

## **Laporan Akhir**

### ***Sistem Rekap Presensi Mahasiswa***

Dosen Pengampu : Muhammad Resa Arif Yudianto M.Kom.



#### **Kelompok 2**

**Aprilia Candra Puspita**

**24051130025**

**Gayuh Aza**

**24051130032**

**PROGRAM STUDI TEKNOLOGI INFORMASI  
DEPARTEMEN PENDIDIKAN TEKNIK ELEKTRONIKA DAN INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS NEGERI YOGYAKARTA  
26 NOVEMBER 2025**

## DAFTAR ISI

<b>BAB 1</b>	<b>3</b>
1. Topik	3
2. Tujuan	3
3. Skenario Input Dan Output (I/O)	3
4. Sample Data	3
<b>BAB 2</b>	<b>4</b>
1. Flowchart dan Pseudocode	4
2. Link Github	4
<b>BAB 3</b>	<b>4</b>
1. Struktur Program	4
2. Penjelasan Modul	4
3. Cuplikan Kode Program	5
4. Integrasi Antar Modul	11
5. Skenario dan Hasil Testing	11
6. Analisis Hasil Pengujian	12
<b>BAB 4</b>	<b>13</b>
<b>BAB 5</b>	<b>13</b>

# BAB 1

## 1. Topik

Program ini dibuat untuk membantu proses rekapitulasi kehadiran mahasiswa selama satu semester secara otomatis. Semua data presensi akan digabung menjadi satu sistem sehingga dosen atau pihak administrasi dapat dengan mudah melihat total kehadiran setiap mahasiswa, mencari siapa yang memiliki jumlah alpa terbanyak, serta menampilkan detail pertemuan yang tidak dihadiri.

## 2. Tujuan

- a. Merekap data presensi secara efisien dengan struktur array.
- b. Menampilkan data total kehadiran mahasiswa yang meliputi Hadir (H), Izin (I), Sakit (S), dan Alpa (A)
- c. Menampilkan mahasiswa dengan jumlah Alpa (A) dengan bubble sort.

## 3. Skenario Input Dan Output (I/O)

- a. Input
  - i. Input pilihan menu dari pengguna.
  - ii. Input status presensi (Hadir, Izin, Sakit, Alpa) berdasar mahasiswa (NIM).
- b. Output

Menu Pilihan

  - i. Rekap Kehadiran  
Neshia Septiarifa H=2, I=1, S=0, A=1  
Luthfian Afif H=4, I=0, S=0, A=0  
Angger Sekarayu L. B. H=1, I=0, S=1, A=2
  - ii. Urutan Berdasarkan Jumlah Alpa  
Angger Sekarayu, Neshia Septiarifa, Luthfian Afif.
  - iii. Pencarian Mahasiswa Berdasarkan NIM  
24051130003  
Detail Ketidakhadiran: P1=S, P2=A, P3=A


## 4. Sample Data

NO	Nama	NIM
1	Neshia Septiarifa	24051130001
2	Luthfian Afif	24051130002
3	Angger Sekarayu L. B.	24051130003

4	Haidar Tafazul Fikri	24051130004
5	Glenferdinza Aghis A.	24051130005
6	Mulisa Kusuma Wardani	24051130006
7	Kartika Dwi Hapsari	24051130007
8	La D'raz Ar Rauf Mamora	24051130008
9	Elfrina Maya Prayela	24051130009
10	Bayu Arya Wijaya	24051130010

## BAB 2

### 1. Flowchart dan Pseudocode

 Flowchart dan Pseudocode

### 2. Link Github

[https://github.com/azaganteng/sistem\\_presensi\\_mahasiswa.git](https://github.com/azaganteng/sistem_presensi_mahasiswa.git)

## BAB 3

### 1. Struktur Program

Struktur utama program ini mengimplementasikan bahasa C++. Data utama disimpan dalam *Global Array Of Struct*, dan operasi dijalankan dengan melalui fungsi modular.

### 2. Penjelasan Modul

#### a. Struct Data

Data ini menampung NIM, nama, status (untuk 5 pertemuan), dan rekap kehadiran (Hadir, izin, sakit, alpa)

#### b. cariMahasiswa()

Mengimplementasikan Sequential Search untuk mencari data mahasiswa.

#### c. hitungRekap()

Menghitung rekapitulasi presensi mahasiswa selama 5 pertemuan dengan mengakumulasi total H/I/S/A

#### d. bubbleSortAlpa()

Mengimplementasikan BubbleSort untuk mengurutkan mahasiswa mana yang memiliki Alpa terbanyak

e. main()

Sebagai integrasi antar modul dengan menampilkan menu program.

### 3. Cuplikan Kode Program

```
#include <string>
#include <algorithm>
#include <iostream>
#include <iomanip>

using namespace std;

struct Data {
    string nim;
    string nama;
    char status[5];
    int hadir;
    int izin;
    int sakit;
    int alpa;
};

Data mahasiswa[10] = {
    {"24051130001", "Neshia",    {'\0', '\0', '\0', '\0', '\0'}},
    {"24051130002", "Luthfian", {'\0', '\0', '\0', '\0', '\0'}},
    {"24051130003", "Angger",   {'\0', '\0', '\0', '\0', '\0'}},
    {"24051130004", "Haidar",   {'\0', '\0', '\0', '\0', '\0'}},
    {"24051130005", "Glen",     {'\0', '\0', '\0', '\0', '\0'}}}
```

```

        {"24051130006", "Mulisa",      {'\0','\0','\0','\0','\0'}},
        {"24051130007", "Kartika",    {'\0','\0','\0','\0','\0'}},
        {"24051130008", "La D'raz",   {'\0','\0','\0','\0','\0'}},
        {"24051130009", "Elfrina",    {'\0','\0','\0','\0','\0'}},
        {"24051130010", "Bayu",       {'\0','\0','\0','\0','\0'}}
    };

int jumlahMhs = 10;

int cariMahasiswa(string nim) {
    for (int i = 0; i < jumlahMhs; i++) {
        if (mahasiswa[i].nim == nim)
            return i;
    }
    return -1;
}

void inputPresensi() {
    string nim;

    cout << "\nMasukkan NIM: ";
    cin >> nim;

    int idx = cariMahasiswa(nim);
    if (idx == -1) {
        cout << "NIM tidak ditemukan!\n";
        return;
    }

    cout << "Nama : " << mahasiswa[idx].nama << endl;
}

```

```

int hari = -1;

for (int i = 0; i < 5; i++) {

    if (mahasiswa[idx].status[i] == '\0') {

        hari = i;

        break;

    }

}

if (hari == -1) {

    cout << "Presensi sudah penuh 5 pertemuan!\n";

    return;

}

cout << "Mengisi presensi pertemuan ke-" << hari + 1 << endl;

cout << "Masukkan status (H/I/S/A): ";

char s;

cin >> s;

mahasiswa[idx].status[hari] = toupper(s);

cout << "Presensi pertemuan ke-" << hari + 1 << " tersimpan!\n";

}

void hitungRekap(Data mahasiswa[], int n) {

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

        mahasiswa[i].hadir = 0;

        mahasiswa[i].izin = 0;

        mahasiswa[i].sakit = 0;

        mahasiswa[i].alpa = 0;

```

```

        for (int j = 0; j < 5; j++) {
            switch (mahasiswa[i].status[j]) {
                case 'H': mahasiswa[i].hadir++; break;
                case 'I': mahasiswa[i].izin++; break;
                case 'S': mahasiswa[i].sakit++; break;
                case 'A': mahasiswa[i].alpa++; break;
            }
        }
    }
}

void bubbleSortAlpa(Data mahasiswa[], int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (mahasiswa[j].alpa < mahasiswa[j + 1].alpa) {
                swap(mahasiswa[j], mahasiswa[j + 1]);
            }
        }
    }
}

void tampilkanRekap(Data mahasiswa[], int n) {
    hitungRekap(mahasiswa, n);

    cout << "\n=== REKAP PRESENSI MAHASISWA ===\n\n";

    cout << left
        << setw(12) << "NIM"

```



```

        << setw(15) << "Nama"

        << setw(7) << "Hadir"

        << setw(7) << "Sakit"

        << setw(7) << "Izin"

        << setw(7) << "Alpa"

        << endl;

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

    cout << left

        << setw(12) << mahasiswa[i].nim

        << setw(15) << mahasiswa[i].nama

        << setw(7) << mahasiswa[i].hadir

        << setw(7) << mahasiswa[i].sakit

        << setw(7) << mahasiswa[i].izin

        << setw(7) << mahasiswa[i].alpa

        << endl;

}

bubbleSortAlpa(mahasiswa, n);

cout << "\n=== MAHASISWA DENGAN ALPA TERTINGGI ===\n";

cout << "Nama : " << mahasiswa[0].nama << endl;

cout << "NIM : " << mahasiswa[0].nim << endl;

cout << "Alpa : " << mahasiswa[0].alpa << endl;

}

void cariData() {

    string nim;

    cout << "\nMasukkan NIM: ";

```

```

cin >> nim;

int idx = cariMahasiswa(nim);

if (idx == -1) {
    cout << "Mahasiswa tidak ditemukan!\n";
    return;
}

hitungRekap(mahasiswa, jumlahMhs);

cout << "\nData ditemukan!\n";
cout << "Nama : " << mahasiswa[idx].nama << endl;
cout << "Hadir : " << mahasiswa[idx].hadir << endl;
cout << "Sakit : " << mahasiswa[idx].sakit << endl;
cout << "Izin : " << mahasiswa[idx].izin << endl;
cout << "Alpa : " << mahasiswa[idx].alpa << endl;
}

int main() {
    int pilih;

    do {
        cout << "\n=== MENU PRESENSI MAHASISWA ===\n";
        cout << "1. Input Presensi\n";
        cout << "2. Tampilkan Rekap\n";
        cout << "3. Cari Mahasiswa\n";
        cout << "4. Keluar\n";
        cout << "Pilih menu: ";
    }
}

```

```

        cin >> pilih;

        switch (pilih) {

            case 1: inputPresensi(); break;

            case 2: tampilkanRekap(mahasiswa, jumlahMhs); break;

            case 3: cariData(); break;

            case 4: cout << "Program selesai.\n"; break;

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

        }

    } while (pilih != 4);

    return 0;

}

```

#### 4. Integrasi Antar Modul

- inputPresensi()** dan **cariData()** memanggil **cariMahasiswa()**
- tampilkanRekap()** memanggil **hitungRekap()** (Untuk data terbaru) dan **bubbleSortAlpa()** (Untuk urutan alpa tertinggi)
- cariData()** memanggil **hitungRekap()** sebelum menampilkan data spesifik mahasiswa.

#### 5. Skenario dan Hasil Testing

Jenis Uji	Pengujian	Hasil yang diharapkan
Normal	Input data mahasiswa (nama, NIM, dan status kehadiran)	Data mahasiswa berhasil tersimpan pada array
Normal	Menampilkan rekap kehadiran seluruh mahasiswa	Menampilkan daftar mahasiswa dengan total H, I, S, dan A untuk setiap mahasiswa

Normal	Sorting berdasarkan jumlah alpa (descending)	Data mahasiswa terurut dari alpa terbanyak ke tersedikit
Normal	Pencarian mahasiswa berdasarkan NIM	Data mahasiswa yang dicari muncul
Edge	Pencarian mahasiswa dengan NIM yang tidak terdaftar	Menampilkan pesan "Mahasiswa dengan "NIM" tidak ditemukan"
Edge	Input pilihan menu di luar opsi	Menampilkan pesan error "Pilihan tidak valid, silakan pilih menu 1-4
Normal	Keluar dari program (pilih menu 4)	Program berhenti

## 6. Analisis Hasil Pengujian

Hasil pengujian menunjukkan bahwa data mahasiswa dapat diinputkan dan status mahasiswa dapat disimpan dalam array. Untuk dapat menginput status, user terlebih dahulu diminta untuk memasukkan NIM dari data yang ingin dimasukkan. Data mahasiswa kemudian akan dicari dan apabila tidak ditemukan akan muncul pemberitahuan bahwa NIM tidak ditemukan dan dikembalikan ke menu. Namun, apabila NIM ditemukan, data mahasiswa akan ditampilkan dan user diminta untuk menginput status pada pertemuan yang tersedia. Apabila array sudah penuh, input status tidak akan diterima dan akan muncul pemberitahuan bahwa presensi sudah penuh. Hal ini menunjukkan bahwa `inputPresensi()` dapat bekerja dengan baik.

Selanjutnya, rekap mahasiswa juga dapat ditampilkan. Data mahasiswa beserta rekapitulasi status presensi ditampilkan dalam bentuk tabel. Hal ini menunjukkan bahwa `hitungRekap()` dan `tampilkanRekap()` dapat bekerja dengan baik. Data mahasiswa dengan alpa tertinggi juga ditampilkan setelah tabel rekapitulasi presensi. Hal ini menunjukkan bahwa `bubbleSortAlpa()` dapat bekerja dengan baik.

Selanjutnya, mahasiswa dapat dicari berdasarkan NIM mahasiswa. Apabila NIM ditemukan, akan tampil data mahasiswa berupa nama dan rekapitulasi presensi mahasiswa tersebut. Namun, apabila NIM tidak ditemukan, muncul pemberitahuan bahwa NIM tidak ditemukan dan user akan dikembalikan ke menu. Hal ini menunjukkan bahwa `cariMahasiswa()` dan `cariData()` dapat bekerja dengan baik.

Terakhir, apabila input untuk pilihan menu bukanlah salah satu dari pilihan menu, akan muncul pemberitahuan bahwa pilihan tidak valid, serta apabila pilihan menu

untuk keluar diinputkan, maka menu akan keluar dari loop dan program pun berhenti. Hal ini menunjukkan bahwa main() dapat bekerja dengan baik.

## BAB 4

1. Identitas Anggota
  - a. Nama = Aprilia Candra Puspita  
NIM = 24051130025  
Peran = membuat kode
  - b. Nama = Gayuh Aza  
NIM = 24051130032  
Peran = membuat dan menggabungkan kode
2. Tanggung Jawab
  - a. Aprilia Candra Puspita = membuat structData(), membuat hitungRekap(), membuat bubbleSortAlpa(), dan membuat tampilkanRekap()
  - b. Gayuh Aza = membuat cariMahasiswa(), membuat inputPresensi(), membuat cariData(), membuat main(), dan menggabungkan seluruh kode
3. Refleksi Individu
  - a. Aprilia Candra Puspita  
Proyek ini mengajarkan untuk mempelajari bahasa pemrograman C++ dan untuk mencari cara untuk menyelesaikan suatu kasus dengan kode program. Tidak mengalami kendala yang berarti dalam pengerjaannya.
  - b. Gayuh Aza  
Proyek ini mengajarkan untuk mempelajari pengaplikasian dari bubble sort dan bahasa pemrograman C++. Tidak mengalami kendala yang berarti dalam pengerjaannya.

## BAB 5

Proyek ini dapat dibuat dengan baik dan berjalan sesuai dengan rancangan awal yang dibuat meskipun mengalami beberapa perubahan seperti teknis penyimpanan data mahasiswa. Setiap modul dapat dipanggil dan bekerja dengan baik sebagaimana fungsi yang diharapkan. Error maupun input yang tidak sesuai juga diharapkan telah diakomodasi dengan pemberitahuan-pemberitahuan yang diberikan oleh program. Uji program dan analisis hasil uji pun dapat dilakukan dan menunjukkan bahwa program dapat bekerja dan berjalan dengan baik.