

JOBSHEET 12

Nama : Muhammad Ibnu Zauzi

NIM : 24410720016

ABSEN : 18

KELAS : 1D

Percobaan 1

1. Buat project baru bernama Rekursif, dan 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.

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

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.

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

5. Jalankan program tersebut. Amati apa yang terjadi!

```
120
120
```

Pertanyaan

1. Fungsi rekursif adalah **fungsi yang memanggil dirinya sendiri** selama proses eksekusinya untuk menyelesaikan masalah yang lebih kecil dari masalah awal
2. fungsi faktorialRekursif digunakan untuk menghitung faktorial sebuah bilangan n. Faktorial didefinisikan sebagai hasil perkalian bilangan bulat positif dari 1 hingga n.
3. Iya, hasil yang diberikan oleh faktorialRekursif() dan faktorialIteratif() adalah sama. Perbedaannya faktorialRekursif memanggil dirinya sendiri hingga kondisi dasar tercapai. FaktorialIteratif Menggunakan perulangan (loop) untuk menyelesaikan tugas

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.

```
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

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

4. Buatlah dua buah variabel bertipe int dengan nama bilangan dan pangkat

```
int bilangan, pangkat;
```

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

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

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

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

7. Jalankan program tersebut. Amati apa yang terjadi!

```
Bilangan yang dihitung: 8
Pangkat: 2
64
```

Pertanyaan

1. Proses pemanggilan fungsi hitungPangkat akan terus berjalan hingga mencapai kondisi dasar yang ditentukan dalam fungsi, yaitu ketika $y=0$.

```
static int hitungPangkat(int x, int y) {
    if (y == 0) {
        System.out.print(s:"1");
        return 1;
    } else {
        System.out.print(x + "x");
        return x * hitungPangkat(x, y - 1);
    }
}

Run | Debug | Tabnine | Edit | Test | Fix | Explain | Document | Ask
public static void main(String[] args) {
    int bilangan, pangkat;
    Scanner sc = new Scanner(System.in);

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

    System.out.print(s:"Deret perhitungan: ");
    int hasil = hitungPangkat(bilangan, pangkat);
    System.out.println(" = " + hasil);
}
```

```
Bilangan yang dihitung: 2
Pangkat: 5
Deret perhitungan: 2x2x2x2x2x1 = 32
```

- 2.

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)

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

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc

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

4. Buatlah sebuah variabel bertipe double dengan nama saldoAwal dan sebuah variabel bertipe int bernama tahun

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

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

```
System.out.print(s:"Jumlah saldo awal : ");
saldoAwal = scanner.nextInt();
System.out.print(s:"Lamanya investasi (tahun) : ");
tahun = scanner.nextInt();
```

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

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

7. Jalankan program tersebut. Amati apa yang terjadi!

```
Jumlah saldo awal : 500
Lamanya investasi (tahun) : 3
Jumlah saldo setelah 3 tahun : 683.8155000000002
```

Pertanyaan

1. **Base Case:** Base case adalah kondisi yang menghentikan rekursi.

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

Recursion Call: Recursion call adalah bagian di mana fungsi memanggil dirinya sendiri

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

2. **Fase Ekspansi (Pemanggilan Rekursif)**

- hitungLaba(100000, 3) memanggil 1.11 * hitungLaba(100000, 2).
- hitungLaba(100000, 2) memanggil 1.11 * hitungLaba(100000, 1).
- hitungLaba(100000, 1) memanggil 1.11 * hitungLaba(100000, 0).

- hitungLaba(100000, 0) adalah base case, mengembalikan 100000.
 - **Fase Substitusi (Penghitungan Balik)**
 - hitungLaba(100000, 0) = 100000.
 - hitungLaba(100000, 1) = 1.11 * 100000 = 111000.
 - hitungLaba(100000, 2) = 1.11 * 111000 = 123210.
 - hitungLaba(100000, 3) = 1.11 * 123210 = 136761.
- Hasil Akhir: 136761.**

TugasFase Ekspansi (Pemanggilan Rekursif)

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

Tabnine | Edit | Test | Explain | Document | Ask

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

Run | Debug | Tabnine | Edit | Test | Fix | Explain | Document | Ask

public static void main(String[] args) {
    int n = 5;
    System.out.println(x:"Hasil menggunakan rekursif:");
    deretDescendingRekursif(n);

    System.out.println(x:"\nHasil menggunakan iteratif:");
    deretDescendingIteratif(n);
}
```

1.

```
Hasil menggunakan rekursif:
5 4 3 2 1 0
Hasil menggunakan iteratif:
5 4 3 2 1 0
```

```

public static int penjumlahanRekursif(int f) {
    if (f == 0){
        return 0;
    } else {
        return f + penjumlahanRekursif(f - 1);
    }
}

Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
public static void main(String[] args) {
    int f = 8;
    int hasil = penjumlahanRekursif(f);
    System.out.println("Hasil penjumlahan dari 1 sampai " + f + " adalah: " + hasil);
}

```

2.

Hasil penjumlahan dari 1 sampai 8 adalah: 36

```

public static int fibonacci(int bulan) {
    if (bulan == 1 || bulan == 2) {
        return 1;
    } else {
        return fibonacci(bulan - 1) + fibonacci(bulan - 2);
    }
}

Run | Debug | Tabnine | Edit | Test | Explain | Document | Ask
public static void main(String[] args) {
    int bulan = 12;
    int totalPasangan = fibonacci(bulan);
    System.out.println("Jumlah total pasangan marmut pada bulan ke-" + bulan + " adalah: " + totalPasangan);
}

```

3.

Jumlah total pasangan marmut pada bulan ke-12 adalah: 144