

LAPORAN PRAKTIKUM
POSTTEST (VI)
ALGORITMA PEMROGRAMAN LANJUT

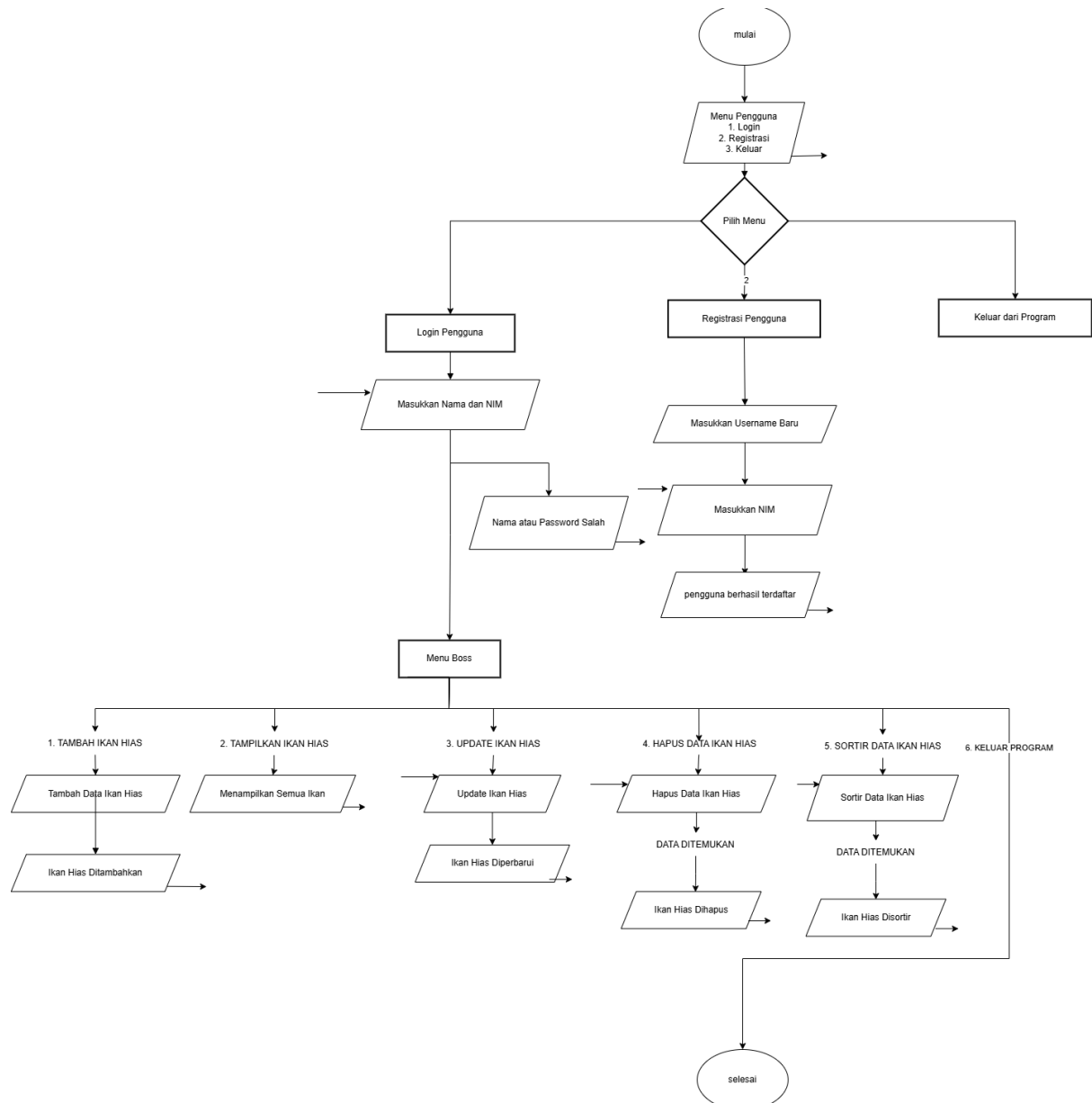


Disusun oleh:
Akhmad Zifa Al-Fatih (2409106025)
Kelas (A2'24)

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA

2025

1. Flowchart



2. Analisis Program

Program ini adalah sistem manajemen ikan hias berbasis C++ yang memungkinkan pengguna untuk menambah, melihat, mengedit, menyortir dan menghapus data ikan di sebuah toko. Pada kali ini tersedia fitur dengan menu user yang mana menu ini dapat menjalankan login, registrasi dan keluar dari program, pengguna baru dapat melakukan registrasi yang mana nanti data user akan disimpan dalam array dan dibatasi sebanyak 100 data pengguna. Pengguna harus login dengan nama dan NIM yang benar. Setelah login, tersedia menu utama dengan beberapa pilihan, seperti menampilkan daftar ikan, menambah ikan baru, mengubah data ikan, menghapus ikan, menyortir ikan dan keluar dari program. Data ikan disimpan dalam array dengan batas maksimal 100 ikan. Program juga memiliki validasi input untuk memastikan data dikelola dengan benar dan memberikan pesan kesalahan jika terjadi kesalahan input. Program ini berguna untuk mengelola data ikan hias di sebuah toko atau koleksi pribadi.

Dengan program ini, pengguna dapat:

- Menyimpan dan mengorganisir data ikan, termasuk nama, jenis, dan harga.
- Melihat daftar ikan yang sudah tersimpan.
- Menambahkan ikan baru ke dalam daftar.
- Mengubah informasi ikan jika ada kesalahan atau perubahan harga.
- Menghapus ikan yang tidak tersedia lagi.
- Menyortir Ikan dengan ketentuan :
Sorting Berdasarkan nama (Sesuai abjad) secara Ascending, Sorting angka(Harga) secara Descending, dan Sortir Huruf (Jenis) secara abjad dengan Ascending.

3. Source Code

A. Inisialisasi dan Library

Bagian ini mengimpor library yang diperlukan dan mendefinisikan namespace yang digunakan, menginisialisasi ukuran maksimum array ikan dan array pengguna, menyimpan variable array untuk menyimpan data ikan dan data pengguna,

Source Code:

```

#include <iostream>
#include <string>
using namespace std;

#define MAX_IKAN 100
#define MAX_PENGGUNA 100

struct DataIkan {
    string nama;
    string jenis;
    int harga;
};

struct DataPengguna {
    string nama;
    string nim;
    DataIkan* ikan; // Menggunakan pointer untuk array ikan
    int jumlahIkan = 0;
};

DataPengguna* dataPengguna = nullptr; // Menggunakan pointer
untuk array pengguna
int jumlahPengguna = 0;

// Deklarasi fungsi dan prosedur
void tampilkanMenuUtama();
void prosesLogin();
void prosesRegister();
void menuManajemenIkan(int indexPengguna);
void tampilkanIkan(int indexPengguna);
void tambahIkan(int indexPengguna);
void ubahIkan(int indexPengguna);
void hapusIkan(int indexPengguna);
int hitungIkanRekursif(int indexPengguna, int index = 0);
void tampilkanInfo(const string &nama);
void tampilkanInfo(const string &nama, const string &nim);
void sortByNamaAscending(int indexPengguna); // Bubble Sort
void sortByHargaDescending(int indexPengguna); // Selection Sort
void sortByJenisAscending(int indexPengguna); // Insertion Sort

```

B. Login

Fungsi ini bertugas untuk melakukan proses login, registrasi dan keluar. Dengan input nama dan NIM, Pengguna memiliki 3 kesempatan untuk login. **Source Code:**

```

void tampilkanMenuUtama() {
    while (true) {
        cout << "\n=== Login Toko Ikan Hias ===" << endl;
        cout << "1. Login\n2. Register\n3. Exit\n";
        cout << "===== " << endl;
        cout << "Pilih menu (1-3): ";

        int pilihan;
        cin >> pilihan;
        cin.ignore();

        switch (pilihan) {
            case 1: prosesLogin(); break;
            case 2: prosesRegister(); break;
            case 3: exit(0);
            default: cout << "\nPilihan tidak valid.\n";
        }
    }
}

void prosesLogin() {
    string nama, nim;
    static int percobaan = 0;

    while (percobaan < 3) {
        cout << "\nMasukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        getline(cin, nim);

        bool loginBerhasil = false;
        int indexPengguna = -1;

        for (int i = 0; i < jumlahPengguna; i++) {
            if ((dataPengguna + i)->nama == nama && (dataPengguna + i)->nim ==
nim) {
                loginBerhasil = true;
                indexPengguna = i;
                break;
            }
        }

        if (loginBerhasil) {
            percobaan = 0;
            cout << "\nLogin berhasil! Selamat Datang " << nama << endl;

```

```

    tampilkanInfo(nama, nim);
    menuManajemenIkan(indexPengguna);
    return;
} else {
    percobaan++;
    cout << "\nLogin gagal. Sisa percobaan: " << 3 - percobaan << endl;
}
}
cout << "\nAnda telah melebihi batas percobaan login.\n";
}

```

```

void prosesRegister() {
    if (jumlahPengguna >= MAX_PENGGUNA) {
        cout << "\nBatas pengguna telah tercapai.\n";
        return;
    }

    cout << "\n=== Register Pengguna ===" << endl;
    cout << "Masukkan nama: ";
    getline(cin, (dataPengguna + jumlahPengguna)->nama);
    cout << "Masukkan NIM: ";
    getline(cin, (dataPengguna + jumlahPengguna)->nim);

    jumlahPengguna++;
    cout << "\nPengguna berhasil terdaftar!\n";
    tampilkanInfo((dataPengguna + jumlahPengguna - 1)->nama);
}

```

```

void tampilkanInfo(const string &nama) {
    cout << "\nInfo Pengguna:\n";
    cout << "Nama: " << nama << "\n";
}

```

```

void tampilkanInfo(const string &nama, const string &nim) {
    tampilkanInfo(nama);
    cout << "NIM: " << nim << "\n";
}

```

C. Tampilan Menu Program

ini menampilkan menu program manajemen ikan hias yang dapat menginput pilihan dari tampilan yang nanti nya akan di proses di bagian program CRUD dengan tambahan menu sortir ikan.

Source Code:


```

void menuManajemenIkan(int indexPengguna) {
    DataPengguna* pengguna = dataPengguna + indexPengguna;

    while (true) {
        cout << "\n===== ";
        cout << "\n=== Program Manajemen Ikan Hias === ";
        cout << "\n===== ";
        cout << "\n1. Tampilkan Ikan          | ";
        cout << "\n2. Tambah Ikan          | ";
        cout << "\n3. Ubah Ikan          | ";
        cout << "\n4. Hapus Ikan          | ";
        cout << "\n5. Sortir Ikan          | ";
        cout << "\n6. Keluar Program          | ";
        cout << "\n===== ";
        cout << "\nPilih menu: ";

        int pilihanMenu;
        cin >> pilihanMenu;
        cin.ignore();

        switch (pilihanMenu) {
            case 1:
                tampilkanIkan(indexPengguna);
                cout << "\nTotal ikan (dihitung secara rekursif): "
                    << hitungIkanRekursif(indexPengguna) << endl;
                break;
            case 2: tambahIkan(indexPengguna); break;
            case 3: ubahIkan(indexPengguna); break;
            case 4: hapusIkan(indexPengguna); break;
            case 5: SortirIkan(indexPengguna); break; //
            case 6: return;
            default: cout << "\nPilihan tidak valid.\n";
        }
    }
}

```

E. Program CRUD

Pada Bagian program CRUD ini, menggunakan dari struktur kendali switch-case untuk menangani berbagai pilihan dalam menu manajemen ikan hias, berikut penjelasan rinci dari switch case CRUD ini.

1. Case 1: Menampilkan Daftar Ikan

Mengecek apakah ada ikan dalam daftar. Jika tidak ada (jumlahIkan == 0), program menampilkan pesan bahwa daftar kosong. Jika ada, program akan mencetak daftar ikan satu per satu dengan informasi: Nomor urut, Nama ikan, Jenis ikan, Harga ikan.

Source Code:

```
void tampilkanIkan(int indexPengguna) {
    DataPengguna* pengguna = &dataPengguna[indexPengguna];

    if (pengguna->jumlahIkan == 0) {
        cout << "\nBelum ada ikan yang tersedia.\n";
    } else {
        cout << "\n=== Daftar Ikan Hias ===\n";
        for (int i = 0; i < pengguna->jumlahIkan; i++) {
            DataIkan* ikan = pengguna->ikan + i;
            cout << "Ikan ke-" << i+1 << ":\n";
            cout << "  Nama: " << ikan->nama << "\n";
            cout << "  Jenis: " << ikan->jenis << "\n";
            cout << "  Harga: Rp" << ikan->harga << "\n";
            cout << "-----\n";
        }
    }
}
```

2. Case 2 : Menambahkan Ikan Baru

Memeriksa apakah masih ada ruang dalam array (jumlahIkan < MAX_IKAN). Jika masih bisa menambah ikan ; Meminta input nama, jenis, dan harga ikan, menyimpannya di array pada indeks jumlahIkan, menambah nilai panjang untuk memperbarui jumlah ikan dalam daftar. Jika daftar sudah penuh, program menampilkan pesan bahwa kapasitas sudah maksimal.

Source code :

```

void tambahIkan(int indexPengguna) {
    DataPengguna* pengguna = dataPengguna + indexPengguna;

    if (pengguna->jumlahIkan >= MAX_IKAN) {
        cout << "\nKapasitas penuh!\n";
        return;
    }

    cout << "\n=== Tambah Ikan ===\n";
    DataIkan ikanBaru;

    cout << "Nama ikan: ";
    getline(cin, ikanBaru.nama);
    cout << "Jenis ikan: ";
    getline(cin, ikanBaru.jenis);
    cout << "Harga ikan: Rp";
    cin >> ikanBaru.harga;
    cin.ignore();

    *(pengguna->ikan + pengguna->jumlahIkan) = ikanBaru;
    pengguna->jumlahIkan++;

    cout << "\nIkan berhasil ditambahkan!\n";
}

```

3. Case 3 : Mengubah Data Ikan

Memeriksa apakah ada ikan dalam daftar. Jika tidak ada (`jumlahIkan == 0`), program menampilkan pesan bahwa daftar kosong. Jika ada meminta pengguna memasukkan nomor ikan yang ingin diubah (`index`). Memeriksa apakah nomor ikan valid. Jika valid, pengguna memasukkan nama, jenis, dan harga baru, lalu data diperbarui. Jika nomor tidak valid, program menampilkan pesan error.

Source code :

```

void ubahIkan(int indexPengguna) {
    DataPengguna* pengguna = &dataPengguna[indexPengguna];

    if (pengguna->jumlahIkan == 0) {
        cout << "\nBelum ada ikan yang tersedia.\n";
        return;
    }

    tampilkanIkan(indexPengguna);
    cout << "\nPilih nomor ikan yang akan diubah: ";
    int index;
    cin >> index;
    cin.ignore();

    if (index < 1 || index > pengguna->jumlahIkan) {
        cout << "\nNomor tidak valid.\n";
        return;
    }

    cout << "\n=== Ubah Ikan ===\n";
    DataIkan* ikan = pengguna->ikan + (index - 1);
    cout << "Nama baru: ";
    getline(cin, ikan->nama);
    cout << "Jenis baru: ";
    getline(cin, ikan->jenis);
    cout << "Harga baru: Rp";
    cin >> ikan->harga;
    cin.ignore();

    cout << "\nData ikan berhasil diubah!\n";
}

```

4. Case 4 : Menghapus Data Ikan

Memeriksa apakah daftar ikan kosong. Jika kosong, pengguna diberi pesan bahwa tidak ada ikan yang bisa dihapus. Jika ada ikan, program meminta nomor ikan yang akan dihapus (index). Validasi nomor ikan apakah berada dalam rentang yang benar. Jika valid, program menggeser elemen setelahnya untuk menutupi posisi ikan yang dihapus. Mengurangi nilai panjang agar jumlah ikan yang tersimpan berkurang. Jika nomor tidak valid, program menampilkan pesan error.

Source code :

```

void hapusIkan(int indexPengguna) {
    DataPengguna* pengguna = dataPengguna + indexPengguna;

    if (pengguna->jumlahIkan == 0) {
        cout << "\nBelum ada ikan yang tersedia.\n";
        return;
    }

    tampilkanIkan(indexPengguna);
    cout << "\nPilih nomor ikan yang akan dihapus: ";
    int index;
    cin >> index;
    cin.ignore();

    if (index < 1 || index > pengguna->jumlahIkan) {
        cout << "\nNomor tidak valid.\n";
        return;
    }

    for (int i = index-1; i < pengguna->jumlahIkan-1; i++) {
        *(pengguna->ikan + i) = *(pengguna->ikan + i + 1);
    }
    pengguna->jumlahIkan--;

    cout << "\nIkan berhasil dihapus!\n";
}

int hitungIkanRekursif(int indexPengguna, int index) {
    DataPengguna* pengguna = dataPengguna + indexPengguna;

    if (index >= pengguna->jumlahIkan) {
        return 0;
    }
    return 1 + hitungIkanRekursif(indexPengguna, index + 1);
}

```

5. Case 5 : Menyortir Data Ikan

Dengan ketentuan penyortiran sesuai di intruksi yaitu : NIM GANJIL

- menggunakan metode sorting huruf secara ascending (bubble sort)
- metode sorting angka secara descending (Selection Sort)

- dan untuk satu metode sorting lainnya dibebaskan (Insertation Sort)

Source code:

```
void SortirIkan(int indexPengguna) {
    cout << "\n=== Menu Sortir Ikan ===";
    cout << "\n1. Sortir berdasarkan Nama (A-Z)";
    cout << "\n2. Sortir berdasarkan Harga (Tertinggi-
Terendah)";
    cout << "\n3. Sortir berdasarkan Jenis (A-Z)";
    cout << "\nPilihan: ";
    int sortChoice;
    cin >> sortChoice;
    cin.ignore();

    switch(sortChoice) {
        case 1:
            sortByNamaAscending(indexPengguna);
            cout << "\nIkan berhasil diurutkan berdasarkan Nama
(A-Z)\n";
            break;
        case 2:
            sortByHargaDescending(indexPengguna);
            cout << "\nIkan berhasil diurutkan berdasarkan Harga
(Tertinggi-Terendah)\n";
            break;
        case 3:
            sortByJenisAscending(indexPengguna);
            cout << "\nIkan berhasil diurutkan berdasarkan Jenis
(A-Z)\n";
            break;
        default: cout << "\nPilihan tidak valid.\n";
    }
}
```

6. Case 6 : Keluar Dari Program

Menampilkan pesan perpisahan. Menghentikan program dengan return 0.

```
case 5: return;
        default: cout << "\nPilihan tidak valid.\n";
    }
}
```

4. Uji Coba dan Hasil Output

(Sertakan tangkapan layar atau hasil output dari program setelah dijalankan.)

```
=== Login Toko Ikan Hias ===
1. Login
2. Register
3. Exit
=====
Pilih menu (1-3): 2

=== Register Pengguna ===
Masukkan nama: zif
Masukkan NIM: 025

Pengguna berhasil terdaftar!

Info Pengguna:
Nama: zif
```

Gambar 4.1 registrasi di menu login


```

=== Login Toko Ikan Hias ===
1. Login
2. Register
3. Exit
=====
Pilih menu (1-3): 1

Masukkan Nama: zif
Masukkan NIM: 025

Login berhasil! Selamat Datang zif

Info Pengguna:
Nama: zif
NIM: 025

=====
=== Program Manajemen Ikan Hias ===
=====
1. Tampilkan Ikan      |
2. Tambah Ikan         |
3. Ubah Ikan           |
4. Hapus Ikan          |
5. Sortir Ikan         |
6. Keluar Program      |
=====

```

Gambar 4.2 Login dan mengeluarkan menu program jika berhasil

```

=== Tambah Ikan ===
Nama ikan: Arwana Legend
Jenis ikan: Tawar
Harga ikan: Rp10000000

Ikan berhasil ditambahkan!

```

```

=== Tambah Ikan ===
Nama ikan: Piranha
Jenis ikan: Tawar
Harga ikan: Rp500000

Ikan berhasil ditambahkan!

```

```
=== Tambah Ikan ===  
Nama ikan: Koi Legend  
Jenis ikan: tawar  
Harga ikan: Rp1000000  
  
Ikan berhasil ditambahkan!
```

Gambar 4.3 Menambahkan Ikan pada pilihan 2

```
Pilih menu: 1  
  
=== Daftar Ikan Hias ===  
Ikan ke-1:  
  Nama: Koi Legend  
  Jenis: tawar  
  Harga: Rp1000000  
-----  
Ikan ke-2:  
  Nama: Piranha  
  Jenis: Tawar  
  Harga: Rp500000  
-----  
Ikan ke-3:  
  Nama: Arwana Legend  
  Jenis: Tawar  
  Harga: Rp1000000  
-----  
  
Total ikan (dihitung secara rekursif): 3
```

Gambar 4.4 Menampilkan data Ikan pada pilihan 1

```
Pilih menu: 3

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi Legend
  Jenis: tawar
  Harga: Rp1000000
-----
Ikan ke-2:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp500000
-----
Ikan ke-3:
  Nama: Arwana Legend
  Jenis: Tawar
  Harga: Rp10000000
-----

Pilih nomor ikan yang akan diubah: 2

=== Ubah Ikan ===
Nama baru: Piranha
Jenis baru: Tawar
Harga baru: Rp1100000

Data ikan berhasil diubah!
```

Gambar 4.5 Mengubah data ikan pada ikan ke 2 dan membuktikan pada pilihan 1

```
Pilih menu: 4

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi Legend
  Jenis: tawar
  Harga: Rp1000000
-----
Ikan ke-2:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp1100000
-----
Ikan ke-3:
  Nama: Arwana Legend
  Jenis: Tawar
  Harga: Rp10000000
-----

Pilih nomor ikan yang akan dihapus: 3

Ikan berhasil dihapus!
```

```

Pilih menu: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi Legend
  Jenis: tawar
  Harga: Rp1000000
-----
Ikan ke-2:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp1100000
-----

Total ikan (dihitung secara rekursif): 2

```

Gambar 4.6 Menghapus ikan ke 1 pada pilihan 4 dan membuktikan dengan pilihan 1

```

Pilih menu: 5

=== Menu Sortir Ikan ===
1. Sortir berdasarkan Nama (A-Z)
2. Sortir berdasarkan Harga (Tertinggi-Terendah)
3. Sortir berdasarkan Jenis (A-Z)
Pilihan: 1

Ikan berhasil diurutkan berdasarkan Nama (A-Z)

```

```

Pilih menu: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi Legend
  Jenis: tawar
  Harga: Rp1000000
-----
Ikan ke-2:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp1100000
-----

Total ikan (dihitung secara rekursif): 2

```

Gambar 4.7 dan 4.8 Menyortir ikan ke dengan pilihan 1 di menu pilihan 5 dan menampilkan kembali dengan pilihan 1

```

Pilih menu: 5

=== Menu Sortir Ikan ===
1. Sortir berdasarkan Nama (A-Z)
2. Sortir berdasarkan Harga (Tertinggi-Terendah)
3. Sortir berdasarkan Jenis (A-Z)
Pilihan: 2

Ikan berhasil diurutkan berdasarkan Harga (Tertinggi-Terendah)

```

```

Pilih menu: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp1100000
-----
Ikan ke-2:
  Nama: Koi Legend
  Jenis: tawar
  Harga: Rp1000000
-----

Total ikan (dihitung secara rekursif): 2

```

Gambar 4.9 dan 4.10 Menyortir ikan ke dengan pilihan 2 di menu pilihan 5 dan menampilkan kembali dengan pilihan 1

(Untuk pilihan 3 di sorting tidak dilakukan karena jenis ikan sama)

```

Pilih menu: 6

=== Login Toko Ikan Hias ===
1. Login
2. Register
3. Exit
=====
Pilih menu (1-3): 3
PS D:\praktikum-apl\post-test\post-test-apl-6>

```

Gambar 4.11 keluar pada pilihan 5 dan mengembalikan ke menu login serta memilih opsi 3

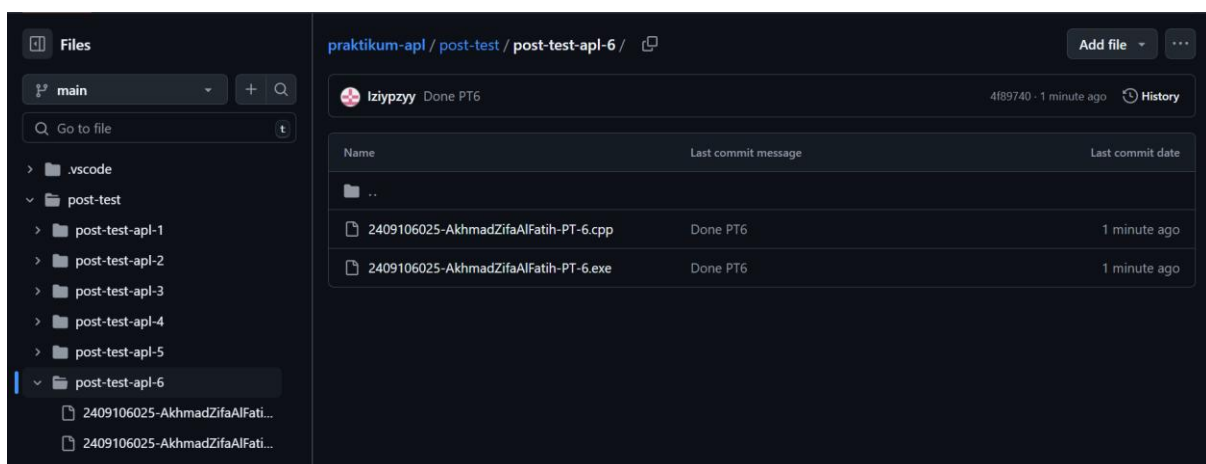
5. Langkah-Langkah Git pada VSCode

```

PS D:\praktikum-apl> git add .
PS D:\praktikum-apl> git commit -m "Done PT6"
[main 4f89740] Done PT6
 3 files changed, 372 insertions(+)
 create mode 100644 .vscode/launch.json
 create mode 100644 post-test/post-test-apl-6/2409106025-AkhmadZifaAlFatih-PT-6.cpp
 create mode 100644 post-test/post-test-apl-6/2409106025-AkhmadZifaAlFatih-PT-6.exe
PS D:\praktikum-apl> git push origin main
Enumerating objects: 11, done.
Counting objects: 100% (11/11), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (8/8), 679.76 KiB | 4.02 MiB/s, done.
Total 8 (delta 1), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (1/1), completed with 1 local object.
To https://github.com/Iziypzyy/praktikum-apl.git
 a89c5bd..4f89740  main -> main
PS D:\praktikum-apl>

```

Gambar 5.1 Langkah git di terminal dari git add,git commit -m "Done PT6",git push origin main



Gambar 5.2 Memeriksa pada github