

# LAPORAN PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

## Pertemuan 6

Nama: Aditya Alfandy

NIM: H1D024103

### A. Alur Kerja Program

Program ini mensimulasikan sistem kendali armada kendaraan antariksa menggunakan konsep Abstract Class dan Polimorfisme. Alur kerja program adalah sebagai berikut:

#### 1. Inisialisasi Objek:

Program dimulai di class UjiGalaksi (Main Class). Langkah pertama adalah membuat dua objek kendaraan yang berbeda karakteristiknya1:

- Pesawat Tempur ("Astra-Fury"): Kapasitas 2 orang, memiliki 8 rudal.
- Kapal Eksplorasi ("Voyager X"): Kapasitas 10 orang, memiliki modul scan level 4.

#### 2. Penerapan Konsep Abstraksi:

Kedua kendaraan tersebut merupakan turunan dari abstract class KendaraanGalaksi. Class ini menyediakan kerangka dasar berupa atribut (nama, energi, kapasitas) dan method abstrak (aktifkanMesin, jelajah, isiEnergi) yang wajib diimplementasikan ulang oleh subclass sesuai spesifikasinya masing-masing2.

#### 3. Skenario Pesawat Tempur:

Program menjalankan urutan aksi untuk pesawat tempur:

- Mesin diaktifkan (memerlukan energi minimal 20%)<sup>3</sup>.
- Menjelajah sejauh 10 km (mengonsumsi energi 3% per km).
- Mencoba menjelajah 30 km lagi (gagal karena sisa energi tidak mencukupi)<sup>4</sup>.
- Menembakkan rudal (mengurangi jumlah amunisi).
- Menampilkan status akhir.

#### 4. Skenario Kapal Eksplorasi:

Program menjalankan urutan aksi untuk kapal eksplorasi:

- Mesin diaktifkan (memerlukan energi minimal 15%)<sup>5</sup>.
- Menjelajah sejauh 15 km (mengonsumsi energi lebih efisien, 2% per km)<sup>6</sup>.
- Melakukan pemindaian (scan) pada planet "Kepler-442b"<sup>7</sup>.
- Menampilkan status akhir.

### B. Fungsi dan Method yang Digunakan

Berikut adalah penjelasan fungsi (method) utama yang digunakan dalam sistem ini, baik yang berasal dari parent class maupun subclass:

No	Nama Method	Tipe Return	Fungsi / Kegunaan
1.	aktifkanMesin()	void	Method abstrak yang diimplementasikan subclass untuk menyalakan mesin. Melakukan pengecekan batas minimum energi (20% untuk Pesawat, 15% untuk Kapal) sebelum mesin bisa on.
2.	jelajah(int jarak)	void	Mengurangi energi berdasarkan jarak tempuh. Pesawat Tempur boros (3%/km) sedangkan Kapal Eksplorasi efisien (2%/km).
3.	tampilStatus()	void	Menampilkan informasi dasar kendaraan seperti nama, sisa energi dalam persen, dan kapasitas penumpang.
4.	tembakRudal(int)	void	Method khusus PesawatTempur. Mengurangi jumlah stok rudal jika stok mencukupi.
5.	scanPlanet(String)	void	Method khusus KapalEksplorasi. Mensimulasikan pengambilan data ilmiah dari planet menggunakan modul scan.

## C. Implementasi Kode (Source Code)

### 1. File: KendaraanGalaksi.java (Abstract Class)

```
public abstract class KendaraanGalaksi {
    private String namaKendaraan;
    private int levelEnergi;
    private int kapasitasPenumpang;

    public KendaraanGalaksi(String namaKendaraan, int kapasitasPenumpang) {
        this.namaKendaraan = namaKendaraan;
        this.kapasitasPenumpang = kapasitasPenumpang;
        this.levelEnergi = 100;
    }

    public String getNamaKendaraan() { return namaKendaraan; }
    public int getLevelEnergi() { return levelEnergi; }

    protected void setLevelEnergi(int energi) {
        this.levelEnergi = energi;
    }

    public final void tampilStatus() {
        System.out.println(namaKendaraan + " | Energi: " + levelEnergi +
            "% | Kapasitas: " + kapasitasPenumpang + " awak");
    }
}
```

```

    public abstract void aktifkanMesin();
    public abstract void jelajah(int jarak);
    public abstract void isiEnergi(int jumlah);
}

```

## 2. File: PesawatTempur.java (Subclass)

```

public class PesawatTempur extends KendaraanGalaksi {
    private int jumlahRudal;

    public PesawatTempur(String nama, int kapasitas, int jumlahRudal) {
        super(nama, kapasitas);
        this.jumlahRudal = jumlahRudal;
    }

    @Override
    public void aktifkanMesin() {
        if (getLevelEnergi() < 20) System.out.println("Energi terlalu rendah!");
        else System.out.println("Mesin pesawat tempur diaktifkan.");
    }

    @Override
    public void jelajah(int jarak) {
        int butuh = jarak * 3; // 3% per km
        if (getLevelEnergi() >= butuh) {
            setLevelEnergi(getLevelEnergi() - butuh);
            System.out.println("Pesawat tempur menjelajah sejauh " + jarak + " km.");
        } else {
            System.out.println("Energi tidak mencukupi untuk menjelajah sejauh " + jarak + "
km.");
        }
    }

    @Override
    public void isiEnergi(int jumlah) {
        setLevelEnergi(getLevelEnergi() + jumlah);
    }

    public void tembakRudal(int jumlah) {
        if (jumlahRudal >= jumlah) {
            jumlahRudal -= jumlah;
            System.out.println("Menembakkan " + jumlah + " rudal!");
        } else System.out.println("Gagal menembak.");
    }
}

```

## 3. File: KapalEksplorasi.java (Subclass)

```

public class KapalEksplorasi extends KendaraanGalaksi {

```

```

private int modulScan;

public KapalEksplorasi(String nama, int kapasitas, int modulScan) {
    super(nama, kapasitas);
    this.modulScan = modulScan;
}

@Override
public void aktifkanMesin() {
    if (getLevelEnergi() < 15) System.out.println("Energi rendah!");
    else System.out.println("Kapal eksplorasi siap berangkat!");
}

@Override
public void jelajah(int jarak) {
    int butuh = jarak * 2;
    if (getLevelEnergi() >= butuh) {
        setLevelEnergi(getLevelEnergi() - butuh);
        System.out.println("Kapal eksplorasi menjelajah sejauh " + jarak + " km.");
    } else {
        System.out.println("Gagal menjelajah.");
    }
}

@Override
public void isiEnergi(int jumlah) {
    setLevelEnergi(getLevelEnergi() + jumlah);
}

public void scanPlanet(String namaPlanet) {
    System.out.println("Melakukan scan pada planet " + namaPlanet +
        " dengan modul level " + modulScan + ".");
}
}

```

#### 4. File: UjiGalaksi.java (Main Class)

```

public class UjiGalaksi {
    public static void main(String[] args) {
        System.out.println("=== UJI SISTEM KENDARAAN GALAKSI ===");

        PesawatTempur astra = new PesawatTempur("Astra-Fury", 2, 8);
        KapalEksplorasi voyager = new KapalEksplorasi("Voyager X", 10, 4);

        System.out.println("\n--- PESAWAT TEMPUR ---");
        astra.aktifkanMesin();
        astra.jelajah(10);
        astra.jelajah(30);
        astra.tembakRudal(3);
        astra.tampilStatus();
    }
}

```

```
        System.out.println("\n--- KAPAL EKSPLORASI ---");
        voyager.aktifkanMesin();
        voyager.jelajah(15);
        voyager.scanPlanet("Kepler-442b");
        voyager.tampilStatus();
    }
}
```

#### D. Hasil Output Program

```
--- PESAWAT TEMPUR ---
Mesin pesawat tempur diaktifkan.
Pesawat tempur menjelajah sejauh 10 km.
Energi tidak mencukupi untuk menjelajah sejauh 30 km.
Menembakkan 3 rudal!
Astra-Fury | Energi: 70% | Kapasitas: 2 orang

--- KAPAL EKSPLORASI ---
Kapal eksplorasi siap berangkat!
Kapal eksplorasi menjelajah sejauh 15 km.
Melakukan scan pada planet Kepler-442b dengan modul level 4.
Voyager X | Energi: 70% | Kapasitas: 10 orang
```