PRAKTIKUM

NAMA :AHMAD FALAHI
NIM :244107020152
KELAS :1D
ABSEN :03

Percobaan 1

1. Membuat repository baru dengan nama **rekursif-jobsheet11**. kemudian melakukan cloning repository mengunakan perintah git clone pada terminal.
2. Membuka folder repository menggunakan visual studio code, kemudian membuat file dengan nama **percobaan1.java**
3. Membuat fungsi static dengan nama **faktorialRekursif(),** dengan tipe data kembalian fungsi int dan memiliki satu parameter dengan tipe data int berupa bilangan yang akan dihitung nilai faktorialnya

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

4. Membuat lagi fungsi static dengan nama **faktorialIteratif(),** dengan tipe data kembalikan fungsi int dan memiliki satu 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;
}
```

5. Membuat fungsi main dan melakukan pemanggilan terhadap kedua fungsi yang telah dibuat sebelumnya, dan tampilkan hasil yang telah didapat

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

Petanyaan:

1. Fungsi rekursif adalah fungsi yang memanggil dirinya sendiri sebagai bagian dari proses penyelesaian suatu masalah
2. Contoh kasus penggunaan fungsi rekursif adalah unuk menghitung factorial dan deret Fibonacci
3. fungsi faktorialRekursif() dan faktorialIteratif() menghasilkan hasil yang sama, alur pada fungsi factorialRekursif() fungsi memanggil dirinya sendiri berulang kali hingga mencapai basis rekursi (n == 0). Sedangkan alur pada fungsi faktorialIteratif() fungsi melakukan perulangan sehingga perulangan akan berhenti ketika kondisi false.

Percobaan 2

1. Membuat file baru dengan nama **bercobaan2.java**.
2. Membuat 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.

```
static int hitungPangkat(int x, int y){
    if(y==0){
        return (1);
    }else{
        return (x*hitungPangkat(x,y-1));
    }
}
```

3. Membuat fungsi main dan deklarasi scanner dengan nama sc.

```
Scanner sc= new Scanner(System.in);
```

4. Membuat dua buah variabel bertipe data int dengan nama **bilangan** dan **pangkat.**

```
int bilangan, pangkat;
```

5. Menambahkan kode untuk menginput nilai.

```
System.out.print(s:"Bilangan yang dihitung: ");
bilangan = sc.nextInt();
System.out.print(s:"Pangkat: ");
pangkat = sc.nextInt();
```

6. Melakukan panggilan fungsi hitungpangkat yang telah dibuat sebelumnya dengan mengirimkan dua nilai parameter.

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

7. Melakukan compile dan run sehingga menghasilkan output berikut.

```
Bilangan yang dihitung: 2
Pangkat: 5
32
```

Pertanyaan:

1. Proses pemanggilan fungsi rekursif hitungPangkat(bilangan, pangkat) akan terus dilakukan hingga mencapai kondisi base case (pangkat = 0).
2. Modifikasi kode :

```
static void cetakDeretPangkat(int basis, int eksponen) {
    StringBuilder deret = new StringBuilder();
    for (int i = 0; i < eksponen; i++) {
        deret.append(basis);
        if (i < eksponen - 1) {
            deret.append(str:"x");
        }
    }
    int hasil = (int) Math.pow(basis, eksponen);
    System.out.println( deret.toString() + " = " + hasil);
}
System.out.print(s:"Deret perhitungan pangkatnya:");
cetakDeretPangkat(bilangan, pangkat);
```

Output:

```
Bilangan yang dihitung: 3
Pangkat: 4
81
Deret perhitungan pangkatnya:3x3x3x3 = 81
```

Percobaan 3

1. Membuat file baru dengan nama **percobaan3.**

2. Membuat fungsi static dengan nama **hitunglaba(),** dengan tipe data kembalian fungsi **double** dan memiliki 2 parameter dengan tipe data int berupa saldoAwal dan tahun.

```
static double hitungLaba(double saldoAwal, double tahun){
    if(tahun==0){
        return (saldoAwal);
    }else{
        return (1.11*hitungLaba(saldoAwal, tahun-1));
    }
}
```

3. Membuat fungsi main dan deklarasi scanner dengan nama sc.

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

4. Membuat sebuah variabel bertipe double dengan nama saldoAwal dan sebuah variable bertipe int dengan nama tahun.

```
double saldoAwal;
int tahun;
```

5. Menambahkan kode untuk menginput nilai saldoAwal dan tahun.

```
System.out.print(s:"Masukkan saldo awal: ");
saldoAwal=sc.nextDouble();
System.out.print(s:"lamanya investasi (tahun): ");
tahun=sc.nextInt();
```

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

```
System.out.print("jumlah saldo setelah "+tahun+" tahun adalah: ");
System.out.println(hitungLaba(saldoAwal, tahun));
```

Percobaan:

1. Base case: if (tahun == 0) { return saldoAwal; }
   Resurcion call: return 1.11 * hitungLaba(saldoAwal, tahun - 1);
2. Face ekspansi:
   1.11 * hitungLaba(10000, 2)
   1.11 * hitungLaba(10000, 1)
   1.11 * hitungLaba(10000, 0)
   hasil = 100000
   Fase substitusi:
   hitungLaba(100000,0)=100000
   hitungLaba(100000,1)=1.11×100000=111000
   hitungLaba(100000,2)=1.11×111000=123210
   hitungLaba(100000,3)=1.11×123210=136363.1

Tugas

1. Kode program:

```java
import java.util.Scanner;

public class Tugas1 {
    public static void deretDescendingRekursif(int n) {
        if (n < 0) {
            return;
        }
        System.out.print(n+" ");
        deretDescendingRekursif(n - 1);
    }

    public static void deretDescendingIteratif(int n) {
        for (int i = n; i >= 0; i--) {
            System.out.print(i);
            System.out.print(s:" ");
        }
    }

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int n;
        System.out.print(s:"Masukkan nilai n: ");
        n=sc.nextInt();

        System.out.print(s:"Menggunakan Rekursif: ");
        deretDescendingRekursif(n);
        System.out.println();

        System.out.print(s:"Menggunakan Iteratif: ");
        deretDescendingIteratif(n);
    }
}
```

Output:

```
Masukkan nilai n: 6
Menggunakan Rekursif: 6 5 4 3 2 1 0
Menggunakan Iteratif: 6 5 4 3 2 1 0
```

2. Kode program:

```java
import java.util.Scanner;

public class Tugas2 {
    public static int penjumlahanRekursif(int n, StringBuilder deret) {
        if (n == 1) {
            deret.append(str:"1");
            return 1;
        }
        deret.append(n).append(str:"+");
        return n + penjumlahanRekursif(n - 1, deret);
    }

    Run | Debug | Run main | Debug main
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int f;
        System.out.print(s:"Masukkan bilangan: ");
        f = sc.nextInt();
        StringBuilder deret = new StringBuilder();
        int hasil = penjumlahanRekursif(f, deret);

        if (deret.length() > 0) {
            deret.setLength(deret.length() - 1);
        }

        System.out.println("Penjumlahan dari 1 hingga " + f + " adalah: " + deret+"1" + "=" + hasil);
    }
}
```

Output:

```
Masukkan bilangan: 9
Penjumlahan dari 1 hingga 9 adalah: 9+8+7+6+5+4+3+2+1=45
```

3. Kode program:

```java
import java.util.Scanner;

public class Tugas3 {
    public static int hitungPasangan(int bulan) {
        if (bulan == 1 || bulan == 2) {
            return 1;
        }
        return hitungPasangan(bulan - 1) + hitungPasangan(bulan - 2);
    }


    public static void main(String[] args) {
        Scanner sc=new Scanner(System.in);
        int jumlahBulan;
        System.out.print(s:"Masukkan jumlah bulan: ");
        jumlahBulan = sc.nextInt();
        int jumlahPasangan = hitungPasangan(jumlahBulan);

        System.out.println("Jumlah pasangan marmut pada akhir bulan ke-" + jumlahBulan + " adalah " + jumlahPasangan
    }
}
```

Output:

```
Masukkan jumlah bulan: 13
Jumlah pasangan marmut pada akhir bulan ke-13 adalah 233
```