

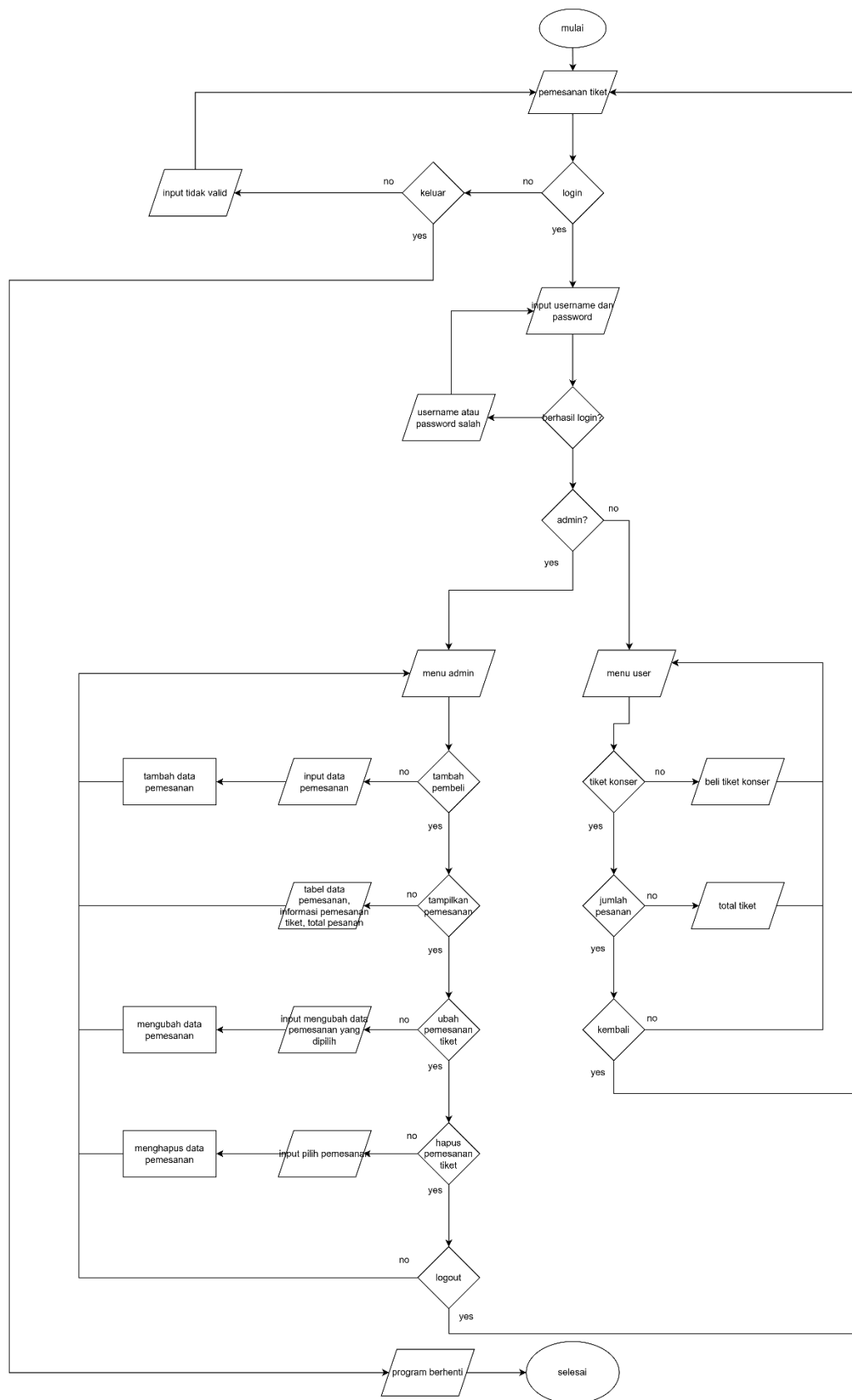
LAPORAN PRAKTIKUM
POSTTEST 7
ALGORITMA PEMROGRAMAN DASAR



Disusun oleh:
Hibrizi Yusriani (2509106131)
Kelas (C2 '25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



Gambar 1 Flowchart

Declare berfungsi sebagai digunakan untuk memberi tahu program bahwa sebuah variabel, fungsi, atau struktur data akan digunakan, dan menentukan jenisnya.

Assign berfungsi sebagai untuk memberi nilai atau tugas. Input berfungsi sebagai untuk memasukan data atau informasi.

if berfungsi sebagai untuk mengecek Keputusan atau mengoreksi apakah benar atau salah.

Fungsi while dalam pemrograman digunakan untuk menjalankan sebuah blok kode berulang kali selama kondisi tertentu bernilai benar (True).

Fungsi for dalam pemrograman digunakan untuk melakukan perulangan dengan jumlah yang sudah diketahui atau bisa ditentukan. Ini sangat berguna saat kamu ingin mengulang sesuatu berdasarkan urutan, seperti daftar, range angka, atau karakter dalam string. output berfungsi untuk menyebut hasilnya.

2. Deskripsi Singkat Program

Program ini ini memungkinkan pengguna untuk membooking tiket konser melalui beberapa fitur, seperti melihat, menambah, mengubah, dan menghapus orang yang membooking tiket konser THE SIGIT

3. Source Code

```
import os
from prettytable import PrettyTable

# ===== VARIABEL GLOBAL =====
data_pemesanan = {
    "Rijie": {"no_hp": "081234567890", "jumlah": 2, "metode_pembayaran":
"Transfer Bank"},
    "Luna": {"no_hp": "082145678921", "jumlah": 1, "metode_pembayaran":
"QRIS"},
}

data_pengguna = {
    "rijie": {"password": "admin123", "level_akses": "admin"}
}

login_status = False # Variabel global tambahan

# ===== FUNGSI =====
```

```

# Fungsi tanpa parameter (menampilkan seluruh pemesanan)
def tampilkan_pemesanan():
    os.system("cls || clear")
    print("=== Daftar Pemesanan ===")
    if len(data_pemesanan) == 0:
        print("Belum ada Pemesanan")
    else:
        tabel = PrettyTable()
        tabel.field_names = ["No", "Nama", "No HP", "Jumlah Tiket",
"Metode Pembayaran"]
        tabel.hrules = 1
        for i, (nama, data) in enumerate(data_pemesanan.items(),
start=1):
            tabel.add_row([i, nama, data["no_hp"], data["jumlah"],
data["metode_pembayaran"]])
            print(tabel)
            input("\nTekan Enter untuk kembali...")

# Fungsi tanpa parameter (login user)
def login():
    global login_status
    os.system("cls || clear")
    print("=====")
    print("      LOGIN SISTEM PEMESANAN      ")
    print("=====")
    username_input = input("Masukkan username: ").lower()
    password_input = input("Masukkan password: ").strip()

    for username, data in data_pengguna.items():
        if username == username_input and data["password"] ==
password_input:
            print("Login berhasil!")
            login_status = True
            return data["level_akses"]
    print("Login gagal! Username atau password salah.")
    return None

# Fungsi dengan parameter (menambah pemesanan)
def tambah_pemesanan(nama, nomor_hp, jumlah_tiket, metode_pembayaran):
    try:
        if nama in data_pemesanan:
            raise ValueError("Nama sudah terdaftar.")
        if not jumlah_tiket.isdigit() or int(jumlah_tiket) <= 0:
            raise ValueError("Jumlah tiket harus berupa angka positif.")
        data_pemesanan[nama] = {
            "no_hp": nomor_hp,
            "jumlah": int(jumlah_tiket),
            "metode_pembayaran": metode_pembayaran
        }
        print(f"Pemesanan atas nama {nama} berhasil ditambahkan!")
    except ValueError as e:
        print(f"Error: {e}")

```

```

# Fungsi dengan parameter (menghapus pemesanan)
def hapus_pemesanan(nama):
    try:
        if nama not in data_pemesanan:
            raise KeyError("Nama tidak ditemukan dalam data pemesanan.")
        del data_pemesanan[nama]
        print(f"Pemesanan {nama} berhasil dihapus.")
    except KeyError as e:
        print(f"Error: {e}")

# ===== PROSEDUR =====

# Prosedur menampilkan menu
def tampilkan_menu():
    os.system("cls || clear")
    print("\n=====")
    print("    BOOKING TIKET KONSER THE SIGIT")
    print("=====")

    table = PrettyTable()
    table.field_names = ["No", "Menu"]
    table.align["No"] = "c"
    table.align["Menu"] = "l"

    table.add_row(["1", "Lihat semua pemesanan"])
    table.add_row(["2", "Tambah pemesanan"])
    table.add_row(["3", "Hapus pemesanan"])
    table.add_row(["4", "Keluar"])
    print(table)

# Prosedur utama menjalankan program
def jalankan_program():
    level_akses = login()
    if not login_status:
        return

    while True:
        tampilkan_menu()
        pilihan = input("Pilih menu (1-4): ").strip()

        if pilihan == "1":
            tampilkan_pemesanan()

        elif pilihan == "2":
            os.system("cls")
            print("=== Tambah Pemesanan ===")
            # variabel lokal
            nama = input("Nama: ").strip()
            nomor_hp = input("Nomor HP: ").strip()
            jumlah_tiket = input("Jumlah tiket: ").strip()
            metode = input("Metode pembayaran: ").strip()
            tambah_pemesanan(nama, nomor_hp, jumlah_tiket, metode)

```

```

        input("\nTekan Enter untuk kembali...")

    elif pilihan == "3":
        os.system("cls")
        print("=== Hapus Pemesanan ===")
        tampilkan_pemesanan()
        nama = input("Masukkan nama yang ingin dihapus: ").strip()
        hapus_pemesanan(nama)
        input("\nTekan Enter untuk kembali...")

    elif pilihan == "4":
        print("Terima kasih! Program selesai.")
        break

    else:
        print("Pilihan tidak valid! Coba lagi.")

# ===== MAIN PROGRAM =====
if __name__ == "__main__":
    jalankan_program()

```

4. Hasil Output

```

=====
      BOOKING TIKET KONSER THE SIGIT
=====
+---+-----+
| No | Menu                |
+---+-----+
| 1  | Lihat semua pemesanan |
| 2  | Tambah pemesanan      |
| 3  | Ubah pemesanan        |
| 4  | Hapus pemesanan       |
| 5  | Keluar                 |
+---+-----+
Pilih menu (1-5): 

```

Gambar 1.1 Hasil output menu pertama

```
=== Daftar Pemesanan ===
+---+---+---+---+---+---+
| No | Nama | No HP | Jumlah Tiket | Metode Pembayaran |
+---+---+---+---+---+---+
| 1 | Rijie | 081234567890 | 2 | Transfer Bank |
+---+---+---+---+---+---+
| 2 | Luna | 082145678921 | 1 | QRIS |
+---+---+---+---+---+---+

Tekan Enter untuk kembali...
```

Gambar 1.2 Menampilkan CRUD Cread

```
=== Tambah Pemesanan ===
Nama: bang ridho
Nomor HP: 08483208245
Jumlah tiket: 1
Metode pembayaran: cash or duel
```

Gambar 1.3 Menampilkan CRUD Create

```
=== UBAH PEMESANAN ===
1. Rijie | 081234567890 | 2 Tiket | Transfer Bank
2. Luna | 082145678921 | 1 Tiket | QRIS
3. bang ridho | 081376283686 | 1 Tiket | cash or duel
Pilih nomor data yang ingin diubah: 3
Nama baru: bang dwiki
No HP baru: 0828372947
Jumlah tiket baru: 1
Metode pembayaran baru: qris
```

Gambar 1.4 Menampilkan CRUD Update

```

=== HAPUS PEMESANAN ===
1. Rije | 081234567890 | 2 Tiket | Transfer Bank
2. Luna | 082145678921 | 1 Tiket | QRIS
3. bang dwiki | 0828372947 | 1 Tiket | qris
Pilih nomor data yang ingin dihapus: 3
Yakin hapus bang dwiki? (y/n): y

```

Gambar 1.5 Menampilkan CRUD Delete

5. Langkah-langkah GIT

5.1 GIT Add

```
PS D:\praktikum-apd> git add .
```

Gambar 2.1

Git Add adalah perintah dalam Git yang digunakan untuk menambahkan perubahan pada file ke **staging area** (atau index).

5.2 GIT Commit

```

PS D:\praktikum-apd> git commit -m "uplodpt7"
[main e8db5f8] uplodpt7
1 file changed, 138 insertions(+)
create mode 100644 post-test/post-test-apd-7/2509106131-Hibriziyusriani.py

```

Gambar 2.2

Git Commit adalah perintah dalam Git yang digunakan untuk menyimpan perubahan yang telah ditambahkan ke **staging area** (melalui **git add**) ke dalam **repository lokal**.

5.3 GIT Push

```

PS D:\praktikum-apd> git push
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 12 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 1.84 KiB | 629.00 KiB/s, done.
Total 5 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/hibriziysriani/praktikum-apd.git
56e2941..e8db5f8 main -> main
PS D:\praktikum-apd> |

```

Gambar 2.3

Git Push adalah perintah dalam Git yang digunakan untuk mengirim (upload) commit dari **repository lokal** ke **repository remote** (seperti GitHub, GitLab, atau server lain).