

**LAPORAN PRAKTIKUM**  
**POSTTEST 1**  
**ALGORITMA PEMROGRAMAN DASAR**



**Disusun oleh:**

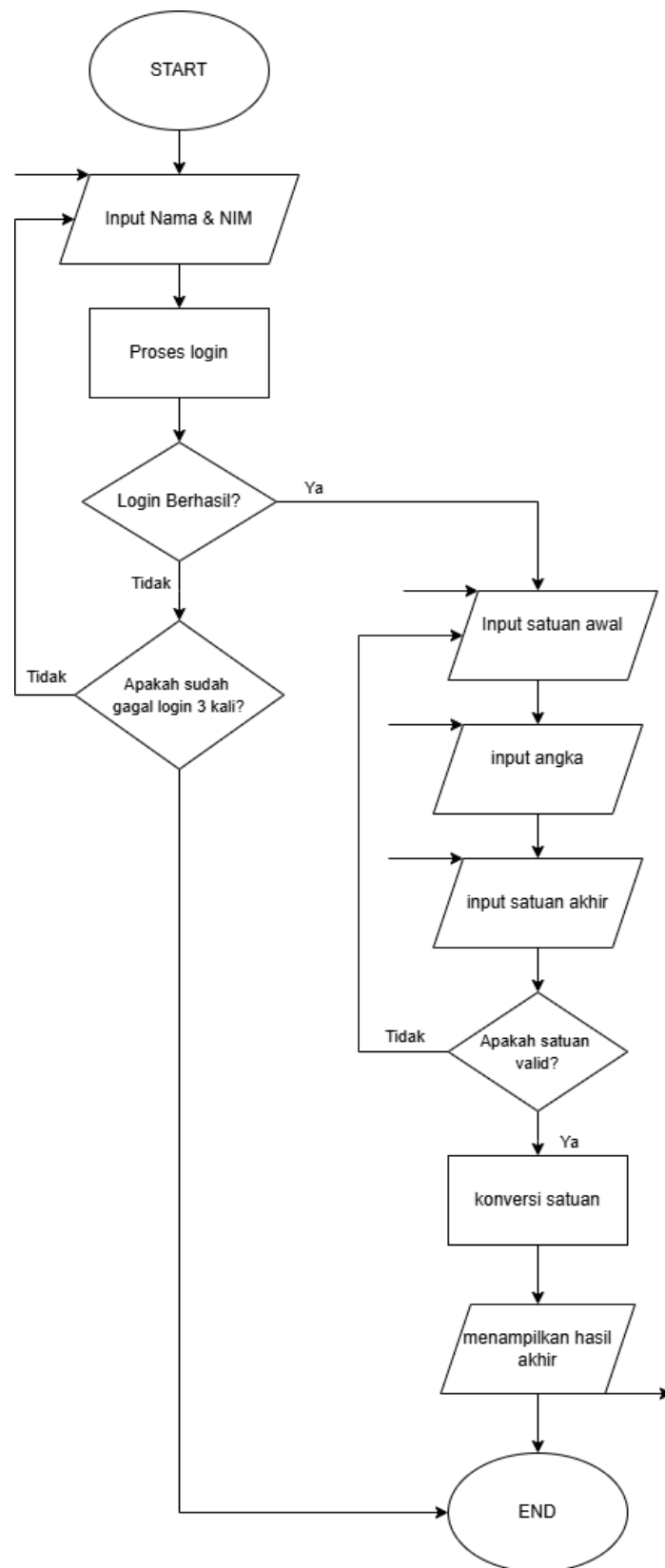
**Niluh Fincy Gloria Nathasia M (2409106012)**

**Kelas (A1'24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**

**2025**

## 1. Flowchart



Gambar 1.1 Flowchart

## 2. Analisis Program

### 2.1 Deskripsi Singkat Program

Program ini merupakan konversi satuan panjang yang berjalan secara berulang hingga user memilih untuk berhenti. Sebelum menggunakan fitur konversi, user harus login dengan memasukkan Nama dan NIM. Jika salah memasukkan sebanyak tiga kali, program akan berhenti. Setelah berhasil login, user dapat memilih jenis konversi yang diinginkan, seperti konversi dari Meter, Centimeter, Mile, atau Foot ke satuan lainnya. Program akan menerima input nilai dari user, melakukan perhitungan konversi, dan menampilkan hasilnya

### 2.2 Penjelasan Alur & Algoritma

#### Penjelasan Alur:

1. **Program berjalan dalam loop utama**
  - Program akan terus berjalan sampai user memilih untuk keluar.
2. **Login User**
  - user diminta untuk memasukkan **Nama** dan **NIM**.
  - Jika user salah memasukkan sebanyak **3 kali berturut-turut**, program akan berhenti.
  - Jika login berhasil, program akan lanjut ke menu utama.
3. **Menampilkan Menu Konversi**

User diberikan beberapa pilihan untuk melakukan konversi satuan panjang. Pilihan menu:

  - User memilih satu dari empat satuan
  - User menginput satuan awal yaitu mi (mile), m (meter), cm (centimeter), ft (foot)
  - User menginput angka yang mau dikonversi
  - User menginput satuan akhir yaitu mi (mile), m (meter), cm (centimeter), ft (foot)
4. **Proses Konversi**

Program akan meminta user memasukkan nilai yang akan dikonversi.

  - Program menghitung hasil konversi berdasarkan rumus yang sesuai.
  - Hasil konversi ditampilkan ke layar.

## 5. Kembali ke Menu Utama atau Keluar

- Setelah konversi selesai, user bisa memilih untuk melakukan konversi lain atau keluar dari program.

### Algoritma:

1. Loop Program Utama
2. Proses Login
3. Menu Konversi
4. Proses Konversi
5. Selesai.

## 1. Mulai Program & Inisialisasi Variabel

Program dimulai dengan mengimpor pustaka yang diperlukan:

```
#include<iostream>
#include<iomanip>
#include<cstdlib>
```

`#include <iostream>` → Digunakan untuk input/output standar (cin dan cout).

`#include <iomanip>` → Digunakan untuk format output yang lebih rapi, meskipun dalam kode ini tidak dimanfaatkan secara eksplisit.

`#include <cstdlib>` → Digunakan untuk fungsi `system("cls")` yang membersihkan layar.

Kemudian, variabel-variabel berikut dideklarasikan:

```
}--+*.
''''string satuan[] = {"mi", "m", "cm", "ft"};
string awal, akhir, nama, nim;
int posisi_awal = -1, posisi_akhir = -1;
double angka, hasil;
```

`satuan[]` → Array yang menyimpan daftar satuan panjang yang didukung.

`awal, akhir` → Menyimpan satuan asal dan tujuan yang dimasukkan oleh user.

`posisi_awal, posisi_akhir` → Variabel untuk mencari indeks satuan di array.

`angka, hasil` → Nilai yang akan dikonversi dan hasil konversinya.

Matriks faktor konversi antar satuan panjang juga dideklarasikan:

```
double faktor_konversi[4][4] = {
    {1, 1609, 160900, 5280},           // mi ke mi, m, cm, ft
    {1.0/1609, 1, 100, 3.281},        // m ke mi, m, cm, ft
    {1.0/160900, 1.0/100, 1, 1.0/30.48}, // cm ke mi, m, cm, ft
    {1.0/5280, 1.0/3.281, 30.48, 1}    // ft ke mi, m, cm, ft
};
```

Matriks ini digunakan untuk mencari faktor konversi antar satuan panjang.

Misalnya, 1 mil = 1609 meter, sehingga pada indeks [0][1] nilainya 1609.

Nilai lainnya diperoleh dengan faktor pembagi sesuai konversi yang berlaku.

## 2. Sistem Login

Program meminta user memasukkan nama dan NIM, lalu mencocokkannya dengan data yang telah ditentukan:

```
int percobaan = 0;
while (percobaan < 3) {
    cout << "=====\n";
    cout << "          LOGIN SISTEM\n";
    cout << "=====\n";
    cout << "Masukkan Nama : "; getline(cin, nama);
    cout << "Masukkan NIM  : "; getline(cin, nim);

    if (nama == "Niluh Fincy Gloria Nathasia M" && nim ==
"2409106012") {
        system("cls"); // Membersihkan layar
        break;
    }

    cout << "Nama atau NIM salah! Coba lagi.\n";
    percobaan++;

    if (percobaan == 3) {
        cout << "Percobaan lebih dari 3 kali. Program berhenti.\n";
        return 1; // Program dihentikan jika salah lebih dari 3
kali.
    }
}
```

Program meminta nama dan NIM menggunakan getline(cin, nama) agar bisa membaca spasi dalam input.

Jika input benar, layar dibersihkan (system("cls")), dan user bisa lanjut ke proses konversi.

Jika input salah, program memberikan maksimal 3 kali percobaan, setelah itu akan dihentikan.

### 3. Input & Validasi Satuan Panjang

Setelah login berhasil, user memasukkan:

1. Satuan asal (mi, m, cm, ft)
2. Nilai yang akan dikonversi
3. Satuan tujuan (mi, m, cm, ft)

```
char lanjut;  
do {  
    cout << "=====\n";  
    cout << "          KONVERSI SATUAN PANJANG\n";  
    cout << "=====\n";  
    cout << "Masukkan satuan awal (mi/m/cm/ft): "; cin >> awal;  
    cout << "Masukkan angka   : "; cin >> angka;  
    cout << "Masukkan satuan akhir (mi/m/cm/ft): "; cin >> akhir;  
    cout << "-----\n";  
}
```

Satuan awal dan akhir disimpan dalam variabel awal dan akhir.

Nilai yang dikonversi disimpan dalam angka.

### 4. Mencari Indeks Satuan dalam Array

Program memeriksa apakah satuan yang dimasukkan valid dengan mencocokkannya dalam array satuan[ ]:

```
posisi_awal = -1;  
posisi_akhir = -1;  
  
for(int i = 0; i < 4; i++){  
    if(awal == satuan[i]) posisi_awal = i;  
    if(akhir == satuan[i]) posisi_akhir = i;  
}
```

Jika satuan valid, indeksnya disimpan dalam posisi\_awal dan posisi\_akhir.

Jika satuan tidak ditemukan, nilainya tetap -1.

Jika satuan tidak valid, program menampilkan pesan kesalahan:

```
if(posisi_awal == -1 || posisi_akhir == -1){  
    cout << "gak ada satuan itu, pastikan cuma pakai:  
(mi/m/cm/ft)\n";  
}
```

## 5. Konversi & Output Hasil

Jika satuan valid, program melakukan perhitungan menggunakan matriks konversi:

```
else {
    hasil = angka * faktor_konversi[posisi_awal][posisi_akhir];
    cout << angka << " " << awal << " = " << hasil << " " << akhir
    << endl;
}
```

Nilai input (angka) dikalikan dengan faktor konversi yang sesuai dalam matriks faktor\_konversi.

Hasil ditampilkan dalam format:

10 mi = 16090 m

## 6. Opsi Melanjutkan atau Keluar

Program menanyakan apakah user ingin melanjutkan atau tidak:

```
cout << "\n lanjut konversi gak? (y/n): "; cin >> lanjut;
cin.ignore();
system("cls");
```

Jika user mengetik 'y' atau 'Y', program mengulangi proses konversi.

Jika mengetik 'n' atau 'N', program keluar dan menampilkan pesan:  
program selesai

## 7. Program Berakhir

Jika user memilih keluar, program mencetak:

```
cout << "program selesai\n";
return 0;
```

return 0 menandakan bahwa program berjalan dengan sukses.

### 3. Source Code

#### 3.1. Fungsi Login.

Fungsi Login berfungsi menyimpan data nama dan NIM, dan terdapat output menu untuk menginput nama dan NIM. User diminta memasukkan Nama dan NIM. Program memeriksa apakah Nama dan NIM sesuai dengan data yang benar. Jika benar, layar dibersihkan, dan program lanjut ke konversi satuan. Jika salah, user diberi kesempatan mencoba lagi. User hanya diberikan maksimal 3 kesempatan. Jika masih salah setelah 3 kali percobaan, program berhenti.

#### Source Code:

```
int percobaan = 0;
while (percobaan < 3) {

    cout << "=====\n";
    cout << "  LOGIN SISTEM\n";
    cout << "=====\n";
    cout << "Masukkan Nama : "; getline(cin, nama);
    cout << "Masukkan NIM : "; getline(cin, nim);

    if (nama == "Niluh Fincy Gloria Nathasia M" && nim == "2409106012") {
        system("cls");
        break;
    }

    cout << "Nama atau NIM salah! Coba lagi.\n";
    percobaan++;

    if (percobaan == 3) {
        cout << "Percobaan lebih dari 3 kali. Program berhenti.\n";
        return 1;
    }
}
```



### 3.2. Fungsi Main Utama

User diminta memasukkan satuan awal, angka, dan satuan tujuan. Program memeriksa apakah satuan yang dimasukkan valid. Jika valid, program melakukan konversi menggunakan faktor yang sesuai. Hasil konversi ditampilkan ke layar. User ditanya apakah ingin melanjutkan konversi atau keluar. Jika lanjut (y/n), layar dibersihkan, dan program mengulang dari awal. Jika keluar (n/N), program menampilkan "program selesai" dan berhenti.

```
char lanjut;
do {
    cout << "=====\n";
    cout << "          KONVERSI SATUAN PANJANG\n";
    cout << "=====\n";
    cout << "Masukkan satuan awal (mi/m/cm/ft): "; cin >> awal;
    cout << "Masukkan angka : "; cin >> angka;
    cout << "Masukkan satuan akhir (mi/m/cm/ft): "; cin >> akhir;
    cout << "-----\n";

    posisi_awal = -1;
    posisi_akhir = -1;

    for(int i = 0; i < 4; i++){
        if(awal == satuan[i]) posisi_awal = i;
        if(akhir == satuan[i]) posisi_akhir = i;
    }

    if(posisi_awal == -1 || posisi_akhir == -1){
        cout << "gak ada satuan itu, pastikan cuma pakai:
(mi/m/cm/ft)\n";
    } else {
        hasil = angka * faktor_konversi[posisi_awal][posisi_akhir];
        cout << angka << " " << awal << " = " << hasil << " " <<
akhir << endl;
    }

    cout << "\n lanjut konversi gak? (y/n): "; cin >> lanjut;
    cin.ignore();
    system("cls");
} while (lanjut == 'y' || lanjut == 'Y');

cout << "program selesai\n";
return 0;
```

## 4. Uji Coba dan Hasil Output

### 4.1 Uji Coba

Disini saya akan menginput nama “Niluh Fincy Gloria Nathasia M” dan NIM “2409106012” karena default sistemnya mengharuskan menginput nama dan NIM seperti di atas.

#### 6. Skenario 1

User memilih satuan mile sebesar 1 mile untuk dikonversikan ke meter, dan user memilih programnya untuk lanjut.

#### 7. Skenario 2

User memilih satuan meter sebesar 1 m untuk dikonversikan ke foot, dan user memilih programnya untuk berhenti

## 4.2 Hasil Output

### Hasil Output Skenario 1:

```
=====
                        LOGIN SISTEM
=====
Masukkan Nama : Niluh Fincy Gloria Nathasia M
Masukkan NIM  : 2409106012
```

Gambar 4.2.1 Menu Login

```
=====
                        KONVERSI SATUAN PANJANG
=====
Masukkan satuan awal (mi/m/cm/ft): mi
Masukkan angka      : 1
Masukkan satuan akhir (mi/m/cm/ft): m
-----
1 mi = 1609 m
```

Gambar 4.2.2 Menu Konversi

lanjut konversi gak? (y/n): y

Gambar 4.2.3 Validasi Lanjut atau Tidak

### Hasil Output Skenario 2:

```
=====
                        LOGIN SISTEM
=====
Masukkan Nama : Niluh Fincy Gloria Nathasia M
Masukkan NIM  : 2409106012
```

Gambar 4.2.4 Menu Login

```
=====
                        KONVERSI SATUAN PANJANG
=====
Masukkan satuan awal (mi/m/cm/ft): m
Masukkan angka      : 1
Masukkan satuan akhir (mi/m/cm/ft): ft
-----
1 m = 3.281 ft
```

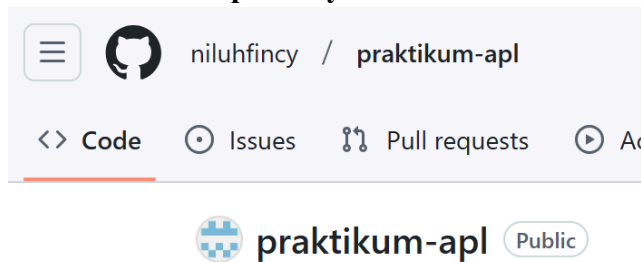
Gambar 4.2.5 Menu Konversi

lanjut konversi gak? (y/n): n

Gambar 4.2.6 Validasi Lanjut atau Tidak

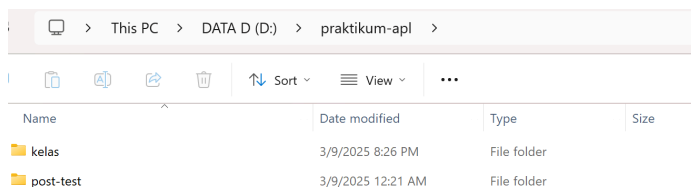
## 5. Langkah-langkah GIT

### 5.1 Membuat Repository Publik di GitHub



Gambar 5.1 Membuat Repository

### 5.2 Membuat Folder



Gambar 5.2 Membuat Folder

### 5.3 Git init

Perintah ini akan membuat folder `.git` didalam direktori yang akan digunakan untuk menyimpan informasi repository.

```
PS D:\praktikum-apl> git init
Initialized empty Git repository in D:/praktikum-apl/.git/
```

Gambar 5.3 Inisialisasi Repository

### 5.4 Git add

Perintah ini digunakan untuk menambahkan semua file dan folder dalam direktori ke staging area.

```
PS D:\praktikum-apl> git add .
```

Gambar 5.5 Menambahkan Semua File

### 5.5 Git Commit

Git commit digunakan untuk menyimpan perubahan di repository lokal dengan pesan deskriptif.

```
PS D:\praktikum-apl> git commit -m "Finish Posttest 1"
```

Gambar 5.5 Commit Posttest 1

### 5.6 Git Remote (Menghubungkan Repository Lokal dengan GitHub)

Git remote digunakan untuk menambahkan repository remote

```
PS D:\praktikum-apl> git remote add origin https://github.com/niluhfincy/praktikum-apl.git
```

Gambar 5.6 Command git remote yang didapat pada GitHub

### 5.7 Git Push (Mengupdate semua yang ada di Repository Lokal)

Git push untuk mengupload perubahan ke repository di GitHub

```
PS D:\praktikum-apl> git push -u origin main
```

Gambar 5.7 Git push untuk mengupload file dan folder yang telah diubah ke repository GitHub