

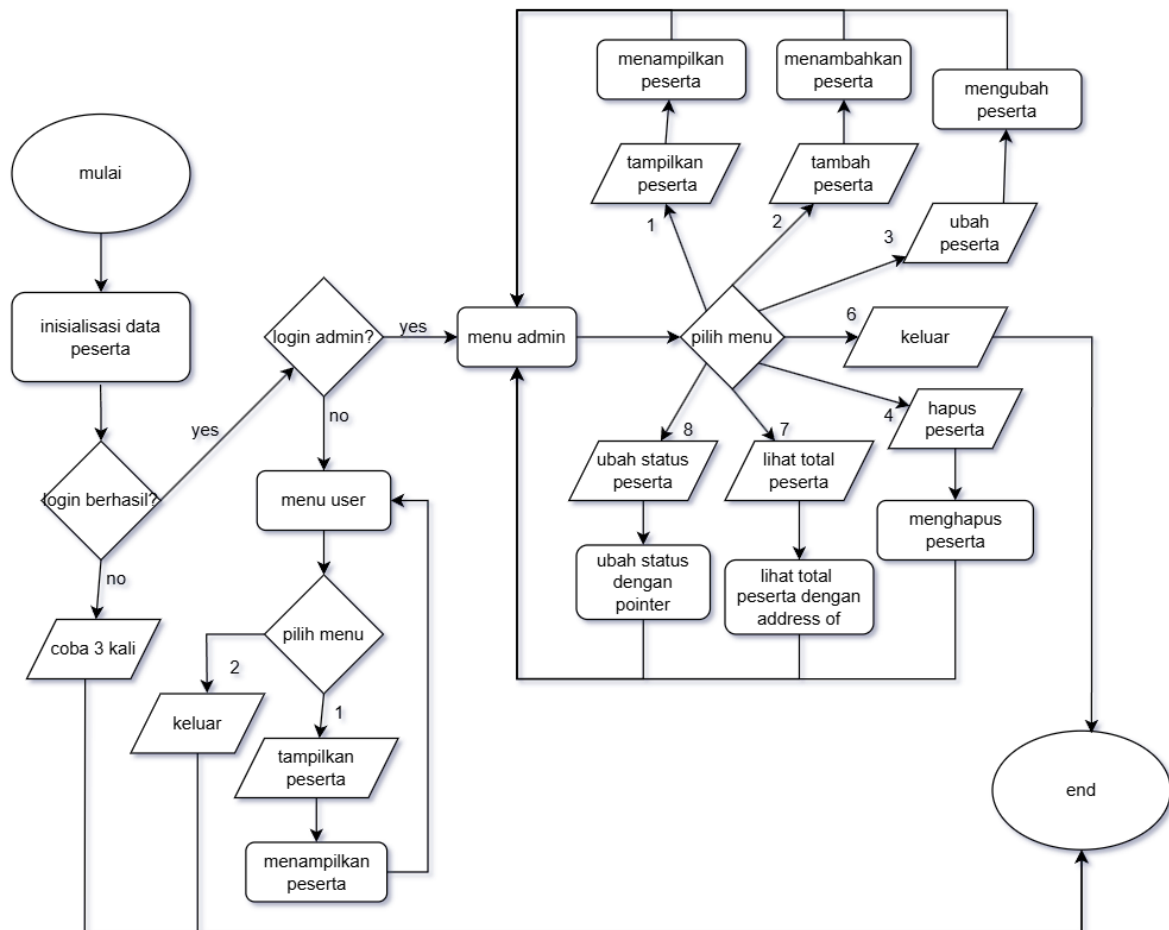
**LAPORAN PRAKTIKUM**  
**POSTTEST (5)**  
**ALGORITMA PEMROGRAMAN LANJUT**



**Disusun oleh:**  
**Avella Salsabella Hermawan (2409106045)**  
**Kelas (A2 '24)**

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

## 1. Flowchart



## 2. Analisis Program

Program ini merupakan aplikasi berbasis bahasa pemrograman C++ yang dirancang untuk mengelola data peserta dalam sebuah sistem seleksi. Program dilengkapi dengan sistem login berbasis peran (role) yang membedakan akses antara admin dan user. Pengguna harus melakukan login terlebih dahulu. Jika pengguna memasukkan nama "admin" dan NIM "000", maka akan masuk sebagai admin. Sementara itu, jika pengguna bukan admin dan NIM terdiri dari tiga digit, maka akan masuk sebagai user. Jika proses login gagal dilakukan sebanyak tiga kali, lebih dari itu program akan berhenti secara otomatis.

Setelah berhasil login, program akan menampilkan menu yang berbeda tergantung peran pengguna. Admin memiliki akses penuh terhadap data, yaitu dapat menampilkan daftar peserta, menambah peserta baru, mengubah data peserta yang sudah ada, dan menghapus peserta dari daftar. Sementara itu, user hanya diberikan akses untuk melihat daftar peserta tanpa bisa mengubah data apa pun.

Program ini menggunakan dua fungsi penerapan pointer. Pertama, fungsi untuk menampilkan total peserta langsung menggunakan parameter address-of (&). Kedua, fungsi untuk mengakses dan mengubah status peserta menggunakan parameter dereference (\*) secara langsung di memori melalui pointer.

### 3. Screenshoot Program

```
2  #include <iostream>
3  #include <string>
4  using namespace std;
5
6  #define MAX_PESERTA 40
7
8  struct Peserta {
9      string nama;
10     string asalNegara;
11     int umur;
12     string posisi;
13     string status;
14 };
15
16 // Prosedur menu
17 void tampilkanMenuAdmin() {
18     cout << "\nMenu Admin:\n"
19     << "1. Tampilkan Peserta\n"
20     << "2. Tambah Peserta\n"
21     << "3. Ubah Peserta\n"
22     << "4. Hapus Peserta\n"
23     << "5. Kembali ke Menu Login\n"
24     << "6. Keluar\n"
25     << "7. Lihat Total Peserta\n"
26     << "8. Ubah Status Peserta\n";
27 }
```

```

void tampilkanMenuUser() {
    cout << "\nMenu User:\n"
        << "1. Tampilkan Peserta\n"
        << "2. Kembali ke Menu Login\n"
        << "3. Keluar\n";
}

// Inisialisasi data
int inisialisasiData(Peserta peserta[], int &jumlahPeserta) {
    Peserta dataAwal[] = {
        {"Kim Sujung", "Korea Selatan", 20, "Dance", "Eliminasi"},
        {"Narai Koko", "Jepang", 19, "Rap", "Lolos"},
        {"Choi Jungeun", "Korea Selatan", 18, "Vokal", "Lolos"},
        {"Nam Yuju", "Korea Selatan", 21, "Rap", "Eliminasi"}
    };
    int jumlahAwal = sizeof(dataAwal) / sizeof(dataAwal[0]);
    for (int i = 0; i < jumlahAwal; i++) {
        peserta[i] = dataAwal[i];
    }
    jumlahPeserta = jumlahAwal;
    return jumlahPeserta;
}

```

```

// Fungsi tambahan dengan address-of operator
void hitungTotalPeserta(int &jumlah) {
    cout << "\nTotal peserta saat ini adalah: " << jumlah << endl;
}

// Fungsi tambahan dengan dereference operator
void ubahStatus(Peserta *p, string statusBaru) {
    p->status = statusBaru;
    cout << "Status peserta berhasil diubah menjadi: " << p->status << endl;
}

```

```

// Login
int login(int &role) {
    string username, nim;
    int attempts = 0;
    while (attempts < 3) {
        cout << "\nLogin Sistem\nMasukkan Nama: ";
        cin >> username;
        cout << "Masukkan NIM: ";
        cin >> nim;

        if (nim.length() == 3 && username != "admin") {
            role = 1;
            return 1;
        } else if (username == "admin" && nim == "000") {
            role = 2;
            return 1;
        }
        cout << "Login gagal! Coba lagi.\n";
        attempts++;
    }
    cout << "Terlalu banyak percobaan. Program berhenti.\n";
    return 0;
}

```

```

// Tampilkan peserta
int tampilkanPeserta(const Peserta peserta[], int jumlahPeserta) {
    if (jumlahPeserta == 0) {
        cout << "Tidak ada data peserta.\n";
        return 0;
    }
    cout << "\nDaftar Peserta I-LAND 2:\n";
    for (int i = 0; i < jumlahPeserta; i++) {
        cout << "Peserta ke-" << i + 1 << "\n"
            << "Nama: " << peserta[i].nama << "\n"
            << "Asal Negara: " << peserta[i].asalNegara << "\n"
            << "Umur: " << peserta[i].umur << "\n"
            << "Posisi: " << peserta[i].posisi << "\n"
            << "Status: " << peserta[i].status << "\n\n";
    }
    return jumlahPeserta;
}

```

```

// Tambah peserta
int tambahPeserta(Peserta peserta[], int &jumlahPeserta) {
    if (jumlahPeserta >= MAX_PESERTA) {
        cout << "Sudah penuh! Tidak bisa menambah peserta lagi.\n";
        return 0;
    }
    cout << "Masukkan Nama: ";
    cin.ignore();
    getline(cin, peserta[jumlahPeserta].nama);
    cout << "Masukkan Asal Negara: ";
    getline(cin, peserta[jumlahPeserta].asalNegara);
    cout << "Masukkan Umur: ";
    cin >> peserta[jumlahPeserta].umur;
    cin.ignore();
    cout << "Masukkan Posisi: ";
    getline(cin, peserta[jumlahPeserta].posisi);
    cout << "Masukkan Status: ";
    getline(cin, peserta[jumlahPeserta].status);

    jumlahPeserta++;
    cout << "Peserta berhasil ditambahkan!\n";
    return jumlahPeserta;
}

```

```

// Ubah peserta
int ubahPeserta(Peserta peserta[], int jumlahPeserta) {
    int index;
    cout << "Masukkan nomor peserta yang akan diubah: ";
    cin >> index;
    cin.ignore();

    if (index < 1 || index > jumlahPeserta) {
        cout << "Nomor peserta tidak valid.\n";
        return 0;
    }

    cout << "Masukkan Nama baru: ";
    getline(cin, peserta[index - 1].nama);
    cout << "Masukkan Asal Negara baru: ";
    getline(cin, peserta[index - 1].asalNegara);
    cout << "Masukkan Umur baru: ";
    cin >> peserta[index - 1].umur;
    cin.ignore();
    cout << "Masukkan Posisi baru: ";
    getline(cin, peserta[index - 1].posisi);
    cout << "Masukkan Status baru: ";
    getline(cin, peserta[index - 1].status);

    cout << "Peserta berhasil diubah!\n";
    return 1;
}

```

```

// Hapus peserta
int hapusPeserta(Peserta peserta[], int &jumlahPeserta) {
    int index;
    cout << "Masukkan nomor peserta yang akan dihapus: ";
    cin >> index;

    if (index < 1 || index > jumlahPeserta) {
        cout << "Nomor peserta tidak valid.\n";
        return 0;
    }

    for (int i = index - 1; i < jumlahPeserta - 1; i++) {
        peserta[i] = peserta[i + 1];
    }
    jumlahPeserta--;
    cout << "Peserta berhasil dihapus!\n";
    return jumlahPeserta;
}

```

```

1 // Main program
2 int main() {
3     Peserta peserta[MAX_PESERTA];
4     int jumlahPeserta = 0;
5     inisialisasiData(peserta, jumlahPeserta);
6
7     int role;
8     do {
9         if (!login(role)) break;
10
11         int pilihan;
12         do {
13             if (role == 2) tampilkanMenuAdmin();
14             else tampilkanMenuUser();
15
16             cout << "Pilihan: ";
17             cin >> pilihan;
18             cin.ignore();
19
20             if (pilihan == 1) {
21                 tampilkanPeserta(peserta, jumlahPeserta);
22             } else if (role == 2 && pilihan == 2) {
23                 tambahPeserta(peserta, jumlahPeserta);
24             } else if (role == 2 && pilihan == 3) {
25                 ubahPeserta(peserta, jumlahPeserta);
26             } else if (role == 2 && pilihan == 4) {
27                 hapusPeserta(peserta, jumlahPeserta);
28             } else if ((role == 2 && pilihan == 5) || (role == 1 && pilihan == 2)) {
29                 break; // kembali ke login
30             } else if ((role == 2 && pilihan == 6) || (role == 1 && pilihan == 3)) {
31                 cout << "Program selesai. Terimakasih banyak :)\n";
32                 return 0;
33             } else if (role == 2 && pilihan == 7) {
34                 hitungTotalPeserta(jumlahPeserta);
35             } else if (role == 2 && pilihan == 8) {
36                 int idx;
37                 cout << "Masukkan nomor peserta yang ingin diubah statusnya: ";
38                 cin >> idx;
39                 cin.ignore();
40                 if (idx >= 1 && idx <= jumlahPeserta) {
41                     string statusBaru;
42                     cout << "Masukkan status baru: ";
43                     getline(cin, statusBaru);
44                     ubahStatus(&peserta[idx - 1], statusBaru);
45                 } else {
46                     cout << "Nomor peserta tidak valid.\n";
47                 }
48             } else {
49                 cout << "Pilihan tidak valid.\n";
50             }
51         } while (true);
52     } while (true);
53
54     return 0;
55 }
56

```

#### 4. Langkah-Langkah Git pada VSCode

```
PS C:\praktikum-apl> git init
Reinitialized existing Git repository in C:/praktikum-apl/.git/
PS C:\praktikum-apl> git add.
git: 'add.' is not a git command. See 'git --help'.

The most similar command is
    add
PS C:\praktikum-apl> git add .
PS C:\praktikum-apl> git commit -m "posttest 5 selesai"
[main 9fa827e] posttest 5 selesai
5 files changed, 319 insertions(+)
create mode 100644 kelas/2409106045-AvellaSalsabellaHermawan-Praktikum-5
create mode 100644 post-test/post-test-5/2409106045-AvellaSalsabellaHemawan-PT-5.pdf
create mode 100644 post-test/post-test-5/2409106045-AvellaSalsabellaHermawan-PT-5.cpp
create mode 100644 post-test/post-test-5/2409106045-AvellaSalsabellaHermawan-PT-5.exe
PS C:\praktikum-apl> git push -u origin main
Enumerating objects: 16, done.
Counting objects: 100% (16/16), done.
Delta compression using up to 8 threads
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 1.40 MiB | 561.00 KiB/s, done.
Total 11 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/avellash/praktikum-apl.git
6dd2e4e..9fa827e  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\praktikum-apl> █
```