

Praktikum **Algoritma dan Pemrograman II**

PENGENALAN ALGORITMA

5002221053 - Putri Ghaida Tsuroyya

5002221072 - Mohammad Febryan Khamim

Kelompok 5



Outline Pembahasan

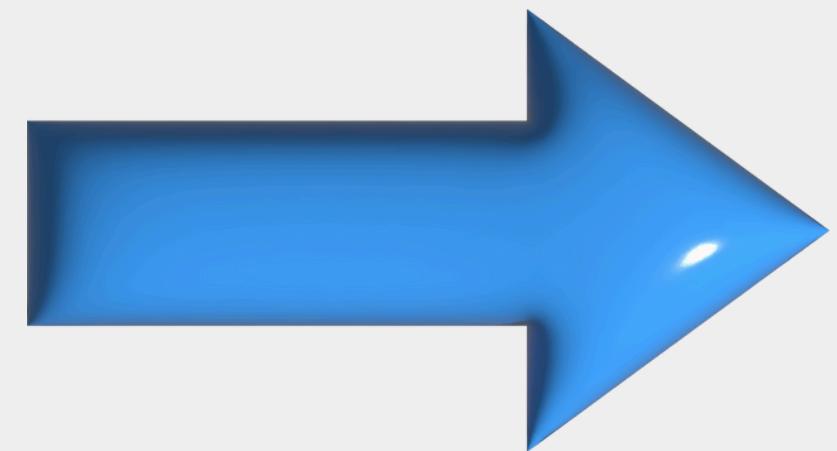
- Analisis Algoritma
- *Correctness of Algorithm*
- Metode Pembuktian Kebenaran Algoritma
- Latihan Soal

Analisis Algoritma

Kelompok 5

Mengapa analisis algoritma penting?

Suatu program
membutuhkan



Program yang bagus?

EFEKTIF & EFISIEN!

Correctness of Algorithm

Kelompok 5

Correctness of Algorithm

Correctness of Algorithm merupakan suatu Teknik dalam melakukan analisis terhadap suatu algoritma.



Valid



Invalid

```
package asdosalpro2;
import java.util.Scanner;
public class AsdosAlpro2 {
    public static void main(String[] args) {
        /**
         * Jika usiamu 30 tahun atau lebih muda, mendapat point 8
         * Jika usiamu antara 31 sampai 39 tahun, mendapat point 7
         * Jika usiamu antara 40 sampai 59 tahun, mendapat point 6
         * Jika usiamu 60 tahun atau lebih tua, mendapat point 4
        */
        int age, point;
        Scanner in = new Scanner(System.in);
        System.out.print("Masukkan umur Anda : ");
        age = in.nextInt();

        if (age <= 30)
            point = 8;
        if (age < 40)
            point = 7;
        if (age < 60)
            point = 6;
        else
            point = 4;

        System.out.println("Nilai point adalah : " + point);
    }
}
```

```
package asdosalpro2;
import java.util.Scanner;
public class AsdosAlpro2 {
    public static void main(String[] args) {
        /**
         * Jika usiamu 30 tahun atau lebih muda, mendapat point 8
         * Jika usiamu antara 31 sampai 39 tahun, mendapat point 7
         * Jika usiamu antara 40 sampai 59 tahun, mendapat point 6
         * Jika usiamu 60 tahun atau lebih tua, mendapat point 4
        */
        int age, point;
        Scanner in = new Scanner(System.in);
        System.out.print("Masukkan umur Anda : ");
        age = in.nextInt();
        if (age <= 30)
            point = 8;
        if (age < 40)
            point = 7;
        if (age < 60)
            point = 6;
        else
            point = 4;
        System.out.println("Nilai point adalah : " + point);
    }
}
```

Terlalu banyak menggunakan if

Urutan pengecekan salah



Metode Pembuktian Kebenaran Algoritma

Kelompok 5

Induksi Matematika

Induksi Matematika adalah sebuah teknik untuk membuktikan berbagai macam *statement*, teorema, atau preposisi dalam matematika.

- **Basis Induksi**

Buktikan bahwa algoritma benar untuk nilai terkecil ($n = 1$)

- **Hipotesis**

Mengasumsikan bahwa untuk ($n=k$) **benar**

- **Langkah Induksi**

Buktikan algoritma benar untuk suatu nilai ($n = k+1$)

Induksi Matematika

Example

```
public static int hitungderet(int n) {  
    return (n * (n + 1)) / 2;  
}
```

Induksi Matematika

Example

```
public static boolean ceklagi(int n) {
    if(Math.sqrt(12*n) < Math.pow(n, 2)) {
        return true;
    }
    else{
        return false;
    }
}
```

Untuk $n > 2$

Invarian Loop

Invarian Loop adalah pernyataan logis yang selalu benar sebelum dan setelah setiap iterasi loop dalam program.

Kelompok 5

- **inisiasi**

Buktikan bahwa **loop invariant** benar sebelum loop mulai berjalan.

- **Maintenance**

Buktikan bahwa jika **loop invariant** benar di suatu iterasi, maka tetap benar di iterasi berikutnya.

- **Terminasi**

Buktikan bahwa ketika loop berhenti, loop invariant memberikan hasil yang benar sesuai spesifikasi algoritma.

Invarian Loop

Example

```
public class SumNumbers {  
    public static void main(String[] args) {  
        int n = 5;  
        int sum = 0;  
        int i = 1;  
  
        while (i <= n) {  
            sum = sum + i;  
            i++;  
        }  
  
        System.out.println("Jumlah dari 1 sampai " + n + " adalah: " + sum);  
    }  
}
```

Metode Formal

Pembuktian formal digunakan untuk menunjukkan bahwa suatu algoritma benar dengan menggunakan logika matematis dan notasi formal.

- **Preconditions**

Keadaan yang harus benar sebelum algoritma dijalankan.

- **Postconditions**

Keadaan yang harus benar setelah algoritma selesai dijalankan.

Summary

Metode	Fokus	Cakupan	Contoh
Induksi Matematika	Membuktikan kebenaran algoritma berbasis pola rekursi atau iterasi menggunakan langkah dasar dan induksi	Algoritma yang memiliki pola rekursi atau perulangan	Faktorial, Fibonacci, Jumlah deret, dan sebagainya
Loop Invariant	Membuktikan kebenaran algoritma dengan memastikan kondisi tertentu tetap benar selama perulangan berjalan	Algoritma yang menerapkan Loop atau Iterasi	Membuktikan kebenaran algoritma seperti binary search atau bubble sort.
Pembuktian Formal	Pembuktian menggunakan logika matematika dan aturan inferensi formal.	Semua jenis algoritma, baik rekursif, iteratif, atau kompleks	Membuktikan kebenaran algoritma sorting atau searching secara formal.

QnA

Tugas

Kelompok 5

Soal no. 1

Buatlah program untuk menghasilkan **output** berupa **boolean** untuk mengecek kebenaran $n! > 2^n$ untuk $n \geq 4$.

(*) *Antisipasi semua kondisi yang mungkin terjadi*

Contoh Output :

Masukkan sebarang nilai n : 5

Dihasilkan nilai $5! = 120$ dan $2^5 = 32$.

Kesimpulan : benar

Setelah membuat program tersebut, buktikan pula kebenaran dari $n! > 2^n$ untuk $n \geq 4$ menggunakan **induksi matematika!**

Soal no. 2

```
import java.util.Scanner;

public class Faktorial {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.print("Masukkan bilangan n: ");
        int n = input.nextInt();
        int faktorial = 1; //

        for (int i = 1; i <= n; i++) {
            faktorial *= i;
        }

        System.out.println("Hasil faktorial: " + faktorial);
    }
}
```

Soal no. 2

- Buktikan kebenaran program diatas menggunakan Invarian Loop.
- Temukan dan jelaskan jika terdapat kesalahan dalam program.
- Jika terdapat kesalahan perbaiki program agar sesuai dengan definisi faktorial.

Thank You