

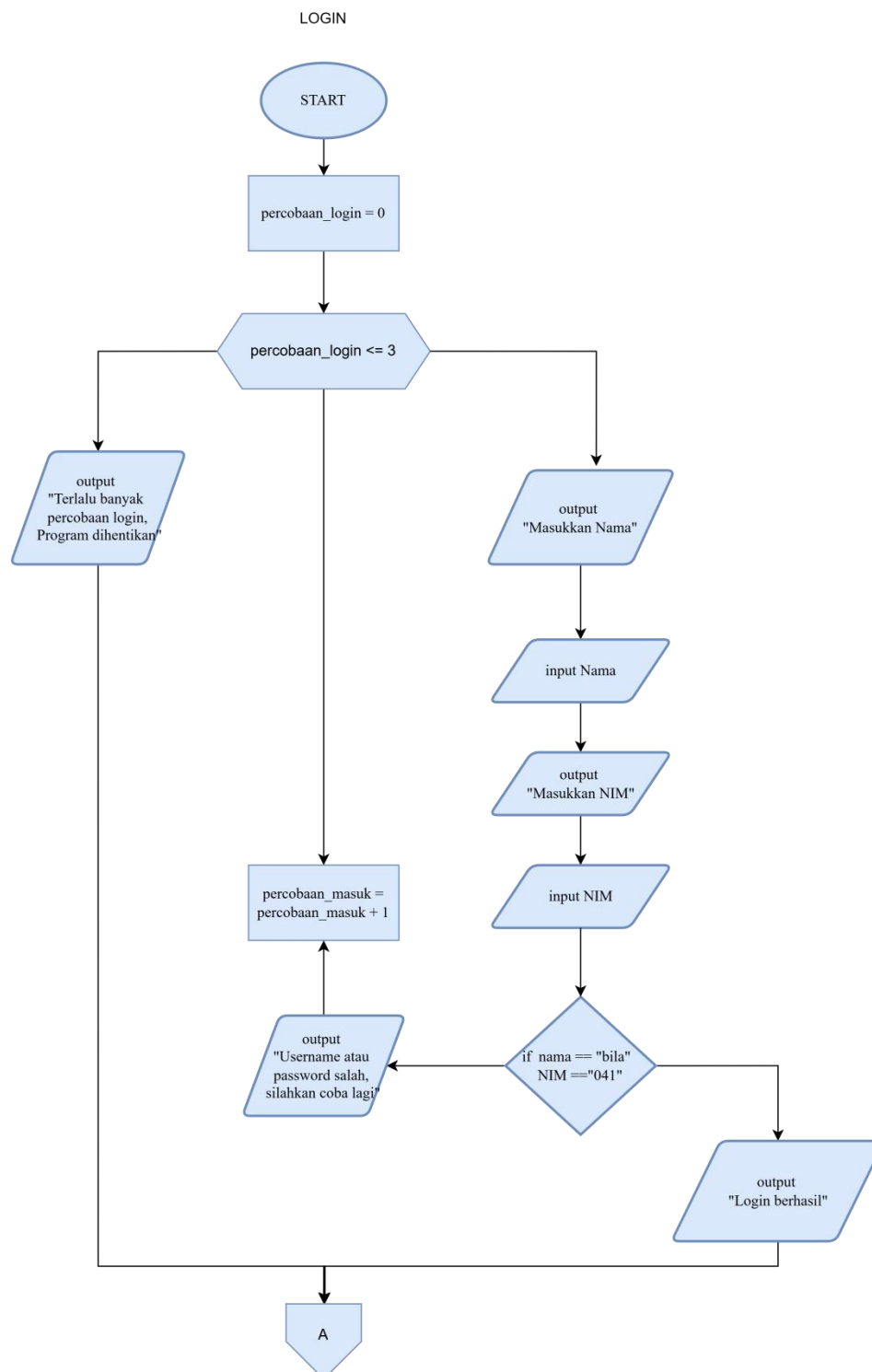
**LAPORAN PRAKTIKUM**  
**POSTTEST 2**  
**ALGORITMA PEMROGRAMAN LANJUT**



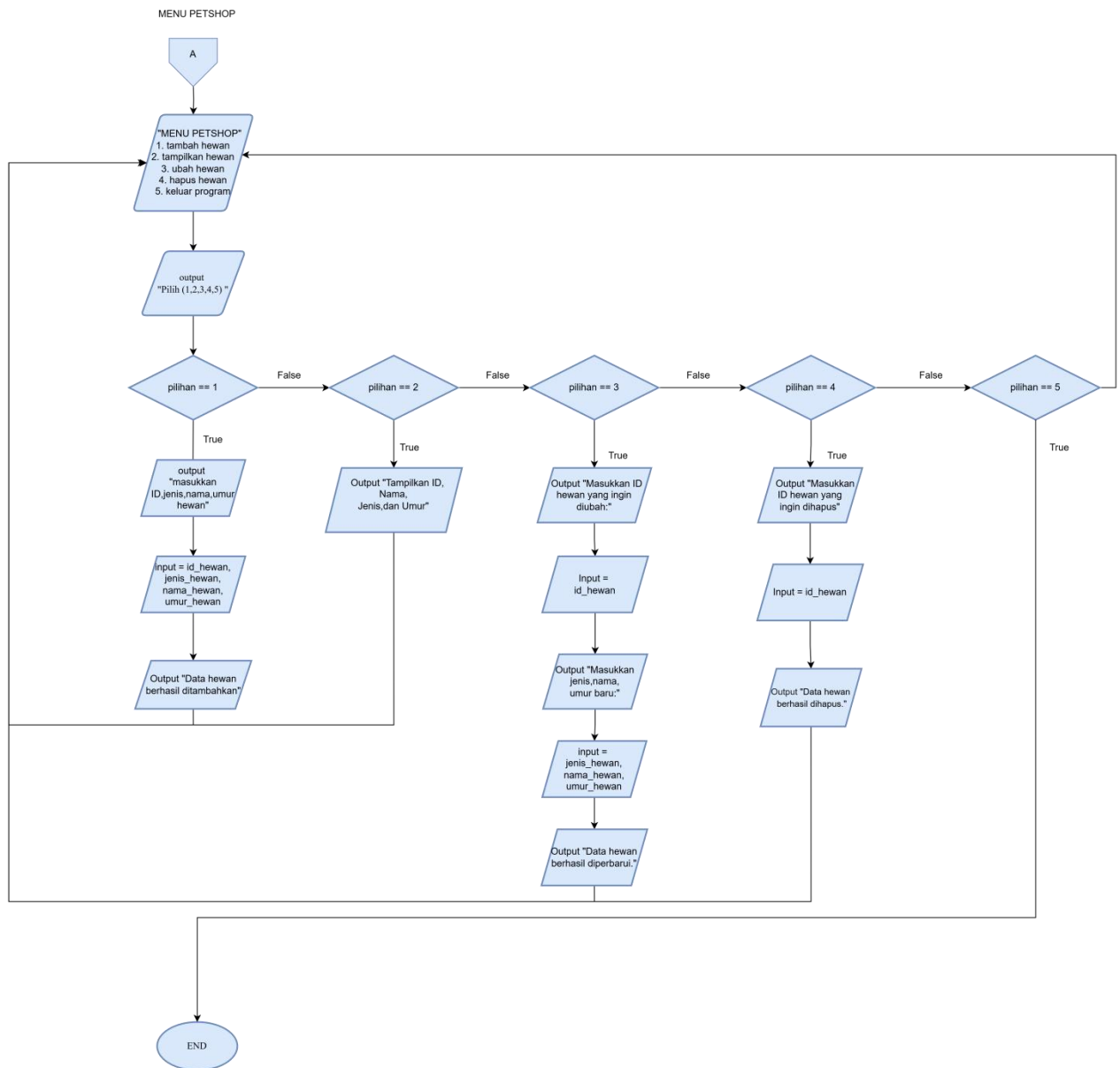
**Disusun oleh:**  
**Nabila Putri Karni (2409106041)**  
**Kelas (A2 '24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

## 1. Flowchart



Gambar 1.1 Flowchart Login



Gambar 1.2 Flowchart Menu Petshop

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program Manajemen Petshop ini bertujuan untuk membantu dalam mengelola data hewan secara efisien. Dengan fitur CRUD. Program ini dapat menambahkan, melihat, memperbarui, dan menghapus data hewan dengan mudah. Lalu manfaat dari program ini yaitu meningkatkan efisiensi dalam pencatatan data hewan dan memudahkan pencarian dan pembaruan data hewan.

### 2.2 Penjelasan Alur & Algoritma

#### ➤ Proses Login

Ketika program dijalankan, pengguna diwajibkan untuk memasukkan Nama dan NIM sebagai ketentuan untuk login. Program akan memeriksa apakah data yang dimasukkan sesuai dengan data login yang telah ditentukan sebelumnya, yaitu "bila" sebagai nama dan "041" sebagai NIM. Jika pengguna salah memasukkan data sebanyak tiga kali berturut-turut, program akan langsung dihentikan dan pengguna tidak dapat mengakses fitur-fitur yang tersedia. Namun, jika pengguna berhasil memasukkan nama dan NIM yang benar, maka akan diberikan akses ke menu utama program yang berisi berbagai opsi untuk mengelola data hewan dalam petshop.

#### ➤ Menu Utama

Setelah login berhasil, program akan menampilkan menu utama yang berisi beberapa pilihan yang dapat dipilih oleh pengguna. Menu ini terdiri dari enam opsi utama, yaitu:

1. **Tambah Hewan** – Digunakan untuk menambahkan data hewan ke dalam sistem.
2. **Tampilkan Hewan** – Menampilkan daftar hewan yang telah disimpan dalam sistem.
3. **Ubah Hewan** – Memungkinkan pengguna untuk mengubah data hewan yang sudah ada.
4. **Hapus Hewan** – Menghapus data hewan berdasarkan ID tertentu.
5. **Keluar Program** – Menutup program sepenuhnya dan menghentikan semua proses.

#### ➤ **Menu Tambah Hewan**

Jika pengguna memilih opsi Tambah Hewan, program akan meminta pengguna untuk memasukkan ID Hewan, Nama Hewan, Jenis Hewan, dan Umur Hewan. Data yang dimasukkan oleh pengguna akan disimpan ke dalam array yang sudah disediakan sebelumnya. Jika kapasitas array belum penuh, data akan berhasil ditambahkan, dan pengguna akan mendapatkan notifikasi bahwa data hewan telah tersimpan. Namun, jika kapasitas array sudah mencapai batas maksimal, program akan menampilkan pesan yang menyatakan bahwa tidak bisa menambahkan lebih banyak hewan.

#### ➤ **Menu Tampilkan Hewan**

Jika pengguna memilih opsi *Tampilkan Hewan*, maka program akan menampilkan seluruh daftar hewan yang telah disimpan dalam sistem. Jika belum ada data hewan yang dimasukkan, program akan menampilkan pesan bahwa belum ada data hewan yang tersedia. Namun, jika terdapat data hewan dalam sistem, program akan mencetak daftar hewan tersebut dalam format tabel yang terdiri dari kolom ID, Nama, Jenis, dan Umur.

#### ➤ **Menu Mengubah Data Hewan**

Jika pengguna ingin memperbarui informasi hewan yang telah tersimpan, mereka dapat memilih opsi Ubah Hewan. Program akan meminta pengguna untuk memasukkan ID Hewan yang ingin diperbarui. Jika ID tersebut ditemukan dalam sistem, maka program akan meminta pengguna untuk memasukkan Nama baru, Jenis baru, dan Umur baru untuk hewan tersebut. Setelah data diubah, program akan memberikan notifikasi bahwa informasi hewan telah diperbarui. Namun, jika ID yang dimasukkan tidak ada dalam sistem, maka program akan menampilkan pesan bahwa hewan tersebut tidak ditemukan dalam database.

### ➤ **Menu Menghapus Data Hewan**

Ketika pengguna ingin menghapus data hewan yang sudah tidak diperlukan, mereka dapat memilih opsi Hapus Hewan. Program akan meminta pengguna untuk memasukkan ID Hewan yang ingin dihapus. Jika ID yang dimasukkan ditemukan dalam sistem, maka data hewan tersebut akan dihapus dan elemen-elemen array akan digeser untuk menghindari adanya ruang kosong dalam daftar. Setelah penghapusan berhasil, program akan menampilkan pesan bahwa data hewan telah berhasil dihapus. Jika ID yang dimasukkan tidak ditemukan dalam sistem, program akan memberi tahu pengguna bahwa hewan dengan ID tersebut tidak ada dalam database.

### ➤ **Keluar Program**

Jika pengguna memilih opsi Keluar Program, maka program akan langsung berhenti sepenuhnya dan semua proses akan dihentikan.

## 2. Source Code

### A. Mendeklarasikan Variabel Global

```
const int limit_hewan = 100;
string id_hewan[limit_hewan];
string jenis_hewan[limit_hewan];
string nama_hewan[limit_hewan];
int umur_hewan[limit_hewan];
int jumlah_hewan = 0;
```

Pada bagian ini, terdapat deklarasi global yang akan digunakan dalam program.

### B. Login

```
string Nama, NIM;
bool login = false;
int percobaan_login = 0;

while (percobaan_login < 3 && !login)
{
    cout << "Masukkan Nama : ";
    cin >> Nama;
    cout << "Masukkan NIM : ";
    cin >> NIM;

    if (Nama == "bila" && NIM == "041")
    {
        login = true;
        cout << "Login berhasil, Welcome " << Nama << "!\n";
    }
    else
    {
        percobaan_login++;
        cout << "Nama atau NIM salah, Silahkan coba lagi!\n";
    }
}
if (!login)
{
    cout << "Terlalu banyak percobaan login, program dihentikan !\n";
    return 0;
}
```

Pada bagian ini, pengguna harus memasukkan Nama dan NIM untuk masuk ke sistem. Jika pengguna salah memasukkan Nama atau NIM sebanyak tiga kali, program akan dihentikan secara otomatis. Jika login berhasil, pengguna bisa mengakses menu utama.

### C. Menu Program Petshop

```
while (true)
{
    cout << "\n=====\\n";
    cout << "|      MENU PET SHOP      |\\n";
    cout << "=====\\n";
    cout << "|  1. TAMBAH HEWAN      |\\n";
    cout << "|  2. TAMPILKAN HEWAN  |\\n";
    cout << "|  3. UBAH HEWAN       |\\n";
    cout << "|  4. HAPUS HEWAN      |\\n";
    cout << "|  5. KELUAR PROGRAM   |\\n";
    cout << "=====\\n";
    cout << "Pilih (1/2/3/4/5): ";
    int pilihan;
    cin >> pilihan;
```

Di dalam perulangan while(true), program akan terus menampilkan menu utama sampai pengguna memilih Keluar Program (pilihan 5). Menu ini berisi opsi untuk menambah, melihat, mengubah, menghapus hewan, atau keluar dari program.

### D. Fitur Menambahkan Data Hewan

```
if (pilihan == 1)
{
    if (jumlah_hewan < limit_hewan)
    {
        cout << "Masukkan ID Hewan: ";
        cin >> id_hewan[jumlah_hewan];
        cout << "Masukkan jenis hewan: ";
        cin >> jenis_hewan[jumlah_hewan];
        cout << "Masukkan Nama Hewan: ";
        cin >> nama_hewan[jumlah_hewan];
        cout << "Masukkan Umur Hewan: ";
        cin >> umur_hewan[jumlah_hewan];
        jumlah_hewan++;
        cout << "Data hewan telah berhasil ditambahkan.\\n ";
    }
    else
    {
        cout << "Maaf, tidak dapat menambahkan hewan lagi\\n";
    }
}
```

Jika jumlah hewan masih di bawah batas maksimal (*limit\_hewan*), program akan meminta pengguna untuk memasukkan ID, Jenis, Nama, dan Umur Hewan, lalu menyimpannya ke dalam array serta menambah jumlah hewan satu, namun jika sudah mencapai batas maksimal, program akan menampilkan pesan error.



## E. Fitur Menampilkan Data Hewan

```
else if (pilihan == 2)
{
    if (jumlah_hewan == 0)
    {
        cout << "Tidak ada data hewan.\n";
    }
    else
    {
        cout << "\n===== DATA HEWAN =====\n";
        cout << "ID\tJenis \tNama\tUmur\n";
        cout << "-----\n";
        for (int i = 0; i < jumlah_hewan; i++)
        {
            cout << id_hewan[i] << "\t" << jenis_hewan[i] << "\t"
<< nama_hewan[i] << "\t" << umur_hewan[i] << "tahun\n";
        }
        cout << "-----\n";
    }
}
```

Jika terdapat data hewan maka program akan mencetak daftar hewan dalam bentuk tabel, tetapi jika tidak ada data hewan maka program akan menampilkan pesan “Tidak ada data hewan”.

## F. Fitur Mengubah Data Hewan

```
else if (pilihan == 3)
{
    string id;
    cout << "Masukkan ID hewan yang ingin diubah: ";
    cin >> id;
    bool terverifikasi = false;
    for (int i = 0; i < jumlah_hewan; i++)
    {
        if (id_hewan[i] == id)
        {
            cout << "Masukkan jenis baru: ";
            cin >> jenis_hewan[i];
            cout << "Masukkan nama baru: ";
            cin >> nama_hewan[i];
            cout << "Masukkan umur baru hewan: ";
            cin >> umur_hewan[i];
            terverifikasi = true;
            cout << "Data hewan berhasil diperbarui.\n";
            break;
        }
    }
    if (!terverifikasi)
        cout << "Hewan tidak terverifikasi!\n";
}
```

Pengguna memasukkan **ID hewan** yang ingin diubah, jika ID ditemukan, pengguna dapat memasukkan data baru, namun jika ID tidak ditemukan, program akan menampilkan pesan error.

## G. Fitur Menghapus Data Hewan

```
else if (pilihan == 4)
{
    string id;
    cout << "Masukkan ID hewan yang ingin dihapus: ";
    cin >> id;
    bool terverifikasi = false;
    for (int i = 0; i < jumlah_hewan; i++)
    {
        if (id_hewan[i] == id)
        {
            for (int j = 1; j < jumlah_hewan - 1; j++)
            {
                jenis_hewan[j] = jenis_hewan[j + 1];
                id_hewan[j] = id_hewan[j + 1];
                nama_hewan[j] = nama_hewan[j + 1];
                umur_hewan[j] = umur_hewan[j + 1];
            }
            jumlah_hewan--;
            terverifikasi = true;
            cout << "Data hewan berhasil dihapus.\n";
            break;
        }
    }
    if (!terverifikasi)
        cout << "Hewan tidak terverifikasi!\n";
}
```

Jika ID ditemukan, program akan menghapus hewan dari array dan menggeser elemen-elemen berikutnya untuk mengisi celah yang kosong, namun jika ID tidak ditemukan, program akan menampilkan pesan error.

## H. Keluar Dari Program

```
else if (pilihan == 5)
{
    cout << "Terima kasih telah menggunakan program petshop ini.\n";
    return 0;
}
else
{
    cout << "Pilihan tidak valid.\n";
}
return 0;
}
```

Pengguna akan langsung keluar dan program akan berhenti jika pengguna memilih opsi ini. Jika pengguna memasukkan angka lebih dari 1-5 maka program akan menampilkan “Pilihan tidak valid” dan akan kembali ke menu utama.

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

#### Skenario 1

- Login dengan nama dan NIM yang benar
- Login berhasil
- Masuk ke menu petshop
- Memilih 1
- Masukkan Id hewan,jenis hewan,nama hewan,umur hewan
- Output : Data hewan telah berhasil ditambahkan
- Kembali ke menu petshop.
- Pilih 2
- Menampilkan tabel data hewan
- Kembali ke menu petshop
- Pilih 3
- Masukkan id hewan yang ingin diubah lalu masukkan jenis baru,nama baru,umur baru.
- Output : Data hewan berhasil diperbarui
- Pilih 2
- Menampilkan data hewan yang telah diubah tadi
- Kembali ke menu petshop
- Pilih 4
- Masukkan id hewan yang ingin dihapus
- Output : data hewan berhasil dhapus
- Kembali ke menu petshop
- Pilih 2
- Output : tidak ada data hewan
- Kembali ke menu petshop
- Pilih 5
- Keluar dari program.

#### Skenario ke 2

- Login dengan nama dan NIM yang salah sebanyak 3 kali
- Program dihentikan.

## 4.2 Hasil Output

```
Masukkan Nama : bila
Masukkan NIM : 041
Login berhasil, welcome bila!

=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 1
Masukkan ID Hewan: 01
Masukkan jenis hewan: kucing
Masukkan Nama Hewan: namja
Masukkan Umur Hewan: 1
Data hewan telah berhasil ditambahkan.
```

Gambar 4.1 Skenario 1 (berhasil)

```
=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 2

===== DATA HEWAN =====
ID      Jenis   Nama    Umur
-----
01      kucing  namja   1tahun
-----
```

Gambar 4.2 Skenario 1 (berhasil)

```

=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 3
Masukkan ID hewan yang ingin diubah: 01
Masukkan jenis baru: anjing
Masukkan nama baru: Park
Masukkan umur baru hewan: 2
Data hewan berhasil diperbarui.

=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 2

===== DATA HEWAN =====
ID      Jenis  Nama  Umur
-----
01      anjing  Park  2tahun
-----

```

Gambar 4.3 Skenario 1 (berhasil)

```
=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 4
Masukkan ID hewan yang ingin dihapus: 01
Data hewan berhasil dihapus.

=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 2
Tidak ada data hewan.
```

Gambar 4.4 Skenario 1 (berhasil)

```
=====
|   MENU PET SHOP   |
=====
|  1. TAMBAH HEWAN  |
|  2. TAMPILKAN HEWAN |
|  3. UBAH HEWAN    |
|  4. HAPUS HEWAN   |
|  5. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5): 5
Terima kasih telah menggunakan program petshop ini.
```

Gambar 4.5 Skenario 1 (berhasil)

```

Masukkan Nama : bil
Masukkan NIM : 041
Nama atau NIM salah, Silahkan coba lagi!
Masukkan Nama : lila
Masukkan NIM : 21
Nama atau NIM salah, Silahkan coba lagi!
Masukkan Nama : ie
Masukkan NIM : 32
Nama atau NIM salah, Silahkan coba lagi!
Terlalu banyak percobaan login, program dihentikan !

```

Gambar 4.6 Skenario 2 (gagal)

## 5. Langkah langkah Git

```

MSI GF63@bilaa MINGW64 ~/Documents/Praktikum apl/praktikum-apl (main)
$ git add .

MSI GF63@bilaa MINGW64 ~/Documents/Praktikum apl/praktikum-apl (main)
$ git commit -m "Finish post test 2"
[main 2abed38] Finish post test 2
3 files changed, 155 insertions(+)
create mode 100644 post-test/post-test-2/2409106041-NabilaPutriKarni-PT-2.cpp
create mode 100644 post-test/post-test-2/2409106041-NabilaPutriKarni-PT-2.exe
create mode 100644 post-test/post-test-2/2409106041-NabilaPutriKarni-PT-2.pdf

MSI GF63@bilaa MINGW64 ~/Documents/Praktikum apl/praktikum-apl (main)
$ git push
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 16 threads
Compressing objects: 100% (7/7), done.
Writing objects: 100% (7/7), 948.91 KiB | 12.32 MiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/lymoonty/praktikum-apl.git
9d574e1..2abed38 main -> main

```

**Git add** berfungsi untuk menambahkan perubahan pada file atau seluruh proyek ke dalam staging area, sehingga perubahan tersebut siap untuk disimpan dalam commit berikutnya.

**Git commit** digunakan untuk menyimpan perubahan yang telah ditambahkan ke staging area secara permanen ke dalam repository lokal dengan menyertakan pesan deskriptif sebagai catatan perubahan.

**Git push** digunakan untuk mengunggah commit dari repository lokal ke repository online, memungkinkan kode sumber dapat diakses dari mana saja atau dibagikan dengan orang lain.