**LAPORAN MINGGU KE 3**

**Penerapan Konsep Inheritance pada Sistem Informasi Produk Pertanian (Agri-POS)**

**NAMA : Hilda Sava Akzena**

**KELAS : 3 ikra**

**NIM : 240202865**

**TUJUAN**

1. Mahasiswa memahami konsep **inheritance (pewarisan class)** dalam pemrograman berorientasi objek (OOP).
2. Mahasiswa mampu membuat **superclass dan subclass** yang merepresentasikan kategori produk pertanian.
3. Mahasiswa dapat menerapkan **pemanggilan konstruktor dan method superclass** menggunakan kata kunci super.
4. Mahasiswa mampu mendemonstrasikan **hierarki class** dalam program yang terstruktur dan reusable.
5. Mahasiswa dapat menjelaskan **perbedaan penggunaan inheritance** dibandingkan dengan penggunaan class tunggal pada sistem informasi produk.

**DASAR TEORI**

1. **Inheritance** adalah konsep dalam OOP yang memungkinkan suatu class mewarisi atribut dan method dari class lain.
2. **Superclass** adalah class induk yang berisi atribut dan method umum.  
   **Subclass** adalah class turunan yang mewarisi dari superclass dan bisa menambah fitur baru.
3. Kata kunci **extends** digunakan untuk membuat hubungan antara superclass dan subclass.
4. Kata kunci **super** digunakan untuk memanggil konstruktor atau method dari superclass.
5. Dengan inheritance, program menjadi lebih **terstruktur, efisien, dan mudah dikembangkan**.
6. Dalam sistem **Agri-POS**, inheritance digunakan untuk membuat class:
   * Produk sebagai superclass,
   * Benih, Pupuk, dan AlatPertanian sebagai subclass yang memiliki atribut khusus masing-masing.

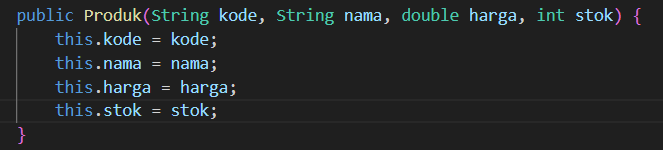
**LANGKAH PRAKTIKUM**

1.Membuat Superclass Produk

* Buat folder dan package sesuai struktur



* Di dalam package model, buat file Produk.java.
* Class ini berfungsi sebagai **superclass (class induk)** yang menyimpan atribut umum untuk semua jenis produk pertanian.
* Tambahkan atribut:kode (String), nama (String), harga (double), stok (int)



**2. Membuat Subclass dari Produk**

Buat tiga class turunan (subclass) di package model, yaitu Benih, Pupuk, dan AlatPertanian.

**a. Benih.java**

* Tambahkan atribut baru varietas.



* Gunakan super() untuk memanggil konstruktor dari Produk.



**b. Pupuk.java**

* Tambahkan atribut jenis.



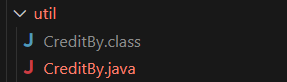
**c. AlatPertanian.java**

* Tambahkan atribut material.



**3**.**Membuat Class CreditBy**

* Buat folder dalam package util sesuai struktur



* Class ini digunakan untuk menampilkan identitas mahasiswa.

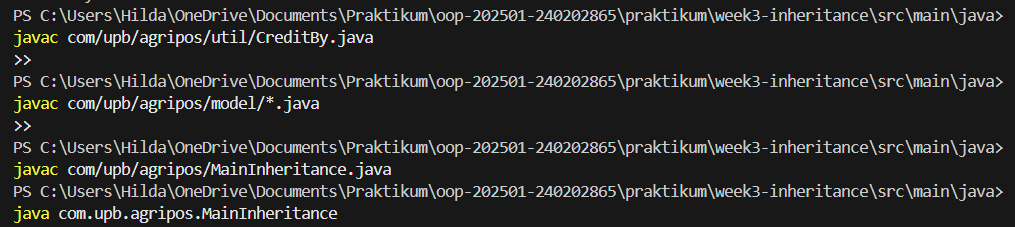


4. **Membuat Class Utama MainInheritance**

* Class ini digunakan untuk menjalankan program dan menguji konsep inheritance.

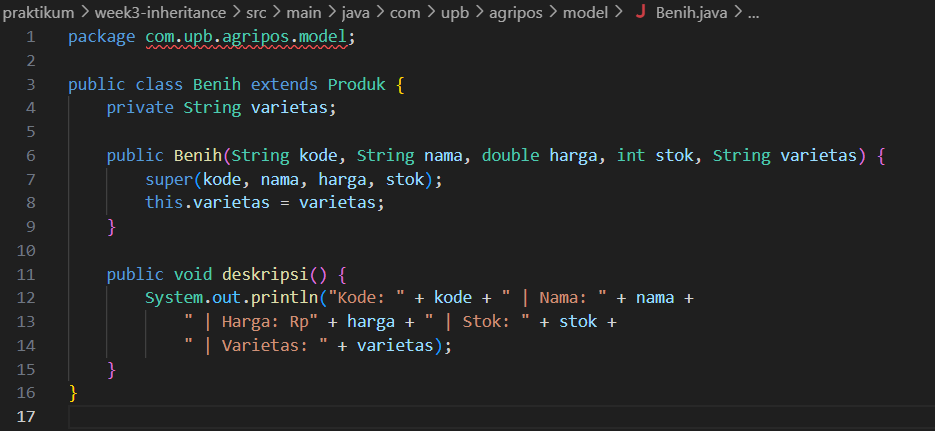
**5.Melakukan komplikasi dan menjalankan program**

* Jalankan perintah berikut di terminal

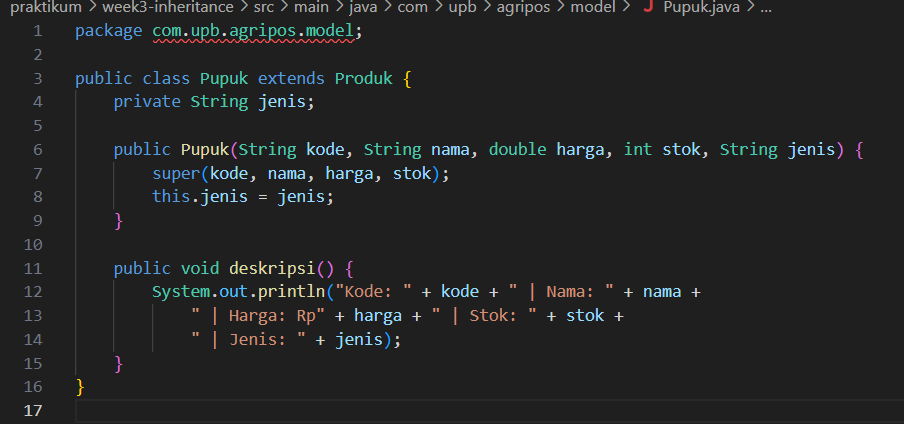


**KODE PROGRAM**

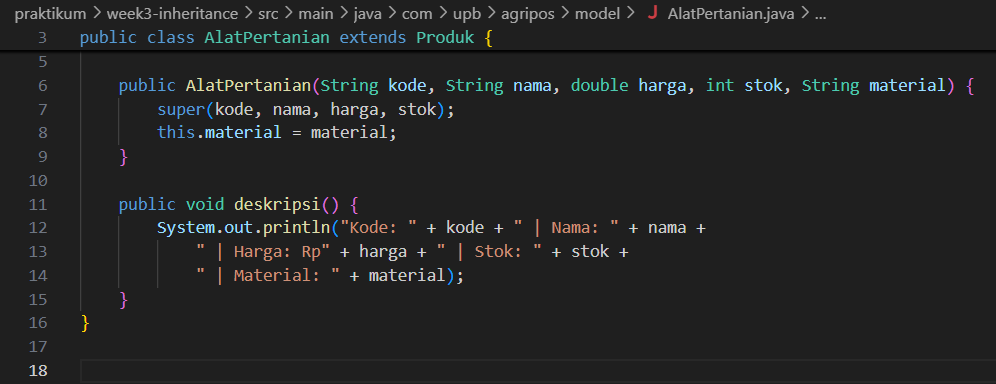
**a.Benih.java**

****

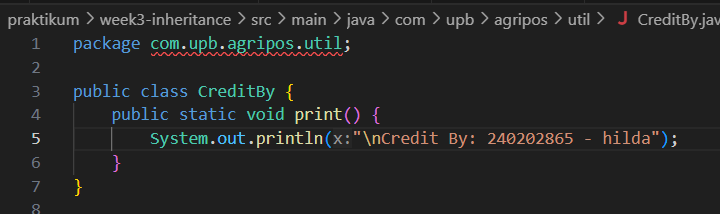
**b. Pupuk.java**

****

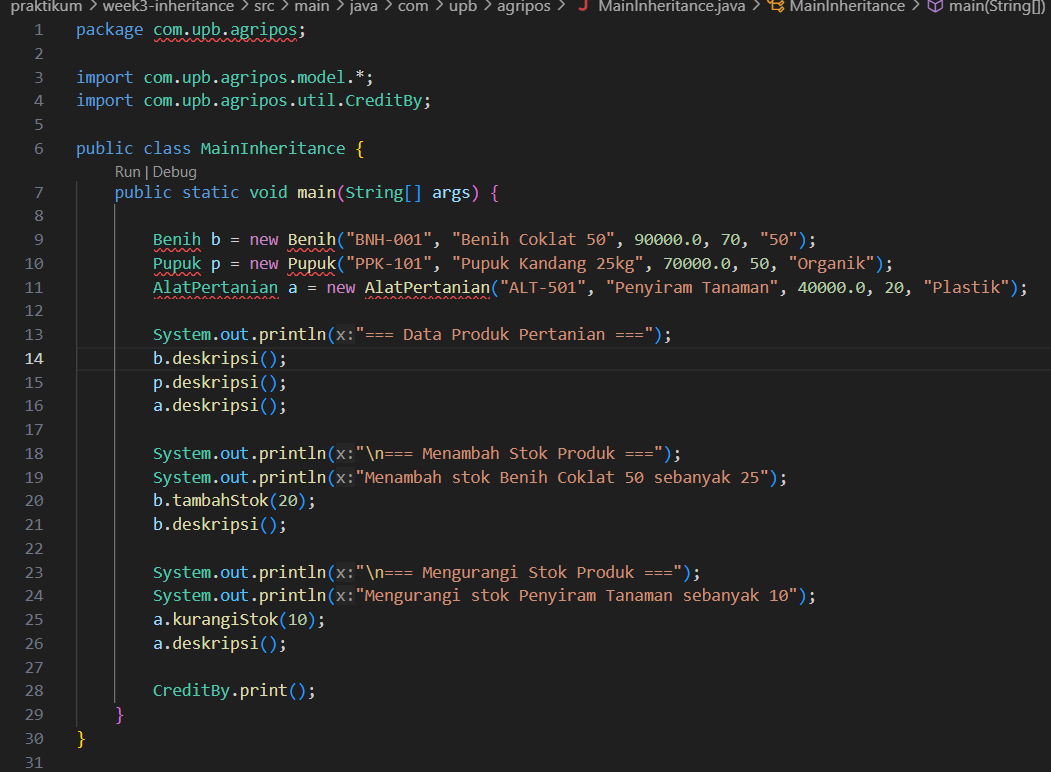
**c. AlatPertanian.java**

****

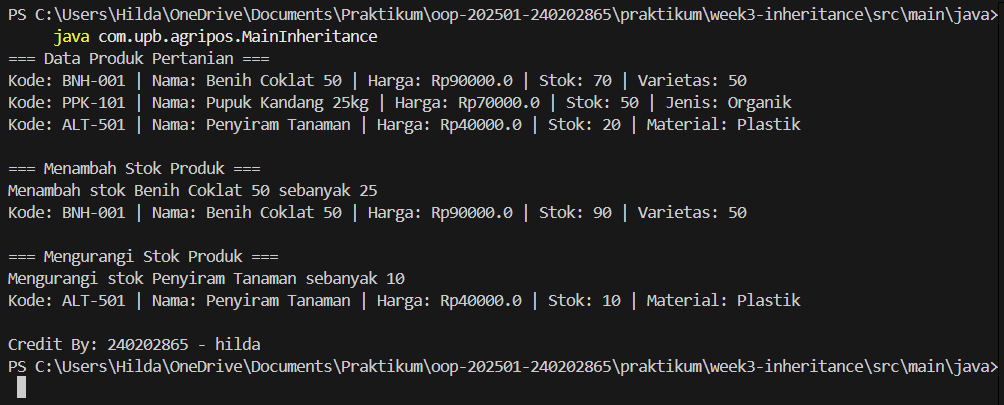
**d. CreditBy.java**

****

**e.MainInheritance.java**

****

**HASIL EKSEKUSI**

****

**ANALISIS**

**1.Jalannya Program**

* Program dimulai dari class MainInheritance.
* Di dalamnya dibuat tiga objek dari subclass yaitu Benih, Pupuk, dan AlatPertanian.
* Setiap subclass menggunakan konstruktor super() untuk memanggil atribut dari superclass Produk.
* Setelah objek dibuat, program menampilkan data masing-masing produk dengan memanggil method getNama(), getVarietas(), getJenis(), dan getMaterial().
* Terakhir, class CreditBy dipanggil untuk menampilkan identitas mahasiswa.
* Semua proses ini menunjukkan bahwa subclass dapat mewarisi atribut dan method dari superclass tanpa perlu menulis ulang kode yang sama.

**2. Perbedaan dengan Minggu Sebelumnya**

* Pada minggu sebelumnya (Bab 2), program hanya menggunakan class tunggal, yaitu Produk, untuk semua jenis produk.
* Pada minggu ini, digunakan konsep inheritance untuk membedakan kategori produk melalui subclass (Benih, Pupuk, dan AlatPertanian).
* Dengan inheritance, struktur kode menjadi lebih terorganisir, efisien, dan mudah dikembangkan, karena setiap jenis produk bisa memiliki atribut dan perilaku khusus tanpa mengubah class utama.

**3.Kendala dan Cara Mengatasinya**

* Kendala: Awalnya program error karena lupa menambahkan super() di konstruktor subclass.  
  Solusi: Menambahkan pemanggilan super(kode, nama, harga, stok) di konstruktor setiap subclass.

**KESIMPULAN**

Dengan menerapkan konsep **inheritance**, program menjadi lebih terstruktur dan mudah dikembangkan. Subclass seperti Benih, Pupuk, dan AlatPertanian dapat mewarisi atribut dari Produk, sehingga kode lebih efisien dan tidak berulang.

**QUIZ**

**1.Apa keuntungan menggunakan inheritance dibanding membuat class terpisah tanpa hubungan?**  
Jawaban: Kode menjadi lebih efisien, tidak berulang, dan lebih mudah dikembangkan karena subclass bisa mewarisi atribut dan method dari superclass.

**2. Bagaimana cara subclass memanggil konstruktor superclass?**  
Jawaban: Dengan menggunakan keyword **super()** di dalam konstruktor subclass.

**3. Berikan contoh kasus di POS pertanian selain Benih, Pupuk, dan Alat Pertanian yang bisa dijadikan subclass.**  
Jawaban: Contohnya **Pestisida**, **Bibit Ikan**, atau **Pakan Ternak** yang semuanya dapat mewarisi atribut umum dari class **Produk**.

**CHECKLIST 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.