

**LAPORAN PRAKTIKUM**  
**STRUKTUR DATA LINIER**  
**MODUL I (REVIEW ARRAY DAN STATIC)**

Dosen Pengampu  
JB. Budi Darmawan S.T., M.Sc.




DISUSUN OLEH :  
AGUSTINUS KEVIN YUDIPRATAMA  
235314029

**PROGRAM STUDI INFORMATIKA**  
**FAKULTAS SAINS DAN TEKNOLOGI**  
**UNIVERSITAS SANATA DHARMA**  
**YOGYAKARTA**  
**2024**

## No. 1

- a. Bagian program yang menjadi base case dan making progress
- Base casenya adalah,  

```
public static void count(int index)
{
    System.out.print(index);
    if (index < 2)
        count(index + 1);
}
```
  - Making progress
    - Metode count dipanggil dengan nilai index sama dengan 3
    - Setelah itu cetak nilai 3
    - Cek 3 kurang dari 2 maka dari itu tidak ada rekusifnya
    - Hasilnya akan mencetak 3 dan program selesai
- b. Outputnya adalah
- 
- c. Jika terjadi error perbaiki.  
(dari program diatas tidak ada error)

## No. 2

- a. Bagian program yang menjadi base case dan making progres
- Base casenya adalah,  

```
public static void count (int index)
{
    if (index < 2)
    {
        count(index+1);
    }
    System.out.print(index);
}
```
  - Making progress
    - Metode count dipanggil dengan nilai index sama dengan 0
      - Cek apakah 0 kurang dari 2 jika iya, metode akan membuat rekusif dengan nambah 1 ke index
    - metode count dipanggil dengan nilai index sama dengan 1

- cek apakah 1 kurang dari 2 jika iya, metode membuat rekusif dengan nambah 1 ke index
  - Metode count dipanggil dengan nilai index sama dengan 2
    - Cek apakah 2 kurang dari 2 jika tidak, tidak terjadi rekusif cukup disitu
  - Program selesai
- b. Outpunya adalah
- 2  
1  
0
- c. Tidak terjadi error

### No. 3

- a. Bagian program yang menjadi base case dan making progres
- Base casenya adalah,
 

```
public static void count (int index)
{
    if (index < 2)
    {
        count(index+1);
        System.out.print(index);
    }
}
```
- b. Making progress
- Metode count dipanggil dengan nilai index sama dengan 3
    - Cek apakah 3 kurang dari 2 jika iya, metode akan membuat rekusif dengan nambah 1 ke index
  - metode count dipanggil dengan nilai index sama dengan 4
    - cek apakah 4 kurang dari 2 jika iya, metode membuat rekusif dengan nambah 1 ke index
  - metode count dipanggil dengan nilai index sama dengan 5
    - cek apakah 5 kurang dari 2 jika iya, metode membuat rekusif dengan nambah 1 ke index
  - metode count dipanggil dengan nilai index sama dengan 6
    - cek apakah 6 kurang dari 2 jika iya, metode membuat rekusif dengan nambah 1 ke index
  - metode count dipanggil dengan nilai index sama dengan 7
    - cek apakah 7 kurang dari 2 jika iya, metode membuat rekusif dengan nambah 1 ke index

No 2

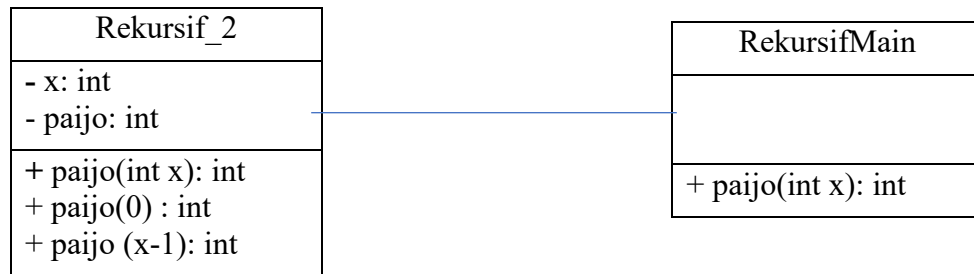
#### A. TUJUAN PRAKTIKUM

- Mahasiswa mengerti tentang method rekusif

#### B. ALGORITMA

- Tujuan dari method tersebut adalah untuk implementasi rekursif untuk menghitung suatu nilai berdasarkan rumus matematika
- Base casenya adalah  
if (x==0)  
return 0 ;  
return 2 \* paijo(x-1) + x \* x
- Making progres

#### C. DIAGRAM UML



#### D. SOURCE CODE

Clasnya

```
public class rekusif_2 {
    public static int paijo (int x){
        if (x==0)
            return 0;
        return 2 * paijo(x-1) + x * x;
    }
}
```

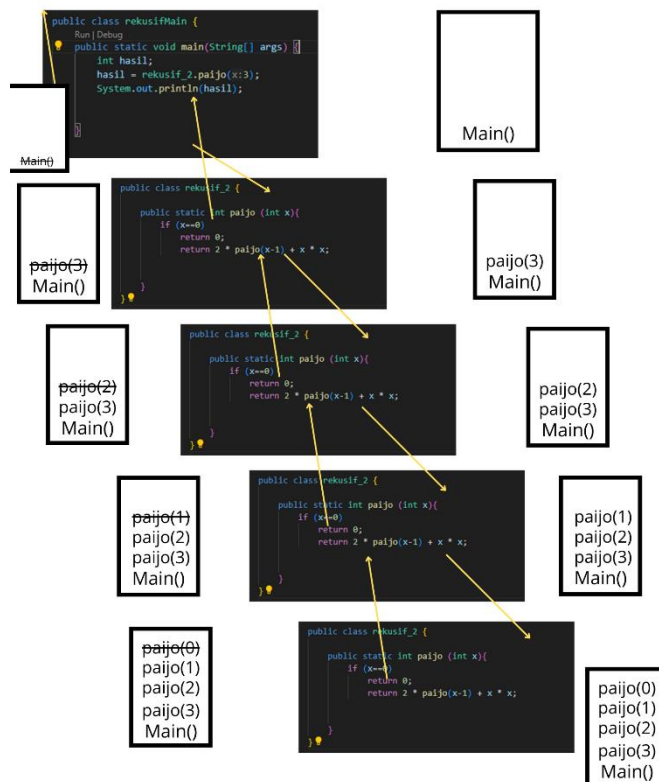
## Mainnya

```
public class rekusifMain {  
    Run | Debug  
    public static void main(String[] args) {  
        int hasil;  
        hasil = rekusif_2.paijo(x:3);  
        System.out.println(hasil);  
    }  
}
```

## Outputnya

```
a4f651b  
21  
PS C:\U
```

## E. ANALISA



No 3

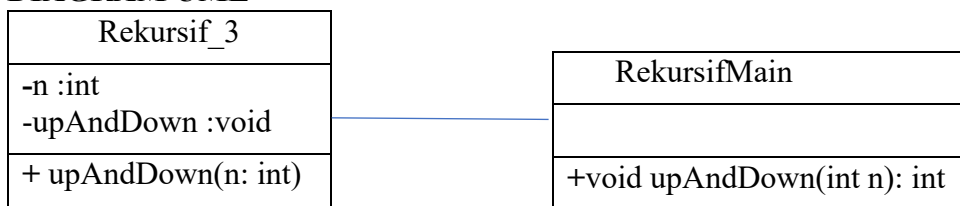
#### A. TUJUAN PRAKTIKUM

- Mahasiswa mengerti tentang method rekusif

#### B. ALGORITMA

- Tujuan dari method tersebut adalah untuk melakukan fungsi rekursif yang mencetak level saat ini dalam pola yang naik turun.
- Base casenya  
If (n<4);  
upAndDown(n+1);
- Making Progresnya
  - Metode awal dipanggil dengan variabel n
  - Mencetak level untuk nilai n saat ini
  - Jika n kurang dari 4, melakukan panggilan rekursif
  - Langkah sama diulang
  - Setelah panggilan rekursif cetak level lagi

#### C. DIAGRAM UML



#### D. SOURCE CODE

classnya

```
public class rekursif_3 {

    public static void upAndDown(int n){
        System.out.print("\nLevel: " + n);
        if (n < 4)
            upAndDown(n+1);
        System.out.print("\nLevel: " + n);
    }
}
```

Mainnya

```
public class rekusifMain {
    Run | Debug
    public static void main(String[] args) {
        rekursif_3.upAndDown(n:0);
    }
}
```

### Outputnya

```
Level: 0  
Level: 1  
Level: 2  
Level: 3  
Level: 4  
Level: 4  
Level: 3  
Level: 2  
Level: 1  
Level: 0
```

### E. ANALISIS

#### No 4

a. Class rekursif

```
public static int Faktorial (int x){  
    if (x==1){  
        return x ;  
  
    }else {  
        return x * Faktorial(x-1);  
    }  
}
```

b. Class iteratif

```
public static int faktorial (int x){  
    int total= 1;  
  
    for (int i = 1; i <= x; i++) {  
        total = i * total;  
  
    }  
    return total;  
}
```

c. Class main

```
public static void main(String[] args) {  
    int hasil;  
  
    hasil = Matematika.faktorial(4);  
    System.out.println(hasil);  
  
}
```

- d. Perbedaannya adalah yang satu dengan metode rekursif yang artinya metode ini memanggil dirinya sendiri untuk menyelesaikan perintahnya. Sedangkan metode iteratif itu menggunakan perulangan dengan for sebagai contoh dia tidak memanggil dirinya sendiri tapi melakukan looping.