

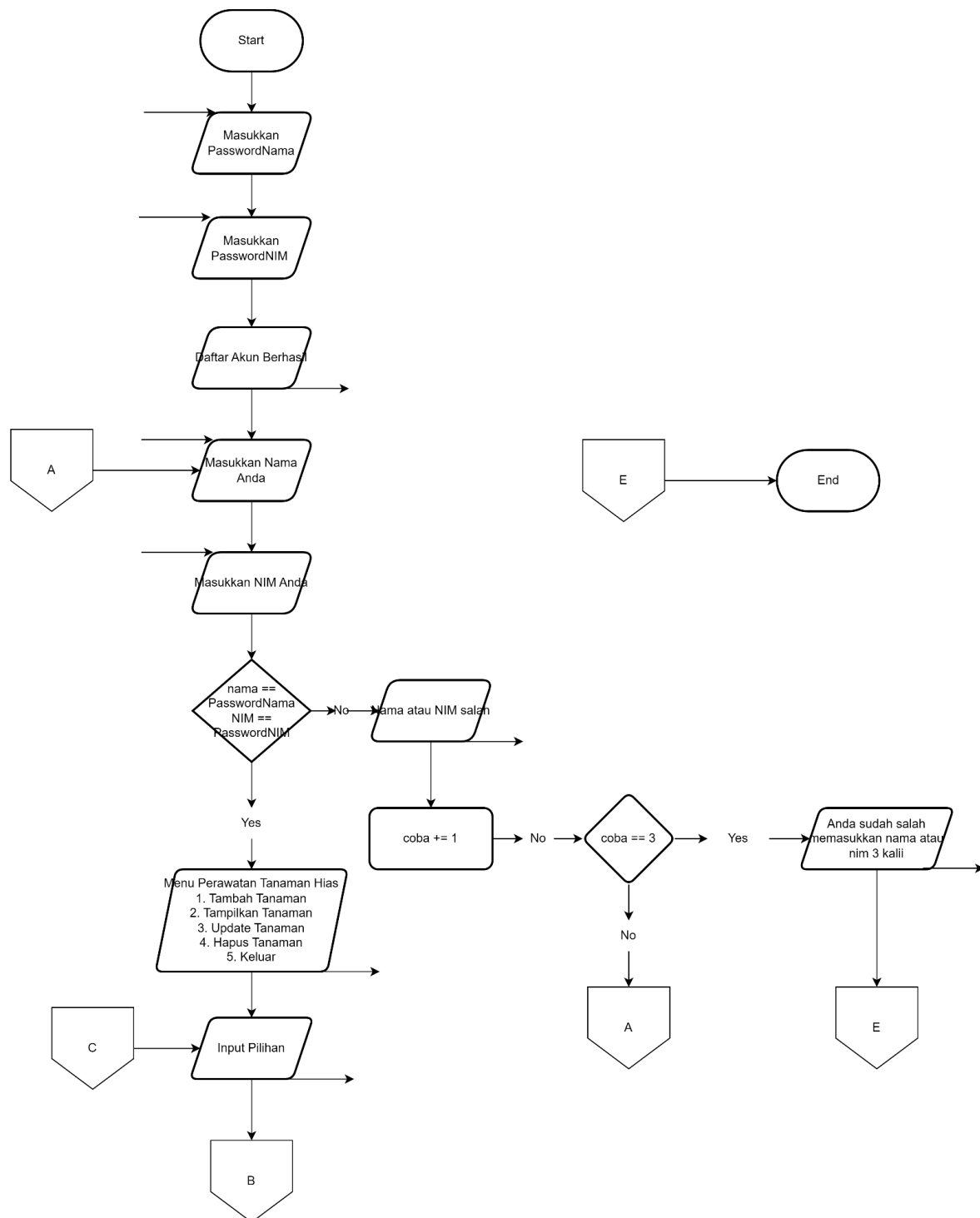
LAPORAN PRAKTIKUM
POSTTEST 2
ALGORITMA PEMROGRAMAN LANJUT



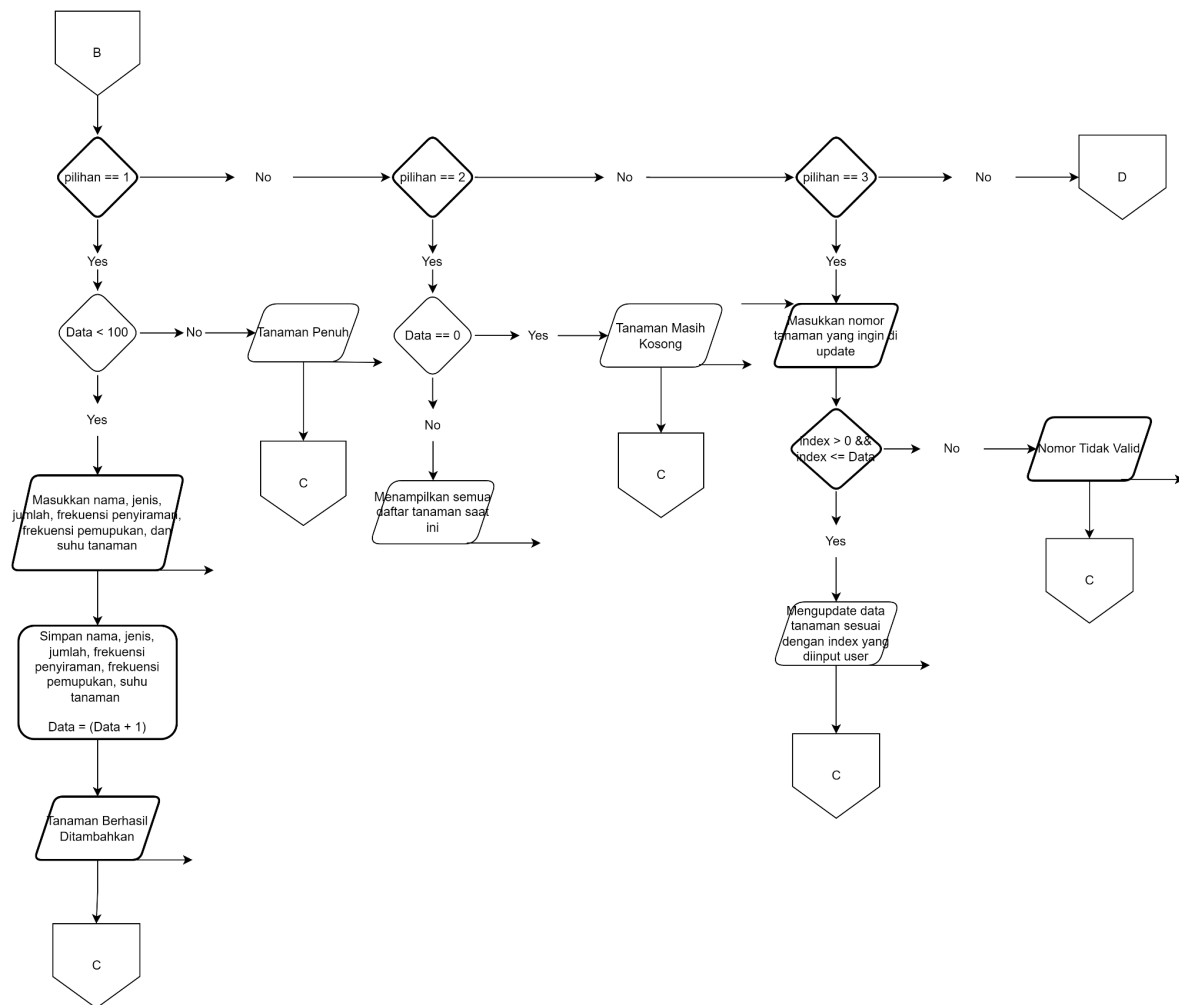
Disusun oleh:
Dimas Firjatullah Islamay (2409106057)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

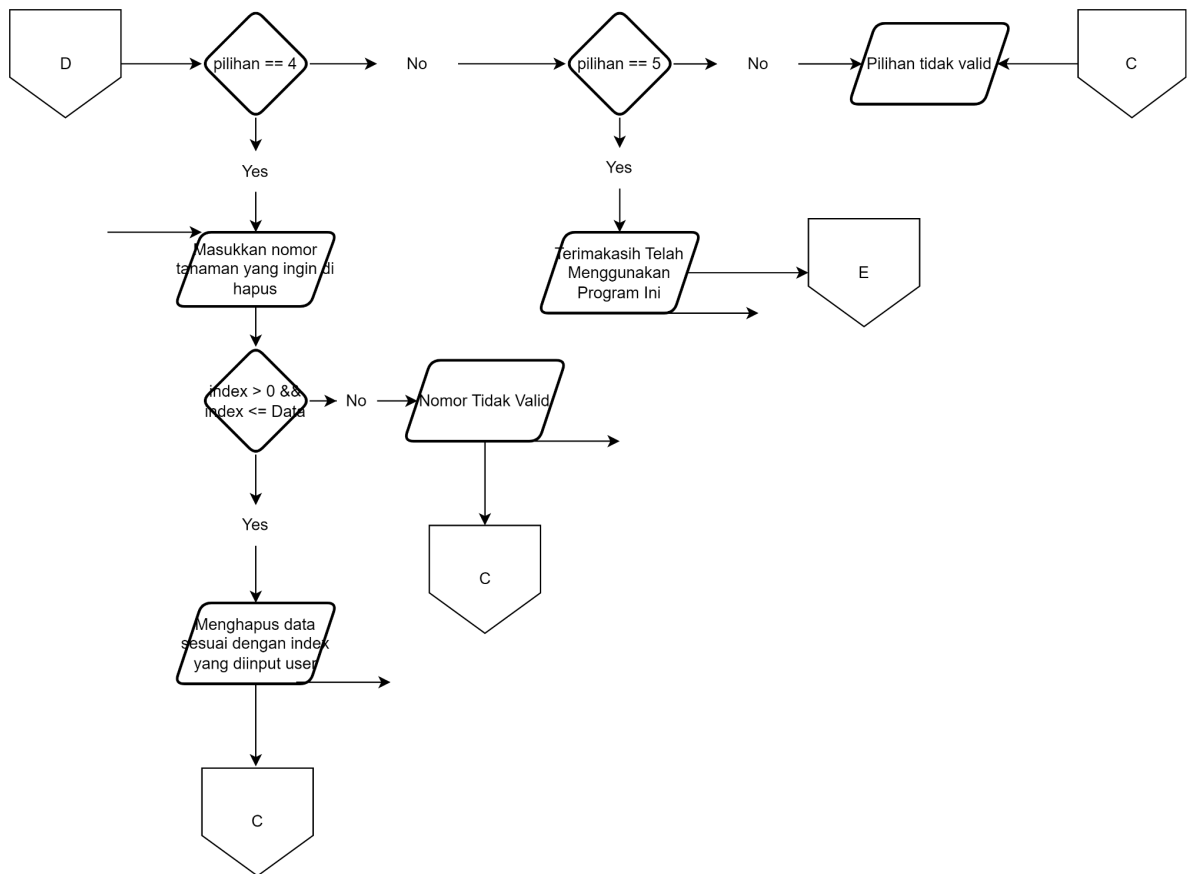
1. Flowchart



Gambar 1.1 Akses login



Gambar 1.2 Fitur Tambah, Tampilkan, Update Tanaman



Gambar 1.2 Fitur Hapus Tanaman dan Keluar

2. Analisis Program

2.1 Deskripsi Singkat Program

Program ini merupakan program manajemen perawatan tanaman hias yang berfungsi untuk mendata nama, jenis, jumlah, frekuensi penyiraman, frekuensi pemupukan, dan suhu ideal tanaman dari seseorang yang suka mengoleksi tanaman hias. Program ini menerapkan beberapa fitur diantaranya seperti sistem login yang harus mencocokkan username dan password yang sudah diatur, fitur CRUD yang dilengkapi index, terdapat array 1 dan 2 dimensi sebagai tempat untuk mendata tanaman, dan terdapat fitur looping yang dimana program akan mengulang terus kecuali user salah memasukkan username/password sebanyak 3 kali atau user memilih keluar program.

2.2 Penjelasan Alur & Algoritma

Berikut beberapa penjelasan alur dari program manajemen perawatan tanaman hias sederhana saya :

1. Saya mendeklarasikan beberapa variabel dengan tipe data yang berbeda-beda seperti tipe data string terdiri dari variabel Nama, NIM, PasswordNama, PasswordNIM, dan DaftarTanaman yang merupakan array 2 dimensi yang terdiri dari 100 baris dan 3 kolom. Lalu ada tipe data integer yang terdiri dari variabel FrekuensiSiram yang merupakan array 1 dimensi dengan ukuran 100, FrekuensiPupuk yang merupakan array 1 dimensi dengan ukuran 100, Coba yang bernilai 0, Pilihan, dan Data yang bernilai 0. Terakhir ada tipe data float yang terdiri dari variabel Suhu yang merupakan array 1 dimensi dengan ukuran 100.
2. Selanjutnya ada blok daftar akun dimana user terlebih dahulu mendaftarkan Nama dan akan disimpan di dalam variabel PasswordNama dan mendaftarkan NIM dan akan dimasukkan ke dalam variabel PasswordNIM.
3. Selanjutnya saya membuat blok kode perulangan while yang didalamnya berfokus pada memasukkan Nama dan NIM. Jadi jika if (Nama == PasswordNama && NIM == PasswordNIM) maka akan muncul output login berhasil dan loop akan diberhentikan menggunakan break, ini berarti apa yang diinput user sudah sesuai dengan PasswordNama dan PasswordNim yang telah ditetapkan di awal tadi. Jika else

atau salah maka dia akan terus looping dan Coba++ yang berarti variabel Coba yang nilai awalnya 0 akan bertambah 1.

4. Selanjutnya terdapat blok kode if (Coba == 3) yang didalamnya berisi output dan return 0, artinya jika nilai variabel coba yang awalnya bernilai 0 telah bernilai 3 maka user sudah salah memasukkan Nama atau NIM sebanyak 3 kali dan program akan otomatis diberhentikan.
5. Selanjutnya ada blok kode perulangan while (true) yang akan terus mengulang kode sampai bertemu dengan break. Lalu terdapat 5 pilihan menu utama tanaman hias, user tinggal memilih mau masuk menu seberapa dengan menginput dan akan disimpan ke variabel Pilihan.
6. Jika variabel pilihan bernilai 1, maka user akan dimasukkan ke blok kode tambah tanaman. Didalamnya terdapat blok kode if (Data < 100), jadi disini Data itu variabel kita untuk menyimpan jumlah tanaman yang telah dimasukkan, jika jumlah tanamannya kurang dari 100 maka user lanjut untuk memasukkan nama tanaman dan disimpan di DaftarTanaman[Data][0], kemudian user lanjut memasukkan jenis tanaman yang disimpan di DaftarTanaman[Data][1], kemudian user lanjut memasukkan jumlah tanaman yang disimpan di DaftarTanaman[Data][2], jadi maksudnya itu semua termasuk array 2d, kan Data itu tadi bernilai 0, maka dia indexnya [0][0], [0][1], [0][2]. Kemudian user dapat memasukkan frekuensi penyiraman dan disimpan ke array FrekuensiSiram[Data], kemudian user dapat memasukkan frekuensi pemupukan dan disimpan ke array FrekuensiPupuk[Data], kemudian user dapat memasukkan suhu tanaman dan disimpan ke Suhu[Data]. Nah jadi yang ini termasuk array 1d, nanti tanaman pertama disimpan ke index 0, lalu jika ada tanaman kedua maka disimpan ke index 1 dan seterusnya. Jika sudah menginput semuanya maka akan ada proses Data++ ini maksudnya data akan bertambah 1 jadi sudah ada 1 tanaman baru yang ditambahkan. Jika jumlah tanamannya lebih dari 100 maka akan pindah ke bagian else, yang berisi output Tanaman Penuh dan akan looping kembali ke menu utama.
7. Jika variabel pilihan bernilai 2, maka user akan dimasukkan ke blok kode tampilkan tanaman. Didalamnya terdapat blok kode if (Data == 0), jika Datanya 0 atau tanamannya memang 0 (gak ada tanamannya) maka akan muncul output Tanaman Masih Kosong dan akan looping kembali ke menu utama. Jika else atau tidak, maka akan muncul daftar tanaman saat ini diikuti for (int i = 0; i < Data; i++), maksudnya ini adalah for loop yang tersimpan di dalam array, maksudnya int i = 0 berarti variabel

i dimulai dari 0 karena index memang selalu dimulai dari 0, lalu selama $Data - 1$ masih lebih banyak dari i , maka akan looping terus lalu i bertambah 1 setiap putaran loop, jadi ujung ujungnya i nilainya akan sama dengan $Data - 1$ (harus dikurang 1 biar menyamakan karena indeks mulainya dari 0 bukan dari 1) dan berhenti sampai disitu. Didalam proses loop tersebut terdapat `cout << i + 1 << "`. " maksudnya adalah $i + 1$ diperlukan karena indeks selalu dimulai dari 0 makanya ditambah 1 biar pas nanti ditampilkan jadi nomor 1 bukan nomor 0. " " ini hanya untuk keperluan biar sedikit rapi. Lalu terdapat `DaftarTanaman[1][0]`, `DaftarTanaman[i][1]` dan seterusnya. Jika i bernilai 0 maka akan menampilkan DaftarTanaman pertama yang berisi nama, jenis, dan jumlah. Jika i bernilai 1 maka akan menampilkan DaftarTanaman kedua dan seterusnya. Begitupun dengan `FrekuensiSiram[i]` dan seterusnya, konsepnya tetap sama yang membedakan cuma yang satu array 2d dan yang satu array 1d. Setelah selesai akan looping kembali ke menu utama.

8. Jika variabel pilihan bernilai 3, maka user akan dimasukkan ke blok kode update tanaman. Didalamnya terdapat variabel index yang berfungsi menyimpan input nomor tanaman dari user lalu terdapat `if (index > 0 && index <= Data)`, maksudnya adalah index harus lebih besar dari 0 dan kurang dari sama dengan $Data$ (tidak boleh lebih dari jumlah yang ada), lalu kita tinggal mengupdate saja tanaman nomor berapa yang kita mau, misal kita ingin mengupdate nama tanaman maka akan muncul `DaftarTanaman[index - 1][0]`, ini menjelaskan bahwa data tanaman akan diambil dari DaftarTanaman, $index - 1$ karena nomor urutnya dimulai dari 1, tetapi arraynya dimulai dari 0, makanya dikurang 1 indexnya. `[0]` disini maksudnya index khusus nama tanaman, kalo `[1]` maka jenis tanaman dan seterusnya. kalau array 1 dimensi maka hanya membutuhkan `[index - 1]` saja karena dia array 1 dimensi bukan 2 dimensi. jika syarat `if` tadi tidak terpenuhi maka dia akan masuk ke `else` dan muncul output nomor tidak valid dan akan looping kembali ke menu utama.
9. Jika variabel pilihan bernilai 4, maka user akan dimasukkan ke blok kode hapus tanaman. Didalamnya juga terdapat variabel `int index` fungsinya juga sama intinya agar user bisa menginput angka nomor tanaman yang mau di hapus, setelah itu juga membutuhkan persyaratan `if` yang sama yaitu `if (index > 0 && index <= Data)` dan fungsinya juga sama dengan blok kode update data. Setelah itu ada `for (int i = index - 1; i < Data - 1; i++)`. Maksud dari kode adalah $index - 1$ karena nomor urutnya dimulai dari 1, tetapi arraynya dimulai dari 0, makanya dikurang 1 indexnya. Lalu selama $Data - 1$ masih lebih banyak dari i , maka akan looping terus lalu i bertambah 1

setiap putaran loop, jadi ujung ujungnya i nilainya akan sama dengan Data - 1 (harus dikurang 1 biar menyamakan karena indeks mulainya dari 0 bukan dari 1) dan berhenti sampai disitu. Didalam for loop nya sendiri terdapat `DaftarTanaman[i][0] = DaftarTanaman[i + 1][0]`, maksudnya adalah i + 1 digunakan untuk menunjuk ke baris berikutnya dalam DaftarTanaman sehingga `DaftarTanaman[i + 1][0]` akan berisi data nama tanaman di baris berikutnya lalu nilai dari `DaftarTanaman[i + 1][0]` akan disalin ke `DaftarTanaman[i][0]` jadi nanti elemen di baris i akan berisi nama tanaman dari baris i + 1 jadi intinya ketika satu data tanaman dihapus, maka data setelahnya akan menggantikan posisi tanaman yang dihapus tersebut. Yang lain seperti `DaftarTanaman[i][1]` dan seterusnya mengikuti konsep dan cara yang sama. Lalu terdapat `Data--`, ini menandakan bahwa satu elemen telah terhapus. Jika else atau tidak memenuhi syarat (`index > 0 && index <= Data`) maka akan menampilkan nomor tidak valid dan akan looping kembali ke menu utama

10. Jika variabel pilihan bernilai 5, maka akan menampilkan output Terimakasih Telah Menggunakan Program Ini, Ini artinya user telah memilih keluar dari program dan looping akan dihentikan menggunakan fungsi break.
11. Jika variabel pilihan bernilai apapun selain 1, 2, 3, 4, dan 5 maka dia akan masuk ke bagian else dari blok kode perulangan while (true) yang berada di langkah ketiga. User akan diperlihatkan output Pilihan Tidak Valid dan program akan mengulang kembali ke menu utama.

3. Source Code

A. Daftar Akun

Fitur ini digunakan agar user dapat masuk ke program dengan cara mendaftarkan Nama dan Nim nya terlebih dahulu.

Source Code:

```
cout << "===== Daftar Akun =====" << endl;
cout << "Masukkan Nama Anda: ";
cin >> PasswordNama;
cout << "Masukkan NIM Anda: ";
cin >> PasswordNIM;
cout << "Daftar Akun Berhasil" << endl;
```

Gambar 3.1

B. Login Akun

Fitur ini digunakan untuk memvalidasi user yang ingin menggunakan aplikasi melalui proses autentikasi dengan mengecek nama dan nim.

Source Code:

```
while (Coba < 3)
{
    cout << "=== Login Akun ===" << endl;
    cout << "Masukkan Nama Anda: ";
    cin >> Nama;
    cout << "Masukkan NIM Anda: ";
    cin >> NIM;

    if (Nama == PasswordNama && NIM == PasswordNIM)
    {
        cout << "Login Berhasil" << endl;
        break;
    }
    else
    {
        cout << "Nama Atau NIM Salah" << endl;
        Coba++;
    }
}

if (Coba == 3)
{
    cout << "Anda Salah Memasukkan Nama Atau NIM 3 Kali" << endl;
```

```

    return 0;
}

```

Gambar 3.2

C. Menu Utama

Terdapat beberapa menu berisi fitur-fitur yang dapat dipilih oleh user setelah berhasil login.

Source Code:

```

while (true)
{
    cout << "=== MENU TANAMAN HIAS ===" << endl;
    cout << "1. Tambah Tanaman" << endl;
    cout << "2. Tampilkan Tanaman" << endl;
    cout << "3. Update Tanaman" << endl;
    cout << "4. Hapus Tanaman" << endl;
    cout << "5. Keluar" << endl;
    cout << "Pilih menu: ";
    cin >> Pilihan;
}

```

Gambar 3.3

D. Tambah Tanaman

Blok kode yang memiliki fitur create untuk menambah Nama, Jenis, Jumlah, FrekuensiSiram, FrekuensiPupuk, dan Suhu dari suatu tanaman.

Source Code:

```

if (Pilihan == 1)
{
    if (Data < 100)
    {
        cout << "Masukkan Nama Tanaman: ";
        cin >> DaftarTanaman[Data][0];
        cout << "Masukkan Jenis Tanaman: ";
        cin >> DaftarTanaman[Data][1];
        cout << "Masukkan Jumlah Tanaman: ";
        cin >> DaftarTanaman[Data][2];
        cout << "Masukkan Frekuensi Penyiraman (Berapa Kali Per
Minggu): ";
        cin >> FrekuensiSiram[Data];
        cout << "Masukkan Frekuensi Pemupukan (Berapa Kali Per Bulan):
";
        cin >> FrekuensiPupuk[Data];
        cout << "Masukkan Suhu Tanaman (Berapa Derajat Celcius): ";
        cin >> Suhu[Data];
        Data++;
    }
}

```

```

        cout << "Tanaman Berhasil Ditambahkan" << endl;
    }
    else
    {
        cout << "Tanaman Penuh" << endl;
    }
}

```

Gambar 3.4

E. Tampilkan Tanaman

Blok kode yang memiliki fitur read dimana user dapat melihat data-data tanaman saat ini.

Source Code:

```

else if (Pilihan == 2)
{
    if (Data == 0)
    {
        cout << "Tanaman Masih Kosong" << endl;
    }
    else
    {
        cout << "=== DAFTAR TANAMAN ===" << endl;
        for (int i = 0; i < Data; i++)
        {
            cout << i + 1 << ". "
                << DaftarTanaman[i][0] << " | "
                << DaftarTanaman[i][1] << " | "
                << DaftarTanaman[i][2] << " | "
                << FrekuensiSiram[i] << "kali/minggu | "
                << FrekuensiPupuk[i] << "kali/bulan | "
                << Suhu[i] << " Derajat Celcius" << endl;
        }
    }
}

```

Gambar 3.5

F. Update Tanaman

Blok kode yang memiliki fitur update pada tanaman dengan index yang dipilih oleh user.

Source Code:

```

else if (Pilihan == 3)
{
    int index;

```

```

        cout << "Masukkan Nomor Tanaman Yang Ingin Diupdate: ";
        cin >> index;

        if (index > 0 && index <= Data)
        {
            cout << "Masukkan Nama Tanaman Baru: ";
            cin >> DaftarTanaman[index - 1][0];
            cout << "Masukkan Jenis Tanaman Baru: ";
            cin >> DaftarTanaman[index - 1][1];
            cout << "Masukkan Jumlah Tanaman Baru: ";
            cin >> DaftarTanaman[index - 1][2];
            cout << "Masukkan Frekuensi Penyiraman Baru (Berapa Kali Per
Minggu): ";
            cin >> FrekuensiSiram[index - 1];
            cout << "Masukkan Frekuensi Pemupukan Baru (Berapa Kali Per
Bulan): ";
            cin >> FrekuensiPupuk[index - 1];
            cout << "Masukkan Suhu Tanaman (Berapa Derajat Celcius): ";
            cin >> Suhu[index - 1];

            cout << "Tanaman Berhasil Diperbarui" << endl;
        }
        else
        {
            cout << "Nomor Tidak Valid" << endl;
        }
    }
}

```

Gambar 3.6

G. Hapus Tanaman

Blok kode yang memiliki fitur delete pada tanaman dengan index yang dipilih oleh user

Source Code:

```

else if (Pilihan == 4)
{
    int index;
    cout << "Masukkan Nomor Tanaman Yang Ingin Dihapus: ";
    cin >> index;

    if (index > 0 && index <= Data)
    {
        for (int i = index - 1; i < Data - 1; i++)
        {
            DaftarTanaman[i][0] = DaftarTanaman[i + 1][0];
            DaftarTanaman[i][1] = DaftarTanaman[i + 1][1];
            DaftarTanaman[i][2] = DaftarTanaman[i + 1][2];
            FrekuensiSiram[i] = FrekuensiSiram[i + 1];
        }
    }
}

```

```

        FrekuensiPupuk[i] = FrekuensiPupuk[i + 1];
        Suhu[i] = Suhu[i + 1];
    }
    Data--;
    cout << "Tanaman Berhasil Dihapus" << endl;
}
else
{
    cout << "Nomor Tidak Valid" << endl;
}
}

```

Gambar 3.7

H. Menu Keluar

Menu ini digunakan untuk menyelesaikan program dan mengakhiri looping.

Source Code:

```

else if (Pilihan == 5)
{
    cout << "TerimaKasih Telah Menggunakan Program Ini" << endl;
    break;
}

```

Gambar 3.8

4. Uji Coba dan Hasil Output

4.1 Uji Coba

Berikut beberapa skenario yang akan digunakan dalam pengujian program saya:

1. User mendaftarkan akun nama dan nim ke program.
2. User login dengan memasukkan nama dan nim.
3. Ketika user berhasil memasukkan nama dan nim maka akan lanjut ke menu utama program.
4. Ketika user salah memasukkan nama atau nim kurang dari 3 kali percobaan maka program akan menampilkan pesan nama atau nim salah, dan program akan mengulang.
5. Ketika user salah memasukkan nama atau nim sebanyak 3 kali maka program akan berhenti.
6. Setelah user berhasil memasukkan nama dan nim dengan benar, user akan dibawa ke halaman menu utama tanaman hias dan user akan diberi pilihan menu.
7. Jika user memilih pilihan nomor 1, maka user akan memasuki halaman tambah tanaman, user akan diminta untuk menginput nama, jenis, jumlah, frekuensi penyiraman, frekuensi pupuk, dan suhu suatu tanaman, lalu program akan kembali ke menu utama.
8. Jika user memilih pilihan nomor 2, maka user akan memasuki halaman tampilkan tanaman, user akan diperlihatkan data tanaman saat ini, lalu program akan kembali ke menu utama.
9. Jika user memilih pilihan nomor 3, maka user akan memasuki halaman update tanaman, user akan diminta untuk mengupdate nomor tanaman yang telah ada dengan mengupdate nama, jenis, jumlah, frekuensi penyiraman, frekuensi pupuk, dan suhu tanaman, lalu program akan kembali ke menu utama.
10. Jika user memilih pilihan nomor 4, maka user akan memasuki halaman hapus tanaman, user akan diminta untuk menghapus nomor tanaman yang telah ada, lalu program akan kembali ke menu utama.
11. Jika user memilih pilihan nomor 5, maka user akan memasuki menu keluar program dan akan keluar program dan looping berhenti.
12. Jika user memilih pilihan selain nomor 1, 2, 3, 4, dan 5 maka akan muncul pesan pilihan tidak valid, lalu program akan kembali ke menu utama.

4.2 Hasil Output

```
===== Daftar Akun =====  
Masukkan Nama Anda: Dimas  
Masukkan NIM Anda: 2409106057  
Daftar Akun Berhasil
```

Gambar 4.1 Pendaftaran Akun

```
=== Login Akun ===  
Masukkan Nama Anda: Dimas  
Masukkan NIM Anda: 2409106057  
Login Berhasil
```

Gambar 4.2 Login Akun Berhasil

```
=== Login Akun ===  
Masukkan Nama Anda: Dimas  
Masukkan NIM Anda: 222222222  
Nama Atau NIM Salah  
=== Login Akun ===  
Masukkan Nama Anda: Cecep  
Masukkan NIM Anda: 2409106057  
Nama Atau NIM Salah
```

Gambar 4.3 Login Akun Gagal

```
=== Login Akun ===  
Masukkan Nama Anda: Dimas  
Masukkan NIM Anda: 222222222  
Nama Atau NIM Salah  
=== Login Akun ===  
Masukkan Nama Anda: Cecep  
Masukkan NIM Anda: 2409106057  
Nama Atau NIM Salah  
=== Login Akun ===  
Masukkan Nama Anda: adf  
Masukkan NIM Anda: 213  
Nama Atau NIM Salah  
Anda Salah Memasukkan Nama Atau NIM 3 Kali  
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-2>
```

Gambar 4.4 Program Terpaksa Berhenti

```
=== MENU TANAMAN HIAS ===  
1. Tambah Tanaman  
2. Tampilkan Tanaman  
3. Update Tanaman  
4. Hapus Tanaman  
5. Keluar
```

Gambar 4.3 Menu Utama


```
Pilih menu: 1
Masukkan Nama Tanaman: Mawar
Masukkan Jenis Tanaman: Bunga
Masukkan Jumlah Tanaman: 25
Masukkan Frekuensi Penyiraman (Berapa Kali Per Minggu): 4
Masukkan Frekuensi Pemupukan (Berapa Kali Per Bulan): 2
Masukkan Suhu Tanaman (Berapa Derajat Celcius): 30
Tanaman Berhasil Ditambahkan
```

Gambar 4.4 Tambah Tanaman

```
Pilih menu: 2
=== DAFTAR TANAMAN ===
1. Mawar | Bunga | 25 | 4kali/minggu | 2kali/bulan | 30 Derajat Celcius
```

Gambar 4.5 Tampilkan Tanaman

```
Pilih menu: 3
Masukkan Nomor Tanaman Yang Ingin Diupdate: 1
Masukkan Nama Tanaman Baru: Tulip
Masukkan Jenis Tanaman Baru: Bunga
Masukkan Jumlah Tanaman Baru: 12
Masukkan Frekuensi Penyiraman Baru (Berapa Kali Per Minggu): 3
Masukkan Frekuensi Pemupukan Baru (Berapa Kali Per Bulan): 3
Masukkan Suhu Tanaman (Berapa Derajat Celcius): 29
Tanaman Berhasil Diperbarui
```

Gambar 4.6 Update Tanaman

```
Pilih menu: 4  
Masukkan Nomor Tanaman Yang Ingin Dihapus: 1  
Tanaman Berhasil Dihapus
```

Gambar 4.7 Hapus Tanaman

```
=== MENU TANAMAN HIAS ===  
1. Tambah Tanaman  
2. Tampilkan Tanaman  
3. Update Tanaman  
4. Hapus Tanaman  
5. Keluar  
Pilih menu: 5  
TerimaKasih Telah Menggunakan Program Ini
```

Gambar 4.8 Keluar

```
Pilih menu: 6  
Pilihan Tidak Valid
```

Gambar 4.9 Pilihan Tidak Valid

5. Langkah-langkah GIT

```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-2> Git add .
```

Gambar 5.1 Git add

Git add digunakan untuk menambahkan file yang ingin di commit di step selanjutnya

```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-2> Git commit -m "Finish Posttest 2"
[main ef6fcf3] Finish Posttest 2
1 file changed, 0 insertions(+), 0 deletions(-)
```

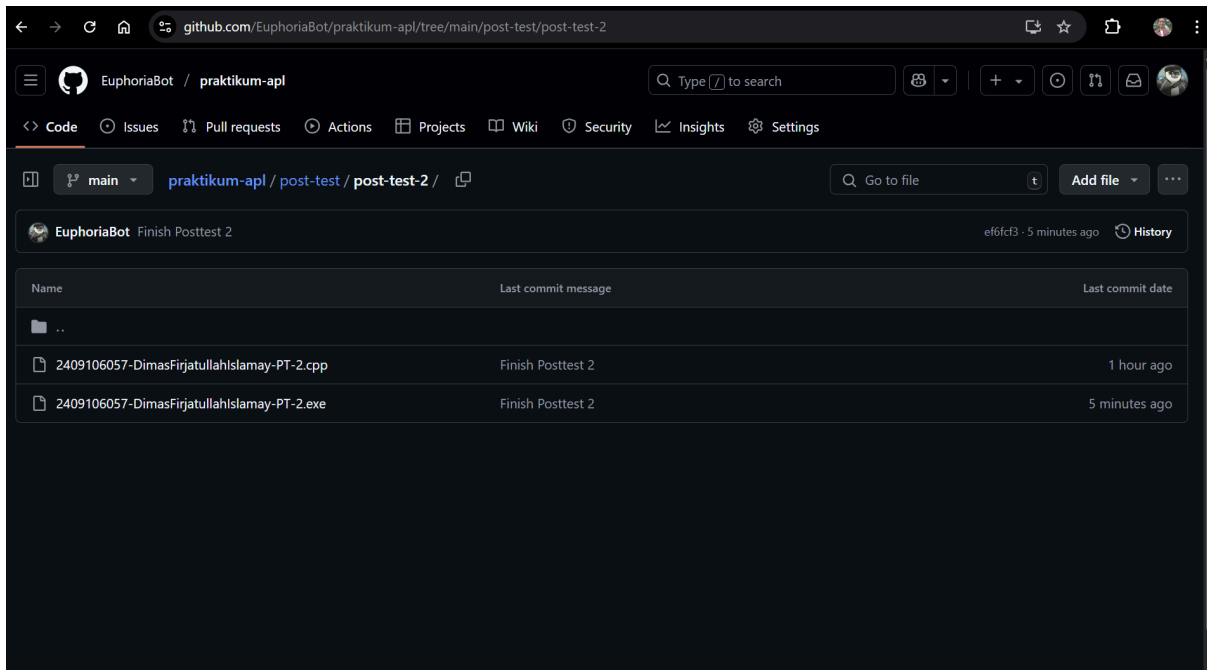
Gambar 5.3 Git commit

Git commit digunakan untuk menyimpan hasil file yang telah ditambahkan

```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-2> git push origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 22 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 439 bytes | 439.00 KiB/s, done.
Total 5 (delta 3), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (3/3), completed with 3 local objects.
To https://github.com/EuphoriaBot/praktikum-apl.git
  19b397e..ef6fcf3  main -> main
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-2> █
```

Gambar 5.5 Git push

Git push digunakan untuk mengupload seluruh commit yang sudah dilakukan dalam satu repository lokal



Gambar 5.6 Github

Berikut contoh jika langkah-langkah Git berhasil dilaksanakan