

LAPORAN MINGGU 2

CLASS DAN OBJECT

Nama : Hilda Sava Alzena

Kelas : 3 IKRA

Nim : 240202865

TUJUAN

Mahasiswa memahami konsep class dan object dalam pemrograman berorientasi objek (OOP) serta mampu membuat dan mengimplementasikan class dengan atribut dan metode untuk membentuk object yang dapat digunakan dalam program Java.

DASAR TEORI

1. Class adalah blueprint atau cetak biru dari suatu objek yang berisi atribut (variabel) dan metode (fungsi) untuk menggambarkan karakteristik serta perilaku suatu entitas.
2. Object adalah instansiasi dari sebuah class, yaitu bentuk nyata yang memiliki data dan dapat menjalankan metode yang didefinisikan di dalam class.
3. Enkapsulasi adalah konsep dalam OOP yang digunakan untuk menyembunyikan data (atribut) agar tidak dapat diakses langsung dari luar class, melainkan melalui metode khusus seperti getter dan setter.
4. Pemrograman Berorientasi Objek (OOP) memudahkan pengelolaan program karena setiap bagian dipisahkan dalam bentuk objek yang saling berinteraksi.
5. Dengan menggunakan class dan object, kode program menjadi lebih terstruktur, mudah dipelihara, dan dapat digunakan kembali (reusable).

LANGKAH PRAKTIKUM

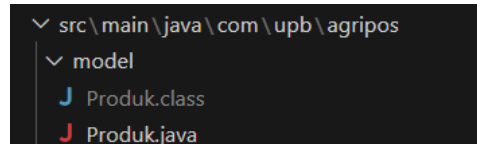
1. Menyiapkan lingkungan kerja

- Buka Visual Studio Code atau IntelliJ IDEA sebagai IDE untuk menulis kode Java.

Pastikan JDK sudah terpasang dan dikonfigurasi dengan benar.

2. Membuat file Produk.java

- Buat file bernama Produk.java di dalam folder src/main/java/com/upb/agripos/ pada package model.

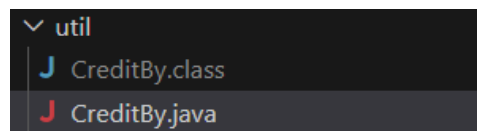


- Tulis kode program untuk mendefinisikan class Produk yang memiliki atribut seperti kode, nama, harga, dan stok.

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

3. Membuat file CreditBy.java

- Berada di folder yang sama, tapi pada package util



- File ini berfungsi untuk menampilkan identitas pembuat program, misalnya:

```
praktikum > week2 > class object > src > main > java > com > upb > agripas > util > CreditBy.java > ...  
1 package com.upb.agripas.util;  
2  
3  
4 public class CreditBy {  
5     public static void print() {  
6         System.out.println("Ncredit by: 240202065 - hilda");  
7     }  
8 }
```

4. Membuat file MainProduk.java

- File utama untuk menjalankan program.
- Tambahkan deklarasi package dan import class yang dibutuhkan

Tulis di bagian atas file:

```
1 package com.upb.agripas;  
2  
3 import com.upb.agripas.model.Produk;  
4 import com.upb.agripas.util.CreditBy;  
5
```

- Buat beberapa objek dengan data berbeda menggunakan konstruktor:

```
Produk p1 = new Produk(kode:"HLD-005", nama:"Benih Coklat 50", harga:90000.0, stok:70);
Produk p2 = new Produk(kode:"SVA-007", nama:"Pupuk Kandang 30kg", harga:70000.0, stok:50);
Produk p3 = new Produk(kode:"ZNA-020", nama:"Penyiram Tanaman 05", harga:40000.0, stok:20);
```

- Menampilkan identitas pembuat program
- Panggil class CreditBy dari package util untuk menampilkan identitas:

```
CreditBy.print();
```

5. Melakukan kompilasi dan menjalankan program

- Jalankan perintah berikut di terminal:

```
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865> cd .\praktikum\
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum> cd .\week2-class-object\src\main\java\
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum\week2-class-object\src\main\java> javac com/upb/agripos/model/Produk.java
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum\week2-class-object\src\main\java> javac com/upb/agripos/util/CreditBy.java
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum\week2-class-object\src\main\java> javac com/upb/agripos/MainProduk.java
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum\week2-class-object\src\main\java> java com.upb.agripos.MainProduk
```

6. Melakukan commit ke GitHub

KODE PROGRAM

1. Produk.java

```
praktikum > week2 class object > src > main > java > com > upb > agripos > model > Produk.java > #Produk > #getStokJ
1 package com.upb.agripos.model;
2
3
4 public class Produk {
5     private String kode;
6     private String nama;
7     private double harga;
8     private int stok;
9
10    public Produk(String kode, String nama, double harga, int stok) {
11        this.kode = kode;
12        this.nama = nama;
13        this.harga = harga;
14        this.stok = stok;
15    }
16
17    public String getKode() { return kode; }
18    public void setKode(String kode) { this.kode = kode; }
19
20    public String getNama() { return nama; }
21    public void setNama(String nama) { this.nama = nama; }
22
23    public double getHarga() { return harga; }
24    public void setHarga(double harga) { this.harga = harga; }
25
26    public int getStok() { return stok; }
27    public void setStok(int stok) { this.stok = stok; }
28
29    public void tambahStok(int jumlah) {
30        stok += jumlah;
31        System.out.println("Berhasil menambah stok " + jumlah + " unit. Stok sekarang: " + stok);
32    }
33 }
```

```

32
33     public void kurangistok(int jumlah) {
34         if (stok >= jumlah) {
35             stok -= jumlah;
36             System.out.println("Berhasil mengurangi stok " + jumlah + " unit. Stok sekarang: " + stok);
37         } else {
38             System.out.println("Stok tidak cukup! Tidak bisa mengurangi " + jumlah + " unit.");
39         }
40     }
41
42     public void tampilkanInfo() {
43         System.out.println("Kode: " + kode + ", Nama: " + nama +
44             ", Harga: " + harga + ", Stok: " + stok);
45     }
46 }

```

2.CreditBy.java

```

praktikum > week2-class-object > src > main > java > com > upb > agripus > util > CreditBy.java > ...
1   package com.upb.agripus.util;
2
3   public class CreditBy {
4       public static void print() {
5           System.out.println(x: "\ncredit by: 240202065 - hilda");
6       }
7   }
8

```

3.MainProduk.java

```

praktikum > week2-class-object > src > main > java > com > upb > agripus > MainProduk.java > MainProduk > main(String[])
1   package com.upb.agripus;
2
3   import com.upb.agripus.model.Produk;
4   import com.upb.agripus.util.CreditBy;
5
6   public class MainProduk {
7       public static void main(String[] args) {
8
9           Produk p1 = new Produk(kode:"HLD-005", nama:"Benih Coklat 50", harga:90000.0, stok:70);
10          Produk p2 = new Produk(kode:"SWA-007", nama:"Pupuk Kandang 30kg", harga:70000.0, stok:50);
11          Produk p3 = new Produk(kode:"ZNA-020", nama:"Penyiram Tanaman 05", harga:40000.0, stok:20);
12
13          System.out.println(x: "--- Info Awal Produk ---");
14          p1.tampilkanInfo();
15          p2.tampilkanInfo();
16          p3.tampilkanInfo();
17
18          System.out.println(x: "\n--- Menambah Stok Produk ---");
19          System.out.println(x: "Menambah stok Benih Coklat 50 sebanyak 25");
20          p1.tambahStok(jumlah:25);
21          p1.tampilkanInfo();
22
23          System.out.println(x: "\n--- Mengurangi Stok Produk ---");
24          System.out.println(x: "Mengurangi stok Penyiram Tanaman 05 sebanyak 10");
25          p3.kurangistok(jumlah:10);
26          p3.tampilkanInfo();
27
28          CreditBy.print();
29      }
30  }
31

```

HASIL EKSEKUSI

```
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum\week2-class-object\src\main\java> java com.upb.agripos.MainProduk
=== Info Awal Produk ===
Kode: HLD-005, Nama: Benih Coklat 50, Harga: 90000.0, Stok: 70
Kode: SVA-007, Nama: Pupuk Kandang 30kg, Harga: 70000.0, Stok: 50
Kode: ZNA-020, Nama: Penyiram Tanaman 05, Harga: 40000.0, Stok: 20

=== Menambah Stok Produk ===
Menambah stok Benih Coklat 50 sebanyak 25
Berhasil menambah stok 25 unit. Stok sekarang: 95
Kode: HLD-005, Nama: Benih Coklat 50, Harga: 90000.0, Stok: 95

=== Mengurangi Stok Produk ===
Mengurangi stok Penyiram Tanaman 05 sebanyak 10
Berhasil mengurangi stok 10 unit. Stok sekarang: 10
Kode: ZNA-020, Nama: Penyiram Tanaman 05, Harga: 40000.0, Stok: 10

credit by: 240202865 - hilda
PS C:\Users\Hilda\Documents\Praktikum\oop-202501-240202865\praktikum\week2-class-object\src\main\java>
```

ANALISIS

1. Jelaskan bagaimana kode berjalan

Program ini menerapkan konsep Class dan Object dalam pemrograman berorientasi objek (OOP).

- Class Produk berfungsi sebagai blueprint yang berisi atribut (kode, nama, harga, stok) serta method seperti tambahStok(), kurangiStok(), dan tampilkanInfo().
- Pada class MainProduk, objek p1, p2, dan p3 dibuat dari class Produk. Masing-masing objek menyimpan data produk yang berbeda.
- Program menampilkan informasi awal produk, lalu melakukan perubahan stok dengan memanggil method tambahStok() dan kurangiStok().
- Terakhir, class CreditBy digunakan untuk menampilkan identitas pembuat program.

Dengan cara ini, setiap produk dikelola secara terpisah melalui object yang berasal dari satu class yang sama, menunjukkan penerapan konsep instansiasi dan enkapsulasi.

2. Perbedaan pendekatan minggu ini dibanding minggu sebelumnya:

Pada minggu ini, pendekatan yang digunakan adalah Object-Oriented Programming (OOP), sedangkan minggu sebelumnya masih menggunakan pemrograman prosedural.

3. Kendala yang dihadapi dan cara mengatasinya:

- Kendala: Error "file not found" atau "class not found" saat menjalankan program.

Penyebab: Struktur folder atau deklarasi package belum sesuai dengan lokasi file.

Solusi: Memastikan struktur direktori sesuai dengan package di awal file (package com.upb.agripos.model;) dan menjalankan perintah dari folder yang tepat.

KESIMPULAN

Dengan menerapkan konsep class dan object, program menjadi lebih terstruktur, mudah dipahami, dan efisien. Mahasiswa dapat memahami cara membuat dan menggunakan class serta object untuk mengelola data dengan prinsip enkapsulasi.

CHEKLIST 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.

QUIZ

1. Mengapa atribut sebaiknya dideklarasikan sebagai private dalam class?

Jawaban:

Atribut sebaiknya dideklarasikan sebagai private untuk menjaga keamanan data (data hiding) agar tidak dapat diakses atau diubah secara langsung dari luar class. Hal ini mencegah perubahan nilai atribut secara tidak sengaja dan menjaga konsistensi data dalam program.

2. Apa fungsi getter dan setter dalam enkapsulasi?

Jawaban:

Getter digunakan untuk mengambil nilai dari atribut yang bersifat private, sedangkan setter digunakan untuk mengubah atau menetapkan nilainya dengan cara yang terkontrol. Keduanya berfungsi sebagai penghubung antara data dalam class dan bagian lain dari program tanpa melanggar prinsip enkapsulasi.

3. Bagaimana cara class Produk mendukung pengembangan aplikasi POS yang lebih kompleks?

Jawaban:

Class Produk menjadi dasar untuk mengelola data barang pada sistem POS, seperti kode produk, nama, harga, dan stok. Dengan struktur ini, pengembang dapat dengan mudah menambah fitur seperti perhitungan total transaksi, manajemen inventori, dan laporan penjualan tanpa harus mengubah banyak bagian dari kode utama.