

Pengembangan Program Rekapitulasi dan Penilaian Mahasiswa Menggunakan Bahasa C++

Rasyid Indra Cahyo¹

¹Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Dian Nuswanto, Kota Kediri

Artikel Info

Kata kunci:

Pemrograman C++
Nilai Mahasiswa
Predikat
Sorting
File Handling

ABSTRAK

Program ini dikembangkan untuk membantu proses rekapitulasi dan evaluasi hasil belajar mahasiswa secara terstruktur dan otomatis menggunakan bahasa pemrograman C++. Fitur utama meliputi input data mahasiswa, perhitungan nilai akhir, penentuan predikat, status kelulusan, perhitungan rekap nilai, dan penyimpanan ke file. Program juga dilengkapi dengan fitur sorting serta analisis tambahan berupa faktorial dari panjang nama mahasiswa. Pengujian dilakukan dengan pendekatan fungsionalitas langsung pada command-line interface (CLI). Hasil menunjukkan bahwa program ini dapat mempercepat proses evaluasi dan dokumentasi nilai mahasiswa dalam skala kecil hingga menengah.

Penulis Korespondensi :

Rasyid Indra Cahyo,
Program Studi Teknik Informatika
Fakultas Ilmu Komputer
Universitas Dian Nuswantoro, Kediri, 64111
Email: emailpenuliskorepondensi@mail.com

1. PENDAHULUAN

Penilaian akademik merupakan bagian penting dalam sistem pendidikan tinggi karena berfungsi untuk mengukur pencapaian kompetensi mahasiswa selama proses pembelajaran. Umumnya, rekapitulasi nilai masih dilakukan secara manual menggunakan spreadsheet atau aplikasi umum yang tidak dirancang khusus untuk pengelolaan data akademik [1]. Kondisi ini sering kali menimbulkan berbagai permasalahan, seperti kesalahan dalam perhitungan, duplikasi data, serta keterlambatan dalam pelaporan hasil belajar [2], [3].

Seiring dengan meningkatnya jumlah mahasiswa dan kompleksitas data yang harus diolah, dibutuhkan sebuah sistem yang mampu mengotomatisasi proses penilaian agar lebih cepat, akurat, dan efisien. Oleh karena itu, penulis mengembangkan aplikasi berbasis C++ yang dirancang untuk mendukung proses input data mahasiswa, menghitung nilai akhir, menentukan predikat, serta mengevaluasi status kelulusan secara otomatis [4], [5], [6], [7], [8].

Aplikasi ini juga dilengkapi dengan fitur untuk menampilkan hasil evaluasi dalam bentuk tabel dan menyimpannya ke dalam file teks, sehingga mempermudah proses dokumentasi serta pelaporan nilai kepada pihak terkait [3], [9]. Dengan adanya sistem ini, diharapkan pengelolaan nilai akademik dapat dilakukan dengan lebih efektif, mengurangi potensi kesalahan manusia (human error), serta meningkatkan kualitas manajemen data di lingkungan perguruan tinggi [10], [11], [12].

2. METODE

2.1. Desain Program

Program dirancang menggunakan paradigma pemrograman prosedural dalam bahasa C++. Struktur data utama adalah array dari struct Mahasiswa, yang menyimpan data individu mahasiswa [7], [13], [14], [15].



2.1. Fitur Utama

Input Mahasiswa: Menggunakan fungsi inputMahasiswa() untuk membaca nama, NIM, nilai tugas, UTS, dan UAS.

```
void inputMahasiswa(Mahasiswa* m, int n) {
    for (int i = 0; i < n; i++) {
        cout << "\nData Mahasiswa ke-" << i+1 << ":"<\n";
        cout << "Nama: "; cin >> m[i].nama;
        cout << "NIM : "; cin >> m[i].nim;
        cout << "Nilai Tugas: "; cin >> m[i].tugas;
        cout << "Nilai UTS : "; cin >> m[i].uts;
        cout << "Nilai UAS : "; cin >> m[i].uas;

        m[i].nilaiAkhir = hitungNilaiAkhir(m[i].tugas, m[i].uts, m[i].uas);
        m[i].predikat = tentukanPredikat(m[i].nilaiAkhir);
        m[i].lulus = cekKelulusan(m[i].nilaiAkhir);
    }
}
```

Gambar 1. Fungsi Input Mahasiswa

Pengolahan Nilai: Perhitungan nilai akhir dilakukan dengan fungsi hitungNilaiAkhir() dengan bobot: tugas (20%), UTS (30%), dan UAS (50%).

```
float hitungNilaiAkhir(float tugas, float uts, float uas) {
    return (0.2 * tugas) + (0.3 * uts) + (0.5 * uas);
}
```

Gambar 2. Perhitungan Nilai Akhir

Predikat dan Kelulusan: Fungsi tentukanPredikat() mengubah nilai numerik menjadi huruf A-E, sedangkan cekKelulusan() menentukan status lulus.

```
char tentukanPredikat(float nilai) {
    if (nilai >= 85) return 'A';
    else if (nilai >= 75) return 'B';
    else if (nilai >= 60) return 'C';
    else if (nilai >= 50) return 'D';
    else return 'E';
}

bool cekKelulusan(float nilai) {
    return nilai >= 60;
}
```

Gambar 3. Fungsi Tentukan Predikat



Rekapitulasi dan Statistik: Fungsi rekapNilai() menghitung rata-rata setiap komponen nilai.

```
void rekapNilai(Mahasiswa* m, int n) {
    float totalTugas = 0, totalUTS = 0, totalUAS = 0;

    for (int i = 0; i < n; i++) {
        totalTugas += m[i].tugas;
        totalUTS += m[i].uts;
        totalUAS += m[i].uas;
    }

    cout << "\n==== Rekap Nilai ====\n";
    cout << "Total Tugas: " << totalTugas << "\n";
    cout << "Total UTS : " << totalUTS << "\n";
    cout << "Total UAS : " << totalUAS << "\n";

    cout << "Rata-rata Tugas: " << totalTugas / n << "\n";
    cout << "Rata-rata UTS : " << totalUTS / n << "\n";
    cout << "Rata-rata UAS : " << totalUAS / n << "\n";
}
```

Gambar 4. Fungsi Rekap Nilai

Penyimpanan File: Fungsi simpanKeFile() menuliskan seluruh data mahasiswa ke dalam file eksternal.

```
void simpanKeFile(Mahasiswa* m, int n, const string& namaFile) {
    ofstream file(namaFile.c_str());

    if (!file) {
        cerr << "Gagal membuka file untuk menulis.\n";
        return;
    }

    file << fixed << setprecision(2);
    file << "Nama\tNIM\tTugas\tUTS\tUAS\tNilai Akhir\tPredikat\tLulus\n";

    for (int i = 0; i < n; i++) {
        file << m[i].nama << "\t"
            << m[i].nim << "\t"
            << m[i].tugas << "\t"
            << m[i].uts << "\t"
            << m[i].uas << "\t"
            << m[i].nilaiAkhir << "\t\t"
            << m[i].predikat << "\t\t"
            << (m[i].lulus ? "Ya" : "Tidak") << "\n";
    }

    file.close();
    cout << "Data berhasil disimpan ke file \"" << namaFile << "\".\n";
}
```

Gambar 5. Fungsi Simpan File

Sorting dan Analisis Tambahan: Fungsi sorting() mengurutkan data berdasarkan nilai akhir (descending), dan faktorial() menghitung faktorial panjang nama mahasiswa sebagai tambahan logika rekreatif.

```
void sorting(Mahasiswa* m, int n) {
    for (int i = 0; i < n - 1; i++) {
        for (int j = i + 1; j < n; j++) {
            if (m[i].nilaiAkhir < m[j].nilaiAkhir) {
                Mahasiswa temp = m[i];
                m[i] = m[j];
                m[j] = temp;
            }
        }
    }
}
```

```

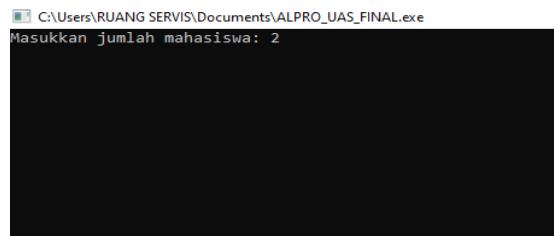
int faktorial(int n) {
    if (n <= 1) return 1;
    return n * faktorial(n - 1);
}

```

Gambar 6. Fungsi Sorting

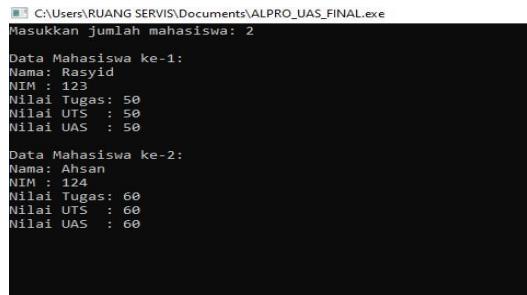
2.3. Alur Program

Meminta input jumlah mahasiswa.



Gambar 7. Input Jumlah Mahasiswa

Melakukan input data mahasiswa.



Gambar 8. Input Data Mahasiswa

Menampilkan dan menyimpan data sebelum dan sesudah sorting.

Gambar 9. Menampilkan Data Mahasiswa Sebelum Sorting

```

Sebelum Diurutkan:
===== Data Mahasiswa =====
Nama      : Rasyid
NIM       : 123
Nilai Akhir: 50 (D) - Tidak Lulus
Faktorial Panjang Nama (6) = 720

Nama      : Ahsan
NIM       : 124
Nilai Akhir: 60 (C) - Lulus
Faktorial Panjang Nama (5) = 120

```

```
Setelah Diurutkan Berdasarkan Nilai Akhir (Descending):

===== Data Mahasiswa =====
Nama      : Ahsan
NIM       : 124
Nilai Akhir: 60 (C) - Lulus
Faktorial Panjang Nama (5) = 120

Nama      : Rasyid
NIM       : 123
Nilai Akhir: 50 (D) - Tidak Lulus
Faktorial Panjang Nama (6) = 720
```

Gambar 10. Menampilkan Data Mahasiswa Sesudah Sorting

Menampilkan rekap nilai dan output di dalam file “data_mahasiswa.txt”.

```
== Rekap Nilai per Mahasiswa ==
Mahasiswa ke-1:
Nama      : Ahsan
NIM       : 124
Nilai Tugas : 60
Nilai UTS   : 60
Nilai UAS   : 60
Nilai Akhir  : 60
Predikat    : C
Kelulusan   : Lulus

Mahasiswa ke-2:
Nama      : Rasyid
NIM       : 123
Nilai Tugas : 50
Nilai UTS   : 50
Nilai UAS   : 50
Nilai Akhir  : 50
Predikat    : D
Kelulusan   : Tidak Lulus
Data berhasil disimpan ke file "data_mahasiswa.txt".
```

Gambar 11. Tampilan Output Data Mahasiswa

Nama	NIM	Tugas	UTS	UAS	Nilai Akhir	Predikat	Lulus
Ahsan	124	60.00	60.00	60.00	60.00	C	Ya
Rasyid	123	50.00	50.00	50.00	50.00	D	Tidak

Gambar 12. Tampilan Data yang Tersimpan dalam File

3. PEMBAHASAN HASIL

Program diuji menggunakan data dummy untuk 3–5 mahasiswa. Hasil menunjukkan:

Program dapat melakukan pengolahan nilai dan predikat dengan benar.

Fungsi sorting mempermudah identifikasi mahasiswa dengan capaian terbaik.

Fitur penyimpanan file menghasilkan laporan data dalam format tabel yang rapi.

Penggunaan faktorial() pada panjang nama mahasiswa menambahkan aspek interaktif dan edukatif dalam logika rekursif.

4. KESIMPULAN

Pengembangan program rekapitulasi dan penilaian mahasiswa berbasis C++ ini berhasil menyediakan fitur-fitur evaluasi akademik yang cepat, akurat, dan terdokumentasi. Program ini cocok digunakan sebagai alat bantu edukasi dalam pengajaran dasar pemrograman, serta dapat dikembangkan lebih lanjut menggunakan GUI atau basis data untuk skala yang lebih besar.



REFERENCES

- [1] R. T. Caesarajmi, J. Raya, T. No, K. Gedong, P. Rebo, and J. Timur, “APLIKASI REKAPITULASI DATA MAHASISWA SEKOLAH VOKASI IPB BERBASIS WEB,” *Jurnal Riset dan Aplikasi Mahasiswa Informatika (JRAMI)*, vol. 01, 2020.
- [2] J. Teknik Informatika, P. Negeri Tanah Laut Jl AYani Km, and P. Tanah Laut Kalimantan Selatan, “Aplikasi Pengelolaan Nilai Akademik Mahasiswa dan DPNA (Daftar Peserta dan Nilai Akhir Herpendi),” 2016.
- [3] E. I. Fenetiruma and J. Lahallo, “RANCANG BANGUN SISTEM PENDATAAN MAHASISWA DAN REKAP NILAI, PRESENSI PRAKTIKUM STIMIK SEPULUH NOPEMBER JAYAPURA BERBASIS WEB Designing Web-based Application of Students Identity, Value Recap, and Attendance of Practice Activities in STIMIK Sepuluh Nopember Jayapura,” 2023.
- [4] R. Amalia and E. D. Putra, “PENGEMBANGAN MODUL MATA KULIAH ALGORITMA DAN PEMROGRAMAN (PEMROGRAMAN C++)”.
- [5] M. Y. Arianti, N. Fitriani, D. Khairani, and S. Tree Adinda, “ANALISIS NILAI AKHIR RAPOR DENGAN PROGRAM C++ SMKISLAMIYAH SEIKAMAHII2021.” [Online]. Available: <http://pkm.uika-bogor.ac.id/index.php/pkm-p/issue/archive>
- [6] I. Ice *et al.*, “PENGENALAN PEMROGRAMAN DASAR DUNIA KODING DENGAN C++.”
- [7] I. Ramadhana and B. Sujatmiko, “PENGEMBANGAN APLIKASI KAMUS BAHASA PEMROGRAMAN C++ BERBASIS ANDROID UNTUK MENINGKATKAN KOMPETENSI KOGNITIF MATA KULIAH STRUKTUR DATA.”
- [8] N. Ariapuri, A. Prasetyo, M. Kom, and I. P. Purbaya, “EFEKTIVITAS APLIKASI KASIR MENGGUNAKAN CODE BLOCKS BERBASIS C++.”
- [9] A. Lubis Ghazali and S. Bunga, “IMPLEMENTASI SISTEM BUSINESS INTELLIGENCE TERHADAP REKAP NILAI PERKULIAHAN MENGGUNAKAN METODE ONLINE ANALYTICAL PROCESSING (OLAP),” 2017.
- [10] L. Penelitian, D. Penerbitan, H. Penelitian, and P. Manufaktur Bandung, “PEMBUATAN PROGRAM SISTEM INFORMASI REKAPITULASI DATA PENDUDUK DESA SUKAPURA BERBASIS WEBSITE GUN GUN MAULANA,” *Ensiklopedia of Journal*, vol. 1, 2019, [Online]. Available: <http://jurnal.ensiklopediaku.org>
- [11] M. Nanak Zakaria, Y. Heru Prasetyo Isnomo, P. Studi Jaringan Telekomunikasi Digital, J. Teknik Elektro, and P. Negeri Malang, “Rancang Bangun Sistem Rekap Nilai Mahasiswa Prodi Jaringan Telekomunikasi Digital Politeknik Negeri Malang Menggunakan Jaringan Fiber Optik Berbasis Website”.
- [12] S. Mauluddin, “SISTEM INFORMASI REKAPITULASI DAFTAR HADIR MAHASISWA (STUDI KASUS : PROGRAM STUDI MANAJEMEN INFORMATIKA).”
- [13] D. Oleh, “PERANCANGAN PROGRAM PENGHITUNG JUMLAH KENDARAAN DI LINTASAN JALAN RAYA SATU ARAH MENGGUNAKAN BAHASA PEMROGRAMAN C++ DENGAN PUSTAKA OPENCV Publikasi Jurnal Skripsi.”
- [14] A. Hendra Saptadi and D. Windi Sari, “ANALISIS ALGORITMA INSERTION SORT, MERGE SORT DAN IMPLEMENTASINYA DALAM BAHASA PEMROGRAMAN C++,” 2012.
- [15] R. W. Arifin and D. Setiyadi, “Algoritma Metode Pengurutan Bubble Sort dan Quick Sort Dalam Bahasa Pemrograman C++,” *INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS*, vol. 4, no. 2, pp. 178–187, 2020.