**LAPORAN PROYEK AKHIR PRAKTIKUM**

**MATA KULIAH ALGORITMA DAN PEMROGRAMAN DASAR**



**Manejemen kamar kos-kosan**

**Oleh:**

**Kelompok 1 A2**

|  |  |
| --- | --- |
| **RAFFI ZIDANE WAHYUDIN** | **2409106037** |
| **MUHAMMAD RIZAL ALFATH** | **2409106039** |
| **MUHAMMAD RASYID** | **2409106042** |
| **RANGGA LAWE** | **2409106043** |

**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 menejemen 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 praktikun algoritma pemprograman dasar

Selama penyusunan laporan dan pembuatan program, kami menghadapi beberapa kendala, seperti kurang nya pemahaman akan materi di luar modul praktikum dan kurangnnya 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

# TAKARIR

Daftar padanan kata bahasa asing dalam bahasa Indonesia yang digunakan adalah

sebagai berikut:

*Database* Basis Data

*Managemen* Mengatur

*Input* Memasukkan

*Python* bahasa program

*User* penguna

*Password* kata sandi

*Akun* tempat penyimpanan

*Login* proses masuk akun

*Rating* penilaian

*CRUD* program

*If/elif/else* fungsi percabangan

*Cctv* kamera pengawas

*Error* kerusakan pada fungsi

# DAFTAR ISI

[KATA PENGANTAR ii](#_Toc195)

[TAKARIR iii](#_Toc12505)

[DAFTAR ISI iv](#_Toc24055)

[DAFTAR GAMBAR vi](#_Toc23854)

[BAB I PENDAHULUAN 1](#_Toc20423)

[1.1 Latar Belakang 1](#_Toc23407)

[1.2 Kebutuhan Fungsional 1](#_Toc12688)

[1.3 Rumusan Masalah 2](#_Toc6544)

[1.4 Batasan Masalah 2](#_Toc24918)

[1.5 Tujuan 2](#_Toc3551)

[BAB II PERANCANGAN 3](#_Toc13093)

[2.1 Analisis Program 3](#_Toc24421)

[2.2 Flowchart 3](#_Toc28768)

[2.3 Konsep/Materi Praktikum yang dipakai 4](#_Toc32365)

[BAB III HASIL DAN PEMBAHASAN 8](#_Toc19128)

[3.1 Tampilan Program 8](#_Toc5683)

[1. Tampilan Menu 8](#_Toc10140)

[2. Menu login 9](#_Toc18690)

[3. Menu admin (pemilik kos) 9](#_Toc4191)

[4. Menu manage kamar 9](#_Toc29259)

[5. Menu manage penghuni 10](#_Toc13423)

[6. Menu lihat pemesanan 11](#_Toc18047)

[7. Menu lihat rating 12](#_Toc5704)

[8. Menu manage account 12](#_Toc9907)

[9. Menu user(penghuni kos) 12](#_Toc271)

[3.2 Source Code 13](#_Toc28241)

[BAB IV PENUTUP 50](#_Toc5114)

[4.1 Kesimpulan 50](#_Toc29540)

[4.2 Saran 50](#_Toc23016)

[DAFTAR PUSTAKA 51](#_Toc30278)

[LAMPIRAN 52](#_Toc22237)

# DAFTAR GAMBAR

[Gambar 1.1 flowchart. 4](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.1 awal program. 8](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.2tampilan menu login . 9](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.3tampilan menu admin. 9](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.4manage kamar dan lihat kamar. 9](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.5tambah kamar. 9](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.6list kamar setelah ditambah. 10](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.7 hapus kamar. 10](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.8 list kamar setelah dihapus. 10](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.9 update kamar. 10](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.10 list kamar setelah diupdate . 10](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.11 manage dan lihat kamar . 11](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.12 tambah penghuni 11](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.13 hapus penghuni. 11](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.14 .jika tidak ada pemesanan 11](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.15 ada pemesanan. 12](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.16 .rating kos-kosan 12](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.17 .untuk manage akun user 12](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.18 untuk akun user dan lihat kamar. 13](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.19 .user rating 13](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.20 .menu akun user yang terkonfirmasi 13](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 2.21 .akun user yang belum terkonfirmasi 13](#_Toc181627902)**[.](#_Toc181627902)**

[Gambar 3.1 lampiran konsul. 52](#_Toc181627902)**[.](#_Toc181627902)**

# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Berkembangnya zaman membuat pengelolaan data yang efektif dan efisien adalah kebutuhan utama dalam segala bidang, termasuk dalam menejemen kos kosan,banyaknya pengelola kos kosan yang mengunakan pelelolaan data secara manual untuk mencatat informasi dan data penting lainnya ,yang mana hal itu sangat tidak efisien dan tidak efektif, oleh karna itu kami membuat program untuk membantu dalam mengelola data kos kosan agar lebih efisien dan efektif mengunakan python yang dirancang untuk kebutuhan menejemen 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 dana password
3. Sistem mampu menampilkan data kos kos an ke user untuk melihat kamar kos kosong
4. Sistem mampu melakukan proses pemberian rating dari user untuk rating kos kosan
5. Sistem mampun nampilkan user untuk dilihat di menu admin
6. Sistem mampu memberi admin untuk memanage kamar
7. Sistem mampu memberi admin untuk memanage penghuni
8. Sistem mampu melihatkan rating untuk admin
9. Sistem mampu melihatkan 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 menejemen 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 menejemen 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 aadmin terdapat menu untuk manage kamar,manage penghuni,lihat rating dan manage akun

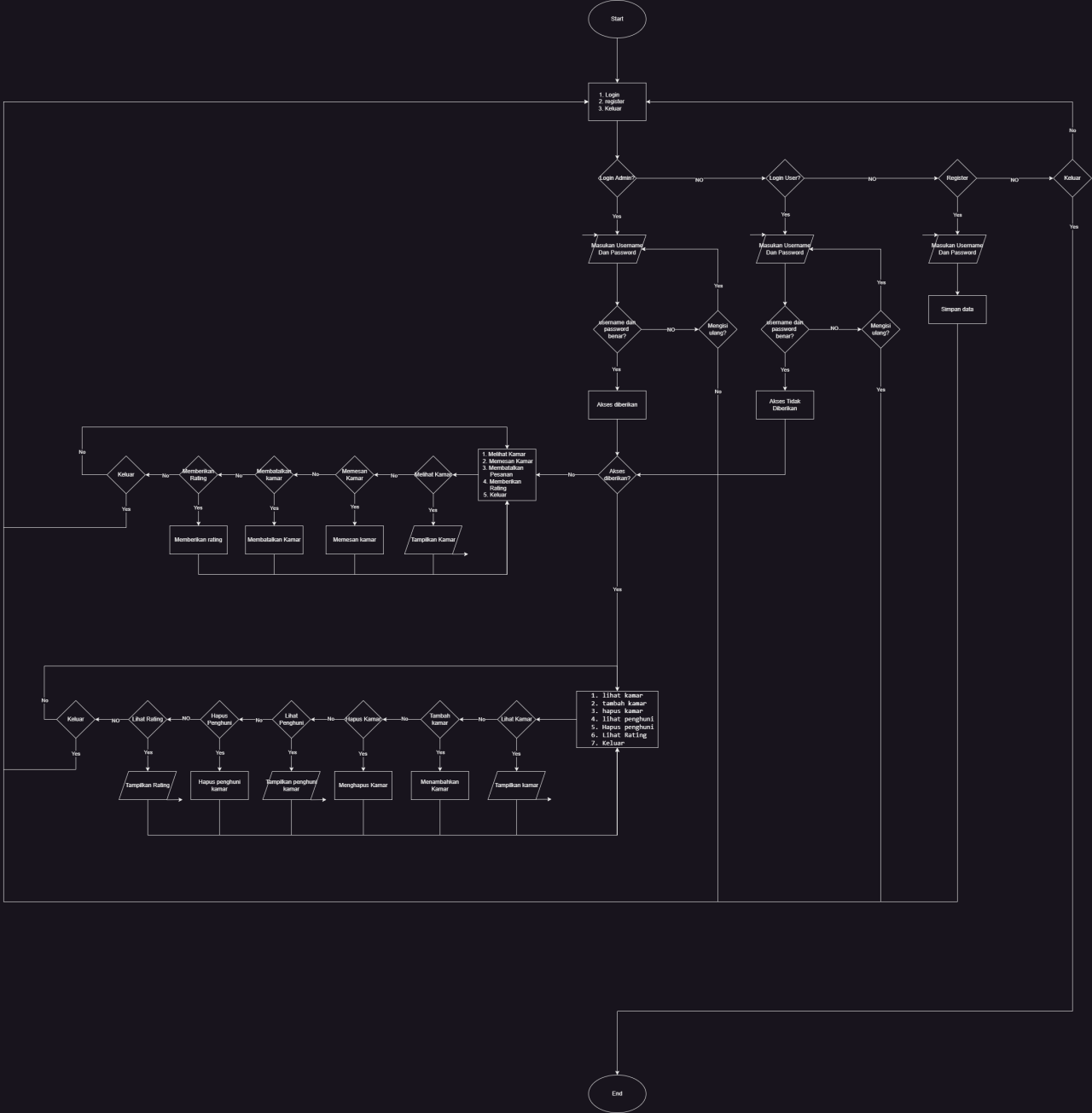
Di menu manage sendiri,seperti namanya,untuk mengatur dari akun dan kamar

Lalu untuk login user,mereka harus registrasi terlebih dahulu lalu login mengunakan 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 flowchar dibawah ialah,dimulai dengan memilih apakah login admin,user,registrasi atau keluar jika memilih login admin makan akan masuk ke input username dan passwod,jika username dan password benar maka akan dilanjutkan ke fungsis fungsi admin,begitu juga jika login user,jika username dan password benar maka akan diaarahkan ke fungsi fungsi penghuni,jika regis maka akan diusuh input untuk username dan password



Gambar 1.1 flowchart

## 2.3 Konsep/Materi Praktikum yang dipakai

Konsep yang kami gunakan dalam pembuatan laporan ini sesuai dengan

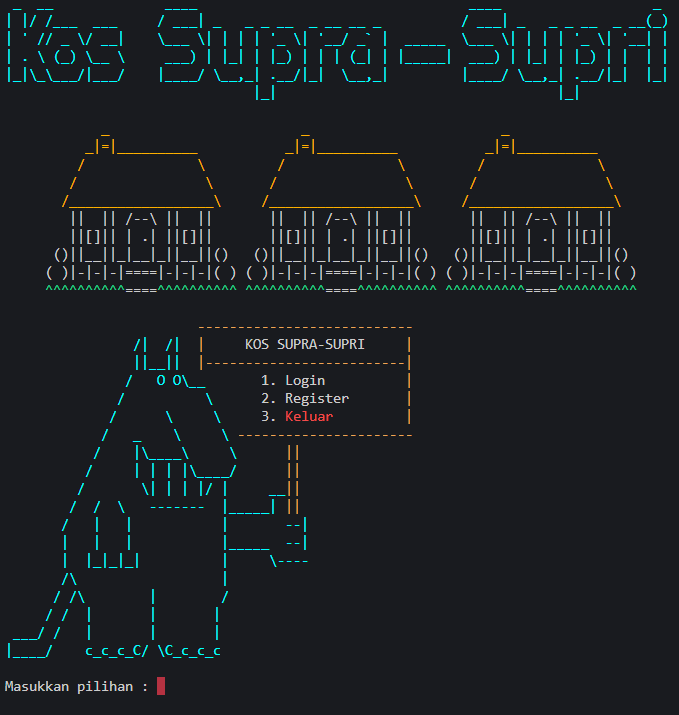
persyaratan yang terdapat pada modul diantaranya:

1. Fungsi dasar: fungsi yang kami gunakan adalah fungsi print. Fungsi print pada program merupakan fungsi yang umum dipakai untuk menampilakan 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 penguna 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 dibilang 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

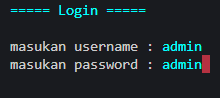
## 3.1 Tampilan Program

### Tampilan Menu

****Menu dimana program dimulai

Gambar 2.1 awal program.

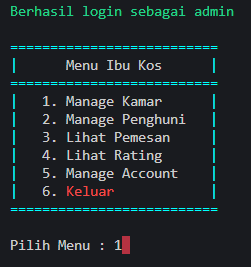
### Menu login



Gambar 2.2 Tampilan menu logim

Menu dimana admin atau user memasukkan username dan password

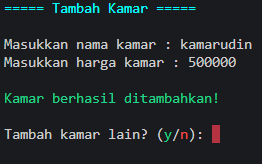
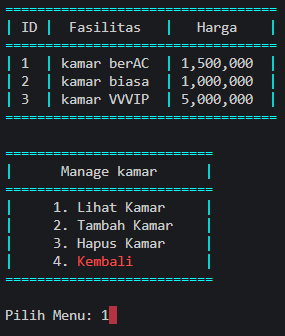
### Menu admin (pemilik kos)



Gambar 2.3 tampilan menu admin

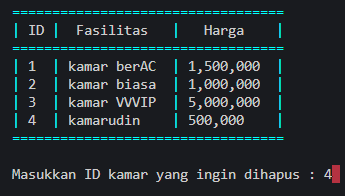
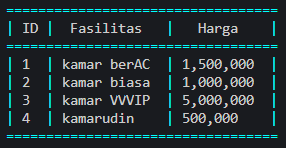
Menu untuk admin (pemilik kos) untuk mengelola kos

### Menu manage kamar



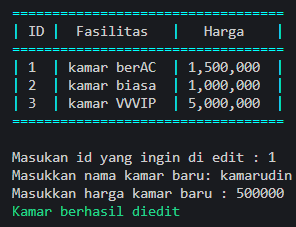
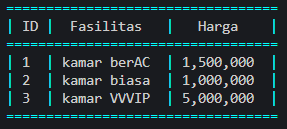
Gambar 2.4 manage kamar dan lihat kamar

Gambar 2.5 tambah kamar



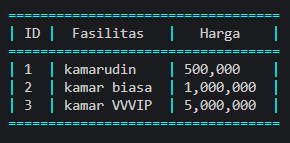
Gambar 2.7 hapus kamar

Gambar 2.6 list kamar setelah ditambah



Gambar 2.9 update kamar

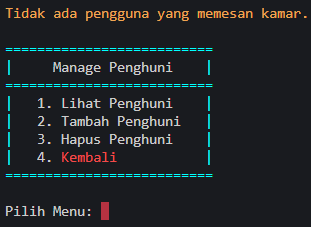
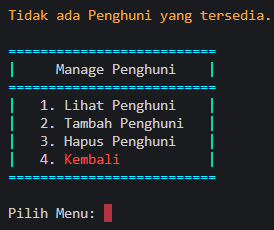
Gambar 2.8 list kamar setelah dihapus



Gambar 2.10 list kamar setelah diupdate

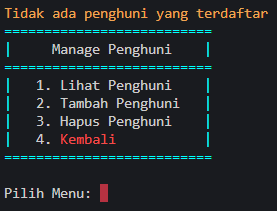
Menu admin untuk manage kamar dengan funsi untuk melihat,menambah,mengupdate dan menghapus kamar

### Menu manage penghuni



Gambar 2.12 tambah penghuni

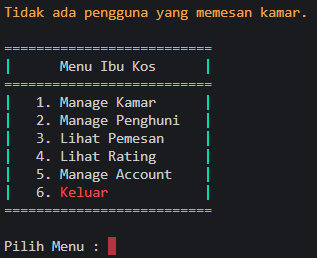
Gambar 2.11 manage dan lihat pengghuni



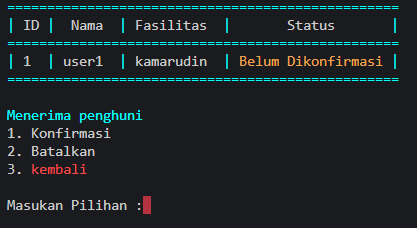
Gambar 2.13 hapus penghuni

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

### Menu lihat pemesanan



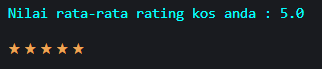
Gambar 2.14 jika tidak ada pemesanan



Gambar 2.15 ada pemesanan

Menu untuk admin melihat pemesanan dan konfirmasi pemesanan kamar kos

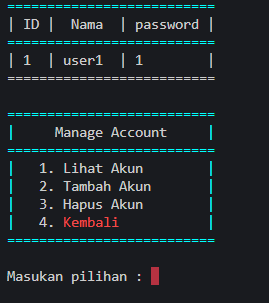
### Menu lihat rating



Gambar 2.16 rating kos-kosan

Menu untuk admin mengetahui rating dari kos-kosan mereka

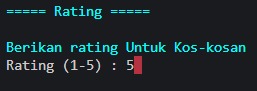
### Menu manage account



Gambar 2.17 untuk manage akun user

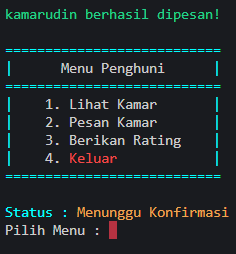
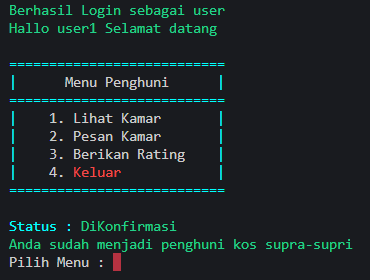
Menu untuk admin melihat user dan menambah serta menghapus user

### Menu user(penghuni kos)



Gambar 2.19 user rating

Gambar 2.18untuk akun user dan lihat kamar



Gambar 2.20 menu akun user yang terkonfirmasi

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 VVVIP', '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("\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}                 /   O O\\\_\_       {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}Pesan 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 Plihan : "))      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\_batalkan(1)            elif pilihan == "2":              konfirmasi\_atau\_batalkan(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\_batalkan(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}{nama} {biru}| {putih}{fasilitas} {biru}| {hijau}{status} {biru}|{putih}")    *#masuk ke pesanan*                  elif cek == 1 :                      print(f"{biru}| {putih}{str(id).ljust(panjang\_maks\_id)} {biru}| {putih}{nama} {biru}|{putih} {fasilitas} {biru}| {kuning}{status} {biru}|{putih}")                    id += 1                  id\_nama\_list.append(penghuni\_atau\_pesanan[x][indeks\_pesanan][0])      print(f"{biru}=" \* (len(tabels) - 105), f"{putih}\n")    *# Memulai program dengan menu login*  print(f"""{biru} \_  \_\_              \_\_\_\_                                  \_\_\_\_                   \_  | |/ /\_\_\_  \_\_\_     / \_\_\_| \_   \_ \_ \_\_  \_ \_\_ \_\_ \_          / \_\_\_| \_   \_ \_ \_\_  \_ \_\_(\_)  | ' // \_ \\/ \_\_|    \\\_\_\_ \\| | | | '\_ \\| '\_\_/ \_` |  \_\_\_\_\_  \\\_\_\_ \\| | | | '\_ \\| '\_\_| |  | . \\ (\_) \\\_\_ \\     \_\_\_) | |\_| | |\_) | | | (\_| | |\_\_\_\_\_|  \_\_\_) | |\_| | |\_) | |  | |  |\_|\\\_\\\_\_\_/|\_\_\_/    |\_\_\_\_/ \\\_\_,\_| .\_\_/|\_|  \\\_\_,\_|         |\_\_\_\_/ \\\_\_,\_| .\_\_/|\_|  |\_|                                 |\_|                                   |\_|           """)    print(f"""  {orange}            \_                        \_                        \_{putih}  {orange}          \_|=|\_\_\_\_\_\_\_\_\_\_           \_|=|\_\_\_\_\_\_\_\_\_\_           \_|=|\_\_\_\_\_\_\_\_\_\_{putih}  {orange}         /              \\         /              \\         /              \\ {putih}  {orange}        /                \\       /                \\       /                \\ {putih}  {orange}       /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\     /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\     /\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\ {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 pengelelolaan 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 projek dan laporan.

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

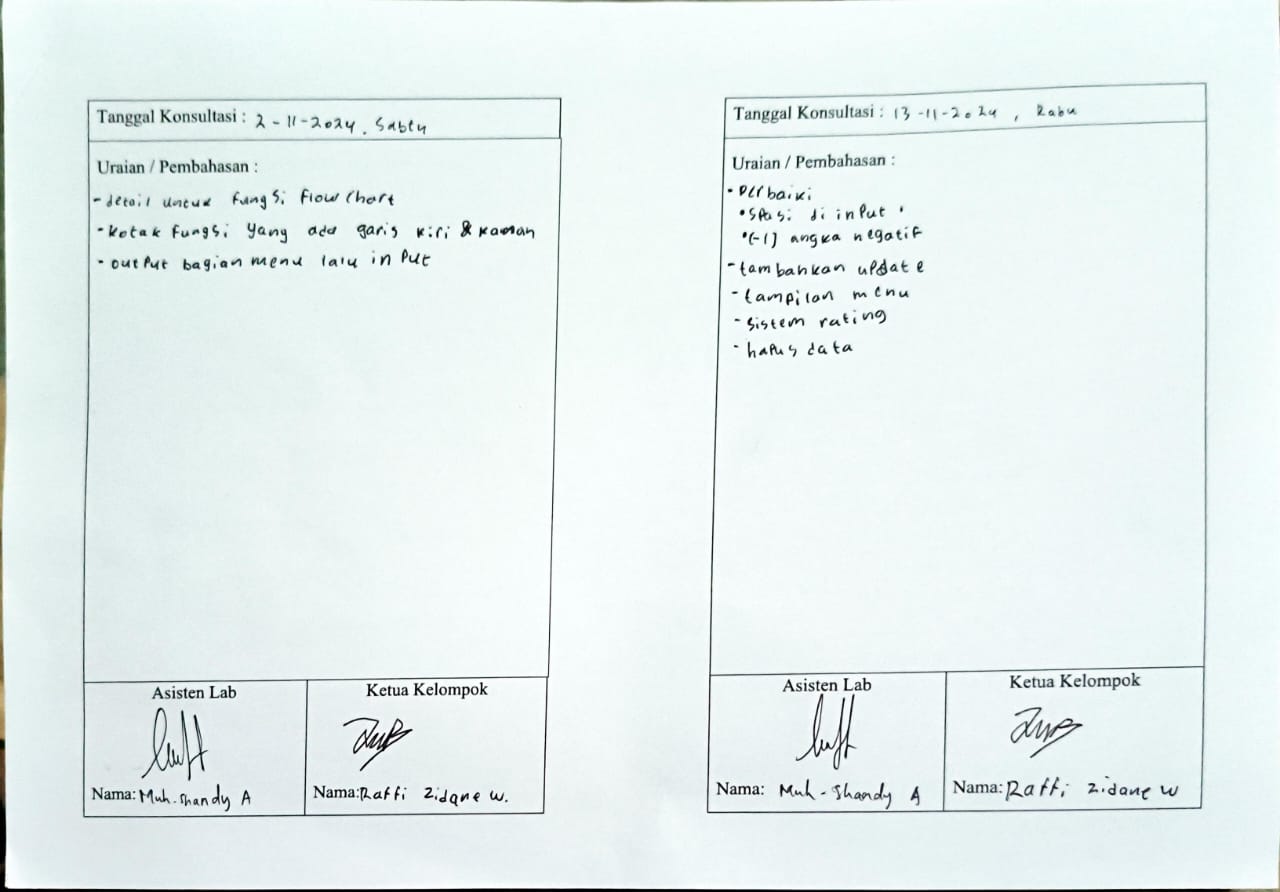
*Newest “Python” questions*. (n.d.). Stack Overflow. <https://stackoverflow.com/questions/tagged/python>

*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 konsul disini dan berikan menamaan gambar juga kemudian masukkan di daftar gambar.

Gambar 3.1 lampiran konsul