

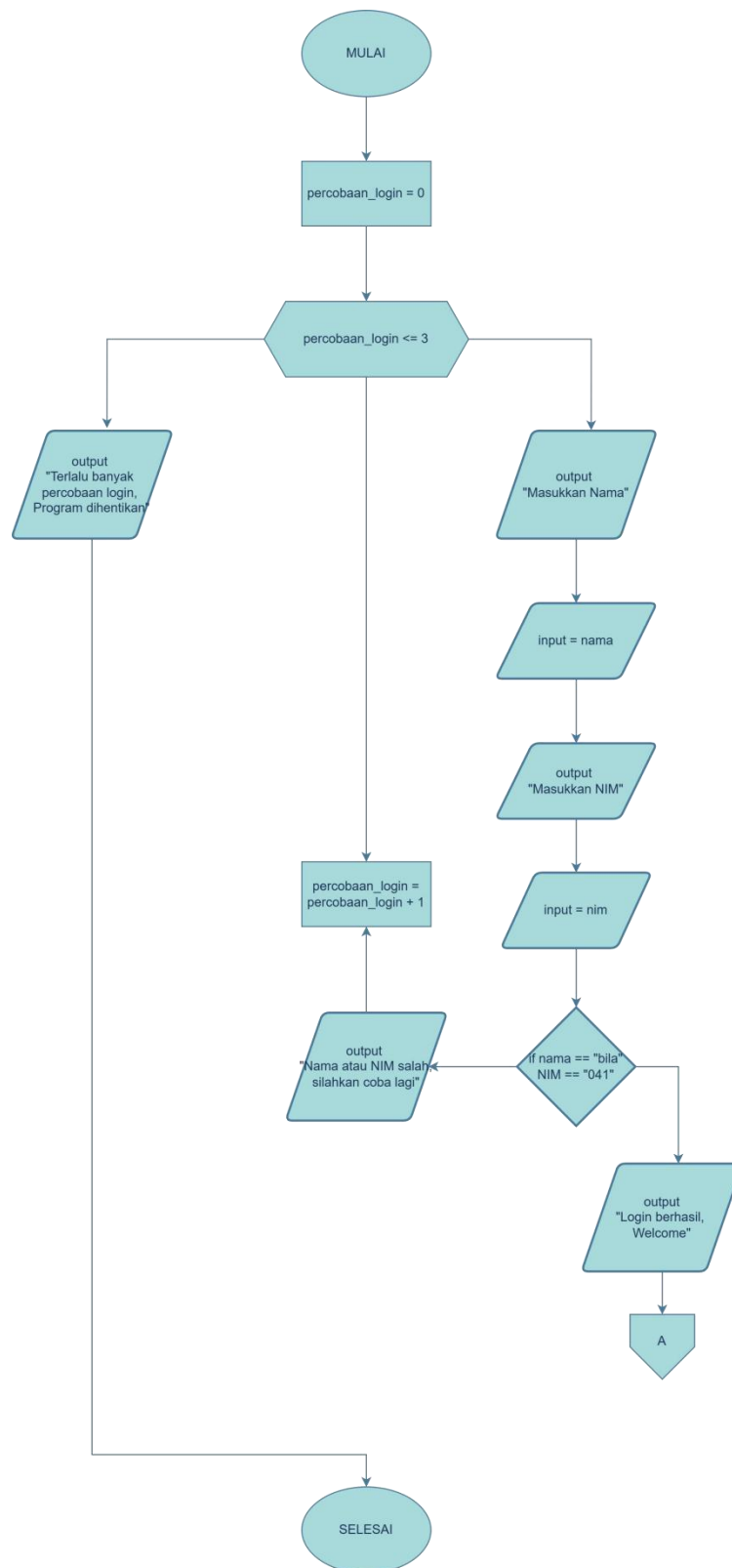
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT



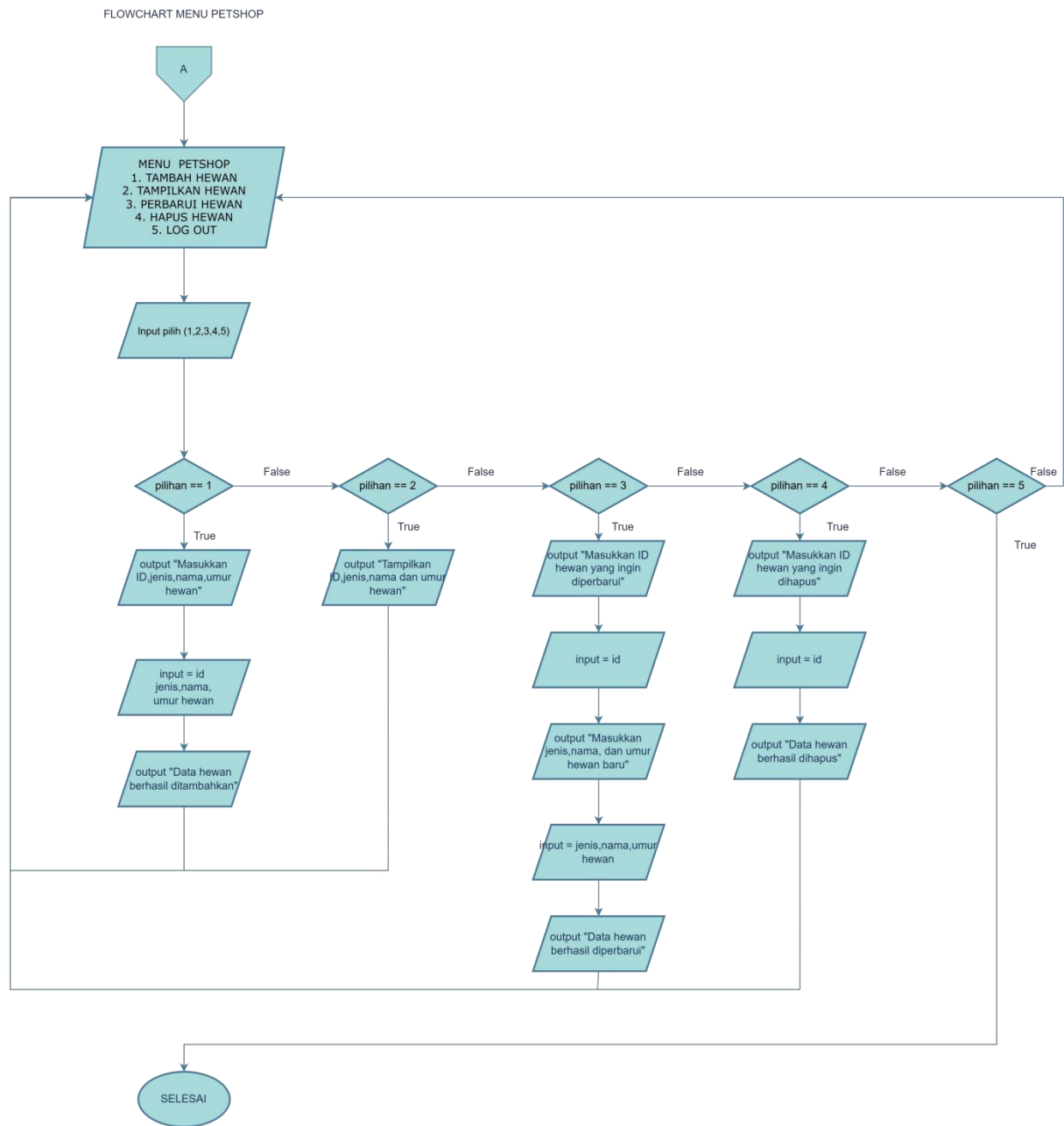
Disusun oleh:
Nabila Putri karni (2409106041)
Kelas (A2 '24)

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, memperbaiki, 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 4 yang kemarin tetapi disini ada perubahan yaitu program ini menerapkan prinsip pointer dimana terdapat fungsi dengan parameter address-of dan terdapat fungsi dengan parameter dereference.

3. Source Code

A. Struct 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. Menambah Hewan

```
void tambah_hewan_ptr(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";
    }
}
```

hewan* daftar artinya adalah pointer yang menunjuk ke array daftar_hewan, dan int* jumlah adalah pointer yang menunjuk ke jumlah hewan yang sedang ada. Di dalam fungsi, ketika kita menulis `cin >> (daftar[*jumlah]).id`, itu maksudnya kita mengambil nilai dari *jumlah (misalnya nilainya 3), lalu mengakses `daftar[3].id` untuk mengisi data hewan ke-4. Setelah data dimasukkan, kita menulis `(*jumlah)++` supaya jumlah hewan bertambah 1, dan karena kita menggunakan pointer, nilai aslinya di `main()` juga ikut bertambah.

D. Menampilkan 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. Mengubah hewan

```
void ubah_hewan_ptr(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 `ubah_hewan_ptr` menggunakan `hewan* daftar` sebagai pointer ke array, dan di dalamnya `(daftar + i)->id` digunakan untuk mengakses id hewan ke-*i* dengan pointer arithmetic (sama seperti `daftar[i].id`), lalu field seperti `jenis`, `nama`, dan `umur` diubah langsung menggunakan pointer melalui `(daftar + i)->` untuk mengakses dan mengisi data baru.

F. Menghapus Hewan

```
void hapus_hewan_ptr(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 `hapus_hewan_ptr` menggunakan pointer `hewan*` `daftar` dan `int*` `jumlah`, lalu melakukan pengecekan dengan `(daftar + i)->id == id` untuk mencari hewan yang ingin dihapus, kemudian menggeser data dengan `*(daftar + j) = *(daftar + j + 1)` yaitu menyalin elemen setelahnya ke posisi saat ini sebagai cara menghapus data, dan terakhir `(*jumlah)--` digunakan untuk mengurangi jumlah hewan karena satu data telah terhapus.

G. 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.

H. Menu Petshop

```
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

I. Menu switch

```
switch (pilihan)
{
    case 1:
        tambah_hewan_ptr(daftar_hewan, &jumlah_hewan);
        break;
    case 2:
        tampilkan_hewan(daftar_hewan, jumlah_hewan);
        break;
    case 3:
        ubah_hewan_ptr(daftar_hewan, jumlah_hewan);
        break;
    case 4:
        hapus_hewan_ptr(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: lily
Masukkan Umur Hewan: 2
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  lily    2 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: kelinci
Masukkan nama baru: mid
Masukkan umur baru hewan: 3
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      kelinci mid      3 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

