

Muhammad Aqil Karomy  
H1D024096  
Shift KRS C  
Shift Baru G

## Pertemuan 6

### DESKRIPSI:

Sistem inventaris mesin menggunakan konsep inheritance dan polymorphism untuk mengelola berbagai jenis mesin (motor, traktor, traktor listrik) dengan perhitungan performa yang berbeda.  
Mendemonstrasikan dynamic dispatch, method overriding, instanceof, dan sorting.

```
defaultMesin.java
class defaultMesin {
    // Atribut mesin yang dapat diakses subclass
    protected String namaMesin;
    protected int tenagaHP;

    // Constructor untuk inisialisasi mesin
    defaultMesin(String nama, int hp) {
        this.namaMesin = nama;
        this.tenagaHP = hp;
    }

    // Setter untuk nama mesin
    void setNamaMesin(String namaMesin) {
        this.namaMesin = namaMesin;
    }

    // Getter untuk nama mesin
    String getNamaMesin() {
        return this.namaMesin;
    }

    // Setter untuk tenaga HP
    void setTenagaHP(int tenagaHP) {
        this.tenagaHP = tenagaHP;
    }
```

```
// Getter untuk tenaga HP
int getTenagaHP() {
    return this.tenagaHP;
}

// Method untuk menampilkan info mesin (akan di-override)
void tampilInfo() {
    System.out.println("Mesin " + namaMesin + " | Tenaga: " + tenagaHP + " HP");
}

// Method untuk menghitung performa mesin (akan di-override)
// Rumus default: tenagaHP × 1.0
double nilaiPerforma() {
    return tenagaHP * 1.0;
}

// Method untuk mendapatkan kategori mesin (akan di-override)
String kategoriMesin() {
    return "Mesin Umum";
}

mesinMotor.java
class mesinMotor extends defaultMesin {
    // Atribut tambahan khusus mesin motor
    private String tipeMotor;

    // Constructor memanggil super untuk inisialisasi atribut parent
    mesinMotor(String nama, int hp, String tipe) {
        super(nama, hp);
        this.tipeMotor = tipe;
    }

    // Setter untuk tipe motor
    void setTipeMotor(String tipeMotor) {
        this.tipeMotor = tipeMotor;
    }

    // Getter untuk tipe motor
    String getTipeMotor() {
```

```

        return this.tipeMotor;
    }

// Override: Menampilkan info dengan format khusus motor
@Override
void tampilInfo() {
    System.out.println("Mesin Motor " + namaMesin + " | Tipe: " + tipeMotor + " | Tenaga:
" + tenagaHP + " HP");
}

// Override: Perhitungan performa motor
// Rumus: tenagaHP × 1.2
@Override
double nilaiPerforma() {
    return tenagaHP * 1.2;
}

// Override: Kategori mesin motor
@Override
String kategoriMesin() {
    return "Mesin Motor";
}

// Method unik untuk mesin motor
void suaraMesin() {
    System.out.println("Brummm! Mesin motor menyala!");
}
}

mesinTraktor.java
class mesinTraktor extends defaultMesin {
    // Atribut tambahan khusus traktor (protected agar bisa diakses subclass)
    protected double kapasitasTarik;

    // Constructor dengan super untuk parent attributes
    mesinTraktor(String nama, int hp, double tarik) {
        super(nama, hp);
        this.kapasitasTarik = tarik;
    }

    // Setter untuk kapasitas tarik
}

```

```

void setKapasitasTarik(double kapasitasTarik) {
    this.kapasitasTarik = kapasitasTarik;
}

// Getter untuk kapasitas tarik
double getKapasitasTarik() {
    return this.kapasitasTarik;
}

// Override: Menampilkan info dengan format khusus traktor
@Override
void tampilInfo() {
    System.out.println("Mesin Traktor " + namaMesin + " | Tarik: " + kapasitasTarik + " ton | "
Tenaga: " + tenagaHP + " HP");
}

// Override: Perhitungan performa traktor
// Rumus: (tenagaHP × 0.9) + (kapasitasTarik × 10)
@Override
double nilaiPerforma() {
    return (tenagaHP * 0.9) + (kapasitasTarik * 10);
}

// Override: Kategori mesin traktor
@Override
String kategoriMesin() {
    return "Mesin Traktor";
}

// Method unik untuk traktor
void suaraMesin() {
    System.out.println("GGGRRRR! Hidup Mesin!");
}
}

mesinTraktorListrik.java
class mesinTraktorListrik extends mesinTraktor {
    // Atribut tambahan khusus traktor listrik
    private double kapasitasBaterai;

    // Constructor memanggil super untuk inisialisasi traktor

```

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

// Setter untuk kapasitas baterai
void setKapasitasBaterai(double kapasitasBaterai) {
    this.kapasitasBaterai = kapasitasBaterai;
}

// Getter untuk kapasitas baterai
double getKapasitasBaterai() {
    return this.kapasitasBaterai;
}

// Override: Menampilkan info dengan format khusus traktor listrik
@Override
void tampilInfo() {
    System.out.println("Mesin Traktor Listrik " + namaMesin + " | Tarik: " + kapasitasTarik +
" ton | Baterai: " + kapasitasBaterai + " kWh | Tenaga: " + tenagaHP + " HP");
}

// Override: Perhitungan performa traktor listrik
// Rumus: (tenagaHP × 1.1) + (kapasitasBaterai × 5)
@Override
double nilaiPerforma() {
    return (tenagaHP * 1.1) + (kapasitasBaterai * 5);
}

// Override: Kategori traktor listrik
@Override
String kategoriMesin() {
    return "Mesin Traktor Listrik";
}

// Override: Suara khusus traktor listrik
@Override
void suaraMesin() {
    System.out.println("Bzzzz! Mesin traktor listrik aktif!");
}
}
```

```
analisisMesin.java
class analisisMesin {
    public static void main(String[] args) {
        // Polymorphism: Array tipe parent dapat menyimpan objek child
        defaultMesin[] mesinArray = new defaultMesin[5];

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

        // Menampilkan data semua mesin
        System.out.println("==> DATA MESIN MEGATECH ==>");
        for (defaultMesin mesin : mesinArray) {
            mesin.tampilInfo(); // Dynamic dispatch: memanggil method sesuai tipe objek
            System.out.println("Kategori: " + mesin.kategoriMesin());
            System.out.println("Performa: " + mesin.nilaiPerforma());
            System.out.println();
        }

        // Menampilkan suara mesin menggunakan instanceof dan downcasting
        System.out.println("==> SUARA MESIN ==>");
        for (defaultMesin mesin : mesinArray) {
            // Cek tipe objek untuk akses method khusus child
            if (mesin instanceof mesinMotor) {
                ((mesinMotor) mesin).suaraMesin(); // Downcasting ke mesinMotor
            } else if (mesin instanceof mesinTraktorListrik) {
                ((mesinTraktorListrik) mesin).suaraMesin(); // Downcasting ke mesinTraktorListrik
            } else if (mesin instanceof mesinTraktor) {
                ((mesinTraktor) mesin).suaraMesin(); // Downcasting ke mesinTraktor
            }
        }

        // Mencari mesin dengan performa tertinggi
        System.out.println();
        System.out.println("==> MESIN PERFORMA TERTINGGI ==>");
        defaultMesin mesinTerbaik = mesinArray[0];
        double performaTertinggi = mesinArray[0].nilaiPerforma();
```

```

for (defaultMesin mesin : mesinArray) {
    if (mesin.nilaiPerforma() > performaTertinggi) {
        performaTertinggi = mesin.nilaiPerforma();
        mesinTerbaik = mesin;
    }
}

mesinTerbaik.tampillInfo();
System.out.println("Performa: " + performaTertinggi);

// Sorting mesin berdasarkan performa (descending) menggunakan Bubble Sort
System.out.println();
System.out.println("== TOP 3 MESIN TERBAIK ==");

for (int i = 0; i < mesinArray.length - 1; i++) {
    for (int j = 0; j < mesinArray.length - i - 1; j++) {
        if (mesinArray[j].nilaiPerforma() < mesinArray[j + 1].nilaiPerforma()) {
            // Swap elements
            defaultMesin temp = mesinArray[j];
            mesinArray[j] = mesinArray[j + 1];
            mesinArray[j + 1] = temp;
        }
    }
}

// Menampilkan top 3 mesin
for (int i = 0; i < 3; i++) {
    System.out.println((i + 1) + ". " + mesinArray[i].getNamaMesin() + " - Performa: " +
mesinArray[i].nilaiPerforma());
}
}
}

```

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 ===

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

Performa: 680.0

==== TOP 3 MESIN TERBAIK ===

1. EcoTrac Z900 - Performa: 680.0
2. Volta FarmX - Performa: 620.0
3. Kubota MX5200 - Performa: 518.0

PENJELASAN ALUR KERJA:

1. Class defaultMesin sebagai parent class dengan atribut dasar (nama, tenaga)
2. Tiga subclass: mesinMotor, mesinTraktor, mesinTraktorListrik dengan atribut tambahan
3. Setiap subclass override method tampilInfo(), nilaiPerforma(), kategoriMesin()
4. Array polymorphic (defaultMesin[]) menyimpan berbagai tipe mesin
5. Loop pertama menampilkan info semua mesin dengan dynamic dispatch
6. Loop kedua menggunakan instanceof untuk downcasting dan panggil method unik (suaraMesin)
7. Algoritma pencarian linear untuk menemukan mesin dengan performa tertinggi
8. Bubble sort untuk mengurutkan mesin berdasarkan performa (descending)

## 9. Menampilkan top 3 mesin dengan performa terbaik

### FUNGSI YANG DIGUNAKAN:

- Constructor dengan super() - inisialisasi multi-level inheritance
- tampilInfo() - override untuk tampilan custom tiap tipe
- nilaiPerforma() - override dengan rumus berbeda tiap tipe:
  - \* defaultMesin:  $HP \times 1.0$
  - \* mesinMotor:  $HP \times 1.2$
  - \* mesinTraktor:  $(HP \times 0.9) + (\text{tarik} \times 10)$
  - \* mesinTraktorListrik:  $(HP \times 1.1) + (\text{baterai} \times 5)$
- kategoriMesin() - override untuk identifikasi tipe
- suaraMesin() - method unik di child classes
- Setter dan Getter - encapsulation untuk akses atribut

### KONSEP OOP:

- Inheritance: Multi-level (defaultMesin → mesinTraktor → mesinTraktorListrik)
- Polymorphism: Array parent menyimpan berbagai child objects
- Method Overriding: Implementasi berbeda untuk tiap subclass
- Dynamic Dispatch: Runtime method resolution
- Downcasting: instanceof untuk akses method khusus child
- Encapsulation: protected dan private access modifiers
- IS-A Relationship: mesinMotor IS-A defaultMesin
- Upcasting: Automatic conversion child to parent type

### ALGORITMA:

- Linear Search: Mencari performa tertinggi  $O(n)$
- Bubble Sort: Sorting descending  $O(n^2)$

Output:

```
analisismesin.java ; java analisismesin
==== 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 ====
Mesin Traktor Listrik EcoTrac Z900 | Tarik: 4.2 ton | Baterai: 70.0 kwh | Tenaga: 300 HP
Performa: 680.0

==== TOP 3 MESIN TERBAIK ====
1. EcoTrac Z900 - Performa: 680.0
2. Volta FarmX - Performa: 620.0
3. Kubota MX5200 - Performa: 518.0
```