
3. Berikan Rating
4. Keluar
=====

Status : DiKonfirmasi
Anda sudah menjadi penghuni kos supra-supri
Pilih Menu : 

```

Gambar 2.20 menu akun user yang terkonfirmasi

```

kamarudin berhasil dipesan!

Menu Penghuni
=====
1. Lihat Kamar
2. Pesan Kamar
3. Berikan Rating
4. Keluar
=====

Status : Menunggu Konfirmasi
Pilih Menu : 

```

Gambar 2.21 akun user yang belum terkonfirmasi

Menu user untuk melihat, memesan, dan memberi rating untuk kos-kosan

### 3.2 Source Code

```

# List untuk menyimpan data pengguna yang terdaftar

users = [{"-"}]

# Data fasilitas yang tersedia dalam bentuk dictionary

```

```

data_fasilitas = {

    1: {'fasilitas': 'kamar berAC', 'harga': 1500000},

    2: {'fasilitas': 'kamar biasa', 'harga': 1000000},

    3: {'fasilitas': 'kamar VVIP', 'harga': 5000000}

}

rating = ["-"]

# nama,fasilitas,status

pesanan = [[],[],[ ]]

penghuni = [[],[],[ ]]

#warna

hijau = "\033[92m"

merah = "\033[91m"

biru = "\033[38;2;0;255;255m"

kuning = "\033[33m"

putih = "\033[0m"

orange = "\033[38;5;214m"

def login(admin_login):

    while True:

        global penghuni_sekarang

        username = input(f"{putih}masukan username : {biru}").strip().replace(" ",
""").lower()

        password = input(f"{putih}masukan password : {biru}").strip().replace(" ",
""").lower()

        if (username, password) == admin_login:

            print("\033c", end="")

            print(f"{hijau}Berhasil login sebagai admin{putih}\n")

            menu_admin() # Memanggil menu_admin setelah login berhasil

```

```

        return

    else:

        for i in range(len(users)):

            if username == users[i][0] and password == users[i][1]:

                print("\033c", end="")

                print(f"{hijau}Berhasil Login sebagai user \nHallo {username} Selamat datang{putih}\n")

                penghuni_sekarang = username

                setting_penghuni_menu() # Memanggil setting_penghuni_menu setelah login user berhasil

                return

            print("\033c", end="")

            print(f""{kuning}Username atau password salah{putih}

1. Coba Lagi

2. Kembali

""")

        pilihan = input("Masukan Pilihan : ")

        if pilihan == "1":

            print("\033c", end="")

            print(f"{biru}==== Login ==== {putih}\n")

            continue

        elif pilihan == "2":

            return

        else:

            while True:

                print("\033c", end="")

```

```

        print(f""{kuning}Maaf Input Salah, Pilihan hanya 1-2{putih}

1. Coba Lagi

2. Kembali

""")

    pilihan = input("Masukan Pilihan : ")

    if pilihan == "1":

        print("\033c", end="")

        print(f""{biru}==== Login ==== {putih}\n")

        break

    elif pilihan == "2":

        return

def register():

    x = True

    while x:

        while True:

            username = input(f""{putih}masukan username : {biru}").strip().replace(" ",

"".lower()

            if len(username) > 15:

                print("\033c", end="")

                print(f""\n{merah}Input terlalu panjang, Maksimal 15 karakter{putih}")

                continue

            elif username.isalnum():

                pass

            else:

                print(f""\n\033[91mUsername Tidak diperbolehkan menggunakan special

karakter\033[0m")

                continue

```

```

        break

    while True:

        password = input(f"{putih}masukan password : {biru}").strip()

        if not password: # Jika password kosong

            print(f"\n{merah>Password tidak boleh kosong!{putih}")

            continue

        break

    for i in range(len(users)):

        if username == users[i][0] or username == "admin":

            print("\033c", end="")

            print(f"\n{merah}Maaf Username yang anda pilih sudah digunakan, silahkan
memasukan username baru{putih}")

            break

        elif i == len(users) - 1:

            x = False

    rating.append("-")

    user = [username,password]

    users.append(user)

    print("\033c", end="")

    print(f"{hijau}Register berhasil{putih}")

# Fungsi untuk menampilkan menu Login

def menu_login():

    while True:

        print(f""{biru}

                {kuning}-----{biru}

                /|  /|  {kuning}|    {putih} KOS SUPRA-SUPRI    {kuning}|{biru}

```

```

    ||__|| {kuning}|-----|{biru}

    /  0 0\\__      {putih}1. Login      {kuning}|{biru}

    /          \\      {putih}2. Register    {kuning}|{biru}

    /      \\      \\      {putih}3. {merah}Keluar      {kuning}|{biru}

    /  _      \\      \\ {kuning}-----{biru}

    /  |\\__\\      \\      {kuning}|{biru}

    /  | | | |\\__/      {kuning}|{biru}

    /      \\| | | | / |      __{kuning}|{biru}

    / /  \\      ----- |__| {kuning}|{biru}

    /  |  |      |      --|

    |  |  |      |__      --|

    |  |_|_|      |      \\----

    /\\          |

    /  \\      |      /

    / /  |      |      |

    __/ /  |      |      |

    |__/_      c_c_c_C/  \\C_c_c_c

    {putih}""")

```

```

pilihan_menu = input(f"Masukkan pilihan : {biru}")

```

```

if pilihan_menu == "1":

```

```

    print("\033c", end="")

```

```

    print(f"{biru}==== Login ==== {putih}\n")

```

```

    login(("admin", "admin"))

```



```

elif pilihan_menu == "2":

    register()

elif pilihan_menu == "3":

    print(f"\n{biru}keluar dari sistem{putih}\n")

    return

else :

    print("\033c", end="")

    print(f"\n{kuning}Pilihan tidak valid{putih}")

# Fungsi untuk menampilkan menu admin
def menu_admin():

    while True:

        print(f""{biru}=====

| {putih}Menu Ibu Kos{biru}      |

=====

| {putih} 1. Manage Kamar {biru}    |

| {putih} 2. Manage Penghuni {biru} |

| {putih} 3. Lihat Pemesan {biru}  |

| {putih} 4. Lihat Rating {biru}   |

| {putih} 5. Manage Account {biru}  |

| {putih} 6. {merah}Keluar{biru}    |

=====

{putih}

""")

        menu = input("Pilih Menu : ")

        if menu == '1':

            print("\033c", end="")

```

```

        manage_kamar()

elif menu == '2':

    print("\033c", end="")

    manage_penghuni()

elif menu == '3':

    print("\033c", end="")

    lihat_pemesan()

elif menu == '4':

    print("\033c", end="")

    lihat_rating()

elif menu == '5':

    print("\033c", end="")

    manage_account()

elif menu == '6':

    print("\033c", end="")

    print(f"{hijau}Anda telah Logout Dari Ibu Kos{putih}")

    return

else :

    print("\033c", end="")

    print(f"{kuning}Pilihan tidak valid{putih}\n")

    menu_admin()

```

```

# Fungsi untuk menampilkan menu penghuni

def setting_penghuni_menu():

    while True:

        print(f""{biru}=====

|      {putih} Menu Penghuni {biru}      |

=====

|      {putih} 1. Lihat Kamar {biru}      |
|      {putih} 2. Pesan Kamar {biru}      |
|      {putih} 3. Berikan Rating {biru}    |
|      {putih} 4.{merah} Keluar {biru}      |

=====

{putih}

        ""

    )

    x = False

    for i in range(len(pesanan)):

        for indeks_pesanan in range(len(pesanan[i])):

            if penghuni_sekarang in pesanan[i][indeks_pesanan]:

                print(f""{biru}Status : {kuning}Menunggu Konfirmasi{putih}")

                x = True

                break

            if x == True:

                break

    x = False

    for i in range(len(penghuni)):

        for indeks_pesanan in range(len(penghuni[i])):

```

```

        if penghuni_sekarang == penghuni[i][indeks_pesanan][0]:

            print(f"{biru}Status : {hijau}DiKonfirmasi{putih}")

            print(f"{hijau}Anda sudah menjadi penghuni kos supra-supri{putih}")

            x = True

            break

    if x == True:

        break

menu_penghuni = input("Pilih Menu : ")

if menu_penghuni == '1':

    print("\033c", end="")

    lihat_kamar()

elif menu_penghuni == '2':

    print("\033c", end="")

    lihat_kamar()

    pesan_kamar()

elif menu_penghuni == '3':

    print("\033c", end="")

    memberikan_rating()

elif menu_penghuni == '4':

    print("\033c", end="")

    print("Keluar dari menu penghuni.")

    break

else:

    print("\033c", end="")

```

```

        print(f"{kuning}Pilihan tidak valid{putih}\n")

def manage_kamar():

    while True:

        print(f""{biru}=====

|      {putih} Manage kamar {biru}      |

=====

|      {putih} 1. Lihat Kamar {biru}      |
|      {putih} 2. Tambah Kamar {biru}      |
|      {putih} 3. Edit Kamar {biru}      |
|      {putih} 4. Hapus Kamar {biru}      |
|      {putih} 5. {merah}Kembali {biru}      |

=====
{putih}

""")

        pilihan = input("Pilih Menu: ")

        if pilihan == "1":

            print("\033c", end="")

            lihat_kamar()

        elif pilihan == "2":

            print("\033c", end="")

            tambah_kamar()

        elif pilihan == "3":

            print("\033c", end="")

            lihat_kamar()

            edit_kamar()

        elif pilihan == "4":

```

```

        print("\033c", end="")

        hapus_kamar()

    elif pilihan == "5":

        print("\033c", end="")

        return

    else:

        print("\033c", end="")

        print(f"{kuning}Pilihan tidak valid{putih}\n")

def manage_account():

    while True:

        print(f""{biru}=====

| {putih} Manage Account {biru} |

=====

| {putih} 1. Lihat Akun {biru} |

| {putih} 2. Tambah Akun {biru} |

| {putih} 3. Hapus Akun {biru} |

| {putih} 4.{merah} Kembali {biru} |

=====

{putih}

""")

        pilihan = input("Masukan pilihan : ")

        if pilihan == "1":

            print("\033c", end="")

            lihat_account()

```

```

elif pilihan == "2":

    print("\033c", end="")

    print(f"{biru}===== Tambah Account =====\n")

    register()

elif pilihan == "3":

    print("\033c", end="")

    lihat_account()

    hapus_account()

elif pilihan == "4":

    print("\033c", end="")

    return

else:

    print("\033c", end="")

    print(f"{kuning}input invalid{putih}")

def manage_penghuni():

    while True:

        print(f""{biru}=====

|   {putih} Manage Penghuni {biru}   |

=====

|   {putih} 1. Lihat Penghuni {biru}   |

|   {putih} 2. Tambah Penghuni {biru}  |

|   {putih} 3. Hapus Penghuni {biru}   |

```

```

| {putih} 4.{merah} Kembali {biru} |

===== {putih}

    """

    pilihan = input("Pilih Menu: ")

    if pilihan == "1":

        print("\033c", end="")

        lihat_penghuni()

    elif pilihan == "2":

        print("\033c", end="")

        lihat_pemesan()

    elif pilihan == "3":

        print("\033c", end="")

        lihat_penghuni()

        hapus_penghuni()

    elif pilihan == "4":

        print("\033c", end="")

        return

    else:

        print("\033c", end="")

        print(f"{kuning}Pilihan tidak valid{putih}\n")

def lihat_kamar():

    #Menghitung panjang maksimum untuk ID

    panjang_maks_id = 0

    for x in data_fasilitas.keys():

        panjang_sekarang = len(str(x))

        if panjang_sekarang > panjang_maks_id:

```



```

        panjang_maks_id = panjang_sekarang

# Menghitung panjang maksimum untuk fasilitas
panjang_maks_fasilitas = 0

for x in data_fasilitas.values():

    panjang_sekarang = len(x['fasilitas'])

    if panjang_sekarang > panjang_maks_fasilitas:

        panjang_maks_fasilitas = panjang_sekarang

# Menghitung panjang maksimum untuk harga
panjang_maks_harga = 0

for nilai_dalam_fasilitas in data_fasilitas.values():

    panjang_harga_sekarang = len(f"{nilai_dalam_fasilitas['harga'],}")

    if panjang_harga_sekarang > panjang_maks_harga:

        panjang_maks_harga = panjang_harga_sekarang

# Membuat panjang maksimum menjadi genap
if panjang_maks_id % 2 != 0:

    panjang_maks_id += 1

if panjang_maks_fasilitas % 2 != 0:

    panjang_maks_fasilitas += 1

if panjang_maks_harga % 2 != 0:

    panjang_maks_harga += 1

tabel = f"{biru}|{putih} {'ID'.center(panjang_maks_id)} {biru}|
{putih}{'Fasilitas'.center(panjang_maks_fasilitas)} {biru}|
{putih}{'Harga'.center(panjang_maks_harga)} {biru}|{putih}"

print(f"{biru}=" * (len(tabel) - 84))

print(tabel)

print(f"{biru}=" * (len(tabel) - 84))

# Menampilkan data dalam tabel

```

```

for id, item in data_fasilitas.items():

    fasilitas = item['fasilitas'].ljust(panjang_maks_fasilitas,)

    harga = f"{item['harga']:,}".ljust(panjang_maks_harga)

    print(f"{biru}| {putih}{str(id).ljust(panjang_maks_id)} {biru}| {putih}{fasilitas}
{biru}| {putih}{harga} {biru}|{putih}")

    print(f"{biru}=" * (len(tabel) - 84), f"{putih}\n")

# Fungsi rekursif untuk menambahkan kamar
def tambah_kamar():

    kamar_id = len(data_fasilitas) + 1

    print("\033c", end="")

    print(f"{biru}===== Tambah Kamar ====={putih}\n")

    # Validasi input nama kamar

    while True:

        nama_kamar = input("Masukkan nama kamar : ").strip()

        try:

            for char in nama_kamar:

                if not (char.isalpha() or char == ' '):

                    raise ValueError(f"{merah}Input hanya boleh berisi alfabet dan
spasi.{putih}")

            teks_bersih = ' '.join(nama_kamar.split())

            jumlah_kata = len(teks_bersih.split())

            if jumlah_kata > 30:

                raise ValueError(f"{merah}Input tidak boleh lebih dari 30 kata, saat ini
ada {jumlah_kata} kata.{putih}")

        except ValueError as error:

            print("\033c", end="")

```

```

        print(f"{biru}==== Tambah Kamar ====={putih}\n")

        print(f"{error}")

        continue

    if not nama_kamar:

        print(f"{merah>Nama kamar tidak boleh kosong!{putih}\n")

    else:

        break

# Validasi input harga kamar

while True:

    try:

        harga_kamar = int(input("Masukkan harga kamar : "))

        if harga_kamar < 0: # Jika harga negatif

            print(f"{merah>Harga kamar tidak boleh negatif!{putih}\n")

        else:

            break

    except ValueError: # Jika input bukan angka

        print("\033c", end="")

        print(f"{biru}==== Tambah Kamar ====={putih}\n")

        print(f"{merah>Input hanya berupa angka!{putih}\n")

data_fasilitas[kamar_id] = {'fasilitas': nama_kamar, 'harga': harga_kamar}

pesanan.append([])

penghuni.append([])

print(f"\n{hijau>Kamar berhasil ditambahkan!{putih}")

```

```

# Meminta pengguna menambah kamar lain atau kembali

lanjut = input(f"\nTambah kamar lain? ({hijau}y{putih}/{merah}n{putih}): ").lower()

if lanjut == 'y':

    print("\033c", end="")

    tambah_kamar() # Rekursif untuk menambah kamar lain

elif lanjut == 'n':

    print("\033c", end="")

    return

else:

    print("\033c", end="")

    print(f"{kuning}Input invalid{putih}")

    return

def edit_kamar():

    while True:

        try:

            pilihan = int(input("Masukan id yang ingin di edit : "))

            break

        except ValueError:

            print("\033c", end="")

            lihat_kamar()

            print(f"{kuning}Input hanya angka{putih}")

    for key in data_fasilitas:

        if pilihan == key:

            while True:

                nama_baru = input("Masukkan nama kamar baru: ").strip()

```

```

try:

    for char in nama_baru:

        if not (char.isalpha() or char == ' '):

            raise ValueError(f"{merah}Input hanya boleh berisi alfabet dan
spasi.{putih}")

    teks_bersih = ' '.join(nama_baru.split())

    jumlah_kata = len(teks_bersih.split())

    if jumlah_kata > 30:

        raise ValueError(f"{merah}Input tidak boleh lebih dari 30 kata,
saat ini ada {jumlah_kata} kata.{putih}")

except ValueError as error:

    print("\033c", end="")

    print(f"{biru}==== Edit Kamar ====={putih}\n")

    print(f"{error}")

    continue

if not nama_baru:

    print(f"{merah>Nama kamar tidak boleh kosong!{putih}\n")

else:

    break

while True:

    try:

        harga_baru = int(input("Masukkan harga kamar baru : "))

        if harga_baru < 0: # Jika harga negatif

            print(f"{kuning}Harga kamar tidak boleh negatif!{putih}\n")

        elif harga_baru <= 100000:

            print(f"{kuning}Harga kamar terlalu murah!{putih}\n")

```

```

        else:

            break

    except ValueError: # Jika input bukan angka

        print("\033c", end="")

        print(f"{biru}===== Edit Kamar ====={putih}\n")

        print(f"{merah}Input hanya berupa angka!{putih}\n")

    data_fasilitas[key]['fasilitas'] = nama_baru

    data_fasilitas[key]['harga'] = harga_baru

    print(f"{hijau}Kamar berhasil diedit{merah}\n")

    return

else:

    print("\033c", end="")

    print(f"{kuning}ID kamar tidak ditemukan{putih}")

    return

def pesan_kamar():

    global penghuni

    for indeks_penghuni in range(len(penghuni)):

        for indeks_dalam in range(len(penghuni[indeks_penghuni])):

            if penghuni_sekarang == penghuni[indeks_penghuni][indeks_dalam][0]:

                print(f"{biru}kamu sudah Diterima menjadi penghuni{putih}")

                print(f"Apakah kamu mau membatalkan kamar\n")

```

```

        print(f"""\n{biru}Batalkan kamar
{putih}1. Batalkan Kamar
2. {merah}Kembali{putih}
""")

        pilihan = input("Masukan pilihan : ")

        if pilihan == "1":

            print("\033c", end="")

            print(f"{hijau}Kamar berhasil dibatalkan{putih}")

            del penghuni[indeks_penghuni][indeks_dalam]

            return

        elif pilihan == "2":

            print("\033c", end="")

            return

        else:

            print("\033c", end="")

            print(f"{kuning}input invalid{putih}")

            return

    for i in range(len(pesanan)):

        for indeks_pesanan in range(len(pesanan[i])):

            if penghuni_sekarang == pesanan[i][indeks_pesanan][0]:

                print(f"{kuning}Maaf Pengguna hanya dapat memesan 1 kamar saja\n{biru}anda
dapat membatalkan pesanan kamar anda dan memilih kamar baru{putih}")

                print(f"""\n{biru}Pesanan Kamar
{putih}1. Batalkan Kamar
2. {merah}Kembali{putih}
""")

```

```

        pilihan = input("Masukan pilihan : ")

        if pilihan == "1":

            print("\033c", end="")

            print(f"{hijau}Kamar berhasil dibatalkan{putih}")

            del pesanan[i][indeks_pesanan]

            return

        elif pilihan == "2":

            print("\033c", end="")

            return

        else:

            print("\033c", end="")

            print(f"{kuning}input invalid{putih}")

            return

    while True:

        try:

            id_kamar = int(input("Masukkan ID kamar yang ingin dipesan : "))

            break

        except ValueError:

            print("\033c", end="")

            lihat_kamar()

            print(f"{merah}Input Hanya Berupa Angka{putih}")

    if id_kamar in data_fasilitas:

        for indeks_fasilitas in range(len(data_fasilitas)):

```



```

        if id_kamar == indeks_fasilitas + 1:

            pesanan_baru = [penghuni_sekarang,data_fasilitas[id_kamar]["fasilitas"],
"Belum Dikonfirmasi"]

            pesanan[indeks_fasilitas].append(pesanan_baru)

            break

        print("\033c", end="")

        print(f"{hijau}{data_fasilitas[id_kamar]['fasilitas']} berhasil dipesan!{putih}\n")

    else:

        print("\033c", end="")

        print(f"{kuning}ID kamar tidak ditemukan.{putih}\n")

# Prosedur untuk menghapus kamar berdasarkan ID
def hapus_kamar():

    lihat_kamar()

    while True:

        try:

            id_kamar = int(input("Masukkan ID kamar yang ingin dihapus : "))

            break

        except ValueError:

            print("\033c", end="")

            print(f"{merah}Input Hanya Berupa Angka{putih}\n")

            lihat_kamar()

    if id_kamar in data_fasilitas:

        kamar = data_fasilitas.pop(id_kamar)

        print("\033c", end="")

        print(f"{hijau}{kamar['fasilitas']} berhasil dihapus.{putih}\n")

    else:

```

```

        print("\033c", end="")

        print(f"{kuning}Fasilitas tidak ditemukan.{putih}\n")

# Prosedur untuk menampilkan daftar pengguna
def lihat_penghuni():

    gak_ada_indeks = 0

    for a in range(len(penghuni)):

        if not penghuni[a]:

            gak_ada_indeks += 1

    if gak_ada_indeks == len(penghuni):

        print(f"{kuning}Tidak ada Penghuni yang tersedia.{putih}\n")

        return

    else :

        tabel(penghuni,0)

# Prosedur untuk menghapus penghuni
def hapus_penghuni():

    ga_ada_indeks = 0

    for x in range(len(penghuni)):

        if not penghuni[x]:

            ga_ada_indeks += 1

    if ga_ada_indeks == len(penghuni):

        print("\033c", end="")

        print(f"{kuning}Tidak ada penghuni yang terdaftar{putih}")

        return

    pilihan = int(input("Masukan Pilihan : "))

```

```

if pilihan >= len(penghuni) or pilihan <= 0:

    print("\033c", end="")

    print(f"{kuning}ID Penghuni Tidak ditemukan{putih}\n")

    return

else:

    p = False

    for i in range(len(penghuni)):

        for indeks_penghuni in range(len(penghuni[i])):

            if id_nama_list[pilihan - 1] == penghuni[i][indeks_penghuni][0]:

                del penghuni[i][indeks_penghuni]

                del id_nama_list[pilihan - 1]

                p = True

                break

            if p == True:

                break

def lihat_account():

    global id_nama_list_account

    panjang_maks_id = 2

    panjang_maks_nama = 4

    panjang_maks_password = 8

    for i in range(1, len(users)):

        panjang_nama = len(users[i][0])

        panjang_password = len(users[i][1])

```

```

        if panjang_nama > panjang_maks_nama:

            panjang_maks_nama = panjang_nama

        if panjang_password > panjang_maks_password:

            panjang_maks_password = panjang_password


panjang_id = len(str(len(users )-1))


if panjang_id > panjang_maks_id:

    panjang_maks_id = panjang_id

if panjang_maks_id % 2 != 0:

    panjang_maks_id += 1

if panjang_maks_nama % 2 != 0:

    panjang_maks_nama += 1

if panjang_maks_password % 2 != 0:

    panjang_maks_password += 1


if len(users) > 1 :

    tabels = f"| {'ID'.center(panjang_maks_id)} | {'Nama'.center(panjang_maks_nama)} | {'password'.center(panjang_maks_password)} | "

    print(f"{biru}="* len(tabels),f"{putih}")

    print(tabels)

    print(f"{biru}="* len(tabels),f"{putih}")

    id = 1

    id_nama_list_account = []

    for i in range(1,len(users)):

```

```

        if len(users) == 1:

            continue

        else:

            print(f"| {str(id).ljust(panjang_maks_id)} |  

{users[i][0].ljust(panjang_maks_nama)} | {users[i][1].ljust(panjang_maks_password)} |")

            id += 1

            id_nama_list_account.append(users[i][0])

        print("=" * len(tabels), "\n")

    else:

        print(f"{kuning}Tidak ada account yang tersedia{putih}\n")

def hapus_account():

    global id_nama_list_account

    x = True

    while x:

        if len(users) == 1:

            print("\033c", end="")

            print(f"T{kuning}idak ada Account yang terdaftar{putih}\n")

            return

        else:

            print(f"\n{biru}==== Hapus Account ====={putih}")

            pilihan = int(input("Masukan ID yang ingin dihapus : "))

            for i in range(1, len(users)):

                if pilihan > (len(users)-1) or pilihan <= 0 :

                    print("\033c", end="")

                    print(f"{kuning}Input invalid{putih}\n")

```

```

return

elif id_nama_list_account[pilihan -1] == users[i][0]:

    print("\033c", end="")

    print(f"{hijau}Account dengan nama {users[i][0]} telah dihapus{putih}")

    del rating[pilihan]

    del id_nama_list_account[pilihan -1]

    x = False

for j in range(len(pesanan)):

    for indeks in range(len(pesanan[j])):

        if pesanan[j][indeks][0] == users[i][0]:

            del pesanan[j][indeks]

            del users[i]

            x = True

            break

    if x == True:

        break

for j in range(len(penghuni)):

    for indeks in range(len(penghuni[j])):

        if penghuni[j][indeks][0] == users[i][0]:

            del penghuni[j][indeks]

            del users[i]

            x = True

            break

```

```

        if x == True:

            break

        if x == False:

            del users[i]

        return

def memberikan_rating():

    for x in range(1, len(users)):

        if users[x][0] == penghuni_sekarang:

            print("\033c", end="")

            print(f"{biru}==== Rating ==== {putih}\n")

            while True:

                print(f"{biru}Berikan rating Untuk Kos-kosan {putih}")

                try:

                    hasil_rating = int(input("Rating (1-5) : "))

                except ValueError:

                    print("\033c", end="")

                    print(f"{biru}==== Rating ==== {putih}\n")

                    print(f"{merah}input Invalid {putih}\n")

                    continue

                if hasil_rating > 5 or hasil_rating < 1:

                    print("\033c", end="")

                    print(f"{biru}==== Rating ==== {putih}\n")

                    print(f"{kuning}Berikan Rating hanya (1-5) {putih}\n")

```

```

        continue

    break

    print("\033c", end="")

    print(f"{hijau}Rating Telah Diberikan, Terima Kasih{putih}\n")

    rating[x] = hasil_rating

def lihat_rating():

    total_rating = 0

    users_rating = 0

    for x in range(1, len(users)):

        if rating[x] == "-":

            continue

        users_rating += 1

        total_rating += rating[x]

        #print(f"{users[x][0]} : {rating[x]}")

    if users_rating == 0:

        print(f"{kuning}tidak ada yang memberi rating kos-kosan anda{putih}\n")

    else:

        nilai_rata2 = total_rating / users_rating

        print(f"{biru}Nilai rata-rata rating kos anda : {nilai_rata2:.1f}{putih}\n")

        for i in range(int(nilai_rata2)):

            print(f"{kuning}★{putih}", end=" ")

        print("\n")

# misal masing masing maks 3

def lihat_pemesan():

```



```

global cek_tabel

cek_tabel = 1

while True:

    gak_ada_indeks = 0

    for a in range(len(pesanan)):

        if not pesanan[a]:

            gak_ada_indeks += 1

    if gak_ada_indeks == len(pesanan):

        print(f"{kuning}Tidak ada pengguna yang memesan kamar.{putih}\n")

        return

    else :

        if cek_tabel == 1:

            tabel(pesanan,1)

        print(f""{biru}Menerima penghuni
{putih}1. Konfirmasi
2. Batalkan
3. {merah}kembali{putih}""")

        pilihan = input("\nMasukan Pilihan :")

        if pilihan == "1":

            konfirmasi_atau_batalan(1)

        elif pilihan == "2":

```

```

        konfirmasi_atau_batalikan(0)

    elif pilihan == "3":

        print("\033c", end="")

        return

    else:

        print("\033c", end="")

        tabel(pesanan,1)

        print(f"{kuning}Pilihan tidak valid{putih}\n")

        cek_tabel = 0

# jika indeks = 1 maka konfirmasi
def konfirmasi_atau_batalikan(indeks):

    global cek_tabel

    ketemu_nama = False

    id_pengguna = int(input("Masukan id Pengguna : "))

    if id_pengguna > len(id_nama_list) or id_pengguna <= 0 :

        print("\033c", end="")

        tabel(pesanan,1)

        print(f"{kuning}ID tidak ditemukan{putih}\n")

        cek_tabel = 0

    else:

        for i in range(len(pesanan)):

            for indeks_pesanan in range(0,len(pesanan[i])):

                if id_nama_list[id_pengguna - 1] == pesanan[i][indeks_pesanan][0]:

```

```

        if indeks == 1:

            penghuni[i].append(pesanan[i][indeks_pesanan])

            penghuni[i][(len(penghuni[i])-1)][2] = "Dikonfirmasi"

            print("\033c", end="")

            nama_tampungan = pesanan[i][indeks_pesanan][0]

            del pesanan[i][indeks_pesanan]

            del id_nama_list[id_pengguna -1]

            cek_tabel = 0

            ketemu_nama = True

            if len(pesanan[i]) > 0:

                tabel(pesanan,1)

        if indeks == 1:

            print(f"{hijau}{nama_tampungan} Berhasil ditambahkan menjadi penghuni{putih}\n")

        else:

            print(f"{hijau}{nama_tampungan} Berhasil Dibatalkan{putih}\n")

        break

    if ketemu_nama == True:

        break

def tabel(penghuni_atau_pesanan,cek):

    global panjang_maks_id, panjang_maks_nama, panjang_maks_fasilitas, panjang_maks_status, tabels, id_nama_list

    panjang_maks_id = 2

    panjang_maks_nama = 4

```

```

panjang_maks_fasilitas = 9

panjang_maks_status = 6


panjang_id = 0

for i in range(len(penghuni_atau_pesanan)):
    for indeks_pesanan in range(len(penghuni_atau_pesanan[i])):
        panjang_nama = len(penghuni_atau_pesanan[i][indeks_pesanan][0])
        panjang_fasilitas = len(penghuni_atau_pesanan[i][indeks_pesanan][1])
        panjang_status = len(penghuni_atau_pesanan[i][indeks_pesanan][2])

        if panjang_nama > panjang_maks_nama:
            panjang_maks_nama = panjang_nama

        if panjang_fasilitas > panjang_maks_fasilitas:
            panjang_maks_fasilitas = panjang_fasilitas

        if panjang_status > panjang_maks_status:
            panjang_maks_status = panjang_status


    panjang_id += len(penghuni[i])

if panjang_id > panjang_maks_id:
    panjang_maks_id = len(str(panjang_id))

if panjang_maks_id % 2 != 0:
    panjang_maks_id += 1

if panjang_maks_nama % 2 != 0:
    panjang_maks_nama += 1

if panjang_maks_fasilitas % 2 != 0:

```

```

        panjang_maks_fasilitas += 1

    if panjang_maks_status % 2 != 0:

        panjang_maks_status += 1

    # mengecek setiap kamar

    tabels = f"{biru}| {putih}{ID'.center(panjang_maks_id)} {biru}|
{putih}{Nama'.center(panjang_maks_nama)} {biru}|
{putih}{Fasilitas'.center(panjang_maks_fasilitas)} {biru}|{putih}
{'Status'.center(panjang_maks_status)} {biru}|{putih}"

    print(f"{biru}=" * (len(tabels) - 105 ))

    print(tabels)

    print(f"{biru}=" * (len(tabels) - 105))

    id = 1

    id_nama_list = []

    for x in range(len(penghuni_atau_pesanan)):

        for indeks_pesanan in range(len(penghuni_atau_pesanan[x])):

            if not penghuni_atau_pesanan[x]:

                continue

            else:

                nama = penghuni_atau_pesanan[x][indeks_pesanan][0].ljust(panjang_maks_nama)

                fasilitas =
penghuni_atau_pesanan[x][indeks_pesanan][1].ljust(panjang_maks_fasilitas)

                status =
penghuni_atau_pesanan[x][indeks_pesanan][2].ljust(panjang_maks_status)

                #masuk ke penghuni

                if cek == 0:

                    print(f"{biru}| {putih}{str(id).ljust(panjang_maks_id)} {biru}|

```



```

{putih}

{putih}      ||  || /--\  ||  ||      ||  || /--\  ||  ||      ||  || /--\
||  ||      {putih}

{putih}      ||[] || | .| ||[] ||      ||[] || | .| ||[] ||      ||[] ||{putih}

{putih}      ()||__||_||_||_||()  ()||__||_||_||_||()  ()||__||_||_||_||()
{putih}

{putih}      ( )|-|-|-|===|-|-|-|( ) ( )|-|-|-|===|-|-|-|( ) ( )|-|-|-|===|-|-|-|( )
{putih}

{hijau}      ^^^^^^^^^{putih}==={hijau}^^^^^^^^^^^^ ^^^^^^^^^{putih}==={hijau}^^^^^^^^^^^^
^^^^^^^^^^^^{putih}==={hijau}^^^^^^^^^^^^ {putih}""")

menu_login()

```

## **BAB IV**

### **PENUTUP**

#### **4.1 Kesimpulan**

Berdasarkan hasil dari program manajemen kos-kosan berbasis CRUD menggunakan Python yang telah kami kembangkan, dapat disimpulkan bahwa:\

1. Merancang sistem CRUD untuk pengelolaan kos lumayan efisien,jika disimpan dalam database
2. Program memberi kemudahan karna tidak perlu buku fisik untuk menyimpan data kos
3. Sistem interface yang simple membantu dalam memahami menu menu

#### **4.2 Saran**

1. Mengintegrasikan program dengan sistem penyimpanan berbasis database, seperti SQLite atau MySQL, agar data dapat disimpan dengan lebih aman dan mudah diakses.
2. Mengembangkan fitur lebih jauh dengan contoh yaitu fitur pencarian spesifik untuk suatu hal
3. Memastikan dokumentasi tiap kode program agar bisa dikembangkan dimasa depan



## DAFTAR PUSTAKA

Referensi buku ataupun website yang dibuat panduan selama pengerjaan proyek dan laporan.

*ASCII Art Archive*. (n.d.). ASCII Art Archive. <https://www.asciart.eu/>

*Newest “Python” questions*. (n.d.). Stack Overflow.

<https://stackoverflow.com/questions/tagged/python>

*W3Schools.com*. (n.d.). <https://www.w3schools.com/python/>

*Belajar Python: Memahami Fungsi dan Prosedur pada Python*. (2017, October 31). Petani



Kode. <https://www.petanikode.com/python-fungsi/>



Mahfud Aska. (2021, February 7). *Aplikasi CRUD Sederhana di Python* [Video]. YouTube.

<https://www.youtube.com/watch?v=nJTeGngAJ4g>

## LAMPIRAN

Lampirkan kartu konsultasi disini dan berikan menamaan gambar juga kemudian masukkan di daftar gambar.

Tanggal Konsultasi : 2-11-2024, Sabtu	
Uraian / Pembahasan : <ul style="list-style-type: none"><li>- detail untuk fungsi flow chart</li><li>- ketuk fungsi yang ada garis kiri &amp; kanan</li><li>- output bagian menu lain in put</li></ul>	
Asisten Lab  Nama: Muh. Shandy A	Ketua Kelompok  Nama: Raffi Zidane W.

Tanggal Konsultasi : 13-11-2024, Rabu	
Uraian / Pembahasan : <ul style="list-style-type: none"><li>- Perbaiki:<ul style="list-style-type: none"><li>* Skema di input</li><li>* (-1) angka negatif</li></ul></li><li>- tambahkan update</li><li>- tampilan menu</li><li>- sistem rating</li><li>- hapus data</li></ul>	
Asisten Lab  Nama: Muh. Shandy A	Ketua Kelompok  Nama: Raffi Zidane W.

Gambar 3.1 lampiran konsultasi