

**LAPORAN PRAKTIKUM  
PERROGRAMAN BERORIENTASI OBJEK**

**MODUL III  
“KELAS DAN OBJEK”**



**Disusun oleh:**

**FRANTINO SENEGAL**

**21102225**

**S1 IF-09 O**

**Dosen Pengampu:**

**DEDDY AGUNG PRABOWO, S.Kom., M.Kom.**

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

## **BAB I**

### **TUJUAN PRAKTIKUM**

1. Mahasiswa diharapkan mampu memahami mengenai objek dan kelas
2. 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

#### A. Membuat Class Baru

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

```
1 package com.pbo.pertemuan3;
2
3 //6 usages
4 public class Sepeda_Motor {
5     //Attribute
6     4 usages
7     String merk;
8     4 usages
9     String tipe ;
10    4 usages
11    int harga;
12
13    //Constructor
14    3 usages
15    public Sepeda_Motor(){ }
16
17    //Method
18    3 usages
19    public void showinfo(){
20        System.out.println("merek : " + merk);
21        System.out.println("Tipe : " + tipe);
22        System.out.println("harga : " + harga);
23        System.out.println();
24    }
25 }
```

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 :

```
7 ▶ public class main_pertemuan3 {  
    no usages  
8 ▶     public static void main(String[] args) {  
9  
10         System.out.println("====DEALER FRANTINO====");  
11         Sepeda_Motor Suzuki = new Sepeda_Motor();  
12         Sepeda_Motor Yamaha = new Sepeda_Motor();  
13         Sepeda_Motor Honda = new Sepeda_Motor();  
14  
15         Suzuki.merk = "Suzuki";  
16         Suzuki.tipe = "GSX 150R";  
17         Suzuki.harga = 20000000;  
18  
19         Yamaha.merk = "Yamaha";  
20         Yamaha.tipe = "YZF R15";  
21         Yamaha.harga = 25000000;  
22  
23         Honda.merk = "Honda";  
24         Honda.tipe = "CBR 150";  
25         Honda.harga = 23500000;  
26  
27         Suzuki.showinfo();  
28         Yamaha.showinfo();  
29         Honda.showinfo();  
}
```

### Screenshot Program

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:  
====DEALER FRANTINO====  
merek :Suzuki  
Tipe :GSX 150R  
harga :20000000  
  
merek :Yamaha  
Tipe :YZF R15  
harga :25000000  
  
merek :Honda  
Tipe :CBR 150  
harga :23500000  
  
Process finished with exit code 0
```

## B. 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!)

```
12      //Constructor ber parameter
      no usages
13      public Sepeda_Motor(String merk, String tipe, int harga) {
14          this.merk = merk;
15          this.tipe = tipe;
16          this.harga = harga;
17      }
```

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

```
8      public static void main(String[] args) {
9
10         System.out.println("====DEALER FRANTINO====");
11         Sepeda_Motor Suzuki = new Sepeda_Motor( merk: "Suzuki", tipe: "G5X 150R", harga: 20000000);
12         Sepeda_Motor Yamaha = new Sepeda_Motor( merk: "Yamaha", tipe: "YZF R15", harga: 25000000);
13         Sepeda_Motor Honda = new Sepeda_Motor( merk: "Honda", tipe: "CBR 150R", harga: 23500000);
14
15         //Menjalankan method showinfo
16         Suzuki.showinfo();
17         Yamaha.showinfo();
18         Honda.showinfo();
19     }
```

## C. 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 :

```
3 public class Sepeda_Motor {
4     //Attribute
4 usages
5     private String merk;
4 usages
6     private String tipe ;
4 usages
7     private int harga;
8
9     //Constructor
3 usages
10    public Sepeda_Motor(){ }
11
12    //Constructor ber parameter
no usages
13    public Sepeda_Motor(String merk, String tipe, int harga) {
14        this.merk = merk;
15        this.tipe = tipe;
16        this.harga = harga;
17    }
18 }
```

#### D. 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 :

```
21 public String getMerk() {
22     return merk;
23 }
24
25 no usages
26 public void setMerk(String merk) {
27     this.merk = merk;
28 }
29
30 no usages
31 public String getTipe() {
32     return tipe;
33 }
34
35 no usages
36 public void setTipe(String tipe) {
37     this.tipe = tipe;
38 }
39
40 no usages
41 public int getHarga() {
42     return harga;
43 }
44
45 no usages
46 public void setHarga(int harga) {
47     this.harga = harga;
48 }
49 }
```

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

```
31 Sepeda_Motor vespa = new Sepeda_Motor( merk: "Vespa",
32     tipe: "Vespa Matic", harga: 40000000);
33 vespa.showinfo();
34
35 //Sebelum
36 System.out.println(
37     "Motor Merek : " + vespa.getMerk() +
38     "Dengan Tipe (Befor) : " + vespa.getTipe()
39 );
40 //Melakukan proses perubahan
41 vespa.setTipe("Sprint");
42
43 //Lihat hasil perubahan
44 System.out.println(
45     "Motor Merek : " + vespa.getMerk() +
46     "Dengan tipe (after) : " + vespa.getTipe()
47 );
48
49 }
50 }
```

### Screenshot Program

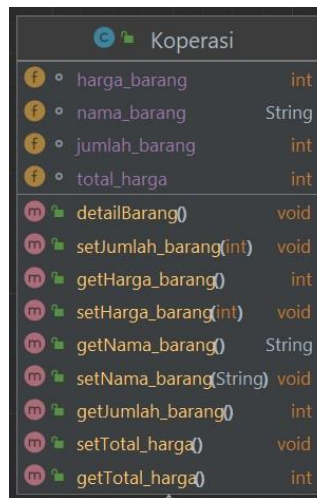
```
Motor Merek : VespaDengan Tipe (Befor) : Vespa Matic  
Motor Merek : VespaDengan tipe (after) : Sprint
```

## 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 atributenya
- Tampilkan detail barang
- Tampilkan total semua harga barang yang telah dibuat

## SourceCode

### Koperasi.java

```
public class Koperasi {  
    //Atrubute  
    String NamaBarang;  
    int HargaSatuan;  
    int JumlahBarang;  
    int TotalHarga;  
  
    //Constractor  
    public void Koperasi(){}  
  
    //Contractor berparameter  
    public Koperasi(String NamaBarang, int HargaSatuan, int  
JumlahBarang) {  
        this.NamaBarang = NamaBarang;  
        this.HargaSatuan = HargaSatuan;  
        this.JumlahBarang = JumlahBarang;  
    }  
  
    //Method  
    public String getNamaBarang() {  
        return NamaBarang;  
    }  
  
    public void setNamaBarang(String NamaBarang) {  
        this.NamaBarang = NamaBarang;  
    }  
  
    public int getHargaSatuan() {  
        return HargaSatuan;  
    }  
  
    public void setHargaSatuan(int HargaSatuan) {  
        this.HargaSatuan = HargaSatuan;  
    }  
  
    public int getJumlahBarang() {  
        return JumlahBarang;  
    }  
  
    public void setJumlahBarang(int JumlahBarang) {  
        this.JumlahBarang = JumlahBarang;  
    }  
  
    public int getTotalHarga() {  
        return TotalHarga;  
    }  
  
    public void setTotalHarga(int totalHarga) {  
        TotalHarga = totalHarga;  
    }  
}
```

```
    }

    public void showinfo() {
        System.out.println("Nama Barang      :" + NamaBarang);
        System.out.println("Harga Satuan   :" + HargaSatuan);
        System.out.println("Jumlah Barang  :" + JumlahBarang);
        System.out.println("Total Harga    :" + TotalHarga);
        System.out.println();
    }
}
```

## SourceCode

### Main.koperasi

```
public class main_koperasi {

    public static void main(String[] args) {
        System.out.println("NAMA : FRANTINO SENEGAL");
        System.out.println("NIM  : 21102225");
        System.out.println();

        System.out.println("====Toko Frantino====");
        Koperasi BukuTulis = new Koperasi("Buku Tulis",
            2000, 20);
        Koperasi Pensil    = new Koperasi("Pensil",
            3000, 10);

        BukuTulis.HargaSatuan = 5000;
        BukuTulis.JumlahBarang = 10;
        BukuTulis.TotalHarga= (BukuTulis.getHargaSatuan() *
BukuTulis.getJumlahBarang());

        Pensil.HargaSatuan = 3000;
        Pensil.JumlahBarang = 20;
        Pensil.TotalHarga= (Pensil.getHargaSatuan() *
Pensil.getJumlahBarang());

        BukuTulis.showinfo();
        Pensil.showinfo();
        System.out.println("=====");
        System.out.println("Total Harga :
"+(Pensil.getHargaSatuan() * Pensil.getJumlahBarang() +
(BukuTulis.getHargaSatuan() * BukuTulis.getJumlahBarang())));
        System.out.println("=====");
    }
}
```

## Screenshoot Program

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Progr
NAMA : FRANTINO SENEGAL
NIM : 21102225

====Toko Frantino====
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

Kelas Koperasi memiliki tiga atribut, yaitu namaBarang (String), hargaBarang (int), dan jumlahBarang (int). Terdapat method getter dan setter untuk masing-masing atribut, sehingga nilai atribut dapat diakses dan dimanipulasi dari luar kelas.

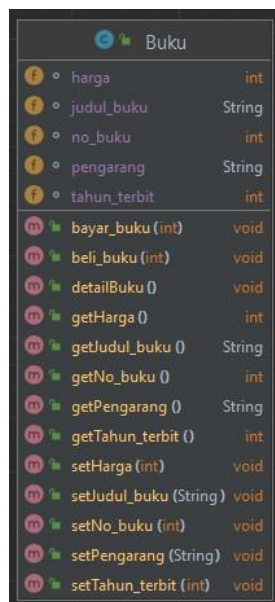
Kelas Main berfungsi untuk membuat objek dari kelas Koperasi, mengisi nilai atribut pada objek tersebut menggunakan method setter, menampilkan detail barang menggunakan method getter, dan menghitung total harga barang. Dalam program ini, terdapat dua objek Koperasi yang telah diinisialisasi dengan nilai atribut awal, yaitu bukuTulis dan pensil. Kemudian, method setter digunakan untuk mengisi nilai atribut pada masing-masing objek Koperasi. Setelah itu, detail barang seperti nama barang, harga satuan, dan jumlah barang ditampilkan menggunakan method getter. Terakhir, total harga barang dihitung dengan mengalikan harga satuan dengan jumlah barang pada masing-masing objek Koperasi, dan kemudian kedua total harga tersebut dijumlahkan dan ditampilkan.

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 :





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.java

```
public class Buku {  
    //Atribut  
    int No_Buku;  
    String Judul_Buku;  
    String Pengarang;  
    int Tahun_Terbit;  
    int Harga;  
  
    public int getNo_Buku() {  
        return No_Buku;  
    }  
  
    public void setNo_Buku(int no_Buku) {  
        No_Buku = no_Buku;  
    }  
  
    public String getJudul_Buku() {  
        return Judul_Buku;  
    }  
  
    public void setJudul_Buku(String judul_Buku) {  
        Judul_Buku = judul_Buku;  
    }  
  
    public String getPengarang() {  
        return Pengarang;  
    }  
}
```

```
}

public void setPengarang(String pengarang) {
    Pengarang = pengarang;
}

public int getTahun_Terbit() {
    return Tahun_Terbit;
}

public void setTahun_Terbit(int tahun_Terbit) {
    Tahun_Terbit = tahun_Terbit;
}

public int getHarga() {
    return Harga;
}

public void setHarga(int harga) {
    Harga = harga;
}

    public void showinfo() {
        System.out.println("NO_Buku      :" + No_Buku);
        System.out.println("Judul_Buku   :" + Judul_Buku);
        System.out.println("Pengarang   :" + Pengarang);
        System.out.println("Tahun_Tebit :" + Tahun_Terbit);
        System.out.println("Harga       :" + Harga);
        System.out.println();
    }
    public void beli(int beli) {

    }

    public void bayar (int bayar){

    }

}
```

## Source Code

### Main.buku

```
public class main_buku {
    public static void clearScreen() {
        System.out.print("\033[H\033[2J");
        System.out.flush();
    }
    public static void main(String[] args) {
        int pilih;
        int jumlah, bayar, totalHarga;

        Scanner input = new Scanner(System.in);
        Buku bk1 = new Buku();
        Buku bk2 = new Buku();

        System.out.println("NAMA : FRANTINO SENEGAL");
        System.out.println("NIM : 21102225");
        System.out.println();
        System.out.println("===List Buku===");
        bk1.setNo_Buku(1);
        bk1.setJudul_Buku("Pemrograman Berbasis Objek dengan Java");
        bk1.setPengarang("Indrajani");
        bk1.setTahun_Terbit(2007);
        bk1.setHarga(70000);

        bk2.setNo_Buku(2);
        bk2.setJudul_Buku("Dasar Pemrograman Java");
        bk2.setPengarang("Abdul Kadir");
        bk2.setTahun_Terbit(2004);
        bk2.setHarga(30000);

        bk1.showinfo();
        bk2.showinfo();

        System.out.println("=====");
        System.out.print("Masukkan No Buku yang ingin dibeli : ");
        pilih = input.nextInt();
        System.out.print("Masukkan jumlah buku yang ingin dibeli : ");
        jumlah = input.nextInt();

        if (pilih == 1){
            totalHarga = bk1.getHarga() * jumlah;
            System.out.println("Total Harga = " + totalHarga);
            System.out.print("Masukkan Jumlah uang : ");
            bayar = input.nextInt();
            if (bayar <= totalHarga){
                clearScreen();
                System.out.println("Mohon maaf uang anda tidak cukup.");
            }
        }
    }
}
```

```
        }else {
            clearScreen();
            System.out.println("===Rincian pembelian buku===");
            bk1.showinfo();
            System.out.println("===Rincian Pembayaran===");
            System.out.println("Jumlah Bayar :" + bayar);
            System.out.println("Kembalian      :" + (bayar-
totalHarga));
        }
    }else if (pilih == 2){
        totalHarga = bk2.getHarga() * jumlah;
        System.out.println("Total Harga : " + totalHarga);
        System.out.println("Masukkan Jumlah Uang : " );
        bayar = input.nextInt();
        if (bayar <= totalHarga){
            System.out.println("Mohon maaf uang anda tidak
cukup.");
        }else {
            System.out.println("=== Rincian Pembayaran ===");
            bk2.showinfo();
            System.out.println("=== Rincian Pembayaran");
            System.out.println("Jumlah Bayar :" + bayar);
            System.out.println("Kembalian      :" + (bayar-
totalHarga));
        }
    }else {
        System.out.println("Pilihan Tidak Tersedia");
    }
    input.close();
}
}
```

## Screenshot Program

```
"C:\Program Files\Java\jdk-19\bin\java.exe" "-javaagent:C:\Program
NAMA : FRANTINO SENEGAL
NIM : 21102225

===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 : 2
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
```

## Deskripsi Program

Program ini terdiri dari dua buah class, yaitu class Buku yang berfungsi untuk menampung data mengenai buku yang tersedia, dan class Main yang berfungsi sebagai program utama untuk melakukan pembelian buku.

Pada class Buku, terdapat beberapa atribut seperti nomor buku, judul buku, pengarang, tahun terbit, dan harga. Setiap atribut tersebut memiliki setter dan getter yang digunakan untuk mengisi dan mengambil nilai atribut.

Pada class Main, terdapat beberapa method seperti `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. Selanjutnya, pada method `bayar_buku()`, user diminta untuk memasukkan jumlah uang yang dibayarkan, kemudian program akan menghitung kembalian dan menampilkan detail buku dan jumlah bayar.

Setelah proses pembelian selesai, program akan menampilkan rincian pembelian buku dan rincian pembayaran.

## **BAB IV**

### **KESIMPULAN**

Dalam pemrograman berorientasi objek, kelas dan objek adalah konsep fundamental yang memungkinkan programmer untuk mengorganisir kode mereka ke dalam unit-unit yang lebih besar yang terdiri dari atribut dan metode yang berfungsi bersama-sama untuk memodelkan suatu entitas atau konsep dalam program.

Dalam pembuatan kelas, dasar teori yang harus dipahami meliputi abstraksi, encapsulation, pewarisan, dan polimorfisme. Dengan menggunakan konsep kelas dan objek, programmer dapat membuat program yang lebih terstruktur, mudah dipelihara, dan dapat diperluas di masa depan.

Dengan demikian, pemahaman yang baik tentang konsep kelas dan objek sangat penting dalam pemrograman berorientasi objek dan sangat membantu dalam pengembangan program yang efektif dan efisien