

LAPORAN PRAKTIKUM POSTTEST 6
ALGORITMA DAN PEMROGRAMAN DASAR



Andi Fairul Amri < B1>

2409106065

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2024

LATAR BELAKANG

Dalam dunia pemrograman, *dictionary* merupakan salah satu struktur data yang sangat penting dan sering digunakan. *Dictionary* adalah tipe data yang menyimpan pasangan kunci-nilai (*key-value pairs*), di mana setiap kunci unik digunakan untuk mengakses nilai yang terkait. Karakteristik utama dari *dictionary* adalah kemampuannya untuk menyediakan akses cepat ke data melalui kunci, yang membuatnya sangat efisien untuk pencarian dan manipulasi data.

Dictionary bersifat *mutable*, artinya data di dalamnya dapat diubah setelah *dictionary* dibuat. Pengguna dapat menambahkan, menghapus, atau memperbarui pasangan kunci-nilai sesuai kebutuhan. Hal ini memberikan fleksibilitas yang besar dalam pengelolaan data. Selain itu, *dictionary* tidak menyimpan data dalam urutan tertentu, sehingga urutan penyimpanan tidak mempengaruhi kinerja atau penggunaan *dictionary*.

Penggunaan *dictionary* sangat luas dalam berbagai aplikasi pemrograman. Keuntungan utama dari *dictionary* adalah efisiensinya dalam akses dan manipulasi data, serta kemampuannya untuk menyimpan berbagai tipe data sebagai nilai, termasuk *list*, *tuple*, dan bahkan *dictionary* lainnya.

SOLUSI

Pada praktikum kali ini, akan dibuat sebuah program yang menggunakan *dictionary* sebagai penerapannya. Program akan dibuat layaknya manajemen data yang menggunakan *dictionary* di dalamnya. Di program ini akan dibuat sistem manajemen data bagi penghuni di sebuah asrama. Dalam sistem manajemen tersebut, terdapat beberapa data seperti nama, universitas, tahun masuk, nomor telepon, dan nomor kamar.

Di dalam program akan memuat halaman autentikasi dan menu utama. Sebelum memasuki menu utama, pengguna harus melewati halaman autentikasi dengan *login* jika telah memiliki akun, dan daftar jika belum memiliki akun. Pengguna yang mendaftar pada halaman daftar akan menjadi pengguna biasa. Berbeda dengan admin yang memiliki akun tersendiri. Ada beberapa menu yang

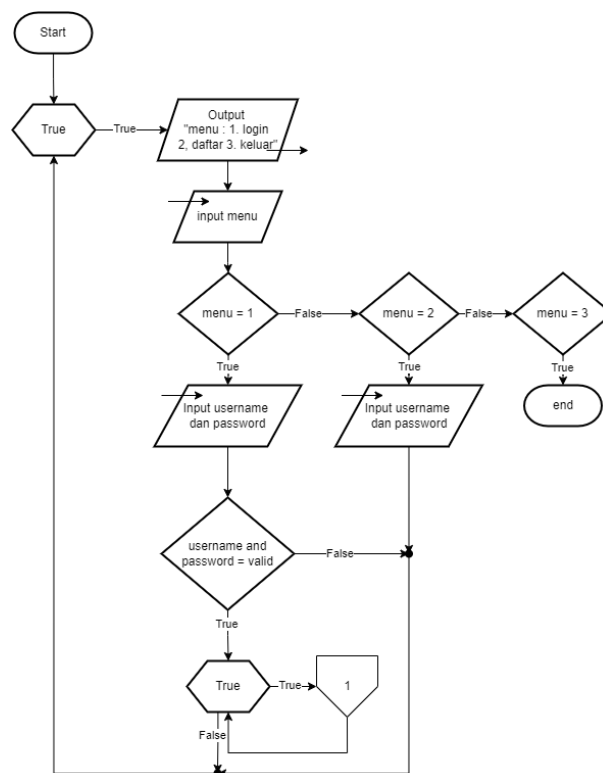
tersedia di dalam program yakni: tambah data, lihat data, ubah data, dan hapus data penghuni. Pengguna biasa hanya dapat mengakses lihat data penghuni. Selebihnya, hanya admin yang dapat mengakses menu tersebut.

Untuk lebih lanjut, berikut *flowchart* dan program *python* yang dibuat.

A. Flowchart

Pembuatan *flowchart* menggunakan aplikasi Draw.io dengan membagi menjadi dua bagian menggunakan *off page reference*. *Flowchart* dibuat dengan menggunakan simbol *while*. Sehingga membentuk perulangan yang akan berulang hingga mencapai kondisi tertentu untuk berhenti.

Berikut gambar *flowchart* pertama



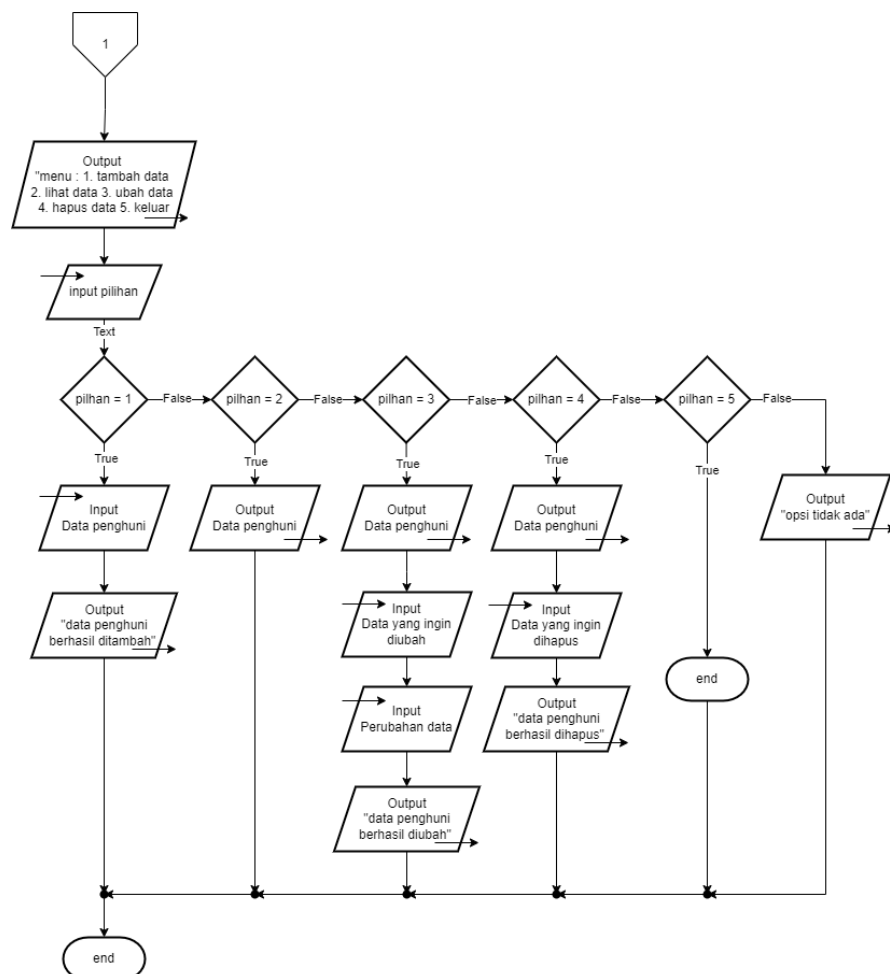
Gambar 1: Flowchart pertama

Pada *flowchart* tersebut terdiri dari :

- *Terminator main*, yang merupakan simbol awal untuk flowchart.
- *While* pertama yang menjadi perulangan jika true , maka akan lanjut ke proses selanjutnya yaitu output menu.

- Jika memilih 1, maka akan meminta masukan username dan password untuk login. Jika memilih 2, maka akan meminta username dan password untuk daftar. Jika memilih 3, maka program akan berhenti
- Setelah login, jika *username* dan *password* bernilai benar maka akan lanjut ke proses selanjutnya. Namun jika bernilai salah maka akan terjadi perulangan di *while* pertama.
- Setelah daftar, username dan password akan tersimpan dan menuju ke perulangan *while* pertama
- *While* kedua yang menjadi perulangan jika bernilai benar maka akan lanjut ke proses selanjutnya.
- *Off page reference* menjadi penghubung antara dua gambar flowchart.

Berikut gambar *flowchart* kedua.



Gambar 2: Flowchart kedua

Pada *flowchart* tersebut terdiri dari.

- *Terminator main*, yang merupakan simbol awal untuk flowchart.
- *Output* menu yang tersedia.
- *Input* variabel menu, yang akan digunakan oleh pengguna dalam memilih menu yang tersedia. Pengguna dapat memilih menu 1-5.
- *Decision* pertama yang menjadi percabangan jika variabel menu $\neq 1$, maka akan lanjut ke percabangan selanjutnya. Namun jika menu = 1 maka akan melanjutkan ke proses di bawahnya yaitu program menambahkan data penghuni. Program ini membutuhkan *input* dari pengguna untuk menjadi data penghuni yang akan dimasukkan ke *dictionary*. Setelah itu, akan menampilkan pesan berhasil. Selanjutnya *flowchart* diakhiri dengan *terminator end* dan akan kembali ke *flowchart* pertama untuk melakukan perulangan.
- *Decision* kedua yang menjadi percabangan jika variabel menu $\neq 2$, maka akan lanjut ke percabangan selanjutnya. Namun jika menu = 2, maka akan melanjutkan ke proses di bawahnya yaitu program melihat data penghuni. Program ini akan menampilkan data penghuni yang telah dimasukkan. Selanjutnya *flowchart* diakhiri dengan *terminator end* dan akan kembali ke *flowchart* pertama untuk melakukan perulangan.
- *Decision* ketiga yang menjadi percabangan jika variabel menu $\neq 3$, maka akan lanjut ke percabangan selanjutnya. Namun jika menu = 3, maka akan melanjutkan ke proses di bawahnya yaitu program mengubah data penghuni. Program ini akan menampilkan data penghuni yang telah dimasukkan. Setelah itu, akan meminta masukan data yang ingin diubah. Lalu, masukkan kembali perubahan data. Selanjutnya *flowchart* diakhiri dengan *terminator end* dan akan kembali ke *flowchart* pertama untuk melakukan perulangan.
- *Decision* keempat yang menjadi percabangan jika variabel menu $\neq 4$, maka akan lanjut ke percabangan selanjutnya. Namun jika menu = 4, maka akan melanjutkan ke proses di bawahnya yaitu program menghapus data penghuni. Program ini akan menampilkan data penghuni yang telah

dimasukkan. Setelah itu, akan meminta masukan data yang ingin dihapus. Lalu, akan muncul pesan berhasil. Selanjutnya *flowchart* diakhiri dengan *terminator end* dan akan kembali ke *flowchart* pertama untuk melakukan perulangan.

- *Decision* kelima yang menjadi percabangan jika variabel menu = 5, maka program akan berakhir dan kembali ke *flowchart* pertama. Namun jika menu \neq 5, maka akan menampilkan pesan pilhan tidak ada dan kembali ke *flowchart* pertama.

B. Program Python

Berikut penjelasan tentang kode yang dibuat:

1. *Import* modul os & getpass

Modul os digunakan untuk menjalankan perintah sistem operasi, seperti membersihkan layar terminal. Getpass digunakan untuk mengambil masukan kata sandi dari pengguna tanpa menampilkannya di layar, menjaga kerahasiaan kata sandi.

2. Data pengguna dan role

User sebagai kamus (*dictionary*) yang menyimpan pasangan username dan password. Diinisialisasi dengan satu pengguna default, yaitu admin. User_role sebagai kamus yang menyimpan peran pengguna. Admin diinisialisasi sebagai “admin”.

3. Data penghuni asrama

Dictionary kosong yang nantinya akan berfungsi sebagai tempat data.

4. Fungsi registrasi pengguna baru

Meminta masukan *username* dan akan ada pengecekan dalam masukan *username* agar tidak terjadi penggandaan dan akan menampilkan pesan kesalahan jika terjadi penggandaan. Selanjutnya meminta masukan dan *password* menggunakan *getpass*. Kemudian username dan password akan ditambahkan dalam kamus user. Lalu dalam kamus (*dictionary*) user_role akan ditambahkan sebagai pengguna. Diakhiri dengan pesan “registrasi berhasil” jika tidak ada kesalahan.

5. Fungsi login

Meminta masukan *username* dan *password* dari pengguna. Memeriksa apakah *username* dan *password* cocok dengan data dalam kamus *user*. Jika cocok, menampilkan pesan sukses dan mengembalikan *username*. Jika tidak cocok, menampilkan pesan kesalahan dan mengembalikan ke menu.

6. Fungsi menambah data penghuni asrama

Meminta masukan data penghuni (nama, universitas, tahun masuk, nomor handphone, nomor kamar). Selanjutnya akan dimasukkan ke dalam *dictionary* penghuni_asrama.

7. Fungsi Menampilkan Data Penghuni

Memeriksa apakah *dictionary* penghuni_asrama kosong. Jika kosong, menampilkan pesan bahwa tidak ada data penghuni. Jika tidak kosong, menampilkan data setiap penghuni dalam *dictionary* penghuni_asrama.

8. Fungsi Memperbarui Data Penghuni

Menampilkan data penghuni yang ada. Selanjutnya, meminta masukan nomor penghuni yang akan diperbarui. Kemudian memeriksa apakah nomor penghuni valid. Jika valid, meminta masukan data baru dan memperbarui data penghuni. Jika tidak valid, menampilkan pesan kesalahan. Dan terdapat penanganan dalam kesalahan masukan yang tidak valid.

9. Fungsi Menghapus Data Penghuni

Menampilkan data penghuni yang ada. Selanjutnya, meminta masukan nomor penghuni yang akan dihapus. Kemudian memeriksa apakah nomor penghuni valid. Jika valid, data akan dihapus dari *dictionary* penghuni_asrama. Jika tidak valid, menampilkan pesan kesalahan. Dan terdapat penanganan dalam kesalahan masukan yang tidak valid.

10. Menu login dan daftar

Terdapat perulangan `while true` yang akan terus berjalan sampai pengguna memilih untuk keluar. Selanjutnya membersihkan layar terminal dan menampilkan menu utama dengan opsi untuk login, daftar, atau keluar. Jika pengguna memilih menu login, maka akan memanggil fungsi login yang telah diprogramkan. Jika pengguna memilih menu daftar, maka akan

memanggil fungsi daftar yang telah diprogramkan. Jika pengguna memilih menu keluar, maka program akan berhenti. Dan ketika login berhasil, program memasuki loop while true kedua untuk menampilkan menu manajemen data penghuni asrama yang meliputi CRUD (*Cread. Read, Update, Delete*).

11. Pemilihan menu utama

Dalam pemilihan menu menggunakan `.strip()` untuk menghapus karakter spasi (atau karakter lain yang ditentukan) dari awal dan akhir *string*.

- Jika pengguna memilih menu 1, program akan melanjutkan ke fungsi menambahkan data penghuni. Pada bagian ini akan ada validasi peran, sehingga hanya admin yang dapat melanjutkan program. Jika bukan admin, maka akan menampilkan pesan kesalahan.
- Jika pengguna memilih menu 2, program akan menampilkan data semua penghuni. Jika data penghuni masih kosong atau belum ada ditambahkan, maka akan menampilkan pesan “data belum tersedia”.
- Jika pengguna memilih menu 3, program akan melanjutkan ke fungsi perbarui data penghuni. Pada bagian ini akan ada validasi peran, sehingga hanya admin yang dapat melanjutkan program. Jika bukan admin, maka akan menampilkan pesan kesalahan.
- Jika pengguna memilih menu 4, program akan melanjutkan ke fungsi hapus data penghuni. Pada bagian ini akan ada validasi peran, sehingga hanya admin yang dapat melanjutkan program. Jika bukan admin, maka akan menampilkan pesan kesalahan.
- Jika pengguna memilih menu 5, maka akan keluar ke halaman *login*.
- Jika pengguna memilih selain dari 1-5, program akan mengeluarkan *output* “opsi tidak valid” dan selanjutnya akan terjadi perulangan.

12. Baris kode `if __name__ == "__main__":`:

Untuk memastikan bahwa kode di dalam blok tersebut hanya dijalankan jika skrip tersebut dijalankan secara langsung, bukan diimpor sebagai modul di skrip lain.

Berikut gambar program python yang telah dibuat.

```
2409106065_Andi Fairul Amri_B1_Posttest6

import os
import getpass

# Data pengguna
user = {"admin": "admin123"} # Akun untuk admin
user_role = {"admin": "admin"}

# Data penghuni asrama
penghuni_asrama = {}

# Registrasi pengguna baru
def daftar():
    os.system('cls || clear')
    print("Registrasi Pengguna Baru".center(50))
    print("=" * 50)
    username = input("Masukkan username: ")
    if username in user:
        print("Username sudah ada!")
        input("Klik Enter untuk melanjutkan . . .")
        return
    password = getpass.getpass("Masukkan password: ")
    user[username] = password
    user_role[username] = "pengguna" # Semua pengguna baru adalah "pengguna"
    print("Registrasi berhasil!")
    print()
    input("Klik Enter untuk melanjutkan . . .")

# Login
def login():
    os.system('cls || clear')
    print("Login".center(50))
    print("=" * 50)
    username = input("Masukkan username: ")
    password = getpass.getpass("Masukkan password (Password akan tertutup): ")
    if user.get(username) == password:
        print("Login berhasil!")
        print()
        input("Enter untuk melanjutkan . . .")
        return username
    else:
        print("Username atau password salah!")
        print()
        input("Enter untuk melanjutkan . . .")
        return None

# Menambahkan data penghuni
def buat_data():
    os.system('cls || clear')
    print("Tambah Data Penghuni".center(50))
    print("=" * 50)
    nama = input("Nama : ")
    universitas = input("Universitas : ")
    tahun_masuk = input("Tahun Masuk : ")
    nomor_hp = input("Nomor Handphone: ")
    nomor_kamar = input("Nomor Kamar : ")
    penghuni_asrama[nama] = {
        "universitas": universitas,
        "tahun_masuk": tahun_masuk,
        "nomor_hp": nomor_hp,
        "nomor_kamar": nomor_kamar
    }
    print()
    print("Data berhasil ditambahkan!")
    print()
    input("Klik Enter untuk melanjutkan . . .")
```

Gambar 3: Program Python

```

# Menampilkan data penghuni
def tampil_data():
    os.system('cls || clear')
    print("Data Penghuni Asrama".center(50))
    print("=" * 50)
    if not penghuni_asrama:
        print("Data belum tersedia.")
        print()
        input("Klik Enter untuk melanjutkan . . .")
        return
    for i, (nama, data) in enumerate(penghuni_asrama.items(), start=1):
        print(f"Penghuni {i}:")
        print()
        print(f" Nama           : {nama}")
        print(f" Universitas      : {data['universitas']}")
        print(f" Tahun Masuk      : {data['tahun_masuk']}")
        print(f" Nomor Handphone: {data['nomor_hp']}")
        print(f" Nomor Kamar      : {data['nomor_kamar']}")
        print("=" * 50)
        print()
        input("Klik Enter untuk melanjutkan . . .")

# Memperbarui data penghuni
def ubah_data():
    tampil_data()
    print()
    print("Perbarui Data Penghuni Asrama".center(50))
    print("=" * 50)
    if not penghuni_asrama:
        return
    nama = input("Masukkan nama penghuni yang akan diperbarui: ")
    if nama in penghuni_asrama:
        universitas = input("Universitas : ")
        tahun_masuk = input("Tahun Masuk : ")
        nomor_hp = input("Nomor Handphone: ")
        nomor_kamar = input("Nomor Kamar : ")
        penghuni_asrama[nama] = {
            "universitas": universitas,
            "tahun_masuk": tahun_masuk,
            "nomor_hp": nomor_hp,
            "nomor_kamar": nomor_kamar
        }
        print("Data berhasil diperbarui!")
        input("Klik Enter untuk melanjutkan . . .")
    else:
        print("Nama penghuni tidak valid!")
        input("Klik Enter untuk melanjutkan . . .")

# Menghapus data penghuni
def hapus_data():
    tampil_data()
    print()
    print("Hapus Data Penghuni Asrama".center(50))
    print("=" * 50)
    if not penghuni_asrama:
        return
    nama = input("Masukkan nama penghuni yang akan dihapus: ")
    if nama in penghuni_asrama:
        del penghuni_asrama[nama]
        print()
        print("Data berhasil dihapus!")
        print()
        input("Klik Enter untuk melanjutkan . . .")
    else:
        print("Nama penghuni tidak valid!")
        print()
        input("Klik Enter untuk melanjutkan . . .")

```

Gambar 4: Program Python 2

```

# Menu utama
def main():
    while True:
        os.system('cls || clear')
        print("=" * 50)
        print("Sistem Manajemen Data Penghuni Asrama".center(50))
        print("=" * 50)
        print("1. Login")
        print("2. Daftar")
        print("3. Keluar")
        print("=" * 50)
        pilihan = input("Pilih opsi: ").strip()
        if pilihan == "1":
            username = login()
            if username:
                while True:
                    os.system('cls || clear')
                    print("=" * 50)
                    print("Sistem Manajemen Data Penghuni Asrama".center(50))
                    print("=" * 50)
                    print(f"Selamat Datang {username}, Silakan pilih menu")
                    print()
                    print("1. Tambah Data Penghuni")
                    print("2. Lihat Data Penghuni")
                    print("3. Perbarui Data Penghuni")
                    print("4. Hapus Data Penghuni")
                    print("5. Logout")
                    print("=" * 50)
                    pilihan = input("Masukkan opsi: ").strip()
                    if pilihan == "1":
                        if user_role[username] == "admin":
                            buat_data()
                        else:
                            print("Hanya admin yang dapat menambah data!")
                            print()
                            input("Klik Enter untuk melanjutkan . . .")
                    elif pilihan == "2":
                        tampil_data()
                    elif pilihan == "3":
                        if user_role[username] == "admin":
                            ubah_data()
                        else:
                            print("Hanya admin yang dapat memperbarui data!")
                            print()
                            input("Klik Enter untuk melanjutkan . . .")
                    elif pilihan == "4":
                        if user_role[username] == "admin":
                            hapus_data()
                        else:
                            print("Hanya admin yang dapat menghapus data!")
                            print()
                            input("Klik Enter untuk melanjutkan . . .")
                    elif pilihan == "5":
                        break
                    else:
                        print("Opsi tidak valid!")
                        input("Klik Enter untuk melanjutkan . . .")
                elif pilihan == "2":
                    daftar()
                elif pilihan == "3":
                    break
                else:
                    print("Opsi tidak valid!")
                    input("Klik Enter untuk melanjutkan . . .")

if __name__ == "__main__":
    main()

```

Gambar 5: Program Python 3

Berikut gambar terminal dari program tersebut.

```
=====
                Sistem Manajemen Data Penghuni Asrama
=====
1. Login
2. Daftar
3. Keluar
=====
Pilih opsi: 1
```

Gambar 5: Terminal output program

```

                        Login
=====
Masukkan username: admin
Masukkan password (Password akan tertutup):
Login berhasil!

Enter untuk melanjutkan . . .
```

Gambar 6: Terminal output program

```
=====
                Sistem Manajemen Data Penghuni Asrama
=====
Selamat Datang admin, Silakan pilih menu

1. Tambah Data Penghuni
2. Lihat Data Penghuni
3. Perbarui Data Penghuni
4. Hapus Data Penghuni
5. Logout
=====
Masukkan opsi: 1
```

Gambar 7: Terminal output program

```

                          Tambah Data Penghuni
=====
Nama          : andi
Universitas   : Universitas Mulawarma
Tahun Masuk   : 2019
Nomor Handphone: 082253419876
Nomor Kamar   : 21

Data berhasil ditambahkan!

Klik Enter untuk melanjutkan . . .

```

Gambar 8: Terminal output program

```

                          Data Penghuni Asrama
=====
Penghuni 1:

Nama          : andi
Universitas   : Universitas Mulawarma
Tahun Masuk   : 2019
Nomor Handphone: 082253419876
Nomor Kamar   : 21
=====

Klik Enter untuk melanjutkan . . .

```

Gambar 8: Terminal output program

```

                          Data Penghuni Asrama
=====
Penghuni 1:

Nama          : andi
Universitas   : Universitas Mulawarman
Tahun Masuk   : 2019
Nomor Handphone: 082253419876
Nomor Kamar   : 21
=====

Klik Enter untuk melanjutkan . . .

                          Perbarui Data Penghuni Asrama
=====
Masukkan nomor penghuni yang akan diperbarui: 1
Nama          : andi
Universitas   : Politeknik Samarinda
Tahun Masuk   : 2019
Nomor Handphone: 082251417157
Nomor Kamar   : 12
Data berhasil diperbarui!
Klik Enter untuk melanjutkan . . .

```

Gambar 9: Terminal output program

```

                                Data Penghuni Asrama
=====
Penghuni 1:

Nama           : andi
Universitas    : Politeknik Samarinda
Tahun Masuk   : 2019
Nomor Handphone: 082251417157
Nomor Kamar    : 12
=====

Klik Enter untuk melanjutkan . . .

                                Hapus Data Penghuni Asrama
=====
Masukkan nomor penghuni yang akan dihapus: 1

Data berhasil dihapus!

Klik Enter untuk melanjutkan . . .
```

Gambar 10: Terminal output program