

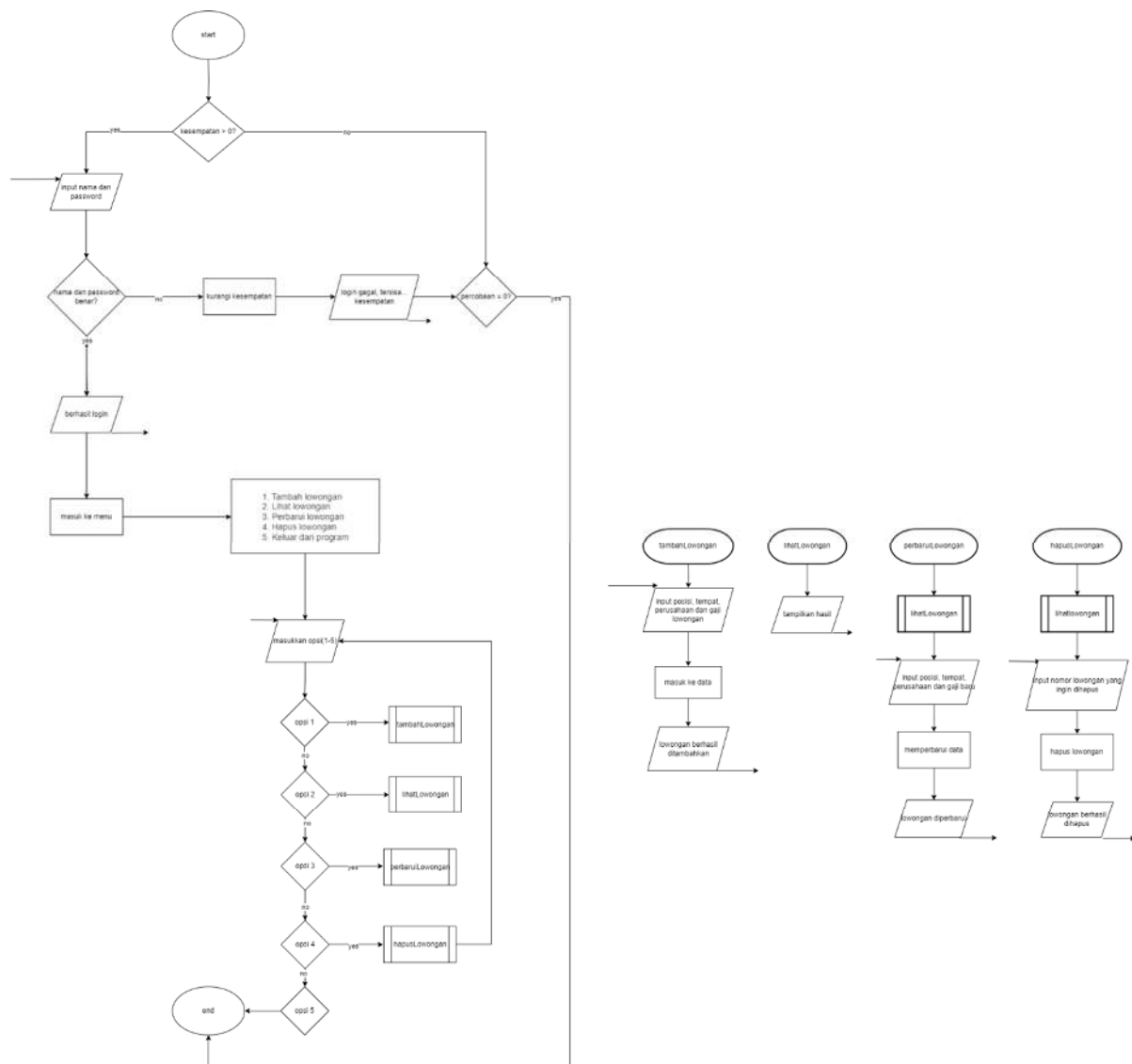
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Rangga Lawe (2409106043)**  
**Kelas (A2'24)**

**PROGRAM STUDI INFORMATIKA**  
**UNIVERSITAS MULAWARMAN**  
**SAMARINDA**  
**2025**

# 1. Flowchart



## 2. Analisis Program

Program ini bertujuan untuk mengelola daftar lowongan kerja dengan sistem login pengguna. Pengguna harus memasukkan nama dan 3 digit terakhir NIM untuk mengakses program. Jika gagal login 3 kali, program akan berhenti. Setelah login berhasil, pengguna dapat menambah, melihat, memperbarui, dan menghapus lowongan kerja. Program terus berjalan hingga pengguna memilih keluar.

### 3. Source Code

#### A. fitur login

Pengguna diminta untuk memasukkan nama dan nim yang sesuai dan jika pengguna salah memasukkan tiga kali maka akan langsung terkeluar dari program.

```
1  while (kesempatan > 0) {
2      cout << "Masukkan Nama: ";
3      cin >> pengguna.nama;
4      cout << "Masukkan NIM (3 digit terakhir): ";
5      cin >> pengguna.nim;
6
7      if (pengguna.nama == "Rangga" && pengguna.nim == "043") {
8          cout << "Login berhasil!\n";
9          break;
10     } else {
11         kesempatan--;
12         cout << "Login gagal! Sisa percobaan: " << kesempatan << "\n";
13     }
14 }
15
16 if (kesempatan == 0) {
17     cout << "Anda gagal login 3 kali. Program berhenti.\n";
18     return 0;
19 }
```

#### B. Fitur tambah lowongan (fungsi dengan parameter address-of)

Pengguna diminta untuk memasukkan suatu lowongan yang terdiri dari posisi, perusahaan, tempat, dan gaji. Lowongan akan langsung dimasukkan dalam data jika jumlah lowongan sedang tidak penuh.

```
1  void tambahLowongan(Lowongan daftar[], int &jumlah) {
2      if (jumlah < MAX_LOWONGAN) {
3          cout << "Posisi: ";
4          cin >> daftar[jumlah].posisi;
5          cout << "Perusahaan: ";
6          cin >> daftar[jumlah].perusahaan;
7          cout << "Tempat: ";
8          cin >> daftar[jumlah].tempat;
9          cout << "Gaji: ";
10         cin >> daftar[jumlah].gaji;
11         jumlah++;
12         cout << "Lowongan ditambahkan!\n";
13     } else {
14         cout << "Daftar lowongan penuh!\n";
15     }
16 }
```

C. Fitur lihat lowongan (fungsi dengan parameter dereference)  
Memungkinkan pengguna untuk melihat lowongan yang tersedia.

```
1 void lihatLowongan(Lowongan *daftar, int jumlah) {
2     if (jumlah > 0) {
3         for (int i = 0; i < jumlah; i++) {
4             cout << i + 1 << ". " << daftar[i].posisi << " - "
5                 << daftar[i].perusahaan << " (" << daftar[i].tempat
6                 << ") Rp" << daftar[i].gaji << "\n";
7         }
8     } else {
9         cout << "Tidak ada lowongan tersedia!\n";
10    }
11 }
```

D. Fitur perbarui data lowongan  
Memungkinkan pengguna untuk memperbarui data lowongan jika terjadi kesalahan ataupun ada informasi baru.

```
1 void perbaruiLowongan(Lowongan daftar[], int jumlah) {
2     int index;
3     if (jumlah > 0) {
4         cout << "Nomor lowongan: ";
5         cin >> index;
6         if (index > 0 && index <= jumlah) {
7             index--;
8             cout << "Posisi baru: ";
9             cin >> daftar[index].posisi;
10            cout << "Perusahaan baru: ";
11            cin >> daftar[index].perusahaan;
12            cout << "Tempat baru: ";
13            cin >> daftar[index].tempat;
14            cout << "Gaji baru: ";
15            cin >> daftar[index].gaji;
16            cout << "Lowongan diperbarui!\n";
17        } else {
18            cout << "Nomor tidak valid!\n";
19        }
20    } else {
21        cout << "Tidak ada lowongan untuk diperbarui!\n";
22    }
23 }
```

E. Fitur hapus lowongan (fungsi dengan parameter address-of)

Dapat menghapus lowongan yang sudah tidak diperlukan atau sudah diambil.

```
1 void hapusLowongan(Lowongan daftar[], int &jumlah) {
2     int index;
3     if (jumlah > 0) {
4         cout << "Nomor lowongan: ";
5         cin >> index;
6         if (index > 0 && index <= jumlah) {
7             for (int i = index - 1; i < jumlah - 1; i++) {
8                 daftar[i] = daftar[i + 1];
9             }
10            jumlah--;
11            cout << "Lowongan dihapus!\n";
12        } else {
13            cout << "Nomor tidak valid!\n";
14        }
15    } else {
16        cout << "Tidak ada lowongan untuk dihapus!\n";
17    }
18 }
```

F. Fitur urutkan lowongan

Dapat melihat urutan dari posisi dan perusahaan di lowongan dari huruf awalnya dan juga bisa melihat gaji yang terurut dari tertinggi ke terendah menggunakan metode sorting yang berbeda-beda.

```
1 void urutkanLowongan(Lowongan daftar[], int jumlah) {
2     int pilihan;
3     cout << "\nUrutkan berdasarkan:\n";
4     cout << "1. Posisi (A ke Z)\n";
5     cout << "2. Gaji (Tertinggi ke Terendah)\n";
6     cout << "3. Perusahaan (Z ke A)\n";
7     cout << "Masukkan pilihan (1-3): ";
8     cin >> pilihan;
```

1. Fitur mengurutkan posisi dengan menggunakan bubble sort.

```
1 case 1: // Bubble Sort
2     for (int i = 0; i < jumlah - 1; i++) {
3         for (int j = 0; j < jumlah - i - 1; j++) {
4             if (daftar[j].posisi > daftar[j + 1].posisi) {
5                 Lowongan sementara = daftar[j];
6                 daftar[j] = daftar[j + 1];
7                 daftar[j + 1] = sementara;
8             }
9         }
10    }
11    cout << "Lowongan diurutkan berdasarkan posisi (A ke Z).\n";
12    break;
```

2. Fitur mengurutkan gaji menggunakan selection sort

```
1 case 2: // Selection Sort
2     for (int i = 0; i < jumlah - 1; i++) {
3         int maks = i;
4         for (int j = i + 1; j < jumlah; j++) {
5             if (daftar[j].gaji > daftar[maks].gaji) {
6                 maks = j;
7             }
8         }
9         Lowongan sementara = daftar[i];
10        daftar[i] = daftar[maks];
11        daftar[maks] = sementara;
12    }
13    cout << "Lowongan diurutkan berdasarkan gaji (tertinggi ke terendah).\n";
14    break;
```

### 3. Fitur mengurutkan perusahaan menggunakan insertion sort

```
1 case 3: // Insertion Sort
2     for (int i = 1; i < jumlah; i++) {
3         Lowongan simpan = daftar[i];
4         int j = i - 1;
5         while (j >= 0 && daftar[j].perusahaan < simpan.perusahaan) {
6             daftar[j + 1] = daftar[j];
7             j--;
8         }
9         daftar[j + 1] = simpan;
10    }
11    cout << "Lowongan diurutkan berdasarkan perusahaan (Z ke A).\n";
12    break;
```

### G. Keluar dari program

Memungkinkan pengguna untuk keluar dari program.

```
1 cout << "Keluar dari program.\n";
2     break;
3     default:
4         cout << "Pilihan tidak valid!\n";
5         break;
6     }
7 } while (pilihan != 5);
8
9 return 0;
```

#### 4. Uji Coba dan Hasil Output

1. output dari login yang berhasil dan fitur tambah lowongan.

```
Masukkan Nama: Ranga
Masukkan NIM (3 digit terakhir): 043
Login berhasil!

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Masukkan opsi(1-5): 1
Posisi: operator
Perusahaan: KPC
Tempat: sangatta
Gaji: 100000
Lowongan ditambahkan!
```

2. output dari fitur lihat lowongan dan perbarui lowongan.

```
1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Masukkan opsi(1-5): 2
1. operator - KPC (sangatta) Rp100000

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Masukkan opsi(1-5): 3
Nomor lowongan: 1
Posisi baru: mekanik
Perusahaan baru: PAMA
Tempat baru: bengalon
Gaji baru: 120000
Lowongan diperbarui!
```



3. fitur hapus lowongan dan keluar dari program.

```
1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Masukkan opsi(1-5): 4
Nomor lowongan: 1
Lowongan dihapus!

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Keluar
Masukkan opsi(1-5): 5
Keluar dari program.
PS C:\Users\Asus GK\OneDrive\Documents\praktikum-apl\Post-test\posttest 3>
```

4. Output dari Fitur mengurutkan posisi dengan menggunakan bubble sort.

```
Urutkan berdasarkan:
1. Posisi (A ke Z)
2. Gaji (Tertinggi ke Terendah)
3. Perusahaan (Z ke A)
Masukkan pilihan (1-3): 1
Lowongan diurutkan berdasarkan posisi (A ke Z).

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Urutkan Lowongan
6. Keluar
Masukkan opsi (1-6): 2
1. hrd - buma (tabang) Rp120000
2. mekanik - pama (bengalon) Rp110000
3. operator - kpc (sangatta) Rp100000
```

5. Output dari Fitur mengurutkan gaji menggunakan selection sort

```
Urutkan berdasarkan:
1. Posisi (A ke Z)
2. Gaji (Tertinggi ke Terendah)
3. Perusahaan (Z ke A)
Masukkan pilihan (1-3): 2
Lowongan diurutkan berdasarkan gaji (tertinggi ke terendah).

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Urutkan Lowongan
6. Keluar
Masukkan opsi (1-6): 2
1. hrd - buma (tabang) Rp120000
2. mekanik - pama (bengalon) Rp110000
3. operator - kpc (sangatta) Rp100000
```

6. Output dari Fitur mengurutkan perusahaan menggunakan insertion sort

```
Urutkan berdasarkan:
1. Posisi (A ke Z)
2. Gaji (Tertinggi ke Terendah)
3. Perusahaan (Z ke A)
Masukkan pilihan (1-3): 3
Lowongan diurutkan berdasarkan perusahaan (Z ke A).

1. Tambah Lowongan
2. Lihat Lowongan
3. Perbarui Lowongan
4. Hapus Lowongan
5. Urutkan Lowongan
6. Keluar
Masukkan opsi (1-6): 2
1. mekanik - pama (bengalon) Rp110000
2. operator - kpc (sangatta) Rp100000
3. hrd - buma (tabang) Rp120000
```

## 5. Langkah-Langkah Git pada VSCode

A. Git add . untuk menyiapkan semua perubahan di file agar bisa disimpan.

```
PS C:\Users\Asus GK\OneDrive\Documents\praktikum-apl> git add .  
PS C:\Users\Asus GK\OneDrive\Documents\praktikum-apl>
```

B. Git commit untuk menyimpan perubahan ke dalam repository lokal

```
PS C:\Users\Asus GK\OneDrive\Documents\praktikum-apl> git commit -m "finish posttest 6"  
[main 3b7d7d6] finish posttest 6  
2 files changed, 208 insertions(+)  
create mode 100644 Post-test/posttest 6/2409106043-Ranggalawe-PT-6.cpp  
create mode 100644 Post-test/posttest 6/2409106043-Ranggalawe-PT-6.exe
```

C. Git push untuk mengirim perubahan ke github atau repository online agar bisa dilihat oleh orang lain.

```
PS C:\Users\Asus GK\OneDrive\Documents\praktikum-apl> git push  
Enumerating objects: 8, done.  
Counting objects: 100% (8/8), done.  
Delta compression using up to 16 threads  
Compressing objects: 100% (6/6), done.  
Writing objects: 100% (6/6), 677.40 KiB | 8.57 MiB/s, done.  
Total 6 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)  
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.  
To https://github.com/Rilzkun/praktikum-apl.git  
77aea82..3b7d7d6 main -> main
```