

Nama: Ahmad Sofyan Badawi

NIM: 244107020073

Kelas: TI-1B

Percobaan 1

Langkah-langkah:

1. Buat project baru bernama **Rekursif**, dan buat file Java dengan nama **Percobaan1**
2. Buat fungsi static dengan nama faktorialRekursif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya.

```
static int faktorialRekursif(int n) {  
    if (n == 0) {  
        return (1);  
    } else {  
        return (n * faktorialRekursif(n - 1));  
    }  
}
```

3. Buat lagi fungsi static dengan nama faktorialIteratif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya.

```
static int faktorialIteratif(int n) {  
    int faktor = 1;  
    for (int i = n; i >= 1; i--) {  
        faktor = faktor * i;  
    }  
    return faktor;  
}
```

4. Buatlah fungsi main dan lakukan pemanggilan terhadap kedua fungsi yang telah dibuat sebelumnya, dan tampilkan hasil yang didapatkan.

```
public static void main(String[] args) {  
    System.out.println(faktorialRekursif(5));  
    System.out.println(faktorialIteratif(5));  
}
```

5. Jalankan program tersebut. Amati apa yang terjadi!
6. Jika ditelusuri, pada saat pemanggilan fungsi faktorialRekursif(5), maka proses yang terjadi dapat diilustrasikan sebagai berikut:

```

5 * hitungFaktorialRekursif(4)
    ↓
4 * hitungFaktorialRekursif(3)
    ↓
3 * hitungFaktorialRekursif(2)
    ↓
2 * hitungFaktorialRekursif(1)
    ↓
return 1

return 2 * 1 = 2

return 3 * 2 = 6

return 4 * 6 = 24

return 5 * 24 = 120

```

Pertanyaan:

1. Apa yang dimaksud dengan fungsi rekursif?
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?
3. Pada Percobaan1, apakah hasil yang diberikan fungsi faktorialRekursif() dan fungsi faktorialIteratif() sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iteratif!

Jawaban:

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri dalam proses eksekusinya
2. Dalam kasus ini, ia memanggil fungsi dalam fungsi itu sendiri

```

static int faktorialRekursif(int n) {
    if (n == 0) {
        return (1);
    } else {
        return (n * faktorialRekursif(n - 1));
    }
}

```

3. Hasilnya sama, Adapun alurnya sebagai berikut:

- faktorialRekursif(5) = **5 * faktorialRekursif(4)**
4 * faktorialRekursif(3)
3 * faktorialRekursif(2)
2 * faktorialRekursif(1)
1 * faktorialRekursif(0)
- faktorialIteratif(5) = **faktor = 1 * 5 = 5**
faktor = 5 * 4 = 20

faktor = 20 * 3 = 60
faktor = 60 * 2 = 120
faktor = 120 * 1 = 120

Percobaan 2

Langkah-langkah:

1. Pada project **Rekursif**, dan buat file Java dengan nama **Percobaan2**
2. Buat fungsi static dengan nama `hitungPangkat()`, dengan tipe data kembalian fungsi `int` dan memiliki 2 parameter dengan tipe data `int` berupa bilangan yang akan dihitung pangkatnya dan bilangan pangkatnya.

```
static int hitungPangkat(int x, int y) {  
    if (y == 0) {  
        return 1;  
    } else {  
        return (x * hitungPangkat(x, y - 1));  
    }  
}
```

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama `sc`.
4. Buatlah dua buah variabel bertipe `int` dengan nama `bilangan` dan `pangkat`.
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print("Bilangan yang dihitung: ");  
bilangan = sc.nextInt();  
System.out.print("Pangkat: ");  
pangkat = sc.nextInt();
```

6. Lakukan pemanggilan fungsi `hitungPangkat` yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```
System.out.println(hitungPangkat(bilangan, pangkat));
```

7. Jalankan program tersebut. Amati apa yang terjadi!

Pertanyaan:

1. Pada **Percobaan2**, terdapat pemanggilan fungsi rekursif `hitungPangkat(bilangan, pangkat)` pada fungsi main, kemudian dilakukan pemanggilan fungsi `hitungPangkat()` secara berulang kali. Jelaskan sampai kapan proses pemanggilan fungsi tersebut akan dijalankan!
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : **hitungPangkat(2,5)** dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

Jawaban:

1. Pemanggilan akan berhenti jika pangkat mencapai nilai 0

2.

```
static int tampilan(int x, int y){
    int temp = x;
    if (y == 0){
        temp = 1;
    }else{
        temp = tampilan(x, y - 1);
        System.out.print(x + "x");
    }
    return temp;
}

Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int bilangan, pangkat, deret;

    System.out.print(s:"Bilangan yang dihitung: ");
    bilangan = sc.nextInt();
    System.out.print(s:"Pangkat: ");
    pangkat = sc.nextInt();

    deret = hitungPangkat(bilangan, pangkat);
    System.out.println(tampilan(bilangan, pangkat) + "=" + deret);
}
```

Percobaan 3

Langkah-langkah:

1. Pada project Rekursif, dan buat file Java dengan nama Percobaan3
2. Buat fungsi static dengan nama hitungLaba(), dengan tipe data kembalian fungsi double dan memiliki 2 parameter dengan tipe data int berupa saldo investor dan lamanya investasi. Pada kasus ini dianggap laba yang ditentukan adalah 11% per tahun. Karena perhitungan laba adalah $\text{laba} * \text{saldo}$, sehingga untuk menghitung besarnya uang setelah ditambah laba adalah $\text{saldo} + \text{laba} * \text{saldo}$. Dalam hal ini, besarnya laba adalah $0.11 * \text{saldo}$, dan saldo dianggap $1 * \text{saldo}$, sehingga $1 * \text{saldo} + 0.11 * \text{saldo}$ dapat diringkas menjadi $1.11 * \text{saldo}$ untuk perhitungan saldo setelah ditambah laba (dalam setahun).

```
static double hitungLaba(double saldo, int tahun) {
    if (tahun == 0) {
        return (saldo);
    } else {
        return (1.11 * hitungLaba(saldo, tahun - 1));
    }
}
```

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama **sc**.
4. Buatlah sebuah variabel bertipe double dengan nama **saldoAwal** dan sebuah variabel bertipe int bernama **tahun**
5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print("Jumlah saldo awal : ");
saldoAwal = sc.nextInt();
System.out.print("Lamanya investasi (tahun) :");
tahun = sc.nextInt();
```

6. Lakukan pemanggilan fungsi **hitungLaba** yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```
System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
System.out.print(d:hitungLaba(saldo:saldoAwal, tahun));
```

7. Jalankan program tersebut. Amati apa yang terjadi!

Pertanyaan:

1. Pada Percobaan3, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!
2. Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai **hitungLaba(100000,3)**.

Jawaban:

1. Base call:

```
return (saldo);
```

Recursion call:

```
return (1.11 * hitungLaba(saldo, tahun - 1));
}
```

2. Fase ekspansi:

$$\begin{aligned} \text{hitungLaba}(100000, 3) &= 1.11 * \text{hitungLaba}(100000, 2) \\ &= 1.11 * (1.11 * \text{hitungLaba}(100000, 1)) \\ &= 1.11 * (1.11 * (1.11 * \text{hitungLaba}(100000, 1))) \end{aligned}$$

Fase substitusi:

$$\begin{aligned} \text{hitungLaba}(100000, 3) &= 1.11 * (1.11 * (1.11 * 100000)) \\ &= 1.11 * (1.11 * 111000) \\ &= 1.11 * (123210) \\ &= 136763,1 \end{aligned}$$

Tugas

1. Buatlah program untuk menampilkan bilangan n sampai 0 dengan menggunakan fungsi rekursif dan fungsi iteratif. (DeretDescendingRekursif).

2. Buatlah program yang di dalamnya terdapat fungsi rekursif untuk menghitung penjumlahan bilangan. Misalnya $f = 8$, maka akan dihasilkan $1+2+3+4+5+6+7+8 = 36$ (PenjumlahanRekursif).
3. Sepasang marmut yang baru lahir (jantan dan betina) ditempatkan pada suatu pembiakan. Setelah dua bulan pasangan marmut tersebut melahirkan sepasang marmut kembar (jantan dan betina). Setiap pasangan marmut yang lahir juga akan melahirkan sepasang marmut juga setiap 2 bulan. Berapa pasangan marmut yang ada pada akhir bulan ke-12? Buatlah programnya menggunakan fungsi rekursif! (Fibonacci). Berikut ini adalah ilustrasinya dalam bentuk tabel.

Bulan ke-	Jumlah Pasangan		Total Pasangan
	Produktif	Belum Produktif	
1	0	1	1
2	0	1	1
3	1	1	2
4	1	2	3
5	2	3	5
6	3	5	8
7	5	8	13
8	8	13	21
9	13	21	34
10	21	34	55
11	34	55	89
12	55	89	144

Jawaban

1. Program:

```

1  import java.util.Scanner;
2  public class DeretDescendingRekursif {
3      Run | Debug
4      public static void main(String[] args) {
5          Scanner sc = new Scanner(System.in);
6          int angka;
7
8          System.out.print(s:"Masukkan angka awal: ");
9          angka = sc.nextInt();
10
11         System.out.println(deretRekursif(angka) + " Ini Rekursif");
12         System.out.print(deretIteratif(angka) + " Ini Iteratif " );
13     }
14
15     static int deretRekursif (int n) {
16         int temp;
17         if (n == 0) {
18             temp = 0;
19         } else {
20             System.out.print (n + " ");
21             temp = (deretRekursif(n - 1));
22         }
23         return temp;
24     }
25
26     static int deretIteratif (int n) {
27         for (int i = n; i > 0; i--) {
28             System.out.print(i + " " );
29         }
30         return 0;
31     }
32 }

```

Output:

```

Masukkan angka awal: 5
5 4 3 2 1 0 Ini Rekursif
5 4 3 2 1 0 Ini Iteratif

```

2. Program:

```

1  import java.util.Scanner;
2  public class PenjumlahanRekursif {
    Run | Debug
3      public static void main(String[] args) {
4          Scanner sc = new Scanner(System.in);
5          int angka;
6
7          System.out.print(s:"Masukkan Angka: ");
8          angka = sc.nextInt();
9
10         System.out.println(x:"Jumlah Deret Angka: ");
11         tampilJumlah(angka);
12         System.out.println("\b = " + penjumlahanRekursif(angka));
13     }
14
15     static int penjumlahanRekursif(int n) {
16         int temp = n;
17         if (n == 0) {
18             temp = n;
19         } else {
20             temp += penjumlahanRekursif(n - 1);
21         }
22         return temp;
23     }
24
25     static void tampilJumlah(int n) {
26         if (n == 0) {
27             return;
28         } else {
29             tampilJumlah(n - 1);
30             System.out.print(n);
31             if (n >= 1) {
32                 System.out.print(s:"+");
33             }
34         }
35     }
36
37 }

```

Output:

```

Masukkan Angka: 5
Jumlah Deret Angka:
1+2+3+4+5 = 15

```

3. Program:


```

TugasFibonacci.java > TugasFibonacci > hitungHamster(int)
1  import java.util.Scanner;
2  public class TugasFibonacci {
    Run | Debug
3      public static void main(String[] args) {
4          Scanner sc = new Scanner(System.in);
5          int bulan, hamster;
6
7          System.out.print(s:"Masukkan berapa bulan pembiakan: ");
8          bulan = sc.nextInt();
9          System.out.println("Banyak pasangan yang ada yaitu: " + hitungHamster(bulan));
10     }
11
12
13     static int hitungHamster(int bulan){
14         int hamster = 1;
15         if (bulan <= 1) {
16             return bulan;
17         }
18         return hitungHamster(bulan - 1) + hitungHamster(bulan-2);
19     }
20
21 }

```

Output:

```

C:\Users\user> java TugasFibonacci
Masukkan berapa bulan pembiakan: 5
Banyak pasangan yang ada yaitu: 5

```