

PRAKTIKUM DASAR PEMROGRAMAN

“Laporan Pada Jobsheet 12”

Oleh:

Hafiz Rizqi Hernanda_1B

NIM (244107020154)





Jurusan Teknologi informasi

Teknik Informatika

Politeknik Negeri Malang

Percobaan 1

1. Buat project baru bernama Rekursif, dan buat file Java dengan nama Percobaan1

 Percobaan1.java 

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 public class Percobaan1 {
2     public static int faktorialRekursif(int n) {
3         if (n == 0) {
4             return 1;
5         } else {
6             return (n * faktorialRekursif(n - 1));
7         }
8     }
}
```

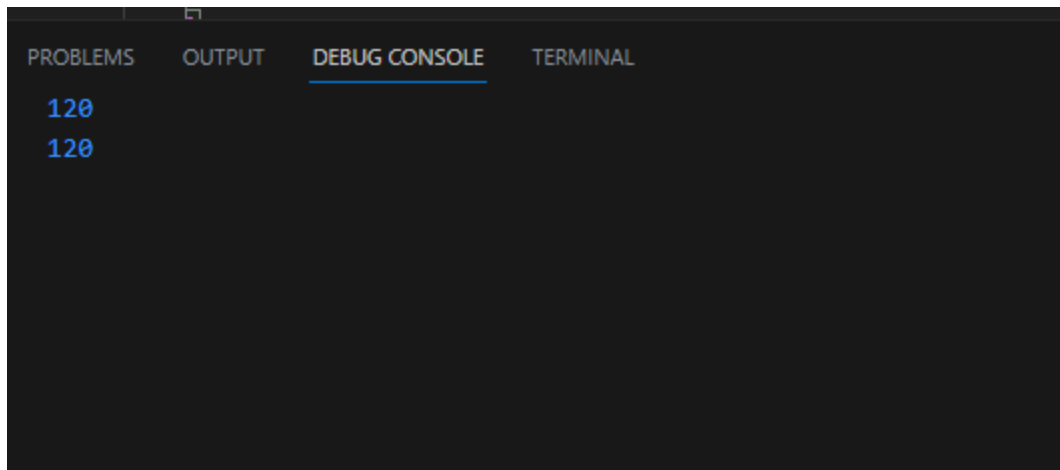
3. Buat lagi fungsi static dengan nama faktorialIteratif(), dengan tipe data kembalian fungsi int dan memiliki 1 parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya.

```
9 public static int faktorialIteratif (int n) {
10     int faktor = 1;
11     for (int i = n; i >= 1; i--) {
12         faktor = faktor * i;
13     }
14     return faktor;
15 }
```

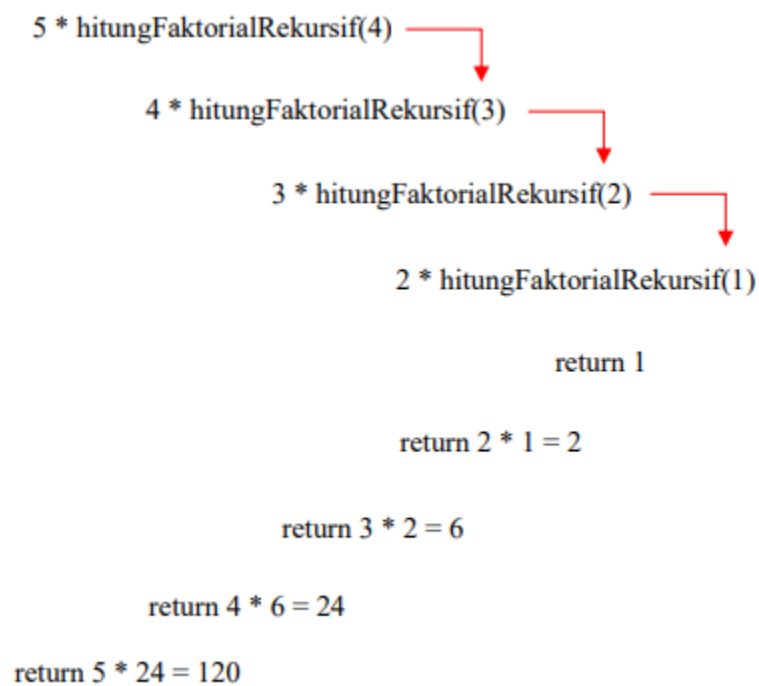
4. Buatlah fungsi main dan lakukan pemanggilan terhadap kedua fungsi yang telah dibuat sebelumnya, dan tampilkan hasil yang didapatkan.

```
Run main | Debug main
16 public static void main(String[] args) {
17     System.out.println(faktorialRekursif(5));
18     System.out.println(faktorialIteratif(5));
19 }
20 }
```

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:



Jawaban Pertanyaan:

1. Apa yang dimaksud dengan fungsi rekursif?
Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri selama proses eksekusi hingga mencapai kondisi tertentu yang dikenal sebagai **base case**. Fungsi ini sering digunakan untuk menyelesaikan masalah yang dapat dipecah menjadi submasalah serupa dengan struktur yang lebih kecil.
2. Bagaimana contoh kasus penggunaan fungsi rekursif ?

```

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

```

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!

Hasil yang di berikan oleh dua buah fungsi sama yaitu 120

Alur fungsi rekursif

faktorialRekursif(5) memanggil faktorialRekursif(4) -> $5 \times (4!)$

faktorialRekursif(4) memanggil faktorialRekursif(3) -> $4 \times (3!)$

Proses ini terus berlanjut hingga faktorialRekursif(0) -> base case (1).

Setelah base case tercapai, nilai dikembalikan secara berjenjang:

- 111 dikembalikan ke faktorialRekursif(1) -> $1 \times 1 = 1$
- 111 dikembalikan ke faktorialRekursif(2) -> $2 \times 1 = 2$, dan seterusnya.

Hasil akhir adalah 120.


Alur fungsi Iteratif

Fungsi ini menggunakan perulangan for untuk menghitung faktorial dengan mengalikan setiap bilangan dari n hingga 1.

1. Variabel faktor diinisialisasi dengan nilai 1.
2. Perulangan berjalan dari $i = n$ hingga $i = 1$.
3. Pada setiap iterasi, nilai faktor diperbarui dengan perkalian faktor * i.
4. Setelah perulangan selesai, nilai akhir faktor adalah hasil faktorial.

Percobaan 2

1. Pada project Rekursif, dan buat file Java dengan nama Percobaan2

 Percobaan2.java >

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.

```
public class Percobaan2 {  
    public 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

```
Kun main | Debug main  
10     public static void main(String[] args) {  
11         Scanner sc = new Scanner(System.in);  
12         System.out.println("Bilangan yang dihitung: ");  
13         int bilangan = sc.nextInt();  
14         System.out.println("Pangkat: ");  
15         int pangkat = sc.nextInt();  
16     }
```

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

```
17         System.out.println(hitungPangkat(bilangan, pangkat));  
18     }
```

7. Jalankan program tersebut. Amati apa yang terjadi!

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  
  
    Bilangan yang dihitung:  
→    2  
  
    Pangkat:  
→    5  
  
    32
```

Jawaban 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!
Proses pemanggilan fungsi `hitungPangkat()` akan terus berjalan hingga mencapai kondisi dasar (base case) yaitu pada nilai parameter `y == 0`.
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh : `hitungPangkat(2,5)` dicetak $2 \times 2 \times 2 \times 2 \times 2 \times 1 = 32$

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

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL


Bilangan yang dihitung:
2

Pangkat:
5

hitungPangkat(2, 5) = 2x2x2x2x2x1 = 32

Percobaan 3

1. Pada project Rekursif, dan buat file Java dengan nama Percobaan3

 Percobaan3.java >

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).

```
1 import java.util.Scanner;
2 public class Percobaan3 {
3     public static double hitungLaba (double saldo, int tahun) {
4         if (tahun == 0) {
5             return (saldo);
6         } else {
7             return (1.11 * hitungLaba(saldo,tahun -1 ));
8         }
9     }
}
```

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

```
Run main | Debug main
10 public static void main(String[] args) {
11     Scanner sc = new Scanner(System.in);
12     System.out.println("Masukkan saldo awal: ");
13     double saldoAwal = sc.nextDouble();
14     System.out.println("Lamanya investasi(Tahun): ");
15     int tahun = sc.nextInt();
16 }
```

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

```
17     System.out.println("Jumlah saldo setelah "+ tahun +" tahun: ");
18     System.out.println(hitungLaba(saldoAwal, tahun));
19 }
20 }
```

7. Jalankan program tersebut. Amati apa yang terjadi!

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Jumlah saldo awal:
→ 1500000

Lamanya investasi(Tahun):
→ 10

Jumlah saldo setelah 10 tahun:
4259131.479103527
```

Jawaban Pertanyaan:

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

Base case

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

Recursion call

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

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

Fase ekspansi

Fungsi memanggil dirinya sendiri sampai base case

1. `hitungLaba(100000,3)`
= $1.11 * \text{hitungLaba}(100000,2)$
2. `hitungLaba(100000,2)`
= $1.11 * \text{hitungLaba}(100000,1)$
3. `hitungLaba(100000,1)`
= $1.11 * \text{hitungLaba}(100000,0)$
4. `hitungLaba(100000,0)`
base case mengembalikan 100000

Fase substitusi

1. $\text{hitungLaba}(100000, 0) = 100000$
2. $\text{hitungLaba}(100000, 1)$
 $= 1.11 * 100000 = 111000$
3. $\text{hitungLaba}(100000, 2)$
 $= 1.11 * 111000 = 123210$
4. $\text{hitungLaba}(100000, 3)$
 $= 1.11 * 123210 = 136763.1$

Hasil Akhir

Jika fungsi dipanggil dengan $\text{hitungLaba}(100000, 3)$, hasil akhirnya adalah:

Jumlah saldo setelah 3 tahun = 136763.1

Tugas

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

```
J DeretDescending11.java > DeretDescending11 > main(String[] args)
1  import java.util.Scanner;
2  public class DeretDescending11 {
3      public static void deretDescendingRekursif(int n) {
4          if (n < 0) {
5              return;
6          }
7          System.out.print(n + " ");
8          deretDescendingRekursif(n - 1);
9      }
10
11     public static void deretDescendingIteratif(int n) {
12         for (int i = n; i >= 0; i--) {
13             System.out.print(i + " ");
14         }
15     }
16     Run main | Debug main
17     public static void main(String[] args) {
18         Scanner sc = new Scanner(System.in);
19
20         System.out.print("Masukkan nilai n: ");
21         int n = sc.nextInt();
22
23         System.out.println("\nHasil menggunakan rekursif:");
24         deretDescendingRekursif(n);
25
26         System.out.println("\n\nHasil menggunakan iteratif:");
27         deretDescendingIteratif(n);
28     }
29 }
30
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

Masukkan nilai n:
→ 7

Hasil menggunakan rekursif:
7 6 5 4 3 2 1 0

Hasil menggunakan iteratif:
7 6 5 4 3 2 1 0

> Please start a debug session to evaluate expressions
```

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).

```
PenjumlahanRekursif11.java > PenjumlahanRekursif11 > main(String[] args)
1  import java.util.Scanner;
2  public class PenjumlahanRekursif11 {
3
4      public static int hitungPenjumlahan(int n) {
5          if (n == 1) {
6              return 1;
7          } else {
8              return n + (hitungPenjumlahan(n - 1));
9          }
10     }
11
12     Run main | Debug main
13     public static void main(String[] args) {
14         Scanner sc = new Scanner(System.in);
15
16         System.out.print("Masukkan nilai f: ");
17         int f = sc.nextInt();
18
19         int hasil = hitungPenjumlahan(f);
20         System.out.println("Hasil penjumlahan dari 1 sampai " + f + " adalah: " + hasil);
21     }
22 }
```

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL
Masukkan nilai f:
→ 8

Hasil penjumlahan dari 1 sampai 8 adalah: 36
```

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

J MarmutFibonacci11.java

```
1  import java.util.Scanner;
2
3  v public class MarmutFibonacci11 {
4
5  v     public static int hitungFibonacci(int bulan) {
6  v         if (bulan == 1 || bulan == 2) {
7             return (1);
8         } else {
9             return (hitungFibonacci(bulan - 1) + hitungFibonacci(bulan - 2));
10        }
11    }
12
13  v     Run main | Debug main
14     public static void main(String[] args) {
15         Scanner scanner = new Scanner(System.in);
16
17         System.out.print("Masukkan bulan ke: ");
18         int bulan = scanner.nextInt();
19
20         int totalPasangan = hitungFibonacci(bulan);
21
22         System.out.println("Jumlah pasangan marmut pada bulan ke-" + bulan + " adalah: " + totalPasangan);
23     }
24 }
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

Masukkan bulan ke-:

→ 12

Jumlah pasangan marmut pada bulan ke-12 adalah: 144