LAPORAN PRAKTIKUM MINGGU 2

Topik: Class dan Object (Produk Pertanian)

A. IDENTITAS

Nama	: Sri Wahyuningsih
Nim	: 240202844
Kelas	: 3IKRA

B. TUJUAN

Praktikum ini memiliki tujuan sebagai berikut:

1. Mahasiswa memahami konsep class dan object dalam pemrograman berorientasi objek (OOP).
2. Mahasiswa dapat menerapkan enkapsulasi menggunakan access modifier dan getter/setter.
3. Mahasiswa mampu membuat class Produk untuk merepresentasikan produk pertanian.
4. Mahasiswa mampu menginstansiasi beberapa objek dan menampilkan informasinya di console.

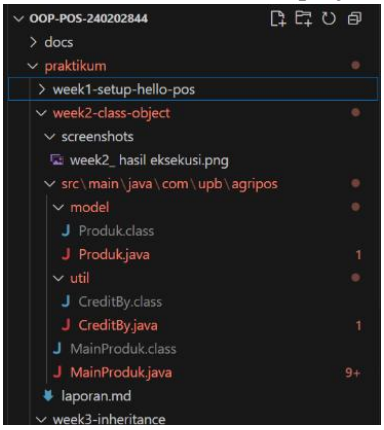
C. DASAR TEORI

Adapun dasar teori yang mendasari praktikum ini, diantaranya:

1. Class adalah *blueprint* dari objek, berisi atribut dan method.
2. Object adalah instansiasi dari class yang merepresentasikan entitas nyata.
3. Enkapsulasi digunakan untuk melindungi data dengan membuat atribut bersifat *private*.
4. Getter dan Setter digunakan untuk mengakses dan mengubah nilai atribut secara aman.Dalam OOP, setiap objek dapat berinteraksi melalui method untuk menghasilkan perilaku tertentu.

D. LANGKAH PRAKTIKUM

1. Membuat struktur folder project:



2. Buat class Produk dengan atribut kode, nama, harga, dan stok, serta method getter dan setter.
3. Buat class CreditBy pada package util yang menampilkan identitas mahasiswa dengan format credit by: <NIM> - <Nama>.
4. Instansiasi minimal tiga produk pertanian dan tampilkan informasinya di console, diakhiri dengan pemanggilan CreditBy.print().
5. Tambahkan method tambahStok(int jumlah) dan kurangiStok(int jumlah) untuk mengelola stok produk.

6. Jalankan perintah kompilasi dengan javac, lalu jalankan program dengan perintah java agar muncul hasil eksekusinya

```
    a85c785..799e882  main -> main
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> javac com/upb/agripos/model/Produk.java
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> javac com/upb/agripos/util/CreditBy.java
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> javac com/upb/agripos/MainProduk.java
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> java com.upb.agripos.MainProduk
=== Info Awal Produk ===
```

7. Melakukan *commit* dan *push*

```
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> git add .
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> git commit -m "upload praktikum 2"
On branch main
Your branch is ahead of 'origin/main' by 2 commits.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
PS C:\Users\sri61\Documents\oop-pos-240202844\praktikum\week2-class-object\src\main\java> git push origin main
Enumerating objects: 29, done.
Counting objects: 100% (29/29), done.
Delta compression using up to 2 threads
Compressing objects: 100% (12/12), done.
Writing objects: 100% (18/18), 1.86 KiB | 272.00 KiB/s, done.
Total 18 (delta 6), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (6/6), completed with 4 local objects.
To https://github.com/sriwaa/oop-202501-240202844.git
   a85c785..799e882  main -> main
```

E. KODE PROGRAM

1. Produk.java

```java
package com.upb.agripos.model;

public class Produk {
    private String kode;
    private String nama;
    private double harga;
    private int stok;

    public Produk(String kode, String nama, double harga, int stok) {
        this.kode = kode;
        this.nama = nama;
        this.harga = harga;
        this.stok = stok;
    }

    public String getKode() { return kode; }
    public void setKode(String kode) { this.kode = kode; }

    public String getNama() { return nama; }
    public void setNama(String nama) { this.nama = nama; }

    public double getHarga() { return harga; }
    public void setHarga(double harga) { this.harga = harga; }

    public int getStok() { return stok; }
    public void setStok(int stok) { this.stok = stok; }

    public void tambahStok(int jumlah) {
        stok += jumlah;
    }

    public int getStok() { return stok; }
    public void setStok(int stok) { this.stok = stok; }

    public void tambahStok(int jumlah) {
        stok += jumlah;
        System.out.println("Berhasil menambah stok " + jumlah + " unit. Stok sekarang: " + stok);
    }

    public void kurangiStok(int jumlah) {
        if (stok >= jumlah) {
            stok -= jumlah;
            System.out.println("Berhasil mengurangi stok " + jumlah + " unit. Stok sekarang: " + stok);
        } else {
            System.out.println("Stok tidak cukup! Tidak bisa mengurangi " + jumlah + " unit.");
        }
    }

    public void tampilkanInfo() {
        System.out.println("Kode: " + kode + ", Nama: " + nama +
                ", Harga: " + harga + ", Stok: " + stok);
    }
}
```

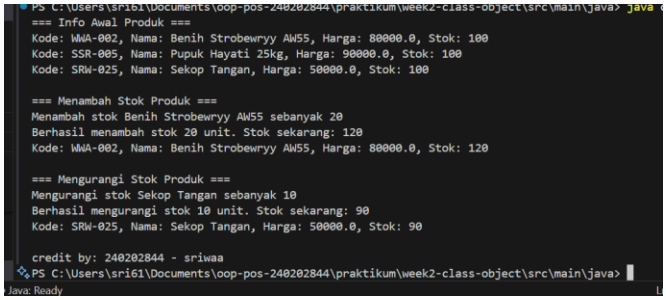
2. CreditBy.java

```java
package com.upb.agripos.util;

public class CreditBy {
    public static void print() {
        System.out.println(x:"\ncredit by: 240202844 - sriwaa");
    }
}
```

3. MainProduk.java

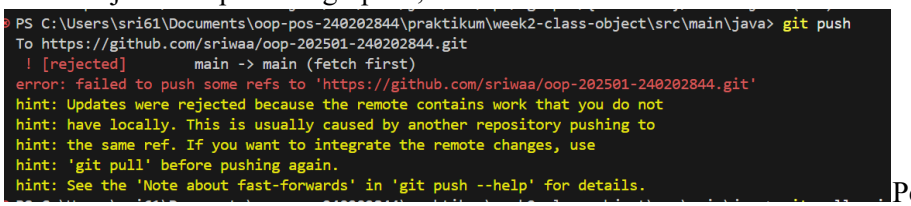
```java
package com.upb.agripos;

import com.upb.agripos.model.Produk;
import com.upb.agripos.util.CreditBy;

public class MainProduk {
    Run | Debug
    public static void main(String[] args) {

        Produk p1 = new Produk("WWA-002", "Benih Strobewryy AW55", 80000.0, 100);
        Produk p2 = new Produk("SSR-005", "Pupuk Hayati 25kg", 90000.0, 100);
        Produk p3 = new Produk("SRW-025", "Sekop Tangan", 50000.0, 100);

        System.out.println(x:"=== Info Awal Produk ===");
        p1.tampilkanInfo();
        p2.tampilkanInfo();
        p3.tampilkanInfo();

        System.out.println(x:"\n=== Menambah Stok Produk ===");
        System.out.println(x:"Menambah stok Benih Strobewryy AW55 sebanyak 20");
        p1.tambahStok(20);
        p1.tampilkanInfo();

        System.out.println(x:"\n=== Mengurangi Stok Produk ===");
        System.out.println(x:"Mengurangi stok Sekop Tangan sebanyak 10");
        p3.kurangiStok(10);
        p3.tampilkanInfo();

        CreditBy.print();
    }
}
```

F. HASIL EKSEKUSI

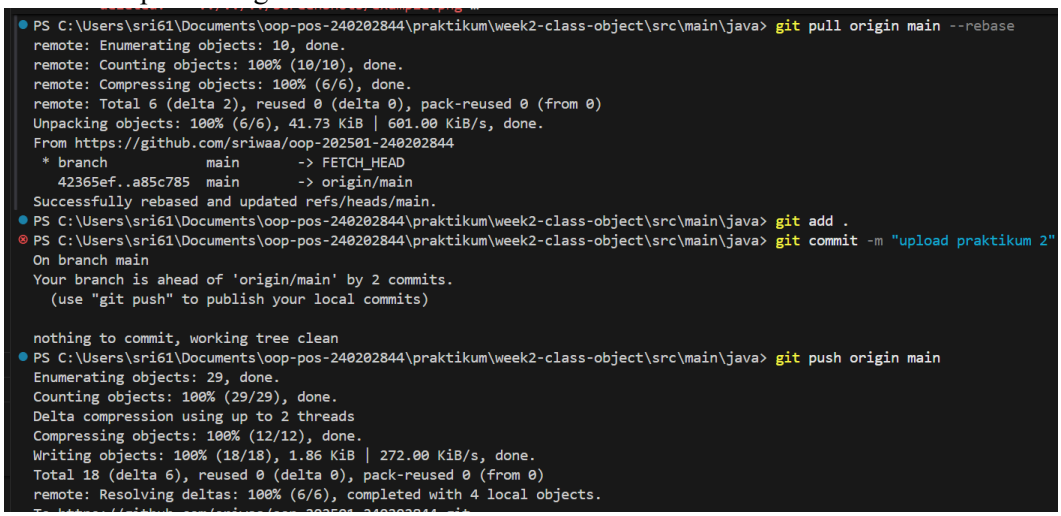


G. ANALISIS

1. Alur kode berjalan:
   a) Program dimulai dari MainProduk, yang membuat beberapa objek dari class Produk.
   b) Setiap objek memiliki atribut seperti kode, nama, harga, dan stok.
   c) Informasi produk ditampilkan menggunakan method *getter* dari class Produk.
   d) Di akhir, class CreditBy digunakan untuk menampilkan identitas mahasiswa sebagai tanda pembuat program.
2. Perbedaan dengan minggu sebelumnya:
   - Minggu sebelumnya hanya membuat program sederhana dengan satu file (tanpa class terpisah).
   - Minggu ini menggunakan konsep *class* dan *object* dengan struktur package (model, util, main), sehingga program lebih terorganisir. Diterapkan prinsip *enkapsulasi* untuk melindungi data.
3. Kendala dan cara mengatasinya:
   Saat menjalankan perintah git push, muncul error:



   Penyebab: Repository di GitHub sudah memiliki update terbaru yang belum ada di lokal.
   Solusi: Jalankan perintah ini untuk menarik perubahan terbaru terlebih dahulu sebelum melakukan push ulang.



H. KESIMPULAN

1. Mahasiswa memahami konsep dasar *class* dan *object* dalam OOP. Penerapan *enkapsulasi* dengan *getter* dan *setter* membantu menjaga keamanan data.
2. Program menjadi lebih modular dan mudah dikembangkan di masa depan.
3. Struktur package membuat kode lebih terorganisir dan profesional.
4. Praktikum ini menjadi dasar penting untuk pengembangan aplikasi POS yang lebih kompleks.

QUIZ

1. Mengapa atribut sebaiknya dideklarasikan sebagai private dalam class?
   Jawaban:
   Atribut sebaiknya dideklarasikan sebagai *private* agar data di dalam class terlindungi dan tidak bisa diakses langsung dari luar class. Hal ini merupakan prinsip *enkapsulasi* yang bertujuan menjaga keamanan data serta mencegah perubahan nilai atribut secara sembarangan.

2. Apa fungsi getter dan setter dalam enkapsulasi?
   Jawaban:
   Fungsi *getter* adalah untuk mengambil atau membaca nilai dari atribut yang bersifat *private*, sedangkan *setter* digunakan untuk mengubah atau menetapkan nilai atribut tersebut secara terkontrol. Dengan getter dan setter, kita bisa menerapkan logika validasi sebelum nilai atribut diubah.

3. Bagaimana cara class Produk mendukung pengembangan aplikasi POS yang lebih kompleks?
   Jawaban:
   Class Produk menjadi fondasi utama dalam aplikasi POS karena mewakili data barang yang dijual. Dengan struktur atribut seperti kode, nama, harga, dan stok, serta method seperti tambahStok() dan kurangiStok(), class ini bisa dikembangkan lebih lanjut untuk fitur transaksi, laporan penjualan, dan manajemen inventori secara terintegrasi.

CHECKLIST KEBERHASILAN

- ✓ Class Produk berhasil dibuat dengan atribut dan method yang lengkap.
- ✓ Class CreditBy berhasil dibuat dan dipanggil di program utama.
- ✓ Objek produk berhasil diinstansiasi dan ditampilkan.
- ✓ Enkapsulasi sudah diterapkan dengan benar.
- ✓ Commit dengan pesan sesuai instruksi berhasil dilakukan.
- ✓ Screenshot hasil eksekusi telah dilampirkan.
- ✓ Laporan singkat telah dibuat.