

**LAPORAN HASIL PRAKTIKUM  
ALGORITMA DAN STRUKTUR DATA  
JOBSHEET 2**



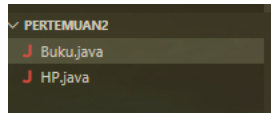
**Oleh:  
DZULFIKAR MUHAMMAD AL GHIFARI  
NIM. 2341760071  
SIB-1F / 08  
D-IV SISTEM INFORMASI BISNIS  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

## PRAKTIKUM 2

### PERCOBAAN 2.1

#### LANGKAH LANGKAH PERCOBAAN 2.1.1

1. Membuat file dan class Buku



2. Deklarasi variable

```
public class Buku {  
    String judul, pengarang;  
    int halaman, stok, harga;  
}
```

3. Melengkapi method

```
Codeium: Explain  
public class Buku {  
    String judul, pengarang;  
    int halaman, stok, harga;  
  
    public Buku(String judul, String pengarang, int halaman, int stok, int harga) {  
        this.judul = judul;  
        this.pengarang = pengarang;  
        this.halaman = halaman;  
        this.stok = stok;  
        this.harga = harga;  
    }  
  
    Codeium: Refactor | Explain | Generate Javadoc | X  
    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);  
    }  
  
    Codeium: Refactor | Explain | Generate Javadoc | X  
    void terjual(int jml) {  
        stok -= jml;  
    }  
  
    Codeium: Refactor | Explain | Generate Javadoc | X  
    void restock(int jml) {  
        stok += jml;  
    }  
  
    Codeium: Refactor | Explain | Generate Javadoc | X  
    void gantiHarga(int hrg) {  
        harga = hrg;  
    }  
}
```

#### VERIFIKASI HASIL PERCOBAAN 2.1.2

The file 'Buku.java' is not executable, please select a main class you want to run.

#### PERTANYAAN 2.1.3

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?

Jawaban

1. Memiliki sesuatu dan melakukan sesuatu
2. Ada 5 attribut
  - Judul
  - Pengarang
  - Halaman
  - Stok
  - Harga
3. Ada 4 method
  - tampilInformasi
  - terjual
  - restock
  - gantiHarga

4.

```
void terjual(int jml) {  
    if(stok > 0){  
        stok -= jml;  
    }else{  
        System.out.println(x:"Stok tidak mencukupi");  
    }  
}
```

5. Karena untuk menampung data jumlah barang yang akan masuk. jadi diperlukan satu parameter bertipe int

## LANGKAH LANGKAH PERCOBAAN 2.2.1

1. Buat file

```
J Buku.java  
J BukuMain08.java
```

2. Menuliskan struktur

```
public class BukuMain08 {  
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X  
    public static void main(String[] args) {}  
}
```

3. Instansiasi

```
public class BukuMain08 {  
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | X  
    public static void main(String[] args) {}  
    Buku bk1 = new Buku(judul:null, pengarang:null, halaman:0, stok:0, harga:0);  
    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();  
}
```

## VERIFIKASI HASIL PERCOBAAN 2.2.2

```
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
```

### PERTANYAAN 2.2.3

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. `Buku bk1 = new Buku();`  
Object yang dihasilkan adalah **bk1**
2. Dengan memanggil nama object kemudian diberi titik dan nama attr atau method yang di inginkan  

```
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();
```
3. Untuk yang pertama tampilInformasi menampilkan data yang di instansiasi tanpa merubah sedikit pun data.  
Untuk yang kedua terdapat perubahan data dengan dipanggilnya method terjual dan gantiHarga yang menyebabkan perubahan data.

### LANGKAH LANGKAH PERCOBAAN 2.3.1

1. Menambahkan konstruktor

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

2. Menambahkan object baru

```
Buku bk2 = new Buku(judul:"Self Reward", pengarang:"Mahera Ayesi", halaman:160, stok:20, harga:59000);  
bk2.terjual(jml:11);  
bk2.tampilInformasi();
```

### VERIFIKASI HASIL PERCOBAAN 2.2.2

```
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: Mahera Ayesi  
Jumlah halaman: 160  
Sisa stok: 18  
Harga: Rp 59000
```

### PERTANYAAN 2.3.3

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!

Jawaban

1.

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

2. Membuat object baru dengan parameter judul, pengarang, hal, stok, harga dalam satu baris. Hal ini bisa dilakukan karena sudah terdapat konstruktor berparameter.
3. Hasilnya akan error, karena object bk 1 tidak memiliki parameter, sehingga harus di tambahkan parameter terlebih dahulu seperti object bk2
4. Method bebas untuk pengaksesannya, karena urutan run method bergantung pada class bukuMain. Kode yang paling atas akan dijalankan terlebih dahulu
- 5.

```
Buku bukuDzulfikar = new Buku(judul:"Dzulfikar orang baik", pengarang:"dzulfikar", halaman:3, stok:50, harga:10000);  
bukuDzulfikar.tampilInformasi();  
bukuDzulfikar.gantiHarga(hrg:20000);  
bukuDzulfikar.tampilInformasi();  
  
Judul: Dzulfikar orang baik  
Pengarang: dzulfikar  
Jumlah halaman: 3  
Sisa stok: 50  
Harga: Rp 10000  
Judul: Dzulfikar orang baik  
Pengarang: dzulfikar  
Jumlah halaman: 3  
Sisa stok: 50  
Harga: Rp 20000
```

## 2.4 LATIHAN PRAKTIKUM

A. Menambahkan 3 method pada class buku

```
Codeium: Refactor | Explain | Generate Javadoc | X
int hitungHargaTotal(int terjual){
    int hargaTotal = harga * terjual;
    System.out.println("Harga Total : " + hargaTotal);
    return hargaTotal;
}

Codeium: Refactor | Explain | Generate Javadoc | X
int hitungDiskon(int hargaTotal){
    int diskon = 0;
    if(hargaTotal > 150000){
        diskon = 12;
        System.out.println("Anda mendapatkan diskon sebesar "+ diskon);
    }else if(hargaTotal >= 75000 ){
        diskon = 5;
        System.out.println("Anda mendapatkan diskon sebesar "+ diskon);
    }else{
        System.out.println(x:"Anda tidak mendapatkan diskon");
    }
    return diskon;
}

Codeium: Refactor | Explain | Generate Javadoc | X
void hitungHargaBayar(int hargaTotal, int diskon){
    int hitungBayar = 0, potongan = 0;
    if(diskon > 0){
        potongan = hargaTotal*diskon/100;
        hitungBayar = hargaTotal - potongan;
    }else{
        hitungBayar = hargaTotal;
    }
    System.out.println("Total yang harus anda bayar "+hitungBayar);
}

Judul: Dzulfikar orang baik
Pengarang: dzulfikar
Jumlah halaman: 3
Sisa stok: 50
Harga: Rp 10000
Harga Total : 80000.0
Anda mendapatkan diskon sebesar 5
Total yang harus anda bayar 76000.0
PE-5-12-POLTEKNIK12560000-2023-2024) Dan seterusnya
```

### B. Program koordinat



```

Codeium: Explain
public class Dragon08 {
    int x, y, width, height;

    public Dragon08(int x, int y, int width, int height) {
        this.x = x;
        this.y = y;
        this.width = width;
        this.height = height;
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    void moveLeft(){
        if(detectX()){
            x--;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    void moveRight(){
        if(detectX()){
            x++;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    void moveUp(){
        if(detectY()){
            y--;
        }
    }

    Codeium: Refactor | Explain | Generate Javadoc | ×
    void moveDown(){
        if(detectY()){
            y++;
        }
    }
}

```

```

Codeium: Refactor | Explain | Generate Javadoc | ×
boolean detectX(){
    boolean status = false;
    if(x<0 || x>width){
        detectCollision();
    }else{
        status = true;
    }
    return status;
}

Codeium: Refactor | Explain | Generate Javadoc | ×
boolean detectY(){
    boolean status = false;
    if(y<0 || y>height){
        detectCollision();
    }else{
        status = true;
    }
    return status;
}

Codeium: Refactor | Explain | Generate Javadoc | ×
void detectCollision(){
    System.out.println(x: "Game Over");
}

```

```

Codeium: Refactor | Codeium: Explain
public class MainDragon08 {
    Run | Debug | Codeium: Refactor | Explain | Generate Javadoc | ×
    public static void main(String[] args) {
        Dragon08 dragon = new Dragon08(x:5, y:5, width:6, height:6);
        dragon.moveRight();
        dragon.moveRight();
        dragon.moveRight();
    }
}

```

3e493fc5e6ad  
Game Over