

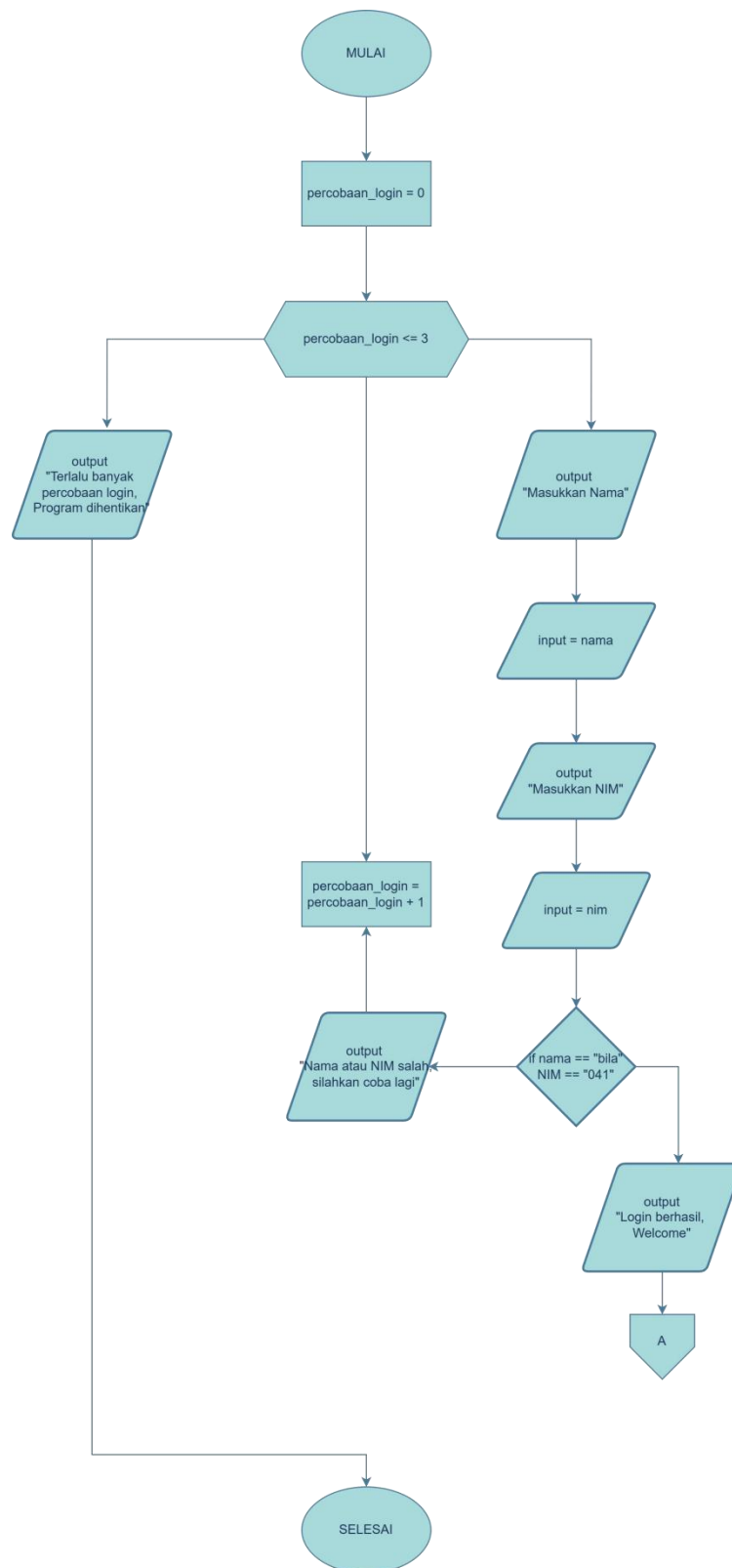
LAPORAN PRAKTIKUM
POSTTEST 6
ALGORITMA PEMROGRAMAN LANJUT



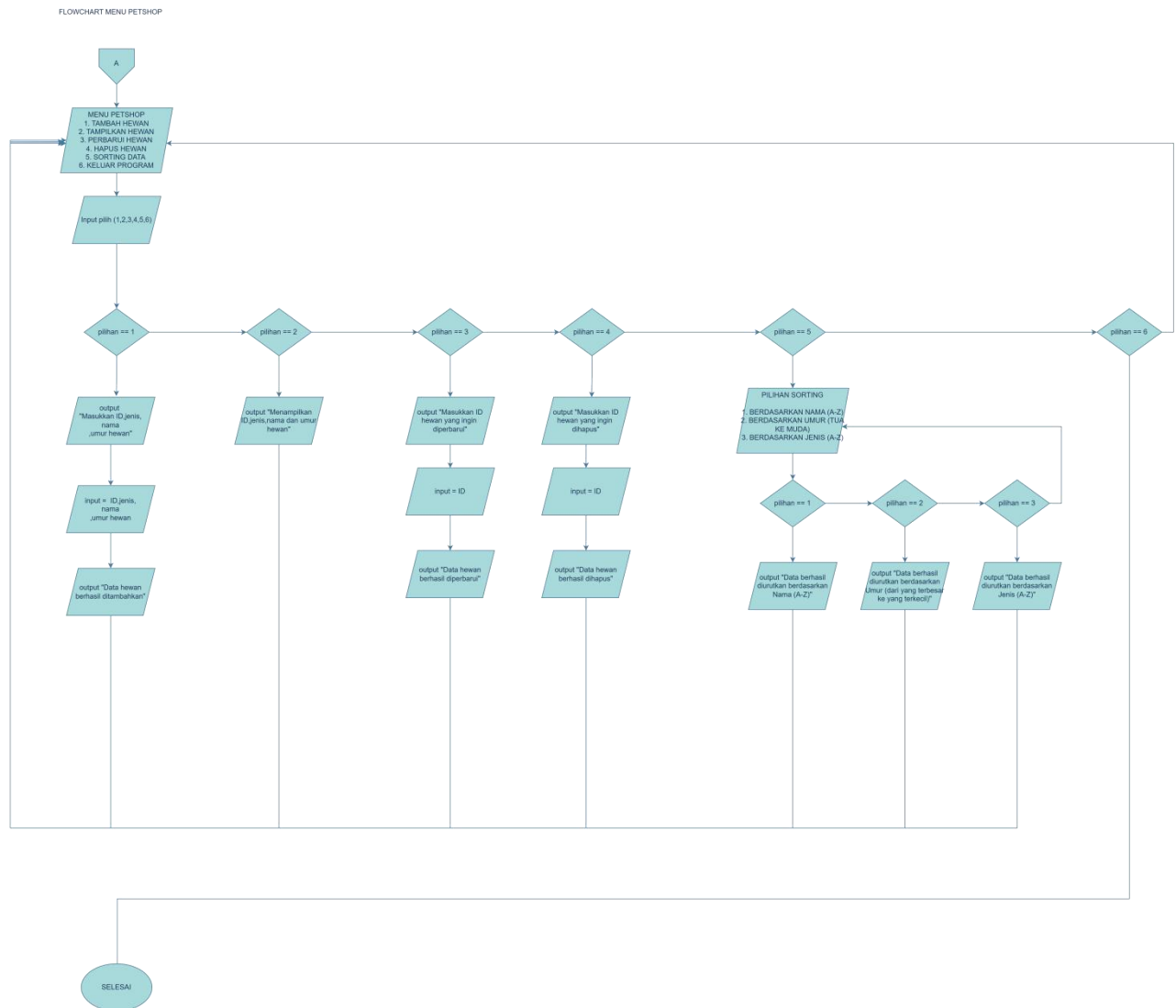
Disusun oleh:
Nabila Putri karni (2409106041)
Kelas (A2 '24)

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 5 yang kemarin tetapi disini ada tambahan yaitu program ini menerapkan minimal 3 metode sorting yang terdapat di modul, karena NIM saya ganjil maka saya menggunakan metode sorting huruf secara ascending, metode sorting angka secara descending, dan yang satu lagi sorting bebas.

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. 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 Sorting Nama (Ascending)

```
void sort_nama_ascending(hewan *daftar, int jumlah)
{
    for (int i = 0; i < jumlah - 1; i++)
    {
        for (int j = i + 1; j < jumlah; j++)
        {
            if ((daftar + i)->nama > (daftar + j)->nama)
            {
                swap(*(daftar + i), *(daftar + j));
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan Nama (A-Z).\n";
}
```

Fungsi `sort_nama_ascending` mengurutkan array hewan berdasarkan nama secara ascending (A-Z) menggunakan metode bubble sort sederhana dengan membandingkan dan menukar elemen jika perlu.

H. Fungsi Sorting Umur (Descending)

```
void sort_umur_descending(hewan *daftar, int jumlah)
{
    for (int i = 0; i < jumlah - 1; i++)
    {
        for (int j = i + 1; j < jumlah; j++)
        {
            if ((daftar + i)->umur < (daftar + j)->umur)
            {
                swap(*(daftar + i), *(daftar + j));
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan Umur ( dari yang terbesar ke yang terkecil).\n";
}
```

Fungsi `sort_umur_descending` mengurutkan array hewan berdasarkan umur secara descending (dari yang paling tua ke yang paling muda) dengan membandingkan dan menukar data menggunakan pointer.

I. Fungsi Sorting Jenis (Ascending)

```
void sort_jenis_ascending(hewan *daftar, int jumlah)
{
    for (int i = 0; i < jumlah - 1; i++)
    {
        for (int j = i + 1; j < jumlah; j++)
        {
            if ((daftar + i)->jenis > (daftar + j)->jenis)
            {
                swap(*(daftar + i), *(daftar + j));
            }
        }
    }
    cout << "Data berhasil diurutkan berdasarkan Jenis Hewan (A-Z).\n";
}
```

Fungsi `sort_jenis_ascending` mengurutkan array hewan berdasarkan jenis hewan secara ascending (A-Z) dengan membandingkan dan menukar data menggunakan pointer.

J. 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.

K. 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. SORTING DATA         |";
    cout << "\n| 6. KELUAR PROGRAM       |";
    cout << "\n=====";
    cout << "\nPilih (1/2/3/4/5/6): ";
    int pilihan;
    cin >> pilihan;
```

Tampilan Menu Petshop.

L. 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:
        int pilihan_sort;
        cout << "\n== PILIHAN SORTING ==";
        cout << "\n1. Berdasarkan Nama (A-Z)";
        cout << "\n2. Berdasarkan Umur (Tua ke Muda)";
        cout << "\n3. Berdasarkan Jenis (A-Z)";
        cout << "\nPilih metode sorting (1/2/3): ";
        cin >> pilihan_sort;

        if (pilihan_sort == 1)
            sort_nama_ascending(daftar_hewan, jumlah_hewan);
        else if (pilihan_sort == 2)
            sort_umur_descending(daftar_hewan, jumlah_hewan);
        else if (pilihan_sort == 3)
            sort_jenis_ascending(daftar_hewan, jumlah_hewan);
        else
            cout << "Pilihan sorting tidak valid.\n";
        break;
    case 6:
        cout << "Terima kasih telah menggunakan program petshop ini.\n";
        return 0;
    default:
        cout << "Pilihan tidak valid.\n";
}

return 0;
}
```

switch(pilihan) berfungsi untuk menangani aksi yang dipilih user dalam menu utama program. Jika user memilih 1, program akan memanggil tambah_hewan_ptr untuk menambahkan data hewan baru; jika memilih 2, program akan menampilkan seluruh data hewan menggunakan tampilkan_hewan; jika memilih 3, program memungkinkan user mengubah data hewan tertentu dengan ubah_hewan_ptr; jika memilih 4, program akan menghapus data hewan berdasarkan ID melalui hapus_hewan_ptr. Pada pilihan 5, program membuka sub-menu sorting, di mana user bisa memilih untuk mengurutkan data berdasarkan nama (A-Z), umur (dari tua ke muda), atau jenis hewan (A-Z) menggunakan fungsi sort_nama_ascending, sort_umur_descending, atau sort_jenis_ascending. Jika memilih 6, program menampilkan pesan terima kasih dan keluar. Apabila input tidak valid, default akan menampilkan pesan bahwa pilihan tidak valid, dan setelah setiap operasi (kecuali keluar), program akan kembali ke menu utama.

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. SORTING DATA  |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 1
Masukkan ID Hewan: 199
Masukkan jenis hewan: kucing
Masukkan Nama Hewan: Reo
Masukkan Umur Hewan: 7
Data hewan telah berhasil ditambahkan.

=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA  |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 1
Masukkan ID Hewan: 169
Masukkan jenis hewan: Anjing
Masukkan Nama Hewan: Andrew
Masukkan Umur Hewan: 3
Data hewan telah berhasil ditambahkan.
```

Gambar 4.1 Tampilan Terminal

```
=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA   |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 1
Masukkan ID Hewan: 40
Masukkan jenis hewan: kelinci
Masukkan Nama Hewan: milo
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. SORTING DATA   |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 1
Masukkan ID Hewan: 153
Masukkan jenis hewan: bunglon
Masukkan Nama Hewan: bubu
Masukkan Umur Hewan: 1
Data hewan telah berhasil ditambahkan.
```

Gambar 4.2 Tampilan Terminal

```
=====
| MENU PET SHOP |
=====
| 1. TAMBAH HEWAN |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN |
| 4. HAPUS HEWAN |
| 5. SORTING DATA |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 2

===== DATA HEWAN =====
ID      Jenis   Nama      Umur
-----
199     kucing   reo       7 tahun
169     anjing   andrew    3 tahun
40      kelinci   milo      2 tahun
153     bunglon   bubu      1 tahun
-----

=====
| MENU PET SHOP |
=====
| 1. TAMBAH HEWAN |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN |
| 4. HAPUS HEWAN |
| 5. SORTING DATA |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 3
Masukkan ID hewan yang ingin diubah: 153
Masukkan jenis baru: bunglon
Masukkan nama baru: bina
Masukkan umur baru hewan: 2
Data hewan berhasil diperbarui.
```

Gambar 4.3 Tampilan Terminal

```
=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN|
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA   |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 2

===== DATA HEWAN =====
ID      Jenis  Nama    Umur
-----
199      kucing reo      7 tahun
169      anjing  andrew  3 tahun
40       kelinci milo    2 tahun
153      bunglon bina    2 tahun
-----
```

Gambar 4.4 Tampilan Terminal

```
=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA  |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 4
Masukkan ID hewan yang ingin dihapus: 153
Data hewan berhasil dihapus.

=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA  |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 2

===== DATA HEWAN =====
ID      Jenis   Nama    Umur
-----
199     kucing   reo      7 tahun
169     anjing   andrew   3 tahun
40      kelinci   milo     2 tahun
-----
```

Gambar 4.5 Tampilan Terminal

```
=====
| MENU PET SHOP |
=====
| 1. TAMBAH HEWAN |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN |
| 4. HAPUS HEWAN |
| 5. SORTING DATA |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 5

== PILIHAN SORTING ==
1. Berdasarkan Nama (A-Z)
2. Berdasarkan Umur (Tua ke Muda)
3. Berdasarkan Jenis (A-Z)
Pilih metode sorting (1/2/3): 1
Data berhasil diurutkan berdasarkan Nama (A-Z).

=====
| MENU PET SHOP |
=====
| 1. TAMBAH HEWAN |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN |
| 4. HAPUS HEWAN |
| 5. SORTING DATA |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 2

===== DATA HEWAN =====
ID      Jenis  Nama  Umur
-----
169     anjing  andrew 3 tahun
40      kelinci milo  2 tahun
199     kucing  reo    7 tahun
-----
```

Gambar 4.6 Tampilan Terminal


```
=====
|  MENU PET SHOP  |
=====
| 1. TAMBAH HEWAN |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN |
| 4. HAPUS HEWAN |
| 5. SORTING DATA |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 5

== PILIHAN SORTING ==
1. Berdasarkan Nama (A-Z)
2. Berdasarkan Umur (Tua ke Muda)
3. Berdasarkan Jenis (A-Z)
Pilih metode sorting (1/2/3): 2
Data berhasil diurutkan berdasarkan Umur ( dari yang terbesar ke yang terkecil).

=====
|  MENU PET SHOP  |
=====
| 1. TAMBAH HEWAN |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN |
| 4. HAPUS HEWAN |
| 5. SORTING DATA |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 2

===== DATA HEWAN =====
ID      Jenis   Nama    Umur
-----
199     kucing  reo     7 tahun
169     anjing  andrew  3 tahun
40      kelinci  milo    2 tahun
-----
```

Gambar 4.7 Tampilan Terminal

```

=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA   |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 5

== PILIHAN SORTING ==
1. Berdasarkan Nama (A-Z)
2. Berdasarkan Umur (Tua ke Muda)
3. Berdasarkan Jenis (A-Z)
Pilih metode sorting (1/2/3): 3
Data berhasil diurutkan berdasarkan Jenis Hewan (A-Z).

=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA   |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 2

===== DATA HEWAN =====
ID      Jenis  Nama    Umur
-----
169     anjing  andrew  3 tahun
40      kelinci  milo    2 tahun
199     kucing   reo     7 tahun
-----

```

Gambar 4.8 Tampilan Terminal

```

=====
|   MENU PET SHOP   |
=====
| 1. TAMBAH HEWAN   |
| 2. TAMPILKAN HEWAN |
| 3. UBAH HEWAN     |
| 4. HAPUS HEWAN    |
| 5. SORTING DATA   |
| 6. KELUAR PROGRAM |
=====
Pilih (1/2/3/4/5/6): 6
Terima kasih telah menggunakan program petshop ini.

```

Gambar 4.9 Tampilan Terminal

5. Langkah-Langkah Git pada VSCode

```
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-6> git add .
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-6> git commit -m "Finish Posttest 6"
[main 0250622] Finish Posttest 6
3 files changed, 252 insertions(+)
create mode 100644 post-test/post-test-6/2409106041-NabilaPutriKarni-PT-6.cpp
create mode 100644 post-test/post-test-6/2409106041-NabilaPutriKarni-PT-6.exe
create mode 100644 post-test/post-test-6/2409106041-NabilaPutriKarni-PT-6.pdf
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-6> 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), 1.06 MiB | 524.00 KiB/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
   bdcdfb1..0250622  main -> main
PS C:\Users\MSI GF63\Documents\Praktikum apl\praktikum-apl\post-test\post-test-6> █
```

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.