**LAPORAN TUGAS PRAKTIKUM   
KONSEP DASAR PEMROGRAMAN**



**Oleh:  
FAWWAZ ALIFIO FARSA  
NIM. 2341720128  
TI-1E / 10**

**D-IV TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI  
POLITEKNIK NEGERI MALANG**

**Praktikum Percobaan 1**

1. Membuat program untuk menghitung nilai akhir mahasiswa.  
   Output:  
   

import java.util.Scanner;

public class Nilai {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        int tugas, kuis, uts, uas;

        double nilaiAkhir;

        String keterangan = "";

        String[] nilaiHuruf = { "A", "B+", "B", "C+", "C", "D", "E" };

        String[] Kualifikasi = { "Sangat Baik", "Lebih dari Baik", "Baik", "Lebih dari Cukup", "Cukup", "Kurang",

                "Gagal" };

        System.out.print("Masukkan Nilai Tugas: ");

        tugas = input.nextInt();

        System.out.print("Masukkan Nilai Kuis: ");

        kuis = input.nextInt();

        System.out.print("Masukkan Nilai UTS: ");

        uts = input.nextInt();

        System.out.print("Masukkan Nilai UAS: ");

        uas = input.nextInt();

        if (tugas >= 0 && tugas <= 100

                && kuis >= 0 && kuis <= 100

                && uts >= 0 && uts <= 100

                && uas >= 0 && uas <= 100) {

            nilaiAkhir = (tugas \* 0.2) + (kuis \* 0.2) + (uts \* 0.3) + (uas \* 0.3);

            if (nilaiAkhir <= 100 && nilaiAkhir > 80) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[0]);

                keterangan = nilaiHuruf[0];

            } else if (nilaiAkhir <= 80 && nilaiAkhir > 73) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[1]);

                keterangan = nilaiHuruf[1];

            } else if (nilaiAkhir <= 73 && nilaiAkhir > 65) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[2]);

                keterangan = nilaiHuruf[2];

            } else if (nilaiAkhir <= 65 && nilaiAkhir > 60) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[3]);

                keterangan = nilaiHuruf[3];

            } else if (nilaiAkhir <= 60 && nilaiAkhir > 50) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[4]);

                keterangan = nilaiHuruf[4];

            } else if (nilaiAkhir <= 50 && nilaiAkhir > 39) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[5]);

                keterangan = nilaiHuruf[5];

            } else if (nilaiAkhir <= 39) {

                System.out.println("Nilai Angka: " + nilaiAkhir);

                System.out.println("Nilai Huruf: " + nilaiHuruf[6]);

                keterangan = nilaiHuruf[6];

            }

            if (keterangan.equals("D") || keterangan.equals("E")) {

                System.out.println("Keterangan: Tidak Lulus");

            } else {

                System.out.println("Keterangan: Lulus");

            }

        } else {

            System.out.println("Nilai tidak valid");

        }

    }

}

**Jawaban Pertanyaan Praktikum Percobaan 1**

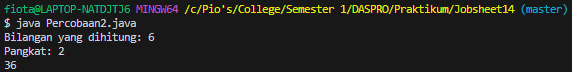
1. Fungsi rekursif merupakan suatu konsep di mana suatu fungsi dapat memanggil dirinya sendiri selama proses eksekusi. Dalam hal ini, fungsi tersebut memecahkan masalah menjadi submasalah yang lebih kecil dan menggunakan hasil dari submasalah tersebut untuk memecahkan masalah secara keseluruhan.
2. Contoh penggunaan fungsi rekursif adalah dalam perhitungan deret Fibonacci. Deret Fibonacci merupakan deret bilangan yang setiap elemennya merupakan penjumlahan dari dua elemen sebelumnya. Dalam hal ini, kita dapat membuat fungsi rekursif untuk menghitung nilai deret Fibonacci ke-n.
3. **Faktorial Rekursif**: Fungsi ini menggunakan rekursi, yang berarti fungsi tersebut memanggil dirinya sendiri dengan parameter yang lebih kecil. Alur jalannya program pada fungsi rekursif dapat dijelaskan sebagai berikut:

* **faktorialRekursif(5)** memanggil **faktorialRekursif(4),** yang kemudian memanggil **faktorialRekursif(3)**, dan seterusnya hingga mencapai **faktorialRekursif(0)**.
* Ketika n menjadi 0, fungsi mulai mengembalikan nilai 1.
* Nilai-nilai ini kemudian dikalikan bersama-sama saat keluar dari panggilan rekursif hingga mencapai nilai faktorial untuk **faktorialRekursif(5)**.

**Faktorial Iteratif**: Fungsi ini menggunakan pendekatan iteratif dengan bantuan perulangan for. Alur jalannya program pada fungsi iteratif dapat dijelaskan sebagai berikut:

* Sebuah perulangan for digunakan untuk mengalikan nilai-nilai dari n hingga 1.
* Setiap iterasi, nilai faktorial (faktor) dikalikan dengan nilai iterasi saat ini.
* Setelah perulangan selesai, nilai faktorial dikembalikan.

**Praktikum Percobaan 2**

1. Buat file baru dengan nama **Percobaan2.java** dan tuliskan kode program sebagai berikut.  
     
   Output:  
   

package Jobsheet14;

import java.util.Scanner;

public class Percobaan2 {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        int bilangan, pangkat;

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

        bilangan = sc.nextInt();

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

        pangkat = sc.nextInt();

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

    }

    static int hitungPangkat(int x, int y) {

        if (y == 0) {

            return (1);

        } else {

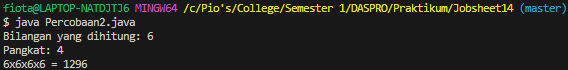
            return (x \* hitungPangkat(x, y - 1));

        }

    }

}

**Jawaban Pertanyaan Praktikum Percobaan 2**

1. Proses pemanggilan fungsi hitungPangkat(x, y) akan dijalankan hingga kondisi berhenti yang didefinisikan oleh base case pada fungsi rekursif tercapai. Base case dalam hal ini adalah ketika nilai y (pangkat) menjadi 0.
2. Memodifikasi program di atas untuk mencetak deret perhitungan pangkatnya.  
   Output:  
   

    static int hitungPangkat(int x, int y) {

        if (y == 0) {

            return (1);

        } else { System.out.print(x);

            if (y > 1) {

                System.out.print("x");

            }

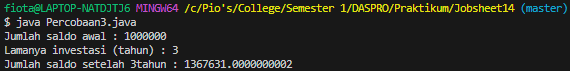
            System.out.print((y == 1) ? " = " : "");

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

        }

    }

**Praktikum Percobaan 3**

1. Membuat file baru dengan nama **Percobaan3.java** dan menuliskan kode program sebagai berikut.  
   Output:  
   

package Jobsheet14;

import java.util.Scanner;

public class Percobaan3 {

    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 investasi (tahun) : ");

        tahun = sc.nextInt();

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

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

    }

    static double hitungLaba(double saldo, int tahun) {

        if (tahun == 0) {

            return (saldo);

        } else {

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

        }

    }

}

**Jawaban Pertanyaan Praktikum Percobaan 3**

1. **Base Case:** Kode **if (tahun == 0)** merupakan base case. Pada saat **tahun** sudah mencapai 0, fungsi mengembalikan nilai **saldo** tanpa melakukan pemanggilan rekursif lagi.

**Recursion Call:** Kode **return 1.11 \* hitungLaba(saldo, tahun - 1);** merupakan pemanggilan rekursif. Fungsi ini memanggil dirinya sendiri dengan nilai **tahun** yang dikurangi 1, sehingga membentuk rekursi hingga base case tercapai.

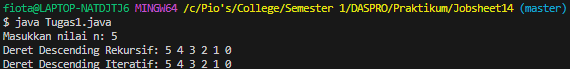
1. **Fase Ekspansi:**

* **hitungLaba(100000, 3)** memanggil **1.11 \* hitungLaba(100000, 2) 1.11 \***
* **hitungLaba(100000, 2)** memanggil **1.11 \* hitungLaba(100000, 1) 1.11 \***
* **hitungLaba(100000, 1)** memanggil **1.11 \* hitungLaba(100000, 0) 1.11 \***
* **hitungLaba(100000, 0)** mencapai base case dan mengembalikan nilai **100000.0**

**Fase Substitusi:**

* Substitusi nilai-nilai dari hasil ekspansi ke dalam ekspresi:  
  **1.11 \* (1.11 \* (1.11 \* 100000.0))**
* Hitung nilai ekspresi secara keseluruhan untuk mendapatkan hasil akhir: **1.11 \* 1.11 \* 1.11 \* 100000.0 = 136763.61100000002**

**Tugas**

1. Membuat class dengan nama **DeretDescendingRekursif.java** dan menuliskan kode program untuk menampilkan bilangan **n** sampai 0 dengan fungsi rekursif dan iteratif.  
   Output:  
   

package Jobsheet14;

import java.util.Scanner;

public class DeretDescendingRekursif {

     public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan nilai n: ");

        int n = sc.nextInt();

        System.out.print("Deret Descending Rekursif: ");

        descendingRekursif(n);

        System.out.print("\nDeret Descending Iteratif: ");

        descendingIteratif(n);

    }

    static void descendingRekursif(int n) {

        if (n >= 0) {

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

            descendingRekursif(n - 1);

        }

    }

    static void descendingIteratif(int n) {

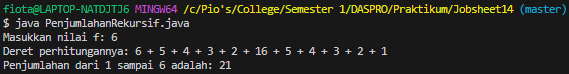
        for (int i = n; i >= 0; i--) {

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

        }

    }

}

1. Membuat class dengan nama **PenjumlahanRekursif.java** yang di dalamnya terdapat fungsi rekursif untuk menghitung penjumlahan bilangan.  
   Output:  
   

package Jobsheet14;

import java.util.Scanner;

public class PenjumlahanRekursif {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan nilai f: ");

        int f = sc.nextInt();

        System.out.print("Deret perhitungannya: ");

        penjumlahanRekursif(f);

        int hasil = penjumlahanRekursif(f);

        System.out.println("\nPenjumlahan dari 1 sampai " + f + " adalah: " + hasil);

    }

    static int penjumlahanRekursif(int n) {

        if (n == 1) {

            System.out.print("1");

            return 1;

        } else {

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

            return n + penjumlahanRekursif(n - 1);

        }

    }

}

1. Membuat class baru dengan nama **CekPrimaRekursif.java** yang di dalamnya terdapat fungsi rekursif untuk mengecek apakah suatu bilangan **n** merupakan bilangan prima atau bukan.  
   Output:  
   

package Jobsheet14;

import java.util.Scanner;

public class CekPrimaRekursif {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        System.out.print("Masukkan sebuah bilangan: ");

        int n = sc.nextInt();

        if (cekPrimaRekursif(n, 2)) {

            System.out.println(n + " adalah bilangan prima.");

        } else {

            System.out.println(n + " bukan bilangan prima.");

        }

    }

    static boolean cekPrimaRekursif(int n, int i) {

        if (n <= 1) {

            return false;

        } else if (i == n) {

            return true;

        } else if (n % i == 0) {

            return false;

        } else {

            return cekPrimaRekursif(n, i + 1);

        }

    }

}

1. Membuat class baru dengan nama **Fibonacci.java** dan menuliskan program untuk menampilkan jumlah pasangan marmot yang ada pada akhir bulan ke-12 dengan pola fibonacci.  
   Output:  
   

package Jobsheet14;

public class Fibonacci {

    public static void main(String[] args) {

        int month = 12;

        int pairs = fibonacci(month);

        for (int i = 0; i < month; i++) {

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

        }

        System.out.println("\nJumlah pasangan marmut pada bulan ke-" + month + " adalah " + pairs);

    }

    public static int fibonacci(int n) {

        if (n <= 1) {

            return n;

        }

        return fibonacci(n - 1) + fibonacci(n - 2);

    }

}