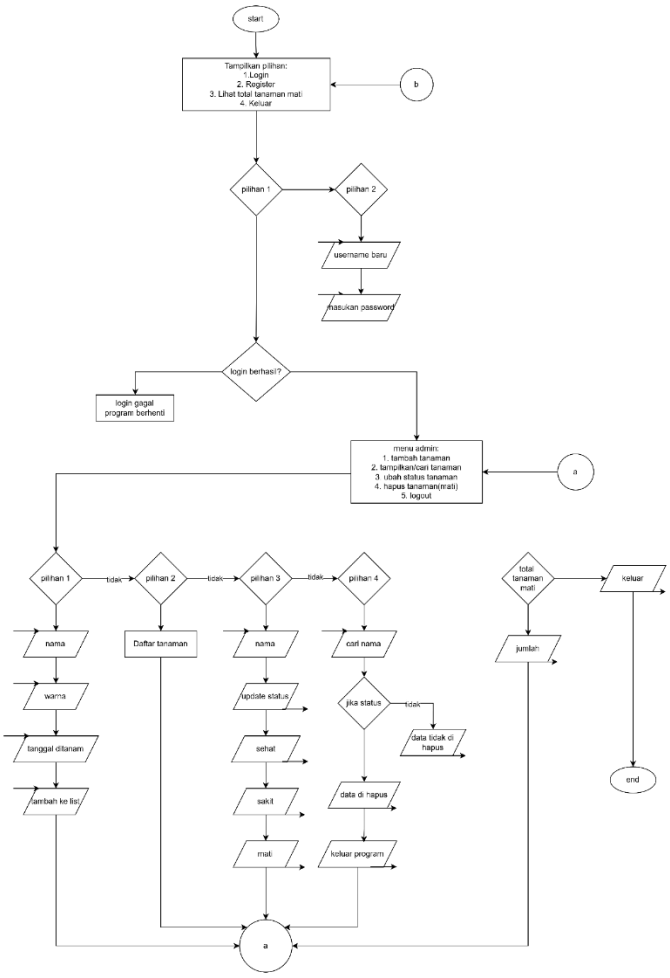
LAPORAN PRAKTIKUM
POSTTEST 8
ALGORITMA PEMROGRAMAN DASAR

Disusun oleh:
Zihni Larasati (2509106010)
Informatika (A'25)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. **Flowchart**



2. **Deskripsi singkat program**

1. Modul **prettytable** digunakan untuk menampilkan data tanaman dalam format tabel yang rapi.
2. Program dipecah menjadi empat file untuk **pemisahan fungsionalitas**: data, fungsi tanaman, menu, dan main program.
3. Logika program **tidak diubah sama sekali**, hanya tampilan dan struktur file yang diperbaiki.

3. Source code

```
akun = {

    "admin": {"password": "andi laras nee", "role": "admin"},
    "AADPA": {"password": "Andi Ahmad dzaky P A", "role": "user"},
    "Laras": {"password": "Z Larasati", "role": "user"}
}

tanaman_dict = {
    "Bunga Mawar": {"warna": "merah", "tanggal": "2022-08-27", "status":
"sehat"},
    "Bunga Tulip": {"warna": "pink", "tanggal": "2024-05-30", "status":
"sehat"},
    "Bunga Kamboja": {"warna": "putih", "tanggal": "2020-09-24", "status":
"sehat"},
    "Bunga Matahari": {"warna": "kuning", "tanggal": "2024-04-02", "status":
"mati"},
    "Bunga Alamanda": {"warna": "ungu", "tanggal": "2023-02-07", "status":
"mati"}
}
```

Menyimpan data login pengguna dalam bentuk dictionary bersarang.

- Setiap username punya password dan role (admin atau user).
- Role menentukan akses fitur di menu.

tanaman_dict:

Menyimpan data tanaman bunga.

- Kunci dictionary = nama tanaman (misal: "Bunga Mawar")
- Nilai = dictionary berisi warna, tanggal ditanam, dan status tanaman.

```
def tampilkan_menu_admin():
    """Menampilkan menu untuk admin"""
    print("=== MENU ADMIN ===")
    print("1. Tambah Tanaman")
    print("2. Tampilkan/Cari Tanaman")
    print("3. Ubah Status Tanaman")
    print("4. Hapus Tanaman (Mati)")
    print("5. Logout")

def tampilkan_menu_user():
    """Menampilkan menu untuk user"""
    print("=== MENU USER ===")
    print("1. Lihat Tanaman")
```

```
    print("2. Ubah Status Tanaman")
    print("3. Logout")
```

Fungsi ini hanya menampilkan daftar pilihan menu sesuai dengan peran pengguna:

- Admin memiliki hak penuh (tambah, ubah, hapus, lihat).
- User hanya bisa melihat dan mengubah status tanaman.

```
def tampilkan_data():
    """Menampilkan semua data tanaman dalam bentuk tabel"""
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== DAFTAR TANAMAN ===")
    if not tanaman_dict:
        print("Belum ada data tanaman.")
    else:
        table = PrettyTable()
        table.field_names = ["Nama Tanaman", "Warna", "Tanggal Ditanam",
"Status"]
        for nama, data in tanaman_dict.items():
            table.add_row([nama, data['warna'], data['tanggal'],
data['status']])
        print(table)
    input("\nTekan Enter untuk melanjutkan...")
```

- Membersihkan layar dengan os.system('cls'/'clear').
- Menampilkan semua data tanaman dalam bentuk tabel rapi menggunakan PrettyTable.
- PrettyTable membuat hasil seperti ini:

```
def ubah_status_tanaman(nama_tanaman):
    """Mengubah status tanaman tertentu"""
    global tanaman_dict
    if nama_tanaman in tanaman_dict:
        status_baru = input("Masukkan status baru (sehat/sakit/mati):
").lower().strip()
        if status_baru in ['sehat', 'sakit', 'mati']:
            tanaman_dict[nama_tanaman]['status'] = status_baru
            return f"Status {nama_tanaman} berhasil diubah menjadi
{status_baru}."
        else:
            return "Status tidak valid."
    else:
        return "Tanaman tidak ditemukan."
```

- Mengubah status tanaman tertentu (contohnya: dari “sehat” jadi “mati”).
- Menggunakan parameter nama_tanaman agar bisa dipanggil secara fleksibel di menu admin/user.
- Mengembalikan pesan hasil perubahan.

```
def tambah_tanaman():
    """Menambahkan tanaman baru"""
    global tanaman_dict
    os.system('cls' if os.name == 'nt' else 'clear')
    print("=== TAMBAH TANAMAN BARU ===")

    nama = input("Masukkan nama tanaman: ").strip()
    warna = input("Masukkan warna bunga: ").strip()
    tanggal = input("Masukkan tanggal ditanam (YYYY-MM-DD): ").strip()

    if nama in tanaman_dict:
        print("Tanaman sudah ada.")
    else:
        tanaman_dict[nama] = {"warna": warna, "tanggal": tanggal, "status":
"sehat"}
        print("Tanaman berhasil ditambahkan.")
    input("Tekan Enter untuk melanjutkan...")
```

- Menambahkan data baru ke tanaman_dict.
- Tidak butuh parameter, semua input diambil langsung dari pengguna.
- Jika nama tanaman sudah ada, program menolak duplikasi.

```
def hitung_tanaman_mati(daftar_tanaman, indeks=0, jumlah=0):
    """Fungsi rekursif untuk menghitung jumlah tanaman mati"""
    if indeks == len(daftar_tanaman):
        return jumlah
    nama = daftar_tanaman[indeks]
    if tanaman_dict[nama]["status"] == "mati":
        jumlah += 1
    return hitung_tanaman_mati(daftar_tanaman, indeks + 1, jumlah)
```

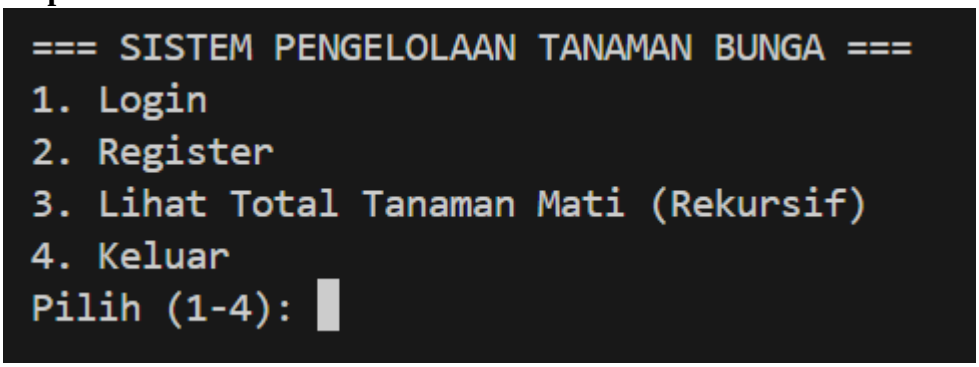
- Menghitung jumlah tanaman yang statusnya mati.
- Menggunakan rekursi (fungsi memanggil dirinya sendiri).
- Parameter:

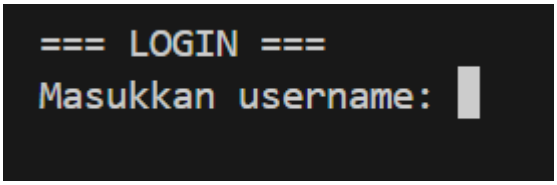
- daftar_tanaman: daftar nama tanaman.
- indeks: posisi yang sedang dihitung.
- jumlah: total tanaman mati sejauh ini

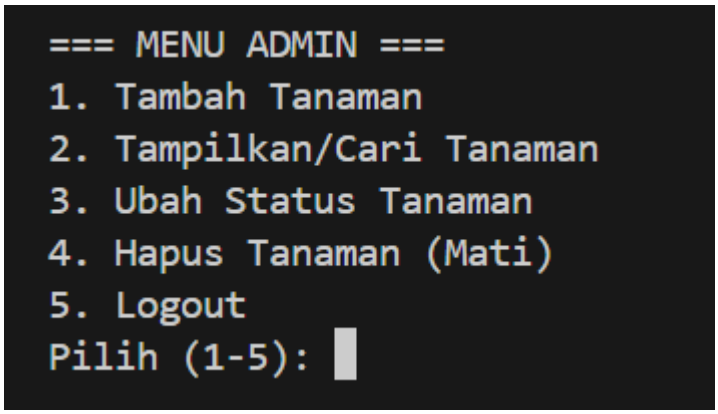
```
  print("Terima kasih telah menggunakan program ini!")
        break
```

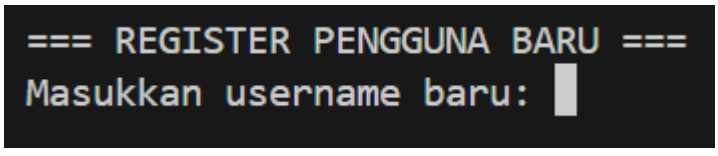
Jika user memilih opsi `4`, program menampilkan pesan perpisahan dan keluar dari loop:

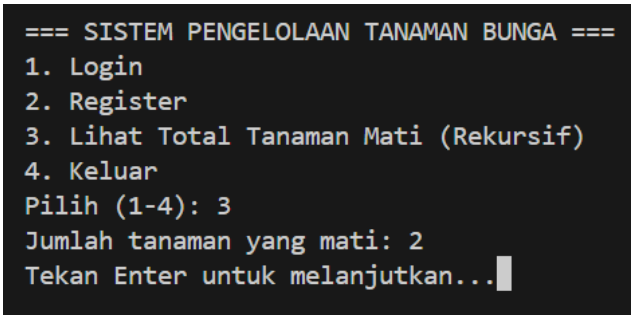
4. Output











```
=== SISTEM PENGELOLAAN TANAMAN BUNGA ===
1. Login
2. Register
3. Lihat Total Tanaman Mati (Rekursif)
4. Keluar
Pilih (1-4): 4
Terima kasih telah menggunakan program ini!
PS D:\praktikum-apd>
```

5. Git

```
PS D:\praktikum-apd> git init
Reinitialized existing Git repository in D:/praktikum-apd/.git/
```

```
PS D:\praktikum-apd> git add .
PS D:\praktikum-apd> git commit -m "posttest8"
[main a6d9198] posttest8
 1 file changed, 208 insertions(+)
 create mode 100644 post-test/post-test-apd-8/2509106010-Zihni Larasati-PT-8.py
```

```
PS D:\praktikum-apd> git push -u origin main
Enumerating objects: 7, done.
Counting objects: 100% (7/7), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 2.31 KiB | 787.00 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/Laras010/praktikum-apd-.git
   9c0bffc..a6d9198  main -> main
```