



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 menggunakan 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;
}

```

```

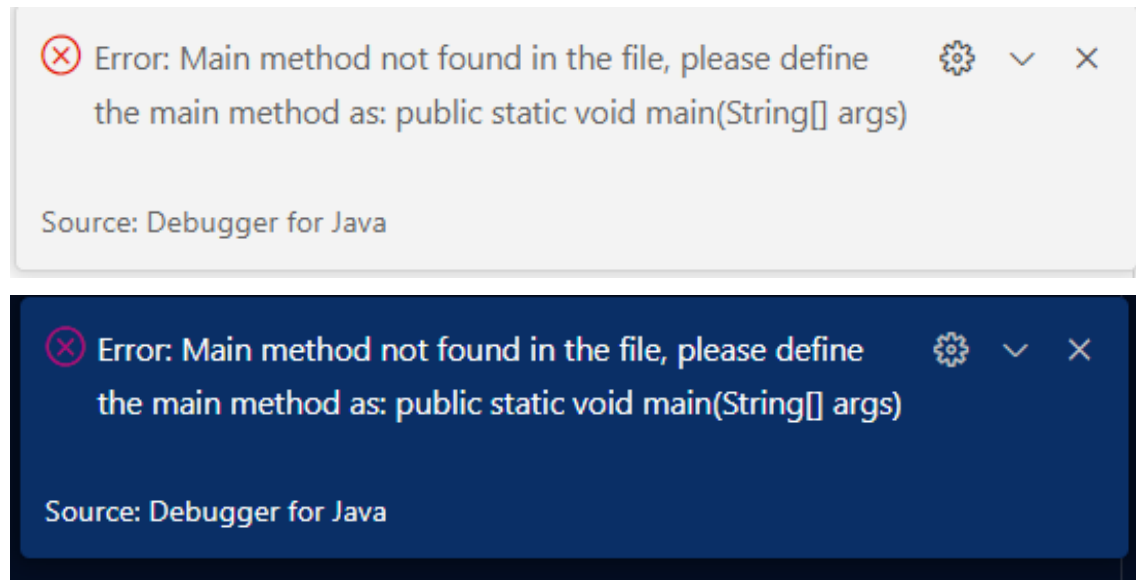
1 public class Buku24 {
2     String judul, pengarang;
3     int halaman, stok, harga;
4
5     void tampilInformasi() {
6         System.out.println("Judul: " + judul);
7         System.out.println("Pengarang: " + pengarang);
8         System.out.println("Jumlah halaman: " + halaman);
9         System.out.println("Sisa stok: " + stok);
10        System.out.println("Harga: Rp. " + harga);
11    }
12
13    void terjual(int jml) {
14        if (stok > 0) {
15            stok -= jml;
16        } else {
17            System.out.println("Stok kosong");
18        }
19    }
20
21    void restock(int jml) {
22        stok += jml;
23    }
24
25    void gantiHarga(int hrg) {
26        harga = hrg;
27    }
28 }
29

```

4. Compile dan run program.

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!
Jawab: objek lebih spesifik daripada class, class lebih umum daripada objek
2. Perhatikan class **Buku** pada Praktikum 1 tersebut, ada berapa atribut yang dimiliki oleh class Buku? Sebutkan apa saja atributnya!
Jawab: ada 5 atribut yaitu, judul;pengarang;jumlah halaman;sis stok;harga
3. Ada berapa method yang dimiliki oleh class tersebut? Sebutkan apa saja methodnya!
Jawab: ada 4 method yaitu, tampilInformasi;terjual;restock;gantiHarga
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)!

```

1 void terjual(int jml) {
2     if (stok > 0) {
3         stok -= jml;
4     } else {
5         System.out.println("Stok kosong");
6     }
7 }

```

Jawab:

5. Menurut Anda, mengapa method **restock()** mempunyai satu parameter berupa bilangan int?

Jawab: karena yang dibutuhkan hanya int saja dan membuat method mudah digunakan, fleksibel, aman, dan efisien

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

Jawab:

<https://github.com/revaniputeri/kampus/tree/main/smt2/Algoritma%20Struktur%20Dasar/PraTek/Pertemuan%202>

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:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src> & 'C:\
va\jdk1.8.0_231\bin\java.exe' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User
e\397d8c8389b25999f949ac0c15344c07\redhat.java\jdt_ws\src_4344ca31\bin' 'Buku
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
PS D:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src>
```

5. Commit dan push kode program ke Github

<https://github.com/revaniputeri/kampus/tree/main/smt2/Algoritma%20Struktur%20Dasar/Praktek/Pertemuan%202>

2.2.2 Verifikasi Hasil Percobaan

Cocokkan hasil compile kode program Anda dengan gambar berikut ini.

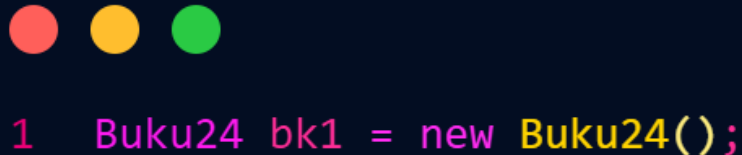
```
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 13
Harga: Rp 71000
Judul: Today Ends Tomorrow Comes
Pengarang: Denanda Pratiwi
Jumlah halaman: 198
Sisa stok: 8
Harga: Rp 60000
```

```
PS D:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src> & 'C:\Program Files\Java\jdk1.8.0_231\bin\java.exe' '-cp' 'C:\Users\LENOVO\AppData\Roaming\Code\User\globalStorage\397d8c8389b25999f949ac0c15344c07\redhat.java\jdt_ws\src_4344ca31\bin' 'BukuMain.java'
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
PS D:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src>
```

2.2.3 Pertanyaan

1. Pada class **BukuMain**, tunjukkan baris kode program yang digunakan untuk proses instansiasi! Apa nama object yang dihasilkan?

Jawab: nama object yang dihasilkan merupakan bk1



```
1  Buku24 bk1 = new Buku24();
```

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

Jawab: dengan menggunakan titik (.)

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

Jawab: karena **tampilInformasi()** yang pertama digunakan untuk menampilkan data yang belum mengalami perubahan, sedangkan **tampilInformasi()** yang kedua digunakan untuk menampilkan data setelah mengalami perubahan.

2.3 Percobaan 3: Membuat Konstruktork

Waktu Percobaan: 60 Menit

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

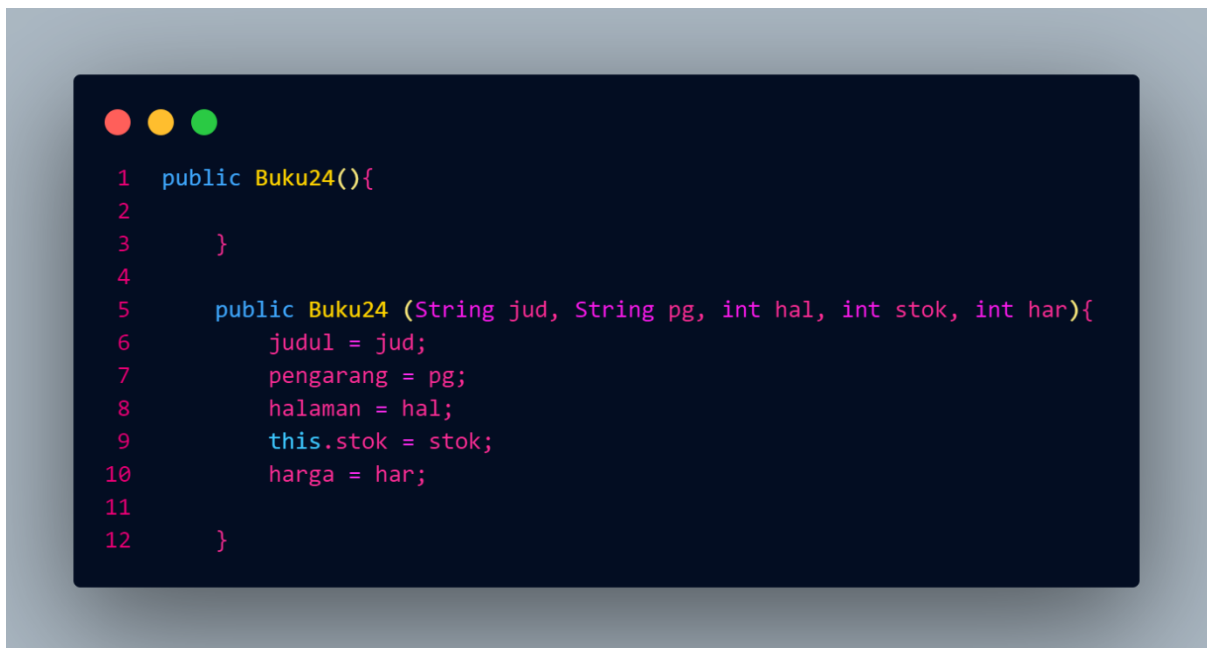
2.3.1 Langkah-langkah Percobaan

1. Buka kembali class **Buku**. Tambahkan dua buah konstruktork di dalam class **Buku** tersebut, yang terdiri dari satu konstruktork default dan satu konstruktork berparameter. Konstruktork merupakan method istimewa, penempatan kode program untuk konstruktork 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.

```
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
PS D:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src> █
```

4. Commit dan push kode program ke Github

<https://github.com/revaniputeri/kampus/tree/main/smt2/Algoritma%20Struktur%20Dasar/Praktek/Pertemuan%202>

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
```



```
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
PS D:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src> █
```

2.3.3 Pertanyaan

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

Jawab:

```
1 Buku24 bk2 = new Buku24("Self Reward", "Maheera Ayesha", 160, 29, 59000);
```

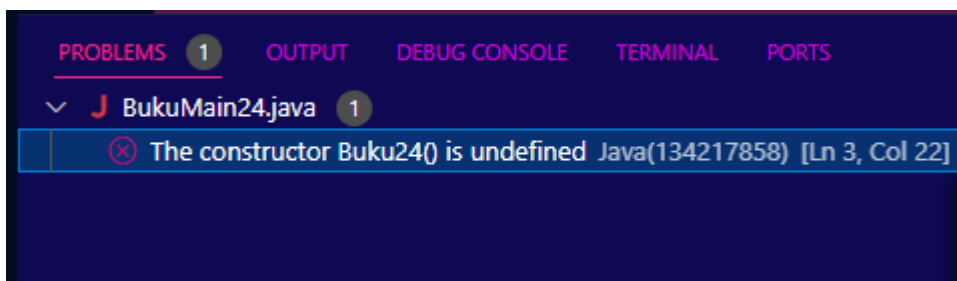
2. Perhatikan class **BukuMain**. Apa sebenarnya yang dilakukan pada baris program berikut?

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

Jawab: deklarasi variable, instansiasi objek, pemanggilan konstruktor, inisialisasi atribut

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

Jawab: terjadi error karena konstruktor tidak terdeteksi di file **Buku24.java**





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

Jawab: Tidak, method di dalam class Buku tidak harus diakses secara berurutan. Akses method di dalam class Buku bersifat independen, artinya Anda dapat mengakses method apa pun dalam urutan apa pun, sesuai kebutuhan.

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

Jawab:

```
1  Buku24 BukuReva = new Buku24("It Ends With Us", "Collen Hoover", 300, 10, 230000);
2      BukuReva.tampilInformasi();
3      }
```

```
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
Judul: It Ends With Us
Pengarang: Collen Hoover
Jumlah halaman: 300
Sisa stok: 10
Harga: Rp. 230000
PS D:\Kulyeahhh\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src> █
```

6. Commit dan push kode program ke Github

<https://github.com/revaniputeri/kampus/tree/main/smt2/Algoritma%20Struktur%20Dasar/Praktek/Pertemuan%202>

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

```

1 void hitungHargaTotal(int jml){
2     totalHarga = harga * jml;
3 }
    
```

- 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

```

1 double diskon = 0;
2 void hitungDiskon() {
3     if (totalHarga > 150000) {
4         diskon = 0.12;
5     } else if (totalHarga > 75_000 && totalHarga < 150_000) {
6         diskon = 0.05;
7     } else {
8         diskon = 0;
9     }
10 }
    
```

- Method **hitungHargaBayar()** digunakan untuk menghitung harga total setelah dikurangi diskon

```
1 void hitungHargaBayar() {
2     hargaBayar = totalHarga - (int) diskon;
3 }
```

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

```
1 Buku24 BukuReva = new Buku24("It Ends With Us", "Collen Hoover", 300, 10, 230000);
2     BukuReva.tampilInformasi();
3     BukuReva.terjual(8);
4     BukuReva.restock(3);
5     BukuReva.gantiHarga(100_000);
6     BukuReva.hitungHargaTotal(8);
7     BukuReva.hitungDiskon();
8     BukuReva.hitungHargaBayar();
9     BukuReva.tampilTransaksi();
```

```
Judul: It Ends With Us
Pengarang: Collen Hoover
Jumlah halaman: 300
Sisa stok: 10
Harga: Rp. 230000
Total Harga: 800000
Total Diskon: 96000
Total Bayar: 704000
```

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 $x > \text{width}$ 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 $y > \text{height}$ maka panggil method **detectCollision()**
- Method **detectCollision()** akan mencetak pesan "Game Over" apabila dragon menyentuh ujung area permainan.



```

1  public class Dragon24 {
2      int x, y, width, height;
3
4      public Dragon24(int w, int h) {
5          width = w;
6          height = h;
7      }
8
9      void printPosition() {
10         System.out.println("Dragon Current Position");
11         System.out.printf("x: %d\ny: %d\n", x, y);
12         System.out.println();
13     }
14
15     void detectCollision(int x, int y) {
16         if (x < 0 || x > width || y < 0 || y > height) {
17             System.out.println("Game Over !");
18             System.exit(0);
19         }
20     }
21
22     void moveRight() {
23         x++;
24         detectCollision(x, y);
25     }
26
27     void moveLeft() {
28         x--;
29         detectCollision(x, y);
30     }
31
32     void moveUp() {
33         y++;
34         detectCollision(x, y);
35     }
36
37     void moveBottom() {
38         y--;
39         detectCollision(x, y);
40     }
41
42 }
    
```

```

1  import java.util.Scanner;
2
3  public class DragonMain24 {
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          Dragon24 dragon = new Dragon24(3, 5);
7
8          while (true) {
9              System.out.println("Move Dragon");
10             System.out.println("1. Move Right\n2. Move Left\n3. Move Up\n4. Move Bottom");
11             System.out.print("Enter: ");
12             int move = sc.nextInt();
13
14             System.out.println();
15
16             switch (move) {
17                 case 1:
18                     dragon.moveRight();
19                     break;
20                 case 2:
21                     dragon.moveLeft();
22                     break;
23                 case 3:
24                     dragon.moveUp();
25                     break;
26                 case 4:
27                     dragon.moveBottom();
28                     break;
29                 default:
30                     break;
31             }
32             dragon.printPosition();
33         }
34     }
35 }

```



```
PS D:\kampus\smt2\Algoritma Struktur Dasar\Praktek\Pertemuan 2\src> & 'C:\Program Files  
s\src_32f1005\bin' 'DragonMain24'  
Move Dragon  
1. Move Right  
2. Move Left  
3. Move Up  
4. Move Bottom  
Enter: 3  
  
Dragon Current Position  
x: 0  
y: 1  
  
Move Dragon  
1. Move Right  
2. Move Left  
3. Move Up  
4. Move Bottom  
Enter: 1  
  
Dragon Current Position  
x: 1  
y: 1  
  
Move Dragon  
1. Move Right  
2. Move Left  
3. Move Up  
4. Move Bottom  
Enter:   
Ready
```

Link github :

<https://github.com/revaniputeri/kampus/tree/main/smt2/Algoritma%20Struktur%20Dasar/Praktek/Pertemuan%202/src>