

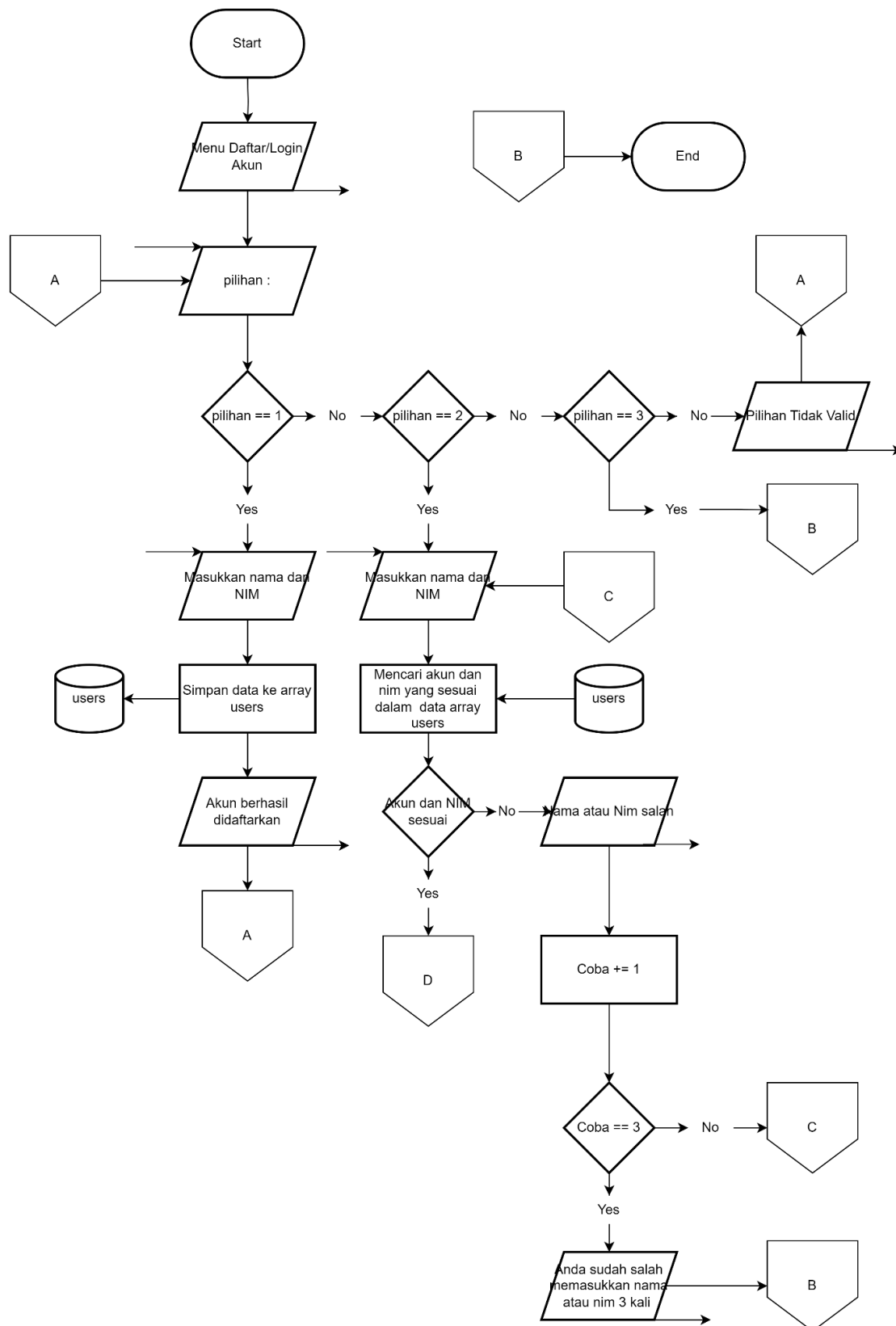
LAPORAN PRAKTIKUM
POSTTEST 3
ALGORITMA PEMROGRAMAN LANJUT



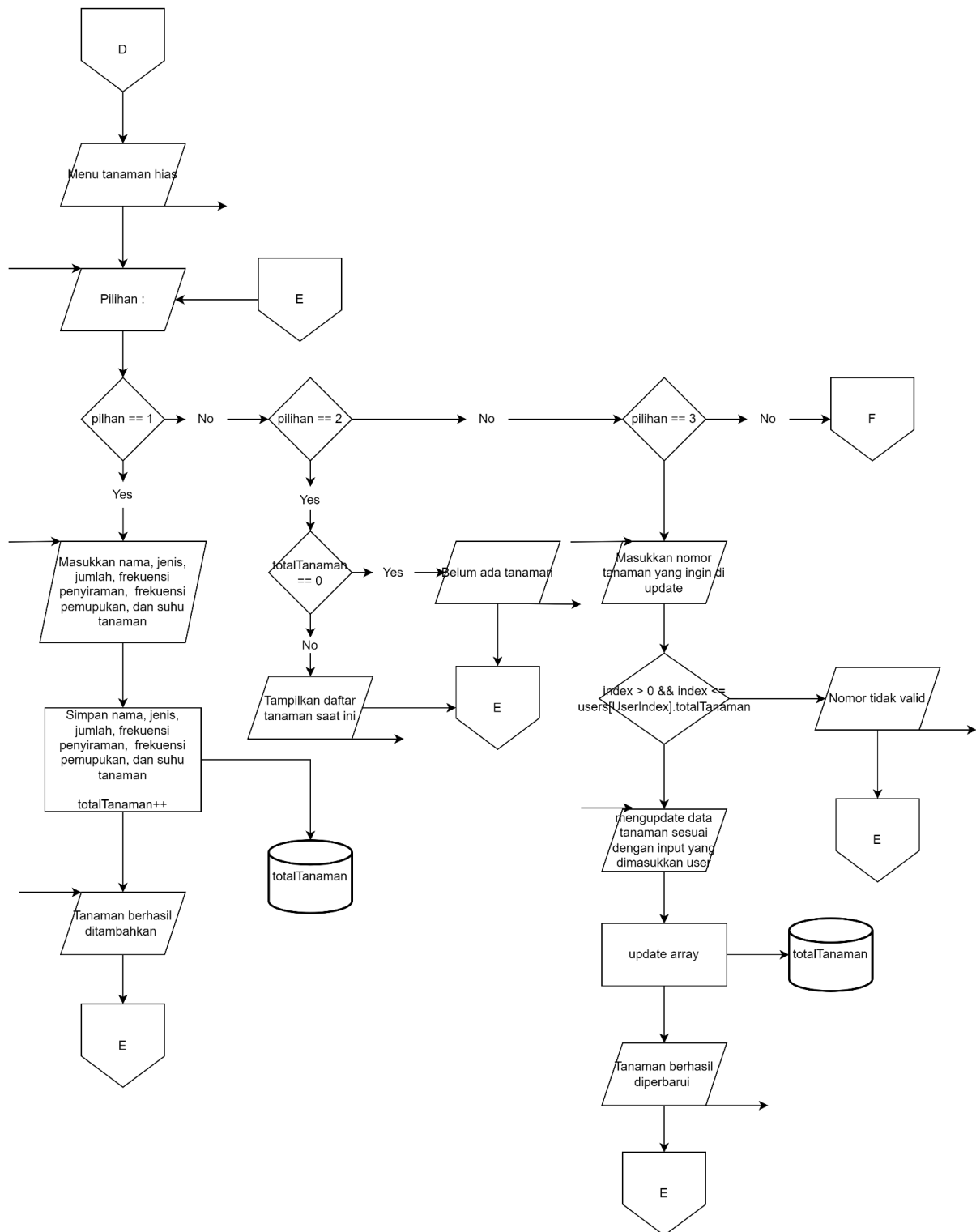
Disusun oleh:
Dimas Firjatullah Islamay (2409106057)
Kelas (B1 '24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

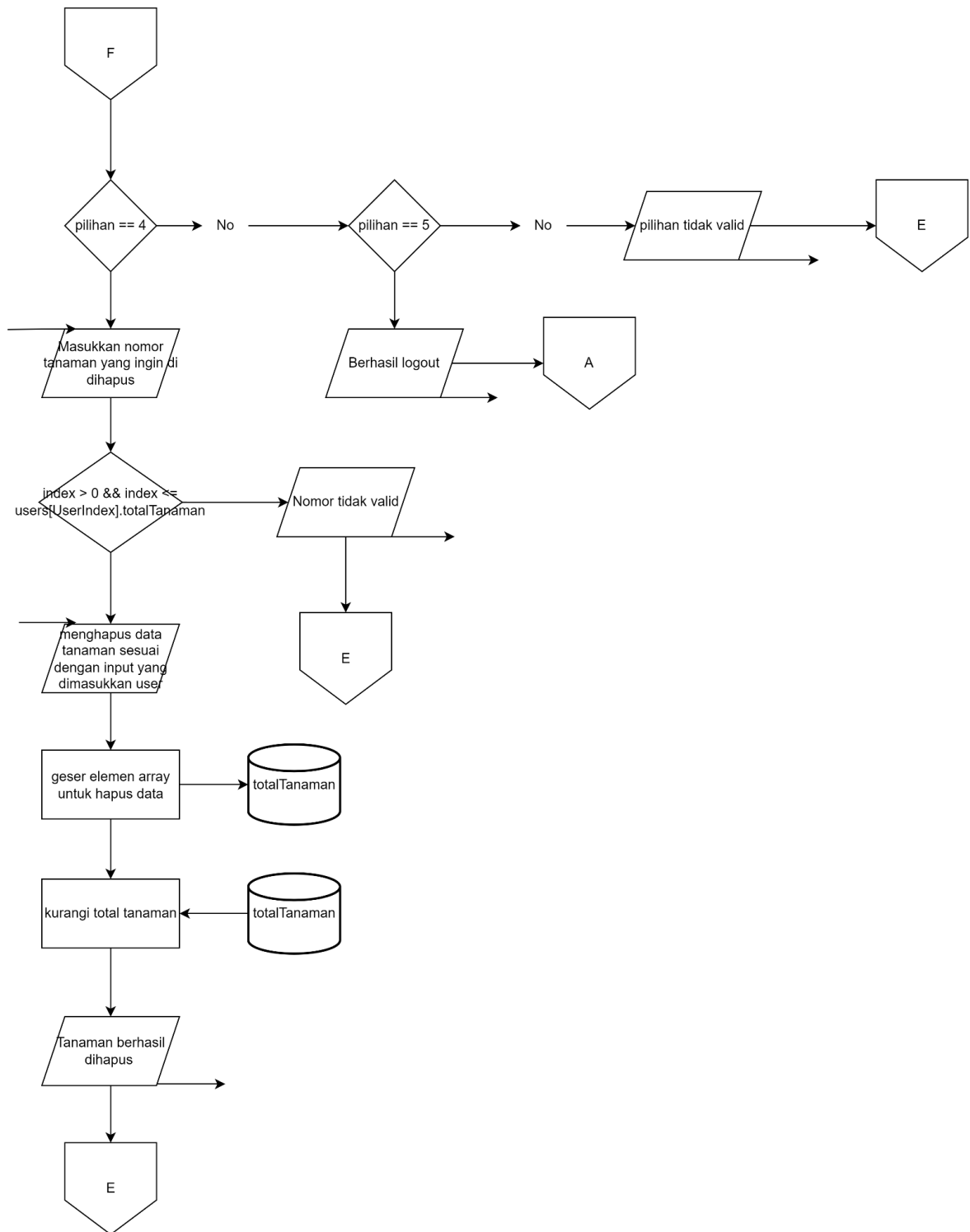
1. Flowchart



Gambar 1.1 Fitur Daftar, Login Akun dan Keluar Program



Gambar 1. 2 Fitur Tambah, Tampilkan, Dan Update Tanaman



Gambar 1.3 Fitur Hapus Tanaman dan Logout

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 fitur daftar akun dan login akun yang dimana user harus registrasi akun dengan nama dan NIM sebagai password lalu mencocokkan username dan password yang sudah diatur, fitur CRUD yang dilengkapi index, terdapat array of struct atau nested struct 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.

3. Source Code

A. Menu Awal

Fitur yang terdapat di menu awal ini merupakan fitur daftar dan login akun dimana user terlebih dahulu mendaftarkan akunnya kemudian dapat melakukan login dengan memasukkan nama dan nim yang sesuai dengan yang didaftarkan

Source Code:

```
while (true)
{
    int Pilihan;
    cout << "==== MENU =====" << endl;
    cout << "1. Daftar Akun" << endl;
    cout << "2. Login" << endl;
    cout << "3. Keluar" << endl;
    cout << "Pilih: ";
    cin >> Pilihan;

    if (Pilihan == 1)
    {
        cout << "Masukkan Nama: ";
        cin >> users[TotalAkun].nama;
        cout << "Masukkan NIM: ";
        cin >> users[TotalAkun].nim;
        TotalAkun++;
        cout << "Akun Berhasil Didaftarkan" << endl;
    }
    else if (Pilihan == 2)
    {
        Coba = 0;
        while (Coba < 3)
        {
            cout << "=== Login ===" << endl;
            cout << "Masukkan Nama: ";
            cin >> Nama;
            cout << "Masukkan NIM: ";
            cin >> NIM;

            UserIndex = -1;
            for (int i = 0; i < TotalAkun; i++)
            {
                if (users[i].nama == Nama && users[i].nim == NIM)
                {
                    UserIndex = i;
                    break;
                }
            }
        }
    }
}
```

```

        if (UserIndex != -1)
        {
            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 3.1

B. Menu Utama

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

Source Code:

```

while (true)
{
    int pilihan;
    cout << "=== MENU TANAMAN HIAS ===" << endl;
    cout << "1. Tambah Tanaman" << endl;
    cout << "2. Lihat Tanaman" << endl;
    cout << "3. Update Tanaman" << endl;
    cout << "4. Hapus Tanaman" << endl;
    cout << "5. Logout" << endl;
    cout << "Pilih: ";
    cin >> pilihan;
}

```

Gambar 3.2

C. 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)

```

```

    {
        int index = users[UserIndex].totalTanaman;
        cout << "Masukkan Nama Tanaman: ";
        cin >> users[UserIndex].tanaman[index].nama;
        cout << "Masukkan Jenis Tanaman: ";
        cin >> users[UserIndex].tanaman[index].jenis;
        cout << "Masukkan Jumlah Tanaman: ";
        cin >> users[UserIndex].tanaman[index].jumlah;
        cout << "Masukkan Frekuensi Penyiraman (Berapa kali per
minggu): ";

        cin >> users[UserIndex].tanaman[index].siram;
        cout << "Masukkan Frekuensi Pemupukan (Berapa kali per
bulan): ";

        cin >> users[UserIndex].tanaman[index].pupuk;
        cout << "Masukkan Suhu Tanaman: ";
        cin >> users[UserIndex].tanaman[index].suhu;
        users[UserIndex].totalTanaman++;
        cout << "Tanaman Berhasil Ditambahkan" << endl;
    }

```

Gambar 3.3

D. Tampilkan Tanaman

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

Source Code:

```

else if (pilihan == 2)
{
    if (users[UserIndex].totalTanaman == 0)
    {
        cout << "Belum Ada Tanaman" << endl;
    }
    else
    {
        cout << "=== DAFTAR TANAMAN ===" << endl;
        for (int i = 0; i < users[UserIndex].totalTanaman; i++)
        {
            cout << i + 1 << ". " <<
users[UserIndex].tanaman[i].nama << " | "
            << users[UserIndex].tanaman[i].jenis << " | "
            << users[UserIndex].tanaman[i].jumlah << "
Tanaman | "
            << users[UserIndex].tanaman[i].siram << "Kali
Siram Per Minggu | "
            << users[UserIndex].tanaman[i].pupuk << "Kali
Pupuk Per Bulan | "

```



```

                                << users[UserIndex].tanaman[i].suhu << "Derajat
Celsius" << endl;
                                }
                            }
    }

```

Gambar 3.4

E. 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 <= users[UserIndex].totalTanaman)
    {
        index--;
        cout << "Masukkan Nama Tanaman Baru: ";
        cin >> users[UserIndex].tanaman[index].nama;
        cout << "Masukkan Jenis Tanaman Baru: ";
        cin >> users[UserIndex].tanaman[index].jenis;
        cout << "Masukkan Jumlah Tanaman Baru: ";
        cin >> users[UserIndex].tanaman[index].jumlah;
        cout << "Masukkan Frekuensi Penyiraman Baru: ";
        cin >> users[UserIndex].tanaman[index].siram;
        cout << "Masukkan Frekuensi Pemupukan Baru: ";
        cin >> users[UserIndex].tanaman[index].pupuk;
        cout << "Masukkan Suhu Tanaman Baru: ";
        cin >> users[UserIndex].tanaman[index].suhu;
        cout << "Tanaman Berhasil Diperbarui" << endl;
    }
    else
    {
        cout << "Nomor Tidak Valid" << endl;
    }
}

```

Gambar 3.5

F. 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 <= users[UserIndex].totalTanaman)
    {
        for (int i = index - 1; i <
users[UserIndex].totalTanaman - 1; i++)
        {
            users[UserIndex].tanaman[i] =
users[UserIndex].tanaman[i + 1];
        }
        users[UserIndex].totalTanaman--;
        cout << "Tanaman Berhasil Dihapus" << endl;
    }
    else
    {
        cout << "Nomor Tidak Valid" << endl;
    }
}
```

Gambar 3.6

G. Menu Logout

Menu ini digunakan untuk logout atau kembali ke menu awal.

Source Code:

```
else if (pilihan == 5)
{
    cout << "Logout Berhasil" << endl;
    break;
}
else
{
    cout << "Pilihan Tidak Valid" << endl;
}
```

Gambar 3.7

H. Menu Keluar

Menu ini digunakan untuk menyelesaikan program.

Source Code:

```
else if (Pilihan == 3)
{
    cout << "Terima kasih telah menggunakan program ini" << endl;
    break;
}
else
{
    cout << "Pilihan Tidak Valid" << endl;
}
}

return 0;
}
```

Gambar 3.8

4. Hasil Output

4.1 Hasil Output

```
==== MENU ====  
1. Daftar Akun  
2. Login  
3. Keluar  
Pilih: █
```

Gambar 4.1 Menu Awal Daftar

```
Pilih: 1  
Masukkan Nama: Dimas  
Masukkan NIM: 2409106057  
Akun Berhasil Didaftarkan  
==== MENU ====  
1. Daftar Akun  
2. Login  
3. Keluar  
Pilih: █
```

Gambar 4.2 Daftar Akun

```
=== Login ===  
Masukkan Nama: DimasKroco  
Masukkan NIM: 2409106057  
Nama Atau NIM Salah  
=== Login ===  
Masukkan Nama: Dimas  
Masukkan NIM: 13123131313  
Nama Atau NIM Salah  
=== Login ===  
Masukkan Nama: Cihuy  
Masukkan NIM: 2000000  
Nama Atau NIM Salah  
Anda Salah Memasukkan Nama Atau NIM 3 Kali  
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-3> █
```

Gambar 4.3 Login Akun Gagal

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

Gambar 4.4 Login Berhasil

```
=== MENU TANAMAN HIAS ===  
1. Tambah Tanaman  
2. Lihat Tanaman  
3. Update Tanaman  
4. Hapus Tanaman  
5. Logout  
Pilih: █
```

Gambar 4.5 Menu Utama Tanaman Hias

```
Pilih: 1
Masukkan Nama Tanaman: Mawar
Masukkan Jenis Tanaman: Bunga
Masukkan Jumlah Tanaman: 20
Masukkan Frekuensi Penyiraman (Berapa kali per minggu): 3
Masukkan Frekuensi Pemupukan (Berapa kali per bulan): 2
Masukkan Suhu Tanaman: 25
Tanaman Berhasil Ditambahkan
```

Gambar 4.6 Tambah Tanaman

```
Pilih: 2
=== DAFTAR TANAMAN ===
1. Mawar | Bunga | 20 Tanaman | 3Kali Siram Per Minggu | 2Kali Pupuk Per Bulan | 25Derajat Celsius
```

Gambar 4.7 Tampilkan Tanaman

```
Pilih: 3
Masukkan Nomor Tanaman Yang Ingin Diupdate: 1
Masukkan Nama Tanaman Baru: Melati
Masukkan Jenis Tanaman Baru: Bunga
Masukkan Jumlah Tanaman Baru: 16
Masukkan Frekuensi Penyiraman Baru: 3
Masukkan Frekuensi Pemupukan Baru: 1
Masukkan Suhu Tanaman Baru: 24
Tanaman Berhasil Diperbarui
```

Gambar 4.8 Update Tanaman

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

Gambar 4.9 Hapus Tanaman

```
Pilih: 5
Logout Berhasil
==== MENU ====
1. Daftar Akun
2. Login
3. Keluar
Pilih: █
```

Gambar 4.10 logout

```
Pilih: 3
Terima kasih telah menggunakan program ini
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-3> █
```

4.11 Keluar Program

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

Gambar 4.12 Pilihan Tidak Valid

5. Langkah-langkah GIT

```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-3> 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-3> git commit -m "Finish Posttest 3"
[main b06ec77] Finish Posttest 3
1 file changed, 0 insertions(+), 0 deletions(-)
```

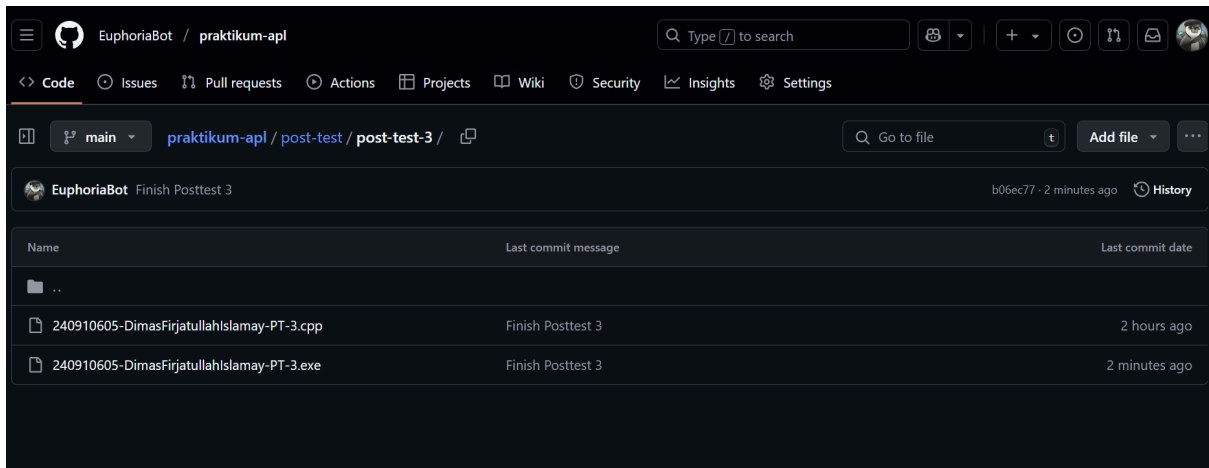
Gambar 5.2 Git commit

Git commit digunakan untuk menyimpan hasil file yang telah ditambahkan

```
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-3> 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), 8.60 KiB | 677.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/EuphoriaBot/praktikum-apl.git
 9c583da..b06ec77 main -> main
PS C:\Users\ASUS\OneDrive\Desktop\praktikum-apl\post-test\post-test-3> █
```

Gambar 5.3 Git push

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



Gambar 5.4 Github

Berikut contoh gambar jika langkah-langkah Git berhasil dilaksanakan