

Template Lembar Kerja Kelompok

Nama & NPM	Topik:	Tanggal:
Syifa Ariqah Pajrianti (G1F0240009) Anggun Syavira Trinanda (G1F024071) Nafiisah Calista Hersa Agustine (G1F024077)	IF dan SWITCH Java	24 September 2024

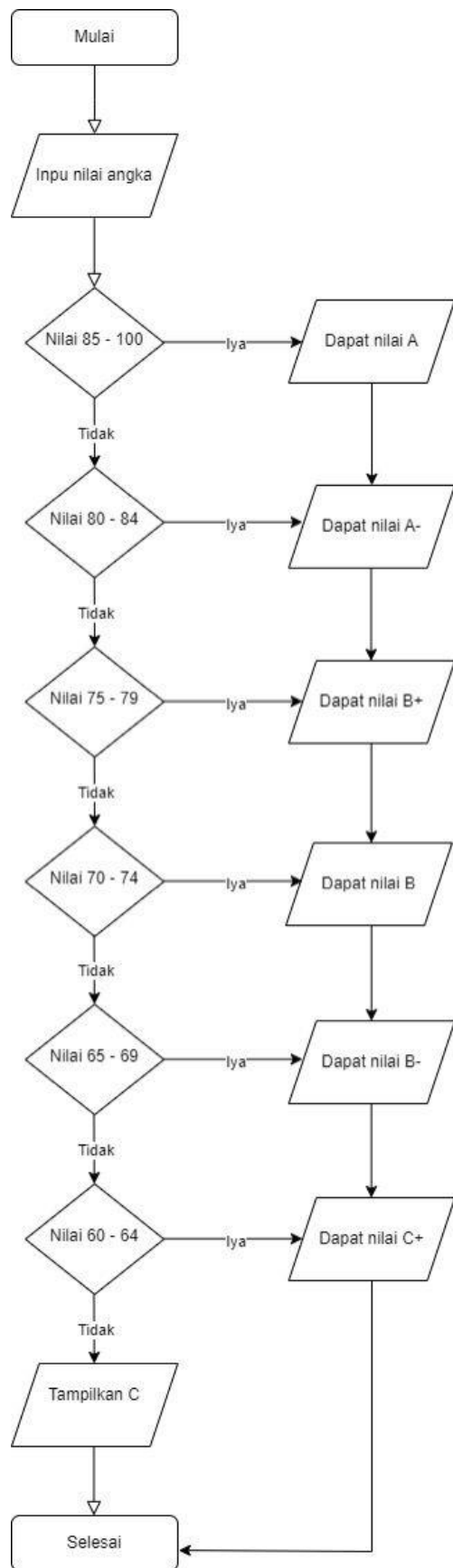
[Nomor 1] Identifikasi Masalah:

1) Uraikan permasalahan dan variabel

Diketahui data penilaian mata kuliah di Universitas Bengkulu sebagai berikut:

Nilai Abjad	Nilai angka Mutu	Rentang Nilai
A	4	85-100
A-	3,75	80-84
B+	3,5	75-79
B	3	70-74
B-	2,75	65-69
C+	2,5	60-64
C	2	1. 59.

- 1.1. Rekomendasikan langkah kerja dan flowchart susunan percabangan sesuai dengan data tersebut.



- 1.2. Desain susunan kode program untuk memeriksa nilai angka yang dimasukkan oleh pengguna ke dalam nilai abjad tertentu.

```

import java.util.Scanner;

public class PercabanganIf {
    public static double nilaiAbjadKeAngka(String nilaiAbjad) {
        switch (nilaiAbjad) {
            case "A":
                return 4.0;
            case "A-":
                return 3.75;
            case "B+":
                return 3.5;
            case "B":
                return 3.0;
            case "B-":
                return 2.75;
            case "C+":
                return 2.5;
            case "C":
                return 2.0;
            default:
                return -1; // Menandakan nilai tidak valid
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        while (true) {
            System.out.print("Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): ");
            String nilaiAbjad = scanner.nextLine().trim();
            double nilaiAngka = nilaiAbjadKeAngka(nilaiAbjad);

            if (nilaiAngka != -1) {
                System.out.printf("Nilai angka untuk %s adalah %.2f%n", nilaiAbjad, nilaiAngka);
            } else {
                System.out.println("Nilai abjad tidak valid.");
            }

            System.out.print("Ingin memasukkan nilai lagi? (ya/tidak): ");
            String lagi = scanner.nextLine().trim().toLowerCase();
            if (!lagi.equals("ya")) {
                break;
            }
        }

        scanner.close();
    }
}

```

2) Rincikan sumber informasi yang relevan (buku / webpage)

Video Materi 1 tentang IF – <https://www.youtube.com/watch?v=G0dfdAFa9iM>

Video Materi 2 tentang SWITCH – <https://www.youtube.com/watch?v=RB4nz4xkisM>

[No 1] Analisis dan Argumentasi

1) Uraikan rancangan solusi yang diusulkan.

Rancangan solusi yang diusulkan dalam kode ini bertujuan untuk mengonversi nilai abjad, seperti A, A-, B+, dan seterusnya, menjadi nilai angka sesuai dengan standar penilaian di universitas. Program ini menggunakan fungsi nilaiAbjadKeAngka, yang memetakan setiap nilai abjad ke nilai angka menggunakan pernyataan switch. Jika nilai yang dimasukkan tidak valid, fungsi ini mengembalikan -1 sebagai indikator kesalahan. Dalam metode main, pengguna diminta untuk memasukkan nilai abjad yang kemudian dibersihkan dari spasi tambahan. Program memanggil fungsi konversi dan memeriksa validitas hasilnya; jika valid, hasil akan dicetak, sementara input yang tidak valid akan diinformasikan kepada pengguna. Untuk meningkatkan pengalaman pengguna, program juga menawarkan opsi untuk memasukkan nilai lagi, sehingga memungkinkan interaksi berulang

2) Analisis solusi, kaitkan dengan permasalahan.

Analisis terhadap solusi ini menunjukkan bahwa pendekatan modular dan terstruktur tidak hanya menjawab permasalahan konversi nilai, tetapi juga mempertimbangkan aspek usability dan maintainability. Dengan memisahkan logika ke dalam fungsi, program menjadi lebih mudah dipelihara, dan penggunaan pernyataan switch menjadikan proses konversi lebih efisien dan jelas. Program ini juga mengatasi masalah kesalahan input dengan memberikan umpan balik yang sesuai kepada pengguna, sehingga mengurangi kebingungan saat memasukkan nilai. Selain itu, opsi untuk memasukkan nilai lagi membuat aplikasi lebih user-friendly, meningkatkan interaksi pengguna secara keseluruhan. Solusi ini tidak hanya efektif dalam konversi nilai, tetapi juga memperhatikan pengalaman pengguna dalam berinteraksi dengan program.

[Nomor 1] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

Kode program (algoritma)

- Mulai
- Deklarasi fungsi
- Input pengguna
- Konversi nilai
- Validasi dan output
- Tanya pengguna
- Tutup scanner
- Selesai

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

Kode di atas merupakan implementasi program Java yang berfungsi untuk mengonversi nilai abjad menjadi nilai angka sesuai dengan standar penilaian di universitas. Di dalamnya, terdapat fungsi `nilaiAbjadKeAngka` yang menggunakan pernyataan `switch` untuk memetakan setiap nilai abjad ke nilai angka yang relevan, dengan pengembalian -1 sebagai indikator jika input tidak valid. Di bagian `main`, program menggunakan `Scanner` untuk membaca input pengguna dan mengulangi proses selama pengguna ingin memasukkan nilai baru. Input dibersihkan dari spasi tambahan menggunakan `trim()`, yang meningkatkan akurasi data. Program kemudian memanggil fungsi konversi dan memeriksa validitas hasilnya, menampilkan nilai angka jika valid atau memberi tahu pengguna jika input tidak sah. Opsi untuk memasukkan nilai lagi menjadikan interaksi pengguna lebih fleksibel dan menyenangkan. Penggunaan struktur modular dan jelas dalam kode ini membuatnya mudah dibaca dan dipelihara, serta memberikan umpan balik yang bermanfaat kepada pengguna saat berinteraksi dengan aplikasi.

b) Uraikan luaran yang dihasilkan

Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): A

Nilai angka untuk A adalah 4.00

Ingin memasukkan nilai lagi? (ya/tidak): YA

Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): A-
 Nilai angka untuk A- adalah 3.75
 Ingin memasukkan nilai lagi? (ya/tidak): YA
 Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): B+
 Nilai angka untuk B+ adalah 3.50
 Ingin memasukkan nilai lagi? (ya/tidak): YA
 Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): B-
 Nilai angka untuk B- adalah 2.75
 Ingin memasukkan nilai lagi? (ya/tidak): YA
 Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): C+
 Nilai angka untuk C+ adalah 2.50
 Ingin memasukkan nilai lagi? (ya/tidak): YA
 Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): C
 Nilai angka untuk C adalah 2.00
 Ingin memasukkan nilai lagi? (ya/tidak): TIDAK

=== Code Execution Successful ===

c) Screenshot/ Capture potongan kode dan hasil luaran

```

1 import java.util.Scanner;
2
3 public class Percobaan17 {
4     public static double nilaiAbjadKeAngka(String nilaiAbjad) {
5         switch (nilaiAbjad) {
6             case "A":
7                 return 4.0;
8             case "A-":
9                 return 3.75;
10            case "B+":
11                return 3.5;
12            case "B":
13                return 3.0;
14            case "B-":
15                return 2.75;
16            case "C+":
17                return 2.5;
18            case "C":
19                return 2.0;
20            default:
21                return -1; // Menandakan nilai tidak valid
22        }
23    }
24
25    public static void main(String[] args) {
26        Scanner scanner = new Scanner(System.in);
27
28        while (true) {
29            System.out.println("Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): ");
30            String nilaiAbjad = scanner.nextLine().trim();
31            double nilaiAngka = nilaiAbjadKeAngka(nilaiAbjad);
32
33            if (nilaiAngka != -1) {
34                System.out.printf("Nilai angka untuk %s adalah %.2f\n", nilaiAbjad, nilaiAngka);
35            } else {
36                System.out.println("Nilai abjad tidak valid.");
37            }
38
39            System.out.print("Ingin memasukkan nilai lagi? (ya/tidak): ");
40            String lagi = scanner.nextLine().trim().toLowerCase();
41            if (!lagi.equals("y")) {
42                break;
43            }
44        }
45
46        scanner.close();
47    }
48 }
  
```

Output

```

Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): A
Nilai angka untuk A adalah 4.00
Ingin memasukkan nilai lagi? (ya/tidak): YA
Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): A-
Nilai angka untuk A- adalah 3.75
Ingin memasukkan nilai lagi? (ya/tidak): YA
Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): B+
Nilai angka untuk B+ adalah 3.50
Ingin memasukkan nilai lagi? (ya/tidak): YA
Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): B-
Nilai angka untuk B- adalah 2.75
Ingin memasukkan nilai lagi? (ya/tidak): YA
Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): C+
Nilai angka untuk C+ adalah 2.50
Ingin memasukkan nilai lagi? (ya/tidak): YA
Masukkan nilai abjad (A, A-, B+, B, B-, C+, C): C
Nilai angka untuk C adalah 2.00
Ingin memasukkan nilai lagi? (ya/tidak): TIDAK

=== Code Execution Successful ===
  
```

[Nomor 1] Kesimpulan

1) Evaluasi

a) Apa konsekuensi dari skenario pemrograman ini?

Skenario pemrograman ini memiliki beberapa konsekuensi penting. Pertama, program sudah mencakup mekanisme validasi input, tetapi pengguna mungkin masih kebingungan jika mereka memasukkan nilai abjad yang tidak terdaftar, sehingga memberikan contoh nilai yang benar bisa meningkatkan kejelasan. Selain itu, dengan menawarkan opsi untuk memasukkan nilai lagi, pengalaman pengguna menjadi lebih interaktif dan fleksibel, meskipun bisa menyebabkan frustrasi jika pengguna terus memasukkan input yang tidak valid. Program akan terus berjalan selama pengguna memilih untuk melakukannya, yang dapat mempengaruhi penggunaan sumber daya. Struktur kode yang modular meningkatkan keterbacaan dan pemeliharaan, tetapi perlu diperhatikan agar tetap terorganisir jika ada penambahan fitur di masa depan. Pengguna masih berpotensi melakukan kesalahan input, sehingga menambahkan fitur untuk validasi dan perbaikan otomatis bisa meningkatkan keandalan. Secara keseluruhan, meskipun program berhasil menjalankan fungsi dasarnya, masih ada beberapa area yang dapat ditingkatkan untuk meningkatkan pengalaman pengguna dan mengurangi kemungkinan kesalahan.

Template Lembar Kerja Kelompok

Nama & NPM	Topik:	Tanggal:																												
Syifa Ariqah Pajrianti (G1F0240009) Anggun Syavira Trinanda (G1F024071) Nafiisah Calista Hersa Agustine (G1F024077)	IF dan SWITCH Java	24 September 2024																												
[Nomor 2] Identifikasi Masalah:																														
<p>1) Uraikan permasalahan dan variabel</p> <p>Konstruksikan kode program dengan IF yang menghitung nilai IPK anda untuk data mata kuliah di semester 1</p> <p>(gunakan data pada tabel dibawah), dengan variasi nilai abjad yang diinput setiap anggota kelompok harus berbeda dan bervariasi (dalam range A hingga C) Contoh MK (kalian dapat mengubah kolom nilai sesuai keinginan masing-masing):</p> <table><tr><th>Nama MK</th><th>SKS</th><th>Nilai</th><th>Contoh Hitung</th></tr><tr><td>Pengantar Teknologi Informasi dan Komunikasi</td><td>2</td><td>A-</td><td>$3.75 * 2 = 7.5$</td></tr><tr><td>Sistem Digital</td><td>3</td><td>C+</td><td>$2.5 * 3 = 7.5$</td></tr><tr><td>Komputer dan Pemrograman</td><td>3</td><td>A</td><td>$4 * 3 = 12$</td></tr><tr><td>Pengantar Sistem multimedia</td><td>2</td><td>B+</td><td>$3.5 * 2 = 7$</td></tr><tr><td>IPK</td><td>10</td><td></td><td>$(7.5 + 7.5 + 12 + 7) / 10$</td></tr><tr><td></td><td></td><td></td><td>IPK = 3.4</td></tr></table> <p>Petunjuk: Rumus perhitungan IPK = (total jumlah (sks * nilai)) / jumlah sks</p> <p>2) Rincikan sumber informasi yang relevan (buku / webpage)</p> <p>Video Materi 1 tentang IF – https://www.youtube.com/watch?v=G0dfdAFa9iM</p> <p>Video Materi 2 tentang SWITCH – https://www.youtube.com/watch?v=RB4nz4xkisM</p>			Nama MK	SKS	Nilai	Contoh Hitung	Pengantar Teknologi Informasi dan Komunikasi	2	A-	$3.75 * 2 = 7.5$	Sistem Digital	3	C+	$2.5 * 3 = 7.5$	Komputer dan Pemrograman	3	A	$4 * 3 = 12$	Pengantar Sistem multimedia	2	B+	$3.5 * 2 = 7$	IPK	10		$(7.5 + 7.5 + 12 + 7) / 10$				IPK = 3.4
Nama MK	SKS	Nilai	Contoh Hitung																											
Pengantar Teknologi Informasi dan Komunikasi	2	A-	$3.75 * 2 = 7.5$																											
Sistem Digital	3	C+	$2.5 * 3 = 7.5$																											
Komputer dan Pemrograman	3	A	$4 * 3 = 12$																											
Pengantar Sistem multimedia	2	B+	$3.5 * 2 = 7$																											
IPK	10		$(7.5 + 7.5 + 12 + 7) / 10$																											
			IPK = 3.4																											
[Nomor 2] Analisis dan Argumentasi																														
<p>1) Uraikan rancangan solusi yang diusulkan.</p> <p>Rancangan solusi untuk program ini bertujuan menghitung Indeks Prestasi Kumulatif (IPK) berdasarkan daftar mata kuliah yang diambil oleh mahasiswa. Pertama, dibuat kelas MataKuliah yang memiliki atribut nama, sks, dan nilai, lengkap dengan konstruktor untuk menginisialisasi nilai-nilai tersebut. Program kemudian menggunakan ArrayList untuk menyimpan beberapa objek MataKuliah, masing-masing mewakili mata kuliah yang diambil. Fungsi nilaiKeNumerik berfungsi untuk mengonversi nilai abjad ke nilai numerik, dengan menggunakan pernyataan if untuk memetakan setiap nilai ke angka yang sesuai. Dalam metode main, total SKS dan total nilai dihitung dengan menjumlahkan SKS dari setiap mata kuliah dan mengalikan SKS dengan nilai numerik. Setelah itu, IPK dihitung dengan membagi total nilai dengan total SKS, dengan pengecekan untuk memastikan bahwa total SKS lebih dari 0. Program kemudian menampilkan total SKS, total nilai, dan IPK yang telah dihitung dalam format dua desimal.</p> <p>2) Analisis solusi, kaitkan dengan permasalahan.</p> <p>Solusi yang diusulkan efektif dalam menghitung IPK, karena strukturnya memisahkan data mata kuliah dan logika perhitungan. Penggunaan ArrayList memungkinkan fleksibilitas dalam menambah atau mengurangi mata kuliah tanpa mengubah struktur dasar program. Fungsi nilaiKeNumerik dengan jelas mendefinisikan cara konversi nilai abjad, yang</p>																														

membuat kode mudah dipahami. Namun, ada beberapa keterbatasan. Misalnya, program ini tidak menangani input dari pengguna, sehingga data mata kuliah yang ada bersifat statis. Ini membatasi kemampuan pengguna untuk memasukkan data sesuai kebutuhan mereka. Selain itu, tidak adanya penanganan untuk nilai yang tidak valid (seperti nilai di luar rentang yang diharapkan) dapat mengakibatkan kesalahan dalam perhitungan. Untuk meningkatkan program, sebaiknya ditambahkan fitur input interaktif dan validasi data untuk meningkatkan keandalan dan fleksibilitasnya. Secara keseluruhan, meskipun solusi ini berfungsi dengan baik, pengembangan lebih lanjut diperlukan untuk mengatasi keterbatasan yang ada.

[Nomor 2] Penyusunan Algoritma dan Kode Program

1) Rancang desain solusi atau algoritma

- mulai
- Definisikan Kelas MataKuliah
- Buat Fungsi untuk Mengonversi Nilai
- Inisialisasi Daftar Mata Kuliah
- Hitung Total SKS dan Total Nilai
- Hitung IPK
- Tampilkan Hasil

2) Tuliskan kode program dan luaran

a) Beri komentar pada kode

Program ini dirancang untuk menghitung Indeks Prestasi Kumulatif (IPK) berdasarkan daftar mata kuliah yang diambil oleh mahasiswa. Kelas `MataKuliah` menyimpan informasi mengenai nama mata kuliah, jumlah SKS (Satuan Kredit Semester), dan nilai yang diperoleh, dengan konstruktor untuk menginisialisasi atribut tersebut. Fungsi `nilaiKeNumerik` mengonversi nilai abjad ke nilai numerik menggunakan serangkaian pernyataan `if`, di mana nilai di bawah C dianggap 0.0. Di dalam metode `main`, program membuat daftar `mataKuliahList` menggunakan `ArrayList`, berisi beberapa objek `MataKuliah` dengan data contoh. Dua variabel, `totalSKS` dan `totalNilai`, diinisialisasi untuk menyimpan total SKS dan nilai yang dihitung. Program melakukan iterasi melalui daftar mata kuliah untuk menghitung total SKS dan total nilai, lalu menghitung IPK dengan membagi `totalNilai` dengan `totalSKS`, dengan pengecekan untuk memastikan `totalSKS` lebih dari 0. Terakhir, program menampilkan total SKS, total nilai, dan IPK yang dihitung dengan format dua desimal. Secara keseluruhan, kode ini terstruktur dengan baik dan efisien dalam menghitung IPK, meskipun ada ruang untuk meningkatkan penanganan nilai tidak valid atau memberikan umpan balik kepada pengguna jika input tidak sesuai.

b) Uraikan luaran yang dihasilkan

Total SKS: 10

Total Nilai: 37.5

IPK: 3.75

=== Code Execution Successful ===

C) Screenshot/ Capture potongan kode dan hasil luaran

```

main.java
import java.util.ArrayList;
import java.util.List;

class MataKuliah {
    String nama;
    int sks;
    String nilai;

    MataKuliah(String nama, int sks, String nilai) {
        this.nama = nama;
        this.sks = sks;
        this.nilai = nilai;
    }
}

public class HitungIPK {

    // Fungsi untuk mengonversi nilai huruf ke nilai numerik
    static double nilaiKeNumerik(String nilai) {
        if (nilai.equals("A")) {
            return 4.0;
        } else if (nilai.equals("A-")) {
            return 3.75;
        } else if (nilai.equals("B+")) {
            return 3.5;
        } else if (nilai.equals("B")) {
            return 3.0;
        } else if (nilai.equals("B-")) {
            return 2.75;
        } else if (nilai.equals("C+")) {
            return 2.5;
        } else if (nilai.equals("C")) {
            return 2.0;
        } else {
            return 0.0; // Nilai di bawah C
        }
    }

    public static void main(String[] args) {
        // Daftar mata kuliah
        List<MataKuliah> mataKuliahList = new ArrayList<>();
        mataKuliahList.add(new MataKuliah("Pengantar Teknologi Informasi dan Komunikasi", 2, "A"));
        mataKuliahList.add(new MataKuliah("Sistem Digital", 2, "B-"));
        mataKuliahList.add(new MataKuliah("Komputer dan Pemrograman", 2, "A"));
        mataKuliahList.add(new MataKuliah("Pengantar Sistem Multimedia", 2, "B-"));

        // Hitung total SKS dan total nilai
        int totalSKS = 0;
        double totalNilai = 0.0;

        // Hitung total SKS dan total nilai
        for (MataKuliah mk : mataKuliahList) {
            totalSKS += mk.sks;
            totalNilai += mk.sks * nilaiKeNumerik(mk.nilai);
        }

        // Hitung IPK
        double ipk = (totalSKS > 0) ? totalNilai / totalSKS : 0.0;

        // Tampilkan hasil
        System.out.println("Total SKS: " + totalSKS);
        System.out.println("Total Nilai: " + totalNilai);
        System.out.printf("IPK: %.2f\n", ipk);
    }
}

```

```

Output

java -cp /tmp/4sfQL80rql/HitungIPK
Total SKS: 10
Total Nilai: 37.5
IPK: 3.75

=== Code Execution Successful ===

```

[Nomor 2] Kesimpulan

- 1) Evaluasi
 - a) Apa konsekuensi dari skenario pemrograman ini?

Konsekuensi dari skenario pemrograman ini mencakup beberapa aspek penting. Pertama, program ini memberikan kemudahan bagi pengguna untuk menghitung Indeks Prestasi Kumulatif (IPK) dengan cara yang terstruktur dan jelas, mengurangi kemungkinan kesalahan manual. Namun, karena data mata kuliah bersifat statis dan tidak ada input dinamis, pengguna tidak dapat menyesuaikan daftar mata kuliah sesuai kebutuhan mereka, yang membatasi fleksibilitas program. Selain itu, tidak adanya penanganan untuk nilai yang tidak valid dapat mengakibatkan kesalahan dalam perhitungan IPK, yang berpotensi menyesatkan pengguna. Program ini juga tidak menangani skenario seperti tahun kabisat atau variasi sistem penilaian yang mungkin ada di institusi pendidikan yang berbeda. Dalam konteks pengembangan perangkat lunak, kurangnya interaktivitas dan validasi input dapat mengurangi pengalaman pengguna secara keseluruhan. Oleh karena itu, meskipun program ini berfungsi untuk tujuan dasarnya, ada kebutuhan mendesak untuk meningkatkan fitur dan penanganan kesalahan agar lebih responsif dan bermanfaat bagi pengguna.

