

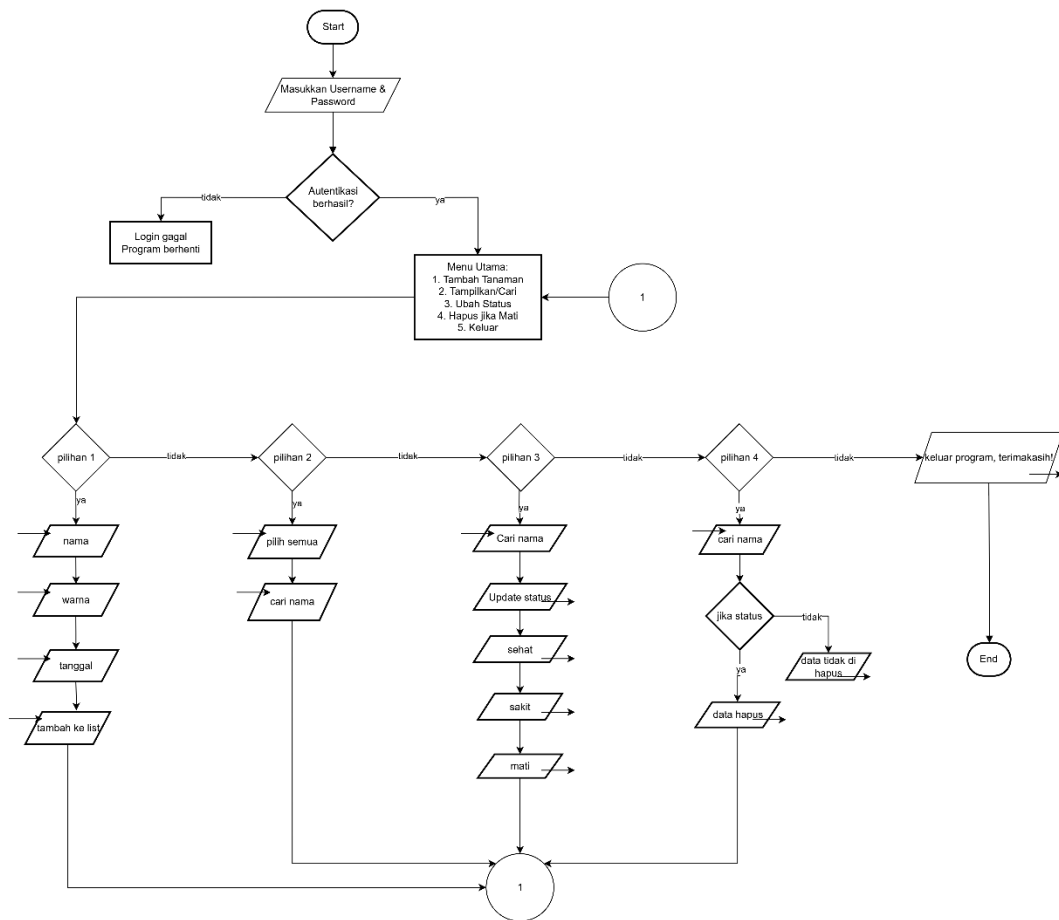
**LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN DASAR**



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

**PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025**

1. Flowchart



2. Deskripsi singkat program

Program ini adalah aplikasi sederhana untuk mengelola koleksi tanaman bunga menggunakan operasi CRUD (Create, Read, Update, Delete). Tema utamanya adalah "Pengelolaan Tanaman Bunga", di mana pengguna dapat menambahkan, melihat, memperbarui status kesehatan, dan menghapus tanaman berdasarkan kondisi tertentu. Program menggunakan struktur data berupa list of dictionaries untuk menyimpan informasi tanaman, dan dilengkapi dengan autentikasi login sederhana.

Tujuan program ini adalah membantu pengguna (dalam hal ini, satu pengguna tetap) dalam mengelola data tanaman bunga, termasuk mencatat nama, warna, tanggal tanam, dan status kesehatan. Program ini dirancang untuk berjalan di terminal dan menggunakan fungsi clear screen untuk memperbaiki tampilan.

Fungsi utama dari pemograman adalah:

- **clear_screen()**: Membersihkan layar terminal untuk membuat tampilan lebih rapi. Menggunakan **os.system()** untuk kompatibilitas dengan Windows atau Linux.
- **login()**: Memeriksa apakah input username dan password sesuai dengan yang ditentukan. Jika benar, mengembalikan **True**; jika tidak, mencetak pesan error dan mengembalikan **False**.
- **create_tanaman()**: Menambahkan tanaman baru ke **tanaman_list**. Pengguna diminta input nama, warna, dan tanggal; status awal selalu "Sehat".
- **read_tanaman()**: Menampilkan daftar tanaman atau mencari berdasarkan nama. Pengguna memilih opsi: tampilkan semua atau cari spesifik.
- **update_tanaman()**: Memperbarui status kesehatan tanaman (misalnya, dari "Sehat" menjadi "Sakit" atau "Mati") berdasarkan nama tanaman yang dimasukkan.
- **delete_tanaman()**: Menghapus tanaman dari **tanaman_list** hanya jika statusnya "Mati". Jika tanaman ditemukan tapi statusnya bukan "Mati", operasi dibatalkan.
- **main()**: Fungsi utama yang mengatur alur program. Memanggil **login()** terlebih dahulu, kemudian memasuki loop menu jika login berhasil.

3. Source code

```
import os

USERNAME = "AADPA"
PASSWORD = "Andi Ahmad dzaky P A"

tanaman_list = [{ 'nama': 'Bunga Mawar', 'warna': 'merah', 'tanggal': '2022-08-27', 'status': 'sehat' },
                 { 'nama': 'Bunga Tulip', 'warna': 'pink', 'tanggal': '2024-05-30', 'status': 'sehat' },
                 { 'nama': 'Bunga Kamboja ', 'warna': 'putih', 'tanggal': '2020-09-24', 'status': 'sehat' },
                 { 'nama': 'Bunga Matahari', 'warna': 'kuning', 'tanggal': '2024-04-02', 'status': 'mati' },
                 { 'nama': 'Bunga Alamanda', 'warna': 'ungu', 'tanggal': '20233-02-07', 'status': 'mati' }]

def clear_screen():
    """Fungsi untuk membersihkan layar terminal."""
    os.system('cls' if os.name == 'nt' else 'clear')

def login():
    """Fungsi untuk autentikasi pengguna."""
    input_username = input("Masukkan username: ")
    input_password = input("Masukkan password: ")
    if input_username == USERNAME and input_password == PASSWORD:
        return True
    else:
        print("Login gagal. Username atau password salah.")
        return False

def create_tanaman():
    """Fungsi untuk menambahkan tanaman baru."""
    nama = input("Masukkan nama tanaman: ")
    warna = input("Masukkan warna bunga: ")
    tanggal = input("Masukkan tanggal ditanam (contoh: YYYY-MM-DD): ")
    status = "Sehat" # Status awal selalu "Sehat"
    tanaman_list.append({ 'nama': nama, 'warna': warna, 'tanggal': tanggal, 'status': status })
    print("Tanaman berhasil ditambahkan.")

def read_tanaman():
    """Fungsi untuk menampilkan daftar tanaman atau mencari berdasarkan nama."""
    if not tanaman_list:
        print("Tidak ada tanaman dalam koleksi.")
        return
```

```

    pilihan = input("1. Tampilkan semua tanaman\n2. Cari berdasarkan
nama\nPilih (1/2): ")
    if pilihan == '1':
        for tanaman in tanaman_list:
            print(f"nama: {tanaman['nama']], warna: {tanaman['warna']],
tanggal: {tanaman['tanggal']], status: {tanaman['status']}")

    elif pilihan == '2':
        nama_cari = input("Masukkan nama tanaman: ")
        for tanaman in tanaman_list:
            if tanaman['nama'].lower() == nama_cari.lower():
                print(f"nama: {tanaman['nama']], warna: {tanaman['warna']],
tanggal: {tanaman['tanggal']], status: {tanaman['status']}")

        print("Tanaman tidak ditemukan.")
    else:
        print("Pilihan tidak valid.")

def update_tanaman():
    """Fungsi untuk mengubah status kesehatan tanaman."""
    nama = input("Masukkan nama tanaman yang ingin diupdate: ")
    for tanaman in tanaman_list:
        if tanaman['nama'].lower() == nama.lower():
            status_baru = input("Masukkan status kesehatan baru
(Sehat/Sakit/Mati): ")
            if status_baru in ['Sehat', 'Sakit', 'Mati']:
                tanaman['status'] = status_baru
                print("Status kesehatan tanaman berhasil diupdate.")
            else:
                print("Status tidak valid. Gunakan Sehat, Sakit, atau Mati.")
        return
    print("Tanaman tidak ditemukan.")

def delete_tanaman():
    """Fungsi untuk menghapus tanaman jika statusnya 'Mati'."""
    nama = input("Masukkan nama tanaman yang ingin dihapus: ")
    for i, tanaman in enumerate(tanaman_list):
        if tanaman['nama'].lower() == nama.lower():
            if tanaman['status'] == 'Mati':
                del tanaman_list[i]
                print("Tanaman berhasil dihapus dari koleksi.")
            else:
                print("Tanaman belum mati, tidak bisa dihapus.")
        return
    print("Tanaman tidak ditemukan.")

def main():
    if not login():

```

```

        return

    while True:
        clear_screen()
        print("=== Pengelolaan Tanaman Bunga ===")
        print("1. Tambah Tanaman")
        print("2. Tampilkan atau Cari Tanaman")
        print("3. Ubah Status Kesehatan")
        print("4. Hapus Tanaman jika Mati")
        print("5. Keluar Program")
        pilihan = input("Pilih operasi 1-5: ")

        if pilihan == '1':
            create_tanaman()
        elif pilihan == '2':
            read_tanaman()
        elif pilihan == '3':
            update_tanaman()
        elif pilihan == '4':
            delete_tanaman()
        elif pilihan == '5':
            print("Keluar program. Terima kasih!")
            break
        else:
            print("Pilihan tidak valid. Silakan pilih 1-5.")

        input("Tekan Enter untuk melanjutkan...")

if __name__ == "__main__":
    main()
import os

USERNAME = "AADPA"
PASSWORD = "Andi Ahmad dzaky P A"

tanaman_list = [{ 'nama': 'Bunga Mawar', 'warna': 'merah', 'tanggal': '2022-08-27', 'status': 'sehat' },
                 { 'nama': 'Bunga Tulip', 'warna': 'pink', 'tanggal': '2024-05-30', 'status': 'sehat' },
                 { 'nama': 'Bunga Kamboja ', 'warna': 'putih', 'tanggal': '2020-09-24', 'status': 'sehat' },
                 { 'nama': 'Bunga Matahari', 'warna': 'kuning', 'tanggal': '2024-04-02', 'status': 'mati' },
                 { 'nama': 'Bunga Alamanda', 'warna': 'ungu', 'tanggal': '20233-02-07', 'status': 'mati' } ]

def clear_screen():
    """Fungsi untuk membersihkan layar terminal."""

```

```

os.system('cls' if os.name == 'nt' else 'clear')

def login():
    """Fungsi untuk autentikasi pengguna."""
    input_username = input("Masukkan username: ")
    input_password = input("Masukkan password: ")
    if input_username == USERNAME and input_password == PASSWORD:
        return True
    else:
        print("Login gagal. Username atau password salah.")
        return False

def create_tanaman():
    """Fungsi untuk menambahkan tanaman baru."""
    nama = input("Masukkan nama tanaman: ")
    warna = input("Masukkan warna bunga: ")
    tanggal = input("Masukkan tanggal ditanam (contoh: YYYY-MM-DD): ")
    status = "Sehat" # Status awal selalu "Sehat"
    tanaman_list.append({'nama': nama, 'warna': warna, 'tanggal': tanggal,
'status': status})
    print("Tanaman berhasil ditambahkan.")

def read_tanaman():
    """Fungsi untuk menampilkan daftar tanaman atau mencari berdasarkan
nama."""
    if not tanaman_list:
        print("Tidak ada tanaman dalam koleksi.")
        return
    pilihan = input("1. Tampilkan semua tanaman\n2. Cari berdasarkan
nama\nPilih (1/2): ")
    if pilihan == '1':
        for tanaman in tanaman_list:
            print(f"nama: {tanaman['nama']], warna: {tanaman['warna']],
tanggal: {tanaman['tanggal']], status: {tanaman['status']}")

    elif pilihan == '2':
        nama_cari = input("Masukkan nama tanaman: ")
        for tanaman in tanaman_list:
            if tanaman['nama'].lower() == nama_cari.lower():
                print(f"nama: {tanaman['nama']], warna: {tanaman['warna']],
tanggal: {tanaman['tanggal']], status: {tanaman['status']}")

        print("Tanaman tidak ditemukan.")
    else:
        print("Pilihan tidak valid.")

def update_tanaman():
    """Fungsi untuk mengubah status kesehatan tanaman."""

```

```

    nama = input("Masukkan nama tanaman yang ingin diupdate: ")
    for tanaman in tanaman_list:
        if tanaman['nama'].lower() == nama.lower():
            status_baru = input("Masukkan status kesehatan baru
(Sehat/Sakit/Mati): ")
            if status_baru in ['Sehat', 'Sakit', 'Mati']:
                tanaman['status'] = status_baru
                print("Status kesehatan tanaman berhasil diupdate.")
            else:
                print("Status tidak valid. Gunakan Sehat, Sakit, atau Mati.")
            return
    print("Tanaman tidak ditemukan.")

def delete_tanaman():
    """Fungsi untuk menghapus tanaman jika statusnya 'Mati'."""
    nama = input("Masukkan nama tanaman yang ingin dihapus: ")
    for i, tanaman in enumerate(tanaman_list):
        if tanaman['nama'].lower() == nama.lower():
            if tanaman['status'] == 'Mati':
                del tanaman_list[i]
                print("Tanaman berhasil dihapus dari koleksi.")
            else:
                print("Tanaman belum mati, tidak bisa dihapus.")
            return
    print("Tanaman tidak ditemukan.")

def main():
    if not login():
        return

    while True:
        clear_screen()
        print("=== Pengelolaan Tanaman Bunga ===")
        print("1. Tambah Tanaman")
        print("2. Tampilkan atau Cari Tanaman")
        print("3. Ubah Status Kesehatan")
        print("4. Hapus Tanaman jika Mati")
        print("5. Keluar Program")
        pilihan = input("Pilih operasi 1-5: ")

        if pilihan == '1':
            create_tanaman()
        elif pilihan == '2':
            read_tan
aman()
        elif pilihan == '3':
            update_tanaman()
        elif pilihan == '4':

```



```

        delete_tanaman()
    elif pilihan == '5':
        print("Keluar program. Terima kasih!")
        break
    else:
        print("Pilihan tidak valid. Silakan pilih 1-5.")

    input("Tekan Enter untuk melanjutkan...")

if __name__ == "__main__":
    main()

```

4. Hasil Output

```

=== Pengelolaan Tanaman Bunga ===
1. Tambah Tanaman
2. Tampilkan atau Cari Tanaman
3. Ubah Status Kesehatan
4. Hapus Tanaman jika Mati
5. Keluar Program
Pilih operasi 1-5: 1
Masukkan nama tanaman: Bunga Melati
Masukkan warna bunga: Putih
Masukkan tanggal ditanam (contoh: YYYY-MM-DD): 20025-10-15
Tanaman berhasil ditambahkan.
Tekan Enter untuk melanjutkan...

```

```

=== Pengelolaan Tanaman Bunga ===
1. Tambah Tanaman
2. Tampilkan atau Cari Tanaman
3. Ubah Status Kesehatan
4. Hapus Tanaman jika Mati
5. Keluar Program
Pilih operasi 1-5: 2
1. Tampilkan semua tanaman
2. Cari berdasarkan nama
Pilih (1/2): 1
nama: Bunga Mawar, warna: merah, tanggal: 2022-08-27, status: sehat
nama: Bunga Melati, warna: Putih, tanggal: 20025-10-15, status: Sehat
Tekan Enter untuk melanjutkan...

```

=== Pengelolaan Tanaman Bunga ===

1. Tambah Tanaman
2. Tampilkan atau Cari Tanaman
3. Ubah Status Kesehatan
4. Hapus Tanaman jika Mati
5. Keluar Program

Pilih operasi 1-5: 2

1. Tampilkan semua tanaman
2. Cari berdasarkan nama

Pilih (1/2): 2

Masukkan nama tanaman: anggrek

Tanaman tidak ditemukan.

Tekan Enter untuk melanjutkan...

=== Pengelolaan Tanaman Bunga ===

1. Tambah Tanaman
2. Tampilkan atau Cari Tanaman
3. Ubah Status Kesehatan
4. Hapus Tanaman jika Mati
5. Keluar Program

Pilih operasi 1-5: 3

Masukkan nama tanaman yang ingin diupdate: Bunga Mawar

Masukkan status kesehatan baru (Sehat/Sakit/Mati): Sakit

Status kesehatan tanaman berhasil diupdate.

Tekan Enter untuk melanjutkan...

=== Pengelolaan Tanaman Bunga ===

1. Tambah Tanaman
2. Tampilkan atau Cari Tanaman
3. Ubah Status Kesehatan
4. Hapus Tanaman jika Mati
5. Keluar Program

Pilih operasi 1-5: 4

Masukkan nama tanaman yang ingin dihapus: Bunga Mawar

Tanaman belum mati, tidak bisa dihapus.

Tekan Enter untuk melanjutkan...

```
=== Pengelolaan Tanaman Bunga ===
```

1. Tambah Tanaman
2. Tampilkan atau Cari Tanaman
3. Ubah Status Kesehatan
4. Hapus Tanaman jika Mati
5. Keluar Program

```
Pilih operasi 1-5: 5
```

```
Keluar program. Terima kasih!
```

```
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 "posttest5"
[main 5510759] posttest5
1 file changed, 114 insertions(+)
create mode 100644 post-test/post-test-apd-5/2509106010-PT-apd-5.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% (4/4), done.
Writing objects: 100% (5/5), 1.65 KiB | 421.00 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
4c16c64..5510759 main -> main
branch 'main' set up to track 'origin/main'.
```

```
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% (4/4), done.
Writing objects: 100% (5/5), 1.65 KiB | 421.00 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
4c16c64..5510759 main -> main
branch 'main' set up to track 'origin/main'.
```