

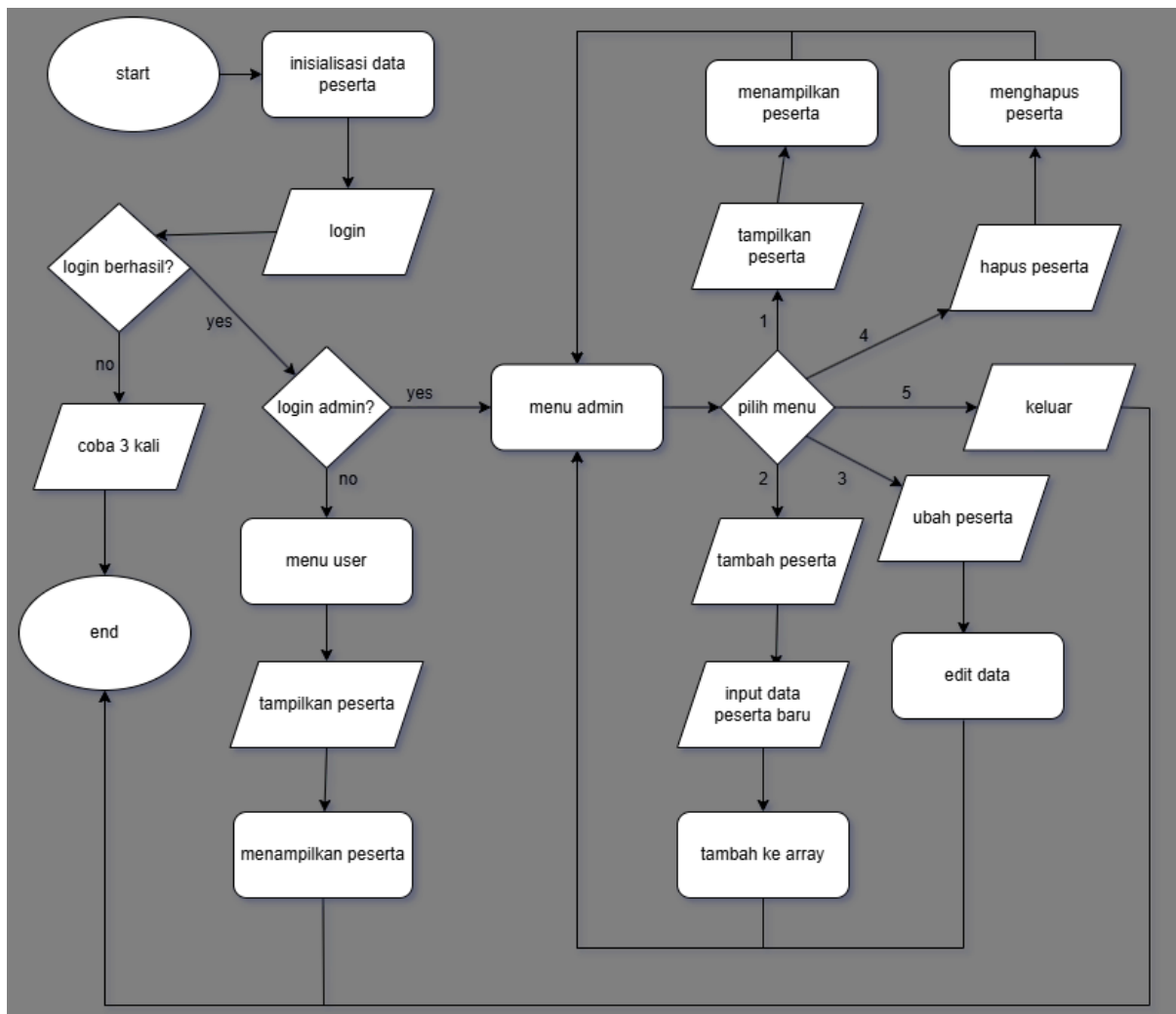
LAPORAN PRAKTIKUM
POSTTEST (4)
ALGORITMA PEMROGRAMAN LANJUT



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

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

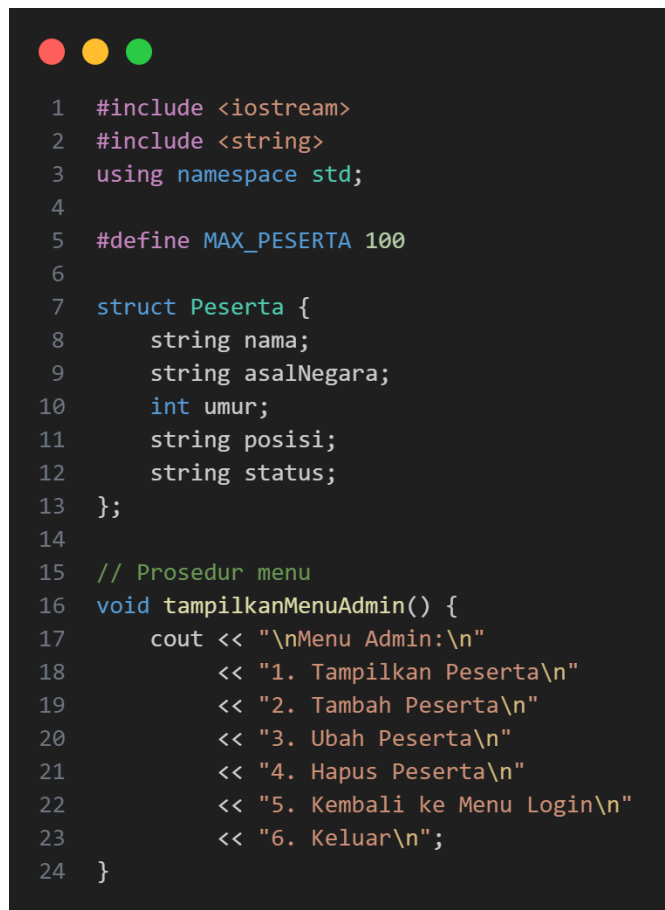
Program ini merupakan aplikasi berbasis bahasa pemrograman C++ yang dirancang untuk mengelola data peserta dalam sebuah sistem seleksi. Program dilengkapi dengan sistem login berbasis peran (role) yang membedakan akses antara admin dan user. Pengguna harus melakukan login terlebih dahulu. Jika pengguna 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. Data peserta disimpan dalam array bertipe struct dengan nama Peserta, yang memuat informasi seperti nama, asal negara, umur, posisi, dan status (lolos/eliminasi).

Program ini menggunakan memisahkan setiap fitur menu ke dalam subprogram. Fungsi-fungsi yang digunakan memakai parameter. Selain itu, program ini juga memanfaatkan prosedur.

3. Screenshoot Program



```
1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  #define MAX_PESERTA 100
6
7  struct Peserta {
8      string nama;
9      string asalNegara;
10     int umur;
11     string posisi;
12     string status;
13 };
14
15 // Prosedur menu
16 void tampilkanMenuAdmin() {
17     cout << "\nMenu Admin:\n"
18         << "1. Tampilkan Peserta\n"
19         << "2. Tambah Peserta\n"
20         << "3. Ubah Peserta\n"
21         << "4. Hapus Peserta\n"
22         << "5. Kembali ke Menu Login\n"
23         << "6. Keluar\n";
24 }
```

```

1 void tampilkanMenuUser() {
2     cout << "\nMenu User:\n"
3         << "1. Tampilkan Peserta\n"
4         << "2. Kembali ke Menu Login\n"
5         << "3. Keluar\n";
6 }
7
8 // Inisialisasi data
9 int inisialisasiData(Peserta peserta[], int &jumlahPeserta) {
10     Peserta dataAwal[] = {
11         {"Kim Sujung", "Korea Selatan", 20, "Dance", "Eliminasi"},
12         {"Narai Koko", "Jepang", 19, "Rap", "Lolos"},
13         {"Choi Jungeun", "Korea Selatan", 18, "Vokal", "Lolos"},
14         {"Nam Yuju", "Korea Selatan", 21, "Rap", "Eliminasi"}
15     };
16     int jumlahAwal = sizeof(dataAwal) / sizeof(dataAwal[0]);
17     for (int i = 0; i < jumlahAwal; i++) {
18         peserta[i] = dataAwal[i];
19     }
20     jumlahPeserta = jumlahAwal;
21     return jumlahPeserta;
22 }

```

```

1 // Login
2 int login(int &role) {
3     string username, nim;
4     int attempts = 0;
5     while (attempts < 3) {
6         cout << "\nLogin Sistem\nMasukkan Nama: ";
7         cin >> username;
8         cout << "Masukkan NIM: ";
9         cin >> nim;
10
11         if (nim.length() == 3 && username != "admin") {
12             role = 1;
13             return 1;
14         } else if (username == "admin" && nim == "000") {
15             role = 2;
16             return 1;
17         }
18         cout << "Login gagal! Coba lagi.\n";
19         attempts++;
20     }
21     cout << "Terlalu banyak percobaan. Program berhenti.\n";
22     return 0;
23 }

```

```

1 // Tampilkan peserta
2 int tampilkanPeserta(const Peserta peserta[], int jumlahPeserta) {
3     if (jumlahPeserta == 0) {
4         cout << "Tidak ada data peserta.\n";
5         return 0;
6     }
7     cout << "\nDaftar Peserta I-LAND 2:\n";
8     for (int i = 0; i < jumlahPeserta; i++) {
9         cout << "Peserta ke-" << i + 1 << "\n"
10             << "Nama: " << peserta[i].nama << "\n"
11             << "Asal Negara: " << peserta[i].asalNegara << "\n"
12             << "Umur: " << peserta[i].umur << "\n"
13             << "Posisi: " << peserta[i].posisi << "\n"
14             << "Status: " << peserta[i].status << "\n\n";
15     }
16     return jumlahPeserta;
17 }

```

```

1 // Tambah peserta
2 int tambahPeserta(Peserta peserta[], int &jumlahPeserta) {
3     if (jumlahPeserta >= MAX_PESERTA) {
4         cout << "Kapasitas penuh! Tidak bisa menambah peserta lagi.\n";
5         return 0;
6     }
7     cout << "Masukkan Nama: ";
8     cin.ignore();
9     getline(cin, peserta[jumlahPeserta].nama);
10    cout << "Masukkan Asal Negara: ";
11    getline(cin, peserta[jumlahPeserta].asalNegara);
12    cout << "Masukkan Umur: ";
13    cin >> peserta[jumlahPeserta].umur;
14    cin.ignore();
15    cout << "Masukkan Posisi: ";
16    getline(cin, peserta[jumlahPeserta].posisi);
17    cout << "Masukkan Status: ";
18    getline(cin, peserta[jumlahPeserta].status);
19
20    jumlahPeserta++;
21    cout << "Peserta berhasil ditambahkan!\n";
22    return jumlahPeserta;
23 }
24

```

```

1 // Hapus peserta
2 int hapusPeserta(Peserta peserta[], int &jumlahPeserta) {
3     int index;
4     cout << "Masukkan nomor peserta yang akan dihapus: ";
5     cin >> index;
6
7     if (index < 1 || index > jumlahPeserta) {
8         cout << "Nomor peserta tidak valid.\n";
9         return 0;
10    }
11
12    for (int i = index - 1; i < jumlahPeserta - 1; i++) {
13        peserta[i] = peserta[i + 1];
14    }
15    jumlahPeserta--;
16    cout << "Peserta berhasil dihapus!\n";
17    return jumlahPeserta;
18 }

```

```

1 // Ubah peserta
2 int ubahPeserta(Peserta peserta[], int jumlahPeserta) {
3     int index;
4     cout << "Masukkan nomor peserta yang akan diubah: ";
5     cin >> index;
6     cin.ignore();
7
8     if (index < 1 || index > jumlahPeserta) {
9         cout << "Nomor peserta tidak valid.\n";
10        return 0;
11    }
12
13    cout << "Masukkan Nama baru: ";
14    getline(cin, peserta[index - 1].nama);
15    cout << "Masukkan Asal Negara baru: ";
16    getline(cin, peserta[index - 1].asalNegara);
17    cout << "Masukkan Umur baru: ";
18    cin >> peserta[index - 1].umur;
19    cin.ignore();
20    cout << "Masukkan Posisi baru: ";
21    getline(cin, peserta[index - 1].posisi);
22    cout << "Masukkan Status baru: ";
23    getline(cin, peserta[index - 1].status);
24
25    cout << "Peserta berhasil diubah!\n";
26    return 1;
27 }

```

```

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 {
34                 cout << "Pilihan tidak valid.\n";
35             }
36         } while (true);
37     } while (true);
38
39     return 0;
40 }
41

```

4. Langkah-Langkah Git pada VSCode

(Sertakan tangkapan layar dari langkah-langkah git [lokal] sampai ke github [cloud].)