

**LAPORAN PRAKTIKUM
PEMROGRAMAN BERORIENTASI OBJEK**

**MODUL III
KELAS AND OBJEK**



**Disusun Oleh:
Nisrina Nurhaliza
21102218
S1 IF-09-0**

**Dosen Pengampu :
Dedy Agung Prabowo, S.Kom., M.Kom**

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2023**

BAB I

TUJUAN PRAKTIKUM

Setelah mengikuti praktikum ini, mahasiswa diharapkan dapat :

1. Mahasiswa diharapkan mampu memahami mengenai objek dan kelas
2. Mahasiswa diharapkan mampu menerjemahkan objek dan kelas dalam bahasa pemrograman

BAB II

TOOLS

1. IntelliJ Community Edition
2. Java SE Development Kit 19

BAB III

DASAR TEORI

a. Kelas dan Objek

Objek adalah kesatuan entitas (benda), baik yang berwujud nyata ataupun hanya suatu sistem atau konsep yang memiliki sifat karakteristik dan fungsi. Dalam kehidupan sehari-hari kita sering menjumpai banyak objek dengan jenis yang sama, contohnya sepeda yang Anda miliki adalah salah satu jenis dari sepeda yang ada di dunia. Sepeda Anda adalah **instance** dari kelas yang disebut kelas sepeda. **Objek** memiliki status dan perilaku. **Objek** adalah turunan dari class, sebuah objek bisa didefinisikan setelah suatu class ada terlebih dahulu. **Objek** membungkus suatu data dan fungsi menjadi suatu unit dalam suatu program, objek merupakan dasar dari modularitas dan struktur dalam sebuah program berorientasi objek.

Objek adalah suatu abstraksi dari suatu problem. Sedangkan **kelas** adalah kumpulan objek yang memiliki atribut dan perilaku yang sama (karakteristik sama). Setiap objek memiliki nilai atribut/state yang unik yang membedakannya dengan objek lain dari kelas yang sama, dan objek memiliki perilaku/behaviour untuk mengakses atribut/state yang dimilikinya.

Kelas bertugas untuk mengumpulkan prosedur/fungsi dan variabel dalam satu tempat. **Kelas** merupakan blueprint dari sebuah objek atau cetakan untuk membuat objek. **Kelas** adalah pemodelan dari objek yang berisi informasi (aturan) tentang sifat karakteristik (data) dan tingkah laku (method) yang dimiliki oleh objek tersebut. Kelas dapat dianalogikan sebagai struktur data dari objek. Perbedaan kelas pada pemrograman berorientasi objek dengan struktur data pada pemrograman terstruktur adalah bahwa kelas pada pemrograman berorientasi objek tidak hanya berisi data saja tetapi juga fungsi-

fungsi yang mengaksesnya, sehingga data dan fungsi harus dirancang secara bersamaan (**kelas = struktur data + fungsi**).

b. Mendefinisikan Kelas

Elemen – elemen dasar dalam mendefinisikan kelas :

1. Fields/Variable

Field atau variable adalah implementasi dari atribut suatu objek. Field atau variable digunakan untuk menyimpan data dari objek. Jenis fields atau variable :

- Instance Variabel

Setiap objek memiliki salinan sendiri dan salinan tersebut memiliki nilai masing masing.

- Class Variabel

Suatu kelas hanya memiliki satu variable jenis ini dan digunakan bersama oleh semua objek dari kelas tersebut.

2. Constructor/Konstruktor

Konstruktor adalah method yang berfungsi untuk menginisialisasi variabel-variabel instance yang akan dimiliki oleh objek. Method konstruktor harus memiliki nama yang sama dengan nama kelas. Konstruktor ini dipanggil pada saat proses instansiasi kelas menjadi objek.

Kegunaan constructor :

- Mengalokasikan ruang bagi sebuah objek
- Memberikan nilai awal terhadap anggota data suatu objek
- Membentuk tugas-tugas umum lainnya

Perlu diketahui :

- Constructor tidak mempunyai nilai balik (bahkan tanpa void)
- Constructor harus diletakkan pada bagian public

Karakteristik constructor :

- Tidak pernah memiliki nilai balikan
- Memiliki nama sama dengan nama kelas

3. Method

Method merupakan fungsi-fungsi implementasi perilaku objek untuk mengakses atribut-atributnya. Method dalam pemrograman Java adalah sebuah blok program terpisah (diluar program utama) yang kita gunakan untuk menyelesaikan masalah khusus. Tujuannya untuk memecah program kompleks menjadi bagian-bagian kecil sehingga nantinya dapat kita gunakan secara berulang-ulang tanpa harus menulis baris kode yang sama.

Method adalah kumpulan program yang mempunyai nama. Program harus dibungkus dalam method. Dengan method kita bias memanggil kumpulan program hanya dengan memanggil nama methodnya, pekerjaan jadi lebih singkat dan tidak boros menuliskan program, program menjadi lebih terstruktur, praktis, dan efisien.

Sebagai ilustrasi, kita ingin membuat kelas **Buku**, yang memiliki objek diantaranya novel dan fiksi. Karena kelas

merupakan abstraksi dari objek, maka pemilihan atribut haruslah yang dapat merepresentasikan objek secara umum. Beberapa atribut/property yang digunakan, yaitu judul dan pengarang yang bertipe char/string, kemudian jumlah (untuk mengetahui berapa banyaknya buku maka bertipe integer). Selain data yang telah didefinisikan sebelumnya, kita juga dapat menentukan method yang dimiliki oleh kelas tersebut diantaranya fungsi untuk mengisi data dan menampilkan data.

Sumber :

1. <https://www.dicoding.com/blog/apa-itu-oop-pada-java-beserta-contohnya/> . Diakses pada tanggal 4 April 2023.
2. <https://www.masgani.com/class-dan-object-pada-pemrograman-java/> Diakses pada tanggal 4 April 2023.
3. <https://kodedasar.com/blog/method-java/> . Diakses pada tanggal 4 April 2023.

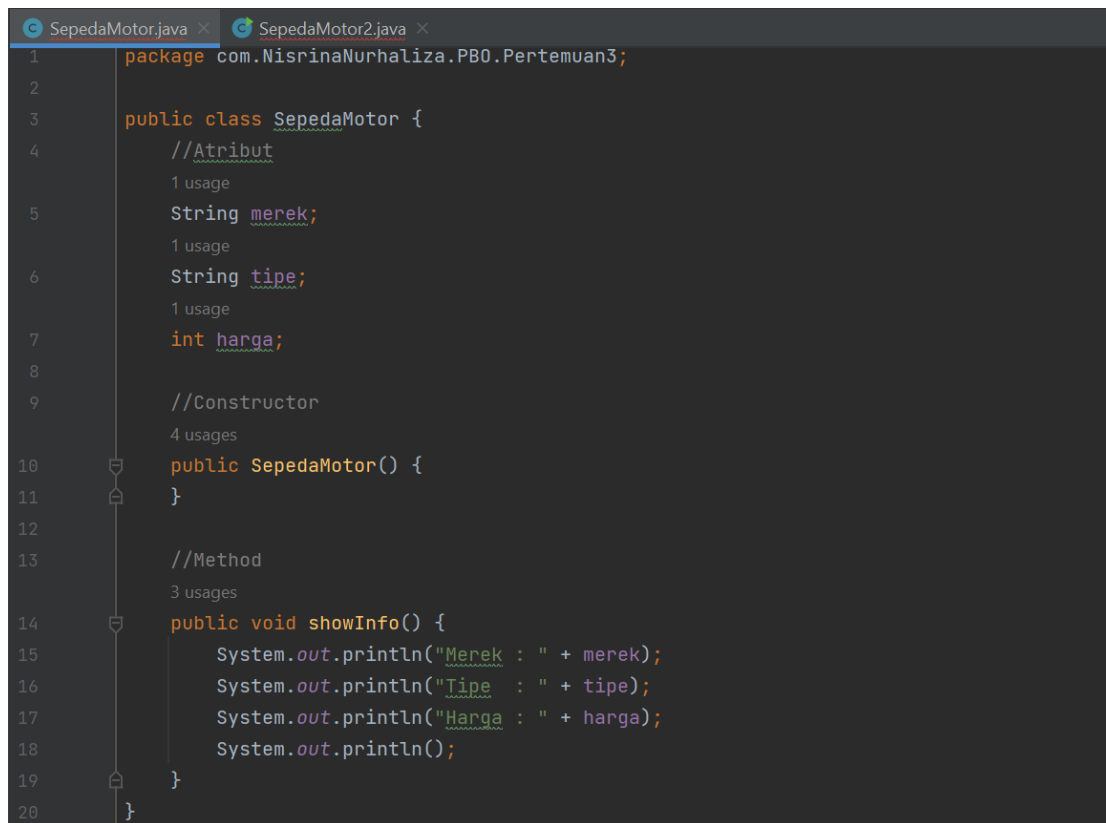
BAB IV

GUIDED

1. Membuah Class Baru

Sebuah class baru dengan nama **SepedaMotor.java**

- Source Code



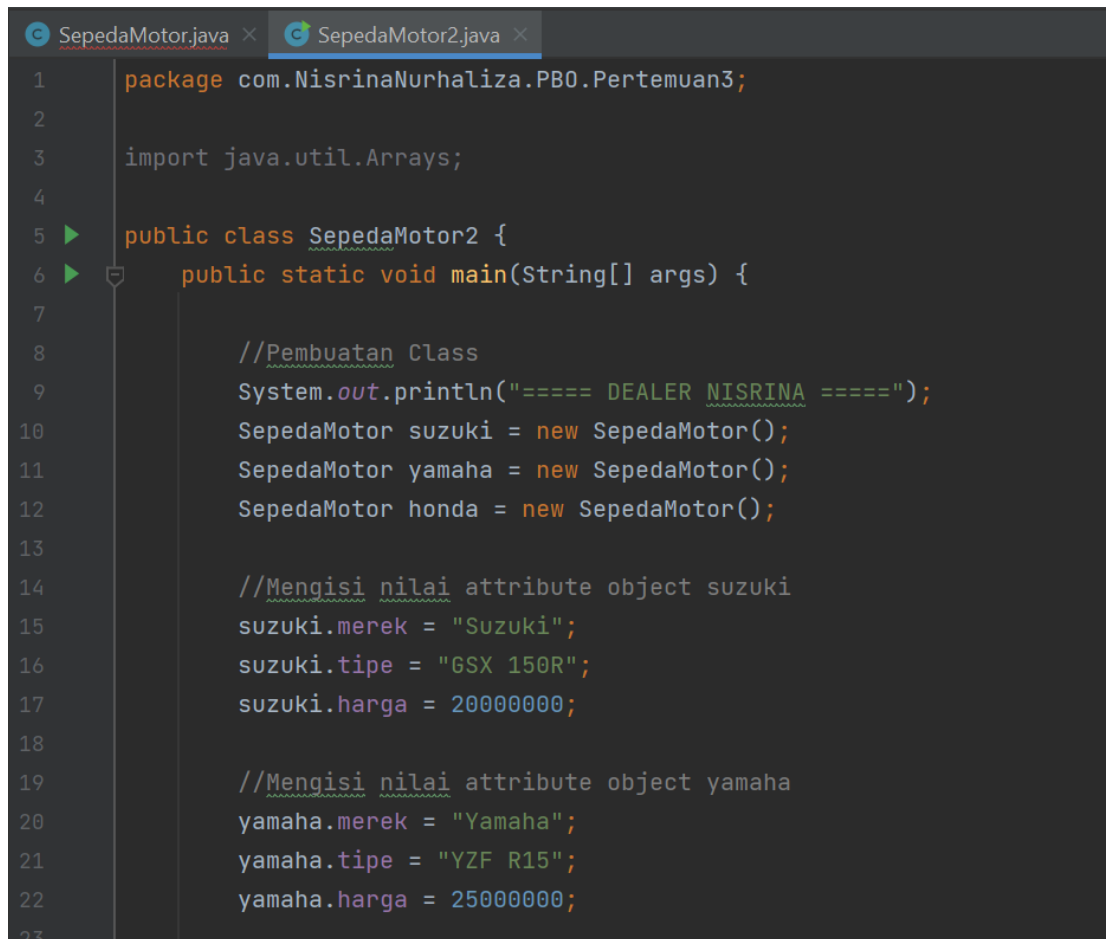
```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 public class SepedaMotor {
4     //Atribut
5     String merek;
6     String tipe;
7     int harga;
8
9     //Constructor
10    public SepedaMotor() {
11    }
12
13    //Method
14    public void showInfo() {
15        System.out.println("Merek : " + merek);
16        System.out.println("Tipe : " + tipe);
17        System.out.println("Harga : " + harga);
18        System.out.println();
19    }
20 }
```

Dapat dilihat, class tersebut tidak memiliki method berikut :

```
public static void main(String[] args){ }
```

Ini dikarenakan class yang barusan kita buat akan kita gunakan sebagai template atau tipe data baru. Sebelumnya, class-class yang kita buat, kita gunakan sebagai program utama atau main program. Agar program kita bias dijalankan, kita memerlukan sebuah program utama. Untuk itu, buatlah sebuah

Java class baru dan beri nama **SepedaMotor2.java** lalu salin dan jalankan kode berikut :



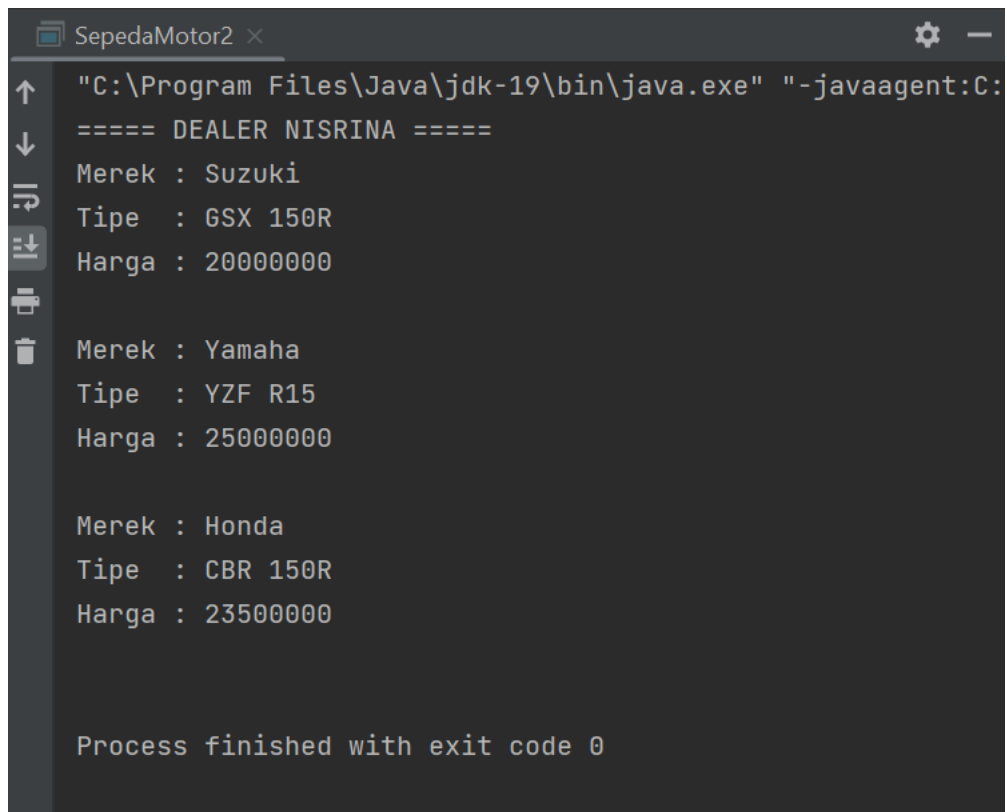
```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 import java.util.Arrays;
4
5 public class SepedaMotor2 {
6     public static void main(String[] args) {
7
8         //Pembuatan Class
9         System.out.println("===== DEALER NISRINA =====");
10        SepedaMotor suzuki = new SepedaMotor();
11        SepedaMotor yamaha = new SepedaMotor();
12        SepedaMotor honda = new SepedaMotor();
13
14        //Mengisi nilai attribute object suzuki
15        suzuki.merek = "Suzuki";
16        suzuki.tipe = "GSX 150R";
17        suzuki.harga = 200000000;
18
19        //Mengisi nilai attribute object yamaha
20        yamaha.merek = "Yamaha";
21        yamaha.tipe = "YZF R15";
22        yamaha.harga = 250000000;
23    }
```

```

23
24      ////Mengisi nilai attribute object Honda
25      honda.merek = "Honda";
26      honda.tipe = "CBR 150R";
27      honda.harga = 235000000;
28
29      suzuki.showInfo();
30      yamaha.showInfo();
31      honda.showInfo();
32  }
33  }

```

- Output Program



```

SepedaMotor2 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:
===== DEALER NISRINA =====
Merek : Suzuki
Tipe  : GSX 150R
Harga : 20000000

Merek : Yamaha
Tipe  : YZF R15
Harga : 25000000

Merek : Honda
Tipe  : CBR 150R
Harga : 23500000

Process finished with exit code 0

```

- Deskripsi Program

Program di atas merupakan contoh implementasi Class dan Objek pada Java. Di dalam program tersebut terdapat dua Class yaitu SepedaMotor dan SepedaMotor2. Class SepedaMotor digunakan untuk menggambarkan sebuah sepeda motor yang memiliki atribut atau variabel berupa merk, tipe, dan harga. Selain itu, pada Class SepedaMotor juga terdapat sebuah method bernama showInfo() yang berfungsi untuk menampilkan informasi mengenai sepeda motor yang dimiliki. Class SepedaMotor2 digunakan sebagai Class utama atau entry point dari program. Pada Class SepedaMotor2, terdapat pembuatan tiga objek SepedaMotor yaitu suzuki, yamaha, dan honda. Nilai atribut atau variabel dari masing-masing objek tersebut diisi dengan nilai yang sesuai. Setelah itu, method showInfo() dipanggil pada setiap objek untuk menampilkan informasi mengenai sepeda motor yang dimiliki.

2. Constructor Ber-Parameter

Daripada melakukan pengisian attribute satu per satu seperti diatas, kita bias melakukan sedikit modifikasi untuk mempermudah kerja kita.

Kembali ke **SepedaMotor.java** lalu tambahkan constructor method baru di kodenya. (**Keterangan** : Anda dapat membuat method dengan nama sama tanpa menyebabkan error asalkan method-method tersebut memiliki parameter berbeda).

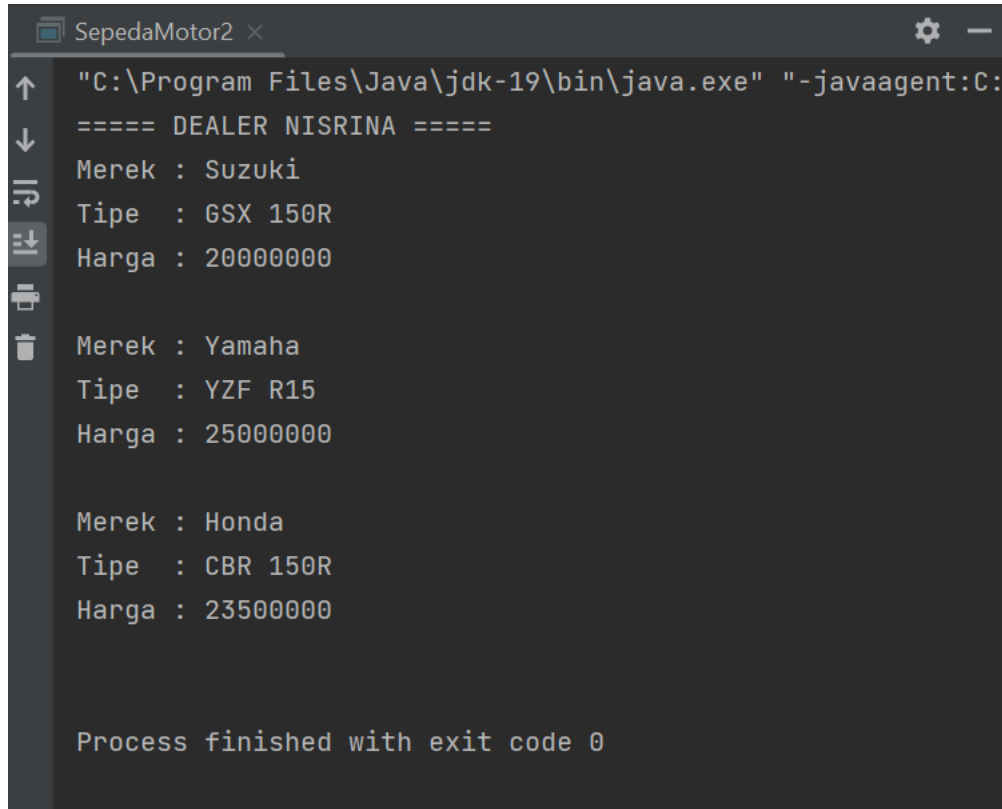
- Source Code

```
//Constructor ber parameter
3 usages
public SepedaMotor(String merek, String tipe, int harga) {
    this.merek = merek;
    this.tipe = tipe;
    this.harga = harga;
}
}
```

Lalu kembali ke **SepedaMotor2.java** dan lakukan modifikasi terhadap source code hingga akhirnya terlihat seperti ini :

```
4
5 public class SepedaMotor2 {
6     public static void main(String[] args) {
7
8         //Pembuatan Class
9         System.out.println("===== DEALER NISRINA =====");
10        SepedaMotor suzuki = new SepedaMotor( merek: "Suzuki", tipe: "GSX 150R", harga: 2000000);
11        SepedaMotor yamaha = new SepedaMotor( merek: "Yamaha", tipe: "YZF R15", harga: 25000000);
12        SepedaMotor honda = new SepedaMotor( merek: "Honda", tipe: "CBR 150R", harga: 23500000);
13
14        //Menjalankan method showInfo()
15        suzuki.showInfo();
16        yamaha.showInfo();
17        honda.showInfo();
18    }
19 }
```

- Output Program



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-19\bin\javaagent.jar" -Djava.class.path=.\SepedaMotor2.jar -Djava.main.class=SepedaMotor2.Main -jar SepedaMotor2.jar

==== DEALER NISRINA =====

Merek : Suzuki
Tipe  : GSX 150R
Harga : 20000000

Merek : Yamaha
Tipe  : YZF R15
Harga : 25000000

Merek : Honda
Tipe  : CBR 150R
Harga : 23500000

Process finished with exit code 0
```

- Deskripsi Program

Program di atas merupakan pengembangan dari program sebelumnya dengan cara membuat objek dengan constructor yang memiliki parameter pada Class **SepedaMotor**. Dalam program ini, constructor SepedaMotor yang memiliki tiga parameter digunakan untuk mengisi nilai atribut objek yang dibuat pada saat objek dibuat. setiap objek langsung diisi dengan nilai yang diberikan pada saat objek dibuat dengan menggunakan constructor berparameter. Hal ini membuat proses pengisian nilai atribut objek menjadi lebih mudah dan singkat. Dalam penggunaan constructor, kita dapat menyederhanakan proses pembuatan objek dan menghindari kesalahan dalam

pengisian nilai atribut objek karena setiap nilai atribut diinisialisasi saat objek dibuat.

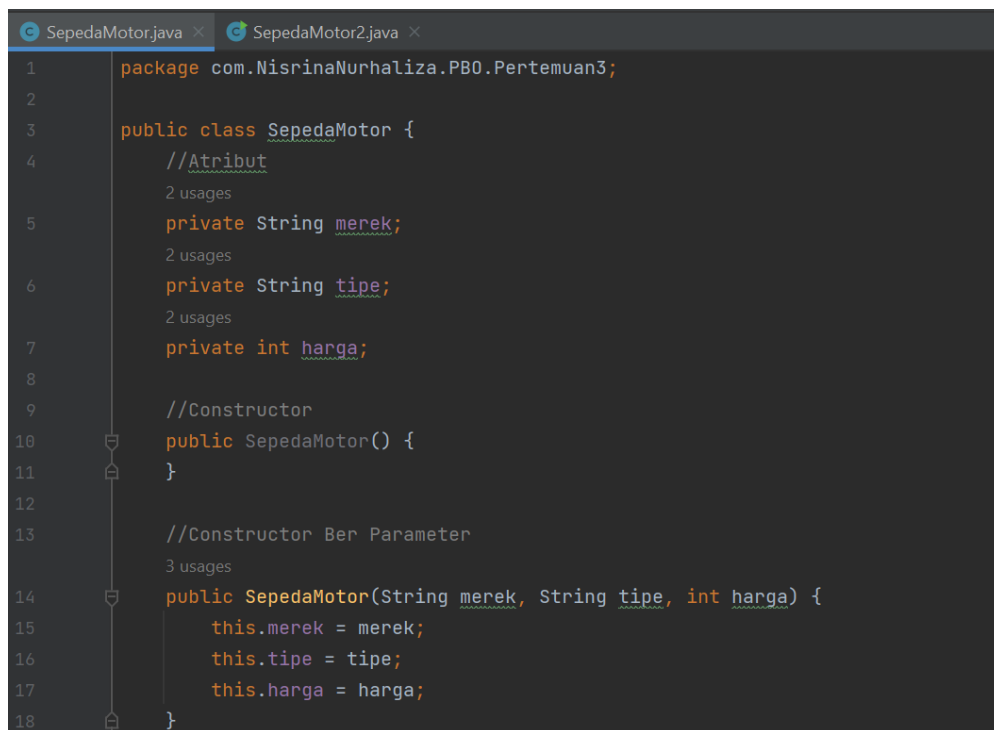
3. Modifier

Di Java, ada yang namanya modifier untuk menentukan apakah sebuah method atau attribute dapat diakses oleh class atau file lain atau tidak. Terdapat tiga modifier di bahasa Java :

1. Public (+) : Attribute atau method tersebut dapat diakses oleh class atau file itu manapun.
2. Private (-) : Attribute atau method tersebut hanya dapat diakses oleh class atau file itu sendiri.
3. Protected (#) : Attribute atau method tersebut hanya dapat diakses oleh class atau file yang berada di package yang sama.

Lakukan modifikasi ke class **SepedaMotor.java**

- Source Code



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 public class SepedaMotor {
4     //Atribut
5     2 usages
6     private String merek;
7     2 usages
8     private String tipe;
9     2 usages
10    private int harga;
11
12    //Constructor
13    public SepedaMotor() {
14    }
15
16    //Constructor Ber Parameter
17    3 usages
18    public SepedaMotor(String merek, String tipe, int harga) {
19        this.merek = merek;
20        this.tipe = tipe;
21        this.harga = harga;
22    }
23 }
```

```

//Method
1 usage
public void showInfo() {
    System.out.println("Merek : " + merek);
    System.out.println("Tipe : " + tipe);
    System.out.println("Harga : " + harga);
    System.out.println();
}
}

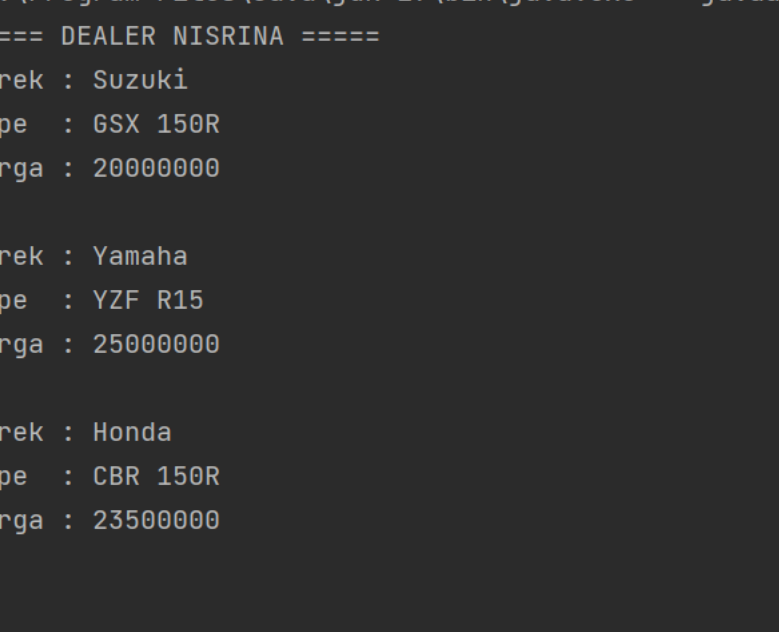
```

```

4
5 ▶ public class SepedaMotor2 {
6 ▶     public static void main(String[] args) {
7
8         //Pembuatan Class
9         System.out.println("===== DEALER NISRINA =====");
10        SepedaMotor suzuki = new SepedaMotor( merek: "Suzuki", tipe: "GSX 150R", harga: 20000000);
11        SepedaMotor yamaha = new SepedaMotor( merek: "Yamaha", tipe: "YZF R15", harga: 25000000);
12        SepedaMotor honda = new SepedaMotor( merek: "Honda", tipe: "CBR 150R", harga: 23500000);
13
14        //Menjalankan method showInfo()
15        suzuki.showInfo();
16        yamaha.showInfo();
17        honda.showInfo();
18    }
19 }

```


- Output Program



```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\Java\jdk-19\bin\java.exe" "-jar SepedaMotor2.jar"

===== DEALER NISRINA =====

Merek : Suzuki
Tipe   : GSX 150R
Harga  : 200000000

Merek : Yamaha
Tipe   : YZF R15
Harga  : 250000000

Merek : Honda
Tipe   : CBR 150R
Harga  : 235000000

Process finished with exit code 0
```

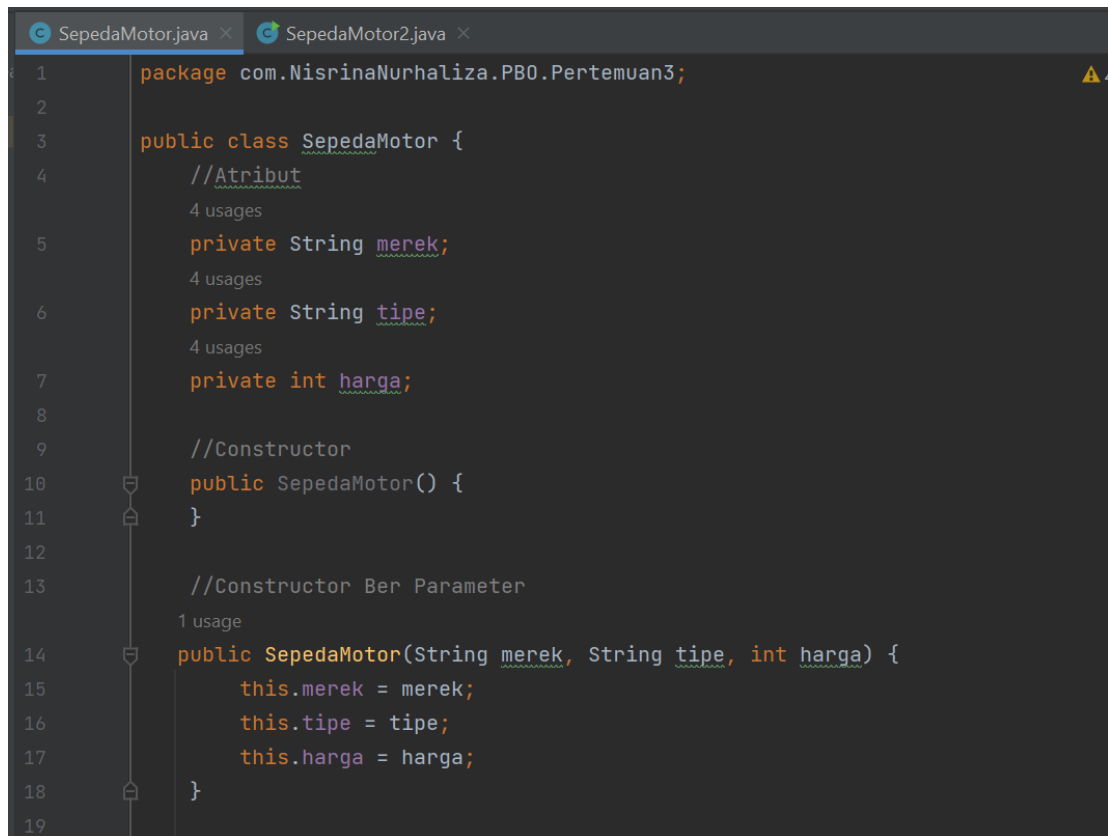
- Deskripsi Program

Program di atas merupakan contoh implementasi class **SepedaMotor** pada bahasa pemrograman Java yang memiliki tiga atribut (merek, tipe, dan harga) dengan hak akses private, sehingga hanya dapat diakses oleh method yang berada pada class tersebut. Terdapat dua constructor, yaitu constructor default tanpa parameter dan constructor dengan parameter merek, tipe, dan harga. Method `showInfo` digunakan untuk menampilkan informasi mengenai objek **SepedaMotor** seperti merek, tipe, dan harga.

4. Getter Dan Setter

Normalnya mencoba mengakses private attribute di class lain tentunya akan menghasilkan error. Namun jika Anda tetap ingin melakukannya, Anda bisa mencoba pendekatan Getter & Setter.

- Source Code :



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 public class SepedaMotor {
4     //Atribut
4 usages
5     private String merek;
6     private String tipe;
7     private int harga;
8
9     //Constructor
10    public SepedaMotor() {
11    }
12
13    //Constructor Ber Parameter
14    1 usage
15    public SepedaMotor(String merek, String tipe, int harga) {
16        this.merek = merek;
17        this.tipe = tipe;
18        this.harga = harga;
19    }
```

```
SepedaMotor.java x SepedaMotor2.java x
20 //Method
21 2 usages
22 public String getMerek() {
23     return merek;
24 }
25
26 public void setMerek(String merek) {
27     this.merek = merek;
28 }
29
30 2 usages
31 public String getTipe() {
32     return tipe;
33 }
34
35 1 usage
36 public void setType(String tipe) {
37     this.tipe = tipe;
38 }
39
40 public int getHarga() {
41     return harga;
42 }
```

```

40
41     public void setHarga(int harga) {
42         this.harga = harga;
43     }
44
45     1 usage
46     public void showInfo() {
47         System.out.println("Merek : " + merek);
48         System.out.println("Tipe : " + tipe);
49         System.out.println("Harga : " + harga);
50         System.out.println();
51     }

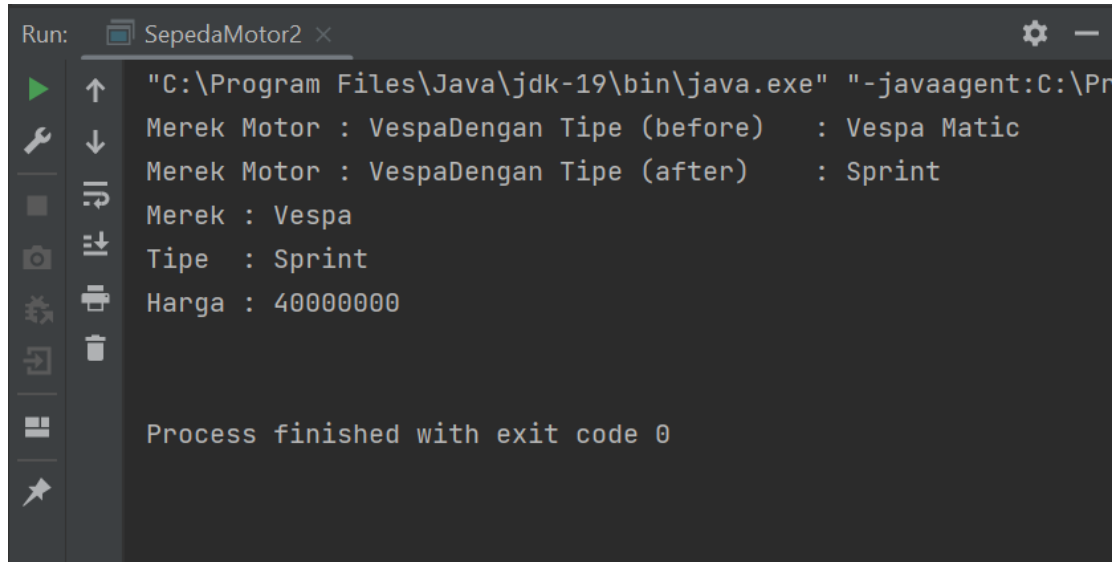
```

```

SepedaMotor.java x SepedaMotor2.java x
5  ▶ public class SepedaMotor2 {
6  ▶  public static void main(String[] args) {
7
8      SepedaMotor vespa = new SepedaMotor( merek: "Vespa", tipe: "Vespa Matic", harga: 40000000);
9
10
11      //Sebelum Perubahan
12      System.out.println("Merek Motor : " + vespa.getMerek() +
13          "Dengan Tipe (before) : " + vespa.getTipe()
14      );
15
16      //Melakukan Proses Perubahan Tipe Motor
17      vespa.setTipe("Sprint");
18
19      //Lihat Hasil Perubahan
20      System.out.println("Merek Motor : " + vespa.getMerek() +
21          "Dengan Tipe (after) : " + vespa.getTipe()
22      );
23
24      vespa.showInfo();
25  }

```

- Output :



```
Run: SepedaMotor2 x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Pr
Merek Motor : VespaDengan Tipe (before) : Vespa Matic
Merek Motor : VespaDengan Tipe (after) : Sprint
Merek : Vespa
Tipe : Sprint
Harga : 40000000

Process finished with exit code 0
```

- Deskripsi Program

Pada program diatas kelas **SepedaMotor2** digunakan untuk melakukan uji coba terhadap method pada kelas **SepedaMotor**. Dalam kelas **SepedaMotor2** terdapat pembuatan objek dari kelas **SepedaMotor** dengan menggunakan constructor ber- parameter. Kemudian dilakukan pengujian pada method getter dan setter untuk atribut "tipe" dengan mengubah nilai atribut "tipe" menggunakan setter dan menampilkan nilai atribut "tipe" menggunakan getter sebelum dan sesudah perubahan. Selain itu, juga dilakukan pemanggilan method "showInfo()" untuk menampilkan informasi dari objek sepeda motor.

UNGUIDED

1. Disebuah koperasi sekolah menjual item sebagai berikut :

Nama Barang	Harga Barang (Satuan)	Jumlah Barang
Buku Tulis	5000	10
Pensil	3000	20

Buatlah **Class** dengan nama **Koperasi.java** untuk menampung beberapa method yang diperlukan seperti **setter** dan **getter** sebagai berikut :

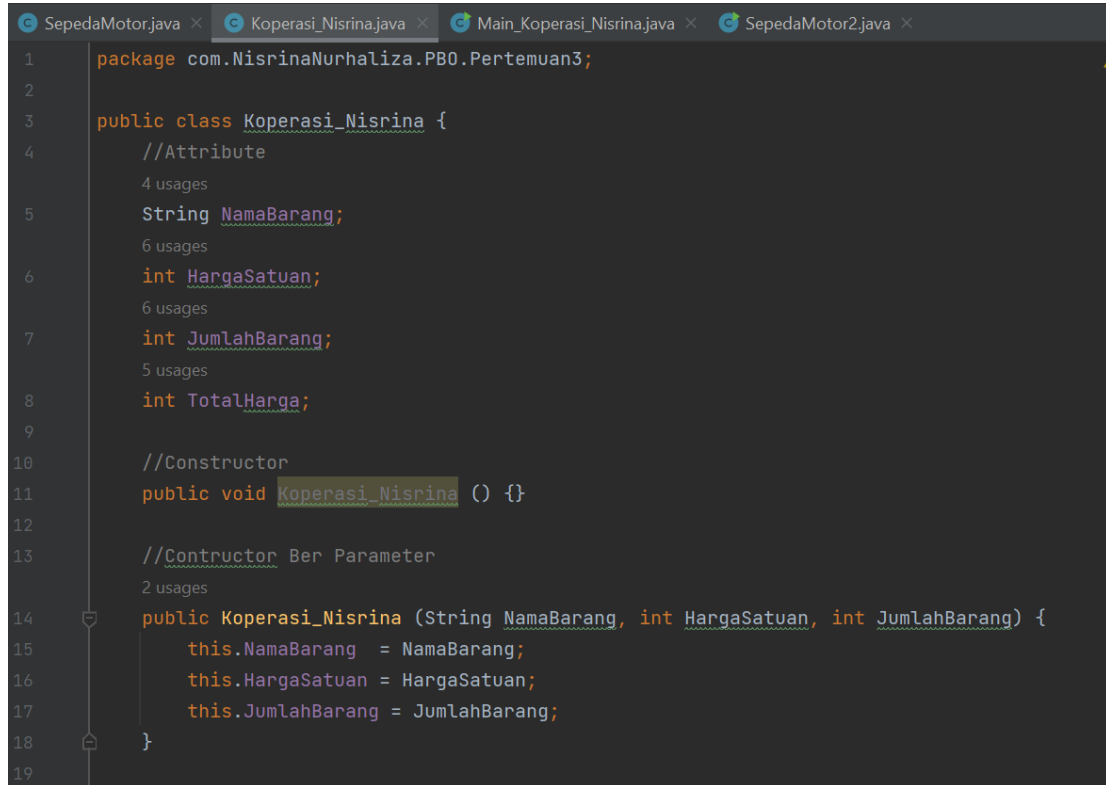


Kemudian buatlah **Class** dengan nama **main.java** dengan ketentuan sebagai berikut :

- Buatlah objek dari **Class Koperasi** minimal 2 seperti contoh data diatas
- Isi semua **method setter** sesuai dengan nilai attributnya
- Tampilkan detail barang
- Tampilkan total semua harga barang yang telah dibuat

Source Code :

- **Koperasi_Nisrina.java**



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 public class Koperasi_Nisrina {
4     //Attribute
5     String NamaBarang;
6     int HargaSatuan;
7     int JumlahBarang;
8     int TotalHarga;
9
10    //Constructor
11    public void Koperasi_Nisrina () {}
12
13    //Constructor Ber Parameter
14    public Koperasi_Nisrina (String NamaBarang, int HargaSatuan, int JumlahBarang) {
15        this.NamaBarang = NamaBarang;
16        this.HargaSatuan = HargaSatuan;
17        this.JumlahBarang = JumlahBarang;
18    }
19 }
```

```
Koperasi_Nisrina.java x Buku_Nisrina.java x Main_Buku_Nisrina.java x Main_Koperasi_Nisrina.java x
20 //Method
21 public String getNamaBarang() {
22     return NamaBarang;
23 }
24
25 public void setNamaBarang(String namaBarang) {
26     NamaBarang = namaBarang;
27 }
28
29 4 usages
30 public int getHargaSatuan() {
31     return HargaSatuan;
32 }
33
34 public void setHargaSatuan(int hargaSatuan) {
35     HargaSatuan = hargaSatuan;
36 }
37
38 4 usages
39 public int getJumlahBarang() {
40     return JumlahBarang;
41 }
```



```
Koperasi_Nisrina.java × Buku_Nisrina.java × Main_Buku_Nisrina.java × Main_Koperasi_Nisrin

39     }
40
41     public void setJumlahBarang(int jumlahBarang) {
42         JumlahBarang = jumlahBarang;
43     }
44
45     public int getTotalHarga() {
46         return TotalHarga;
47     }
48
49     public void setTotalHarga(int totalHarga) {
50         TotalHarga = totalHarga;
51     }
52
53     2 usages
54     public void showInfo() {
55         System.out.println("Nama Barang : " + NamaBarang);
56         System.out.println("Harga Satuan : " + HargaSatuan);
57         System.out.println("Jumlah Barang : " + JumlahBarang);
58         System.out.println("Total Harga : " + TotalHarga);
59         System.out.println();
60     }
```

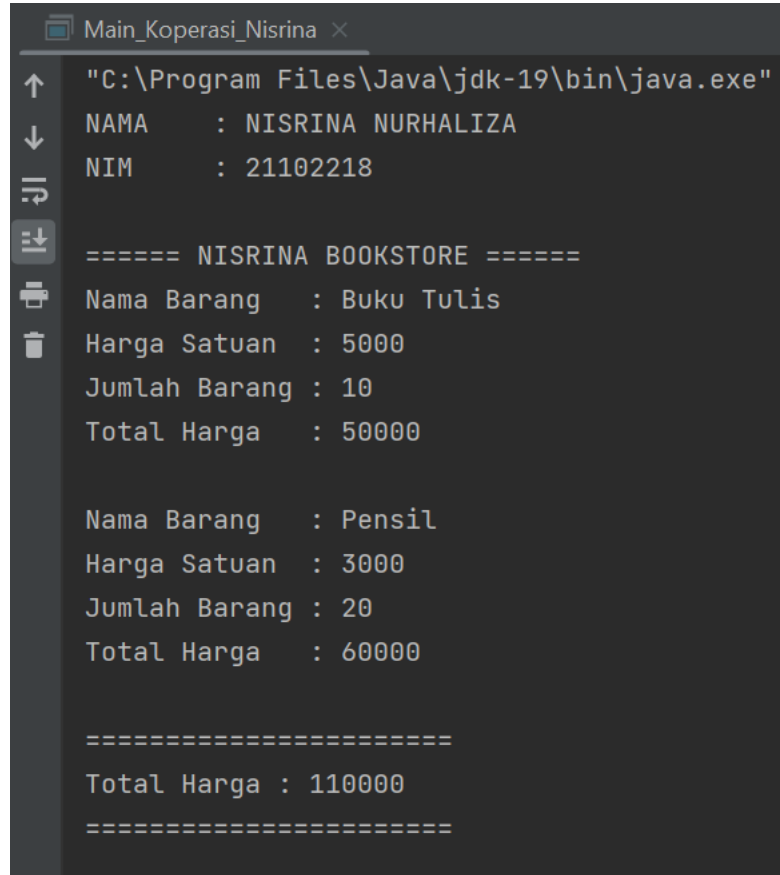
- **Main_Koperasi_Nisrina.java**

```

Koperasi_Nisrina.java x Buku_Nisrina.java x Main_Buku_Nisrina.java x Main_Koperasi_Nisrina.java x SepedaMot
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 import java.util.Arrays;
4
5 public class Main_Koperasi_Nisrina {
6
7     public static void main(String[] args) {
8         System.out.println("NAMA : NISRINA NURHALIZA");
9         System.out.println("NIM : 21102218");
10        System.out.println();
11
12        System.out.println("===== NISRINA BOOKSTORE =====");
13        Koperasi_Nisrina BukuTulis = new Koperasi_Nisrina( NamaBarang: "Buku Tulis",
14            HargaSatuan: 2000, JumlahBarang: 20);
15        Koperasi_Nisrina Pensil = new Koperasi_Nisrina( NamaBarang: "Pensil",
16            HargaSatuan: 3000, JumlahBarang: 10);
17
18        BukuTulis.HargaSatuan = 5000;
19        BukuTulis.JumlahBarang = 10;
20        BukuTulis.TotalHarga = (BukuTulis.getHargaSatuan() *
21            BukuTulis.getJumlahBarang());
22
23        Pensil.HargaSatuan = 3000;
24        Pensil.JumlahBarang = 20;
25        Pensil.TotalHarga = (Pensil.getHargaSatuan() *
26            Pensil.getJumlahBarang());
27
28        BukuTulis.showInfo();
29        Pensil.showInfo();
30        System.out.println("=====");
31        System.out.println(("Total Harga : "
32            + (Pensil.getHargaSatuan() * Pensil.getJumlahBarang() +
33            (BukuTulis.getHargaSatuan() * BukuTulis.getJumlahBarang())));
34
35        System.out.println("===== ");
36    }
37 }

```

Output :



```
"C:\Program Files\Java\jdk-19\bin\java.exe"
NAMA      : NISRINA NURHALIZA
NIM       : 21102218

==== NISRINA BOOKSTORE =====
Nama Barang : Buku Tulis
Harga Satuan : 5000
Jumlah Barang : 10
Total Harga : 50000

Nama Barang : Pensil
Harga Satuan : 3000
Jumlah Barang : 20
Total Harga : 60000

=====
Total Harga : 110000
=====
```

Deskripsi Program

Program di atas merupakan program untuk menampilkan Class koperasi yang menjual beberapa item. Class **Koperasi_Nisrina.java** memiliki tiga atribut yaitu NamaBarang (String), HargaSatuan(int), dan JumlahBarang(int). Menggunakan method Getter dan Setter pada masing masing atribut sehingga nilai atribut dapat diakses dan dimanipulasi dari luar Class.

Class **Main_Koperasi_Nisrina** memiliki fungsi untuk membuat objek dari Class **Koperasi_Nisrina**. Menggunakan method setter pada objek tersebut

yang digunakan untuk mengisi nilai atribut. Menggunakan method getter untuk menampilkan detail barang dan terdapat juga perhitungan total harga barang.

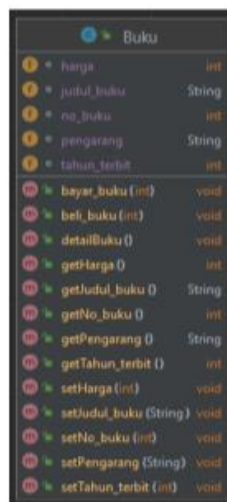
Terdapat dua objek Koperasi yang telah diinisialisasi dengan nilai atribut awal yaitu BukuTulis dan Pensil. Adanya method setter berfungsi untuk mengisi nilai atribut pada masing masing objek Koperasi. Method getter menampilkan detail barang seperti Nama barang, Harga Satuan dan Jumlah Barang. Total Harga Barang didapat dengan cara mengalikan Harga Satuan dengan Jumlah Barang pada masing masing objek Koperasi dan kedua total harga dari barang tersebut dijumlahkan lalu ditampilkan.

2. Disebuah took buku menjual beberapa buku dengan kriteria sebagai berikut :

No_Buku	1
Judul_Buku	Pemrograman Berbasis Objek dengan Java
Pengarang	Indrajani
Tahun_Terbit	2007
Harga	70000

No_Buku	2
Judul_Buku	Dasar Pemrograman Java
Pengarang	Abdul Kadir
Tahun_Terbit	2004
Harga	30000

Buatlah **Class** dengan nama **Buku.java** untuk menampung beberapa method yang diperlukan seperti **Getter and Setter** sebagai berikut :



Kemudian buatlah **Class** dengan nama **main.java** dengan ketentuan sebagai berikut :

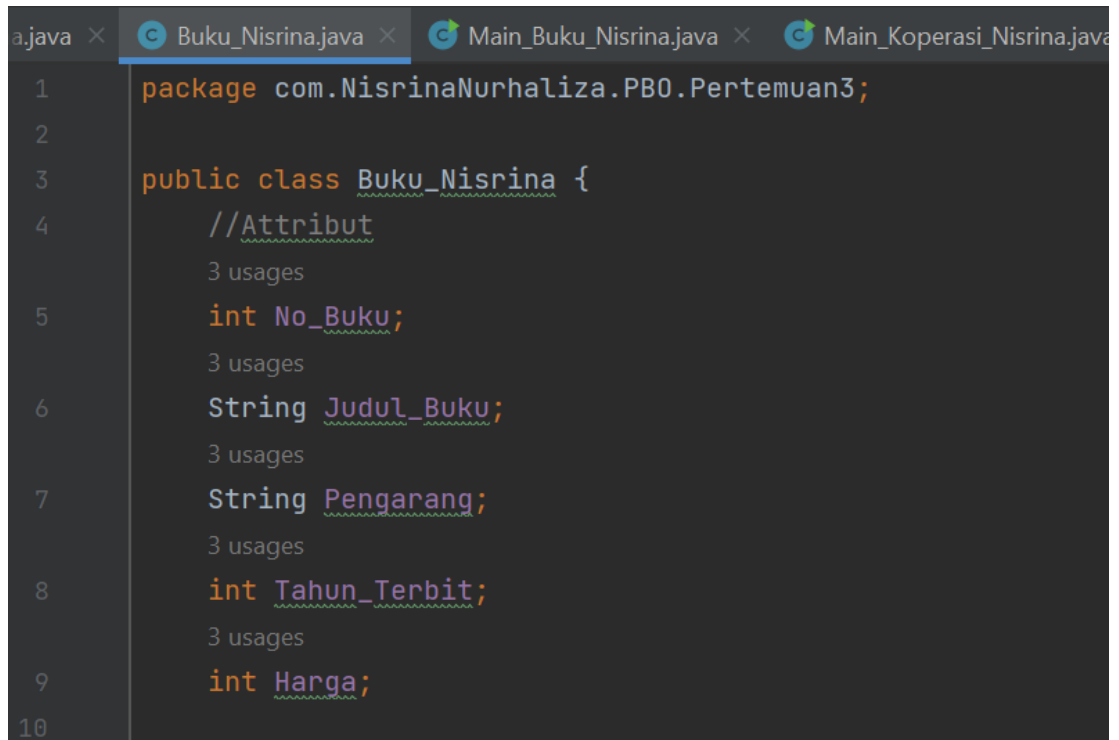
- Buatlah objek dari **Class Buku** minimal 2 seperti contoh data diatas
- Isi semua **method setter** sesuai dengan nilai atributnya
- Tampilkan detail buku
- Terdapat inputan pembelian buku dengan parameter nomor buku
- Terdapat inputan jumlah buku yang ingin dibeli
- Terdapat inputan jumlah bayar

Pada bagian method void beli_buku (int) menampilkan total harga dari jumlah buku yang ingin dibeli.

Pada bagian method void bayar_buku (int) dapat menghitung kembalian dengan rumus jumlah_bayar – harga_buku menampilkan kembali detail buku, jumlah bayar.

Source Code :

- **Buku_Nisrina.java**



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 public class Buku_Nisrina {
4     //Attribut
5     int No_Buku;
6     String Judul_Buku;
7     String Pengarang;
8     int Tahun_Terbit;
9     int Harga;
```

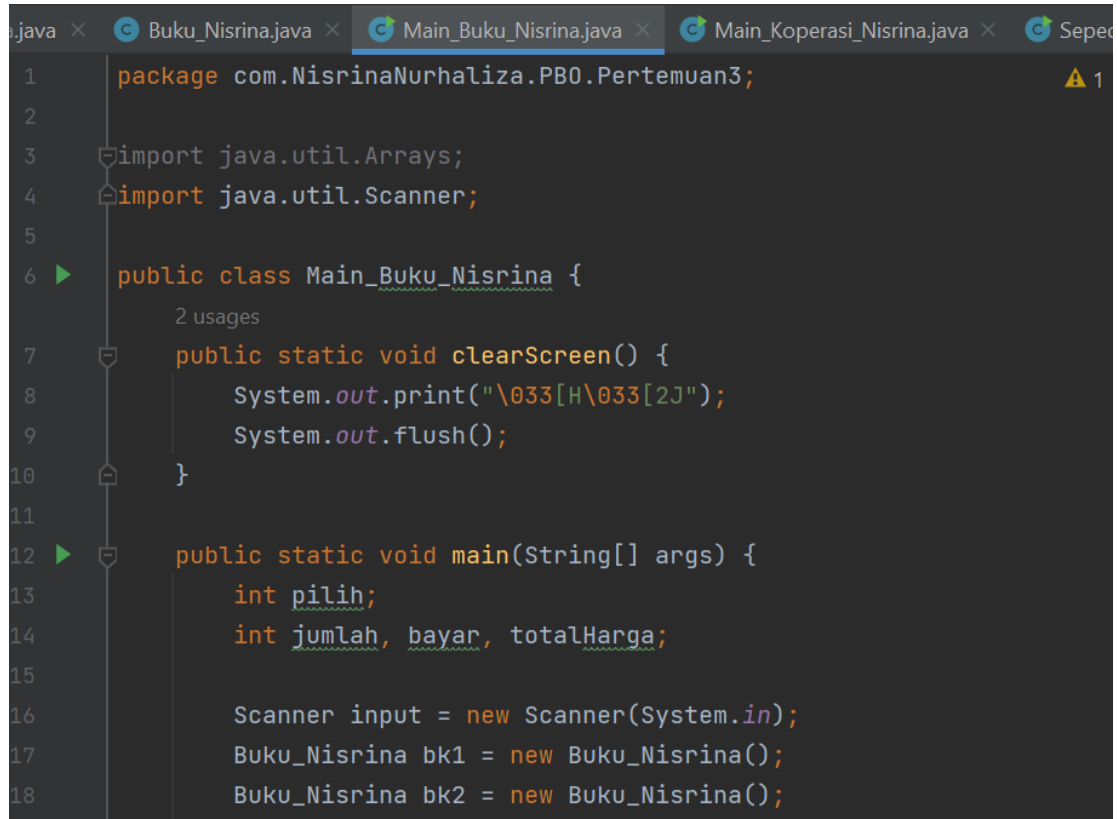
```
a.java x Buku_Nisrina.java x Main_Buku_Nisrina.java x Main_Koperasi_Nisrina.java x
11 public int getNo_Buku() {
12     return No_Buku;
13 }
14
15 2 usages
16 public void setNo_Buku(int no_Buku) {
17     No_Buku = no_Buku;
18 }
19
20 public String getJudul_Buku() {
21     return Judul_Buku;
22 }
23
24 2 usages
25 public void setJudul_Buku(String judul_Buku) {
26     Judul_Buku = judul_Buku;
27 }
28
29 public String getPengarang() {
30     return Pengarang;
31 }
```



```
a.java × Buku_Nisrina.java × Main_Buku_Nisrina.java × Main_Koperasi_Nisrina.java × Se
30      2 usages
31      public void setPengarang(String pengarang) {
32          Pengarang = pengarang;
33      }
34
35      public int getTahun_Terbit() {
36          return Tahun_Terbit;
37      }
38
39      2 usages
40      public void setTahun_Terbit(int tahun_Terbit) {
41          Tahun_Terbit = tahun_Terbit;
42      }
43
44      2 usages
45      public int getHarga() {
46          return Harga;
47      }
48
49      2 usages
50      public void setHarga(int harga) {
51          Harga = harga;
52      }
```

```
50
4 usages
51 public void showInfo() {
52     System.out.println("No Buku           : " + No_Buku);
53     System.out.println("Judul Buku         : " + Judul_Buku);
54     System.out.println("Pengarang         : " + Pengarang);
55     System.out.println("Tahun Terbit       : " + Tahun_Terbit);
56     System.out.println("Harga           : " + Harga);
57     System.out.println();
58 }
59 public void beli (int beli) {
60
61 }
62 public void bayar (int bayar) {
63
64 }
65 }
```

- **Main_Buku_Nisrina.java**



```
1 package com.NisrinaNurhaliza.PB0.Pertemuan3;
2
3 import java.util.Arrays;
4 import java.util.Scanner;
5
6 public class Main_Buku_Nisrina {
7     public static void clearScreen() {
8         System.out.print("\033[H\033[2J");
9         System.out.flush();
10    }
11
12    public static void main(String[] args) {
13        int pilih;
14        int jumlah, bayar, totalHarga;
15
16        Scanner input = new Scanner(System.in);
17        Buku_Nisrina bk1 = new Buku_Nisrina();
18        Buku_Nisrina bk2 = new Buku_Nisrina();
```

```

19
20     System.out.println("NAMA      : NISRINA NURHALIZA");
21     System.out.println("NIM       : 21102218");
22     System.out.println();
23     System.out.println("===== LIST OF BOOKS =====");
24     bk1.setNo_Buku(1);
25     bk1.setJudul_Buku("Pemrograman Berbasis Objek dengan Java");
26     bk1.setPengarang("Indrajani");
27     bk1.setTahun_Terbit(2007);
28     bk1.setHarga(70000);
29
30     bk2.setNo_Buku(2);
31     bk2.setJudul_Buku("Dasar Pemrograman Java");
32     bk2.setPengarang("Abdul Kadir");
33     bk2.setTahun_Terbit(2004);
34     bk2.setHarga(30000);
35
36     bk1.showInfo();
37     bk2.showInfo();
38

```

```

39
40     System.out.print("Masukkan No Buku Yang Ingin Dibeli      : ");
41     pilih = input.nextInt();
42     System.out.print("Masukkan Jumlah Buku Yang Ingin Dibeli : ");
43     jumlah = input.nextInt();
44
45     if (pilih == 1) {
46         totalHarga = bk1.getHarga() * jumlah;
47         System.out.println("Total Harga          : " + totalHarga);
48         System.out.print("Masukkan Jumlah Uang : ");
49         bayar = input.nextInt();
50         if (bayar <= totalHarga) {
51             clearScreen();
52             System.out.println("Mohon Maaf Uang Anda Tidak Cukup.");
53         } else {
54             clearScreen();
55             System.out.println();
56             System.out.println("===== Rincian Pembelian Buku =====");
57             bk1.showInfo();
58             System.out.println("===== Rincian Pembayaran =====");
59             System.out.println("Jumlah Bayar      : " + bayar);
60             System.out.println("Kembalian         : " + (bayar - totalHarga));
61         }
62     } else if (pilih == 2) {

```

```

61     }
62 }else if (pilih == 2) {
63     totalHarga = bk2.getHarga() * jumlah;
64     System.out.println("Total Harga      : " + totalHarga);
65     System.out.print("Masukkan Jumlah Uang : ");
66     bayar = input.nextInt();
67     if (bayar <= totalHarga) {
68         System.out.println("Mohon Maaf Uang Anda Tidak Cukup.");
69
70     } else {
71         System.out.println("==== Rincian Pembelian Buku =====");
72         bk2.showInfo();
73         System.out.println("==== Rincian Pembayaran =====");
74         System.out.println("Jumlah Bayar      : " + bayar);
75         System.out.println("Kembalian         : " + (bayar - totalHarga));
76     }
77 }else {
78     System.out.println("Pilihan Tidak Tersedia");
79 }
80 input.close();
81 }
82 }

```

Output :

- Jika pembayaran dilakukan dengan benar

[illegible]

===== Rincian Pembelian Buku =====

No Buku : 1

Judul Buku : Pemrograman Berbasis Objek dengan Java

Pengarang : Indrajani

Tahun Terbit : 2007

Harga : 70000

===== Rincian Pembayaran =====

Jumlah Bayar :150000

Kembalian :10000

- Jika pembayaran kurang

```
Run: Main_Buku_Nisrina x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\
NAMA      : NISRINA NURHALIZA
NIM       : 21102218

===== LIST OF BOOKS =====
No Buku      : 1
Judul Buku   : Pemrograman Berbasis Objek dengan Java
Pengarang    : Indrajani
Tahun Terbit : 2007
Harga        : 70000

No Buku      : 2
Judul Buku   : Dasar Pemrograman Java
Pengarang    : Abdul Kadir
Tahun Terbit : 2004
Harga        : 30000

=====
Masukkan No Buku Yang Ingin Dibeli      : 2
Masukkan Jumlah Buku Yang Ingin Dibeli : 2
Total Harga                             : 60000
Masukkan Jumlah Uang : 40000
Mohon Maaf Uang Anda Tidak Cukup.
```


- Jika menginputkan data yang tidak tersedia

```
Run: Main_Buku_Nisrina x
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Pro
NAMA      : NISRINA NURHALIZA
NIM       : 21102218

===== LIST OF BOOKS =====
No Buku      : 1
Judul Buku   : Pemrograman Berbasis Objek dengan Java
Pengarang    : Indrajani
Tahun Terbit : 2007
Harga        : 70000

No Buku      : 2
Judul Buku   : Dasar Pemrograman Java
Pengarang    : Abdul Kadir
Tahun Terbit : 2004
Harga        : 30000

=====
Masukkan No Buku Yang Ingin Dibeli      : 4
Masukkan Jumlah Buku Yang Ingin Dibeli : 10
Pilihan Tidak Tersedia
```

Deskripsi Program

Program di atas merupakan program untuk menampilkan beberapa barang di toko buku. Pada program diatas menggunakan *System.out.flush* dimana berfungsi untuk meminta server mengirim output yang saat ini di-buffer ke browser. Konfigurasi server mungkin tidak selalu memungkinkan hal ini terjadi. Terdapat *clearScreen* yang berfungsi untuk membersihkan layar. Program diatas juga terdapat *System.out.println("\033[H\033[2J");* yang memiliki fungsi sama dengan *System.out.println();* yaitu untuk inputan kosong atau biasa digunakan sebagai spasi baris. Program di atas menggunakan objek *Scanner* untuk mengambil input dari pengguna dan menampilkan output di layar.

Program diatas terdiri dari dua buah Class yaitu **Buku_Nisrina** yang berfungsi untuk menampung data mengenai buku yang tersedia dan **Main_Buku_Nisrina** yang berfungsi untuk sebagai program utama untuk melakukan transaksi pembelian buku. Dalam Class **Buku_Nisrina** terdapat beberapa attribute seperti No Buku, Judul Buku, Pengarang, Tahun Terbit dan Harga dimana setiap attribute memiliki getter dan setter untuk mengambil serta mengisi nilai attribute. Dalam Class **Main_Buku_Nisrina** terdapat method *beli_buku()* dan *bayar_buku()* yang digunakan untuk melakukan proses pembelian buku. Pada method *beli_buku()*, user diminta untuk memasukkan nomor buku dan jumlah buku yang ingin dibeli kemudian program akan menampilkan total harga pembelian. Pada method *bayar_buku()*, user diminta untuk memasukkan jumlah uang yang harus dibayarkan kemudian program akan menghitung kembalian dan menampilkan detail buku serta jumlah bayar. Setelah proses pembelian buku selesai maka program akan menampilkan rincian pembelian buku serta rincian pembayaran.

BAB V

KESIMPULAN

Pada praktikum pertemuan kedua, diharapkan mahasiswa mampu memahami mengenai Class dan Objek yang memiliki fungsi dan kegunaan masing masing dalam penyelesaian sebuah masalah dalam suatu pemrograman. Class dan Objek merupakan dasar dalam pemrograman berorientasi objek.

Class digunakan sebagai sebuah blueprint atau cetak biru untuk membuat objek. Objek sendiri dapat dianggap sebagai sebuah entitas yang memiliki keadaan (state) dan perilaku (behavior) tertentu. Dengan menggunakan konsep Class dan Objek, pemrograman dapat memecah program menjadi bagian-bagian yang lebih kecil dan terorganisir serta mempermudah pengembangan dan perawatan kode.

Objek adalah suatu abstraksi dari suatu problem. Sedangkan kelas adalah kumpulan objek yang memiliki atribut dan perilaku yang sama (karakteristik sama). Setiap objek memiliki nilai atribut/state yang unik yang membedakannya dengan objek lain dari kelas yang sama, dan objek memiliki perilaku/behaviour untuk mengakses atribut/state yang dimilikinya.

Dalam Java, sebuah Class dapat memiliki variabel, method, dan konstruktor. Variabel digunakan untuk menyimpan data atau keadaan objek, method digunakan untuk memberikan perilaku atau aksi pada objek, dan konstruktor digunakan untuk menginisialisasi objek ketika objek tersebut dibuat. Selain itu, Java juga mendukung konsep enkapsulasi, di mana variabel pada sebuah Class dapat diakses dan dimanipulasi hanya melalui method yang tersedia pada Class tersebut.