

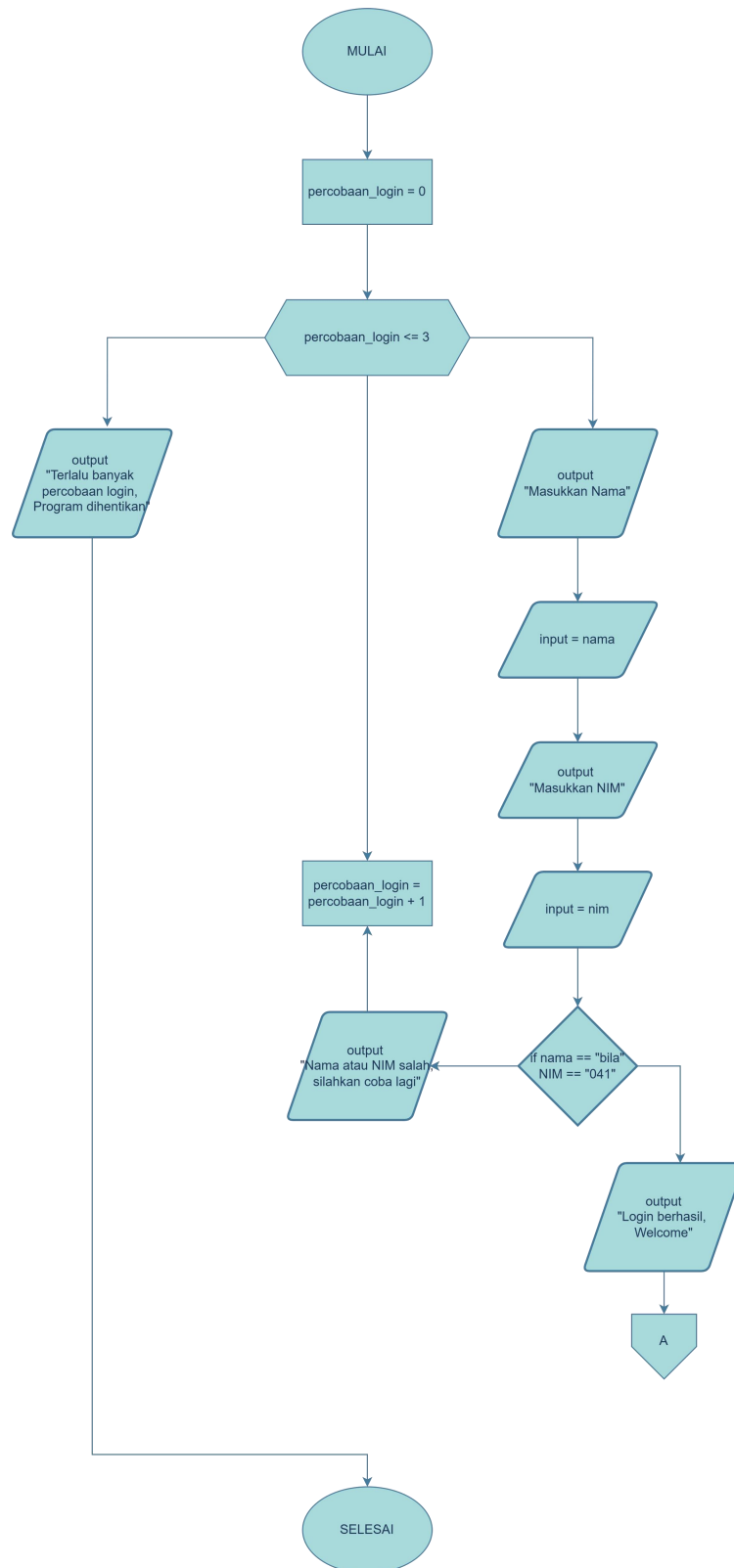
**LAPORAN PRAKTIKUM**  
**POSTTEST 4**  
**ALGORITMA PEMROGRAMAN LANJUT**



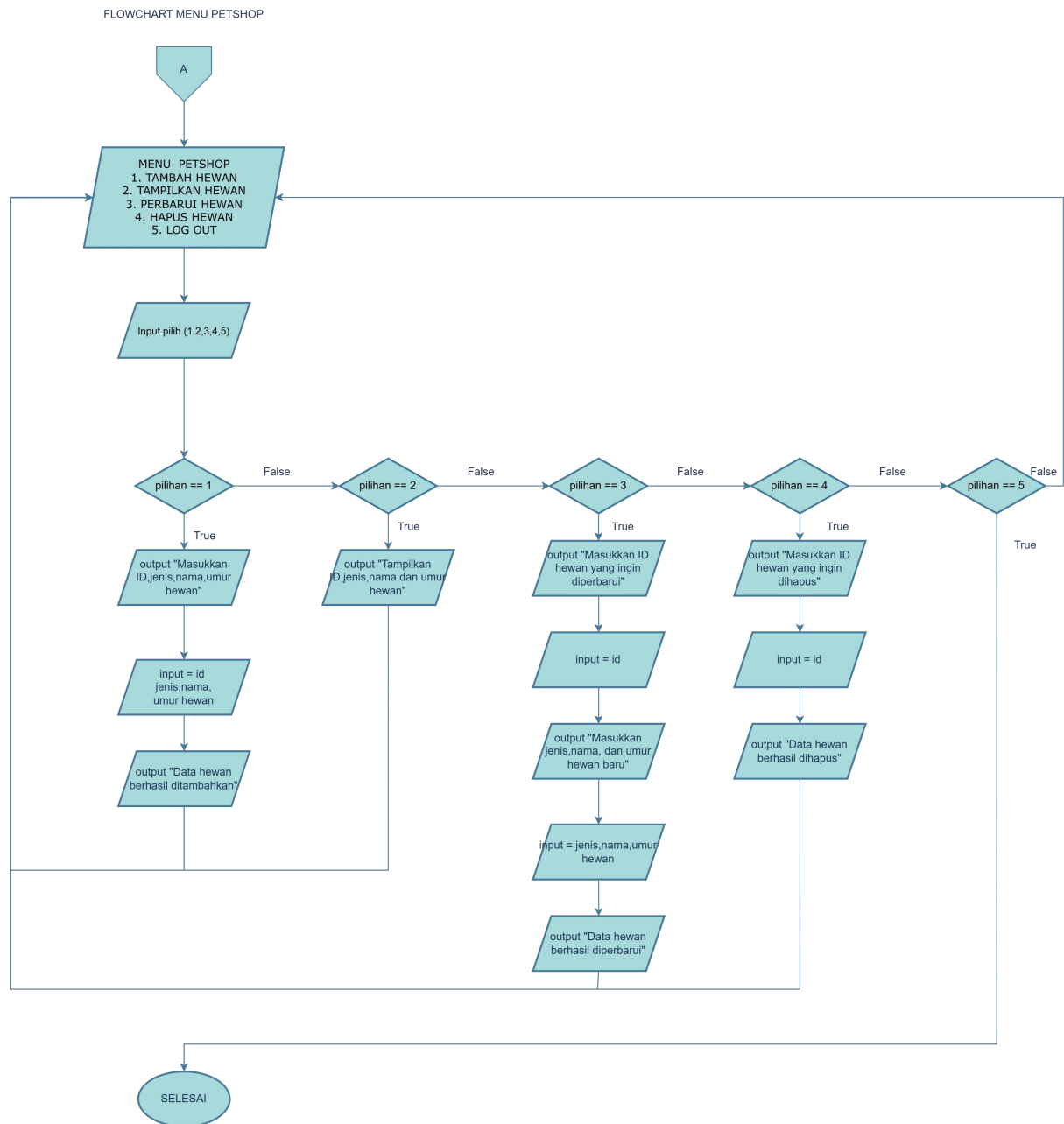
**Disusun oleh:**  
**Nabila Putri karni (2409106041)**  
**Kelas (A2 '24)**

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

## 1. Flowchart



Gambar 1.1 Flowchart Login



Gambar 1.2 Flowchart Menu Petshop

## 2. Analisis Program

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. Sebenarnya program ini sama seperti posttest 3 yang kemarin tetapi disini ada perubahan yaitu masing-masing fitur yang terdapat di menu di pisahkan dari program utama menjadi sub program, fungsi wajib menggunakan parameter dan wajib menggunakan prosedur.

## 3. Source Code

### A. Struck Hewan

```
struct hewan
{
    string id;
    string jenis;
    string nama;
    int umur;
};
```

Struct hewan adalah tipe data buatan yang menyimpan informasi hewan berupa ID unik, jenis (seperti anjing atau kucing), nama, dan umur dalam satuan tahun.

### B. Variabel Global

```
hewan daftar_hewan[limit_hewan];
int jumlah_hewan = 0;
```

daftar\_hewan adalah array yang menyimpan data hingga 100 hewan, sedangkan jumlah\_hewan mencatat jumlah hewan yang telah ditambahkan ke dalam array tersebut.

### C. Tambah hewan

```
void tambah_hewan(hewan daftar[], int &jumlah)
{
    if (jumlah < limit_hewan)
    {
        cout << "Masukkan ID Hewan: ";
        cin >> daftar[jumlah].id;
        cout << "Masukkan jenis hewan: ";
        cin >> daftar[jumlah].jenis;
        cout << "Masukkan Nama Hewan: ";
        cin >> daftar[jumlah].nama;
        cout << "Masukkan Umur Hewan: ";
        cin >> daftar[jumlah].umur;
        jumlah++;
        cout << "Data hewan telah berhasil ditambahkan.\n";
    }
    else
    {
        cout << "Maaf, tidak dapat menambahkan hewan lagi.\n";
    }
}
```

Fungsi ini menambahkan data hewan baru ke array dengan meminta input ID, jenis, nama, dan umur menggunakan cin, lalu menyimpan datanya ke dalam array daftar[] dan menambahkan nilai jumlah++.

### D. Tampilkan Hewan

```
void tampilkan_hewan(const hewan daftar[], int jumlah)
{
    if (jumlah == 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; i++)
        {
            cout << daftar[i].id << "\t"
                << daftar[i].jenis << "\t"
                << daftar[i].nama << "\t"
                << daftar[i].umur << " tahun\n";
        }
        cout << "-----\n";
    }
}
```

Fungsi ini menampilkan semua data hewan dalam daftar, dan jika belum ada data (jumlah == 0), maka akan menampilkan pesan "Tidak ada data hewan".

## E. Ubah Hewan

```
void ubah_hewan(hewan daftar[], int jumlah)
{
    string id;
    cout << "Masukkan ID hewan yang ingin diubah: ";
    cin >> id;
    bool ditemukan = false;
    for (int i = 0; i < jumlah; i++)
    {
        if (daftar[i].id == id)
        {
            cout << "Masukkan jenis baru: ";
            cin >> daftar[i].jenis;
            cout << "Masukkan nama baru: ";
            cin >> daftar[i].nama;
            cout << "Masukkan umur baru hewan: ";
            cin >> daftar[i].umur;
            ditemukan = true;
            cout << "Data hewan berhasil diperbarui.\n";
            break;
        }
    }
    if (!ditemukan)
        cout << "Hewan tidak ditemukan!\n";
}
```

Fungsi ini meminta input ID hewan yang ingin diubah, lalu jika ditemukan akan mengganti data lama (jenis, nama, umur), dan jika tidak ditemukan akan menampilkan pesan "Hewan tidak ditemukan!".

## G. Menghapus Hewan

```
void hapus_hewan(hewan daftar[], int &jumlah)
{
    string id;
    cout << "Masukkan ID hewan yang ingin dihapus: ";
    cin >> id;
    bool ditemukan = false;
    for (int i = 0; i < jumlah; i++)
    {
        if (daftar[i].id == id)
        {
            for (int j = i; j < jumlah - 1; j++)
            {
                daftar[j] = daftar[j + 1];
            }
            jumlah--;
            ditemukan = true;
            cout << "Data hewan berhasil dihapus.\n";
            break;
        }
    }
    if (!ditemukan)
        cout << "Hewan tidak ditemukan\n";
}
```

Fungsi ini meminta ID hewan yang ingin dihapus, lalu jika ditemukan datanya akan digeser ke kiri untuk menutup celah dan jumlah hewan dikurangi satu.

## H. Fungsi login

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

    if (login_program(Nama, NIM))
    {
        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;
}
```

Maksimal 3 kali kesempatan login jika gagal maka akan keluar dari program.

## I. Menu pet shop

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

Tampilan menu petshop.

## J. Menu switch

```
switch (pilihan)
{
    case 1:
        tambah_hewan(daftar_hewan, jumlah_hewan);
        break;
    case 2:
        tampilkan_hewan(daftar_hewan, jumlah_hewan);
        break;
    case 3:
        ubah_hewan(daftar_hewan, jumlah_hewan);
        break;
    case 4:
        hapus_hewan(daftar_hewan, jumlah_hewan);
        break;
    case 5:
        cout << "Terima kasih telah menggunakan program petshop ini.\n";
        return 0;
    default:
        cout << "Pilihan tidak valid.\n";
}
```

Program menggunakan struktur switch untuk mengeksekusi menu yang dipilih pengguna berdasarkan input angka (1–5), yang masing-masing terhubung ke fungsi tambah, tampil, ubah, hapus, atau keluar dari program.



#### 4. 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: 12
Masukkan jenis hewan: kucing
Masukkan Nama Hewan: nami
Masukkan Umur Hewan: 3
Data hewan telah berhasil ditambahkan.

=====
|   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
-----
12      kucing  nami   3 tahun
-----
```

Gambar 4.1 Tampilan Terminal

```
=====
|   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: 12
Masukkan jenis baru: anjing
Masukkan nama baru: bobo
Masukkan umur baru hewan: 1
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
-----
12      anjing bobo  1 tahun
-----
```

Gambar 4.2 Tampilan Terminal

```
=====
|   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: 12
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.

=====
|   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 4.3 Tampilan Terminal

## 5. Langkah-Langkah Git pada VSCode

```
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-4> git add .
PS C:\Users GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-4>
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-4> git commit -m "Finish Posttest 4"
[main 032493b] Finish Posttest 4
 3 files changed, 183 insertions(+)
 create mode 100644 post-test/post-test-4/2409106041-NabilaPutriKarni-PT-4.cpp
 create mode 100644 post-test/post-test-4/2409106041-NabilaPutriKarni-PT-4.exe
 create mode 100644 post-test/post-test-4/2409106041-NabilaPutriKarni-PT-4.pdf
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-4> 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), 851.06 KiB | 23.64 MiB/s, done.
Total 7 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/lymoonty/praktikum-apl.git
 b2600b1..032493b main -> main
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-4>
```

Gambar 5.1

**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.