

LAPORAN PRAKTIKUM MINGGU 3

TOPIK = INHERITANCE

NAMA : ALVIRA LIBRA RAMADHANI

NIM : 240202851

KELAS : 3IKRA

1. TUJUAN

- Mahasiswa mampu menjelaskan konsep inheritance (pewarisan class) dalam OOP.
- Mahasiswa mampu membuat superclass dan subclass untuk produk pertanian.
- Mahasiswa mampu mendemonstrasikan hierarki class melalui contoh kode.
- Mahasiswa mampu menggunakan super untuk memanggil konstruktor dan method parent class.
- Mahasiswa mampu membuat laporan praktikum yang menjelaskan perbedaan penggunaan inheritance dibanding class tunggal.

2. DASAR TEORI

Inheritance adalah mekanisme dalam OOP yang memungkinkan suatu class mewarisi atribut dan method dari class lain.

- Superclass → class induk yang mendefinisikan atribut umum.
- Subclass → class turunan yang mewarisi atribut/method superclass, dan dapat menambahkan atribut/method baru.
- super digunakan untuk memanggil konstruktor atau method superclass.

Dalam konteks aplikasi Agri-POS, class Produk bisa menjadi superclass, sedangkan Benih, Pupuk, dan AlatPertanian menjadi subclass. Hal ini membuat kode lebih reusable, terstruktur, dan mudah dikembangkan.

3. LANGKAH PRAKTIKUM

- Membuat Superclass Produk
 - Gunakan class Produk dari minggu sebelumnya (Bab 2) sebagai superclass.
- Membuat Subclass Baru
 - Benih.java → atribut tambahan: varietas.

```
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     }
}
```

- Pupuk.java → atribut tambahan: jenis pupuk (misal: Urea, NPK, Organik).

```

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     }

```

c. AlatPertanian.java → atribut tambahan: material (misal: baja, kayu, plastik).

```

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     }

```

3. Membuat Main Class

- Instansiasi minimal satu objek dari tiap subclass.
- Tampilkan data produk menggunakan method yang diwarisi dari superclass.

4. Menambahkan CreditBy

- Buat file CreditBy.java di folder util untuk menampilkan identitas mahasiswa (Nama dan NIM).

```

1 package com.upb.agripos.util;
2
3 public class CreditBy {
4     public static void print() {
5         System.out.println(x:"\nCredit By: 240202851 - alvira libra");
6     }
7 }

```

5. Intruksi dengan perintah berikut

```

PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java> javac com/upb/agripos/util/CreditBy.java
PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java> javac com/upb/agripos/model/*.java
PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java> javac com/upb/agripos/MainInheritance.java
PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java> java com.upb.agripos.MainInheritance

```

6. Commit dan Push ke GitHub

- Gunakan pesan commit:

```

PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java> git add .
PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java> git commit -m "week3-inheritance"
[main ee8e8dc] week3-inheritance

```

4. KODE PROGRAM

- Produk.java

```

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 kurangiStok(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     }

```

b. Benih.java

```

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 }

```

c. Pupuk.java

```

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 }

```

d. 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 }

```

e. CreditBy.java

```

1  package com.upb.agripos.util;
2
3  public class CreditBy {
4      public static void print() {
5          System.out.println(x:"\nCredit By: 240202851 - alvira libra");
6      }
7  }
8

```

f. MainInheritance.java

```

1 package com.upb.agripos;
2
3 import com.upb.agripos.model.*;
4 import com.upb.agripos.util.CreditBy;
5
6 public class MainInheritance {
7     Run main | Debug main | Run | Debug
8     public static void main(String[] args) {
9
10         Benih b = new Benih("BNH-001", "Benih Cabai AL10", 80000.0, 100, "AL10");
11         Pupuk p = new Pupuk("PPK-101", "Pupuk Organik 25kg", 90000.0, 100, "Organik");
12         AlatPertanian a = new AlatPertanian("ALT-501", "Cangkul Kecil", 50000.0, 100, "Baja");
13
14         System.out.println(x:"=== Data Produk Pertanian ===");
15         b.deskripsi();
16         p.deskripsi();
17         a.deskripsi();
18     }
19 }

```

```

17
18     System.out.println(x:"\n=== Menambah Stok Produk ===");
19     System.out.println(x:"Menambah stok Benih Cabai AL10 sebanyak 50");
20     b.tambahStok(50);
21     b.deskripsi();
22
23     System.out.println(x:"\n=== Mengurangi Stok Produk ===");
24     System.out.println(x:"Mengurangi stok Cangkul Kecil sebanyak 50");
25     a.kurangiStok(50);
26     a.deskripsi();
27
28     CreditBy.print();
29
30 }

```

5. HASIL EKSEKUSI

```

=== Data Produk Pertanian ===
Kode: BNH-001 | Nama: Benih Cabai AL10 | Harga: Rp80000.0 | Stok: 100 | Varietas: AL10
Kode: PPK-101 | Nama: Pupuk Organik 25kg | Harga: Rp90000.0 | Stok: 100 | Jenis: Organik
Kode: ALT-501 | Nama: Cangkul Kecil | Harga: Rp50000.0 | Stok: 100 | Material: Baja

=== Menambah Stok Produk ===
Menambah stok Benih Cabai AL10 sebanyak 50
Kode: BNH-001 | Nama: Benih Cabai AL10 | Harga: Rp80000.0 | Stok: 150 | Varietas: AL10

=== Mengurangi Stok Produk ===
Mengurangi stok Cangkul Kecil sebanyak 50
Kode: ALT-501 | Nama: Cangkul Kecil | Harga: Rp50000.0 | Stok: 50 | Material: Baja

Credit By: 240202851 - alvira libra
PS C:\Users\LENOVO\oop-202501-240202851\praktikum\week3-inheritance\src\main\java>

```

➤ Analisis

1. Bagaimana kode berjalan

- Program diawali dari class MainInheritance. Di dalamnya dibuat tiga objek dari subclass yaitu Benih, Pupuk, dan AlatPertanian. Setiap objek memanggil **constructor** milik superclass Produk untuk mengisi atribut umum seperti kode, nama, harga, dan stok, lalu menambahkan atribut khusus sesuai jenis produknya.
- Saat dijalankan, program memanggil **method tampilkanInfo()** dari masing-masing objek. Method ini menampilkan data produk yang telah

diisi. Karena menggunakan **inheritance**, subclass dapat menggunakan atribut dan method dari superclass tanpa perlu menulis ulang kodenya.

- c. Dengan begitu, program berjalan dari pembuatan objek → pemanggilan method → menampilkan hasil informasi produk ke layar.
2. Kendala yang dihadapi dan cara mengatasinya
Kendalanya muncul error saat kompilasi karena nama file dan class tidak sesuai atau file salah folder.
Solusinya, samakan nama file dengan class dan pastikan file ada di folder sesuai package.

6. KESIMPULAN

Dari praktikum ini dapat disimpulkan bahwa konsep **inheritance** memudahkan pengembangan program karena subclass dapat menggunakan kembali atribut dan method dari superclass tanpa menulis ulang. Hal ini membuat kode lebih **efisien, terstruktur, dan mudah dikembangkan** untuk berbagai jenis produk pertanian.

7. CHEKLIST KEBERHASILAN

- ☒ Superclass Produk digunakan kembali tanpa duplikasi kode.
- ☒ Subclass Benih, Pupuk, dan AlatPertanian berhasil dibuat dengan atribut tambahan.
- ☒ Program berjalan menampilkan objek dari setiap subclass.
- ☒ CreditBy ditampilkan dengan benar.
- ☒ Commit sesuai instruksi.
- ☒ Laporan singkat lengkap dengan analisis.

8. QUIZ

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

Jawaban:

Inheritance membuat kode lebih efisien karena subclass dapat mewarisi atribut dan method dari superclass. Dengan begitu, tidak perlu menulis ulang kode yang sama di setiap class, sehingga program lebih mudah dikelola dan dikembangkan.

2. Bagaimana cara subclass memanggil konstruktor superclass?

Jawaban:

Subclass memanggil konstruktor superclass dengan menggunakan keyword `super(...)` pada baris pertama konstruktor subclass, sambil mengirimkan parameter yang sesuai.

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

Jawaban:

Contohnya ObatTanaman, Pestisida, atau BibitSayur, yang masing-masing bisa menjadi subclass dari class Produk dengan atribut tambahan seperti jenis obat, dosis, atau masa kadaluarsa.

