

Laporan Praktikum Pemrograman Berorientasi Objek – Pertemuan 5

Dibuat oleh : NindyaAlif Romland

NIM : H1D024031

Shift Baru : I

Shift KRS : G

Deskripsi Program

Program ini dibuat untuk mendata berbagai jenis mesin menggunakan konsep inheritance dan polimorfisme. Terdapat class induk defaultMesin yang menyimpan data umum mesin, lalu subclass mesinMotor, mesinTraktor, dan mesinTraktorListrik yang memiliki atribut tambahan dan perhitungan performa berbeda melalui method override. Program kemudian membuat beberapa objek mesin, menampilkan informasi dan performanya, memanggil suara mesin dengan pengecekan instanceof, mencari mesin dengan performa tertinggi, serta mengurutkan tiga mesin terbaik berdasarkan performa. Program ini menunjukkan bagaimana perilaku objek bisa berbeda meskipun menggunakan referensi tipe class induk.

Alur Kerja Program

Program membuat array berisi 5 objek mesin dari berbagai subclass, lalu menampilkan info, kategori, dan performa setiap mesin melalui method override. Setelah itu, program memanggil suaraMesin() dengan pengecekan instanceof, mencari mesin dengan performa tertinggi, kemudian mengurutkan seluruh mesin secara descending untuk menampilkan 3 mesin terbaik. Semua proses ini memanfaatkan polimorfisme sehingga satu referensi defaultMesin dapat menangani berbagai jenis mesin dengan perilaku berbeda.

Fungsi Yang Digunakan

1. `tampilInfo()`
Menampilkan informasi lengkap mesin (nama, tenaga, dan atribut spesifik sesuai tipe mesin).
2. `nilaiPerforma()`
Menghitung nilai performa mesin dengan rumus berbeda setiap tipe (motor: HP x 1.2, traktor: HP x 0.9 + tarik x 10, traktor listrik: HP x 1.1 + baterai x 5).
3. `kategoriMesin()`
Mengembalikan kategori mesin ("Mesin Motor", "Mesin Traktor", atau "Mesin Traktor Listrik").

4. suaraMesin()

Menampilkan suara khas setiap mesin saat menyala (hanya ada di mesinMotor, mesinTraktor, dan mesinTraktorListrik).

5. instanceof

Operator untuk mengecek tipe objek sebelum memanggil method suaraMesin() (downcasting).

6. Array/Loop

Menyimpan dan mengiterasi semua objek mesin untuk pemrosesan data.

7. Sorting

Mengurutkan mesin berdasarkan nilai performa secara descending untuk menampilkan top 3 terbaik.

Hasil Output

```
*** DATA MESIN MEGATECH ***
Mesin Motor Honda Supra X | Tipe: Bebek | Tenaga: 125 HP
Kategori: Mesin Motor
Performa: 150.0

Mesin Traktor Kubota MX5200 | Tarik: 5.0 ton | Tenaga: 520 HP
Kategori: Mesin Traktor
Performa: 518.0

Mesin Traktor Listrik EcoTrac Z900 | Tarik: 4.2 ton | Baterai: 70.0 kWh | Tenaga: 300 HP
Kategori: Mesin Traktor Listrik
Performa: 680.0

Mesin Motor Yamaha R25 | Tipe: Sport | Tenaga: 250 HP
Kategori: Mesin Motor
Performa: 300.0

Mesin Traktor Listrik Volta FarmX | Tarik: 3.5 ton | Baterai: 80.0 kWh | Tenaga: 200 HP
Kategori: Mesin Traktor Listrik
Performa: 620.0
```

```
*** SUARA MESIN ***
Brummm! Mesin motor menyala!
GGGRRRR! Hidup mesinnn!
Bzzzzz! Mesin traktor listrik aktif!
Brummm! Mesin motor menyala!
Bzzzzz! Mesin traktor listrik aktif!

*** MESIN PERFORMA TERTINGGI ***
EcoTrac Z900 → 680.0

*** TOP 3 MESIN TERBAIK ***
1. EcoTrac Z900 → 680.0
2. Volta FarmX → 620.0
3. Kubota MX5200 → 518.0

Process finished with exit code 0
```

```

import java.util.Arrays;

public class analisisMesin {
    public static void main(String[] args) {

        System.out.println("==== DATA MESIN MEGATECH ===");

        defaultMesin[] mesin = new defaultMesin[5];

        mesin[0] = new mesinMotor("Honda Supra X", 125, "Bebek");
        mesin[1] = new mesinTraktor("Kubota MX5200", 520, 5.0);
        mesin[2] = new mesinTraktorListrik("EcoTrac Z900", 300, 4.2, 70);
        mesin[3] = new mesinMotor("Yamaha R25", 250, "Sport");
        mesin[4] = new mesinTraktorListrik("Volta FarmX", 200, 3.5, 80);

        // 1. Tampilkan info semua mesin
        for (defaultMesin m : mesin) {
            m.tampilInfo();
            System.out.println("Kategori: " + m.kategoriMesin());
            System.out.println("Performa: " + m.nilaiPerforma());
            System.out.println();
        }

        // 2. Suara mesin (polimorfisme + instanceof)
        System.out.println("==== SUARA MESIN ===");
        for (defaultMesin m : mesin) {
            if (m instanceof mesinMotor) {
                System.out.println(((mesinMotor) m).suaraMesin());
            } else if (m instanceof mesinTraktorListrik) {
                System.out.println(((mesinTraktorListrik) m).suaraMesin());
            } else if (m instanceof mesinTraktor) {
                System.out.println(((mesinTraktor) m).suaraMesin());
            }
        }

        // 3. Mesin Performa Tertinggi
        defaultMesin terbaik = mesin[0];
        for (defaultMesin m : mesin) {
            if (m.nilaiPerforma() > terbaik.nilaiPerforma()) {
                terbaik = m;
            }
        }
    }
}

```

```

        }
    }

System.out.println("\n==== MESIN PERFORMA TERTINGGI ===");
System.out.println(terbaik.namaMesin + " → " + terbaik.nilaiPerforma());

// 4. Sort performa descending
Arrays.sort(mesin, (a, b) -> Double.compare(b.nilaiPerforma(), a.nilaiPerforma()));

System.out.println("\n==== TOP 3 MESIN TERBAIK ===");
for (int i = 0; i < 3; i++) {
    System.out.println((i + 1) + ". " + mesin[i].namaMesin +
        " → " + mesin[i].nilaiPerforma());
}
}

}

public class defaultMesin {
    protected String namaMesin;
    protected int tenagaHP;

    public defaultMesin(String namaMesin, int tenagaHP) {
        this.namaMesin = namaMesin;
        this.tenagaHP = tenagaHP;
    }

    public void tampilInfo() {
        System.out.println("Mesin " + namaMesin + " | Tenaga: " + tenagaHP + " HP");
    }

    public double nilaiPerforma() {
        return tenagaHP * 1.0;
    }

    public String kategoriMesin() {
        return "Mesin Umum";
    }
}

```

```
public class mesinMotor extends defaultMesin {
    private String tipeMotor;

    public mesinMotor(String nama, int hp, String tipeMotor) {
        super(nama, hp);
        this.tipeMotor = tipeMotor;
    }

    @Override
    public void tampilInfo() {
        System.out.println("Mesin Motor " + namaMesin + " | Tipe: " + tipeMotor + " | Tenaga: " +
        tenagaHP + " HP");
    }

    @Override
    public double nilaiPerforma() {
        return tenagaHP * 1.2;
    }

    @Override
    public String kategoriMesin() {
        return "Mesin Motor";
    }

    public String suaraMesin() {
        return "Brummm! Mesin motor menyala!";
    }
}

public class mesinTraktor extends defaultMesin {
    protected double kapasitasTarik;

    public mesinTraktor(String nama, int hp, double kapasitasTarik) {
        super(nama, hp);
        this.kapasitasTarik = kapasitasTarik;
    }

    @Override
    public void tampilInfo() {
        System.out.println("Mesin Traktor " + namaMesin +
        " | Tarik: " + kapasitasTarik + " ton | Tenaga: " + tenagaHP + " HP");
    }
}
```

```
}

@Override
public double nilaiPerforma() {
    return (tenagaHP * 0.9) + (kapasitasTarik * 10);
}

@Override
public String kategoriMesin() {
    return "Mesin Traktor";
}

public String suaraMesin() {
    return "GGGRRRR! Hidup mesinnn!";
}
}

public class mesinTraktorListrik extends mesinTraktor {
    private double kapasitasBaterai;

    public mesinTraktorListrik(String nama, int hp, double tarik, double baterai) {
        super(nama, hp, tarik);
        this.kapasitasBaterai = baterai;
    }

    @Override
    public void tampilInfo() {
        System.out.println("Mesin Traktor Listrik " + namaMesin +
            " | Tarik: " + kapasitasTarik + " ton | Baterai: " + kapasitasBaterai +
            " kWh | Tenaga: " + tenagaHP + " HP");
    }

    @Override
    public double nilaiPerforma() {
        return (tenagaHP * 1.1) + (kapasitasBaterai * 5);
    }

    @Override
    public String kategoriMesin() {
        return "Mesin Traktor Listrik";
    }
}
```

```
public String suaraMesin() {  
    return "Bzzzzz! Mesin traktor listrik aktif!";  
}  
}
```