**LAPORAN DASPRO**

**JOBSHET 12**

**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.

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

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

5. Jalankan program tersebut. Amati apa yang terjadi!

public class percobaan1 {

    static int faktorialRekursif(int n) {

        if (n == 0) {

            return (1);

        } else {

            return (n + faktorialRekursif(n -1));

        }

    }

    static int faktorialIteratif(int n) {

        int faktor = 1;

        for (int i = n; i >= 1; i--) {

            faktor = faktor \* i;

        }

        return faktor;

    }

    public static void main(String[] args) {

        System.out.println(faktorialRekursif(5));

        System.out.println(faktorialIteratif(5));

    }

}

OUTPUT



**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.
2. Contoh kasus penggunaan fungsi rekursif

if (n == 0) {

return 1;

} else { return n \* faktorial(n - 1);

1. Fungsi faktorialRekursif() memiliki kesalahan tidak menghitung faktorial, tetapi menghitung jumlah bilangan dari nnn hingga 1 (deret aritmatika)

5+4+3+2+1=15

Fungsi faktorialIteratif() benar-benar menghitung faktorial karena menggunakan operasi perkalian pada setiap iteras

5!=5×4×3×2×1=120

**PERCOBAAN 2**

1. Pada project Rekursif, dan buat file Java 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.

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc

4. Buatlah dua buah variabel bertipa 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

7. Jalankan program tersebut. Amati apa yang terjadi!

import java.util.Scanner;

public class Percobaan2 {

    static int hitungPangkat(int x, int y) {

        if (y == 0) {

            return (1);

        } else {

            return (x \* hitungPangkat(x, y - 1));

        }

    }

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int bilangan, pangkat;

        System.out.print("Bilangan yang akan dihitung: ");

        bilangan = sc.nextInt();

        System.out.print("Pangkat: ");

        pangkat = sc.nextInt();

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

    }

}

OUTPUT



**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 2x2x2x2x2x1 = 32

**JAWABAN**

1. Proses pemanggilan fungsi rekursif hitungPangkat(bilangan, pangkat) akan terus dijalankan sampai mencapai **kondisi dasar** yang ditentukan dalam kode:

if (y == 0) {

return 1;

}

Yang mana apabila Ketika nilai pangkat (y) menjadi **0**, fungsi tidak akan memanggil dirinya sendiri lagi.

1. import java.util.Scanner;
2. public class percobaan2\_soal {
3. static String deret = "";
4. static int hitungPangkat(int x, int y) {
5. if (y == 0) {
6. deret += "1";
7. return (1);
8. } else {
9. deret += x + "x";
10. return x \* hitungPangkat(x, y - 1);
11. }
12. }
13. public static void main(String[] args) {
14. Scanner sc = new Scanner(System.in);
15. int bilangan, pangkat;
16. System.out.print("Bilangan yang akan dihitung: ");
17. bilangan = sc.nextInt();
18. System.out.print("Pangkat: ");
19. pangkat = sc.nextInt();
20. int hasil = hitungPangkat(bilangan, pangkat);
21. System.out.println(deret + " = " + hasil);
22. }
23. }

**PERCOBAAN 3**

1. Pada project Rekursif, dan buat file Java 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)

3. Buatlah fungsi main dan deklarasikan Scanner dengan nama sc

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

5. Tambahkan kode berikut ini untuk menerima input dari keyboard

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

import java.util.Scanner;

public class percobaan3 {

    static double hitunglaba(double saldo, int tahun) {

    if (tahun == 0) {

        return (saldo);

    }else {

        return (1.11 \* hitunglaba(saldo, tahun - 1));

    }

}

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        double saldoAwal;

        int tahun;

        System.out.print("jumlah saldo awal : ");

        saldoAwal = sc.nextInt();

        System.out.print("Lamanya ivestasi (tahun) : ");

        tahun = sc.nextInt();

        System.out.println("Jumlah saldo setelah " + tahun + " tahun : ");

        System.out.print(hitunglaba(saldoAwal, tahun));

    }

}

**PERTANYAAN**

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

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

**JAWABAN**

1. Base casenya terdapat pada di mana rekursi berhenti ketika tahun == 0

Sedangkan Recursion Callnya terdapat pada return (1.11 \* hitunglaba(saldo, tahun - 1))

1. Fase Ekspansi

 hitunglaba(100000, 3) → 1.11×hitunglaba(100000,2)1.11 \times \text{hitunglaba}(100000, 2)1.11×hitunglaba(100000,2)

 hitunglaba(100000, 2) → 1.11×hitunglaba(100000,1)1.11 \times \text{hitunglaba}(100000, 1)1.11×hitunglaba(100000,1)

 hitunglaba(100000, 1) → 1.11×hitunglaba(100000,0)1.11 \times \text{hitunglaba}(100000, 0)1.11×hitunglaba(100000,0)

 hitunglaba(100000, 0) → 100000100000100000 (Base case)

Fase Ekspansi

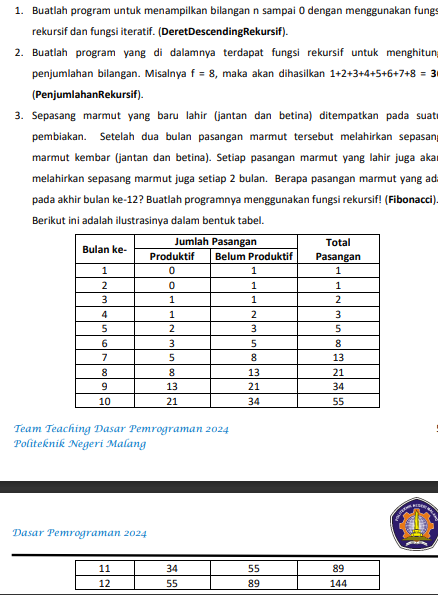
 hitunglaba(100000, 0) → 100000100000100000

 hitunglaba(100000, 1) → 1.11×100000=1110001.11 \times 100000 = 1110001.11×100000=111000

 hitunglaba(100000, 2) → 1.11×111000=1232101.11 \times 111000 = 1232101.11×111000=123210

 hitunglaba(100000, 3) → 1.11×123210=136753.11.11 \times 123210 = 136753.11.11×123210=136753.1

**TUGAS**



**JAWABAN**

public class DeretTurun {

    public static void deretTurunRekursif(int n) {

        if (n < 0) return;

        System.out.print(n + " ");

        deretTurunRekursif(n - 1);

    }

    public static void deretTurunIteratif(int n) {

        for (int i = n; i >= 0; i--) {

            System.out.print(i + " ");

        }

    }

    public static void main(String[] args) {

        int angka = 5;

        System.out.println("Rekursif:");

        deretTurunRekursif(angka);

        System.out.println("\nIteratif:");

        deretTurunIteratif(angka);

    }

}

public class PenjumlahanRekursif {

    public static int jumlahRekursif(int n) {

        if (n == 1) return 1;

        return n + jumlahRekursif(n - 1);

    }

    public static void main(String[] args) {

        int angka = 5;

        System.out.println("Penjumlahan dari 1 sampai " + angka + " adalah: " + jumlahRekursif(angka));

    }

}

public class Marmut {

    public static int hitungFibonacci(int bulan) {

        if (bulan == 1 || bulan == 2) return 1;

        return hitungFibonacci(bulan - 1) + hitungFibonacci(bulan - 2);

    }

    public static void main(String[] args) {

        int bulanKe = 12;

        System.out.print("Deret Fibonacci hingga bulan ke-" + bulanKe + ": ");

        for (int i = 1; i <= bulanKe; i++) {

            System.out.print(hitungFibonacci(i) + " ");

        }

    }

}