



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUM INI SAYA BUAT DENGAN USAHA SENDIRI TANPA MENGGUNAKAN BANTUAN ORANG LAIN. SEMUA MATERI YANG SAYA AMBIL DARI SUMBER LAIN SUDAH SAYA CANTUMKAN SUMBERNYA DAN TELAH SAYA TULIS ULANG DENGAN BAHASA SAYA SENDIRI.

SAYA SANGGUP MENERIMA SANKSI JIKA MELAKUKAN KEGIATAN PLAGIASI, TERMASUK SANKSI TIDAK LULUS MATA KULIAH INI.

NIM	71251188
Nama Lengkap	ANGELA REVALINE SETIAWAN
Minggu ke / Materi	03 / Flowchart dan Pseudocode

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS KRISTEN DUTA WACANA
YOGYAKARTA
2026

BAGIAN 1: MATERI MINGGU INI (40%)

Link Github : https://github.com/angelarevalines/71251188_valine.git

1. Algoritma

Pada minggu kedua ini kita akan membahas Algoritma, Flowchart, dan Pseudocode. Sekarang kita akan membahas Algoritma terlebih dahulu. Algoritma adalah langkah-langkah yang logis dan sistematis yang disusun dengan tujuan untuk menyelesaikan suatu masalah. Tujuan dari algoritma itu sendiri adalah memberikan petunjuk terkait langkah-langkah logika penyelesaian masalah dalam bentuk yang lebih mudah untuk dipahami oleh manusia. Algoritma ini digunakan sebagai acuan yang membantu dalam mengembangkan ke dalam program komputer.

Ada 3 macam bentuk notasi algoritma (penulisan), antara lain :

- 1) Uraian Deskriptif
- 2) *Flowchart* / Diagram Alir
- 3) *Pseudocode*

2. Uraian Deskriptif

Contoh masalah = menyelesaikan permasalahan dalam menghitung Luas dan Keliling suatu lingkaran (Algoritma_Hitung_Luas_dan_Keliling_Lingkaran)

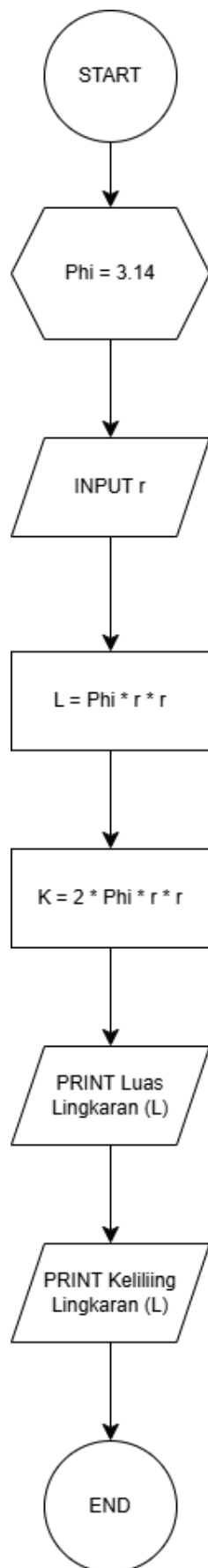
Deskripsi :

1. Masukkan jari-jari lingkaran (r).
2. Hitung luas lingkaran dengan rumus $L = p * r^{**2}$.
3. Hitung keliling lingkaran dengan rumus $K = 2 * p * r$.
4. Tampilkan luas lingkaran.
5. Tampilkan keliling lingkaran.

3. *Flowchart* / Diagram Alir

Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma dalam suatu program, yang menunjukkan arah alur program tersebut.

Contoh *flowchart* Algoritma_Hitung_Luas_dan_Keliling_Lingkaran :



Gambar 3.1 : Flowchart mencari luas dan keliling lingkaran.

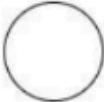

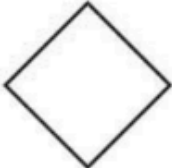
Flowchart sangat membantu analis dan programmer untuk dapat memecahkan masalah ke dalam bagian-bagian yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasiannya.







Fungsi dari *flowchart* :

1. Untuk mendesain program
2. Untuk merepresentasikan program

4. Notasi *Flowchart*

Setiap notasi dalam *flowchart* memiliki artinya masing-masing. Berikut ini adalah notasi yang sering digunakan.

	<p>Flow</p> <p>Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.</p>
	<p>On-Page Reference</p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.</p>
	<p>Off-Page Reference</p> <p>Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.</p>
	<p>Terminator</p> <p>Simbol yang menyatakan awal atau akhir suatu program.</p>
	<p>Process</p> <p>Simbol yang menyatakan suatu proses yang dilakukan komputer.</p>
	<p>Decision</p> <p>Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.</p>

	Input/output Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	Manual Operation Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	Document Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	Display Simbol yang menyatakan peralatan output yang digunakan.
	Preparation Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

5. Pseudocode

Notasi yang hampir menyerupai notasi Bahasa Pemrograman tingkat tinggi seperti Bahasa C dan Python. Strukturnya dapat dibagi menjadi 3 bagian, antara lain :

1. Bagian kepala (header)
2. Bagian Deklarasi (definisi variable)
3. Bagian Deskripsi (rincian langkah-langkah)

Contohnya :

Algoritma Luas_Persegi_Panjang

```
{Menghitung sebuah luas persegi panjang apabila
Diberikan panjang dan lebar persegi panjang}
```

Deklarasi

```
{Definisi variable}  
  
float panjang, lebar, luas
```

Deskripsi

```
INPUT (panjang, lebar)  
  
luas <- panjang * lebar  
  
PRINT (luas)
```

6. Notasi *Pseudocode*

Berikut ini notasi yang akan sering kita gunakan dalam *pseudocode*.

- 1) INPUT berfungsi untuk menunjukkan proses memasukkan suatu isi variable.
- 2) OUTPUT berfungsi untuk menunjukkan proses keluarnya hasil yang diinginkan.
- 3) WHILE berfungsi untuk sebuah perulangan yang telah memiliki iterasi awal.
- 4) FOR berfungsi untuk sebuah perulangan perhitungan iterasi.
- 5) REPEAT - UNTIL berfungsi untuk sebuah perulangan yang memiliki kondisi akhir.
- 6) IF - THEN - ELSE berfungsi untuk mengambil keputusan dari beberapa kondisi yang ada.

Sumber Materi : Modul 03 Praktikum Algoritma dan Pemrograman ([03 - Flowchart dan Pseudocode](#))

BAGIAN 2: LATIHAN MANDIRI (60%)

Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

Link Github : https://github.com/angelarevalines/71251188_valine.git

SOAL 1

Algoritma untuk menentukan apakah suatu bilangan merupakan bilangan prima atau bukan.

Pseudocode :

Algoritma Menentukan_Bilangan_Prima

Deklarasi

```
    int bilangan
```

Deskripsi

```
    START
```

```
        INPUT bilangan
```

```
        IF bilangan < 2 THEN
```

```
            PRINT Bukan bilangan Prima
```

```
        ELSE
```

```
            i = 2
```

```
            batas = bilangan ** 0.5
```

```
            WHILE i <= batas DO
```

```
                IF bilangan % i == 0 THEN
```

```
                    PRINT Bukan bilangan prima
```

```
                    STOP
```

```
                ENDIF
```

```
                i = i + 1
```

```
            ENDWHILE
```

```
            PRINT Bilangan Prima
```

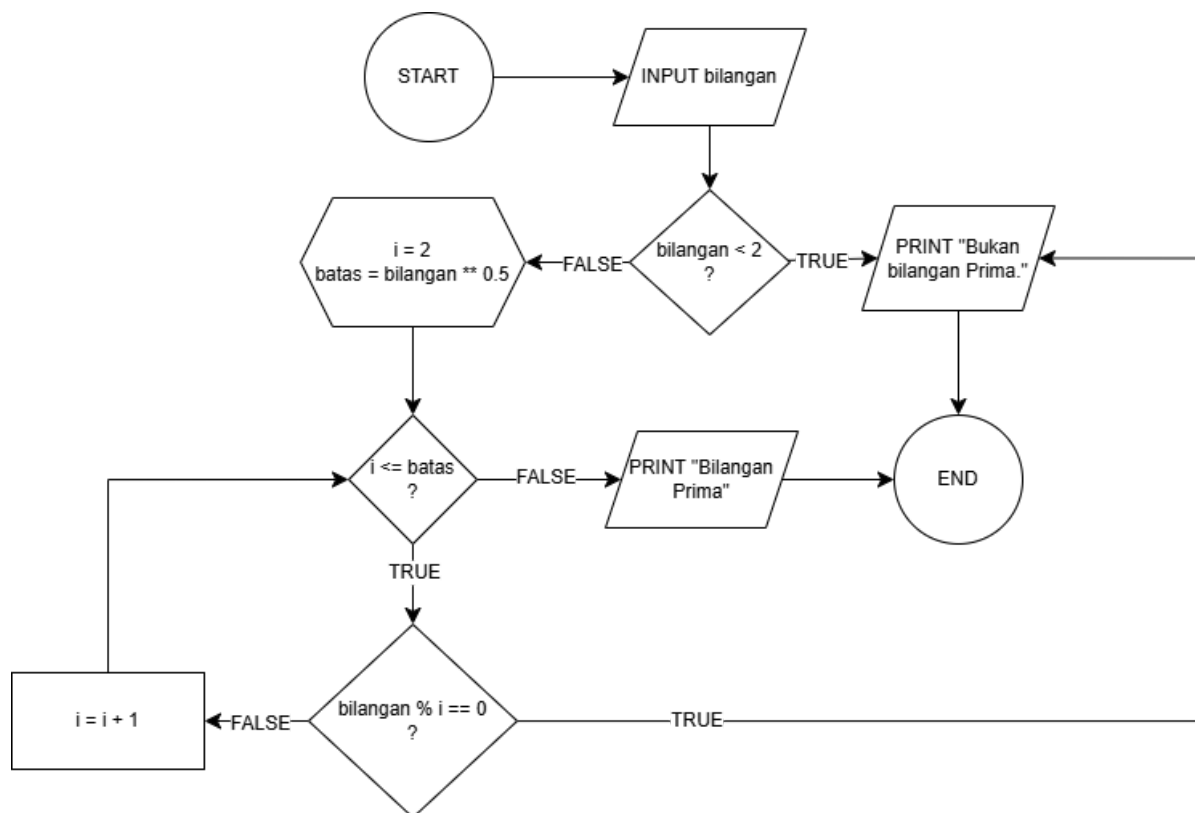
```
        ENDIF
```

```
    END
```

Penjelasan Pseudocode :

1. Algoritma ini berfungsi untuk menentukan apakah suatu bilangan yang diinput merupakan bilangan prima atau tidak.
2. START berfungsi untuk mengawali suatu algoritma pemrograman.
3. Untuk menjalankan program ini, user perlu menginput bilangan integer.
4. Bilangan harus lebih besar dari 2, jika tidak maka program akan menampilkan "Bukan bilangan Prima". Karena angka prima terkecil adalah 2.
5. Program akan melakukan perulangan jika bilangnya lebih besar atau sama dengan 2.
6. Dalam program ini telah disiapkan $i = 2$ dan $\text{batas} = \text{bilangan} ** 0.5$ untuk pengecekan.
7. Program akan terus melakukan perulangan selama nilai $i \leq \text{batas}$ dan mengecek apakah $\text{bilangan} \% i == 0$. Jika iya maka program akan menampilkan "Bukan bilangan Prima"
8. Tetapi jika tidak program akan lanjut ke $i = i + 1$ dan kembali melakukan pengecekan.
9. Jika sudah program akan menampilkan "Bilangan Prima"
10. END berfungsi untuk mengakhiri suatu algoritma pemrograman.

Flowchart :



SOAL 2

Algoritma untuk menentukan jenis akar dari suatu persamaan kuadrat.

Pseudocode :

Algoritma Jenis_Akar_Persamaan_Kuadrat

Deklarasi

```
float a, b, c
```

Deskripsi

```
START
```

```
    INPUT a, b, c
```

```
    D <- b**2 - 4 * a * c
```

```
    IF D > 0 THEN
```

```
        PRINT Akar Real dan Berlainan
```

```
    ELSE IF D = 0 THEN
```

```
        PRINT Akar Real dan Kembar
```

```
    ELSE
```

```
        PRINT Akar Imajiner / Tidak Real / Khayal
```

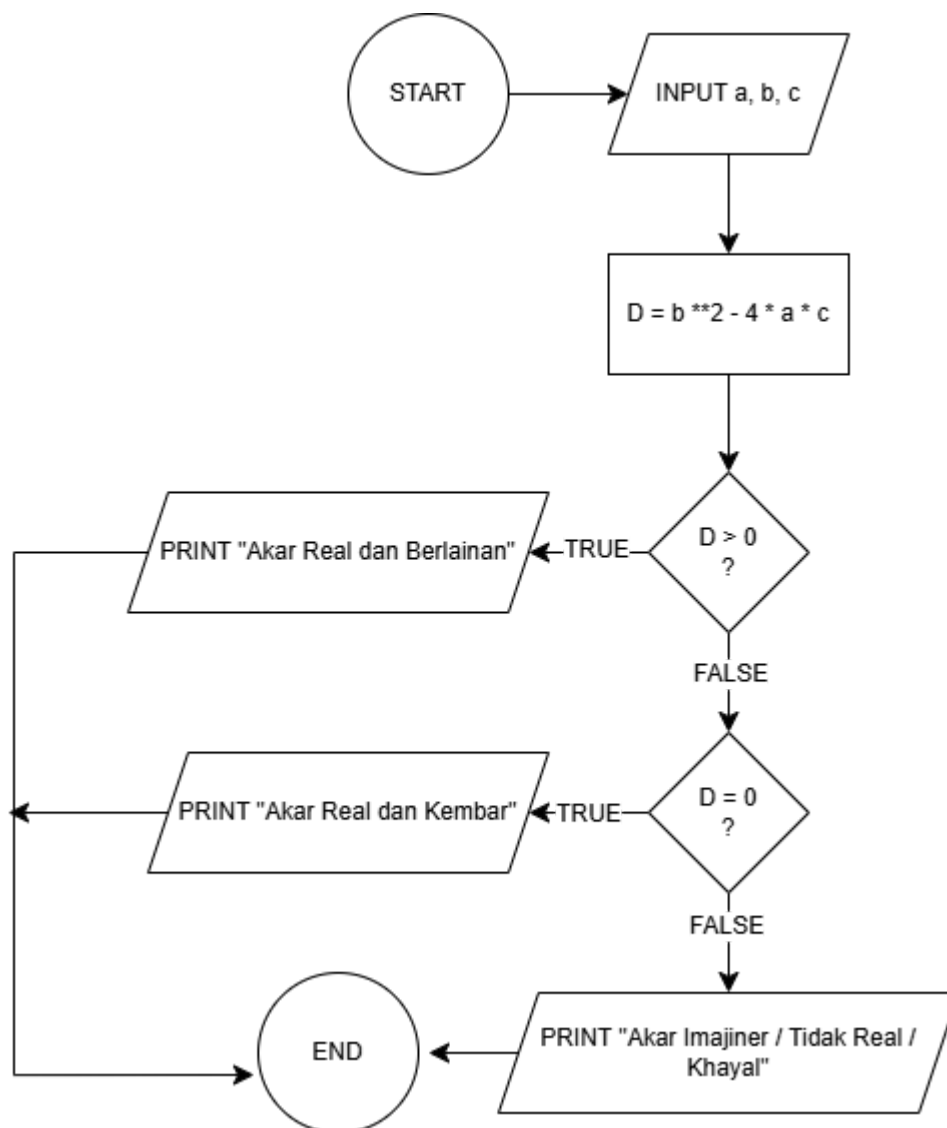
```
    ENDIF
```

```
END
```

Penjelasan Pseudocode :

1. Algoritma ini berfungsi untuk mencari jenis akar dari suatu persamaan kuadrat.
2. START berfungsi untuk mengawali suatu algoritma pemrograman.
3. Untuk mencari nilai diskriminannya, user perlu menginput nilai a, b, dan c dari suatu persamaan kuadrat.
4. Masukkan rumus untuk mencari nilai diskriminan yaitu $D = b^2 - 4 * a * c$.
5. Dalam soal sudah diberikan beberapa kondisi untuk menentukan jenis akar.
6. Jika nilai $D > 0$, maka program akan menampilkan "Akar Real dan Berlainan".
7. Jika tidak, program akan lanjut mengecek apakah nilai $D = 0$ atau tidak. Jika iya, maka program akan menampilkan "Akar Real dan Kembar".
8. Jika tidak, program akan langsung menampilkan "Akar Imajiner / Tidak Real / Khayal".
9. END berfungsi untuk mengakhiri suatu algoritma pemrograman.

Flowchart :



SOAL 3

Algoritma untuk menghitung nilai faktorial dari suatu bilangan.

Pseudocode :

Algoritma Hitung_Nilai_Faktorial

Deklarasi

```
int bilangan
```

Deskripsi

```
START
```

```
    INPUT bilangan
```

```
    IF bilangan < 0 THEN
```

```
        PRINT bilangan harus >= 0!
```

```
    ELSE
```

```
        WHILE bilangan >= i DO
```

```
            faktorial <- faktorial * i
```

```
            i <- i + 1
```

```
        ENDWHILE
```

```
        PRINT "Hasil Faktorial = ", Faktorial
```

```
    ENDIF
```

```
END
```

Penjelasan Pseudocode :

1. Algoritma ini berfungsi untuk menghitung nilai faktorial dari suatu bilangan.
2. START berfungsi untuk mengawali suatu algoritma pemrograman.
3. Untuk menghitung nilai faktorial, user perlu menginput bilangan integer.
4. Bilangan harus lebih besar dari 0, jika tidak maka program akan menampilkan "Bilangan harus >= 0".
5. Program akan melakukan perulangan dari 1 hingga bilangan yang diinput oleh user untuk menghitung faktorialnya. Faktorial adalah perkalian dari 1 hingga bilangan yang diinput oleh user. Contoh : input = 3, berarti $1 * 2 * 3 = 6$.
6. Agar mendapatkan nilai Faktorial, berarti Faktorial harus dikalikan oleh i agar nilai Faktorial yang awalnya 1 dapat bertambah terus sesuai dengan Faktorialnya.
7. ENDWHILE untuk mengakhiri perulangan yang dilakukan oleh program.
8. PRINT supaya program dapat menampilkan hasil faktorial yang telah dihitung.
9. END berfungsi untuk mengakhiri suatu algoritma pemrograman.

Flowchart :

