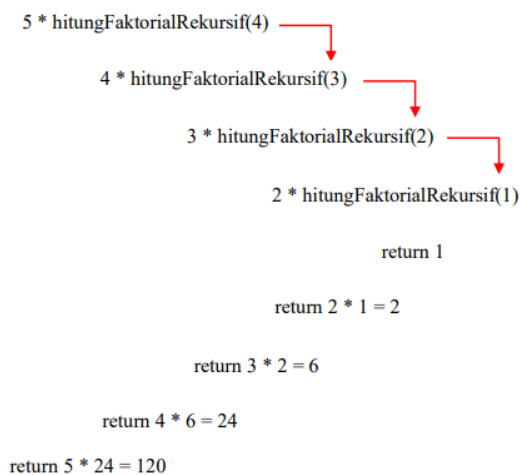


Jobsheet 12

Nama : Nafisah Aliyah K
NIM : 244107060066
Kelas : SIB 1A
Link repo : <https://github.com/pichaachuu/daspro-jobsheet12.git>

Percobaan 1

1. Apa yang dimaksud dengan fungsi rekursif?
⇒ Fungsi yang memanggil fungsi itu sendiri yang dilaksanakan sampai keadaan base case terpenuhi.
2. Pada Percobaan1, apakah hasil yang diberikan fungsi faktorialRekursif() dan fungsi faktorialIteratif() sama? Jelaskan perbedaan alur jalannya program pada penggunaan fungsi rekursif dan fungsi iterative
⇒ Hasilnya sama.
⇒ Alur fungsi rekursif



- ⇒ Alur fungsi iterative
- `I = 5, 5 >= 1? True, 5 = 1*5, i—(5-1=4)`
 - `I = 4, 4 >= 1? True, 20 = 5*4, i—(4-1=3)`
 - `I = 3, 3 >= 1? True, 60 = 20*3, i—(3-1=2)`
 - `I = 2, 2 >= 1? True, 120 = 60*2, i—(2-1=1)`
 - `I = 1, 1 >= 1? True, 120 = 120*1, i—(1-1=0)`
 - `I = 0, I >= 0? False, kembalikan nilai factor`

Percobaan 2

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!
⇒ Input bilangan (2) dan pangkat (5)
⇒ Hitung hasil dengan memanggil fungsi `hitungPangkat`, parameter x diisi dengan bilangan dan parameter y diisi dengan pangkat
⇒ `5 == 0? False, 2 * hitungPangkat(2, 5-1)`

- ⇒ 4 == 0? False, 2 * hitungPangkat(2, 4-1)
 - ⇒ 3 == 0? False, 2 * hitungPangkat(2, 3-1)
 - ⇒ 2 == 0? False, 2 * hitungPangkat(2, 2-1)
 - ⇒ 1 == 0? False, 2 * hitungPangkat(2, 1-1)
 - ⇒ 0, Return 1
 - ⇒ 2 * 2 = 4
 - ⇒ 4 * 2 = 8
 - ⇒ 8 * 2 = 16
 - ⇒ 16 * 2 = 32
2. Tambahkan kode program untuk mencetak deret perhitungan pangkatnya. Contoh: hitungPangkat(2,5) dicetak 2x2x2x2x2x1 = 32


```

1  import java.util.*;;
2  public class Percobaan2 {
3      static int hitungPangkat(int x, int y){
4          if (y==0) {
5              System.out.print("x1");
6              return(1);
7          } else {
8              System.out.print(x);
9              if (y > 1) {
10                 System.out.print("x");
11             }
12             return x * hitungPangkat(x, y-1);
13         }
14     }
15
16     public static void main(String[] args) {
17         Scanner input13 = new Scanner(System.in);
18
19         System.out.print("Bilangan yang dihitung: ");
20         int bilangan = input13.nextInt();
21         System.out.print("Pangkat: ");
22         int pangkat = input13.nextInt();
23
24         System.out.print("Hasil perhitungan: ");
25         int hasil = hitungPangkat(bilangan, pangka
26 t); System.out.println(" = " + hasil);
27         input13.close();
28     }
29 }
30

```

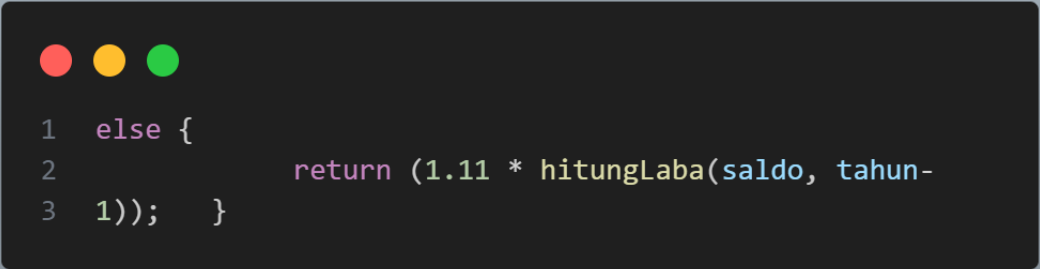
Percobaan 3

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



```
1  if (tahun == 0) {
2      return (saldo);
3  }
```

- Base call



```
1  else {
2      return (1.11 * hitungLaba(saldo, tahun-1));
3  }
```

- Recursion call

2. Jabarkan trace fase ekspansi dan fase substitusi algoritma perhitungan laba di atas jika diberikan nilai `hitungLaba(100000.3)`
 - ⇒ Input `saldoAwal` (100000.3) dan `tahun` (2)
 - ⇒ Memanggil fungsi `hitungLaba`
 - ⇒ Parameter `double` diisi dengan `saldoAwal` dan parameter `tahun` diisi dengan `tahun` yang berada di `main`
 - ⇒ `2 == 0`? False, `1.11 * hitungLaba (100000.3, 2-1)`
 - ⇒ `1 == 0`? False, `1.11 * hitungLaba (100000.3, 1-1)`
 - ⇒ `0 == 0`? True, return `saldo`
 - ⇒ `1.11 * 100000.3 = 111000.333`
 - ⇒ `1.11 * 111000.333 = 123210.36963`

Tugas

```
1 import java.util.*;
2 public class Tugas {
3     public static void main(String[] args) {
4         Scanner input13 = new Scanner(System.in);
5         System.out.print("Masukkan jumlah angka yang ingin dihitung (n): ");
6         int n = input13.nextInt();
7
8         System.out.println("Total dari " + n + " angka yang dimasukkan adalah: " + fungsiIteratif(n,input13));
9         System.out.println("Total dari " + n + " angka yang dimasukkan adalah: " + fungsiRekursif(n, input1
10 3));
11         input13.close();
12     }
13
14     static int fungsiIteratif (int angka, Scanner input13){
15         int total = 0;
16
17         for (int i = angka-1; i >= 0; i--) {
18             System.out.print("Masukkan angka ke-" + (i+1) + ": ");
19             int input = input13.nextInt();
20             total += input;
21         }
22         return total;
23     }
24
25     static int fungsiRekursif (int angka, Scanner input13){
26         if (angka <= 0) {
27             return (0);
28         } else {
29             System.out.print("Masukkan angka ke-" + angka + ": ");
30             int input = input13.nextInt();
31             return input + fungsiRekursif(angka-1, input13);
32         }
33     }
34 }
35
```

```
Masukkan jumlah angka yang ingin dihitung (n): 4
Masukkan angka ke-4: 100
Masukkan angka ke-3: 80
Masukkan angka ke-2: 75
Masukkan angka ke-1: 65
Total dari 4 angka yang dimasukkan adalah: 320
Masukkan angka ke-4: 100
Masukkan angka ke-3: 80
Masukkan angka ke-2: 75
Masukkan angka ke-1: 65
Total dari 4 angka yang dimasukkan adalah: 320
```