

PRAKTIKUM DASAR PEMROGRAMAN

JOBSHEET 14

Fungsi 1



Nama

Dimas Adi Bayu Samudra

NIM

2341720169

Kelas

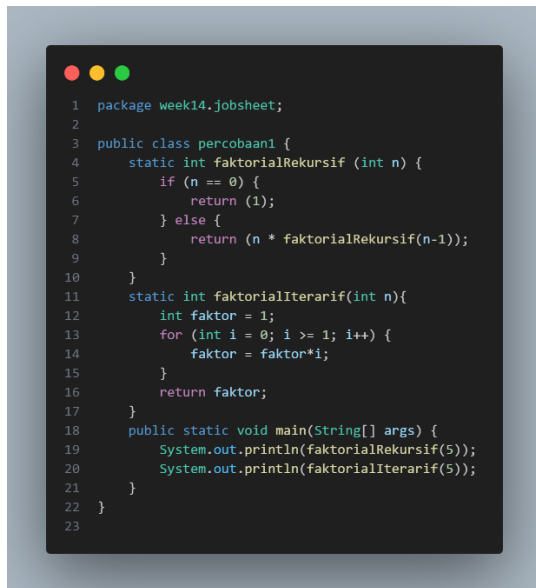
1A

JURUSAN TEKNOLOGI INFORMASI

PROGRAM STUDI D-IV TEKNIK INFORMATIKA

2.1 Percobaan 1

Input :



```
1 package week14.jobsheet;
2
3 public class percobaan1 {
4     static int faktorialRekursif (int n) {
5         if (n == 0) {
6             return (1);
7         } else {
8             return (n * faktorialRekursif(n-1));
9         }
10    }
11    static int faktorialIteratif(int n){
12        int faktor = 1;
13        for (int i = 0; i >= 1; i++) {
14            faktor = faktor*i;
15        }
16        return faktor;
17    }
18    public static void main(String[] args) {
19        System.out.println(faktorialRekursif(5));
20        System.out.println(faktorialIteratif(5));
21    }
22 }
23
```

Output :



```
120
1
```

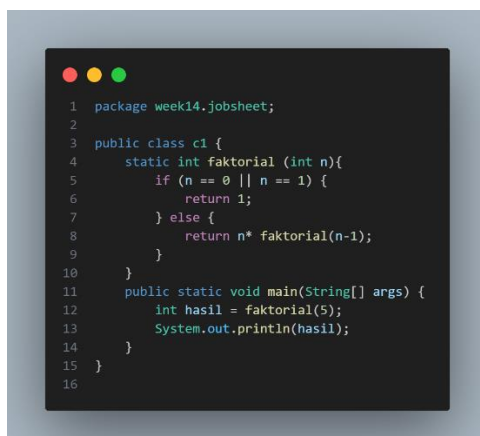
Pertanyaan :

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!

Jawab :

1. pada fungsi rekursif , didalam fungsi tersebut yang di panggil oleh fungsi itu sendiri, dengan begitu proses pemanggilan fungsi akan terjadi secara berulang-ulang

2. Input :



```
1 package week14.jobsheet;
2
3 public class c1 {
4     static int faktorial (int n){
5         if (n == 0 || n == 1) {
6             return 1;
7         } else {
8             return n* faktorial(n-1);
9         }
10    }
11    public static void main(String[] args) {
12        int hasil = faktorial(5);
13        System.out.println(hasil);
14    }
15 }
16
```

Output :

120

3. Pada fungsi rekursif menggunakan if. if dideklarasikan dengan kondisi jika 'n == 0' maka return 1. Tetapi, jika belum memenuhi kondisi maka akan masuk else dengan return n dikalikan dengan parameter 'n-1'. Nantinya akan terus mengulang sampai 'n==0'. - Pada fungsi Iteratif menggunakan perulangan for. For dideklarasikan dengan int i diinisialisasi dengan 1, kemudian deklarasi kondisi selama i lebih dari sama dengan 1, maka akan terus perulangan akan terus berjalan. Kemudian i akan dikurangi setiap perulangan. Di dalamnya deklarasi int factor kemudian factor akan dikali dengan I dan hasilnya disimpan di variable factor. Perkalian akan terus dilakukan sesuai dengan kondisi yang dideklarasikan. Kemudian return factor.

2.2 Percobaan 2

Input :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class percobaan2 {
6     static int hitungPangkat(int x, int y){
7         if (y == 0) {
8             return (1);
9         } else {
10            return (x * hitungPangkat(x, y -1));
11        }
12    }
13
14    public static void main(String[] args) {
15        Scanner sc = new Scanner(System.in);
16        int bilangan, pangkat;
17        System.out.println("Bilangan yang dihitung ; ");
18        bilangan = sc.nextInt();
19        System.out.println("Pangkat ; ");
20        pangkat = sc.nextInt();
21
22        System.out.println(hitungPangkat(bilangan, pangkat));
23    }
24 }
25
```

Output :

```
Bilangan yang dihitung ;
5
Pangkat ;
5
3125
```

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. Jawab :

1. Proses pemanggilan akan terus dijalankan selama y/bilpangkat !=0. Ketika y/bilpangkat sudah == 0, maka program akan berhenti.

2. hasil modif :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class percobaan2 {
6     static int hitungPangkat(int x, int y) {
7         if (y == 0) {
8             return 1;
9         } else {
10             int hasilPangkat = x * hitungPangkat(x, y - 1);
11             System.out.print(x);
12             if (y > 1) {
13                 System.out.print("x");
14             }
15             return hasilPangkat;
16         }
17     }
18
19     public static void main(String[] args) {
20         Scanner sc = new Scanner(System.in);
21         int bilangan, pangkat;
22         System.out.print("Bilangan yang dihitung : ");
23         bilangan = sc.nextInt();
24         System.out.print("Pangkat : ");
25         pangkat = sc.nextInt();
26
27         System.out.print("Hasil Pangkat: ");
28         int hasil = hitungPangkat(bilangan, pangkat);
29         System.out.println(" = " + hasil);
30     }
31 }
32
```

2.3 Percobaan 3

Input :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class Percobaan3 {
6     static double hitunglaba (double saldo,int tahun ) {
7         if (tahun == 0) {
8             return (saldo);
9         } else {
10             return (1.11 * hitunglaba(saldo, tahun -1 ));
11         }
12     }
13
14     public static void main(String[] args) {
15         Scanner sc = new Scanner(System.in);
16         double saldoAwal;
17         int tahun;
18
19         System.out.print("Jumlah Saldo Awal : ");
20         saldoAwal = sc.nextInt();
21         System.out.print("lamanya investasi (tahun) : ");
22         tahun = sc.nextInt();
23         System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
24         System.out.print(hitunglaba(saldoAwal, tahun));
25     }
26 }
27
```

Output :

```
Jumlah Saldo Awal : 100000
Lamanya investasi (tahun) : 5
Jumlah saldo setelah 5 tahun : 168505.81551000007
```

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).

Jawab :

1. Pada blok kode di atas, program menguji apakah variabel tahun sama dengan 0. Jika iya, maka fungsi hitunglaba akan mengembalikan nilai saldo tanpa melakukan pemanggilan rekursif lagi. Ini adalah kondisi dasar atau "base case" yang menghentikan rekursi.

2. Fase ekspansi

$$\begin{aligned} &\text{hitungLaba}(100000, 3) \\ &= 1.11 * \text{hitungLaba}(100000, 2) \\ &= 1.11 * (1.11 * \text{hitungLaba}(100000, 1)) \\ &= 1.11 * (1.11 * (1.11 * \text{hitungLaba}(100000, 0))) \end{aligned}$$

- Fase substitusi

$$\begin{aligned} &= 1.11 * (1.11 * (1.11 * 100000)) \\ &= 1.11 * (1.11 * 111.000) \\ &= 1.11 * 123.210, 2 \\ &= 136.763,1 \end{aligned}$$

Tugas

1. Input :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class Tugas1 {
6     static int hitungRekrusif(int n) {
7         if (n == 0) {
8             System.out.print(0 + " =");
9             return 0;
10        } else {
11            System.out.print(n + " - ");
12            return (hitungRekrusif(n-1));
13        }
14    }
15    static int hitungIteratif (int n) {
16        int hitung = n;
17        for (int i = 0; i <= n; i++) {
18            hitung = n-i;
19            if (i <= n) {
20                System.err.print(hitung + " - ");
21            }
22        }
23        return hitung;
24    }
25    public static void main(String[] args) {
26        Scanner sc = new Scanner(System.in);
27
28        int angka;
29        System.out.print("Masukan angka : ");
30        angka = sc.nextInt();
31        sc.close();
32        System.out.println("Fungsi rekrusif : ");
33        System.out.println(hitungRekrusif(angka));
34        System.out.println("\nFungsi iteratif");
35        int hasil = hitungIteratif(angka);
36        System.out.print(" = " + hasil);
37    }
38 }
39
```

Output :

```
Masukan angka : 10
Fungsi rekrusif :
10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 - 0 = 0

Fungsi iteratif
10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 - 0 - = 0
```

2. Input :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class Tugas2 {
6     static int penjumlahanRekrusif(int n) {
7         if (n == 1) {
8             System.out.print("1");
9             return 1;
10        } else {
11            int hasil = penjumlahanRekrusif(n - 1) + n;
12            System.out.print(" + " + n);
13            return hasil;
14        }
15    }
16    public static void main(String[] args) {
17        Scanner sc = new Scanner(System.in);
18
19        int angka;
20        System.out.print("Masukan angka: ");
21        angka = sc.nextInt();
22        sc.close();
23        int result = penjumlahanRekrusif(angka);
24        System.out.print(" = " + result);
25    }
26 }
27
```

Output :

```
Masukan angka: 8
1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36
```

3. Input :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class Tugas3 {
6     static boolean isPrima(int angka, int n) {
7         if (n <= 1) {
8             return false;
9         } else if (n > Math.sqrt(angka)) {
10            return true;
11        } else if (n % n == 0) {
12            return false;
13        } else {
14            return isPrima(angka, n + 1);
15        }
16    }
17    public static void main(String[] args) {
18        Scanner sc = new Scanner(System.in);
19        System.out.print("Masukan angka : ");
20        int angka = sc.nextInt();
21        sc.close();
22
23        if (isPrima(angka, 2)) {
24            System.out.println(angka + " adalah bilangan prima. ");
25        } else {
26            System.out.println(angka + " bukan bilangan prima");
27        }
28    }
29 }
30
```

Output :

```
Masukan angka : 5
5 bukan bilangan prima
```

```
Masukan angka : 4
4 bukan bilangan prima
```

```
Masukan angka : 19
19 bukan bilangan prima
```

4. Input :

```
1 package week14.jobsheet;
2
3 import java.util.Scanner;
4
5 public class Tugas4 {
6     public static int fibonacci (int n) {
7         if (n == 0 || n == 1) {
8             return n;
9         } else {
10             return fibonacci(n - 1) + fibonacci(n - 2);
11         }
12     }
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15         System.out.print("Masukan bulan ke- ");
16         int bulan = sc.nextInt();
17         sc.close();
18
19         int Jumlah = fibonacci(bulan);
20         System.out.println("Jumlah pasangan marmut pada akhir bulan ke- " + bulan + " adalah " + Jumlah);
21     }
22 }
23
```

Output :

```
Masukan bulan ke- 5
Jumlah pasangan marmut pada akhir bulan ke- 5 adalah 5
```