

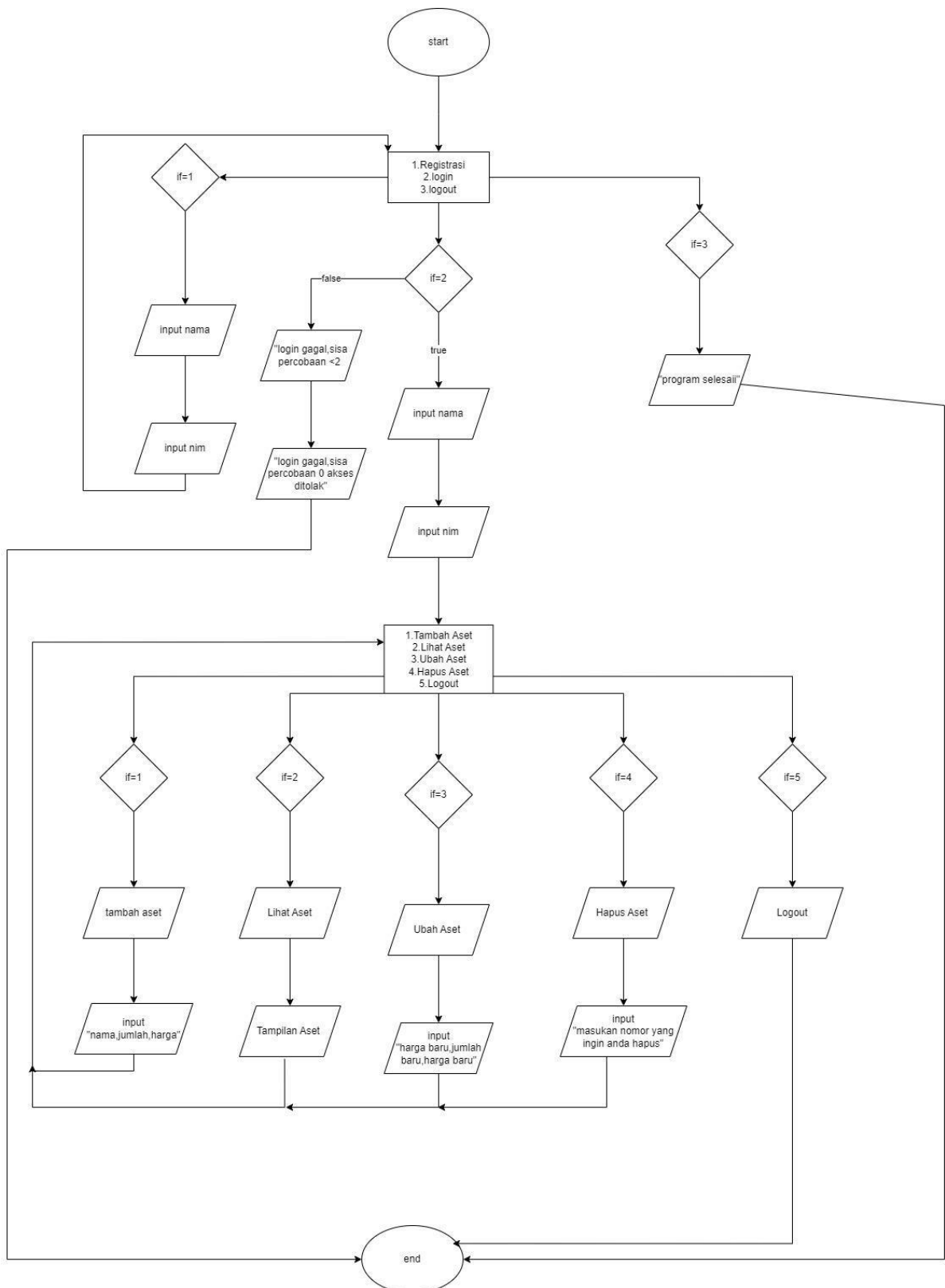
LAPORAN PRAKTIKUM
POSTTEST 5
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Gilbert Finsen Zakaria Lian (2409106110) Kelas
(C'2)

PROGRAM STUDI
INFORMATIKA UNIVERSITAS
MULAWARMAN SAMARINDA
2025

1. Flowchart



2. Analisis Program

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

Tujuan Program

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.

3. Source Code

Header & Namespace

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

- `#include <iostream>` → untuk input/output (cin, cout).
 - `#include <string>` → untuk menggunakan tipe data string.
 - `#include <iomanip>` → untuk manipulasi tampilan output (contoh: setw).
 - `using namespace std;` → supaya tidak perlu menulis `std :` terus.
-

Struct Definition

```
struct Aset {
    string nama;
    int jumlah;
    double harga;
};
```

- Struct Aset menyimpan informasi satu aset: nama, jumlah, dan harga.

```
struct User {
    string nama;
    string nim;
    Aset daftarAset[100];
    int jumlahAsetTersimpan = 0;
};
```

- Struct User menyimpan nama dan NIM.
- daftarAset [100] adalah array dari Aset, maksimal 100 aset.
- jumlahAsetTersimpan melacak jumlah aset user saat ini.

Array Global

```
User users[100];
int totalUsers = 0;
```

- users [100]: array untuk menyimpan maksimal 100 user.
- totalUsers: melacak jumlah user yang sudah mendaftar.

Function Overloading

```
void tampilkanAset(const Aset &aset, int index) {
... }
void tampilkanAset(const Aset &aset) { ... }
```

- Dua fungsi dengan nama sama (tampilkanAset) tapi parameter berbeda.
- Menampilkan informasi aset secara berformat (pakai setw).

Fungsi Rekursif Total Nilai Aset

```
double hitungTotalNilaiAset(const Aset daftar[], int jumlah) {
    if (jumlah == 0) return 0;
    return (daftar[jumlah - 1].jumlah * daftar[jumlah -
1].harga) +
        hitungTotalNilaiAset(daftar, jumlah - 1);
}
```

- Menghitung total nilai aset dengan rekursi:

Basis: jika jumlah == 0, return 0.

Rekursi: ambil 1 aset terakhir, lalu tambah dengan hasil rekursif dari sisa aset.

Registrasi

```
void registrasi(User users[], int &totalUsers) {  
... }
```

- Menginput data user baru dan menambahkannya ke array users.
- Gunakan `cin >> ws` supaya input string bisa menerima spasi.

Login

```
int login(User users[], int totalUsers) {  
... }
```

- Menerima input nama dan NIM.
 - Cek ke semua user yang terdaftar.
 - Return indeks user jika ditemukan.
 - Maksimal 3 kali percobaan login.
-

CRUD Aset

```
void tambahAset(User &user) {  
... }
```

- Input nama, jumlah, harga aset.
- Simpan ke user.daftarAset.
- Tambah jumlahAsetTersimpan.

```
void lihatAset(User &user) {
... }
```

- Tampilkan seluruh aset user.
- Tampilkan total nilai aset (panggil fungsi rekursif).

```
void ubahAset(User &user) { ...
}
```

- Input nomor aset.
- Ubah nama, jumlah, harga aset.

```
void hapusAset(User &user) {
... }
```

- Hapus aset dengan cara geser array ke atas.
- Kurangi jumlahAsetTersimpan.

Menu Aset (User sudah login)

```
void menuAset(User &user) {  
    ... }
```

- Menu pilihan:
 1. Tambah Aset
 2. Lihat Aset
 3. Ubah Aset
 4. Hapus Aset
 5. Logout
- Gunakan do-while agar terus berjalan sampai user pilih logout.

Main Function

```
int main() {  
    int pilihan;  
    do {  
        cout << "\n1. Registrasi\n2. Login\n3. Keluar\nPilihan: ";  
        cin >> pilihan;  
        cin.ignore();  
  
        if (pilihan == 1) registrasi(users, totalUsers);  
        else if (pilihan == 2) {  
            int userIndex = login(users, totalUsers);  
            if (userIndex != -1) menuAset(users[userIndex]);  
        }  
    } while (pilihan != 3);  
  
    cout << "Program selesai." << endl;  
    return 0;  
}
```


- Menampilkan menu utama (Registrasi, Login, Keluar).
 - Jika login berhasil, lanjut ke menu aset.
 - Loop utama akan terus berjalan hingga user memilih keluar.
-

Fungsi Pointer yang Ditambahkan

```
void inisialisasiAset(Aset *aset, string nama, int jumlah, double harga) {  
    aset->nama = nama;  
    aset->jumlah = jumlah;  
    aset->harga = harga;  
}
```

- Fungsi pointer (*aset) digunakan untuk mengisi data ke struct dari alamat memori.

```
void tampilkanDetailAset(Aset *aset) {  
    cout << "Detail Aset:\n";  
    cout << "Nama: " << aset->nama << endl;  
    cout << "Jumlah: " << aset->jumlah << endl;  
    cout << "Harga: " << aset->harga << endl;  
}
```

- Menampilkan aset dengan akses pointer (->).

```
void tambahJumlah(int *jumlah, int tambahan) {  
    *jumlah += tambahan;  
}
```

- Menggunakan parameter `int*` (address-of) dan operasi dereference untuk menambah nilai jumlah aset.

4. Uji Coba dan Hasil Output

```
1. Registrasi
2. Login
3. Keluar
Pilihan: 2
Nama: Lian
NIM: 110
Login berhasil!
```

4.1 Halaman Registrasi

```
1. Registrasi
2. Login
3. Keluar
Pilihan: 1
Nama: Lian
NIM: 110
Registrasi berhasil. Silakan login.
```

Gambar 4.1

4.2 Halaman Login

```
1. Registrasi
2. Login
3. Keluar
Pilihan: 2
Nama: Lian
NIM: 110
Login berhasil!
```

Gambar 4.2

4.3 Pilihan Menu

```
Menu:  
1. Tambah Aset  
2. Lihat Daftar Aset  
3. Ubah Aset  
4. Hapus Aset  
5. Logout  
Pilihan Anda: |
```

Gambar 4.3

4.4 Tambah Aset

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 1
Nama Aset: meja
Jumlah: 1
Harga: 100
Aset berhasil ditambahkan.
```

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 1
Nama Aset: kursi
Jumlah: 4
Harga: 350
Aset berhasil ditambahkan.
```

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 1
Nama Aset: TV
Jumlah: 1
Harga: 500
Aset berhasil ditambahkan.
```

Gambar 4.4

4.5 Lihat Aset

Menu:

1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout

Pilihan Anda: 2

Daftar Aset:

No	Nama Aset	Jumlah	Harga
1	meja	1	100
2	kursi	4	350
3	TV	1	500

Gambar 4.5

4.6 Ubah Aset

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 3
Masukkan nomor aset yang ingin diubah: 1
Nama Aset Baru: mobil
Jumlah Baru: 1
Harga Baru: 9000
Aset berhasil diubah.

Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 2

Daftar Aset:
  No      Nama Aset      Jumlah      Harga
-----
  1      mobil          1          9000
  2      kursi           4          350
  3         TV           1          500
```

Gambar 4.6

4.7 Hapus Aset

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 4
Masukkan nomor aset yang ingin dihapus: 3
Aset berhasil dihapus.
```

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 2
```

Daftar Aset:

No	Nama Aset	Jumlah	Harga
1	mobil	1	9000
2	kursi	4	350

Gambar 4.7

4.8 Logout dan keluar program

```
Menu:
1. Tambah Aset
2. Lihat Daftar Aset
3. Ubah Aset
4. Hapus Aset
5. Logout
Pilihan Anda: 5
Logout berhasil.

1. Registrasi
2. Login
3. Keluar
Pilihan: 3
Program selesai.
```

Gambar 4.8

4.9 Akses Di Tolak Saat Gagal Login

```
1. Registrasi
2. Login
3. Keluar
Pilihan: 2
Nama: li
NIM: 123
Login gagal. Sisa percobaan: 2
Nama: li
NIM: 123
Login gagal. Sisa percobaan: 1
Nama: li
NIM: 123
Login gagal. Sisa percobaan: 0
Akses ditolak.
```

Gambar 4.9

Gambar 4.1 Contoh

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

5. Langkah-Langkah Git

Git add

```
ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl/Post-test-4 (main)
$ git add .
```

Git commit -m

```
ASUS@LAPTOP-Lian MINGW64 /d/Praktikum-apl/Post-test-4 (main)
$ git commit -m "post-test-4"
[main 8596355] post-test-4
7 files changed, 181 insertions(+)
create mode 100644 Post-test-4/2409106110-Gilbert Finsen Zakaria Lian-PT-3.docx
create mode 100644 Post-test-4/2409106110_GilbertFinsenZakariaLian_PT4.cpp
create mode 100644 Post-test-4/2409106110_GilbertFinsenZakariaLian_PT4.exe
create mode 100644 Post-test-4/Dokumen tanpa judul.docx
create mode 100644 Post-test-4/~$09106110-Gilbert Finsen Zakaria Lian-PT-3.docx
create mode 100644 Post-test-4/~$kumen tanpa judul.docx
create mode 100644 Post-test-4/~WRL0511.tmp
```

Git push

```
$ git push
Enumerating objects: 10, done.
Counting objects: 100% (10/10), done.
Delta compression using up to 2 threads
Compressing objects: 100% (9/9), done.
Writing objects: 100% (9/9), 897.59 KiB | 2.79 MiB/s, done.
Total 9 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/glbtrlan/Praktikum-apl.git
6ff6d11..8596355 main -> main
```