



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

Jobsheet 02

Class dan Object

1. Kompetensi

- Mahasiswa dapat memahami deskripsi dari class dan object
- Mahasiswa memahami implementasi dari class
- Mahasiswa dapat memahami implementasi dari attribute
- Mahasiswa dapat memahami implementasi dari method
- Mahasiswa dapat memahami implementasi dari proses instansiasi
- Mahasiswa dapat memahami implementasi dari try-catch
- Mahasiswa dapat memahami proses pemodelan class diagram menggunakan *UML*

2. Pendahuluan

2.1 Class dan Object

Pada pertemuan sebelumnya anda sudah diberikan banyak penjelasan secara *semantic* (makna) mengenai deskripsi dari class dan object. Secara singkat class adalah abstraksi dari sebuah object (nyata ataupun tdk nyata) (roger s pressman). Apabila kita ingin membuat class mahasiswa, maka kita perlu melakukan abstraksi (mengindikasikan bagian – bagian penting yang merepresentasikan benda itu sendiri) dari object mahasiswa itu sendiri. Contoh salah satu attribute yang mengidentifikasi jika seseorang itu mahasiswa adalah **Nim** (Nomor Induk Mahasiswa), dan **Nim** tidak akan anda temui pada attribute dosen. Selain attribute abstraksi juga digunakan untuk behavior (perilaku) , contoh salah satu perilaku yang bisa dilakukan oleh mahasiswa adalah mengikuti **UAS**, dan anda juga tidak akan pernah menemui perilaku tersebut pada object dosen. Oleh karena itu sangat mudah untuk seorang perancang system dalam memodelkan sebuah class dari sebuah object tertentu.

Setelah kita memahami secara semantic pengertian dari class dan object, maka langkah selanjutnya adalah bagaimana cara melakukan implementasi class pada pendekatan Object Oriented Programming, terutama pada bahasa pemrograman java. Berikut adalah sintaks dari deklarasi class pada pemrograman java :



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

```
<modifier> class <nama_class> {  
    //deklarasi atribut dan method  
}
```

Aturan penulisan class adalah sebagai berikut:

1. Berupa kata benda,
2. Diawali dengan **HURUF BESAR**,
3. Jika terdiri dari lebih dari 1 kata, maka antar kata satu dengan kata yang lain digandeng, dan tiap huruf awal dari tiap kata menggunakan **HURUF BESAR**.

Untuk Access Modifier tidak dibahas pada jobsheet ini, melainkan akan dibahas pada jobsheet berikutnya.

Contoh deklarasi class:

```
public class Mahasiswa{  
  
}
```

2.2 Attribute

Untuk melakukan pendeklarasian *attribute* dapat dilakukan dengan sintaks sebagai berikut:

```
<modifier> <tipe> <nama_atribut> ;
```

Aturan penulisan atribut adalah sebagai berikut:

1. Berupa kata benda,
2. Diawali dengan HURUF KECIL,
3. Jika terdiri dari 2 atau lebih kata, kata pertama diawali HURUF KECIL sedangkan kata selanjutnya diawali HURUF BESAR. Dan antar kata disambung tidak (dipisah).



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

Contoh deklarasi *attribute*:

```
public class Mahasiswa{  
  
    public int nim;  
    public String nama;  
    public String alamat;  
    public float luas;  
  
}
```

Nb : *attribute* yang dituliskan dengan **huruf tebal**.

2.3 Method

Method adalah suatu blok dari program yang berisi kode program nama dan properti yang dapat digunakan kembali. Method dapat mempunyai nilai balik atau tidak. Method yang tidak mempunyai nilai balik dipanggil dalam pernyataan yang akan dikerjakan, sedangkan method yang mempunyai nilai balik dipanggil dari suatu ekspresi. Kata kunci untuk mengembalikan/mengeluarkan suatu nilai adalah return

Method dengan tipe data void, berarti tidak memiliki nilai balik, berarti tidak memerlukan kata kunci return di dalamnya. Method dengan tipe data bukan void, berarti memerlukan suatu nilai balik, yaitu harus memerlukan return di dalamnya

Deklarasi method dapat dilakukan dengan sintaks sebagai berikut:

```
public class Mahasiswa {  
  
    <modifier> <tipe_data> <nama_metode> ([daftar_argumen]) {  
        //statement  
    }  
  
}
```



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

Contoh method dengan tipe void dan method yang mengembalikan nilai (*return*)

```
public void sayHello(){  
    System.out.println("Hello World!!");  
}
```

TIDAK PERLU
RETURN /
TIDAK ADA NILAI
KEMBALIAN

```
public int tambah (int a, int b){  
    int hasil = a+b;  
    return hasil;  
}
```

• TIPE DATA METHOD INT, BERARTI
METHOD TSB HARUS
MENGEMBALIKAN NILAI INT
• HARUS ADA RETURN

Aturan penulisan method adalah sebagai berikut:

1. Berupa kata kerja,
2. Diawali dengan **HURUF KECIL**,
3. Jika terdiri dari 2 atau lebih kata, kata pertama diawali HURUF KECIL sedangkan kata selanjutnya diawali HURUF BESAR. Dan antar kata disambung tidak (dipisah).

Contoh deklarasi method:

```
public void tampil(){  
  
    System.out.println ("Hallo PBO!!");  
}  
  
public int tambah(int a, int b){  
    return a+b;  
}
```



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

2.4 Object

Setelah Class dibuat, langkah selanjutnya adalah membuat Object. Proses pembuatan Object dari suatu Class disebut instansiasi. Format dasar instansiasi adalah sebagai berikut:

```
NamaClass namaObject = new NamaClass();
```

Proses membuat objek dari suatu class adalah INSTANSIASI, dan ditandai kata kunci new. Aturan penulisan objek adalah sama seperti penulisan atribut. Contoh :

```
Random r = new Random();  
Pegawai p2 = new Pegawai();  
Mahasiswa mhs1= new Mahasiswa();
```

2.5 Try – catch

Untuk menangani **error** di Java, digunakan sebuah statement yang bernama *try - catch*. Statement tersebut digunakan untuk mengurung eksekusi yang menampilkan *error* dan dapat membuat program tetap berjalan tanpa dihentikan secara langsung. *Error* yang ditangani oleh *try – catch* biasa disebut dengan **exception**.

Ada beberapa hal yang perlu diingat ketika akan menggunakan try - catch di Java:

1. Kita dapat membuat multiple try-catch,
2. Kita dapat menambahkan statement finally untuk menangani berbagai hal ketika error terjadi atau tidak,
3. Kita dapat membuat exception sendiri disamping menggunakan bawaan Java.

Untuk melihat hasil dari implementasi dari try-catch, maka kita perlu melakukan komparasi sintaks tanpa try-catch dengan sintaks yang menggunakan try-catch. Berikut percobaanya :



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

Tanpa menggunakan try-catch :

```
public class SourceErrorExp {  
    public static void main(String[]args){  
        System.out.println("awal program");  
  
        int x = 10;  
        x = x / 0;  
  
        System.out.println(x);  
        System.out.println("akhir program");  
    }  
}
```

Hasil :

```
$ javac SourceErrorExp.java  
$ java SourceErrorExp  
awal program  
Exception in thread "main" java.lang.ArithmeticException: / by zero  
    at SourceErrorExp.main(SourceErrorExp.java:7)
```

Berbeda bila kita kurung operasi pembagian nol diatas dengan try - catch, maka hasil eksekusi program akan sedikit berbeda:

```
public class TanpaTC {  
    public static void main(String[]args){  
        System.out.println("awal program");  
  
        int x = 10;  
        try{  
            x = x / 0;  
        }catch(Exception e){  
            e.printStackTrace();  
            System.out.println("error karena pembagian nol");  
        }  
    }  
}
```



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

```
        System.out.println(x);  
        System.out.println("akhir program");  
    }  
}
```

Hasil :

```
$ javac SourceErrorExp2  
$ java SourceErrorExp2  
awal program  
java.lang.ArithmeticException: / by zero  
    at SourceErrorExp2.main(SourceErrorExp2.java:10)  
error karena pembagian nol  
10  
akhir program
```

3. *Unified Modeling Language (UML)*

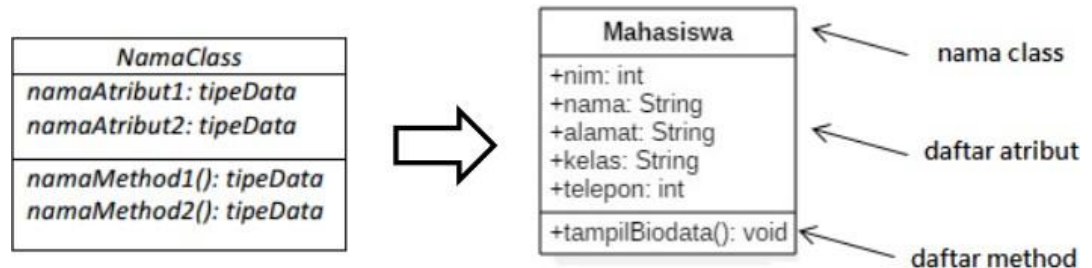
Unified Modeling Language (UML) adalah tujuan umum, perkembangan, bahasa pemodelan di bidang rekayasa perangkat lunak , yang dimaksudkan untuk menyediakan cara standar untuk memvisualisasikan desain sistem. UML menyediakan sembilan jenis diagram yaitu Diagram kelas (Class Diagram), Diagram paket (Package Diagram), Diagram use-case (Usecase Diagram), Diagram interaksi dan sequence (Sequence Diagram), Diagram komunikasi (Communication Diagram), Diagram statechart (Statechart Diagram), Diagram aktivitas (Activity Diagram), Diagram komponen (Component Diagram), dan Diagram deployment (deployment diagram). Pada materi ini yang akan dipelajari adalah diagram class (class diagram).

Class diagram adalah sebuah class yang menggambarkan struktur dan penjelasan class, paket, dan objek serta hubungan satu sama lain seperti pewarisan, asosiasi, dan lain-lain. Class diagram juga menjelaskan hubungan antar class dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan. Class juga memiliki 3 area pokok (utama) yaitu : nama, atribut, dan operasi. Nama berfungsi untuk member identitas pada sebuah kelas, atribut fungsinya adalah untuk member karakteristik pada data yang dimiliki suatu objek di dalam kelas, sedangkan



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

operasi fungsinya adalah memberikan sebuah fungsi ke sebuah objek. Berikut ini merupakan contoh dari class diagram:



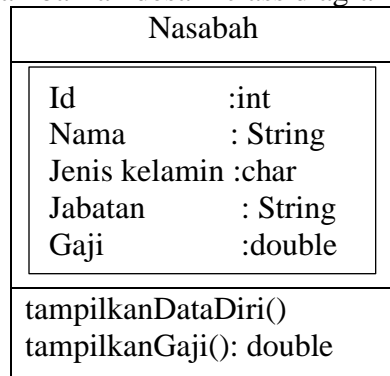
4. Percobaan

4.1 Percobaan 1: Membuat Class Diagram

Studi Kasus 1:

Dalam suatu perusahaan salah satu data yang diolah adalah data karyawan. Setiap karyawan memiliki id, nama, jenis kelamin, jabatan, jabatan, dan gaji. Setiap mahasiswa juga bisa menampilkan data diri pribadi dan melihat gajinya.

1. Gambarkan desain class diagram dari studi kasus 1!



2. Sebutkan Class apa saja yang bisa dibuat dari studi kasus 1!,

- Class Karyawan
- Class Main

3. Sebutkan atribut beserta tipe datanya yang dapat diidentifikasi dari masing-masing class dari studi kasus 1!



NAMA : Fransiscus Farrel Edric W

NIM : 2241720032

KELAS : 2C

- Id : int
- Nama :String
- jenisKelamin: char
- jabatan : String
- Gaji :int

4. Sebutkan method-method yang sudah anda buat dari masing-masing class pada studi kasus 1!

4.2 Percobaan 2: Membuat dan mengakses anggota suatu class

Studi Kasus 2:

Perhatikan class diagram dibawah ini. Buatlah program berdasarkan class diagram tersebut!



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

Mahasiswa
+nim: int
+nama: String
+alamat: String
+kelas: String
+tampilBiodata(): void

Langkah kerja:

1. Bukalah text editor atau IDE, misalnya Notepad ++ / netbeans.
2. Ketikkan kode program berikut ini:

```
1 public class Mahasiswa {  
2     public int nim;  
3     public String nama;  
4     public String alamat;  
5     public String kelas;  
6  
7     public void tampilBiodata() {  
8         System.out.println ("Nim      : "+nim);  
9         System.out.println ("Nama      : "+nama);  
10        System.out.println ("Alamat   : "+alamat);  
11        System.out.println ("Kelas   : "+kelas);  
12    }  
13 }
```

3. Simpan dengan nama file Mahasiswa.java.
4. Untuk dapat mengakses anggota-anggota dari suatu obyek, maka harus dibuat instance dari class tersebut terlebih dahulu. Berikut ini adalah cara pengaksesan anggota-anggota dari class Mahasiswa dengan membuka file baru kemudian ketikkan kode program berikut:



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

```
1 public class TestMahasiswa {  
2     public static void main (String args[]){  
3         Mahasiswa mhs1=new Mahasiswa();  
4         mhs1.nim=101;  
5         mhs1.nama="Lestari";  
6         mhs1.alamat="Jl. Vinolia No 1A";  
7         mhs1.kelas="1A";  
8         mhs1.tampilBiodata();  
9     }  
10 }
```

5. Simpan file dengan TestMahasiswa.java

6. Jalankan class TestMahasiswa

```
--- maven-compiler-plugin:3.1:compile (default-compile) {  
  Changes detected - recompiling the module!  
  Compiling 2 source files to C:\Users\Farrel Edric\OneDrive\Documents\workspace\test\src\main\java\com\example\test\mahasiswa\:  
  --- exec-maven-plugin:3.0.0:exec (default-cli) @ OOPP2 ---  
  NIM : 101  
  NAMA : Lestari  
  ALAMAT: JL. Vinolia NO 1A  
  KELAS : 1A  
  -----  
  BUILD SUCCESS  
  -----  
  Total time: 2.670 s  
  Finished at: 2023-09-07T08:52:36+07:00  
  -----
```

7. Jelaskan pada bagian mana proses pendeklarasian atribut pada program diatas!

- Nim : Atribut ini mewakili Nomor Induk Mahasiswa
- Nama : Atribut ini mewakili nama mahasiswa
- Alamat: Atribut ini mewakili alamat mahasiswa.
- Kelas : Atribut ini mewakili kelas di mana mahasiswa tersebut terdaftar.

8. Jelaskan pada bagian mana proses pendeklarasian method pada program diatas!

- Untuk mendeklarasikan method perlu dilakukan inisialisasi. Pada program di atas instansiasi terjadi pada kode program “mahasiswa mh1 = new mahasiswa();”

Kemudian dapat memanggil method sesuai nama yang diinginkan. Dan akan secara otomatis akan dijalankan di main.

9. Berapa banyak objek yang di instansiasi pada program diatas!

```
Mahasiswa mhs1=new Mahasiswa();
```

10. Apakah yang sebenarnya dilakukan pada sintaks program “mhs1.nim=101” ?

- Memberikan nilai 101 kepada atribut nim dari objek mhs1

11. Apakah yang sebenarnya dilakukan pada sintaks program “mhs1.tampilBiodata()” ?

- memanggil (menjalankan) method tampilBiodata() pada objek mhs1.
- Jadi, ketika baris ini dieksekusi, program akan menjalankan method tampilBiodata() pada objek



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

mhs1. Method ini akan mencetak informasi biodata mahasiswa (NIM, NAMA, ALAMAT, dan KELAS) ke layar menggunakan perintah System.out.println.

12. Instansiasi 2 objek lagi pada program diatas!

```
--- exec-maven-plugin:3.0.0:exec (default-cli) @ OOPP2 ---  
NIM : 101  
NAMA : Lestari  
ALAMAT: JL. Vinolia NO 1A  
KELAS : 1A  
NIM : 102  
NAMA : Edd  
ALAMAT: JL. Lapu-Lapu NO 2B  
KELAS : 2B  
NIM : 103  
NAMA : Bojem  
ALAMAT: JL. Poqe NO 3C  
KELAS : 3C
```

```
-----  
BUILD SUCCESS  
-----
```

4.3 Percobaan 3: Menulis method yang memiliki argument/parameter dan memiliki return

Langkah kerja:

1. Bukalah text editor atau IDE, misalnya Notepad ++ / netbeans.
2. Ketikkan kode program berikut ini:



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

```
1 public class Barang {  
2     public String namaBrg;  
3     public String jenisBrg;  
4     public int stok;  
5  
6     public void tampilBarang(){  
7         System.out.println ("Nama Barang : "+namaBrg);  
8         System.out.println ("Jenis Barang : "+jenisBrg);  
9         System.out.println ("Stok : "+stok);  
10    }  
11  
12    //method dengan argumen dan nilai balik (return)  
13    public int tambahStok(int brgMasuk){  
14        int stokBaru=brgMasuk+stok;  
15        return stokBaru;  
16    }  
17 }
```

3. Simpan dengan nama file Barang.java
4. Untuk dapat mengakses anggota-anggota dari suatu obyek, maka harus dibuat instance dari class tersebut terlebih dahulu. Berikut ini adalah cara pengaksesan anggota-anggota dari class Barang dengan membuka file baru kemudian ketikkan kode program berikut:

```
1 public class TestBarang{  
2     public static void main (String args[]){  
3         Barang brg1=new Barang();  
4         brg1.namaBrg="Pensil";  
5         brg1.jenisBrg="ATK";  
6         brg1.stok=10;  
7         brg1.tampilBarang();  
8         // menampilkan dan mengisi argumen untuk menambahkan stok barang  
9         System.out.println ("Stok Baru adalah " +brg1.tambahStok(20));  
10    }  
11 }
```

5. Simpan dengan nama file TestBarang.java
 6. Jalankan program tersebut!
 7. Apakah fungsi argumen dalam suatu method?
- Argumen dalam suatu method berfungsi sebagai input yang dibutuhkan oleh method contoh nya seperti pada program diatas yaitu argument nya adalah 20 , dengan angka 20 yang dikirimkan ke method sebagai argument maka kita akan memberitau method tambahStok untuk menambahkan 20 ke objek brg1



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

8. Ambil kesimpulan tentang kegunaan dari kata kunci return , dan kapan suatu method harus memiliki return!

- "return " digunakan untuk mengembalikan nilai dari suatu method, suatu method harus memiliki return saat method memiliki tipe data selain void yang perlu untuk mengembalikan nilai seperti hasil sebuah perhitungan



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

4.4 Tugas

1. Suatu toko persewaan video game salah satu yang diolah adalah peminjaman, dimana data yang dicatat ketika ada orang yang melakukan peminjaman adalah id, nama member, nama game, dan harga yang harus dibayar. Setiap peminjaman bisa menampilkan data hasil peminjaman dan harga yang harus dibayar. Buatlah class diagram pada studi kasus diatas!

Penjelasan:

- Harga yang harus dibayar diperoleh dari lama sewa x harga.
 - Diasumsikan 1x transaksi peminjaman game yang dipinjam hanya 1 game saja.
2. Buatlah program dari class diagram yang sudah anda buat di no 1!

Class peminjaman

```
public class Peminjaman {  
    private int id;  
    private String namaMember;  
    private String namaGame;  
    private int lamaPinjam;  
    private double harga;  
  
    public Peminjaman(int id, String namaMember, String namaGame,  
        this.id = id;  
        this.namaMember = namaMember;  
        this.namaGame = namaGame;  
        this.lamaPinjam = lamaPinjam;  
        this.harga = harga;  
    }  
  
    public double hitungHargaTotal() {  
        return lamaPinjam * harga;  
    }  
  
    public void tampilData() {  
        System.out.println("ID Peminjaman: " + id);  
        System.out.println("Nama Member: " + namaMember);  
        System.out.println("Nama Game: " + namaGame);  
        System.out.println("Lama Pinjam: " + lamaPinjam + " hari");  
        System.out.println("Harga Total: Rp " + hitungHargaTotal());  
    }  
}
```



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

Class test peminjaman

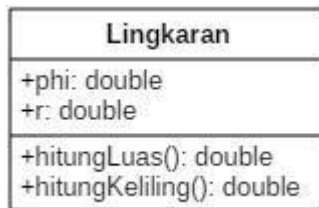
```
package oopp2;

/**
 *
 * @author Farrel Edric
 */
public class TestPeminjaman {
    public static void main(String[] args) {
        Peminjaman peminjaman1 =
            new Peminjaman( id:1, namaMember: "John", namaGame: "Game
Peminjaman peminjaman2 =
            new Peminjaman( id:2, namaMember: "Alice", namaGame: "Gan

        System.out.println( x: "Data Peminjaman 1:");
        peminjaman1.tampilData();

        System.out.println( x: "\nData Peminjaman 2:");
        peminjaman2.tampilData();
    }
}
```

3. Buatlah program sesuai dengan class diagram berikut ini:





NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

```
j --- exec-maven-plugin:3.0.0:exec (default-cli) @ OOPP2 ---
Masukkan jari-jari lingkaran: 8

Lingkaran
Jari-jari: 8.0
1. Hitung Luas
2. Hitung Keliling
3. Keluar
Pilih operasi (1/2/3): 1
Luas: 200.96
Masukkan jari-jari lingkaran: 2

Lingkaran
Jari-jari: 2.0
1. Hitung Luas
2. Hitung Keliling
3. Keluar
Pilih operasi (1/2/3): 3
- Keluar dari program.
-----
BUILD SUCCESS
-----
Total time: 12.524 s
- 2022-06-07T00:46:55+07:00
```

4. Buatlah program sesuai dengan class diagram berikut ini:



Deskripsi / Penjelasan :

- Nilai atribut hargaDasar dalam Rupiah dan atribut diskon dalam %
- Method hitungHargaJual() digunakan untuk menghitung harga jual dengan perhitungan berikut ini:

harga jual = harga dasar – (diskon x harga dasar)



NAMA : Fransiscus Farrel Edric W
NIM : 2241720032
KELAS : 2C

- Class barang

```
public class Barang {
    private String kode;
    private String namaBarang;
    private int hargaDasar;
    private float diskon;

    // Konstruktor
    public Barang(String kode, String namaBarang, int hargaDasar,
        float diskon) {
        this.kode = kode;
        this.namaBarang = namaBarang;
        this.hargaDasar = hargaDasar;
        this.diskon = diskon;
    }

    // Method untuk menghitung harga jual
    public int hitungHargaJual() {
        float diskonDecimal = diskon / 100; // Mengubah diskon dari
        int hargaJual = (int) (hargaDasar - (diskonDecimal * hargaDasar));
        return hargaJual;
    }

    // Method untuk menampilkan data barang
    public void tampilData() {
        System.out.println("Kode: " + kode);
        System.out.println("Nama Barang: " + namaBarang);
        System.out.println("Harga Dasar: Rp " + hargaDasar);
        System.out.println("Diskon: " + diskon + "%");
    }
}
```

- Class main

```
public class TestBarang {
    public static void main(String[] args) {
        // Membuat objek barang
        Barang barang1 = new Barang(kode: "001", namaBarang: "Laptop",
            hargaDasar: 7000000, diskon: 10);
        Barang barang2 = new Barang(kode: "002", namaBarang: "Smartphone",
            hargaDasar: 3500000, diskon: 5);

        // Menampilkan data barang
        System.out.println("Data Barang 1:");
        barang1.tampilData();

        System.out.println("Data Barang 2:");
        barang2.tampilData();
    }
}
```

- Run

```
--- exec-maven-plugin:3.0.0:exec (default)
Data Barang 1:
Kode: 001
Nama Barang: Laptop
Harga Dasar: Rp 7000000
Diskon: 10.0%
Harga Jual: Rp 6300000

Data Barang 2:
Kode: 002
Nama Barang: Smartphone
Harga Dasar: Rp 3500000
Diskon: 5.0%
Harga Jual: Rp 3325000

BUILD SUCCESS
```



NAMA : Fransiscus Farrel Edric W

NIM : 2241720032

KELAS : 2C

- Method tampilData() digunakan untuk menampilkan nilai dari kode, namaBarang, hargaDasar, diskon dan harga jual.