

**LAPORAN PRAKTIKUM
PEMOGRAMAN BERORIENTASI OBJEK**

**MODUL III
“KELAS DAN OBJEK”**



**Disusun oleh:
Laras Wahyu Adiningsih
21102214**

**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

- a. Mahasiswa diharapkan mampu memahami mengenai objek dan kelas
- b. Mahasiswa diharapkan mampu menerjemahkan objek dan kelas dalam bahasa pemrograman

BAB II

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 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 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 Field atau variable :

- Instance variable
Setiap objek memiliki salinan sendiri dan salinan tersebut memiliki nilai masing-masing.
- Class variable
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 konstruktor :

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

Perlu diketahui :

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

Karakteristik constructor :

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

3. Methods

Method merupakan fungsi-fungsi implementasi perilaku objek untuk mengakses atribut-atributnya.

Method adalah kumpulan program yang mempunyai nama. Program harus dibungkus dalam method. Dengan method kita bisa 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.

BAB III

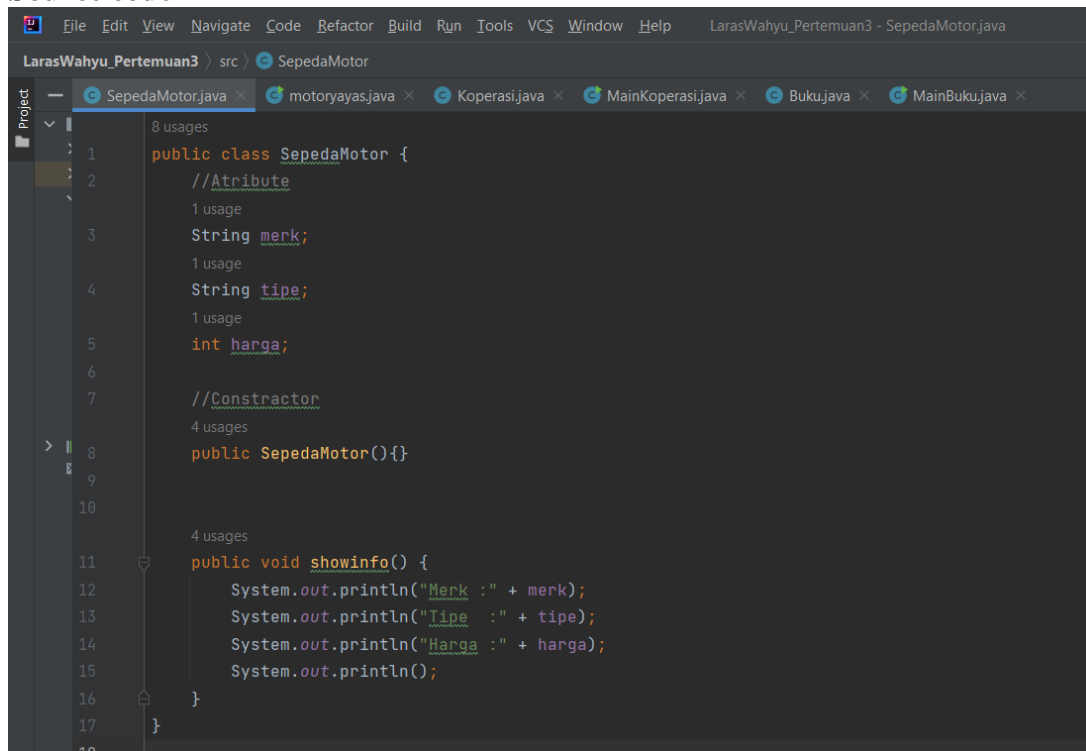
GUIDED

1. Guided 1

Membuat Class Baru

Buatlah sebuah class baru dengan nama “Sepeda Motor”, lalu ketikkan kode berikut :

Source code

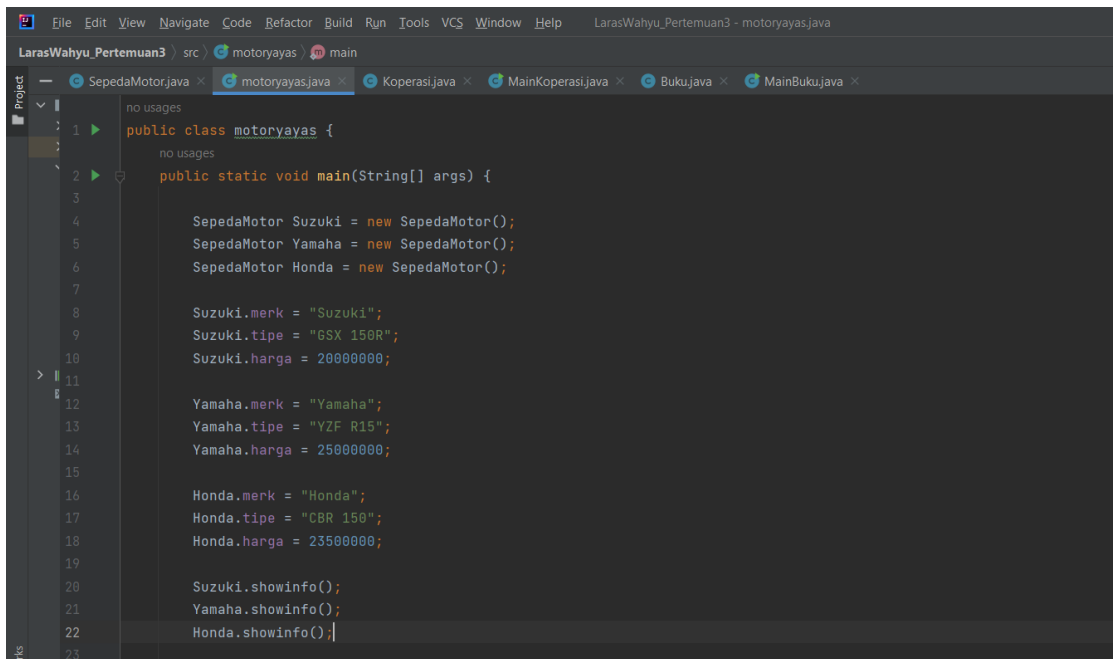


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - SepedaMotor.java
LarasWahyu_Pertemuan3 > src > SepedaMotor
Project
  SepedaMotor.java x motoryayas.java x Koperasi.java x MainKoperasi.java x Buku.java x MainBuku.java x
  8 usages
  1 public class SepedaMotor {
  2     //Attribute
  3     String merk;
  4     String tipe;
  5     int harga;
  6
  7     //Constructor
  8     public SepedaMotor(){}
  9
 10
 11     4 usages
 12     public void showinfo() {
 13         System.out.println("Merk : " + merk);
 14         System.out.println("Tipe : " + tipe);
 15         System.out.println("Harga : " + harga);
 16         System.out.println();
 17     }
 18 }
```

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 bisa dijalankan, kita memerlukan sebuah program utama. Untuk itu, buatlah sebuah Java class baru dan beri nama Main.java, lalu salin dan jalankan kode berikut :



```
1 public class motoryayas {
2     public static void main(String[] args) {
3
4         SepedaMotor Suzuki = new SepedaMotor();
5         SepedaMotor Yamaha = new SepedaMotor();
6         SepedaMotor Honda = new SepedaMotor();
7
8         Suzuki.merk = "Suzuki";
9         Suzuki.tipe = "GSX 150R";
10        Suzuki.harga = 20000000;
11
12        Yamaha.merk = "Yamaha";
13        Yamaha.tipe = "YZF R15";
14        Yamaha.harga = 25000000;
15
16        Honda.merk = "Honda";
17        Honda.tipe = "CBR 150";
18        Honda.harga = 23500000;
19
20        Suzuki.showinfo();
21        Yamaha.showinfo();
22        Honda.showinfo();
23    }
```

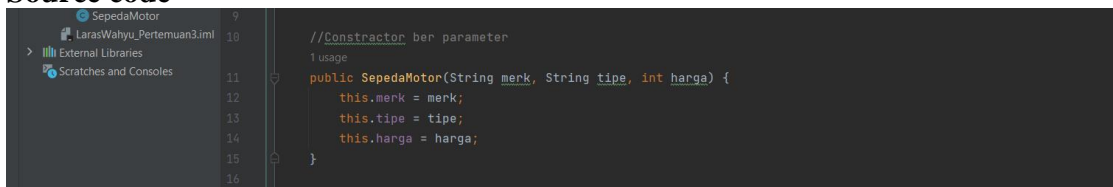
2. Guided 2

Constructor Ber-Parameter

Daripada melakukan pengisian attribute satu per satu seperti di atas, kita bisa melakukan sedikit modifikasi untuk mempermudah kerja kita!

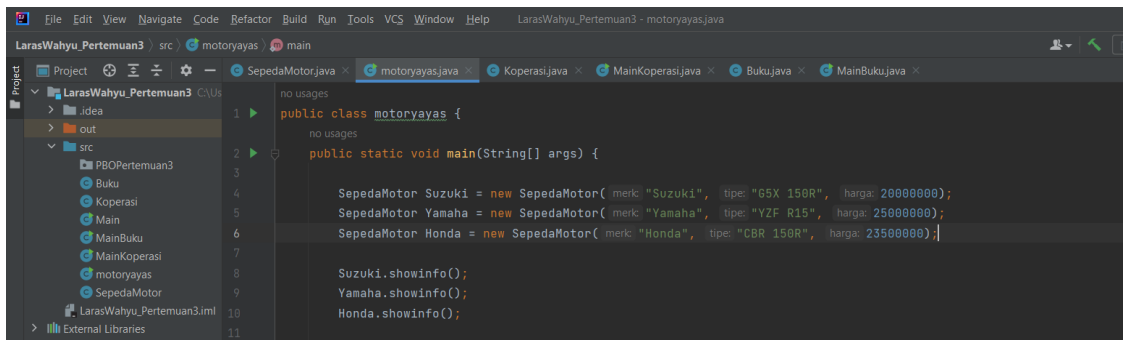
Kembali ke SepedaMotor.java, lalu tambahkan constructor method baru di kodenya. (Keterangan: Anda dapat membuat method dengan nama sama tanpa menyebabkan error asalkan method-method tersebut memiliki parameter berbeda!)

Source code



```
10 //Constructor ber parameter
11 1 usage
12 public SepedaMotor(String merk, String tipe, int harga) {
13     this.merk = merk;
14     this.tipe = tipe;
15     this.harga = harga;
16 }
```

Lalu Kembali ke **Main.java** dan lakukan modifikasi terhadap source codenya hingga akhirnya terlihat seperti ini :



```
1 public class motoryayas {
2     public static void main(String[] args) {
3
4         SepedaMotor Suzuki = new SepedaMotor( merk: "Suzuki", tipe: "G5X 150R", harga: 20000000);
5         SepedaMotor Yamaha = new SepedaMotor( merk: "Yamaha", tipe: "YZF R15", harga: 25000000);
6         SepedaMotor Honda = new SepedaMotor( merk: "Honda", tipe: "CBR 150R", harga: 23500000);
7
8         Suzuki.showinfo();
9         Yamaha.showinfo();
10        Honda.showinfo();
11    }
```

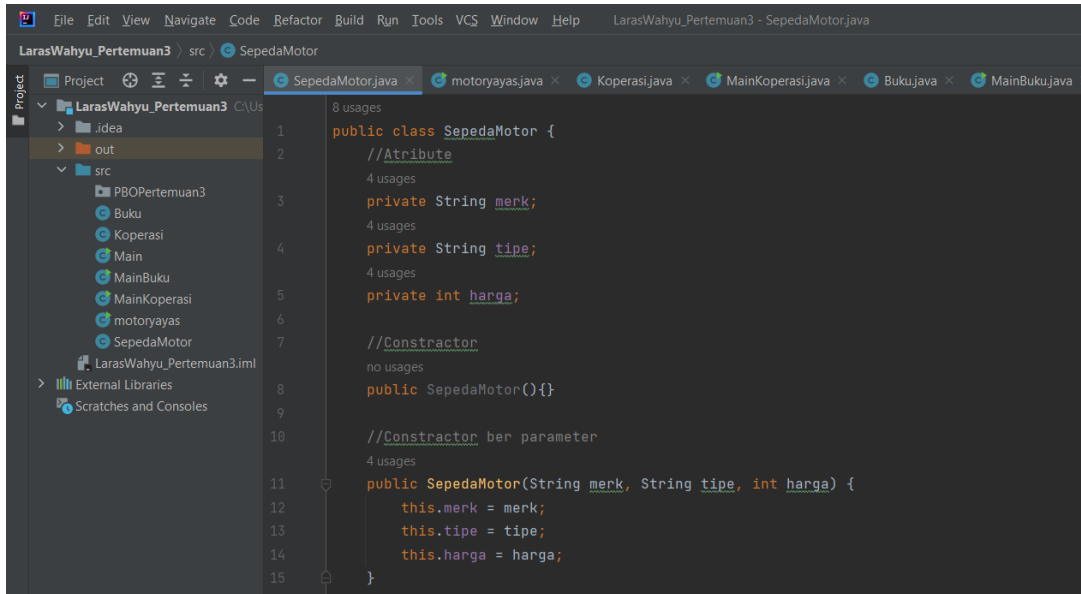
3. Guided 3 Modifier

Di Java, ada yang namanya modifier yang 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 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** hingga hasil akhirnya terlihat seperti berikut :

Source code



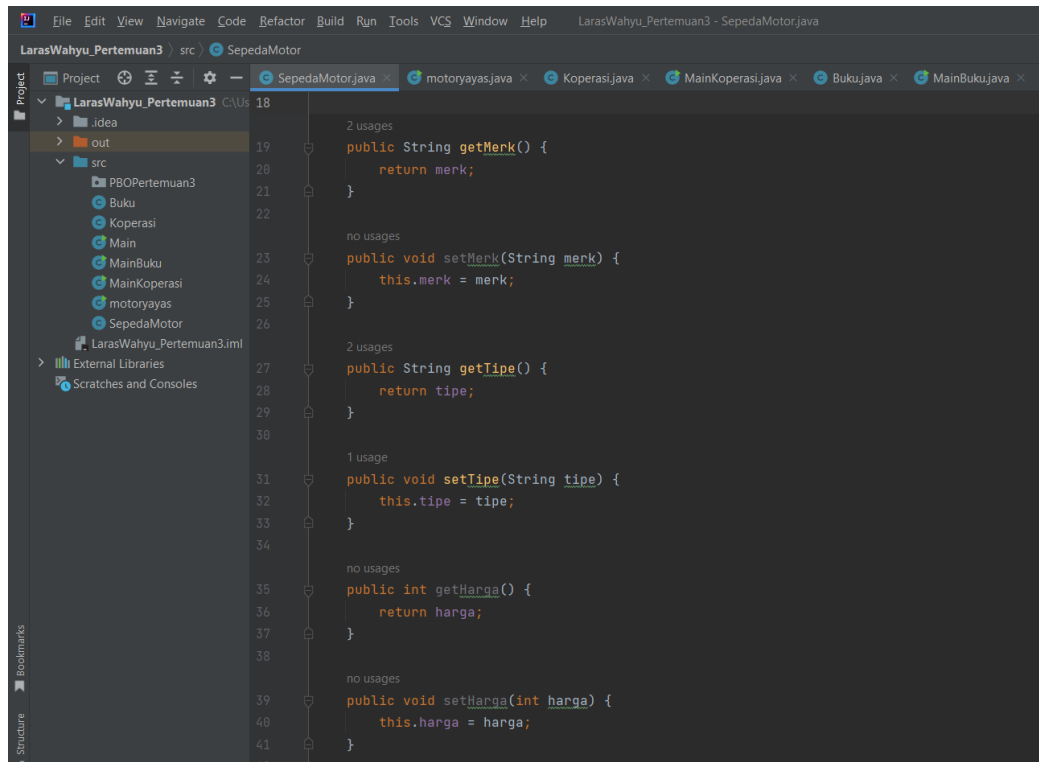
```
1 public class SepedaMotor {
2     //Attribute
3     private String merk;
4     private String tipe;
5     private int harga;
6
7     //Constructor
8     public SepedaMotor() {}
9
10    //Constructor ber parameter
11    public SepedaMotor(String merk, String tipe, int harga) {
12        this.merk = merk;
13        this.tipe = tipe;
14        this.harga = harga;
15    }
16 }
```

4. Guided 4 Getter & 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.

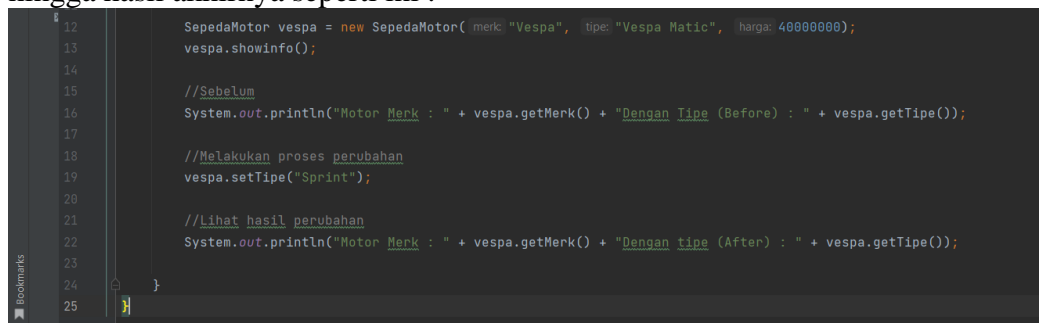
Pada **SepedaMotor.java** tambahkan method-method berikut :

Source code



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - SepedaMotor.java
LarasWahyu_Pertemuan3 / src / SepedaMotor
Project
  LarasWahyu_Pertemuan3
    out
    src
      PBOPertemuan3
      Buku
      Koperasi
      Main
      MainBuku
      MainKoperasi
      motoryayas
      SepedaMotor
    LarasWahyu_Pertemuan3.iml
  External Libraries
  Scratches and Consoles
Structure
  Bookmarks
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
2 usages
public String getMerk() {
    return merk;
}
no usages
public void setMerk(String merk) {
    this.merk = merk;
}
2 usages
public String getTipe() {
    return tipe;
}
1 usage
public void setTipe(String tipe) {
    this.tipe = tipe;
}
no usages
public int getHarga() {
    return harga;
}
no usages
public void setHarga(int harga) {
    this.harga = harga;
}
```

Untuk uji coba, Kembali ke class Main.java, lakukan modifikasi ke seluruh isi kodenya hingga hasil akhirnya seperti ini :



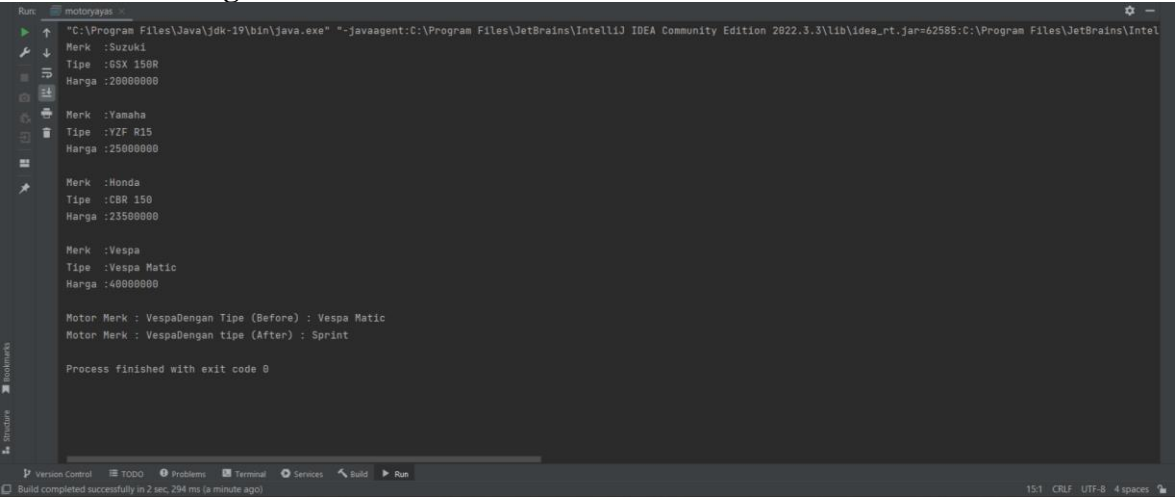
```
12
13
14
15
16
17
18
19
20
21
22
23
24
25
SepedaMotor vespa = new SepedaMotor(merk: "Vespa", tipe: "Vespa Matic", harga: 40000000);
vespa.showinfo();

//Sebelum
System.out.println("Motor Merk : " + vespa.getMerk() + "Dengan Tipe (Before) : " + vespa.getTipe());

//Melakukan proses perubahan
vespa.setTipe("Sprint");

//Lihat hasil perubahan
System.out.println("Motor Merk : " + vespa.getMerk() + "Dengan tipe (After) : " + vespa.getTipe());
}
```


Screenshoot Program



The screenshot shows the 'Run' window of an IDE. The title bar indicates the process is 'motorayaya'. The command line shows the Java runtime environment. The output displays data for four motorcycle models: Suzuki, Yamaha, Honda, and Vespa. It also shows a comparison of the Vespa model before and after a change, and a final status message.

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.3\lib\idea_rt.jar=62585:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.3\bin" -Dfile.encoding=UTF-8
Merk :Suzuki
Tipe :GSX 150R
Harga :20000000

Merk :Yamaha
Tipe :YZF R15
Harga :25000000

Merk :Honda
Tipe :CBR 150
Harga :23500000

Merk :Vespa
Tipe :Vespa Matic
Harga :40000000

Motor Merk : VespaDengan Tipe (Before) : Vespa Matic
Motor Merk : VespaDengan tipe (After) : Sprint

Process finished with exit code 0
```

At the bottom, a status bar indicates 'Build completed successfully in 2 sec, 294 ms (a minute ago)' and the file encoding is 'UTF-8'.

UNGUIDED

1. Unguided 1

Disebuah koperasi sekolah menjual item sebagai berikut :

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

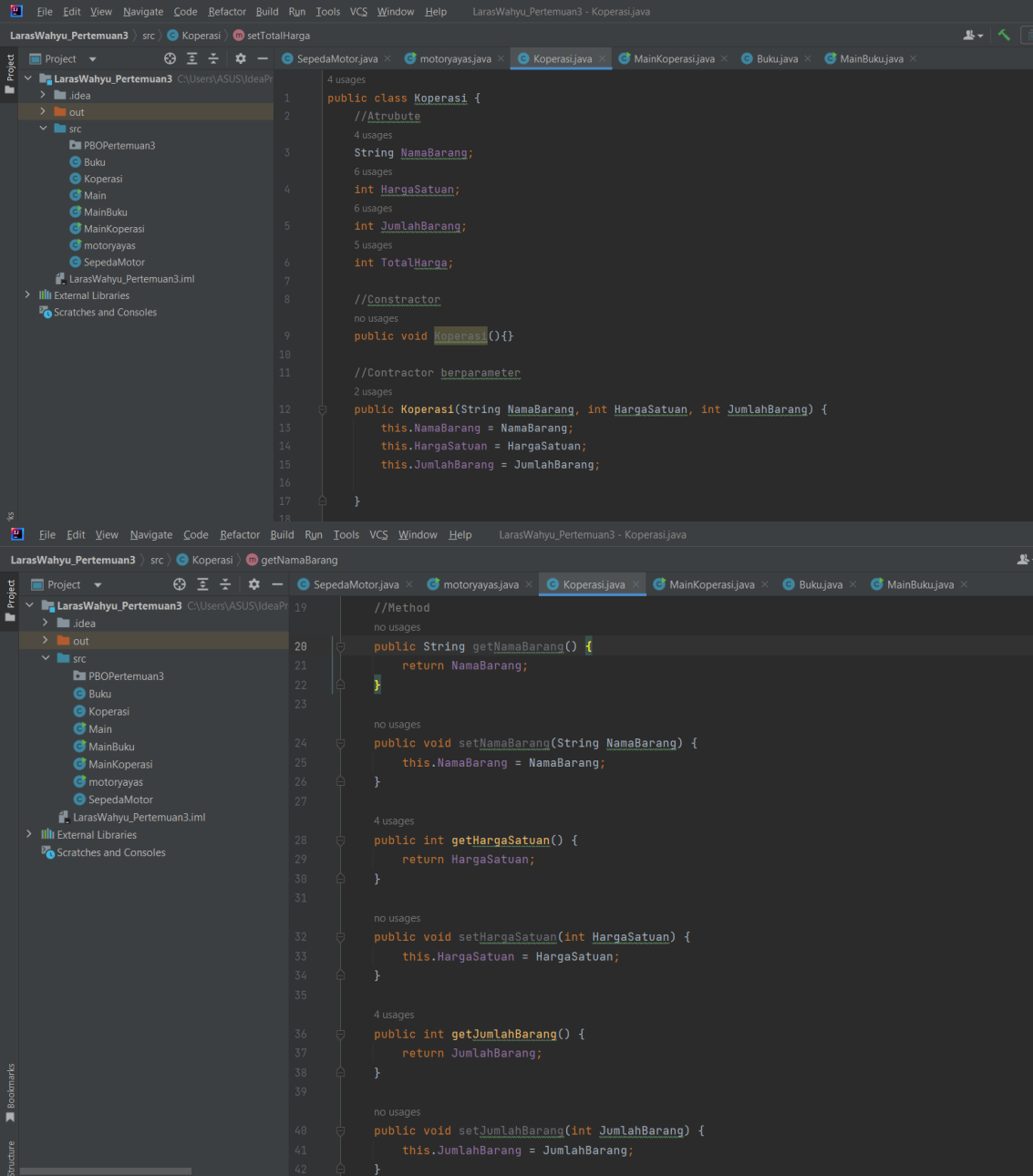
Buatlah **Class** dengan nama **Koperasi.java** untuk menampung beberapa method yang diperlukan seperti **setter** dan **getter** sebagai berikut :



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

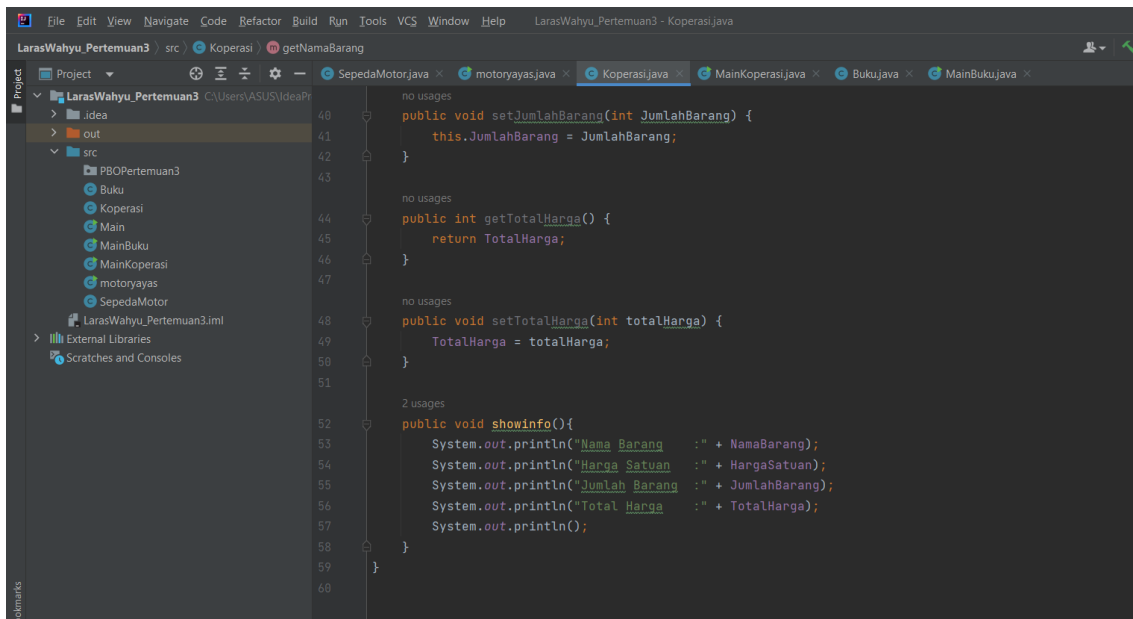
- Buatlah objek dari **Class Koperasi** minimal 2 seperti contoh data diatas
- Isi semua **method setter** sesuai dengan nilai attributenya
- Tampilkan detail barang
- Tampilkan total semua harga barang yang telah dibuat

Source code Koperasi.Java



The screenshot displays an IDE window for a project named 'LarasWahyu_Pertemuan3'. The 'Project' view on the left shows the file structure, including a 'src' directory with files like 'PBOPertemuan3', 'Buku', 'Koperasi', 'Main', 'MainBuku', 'MainKoperasi', 'motoryayas', and 'SepedaMotor'. The 'Koperasi.java' file is selected, and its source code is visible in the main editor. The code defines a 'Koperasi' class with attributes 'NamaBarang', 'HargaSatuan', 'JumlahBarang', and 'TotalHarga'. It includes a constructor and several getter and setter methods. The code is as follows:

```
1 public class Koperasi {
2     //Attribute
3     String NamaBarang;
4     int HargaSatuan;
5     int JumlahBarang;
6     int TotalHarga;
7
8     //Constructor
9     public void Koperasi() {}
10
11    //Constructor berparameter
12    public Koperasi(String NamaBarang, int HargaSatuan, int JumlahBarang) {
13        this.NamaBarang = NamaBarang;
14        this.HargaSatuan = HargaSatuan;
15        this.JumlahBarang = JumlahBarang;
16    }
17
18    //Method
19    public String getNamaBarang() {
20        return NamaBarang;
21    }
22
23    public void setNamaBarang(String NamaBarang) {
24        this.NamaBarang = NamaBarang;
25    }
26
27    public int getHargaSatuan() {
28        return HargaSatuan;
29    }
30
31    public void setHargaSatuan(int HargaSatuan) {
32        this.HargaSatuan = HargaSatuan;
33    }
34
35    public int getJumlahBarang() {
36        return JumlahBarang;
37    }
38
39    public void setJumlahBarang(int JumlahBarang) {
40        this.JumlahBarang = JumlahBarang;
41    }
42 }
```

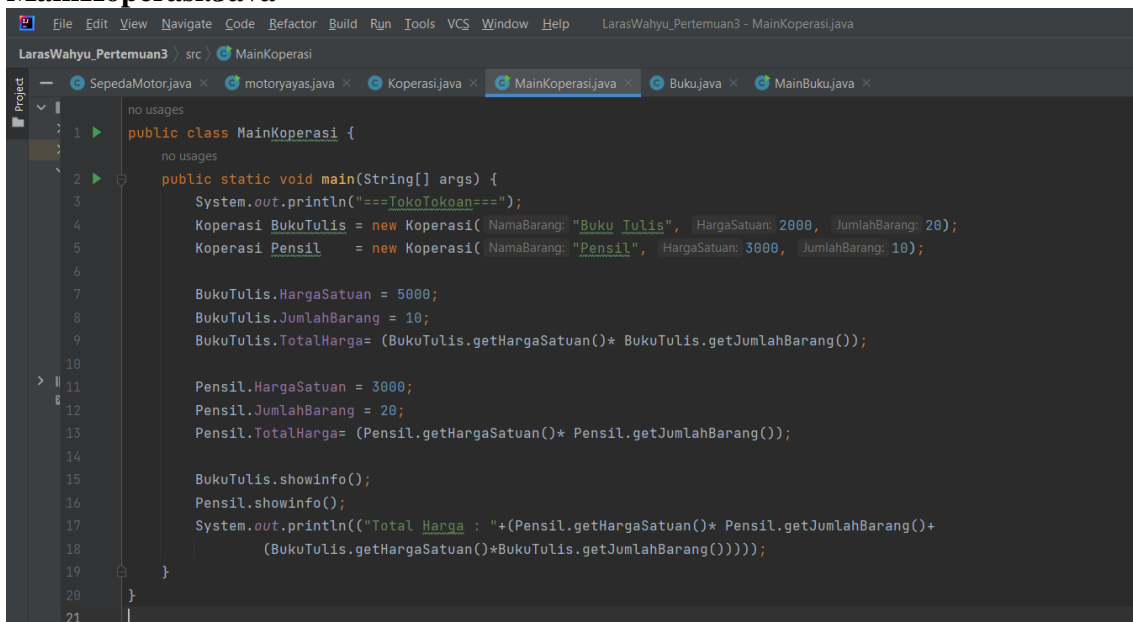


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - Koperasi.java
LarasWahyu_Pertemuan3 src Koperasi Koperasi.java MainKoperasi.java Buku.java MainBuku.java

Project
  LarasWahyu_Pertemuan3
    .idea
    out
    src
      PBOPertemuan3
        Buku
        Koperasi
        Main
        MainBuku
        MainKoperasi
        motoryayas
        SepedaMotor
      LarasWahyu_Pertemuan3.iml
    External Libraries
    Scratches and Consoles

40 public void setJumlahBarang(int JumlahBarang) {
41     this.JumlahBarang = JumlahBarang;
42 }
43
44 no usages
45 public int getTotalHarga() {
46     return TotalHarga;
47 }
48
49 no usages
50 public void setTotalHarga(int totalHarga) {
51     TotalHarga = totalHarga;
52 }
53
54 2 usages
55 public void showinfo(){
56     System.out.println("Nama Barang : " + NamaBarang);
57     System.out.println("Harga Satuan : " + HargaSatuan);
58     System.out.println("Jumlah Barang : " + JumlahBarang);
59     System.out.println("Total Harga : " + TotalHarga);
60     System.out.println();
61 }
```

Source code MainKoperasi.Java

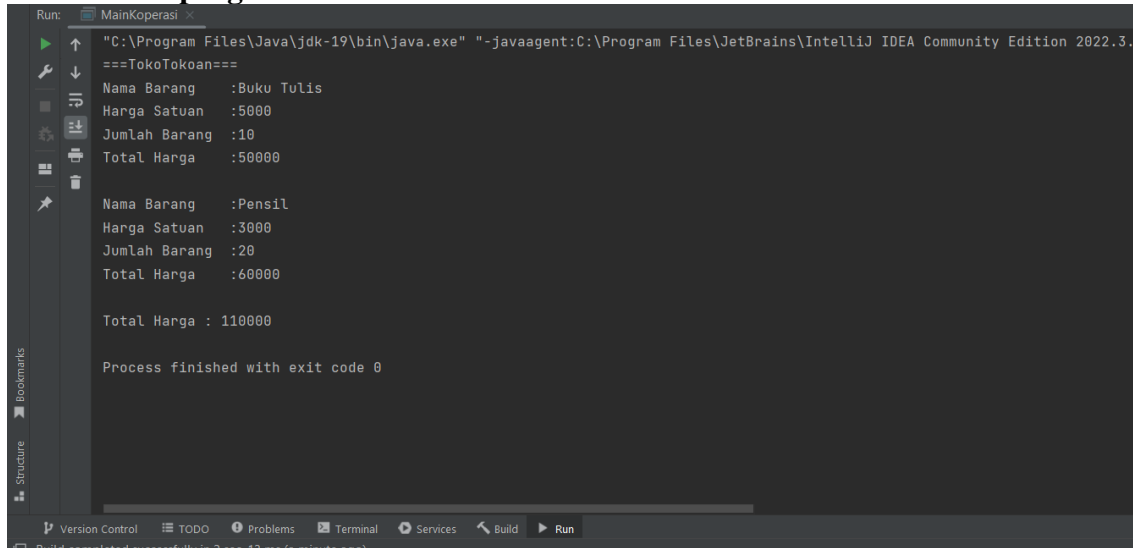


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - MainKoperasi.java
LarasWahyu_Pertemuan3 src MainKoperasi
MainKoperasi.java Buku.java MainBuku.java

Project
  LarasWahyu_Pertemuan3
    .idea
    out
    src
      PBOPertemuan3
        Buku
        Koperasi
        Main
        MainBuku
        MainKoperasi
        motoryayas
        SepedaMotor
      LarasWahyu_Pertemuan3.iml
    External Libraries
    Scratches and Consoles

no usages
1 public class MainKoperasi {
2     no usages
3     public static void main(String[] args) {
4         System.out.println("===TokoTokoan===");
5         Koperasi BukuTulis = new Koperasi( NamaBarang: "Buku Tulis", HargaSatuan: 2000, JumlahBarang: 20);
6         Koperasi Pensil = new Koperasi( NamaBarang: "Pensil", HargaSatuan: 3000, JumlahBarang: 10);
7
8         BukuTulis.HargaSatuan = 5000;
9         BukuTulis.JumlahBarang = 10;
10        BukuTulis.TotalHarga= (BukuTulis.getHargaSatuan()* BukuTulis.getJumlahBarang());
11
12        Pensil.HargaSatuan = 3000;
13        Pensil.JumlahBarang = 20;
14        Pensil.TotalHarga= (Pensil.getHargaSatuan()* Pensil.getJumlahBarang());
15
16        BukuTulis.showinfo();
17        Pensil.showinfo();
18        System.out.println(("Total Harga : "+(Pensil.getHargaSatuan()* Pensil.getJumlahBarang()+
19            (BukuTulis.getHargaSatuan()*BukuTulis.getJumlahBarang()))));
20    }
21 }
```

Screenshoot program



```
Run: MainKoperasi
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2022.3.
===TokoTokoan===
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

Process finished with exit code 0
```

Deskripsi program

Pada Program diatas kelas Main memiliki fungsi membuat objek dari kelas Koperasi, lalu mengisi nilai atribut pada objek yang telah dibuat menggunakan method setter dan menampilkan detail barang menggunakan method getter. Setelah semua dimasukan maka total harga barang akan dihitung.

2. Unguided 2

Disebuah toko 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 **setter** dan **getter** sebagai berikut :

Buku		
f	◦ harga	int
f	◦ judul_buku	String
f	◦ no_buku	int
f	◦ pengarang	String
f	◦ tahun_terbit	int
m	📌 bayar_buku (int)	void
m	📌 beli_buku (int)	void
m	📌 detailBuku ()	void
m	📌 getHarga ()	int
m	📌 getJudul_buku ()	String
m	📌 getNo_buku ()	int
m	📌 getPengarang ()	String
m	📌 getTahun_terbit ()	int
m	📌 setHarga (int)	void
m	📌 setJudul_buku (String)	void
m	📌 setNo_buku (int)	void
m	📌 setPengarang (String)	void
m	📌 setTahun_terbit (int)	void

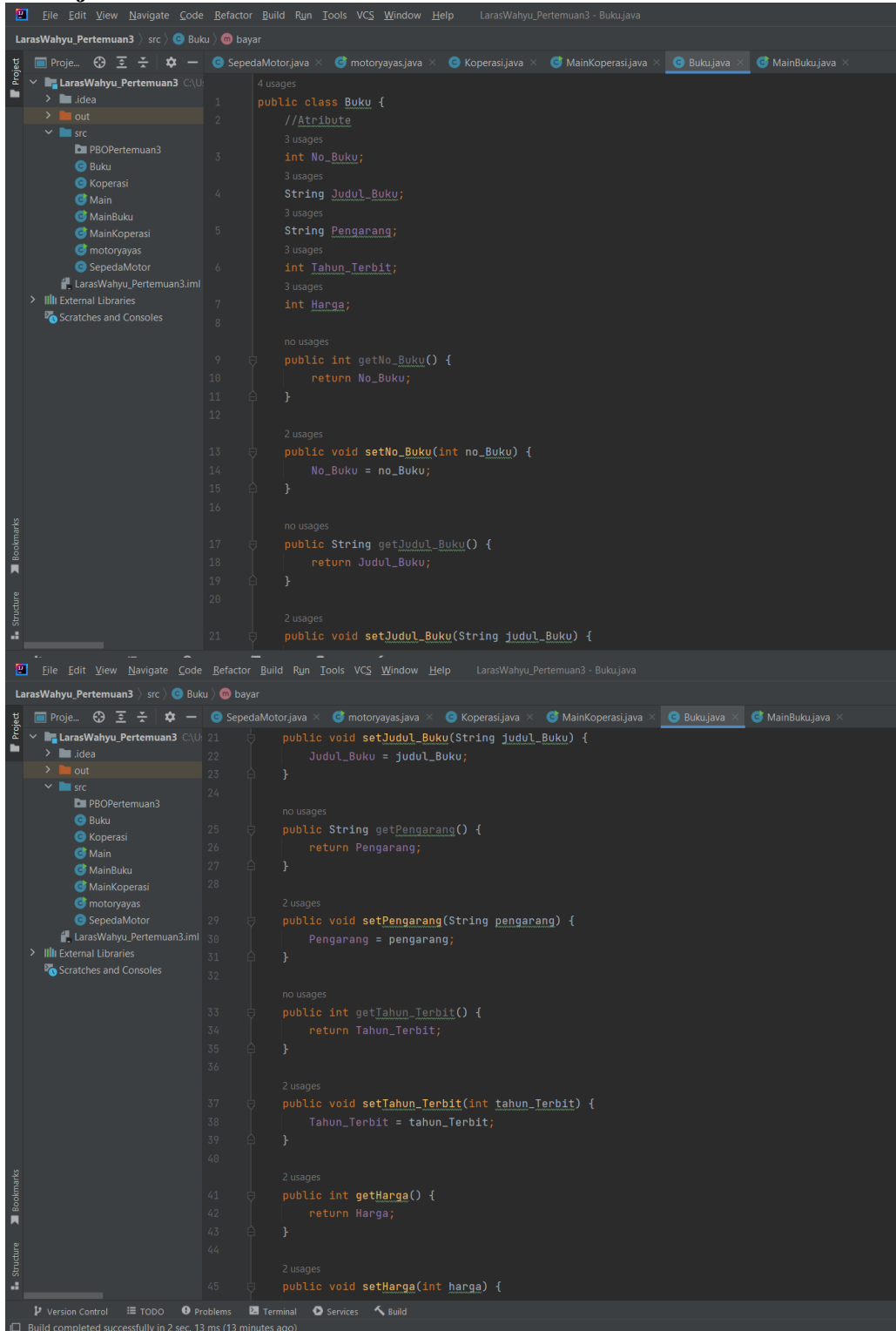
Kemudian buatlah **Class** dengan nama **main.java** dengan ketentuan sebagai berikut

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

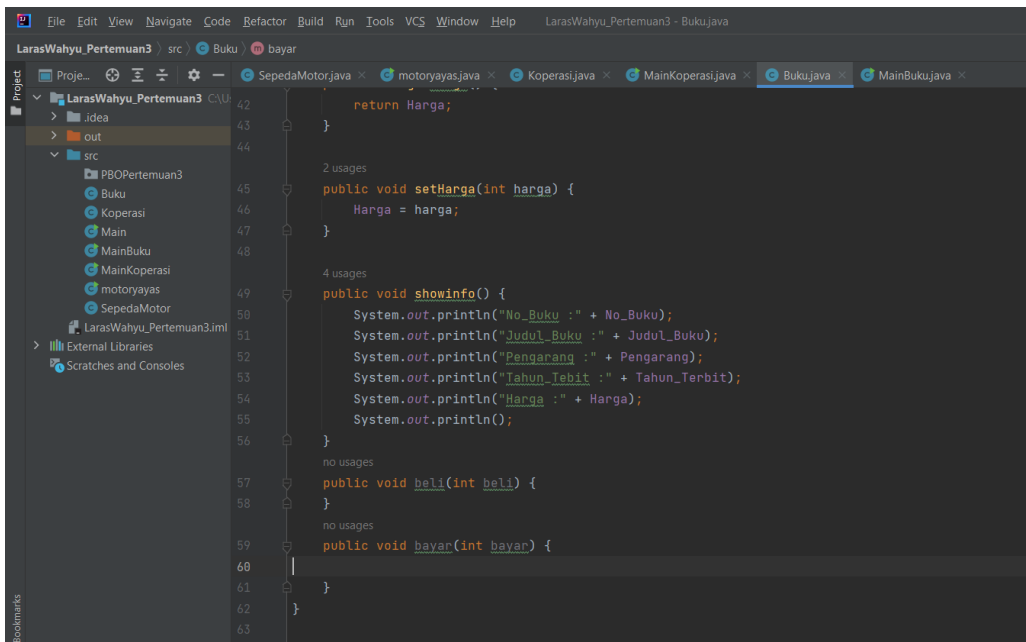
Pada bagian method void beli_buku(int) menampilkan total harga dari jumlah buku yang ingin dibeli.

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.java



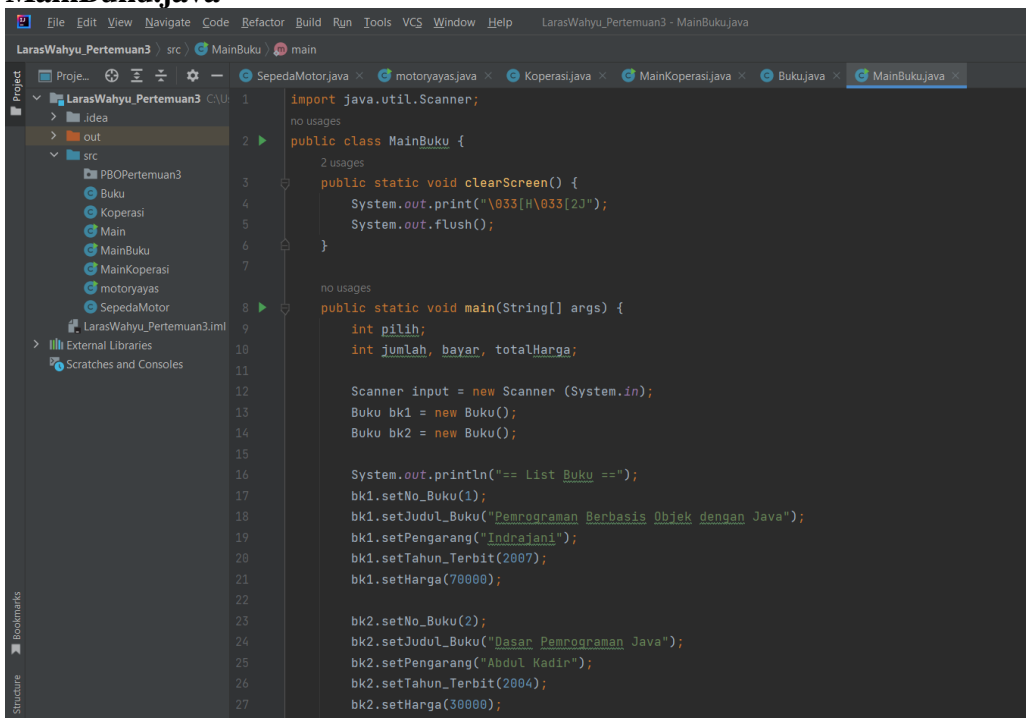
```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - Buku.java
LarasWahyu_Pertemuan3 src Buku bayar
Project
  LarasWahyu_Pertemuan3 C:\U
    .idea
    out
    src
      PBOPertemuan3
        Buku
        Koperasi
        Main
        MainBuku
        MainKoperasi
        motoryayas
        SepedaMotor
      LarasWahyu_Pertemuan3.iml
    External Libraries
    Scratches and Consoles
  Structure
  Bookmarks
1 public class Buku {
2     //Attribute
3     int No_Buku;
4     String Judul_Buku;
5     String Pengarang;
6     int Tahun_Terbit;
7     int Harga;
8
9     no usages
10    public int getNo_Buku() {
11        return No_Buku;
12    }
13
14    2 usages
15    public void setNo_Buku(int no_Buku) {
16        No_Buku = no_Buku;
17    }
18
19    no usages
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    no usages
30    public String getPengarang() {
31        return Pengarang;
32    }
33
34    2 usages
35    public void setPengarang(String pengarang) {
36        Pengarang = pengarang;
37    }
38
39    no usages
40    public int getTahun_Terbit() {
41        return Tahun_Terbit;
42    }
43
44    2 usages
45    public void setTahun_Terbit(int tahun_Terbit) {
46        Tahun_Terbit = tahun_Terbit;
47    }
48
49    2 usages
50    public int getHarga() {
51        return Harga;
52    }
53
54    2 usages
55    public void setHarga(int harga) {
56
57    }
58    }
59
60    Build completed successfully in 2 sec. 13 ms (13 minutes ago)
```



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - Buku.java
LarasWahyu_Pertemuan3 > src > Buku > bayar
Project
  LarasWahyu_Pertemuan3
    .idea
    out
    src
      PBOPertemuan3
        Buku
        Koperasi
        Main
        MainBuku
        MainKoperasi
        motoryayas
        SepedaMotor
      LarasWahyu_Pertemuan3.iml
    External Libraries
    Scratches and Consoles
  Bookmarks
  Structure

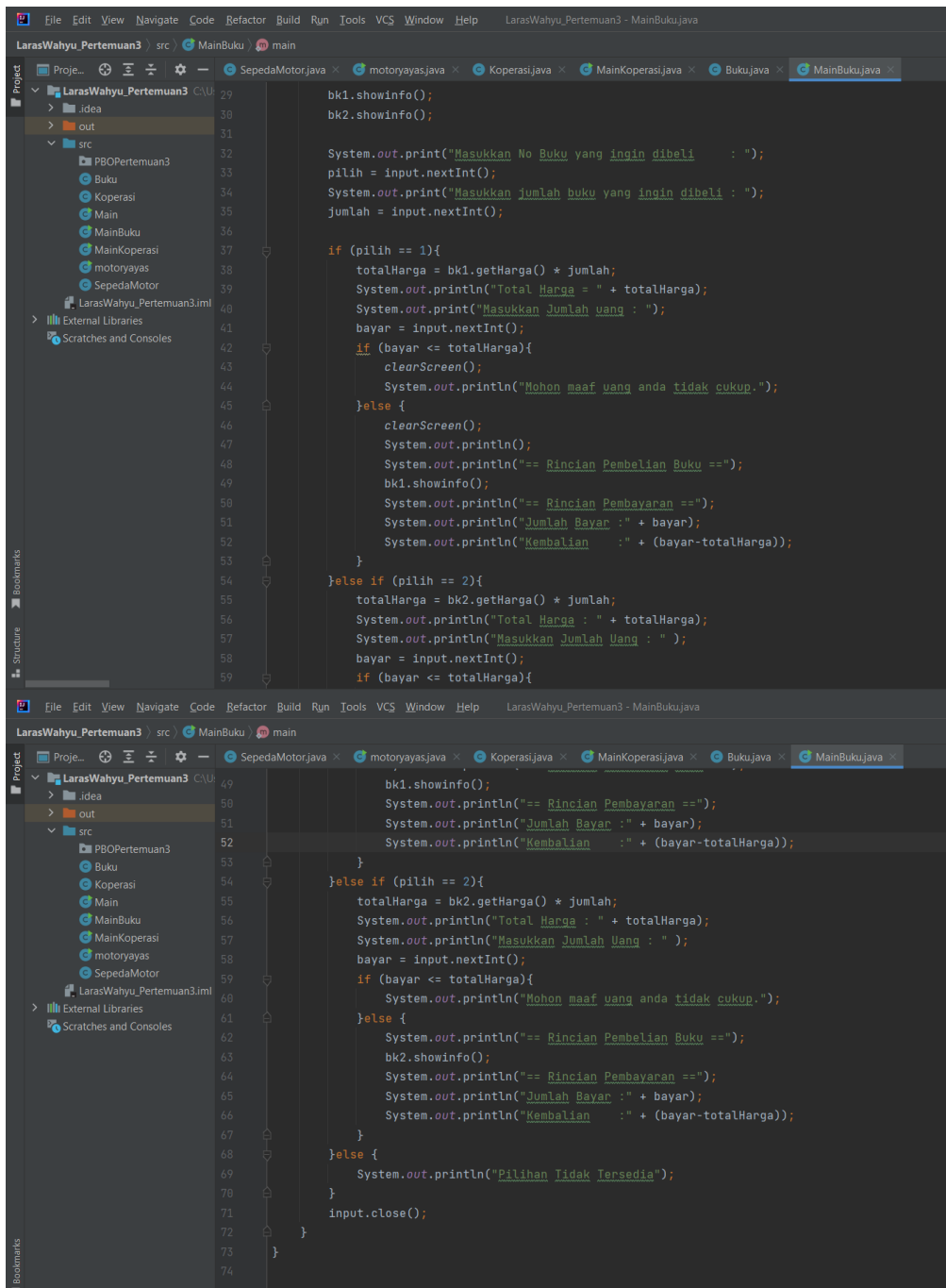
42   return Harga;
43   }
44
45   2 usages
46   public void setHarga(int harga) {
47       Harga = harga;
48   }
49
50   4 usages
51   public void showinfo() {
52       System.out.println("No_Buku : " + No_Buku);
53       System.out.println("Judul_Buku : " + Judul_Buku);
54       System.out.println("Pengarang : " + Pengarang);
55       System.out.println("Tahun_Terbit : " + Tahun_Terbit);
56       System.out.println("Harga : " + Harga);
57       System.out.println();
58   }
59   no usages
60   public void beli(int beli) {
61   }
62   no usages
63   public void bayar(int bayar) {
64   }
65   }
```

Source code MainBuku.java

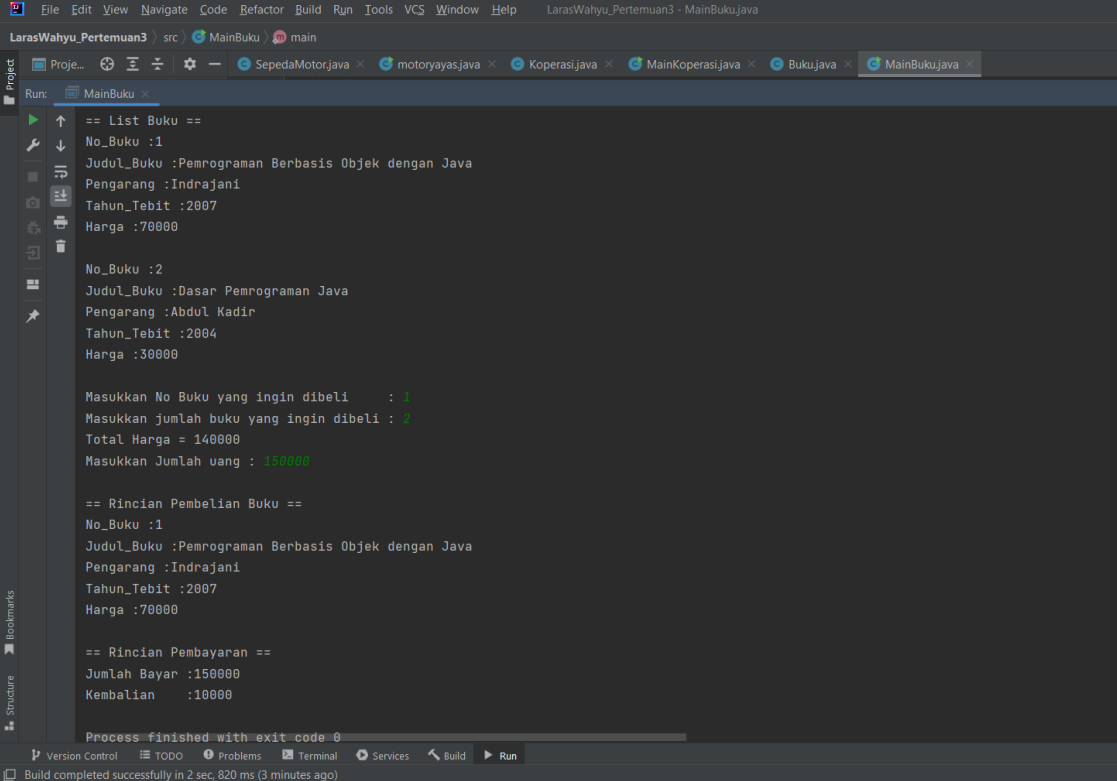


```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - MainBuku.java
LarasWahyu_Pertemuan3 > src > MainBuku > main
Project
  LarasWahyu_Pertemuan3
    .idea
    out
    src
      PBOPertemuan3
        Buku
        Koperasi
        Main
        MainBuku
        MainKoperasi
        motoryayas
        SepedaMotor
      LarasWahyu_Pertemuan3.iml
    External Libraries
    Scratches and Consoles
  Bookmarks
  Structure

1   import java.util.Scanner;
2
3   no usages
4   public class MainBuku {
5       2 usages
6       public static void clearScreen() {
7           System.out.print("\033[H\033[2J");
8           System.out.flush();
9       }
10
11   no usages
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 bk1 = new Buku();
18       Buku bk2 = new Buku();
19
20       System.out.println("== List Buku ==");
21       bk1.setNo_Buku(1);
22       bk1.setJudul_Buku("Pemrograman Berbasis Objek dengan Java");
23       bk1.setPengarang("Indrajani");
24       bk1.setTahun_Terbit(2007);
25       bk1.setHarga(70000);
26
27       bk2.setNo_Buku(2);
28       bk2.setJudul_Buku("Dasar Pemrograman Java");
29       bk2.setPengarang("Abdul Kadir");
30       bk2.setTahun_Terbit(2004);
31       bk2.setHarga(30000);
32   }
```

Screenshoot program



```
File Edit View Navigate Code Refactor Build Run Tools VCS Window Help LarasWahyu_Pertemuan3 - MainBuku.java
LarasWahyu_Pertemuan3 > src > MainBuku > main
Project: MainBuku
Run: MainBuku
== List Buku ==
No_Buku :1
Judul_Buku :Pemrograman Berbasis Objek dengan Java
Pengarang :Indrajani
Tahun_Tebit :2007
Harga :70000

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

Masukkan No Buku yang ingin dibeli : 1
Masukkan jumlah buku yang ingin dibeli : 1
Total Harga = 140000
Masukkan Jumlah uang : 150000

== Rincian Pembelian Buku ==
No_Buku :1
Judul_Buku :Pemrograman Berbasis Objek dengan Java
Pengarang :Indrajani
Tahun_Tebit :2007
Harga :70000

== Rincian Pembayaran ==
Jumlah Bayar :150000
Kembalian :10000

Process finished with exit code 0
Build completed successfully in 2 sec, 820 ms (3 minutes ago)
```

BAB IV

KESIMPULAN

Kesimpulan pada Praktikum ini adalah kita belajar tentang kelas dan objek. Kelas dapat didefinisikan sebagai template/blueprint yang menjelaskan perilaku/status yang didukung oleh objek dari tipenya. Objek adalah turunan dari kelas, sebuah objek bisa didefinisikan setelah suatu kelas 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.

Perbedaannya adalah kelas merupakan pemodelan dari objek yang berisi informasi tentang karakteristik(data) dan tingkah laku yang dimiliki oleh objek tersebut(metode), sedangkan objek merupakan perwujudan dari suatu kelas.