



## **TUGAS BESAR PROGRAMA KOMPUTER**

**“Penerapan Python Pada Perancangan Sistem Objek Wisata”**

**Oleh:**

**Kelompok 7A**

**Anggota:**

Muhammad Rayhan	(2410932033)
Kirana Dwiva Ninda	(2410931005)
Nadhirah Zaskya Syanela	(2410932029)

**Asisten Pembimbing:**

Andre Nugraha Akbar

Hanifa Aulia Rahma

**Dosen Pengampu:**

Dr. Eng. Ardhian Agung Yulianto S.Kom, M.T



**LABORATORIUM SISTEM INFORMASI DAN KEPUTUSAN  
DEPARTEMEN TEKNIK INDUSTRI  
FAKULTAS TEKNIK  
UNIVERSITAS ANDALAS  
2025**

# DAFTAR ISI

## ***COVER***

## **LEMBAR ASISTENSI**

## **DAFTAR ISI.....i**

## **DAFTAR GAMBAR.....ii**

## **DAFTAR LAMPIRAN.....iv**

## **BAB I PENDAHULUAN**

### **1.1 Gambaran Umum.....1**

### **1.2 Tujuan.....2**

## **BAB II ALGORITMA DAN FLOWCHART**

### **2.1 Algoritma Deskriptif.....3**

### **2.2 *Syntax*.....5**

### **2.3 *Flowchart*.....17**

## **BAB III ANALISIS PROGRAM**

### **3.1 Analisis Penerapan Modul pada Program.....29**

#### **3.1.1 Analisis Modul 1 – Pengenalan Komputer dan Dasar Bahasa Python.....29**

#### **3.1.2 Analisis Modul 2 – Konsep Keputusan.....32**

#### **3.1.3 Analisis Modul 3 – Perulangan (*loop*).....34**

#### **3.1.4 Analisis Modul 4 – Array.....37**

#### **3.1.5 Analisis Modul 5 – Program Berorientasi Objek (OOP).....40**

## **DAFTAR PUSTAKA**

## **LAMPIRAN**

## DAFTAR GAMBAR

- Gambar 2.1** *Flowchart menu login*
- Gambar 2.2** *Flowchart menu login (Lanjutan)*
- Gambar 2.3** *Flowchart menu User*
- Gambar 2.4** *Flowchart menu User (Lanjutan)*
- Gambar 2.5** *Flowchart menu Admin*
- Gambar 2.6** *Flowchart menu Admin (Lanjutan)*
- Gambar 2.7** *Flowchart menu Admin (Lanjutan)*
- Gambar 2.8** *Flowchart menu Admin (Lanjutan)*
- Gambar 2.9** *Flowchart menu Admin (Lanjutan)*
- Gambar 2.10** *Flowchart menu Admin (Lanjutan)*
- Gambar 2.11** *Flowchart menu Admin (Lanjutan)*
- Gambar 3.1** Contoh *String* Dalam Program
- Gambar 3.2** Contoh *Boolean*
- Gambar 3.3** Contoh *Variabel*
- Gambar 3.4** Contoh Perintah *Print*
- Gambar 3.5** Contoh Perintah *Input*
- Gambar 3.6** Contoh Kondisi *IF*
- Gambar 3.7** Contoh Kondisi *ELIF*
- Gambar 3.8** Contoh Kondisi *Else*
- Gambar 3.9** Contoh *While Statement*
- Gambar 3.10** Contoh *For Statement*
- Gambar 3.11** Contoh *Break*
- Gambar 3.12** Contoh *Continue*
- Gambar 3.13** Contoh *List*
- Gambar 3.14** Contoh *Dictionary*
- Gambar 3.15** Contoh *Dictionary* Dalam *List*
- Gambar 3.16** Contoh *Append*
- Gambar 3.17** Contoh *Remove*
- Gambar 3.18** Contoh *Class*
- Gambar 3.19** Contoh *Method Constructor*

**Gambar 3.20** Contoh *Method*

**Gambar 3.21** Contoh *Encapsulation*

**Gambar 3.22** Contoh *Polymorphism*

## **DAFTAR LAMPIRAN**

**Lampiran A** (*Output* Program)

**Lampiran B** (Kartu Praktikum)

# **BAB I**

## **PENDAHULUAN**

Bab ini memuat penjelasan tentang gambaran umum dan tujuan dari pembuatan program. Bab ini menjelaskan latar belakang pembuatan program, manfaat program secara umum, dan solusi terhadap permasalahan yang ada, serta tujuan khusus dari pembuatan program, sehingga dapat memberikan Gambaran awal dari program yang dibuat

### **1.1 Gambaran Umum**

Pariwisata adalah salah satu sektor yang sangat berdampak akan peluang dan pertumbuhan ekonomi, pelestarian budaya, dan kesejahteraan masyarakat. Hal ini penting untuk diperhatikan karena masih banyak situs dan objek wisata yang masih belum dikelola secara optimal, terutama terkait ketersediaan fasilitas, kemudahan akses, dan informasi lainnya untuk menarik minat wisatawan. Oleh karena itu, pengelolaan objek wisata perlu diperhatikan dan dilakukan pembaharuan.

Perkembangan teknologi memudahkan kita akan akses informasi dan mengoptimalkan dalam manajemen data. Salah satu manfaatnya adalah pengelolaan dan pempromosian objek wisata dengan suatu aplikasi yang dapat ditujukan pada wisatawan sebagai promosi wisata dan bagi manajemen untuk memberikan informasi terkait objek wisata secara *real time*. Pemanfaatan ini dapat direalisasikan dengan suatu program atau aplikasi yang efisien dan efektif. Salah satu bahasa pemrograman untuk pengembangan program ini adalah Python, di mana bahasa pemrograman ini memiliki keunggulan fleksibel, sederhana dan adaptif dibanding bahasa pemrograman lain.

Program “Penerapan Python pada Perancangan Sistem Objek Wisata” adalah suatu program yang merujuk pada studi kasus perancangan sebuah sistem berbahasa pemrograman Python yang memfasilitasi objek wisata di Kota Padang,

Batam, dan Bali, yang memiliki permasalahan utama terkait kurangnya suatu sistem wisata yang interaktif, mudah diakses, dan ramah pengguna. Sistem ini menggunakan konsep OOP (*Object Oriented Programming*) yang berarti program dirancang berfokus pada objek untuk penyelesaian masalah sehingga penulisan kode lebih efektif dan efisien. Selain penggunaan OOP, fitur-fitur lain dalam Python seperti struktur data (list dan dictionary) dan konsep keputusan, dan perulangan.

Program ini dirancang untuk membantu dalam pengelolaan informasi hotel, tempat wisata dan rumah makan di tiga kota, yaitu Padang, Batam, dan Bali. Akses yang dapat digunakan adalah admin dan user. Akses admin mempunyai akses penuh dalam mengelola data, seperti menambah data, menghapus data, mengedit data, dan menampilkan data. Sedangkan akses user hanya mempunyai akses untuk menampilkan dan mencari data dengan parameter tertentu. Hal ini diharapkan dapat memberikan fasilitas pengelolaan data yang sederhana namun efisien oleh pihak pengelola wisata lokal yang ingin mengelola informasi wisata secara digital dan kemudahan akses informasi bagi pengguna umum.

## **1.2 Tujuan**

Tujuan dari pembuatan program ini, yaitu :

1. Memberikan info terkait Objek Wisata yang ada di Kota Padang, Batam, dan Bali.
2. Memberikan akses kepada pengguna (*user*) untuk mencari informasi Objek Wisata di Kota Padang, Batam, dan Bali.
3. Memberikan akses kepada *admin* untuk menambahkan, menghapus, dan mengedit data informasi Objek Wisata di Kota Padang, Batam, dan Bali.

## **BAB II**

### **ALGORITMA DAN FLOWCHART**

Bab ini memuat algoritma deskriptif, sintaks atau kode program, dan *flowchart* yang disusun untuk memudahkan representasi alur pemrograman. Algoritma disusun secara berurutan dalam bentuk teks, sintaks berupa kode-kode yang merupakan iterasi dalam program, dan *flowchart* berupa diagram alir yang memudahkan penggambaran proses atau alur program secara visual.

#### **2.1 Algoritma Deskriptif**

0. Mulai program
1. Tampilkan menu utama: 1. Sign Up, 2. Login, 3. Keluar
2. *Input* pilihan menu
3. Jika pilih menu Sign Up, lanjutkan ke nomor 6. Jika tidak, tampilkan "Pilihan tidak *valid*" dan kembali ke menu utama
4. Jika pilih menu Login, lanjutkan ke nomor 15. Jika tidak, tampilkan "Pilihan tidak *valid*" dan kembali ke menu utama
5. Jika pilih menu 3, lanjutkan ke nomor 64. Jika tidak, tampilkan "Pilihan tidak *valid*" dan kembali ke menu utama
6. *Input username*
7. *Input password*
8. *Input role (admin/user)*
9. Validasi *input*
10. Jika *username* kosong atau sudah ada, lanjutkan ke nomor 12
11. Jika *role* diinputkan bukan *admin* atau *user*, lanjutkan ke nomor 12
12. Tampilkan pesan *error*, Kembali ke nomor 6
13. Jika semua *valid*, lanjutkan ke nomor 14.
14. Simpan data pengguna
15. Kembali ke nomor 1
16. *Input username* dan *password*



17. Validasi *input* kombinasi *username* dan *password*
18. Jika salah, lanjutkan ke nomor 20
19. Jika benar, periksa *role* pengguna, lanjutkan ke nomor 21
20. Tampilkan pesan *error*. Kembali nomor 16
21. Jika *role* sebagai *admin*, lanjut ke nomor 23
22. Jika *role* sebagai *user*, lanjut ke nomor 53
23. Tampilkan menu *admin* : 1. Tampilkan Data, 2. Tambah Data, 3. Hapus Data, 4. Edit Data, 5. Keluar
24. *Input* pilihan menu *admin*
25. Jika pilih selain Keluar (1-4), lanjut ke nomor 27
26. Jika pilih Keluar, lanjutkan ke nomor 64
27. *Input* kota (1. Padang/2. Batam/3. Bali)
28. *Input* jenis data (1. Hotel/2. Wisata/3. Rumah Makan)
29. Sistem memfilter data sesuai kota dan jenis data
30. Jika pilih 1. Tampilkan Data, tampilkan data, lanjutkan ke nomor 39
31. Jika pilih 2. Tambah Data, lanjutkan ke nomor 34
32. Jika pilih 3. Hapus Data, lanjutkan ke nomor 42
33. Jika pilih 4. Edit Data, lanjutkan ke nomor 44
34. *Input* kode
35. *Input* nama
36. *Input* deskripsi
37. *Input* *contact person*
38. Tampilkan data berhasil ditambahkan
39. Tampilkan menu ingin lanjut atau tidak (y/n)
40. Jika y, Kembali ke nomor 27
41. Jika n, Kembali ke 23
42. Masukkan kode
43. Tampilkan data berhasil dihapus, Kembali ke nomor 39
44. Masukkan kode yang akan diubah
45. Tampilkan data yang dapat diubah (a. Lokasi/b. nama/c. deskripsi/d. *contact person*)
46. Masukkan pilihan yang akan diedit (a/b/c/d)

47. Jika pilihan a, inputkan Lokasi terbaru, lanjutkan ke nomor 51
48. Jika pilihan b, inputkan nama terbaru, lanjutkan ke nomor 51
49. Jika pilihan c, inputkan deskripsi terbaru, lanjutkan ke nomor 51
50. Jika pilihan d, inputkan *contact person* terbaru, lanjutkan ke nomor 51
51. Tampilkan data telah diperbarui, kembali ke nomor 39
52. Tampilkan program selesai
53. Tampilkan menu *user*: 1. Tampilkan data berdasarkan kota, 2. Keluar
54. *Input* menu (1/2)
55. Jika pilih 1. Tampilkan data berdasarkan kota, lanjutkan ke nomor 57
56. Jika pilih 2. Keluar, lanjutkan ke nomor 64
57. Pilih kota (1. Padang/2. Batam/3. Bali)
58. Pilih jenis data (1. Hotel/2. Wisata/3. Rumah Makan)
59. Sistem memfilter data
60. Tampilkan data
61. Tampilkan menu ingin lanjut atau tidak(y/n)
62. Jika y, Kembali ke nomor 57
63. Jika n, kembali ke nomor 53
64. Selesai

## 2.2 *Syntax*

*Syntax* adalah aturan penulisan kode pada program yang harus dipatuhi agar program dapat berjalan.

class DataDasar:

```
def __init__(self, kode, lokasi, nama, desc, cp):  
    self.kode = kode  
    self.lokasi = lokasi  
    self.nama = nama  
    self.desc = desc  
    self.cp = cp
```

class Data:

```
def __init__(self):
```

```
    self.namah_daftar = []
```

```
    self.namaw_daftar = []
```

```
    self.namar_daftar = []
```

```
def data_ahotel(self):
```

```
    datah = [
```

```
        {"kode": "PDH01", "lokasi": "Padang", "nama": "Mercure Padang",
```

```
"desc": "Western Food", "cp": "0811"},
```

```
        {"kode": "PDH02", "lokasi": "Padang", "nama": "Grand Zuri Padang",
```

```
"desc": "Tengah Kota", "cp": "0812"},
```

```
        {"kode": "BTH01", "lokasi": "Batam", "nama": "Batam Marriott Hotel  
Harbour Bay", "desc": "Sea View", "cp": "0813"},
```

```
        {"kode": "BTH02", "lokasi": "Batam", "nama": "The Music Hotel ",  
"desc": "Budget-friendly", "cp": "0814"},
```

```
        {"kode": "BLH01", "lokasi": "Bali", "nama": "Neo Kuta Jelantik Hotel",  
"desc": "Budget-friendly", "cp": "0815"},
```

```
        {"kode": "BLH02", "lokasi": "Bali", "nama": "Hotel Siesta Legian",  
"desc": "Dekat Pantai", "cp": "0816"},
```

```
    ]
```

```
    for dh in datah:
```

```
        self.namah_daftar.append(DataDasar(dh["kode"], dh["lokasi"],  
dh["nama"], dh["desc"], dh["cp"]))
```

```
def data_awis(self):
```

```
    dataw = [
```

```
        {"kode": "PDW01", "lokasi": "Padang", "nama": "Pantai Air Manis",  
"desc": "Legenda Malin Kundang", "cp": "0821"},
```

```
        {"kode": "PDW02", "lokasi": "Padang", "nama": "Jembatan Siti Nurbaya",  
"desc": "Pemandangan Pelabuhan", "cp": "0822"},
```

```

        {"kode": "BTW01", "lokasi": "Batam", "nama": "Mega wisata Ocarina
Theme Park", "desc": "Taman Hiburan Keluarga", "cp": "0823"},
        {"kode": "BTW02", "lokasi": "Batam", "nama": "Miniature House
Indonesia", "desc": "Free-Entry", "cp": "0824"},
        {"kode": "BLW01", "lokasi": "Bali", "nama": "Pura Uluwatu", "desc":
"Kuil Hindu", "cp": "0825"},
        {"kode": "BLW02", "lokasi": "Bali", "nama": "Garuda Wisnu Kencana",
"desc": "Patung raksasa", "cp": "0826"}
    ]
    for dw in dataw:
        self.namaw_daftar.append(DataDasar(dw["kode"], dw["lokasi"],
dw["nama"], dw["desc"], dw["cp"]))

    def data_arm(self):
        datar = [
            {"kode": "PDR01", "lokasi": "Padang", "nama": "Ayam Bakar Sing A
Song ", "desc": "Spesialis ayam bakar", "cp": "0831"},
            {"kode": "PDR02", "lokasi": "Padang", "nama": "RM Lamun Ombak",
"desc": "Masakan Padang", "cp": "0832"},
            {"kode": "BTR01", "lokasi": "Batam", "nama": "Wey Wey Seafood ",
"desc": "Spesialis seafood", "cp": "0833"},
            {"kode": "BTR02", "lokasi": "Batam", "nama": "My Garden Resto",
"desc": "Cafe berkonsep taman", "cp": "0834"},
            {"kode": "BLR01", "lokasi": "Bali", "nama": "Kala Uluwatu", "desc":
"Greek-Inspired Restaurant", "cp": "0835"},
            {"kode": "BLR02", "lokasi": "Bali", "nama": "Warung Laota", "desc":
"Hidangan ala Hongkong", "cp": "0836"}
        ]
        for dr in datar:
            self.namar_daftar.append(DataDasar(dr["kode"], dr["lokasi"], dr["nama"],
dr["desc"], dr["cp"]))

```

```

def tampil_data(self, data_list):
    if not data_list:
        print("Belum ada data")
    else:
        print("=" * 106)
        print("\n| {:^5} | {:^10} | {:^35} | {:^25} | {:^14} |".format("Kode",
"Lokasi", "Nama", "Deskripsi", "Contact Person"))
        for item in data_list:
            print("-" * 106)
            print("| {:^5} | {:^10} | {:^35} | {:^25} | {:^14} |".format(item.kode,
item.lokasi, item.nama, item.desc, item.cp))
            print("=" * 106)

def tambah_data(self, kota_choise, data_list):
    kode = input("Kode: ").upper()
    lokasi = kota_choise
    nama = input("Nama: ").capitalize()
    desc = input("Deskripsi: ").capitalize()
    cp = input("Contact Person: ")
    data_list.append(DataDasar(kode, lokasi, nama, desc, cp))
    print("Data berhasil ditambahkan")

def hapus_data(self, data_list):
    kode = input("Masukkan Kode: ").upper()
    for item in data_list:
        if item.kode == kode:
            data_list.remove(item)
            print("Data berhasil dihapus")
            return
    print("Kode tidak valid")

def edit_data(self, item):

```

```

print("Data yang dapat diubah:")
print("a. Lokasi")
print("b. Nama")
print("c. Deskripsi")
print("d. Contact Person")

pilihan = input("Masukkan pilihan yang akan di edit: ")

if pilihan == "a":
    lokasi = input(f"Lokasi [{item.lokasi}]: ").capitalize()
    item.lokasi = lokasi if lokasi else item.lokasi
elif pilihan == "b":
    nama = input(f>Nama [{item.nama}]: ").capitalize()
    item.nama = nama if nama else item.nama
elif pilihan == "c":
    desc = input(f"Deskripsi [{item.desc}]: ").capitalize()
    item.desc = desc if desc else item.desc
elif pilihan == "d":
    cp = input(f"Contact Person [{item.cp}]: ").capitalize()
    item.cp = cp if cp else item.cp
else:
    print("Pilihan tidak valid")

print(f>Data {item.nama} telah diperbarui")

def edit_data_menu(self, data_list):
    kode = input("Masukkan kode yang akan diubah: ").upper()
    for item in data_list:
        if item.kode == kode:
            self.edit_data(item)
            return
    print("Data tidak ditemukan")

```

```

class SistemPariwisata:
    def __init__(self):
        self.users = {} # Menyimpan username dan password
        self.data = Data()
        self.data.data_ahotel()
        self.data.data_awis()
        self.data.data_arm()

    def sign_up(self):
        print("\n=== Sign Up ===")
        username = input("Masukkan username: ")
        password = input("Masukkan password: ")
        role = input("Masukkan peran (admin/user): ").lower()

        if role not in ["admin", "user"]:
            print("Peran tidak valid. Harap masukkan 'admin' atau 'user'.")
            return

        if username in self.users:
            print("Username sudah terdaftar. Silakan gunakan username lain.")
            return

        if not username or not password:
            print("Username dan password tidak boleh kosong!")
            return

        self.users[username] = {"password": password, "role": role}
        print("\n!!!Pendaftaran berhasil. Silakan Login !!!")

    def login(self):
        while True:

```

```

print("\n=== Login Sistem Pariwisata ===")
username = input("Username: ")
password = input("Password: ")

if username in self.users and self.users[username]["password"] ==
password:
    role = self.users[username]["role"]
    if role == "admin":
        print("\n!!!Berhasil login sebagai admin!!!")
        self.menu_admin()
    else:
        print("\n!!!Berhasil login sebagai user!!!")
        self.menu_user()
    break
else:
    print("Login gagal! Username atau password salah. Silahkan coba lagi")
    continue

def menu_admin(self):
    while True:
        print("\n===SELAMAT DATANG DI MENU ADMIN===")
        print("\nMenu Admin")
        print("1. Tampilkan Data")
        print("2. Tambah Data")
        print("3. Hapus Data")
        print("4. Edit Data")
        print("5. Keluar")

        pilihan = input("Silakan masukkan menu (1/2/3/4/5): ")
        if pilihan == "5":
            print("Program selesai. Terimakasih sudah menggunakan program
kami!")

```



```

        exit()
elif pilihan in ["1", "2", "3", "4"]:
    print("\nPilih kota:")
    print("1. Padang")
    print("2. Batam")
    print("3. Bali")
    kota_choice = input("Masukkan pilihan kota (1/2/3): ")

    if kota_choice == "1":
        kota = "Padang"
    elif kota_choice == "2":
        kota = "Batam"
    elif kota_choice == "3":
        kota = "Bali"
    else:
        print("Pilihan tidak valid.")
        continue
while True:
    print("\nData yang dapat dipilih:")
    print("1. Hotel")
    print("2. Wisata")
    print("3. Rumah Makan")
    print("x. Kembali ke menu utama")
    sub_pilihan = input("Silakan masukkan pilihan: ").lower()

    if sub_pilihan == "x":
        break
    if sub_pilihan == "1":
        data_list = self.data.namah_daftar
    elif sub_pilihan == "2":
        data_list = self.data.namaw_daftar
    elif sub_pilihan == "3":

```

```

        data_list = self.data.namar_daftar
    else:
        print("Pilihan tidak valid.")
        continue

    filtered_data = [item for item in data_list if item.lokasi.lower() ==
kota.lower()]

    if pilihan == "1":
        self.data.tampil_data(filtered_data)
    elif pilihan == "2":
        self.data.tampil_data(filtered_data)
        self.data.tambah_data(kota, data_list)
    elif pilihan == "3":
        self.data.tampil_data(filtered_data)
        self.data.hapus_data(data_list)
    elif pilihan == "4":
        self.data.tampil_data(filtered_data)
        self.data.edit_data_menu(data_list)
    lanjut = input("Apakah ingin lanjut? (y/n): ").lower()
    if lanjut != "y":
        break

    else:
        print("Pilihan tidak valid. Silahkan coba lagi")

def menu_user(self):
    while True:
        print("\n===SELAMAT DATANG DI MENU USER===")
        print("\nMenu User")
        print("1. Tampilkan Data Berdasarkan Kota")
        print("2. Keluar")

```

```

pilihan = input("Silakan masukkan menu (1/2): ")

if pilihan == "1":
    print("\nPilih kota:")
    print("1. Padang")
    print("2. Batam")
    print("3. Bali")
    kota_choice = input("Masukkan pilihan kota (1/2/3): ")

    if kota_choice == "1":
        kota = "Padang"
    elif kota_choice == "2":
        kota = "Batam"
    elif kota_choice == "3":
        kota = "Bali"
    else:
        print("Pilihan tidak valid.")
        continue

    print("\nData apa yang ingin ditampilkan?")
    print("1. Hotel")
    print("2. Wisata")
    print("3. Rumah Makan")
    print("x. Kembali ke menu utama")
    sub_pilihan = input("Silakan masukkan pilihan: ").lower()

    if sub_pilihan == "1":
        self.data.tampil_data([h for h in self.data.namah_daftar if
h.lokasi.lower() == kota.lower()])
    elif sub_pilihan == "2":

```

```

        self.data.tampil_data([w for w in self.data.namaw_daftar if
w.lokasi.lower() == kota.lower()])
        elif sub_pilihan == "3":
            self.data.tampil_data([r for r in self.data.namar_daftar if
r.lokasi.lower() == kota.lower()])
            elif sub_pilihan == "x":
                continue
            else:
                print("Pilihan tidak tersedia.")
                continue

        lanjut = input("Apakah ingin lanjut? (y/n): ").lower()
        if lanjut != "y":
            print("Program selesai. Terimakasih sudah menggunakan program
kami!")
            break

        elif pilihan == "2":
            print("Program selesai. Terimakasih sudah menggunakan program
kami!")
            exit()
        else:
            print("Pilihan tidak valid. Silakan coba lagi.")

def tampilkan_data_kota(self, kota):
    print(f"Data untuk kota {kota}:")
    if kota == "Padang":
        self.data.tampil_data(self.data.namah_daftar)
        self.data.tampil_data(self.data.namaw_daftar)
        self.data.tampil_data(self.data.namar_daftar)
    elif kota == "Batam":
        self.data.tampil_data(self.data.namah_daftar)

```

```

        self.data.tampil_data(self.data.namaw_daftar)
        self.data.tampil_data(self.data.namar_daftar)
    elif kota == "Bali":
        self.data.tampil_data(self.data.namah_daftar)
        self.data.tampil_data(self.data.namaw_daftar)
        self.data.tampil_data(self.data.namar_daftar)

def get_data_list(self, pilihan, kota):
    if pilihan == "1":
        return [h for h in self.data.namah_daftar if h.lokasi.lower() ==
kota.lower()]
    elif pilihan == "2":
        return [w for w in self.data.namaw_daftar if w.lokasi.lower() ==
kota.lower()]
    elif pilihan == "3":
        return [r for r in self.data.namar_daftar if r.lokasi.lower() == kota.lower()]

def main():
    sistem = SistemPariwisata()
    while True:
        print("\n=== Selamat Datang di Sistem Objek Pariwisata ===")
        print("Kota Padang Batam Bali")
        print("\n===MENU UTAMA===")
        print("1. Sign Up")
        print("2. Login")
        print("3. Keluar")
        pilihan = input("Silakan masukkan pilihan (1/2/3): ")

        if pilihan == "1":
            sistem.sign_up()
        elif pilihan == "2":

```

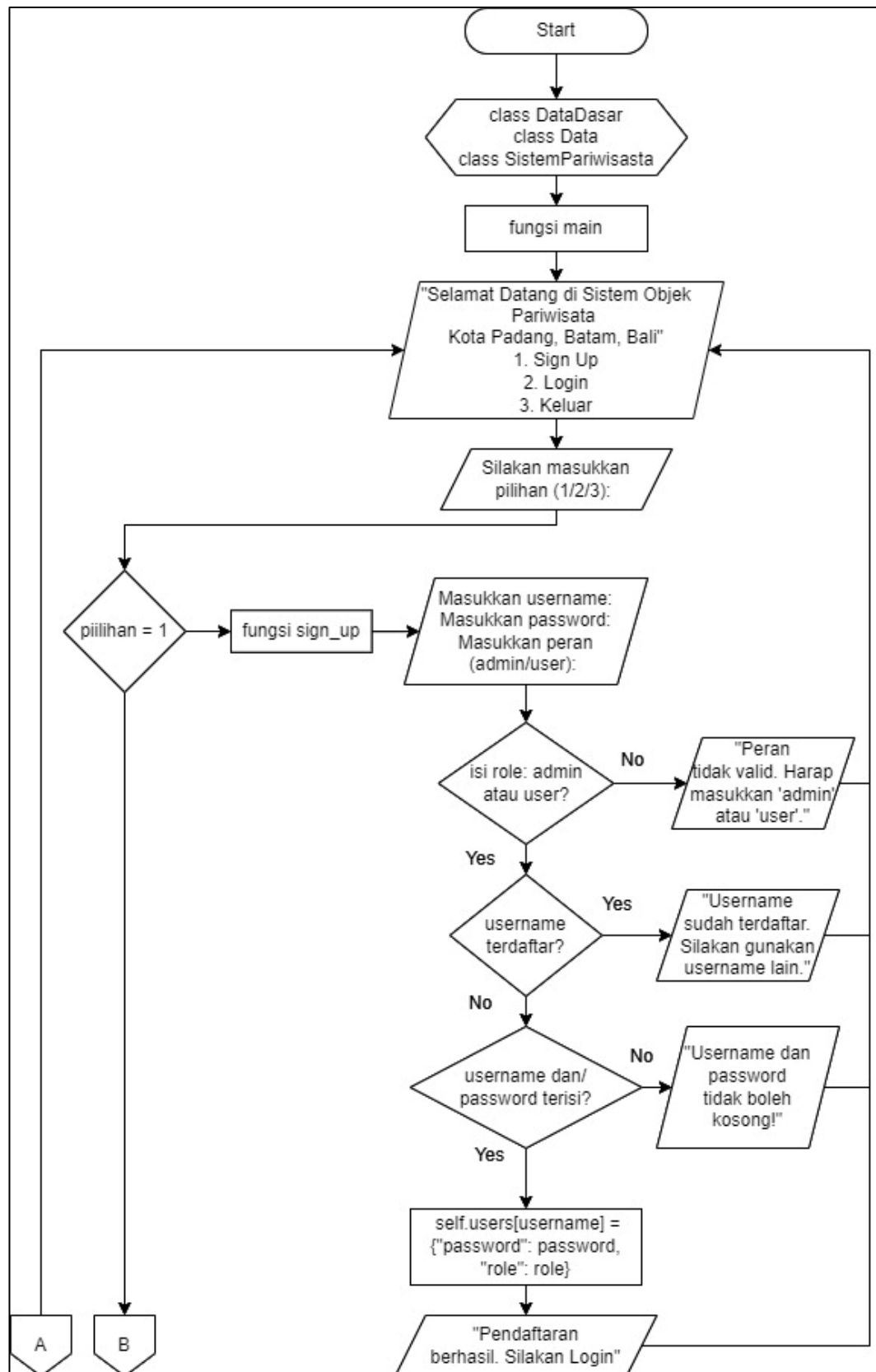
```
sistem.login()
elif pilihan == "3":
    print("Program selesai. Terimakasih sudah menggunakan program kami!")
    break
else:
    print("Pilihan tidak valid. Silakan coba lagi.")

if __name__ == "__main__":
    main()
```

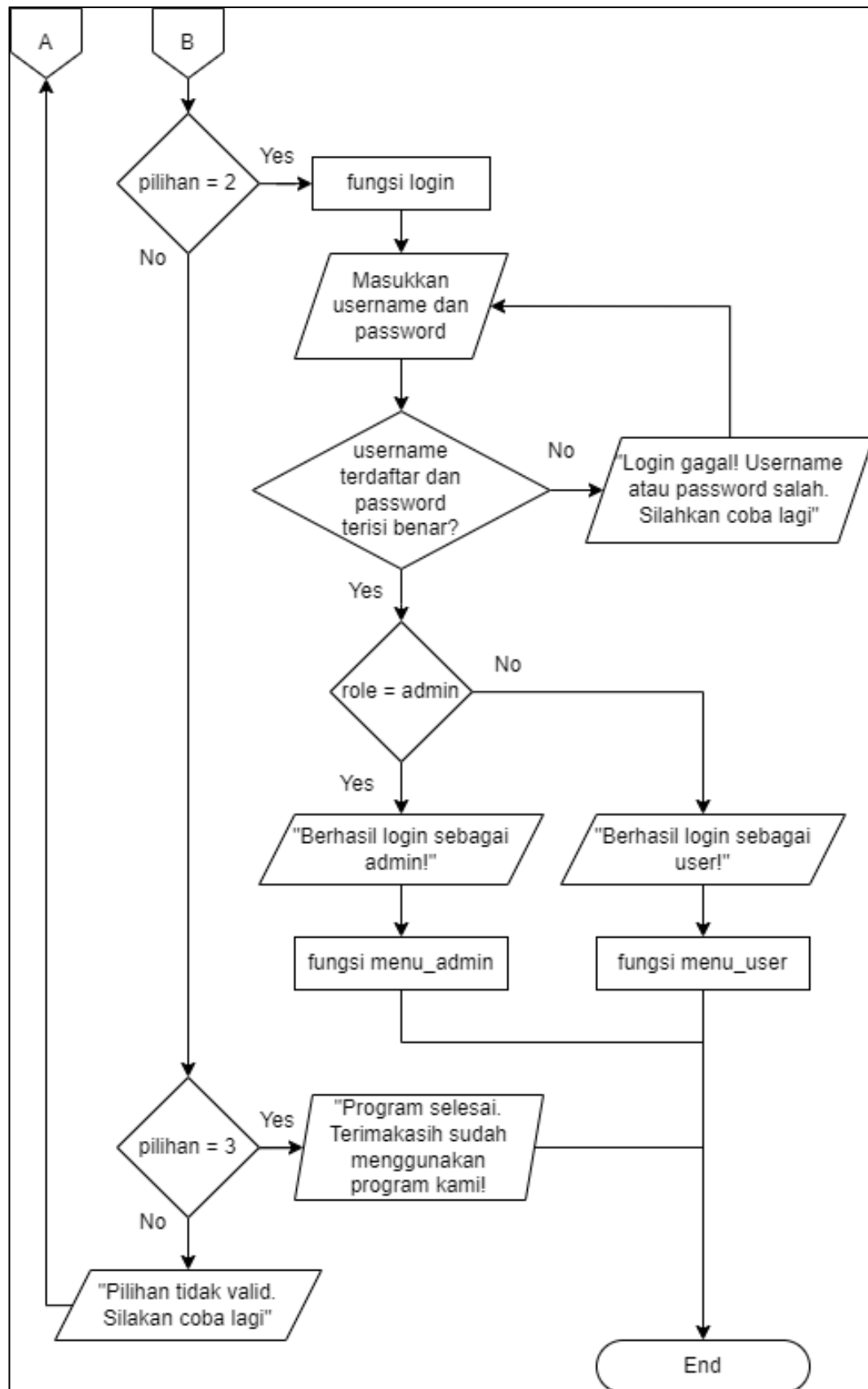
### **2.3    *Flowchart***

Flowchart ini menggambarkan alur jalannya program sebuah sistem pariwisata dengan 3 jenis menu utama, yaitu menu login, menu akses sebagai admin, dan menu akses sebagai user.

### 2.3.1 Flowchart Menu Login



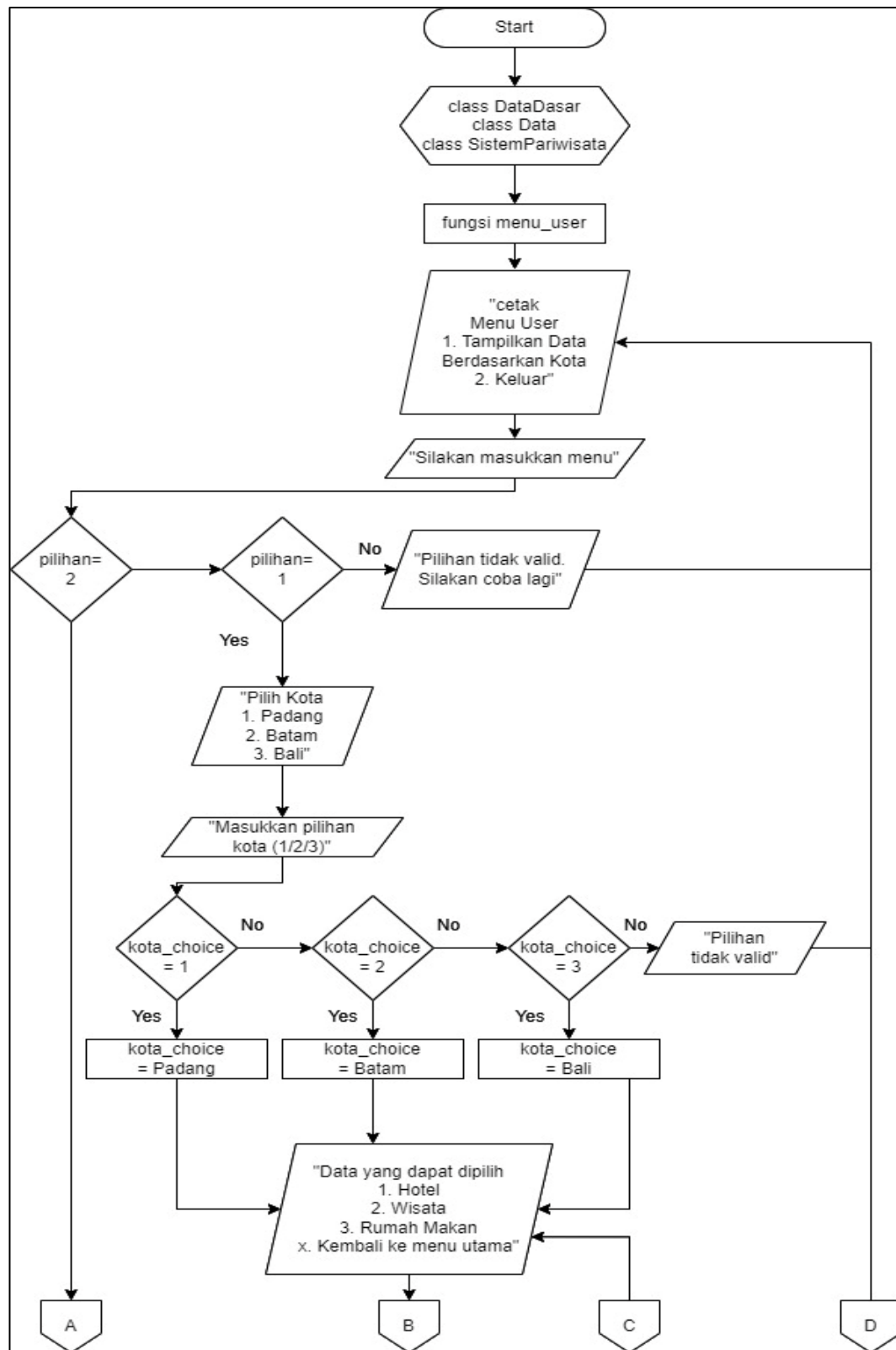
Gambar 2.1 Flowchart Login



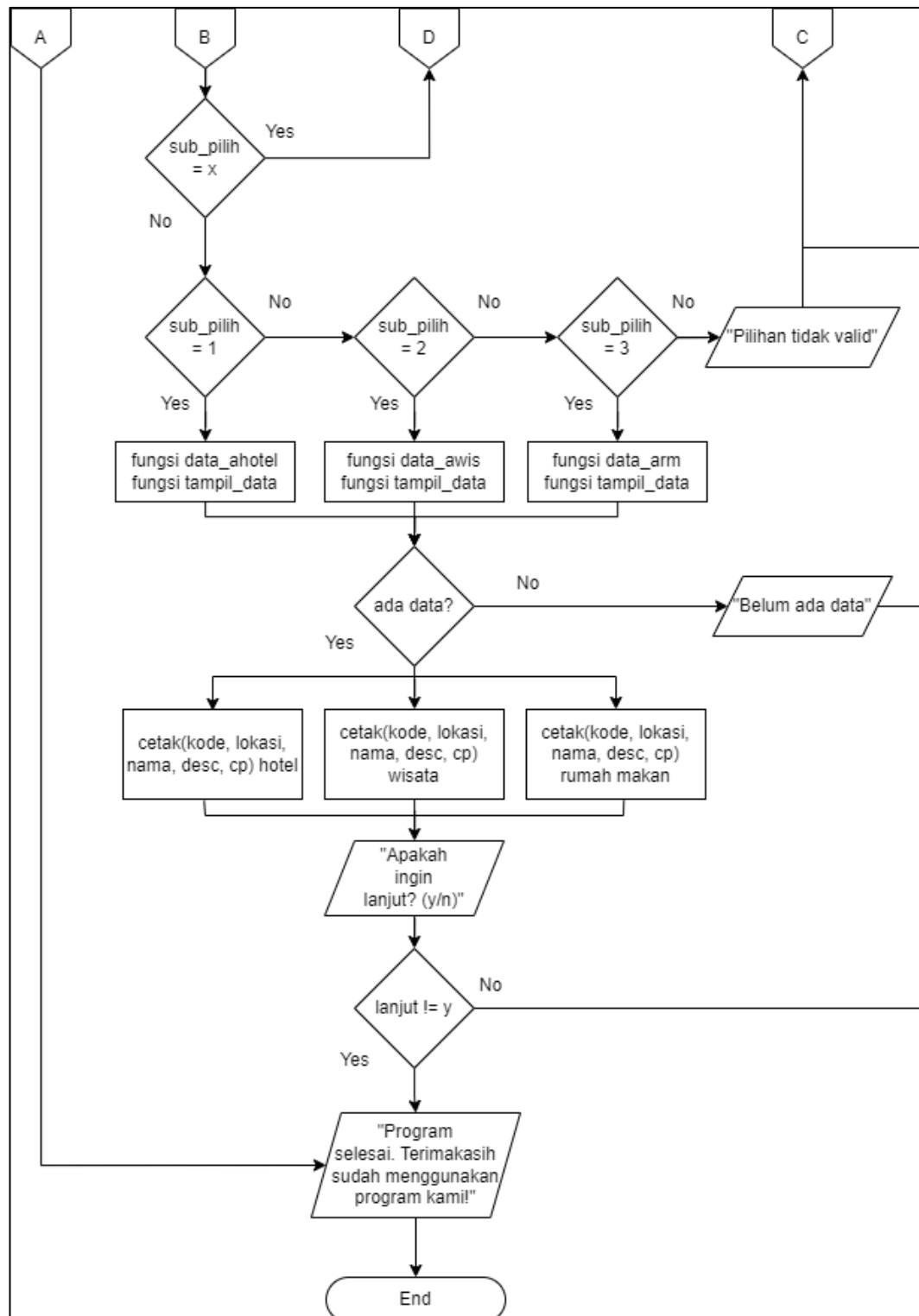
**Gambar 2.2** Flowchart Login (Lanjutan)



### 2.3.2 Flowchart Menu User

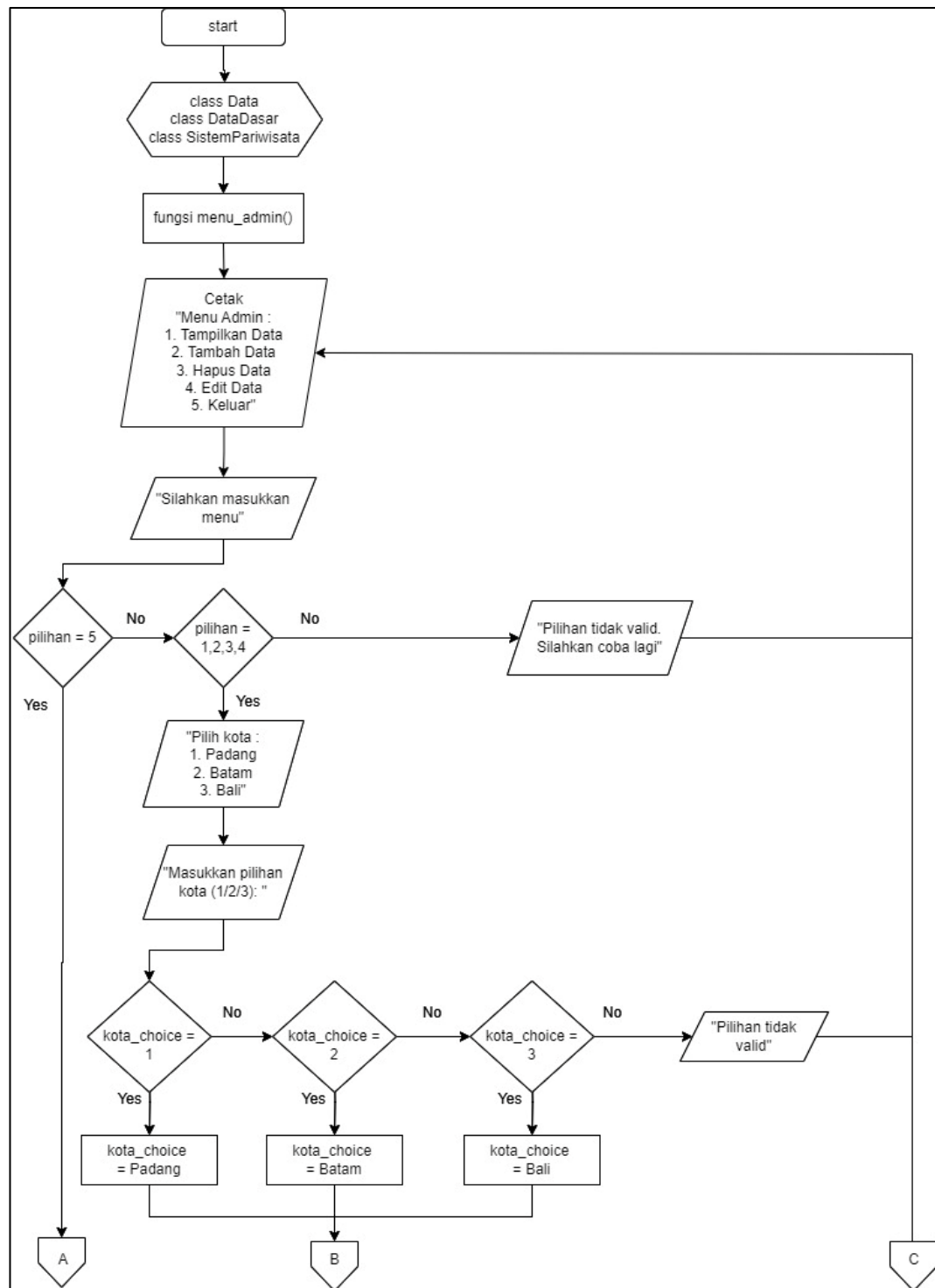


**Gambar 2.3** Flowchart User

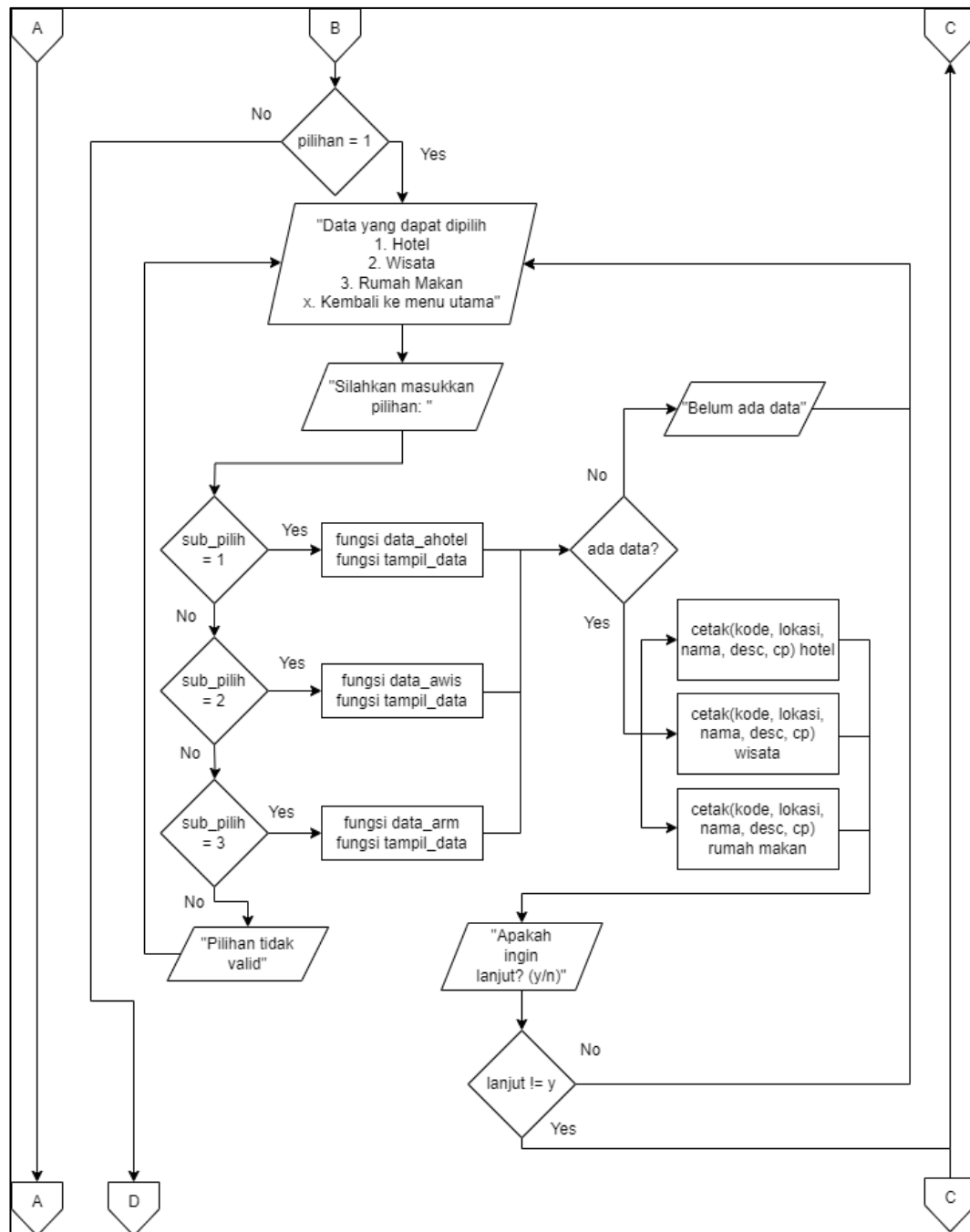


**Gambar 2.4** Flowchart User (Lanjutan)

### 2.3.3 Flowchart Menu Admin

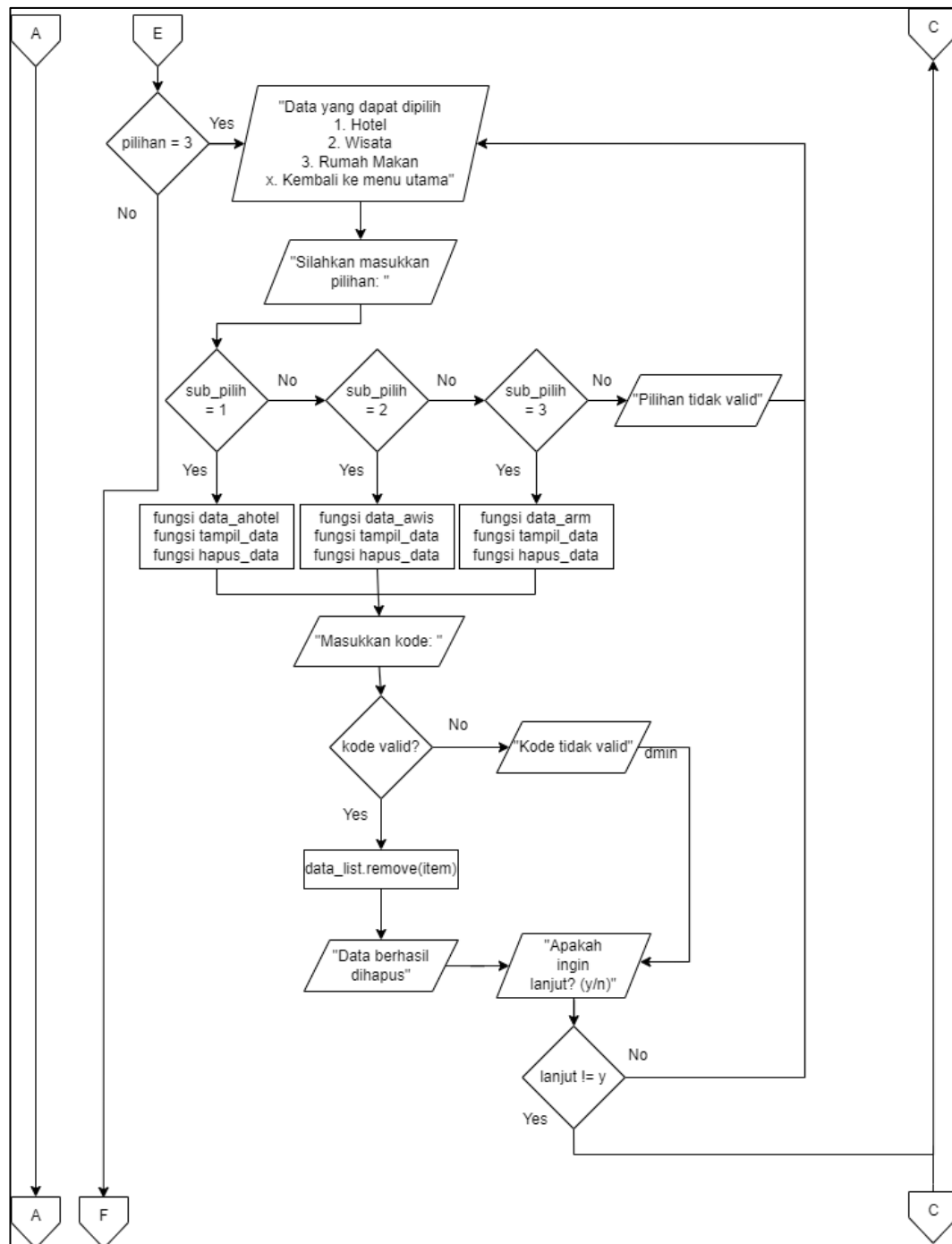


**Gambar 2.5** Flowchart Admin

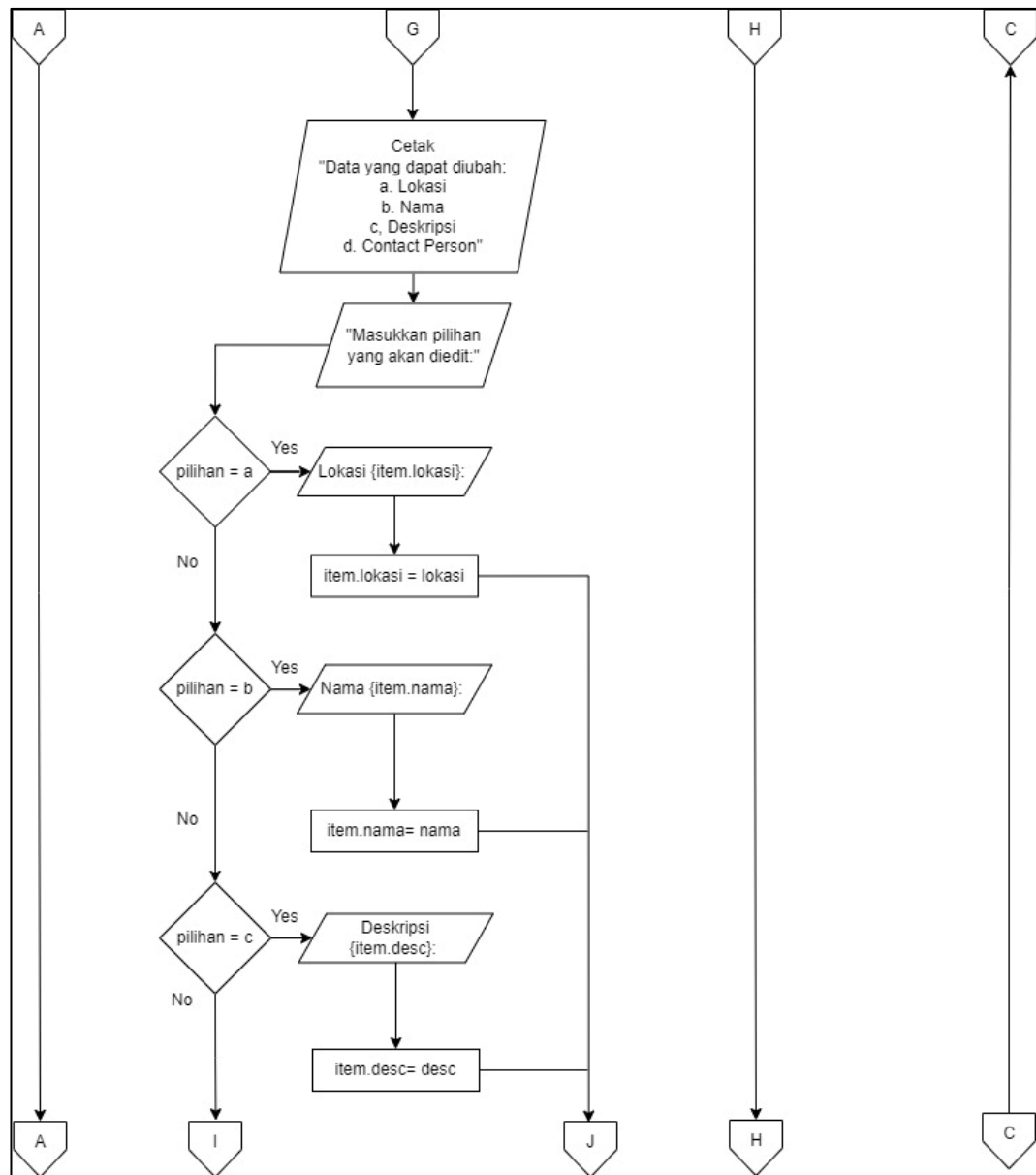


**Gambar 2.6** Flowchart Admin (Lanjutan)

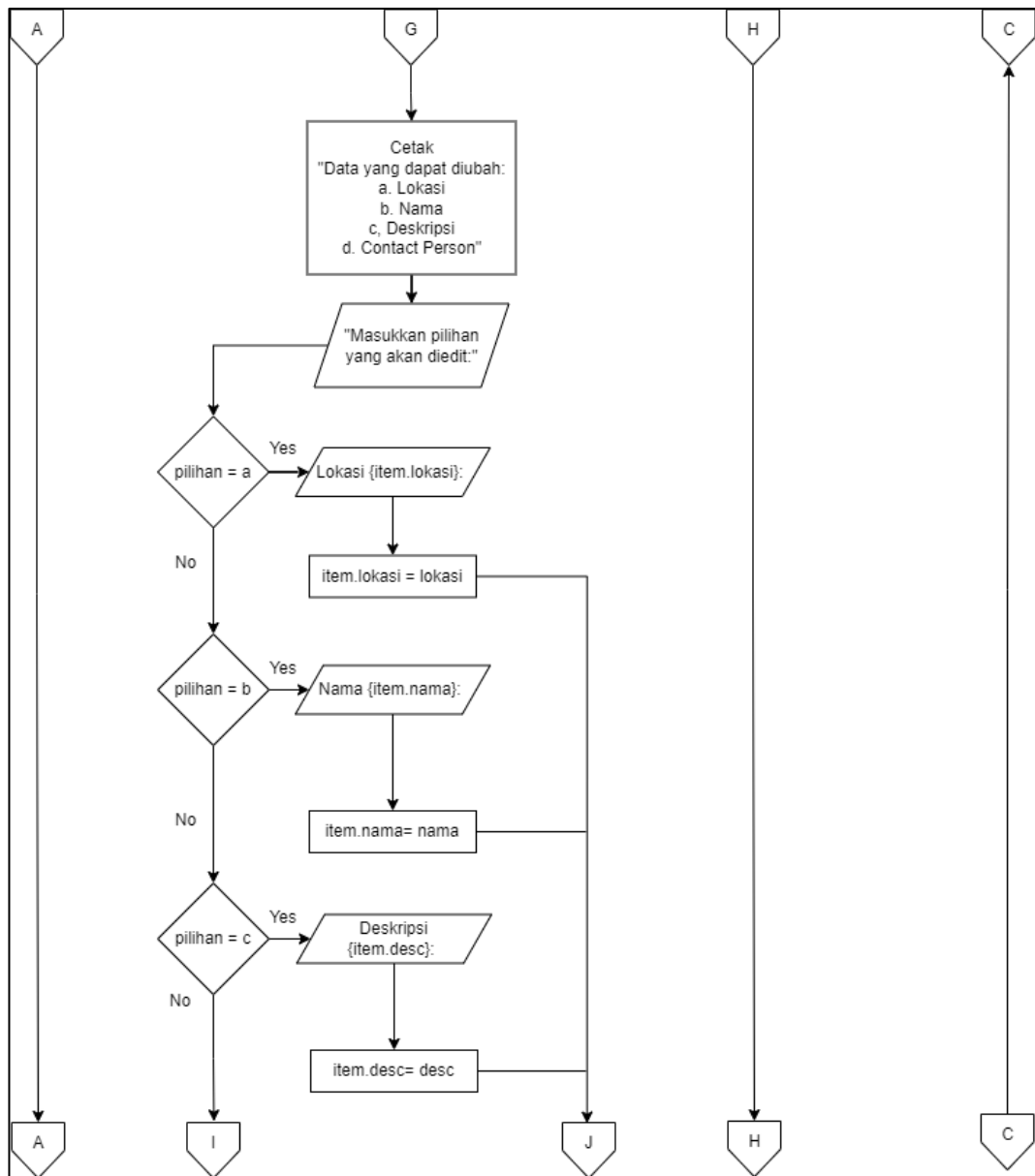




**Gambar 2.8** Flowchart Admin (Lanjutan)

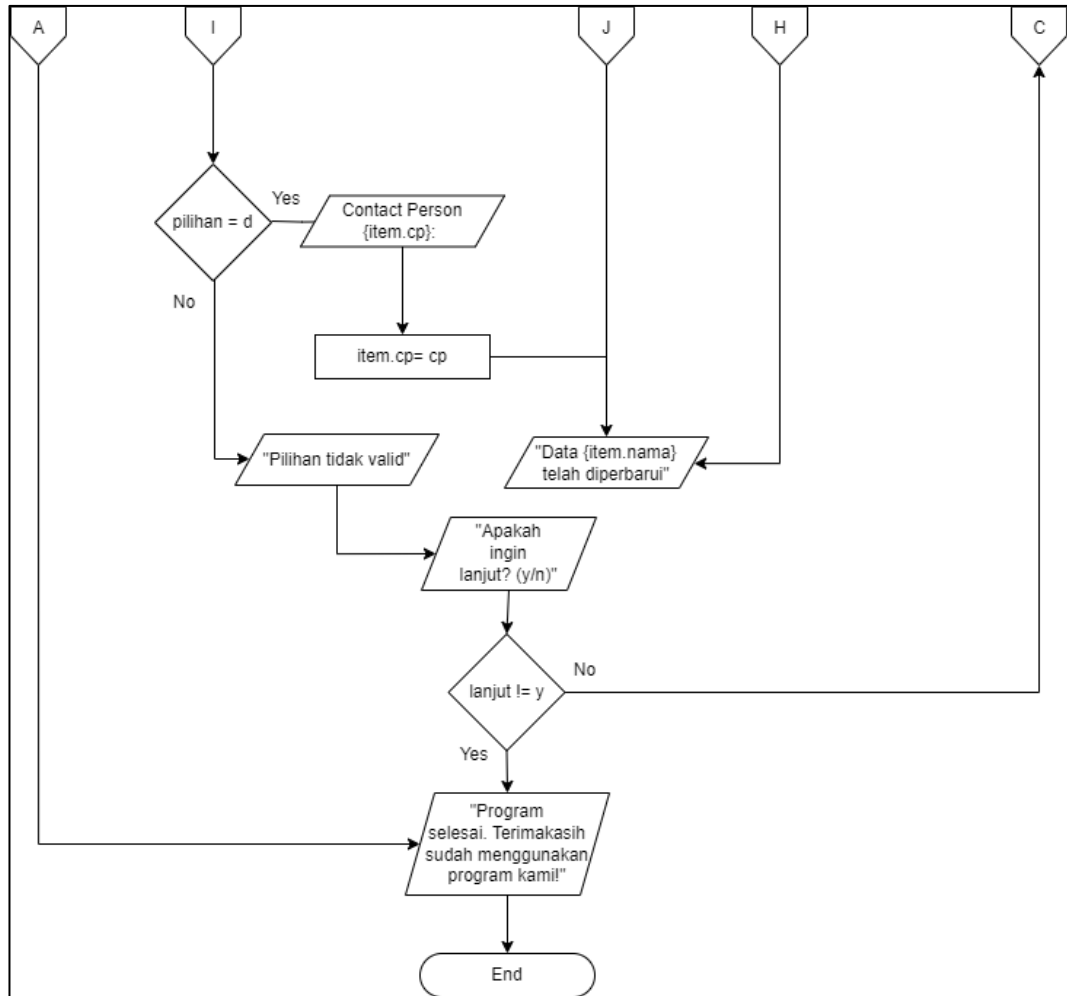


**Gambar 2.9** *Flowchart Admin (Lanjutan)*



**Gambar 2.10** *Flowchart Admin (Lanjutan)*





**Gambar 2.11** *Flowchart Admin (Lanjutan)*

## **BAB III**

### **ANALISIS PROGRAM**

Bab ini memuat analisis program terhadap modul-modul praktikum yang telah dipelajari. Analisis ini dilakukan untuk memperdalam pemahaman teknis, penyusunan sistem yang fungsional, terstruktur, dan menghasilkan output yang diharapkan

#### **3.1 Analisis Penerapan Modul pada Program**

Dalam praktik Pemrograman Komputer dengan basis bahasa pemrograman Python, dipelajari beberapa modul dasar pengenalan dan penggunaan Python dalam pengembangan suatu program proses atau aplikasi yang efektif, efisien dan ramah pengguna. Setiap modul dirancang untuk memperkenalkan dan melatih praktikan terkait konsep dasar pemrograman secara bertahap, yang kemudian diintegrasikan dengan studi kasus nyata. Studi kasus yang diangkat adalah pengelolaan sistem informasi pariwisata hotel, tempat wisata, dan rumah makan di tiga kota, yaitu Padang, Batam, dan Bali, serta sistem login multi-peran (admin dan pengguna).

Sistem ini dirancang sistematis dengan memanfaatkan kelima modul untuk menciptakan struktur program. Modul satu sebagai pengenalan *syntax* dan dasar program. Modul dua sebagai alur pengambilan logika berdasarkan keputusan. Modul tiga sebagai sistem perulangan digunakan dalam fitur pengolahan data dan navigasi menu. Modul empat sebagai membuat dan mengolah data yang tersedia. Modul lima sebagai pembentuk kerangka yang menggabungkan semua modul dan konsep agar menciptakan suatu program yang terorganisir dan siap digunakan.

##### **3.1.1 Analisis Modul 1 – Pengenalan Komputer dan Dasar Bahasa Python**

Modul ini membahas pengantar dasar komputer dan pemrograman, mencakup pengertian komputer, sejarah, sistem kerja, prinsip kerja, jaringan, sistem operasi, serta dasar-dasar algoritma dan *flowchart*. Dijelaskan pula struktur dasar

bahasa Python, tipe data, kata kunci (*keywords*), *identifier*, prosedur *input/output*, serta pengenalan *tools* pemrograman. *Flowchart* dan algoritma digunakan untuk menggambarkan proses eksekusi program secara terstruktur, membantu menyelesaikan masalah secara logis dan sistematis sebelum diterjemahkan ke dalam kode.

Python dikenalkan sebagai bahasa pemrograman tingkat tinggi yang mudah digunakan, bersifat interaktif, berorientasi objek, dan serbaguna. Python menyediakan fitur seperti *identifier* dan *keyword* untuk membuat kode yang terstruktur dan mudah dibaca. Pemahaman terhadap konsep-konsep ini menjadi dasar penting bagi pemula untuk mulai menulis program, termasuk dalam membangun aplikasi sederhana seperti sistem informasi pariwisata berbasis teks yang dikembangkan pada studi kasus ini.

Dalam modul ini juga dipelajari tipe-tipe data pada Python, yaitu:

1. *String* : menyatakan nilai teks, seperti huruf, tanda baca, dan karakter spesial lainnya. Data *string* diapit oleh tanda petik, baik itu petik satu ( ' ') atau petik dua ( " ").

```
nama = input("Nama: ")
```

**Gambar 3.1** Contoh *string* dalam program

Berdasarkan **Gambar 3.1**, “nama” adalah variabel bertipe *string* yang akan didefinisikan dari perintah “input”. Dalam perintah ini, terdapat teks yang diapit oleh petik dua ( “ ” ) untuk memberikan arahan apa yang harus diinputkan dan dapat diisi dengan teks atau angka untuk oleh pengguna. Program meminta pengguna mengetik nama, lalu menyimpannya ke dalam variabel nama.

2. *Boolean* : menyatakan nilai benar (*True*) atau salah (*False*)

```
if username in self.users and self.users[username]["password"] == password:
```

**Gambar 3.2** Contoh *Boolean*

Berdasarkan **Gambar 3.2**, program tersebut adalah penerapan *Boolean* dengan logika di mana kondisi yang harus terpenuhi jika variabel “username” dan “password” yang dimasukkan terdaftar dalam “self.users”. Apabila memenuhi, maka kondisi akan dianggap *True*. “username in self.user” bernilai *True* jika *user* ditemukan. “self.users[username][“password”] == password” berfungsi untuk memeriksa kesesuaian *password*. Kedua kondisi tersebut harus bernilai *True* agar program dapat berlanjut ke proses *login*.

3. Variabel : simbol untuk mewakili nilai tertentu. Penulisannya tidak boleh dipisah oleh spasi. Untuk variabel yang terdiri dari 2 suku kata, dipisahkan dengan simbol *underscore* (\_).

```
kota_choice = input("Masukkan pilihan kota (1/2/3): ")
```

**Gambar 3.3** Contoh Variabel

Berdasarkan **Gambar 3.3**, “kota\_choice” adalah variabel yang didefinisikan oleh *input* pengguna, yang kali ini merupakan *input string* dan diarahkan untuk memilih antara 1/2/3. Variable “kota\_choice” menyimpan *input* pilihan kota dari pengguna. Sesuai aturan Python, nama variabel menggunakan *underscore* (\_) karena variabel tersebut terdiri dari lebih dari satu suku kata.

Fitur-fitur Python yang digunakan pada program.

1. *Print* : digunakan untuk menampilkan *output* yang dihasilkan dari eksekusi program yang ingin ditampilkan

```
print("\n=== Selamat Datang di Sistem Objek Pariwisata ===")
print("Kota Padang Batam Bali")
print("\n==MENU UTAMA==")
print("1. Sign Up")
print("2. Login")
print("3. Keluar")
```

**Gambar 3.4** Contoh Perintah *Print*

Berdasarkan **Gambar 3.4**, Perintah "print()" dalam program tersebut akan mencetak kalimat yang berada di dalam kurung dan di dalam tanda petik (" ") dan menampilkannya ke layar.

2. *Input* : digunakan untuk memasukkan data yang dibutuhkan dalam eksekusi program

```
pilihan = input("Silakan masukkan pilihan (1/2/3): ")
```

**Gambar 3.5** Contoh Perintah *Input*

Berdasarkan **Gambar 3.5**, Saat baris "input()" dijalankan, program akan menampilkan teks "Silakan masukkan pilihan (1/2/3):" di terminal dan menunggu *input* dari pengguna. Nilai (1/2/3) yang diketik oleh pengguna akan disimpan dan menentukan proses selanjutnya. Kemudian proses selanjutnya yang ingin dilakukan pengguna akan berjalan.

### 3.1.2 Analisis Modul 2 – Konsep Keputusan

Konsep keputusan dalam pemrograman merujuk pada proses pengambilan Keputusan yang dilakukan oleh program untuk menentukan Tindakan yang akan diambil berdasarkan suatu kondisi. Salah satu ciri khas pemrograman terletak pada kemampuannya untuk membuat struktur logika yang kemudian dimanfaatkan agar pemrogram dapat membuat percabangan alur program. Dengan konsep ini, pemrogram dapat mengatur apakah sebuah perintah akan dijalankan atau tidak. Pada Python, ada beberapa *statement/kondisi*. Diantaranya :

1. Kondisi *IF*

*Statement* ini digunakan untuk menguji suatu kondisi Dimana jika kondisi tersebut bernilai benar (*True*) maka pernyataan atau program dibawah blok *IF* tersebut akan dieksekusi oleh komputer. Namun, jika kondisi tersebut bernilai salah maka blok program tersebut akan diabaikan.

```
if role not in ["admin", "user"]:  
    print("Peran tidak valid. Harap masukkan 'admin' atau 'user'.")  
    return
```

**Gambar 3.6** Contoh Kondisi *IF*

Berdasarkan **Gambar 3.6**, pengondisian *IF* berlaku apabila variabel *role* tidak terpenuhi dalam peran *admin* atau *user* maka program yang berada di bawah blok *IF* tersebut akan dieksekusi. “role not in [“admin”, “user”]” memeriksa apakah nilai *role* tidak termasuk dalam *list* [“admin”, “user”]. Jika kondisi benar (*True*), maka blok “print()” akan dieksekusi. Program ini digunakan untuk validasi *input* peran, agar hanya bisa *admin* atau *user* yang dapat melanjutkan program.

## 2. Kondisi *Else If* atau *ELIF*

*Statement* ini digunakan untuk menguji suatu pernyataan dengan proses pemilihan kondisi dapat dilakukan dengan beberapa kondisi lainnya atau kondisi alternatif. Jika kondisi tersebut bernilai benar (*True*) maka pernyataan atau program di bawah blok *IF* akan dieksekusi oleh komputer. Sedangkan jika kondisi tersebut bernilai salah (*False*) maka komputer akan mengeksekusi pernyataan atau program di bawah blok *IF* berikutnya.

```
if kota_choice == "1":  
    kota = "Padang"  
elif kota_choice == "2":  
    kota = "Batam"  
elif kota_choice == "3":  
    kota = "Bali"
```

**Gambar 3.7** Contoh Kondisi *ELIF*

Berdasarkan **Gambar 3.7**, kondisi yang terdapat pada program adalah apabila kondisi *IF* tidak terpenuhi maka akan lanjut ke kondisi selanjutnya (*ELIF*) yang mana memiliki alur perintah atau pengondisian lain yang berbeda dari kondisi lainnya. Pertama, akan diperiksa apakah “kota\_choice” == 1”. Jika iya, maka isi variabel “kota” adalah “Padang”. Jika tidak, maka Python akan mengevaluasi *ELIF*.

*ELIF* hanya dievaluasi jika *IF* sebelumnya bernilai salah. Jika pengguna memilih “2” maka isi dari variabel “kota” adalah “Batam”.

### 3. Kondisi *ELSE*

*Statement* ini digunakan untuk mengeksekusi kondisi selain yang sudah diperiksa sebelumnya oleh *IF* dan *ELIF* (opsional). Artinya, jika kondisi *IF* atau *ELIF* tidak terpenuhi, maka program *ELSE* akan berjalan.

```
if kota_choice == "1":
    kota = "Padang"
elif kota_choice == "2":
    kota = "Batam"
elif kota_choice == "3":
    kota = "Bali"
else:
    print("Pilihan tidak valid.")
```

**Gambar 3.8** Contoh Kondisi *ELSE*

Berdasarkan **Gambar 3.8**, kondisi *ELSE* merupakan kondisi lain atau tidak terpenuhi dari *IF* dan *ELIF* sebelumnya. *ELSE* digunakan untuk menutup kemungkinan terjadi *loop* atau *error* dari kondisi *IF* dan/atau *ELIF* pada program. Jika pengguna memasukkan pilihan selain yang ada pada *IF* atau *ELIF*, maka teks “Pilihan tidak valid.” ditampilkan.

#### 3.1.3 Analisis Modul 3 – Perulangan (*loop*)

Konsep perulangan merupakan perulangan suatu proses guna menghasilkan suatu urutan *output*. Konsep ini menjadikan komputer dapat diinstruksikan untuk berkerja pada hal yang sama secara berulang-ulang. Pada konsep ini, perulangan program dapat dilakukan sejumlah kali hingga tercapainya suatu kondisi berhenti yang telah ditentukan. Pernyataan atau perintah *loop* pada Python terdapat dua *statement* yaitu *While Statement* dan *For Statement*

## 1. *While Statement*

Perulangan *while* digunakan untuk mengulang sekumpulan kode program/perintah program selama kondisi yang diberikan bernilai benar (*True*). Kondisi ini akan diperiksa sebelum *loop* dijalankan, dan jika benar, kode perintah dibawahnya akan dieksekusi. Jenis perulangan ini digunakan saat jumlah pengulangan yang dilakukan belum diketahui dan akan berhenti saat kondisi tertentu terpenuhi.

```
def login(self):
    while True:
        print("\n== Login Sistem Pariwisata ==")
        username = input("Username: ")
        password = input("Password: ")

        if username in self.users and self.users[username]["password"] == password:
            role = self.users[username]["role"]
            if role == "admin":
                print("\n!!!Berhasil login sebagai admin!!!")
                self.menu_admin()
            else:
                print("\n!!!Berhasil login sebagai user!!!")
                self.menu_user()
            break
        else:
            print("Login gagal! Username atau password salah. Silahkan coba lagi")
            continue
```

**Gambar 3.9** Contoh *While Statement*

Kondisi pada **Gambar 3.9** adalah apabila penginputan variabel di dalam fungsi login gagal, maka kondisi di dalam *While* bernilai benar (*True*) maka proses login akan terus berulang dan apabila pengguna berhasil memasukkan variabel yang benar, maka pengulangan akan berhenti yang ditandai dengan pengondisian *break*.

## 2. *For Statement*

Perulangan digunakan untuk mengulang kode program apabila jumlah perulangan dalam suatu proses telah diketahui di awal. Syarat penggunaannya adalah harus ada urutan yang dapat diiterasi, seperti *list* atau *string*. Perulangan akan berlangsung hingga dicapai item terakhir dari urutan.



```
for item in data_list:
    print("-" * 106)
    print("{:^5} | {:^10} | {:^35} | {:^25} | {:^14} |".format(item.kode, item.lokasi, item.nama, item.desc, item.cp))
    print("=" * 106)
```

**Gambar 3.10** Contoh *For Statement*

Kondisi pada **Gambar 3.10** adalah perulangan perintah *print* dengan keterangan di dalam perintah tersebut akan terus berulang hingga semua data yang dituju dicetak semua dan proses perulangan berhenti. “for item in data\_list” artinya perulangan terhadap setiap elemen dalam “data\_list”. Ini digunakan untuk menampilkan data yang sudah disimpan

Dalam konsep perulangan terdapat kontrol-kontrol untuk membantu program agar lebih fleksibel, adaptif terhadap kondisi, dan efisiensi perulangan untuk kasus khusus. Berikut adalah jenis-jenis nya :

#### 1. *Break*

Digunakan untuk mengubah aliran di dalam suatu *loop*. *Break* akan menghentikan iterasi yang sedang berlangsung dan mengeksekusi pernyataan berikutnya di luar *loop*.

```
if lanjut != "y":
    print("Program selesai. Terimakasih sudah menggunakan program kami!")
    break
```

**Gambar 3.11** Contoh *Break*

Pada **Gambar 3.11**, kondisi *break* akan berjalan saat keputusan tidak sama dengan “y”, yang mana akan menjalankan perintah cetak dan perulangan dihentikan dengan kondisi *break*.

#### 2. *Continue*

Digunakan untuk menghentikan perintah yang sedang berlangsung di dalam suatu *loop* dan memindahkan kendali ke kode perintah berikutnya di dalam *loop* tersebut.

```
else:  
    print("Pilihan tidak valid.")  
    continue
```

**Gambar 3.12** Contoh *Continue*

Pada **Gambar 3.12**, *continue* akan berjalan dalam kondisi *ELSE* di mana program dirancang untuk melewati iterasi sekarang dan melanjutkan ke iterasi berikutnya. Pada contoh, setelah kondisi *ELSE*, *continue* akan membuat program melewati sisa kode dalam *loop* saat ini, lalu melanjutkan ke iterasi berikutnya. Ini dilakukan agar program tidak mengeksekusi kode di bawahnya jika kondisi tertentu tidak valid.

#### 3.1.4 Analisis Modul 4 – Array

*Array* merupakan sekumpulan variabel yang memiliki tipe data yang sama dan dinyatakan dengan nama yang sama secara berurutan. Artinya, *array* memungkinkan program untuk menyimpan data maupun referensi objek dalam jumlah banyak. *Array* adalah variabel tunggal yang dapat diakses atau dimodifikasi dengan menggunakan indeks. Indeks adalah fungsi yang digunakan untuk mengembalikan indeks kemunculan pertama yang nilainya disebutkan dalam argumen.

Dalam python, ada beragam bentuk format *array*. Seperti *list*, *tuple*, *set*, dan *dictionary*.

##### 1. *List*

*List* adalah struktur data yang menyimpan koleksi data dengan tipe data yang bisa saja berbeda, misalnya *string*, *integer*, dan sebagainya. *List* bersifat *mutable* yang artinya data dalam *list* dapat diubah. Pengguna dapat melakukan perubahan data seperti menambah, menghapus, atau mengubah elemen-elemen yang terdapat di dalam *list*. Objek *list* dibuat dengan tanda '[' ]'.

```
self.namah_daftar = []  
self.namaw_daftar = []  
self.namar_daftar = []
```

**Gambar 3.13** Contoh *List*

Pada **Gambar 3.13**, *list* ini berisikan data hotel, tempat wisata, dan rumah makan yang akan dikelola nantinya. Setiap elemen dalam *list* adalah sebuah *dictionary* yang berisi informasi seperti kode, lokasi, nama hotel, deskripsi, dan *contact person*.

## 2. *Dictionary*

*Dictionary* berisi *key* dan *value* pada setiap urutannya. Setiap *key* dipisahkan dari *value*-nya oleh titik dua (:), item dipisahkan koma, dan semuanya tertutup dalam kurung kurawal '{ }'.

```
self.users[username] = {"password": password, "role": role}
```

**Gambar 3.14** Contoh *Dictionary*

Pada **Gambar 3.14**, ini merupakan salah satu pasangan *key-value* yang dimana *key* "password" memiliki *value* variabel "password" yang mana pada bagian ini merupakan dimasukkannya data *sign-up* ke dalam *dictionary*. Data pengguna disimpan dalam *dictionary* "self.users", dengan "username" sebagai *key* utama. Nilainya berupa *dictionary* lagi yang berisi dua pasang informasi, yaitu "password" dan "role". Struktur ini sangat berguna dalam manajemen data *login* multi-peran karena fleksibel dan mudah diakses.

## 3. *Dictionary* dalam *list*

*Dictionary* dalam *list* adalah struktur data kompleks dalam Python yang menggunakan tipe data *list* dan tipe data *dictionary*. Dalam hal ini, *list* menyimpan beberapa elemen *dictionary* yang masing-masing *dictionary* itu menyimpan data

dalam bentuk *key* dan *value*. Konsep ini digunakan untuk menyimpan data yang terstruktur dalam satu variabel dan mudah di proses pada *loop*.

```
data_ahotel(self):
    datah = [
        {"kode": "PDH01", "lokasi": "Padang", "nama": "Mercure Padang", "desc": "Western Food", "cp": "0811"},
        {"kode": "PDH02", "lokasi": "Padang", "nama": "Grand Zuri Padang", "desc": "Tengah Kota", "cp": "0812"},
        {"kode": "BTH01", "lokasi": "Batam", "nama": "Batam Marriott Hotel Harbour Bay", "desc": "Sea View", "cp": "0813"},
        {"kode": "BTH02", "lokasi": "Batam", "nama": "The Music Hotel", "desc": "Budget-friendly", "cp": "0814"},
        {"kode": "BLH01", "lokasi": "Bali", "nama": "Neo Kuta Jelantik Hotel", "desc": "Budget-friendly", "cp": "0815"},
        {"kode": "BLH02", "lokasi": "Bali", "nama": "Hotel Siesta Legian", "desc": "Dekat Pantai", "cp": "0816"},
    ]
```

**Gambar 3.15** Contoh *Dictionary* dalam *list*

Berdasarkan **Gambar 3.15**, “datah” adalah variabel yang digunakan untuk mendefinisikan *list of dictionaries* tersebut. Setiap data memiliki data dan informasi yang lebih lengkap sehingga lebih mudah menyortir banyak data tanpa mendefinisikan seperti nama dan lokasi. Variabel “datah” menyimpan banyak *dictionary*, yang masing-masing mewakili satu hotel. Setiap *dictionary* memiliki *key* seperti “kode”, “lokasi”, “nama”, “desc”, dan “cp”. *Dictionary* dalam *list* memungkinkan penyimpanan banyak entitas dengan struktur data yang seragam.

Fungsi yang digunakan pada *list* :

a. *Append()*

Digunakan untuk menambahkan nilai objek atau data di akhir *list*.

```
tambah_data(self, kota_choise, data_list):
    kode = input("Kode: ")
    lokasi = kota_choise
    nama = input("Nama: ")
    desc = input("Deskripsi: ")
    cp = input("Contact Person: ")
    data_list.append(DataDasar(kode, lokasi, nama, desc, cp))
    print("Data berhasil ditambahkan")
```

**Gambar 3.16** Contoh *append()*

Berdasarkan **Gambar 3.16**, “data\_list” adalah *list* yang menyimpan objek hotel. “.append()” merupakan *method list* yang berfungsi menambahkan data-data yang baru saja diinputkan di akhir *list* yang telah ada.

“self.namah\_daftar.append(DataDasar(kode, lokasi, nama, desc, cp))” digunakan untuk menambahkan data hotel baru ke dalam *list* namah\_daftar, setelah objek baru dibuat menggunakan konstruktor DataDasar(...). Metode *append()* merupakan cara standar di Python untuk menambahkan data ke list secara dinamis selama program berjalan.

b. *Remove()*

Digunakan untuk menghapus nilai objek atau data pada list

```
def hapus_data(self, data_list):
    kode = input("Masukkan Kode: ")
    for item in data_list:
        if item.kode == kode:
            data_list.remove(item)
            print("Data berhasil dihapus")
```

**Gambar 3.17** Contoh remove()

Pada **Gambar 3.17**, “data\_list” adalah *list* yang telah ada dan sedang aktif di dalam program. Program mencari “item” atau objek dalam “data\_list” yang cocok dengan kode yang dicari atau diinputkan. Kemudian “.remove(item)” akan menghapus item atau objek dari *list* berdasarkan referensinya.

### 3.1.5 Analisis Modul 5 – Program Berorientasi Objek (OOP)

OOP (*Object Oriented Programming*) adalah sebuah metode atau teknik pemrograman yang mengutamakan konsep objek sebagai pemodelan dari dunia nyata dalam membuat program.

1. *Class*

*Class* adalah suatu struktur data pada sebuah program yang dapat mewakili sejumlah objek dalam suatu jenis tertentu yang menyimpan data bersamaan dengan

nilai dan atributnya. *Class* menyediakan ide tentang bagaimana sesuatu harus didefinisikan.

```
class DataDasar:
    def __init__(self, kode, lokasi, nama, desc, cp):
        self.kode = kode
        self.lokasi = lokasi
        self.nama = nama
        self.desc = desc
        self.cp = cp
```

**Gambar 3.18** Contoh *Class*

Berdasarkan **Gambar 3.18**, *Class* “DataDasar” mendefinisikan cetakan untuk satu entitas data. *class* ini akan digunakan untuk membuat objek hotel/wisata/rumah makan. Atribut-atributnya yaitu kode, lokasi, nama, deskripsi, dan *contact person*, serta dapat dilengkapi dengan *method* untuk manipulasi data. Ini dilakukan agar data lebih terstruktur dan *reusable*.

## 2. *Method Constructor*

*Method* ‘`__init__`’ adalah sebuah *method* khusus yang dapat terpanggil secara otomatis ketika objek *class* tersebut dibuat. Singkatnya, *method* ‘`__init__`’ adalah konstruktor yang digunakan untuk menginisialisasi objek

```
def __init__(self):
    self.namah_daftar = []
    self.namaw_daftar = []
    self.namar_daftar = []
```

**Gambar 3.19** Contoh *Method Constructor*

Pada **Gambar 3.19**, objek akan dibuat dari *class* yang mana akan membuat data dari isi konstruktor dan mengisinya kedalam *list*. Saat objek dari “DataDasar” dibuat, konstruktor akan menerima parameter seperti kode, lokasi, dan lainnya kemudian menyimpannya ke dalam atribut *self*. “`__init__`” adalah *method constructor* yang membuatnya akan otomatis dipanggil saat objek dibuat.

“self.kode = kode” menyimpan parameter ke dalam atribut objek. Dan “self” merujuk pada objek itu sendiri. Penggunaan “\_\_init\_\_” meningkatkan efisiensi dan kerapihan kode.

### 3. *Method*

Operasi atau *method* pada sebuah kelas hampir sama dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas dapat memiliki lebih dari satu *method*. *Method* merupakan fungsi yang dapat dilakukan terhadap objek atau yang dilakukan oleh objek.

```
def sign_up(self):
    print("\n=== Sign Up ===")
    username = input("Masukkan username: ")
    password = input("Masukkan password: ")
    role = input("Masukkan peran (admin/user): ").lower()

    if role not in ["admin", "user"]:
        print("Peran tidak valid. Harap masukkan 'admin' atau 'user'.")
        return

    if username in self.users:
        print("Username sudah terdaftar. Silakan gunakan username lain.")
        return

    if not username or not password:
        print("Username dan password tidak boleh kosong!")
        return

    self.users[username] = {"password": password, "role": role}
    print("\n!!!Pendaftaran berhasil. Silakan Login !!!")
```

**Gambar 3.20** Contoh *Method*

Pada **Gambar 3.20**, terdapat pengondisian terhadap input yang diberikan di mana ini memproses data dalam fungsi “sign\_up” untuk dijadikan objek yang dapat digunakan untuk fungsi *login*. Pada contoh tersebut, *method* “tampil\_data(self, data\_list)” digunakan untuk mencetak informasi dari setiap elemen dalam *list*. Karena *method* ini berada di dalam *class*, maka harus menggunakan parameter *self* untuk mengakses atribut atau *method* lain di *class* tersebut. Ini mencerminkan konsep OOP yaitu perilaku (*method*) dan data (atribut) digabung dalam satu kesatuan yang logis.

#### 4. *Encapsulation*

Membungkus data dan *method* dalam sebuah objek untuk melindungi data dan membuat objek lebih terstruktur.

```
def edit_data(self, item):
    print("Data yang dapat diubah:")
    print("a. Lokasi")
    print("b. Nama")
    print("c. Deskripsi")
    print("d. Contact Person")

    pilihan = input("Masukkan pilihan yang akan di edit: ")

    if pilihan == "a":
        lokasi = input(f"Lokasi [{item.lokasi}]: ")
        item.lokasi = lokasi if lokasi else item.lokasi
    elif pilihan == "b":
        nama = input(f>Nama [{item.nama}]: ")
        item.nama = nama if nama else item.nama
    elif pilihan == "c":
        desc = input(f"Deskripsi [{item.desc}]: ")
        item.desc = desc if desc else item.desc
    elif pilihan == "d":
        cp = input(f"Contact Person [{item.cp}]: ")
        item.cp = cp if cp else item.cp
    else:
        print("Pilihan tidak valid")

    print(f>Data {item.nama} telah diperbarui")
```

**Gambar 3.21** Contoh *Encapsulation*

Berdasarkan **Gambar 3.21**, *Admin* tidak dapat mengubah “item.nama” secara langsung, tetapi harus melalui *method* seperti *method* “edit\_data()”. Ini adalah bentuk menjaga integritas data dan membuat program lebih aman, karena logika validasi dan pemrosesan bisa dikontrol di dalam *method*.

#### 5. *Polymorphism*



Memungkinkan objek memiliki lebih dari satu bentuk, sehingga objek memiliki kemampuan untuk menggunakan *method* yang sama tetapi menghasilkan *output* atau perilaku yang berbeda tergantung pada tipe data.

```
def tampil_data(self, data_list):
    if not data_list:
        print("Belum ada data")
    else:
        print("-" * 100)
        print("\n | {:^5} | {:^10} | {:^35} | {:^25} | {:^14} |".format("Kode", "Lokasi", "Nama", "Deskripsi", "Contact Person"))
        for item in data_list:
            print("-" * 100)
            print(" | {:^5} | {:^10} | {:^35} | {:^25} | {:^14} |".format(item.kode, item.lokasi, item.nama, item.desc, item.cp))
        print("-" * 100)
```

**Gambar 3.22** Contoh *Polymorphism*

Berdasarkan **Gambar 3.22**, *Method* “tampil\_data(data\_list)” dipanggil tiga kali dengan argumen berbeda, yaitu “namah\_daftar”, “namaw\_daftar”, dan “namar\_daftar”. Meskipun *method*-nya tetap satu, *method* ini dapat berlaku ke semua tipe objek “DataDasar” yang ada di *list* berbeda. Artinya, *method* ini dapat digunakan untuk menampilkan data hotel, data wisata, atau data rumah makan secara

## DAFTAR PUSTAKA

- Husni, & Rachmad, A. (2021). *ALGORITMA PEMROGRAMAN : Pendekatan Praktis Menggunakan Python*. Tasikmalaya: Perkumpulan Rumah Cemerlang Indonesia.
- Kadir, A. (2019). *Logika Pemrograman PYTHON*. Jakarta: PT Elex Media Komputindo.
- Kadir, A. (2021). *Dasar Logika Pemrograman Komputer (Update Version)*. Jakarta: PT Elex Media Komputindo.
- Rahmatudin, J., & Sumliyah. (2024). *PEMROGRAMAN PYTHON*. Malang: PT. Literasi Nusantara Abadi Grup.
- Siska, S. T. (2023). *Algoritma Pemrograman Python*. Yogyakarta: PT. Penamuda Media.
- Sutomo, B., Nurdiyanto, H., & Saprudin, U. (2023). *PYTHON BASIC PROGRAMING*. Malang: PT. Literasi Nusantara Abadi Group.
- Umam, K. (2021). *ALGORITMA DAN PEMROGRAMAN KOMPUTER DENGAN PYTHON*. Pemekasaan: Duta Media Publishing.
- Yuliani, O., & Prasajo, J. (2015). Rancang Bangun Sistem Informasi Obyek Wisata Berbasis Web Menggunakan Metode USER CENTERED DESIGN. *Jurnal Angkasa*, 149-156.
- Yuniarti, W. D. (2019 ). *Dasar-Dasar Pemrograman dengan Python*. Yogyakarta: Penerbit DEEPUBLISH.

# LAMPIRAN

**LAMPIRAN A**  
**(*Output* Program)**

```
=== Selamat Datang di Sistem Objek Pariwisata ===  
Kota Padang Batam Bali  
  
==MENU UTAMA==  
1. Sign Up  
2. Login  
3. Keluar  
Silakan masukkan pilihan (1/2/3): 1  
  
=== Sign Up ===  
Masukkan username: user  
Masukkan password: user123  
Masukkan peran (admin/user): user  
  
!!!Pendaftaran berhasil. Silakan Login !!!  
  
=== Selamat Datang di Sistem Objek Pariwisata ===  
Kota Padang Batam Bali  
  
==MENU UTAMA==  
1. Sign Up  
2. Login  
3. Keluar  
Silakan masukkan pilihan (1/2/3): 2  
  
=== Login Sistem Pariwisata ===  
Username: user  
Password: user123  
  
!!!Berhasil login sebagai user!!!  
  
===SELAMAT DATANG DI MENU USER===  
  
Menu User  
1. Tampilkan Data Berdasarkan Kota  
2. Keluar  
Silakan masukkan menu (1/2): 2  
Program selesai. Terimakasih sudah menggunakan program kami!
```

**Lampiran A.1** *Output Login User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.2** *Output Hotel Padang Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDR01 | Padang | Ayam Bakar Sing A Song | Spesialis ayam bakar | 0831 |
=====
| PDR02 | Padang | RM Lamun Ombak | Masakan Padang | 0832 |
=====

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.3** *Output Rumah Makan Padang Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 2

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDW01 | Padang | Pantai Air Manis | Legenda Malin Kundang | 0821 |
=====
| PDW02 | Padang | Jembatan Siti Nurbaya | Pemandangan Pelabuhan | 0822 |
=====

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.4** *Output Wisata Padang Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 2

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BTH01 | Batam | Batam Marriott Hotel Harbour Bay | Sea View | 0813 |
=====
| BTH02 | Batam | The Music Hotel | Budget-friendly | 0814 |
=====

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.5** *Output Hotel Batam Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 2

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3
=====

| Kode | Lokasi | Nama | Deskripsi | Contact Person |
-----|-----|-----|-----|-----|
| BTR01 | Batam | Wey Wey Seafood | Spesialis seafood | 0833 |
-----|-----|-----|-----|-----|
| BTR02 | Batam | My Garden Resto | Cafe berkonsep taman | 0834 |
-----|-----|-----|-----|-----|

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.6** *Output Rumah Makan Batam Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 2

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 2
=====

| Kode | Lokasi | Nama | Deskripsi | Contact Person |
-----|-----|-----|-----|-----|
| BTW01 | Batam | Mega wisata Ocarina Theme Park | Taman Hiburan Keluarga | 0823 |
-----|-----|-----|-----|-----|
| BTW02 | Batam | Miniature House Indonesia | Free-Entry | 0824 |
-----|-----|-----|-----|-----|

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.7** *Output Wisata Batam Menu User*



```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 3

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1
=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
|-----|-----|-----|-----|-----|
| BLH01 | Bali | Neo Kuta Jelantik Hotel | Budget-friendly | 0815 |
|-----|-----|-----|-----|-----|
| BLH02 | Bali | Hotel Siesta Legian | Dekat Pantai | 0816 |
|-----|-----|-----|-----|-----|
Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.8** *Output Hotel Bali Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 3

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3
=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
|-----|-----|-----|-----|-----|
| BLR01 | Bali | Kala Uluwatu | Greek-Inspired Restaurant | 0835 |
|-----|-----|-----|-----|-----|
| BLR02 | Bali | Warung Laota | Hidangan ala Hongkong | 0836 |
|-----|-----|-----|-----|-----|
Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.9** *Output Rumah Makan Bali Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 3

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 2

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BLW01 | Bali | Pura Uluwatu | Kuil Hindu | 0825 |
=====
| BLW02 | Bali | Garuda Wisnu Kencana | Patung raksasa | 0826 |
=====

Apakah ingin lanjut? (y/n): y

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.10** *Output Wisata Bali Menu User*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: x

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.11** *Output Pilihan X Menu User*

```
===SELAMAT DATANG DI MENU USER===
```

```
Menu User
```

1. Tampilkan Data Berdasarkan Kota
2. Keluar

```
Silakan masukkan menu (1/2): 3
```

```
Pilihan tidak valid. Silakan coba lagi.
```

```
===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.12** *Output Pilihan Invalid Menu User*

```
===SELAMAT DATANG DI MENU USER===
```

```
Menu User
```

1. Tampilkan Data Berdasarkan Kota
2. Keluar

```
Silakan masukkan menu (1/2): 1
```

```
Pilih kota:
```

1. Padang
2. Batam
3. Bali

```
Masukkan pilihan kota (1/2/3): 4
```

```
Pilihan tidak valid.
```

```
===SELAMAT DATANG DI MENU USER===
```

```
Menu User
```

1. Tampilkan Data Berdasarkan Kota
2. Keluar

```
Silakan masukkan menu (1/2): 1
```

**Lampiran A.12** *Output Pilihan Invalid Menu User (Lanjutan)*

```
Silakan masukkan menu (1/2): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data apa yang ingin ditampilkan?
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 4
Pilihan tidak tersedia.

===SELAMAT DATANG DI MENU USER===
```

**Lampiran A.12** *Output Pilihan Invalid Menu User (Lanjutan)*

```
===SELAMAT DATANG DI MENU USER===

Menu User
1. Tampilkan Data Berdasarkan Kota
2. Keluar
Silakan masukkan menu (1/2): 2
Program selesai. Terimakasih sudah menggunakan program kami!
```

**Lampiran A.13** *Output Pilihan Keluar Menu User*

```
=== Selamat Datang di Sistem Objek Pariwisata ===  
Kota Padang Batam Bali  
  
==MENU UTAMA==  
1. Sign Up  
2. Login  
3. Keluar  
Silakan masukkan pilihan (1/2/3): 1  
  
=== Sign Up ===  
Masukkan username: admin  
Masukkan password: admin123  
Masukkan peran (admin/user): admin  
  
!!!Pendaftaran berhasil. Silakan Login !!!  
  
=== Selamat Datang di Sistem Objek Pariwisata ===  
Kota Padang Batam Bali  
  
==MENU UTAMA==  
1. Sign Up  
2. Login  
3. Keluar  
Silakan masukkan pilihan (1/2/3): 2  
  
=== Login Sistem Pariwisata ===  
Username: admin  
Password: admin123  
  
!!!Berhasil login sebagai admin!!!  
  
===SELAMAT DATANG DI MENU ADMIN===  
  
Menu Admin  
1. Tampilkan Data  
2. Tambah Data  
3. Hapus Data  
4. Edit Data  
5. Keluar  
Silakan masukkan menu (1/2/3/4/5): 5  
Program selesai. Terimakasih sudah menggunakan program kami!
```

**Lampiran A.14** *Output Login Admin*

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
```

**Lampiran A.15** *Output Tampilkan Data Padang Menu Admin*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 2

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDW01 | Padang | Pantai Air Manis | Legenda Malin Kundang | 0821 |
=====
| PDW02 | Padang | Jembatan Siti Nurbaya | Pemandangan Pelabuhan | 0822 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDR01 | Padang | Ayam Bakar Sing A Song | Spesialis ayam bakar | 0831 |
=====
| PDR02 | Padang | RM Lamun Ombak | Masakan Padang | 0832 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
```

**Lampiran A.15** *Output Tampilkan Data Padang Menu Admin (Lanjutan)*

```
===SELAMAT DATANG DI MENU ADMIN===

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 2

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BTH01 | Batam | Batam Marriott Hotel Harbour Bay | Sea View | 0813 |
=====
| BTH02 | Batam | The Music Hotel | Budget-friendly | 0814 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
```

**Lampiran A.16** *Output Tampilkan Data Batam Menu Admin*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 2

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BTW01 | Batam | Mega wisata Ocarina Theme Park | Taman Hiburan Keluarga | 0823 |
=====
| BTW02 | Batam | Miniature House Indonesia | Free-Entry | 0824 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BTR01 | Batam | Wey Wey Seafood | Spesialis seafood | 0833 |
=====
| BTR02 | Batam | My Garden Resto | Cafe berkonsep taman | 0834 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
```

**Lampiran A.16** *Output Tampilkan Data Batam Menu Admin (Lanjutan)*

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 3

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BLH01 | Bali | Neo Kuta Jelantik Hotel | Budget-friendly | 0815 |
=====
| BLH02 | Bali | Hotel Siesta Legian | Dekat Pantai | 0816 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
1. Hotel
```

**Lampiran A.17** *Output Tampilkan Data Bali Menu Admin*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 2

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BLW01 | Bali | Pura Uluwatu | Kuil Hindu | 0825 |
=====
| BLW02 | Bali | Garuda Wisnu Kencana | Patung raksasa | 0826 |
=====

Apakah ingin lanjut? (y/n): y

Data yang dapat dipilih:
```

**Lampiran A.17** *Output Tampilkan Data Bali Menu Admin (Lanjutan)*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| BLR01 | Bali | Kala Uluwatu | Greek-Inspired Restaurant | 0835 |
=====
| BLR02 | Bali | Warung Laota | Hidangan ala Hongkong | 0836 |
=====
```

**Lampiran A.17** *Output Tampilkan Data Bali Menu Admin (Lanjutan)*



```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 2

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====

Kode: pdh03
Nama: pangeran beach
Deskripsi: dekat pantai
Contact Person: 0895
Data berhasil ditambahkan
Apakah ingin lanjut? (y/n): █
```

Lampiran A.18 *Output Tambah Data Menu Admin*

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====
| PDH03 | Padang | Pangeran beach | Dekat pantai | 0895 |
=====

Apakah ingin lanjut? (y/n): █
```

Lampiran A.19 *Output Hasil Tambah Data Menu Admin*

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 3

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====
| PDH03 | Padang | Pangeran beach | Dekat pantai | 0895 |
=====

Masukkan Kode: pdh03
Data berhasil dihapus
Apakah ingin lanjut? (y/n): y
```

Lampiran A.20 Output Hapus Data Menu Admin

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====

Apakah ingin lanjut? (y/n):
```

Lampiran A.21 Output Hasil Hapus Data Menu Admin

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 4

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
=====
Masukkan kode yang akan diubah: pdh02
```

Lampiran A.22 *Output Edit Data Menu Admin*

```
====SELAMAT DATANG DI MENU ADMIN====

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 1

Pilih kota:
1. Padang
2. Batam
3. Bali
Masukkan pilihan kota (1/2/3): 1

Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
=====
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
=====
| PDH02 | Padang | Grand zuri | Tengah Kota | 0812 |
=====
Apakah ingin lanjut? (y/n): █
```

Lampiran A.23 *Output Hasil Edit Data Menu Admin*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 1

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
|-----|-----|-----|-----|-----|
| PDH01 | Padang | Mercure Padang | Western Food | 0811 |
|-----|-----|-----|-----|-----|
| PDH02 | Padang | Grand Zuri Padang | Tengah Kota | 0812 |
|-----|-----|-----|-----|-----|

Masukkan Kode: pdg01
Kode tidak valid
Apakah ingin lanjut? (y/n):
```

**Lampiran A.24** *Output Pilihan Invalid Menu Admin*

```
===SELAMAT DATANG DI MENU ADMIN===

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): admin
Pilihan tidak valid. Silahkan coba lagi

===SELAMAT DATANG DI MENU ADMIN===
```

**Lampiran A.24** *Output Pilihan Invalid Menu Admin (Lanjutan)*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: 3

=====
| Kode | Lokasi | Nama | Deskripsi | Contact Person |
|-----|-----|-----|-----|-----|
| BLR01 | Bali | Kala Uluwatu | Greek-Inspired Restaurant | 0835 |
|-----|-----|-----|-----|-----|
| BLR02 | Bali | Warung Laota | Hidangan ala Hongkong | 0836 |
|-----|-----|-----|-----|-----|

Apakah ingin lanjut? (y/n): n

===SELAMAT DATANG DI MENU ADMIN===
```

**Lampiran A.25** *Output Pilihan N Menu Admin*

```
Data yang dapat dipilih:
1. Hotel
2. Wisata
3. Rumah Makan
x. Kembali ke menu utama
Silakan masukkan pilihan: x

===SELAMAT DATANG DI MENU ADMIN===
```

**Lampiran A.26** *Output Pilihan X Menu Admin*


```
===SELAMAT DATANG DI MENU ADMIN===

Menu Admin
1. Tampilkan Data
2. Tambah Data
3. Hapus Data
4. Edit Data
5. Keluar
Silakan masukkan menu (1/2/3/4/5): 5
Program selesai. Terimakasih sudah menggunakan program kami!
```


**Lampiran A.27** *Output Pilihan Keluar Menu Admin*

**LAMPIRAN B**  
**(Kartu Praktikum)**





KARTU ABSENSI PRAKTIKUM  
PROGRAMA KOMPUTER  
2025




---

Nama : Muhammad Rayhan

NIM : 2410932033


Kelompok : GA

Asisten Pembimbing : Hadyen Zada Fitra




No	Modul	Waktu	Paraf
1	Modul 1 – Pengenalan Komputer dan Dasar Bahasa Python	06-05-2025 / shift 5	✓
2	Modul 2 – Konsep Keputusan	Kamis / 14.30 – 15.50	✓
3	Modul 3 – Perulangan (Loop)	Kamis / 14.30 – 15.50	✓
4	Modul 4 – Array	Kamis / 09.20 – 11.00	✓
5	Modul 5 – Pemrograman Berorientasi Objek (OOP)	Kamis / 16.00 – 17.40	✓
6	Presentasi Tugas Besar		

**Lampiran B.1** Kartu Praktikum Muhammad Rayhan



KARTU ABSENSI PRAKTIKUM  
PROGRAMA KOMPUTER  
2025




---

Nama : Nadhirah Zaskya Syanela

NIM : 2410932029

Kelompok : 8A

Asisten Pembimbing : Hanna Izzani Rizqita



No	Modul	Waktu	Paraf
1	Modul 1 – Pengenalan Komputer dan Dasar Bahasa Python	Kamis 06 Maret 2025 (Shift 5)	✓
2	Modul 2 – Konsep Keputusan	Kamis, 13-Maret-2025 / 14.30-15.50	✓
3	Modul 3 – Perulangan (Loop)	Kamis / 14.30 – 15.50	✓
4	Modul 4 – Array	Jumat / 9.20 – 11.00	✓
5	Modul 5 – Pemrograman Berorientasi Objek (OOP)	Kamis / 16.00 – 17.40	✓
6	Presentasi Tugas Besar		

**Lampiran B.2** Kartu Praktikum Nadhirah Zaskya Syanela



KARTU ABSENSI PRAKTIKUM  
PROGRAMA KOMPUTER  
2025



Nama : Kirana Dwiva Ninda  
NIM : 2410931005  
Kelompok : 5A  
Asisten Pembimbing : Yosentati Fahdi



No	Modul	Waktu	Paraf
1	Modul 1 – Pengenalan Komputer dan Dasar Bahasa Python	6/3/2025 / 19.30-19.50	✓
2	Modul 2 – Konsep Keputusan	12/3/2025 / 19.50-19.50	✓
3	Modul 3 – Perulangan (Loop)	Kamis / 14.30-15.00	✓
4	Modul 4 – Array	Jumat / 09.30-11.00	Basil
5	Modul 5 – Pemograman Berorientasi Objek (OOP)	Kamis / 16.00-17.00	✓
6	Presentasi Tugas Besar		

**Lampiran B.3** Kartu Praktikum Kirana Dwiva Ninda