

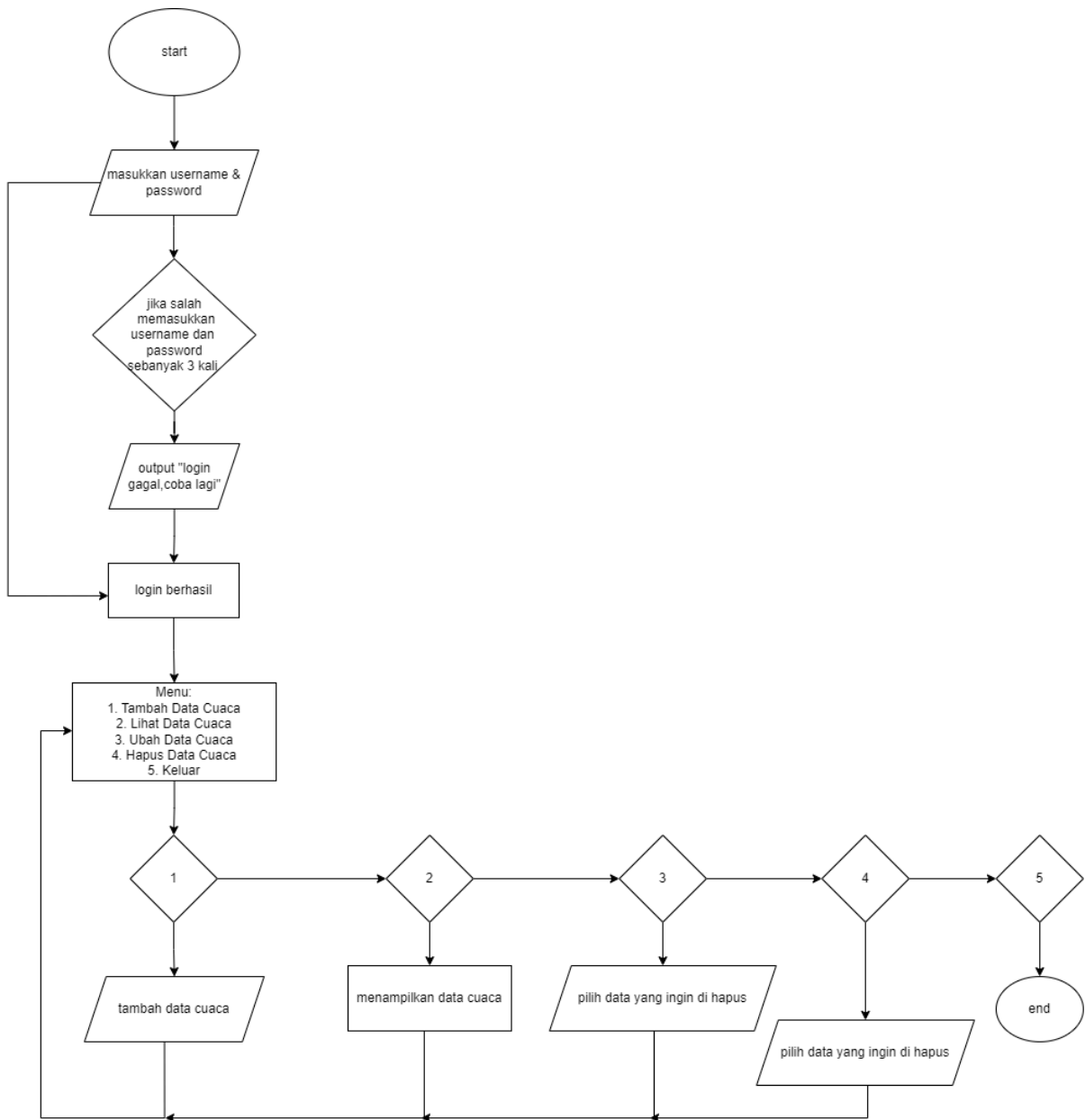
LAPORAN PRAKTIKUM
POSTTEST 4
ALGORITMA PEMROGRAMAN LANJUT



Disusun oleh:
Syalomitha Novindriani Depe (2409106098)
Kelas C'24

PROGRAM STUDI INFORMATIKA
UNIVERSITAS MULAWARMAN
SAMARINDA
2025

1. Flowchart



2. Analisis Program

Program manajemen data cuaca ini sudah memenuhi kebutuhan dasar dengan fitur CRUD (Create, Read, Update, Delete) dan sistem login sederhana. Kelebihannya terletak pada struktur modular dan tampilan data yang terorganisir. Namun, program masih memiliki keterbatasan dalam hal validasi input, penyimpanan data permanen, dan fitur tambahan seperti pencarian/sorting. Untuk pengembangan selanjutnya, disarankan untuk:

- 1. Memperkuat validasi input*
- 2. Menambahkan penyimpanan ke file*
- 3. Mengimplementasikan fitur pencarian/sorting data*
- 4. Memperbaiki user experience*

Program ini layak sebagai basis yang baik, tetapi memerlukan penyempurnaan untuk penggunaan yang lebih profesional.

3. Source Code

```
#include <iostream>

#include <string>

using namespace std;

#define MAX_DATA 100

struct Cuaca {

    string kota;

    float suhu;

    int kelembaban;

};

Cuaca daftarCuaca[MAX_DATA];

int jumlahData = 0;

// prototype fungsi dan prosedur

bool loginSystem();

void displayMainMenu();

void tambahDataCuaca();

string tampilkanDataCuaca();

void ubahDataCuaca();

void hapusDataCuaca();

void keluarProgram();

float hitungRataSuhu();

float hitungRataKelembaban();
```

```

// fungsi login

bool loginSystem() {

    string username, password;

    int attempts = 0;

    while (attempts < 3) {

        cout << "Masukkan Username: ";

        getline(cin, username);

        cout << "Masukkan Password: ";

        getline(cin, password);

        if (username == "Syalomitha Novindriani Depe" && password == "098") {

            return true;

        }

        cout << "Login gagal. Coba lagi.\n";

        attempts++;

    }

    return false;

}

// menambahkan data cuaca

void tambahDataCuaca() {

    if (jumlahData >= MAX_DATA) {

```

```

        cout << "Data sudah penuh!\n";

        return;
    }

    cout << "Masukkan nama kota: ";

    cin.ignore();

    getline(cin, daftarCuaca[jumlahData].kota);

    cout << "Masukkan suhu (°C): ";

    cin >> daftarCuaca[jumlahData].suhu;

    cout << "Masukkan kelembaban (%): ";

    cin >> daftarCuaca[jumlahData].kelembaban;

    jumlahData++;

    cout << "Data berhasil ditambahkan!\n";
}

// menampilkan data cuaca (tanpa rekursi)
string tampilkanDataCuaca() {

    if (jumlahData == 0) {

        return "Belum ada data cuaca.\n";

    }

    string output = "\nDaftar Cuaca:\n";

    output += "=====\n";

    output += "| No | Kota | Suhu (°C) | Kelembaban (%) |\n";

    output += "=====\n";

```

```

// loop biasa
for (int i = 0; i < jumlahData; i++) {
    output += "| " + to_string(i + 1) + " | ";

    output += daftarCuaca[i].kota + string(15 -
daftarCuaca[i].kota.length(), ' ') + "| ";

    output += to_string(daftarCuaca[i].suhu) + " | ";

    output += to_string(daftarCuaca[i].kelembaban) + "% | \n";
}

output += "=====\n";

// menambahkan informasi rata-rata
output += "Rata-rata Suhu: " + to_string(hitungRataSuhu()) + "°C\n";

output += "Rata-rata Kelembaban: " + to_string(hitungRataKelembaban()) +
"%\n";

return output;
}

// menghitung rata-rata suhu
float hitungRataSuhu() {
    if (jumlahData == 0) return 0;

    float total = 0;

    for (int i = 0; i < jumlahData; i++) {
        total += daftarCuaca[i].suhu;
    }
}

```

```

    }

    return total / jumlahData;
}

// menghitung rata-rata kelembaban
float hitungRataKelembaban() {
    if (jumlahData == 0) return 0;

    float total = 0;

    for (int i = 0; i < jumlahData; i++) {
        total += daftarCuaca[i].kelembaban;
    }

    return total / jumlahData;
}

// mengubah data cuaca
void ubahDataCuaca() {
    if (jumlahData == 0) {
        cout << "Belum ada data untuk diubah.\n";
        return;
    }

    int index;

    cout << tampilkanDataCuaca();

    cout << "Masukkan nomor data yang akan diubah: ";

    cin >> index;
}

```



```

        if (index < 1 || index > jumlahData) {

            cout << "Nomor tidak valid!\n";

            return;

        }

        cout << "Masukkan nama kota baru: ";

        cin.ignore();

        getline(cin, daftarCuaca[index - 1].kota);

        cout << "Masukkan suhu baru (\u00B0C): ";

        cin >> daftarCuaca[index - 1].suhu;

        cout << "Masukkan kelembaban baru (%): ";

        cin >> daftarCuaca[index - 1].kelembaban;

        cout << "Data berhasil diubah!\n";

    }

// menghapus data cuaca
void hapusDataCuaca() {

    if (jumlahData == 0) {

        cout << "Belum ada data untuk dihapus.\n";

        return;

    }

    int index;

    cout << tampilkanDataCuaca();

```

```

    cout << "Masukkan nomor data yang akan dihapus: ";

    cin >> index;

    if (index < 1 || index > jumlahData) {

        cout << "Nomor tidak valid!\n";

        return;

    }

    for (int i = index - 1; i < jumlahData - 1; i++) {

        daftarCuaca[i] = daftarCuaca[i + 1];

    }

    jumlahData--;

    cout << "Data berhasil dihapus!\n";

}

// menampilkan menu utama

void displayMainMenu() {

    cout << "\n=====\\n";

    cout << " |          MENU UTAMA          |\\n";

    cout << "=====\\n";

    cout << " | 1 | Tambah Data Cuaca          |\\n";

    cout << " | 2 | Tampilkan Data Cuaca       |\\n";

    cout << " | 3 | Ubah Data Cuaca            |\\n";

    cout << " | 4 | Hapus Data Cuaca           |\\n";

    cout << " | 5 | Keluar                     |\\n";

```

```

        cout << "=====\n";

        cout << "Pilih menu: ";
    }

// keluar program
void keluarProgram() {
    cout << "Keluar dari program.\n";
}

int main() {
    if (!loginSystem()) {
        cout << "Terlalu banyak percobaan gagal. Program berhenti.\n";
        return 0;
    }

    int pilihan;

    do {
        displayMainMenu();

        cin >> pilihan;

        switch (pilihan) {
            case 1:
                tambahDataCuaca();
                break;

            case 2:
                cout << tampilkanDataCuaca();

```

```
        break;

    case 3:

        ubahDataCuaca();

        break;

    case 4:

        hapusDataCuaca();

        break;

    case 5:

        keluarProgram();

        break;

    default:

        cout << "Pilihan tidak valid!\n";

        break;

    }

} while (pilihan != 5);

return 0;

}
```

4. Uji Coba dan Hasil Output

1. Login

```
Masukkan Username: Syalomitha Novindriani Depe  
Masukkan Password: 098
```

Gambar 4.1 Login

2. Menu Utama

```
=====
|                MENU UTAMA                |
=====
| 1 | Tambah Data Cuaca                    |
| 2 | Tampilkan Data Cuaca                  |
| 3 | Ubah Data Cuaca                       |
| 4 | Hapus Data Cuaca                      |
| 5 | Keluar                                |
=====
Pilih menu: █
```

Gambar 4.2 Menu Utama

3. Tambah Data Cuaca

```
=====
|          MENU UTAMA          |
=====
| 1 | Tambah Data Cuaca      |
| 2 | Tampilkan Data Cuaca   |
| 3 | Ubah Data Cuaca        |
| 4 | Hapus Data Cuaca       |
| 5 | Keluar                  |
=====
Pilih menu: 1
Masukkan nama kota: bandung
Masukkan suhu (TC): 26
Masukkan kelembaban (%): 10
Data berhasil ditambahkan!
```

Gambar 4.3 Tambah Data Cuaca

4. Tampilkan Data Cuaca

```
=====
| 1 | Tambah Data Cuaca      |
| 2 | Tampilkan Data Cuaca   |
| 3 | Ubah Data Cuaca        |
| 4 | Hapus Data Cuaca       |
| 5 | Keluar                  |
=====
Pilih menu: 2

Daftar Cuaca:
=====
| No | Kota          | Suhu (TC) | Kelembaban (%) |
=====
| 1  | bandung       | 26.000000  | 10%             |
=====

=====
```

Gambar 4.4 Tampilkan Data Cuaca

5. Ubah Data Cuaca

```
=====
Pilih menu: 3
Masukkan nomor data yang akan diubah: 1
Masukkan nama kota baru: jakarta
Masukkan suhu baru (T C): 30
Masukkan kelembaban baru (%): 15
Data berhasil diubah!
=====
```

Gambar 4.5 Ubah Data Cuaca

6. Hapus Data Cuaca

```
=====
Pilih menu: 4
Masukkan nomor data yang akan dihapus: 1
Data berhasil dihapus!
=====
```

Gambar 4.5 Hapus Data Cuaca

7. Keluar

```
=====
Pilih menu: 5
Keluar dari program.
```

Gambar 4.6 Keluar Dari Program

5. Langkah-Langkah Git pada VSCode

```
MINGW64:c:/Users/ASUS/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-4
ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-4 (main)
$ git add .
ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-4 (main)
$ git commit -m "post-test-4"
[main 7cc9788] post-test-4
3 files changed, 224 insertions(+)
create mode 100644 Post-test/Post-test-4/2409106098-SyalomithaNovindrianiDepe-PT-4.cpp
create mode 100644 Post-test/Post-test-4/2409106098-SyalomithaNovindrianiDepe-PT-4.exe
create mode 100644 Post-test/Post-test-4/2409106098-SyalomithaNovindrianiDepe-PT-4.pdf
ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-4 (main)
$ git push --force origin main
Enumerating objects: 67, done.
Counting objects: 100% (67/67), done.
Delta compression using up to 16 threads
Compressing objects: 100% (43/43), done.
Writing objects: 100% (67/67), 3.35 MiB | 947.00 KiB/s, done.
Total 67 (delta 17), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (17/17), done.
To https://github.com/syalomithaaa/Praktikum-APL.git
+ ce6c79e...7cc9788 main -> main (forced update)
ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-4 (main)
$ |
```

- *git add .*

digunakan untuk menambahkan semua perubahan (file baru, perubahan, dan penghapusan) ke dalam staging area. Ini mempersiapkan perubahan sebelum di-commit.

- *git commit -m "post-test-3"*

menyimpan perubahan ke repository dengan commit dan pesan update

File yang dikomit:

1. *Post-test/Post-test-4/2409106098-SyalomithaNovindrianiDepe-PT-4.cpp*
2. *Post-test/Post-test-4/2409106098-SyalomithaNovindrianiDepe-PT-4.exe*
3. *Post-test/Post-test-4/2409106098-SyalomithaNovindrianiDepe-PT-4.pdf*

- ***git push -force origin main***

memaksa mengunggah branch main ke remote, menggantikan riwayat commit sebelumnya