

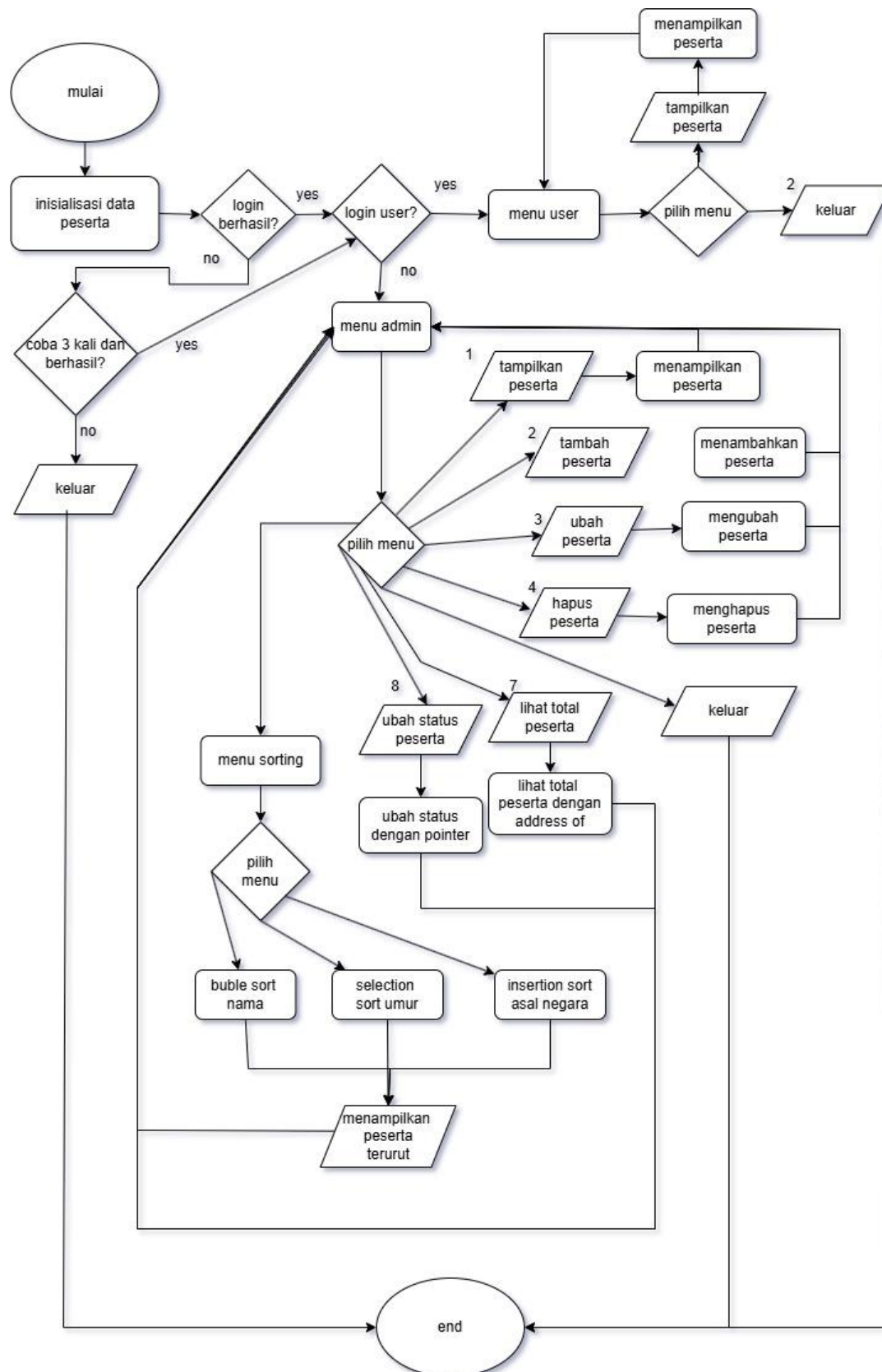
LAPORAN PRAKTIKUM
POSTTEST (6)
ALGORITMA PEMROGRAMAN LANJUT



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.

Program ini juga menggunakan fitur sorting yang memungkinkan admin mengurutkan data peserta berdasarkan kriteria tertentu agar lebih mudah dianalisis atau ditampilkan secara sistematis. Terdapat tiga metode sorting yang saya gunakan:

1. **Sorting Nama (A-Z)** – Menggunakan *Bubble Sort*, mengurutkan nama peserta secara alfabetis dari huruf awal ke akhir.
2. **Sorting Umur (Tertua - Termuda)** – Menggunakan *Selection Sort*, mengurutkan data berdasarkan umur dari yang paling tua ke paling muda.
3. **Sorting Asal Negara (A-Z)** – Menggunakan *Insertion Sort*, mengurutkan peserta berdasarkan nama negara secara alfabetis.

3. Screenshoot Program

```
4. #include <iostream>
5. #include <string>
6. using namespace std;
7.
8. #define MAX_PESERTA 40
9.
10. struct Peserta {
11.     string nama;
12.     string asalNegara;
```

```

13.     int umur;
14.     string posisi;
15.     string status;
16. };
17.
18. // Prosedur menu
19. void tampilkanMenuAdmin() {
20.     cout << "\nMenu Admin:\n"
21.         << "1. Tampilkan Peserta\n"
22.         << "2. Tambah Peserta\n"
23.         << "3. Ubah Peserta\n"
24.         << "4. Hapus Peserta\n"
25.         << "5. Kembali ke Menu Login\n"
26.         << "6. Keluar\n"
27.         << "7. Lihat Total Peserta\n"
28.         << "8. Ubah Status Peserta\n"
29.         << "9. Menu Sorting\n";
30. }
31.
32. void tampilkanMenuUser() {
33.     cout << "\nMenu User:\n"
34.         << "1. Tampilkan Peserta\n"
35.         << "2. Kembali ke Menu Login\n"
36.         << "3. Keluar\n";
37. }
38.
39. // Inisialisasi data
40. int inisialisasiData(Peserta peserta[], int &jumlahPeserta) {
41.     Peserta dataAwal[] = {
42.         {"Kim Sujung", "Korea Selatan", 20, "Dance", "Eliminasi"},
43.         {"Narai Koko", "Jepang", 19, "Rap", "Lolos"},
44.         {"Choi Jungeun", "Korea Selatan", 18, "Vokal", "Lolos"},
45.         {"Nam Yuju", "Korea Selatan", 21, "Rap", "Eliminasi"}
46.     };
47.     int jumlahAwal = sizeof(dataAwal) / sizeof(dataAwal[0]);
48.     for (int i = 0; i < jumlahAwal; i++) {
49.         peserta[i] = dataAwal[i];
50.     }
51.     jumlahPeserta = jumlahAwal;
52.     return jumlahPeserta;
53. }
54.
55. // Fungsi tambahan dengan address-of operator
56. void hitungTotalPeserta(int &jumlah) {
57.     cout << "\nTotal peserta saat ini adalah: " << jumlah << endl;
58. }
59.
60. // Fungsi tambahan dengan dereference operator

```

```

61. void ubahStatus(Peserta *p, string statusBaru) {
62.     p->status = statusBaru;
63.     cout << "Status peserta berhasil diubah menjadi: " << p->status
<< endl;
64. }
65.
66. // Sorting
67. void bubbleSortNama(Peserta peserta[], int jumlahPeserta) {
68.     for (int i = 0; i < jumlahPeserta - 1; i++) {
69.         for (int j = 0; j < jumlahPeserta - i - 1; j++) {
70.             if (peserta[j].nama > peserta[j + 1].nama) {
71.                 swap(peserta[j], peserta[j + 1]);
72.             }
73.         }
74.     }
75. }
76.
77. void selectionSortUmur(Peserta peserta[], int jumlahPeserta) {
78.     for (int i = 0; i < jumlahPeserta - 1; i++) {
79.         int maxIdx = i;
80.         for (int j = i + 1; j < jumlahPeserta; j++) {
81.             if (peserta[j].umur > peserta[maxIdx].umur) {
82.                 maxIdx = j;
83.             }
84.         }
85.         swap(peserta[i], peserta[maxIdx]);
86.     }
87. }
88.
89. void insertionSortNegara(Peserta peserta[], int jumlahPeserta) {
90.     for (int i = 1; i < jumlahPeserta; i++) {
91.         Peserta key = peserta[i];
92.         int j = i - 1;
93.         while (j >= 0 && peserta[j].asalNegara > key.asalNegara) {
94.             peserta[j + 1] = peserta[j];
95.             j--;
96.         }
97.         peserta[j + 1] = key;
98.     }
99. }
100.
101. // Login
102. int login(int &role) {
103.     string username, nim;
104.     int attempts = 0;
105.     while (attempts < 3) {
106.         cout << "\nLogin Sistem\nMasukkan Nama: ";
107.         cin >> username;

```

```

108.         cout << "Masukkan NIM: ";
109.         cin >> nim;
110.
111.         if (nim.length() == 3 && username != "admin") {
112.             role = 1;
113.             return 1;
114.         } else if (username == "admin" && nim == "000") {
115.             role = 2;
116.             return 1;
117.         }
118.         cout << "Login gagal! Coba lagi.\n";
119.         attempts++;
120.     }
121.     cout << "Terlalu banyak percobaan. Program berhenti.\n";
122.     return 0;
123. }
124.
125. // Tampilkan peserta
126. int tampilkanPeserta(const Peserta peserta[], int jumlahPeserta) {
127.     if (jumlahPeserta == 0) {
128.         cout << "Tidak ada data peserta.\n";
129.         return 0;
130.     }
131.     cout << "\nDaftar Peserta I-LAND 2:\n";
132.     for (int i = 0; i < jumlahPeserta; i++) {
133.         cout << "Peserta ke-" << i + 1 << "\n"
134.             << "Nama: " << peserta[i].nama << "\n"
135.             << "Asal Negara: " << peserta[i].asalNegara << "\n"
136.             << "Umur: " << peserta[i].umur << "\n"
137.             << "Posisi: " << peserta[i].posisi << "\n"
138.             << "Status: " << peserta[i].status << "\n\n";
139.     }
140.     return jumlahPeserta;
141. }
142.
143. // Tambah peserta
144. int tambahPeserta(Peserta peserta[], int &jumlahPeserta) {
145.     if (jumlahPeserta >= MAX_PESERTA) {
146.         cout << "Sudah penuh! Tidak bisa menambah peserta lagi.\n";
147.         return 0;
148.     }
149.     cout << "Masukkan Nama: ";
150.     cin.ignore();
151.     getline(cin, peserta[jumlahPeserta].nama);
152.     cout << "Masukkan Asal Negara: ";
153.     getline(cin, peserta[jumlahPeserta].asalNegara);
154.     cout << "Masukkan Umur: ";
155.     cin >> peserta[jumlahPeserta].umur;

```

```

156.     cin.ignore();
157.     cout << "Masukkan Posisi: ";
158.     getline(cin, peserta[jumlahPeserta].posisi);
159.     cout << "Masukkan Status: ";
160.     getline(cin, peserta[jumlahPeserta].status);
161.
162.     jumlahPeserta++;
163.     cout << "Peserta berhasil ditambahkan!\n";
164.     return jumlahPeserta;
165. }
166.
167. // Ubah peserta
168. int ubahPeserta(Peserta peserta[], int jumlahPeserta) {
169.     int index;
170.     cout << "Masukkan nomor peserta yang akan diubah: ";
171.     cin >> index;
172.     cin.ignore();
173.
174.     if (index < 1 || index > jumlahPeserta) {
175.         cout << "Nomor peserta tidak valid.\n";
176.         return 0;
177.     }
178.
179.     cout << "Masukkan Nama baru: ";
180.     getline(cin, peserta[index - 1].nama);
181.     cout << "Masukkan Asal Negara baru: ";
182.     getline(cin, peserta[index - 1].asalNegara);
183.     cout << "Masukkan Umur baru: ";
184.     cin >> peserta[index - 1].umur;
185.     cin.ignore();
186.     cout << "Masukkan Posisi baru: ";
187.     getline(cin, peserta[index - 1].posisi);
188.     cout << "Masukkan Status baru: ";
189.     getline(cin, peserta[index - 1].status);
190.
191.     cout << "Peserta berhasil diubah!\n";
192.     return 1;
193. }
194.
195. // Hapus peserta
196. int hapusPeserta(Peserta peserta[], int &jumlahPeserta) {
197.     int index;
198.     cout << "Masukkan nomor peserta yang akan dihapus: ";
199.     cin >> index;
200.
201.     if (index < 1 || index > jumlahPeserta) {
202.         cout << "Nomor peserta tidak valid.\n";
203.         return 0;

```

```

204.     }
205.
206.     for (int i = index - 1; i < jumlahPeserta - 1; i++) {
207.         peserta[i] = peserta[i + 1];
208.     }
209.     jumlahPeserta--;
210.     cout << "Peserta berhasil dihapus!\n";
211.     return jumlahPeserta;
212. }
213.
214. // Main program
215. int main() {
216.     Peserta peserta[MAX_PESERTA];
217.     int jumlahPeserta = 0;
218.     inisialisasiData(peserta, jumlahPeserta);
219.
220.     int role;
221.     do {
222.         if (!login(role)) break;
223.
224.         int pilihan;
225.         do {
226.             if (role == 2) tampilkanMenuAdmin();
227.             else tampilkanMenuUser();
228.
229.             cout << "Pilihan: ";
230.             cin >> pilihan;
231.             cin.ignore();
232.
233.             if (pilihan == 1) {
234.                 tampilkanPeserta(peserta, jumlahPeserta);
235.             } else if (role == 2 && pilihan == 2) {
236.                 tambahPeserta(peserta, jumlahPeserta);
237.             } else if (role == 2 && pilihan == 3) {
238.                 ubahPeserta(peserta, jumlahPeserta);
239.             } else if (role == 2 && pilihan == 4) {
240.                 hapusPeserta(peserta, jumlahPeserta);
241.             } else if ((role == 2 && pilihan == 5) || (role == 1
&& pilihan == 2)) {
242.                 break;
243.             } else if ((role == 2 && pilihan == 6) || (role == 1
&& pilihan == 3)) {
244.                 cout << "Program selesai. Terimakasih banyak :)\n";
245.                 return 0;
246.             } else if (role == 2 && pilihan == 7) {
247.                 hitungTotalPeserta(jumlahPeserta);
248.             } else if (role == 2 && pilihan == 8) {
249.                 int idx;

```



```

250.             cout << "Masukkan nomor peserta yang ingin diubah
statusnya: ";
251.             cin >> idx;
252.             cin.ignore();
253.             if (idx >= 1 && idx <= jumlahPeserta) {
254.                 string statusBaru;
255.                 cout << "Masukkan status baru: ";
256.                 getline(cin, statusBaru);
257.                 ubahStatus(&peserta[idx - 1], statusBaru);
258.             } else {
259.                 cout << "Nomor peserta tidak valid.\n";
260.             }
261.         } else if (role == 2 && pilihan == 9) {
262.             int opsiSort;
263.             cout << "\nMenu Sorting:\n"
264.                 << "1. Urutkan Nama (A-Z)\n"
265.                 << "2. Urutkan Umur (Tertua - Termuda)\n"
266.                 << "3. Urutkan Asal Negara (A-Z)\n"
267.                 << "Pilihan: ";
268.             cin >> opsiSort;
269.
270.             if (opsiSort == 1) {
271.                 bubbleSortNama(peserta, jumlahPeserta);
272.                 cout << "Peserta berhasil diurutkan
berdasarkan nama (A-Z).\n";
273.             } else if (opsiSort == 2) {
274.                 selectionSortUmur(peserta, jumlahPeserta);
275.                 cout << "Peserta berhasil diurutkan
berdasarkan umur (Tertua - Termuda).\n";
276.             } else if (opsiSort == 3) {
277.                 insertionSortNegara(peserta, jumlahPeserta);
278.                 cout << "Peserta berhasil diurutkan
berdasarkan asal negara (A-Z).\n";
279.             } else {
280.                 cout << "Pilihan sorting tidak valid.\n";
281.             }
282.         } else {
283.             cout << "Pilihan tidak valid.\n";
284.         }
285.     } while (true);
286. } while (true);
287.
288.     return 0;
289. }

```

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 .
PS C:\praktikum-apl> git commit -m "posttest 6 selesai"
[main f0d28c5] posttest 6 selesai
PS C:\praktikum-apl> git pull
Already up to date.
PS C:\praktikum-apl> git push -u origin main
Enumerating objects: 14, done.
Counting objects: 100% (14/14), done.
Delta compression using up to 8 threads
Compressing objects: 100% (10/10), done.
Writing objects: 100% (10/10), 1.03 MiB | 161.00 KiB/s, done.
Total 10 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/avellash/praktikum-apl.git
   f19e1f5..f0d28c5  main -> main
branch 'main' set up to track 'origin/main'.
PS C:\praktikum-apl> █
```