LAPORAN PRAKTIKUM

POSTTEST (III)

ALGORITMA PEMROGRAMAN LANJUT

Disusun oleh:

Akhmad Zifa Al-Fatih (2409106025)

Kelas (A2‘24)

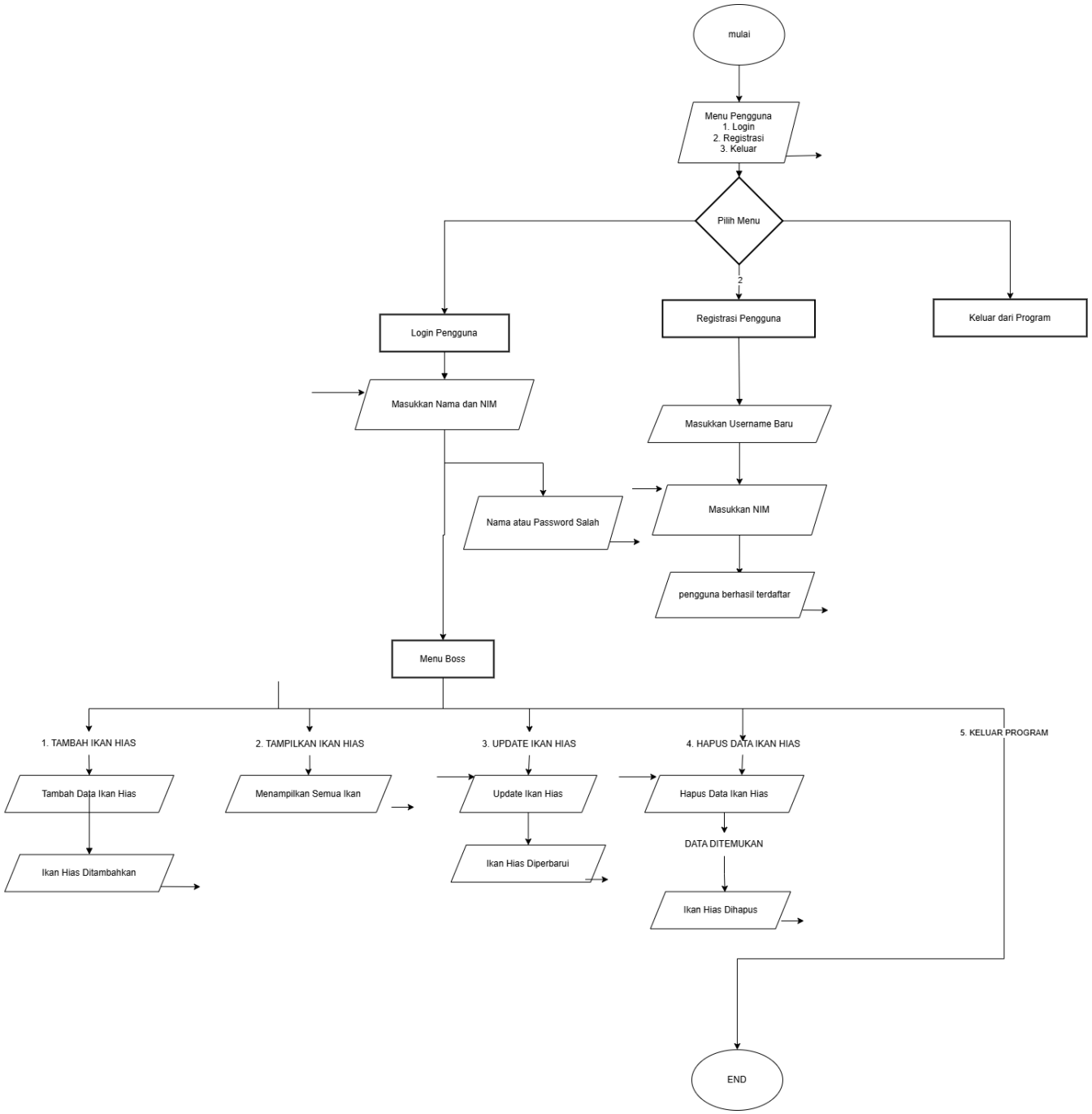
PROGRAM STUDI INFORMATIKA

UNIVERSITAS MULAWARMAN

SAMARINDA

2025

1. Flowchart


mulai
Menu Pengguna
1. Login
2. Registrasi
3. Keluar
Pilih Menu
1
2
3
Login Pengguna
Registrasi Pengguna
Keluar dari Program
Masukkan Nama dan NIM
Masukkan Username Baru
Nama atau Password Salah
Masukkan NIM
pengguna berhasil terdaftar
Menu Boss
1. TAMBAH IKAN HIAS
2. TAMPILKAN IKAN HIAS
3. UPDATE IKAN HIAS
4. HAPUS DATA IKAN HIAS
5. KELUAR PROGRAM
Tambah Data Ikan Hias
Menampilkan Semua Ikan
Update Ikan Hias
Hapus Data Ikan Hias
Ikan Hias Ditambahkan
Ikan Hias Diperbarui
DATA DITEMUKAN
Ikan Hias Dihapus
END

2. Analisis Program

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

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 // Ukuran maksimum array pengguna

// Struct untuk menyimpan data ikan
struct DataIkan {
    string nama;
    string jenis;
    int harga;
};

// Struct untuk menyimpan data pengguna
struct DataPengguna {
    string nama;
    string nim;
    DataIkan ikan[MAX_IKAN]; // Setiap pengguna memiliki array ikan
    int jumlahIkan = 0; // Jumlah ikan yang dimiliki pengguna
};


// Variabel array untuk menyimpan data pengguna
DataPengguna dataPengguna[MAX_PENGGUNA];
int jumlahPengguna = 0; // Jumlah pengguna saat ini
```

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:

```
int main() {
    // Variabel untuk login
    string nama, nim;
    int percobaan = 0;

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

        if (pilihan == 1) { // Login
            while (percobaan < 3) {
                cout << "Masukkan Nama: ";
                getline(cin, nama);
                cout << "Masukkan NIM: ";
                getline(cin, nim);
                cout << endl << endl;

                bool loginBerhasil = false;
                int indexPengguna = -1;

                // Cek apakah nama dan NIM cocok
                for (int i = 0; i < jumlahPengguna; i++) {
                    if (dataPengguna[i].nama == nama && dataPengguna[i].nim ==
nim) {
                        loginBerhasil = true;
                        indexPengguna = i; // Simpan index pengguna yang login
                        break;
                    }
                }

                if (loginBerhasil) {
                    cout << "Login berhasil! Selamat Datang " << nama << endl <<
endl;


                } else {
                    percobaan++;
                    cout << "Login gagal. Sisa percobaan: " << 3 - percobaan <<
endl;
                    break;
                }
            }
```

```
        } else if (pilihan == 2) { // Register
            if (jumlahPengguna >= MAX_PENGGUNA) {
                cout << "Batas pengguna telah tercapai. Tidak bisa mendaftar
lagi." << endl;
            } else {
                cout << "\n=== Register Pengguna ===" << endl;
                cout << "Masukkan nama: ";
                getline(cin, dataPengguna[jumlahPengguna].nama);
                cout << "Masukkan NIM: ";
                getline(cin, dataPengguna[jumlahPengguna].nim);
                jumlahPengguna++;
                cout << "Pengguna berhasil terdaftar!" << endl;
            }
        } else if (pilihan == 3) { // Exit
            cout << "\nTerima kasih! Program selesai." << endl;
            return 0;
        } else {
            cout << "\nPilihan tidak valid. Silakan coba lagi." << 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:

```
 // Menu manajemen ikan setelah login berhasil

while (true) {
     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 << "Pilih menu: ";
        int pilihanMenu;
        cin >> pilihanMenu;
        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 (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:

```
switch (pilihanMenu) {
    case 1: // Tampilkan Ikan
        if (dataPengguna[indexPengguna].jumlahIkan == 0) {
            cout << "\nBelum ada ikan yang tersedia." << endl;
        } else {
            cout << "\n=== Daftar Ikan Hias ===" << endl;
            for (int i = 0; i < dataPengguna[indexPengguna].jumlahIkan; i++) {
                cout << "Ikan ke-" << i + 1 << ":" << endl;
                cout << "  Nama: " << dataPengguna[indexPengguna].ikan[i].nama <<
endl;
                cout << "  Jenis: " << dataPengguna[indexPengguna].ikan[i].jenis
<< endl;
                cout << "  Harga: Rp" <<
dataPengguna[indexPengguna].ikan[i].harga << endl;
                cout << "-------------------------" << endl;
            }
         }
         break;
```

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 :

```
case 2: // Tambah Ikan
    if (dataPengguna[indexPengguna].jumlahIkan < MAX_IKAN) {
        cout << "\n=== Tambah Ikan ===" << endl;
        cout << "Masukkan nama ikan: ";
        getline(cin,
dataPengguna[indexPengguna].ikan[dataPengguna[indexPengguna].jumlahIkan].nama);
        cout << "Masukkan jenis ikan: ";
        getline(cin,
dataPengguna[indexPengguna].ikan[dataPengguna[indexPengguna].jumlahIkan].jenis);
        cout << "Masukkan harga ikan: ";
        cin >>
dataPengguna[indexPengguna].ikan[dataPengguna[indexPengguna].jumlahIkan].harga;
        cin.ignore();
        dataPengguna[indexPengguna].jumlahIkan++;
        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 (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 :

```
case 3: // Ubah Ikan
    if (dataPengguna[indexPengguna].jumlahIkan == 0) {
        cout << "\nBelum ada ikan yang tersedia untuk diubah." << endl;
    } else {
        cout << "\n=== Ubah Ikan ===" << endl;
        cout << "Masukkan nomor ikan yang akan diubah: ";
        int index;
        cin >> index;
        cin.ignore();
```

```
        if (index > 0 && index <= dataPengguna[indexPengguna].jumlahIkan) {
            cout << "Masukkan nama ikan baru: ";
            getline(cin, dataPengguna[indexPengguna].ikan[index -1].nama);
            cout << "Masukkan jenis ikan baru: ";
            getline(cin, dataPengguna[indexPengguna].ikan[index -1].jenis);
            cout << "Masukkan harga ikan baru: ";
            cin >> dataPengguna[indexPengguna].ikan[index - 1].harga;
            cin.ignore();
            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 (dataPengguna[indexPengguna].jumlahIkan == 0) {
        cout << "\nBelum ada ikan yang tersedia untuk dihapus." << endl;
    } else {
        cout << "\n=== Hapus Ikan ===" << endl;
        cout << "Masukkan nomor ikan yang akan dihapus: ";
        int index;
        cin >> index;
        cin.ignore();

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

5. Case 5 : Keluar Dari Program

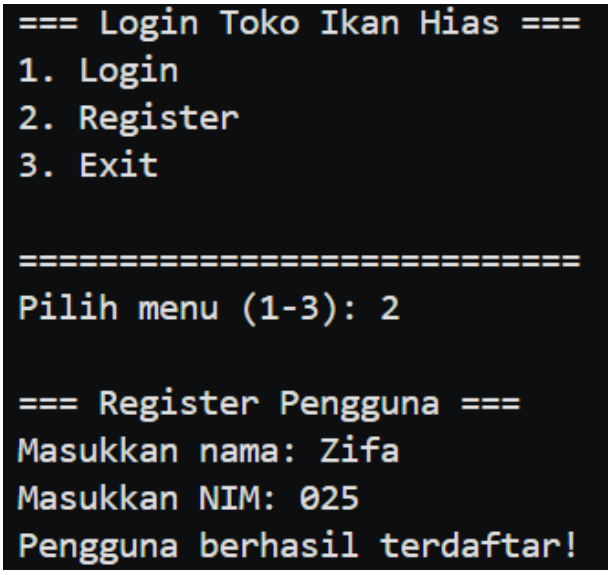
Menampilkan pesan perpisahan. Menghentikan program dengan return 0.

Source code:

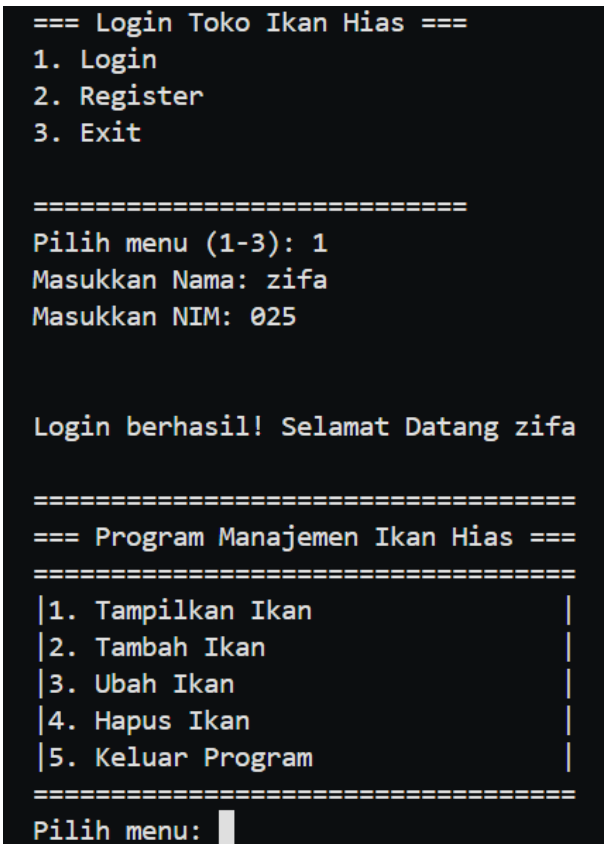
```
case 5: // Keluar
    cout << "\nTerima kasih! Kembali ke menu utama." << endl;
    exit

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

4. Uji Coba dan Hasil Output

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



Gambar 4.1 regstrasi di menu login



Gambar 4.2 Login dan mengeluarkan menu program jika berhasil

```
Pilih menu: 2

=== Tambah Ikan ===
Masukkan nama ikan: Koi legend
Masukkan jenis ikan: Tawar
Masukkan harga ikan: 1000000
Ikan berhasil ditambahkan!
==================================
=== Program Manajemen Ikan Hias ===
==================================
|1. Tampilkan Ikan                |
|2. Tambah Ikan                   |
|3. Ubah Ikan                     |
|4. Hapus Ikan                    |
|5. Keluar Program                |
==================================
Pilih menu:
```

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
------------------------
```

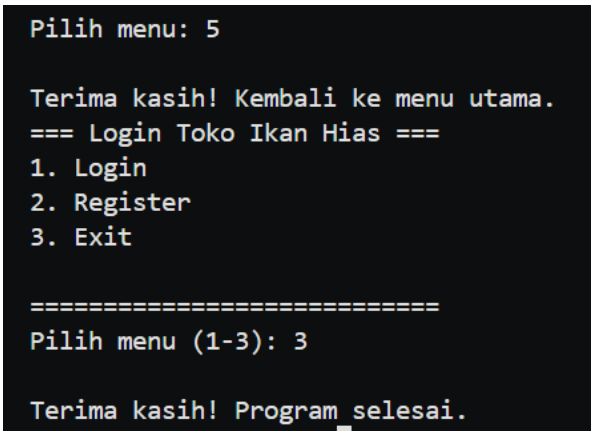
Gambar 4.4 Menampilkan data Ikan pada pilihan 1

```
=== Ubah Ikan ===
Masukkan nomor ikan yang akan diubah: 2
Masukkan nama ikan baru: Arwana
Masukkan jenis ikan baru: Tawar
Masukkan harga ikan baru: 2000000
Ikan berhasil diubah!
==================================
=== Program Manajemen Ikan Hias ===
==================================
|1. Tampilkan Ikan                |
|2. Tambah Ikan                   |
|3. Ubah Ikan                     |
|4. Hapus Ikan                    |
|5. Keluar Program                |
==================================
Pilih menu: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Koi legend
  Jenis: Tawar
  Harga: Rp1000000
------------------------
Ikan ke-2:
  Nama: Arwana
  Jenis: Tawar
  Harga: Rp2000000
------------------------
```

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

```
Pilih menu: 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                |
==================================
Pilih menu: 1

=== Daftar Ikan Hias ===
Ikan ke-1:
  Nama: Arwana
  Jenis: Tawar
  Harga: Rp2000000
------------------------
```

Gambar 4.6 Menghapus ikan ke 1 pada pilihan 4

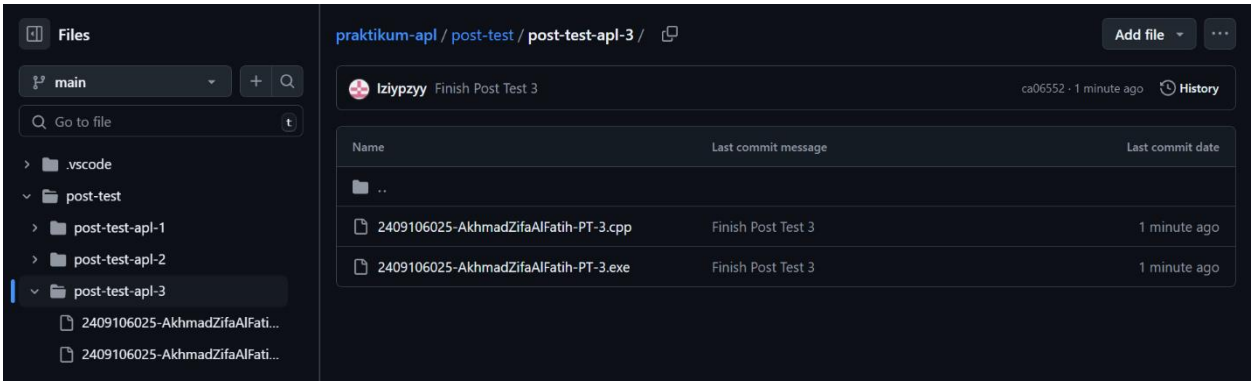


Gambar 4.7 keluar pada pilihan 5 dan mengembalikan ke menu login serta memilih opsi 3

5. Langkah-Langkah Git pada VSCode

```
PS D:\
PS D:\                Show Commit (ctrl + click)  it -m "Finish Post Test 3"
[main ca06552] Finish Post Test 3
 4 files changed, 226 insertions(+)
 create mode 100644 .vscode/tasks.json
 create mode 100644 post-test/post-test-apl-3/2409106025-AkhmadZifaAlFatih-PT-3.cpp
 create mode 100644 post-test/post-test-apl-3/2409106025-AkhmadZifaAlFatih-PT-3.exe
PS D:\praktikum-apl> git push origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (10/10), 695.04 KiB | 3.55 MiB/s, done.
Total 10 (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/Iziypzyy/praktikum-apl.git
   5cf6f4c..ca06552  main -> main
PS D:\praktikum-apl>
```

Gambar 5.1 Langkah git di terminal dari git add,git commit -m "Finish Post Test 3" ,git push origin main



Gambar 5.2 Memeriksa pada github