

LAPORAN AKHIR PBO
KALKULATOR BERBASIS PBO DENGAN FITUR TAMBAHAN
MENGHITUNG IPK



Kelompok 2:

OLEH:

Ferdi Maulana Ikhsan	2310506006	Teknologi Informasi
Ikhsan Budi Wicaksono	2310506007	Teknologi Informasi
Febriani Dewi Alviah	2310506010	Teknologi Informasi
Resnu Mukti Ismail Hanif	2310506012	Teknologi Informasi
Muhammad Ilham Fahrezy	2320506022	Teknologi Informasi

Program Studi S1 Teknologi Informasi Fakultas Teknik
Universitas Tidar
Genap 2024/2025

I. PENDAHULUAN

a. Latar Belakang

Perkembangan teknologi merupakan dan meningkatkan kompleksitas perhitungan pada berbagai bidang, kebutuhan nantinya akan diaplikasikan pada kalkulator yang lebih dari sekedar alat hitung dasar yang mendesak. Pengguna saat ini menginginkan adanya kalkulator yang mampu untuk menangani operasi matematis yang lebih rumit, seperti dengan adanya modulus dan pangkat, serta fitur tambahan untuk menghitung IPK dan menyimpan riwayat perhitungan. Hal seperti ini sangat berguna untuk efisien pada waktu dan akurasi dalam menyelesaikan tugas-tugas akademis maupun profesional.

Pada perancangan aplikasi yang nantinya akan dibuat dengan menerapkan konsep PBO sehingga aplikasi menciptakan struktur kode yang lebih terorganisir dan mudah dipelihara. Sehingga nantinya aplikasi yang dibuat dapat menjadi alat bantu yang handal dalam menyelesaikan berbagai jenis perhitungan.

b. Rumusan Masalah

1. Pada perancangan aplikasi jelaskan diagram kelas yang digunakan pada aplikasi ?
2. Fitur apa saja yang nantinya ada pada aplikasi kalkulator ?

c. Tujuan

1. Memenuhi tugas akhir PBO.
2. Mengimplementasikan konsep pada PBO pada sebuah aplikasi.
3. Mengembangkan struktur kelas yang efisien.
4. Menyediakan fungsionalitas dari aplikasi dengan fitur-fitur yang tersedia.

II. LANDASAN TEORI DAN KODE PROGRAM

a. Dasar Teori

1. Diagram Kelas pada Perancangan Aplikasi Kalkulator

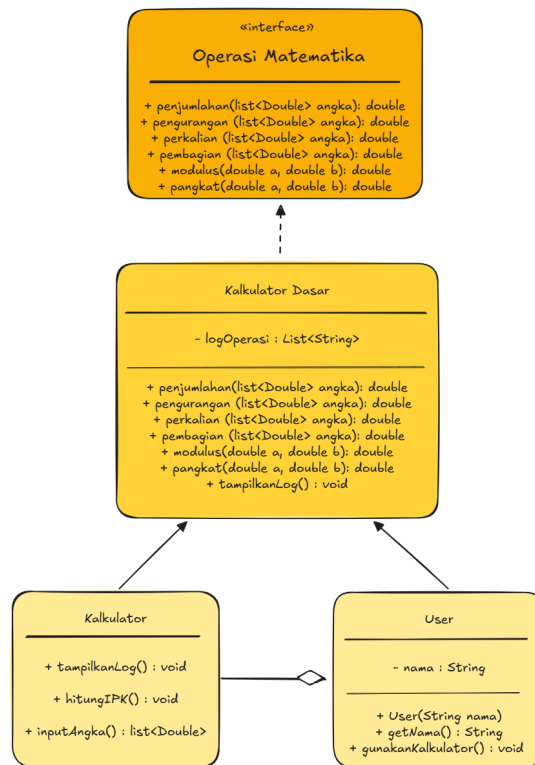


Diagram diatas merupakan diagram kelas UML atau Unified Modeling Language yang menggambarkan struktur kelas-kelas pada suatu program yang berbasis PBO dengan berisi elemen pada diagram seperti berikut

a. Interface

Merupakan kelas yang fungsinya untuk menyediakan deklarasi metode yang harus diimplementasikan pada kelas lain yang mengimolementasikan interface ini. Menggunakan metode yang dideklarasikan yaitu penjumlahan, pengurangan, perkalian, pembagian, modulus, dan pangkat.

b. Kelas KalkulatorDasar

Kelas yang mengimplementasikan interface, sehingga kelas ini wajib untuk menyediakan implementasi semua metod eyang

ada pada interface dengan atribut logOperasi yang berfungsi untuk menyimpan catatan operasi yang dilakukan. Metode yang digunakan adalah semua metode yang ada pada interface dan tampilkanLog().

c. Kelas Kalkulator

Kelas Kalkulator merupakan subclass dari KalkulatorDasar dengan menggunakan metode tampilkanlog(), hitungIPK(), dan inputAngka().

d. Kelas User

Kelas user memiliki atribut nama yang bersifat private dengan kontruktor user(String nama) dengan menggunakan metode getNama() dan gunakanKalkulator().

Hubungan antarkelas yang terjadi pada perancangan aplikasi kalkulator seperti berikut:

a. Implementasi

KalkulatorDasar mengimplementasikan Interface dengan secara tidak langsung wajib menyediakan implementasi untuk metode-metode dalam interface.

b. Pewarisan

Kalkulator merupakan subclass dari KalkulatorDasar sehingga mewarisi atribut dan metode dari kelas induknya.

c. Asosiasi

Kelas User memiliki asosiasi dengan kalkulator ditunjukkan melalui metode gunakanKalkulator().

d. Agregasi

User menggunakan Kalkulator.

2. Fitur Aplikasi Kalkulator

a. Operasi Matematika Dasar

Pada kelas KalkulatorDasar dan diambil dari metode pada interface, kalkulator mendukung operasi berikut:

1. Penjumlahan yaitu menjumlahkan semua angka pada list.
2. Pengurangan yaitu fitur mengurangi semua angka dalam list secara berurutan.
3. Perkalian yaitu fitur mengalikan semua angka pada list.
4. Pembagian merupakan membagi angka-angka pada list dengan cara yang berurutan.
5. Modulus merupakan fitur mengembalikan sisa pembagian dari dua angka.
6. Pangkat merupakan menghitung hasil pemangkatan a dan b.

b. Pencatatan Log Operasi

Berisi fitur pada operasi yang dilakukan akan dicatat pada atribut logOperasi yang berupa list string yang menyimpan informasi tentang operasi yang dilakukan dengan fungsi tambahan pengguna dapat melihat riwayat operasi melalui metode tampilkanLog() pada kelas KalkulatorDasar.

c. Input Data

Fitur untuk memasukkan data dengan metode inputAngka() memungkinkan pengguna memasukkan daftar angka pada bentuk List<Double>.

d. Hitung IPK

Khusus pada kelas Kalkulator menyediakan fitur tambahan berupa metode hitungIPK() untuk menghitung IPK berdasarkan input tertentu.

b. Kode Program

1. Interface OperasiMatematika

```
interface OperasiMatematika {  
    double penjumlahan(List<Double> angka);  
    double pengurangan(List<Double> angka);  
    double perkalian(List<Double> angka);  
    double pembagian(List<Double> angka);  
    double modulus(double a, double b);  
    double pangkat(double a, double b);  
}
```

Interface ini memuat operasi-operasi matematika yang dibutuhkan untuk proyek ini. Operasi-operasi ini adalah operasi dasar matematika yang sering ditemui. Parameter yang dimiliki oleh penjumlahan, pengurangan, perkalian dan pembagian berbentuk List, karena pada proyek ini kami mengharapkan dapat terjadi operasi matematika lebih dari 2 bilangan, sedangkan modulus dan pangkat kami hanya beri 2 parameter saja.

2. Kelas Abstrak KalkulatorDasar

```
abstract class KalkulatorDasar implements OperasiMatematika {
    protected List<String> logOperasi = new ArrayList<>();

    public double penjumlahan(List<Double> angka) {
        double hasil = 0;
        for (double num : angka) {
            hasil += num;
        }
        logOperasi.add("Penjumlahan: " + angka + " = " + hasil);
        return hasil;
    }

    public double pengurangan(List<Double> angka) {
        if (angka.isEmpty()) return 0;
        double hasil = angka.get(0);
        for (int i = 1; i < angka.size(); i++) {
            hasil -= angka.get(i);
        }
        logOperasi.add("Pengurangan: " + angka + " = " + hasil);
        return hasil;
    }

    public double perkalian(List<Double> angka) {
        if (angka.isEmpty()) return 1;
        double hasil = 1;
        for (double num : angka) {
            hasil *= num;
        }
        logOperasi.add("Perkalian: " + angka + " = " + hasil);
        return hasil;
    }

    public double pembagian(List<Double> angka) {
        if (angka.isEmpty()) return 0;
        double hasil = angka.get(0);
        for (int i = 1; i < angka.size(); i++) {
            if (angka.get(i) == 0) {
                logOperasi.add("Pembagian: Error (pembagi tidak boleh nol)");
                throw new ArithmeticException("Pembagi tidak boleh nol");
            }
            hasil /= angka.get(i);
        }
        logOperasi.add("Pembagian: " + angka + " = " + hasil);
        return hasil;
    }

    public double modulus(double a, double b) {
        double hasil = a % b;
        logOperasi.add("Modulus: " + a + " % " + b + " = " + hasil);
        return hasil;
    }

    public double pangkat(double a, double b) {
        double hasil = Math.pow(a, b);
        logOperasi.add("Pangkat: " + a + "^" + b + " = " + hasil);
        return hasil;
    }

    public abstract void tampilkanLog();
}
```

Kelas abstrak ini berfungsi untuk membuat parameter lebih lanjut dari operasi-operasi matematika yang sebelumnya sudah dibuat. Kelas ini juga mengimplementasi interface OperasiMatematika agar dapat digunakan pada kelas ini. Kelas ini selain membuat parameter lebih lanjut untuk operasi matematika, kelas ini juga membuat sebuah atribut baru yaitu LogOperasi yang berfungsi untuk menyimpan history

operasi yang digunakan oleh pengguna dengan menggunakan List dan untuk menampilkannya dengan menggunakan method tampilkanLog.

3. Kelas Kalkulator

```
class Kalkulator extends KalkulatorDasar {

    @Override
    public void tampilkanLog() {
        System.out.println("\nLog Operasi:");
        for (String log : logOperasi) {
            System.out.println(log);
        }
    }

    // Fitur tambahan: Menghitung IPK
    public void hitungIPK() {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Masukkan jumlah mata kuliah: ");
        int jumlahMataKuliah = scanner.nextInt();

        double totalSKS = 0;
        double totalBobot = 0;

        for (int i = 1; i <= jumlahMataKuliah; i++) {
            System.out.println("\nMata Kuliah ke-" + i);
            System.out.print("Masukkan jumlah SKS: ");
            int sks = scanner.nextInt();

            System.out.print("Masukkan nilai (A, B, C, D, E): ");
            char nilai = scanner.next().toUpperCase().charAt(0);

            double bobotNilai;
            switch (nilai) {
                case 'A':
                    bobotNilai = 4.0;
                    break;
                case 'B':
                    bobotNilai = 3.0;
                    break;
                case 'C':
                    bobotNilai = 2.0;
                    break;
                case 'D':
                    bobotNilai = 1.0;
                    break;
                case 'E':
                    bobotNilai = 0.0;
                    break;
                default:
                    System.out.println("Nilai tidak valid, masukkan A, B, C, D, atau E.");
                    i--; // Ulangi input untuk mata kuliah ini
                    continue;
            }

            totalSKS += sks;
            totalBobot += bobotNilai * sks;
        }

        double ipk = totalBobot / totalSKS;
        System.out.println("\nTotal SKS: " + totalSKS);
        System.out.println("Total Bobot: " + totalBobot);
        System.out.printf("IPK Anda adalah: %.2f\n", ipk);

        logOperasi.add("Perhitungan IPK: Total SKS = " + totalSKS + ", Total Bobot = " + totalBobot + ", IPK = " + ipk);
    }

    public List<Double> inputAngka() {
        Scanner scanner = new Scanner(System.in);
        List<Double> angka = new ArrayList<>();
        boolean selesai = false;

        System.out.println("Masukkan angka satu per satu (ketik 'selesai' untuk berhenti):");

        while (!selesai) {
            System.out.print("Masukkan angka: ");
            String input = scanner.nextLine();
            if (input.equalsIgnoreCase("selesai")) {
                selesai = true;
            } else {
                try {
                    double num = Double.parseDouble(input);
                    angka.add(num);
                } catch (NumberFormatException e) {
                    System.out.println("Input tidak valid. Masukkan angka atau ketik 'selesai'.");
                }
            }
        }

        return angka;
    }
}
```

Kelas Kalkulator adalah ini adalah kelas yang digunakan untuk membuat sebuah fitur utama yaitu menghitung IPK. Kelas ini melakukan override tampilkanLog yang berfungsi agar saat user

menggunakan fitur IPK ini nantinya LogOperasi dapat menyimpan rekap penggunaan fitur IPK. Kelas ini juga membuat sebuah case yang berfungsi untuk membulatkan IPK jika A berarti IPKnya 4.0 dan seterusnya. Pada kelas ini juga ada operasi untuk menghitung IPK juga dengan BobotNilai dibagi dengan jumlahSKS.

4. Scanner

```
public List<Double> inputAngka() {
    Scanner scanner = new Scanner(System.in);
    List<Double> angka = new ArrayList<>();
    boolean selesai = false;

    System.out.println("Masukkan angka satu per satu (ketik 'selesai' untuk berhenti:");

    while (!selesai) {
        System.out.print("Masukkan angka: ");
        String input = scanner.nextLine();
        if (input.equalsIgnoreCase("selesai")) {
            selesai = true;
        } else {
            try {
                double num = Double.parseDouble(input);
                angka.add(num);
            } catch (NumberFormatException e) {
                System.out.println("Input tidak valid. Masukkan angka atau ketik 'selesai'.");
            }
        }
    }
    return angka;
}
```

Scanner ini berfungsi untuk melakukan input dari keyboard berbasis DOS. User dapat menginputkan angka dan memilih operasi dengan menggunakan scanner ini.

5. User

```
class User {
    private String nama;

    public User(String nama) {
        this.nama = nama;
    }

    public String getNama() {
        return nama;
    }
}
```

```

public void gunakanKalkulator() {
    Kalkulator kalkulator = new Kalkulator();
    Scanner scanner = new Scanner(System.in);

    System.out.println("Selamat datang, " + nama + "!");

    boolean selesai = false;
    while (!selesai) {
        System.out.println("\nPilih operasi:");
        System.out.println("1. Penjumlahan");
        System.out.println("2. Pengurangan");
        System.out.println("3. Perkalian");
        System.out.println("4. Pembagian");
        System.out.println("5. Modulus");
        System.out.println("6. Pangkat");
        System.out.println("7. Hitung IPK");
        System.out.println("8. Tampilkan Log");
        System.out.println("9. Keluar");
        System.out.print("Pilihan: ");
        int pilihan = scanner.nextInt();

        if (pilihan == 9) {
            selesai = true;
            break;
        }

        if (pilihan >= 1 && pilihan <= 6) {
            if (pilihan == 5 || pilihan == 6) {
                System.out.print("Masukkan angka pertama: ");
                double a = scanner.nextDouble();
                System.out.print("Masukkan angka kedua: ");
                double b = scanner.nextDouble();

                double hasil = (pilihan == 5) ? kalkulator.modulus(a, b) : kalkulator.pangkat(a, b);
                System.out.println("Hasil: " + hasil);
            } else {
                List<Double> angka = kalkulator.inputAngka();
                if (angka.isEmpty()) {
                    System.out.println("Tidak ada angka yang dimasukkan!");
                    continue;
                }

                double hasil = 0;
                switch (pilihan) {
                    case 1:
                        hasil = kalkulator.penjumlahan(angka);
                        break;
                    case 2:
                        hasil = kalkulator.pengurangan(angka);
                        break;
                    case 3:
                        hasil = kalkulator.perkalian(angka);
                        break;
                    case 4:
                        hasil = kalkulator.pembagian(angka);
                        break;
                }
                System.out.println("Hasil: " + hasil);
            }
        } else if (pilihan == 7) {
            kalkulator.hitungIPK();
        } else if (pilihan == 8) {
            kalkulator.tampilkanLog();
        } else {
            System.out.println("Pilihan tidak valid!");
        }
    }
    scanner.close();
}

```

Kelas User ini memiliki atribut baru yaitu nama, gunanya agar pengguna dapat memasukkan namanya. Atribut tersebut menggunakan modifier private untuk melindunginya dan menggunakan getter agar dapat diakses oleh kelas lainnya. Selain berfungsi untuk menambahkan atribut nama, kelas ini berfungsi untuk membuat parameter untuk scanner dengan case seperti 1 sebagai penjumlahan dan parameter

lainnya seperti menghentikan scanner.

6. Kelas Main

```
public class ProjectPBO_Akhir {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
  
        System.out.print("Masukkan nama Anda: ");  
        String nama = scanner.nextLine();  
  
        User user = new User(nama);  
        user.gunakanKalkulator();  
    }  
}
```

Kelas ini berfungsi untuk menjalankan kode program.

c. Output

1. Masukkan Nama

```
Masukkan nama Anda: Resnu|
```

2. Tampilan Awal

```
Selamat datang, Resnu!
```

```
Pilih operasi:
```

```
1. Penjumlahan  
2. Pengurangan  
3. Perkalian  
4. Pembagian  
5. Modulus  
6. Pangkat  
7. Hitung IPK  
8. Tampilkan Log  
9. Keluar  
Pilihan: |
```

3. Penjumlahan

```
Pilihan: 1  
Masukkan angka satu per satu (ketik 'selesai' untuk berhenti):  
Masukkan angka: 1  
Masukkan angka: 2  
Masukkan angka: 3  
Masukkan angka: 4  
Masukkan angka: 5  
Masukkan angka: selesai  
Hasil: 15.0
```

4. Pengurangan

```
Pilih operasi:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Modulus
6. Pangkat
7. Hitung IPK
8. Tampilkan Log
9. Keluar
Pilihan: 2
Masukkan angka satu per satu (ketik 'selesai' untuk berhenti):
Masukkan angka: 5
Masukkan angka: 4
Masukkan angka: 3
Masukkan angka: 2
Masukkan angka: 1
Masukkan angka: selesai
Hasil: -5.0
```

5. Perkalian

```
Pilih operasi:
1. Penjumlahan
2. Pengurangan
3. Perkalian
4. Pembagian
5. Modulus
6. Pangkat
7. Hitung IPK
8. Tampilkan Log
9. Keluar
Pilihan: 3
Masukkan angka satu per satu (ketik 'selesai' untuk berhenti):
Masukkan angka: 1
Masukkan angka: 2
Masukkan angka: 3
Masukkan angka: 4
Masukkan angka: 5
Masukkan angka: selesai
Hasil: 120.0
```

6. Pembagian

```
Pilihan: 4
Masukkan angka satu per satu (ketik 'selesai' untuk berhenti):
Masukkan angka: 0
Masukkan angka: 2
Masukkan angka: selesai
Hasil: 0.0
```

Exception Handling

```
Pilihan: 4
Masukkan angka satu per satu (ketik 'selesai' untuk berhenti):
Masukkan angka: 2
Masukkan angka: 0
Masukkan angka: selesai
Exception in thread "main" java.lang.ArithmeticException: Pembagi tidak boleh nol
```

7. Modulus

```
Pilihan: 5
Masukkan angka pertama: 5
Masukkan angka kedua: 4
Hasil: 1.0
```

8. Pangkat

```
Pilihan: 6
Masukkan angka pertama: 5
Masukkan angka kedua: 4
Hasil: 625.0
```

9. Hitung IPK

```
Pilihan: 7
Masukkan jumlah mata kuliah: 5

Mata Kuliah ke-1
Masukkan jumlah SKS: 4
Masukkan nilai (A, B, C, D, E): A

Mata Kuliah ke-2
Masukkan jumlah SKS: 3
Masukkan nilai (A, B, C, D, E): B

Mata Kuliah ke-3
Masukkan jumlah SKS: 2
Masukkan nilai (A, B, C, D, E): C

Mata Kuliah ke-4
Masukkan jumlah SKS: 1
Masukkan nilai (A, B, C, D, E): D

Mata Kuliah ke-5
Masukkan jumlah SKS: 2
Masukkan nilai (A, B, C, D, E): E

Total SKS: 12.0
Total Bobot: 30.0
IPK Anda adalah: 2,50
```

10. Tampilkan Log

```
Pilihan: 8

Log Operasi:
Modulus:  $5.0 \% 4.0 = 1.0$ 
Pangkat:  $5.0^{4.0} = 625.0$ 
Perhitungan IPK: Total SKS = 12.0, Total Bobot = 30.0, IPK = 2.5
```

11. Keluar

```
Pilih operasi:  
1. Penjumlahan  
2. Pengurangan  
3. Perkalian  
4. Pembagian  
5. Modulus  
6. Pangkat  
7. Hitung IPK  
8. Tampilkan Log  
9. Keluar  
Pilihan: 9
```

```
-----  
BUILD SUCCESS
```

d. Kesimpulan

Penerapan konsep PBO yang dirancang dengan menerapkan konsep PBO dengan baik melalui penggunaan struktur kelas yang terorganisir. Penggunaan inheritance, interface, agregasi, dan komposisi untuk memungkinkan adanya pengembangan aplikasi yang modular dan mudah dipelihara. Fungsionalitas lengkap sehingga menjadikan aplikasi kalkulator ini lebih komprehensif dan memenuhi kebutuhan pengguna di berbagai konteks akademis dan profesional. Riwayat perhitungan yang menjadi nilai tambah aplikasi. Sehingga diharapkan dengan adanya aplikasi ini memiliki kontribusi terhadap proses pembelajaran.

III. DAFTAR PUSTAKA