



"PROGRAM PENCARIAN KOST"

PROJECT TEAM PRAKTIKUM PROGRAMA KOMPUTER
2023

KELOMPOK 09 :

FERY KURNIAWAN	(I0322044)
FIKI ANREANO SIAGIAN	(I0322045)
FITRI NIDHOMUN NI'MAH	(I0322046)
IVAN PRAMUDYA ARTURA	(I0322055)
KHANSA KHALIDA KENNEDY	(I0322067)



BAB I

DESKRIPSI MASALAH

Bab ini menjelaskan mengenai latar belakang masalah, rumusan masalah, dan tujuan pembuatan program pencarian kost.

1.1 Latar Belakang

Subbab ini menjelaskan mengenai latar belakang pembuatan program pencarian kost

Sekarang ini kehidupan manusia tidak lepas dari kemajuan pendidikan dan teknologi mengingat zaman sudah berkembang pesat. Maka dari itu seiring perkembangan zaman kebutuhan akan pendidikan bagi para generasi penerus bangsa terus meningkat. Setiap tahunnya banyak siswa/siswi SMA atau SMK yang melanjutkan pendidikan ke jenjang yang lebih tinggi atau yang biasa dikenal dengan universitas atau perguruan tinggi. Surakarta atau yang biasa disebut Solo menjadi salah satu kota incaran para siswa/siswi SMA dan SMK dalam melanjutkan pendidikan mereka baik di perguruan tinggi negeri atau swasta. Dalam menempuh pendidikan perguruan tinggi banyak siswa/siswi yang berasal dari luar kota. Jauhnya letak rumah dengan lokasi perguruan tinggi mengharuskan para mahasiswa/mahasiswi untuk mencari kost sebagai tempat tinggal sementara mereka selama menjadi mahasiswa/mahasiswi.

Luasnya Kota Solo menyebabkan sulitnya mencari informasi tempat tinggal sementara bagi para mahasiswa/mahasiswi. Pada saat ini para mahasiswa mencari tempat kost dengan cara mengunjungi satu tempat ke tempat lain untuk menemukan tempat kost yang diinginkan. Namun cara ini kurang efektif bagi para mahasiswa yang tinggal jauh dari perguruan tinggi tempat mereka menimba ilmu. Adanya perkembangan teknologi diharapkan mampu mempermudah para mahasiswa dalam mencari kost dengan terpercaya tanpa datang langsung ke lokasi. Program ini dapat menjadi sarana penyediaan informasi terkait pencarian kost yang efektif bagi mahasiswa secara mudah, cepat, dan terpercaya.

1.2 Rumusan Masalah

Subbab ini menjelaskan mengenai rumusan masalah yang terjadi.

- Maraknya pencarian kost bagi mahasiswa di Kota Solo.
- Minimnya informasi mengenai detail kost bagi para calon mahasiswa.

1.3 Tujuan

Subbab ini menjelaskan tujuan pembuatan program.

- Memudahkan para mahasiswa/mahasiswi dalam mencari kost di Kota Solo.
- Memudahkan para mahasiswa/mahasiswi dalam memfilter secara detail kamar kost yang sesuai dengan *budget*.

BAB II FLOWCHART

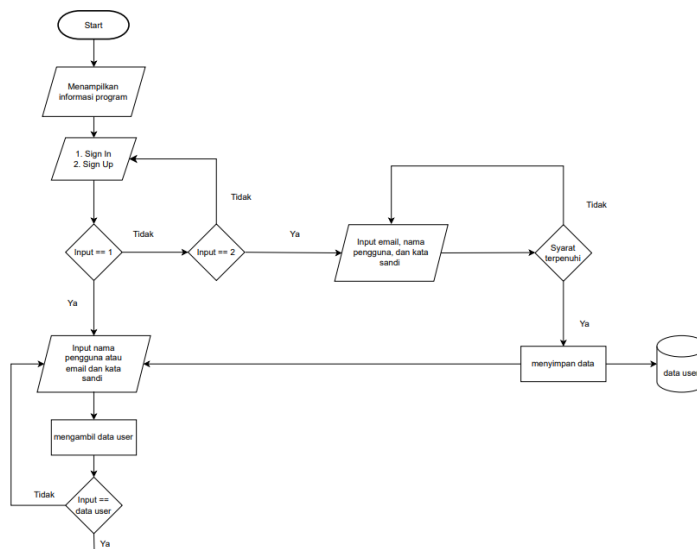
Bab ini menjelaskan mengenai Flowchart.

2.1 Flowchart Program

Subbab ini menjelaskan mengenai *flowchart* program pencarian kost yang terdiri dari tahap *login*, pencarian kost, dan pembayaran.

2.1.1 Flowchart Login

Subbab ini menjelaskan mengenai *flowchart* login.



Gambar 2.1 Flowchart Program Sign In Dan Sign Up

Menu awal yang akan ditampilkan oleh program yaitu:

1. Sign In
2. Sign Up

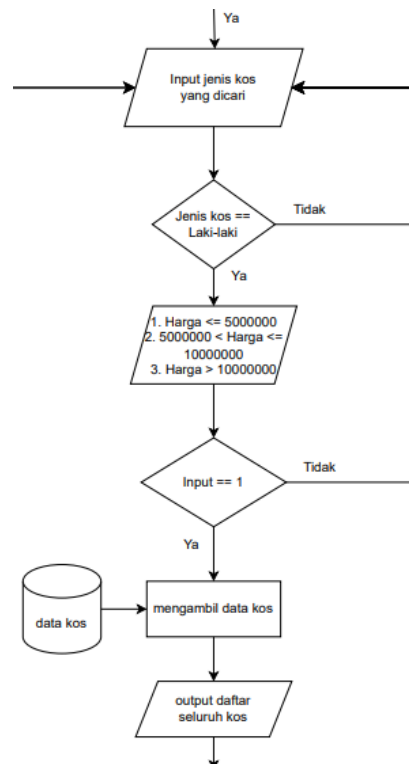
Program akan memberi petunjuk kepada *user* berupa perintah untuk memilih menu *Sign Up* jika belum memiliki akun atau memilih *Sign In* jika sudah memiliki akun. Jika *user* memilih menu *Sign In*, maka *user* diminta untuk memasukkan *username* atau email, dan *password* sebagai validasi bahwa data *user* sebelumnya sudah tersimpan pada data csv. Jika *user* merupakan pengguna baru, maka *user* harus memilih menu *Sign Up* untuk membuat sebuah akun pencarian kost dengan memasukkan *email*, *username*, dan *password* yang nantinya secara otomatis akan tersimpan pada data csv.

Setelah membuat akun baru pada menu *Sign Up*, *user* akan diminta untuk masuk ke menu *Sign In* dan memasukkan kembali *username* atau email yang sudah didaftarkan tadi.

Setelah *user* berhasil *login* dengan akun lama ataupun baru, program akan memberikan *output* berupa ucapan selamat datang sesuai *username* yang dimasukkan *user*. Namun apabila *username*, email, dan *password* yang dimasukkan salah, maka *user* akan diminta untuk memasukkan ulang hingga data tersebut benar sesuai dengan apa yang tersimpan pada *database*.

2.1.2 Flowchart Pencarian kost

Subbab ini menjelaskan mengenai *flowchart* pencarian kost.



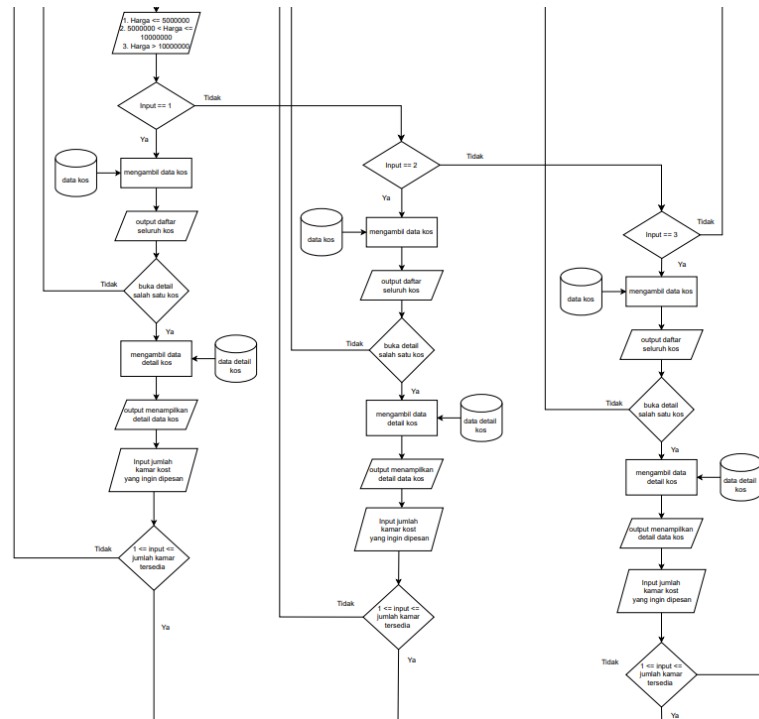
Gambar 2.2 Flowchart Program Pencarian kost

Setelah *user* berhasil *login* dengan *username*, email, dan *password* yang benar, program akan mengarahkan *user* untuk memilih jenis kost, yaitu kost perempuan atau laki-laki. Setelah memilih jenis kost, program akan menampilkan pilihan 3 *range* harga yang sudah difilter sesuai kebutuhan *user*, yaitu:

1. Harga \leq Rp 5.000.000,00
2. Rp 5.000.000,00 < Harga \leq Rp 10.000.000,00

3. Harga > Rp 10.000.000,00

User diminta untuk memilih salah satu dari ketiga *range* harga yang ditampilkan. Setelah memilih, program akan memberikan *output* berupa *list* daftar kost sesuai harga yang dipilih oleh *user*.



Gambar 2.3 Flowchart Program Pencarian Kost

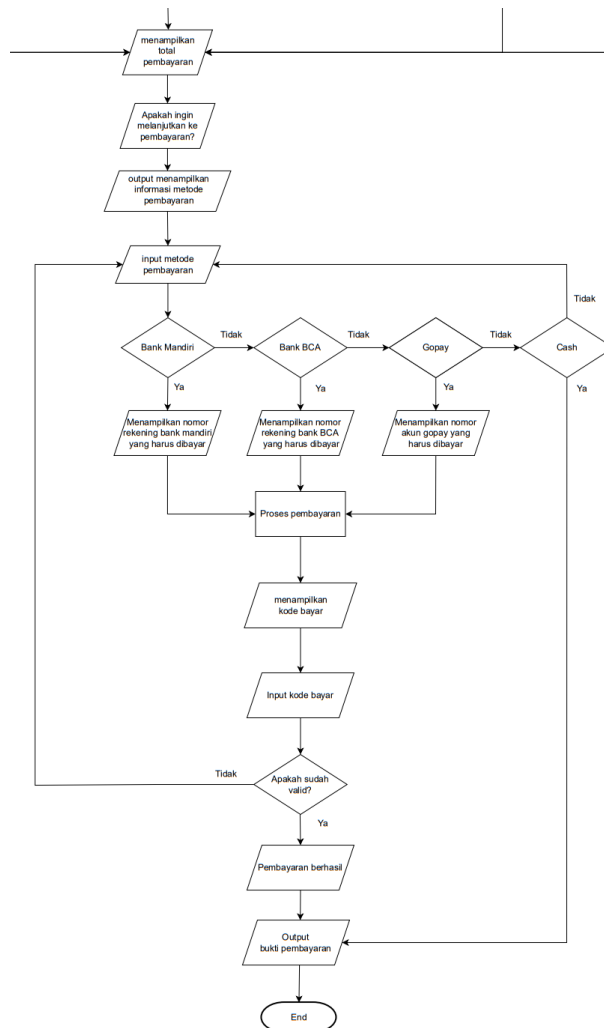
Setelah muncul daftar kost sesuai *range* harga, program akan memberikan pilihan berupa (ya/tidak) pada *user* untuk membuka detail salah satu kost. Jika *user* memilih ‘tidak’, maka program akan otomatis mengarahkan *user* ke menu awal pemilihan jenis kost. Jika *user* memilih ‘ya’, maka program akan meminta *user* untuk memasukkan nomor kost yang ingin dibuka detailnya. Setelah itu, program akan menampilkan *output* berupa detail kost yang meliputi nama kost, alamat, ukuran, harga sewa/tahun, fasilitas, narahubung, dan juga jumlah kamar yang masih tersedia sesuai nomor kost yang dimasukkan *user*.

Setelah itu, *user* akan diminta memasukkan jumlah kamar kost yang ingin disewa. Apabila jumlah kamar kost yang dimasukkan *user* ternyata melebihi jumlah kapasitas kamar yang tersedia, maka program akan memunculkan keterangan bahwa jumlah kamar yang tersedia tidak cukup, dan secara otomatis program akan mengarahkan *user* ke menu awal pemilihan

jenis kost. Namun, jika jumlah kamar yang dimasukkan *user* ternyata kurang dari atau sama dengan (\leq), atau dengan kata lain memenuhi jumlah kamar yang tersedia, maka program secara otomatis akan mengkalkulasi dan menampilkan total harga yang harus dibayarkan *user* jika ingin menyewa kost tersebut.

2.1.3 Flowchart Pembayaran

Subbab ini menjelaskan mengenai *flowchart* pembayaran.



Gambar 2.4 Flowchart Program Pembayaran

Jika jumlah kamar yang dimasukkan *user* ternyata kurang dari atau sama dengan (\leq) jumlah kamar yang tersedia, maka program secara otomatis akan mengkalkulasi dan menampilkan total harga yang harus dibayarkan *user* jika ingin menyewa kost tersebut.

Setelah itu program akan memberikan pilihan berupa (ya/tidak) kepada *user* untuk melanjutkan ke menu pembayaran. Jika *user* memilih ‘tidak’, maka program akan mengarahkan *user* ke menu awal pemilihan jenis kost. Namun jika *user* memilih ‘ya’, program akan menampilkan beberapa metode pembayaran yang harus dipilih *user* untuk melanjutkan transaksi, yang meliputi:

1. Bank mandiri
2. Bank BCA
3. Gopay
4. Cash

Jika *user* memilih metode pembayaran *cash*, maka program akan menampilkan total harga yang harus dibayar dan *user* dapat menghubungi pemilik kost melalui narahubung yang sudah tertera di detail kost untuk melanjutkan proses transaksi secara langsung. Sedangkan jika *user* memilih metode pembayaran melalui Bank Mandiri, Bank BCA, ataupun Gopay, *user* akan diberikan nomor rekening dan kode bayar untuk memvalidasi pembayaran. Setelah diberikan kode bayar, *user* harus memasukkan kode pada metode bayar yang dipilih. Jika kode bayar yang dimasukkan salah, maka *user* akan diminta untuk memilih kembali metode pembayaran. Namun jika kode bayar yang dimasukkan benar, program akan memvalidasi dan memberi tahu *user* bahwa proses pembayaran berhasil. Setelah proses pembayaran berhasil, program akan menampilkan *output* berupa struk/bukti pembayaran.

BAB III KODE PROGRAM

3.1 Melakukan *Import* Modul yang akan digunakan pada Program

```
import csv
import random
```

Gambar 3.1 Kode Program *Import* Modul

Kode program “import csv” digunakan untuk memanggil modul csv sehingga pengguna dapat dengan mudah membaca dan menulis file csv dalam Python. Sedangkan kode program “import random” digunakan pada fungsi kode bayar untuk memanggil angka secara acak.

3.2 Mendefinisikan Fungsi Menu Awal

```
#menu awal
def menu():
    print("=====")
    print("=                      SELAMAT DATANG DI PROGRAM PENCARIAN KOST!                      =")
    print("=====")
    menu()
```

Gambar 3.2 Fungsi Menu Awal

Fungsi ini digunakan untuk menampilkan tampilan awal program ketika program akan dijalankan.

3.3 Mendefinisikan Fungsi Membaca dan Menulis Data

```
def read_csv(datauser):
    data = []
    with open(datauser, 'r') as file:
        reader = csv.reader(file)
        for row in reader:
            data.append(row)
    return data

def write_csv(datauser, data):
    with open(datauser, 'w', newline='') as file:
        writer = csv.writer(file)
        writer.writerows(data)
```

Gambar 3.3 Fungsi Membaca dan Menulis Data

Fungsi ini digunakan untuk membaca dan menulis/menambah data pada *database*.

3.4 Mendefinisikan Fungsi *Sign In*

```
#sign in
def signin():
    print("\nSign In")
    username = input("Masukkan email atau username akun Anda: ")
    password = input("Masukkan password akun Anda : ")
    data = read_csv('datauser.csv')
    for row in data:
        if (row[0] == username or row[1] == username) and row[2] == password:
            print("===== \n")
            print("INFORMASI PENGGUNA")
            print("Selamat datang,", row[1])
            print("Sign In telah berhasil.")
            return
    print("Username atau password Anda tidak valid.")
    signin()
```

Gambar 3.4 Fungsi *Sign In*

Fungsi ini diperuntukkan bagi pengguna yang sudah memiliki akun. Pada fungsi ini terdapat fungsi **read_csv(datauser)** yang digunakan untuk membaca data yang sudah tersimpan pada *database*, sehingga pengguna tidak perlu membuat akun baru dan cukup dengan memasukkan *username* atau *email* dan *password* dari akun yang sudah didaftarkan sebelumnya.

3.5 Mendefinisikan Fungsi *Sign Up*

```
#sign up
def signup():
    print("\nSign Up")
    email = input("Masukkan email : ")
    username = input("Masukkan username : ")
    password = input("Masukkan password : ")
    data = read_csv('datauser.csv')
    for row in data:
        if row[0] == email or row[1] == username:
            print("Email atau username telah digunakan.\n")
            signup()
            return
    data.append([email, username, password])
    write_csv('datauser.csv', data)
    print("Akun berhasil dibuat, silakan lakukan Sign In.")
    signin()
```

Gambar 3.5 Fungsi *Sign Up*

Fungsi ini diperuntukkan bagi pengguna yang belum memiliki akun untuk membuat akun baru dengan memasukkan *email*, *username*, dan *password* yang nantinya akan otomatis tersimpan pada *database*. Fungsi ini juga dilengkapi percabangan “IF” sebagai syarat apabila data yang dimasukkan pengguna telah digunakan, maka program akan meminta pengguna untuk memasukkan data lain.

3.6 Mendefinisikan Fungsi Akun

```
#akun
def akun():
    print("\nPilih Sign Up jika belum memiliki akun atau pilih Sign In jika sudah memiliki akun.")
    print("1. Sign In")
    print("2. Sign Up")
    try:
        opsi = int(input("Pilih opsi (1/2) : "))
        if opsi == 1:
            signin()
        elif opsi == 2:
            signup()
        else:
            print("Pilihan tidak valid.")
            akun()
    except ValueError:
        print("Kesalahan: Masukan harus berupa angka (1/2).")
        akun()
```

Gambar 3.6 Fungsi Akun

Fungsi ini akan menampilkan pilihan menu untuk masuk ke dalam program yang dapat diakses oleh pengguna. Pada fungsi ini digunakan juga penanganan eksepsi yang akan memberikan keterangan mengenai *input* yang dimasukkan pengguna invalid karena tidak sesuai dengan format yang ditentukan.

3.7 Mendefinisikan Fungsi Buka Detail kost

```
def buka_detail_data(pilihkost):
    with open('datakost.txt') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=',')
        for i in csv_reader:
            if i[0] == pilihkost:
                print('=====')
                print("DETAIL KOST")
                print("")
                print(f>Nama Kost : {i[1]}")
                print(f>Alamat : {i[3]}")
                print(f>Ukuran : {i[4]}")
                print(f>Harga Kost/Tahun : {i[5]}")
                print(f>Fasilitas : {i[6]}")
                print(f>Narahubung : {i[7]}")
                print(f>Jumlah Kamar yang Tersedia : {i[8]}")
                print('=====')
                return True
        return False
```

Gambar 3.7 Fungsi Buka Detail kost

Fungsi ini digunakan untuk menampilkan detail sebuah kost berdasarkan jenis kost dan *range* harga kost yang telah dipilih oleh pengguna. Kemudian fungsi ini memanggil detail data kost pada *database*.

3.8 Mendefinisikan Fungsi Aplikasi

```
def aplikasi():
    global total_harga
    with open('datakost.txt') as csv_file:
        csv_reader = csv.reader(csv_file, delimiter=',')
        print("\nSelamat datang di Program Pencarian Kost!")
        print("Pilih Jenis Kost (L/P)")
        daftar_kost = []
        jeniskost = input("Jenis Kost: ")
        if jeniskost.upper() == 'L':
            print("\nBerikut Daftar Harga Kost/Tahun:")
            print("1. Kurang dari Rp. 5.000.001")
            print("2. Rp. 5.000.001 sampai Rp. 10.000.000")
            print("3. Lebih dari Rp. 10.000.000")
        try:
            harga = int(input("Pilih Daftar Harga Kost/Tahun (1/2/3) : "))
            if harga == 1:
                print('=====')
                print("DAFTAR KOST:")
                for i in csv_reader:
                    if int(i[5]) <= 5000000 and i[2] == 'L':
                        daftar_kost.append(i)
                        print(f'{i[0]}. {i[1]}')
            elif harga == 2:
                print('=====')
                print("DAFTAR KOST:")
                for i in csv_reader:
                    if 5000000 < int(i[5]) <= 10000000 and i[2] == 'L':
                        daftar_kost.append(i)
                        print(f'{i[0]}. {i[1]}')
            elif harga == 3:
                print('=====')
                print("DAFTAR KOST:")
                for i in csv_reader:
                    if int(i[5]) > 10000000 and i[2] == 'L':
                        daftar_kost.append(i)
                        print(f'{i[0]}. {i[1]}')
            else:
                print("Pilihan tidak valid.")
                print('=====')
                aplikasi()
        except ValueError:
            print("Kesalahan: Masukan harus berupa angka (1/2/3).")
            aplikasi()
```

Gambar 3.8.1 Fungsi Aplikasi

```
print('=====')
print("\nApakah Anda ingin membuka detail salah satu kost?")
print("1. Ya")
print("2. Tidak")
try:
    pilihan = int(input("Masukkan pilihan (1/2): "))
    if pilihan == 1:
        pilihkost = input("Pilih nomor kost pada daftar untuk melihat detail: ")
        kost_ditemukan = False
        for kost in daftar_kost:
            if kost[0] == pilihkost:
                buka_detail_data(pilihkost)
                kost_ditemukan = True
                print("")
                jumlahkamar = int(input("Masukkan jumlah kamar kost yang ingin disewa: "))
                if 1 <= jumlahkamar <= int(kost[8]):
                    total_harga = jumlahkamar * int(kost[5])
                    print(f'Total harga sewa: Rp {total_harga}')
                    print("Apakah anda ingin melanjutkan ke menu pembayaran?")
                    print("1. Ya")
                    print("2. Tidak")
                    try:
                        keputusan = int(input("Masukkan pilihan (1/2): "))
                        if keputusan == 1:
                            pembayaran()
                        elif keputusan == 2:
                            print("Silakan kembali ke menu awal")
                            aplikasi()
                        else:
                            print("Pilihan tidak valid.")
                    except ValueError:
                        print("Kesalahan: Masukan harus berupa angka (1/2).")
                        aplikasi()
                else:
                    print("Kamar yang tersedia tidak cukup, akan dikembalikan ke menu awal.")
                    aplikasi()
        if not kost_ditemukan:
            print("Detail kos tidak tersedia.")
            aplikasi()
    elif pilihan == 2:
        aplikasi()
    else:
        print("Pilihan tidak valid.")
        aplikasi()
except ValueError:
    print("Kesalahan: Masukan harus berupa angka (1/2).")
    aplikasi()
```

Gambar 3.8.2 Fungsi Aplikasi

Fungsi aplikasi digunakan untuk menampilkan daftar kost yang sudah difilter berdasarkan jenis kost (L/P) dan *range* harga kost per tahun dengan memanggil nama kost pada *database*. Pada fungsi ini pengguna diminta memasukkan tipe data integer, yaitu berupa angka (1/2/3) untuk memilih *range* harga. Pada fungsi ini digunakan juga penanganan eksepsi yang akan memberikan keterangan mengenai *input* yang dimasukkan pengguna tidak valid karena tidak sesuai dengan format yang ditentukan.

Setelah muncul *output* berupa daftar kost berdasarkan jenis kost dan *range* harga kost per tahun yang telah dipilih pengguna, program akan melanjutkan untuk membuka salah satu detail kost yang terdapat pada daftar kost.

3.9 Mendefinisikan Fungsi Pembayaran

```
def pembayaran():
    print("\nPilih metode pembayaran:")
    print("1. BANK MANDIRI")
    print("2. BANK BCA")
    print("3. GOPAY")
    print("4. CASH")
    try:
        pilihan_metode = int(input("Masukkan nomor metode pembayaran yang dipilih: "))
        if pilihan_metode == 1:
            print("Anda memilih pembayaran dengan BANK MANDIRI : 1234567890 .")
            total_pembayaran = int(input("Masukkan total pembayaran: "))
            if total_pembayaran == total_harga:
                kode_bayar = generate_kode_pembayaran()
                print("Kode pembayaran:", kode_bayar)
                kode_input = input("Masukkan kode bayar: ")
                if kode_input == kode_bayar:
                    print("Kode pembayaran valid.")
                    print("Pembayaran berhasil")
                    bukti_pembayaran(pilihan_metode, kode_bayar, total_harga)
                else:
                    print("Kode pembayaran tidak valid, pembayaran gagal. Silakan lakukan pembayaran ulang.")
                    pembayaran()
            else:
                print("Total pembayaran tidak valid, pembayaran gagal. Silakan lakukan pembayaran ulang.")
                pembayaran()
    except:
```

Gambar 3.9 Fungsi Pembayaran

Fungsi pembayaran digunakan untuk menampilkan metode pembayaran setelah pengguna setuju untuk melanjutkan ke tahap transaksi.

3.10 Mendefinisikan Fungsi Kode Bayar Acak

```
def generate_kode_pembayaran():
    kode_bayar = ''.join(random.choices('0123456789', k=6))
    return kode_bayar
```

Gambar 3.10 Fungsi Kode Bayar Acak

Fungsi ini digunakan untuk menampilkan kode bayar berupa angka acak sebanyak 6 angka.

3.11 Mendefinisikan Fungsi Bukti Pembayaran

```
def bukti_pembayaran(pilihan_metode, kode_bayar, total_harga ):
    print("Bukti Pembayaran:")
    if pilihan_metode == 1:
        metode_pembayaran = "BANK MANDIRI"
    elif pilihan_metode == 2:
        metode_pembayaran = "BANK BCA"
    elif pilihan_metode == 3:
        metode_pembayaran = "GOPAY"
    else:
        metode_pembayaran = "CASH"
    print()
    print('=====')
    print("                STRUK PEMESANAN                ")
    print('=====')
    print("                Metode Pembayaran:", metode_pembayaran)
    print("                Dengan Kode Bayar:", kode_bayar)
    print("                Total pembayaran :", total_harga)
    print("                Pemesanan kost sudah otomatis masuk kedalam sistem")
    print("                Terima kasih atas pembayaran Anda.\n")
```

Gambar 3.11 Fungsi Bukti Pembayaran

Fungsi ini digunakan untuk menampilkan bukti pembayaran. Bukti pembayaran ini akan muncul sebagai *output* ketika pengguna telah berhasil melakukan transaksi.

BAB IV

HASIL RUNNING PROGRAM

Bab ini menjelaskan tentang *running* program pencarian kost.

4.1 HASIL RUNNING PROGRAM

4.1.1 Login

```
Pilih Sign Up jika belum memiliki akun atau pilih Sign In jika sudah memiliki akun.
1. Sign In
2. Sign Up
Pilih opsi (1/2)                : 1

Sign In
Masukkan email atau username akun Anda: nidhomun
Masukkan password akun Anda       : 123
Username atau password Anda tidak valid.

Sign In
Masukkan email atau username akun Anda: nidhomun
Masukkan password akun Anda       : passwordnidhomun
=====

INFORMASI PENGGUNA

Selamat datang, nidhomun
Sign In telah berhasil.

Selamat datang di Program Pencarian Kost!
Pilih Jenis Kost (L/P)
Jenis Kost: █
```

Gambar 4.1 Hasil *Running* Program Sign In

```
Pilih Sign Up jika belum memiliki akun atau pilih Sign In jika sudah memiliki akun.
1. Sign In
2. Sign Up
Pilih opsi (1/2)                : 2

Sign Up
Masukkan email                   : nidhomun@gmail.com
Masukkan username                 : mumun
Masukkan password                 : 123
Akun berhasil dibuat, silakan lakukan Sign In.

Sign In
Masukkan email atau username akun Anda: mumun
Masukkan password akun Anda       : 123
=====

INFORMASI PENGGUNA

Selamat datang, mumun
Sign In telah berhasil.

Selamat datang di Program Pencarian Kost!
Pilih Jenis Kost (L/P)
Jenis Kost: █
```

Gambar 4.2 Hasil *Running* Program Sign Up

Dari hasil tersebut diketahui bahwa pengguna dapat memilih menu *Sign In* atau *Sign Up*. Jika pengguna sudah memiliki akun, pengguna dapat memilih menu *Sign In* untuk masuk ke program. Namun, apabila pengguna belum mempunyai akun, pengguna dapat membuat akun dengan memilih menu *Sign Up*. Pengguna akan diminta untuk memasukkan *email*, *username*, dan *password* baru yang akan tersimpan otomatis di data csv. Kemudian jika akun berhasil dibuat, pengguna akan diminta untuk melakukan *Sign In* ulang dengan *email*, *username*, dan *password* baru yang telah dibuat sebelumnya.

4.1.2 Pencarian kost

```

INFORMASI PENGGUNA

Selamat datang, mumun
Sign In telah berhasil.

Selamat datang di Program Pencarian Kost!
Pilih Jenis Kost (L/P)
Jenis Kost: p

Berikut Daftar Harga Kost/Tahun:
1. Kurang dari Rp. 5.000.000
2. Rp. 5.000.000 sampai Rp. 10.000.000
3. Lebih dari Rp. 10.000.000
Pilih Daftar Harga Kost/Tahun (1/2/3) : 2
=====

DAFTAR KOST:
4. Kost Bu Erna
6. Wisma Sri
7. Kost Cipto
8. Kost Al-Azhar
17. Kost Rinumpaka 2
27. Wisma Putri Cahaya
28. Wisma Linaya
30. Wisma Putri Maheswari
=====

Apakah Anda ingin membuka detail salah satu kost?
1. Ya
2. Tidak
Masukkan pilihan (1/2): 1
Pilih nomor kost pada daftar untuk melihat detail: 28
=====

DETAIL KOST

Nama Kost          : Wisma Linaya
Alamat             : Jl Halimun No 59
Ukuran            : 3x2 m
Harga Kost/Tahun   : 6000000
Fasilitas          : KM Dalam
Narahubung        : 082323559197 (Asti)
Jumlah Kamar yang Tersedia : 1
=====

Masukkan jumlah kamar kost yang ingin disewa: 1
Total harga sewa: Rp 6000000
Apakah anda ingin melanjutkan ke menu pembayaran?
1. Ya
2. Tidak
Masukkan pilihan (1/2): 1

```

Gambar 4.3 Hasil Running Program Pencarian kost

Hasil *running* menampilkan menu jenis kost untuk (L/P). Kemudian program akan menampilkan pilihan 3 *range* harga sesuai kebutuhan pengguna yaitu:

1. Harga \leq Rp 5.000.000,00
2. Rp 5.000.000,00 < Harga \leq Rp 10.000.000,00
3. Harga > Rp 10.000.000,00

Setelah pengguna memilih *range* harga yang tersedia, program akan memberikan *output* berupa *list* daftar kost sesuai harga yang dipilih oleh pengguna. Pengguna akan diminta untuk memilih salah satu kost yang tersedia setelah program memberikan *output* mengenai *list* daftar kost sesuai harga yang diminta. Setelah itu program akan memberikan *output* berupa detail kost yaitu (nama kost, alamat, ukuran, harga kost/tahun, fasilitas, narahubung, dan jumlah kamar yang tersedia). Kemudian pengguna akan diminta untuk memasukkan jumlah kamar yang akan dipesan. Jika jumlah

kamar yang akan dipesan kurang dari sama dengan jumlah kamar yang tersedia, maka program akan menampilkan *output* berupa total harga sewa. Jika jumlah kamar yang dipesan lebih dari jumlah kamar yang tersedia maka program akan menampilkan *output* “Kamar yang tersedia tidak cukup”.

4.1.3 Pembayaran

```
Masukkan jumlah kamar kost yang ingin disewa: 1
Total harga sewa: Rp 6000000
Apakah anda ingin melanjutkan ke menu pembayaran?
1. Ya
2. Tidak
Masukkan pilihan (1/2): 1

Pilih metode pembayaran:
1. BANK MANDIRI
2. BANK BCA
3. GOPAY
4. CASH
Masukkan nomor metode pembayaran yang dipilih: 1
Anda memilih pembayaran dengan BANK MANDIRI : 1234567890 .
Masukkan total pembayaran: 6000000
Kode pembayaran: 317350
Masukkan kode bayar: 317350
Kode pembayaran valid.
Pembayaran berhasil
Bukti Pembayaran:

=====
                        STRUK PEMESANAN
=====

Metode Pembayaran: BANK MANDIRI
Dengan Kode Bayar: 317350
Total pembayaran : 6000000
Pemesanan kost sudah otomatis masuk kedalam sistem
Terima kasih atas pembayaran Anda.
```

Gambar 4.4 Hasil *Running* Program Pembayaran

Jika jumlah kamar yang dimasukkan pengguna tersedia dan pengguna ingin memesan kamar, maka pengguna akan diberikan total harga sesuai jumlah kamar yang dipesan. Setelah itu pengguna akan diberikan pilihan untuk memilih metode pembayaran yaitu ada:

1. Bank mandiri
2. Bank BCA
3. Gopay
4. Cash

Jika pengguna memilih *cash*, maka pengguna akan melakukan transaksi kepada pemilik kost secara langsung untuk menindaklanjuti proses pemesanan kost secara langsung. Sedangkan untuk metode pembayaran Bank

Mandiri, BCA, dan Gopay, pengguna akan diberikan nomor rekening dan kode bayar untuk melakukan validasi pembayaran. Pengguna harus memasukkan kode bayar pada metode bayar yang dipilih. Jika benar, program akan memberikan struk pembayaran.