

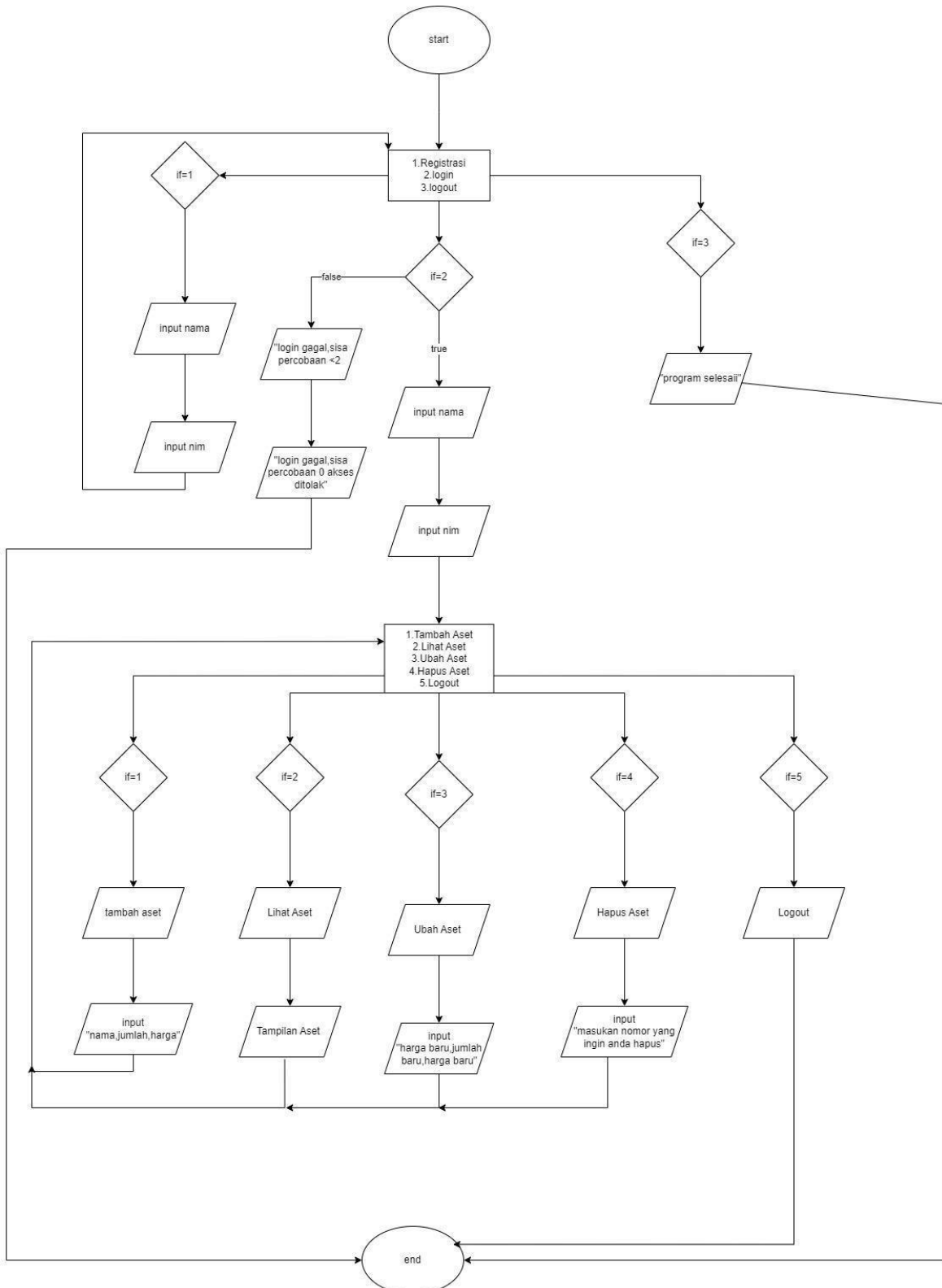
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**



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

- Struct Aset menyimpan informasi tiap aset: nama (string), jumlah (int), harga (double).

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

- Struct Users menyimpan data pengguna: nama, NIM, array aset sebanyak 100, dan penghitung jumlah aset yang dimiliki user.

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

- Array users menampung maksimal 100 pengguna.
- totalUsers menghitung jumlah user yang terdaftar.

### Fungsi Overload Tampilkan Aset

```
void tampilkanAset(const Aset &aset, int index) {
    cout << setw(5) << index + 1 << setw(20) << aset.nama
        << setw(10) << aset.jumlah << setw(15) << aset.harga << endl;
}
```

- Fungsi menampilkan 1 aset dengan format tabel dan nomor urut (index + 1)

```
void tampilkanAset(const Aset &aset) {
    cout << "Aset: " << aset.nama << ", Jumlah: " << aset.jumlah
        << ", Harga: " << aset.harga << endl;
}
```

- Overload untuk menampilkan 1 aset tanpa format tabel.

### Fungsi Rekursif Hitung 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);
}
```

- Fungsi rekursif untuk menghitung total nilai seluruh aset = jumlah × harga.

### Fungsi Pointer

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

- Menampilkan detail aset menggunakan pointer (->).

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

- Menambahkan jumlah aset dengan parameter by address.

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

- Mengisi data aset menggunakan pointer.

## Registrasi dan Login

```
void registrasi(User users[], int &totalUsers) {
    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++;
    } else {
        cout << "User telah penuh." << endl;
    }
}
```

- Menambahkan user baru ke array users.

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

```

- Autentikasi user, maksimal 3 percobaan.

---

## Tambah, Lihat, Ubah, dan Hapus Aset

```

void tambahAset(User &user) {
    if (user.jumlahAsetTersimpan < 100) {
        string nama;
        int jumlah;
        double harga;

        cout << "Nama Aset: ";
        getline(cin >> ws, nama);
        cout << "Jumlah: ";
        cin >> jumlah;
        cout << "Harga: ";
        cin >> harga;

        inisialisasiAset(&user.daftarAset[user.jumlahAsetTersimpan],
nama, jumlah, harga);
        user.jumlahAsetTersimpan++;
        cout << "Aset berhasil ditambahkan." << endl;
    } else {
        cout << "Daftar aset penuh." << endl;
    }
}

```

- Menambahkan aset baru ke user.

```

void lihatAset(User &user) {
    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++) {
            tampilkanAset(user.daftarAset[i], i);
        }

        double total = hitungTotalNilaiAset(user.daftarAset,
user.jumlahAsetTersimpan);
        cout << "Total nilai aset: " << total << endl;

        int pilih;
        cout << "\nIngin lihat detail aset tertentu? (nomor atau 0 untuk
lewati): ";
        cin >> pilih;
        if (pilih > 0 && pilih <= user.jumlahAsetTersimpan) {
            tampilkanDetailAset(&user.daftarAset[pilih - 1]);
        }
        else {
            cout << "Daftar aset kosong." << endl;
        }
    }
}

```

- Menampilkan semua aset dan bisa lihat detail salah satu.

```

void ubahAset(User &user) {
    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;

        int tambahan;
        cout << "Tambahan Jumlah (boleh 0): ";
        cin >> tambahan;
        tambahJumlah(&user.daftarAset[nomor].jumlah,
tambahan);
        cout << "Aset berhasil diubah." << endl;
    } else {
        cout << "Nomor aset tidak valid." << endl;
    }
}

```



- Mengubah aset berdasarkan nomor.

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

- Menghapus aset dan menggeser elemen setelahnya.

---

## Sorting

Tiga fungsi sorting ditambahkan (lihat bagian file sebelumnya):

- `urutkanAsetNamaAscending ()` — berdasarkan nama (huruf, ascending).
- `urutkanAsetJumlahDescending ()` — berdasarkan jumlah (angka, descending).
- `urutkanAsetHargaAscending ()` — berdasarkan harga (bebas).

---

## Menu Aset

```
void menuAset(User &user) {
    int pilihan;
    do {
        cout << "\nMenu:\n1. Tambah Aset\n2. Lihat Daftar Aset\n3. Ubah
Aset\n4. Hapus Aset\n"
        << "5. Urutkan Aset Nama (Asc)\n6. Urutkan Jumlah (Desc)\n7.
Urutkan Harga (Asc)\n"
        << "8. Logout\nPilihan Anda: ";
```

```

        cin >> pilihan;
        cin.ignore();
        switch (pilihan) {
            case 1: tambahAset(user); break;
            case 2: lihatAset(user); break;
            case 3: ubahAset(user); break;
            case 4: hapusAset(user); break;
            case 5: urutkanAsetNamaAscending(user); break;
            case 6: urutkanAsetJumlahDescending(user); break;
            case 7: urutkanAsetHargaAscending(user); break;
            case 8: cout << "Logout berhasil." << endl; break;
            default: cout << "Pilihan tidak valid." << endl;
        }
    } while (pilihan != 8);
}

```

- Menu CRUD dan sorting.

---

### Main Program

```

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;
}

```

- Program utama yang memanggil registrasi, login, dan menu 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.
```

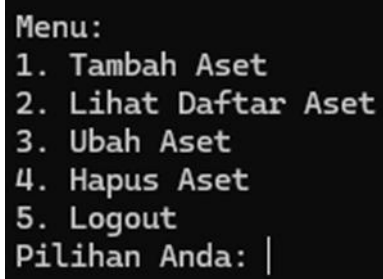
Gambar 4.1

#### 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
```