

## Modul 2 Class dan Object

### Tugas Pendahuluan

1. Apa yang dimaksud dengan class..?
2. Apa yang dimaksud dengan Object..?
3. Apa perbedaan antara class dan object..?

### Materi

#### 1. Class

Dalam pemrograman berorientasi objek (OOP), sebuah class merupakan sebuah blueprint atau template yang mendefinisikan struktur, perilaku, dan karakteristik suatu objek. Class digunakan sebagai kerangka dasar untuk membuat objek-objek dengan atribut-atribut (variabel anggota) dan perilaku (metode).

Class menggambarkan suatu jenis objek atau entitas tertentu yang memiliki properti dan perilaku yang sama. Properti dalam class dikenal sebagai variabel anggota (instance variables) yang merepresentasikan data atau informasi yang terkait dengan objek tersebut. Sementara itu, perilaku dalam class diimplementasikan melalui metode, yang adalah fungsi-fungsi atau prosedur-prosedur yang dapat dilakukan oleh objek tersebut.

Dalam class, Anda dapat mendefinisikan variabel anggota, metode, konstruktor, dan metode khusus lainnya. Variabel anggota menyimpan data yang spesifik untuk setiap objek yang dibuat dari class tersebut. Metode digunakan untuk memanipulasi data, menjalankan operasi, atau memberikan fungsionalitas tertentu terhadap objek. Konstruktor adalah metode khusus yang digunakan untuk menginisialisasi objek saat objek dibuat. Metode khusus lainnya, seperti getter dan setter, dapat digunakan untuk mengakses dan mengubah nilai variabel anggota pada objek.

Contoh class misalkan class Mobil, menggambarkan blueprint atau template untuk menciptakan objek-objek mobil. Class tersebut mendefinisikan atribut-atribut seperti merk, warna, dan tahun produksi, serta perilaku-perilaku seperti menyalakan, mematikan, dan menampilkan informasi mobil.

Dengan menggunakan konsep class dalam OOP, kita dapat membuat banyak objek yang berbeda dengan menggunakan struktur dan perilaku yang sama yang telah didefinisikan dalam class tersebut.

Contoh Class Mobil :

```
1. public class Mobil {
2.     // Variabel anggota (instance variables)
3.     private String merk;
4.     private String warna;
5.     private int tahunProduksi;
6.
7.     // Konstruktor
8.     public Mobil(String merk, String warna, int tahunProduksi) {
9.         this.merk = merk;
10.        this.warna = warna;
```

```

11.         this.tahunProduksi = tahunProduksi;
12.     }
13.
14.     // Metode getter dan setter untuk variabel anggota
15.     public String getMerk() {
16.         return merk;
17.     }
18.
19.     public void setMerk(String merk) {
20.         this.merk = merk;
21.     }
22.
23.     public String getWarna() {
24.         return warna;
25.     }
26.
27.     public void setWarna(String warna) {
28.         this.warna = warna;
29.     }
30.
31.     public int getTahunProduksi() {
32.         return tahunProduksi;
33.     }
34.
35.     public void setTahunProduksi(int tahunProduksi) {
36.         this.tahunProduksi = tahunProduksi;
37.     }
38.
39.     // Metode lain
40.     public void start() {
41.         System.out.println("Mobil " + merk + " dinyalakan.");
42.     }
43.
44.     public void stop() {
45.         System.out.println("Mobil " + merk + " dimatikan.");
46.     }
47.
48.     public void info() {
49.         System.out.println("Mobil " + merk + " (" + warna + ")");
50.         System.out.println("Tahun Produksi: " + tahunProduksi);
51.     }
52. }

```

Dalam contoh di atas, class Mobil memiliki beberapa variabel anggota (instance variables) yaitu merk, warna, dan tahunProduksi. Class Mobil juga memiliki konstruktor yang digunakan untuk menginisialisasi nilai variabel anggota pada saat objek Mobil dibuat.

Terdapat juga metode getter dan setter untuk masing-masing variabel anggota, yang digunakan untuk mengakses dan mengubah nilai variabel tersebut.

Selain itu, terdapat juga beberapa metode lain seperti start() yang mencetak pesan mobil dinyalakan, stop() yang mencetak pesan mobil dimatikan, dan info() yang mencetak informasi lengkap tentang mobil seperti merk, warna, dan tahun produksi.

Dengan menggunakan definisi class Mobil di atas, Anda dapat membuat objek-objek Mobil yang memiliki atribut-atribut seperti merk, warna, dan tahun produksi, serta dapat melakukan

operasi-operasi seperti menyalakan dan mematikan mobil, serta menampilkan informasi tentang mobil tersebut.

#### Contoh Class Mahasiswa

```
1. public class Mahasiswa {
2.     // Variabel anggota (instance variables)
3.     private String nim;
4.     private String nama;
5.     private String jurusan;
6.
7.     // Konstruktor
8.     public Mahasiswa(String nim, String nama, String jurusan) {
9.         this.nim = nim;
10.        this.nama = nama;
11.        this.jurusan = jurusan;
12.    }
13.
14.    // Metode getter dan setter untuk variabel anggota
15.    public String getNim() {
16.        return nim;
17.    }
18.
19.    public void setNim(String nim) {
20.        this.nim = nim;
21.    }
22.
23.    public String getNama() {
24.        return nama;
25.    }
26.
27.    public void setNama(String nama) {
28.        this.nama = nama;
29.    }
30.
31.    public String getJurusan() {
32.        return jurusan;
33.    }
34.
35.    public void setJurusan(String jurusan) {
36.        this.jurusan = jurusan;
37.    }
38.
39.    // Metode lain
40.    public void info() {
41.        System.out.println("NIM: " + nim);
42.        System.out.println("Nama: " + nama);
43.        System.out.println("Jurusan: " + jurusan);
44.    }
45. }
```

Dalam contoh di atas, class Mahasiswa memiliki tiga variabel anggota (instance variables) yaitu nim, nama, dan jurusan. Class Mahasiswa juga memiliki konstruktor yang digunakan untuk menginisialisasi nilai variabel anggota pada saat objek Mahasiswa dibuat.

Terdapat juga metode getter dan setter untuk masing-masing variabel anggota, yang digunakan untuk mengakses dan mengubah nilai variabel tersebut. Selain itu, terdapat juga metode info() yang mencetak informasi lengkap tentang mahasiswa seperti NIM, Nama, dan Jurusan.

Dengan menggunakan definisi class Mahasiswa di atas, Anda dapat membuat objek-objek Mahasiswa yang memiliki atribut-atribut seperti NIM, Nama, dan Jurusan, serta dapat memanggil metode info() untuk menampilkan informasi tentang mahasiswa tersebut.

Contoh Class Buku :

```
1. public class Buku {
2.     // Variabel anggota (instance variables)
3.     private String judul;
4.     private String penulis;
5.     private int tahunTerbit;
6.
7.     // Konstruktor
8.     public Buku(String judul, String penulis, int tahunTerbit) {
9.         this.judul = judul;
10.        this.penulis = penulis;
11.        this.tahunTerbit = tahunTerbit;
12.    }
13.
14.    // Metode getter dan setter untuk variabel anggota
15.    public String getJudul() {
16.        return judul;
17.    }
18.
19.    public void setJudul(String judul) {
20.        this.judul = judul;
21.    }
22.
23.    public String getPenulis() {
24.        return penulis;
25.    }
26.
27.    public void setPenulis(String penulis) {
28.        this.penulis = penulis;
29.    }
30.
31.    public int getTahunTerbit() {
32.        return tahunTerbit;
33.    }
34.
35.    public void setTahunTerbit(int tahunTerbit) {
36.        this.tahunTerbit = tahunTerbit;
37.    }
38.
39.    // Metode lain
40.    public void info() {
41.        System.out.println("Judul: " + judul);
42.        System.out.println("Penulis: " + penulis);
43.        System.out.println("Tahun Terbit: " + tahunTerbit);
44.    }
45. }
```

Dalam contoh di atas, class Buku memiliki tiga variabel anggota (instance variables) yaitu judul, penulis, dan tahunTerbit. Class Buku juga memiliki konstruktor yang digunakan untuk menginisialisasi nilai variabel anggota pada saat objek Buku dibuat. Terdapat juga metode getter dan setter untuk masing-masing variabel anggota, yang digunakan untuk mengakses dan mengubah nilai variabel tersebut. Selain itu, terdapat juga metode info() yang mencetak informasi lengkap tentang buku seperti Judul, Penulis, dan Tahun Terbit.

Dengan menggunakan definisi class Buku di atas, Anda dapat membuat objek-objek Buku yang memiliki atribut-atribut seperti Judul, Penulis, dan Tahun Terbit, serta dapat memanggil metode info() untuk menampilkan informasi tentang buku tersebut.

## 2. Objek

Dalam pemrograman berorientasi objek (OOP), objek dan class adalah dua konsep yang saling terkait namun memiliki perbedaan yang penting.

### **Class:**

1. Class adalah sebuah blueprint atau template yang mendefinisikan struktur, perilaku, dan karakteristik umum dari suatu objek.
2. Class menyediakan kerangka dasar untuk menciptakan objek-objek yang memiliki atribut dan perilaku yang sama.
3. Class digunakan untuk mendefinisikan variabel anggota (instance variables), metode, dan konstruktor yang akan dimiliki oleh objek-objek yang dibuat dari class tersebut.
4. Class hanya merupakan representasi konseptual atau abstrak dari suatu objek, dan tidak memiliki keberadaan fisik di dalam program.

### **Objek:**

1. Objek adalah instansi konkret atau nyata dari suatu class.
2. Objek merupakan manifestasi atau hasil dari class yang telah diinstansiasi atau dibuat.
3. Objek memiliki keberadaan fisik di dalam program dan dapat digunakan untuk melakukan operasi dan menyimpan data.
4. Setiap objek yang dibuat dari class memiliki keunikannya sendiri dan dapat memiliki nilai yang berbeda untuk variabel anggota (instance variables) yang dimilikinya.

### **Perbedaan antara objek dan class:**

1. Class adalah kerangka dasar atau blueprint yang mendefinisikan struktur dan perilaku objek, sedangkan objek adalah instansi konkret atau nyata dari class tersebut.
2. Class hanya memiliki definisi dan tidak memiliki keberadaan fisik, sedangkan objek memiliki keberadaan fisik dan dapat dioperasikan di dalam program.
3. Class digunakan untuk membuat banyak objek dengan atribut dan perilaku yang sama, sedangkan objek mewakili satu entitas tunggal yang memiliki nilai unik untuk atributnya.
4. Class berfungsi sebagai template atau kerangka untuk menciptakan objek, sementara objek adalah entitas yang dapat melakukan operasi dan menyimpan data.

Dalam OOP, kita menggunakan class sebagai blueprint untuk membuat objek-objek yang memiliki karakteristik dan perilaku yang sama. Class memberikan struktur dan definisi, sementara objek merupakan manifestasi nyata dari class tersebut yang dapat berinteraksi dengan program.

Contoh object dari class mobil diatas :

```
1. public class Main {
2.     public static void main(String[] args) {
3.         // Membuat objek mobil1
4.         Mobil mobil1 = new Mobil("Toyota", "Merah", 2022);
5.
6.         // Mengakses atribut dan memanggil metode objek mobil1
7.         System.out.println("Merk mobil1: " + mobil1.getMerk());
8.         System.out.println("Warna mobil1: " + mobil1.getWarna());
9.         System.out.println("Tahun Produksi mobil1: " + mobil1.getTahunProduksi());
10.        mobil1.start();
11.        mobil1.info();
12.
13.        System.out.println();
14.
15.        // Membuat objek mobil2
16.        Mobil mobil2 = new Mobil("Honda", "Biru", 2021);
17.
18.        // Mengakses atribut dan memanggil metode objek mobil2
19.        System.out.println("Merk mobil2: " + mobil2.getMerk());
20.        System.out.println("Warna mobil2: " + mobil2.getWarna());
21.        System.out.println("Tahun Produksi mobil2: " + mobil2.getTahunProduksi());
22.        mobil2.start();
23.        mobil2.info();
24.    }
25. }
```

Dalam contoh di atas, kita membuat dua objek mobil1 dan mobil2 berdasarkan class Mobil. Setiap objek memiliki atribut-atribut seperti merk, warna, dan tahun produksi yang diberikan melalui konstruktor saat objek dibuat.

Kemudian, kita mengakses atribut-atribut dan memanggil metode pada masing-masing objek. Misalnya, kita menggunakan metode getMerk(), getWarna(), dan getTahunProduksi() untuk mengakses nilai atribut pada objek, serta memanggil metode start() dan info() pada objek untuk menjalankan operasi dan menampilkan informasi mobil.

Dengan cara ini, kita dapat membuat objek-objek yang berbeda dengan atribut-atribut yang berbeda pula, tetapi memiliki perilaku yang sama seperti yang telah didefinisikan dalam class Mobil.

File: Mobil.java

```
1. public class Mobil {
2.     private String merk;
3.     private String warna;
4.     private int tahunProduksi;
5.
6.     public Mobil(String merk, String warna, int tahunProduksi) {
7.         this.merk = merk;
8.         this.warna = warna;
9.         this.tahunProduksi = tahunProduksi;
10.    }
11.
12.    public String getMerk() {
13.        return merk;
14.    }
15. }
```

```

16.     public void setMerk(String merk) {
17.         this.merk = merk;
18.     }
19.
20.     public String getWarna() {
21.         return warna;
22.     }
23.
24.     public void setWarna(String warna) {
25.         this.warna = warna;
26.     }
27.
28.     public int getTahunProduksi() {
29.         return tahunProduksi;
30.     }
31.
32.     public void setTahunProduksi(int tahunProduksi) {
33.         this.tahunProduksi = tahunProduksi;
34.     }
35.
36.     public void start() {
37.         System.out.println("Mobil dinyalakan");
38.     }
39.
40.     public void info() {
41.         System.out.println("Merk: " + merk);
42.         System.out.println("Warna: " + warna);
43.         System.out.println("Tahun Produksi: " + tahunProduksi);
44.     }
45. }
46.

```

File: Main.java

```

1. public class Main {
2.     public static void main(String[] args) {
3.         Mobil mobil1 = new Mobil("Toyota", "Merah", 2022);
4.
5.         System.out.println("Merk mobil1: " + mobil1.getMerk());
6.         System.out.println("Warna mobil1: " + mobil1.getWarna());
7.         System.out.println("Tahun Produksi mobil1: " + mobil1.getTahunProduksi());
8.         mobil1.start();
9.         mobil1.info();
10.    }
11. }

```

Simpan kedua file tersebut dengan nama yang sesuai, yaitu "Mobil.java" dan "Main.java". Kemudian, kompilasi dan jalankan program dengan perintah berikut pada command line:

```

javac Mobil.java
javac Main.java
java Main

```

Dengan demikian, program akan mengkompilasi kedua file .java menjadi bytecode dan menjalankan program Main, sehingga Anda dapat melihat output program yang mencetak informasi tentang mobil.

File Mahasiswa.java

```

1. public class Mahasiswa {
2.     private String nim;
3.     private String nama;
4.     private String jurusan;
5.
6.     public Mahasiswa(String nim, String nama, String jurusan) {
7.         this.nim = nim;
8.         this.nama = nama;
9.         this.jurusan = jurusan;
10.    }
11.
12.    public String getNim() {
13.        return nim;
14.    }
15.
16.    public void setNim(String nim) {
17.        this.nim = nim;
18.    }
19.
20.    public String getNama() {
21.        return nama;
22.    }
23.
24.    public void setNama(String nama) {
25.        this.nama = nama;
26.    }
27.
28.    public String getJurusan() {
29.        return jurusan;
30.    }
31.
32.    public void setJurusan(String jurusan) {
33.        this.jurusan = jurusan;
34.    }
35.
36.    public void info() {
37.        System.out.println("NIM: " + nim);
38.        System.out.println("Nama: " + nama);
39.        System.out.println("Jurusan: " + jurusan);
40.    }
41.
42.    public static void main(String[] args) {
43.        Mahasiswa mahasiswa1 = new Mahasiswa("123456789", "John Doe", "Teknik
Informatika");
44.
45.        System.out.println("Data Mahasiswa 1:");
46.        mahasiswa1.info();
47.
48.        System.out.println();
49.
50.        Mahasiswa mahasiswa2 = new Mahasiswa("987654321", "Jane Smith", "Sistem
Informasi");
51.
52.        System.out.println("Data Mahasiswa 2:");
53.        mahasiswa2.info();
54.    }
55. }

```



Dalam contoh di atas, kita menyertakan definisi class "Mahasiswa" dan juga blok kode main() untuk menjalankan program.

Pada blok main(), kita membuat dua objek mahasiswa1 dan mahasiswa2 berdasarkan class Mahasiswa. Setiap objek memiliki atribut-atribut seperti NIM, Nama, dan Jurusan yang diberikan melalui konstruktor saat objek dibuat.

Kemudian, kita mengakses atribut-atribut objek menggunakan metode getter, memanggil metode info() pada objek, dan mencetak hasilnya menggunakan System.out.println().

Anda dapat menyimpan program tersebut dalam sebuah file dengan ekstensi .java, misalnya MahasiswaProgram.java. Kemudian, Anda dapat mengkompilasi dan menjalankan program tersebut menggunakan perintah-perintah berikut pada command line:

```
javac MahasiswaProgram.java
java MahasiswaProgram
```

Dengan demikian, program akan mengkompilasi file .java menjadi bytecode dan menjalankannya, sehingga Anda dapat melihat output program yang mencetak informasi tentang mahasiswa.

### 3. Konstruktor (constructor)

Konstruktor (constructor) dalam Java adalah metode khusus yang digunakan untuk menginisialisasi objek dari suatu kelas. Konstruktor memiliki nama yang sama dengan nama kelasnya dan tidak mengembalikan nilai (tidak ada tipe pengembalian yang didefinisikan).

Berikut adalah contoh penggunaan konstruktor dalam sebuah kelas "Mobil":

```
1. public class Mobil {
2.     private String merk;
3.     private String warna;
4.
5.     // Konstruktor
6.     public Mobil(String merk, String warna) {
7.         this.merk = merk;
8.         this.warna = warna;
9.     }
10.
11.    // Metode lainnya
12.    public void infoMobil() {
13.        System.out.println("Mobil " + merk + " berwarna " + warna);
14.    }
15.
16.    public static void main(String[] args) {
17.        Mobil mobil1 = new Mobil("Toyota", "Hitam");
18.        mobil1.infoMobil();
19.
20.        Mobil mobil2 = new Mobil("Honda", "Merah");
21.        mobil2.infoMobil();
22.    }
23. }
```

Penjelasan:

1. Dalam kelas "Mobil", terdapat konstruktor yang menerima dua parameter merk dan warna. Ketika objek mobil dibuat, konstruktor ini akan dijalankan untuk menginisialisasi nilai merk dan warna.
2. Metode infoMobil() digunakan untuk mencetak informasi tentang mobil, yaitu merk dan warnanya.
3. Dalam metode main, objek "mobil1" dan "mobil2" dibuat dengan menggunakan konstruktor. Nilai yang diberikan sebagai argumen akan digunakan untuk menginisialisasi atribut merk dan warna pada setiap objek.
4. Setelah objek dibuat, metode infoMobil() dipanggil untuk mencetak informasi tentang masing-masing mobil.

Output yang dihasilkan akan menjadi:

```
Mobil Toyota berwarna Hitam
Mobil Honda berwarna Merah
```

Dalam contoh ini, konstruktor digunakan untuk menginisialisasi atribut merk dan warna saat objek mobil dibuat. Anda dapat menambahkan lebih banyak konstruktor dengan berbagai parameter atau tanpa parameter sesuai kebutuhan Anda.

## Latihan

### Latihan 1:

Anda coba program dibawah ini dan kerjakan sesuai petunjuk.

```
1. public class Mahasiswa {
2.     private String nim;
3.     private String nama;
4.     private String jurusan;
5.
6.     public Mahasiswa(String nim, String nama, String jurusan) {
7.         this.nim = nim;
8.         this.nama = nama;
9.         this.jurusan = jurusan;
10.    }
11.
12.    public String getNim() {
13.        return nim;
14.    }
15.
16.    public void setNim(String nim) {
17.        this.nim = nim;
18.    }
19.
20.    public String getNama() {
21.        return nama;
22.    }
23.
24.    public void setNama(String nama) {
25.        this.nama = nama;
26.    }
27.
28.    public String getJurusan() {
29.        return jurusan;
30.    }
```

```

31.
32.     public void setJurusan(String jurusan) {
33.         this.jurusan = jurusan;
34.     }
35.
36.     public void info() {
37.         System.out.println("NIM: " + nim);
38.         System.out.println("Nama: " + nama);
39.         System.out.println("Jurusan: " + jurusan);
40.     }
41. }

```

```

1. public class Main {
2.     public static void main(String[] args) {
3.         Mahasiswa mhs1 = new Mahasiswa("123456789", "John Doe", "Teknik Informatika");
4.         Mahasiswa mhs2 = new Mahasiswa("987654321", "Jane Smith", "Sistem Informasi");
5.
6.         System.out.println("Data Mahasiswa 1:");
7.         mhs1.info();
8.
9.         System.out.println();
10.
11.        System.out.println("Data Mahasiswa 2:");
12.        mhs2.info();
13.
14.        System.out.println();
15.
16.        // Latihan 1: Ubah Nama Mahasiswa 1 menjadi "Michael Johnson"
17.        mhs1.setNama("Michael Johnson");
18.        System.out.println("Data Mahasiswa 1 (Setelah Diubah Nama):");
19.        mhs1.info();
20.
21.        System.out.println();
22.
23.        // Latihan 2: Tampilkan NIM dan Jurusan Mahasiswa 2
24.        System.out.println("NIM Mahasiswa 2: " + mhs2.getNim());
25.        System.out.println("Jurusan Mahasiswa 2: " + mhs2.getJurusan());
26.    }
27. }

```

Dalam latihan ini, kita membuat objek mahasiswa mhs1 dan mhs2 berdasarkan class Mahasiswa. Kemudian, kita memanggil metode info() pada masing-masing objek untuk menampilkan informasi mahasiswa.

Selanjutnya, terdapat dua latihan praktikum yang dapat Anda coba:

1. Mengubah Nama Mahasiswa 1 menjadi "Michael Johnson". Dalam latihan ini, Anda akan menggunakan metode setNama() pada objek mhs1 untuk mengubah nama mahasiswa, kemudian menampilkan kembali informasi mahasiswa tersebut menggunakan metode info().
2. Menampilkan NIM dan Jurusan Mahasiswa 2. Dalam latihan ini, Anda akan menggunakan metode getNim() dan getJurusan() pada objek mhs2 untuk mengambil nilai NIM dan Jurusan mahasiswa tersebut, kemudian mencetaknya menggunakan System.out.println().

Anda dapat menyimpan kedua file kode di atas dengan nama yang sesuai, yaitu "Mahasiswa.java" dan "Main.java". Kemudian, Anda dapat mengkompilasi dan menjalankan program tersebut menggunakan perintah-perintah berikut pada command line:

```
javac Mahasiswa.java
javac Main.java
java Main
```

Dari Latihan diatas coba amati hasilnya.

## Latihan 2:

Silahkan di coba program berikut ini dan kerjakan sesuai petunjuk :

```
1. public class Buku {
2.     private String judul;
3.     private String pengarang;
4.     private int tahunTerbit;
5.
6.     public Buku(String judul, String pengarang, int tahunTerbit) {
7.         this.judul = judul;
8.         this.pengarang = pengarang;
9.         this.tahunTerbit = tahunTerbit;
10.    }
11.
12.    public String getJudul() {
13.        return judul;
14.    }
15.
16.    public void setJudul(String judul) {
17.        this.judul = judul;
18.    }
19.
20.    public String getPengarang() {
21.        return pengarang;
22.    }
23.
24.    public void setPengarang(String pengarang) {
25.        this.pengarang = pengarang;
26.    }
27.
28.    public int getTahunTerbit() {
29.        return tahunTerbit;
30.    }
31.
32.    public void setTahunTerbit(int tahunTerbit) {
33.        this.tahunTerbit = tahunTerbit;
34.    }
35.
36.    public void info() {
37.        System.out.println("Judul: " + judul);
38.        System.out.println("Pengarang: " + pengarang);
39.        System.out.println("Tahun Terbit: " + tahunTerbit);
40.    }
41. }
```

```
1. public class Main {
```

```

2.     public static void main(String[] args) {
3.         Buku buku1 = new Buku("Harry Potter", "J.K. Rowling", 1997);
4.         Buku buku2 = new Buku("The Great Gatsby", "F. Scott Fitzgerald", 1925);
5.
6.         System.out.println("Data Buku 1:");
7.         buku1.info();
8.
9.         System.out.println();
10.
11.        System.out.println("Data Buku 2:");
12.        buku2.info();
13.
14.        System.out.println();
15.
16.        // Latihan 1: Ubah Tahun Terbit Buku 1 menjadi 2001
17.        buku1.setTahunTerbit(2001);
18.        System.out.println("Data Buku 1 (Setelah Diubah Tahun Terbit):");
19.        buku1.info();
20.
21.        System.out.println();
22.
23.        // Latihan 2: Tampilkan Pengarang Buku 2
24.        System.out.println("Pengarang Buku 2: " + buku2.getPengarang());
25.    }
26. }

```

Dalam latihan ini, kita membuat objek buku buku1 dan buku2 berdasarkan class Buku. Kemudian, kita memanggil metode info() pada masing-masing objek untuk menampilkan informasi buku.

Selanjutnya, terdapat dua latihan praktikum yang dapat Anda coba:

1. Mengubah Tahun Terbit Buku 1 menjadi 2001. Dalam latihan ini, Anda akan menggunakan metode setTahunTerbit() pada objek buku1 untuk mengubah tahun terbit buku, kemudian menampilkan kembali informasi buku tersebut menggunakan metode info().
2. Menampilkan Pengarang Buku 2. Dalam latihan ini, Anda akan menggunakan metode getPengarang() pada objek buku2 untuk mengambil nilai pengarang buku tersebut, kemudian mencetaknya menggunakan System.out.println().

Anda dapat menyimpan kedua file kode di atas dengan nama yang sesuai, yaitu "Buku.java" dan "Main.java". Kemudian, Anda dapat mengkompilasi dan menjalankan program tersebut menggunakan perintah-perintah berikut pada command line:

```

javac Buku.java
javac Main.java
java Main

```

Setelah itu, Anda dapat melihat output program yang mencetak informasi buku dan hasil dari latihan-latihan praktikum.

## Tugas

### Tugas Praktikum:

1. Buatlah class "Karyawan" dengan atribut seperti "nip", "nama", dan "jabatan". Sertakan juga metode getter dan setter untuk masing-masing atribut. Selanjutnya, buatlah objek karyawan dan tampilkan informasinya.
2. Buatlah class "Barang" dengan atribut seperti "kode", "namabarang", dan "harga". Sertakan juga metode getter dan setter untuk masing-masing atribut. Selanjutnya, buatlah objek barang dan tampilkan informasinya.
3. Buatlah class "Perpustakaan" dengan atribut seperti "nama", "lokasi", dan "jumlahBuku". Sertakan juga metode getter dan setter untuk masing-masing atribut. Selanjutnya, buatlah objek perpustakaan dan tampilkan informasinya.

### Tugas Teori:

1. Jelaskan apa yang dimaksud dengan class dalam pemrograman berorientasi objek. Berikan contoh sederhana untuk mengilustrasikan penggunaan class.
2. Jelaskan apa yang dimaksud dengan objek dalam pemrograman berorientasi objek. Berikan contoh sederhana untuk mengilustrasikan pembuatan dan penggunaan objek.
3. Jelaskan perbedaan antara class dan objek dalam pemrograman berorientasi objek.
4. Jelaskan konsep enkapsulasi dalam pemrograman berorientasi objek. Mengapa enkapsulasi penting dalam pengembangan perangkat lunak?
5. Jelaskan konsep pewarisan (inheritance) dalam pemrograman berorientasi objek. Berikan contoh sederhana untuk mengilustrasikan pewarisan antara dua class.
6. Pastikan untuk memberikan penjelasan yang jelas dan contoh yang sesuai dalam menjawab setiap pertanyaan pada tugas teori.