

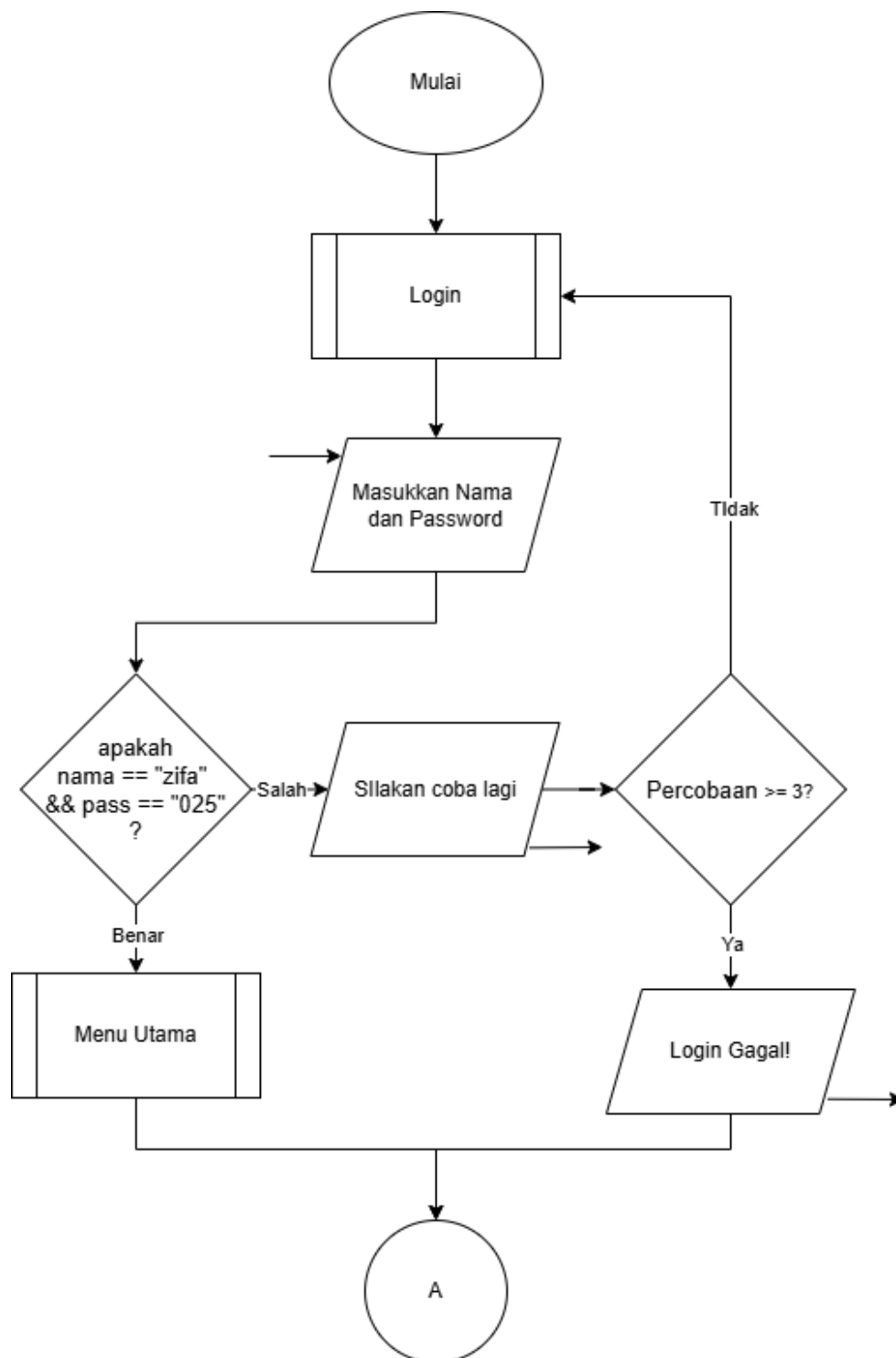
**LAPORAN PRAKTIKUM  
POSTTEST (II)  
ALGORITMA PEMROGRAMAN LANJUT**



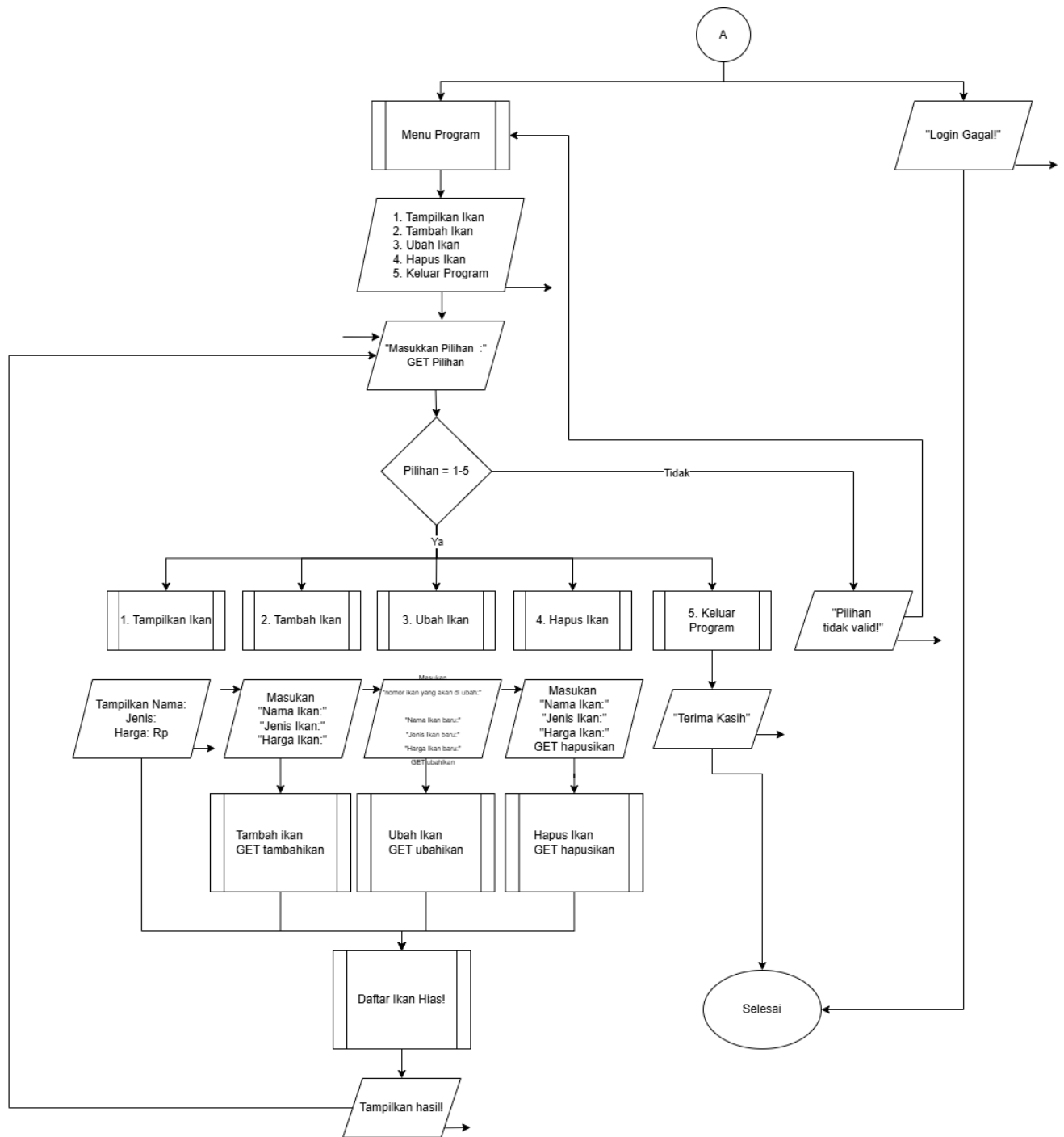
**Disusun oleh:  
Akhmad Zifa Al-Fatih (2409106025)  
Kelas (A2 '24)**

**PROGRAM STUDI INFORMATIKA  
UNIVERSITAS MULAWARMAN  
SAMARINDA  
2025**

## 1. Flowchart



Gambar 1.1 Flowchart login



Gambar 1.2 Flowchart Setelah login

## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini adalah sistem manajemen ikan hias berbasis C++ yang memungkinkan pengguna untuk menambah, melihat, mengedit, dan menghapus data ikan di sebuah toko.

Sebelum mengakses fitur utama, 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, 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.

### **2.2 Penjelasan Alur & Algoritma**

#### **Alur Kerja Program**

Adapun alur dari program yang saya buat terdiri dari 5 bagian fungsi yaitu; Fungsi login sebagai ketentuan dari post-test kali ini dengan menggunakan, menginput dan memproses nama dan nim sebagai password untuk login, kemudian fungsi tampilan menu, ketiga fungsi looping untuk mengembalikan ke tampilan menu program, keempat proses CRUD dan terakhir untuk keluar dari program.

#### **1. Login**

- Program meminta nama dan NIM pengguna.
- Jika sesuai dengan data valid (Zifa dan 025), pengguna bisa masuk.
- Jika salah, diberikan tiga kali percobaan sebelum program berhenti.

#### **2. Menampilkan Menu program**

Setelah login berhasil, pengguna akan melihat menu utama dengan pilihan:

- 1. Tampilkan Ikan → Menampilkan daftar ikan yang sudah tersimpan.
- 2. Tambah Ikan → Menambahkan ikan baru dengan nama, jenis, dan harga.

- 3. Ubah Ikan → Mengedit informasi ikan yang sudah ada.
- 4. Hapus Ikan → Menghapus ikan dari daftar.
- 5. Keluar Program → Menghentikan program.

### **3. Proses Sesuai Pilihan Pengguna**

- Jika memilih 1 (Tampilkan Ikan) Program menampilkan daftar ikan jika ada. Jika kosong, muncul pesan bahwa belum ada ikan.
- Jika memilih 2 (Tambah Ikan) Program meminta pengguna memasukkan nama, jenis, dan harga ikan, lalu menyimpannya ke dalam array.
- Jika memilih 3 (Ubah Ikan) Program meminta nomor ikan yang ingin diubah, lalu pengguna memasukkan data baru.
- Jika memilih 4 (Hapus Ikan) Program meminta nomor ikan yang ingin dihapus, lalu menggeser data yang tersisa agar tidak ada data kosong di tengah.
- Jika memilih 5 (Keluar Program) Program menampilkan pesan perpisahan dan berhenti.

### **4. Program Berjalan dalam Looping**

- Setelah menyelesaikan satu operasi, program kembali ke menu utama hingga pengguna memilih keluar.
- Jika pengguna memasukkan pilihan yang salah, akan muncul pesan error dan diminta memilih ulang.

### 3. Source Code

#### A. Inisialisasi dan Library

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

**Source Code:**

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

#define MAX_IKAN 100

string namaIkan[MAX_IKAN];
string jenisIkan[MAX_IKAN];
int hargaIkan[MAX_IKAN];
int panjang = 0; // Jumlah elemen saat ini
```

#### B. Login

Fungsi ini bertugas untuk melakukan proses login dengan meminta input nama dan NIM, Pengguna memiliki 3 kesempatan untuk login.

**Source Code:**

```
int main() {
    // Variabel untuk Login
    string nama, nim;
    string namavalid = "Zifa";
    string nimvalid = "025";
    int percobaan = 0;

    // Proses Login
    while (percobaan < 3) {
        cout << "=== Login Toko Ikan Hias ===" << endl;
        cout << "Masukkan Nama: ";
        getline(cin, nama);
        cout << "Masukkan NIM: ";
        getline(cin, nim);
        cout << endl;

        if (nama == namavalid && nim == nimvalid) {
            cout << "Login berhasil! Selamat Datang Iciboss " << namavalid << endl

```

```

<< endl;
        break;
    }

    else {
        percobaan++;
        cout << "kasi bener dong, sisa percobaan : " << 3 - percobaan << endl;
    }
}

if (percobaan == 3) {
    cout << "Yah input user pass nya salah mulu, program berhenti." << endl;
    return 0;
}

```

### 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.

**Source Code:**

```

// Menampilkan menu
cout << "=====\n";
cout << "=== Program Manajemen Ikan Hias ===\n";
cout << "=====\n";
cout << "|1. Tampilkan Ikan          |\n";
cout << "|2. Tambah Ikan           |\n";
cout << "|3. Ubah Ikan             |\n";
cout << "|4. Hapus Ikan            |\n";
cout << "|5. Keluar Program        |\n";
cout << "=====\n";
cout << "Masukan pilihan: ";
cin >> pilihan;
cin.ignore();

```

### D. 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 (`panjang == 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:**

```
switch (pilihan) {
    case 1: // Tampilkan Ikan
        if (panjang == 0) {
            cout << "\nBelum ada ikan yang tersedia." << endl;
        } else {
            cout << "\n=== Daftar Ikan Hias ===" << endl;
            for (int i = 0; i < panjang; i++) {
                cout << "Ikan ke-" << i + 1 << ":" << endl;
                cout << "  Nama: " << namaIkan[i] << endl;
                cout << "  Jenis: " << jenisIkan[i] << endl;
                cout << "  Harga: Rp" << hargaIkan[i] << endl;
                cout << "-----" << endl;
            }
        }
        break;
```

## 2. Case 2 : Menambahkan Ikan Baru

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

**Source code :**

```
case 2: // Tambah Ikan
    if (panjang < MAX_IKAN) {
        cout << "\n=== Tambah Ikan ===" << endl;
        cout << "Masukkan nama ikan: ";
        getline(cin, namaIkan[panjang]);
        cout << "Masukkan jenis ikan: ";
        getline(cin, jenisIkan[panjang]);
        cout << "Masukkan harga ikan: ";
        cin >> hargaIkan[panjang];
        cin.ignore();
        panjang++;
        cout << "Ikan berhasil ditambahkan!" << endl;
    } else {
        cout << "\nKapasitas penuh! Tidak bisa menambah ikan lagi." <<
endl;
```



```
}  
break;
```

### 3. Case 3 : Mengubah Data Ikan

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

Source code :

```
case 3: // Ubah Ikan  
    if (panjang == 0) {  
        cout << "\nBelum ada ikan yang tersedia untuk diubah." <<  
endl;  
    } else {  
        cout << "\n== Ubah Ikan ==" << endl;  
        cout << "Masukkan nomor ikan yang akan diubah: ";  
        cin >> index;  
        cin.ignore(); // Membersihkan buffer  
  
        if (index > 0 && index <= panjang) {  
            cout << "Masukkan nama ikan baru: ";  
            getline(cin, namaIkan[index - 1]);  
            cout << "Masukkan jenis ikan baru: ";  
            getline(cin, jenisIkan[index - 1]);  
            cout << "Masukkan harga ikan baru: ";  
            cin >> hargaIkan[index - 1];  
            cin.ignore(); // Membersihkan buffer  
            cout << "Ikan berhasil diubah!" << endl;  
        } else {  
            cout << "\nNomor ikan tidak valid." << endl;  
        }  
    }  
    break;
```

### 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 :

```

case 4: // Hapus Ikan
    if (panjang == 0) {
        cout << "\nBelum ada ikan yang tersedia untuk dihapus." <<
endl;
    } else {
        cout << "\n== Hapus Ikan ==" << endl;
        cout << "Masukkan nomor ikan yang akan dihapus: ";
        cin >> index;
        cin.ignore(); // Membersihkan buffer

        if (index > 0 && index <= panjang) {
            for (int i = index - 1; i < panjang - 1; i++) {
                namaIkan[i] = namaIkan[i + 1];
                jenisIkan[i] = jenisIkan[i + 1];
                hargaIkan[i] = hargaIkan[i + 1];
            }
            panjang--;
            cout << "Ikan berhasil dihapus!" << endl;
        } else {
            cout << "\nNomor ikan tidak valid." << endl;
        }
    }
    break;

```

## 5. Case 5 : Keluar Dari Program

Menampilkan pesan perpisahan. Menghentikan program dengan return 0.

**Source code:**

```

case 5: // Keluar
    cout << "\nSemangat iciboss!, Terima kasih!" << endl;
    return 0; // Keluar dari program

default:
    cout << "\nPilihan tidak valid. Silakan coba lagi." << endl;
    break;
}

```

## E. Perulangan Pada Program

Program akan terus menampilkan menu dan memproses pilihan pengguna hingga pengguna memilih untuk keluar (pilihan 5).

**Source Code:**

```
int pilihan, index;
while(true) {

}

cout << "\nProgram selesai. Terima kasih!" << endl;

return 0;
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

1. Login salah
2. Login gagal/ lebih dari 3 percobaan
3. Login berhasil
4. Menginput pilihan kurang dari 1 dan lebih dari 5
5. Menginput pilihan 1 (sebelum ada data)
6. Menginput pilihan 1 (setelah ada data)
7. Menginput pilihan 2
8. Menginput pilihan 3
9. Menginput pilihan 4
10. Menginput pilihan 5

```
=== Login Toko Ikan Hias ===  
Masukkan Nama: alamak  
Masukkan NIM: 111
```

#### 4.1.1 Login salah.

```
=== Login Toko Ikan Hias ===  
Masukkan Nama: alamak  
Masukkan NIM: 111  
  
kasi bener dong, sisa percobaan : 2  
=== Login Toko Ikan Hias ===  
Masukkan Nama: cih  
Masukkan NIM: 222  
  
kasi bener dong, sisa percobaan : 1  
=== Login Toko Ikan Hias ===  
Masukkan Nama: siroyo  
Masukkan NIM: 234  
  
kasi bener dong, sisa percobaan : 0  
Yah input user pass nya salah mulu, program berhenti.
```

#### 4.1.2 Login gagal atau lebih dari 3 percobaan.

```

=== Login Toko Ikan Hias ===
Masukkan Nama: Zifa
Masukkan NIM: 025

Login berhasil! Selamat Datang Iciboss Zifa

=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 

```

4.1.3 Login berhasil

```

Masukan pilihan: 6

Pilihan tidak valid. Silakan coba lagi.

=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 

```

4.1.4 Menginput pilihan kurang dari 1 dan lebih dari 5

```

Masukan pilihan: 1

Belum ada ikan yang tersedia.
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====

```

4.1.5 Menginput pilihan 1 (sebelum ada data)

```

Masukan pilihan: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp500000
-----
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====

```

4.1.6 Menginput pilihan 1 (setelah ada data)

```
Masukan pilihan: 2

=== Tambah Ikan ===
Masukkan nama ikan: Piranha
Masukkan jenis ikan: Tawar
Masukkan harga ikan: 500000
Ikan berhasil ditambahkan!
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
```

#### 4.1.7 Menginput pilihan 2

```

Masukan pilihan: 3

=== Ubah Ikan ===
Masukkan nomor ikan yang akan diubah: 1
Masukkan nama ikan baru: Arwana Platinum
Masukkan jenis ikan baru: tawar
Masukkan harga ikan baru: 1000000
Ikan berhasil diubah!
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan          |
|2. Tambah Ikan             |
|3. Ubah Ikan               |
|4. Hapus Ikan              |
|5. Keluar Program          |
=====
Masukan pilihan: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Arwana Platinum
  Jenis: tawar
  Harga: Rp1000000
-----

```

#### 4.1.8 Menginput pilihan 3



```
Masukan pilihan: 4

=== Hapus Ikan ===
Masukkan nomor ikan yang akan dihapus: 1
Ikan berhasil dihapus!
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan          |
|2. Tambah Ikan             |
|3. Ubah Ikan               |
|4. Hapus Ikan              |
|5. Keluar Program          |
=====
Masukan pilihan: 1

Belum ada ikan yang tersedia.
```

#### 4.1.9 Menginput pilihan 4

```
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan          |
|2. Tambah Ikan             |
|3. Ubah Ikan               |
|4. Hapus Ikan              |
|5. Keluar Program          |
=====
Masukan pilihan: 5

Semangat iciboss!, Terima kasih!
```

#### 4.1.10 Menginput pilihan 5

## 4.2 Hasil Output

```
Masukkan Nama: Zifa
Masukkan NIM: 025

Login berhasil! Selamat Datang Icboss Zifa

=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan          |
|2. Tambah Ikan            |
|3. Ubah Ikan              |
|4. Hapus Ikan             |
|5. Keluar Program         |
=====
Masukan pilihan: 1

Belum ada ikan yang tersedia.
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan          |
|2. Tambah Ikan            |
|3. Ubah Ikan              |
|4. Hapus Ikan             |
|5. Keluar Program         |
=====
Masukan pilihan: 2

=== Tambah Ikan ===
Masukkan nama ikan: Arwana
Masukkan jenis ikan: Tawar
Masukkan harga ikan: 1000000
Ikan berhasil ditambahkan!
```

Gambar 4.2.1 login, tampilan menu, pilihan 1 (sebelum memasukan data ikan). Pilihan 2 memasukan data ikan arwana.

```
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan |
|2. Tambah Ikan    |
|3. Ubah Ikan      |
|4. Hapus Ikan     |
|5. Keluar Program |
=====
Masukan pilihan: 2

=== Tambah Ikan ===
Masukkan nama ikan: Piranha
Masukkan jenis ikan: Tawar
Masukkan harga ikan: 500000
Ikan berhasil ditambahkan!
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan |
|2. Tambah Ikan    |
|3. Ubah Ikan      |
|4. Hapus Ikan     |
|5. Keluar Program |
=====
Masukan pilihan: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Arwana
  Jenis: Tawar
  Harga: Rp1000000
-----

Ikan ke-2:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp500000
-----
```

Gambar 4.2.2 dan 4.2.3 pilihan 2 memasukan data ikan piranha, pilihan 1 menampilkan data ikan arwana dan piranha.

```

=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 3

=== Ubah Ikan ===
Masukkan nomor ikan yang akan diubah: 1
Masukkan nama ikan baru: Koi Legends
Masukkan jenis ikan baru: Tawar
Masukkan harga ikan baru: 2500000
Ikan berhasil diubah!
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi Legends
  Jenis: Tawar
  Harga: Rp2500000
-----

```

Gambar 4.2.4 Pilihan 3 mengubah data ikan index 1 jadi data ikan Koi legends, pilihan 1 melihat data yang diubah.

```

Ikan ke-2:
  Nama: Piranha
  Jenis: Tawar
  Harga: Rp500000
-----
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 4

=== Hapus Ikan ===
Masukkan nomor ikan yang akan dihapus: 2
Ikan berhasil dihapus!
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi Legends
  Jenis: Tawar

```

Gambar 4.2.5 tampilan data pilihan 1, pilihan 4 menghapus data ikan index 2, pilihan 1 untuk melihat data ikan yang dihapus.

```
Harga: Rp2500000
-----
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 6

Pilihan tidak valid. Silakan coba lagi.
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 11

Pilihan tidak valid. Silakan coba lagi.
=====
=== Program Manajemen Ikan Hias ===
=====
|1. Tampilkan Ikan      |
|2. Tambah Ikan         |
|3. Ubah Ikan           |
|4. Hapus Ikan          |
|5. Keluar Program      |
=====
Masukan pilihan: 5

Semangat iciboss!, Terima kasih!
PS D:\praktikum-apl>
```

Gambar 4.2.6 dan 4.2.7 menginput pilihan 6 dan 11 untuk mengecek apakah hasil sesuai, pilihan 5 dan outputnya.

## 5. Langkah-langkah Git.

```
PS D:\praktikum-apl> git init
Reinitialized existing Git repository in D:/praktikum-apl/.git/
PS D:\praktikum-apl> git add .
PS D:\praktikum-apl> git commit -m "Finish Post-test 2"
[main 4af90e3] Finish Post-test 2
 3 files changed, 153 insertions(+)
 create mode 100644 post-test/post-test-apl-2/2409106025-AkhmadZifaAlFatih-PT-2.cpp
 create mode 100644 post-test/post-test-apl-2/2409106025-AkhmadZifaAlFatih-PT-2.exe
 create mode 100644 post-test/post-test-apl-2/2409106025-AkhmadZifaAlFatih-PT-2.pdf
PS D:\praktikum-apl> git remote add origin https://github.com/Iziypzyy/praktikum-apl.git
error: remote origin already exists.
PS D:\praktikum-apl> git branch -m main
git: 'branch' is not a git command. See 'git --help'.

The most similar command is
    branch
PS D:\praktikum-apl> git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (6/6), done.
Writing objects: 100% (7/7), 1.47 MiB | 968.00 KiB/s, done.
Total 7 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/Iziypzyy/praktikum-apl.git
   0e5e0cd..4af90e3  main -> main
branch 'main' set up to track 'origin/main'.
PS D:\praktikum-apl> █
```

Gambar 5.1 hasil git di terminal

### 1. git init

Perintah ini digunakan untuk menginisialisasi repository Git di dalam folder proyek. Setelah menjalankan perintah ini, Git akan membuat folder git tersembunyi di dalam proyek yang akan menyimpan semua informasi versi.

### 2. git add.

Perintah ini digunakan untuk menambahkan semua perubahan ke dalam staging area, sehingga siap untuk dikomit.

3. `git commit -m "Pesan"`

Menyimpan perubahan dengan pesan commit.

4. `git remote add origin git@github.com:iziypzyy/praktikum-apl.git` Menghubungkan repository lokal ke GitHub.

5. `git branch -M main`

Mengubah nama branch utama menjadi main. Secara default, branch utama di git dulu bernama master.

6. `git push-u origin main`

mengunggah (push) kode dari repository lokal ke repository remote (GitHub) dan menetapkan branch default.