

Laporan praktek Jobsheet 12

Fungsi 2

Nama : Gaduh Prakoso

Kelas : TI-1D

Absen : 10

NIM : 244107020150

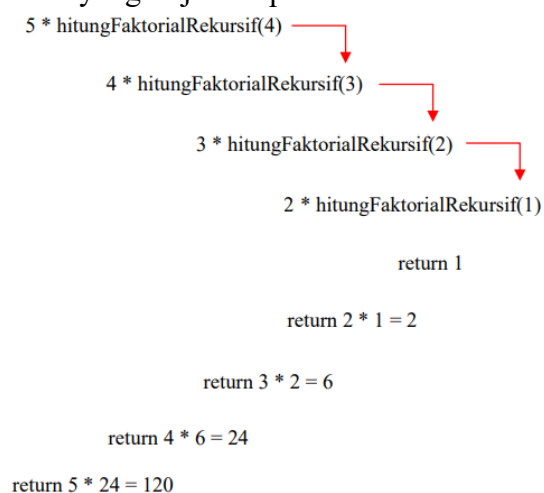
1. Praktikum

1.1 Percobaan 1

- 1) Buat project baru bernama Rekursif, dan buat file Java dengan nama Percobaan1
- 2) Buat program seperti dibawah ini:

```
public class Percobaan1{  
  
    public static int faktorialRekursif(int n){  
        if (n == 0) {  
            return (1);  
        }else {  
            return (n* faktorialRekursif(n-1));  
        }  
    }  
  
    public static int faktorialIteratif(int n){  
        int faktor = 1;  
        for (int i = n; i >= 1; i--){  
            faktor = faktor * i;  
        }  
        return faktor;  
    }  
    Run | Debug  
    public static void main(String[] args) {  
        System.out.println(faktorialRekursif(n:5));  
        System.out.println(faktorialIteratif(n:5));  
    }  
}
```

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



Pertanyaan

- 1) Apa yang dimaksud dengan fungsi rekursif?
 - Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri dalam proses eksekusi untuk menyelesaikan suatu masalah. Fungsi ini biasanya digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi submasalah yang lebih kecil dengan struktur yang serupa
- 2) Bagaimana contoh kasus penggunaan fungsi rekursif ?
 - Menghitung hasil pangkat sebuah bilangan
 - mencari faktorial sebuah bilangan
 - mencari deret fibonacci
- 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!
 - Iya keduanya sama sama menghasilkan nilai 120, yang membedakan adalah cara pengerjaannya.
 - **Rekursif:** Fungsi memanggil dirinya sendiri hingga mencapai kondisi dasar ($n == 0$), lalu hasil dihitung saat "kembali" dari setiap pemanggilan.
 - **Iteratif:** Menggunakan perulangan for untuk menghitung faktorial secara bertahap, mulai dari n hingga 1.

1.2 Percobaan 2

- 1) Pada project Rekursif buat file Java dengan nama Percobaan2
- 2) Buat program seperti dibawah ini:

```
Percobaan2.java > ...
1  import java.util.Scanner;
2
3  public class Percobaan2 {
4      static int hitungPangkat(int x,int y){
5          if (y == 0){
6              return(1);
7          }else {
8              return (x*hitungPangkat(x, y-1));
9          }
10     }
11     Run | Debug
12     public static void main(String[] args){
13         Scanner sc = new Scanner(System.in);
14
15         int bilangan;
16         int pangkat;
17
18         System.out.print(s:"Bilangan yang dihitung: ");
19         bilangan = sc.nextInt();
20         System.out.print(s:"Pangkat: ");
21         pangkat = sc.nextInt();
22
23         System.out.println(hitungPangkat(bilangan, pangkat));
24     }
25 }
```

- 3) Jalankan program tersebut. Amati apa yang terjadi!

Pertayaan

- 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!
 - Fungsi `hitungPangkat` akan dipanggil berulang kali selama nilai `y` (`pangkat`) lebih besar dari 0. Setiap kali, nilai `y` akan dikurangi 1. Proses ini berhenti ketika `y` mencapai 0, yang menjadi kondisi berhenti (`base case`), dan fungsi mengembalikan nilai 1. Setelah itu, hasil dari setiap pemanggilan akan dihitung kembali dan dikembalikan ke pemanggil sebelumnya sampai akhirnya nilai total diprint di `main`.
- 2) Tambahkan kode program untuk mencetak deret perhitungannya.
Contoh : `hitungPangkat(2,5)` dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

```
Percobaan2.java > ...
1  import java.util.Scanner;
2
3  public class Percobaan2 {
4      static int hitungPangkat(int x, int y) {
5          if (y == 0) {
6              System.out.print(s:"1");
7              return 1;
8          } else {
9              System.out.print(x + "x");
10             return x * hitungPangkat(x, y - 1);
11         }
12     }
13
14     Run | Debug
15     public static void main(String[] args) {
16         Scanner sc = new Scanner(System.in);
17
18         int bilangan;
19         int pangkat;
20
21         System.out.print(s:"Bilangan yang dihitung: ");
22         bilangan = sc.nextInt();
23         System.out.print(s:"Pangkat: ");
24         pangkat = sc.nextInt();
25
26         System.out.print(s:"Deret perhitungannya: ");
27         int hasil = hitungPangkat(bilangan, pangkat);
28         System.out.println(" = " + hasil);
29     }
}
```

1.3 Percobaan 3

- 1) Pada project Rekursif, dan buat file Java dengan nama `Percobaan3`
- 2) Buat program seperti dibawah ini:

```

J Percobaan3.java > ...
1  import java.util.Scanner;
2
3  public class Percobaan3 {
4      static double hitungLaba (double saldo,int tahun) {
5          if (tahun == 0){
6              return (saldo);
7          } else {
8              return (1.11 * hitungLaba(saldo,tahun - 1));
9          }
10     }
11     Run | Debug
12     public static void main(String[] args) {
13         Scanner sc = new Scanner(System.in);
14
15         double saldoAwal;
16         int tahun;
17
18         System.out.print(s:"Jumlah saldo awal: ");
19         saldoAwal = sc.nextInt();
20         System.out.print(s:"Lamanya investasi (tahun): ");
21         tahun = sc.nextInt();
22
23         System.out.print("Jumlah saldo setelah " + tahun + " tahun: ");
24         System.out.print(hitungLaba(saldoAwal, tahun));
25     }
26 }

```

3) Jalankan program tersebut. Amati apa yang terjadi!

Pertanyaan

- 1) Pada Percobaan3, sebutkan blok kode program manakah yang merupakan “base case” dan “recursion call”!

```

J Percobaan3.java > ...
1  import java.util.Scanner;
2
3  public class Percobaan3 {
4      static double hitungLaba(double saldo, int tahun) {
5          if (tahun == 0) { // Base Case
6              return (saldo);
7          } else {
8              return (1.11 * hitungLaba(saldo, tahun - 1)); //Recursion Call
9          }
10     }
11 }

```

- 2) Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai hitungLaba(100000,3)

Fase Ekspansi:

- Fungsi hitungLaba(100000, 3) dipanggil, karena tahun tidak 0, maka memanggil hitungLaba(100000, 2).
- Fungsi hitungLaba(100000, 2) dipanggil, karena tahun tidak 0, maka memanggil hitungLaba(100000, 1).
- Fungsi hitungLaba(100000, 1) dipanggil, karena tahun tidak 0, maka memanggil hitungLaba(100000, 0).
- Fungsi hitungLaba(100000, 0) dipanggil dan mengembalikan saldo 100000 (base case).

Fase Substitusi:

- Fungsi hitungLaba(100000, 1) menggantikan hitungLaba(100000, 0) dengan 100000 dan menghitung $1.11 * 100000 = 111000$.

- Fungsi hitungLaba(100000, 2) menggantikan hitungLaba(100000, 1) dengan 111000 dan menghitung $1.11 * 111000 = 123210$.
- Fungsi hitungLaba(100000, 3) menggantikan hitungLaba(100000, 2) dengan 123210 dan menghitung $1.11 * 123210 = 136771.1$.

Hasil Akhir:

Saldo setelah 3 tahun adalah 136771.1.

2. Tugas