

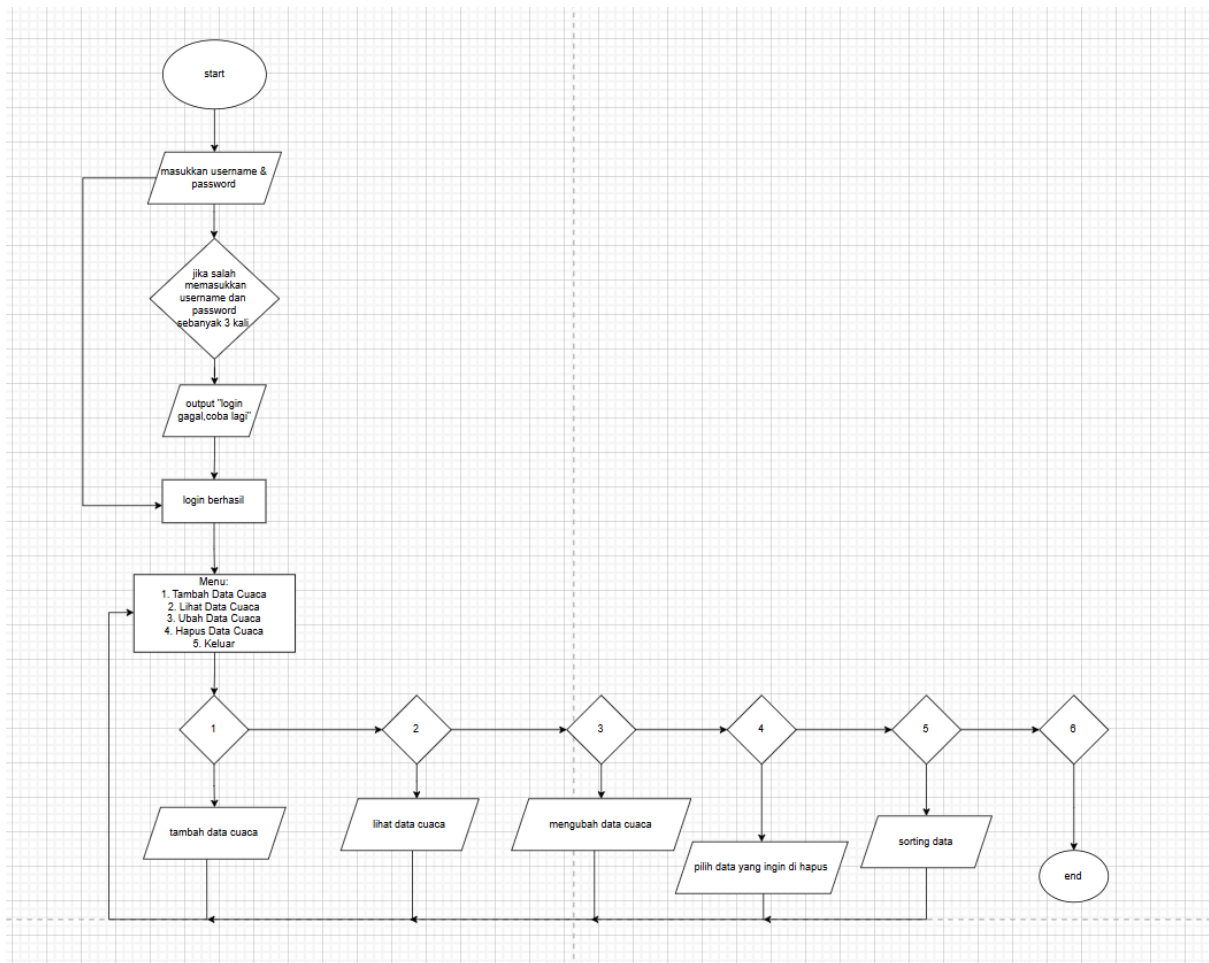
**LAPORAN PRAKTIKUM**  
**POSTTEST 6**  
**ALGORITMA PEMROGRAMAN LANJUT**



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

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

## 1. Flowchart



## **2. Analisis Program**

### **2.1 Deskripsi Singkat Program**

Program ini merupakan implementasi yang baik untuk sistem manajemen data cuaca sederhana dengan fitur dasar yang lengkap. Dengan beberapa perbaikan terutama dalam hal validasi input dan manajemen memori, program ini bisa menjadi lebih robust dan user-friendly. Penggunaan algoritma sorting yang berbeda menunjukkan pemahaman yang baik tentang berbagai teknik pengurutan data.

### 3. Source Code

```
#include <iostream>

#include <string>

#include <algorithm> // untuk swap

using namespace std;

#define MAX_DATA 100

struct Cuaca {

    string kota;

    float suhu;

    int kelembaban;

};

Cuaca daftarCuaca[MAX_DATA];

int jumlahData = 0;

// prototipe pointer

bool loginSystem();

void displayMainMenu();

void tambahDataCuaca(Cuaca *data);

void tampilkanDataCuaca(const Cuaca *data, int size);

void ubahDataCuaca(Cuaca *data, int index);

void hapusDataCuaca(int *size);

float hitungRata(const Cuaca *data, int size, char jenis);
```

```

// sorting baru

void mergeSort(Cuaca *data, int left, int right);

void merge(Cuaca *data, int left, int mid, int right);

void bubbleSortSuhuAsc(Cuaca *data, int size);

void selectionSortKelembabanDesc(Cuaca *data, int size);

void displaySortingMenu();

void sortingData(Cuaca *data, int *size);


// login (ga diubah)

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;
    }

// tambah data cuaca dgn pointer ke array
void tambahDataCuaca(Cuaca *data) {
    if (jumlahData >= MAX_DATA) {
        cout << "Data sudah penuh!\n";
        return;
    }

    cout << "Masukkan nama kota: ";
    cin.ignore();
    getline(cin, data[jumlahData].kota);
    cout << "Masukkan suhu (°C): ";
    cin >> data[jumlahData].suhu;
    cout << "Masukkan kelembaban (%): ";
    cin >> data[jumlahData].kelembaban;

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

// nampilkan data cuaca pointer ke array
void tampilkanDataCuaca(const Cuaca *data, int size) {
    if (size == 0) {

```

```

        cout << "Belum ada data cuaca.\n";

        return;
    }

    cout << "\nDaftar Cuaca:\n";

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

    cout << "| No | Kota                | Suhu (°C) | Kelembaban (%) |\n";

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

    for (int i = 0; i < size; i++) {

        cout << "| " << (i + 1) << " | ";

        cout << data[i].kota << string(15 - data[i].kota.length(), ' ') << "|";

        cout << data[i].suhu << "      | ";

        cout << data[i].kelembaban << "%      |\n";

    }

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

    // nampilkan rata-rata menggunakan pointer

    cout << "Rata-rata Suhu: " << hitungRata(data, size, 's') << "°C\n";

    cout << "Rata-rata Kelembaban: " << hitungRata(data, size, 'k') << "%\n";

}

// ubah data cuaca dgn pointer ke elemen array
void ubahDataCuaca(Cuaca *data, int index) {

    cout << "Masukkan nama kota baru: ";

```

```

        cin.ignore();

        getline(cin, data[index].kota);

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

        cin >> data[index].suhu;

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

        cin >> data[index].kelembaban;

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

// hapus data cuaca dgn pointer ke jumlah data
void hapusDataCuaca(int *size) {

    if (*size == 0) {

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

        return;

    }

    int index;

    tampilkanDataCuaca(daftarCuaca, *size);

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

    cin >> index;

    if (index < 1 || index > *size) {

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

        return;

    }

```



```

        for (int i = index - 1; i < *size - 1; i++) {
            daftarCuaca[i] = daftarCuaca[i + 1];
        }

        (*size)--;

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

// hitung rata-rata (suhu atau kelembaban)
float hitungRata(const Cuaca *data, int size, char jenis) {
    if (size == 0) return 0;

    float total = 0;

    for (int i = 0; i < size; i++) {
        if (jenis == 's') {
            total += data[i].suhu;
        } else {
            total += data[i].kelembaban;
        }
    }

    return total / size;
}

// merge sort untuk sorting huruf (kota) secara descending
void merge(Cuaca *data, int left, int mid, int right) {

```

```

int n1 = mid - left + 1;

int n2 = right - mid;

Cuaca L[n1], R[n2];

for (int i = 0; i < n1; i++)
    L[i] = data[left + i];
for (int j = 0; j < n2; j++)
    R[j] = data[mid + 1 + j];

int i = 0, j = 0, k = left;
while (i < n1 && j < n2) {
    if (L[i].kota.compare(R[j].kota) >= 0) { // descending
        data[k] = L[i];
        i++;
    } else {
        data[k] = R[j];
        j++;
    }
    k++;
}

while (i < n1) {
    data[k] = L[i];
    i++;
    k++;
}

```

```

    }

    while (j < n2) {

        data[k] = R[j];

        j++;

        k++;

    }

}

void mergeSort(Cuaca *data, int left, int right) {

    if (left < right) {

        int mid = left + (right - left) / 2;

        mergeSort(data, left, mid);

        mergeSort(data, mid + 1, right);

        merge(data, left, mid, right);

    }

}

// bubble Sort untuk sorting suhu secara ascending
void bubbleSortSuhuAsc(Cuaca *data, int size) {

    for (int i = 0; i < size - 1; i++) {

        for (int j = 0; j < size - i - 1; j++) {

            if (data[j].suhu > data[j + 1].suhu) {

                swap(data[j], data[j + 1]);

            }

        }

    }

}

```

```

    }
}

// selection sort untuk sorting kelembaban secara descending
void selectionSortKelembabanDesc(Cuaca *data, int size) {
    for (int i = 0; i < size - 1; i++) {
        int max_idx = i;
        for (int j = i + 1; j < size; j++) {
            if (data[j].kelembaban > data[max_idx].kelembaban) {
                max_idx = j;
            }
        }
        if (max_idx != i) {
            swap(data[i], data[max_idx]);
        }
    }
}

// menu sorting
void displaySortingMenu() {
    cout << "\n=====\\n";
    cout << "|          MENU SORTING          |\\n";
    cout << "=====\\n";
    cout << "| 1 | Sort by Kota (Descending) |\\n";
    cout << "| 2 | Sort by Suhu (Ascending)  |\\n";
    cout << "| 3 | Sort by Kelembaban (Desc) |\\n";
}

```

```

        cout << "| 4 | Kembali ke Menu Utama      |\n";

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

        cout << "Pilih menu sorting: ";
    }

// melakukan sorting
void sortingData(Cuaca *data, int *size) {
    int pilihan;

    do {
        displaySortingMenu();

        cin >> pilihan;

        switch (pilihan) {
            case 1:
                mergeSort(data, 0, *size - 1);

                cout << "Data berhasil diurutkan berdasarkan Kota
(Descending)\n";

                tampilkanDataCuaca(data, *size);

                break;

            case 2:
                bubbleSortSuhuAsc(data, *size);

                cout << "Data berhasil diurutkan berdasarkan Suhu
(Ascending)\n";

                tampilkanDataCuaca(data, *size);

                break;

            case 3:
                selectionSortKelembabanDesc(data, *size);

```

```

        cout << "Data berhasil diurutkan berdasarkan Kelembaban
(Descending)\n";

        tampilkanDataCuaca(data, *size);

        break;

    case 4:

        cout << "Kembali ke Menu Utama\n";

        break;

    default:

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

        break;

    }

} while (pilihan != 4);
}

// 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 | Sorting Data Cuaca         | \n";

    cout << " | 6 | Keluar                     | \n";

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

    cout << "Pilih menu: ";

```

```

}

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(daftarCuaca);

                break;

            case 2:

                tampilkanDataCuaca(daftarCuaca, jumlahData);

                break;

            case 3:

                if (jumlahData == 0) {

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

                    break;

                }

                int index;

```

```

        tampilkanDataCuaca(daftarCuaca, jumlahData);

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

        cin >> index;

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

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

            break;

        }

        ubahDataCuaca(daftarCuaca, index-1); // pointer ke elemen

        break;

    case 4:

        hapusDataCuaca(&jumlahData); // pointer ke jumlah data

        break;

    case 5:

        sortingData(daftarCuaca, &jumlahData);

        break;

    case 6:

        cout << "Keluar dari program.\n";

        break;

    default:

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

        break;

    }

} while (pilihan != 6);

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 (T C) | 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. Sorting Data

```
=====
|          MENU SORTING          |
=====
| 1 | Sort by Kota (Descending) |
| 2 | Sort by Suhu (Ascending)  |
| 3 | Sort by Kelembaban (Desc) |
| 4 | Kembali ke Menu Utama     |
=====
Pilih menu sorting: 1
Data berhasil diurutkan berdasarkan Kota (Descending)

Daftar Cuaca:
=====
| No | Kota          | Suhu (°C) | Kelembaban (%) |
=====
| 1  | jogja         | 28        | 15%            |
=====
Rata-rata Suhu: 28°C
```


Gambar 4.6 Sorting Data

8. Keluar

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

Gambar 4.7 Keluar Dari Program

## 5. Langkah-Langkah Git pada VSCode

 MINGW64:/c:/Users/ASUS/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6

```
ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (main)
$ git init
Initialized empty Git repository in C:/Users/ASUS/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6/.git/

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git add .

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git commit -m "Post-test-6"
[master (root-commit) e84629c] Post-test-6
 2 files changed, 336 insertions(+)
 create mode 100644 2409106098-SyalomithaNovindrianiDepe-PT-6.cpp
 create mode 100644 2409106098-SyalomithaNovindrianiDepe-PT-6.exe

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git remote add origin link-repo-git

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'link-repo-git'

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git push origin main
error: src refspec main does not match any
error: failed to push some refs to 'link-repo-git'

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'link-repo-git'

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git add .

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git pull
fatal: 'link-repo-git' does not appear to be a git repository
fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git push -u origin main
error: src refspec main does not match any
error: failed to push some refs to 'link-repo-git'

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ git push
fatal: The current branch master has no upstream branch.
To push the current branch and set the remote as upstream, use

    git push --set-upstream origin master

To have this happen automatically for branches without a tracking
upstream, see 'push.autoSetupRemote' in 'git help config'.

ASUS@LAPTOP-GDSVM82C MINGW64 ~/OneDrive/文档/GITHUB/Praktikum-APL/Post-test/Post-test-6 (master)
$ |
```

- ***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-6"***

*menyimpan perubahan ke repository dengan commit dan pesan update*

*File yang dikomit:*

1. *Post-test/Post-test-6/2409106098-SyalomithaNovindrianiDepe-PT-6.cpp*
2. *Post-test/Post-test-6/2409106098-SyalomithaNovindrianiDepe-PT-6.exe*
3. *Post-test/Post-test-6/2409106098-SyalomithaNovindrianiDepe-PT-6.pdf*

- ***git pull***

*digunakan untuk memperbarui branch lokal dari remote dengan cara menyusun ulang (rebase) commit lokal di atas perubahan terbaru dari remote*

- ***git push origin main***

*digunakan untuk mengunggah perubahan dari branch main di komputer lokal ke repository utama (remote) agar perubahan tersebut tersimpan dan bisa diakses secara online.*