

## LAPORAN PRAKTIKUM MINGGU 3

Topik: Penerapan *Inheritance* pada Program Point of Sales (POS) Pertanian

### A. IDENTITAS

Nama	:	Sri Wahyuningsih
Nim	:	240202844
Kelas	:	3IKRA

### B. TUJUAN

Adapun tujuan dari praktikum ini adalah agar mahasiswa mampu:

1. Menjelaskan konsep inheritance (pewarisan class) dalam pemrograman berorientasi objek (OOP).
2. Membuat superclass dan subclass untuk merepresentasikan produk pertanian secara terstruktur.
3. Mendemonstrasikan hierarki class melalui contoh kode program yang saling berhubungan.
4. Menggunakan keyword super untuk memanggil konstruktor dan method dari parent class.
5. Menyusun laporan praktikum yang menjelaskan perbedaan penggunaan inheritance dibanding class tunggal.

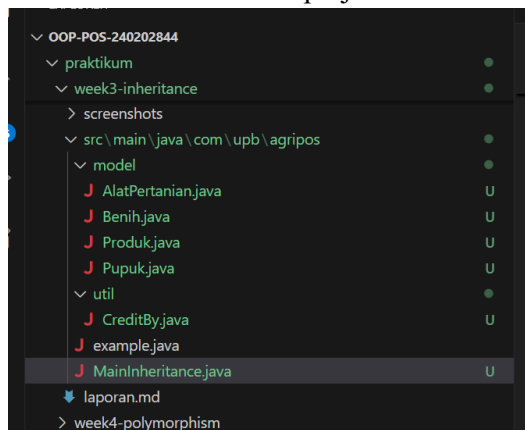
### C. DASAR TEORI

Adapun dasar teori yang mendasari praktikum ini, di antaranya:

1. Inheritance (Pewarisan) adalah konsep dalam OOP yang memungkinkan sebuah class (subclass) mewarisi atribut dan method dari class lain (superclass), sehingga kode dapat digunakan kembali dan lebih efisien.
2. Keyword extends digunakan pada subclass untuk menunjukkan bahwa class tersebut mewarisi dari superclass.
3. Keyword super() digunakan untuk memanggil konstruktor dari superclass di dalam konstruktor subclass.
4. Code reusability dan kemudahan pengembangan merupakan keunggulan utama dari penerapan inheritance dalam pemrograman berorientasi objek.

### D. LANGKAH PRAKTIKUM

1. Membuat struktur folder project:



2. Membuat superclass Produk dari hasil praktikum Bab 2.
3. Membuat tiga subclass:
  - Benih.java → atribut tambahan varietas
  - Pupuk.java → atribut tambahan jenis
  - AlatPertanian.java → atribut tambahan material
4. Membuat main class MainInheritance.java untuk instansiasi objek dari masing-masing subclass.
5. Menampilkan data produk dengan memanfaatkan inheritance.
6. Menambahkan class CreditBy.java untuk menampilkan identitas mahasiswa.
7. Commit dan push ke GitHub dengan pesan:  
*week3-inheritance*

## E. KODE PROGRAM

### 1. Produk.java

```
praktikum > week3-inheritance > src > main > java > com > upb > agripos > model > J Produk.java > Produk > P
1 package com.upb.agripos.model;
2
3 public class Produk {
4     protected String kode;
5     protected String nama;
6     protected double harga;
7     protected int stok;
8
9     public Produk(String kode, String nama, double harga, int stok) {
10        this.kode = kode;
11        this.nama = nama;
12        this.harga = harga;
13        this.stok = stok;
14    }
15
16    public void tambahStok(int jumlah) {
17        this.stok += jumlah;
18    }
19
20    public void kurangStok(int jumlah) {
21        if (jumlah <= stok) {
22            this.stok -= jumlah;
23        } else {
24            System.out.println(x:"Stok tidak mencukupi!");
25        }
26    }
27
28    public void tampilkanInfo() {
29        System.out.println("Kode: " + kode + " | Nama: " + nama +
30            " | Harga: Rp" + harga + " | Stok: " + stok);
31    }
32 }
33
```

### 2. Benih.java

```
praktikum > week3-inheritance > src > main > java > com > upb > agripos > model > J Benih.java > Benih
1 package com.upb.agripos.model;
2
3 public class Benih extends Produk {
4     private String varietas;
5
6     public Benih(String kode, String nama, double harga, int stok, String varietas) {
7         super(kode, nama, harga, stok);
8         this.varietas = varietas;
9     }
10
11    public void deskripsi() {
12        System.out.println("Kode: " + kode + " | Nama: " + nama +
13            " | Harga: Rp" + harga + " | Stok: " + stok +
14            " | Varietas: " + varietas);
15    }
16 }
17
```

### 3. Pupuk.java

```
praktikum > week3-inheritance > src > main > java > com > upb > agripos > model > J Pupuk.java > Pupuk > PupukString
1 package com.upb.agripos.model;
2
3 public class Pupuk extends Produk {
4     private String jenis;
5
6     public Pupuk(String kode, String nama, double harga, int stok, String jenis) {
7         super(kode, nama, harga, stok);
8         this.jenis = jenis;
9     }
10
11    public void deskripsi() {
12        System.out.println("Kode: " + kode + " | Nama: " + nama +
13            " | Harga: Rp" + harga + " | Stok: " + stok +
14            " | Jenis: " + jenis);
15    }
16 }
17
```

### 4. AlatPertanian.java

```
praktikum > week3-inheritance > src > main > java > com > upb > agripos > model > J AlatPertanian.java > ...
1 package com.upb.agripos.model;
2
3 public class AlatPertanian extends Produk {
4     private String material;
5
6     public AlatPertanian(String kode, String nama, double harga, int stok, String material) {
7         super(kode, nama, harga, stok);
8         this.material = material;
9     }
10
11    public void deskripsi() {
12        System.out.println("Kode: " + kode + " | Nama: " + nama +
13            " | Harga: Rp" + harga + " | Stok: " + stok +
14            " | Material: " + material);
15    }
16 }
17
```

## 5. CreditBy.java

```
praktikum > week3-inheritance > src > main > java > com > upb > agripas > util > J CreditBy.java > ...
1 package com.upb.agripas.util;
2
3 public class CreditBy {
4     public static void print() {
5         System.out.println(x:"\nCredit By: 240202844 - sriwaa");
6     }
7 }
8
```

## 6. MainInheritance.java

```
praktikum > week3-inheritance > src > main > java > com > upb > agripas > J MainInheritance.java > ...
1 package com.upb.agripas;
2
3 import com.upb.agripas.model.*;
4 import com.upb.agripas.util.CreditBy;
5
6 public class MainInheritance {
7     RunMain
8     public static void main(String[] args) {
9
10         Benih b = new Benih(kode:"MMA-002", nama:"Benih Strobery AW55", harga:80000.0, stok:100, varietas:"AW55");
11         Pupuk p = new Pupuk(kode:"SSR-005", nama:"Pupuk Hayati 25kg", harga:90000.0, stok:50, jenis:"Hayati");
12         AlatPertanian a = new AlatPertanian(kode:"SRW-025", nama:"Sekop Tangan", harga:50000.0, stok:75, material:"Baja");
13
14         System.out.println(x:"== Data Produk Pertanian ==");
15         b.deskripsi();
16         p.deskripsi();
17         a.deskripsi();
18
19         System.out.println(x:"== Menambah Stok Produk ==");
20         System.out.println(x:"Menambah stok Benih Stroberi AW55 sebanyak 20");
21         b.tambahStok(jumlah:20);
22         b.deskripsi();
23
24         System.out.println(x:"== Mengurangi Stok Produk ==");
25         System.out.println(x:"Mengurangi stok Sekop Tangan sebanyak 10");
26         a.kurangiStok(jumlah:10);
27         a.deskripsi();
28
29         CreditBy.print();
30     }
31 }
```

## F. HASIL EKSEKUSI

```
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week3-inheritance\src\main\java> java com.u
=== Data Produk Pertanian ===
Kode: MMA-002 | Nama: Benih Strobery AW55 | Harga: Rp80000.0 | Stok: 100 | Varietas: AW55
Kode: SSR-005 | Nama: Pupuk Hayati 25kg | Harga: Rp90000.0 | Stok: 50 | Jenis: Hayati
Kode: SRW-025 | Nama: Sekop Tangan | Harga: Rp50000.0 | Stok: 75 | Material: Baja

=== Menambah Stok Produk ===
Menambah stok Benih Stroberi AW55 sebanyak 20
Kode: MMA-002 | Nama: Benih Strobery AW55 | Harga: Rp80000.0 | Stok: 120 | Varietas: AW55

=== Mengurangi Stok Produk ===
Mengurangi stok Sekop Tangan sebanyak 10
Kode: SRW-025 | Nama: Sekop Tangan | Harga: Rp50000.0 | Stok: 65 | Material: Baja

Credit By: 240202844 - sriwaa
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week3-inheritance\src\main\java>
```

## G. ANALISIS

- Praktikum ini diawali dengan pembuatan tiga objek dari subclass:

- Benih
- Pupuk
- AlatPertanian

Ketiga subclass tersebut mewarisi atribut dan method dari superclass Produk, seperti kode, nama, harga, stok, serta method *tambahStok()* dan *kurangiStok()*. Selain itu, setiap subclass memiliki atribut tambahan:

- Benih → varietas
- Pupuk → jenis
- AlatPertanian → material

Masing-masing subclass juga memiliki method tambahan *deskripsi()* yang menampilkan data lengkap produk dengan semua atributnya.

Program kemudian menampilkan informasi awal semua produk, menambah stok pada salah satu produk, mengurangi stok produk lainnya, dan akhirnya menampilkan identitas mahasiswa menggunakan *CreditBy.print()*.

2. Perbedaan Pendekatan Minggu Ini Dibanding Minggu Sebelumnya. Minggu 2 (Class & Object):

Semua class dibuat terpisah tanpa hubungan antar class. Atribut dan method yang mirip harus ditulis ulang di setiap class.

Minggu 3 (Inheritance):

Sudah menerapkan konsep pewarisan (inheritance).

Class Benih, Pupuk, dan AlatPertanian tidak perlu mendefinisikan ulang atribut umum, karena sudah diwarisi dari Produk.

→ Pendekatan ini membuat kode lebih efisien, rapi, mudah diperluas, dan minim duplikasi.

3. Kendala yang Dihadapi dan Cara Mengatasinya

Kendala:

Saat menjalankan perintah git status, muncul pesan:

Untracked files present (use "git add" to track) yang menunjukkan bahwa file baru seperti:

- praktikum/week3-inheritance/src/main/java/com/upb/agripas/MainInheritance.
- java folder model
- folder util

belum terdaftar di Git.

Penyebab:

File baru hasil praktikum belum pernah ditambahkan ke *staging area*, jadi Git belum mengenalinya.

Solusi:

Menjalankan perintah berikut:

```
git add praktikum/week3-inheritance/  
git commit -m "week3-inheritance"  
git push origin main
```

Setelah itu, semua file sudah terlacak dan berhasil di-*push* ke GitHub.

## H. KESIMPULAN

Berdasarkan hasil praktikum yang telah dilakukan, dapat disimpulkan bahwa:

1. Inheritance (pewarisan) memungkinkan sebuah class mewarisi atribut dan method dari class lain, sehingga mengurangi duplikasi kode.
2. Penggunaan superclass (Produk) dan subclass (Benih, Pupuk, AlatPertanian) membuat struktur program lebih terorganisir dan mudah dikembangkan.
3. Keyword `extends` digunakan untuk mendefinisikan hubungan pewarisan antara subclass dan superclass.
4. Keyword `super()` digunakan untuk memanggil konstruktor atau method milik superclass.
5. Penerapan inheritance pada program POS pertanian membantu dalam mengelompokkan berbagai jenis produk yang memiliki kesamaan atribut, tetapi berbeda dalam detailnya.
6. Praktikum ini memperkuat pemahaman mahasiswa tentang konsep OOP, khususnya dalam mengimplementasikan hubungan hierarki antar class.

## QUIZ

1. Apa keuntungan menggunakan inheritance dibanding membuat class terpisah tanpa hubungan?

**Jawaban:**

Dengan inheritance, kita bisa menghindari pengulangan kode (code duplication) karena atribut dan method umum cukup ditulis sekali di superclass dan otomatis dimiliki oleh semua subclass. Selain itu, inheritance membuat kode lebih terstruktur, mudah dipelihara, dan lebih fleksibel untuk pengembangan di masa depan.

2. Bagaimana cara subclass memanggil konstruktor superclass?

**Jawaban:**

Subclass memanggil konstruktor superclass dengan menggunakan keyword `super()` di dalam konstruktor subclass. Contoh:

```
public class Benih extends Produk {  
    public Benih(String kode, String nama, double harga, int stok, String varietas) {  
        super(kode, nama, harga, stok); // memanggil konstruktor superclass (Produk)  
        this.varietas = varietas;  
    }  
}
```

3. Berikan contoh kasus di POS pertanian selain Benih, Pupuk, dan Alat Pertanian yang bisa dijadikan subclass.

**Jawaban:**

Contohnya adalah ObatTanaman (misalnya pestisida atau fungisida). Subclass ini bisa punya atribut tambahan seperti:

~ jenisObat (misalnya: insektisida, fungisida, herbisida)

~ dosis (misalnya: 20ml per liter air)

Contoh kode singkat:

```
public class ObatTanaman extends Produk {  
    private String jenisObat;  
    private String dosis;  
    public ObatTanaman(String kode, String nama, double harga, int stok, String  
        jenisObat, String dosis) {  
        super(kode, nama, harga, stok);  
        this.jenisObat = jenisObat;  
        this.dosis = dosis;  
    }  
}
```

## CHECKLIST KEBERHASILAN

- |   |
|---|
| <ul style="list-style-type: none"><li>✓ Superclass Produk digunakan kembali tanpa duplikasi kode.</li><li>✓ Subclass Benih, Pupuk, dan AlatPertanian berhasil dibuat dengan atribut tambahan.</li><li>✓ Program berjalan menampilkan objek dari setiap subclass.</li><li>✓ CreditBy ditampilkan dengan benar.</li><li>✓ Commit sesuai instruksi.</li><li>✓ Laporan singkat lengkap dengan analisis.</li></ul> |
|---|