

PRAKTIKUM ALGORITMA DAN STRUKTUR DATA
JOBSHEET 2 PERTEMUAN KE-2



NAMA : DIMAS ADI BAYU SAMUDRA

KELAS : 1A

NO. ABSEN : 8

NIM : 2341720169

PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNOLOGI INFORMASI
POLITEKNIK NEGERI MALANG
2024

PRAKTIKUM 1

Code :

```
public class Buku08 {
    String judul, pengarang;
    int halaman, stock, harga;

    void tampilInformasi () {
        System.out.println("Judul : " + judul);
        System.out.println("Pengarang : " + pengarang);
        System.out.println("Halaman : " + halaman);
        System.out.println("Stock : " + stock);
        System.out.println("Harga : " + harga);
    }

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

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

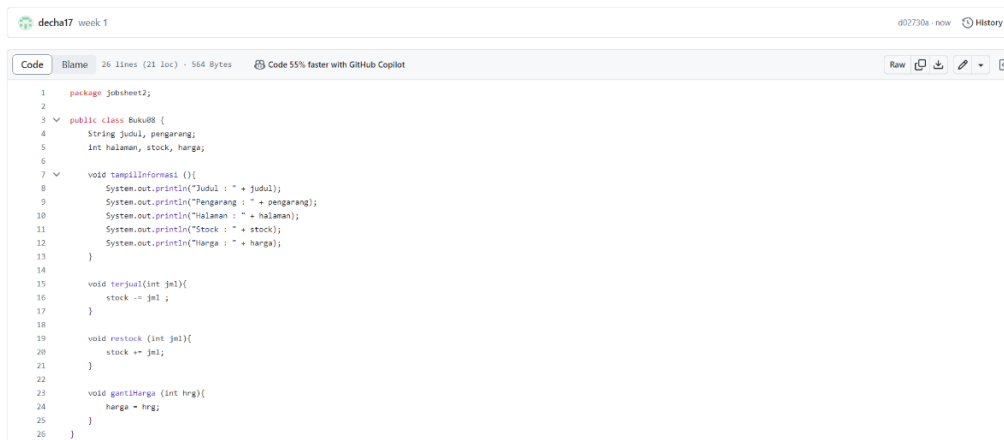
    void gantiHarga (int hrg) {
        harga = hrg;
    }
}
```

Hasil :

**Error: Could not find or load main class BukuMain08
Caused by: java.lang.ClassNotFoundException: BukuMain08**

Github :

<https://github.com/decha17>



The screenshot shows a GitHub repository page for a user named 'decha17'. The repository is titled 'week 1'. The file 'Buku08.java' is selected, showing its source code. The code is a Java class named 'Buku08' with attributes 'judul', 'pengarang', 'halaman', 'stock', and 'harga'. It includes methods for displaying information, handling sales, restocking, and changing prices. The code is formatted with syntax highlighting and line numbers. The repository has 26 lines of code, 21 loc, and 564 bytes. A note indicates that the code is 55% faster with GitHub Copilot.

Pertanyaan :

1. Sebutkan dua karakteristik class atau object!
2. Perhatikan class Buku pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
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)!
5. Menurut Anda, mengapa method restock() mempunyai satu parameter berupa bilangan int?
6. Commit dan push kode program ke Github

Jawaban :

1. Class masih berupa rancangan yang bersifat umum sedangkan objek merupakan penggambaran dari objek yang ada di dunia nyata dan sudah berbentuk class. Objek bersifat lebih spesifik.

2. Ada 5 atribut yang dimiliki oleh class buku, diantaranya judul dan pengarang berupa string lalu halaman, stok, harga berupa int .

3. Method yang ada pada class tersebut ada 4, yaitu

tampilInformasi(): void

terjual(jml: int): void

restock(n: int): void

gantiHarga(hrg: int): int

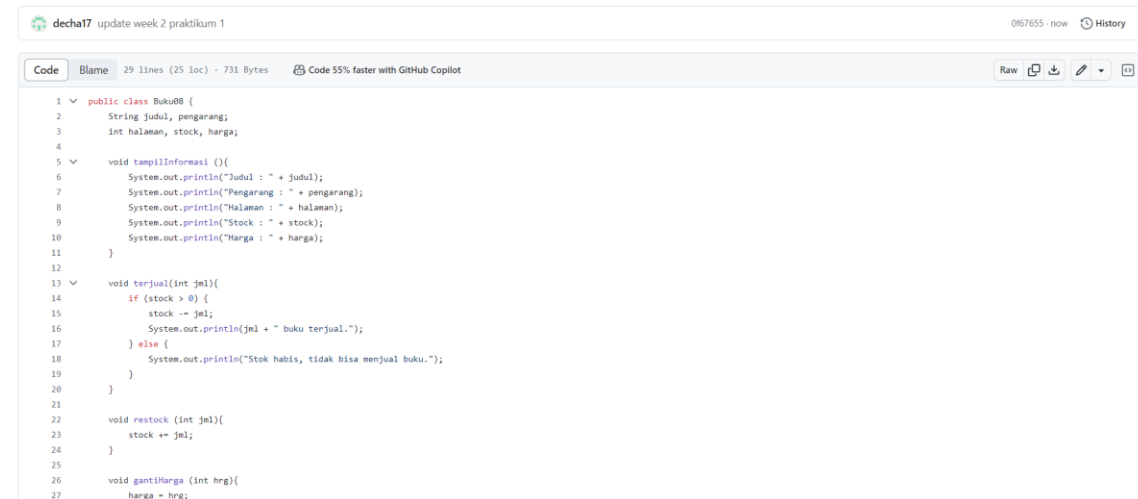
4.

```
void terjual(int jml){
    if (stock > 0) {
        stock -= jml;
        System.out.println(jml + " buku terjual.");
    } else {
        System.out.println("Stok habis, tidak bisa menjual buku.");
    }
}
```

```
Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 0
Harga : 71000
Stok habis, tidak bisa menjual buku.
Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 0
Harga : 60000
```

5. Menurut saya Method restock harus memiliki parameter berupa int dikarenakan ada nilai yang harus di ambil yaitu variabel jml agar bisa melakukan perhitungan pada method restock yang di mana membutuhkan variabel jml.

6. Github :



```
1 public class Buku08 {
2     String judul, pengarang;
3     int halaman, stock, harga;
4
5     void tampilInformasi(){
6         System.out.println("Judul : " + judul);
7         System.out.println("Pengarang : " + pengarang);
8         System.out.println("Halaman : " + halaman);
9         System.out.println("Stock : " + stock);
10        System.out.println("Harga : " + harga);
11    }
12
13    void terjual(int jml){
14        if (stock > 0) {
15            stock -= jml;
16            System.out.println(jml + " buku terjual.");
17        } else {
18            System.out.println("Stok habis, tidak bisa menjual buku.");
19        }
20    }
21
22    void restock (int jml){
23        stock += jml;
24    }
25
26    void gantiHarga (int hrg){
27        harga = hrg;
28    }
29 }
```

PRATIUM 2

Code :

```
public class BukuMain08 {
    public static void main(String[] args) {
        Buku08 bk1 = new Buku08();
        bk1.judul = "Today Ends Tomorrow Comes";
        bk1.pengarang = "Denanda Pratiwi";
        bk1.halaman = 198;
        bk1.stock = 13;
        bk1.harga = 71000;

        bk1.tampilInformasi();
        bk1.terjual(5);
        bk1.gantiHarga(60000);
        bk1.tampilInformasi();
    }
}
```

Hasil :

```
Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 13
Harga : 71000
Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 8
Harga : 60000
```

Pertanyaan :

1. Pada class BukuMain, tunjukkan baris kode program yang digunakan untuk proses instansiasi!

Apa nama object yang dihasilkan?

2. Bagaimana cara mengakses atribut dan method dari suatu objek?

3. Mengapa hasil output pemanggilan method tampilInformasi() pertama dan kedua berbeda?

Jawaban :

1. Nama objek yang dihasilkan yaitu bk1

```
Buku08 bk1 = new Buku08();
```

2. Menggunakan tanda titik “.” Sebagai penghubung antara namaObjek dengan namaAtribut/Method. Format penulisan seperti dibawah ini

Cara mengakses atribut :

```
namaObject.namaAtribut = nilai;
```

Cara mengakses Method

```
namaObjek.namaMethod = nilai ;
```

3. Dikarenakan adanya perubahan pada stock dan juga harga, pada tampilInformasi pertama menunjukkan stock awal dan harga yang belum dirubah lalu di tampilInformasi kedua terdapat pergantian stock dan juga pergantian harga sehingga terjadi perubahan output.

PRAKTIUM 3**Code 1:**

```
public class Buku08 {
    String judul, pengarang;
    int halaman, stock, harga;

    void tampilInformasi () {
        System.out.println("Judul : " + judul);
        System.out.println("Pengarang : " + pengarang);
        System.out.println("Halaman : " + halaman);
        System.out.println("Stock : " + stock);
        System.out.println("Harga : " + harga);
    }

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

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

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

```

    public Buku08 () {
    }
    public Buku08 (String jud, String pg, int hal, int stock, int har){
        judul = jud;
        pengarang = pg;
        halaman = hal;
        this.stock = stock;
        harga = har;
    }
}

```

Code 2:

```

public class BukuMain08 {
    public static void main(String[] args) {
        Buku08 bk1 = new Buku08();
        bk1.judul = "Today Ends Tomorrow Comes";
        bk1.pengarang = "Denanda Pratiwi";
        bk1.halaman = 198;
        bk1.stock = 13;
        bk1.harga = 71000;

        bk1.tampilInformasi();
        bk1.terjual(5);
        bk1.gantiHarga(60000);
        bk1.tampilInformasi();

        Buku08 bk2 = new Buku08("Self reward ", "Maheera Ayesha", 160, 29,
59000);
        bk2.terjual(11);
        bk2.tampilInformasi();

    }
}

```

Hasil :

```

Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 13
Harga : 71000
Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 8
Harga : 60000
Judul : Self reward
Pengarang : Maheera Ayesha
Halaman : 160
Stock : 18
Harga : 59000

```

Pertanyaan :

1. Pada class Buku di Percobaan 3, tunjukkan baris kode program yang digunakan untuk mendeklarasikan konstruktor berparameter!
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);
```
3. Hapus konstruktor default pada class Buku, kemudian compile dan run program. Bagaimana hasilnya? Jelaskan mengapa hasilnya demikian!
4. Setelah melakukan instansiasi object, apakah method di dalam class Buku harus diakses secara berurutan? Jelaskan alasannya!
5. Buat object baru dengan nama buku<NamaMahasiswa> menggunakan konstruktor berparameter dari class Buku!
6. Commit dan push kode program ke Github

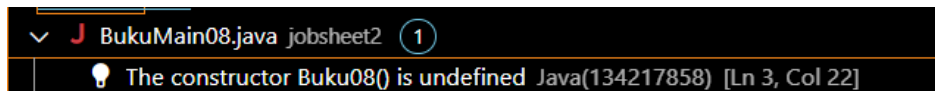
Jawaban :

1.

```
public Buku08 (String jud, String pg, int hal, int stock, int har){  
    judul = jud;  
    pengarang = pg;  
    halaman = hal;  
    this.stock = stock;  
    harga = har;  
}
```

2. Memberikan nilai pada variabel yang menjadi parameter pada method public Buku08()

3. Program tidak bisa di jalankan , dikarenakan konstruktor default tidak ada dan tidak di definisikan pada class Buku. Selain itu juga kita juga menggunakan konstruktor tanpa parameter pada Buku08, didalam kode kita yang berada di tempat lain, tetapi tidak ada konstruktor tanpa parameter yang didefinisikan dalam kelas Buku08.





4. Tidak, method di dalam class Buku08 tidak harus diakses secara berurutan setelah melakukan instansiasi objek. Dalam pemrograman berorientasi objek, urutan pemanggilan method pada suatu objek tidak memiliki keterkaitan yang kaku dengan urutan deklarasi method di dalam class tersebut.

5.

```
public Buku08 (String jud, String pg, int hal, int stock, int har){  
    judul = jud;  
    pengarang = pg;  
    halaman = hal;  
    this.stock = stock;  
}
```

```
        harga = har;
    }
```

6.

 Buku08.java	Update Buku08.java	now
 BukuMain08.java	week 1 tambahan	2 days ago

Praktikum 4

Latihan Praktikum Tugas 1

Code 1

```
public class Tugas1_08 {

    int terjual = 0;
    String judul, pengarang;
    int halaman, stock, harga;
    int hargaTotal ;
    double diskon ;
    int hrgaBayar;

    void tampilInformasi (){
        System.out.println("Judul : " + judul);
        System.out.println("Pengarang : " + pengarang);
        System.out.println("Halaman : " + halaman);
        System.out.println("Stock : " + stock);
        System.out.println("Harga : " + harga);
    }

    int terjual(int jml){
        stock -= jml ;
        return jml;
    }

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

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

    int hitungHargaTotal(int jml){
        hargaTotal = harga * jml;
        System.out.println("Harga total: "+ hargaTotal);
        return hargaTotal;
    }

    int hitungDiskon() {
        if (hargaTotal > 150000) {
            diskon = hargaTotal * 0.88;
        } else if (hargaTotal >= 75000 || hargaTotal <= 150000){
            diskon = hargaTotal * 0.95;
        } else if(hargaTotal < 75000){
            System.out.println("Tidak ada diskon ");
        }
    }
}
```



```

        }
        System.out.println("Hitung harga setelah di diskon : " +
diskon);
        return (int) diskon;
    }
    int hargaBayar(){
        hrgaBayar = (int) diskon;
        System.out.println("Harga yang harus di bayar : " + hrgaBayar);
        return hrgaBayar;
    }
}

```

Code 2:

```

public class Tugas1_08_main {
    public static void main(String[] args) {
        Tugas1_08 bk1 = new Tugas1_08();
        bk1.judul = "Today Ends Tomorrow Comes";
        bk1.pengarang = "Denanda Pratiwi";
        bk1.halaman = 198;
        bk1.stock = 13;
        bk1.harga = 71000;

        bk1.tampilInformasi();
        bk1.terjual(5);
        bk1.gantiHarga(60000);
        bk1.tampilInformasi();
        bk1.hitungHargaTotal(5);
        bk1.hitungDiskon();
        bk1.hargaBayar();

    }
}

```

Hasil run :

```

Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 13
Harga : 71000
Judul : Today Ends Tomorrow Comes
Pengarang : Denanda Pratiwi
Halaman : 198
Stock : 8
Harga : 60000
Harga total: 300000
Hitung harga setelah di diskon : 264000.0
Harga yang harus di bayar : 264000

```

Latihan Praktikum tugas 2:

Code 1 :

```
public class Tugas2_Dragon {
    int x, y, height , width;

    public Tugas2_Dragon(int i, int j, int k, int l) {
        x = i;
        y = j;
        height = k;
        width = l;
    }

    void moveLeft(){
        if (x > 0 || x < width) {
            x--;
        }
    }

    void moveRight(){
        if (x > 0 || x < width) {
            x++;
        }
    }

    void moveUp(){
        if (y > 0 || x < height) {
            y--;
        }
    }

    void moveDown(){
        if (y > 0 || x < height) {
            y++;
        }
    }

    void printPosition(){
        System.out.println("Posisi Dragon (x, y) (" + x + ", " + y +
        ")");
        if (x == 0 || y == 0) {
            detectCollision(x, y);
        }
    }

    void detectCollision(int x, int y){
        System.out.println("Game Over ");
    }
}
```

```
public class Tugas2_DragonMain {  
    public static void main(String[] args) {  
        Tugas2_Dragon dragon = new Tugas2_Dragon(3,5,10,10);  
        dragon.printPosition();  
        dragon.moveUp();  
        dragon.printPosition();  
        dragon.moveLeft();  
        dragon.printPosition();  
        dragon.moveRight();  
        dragon.printPosition();  
        dragon.moveUp();  
        dragon.printPosition();  
        dragon.moveUp();  
        dragon.printPosition();  
        dragon.moveLeft();  
        dragon.printPosition();  
        dragon.moveUp();  
        dragon.printPosition();  
        dragon.moveUp();  
        dragon.printPosition();  
    }  
}
```

Hasil :

```
Posisi Dragon (x, y) (3, 5)  
Posisi Dragon (x, y) (3, 4)  
Posisi Dragon (x, y) (2, 4)  
Posisi Dragon (x, y) (3, 4)  
Posisi Dragon (x, y) (3, 3)  
Posisi Dragon (x, y) (3, 2)  
Posisi Dragon (x, y) (2, 2)  
Posisi Dragon (x, y) (2, 1)  
Posisi Dragon (x, y) (2, 0)  
Game Over
```