

Laporan Praktikum Pemrograman Berorientasi Objek – Pertemuan 6

Dibuat oleh : NindyaAlif Romland

NIM : H1D024031

Shift Baru : I

Shift KRS : G

Deskripsi Program

Program ini membuat sistem pengelolaan kendaraan galaksi menggunakan abstract class dan polimorfisme. Semua kendaraan memiliki atribut dasar seperti nama, level energi, dan kapasitas penumpang, serta method untuk menampilkan status. Subclass seperti Pesawat Tempur dan Kapal Eksplorasi mengimplementasikan behavior berbeda (mengaktifkan mesin, menjelajah, mengisi energi, dan fitur khusus seperti menembak rudal atau scan planet). Program menguji perilaku kendaraan melalui class UjiGalaksi dan menampilkan status, aksi, dan output sesuai urutan misi.

Alur Kerja Program

Program pertama-tama membuat objek kendaraan galaksi dari subclass PesawatTempur dan KapalEksplorasi dengan atribut spesifiknya. Kemudian, masing-masing kendaraan mengaktifkan mesin menggunakan implementasi subclass, menjelajah galaksi dengan pengurangan energi sesuai aturan, dan melakukan aksi khusus seperti menembak rudal atau scan planet. Setelah semua aksi selesai, program memanggil method tampilStatus() untuk menampilkan kondisi akhir kendaraan. Semua interaksi dilakukan dengan memanfaatkan abstract class sebagai fondasi dan polimorfisme agar method yang sama memiliki perilaku berbeda sesuai jenis kendaraan.

Fungsi Yang Digunakan

1. aktifkanMesin()
Menghidupkan mesin kendaraan. Setiap subclass memiliki cara berbeda dalam menyalakan mesinnya.
2. jelajah(int jarak)
Membuat kendaraan bergerak sejauh jarak tertentu. Mengurangi level energi sesuai konsumsi kendaraan.
3. isiEnergi(int jumlah)
Mengisi energi kendaraan hingga maksimum 100% dan menampilkan progres pengisian.
4. tampilStatus()
Menampilkan status kendaraan: nama, level energi, dan kapasitas penumpang.

5. tembakRudal(int jumlah)

(Hanya Pesawat Tempur) Menembakkan rudal jika persediaan mencukupi, menampilkan pesan aksi.

6. scanPlanet(String namaPlanet)

(Hanya Kapal Eksplorasi) Melakukan pemindaian planet menggunakan modul yang tersedia.

Hasil Output

```
== UJI SISTEM KENDARAAN GALAKSI ==

--- 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

Process finished with exit code 0
```

```
public class KapalEksplorasi extends KendaraanGalaksi { 2 usages
    private int modulScan; 2 usages

    public KapalEksplorasi(String name, int kapasitas, int modulScan) { 1 usage
        super(name, kapasitas);
        this.modulScan = modulScan;
    }

    @Override 2 usages
    public void aktifkanMesin() {
        if (getLevelEnergi() < 15) {
            System.out.println("Energi tidak mencukupi untuk memulai ekspedisi!");
        } else {
            System.out.println("Kapal eksplorasi siap berangkat!");
        }
    }

    @Override 3 usages
    public void jelajah(int jarak) {
        int konsumsi = jarak * 2; // 2% per km
        if (getLevelEnergi() < konsumsi) {
            System.out.println("Energi tidak mencukupi untuk menjelajah sejauh " + jarak + " km.");
        } else {
    }
```

```
        kurangiEnergi(konsumsi);
        System.out.println("Kapal eksplorasi menjelajah sejauh " + jarak + " km.");
    }

    @Override no usages
    public void isiEnergi(int jumlah) {
        tambahEnergi(jumlah);
        System.out.println("Energi kapal eksplorasi bertambah menjadi " + getLevelEnergi() + "%.");
    }

    public void scanPlanet(String namaPlanet) { 1 usage
        System.out.println("Melakukan scan pada planet " + namaPlanet + " dengan modul level " + modulScan);
    }
}
```

```
abstract class KendaraanGalaksi { 2 usages 2 inheritors
    private String namakendaraan; 3 usages
    private int levelEnergi; 8 usages
    private int kapasitasPenumpang; 3 usages

    public KendaraanGalaksi(String namakendaraan, int kapasitasPenumpang) { 2 usages
        this.namakendaraan = namakendaraan;
        this.kapasitasPenumpang = kapasitasPenumpang;
        this.levelEnergi = 100; // default
    }

    public String getNamakendaraan() { return namakendaraan; } no usages
    public int getLevelEnergi() { return levelEnergi; } 6 usages
    public int getKapasitasPenumpang() { return kapasitasPenumpang; } no usages

    public final void tampilStatus() { 2 usages
        System.out.println(namakendaraan + " | Energi: " + levelEnergi + "% | Kapasitas: " + kapasitasPenumpang);
    }

    protected void kurangiEnergi(int jumlah) { 2 usages
        levelEnergi -= jumlah;
        if (levelEnergi < 0) levelEnergi = 0;
    }
}
```

```
protected void tambahEnergi(int jumlah) { 2 usages
    levelEnergi += jumlah;
    if (levelEnergi > 100) levelEnergi = 100;
}

public abstract void aktifkanMesin(); 2 usages 2 implementations
public abstract void jelajah(int jarak); 3 usages 2 implementations
public abstract void isiEnergi(int jumlah); no usages 2 implementations
}
```

```
public class PesawatTempur extends KendaraanGalaksi{ no usages
    private int jumlahRudal; 3 usages

    public PesawatTempur(String nama, int kapasitas, int jumlahRudal) { no usages
        super(nama, kapasitas);
        this.jumlahRudal = jumlahRudal;
    }

    @Override no usages
    public void aktifkanMesin() {
        if (getLevelEnergi() < 20) {
            System.out.println("Energi terlalu rendah! Mesin tidak dapat diaktifkan.");
        } else {
            System.out.println("Mesin pesawat tempur diaktifkan.");
        }
    }

    @Override no usages
    public void jetalah(int jarak) {
        int konsumsi = jarak * 3; // 3% per km
        if (getLevelEnergi() < konsumsi) {
            System.out.println("Energi tidak mencukupi untuk menjelajah sejauh " + jarak + " km.");
        } else {
            System.out.println("Ruang energi(konsumsi);");
            System.out.println("Pesawat tempur menjelajah sejauh " + jarak + " km.");
        }
    }

    @Override no usages
    public void isiEnergi(int jumlah) {
        tambahEnergi(jumlah);
        System.out.println("Energi pesawat tempur bertambah menjadi " + getLevelEnergi() + "%.");
    }

    public void tembakRudal(int jumlah) { no usages
        if (jumlahRudal >= jumlah) {
            jumlahRudal -= jumlah;
            System.out.println("Menembakkan " + jumlah + " rudal!");
        } else {
            System.out.println("Rudal tidak cukup untuk menembak " + jumlah + " kali!");
        }
    }
}
```

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

        PesawatTempur pesawat = new PesawatTempur( nama: "Astra-Fury", kapasitas: 2, jumlahRudal: 8 );
        KapalEksplorasi kapal = new KapalEksplorasi( nama: "Voyager X", kapasitas: 10, modulScan: 4 );

        System.out.println("== PESAWAT TEMPUR ==");
        pesawat.aktifkanMesin();
        pesawat.jetalah( jarak: 10 );
        pesawat.jelajah( jarak: 30 );
        pesawat.tembakRudal( jumlah: 3 );
        pesawat.tampilStatus();

        System.out.println();

        System.out.println("== KAPAL EKSPLORASI ==");
        kapal.aktifkanMesin();
        kapal.jelajah( jarak: 15 );
        kapal.scanPlanet( namaPlanet: "Kepler-442b" );
        kapal.tampilStatus();
    }
}
```

```
    }
}
```