

Nama : Galang Satriyo Anorogo Winnada

NIM : 254107020231

Kelas : 1 – H

Mata Kuliah : Praktikum Daspro

JOBSHEET 12

1. Tujuan

Mahasiswa mampu memahami konsep fungsi rekursif

Mahasiswa mampu mengimplementasikan fungsi rekursif dalam kode program

2. Praktikum

2.1 Percobaan 1:

Waktu Percobaan: 60 menit

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung nilai faktorial dari suatu bilangan dengan menggunakan fungsi rekursif. Selain itu, akan dibuat juga fungsi untuk menghitung nilai faktorial dengan menggunakan algoritma iteratif sebagai pembandingnya.

1. Buat folder baru pada repositori lokal Anda, beri nama jobsheet12. Selanjutnya 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.

```
1 package jobsheet12;
2
3 public class percobaan11 {
4     static int faktorialRekursif(int n) {
5         if (n == 0) {
6             return (1);
7         } else {
8             return (n * faktorialRekursif(n - 1));
9         }
10    }
```

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

```
Run | Debug
public static void main(String[] args) {
    System.out.println(faktorialRekursif(n: 5));
    System.err.println(faktorialIteratif(n: 5));
}
```

5. Jalankan program tersebut.

```
120
120
PS C:\Users\shelfy\OneDrive\Documents\MATA KULIAH\KODINGAN\JOBSHEET\PraktikumDaspro> 
```

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

7. Commit program Anda ke Github dengan pesan “Percobaan 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!

Jawaban

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri untuk menyelesaikan suatu masalah.

Ciri utama fungsi rekursif:

- Memiliki pemanggilan diri sendiri

- Memiliki kondisi berhenti (base case) agar tidak terjadi perulangan tak hingga

2. Beberapa contoh penggunaan fungsi rekursif:

- Menghitung faktorial (seperti pada program Anda)
- Deret Fibonacci
- Menelusuri folder dan subfolder
- Pencarian data pada struktur pohon (tree)
- Algoritma pembagian masalah (divide and conquer)

3. Hasilnya sama Baik fungsi rekursif maupun iteratif sama-sama menghitung faktorial dari angka 5

Alur Rekursif

1. faktorialRekursif(5) dipanggil
2. Memanggil faktorialRekursif(4)
3. Terus berlangsung sampai n == 0
4. Hasil dikalikan saat kembali ke atas

Alur Iteratif

1. Variabel faktor dimulai dari 1
2. Perulangan dari n sampai 1
3. Dikalikan terus di dalam satu fungsi
4. Hasil langsung didapat tanpa pemanggilan fungsi ulang

2.2 Percobaan 2:

Waktu Percobaan: 60 menit

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung pangkat sebuah bilangan dengan menggunakan fungsi rekursif.

1. Buat file Java baru 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.

```
package jobsheet12;

public class percobaan211 {
    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
6. Lakukan pemanggilan fungsi hitungPangkat yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

```
Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    int bilangan, pangkat;

    System.out.println("Bilangan yang dihitung: ");
    bilangan = sc.nextInt();
    System.out.println("Pangkat: ");
    pangkat = sc.nextInt();
    System.out.println(hitungPangkat(bilangan, pangkat));
}
```

7. Jalankan program tersebut.

```
Bilangan yang dihitung:
5
Pangkat:
3
125
PS C:\Users\shelfy\OneDrive\Documents\MATA KULIAH\KODINGAN\JOBSITE\PraktikumDaspro>
```

8. Commit program Anda ke Github dengan pesan “Percobaan 2”

Pertanyaan

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

JAWABAN

1. akan terus berjalan selama nilai y (pangkat) belum bernilai 0 proses akan BERHENTI ketika mencapai kondisi dasar (base case) setelah $y == 0$, fungsi tidak memanggil dirinya lagi, lalu hasil dikembalikan ke atas hingga menghasilkan nilai akhir pemanggilan rekursif berjalan sampai nilai pangkat (y) menjadi 0.

```

import java.util.Scanner;
public class percobaan31 {
    static int hitungAngka (int x, int y) {
        if (y == 0) {
            System.out.print(x);
            return (1);
        } else {
            System.out.print(x + "*");
            return (x * hitungAngka(x, y - 1));
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int bilangan, pangkat;
        System.out.print("Bilangan yang dihitung: ");
        bilangan = sc.nextInt();
        System.out.print("Pangkat: ");
        pangkat = sc.nextInt();
        System.out.print("Hasil: ");
        int hasil = hitungAngka(bilangan, pangkat);
        System.out.println("*" + hasil);
    }
}

```

2.

Bilangan yang dihitung: 2

Pangkat: 5

$2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

2.3 Percobaan 3:

Waktu Percobaan : 60 menit

Pada percobaan ini akan dilakukan pembuatan program untuk menghitung jumlah uang investor yang digunakan sebagai investasi setelah mendapatkan laba selama beberapa tahun dengan menggunakan fungsi rekursif.

1. Buat file Java baru 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 laba * saldo, sehingga untuk menghitung besarnya uang setelah ditambah laba adalah saldo + laba * saldo. Dalam hal ini, besarnya laba adalah 0.11 * saldo, dan saldo dianggap 1 * saldo, sehingga $1 * saldo + 0.11 * saldo$ dapat diringkas menjadi $1.11 * saldo$ untuk perhitungan saldo setelah ditambah laba (dalam setahun).

```

5  public class percobaan31 {
6
7      static double hitungLaba(double saldo, int tahun) {
8          if (tahun == 0) {
9              return saldo;
10         } else {
11             return 1.11 * hitungLaba(saldo, tahun - 1);
12         }
13     }

```

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.

```

3 import java.util.Scanner;
4
5 public class percobaan31 {
6
7     static double hitungLaba(double saldo, int tahun) {
8         if (tahun == 0) {
9             return saldo;
10        } else {
11            return 1.11 * hitungLaba(saldo, tahun - 1);
12        }
13    }
14
15    public static void main(String[] args) {
16        Scanner sc = new Scanner(System.in);
17
18        double saldoAwal;
19        int tahun;
20
21        System.out.print("Jumlah saldo awal : ");
22        saldoAwal = sc.nextInt();
23        System.out.print("Lamanya investasi (tahun) : ");
24        tahun = sc.nextInt();
25        System.out.print("Jumlah saldo setelah " + tahun + " tahun : ");
26        System.out.print(hitungLaba(saldoAwal, tahun));
27    }
}

```

7. Jalankan program tersebut.

```

Jumlah saldo awal : 100000
Lamanya investasi (tahun) : 2
Jumlah saldo setelah 2 tahun : 123210.0000000003
PS C:\Users\shelfy\OneDrive\Documents\MATA KULIAH\KODINGAN\JOBSITE\PraktikumDaspro> []

```

8. Commit program Anda ke Github dengan pesan “Percobaan 3”

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 Case

Bagian ini adalah kondisi berhentinya rekursi:

```

if (tahun == 0) {
    return saldo;
}

```

Jika tahun == 0, maka fungsi tidak memanggil dirinya lagi dan langsung mengembalikan nilai saldo.

Recursion Call

Bagian ini adalah pemanggilan fungsi secara berulang:

```

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

```

Di sinilah fungsi hitungLaba() memanggil dirinya sendiri dengan nilai tahun yang dikurangi 1.

2. Fase Ekspansi (Pemanggilan Turun) Fungsi akan terus memanggil dirinya sampai tahun == 0:

$$\text{hitungLaba}(100000, 3)$$

$$= 1.11 * \text{hitungLaba}(100000, 2)$$

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

Fase Substitusi (Pengembalian Nilai Naik) Mulai dari base case:

$$\text{hitungLaba}(100000, 0) = 100000$$

Naik satu per satu:

$$\text{hitungLaba}(100000, 1)$$

$$= 1.11 * 100000$$

$$= 111000$$

$$\text{hitungLaba}(100000, 2)$$

$$= 1.11 * 111000$$

$$= 123210$$

$$\text{hitungLaba}(100000, 3)$$

$$= 1.11 * 123210$$

$$= 136763.1$$

Hasil akhir

$$\text{hitungLaba}(100000, 3) = 136763.1$$

Artinya, saldo Rp100.000 setelah 3 tahun dengan laba 11% per tahun menjadi: Rp136.763,10

Tugas

Waktu Pengerjaan 120 menit

1. Buatlah program untuk menampilkan bilangan n sampai 0 dengan menggunakan fungsi rekursif dan fungsi iteratif. (DeretDescendingRekursif). Commit dan push program Anda ke Github dengan pesan “Tugas1”
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). Commit dan push program Anda ke Github dengan pesan “Tugas2”
3. Sepasang marmut yang baru lahir (jantan dan betina) ditempatkan pada suatu pembiakan. Setelah dua bulan pasangan marmut tersebut melahirkan sepasangmarmut 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

Commit dan push program Anda ke Github dengan pesan “Tugas3”

Jawab

```
package jobsheet12;
import java.util.Scanner;

public class Tugas1 {
    static void faktorialRekursif(int n) {
        if (n < 0) {
            return;
        } else {
            System.out.print(n + " ");
            faktorialRekursif(n - 1);
        }
    }
    static void faktorialIteratif(int n) {
        for (int i = n; i >= 0; i--) {
            System.out.print(i + " ");
        }
    }
}
Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int n;

    System.out.print("Masukkan nilai n: ");
    n = sc.nextInt();
    System.out.println("Deret Descending Rekursif:");
    faktorialRekursif(n);
    System.out.println("Deret Descending Iteratif:");
    faktorialIteratif(n);
}

1. 0
```

Penjelasan Singkat

- Rekursif memanggil fungsi sendiri dengan $n - 1$
- Iteratif menggunakan perulangan for
- Keduanya menghasilkan output yang sama, tapi cara kerjanya berbeda

```
package jobsheet12;
import java.util.Scanner;

public class Tugas2 {
    static int hitungJumlah(int f) {
        if (f == 1) {
            return 1;
        } else {
            return f + hitungJumlah(f - 1);
        }
    }
}
Run | Debug
public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    int f;
    System.out.print("Masukkan nilai f: ");
    f = sc.nextInt();
    int hasil = hitungJumlah(f);
    System.out.println("Hasil penjumlahan = " + hasil);
}

2. 1
```

$$\text{Karena : } 1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 = 36$$

Base Case: $f == 1 \rightarrow$ berhenti rekursi: $f + \text{hitungJumlah}(f - 1) \rightarrow$ menjumlahkan dari atas ke bawah proses berhenti saat mencapai angka 1

```

package jobsheet12;
import java.util.Scanner;
public class Tugas3 {
    static int fibonaci(int n) {
        if (n == 1 || n == 2) {
            return 1;
        } else {
            return fibonaci(n - 1) + fibonaci(n - 2);
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int bulan;
        System.out.print("Masukkan bulan ke-: ");
        bulan = sc.nextInt();
        int jumlahPasangan = fibonaci(bulan);
        System.out.println("Jumlah pasangan marmut pada akhir bulan ke-" + bulan + " = " + jumlahPasangan);
    }
}

```

```

Masukkan bulan ke-: 12
Jumlah pasangan marmut pada akhir bulan ke-12 = 144
PS C:\Users\shelfy\OneDrive\Documents\MATA KULIAH\KODINGAN\JOBSHEET\PraktikumDaspro> 

```

artinya, jumlah pasangan marmut pada akhir bulan ke-12 adalah 144 pasang, sesuai dengan tabel ilustrasi.

Base Case:

```
if (n == 1 || n == 2)
```

Recursion Call:

```
fibonacci(n - 1) + fibonacci(n - 2);
```