

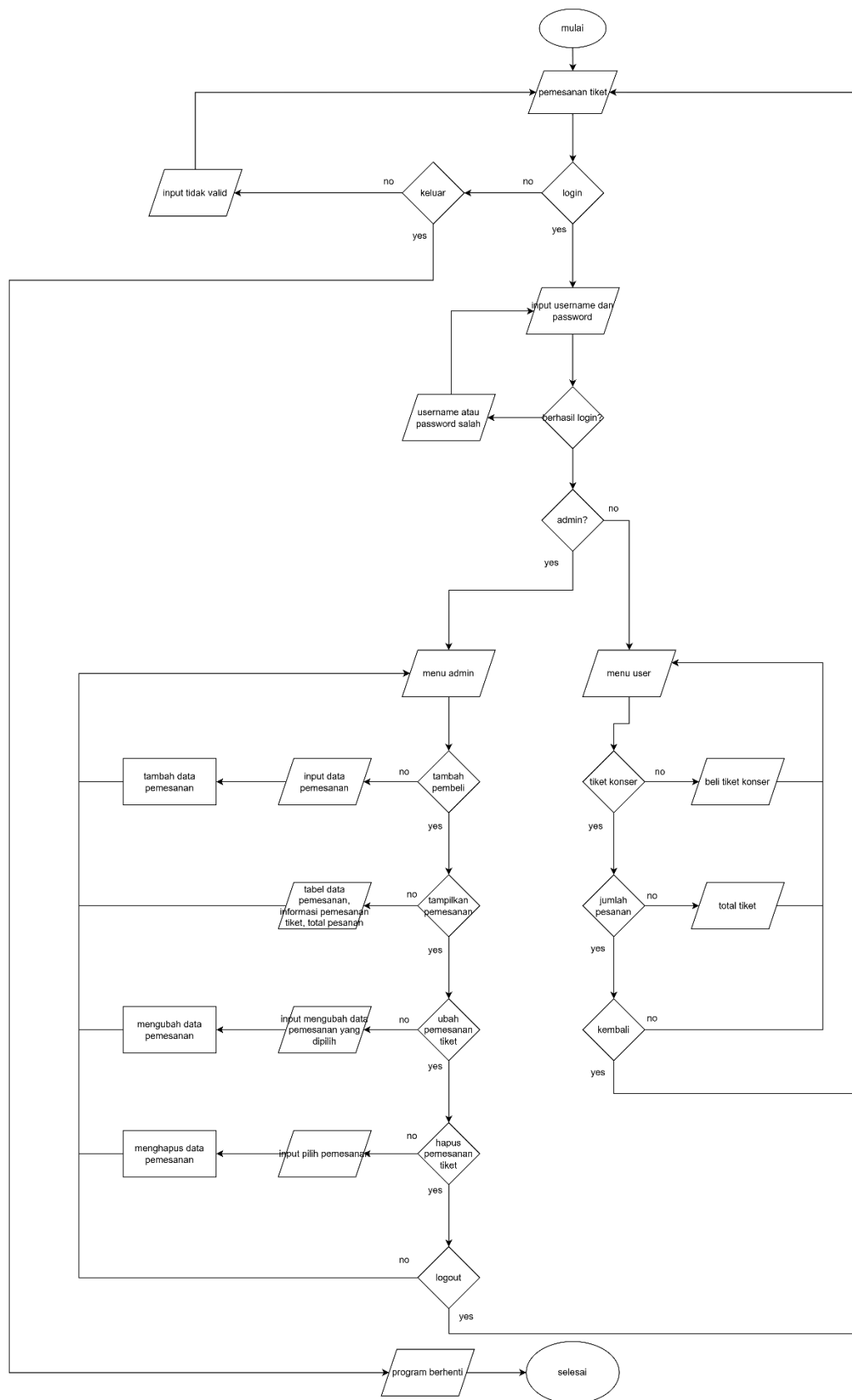
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN DASAR



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

# Mengubah struktur data dari list of tuples menjadi dictionary
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 SYSTEM =====
```

```

os.system("cls || clear")
print("=====")
print("      LOGIN SISTEM PEMESANAN      ")
print("=====")

username_input = input("Masukkan username: ").lower()
password_input = input("Masukkan password: ").strip()

# Verifikasi login
level_akses = None
for username, data in data_pengguna.items():
    if username == username_input and data["password"] == password_input:
        level_akses = data["level_akses"]
        break

if level_akses is None:
    print("Login gagal! Username atau password salah.")
    exit()

while True:
    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", "Ubah pemesanan"])
    table.add_row(["4", "Hapus pemesanan"])
    table.add_row(["5", "Keluar"])

    print(table)

    # Input pilihan menu
    menu = input("Pilih menu (1-5): ").strip()

    # READ
    if menu == "1":
        os.system("cls")
        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...")

# CREATE
elif menu == "2":
    os.system("cls")
    print("=== Tambah Pemesanan ===")
    nama = input("Nama: ").strip()
    if nama in data_pemesanan:
        print("Nama sudah ada, gunakan nama lain.")
    else:
        nomor_hp = input("Nomor HP: ").strip()
        jumlah_tiket_input = input("Jumlah tiket: ").strip()
        metode_pembayaran = input("Metode pembayaran: ").strip()
        if not nama or not nomor_hp or not jumlah_tiket_input or not
metode_pembayaran:
            print("Semua data harus diisi.")
        elif not jumlah_tiket_input.isdigit() or int(jumlah_tiket_input)
<= 0:
            print("Jumlah tiket harus angka lebih dari 0.")
        else:
            jumlah_tiket = int(jumlah_tiket_input)
            data_pemesanan[nama] = {
                "no_hp": nomor_hp,
                "jumlah": jumlah_tiket,
                "metode_pembayaran": metode_pembayaran
            }
            print("Pemesanan berhasil ditambahkan.")

# UPDATE
elif menu == "3":
    print("\n=== UBAH PEMESANAN ===")
    if not data_pemesanan:
        print("Belum ada data untuk diubah.")
    else:
        for i, (nama, data) in enumerate(data_pemesanan.items(),
start=1):
            print(f"{i}. {nama} | {data['no_hp']} | {data['jumlah']}
Tiket | {data['metode_pembayaran']}")

```

```

        pilih = input("Pilih nomor data yang ingin diubah: ").strip()
        if not pilih.isdigit():
            print("Input harus berupa angka!")
        else:
            pilih = int(pilih)
            if not (1 <= pilih <= len(data_pemesanan)):
                print("Nomor tidak valid!")
            else:
                nama_lama = list(data_pemesanan.keys())[pilih - 1]
                nama_baru = input("Nama baru: ").strip()
                hp_baru = input("No HP baru: ").strip()
                jumlah_baru_input = input("Jumlah tiket baru: ").strip()
                bayar_baru = input("Metode pembayaran baru: ").strip()

                if not nama_baru or not hp_baru or not jumlah_baru_input
or not bayar_baru:
                    print("Semua data harus diisi.")
                    elif not jumlah_baru_input.isdigit() or
int(jumlah_baru_input) <= 0:
                        print("Jumlah tiket harus angka lebih dari 0.")
                    else:
                        jumlah_baru = int(jumlah_baru_input)
                        # Jika nama berubah, hapus yang lama dan tambah yang
baru

                        if nama_baru != nama_lama:
                            del data_pemesanan[nama_lama]
                            data_pemesanan[nama_baru] = {
                                "no_hp": hp_baru,
                                "jumlah": jumlah_baru,
                                "metode_pembayaran": bayar_baru
                            }
                        print("Data berhasil diperbarui!")

# DELETE
elif menu == "4":
    print("\n=== HAPUS PEMESANAN ===")
    if not data_pemesanan:
        print("Belum ada data untuk dihapus.")
    else:
        for i, (nama, data) in enumerate(data_pemesanan.items(),
start=1):
            print(f"{i}. {nama} | {data['no_hp']} | {data['jumlah']}
Tiket | {data['metode_pembayaran']}")

        pilih = input("Pilih nomor data yang ingin dihapus: ").strip()
        if not pilih.isdigit():
            print("Input harus berupa angka!")
        else:

```

```

        pilih = int(pilih)
        if not (1 <= pilih <= len(data_pemesanan)):
            print("Nomor tidak valid!")
        else:
            nama_hapus = list(data_pemesanan.keys())[pilih - 1]
            konfirmasi = input(f"Yakin hapus {nama_hapus}? (y/n):
").lower().strip()
            if konfirmasi == "y":
                del data_pemesanan[nama_hapus]
                print("Data berhasil dihapus!")
            else:
                print("Batal dihapus.")

# EXIT
elif menu == "5":
    print("Terima kasih! Program selesai.")
    break

else:
    print("Pilihan menu tidak valid, coba lagi!")

```

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. Rijie | 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 "upload pt 6"
[main 7fb119b] upload pt 6
4 files changed, 293 insertions(+)
create mode 100644 kelas/pertemuan-6/pertemuan6.py
rename post-test/{ => post-test-apd-3}/post-test-apd-2/2506106131-HibriziYusriani-pt-2.pdf (100%)
rename post-test/{ => post-test-apd-3}/post-test-apd-2/2509106131-HibriziYusriani_pt-2.py (100%)
create mode 100644 post-test/post-test-apd-6/2509106131-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: 13, done.
Counting objects: 100% (13/13), done.
Delta compression using up to 12 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 3.16 KiB | 648.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/hibriziysriani/praktikum-apd.git
c80c036..7fb119b main -> main
```

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).