**LAPORAN PRAKTIKUM POSTTEST 3**

**ALGORITMA PEMROGRAMAN LANJUT**



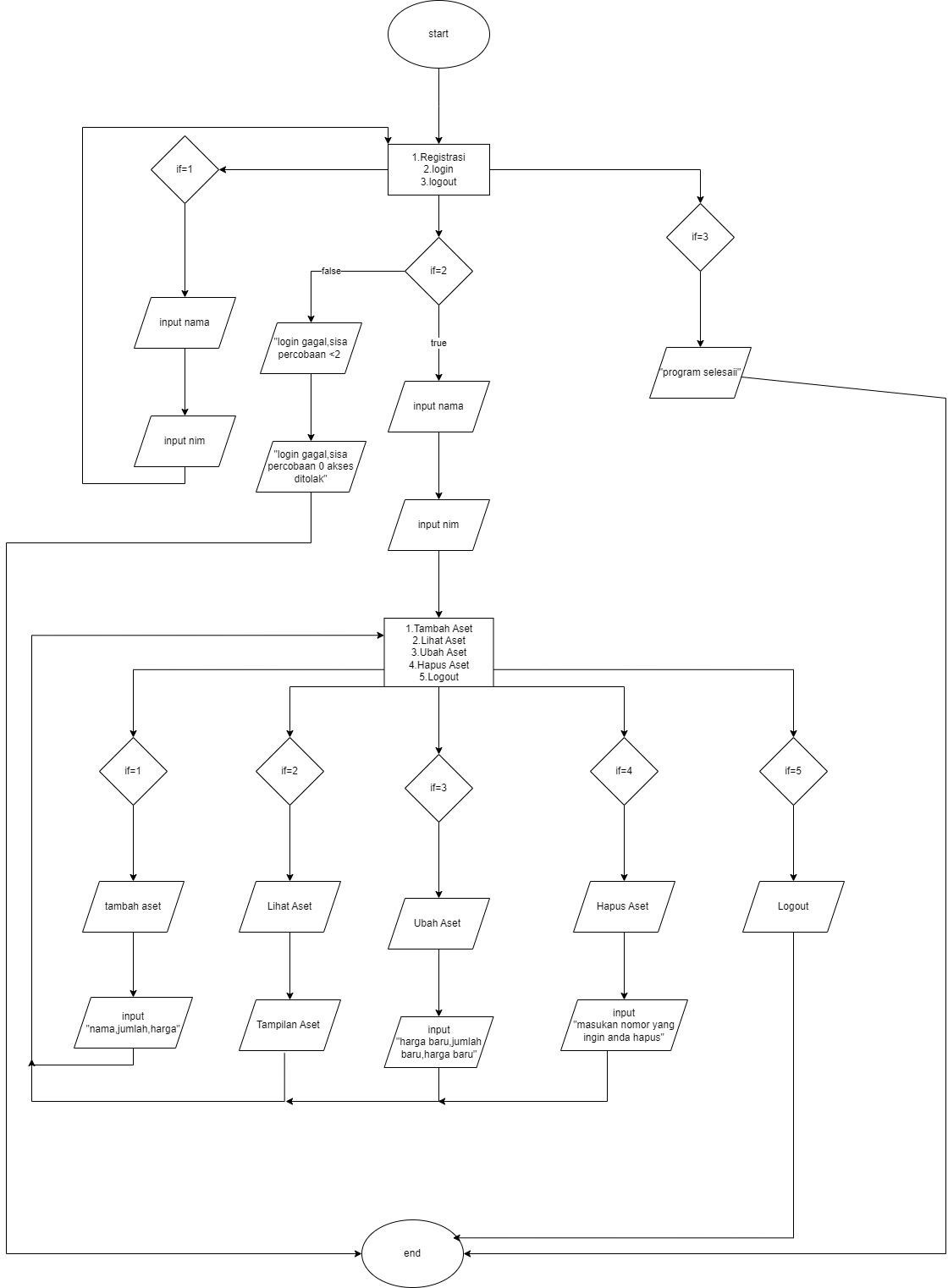
**Disusun oleh:**

**Gilbert Finsen Zakaria Lian (2409106110) Kelas (C‘2)**

**PROGRAM STUDI INFORMATIKA UNIVERSITAS MULAWARMAN SAMARINDA**

**2025**

# Flowchart



1. **Analisis Program**

*(Tuliskan secara ringkas tujuan dan fungsi/manfaat utama dari program yang dibuat.)*

Tujuan P**rogram**

Program ini bertujuan untuk mengelola data aset dengan fitur multiuser menggunakan konsep array of struct dan nested struct. Setiap user dapat memiliki daftar aset pribadi yang dapat dikelola secara mandiri.

# Fungsi dan Manfaat Utama

1. **Registrasi dan Login**: Memungkinkan pengguna untuk membuat akun baru dan masuk dengan validasi hingga 3 kali percobaan.
2. **Manajemen Aset**: Pengguna dapat menambah, melihat, mengubah, dan menghapus data aset yang mereka miliki.
3. **Penggunaan Struct**: Mengorganisir data aset dan pengguna secara efisien menggunakan nested struct.
4. **Data Terpisah per User**: Setiap user memiliki data aset yang tersimpan secara terpisah, menjaga keamanan dan privasi data.
5. **Antarmuka Sederhana**: Menyediakan menu navigasi yang mudah dipahami untuk mengelola aset.

# Source Code

* 1. **Header File dan Pengaturan Dasar**

Source Code:

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

* 1. **Struct**

struct Aset {

string nama; int jumlah; double harga;

**};**

struct User {

string nama; string nim;

Aset daftarAset[100];

int jumlahAsetTersimpan = 0;

**};**

* 1. **Variabel Global**

User users[100]; int totalUsers = 0;

* 1. **Fungsi Registrasi**

int registrasi() {

if (totalUsers < 100) { cout << "Nama: ";

getline(cin >> ws, users[totalUsers].nama); cout << "NIM: ";

getline(cin, users[totalUsers].nim);

cout << "Registrasi berhasil. Silakan login." << endl; totalUsers++;

return 1;

} else {

cout << "User telah penuh." << endl; return 0;

**}**

**}**

* 1. **Fungsi Login**

int login() {

string nama, nim; int percobaan = 0;

while (percobaan < 3) { cout << "Nama: ";

getline(cin >> ws, nama); cout << "NIM: "; getline(cin, nim);

for (int i = 0; i < totalUsers; i++) {

if (users[i].nama == nama && users[i].nim == nim) { cout << "Login berhasil!" << endl;

return i;

**}**

**}**

percobaan++;

cout << "Login gagal. Sisa percobaan: " << 3 - percobaan <<

endl;

**}**

cout << "Akses ditolak." << endl; return -1;

**}**

* 1. **Fungsi Menu Aset**

int menuAset(User &user)

* 1. **Tambah Aset**

case 1: {

if (user.jumlahAsetTersimpan < 100) { cout << "Nama Aset: ";

getline(cin, user.daftarAset[user.jumlahAsetTersimpan].nama); cout << "Jumlah: ";

cin >> user.daftarAset[user.jumlahAsetTersimpan].jumlah; cout << "Harga: ";

cin >> user.daftarAset[user.jumlahAsetTersimpan].harga; user.jumlahAsetTersimpan++;

cout << "Aset berhasil ditambahkan." << endl;

} else {

cout << "Daftar aset penuh." << endl;

**}**

break;

**}**

* 1. **Lihat Daftar Aset**

case 2: {

if (user.jumlahAsetTersimpan > 0) { cout << "\nDaftar Aset:\n";

cout << setw(5) << "No" << setw(20) << "Nama Aset" << setw(10)

<< "Jumlah" << setw(15) << "Harga" << endl;

cout << " "

<< endl;

for (int i = 0; i < user.jumlahAsetTersimpan; i++) { cout << setw(5) << i + 1 << setw(20) <<

user.daftarAset[i].nama << setw(10) << user.daftarAset[i].jumlah << setw(15) << user.daftarAset[i].harga << endl;

**}**

} else {

cout << "Daftar aset kosong." << endl;

**}**

break;

**}**

* 1. **Ubah Aset**

case 3: {

int nomor;

cout << "Masukkan nomor aset yang ingin diubah: "; cin >> nomor;

if (nomor > 0 && nomor <= user.jumlahAsetTersimpan) { nomor--;

cout << "Nama Aset Baru: ";

getline(cin >> ws, user.daftarAset[nomor].nama); cout << "Jumlah Baru: ";

cin >> user.daftarAset[nomor].jumlah; cout << "Harga Baru: ";

cin >> user.daftarAset[nomor].harga; cout << "Aset berhasil diubah." << endl;

} else {

cout << "Nomor aset tidak valid." << endl;

**}**

break;

**}**

* 1. **Hapus Aset**

case 4: {

int nomor;

cout << "Masukkan nomor aset yang ingin dihapus: "; cin >> nomor;

if (nomor > 0 && nomor <= user.jumlahAsetTersimpan) { nomor--;

for (int i = nomor; i < user.jumlahAsetTersimpan - 1; i++) { user.daftarAset[i] = user.daftarAset[i + 1];

**}**

user.jumlahAsetTersimpan--;

cout << "Aset berhasil dihapus." << endl;

} else {

cout << "Nomor aset tidak valid." << endl;

**}**

break;

**}**

* 1. **Fungsi Main**

int main() {

int pilihan; do {

cout << "\n1. Registrasi\n2. Login\n3. Keluar\nPilihan: "; cin >> pilihan;

cin.ignore();

if (pilihan == 1) registrasi(); else if (pilihan == 2) {

int userIndex = login();

if (userIndex != -1) menuAset(users[userIndex]);

**}**

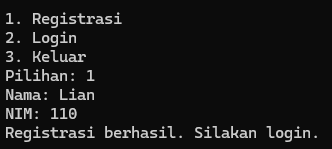
} while (pilihan != 3);

cout << "Program selesai." << endl; return 0;

**}**

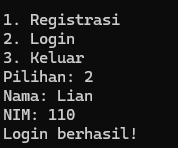
# Uji Coba dan Hasil Output

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

* 1. **Halaman Untuk Registrasi**

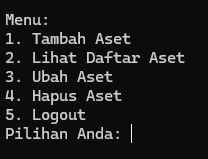
**Gambar 4.1**

* 1. **Halaman Login**



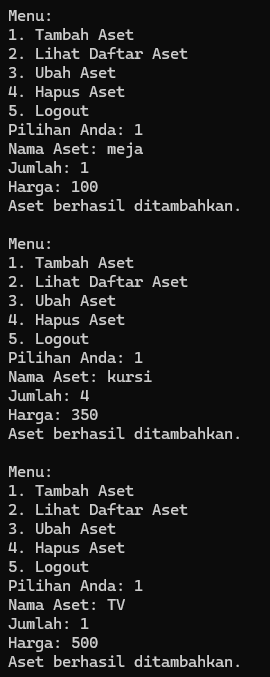
**Gambar 4.2**

* 1. **Pilihan Menu**



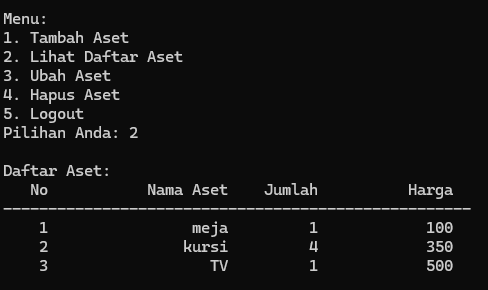
**Gambar 4.3**

* 1. **Tambah Aset**



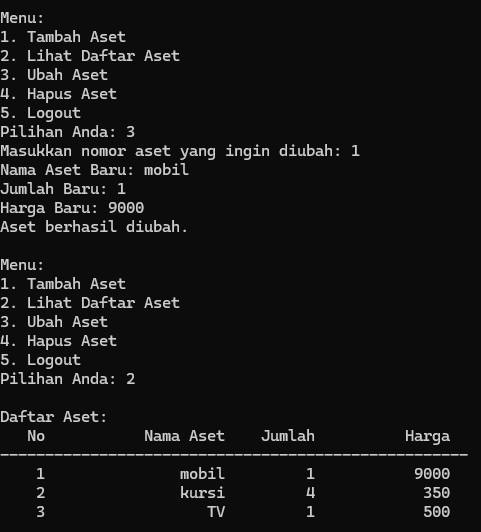
**Gambar 4.4**

* 1. **Lihat Aset**



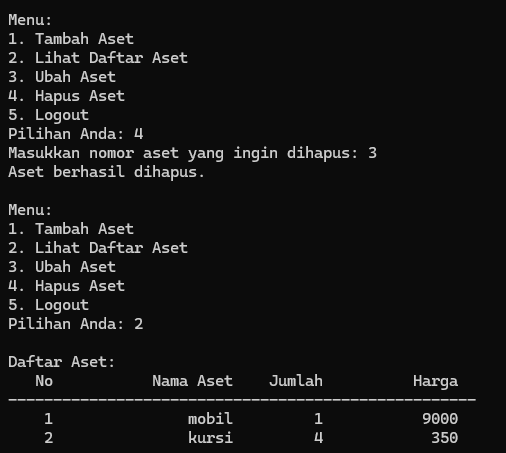
**Gambar 4.5**

* 1. **Ubah Aset**



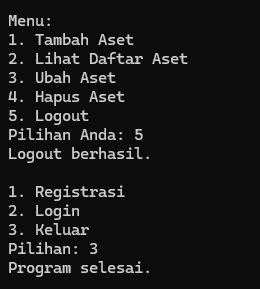
**Gambar 4.6**

* 1. **Hapus Aset**



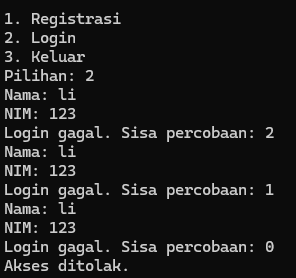
Gambar 4.7

* 1. **Logout dan keluar program**



Gambar 4.8

* 1. **Akses Di Tolak Saat Gagal Login**



Gambar 4.9

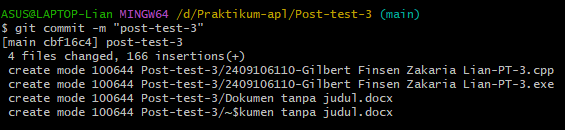
Gambar 4.1 Contoh

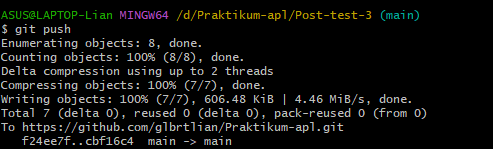
Gambar <nomor bab>.[index] <judul-gambar>

# Langkah-Langkah Git

**Git add**



**Git commit –m**

**Git push**