

LAPORAN TUGAS 3

INHERITANCE

Mata Kuliah : Pemrograman Berbasis Objek

Dosen : Irsyad Arif Mashudi, S.Kom., M.Kom



Ilham Dharma Atmaja

24410702020

Kelas :TI 2D

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG TAHUN 2025

1. Pendahuluan

1.1 Latar Belakang

Perguruan tinggi sebagai institusi pendidikan tinggi memiliki tata tertib yang harus dipatuhi oleh seluruh sivitas akademika, baik mahasiswa maupun dosen. Tata tertib ini dibuat untuk menjaga kedisiplinan, etika, serta menciptakan suasana akademik yang kondusif.

Namun, dalam praktiknya sering terjadi pelanggaran tata tertib yang dilakukan oleh mahasiswa maupun dosen. Oleh karena itu, dibutuhkan suatu sistem yang dapat membantu mencatat, memantau, dan menampilkan riwayat pelanggaran tata tertib di kampus.

Dengan adanya sistem informasi tata tertib kampus yang berbasis Java dan berorientasi objek, diharapkan pencatatan pelanggaran dapat dilakukan dengan lebih terstruktur dan terorganisir.

1.2 Tujuan

Adapun tujuan dari praktikum ini adalah:

1. Mengimplementasikan perancangan **Class Diagram** ke dalam program Java.
2. Menerapkan konsep **Object Oriented Programming (OOP)** terutama **pewarisan (inheritance)** dan **enkapsulasi**.
3. Membuat program sederhana untuk mendukung proses **pencatatan dan pelaporan pelanggaran tata tertib** kampus.
4. Melatih kemampuan analisis dan pemrograman berbasis objek pada kasus nyata.

2. Landasan Teori

2.1 Object Oriented Programming (OOP)

OOP adalah paradigma pemrograman yang berorientasi pada objek. Konsep utama OOP antara lain:

- **Class dan Object:** Class adalah blueprint, sedangkan object adalah instance dari class.
- **Inheritance (Pewarisan):** memungkinkan suatu class mewarisi atribut dan method dari class lain.
- **Encapsulation:** pembungkusan data agar lebih aman dengan mengatur aksesibilitas.
- **Polymorphism:** kemampuan untuk menggunakan satu nama method dengan implementasi berbeda.

2.2 UML Class Diagram

UML (Unified Modeling Language) adalah bahasa standar untuk memodelkan sistem berorientasi objek. **Class Diagram** menggambarkan hubungan antar kelas, atribut, dan method dalam sebuah sistem.

2.3 Java

Java adalah bahasa pemrograman berorientasi objek yang banyak digunakan untuk pengembangan aplikasi. Java memiliki keunggulan multiplatform, library yang lengkap, serta dukungan penuh terhadap paradigma OOP.

3. Analisis dan Perancangan Sistem

3.1 Deskripsi Masalah

Sistem yang akan dibangun harus dapat:

1. Mencatat daftar tata tertib kampus beserta sanksinya.
2. Mencatat pelanggaran yang dilakukan mahasiswa maupun dosen.
3. Menampilkan riwayat pelanggaran dari masing-masing anggota kampus.

3.2 Class Diagram

Berdasarkan analisis, sistem dirancang menggunakan Class Diagram berikut:

- AnggotaKampus (superclass)
- Mahasiswa (subclass AnggotaKampus)
- Dosen (subclass AnggotaKampus)
- TataTertib (menyimpan peraturan dan sanksi)
- Pelanggaran (mencatat pelanggaran oleh anggota kampus)
- SistemInformasiTataTertib (mencatat dan menampilkan riwayat pelanggaran)
- Main (kelas eksekusi)

(Sisipkan gambar Class Diagram sesuai soal)

4. Implementasi Program

4.1 Kode Program

(Lampirkan semua kode program yang sudah aku kasih di jawaban sebelumnya, kelas per kelas).

4.2 Penjelasan Kode

- **AnggotaKampus:** menyimpan data umum (nama, id).
- **Mahasiswa:** subclass dengan atribut tambahan nim dan program studi.
- **Dosen:** subclass dengan atribut tambahan nip dan departemen.
- **TataTertib:** menyimpan peraturan dan sanksi.
- **Pelanggaran:** mencatat pelanggaran yang dilakukan anggota kampus.
- **SistemInformasiTataTertib:** mencatat pelanggaran dan menampilkan riwayat.
- **Main:** berisi contoh implementasi pembuatan objek, pencatatan pelanggaran, dan menampilkan riwayat.

5. Hasil dan Pembahasan

5.1 Verifikasi Program

1. Class AnggotaKampus

```
1 public class AnggotaKampus {
2     protected String nama;
3     protected String id;
4
5     public AnggotaKampus(String nama, String id) {
6         this.nama = nama;
7         this.id = id;
8     }
9
10    public String getNama() {
11        return nama;
12    }
13
14    public String getId() {
15        return id;
16    }
17 }
18
```

2. Class Mahasiswa

```
1 public class Mahasiswa extends AnggotaKampus {
2     private String nim;
3     private String programStudi;
4
5     public Mahasiswa(String nama, String id, String nim, String programStudi) {
6         super(nama, id);
7         this.nim = nim;
8         this.programStudi = programStudi;
9     }
10
11    public String getNim() {
12        return nim;
13    }
14
15    public String getProgramStudi() {
16        return programStudi;
17    }
18 }
19
```

3. Class Dosen

```

1  public class Dosen extends AnggotaKampus {
2      private String nip;
3      private String departemen;
4
5      public Dosen(String nama, String id, String nip, String departemen) {
6          super(nama, id);
7          this.nip = nip;
8          this.departemen = departemen;
9      }
10
11     public String getNip() {
12         return nip;
13     }
14
15     public String getDepartemen() {
16         return departemen;
17     }
18 }
19

```

4. Class Pelanggaran

```

1  public class Pelanggaran {
2      private AnggotaKampus pelanggar;
3      private TataTertib peraturanDilanggat;
4      private String tanggal;
5
6      public Pelanggaran(AnggotaKampus pelanggar, TataTertib peraturanDilanggat, String tanggal) {
7          this.pelanggar = pelanggar;
8          this.peraturanDilanggat = peraturanDilanggat;
9          this.tanggal = tanggal;
10     }
11
12     public AnggotaKampus getPelanggar() {
13         return pelanggar;
14     }
15
16     public TataTertib getPeraturanDilanggat() {
17         return peraturanDilanggat;
18     }
19
20     public String getTanggal() {
21         return tanggal;
22     }
23 }
24

```

5. Class TataTertib

```

1  public class TataTertib {
2      private String peraturan;
3      private String sanksi;
4
5      public TataTertib(String peraturan, String sanksi) {
6          this.peraturan = peraturan;
7          this.sanksi = sanksi;
8      }
9
10     public String getPeraturan() {
11         return peraturan;
12     }
13
14     public String getSanksi() {
15         return sanksi;
16     }
17
18     public void terapkanSanksi(AnggotaKampus anggota) {
19         System.out.println("Sanksi diterapkan kepada " + anggota.getNama() + ": " + sanksi);
20     }
21 }
22

```

6. Class SistemInformasiTataTertib

```
1 import java.util.ArrayList;
2
3 public class SistemInformasiTataTertib {
4     private ArrayList<Pelanggaran> daftarPelanggaran = new ArrayList<>();
5
6     public void catatPelanggaran(Pelanggaran pelanggaran) {
7         daftarPelanggaran.add(pelanggaran);
8         System.out.println("Pelanggaran dicatat: " + pelanggaran.getPelanggaran().getNama() +
9             " melanggar " + pelanggaran.getPeraturanDilanggar().getPeraturan() +
10             " pada " + pelanggaran.getTanggal());
11     }
12
13     public void lihatRiwayatPelanggaran(AnggotaKampus anggota) {
14         System.out.println("Riwayat pelanggaran " + anggota.getNama() + " :");
15         for (Pelanggaran p : daftarPelanggaran) {
16             if (p.getPelanggaran().equals(anggota)) {
17                 System.out.println("- " + p.getPeraturanDilanggar().getPeraturan() +
18                     " | Sanksi: " + p.getPeraturanDilanggar().getSanksi() +
19                     " | Tanggal: " + p.getTanggal());
20             }
21         }
22     }
23 }
24
```

7. Main

```
1 public class Main {
2     public static void main(String[] args) {
3         // Membuat data anggota kampus
4         Mahasiswa mhs1 = new Mahasiswa("Ilham", "A01", "2141720140", "Teknik Informatika");
5         Dosen dosen1 = new Dosen("Pak Budi", "D01", "1987654321", "Informatika");
6
7         // Membuat tata tertib
8         TataTertib tt1 = new TataTertib("Tidak boleh terlambat masuk kelas", "Teguran tertulis");
9         TataTertib tt2 = new TataTertib("Dilarang merokok di area kampus", "Skorsing 3 hari");
10
11         // Sistem
12         SistemInformasiTataTertib sistem = new SistemInformasiTataTertib();
13
14         // Mencatat pelanggaran
15         Pelanggaran pel1 = new Pelanggaran(mhs1, tt1, "2025-09-30");
16         Pelanggaran pel2 = new Pelanggaran(dosen1, tt2, "2025-09-29");
17
18         sistem.catatPelanggaran(pel1);
19         sistem.catatPelanggaran(pel2);
20
21         // Melihat riwayat pelanggaran
22         sistem.lihatRiwayatPelanggaran(mhs1);
23         sistem.lihatRiwayatPelanggaran(dosen1);
24     }
25 }
26
```

5.2 Verifikasi Output

```
Pelanggaran dicatat: Ilham melanggar Tidak boleh terlambat masuk kelas pada 2025-09-30
Pelanggaran dicatat: Pak Budi melanggar Dilarang merokok di area kampus pada 2025-09-29
Riwayat pelanggaran Ilham:
- Tidak boleh terlambat masuk kelas | Sanksi: Teguran tertulis | Tanggal: 2025-09-30
Riwayat pelanggaran Pak Budi:
- Dilarang merokok di area kampus | Sanksi: Skorsing 3 hari | Tanggal: 2025-09-29
```

PS: C:\Users\ILHAM_DHARMA_A\OneDrive\Documents\SEMESTER 3\PRO Pak Irsyad\TEORT\Tugas 3\