LAPORAN PRAKTIKUM

**POSTTEST 1**

**ALGORITMA PEMROGRAMAN LANJUT**

****

**Disusun oleh:**

**Nama (2409106044)**

**Kelas (A2 ‘24)**

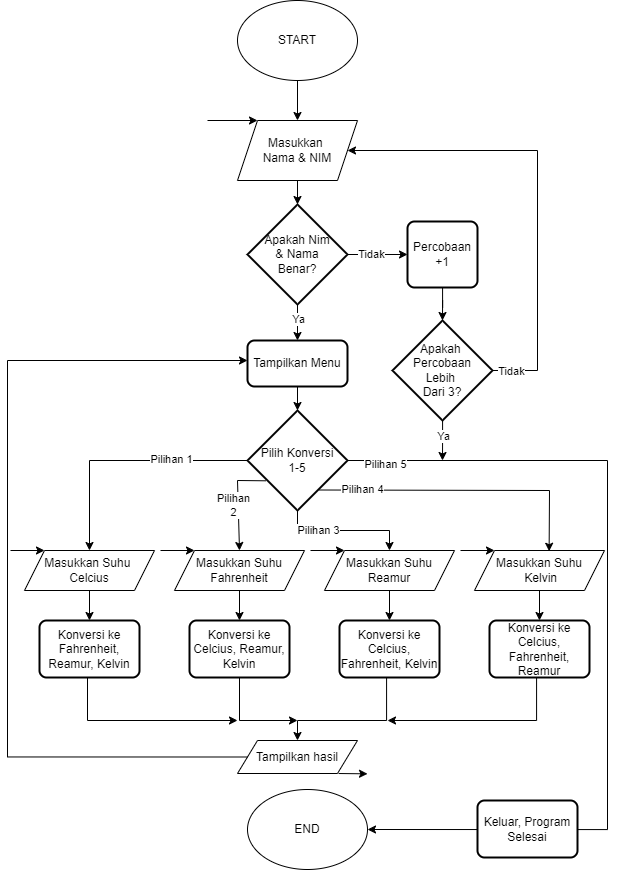
**PROGRAM STUDI INFORMATIKA**

**UNIVERSITAS MULAWARMAN**

**SAMARINDA**

**2025**

# 1. Flowchart



# 2. Analisis Program

## 2.1 Deskripsi Singkat Program

Program ini bertujuan untuk melakukan konversi suhu dari satu satuan ke satuan lainnya, seperti Celcius ke Fahrenheit, Reamur, dan Kelvin. Program ini juga mempunyai fitur login yang membatasi pengguna untuk tiga kali percobaan.

## 2.2 Penjelasan Alur & Algoritma

**Algoritma:**

* Program ini dimulai dengan meminta pengguna untuk login menggunakan nama dan 3 angka NIM terakhir yang telah ditentukan.
* Jika login berhasil, pengguna diarahkan ke menu utama yang berisi pilihan konversi suhu.
* Jika login gagal, pengguna diberikan 3 percobaan dan jika percobaan habis maka program akan berhenti.
* Pengguna memilih salah satu dari empat jenis konversi suhu yang tersedia.
* Program meminta input suhu dari pengguna dan mengonversinya ke satuan lainnya sesuai pilihan pengguna.
* Hasil konversi ditampilkan.
* Jika pengguna memilih nomor lima maka akan program akan berhenti.

**Penjelasan Detail Blok Code/Fungsi Pada Program:**

* **Login:**
* Login menggunakan nama dan 3 NIM terakhir sebagai password, yaitu pada program ini dengan nama (Rangga) dan 3 NIM terakhir (044).
* Pengguna Diberi 3 Percobaan Login.
* Jika pengguna gagal login lebih dari 3 kali, maka program berhenti.

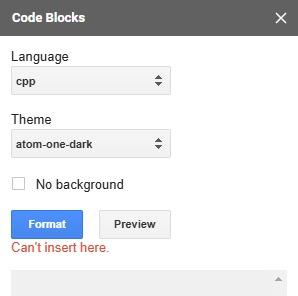
Jika login berhasil maka diarahkan ke Menu.

* **Menu:**
* Menu Utama Memiliki 5 Bagian Yaitu.
* Konversi Celcius ke Fahrenheit, Reamur, Kelvin.
* Konversi Farenheit ke Celcius, Reamur, Kelvin.
* Konversi Reamur ke Celcius, Fahrenheit, Kelvin.
* Konversi Kelvin ke Celcius, Fahrenheit, Reamur.
* Dan keluar dari program.
* **Konversi:**
* Program akan meminta input untuk memilih konversi apa yang diingkan pengguna
* Jika sudah memilih jenis konversi, maka program akan meminta input suhu
* Program akan menampilkan hasil konversi

# 3. Source Code

*(Pada bagian ini, sertakan hanya bagian kode yang berisi fitur penting dari program. Tidak perlu memasukkan seluruh kode, cukup bagian yang relevan dengan fungsionalitas utama.)*

**Catatan:**

* **Gunakan ekstensi Code Blocks di Google Docs** untuk menampilkan kode dengan format yang lebih rapi.
* Unduh dan instal ekstensi di sini:  
  [**Code Blocks - Google Workspace Marketplace**](https://workspace.google.com/marketplace/app/code_blocks/100740430168)
* **Jangan gunakan tabel** untuk menampilkan kode program.
* Untuk **Source Code Font Uk : 10**
* Gunakan **“Language”** dan **“Theme”** yang sudah ditentukan

**Contoh Format dalam Laporan:**

### **A. Fitur Tambah Data**

Fitur ini digunakan untuk menambahkan data baru ke dalam array. Data yang ditambahkan berupa ID dan nama.

**Source Code:**

|  |
| --- |
| **void jumpSearch(int jumlahAyam) {  if (anggotaHead == nullptr) {  cout << "Tidak ada anggota yang diterima.\n";  return;  }   *// Pertama urutkan data berdasarkan jumlah ayam*  mergeSort(anggotaHead);   *// Menghitung panjang linked list*  int length = getLength(anggotaHead);  int jump = sqrt(length);    cout << "\nHasil Pencarian Berdasarkan Jumlah Ayam " << jumlahAyam << ":\n";    Anggota\* current = anggotaHead;  Anggota\* prev = nullptr;  bool found = false;   *// Jump search yang dimodifikasi untuk linked list*  while (current != nullptr && current->jumlahAyam <= jumlahAyam) {  prev = current;    *// Lompat ke depan sejauh jump langkah*  for (int i = 0; i < jump && current != nullptr; i++) {  current = current->next;  }    *// Jika sudah mencapai akhir list, lakukan linear search dari prev*  if (current == nullptr || current->jumlahAyam > jumlahAyam) {  *// Linear search dari prev sampai current atau sampai nilai lebih besar*  Anggota\* linearSearch = prev;  while (linearSearch != current && linearSearch != nullptr &&   linearSearch->jumlahAyam <= jumlahAyam) {  if (linearSearch->jumlahAyam == jumlahAyam) {  cout << "Ditemukan:\n";  cout << "Nama: " << linearSearch->namaPeternak   << " | Jumlah Ayam: " << linearSearch->jumlahAyam   << " | Alamat: " << linearSearch->alamat << " |\n";  found = true;  }  linearSearch = linearSearch->next;  }  break;  }  }   if (!found) {  cout << "Tidak ditemukan anggota dengan jumlah ayam " << jumlahAyam << endl;  } }** |

|  |
| --- |
| void tambahData(int &jumlah, int id[], string nama[]) {  cout << "Masukkan ID: ";  cin >> id[jumlah];  cout << "Masukkan Nama: ";  cin.ignore();  getline(cin, nama[jumlah]);  jumlah++;  cout << "Data berhasil ditambahkan!\n"; } |

# 4. Uji Coba dan Hasil Output

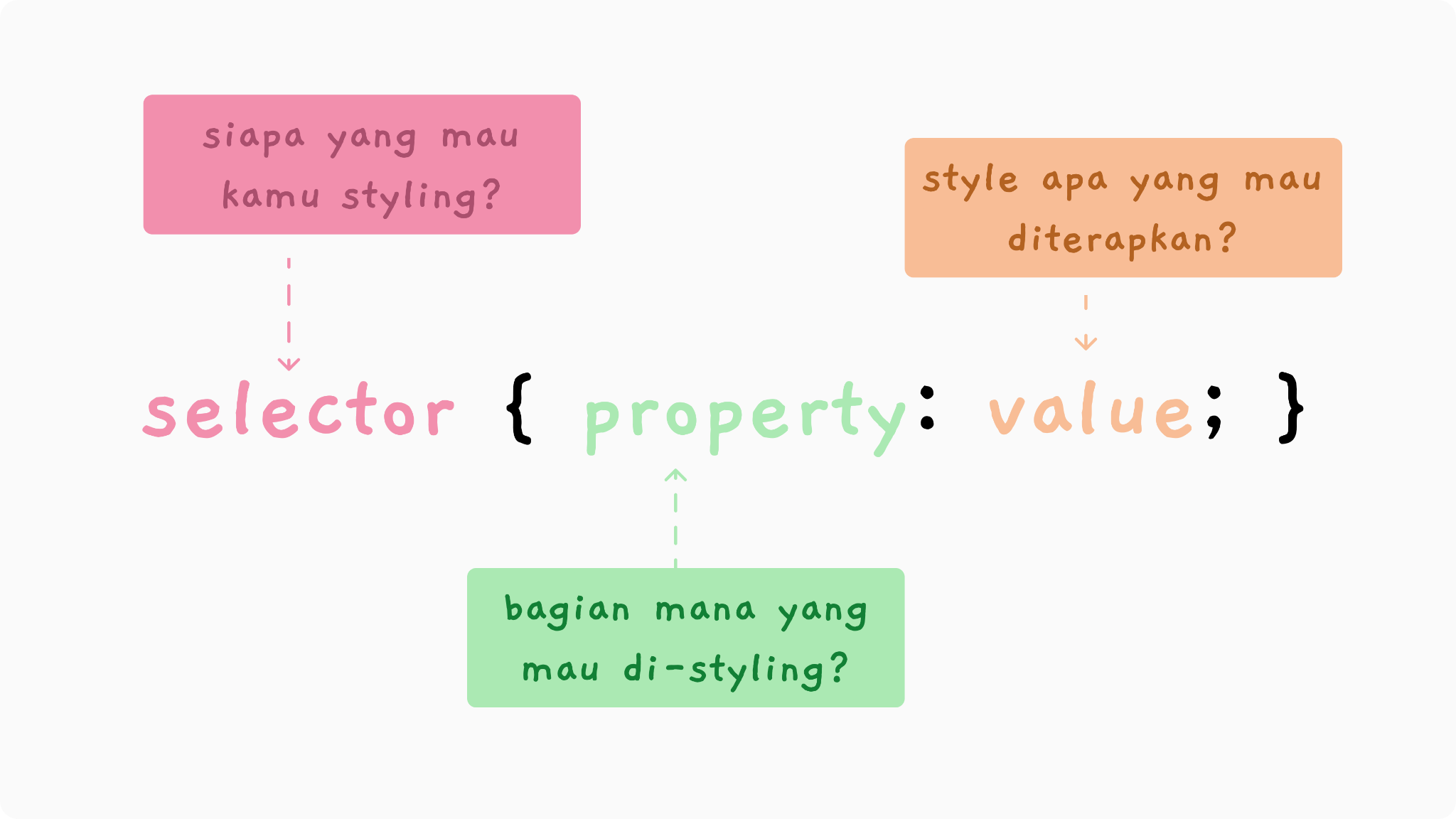
## 4.1 Uji Coba

*(Jelaskan skenario yang digunakan untuk menguji program, misalnya dengan berbagai jenis input.)*

1. Skenario 1
2. Skenario 2

## 4.2 Hasil Output

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



Gambar 4.1 Contoh

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