Name : Azaria Cindy Sahasika
Number Id : 2341760169 / 06
Class : 1G – Business Information System
Lesson : Algorithm and Data Structure
Material : Jobsheet 2
Github Link : https://github.com/azariacindy/algorithm-ds

JOBSHEET II
OBJECT

1. Tujuan Praktikum

Setelah melakukan materi praktikum ini, mahasiswa mampu:

1. Mengenal objek dan class sebagai konsep mendasar pada pemrograman berorientasi objek
2. Mendeklarasikan class, atribut dan method
3. Membuat objek (instansiasi)
4. Mengakses atribut dan method dari suatu objek
5. Menerapkan konstruktor

2. Praktikum

2.1 Percobaan 1: Deklarasi Class, Atribut dan Method

Waktu Percobaan : 50 Menit

Pada Percobaan 1 ini dilakukan pembuatan class beserta atribut dan method yang dimilikinya. Perhatikan Class Diagram berikut ini:

Buku
judul: String pengarang: String halaman: int stok: int harga: int
tampilInformasi(): void terjual(jml: int): void restock(n: int): void gantiHarga(hrg: int): int

Berdasarkan class diagram tersebut, akan dibuat program mengguanakan bahasa Java.

2.1.1 Langkah-langkah Percobaan

1. Buka text editor. Buat file baru, beri nama **Buku<NoAbsen>.java**
2. Lengkapi class **Buku** dengan atribut yang telah digambarkan di dalam class diagram tersebut

```
String judul, pengarang;
int halaman, stok, harga;
```

3. Lengkapi class **Buku** dengan method yang telah digambarkan di dalam class diagram tersebut

```
void tampilInformasi() {
    System.out.println("Judul: " + judul);
    System.out.println("Pengarang: " + pengarang);
    System.out.println("Jumlah halaman: " + halaman);
    System.out.println("Sisa stok: " + stok);
    System.out.println("Harga: Rp " + harga);
}

void terjual(int jml) {
    stok -= jml;
}

void restock(int jml) {
    stok += jml;
}

void gantiHarga(int hrg) {
    harga = hrg;
}
```

4. Compile dan run program.

```
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> javac book06.java
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> java book06
Error: Main method not found in class book06, please define the main method as:
   public static void main(String[] args)
or a JavaFX application class must extend javafx.application.Application
```

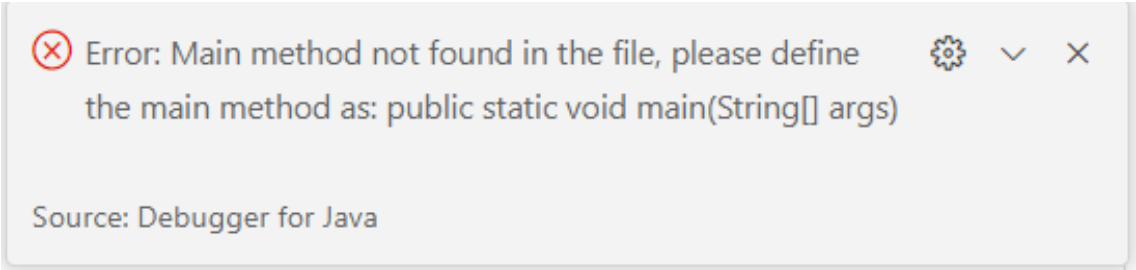
jobsheet2 > code > book06.java > ...

Click here to ask Blackbox to help you code faster | Comment Code |

```java
public class book06 {

    String title, author;
    int page, stock, price;

    void showInformation() {
        System.out.println("Title: " + title);
        System.out.println("Author: " + author);
        System.out.println("Number of pages: " + page);
        System.out.printf(format:"Price: Rp.%d%n", price);
        System.out.println("Stock: " + stock);
    }

    void sold(int amount) {
        if (stock > 0) {
            stock -= amount;
        } else {
            System.out.println(x:"Sorry, the book is out of stock!");
        }
    }

    void restock(int amount) {
        stock += amount;
        System.out.println(x:"Restocked successfully.");
    }

    void priceChange(int prc) {
        price = prc;
        System.out.println(x:"Price changed successfully.");
    }
}
```

2.1.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.



2.1.3 Pertanyaan

1. Sebutkan dua karakteristik class atau object!
   - ➔ Class(blueprint): if this Class has several Objects then just Attributes and Methods in the Object.
   - ➔ Object: this Object contains Attributes and Methods
2. Perhatikan class **Buku** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!
   - title: String
   - author: String
   - page: int
   - stock: int
   - price: int
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
   - showInformation(): void
   - sold(int amount) : void
   - restock(int amount) : void
   - priceChange(int prc): int
4. Perhatikan method **terjual()** yang terdapat di dalam class **Buku**. Modifikasi isi method tersebut sehingga proses pengurangan hanya dapat dilakukan jika stok masih ada (lebih besar dari 0)!

```
void sold(int amount) {
    if (stock > 0) {
        stock -= amount;
    } else {
        System.out.println(x:"Sorry, the book is out of stock!");
    }
}
```

5. Menurut Anda, mengapa method **restock()** mempunyai satu parameter berupa bilangan int?
   - ➔ because for the amount of stock to be added to be an integer.
6. **Commit dan push kode program ke Github**

2.2 Percobaan 2: Instansiasi Object, serta Mengakses Atribut dan Method

Waktu Percobaan: 50 Menit

Sampai tahap ini, class **Buku** telah berhasil dibuat pada Percobaan 1. Selanjutnya, apabila class Buku tersebut ingin digunakan dan diakses atribut serta method-nya, maka perlu dibuat object/instance dari class **Buku** terlebih dahulu melalui proses instansiasi.

2.2.1 Langkah-langkah Percobaan

1. Buat file baru, beri nama **BukuMain<NoAbsen>.java**

2. Tuliskan struktur dasar bahasa pemrograman Java yang terdiri dari fungsi **main()**
3. Di dalam fungsi **main(),** lakukan instansiasi, kemudian lanjutkan dengan mengakses atribut dan method dari objek yang telah terbentuk.

```
Buku bk1 = new Buku();
bk1.judul = "Today Ends Tomorrow Comes";
bk1.pengarang = "Denanda Pratiwi";
bk1.halaman = 198;
bk1.stok = 13;
bk1.harga = 71000;

bk1.tampilInformasi();
bk1.terjual(jml:5);
bk1.gantiHarga(hrg:60000);
bk1.tampilInformasi();
```

4. Compile dan run program.

```
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> java bookMain06
Title: Today Ends Tomorrow Comes
Author: Denanda Pratiwi
Number of pages: 198
Price: Rp.71000
Stock: 13
Price changed successfully.
Title: Today Ends Tomorrow Comes
Author: Denanda Pratiwi
Number of pages: 198
Price: Rp.60000
Stock: 8
```

jobsheet2 > code > bookMain06.java > bookMain06 > main(String[])

```
Click here to ask Blackbox to help you code faster | Comment Code |
1   public class bookMain06 {
        Run | Debug
2       public static void main(String[] args) {
3           book06 bk1 = new book06();
4           bk1.title = "Today Ends Tomorrow Comes";
5           bk1.author = "Denanda Pratiwi";
6           bk1.page = 198;
7           bk1.stock = 13;
8           bk1.price = 71000;
9
10          bk1.showInformation();
11          bk1.sold(amount:5);
12          bk1.priceChange(prc:60000);
13          bk1.showInformation();
14      }
15  }
```

5. **Commit dan push kode program ke Github**

2.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
```

2.2.3 Pertanyaan

1. Pada class **BukuMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?
   - ➔ Object: bk1
   - ➔ Instantiation: `book06 bk1 = new book06();`
2. Bagaimana cara mengakses atribut dan method dari suatu objek?
   - ➔ Using notation ('.')
   - ➔ namaObjet.namaMethod();
   - ➔
```
bk1.showInformation();
bk1.sold(amount:5);
bk1.priceChange(prc:60000);
bk1.showInformation();
```
3. Mengapa hasil output pemanggilan method **tampilInformasi()** pertama dan kedua berbeda?
   - ➔ showInformation() includes references to attributes that can be changed by other methods or outside the method itself, changes in attribute values between two invocations may cause differences in output results.

2.3 Percobaan 3: Membuat Konstruktor

Waktu Percobaan: 60 Menit

Pada percobaan ini, dilakukan pembuatan kode program untuk mengimplementasikan berbagai macam konstruktor berdasarkan parameternya.

2.3.1 Langkah-langkah Percobaan

1. Buka kembali class **Buku**. Tambahkan dua buah konstruktor di dalam class **Buku** tersebut, yang terdiri dari satu konstruktor default dan satu konstruktor berparameter. Konstruktor merupakan method Istimewa, penempatan kode program untuk konstruktor dapat diperlakukan sama seperti method yang lain (setelah atribut).

```
public Buku() {

}

public Buku(String jud, String pg, int hal, int stok, int har) {
    judul = jud;
    pengarang = pg;
    halaman = hal;
    this.stok = stok;
    harga = har;
}
```

*Catatan: Apabila nama parameter sama dengan nama atribut, maka untuk merujuk pada variabel atribut ditambahkan sintaks **this** di depan nama **atribut***

2. Buka kembali class **BukuMain**. Buat sebuah object lagi bernama **bk2** dengan menggunakan konstruktor berparameter.

```
Buku bk1 = new Buku();
bk1.judul = "Today Ends Tomorrow Comes";
bk1.pengarang = "Denanda Pratiwi";
bk1.halaman = 198;
bk1.stok = 13;
bk1.harga = 71000;

bk1.tampilInformasi();
bk1.terjual(jml:5);
bk1.gantiHarga(hrg:60000);
bk1.tampilInformasi();
```

```
Buku bk2 = new Buku(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);
bk2.terjual(jml:11);
bk2.tampilInformasi();
```

3. Compile dan run program.

```
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> java bookMain06
==========================
Title: Today Ends Tomorrow Comes
Author: Denanda Pratiwi
Number of pages: 198
Stock: 13
Price: Rp.71000
==========================
Price changed successfully.
==========================
Title: Today Ends Tomorrow Comes
Author: Denanda Pratiwi
Number of pages: 198
Stock: 8
Price: Rp.60000
==========================
==========================
Title: Self Reward
Author: Mahera Ayesha
Number of pages: 160
Stock: 18
Price: Rp.59000
==========================
```

4. **Commit dan push kode program ke Github**

2.3.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

```
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
Judul: Self Reward
Pengarang: Maheera Ayesha
Jumlah halaman: 160
Sisa stok: 18
Harga: Rp 59000
```

2.3.3 Pertanyaan

1. Pada class **Buku** di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!

```
public book06(String tit, String aut, int pg, int stk, int pric) {
    title = tit;
    author = aut;
    page = pg;
    stock = stk;
    price = pric;
}
```

2. Perhatikan class **BukuMain**. Apa sebenarnya yang dilakukan pada baris program berikut?

```
Buku bk2 = new Buku(jud:"Self Reward", pg:"Maheera Ayesha", hal:160, stok:29, har:59000);
```

➔ instantiate new objects using parameterized constructors.

3. Hapus konstruktor default pada class **Buku**, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!

➔ the result will be an error because bk2 is undefined, because the constructor in book06 corresponds to bookMain06 in bk2.

```
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> javac bookMain06.java
bookMain06.java:15: error: constructor book06 in class book06 cannot be applied to given types;
        book06 bk2 = new book06("Self Reward", "Mahera Ayesha", 160, 29, 59000);
                     ^
  required: no arguments
  found:    String,String,int,int,int
  reason: actual and formal argument lists differ in length
1 error
```

4. Setelah melakukan instansiasi object, apakah method di dalam class **Buku** harus diakses secara berurutan? Jelaskan alasannya!

➔ No, because the methods in the Book class don't have to be accessed sequentially after instantiating the object. Objects can call methods in the Book class in any order they want, as long as they have the right to do so.

5. Buat object baru dengan nama **buku<NamaMahasiswa>** menggunakan konstruktor berparameter dari class **Buku**!

```
jobsheet2 > code > J bookCindy.java > ...
      Click here to ask Blackbox to help you code faster | Comment Code |
1   public class bookCindy {
        Run | Debug
2       public static void main(String[] args) {
3           book06 cindy = new book06(tit:"Wonwoo's", aut:"Azaria Cindy Sahasika", pg:50, stk:10, pric:132000);
4           cindy.sold(amount:6);
5           cindy.showInformation();
6       }
7   }
```

```
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> java bookCindy
============================
Title: Wonwoo's
Author: Azaria Cindy Sahasika
Number of pages: 50
Stock: 4
Price: Rp.132000
============================
```

6. **Commit dan push kode program ke Github**

2.4 Latihan Praktikum

Waktu : 150 Menit

1. Pada class Buku yang telah dibuat, tambahkan tiga method yaitu **hitungHargaTotal()**, **hitungDiskon()**, dan **hitungHargaBayar()** dengan penjelasan sebagai berikut:

- Method **hitungHargaTotal()** digunakan untuk menghitung harga total yang merupakan perkalian antara harga dengan jumlah buku yang terjual
- Method **hitungDiskon()** digunakan untuk menghitung diskon dengan aturan berikut:
  - Jika harga total lebih dari 150000, maka harga didiskon sebesar 12%
  - Jika harga total antara 75000 sampai 150000, maka harga didiskon sebesar 5%
  - Jika harga total kurang dari 75000, maka harga tidak didiskon
- Method **hitungHargaBayar()** digunakan untuk menghitung harga total setelah dikurangi diskon

Class diagram **Buku** setelah penambahan ketiga method tersebut adalah sebagai berikut.

Buku
judul: String pengarang: String halaman: int stok: int harga: int
tampilInformasi(): void terjual(jml: int): void restock(n: int): void gantiHarga(hrg: int): int **hitungHargaTotal(): int** **hitungDiskon(): int** **hitungHargaBayar(): int**

ANSWER:

```
// Method to calculate total price
double calculateTotalPrice(double quantity) {
    return price * quantity;
}

// Method to calculate discount
double calculateDiscount(double totalPrice) {
    if (totalPrice > 150000) {
        return 0.12; // 12% discount
    } else if (totalPrice >= 75000 && totalPrice <= 150000) {
        return 0.05; // 5% discount
    } else {
        return 0; // No discount
    }
}

// Method to calculate final price after discount
double calculateFinalPrice(double quantity) {
    double totalPrice = calculateTotalPrice(quantity);
    double discountRate = calculateDiscount(totalPrice);
    double discountAmount = totalPrice * discountRate;
    return totalPrice - discountAmount;
}
```

2. Buat program berdasarkan class diagram berikut ini!

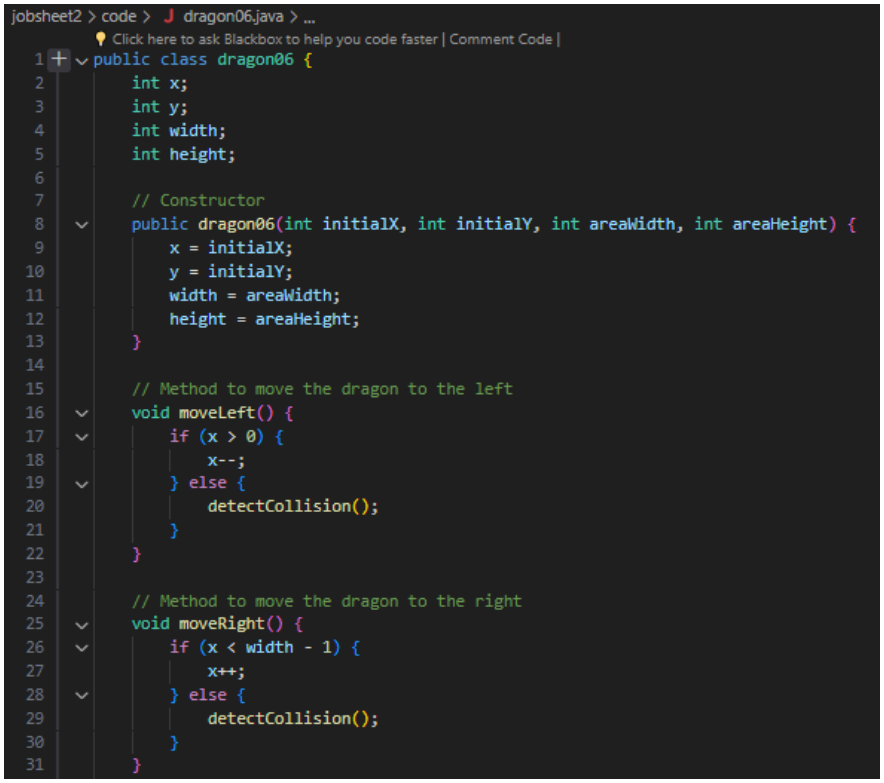
Dragon
x: int y: int width: int height: int
moveLeft(): void moveRight(): void moveUp(): void moveDown(): void printPosition(): void detectCollision(x: int, y: int): void

Penjelasan dari atribut dan method pada class Dragon tersebut adalah sebagai berikut:

- Atribut **x** digunakan untuk menyimpan posisi koordinat x (mendatar) dari dragon, sedangkan atribut **y** untuk posisi koordinat y (vertikal)
- Atribut **width** digunakan untuk menyimpan lebar dari area permainan, sedangkan **height** untuk menyimpan panjang area
- Method **moveLeft()** digunakan untuk mengubah posisi dragon ke kiri (koordinat x akan berkurang 1), sedangkan **moveRight()** untuk bergerak ke kanan (koordinat x akan bertambah 1). Perlu diperhatikan bahwa koordinat x tidak boleh lebih kecil dari 0 atau lebih besar dari nilai width. Jika koordinat x < 0 atau x > width maka panggil method **detectCollision()**
- Method moveUp() digunakan untuk mengubah posisi dragon ke atas (koordinat y akan berkurang 1), sedangkan moveDown() untuk bergerak ke bawah (koordinat y akan bertambah 1). Perlu diperhatikan bahwa koordinat y tidak boleh lebih kecil dari 0 atau lebih besar dari nilai height. Jika koordinat y < 0 atau y > height maka panggil method **detectCollision()**
- Method **detectCollision()** akan mencetak pesan “Game Over” apabila dragon menyentuh ujung area permainan.

ANSWER:

```
PS D:\cooleyah\smstr2\algorithm and data structure\assignment\jobsheet2\code> java dragon06
Dragon position: (1, 3)
Dragon position: (1, 2)
Dragon position: (2, 2)
Dragon position: (2, 3)
```



```java
public class dragon06 {
    int x;
    int y;
    int width;
    int height;

    // Constructor
    public dragon06(int initialX, int initialY, int areaWidth, int areaHeight) {
        x = initialX;
        y = initialY;
        width = areaWidth;
        height = areaHeight;
    }

    // Method to move the dragon to the left
    void moveLeft() {
        if (x > 0) {
            x--;
        } else {
            detectCollision();
        }
    }

    // Method to move the dragon to the right
    void moveRight() {
        if (x < width - 1) {
            x++;
        } else {
            detectCollision();
        }
    }
```

```java
    // Method to move the dragon up
    void moveUp() {
        if (y > 0) {
            y--;
        } else {
            detectCollision();
        }
    }

    // Method to move the dragon down
    void moveDown() {
        if (y < height - 1) {
            y++;
        } else {
            detectCollision();
        }
    }

    // Method to print the current position of the dragon
    void printPosition() {
        System.out.println("Dragon position: (" + x + ", " + y + ")");
    }

    // Method to detect collision and print "Game Over" message
    void detectCollision() {
        System.out.println(x:"Game Over");
    }

    Run | Debug
    public static void main(String[] args) {
        // Example Usage
        dragon06 dragon = new dragon06(initialX:2, initialY:3, areaWidth:5, areaHeight:6);

        // Move the dragon around
        dragon.moveLeft();
        dragon.printPosition();

        dragon.moveUp();
        dragon.printPosition();

        dragon.moveRight();
        dragon.printPosition();

        dragon.moveDown();
        dragon.printPosition();
```