



Laporan Praktikum Algoritma & Pemrograman

Semester Genap 2025/2026

SAYA MENYATAKAN BAHWA LAPORAN PRAKTIKUMINI 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	71251214
Nama Lengkap	Raffiel Gideon Nyolo Nyolo
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%)

Algoritma

Algoritma biasanya didefinisikan sebagai rangkaian terurut langkah-langkah yang logis dan sistematis yang disusun untuk menyelesaikan suatu masalah. Tujuan algoritma adalah memberikan petunjuk tentang langkah-langkah logika penyelesaian masalah dalam bentuk yang mudah dipahami nalar manusia sebagai acuan yang membantu dalam mengembangkan program komputer. Pemahaman terhadap algoritma akan mencegah sejak dini kemungkinan terjadinya kesalahan logika pada program komputer yang dikembangkan.

Penulisan (Notasi algoritma)

Terdapat tiga bentuk notasi algoritma, antara lain:

1. Uraian deskriptif
 2. *Flowchart / Diagram Alir*
 3. *Pseudocode*
1. Uraian deskriptif

Notasi algoritma ini adalah bentuk penjelasan algoritma menggunakan kalimat deskriptif (bahasa sehari-hari yang terstruktur) untuk menjelaskan langkah-langkah penyelesaian suatu masalah tanpa menggunakan simbol flowchart atau sintaks bahasa pemrograman tertentu. Uraian materi biasanya ditulis secara runtut dan logis, menjelaskan proses dari awal hingga akhir secara jelas agar mudah dipahami manusia, sehingga cocok digunakan pada tahap perancangan awal sebelum algoritma diubah menjadi pseudocode atau kode program.

Contoh: menyelesaikan permasalahan menghitung luas dan keliling suatu lingkaran Algoritma Hitung_Luas_dan_Keliling_Lingkaran.

Deskripsi:

1. Masukan jari-jari lingkaran (r).
2. Hitung luas lingkaran dengan rumus $L = \pi * r^2$.
3. Hitung keliling lingkaran dengan rumus $K = 2 * \pi * r$.

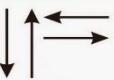
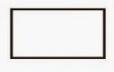
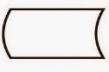
4. Tampilkan luas lingkaran.
5. Tampilkan keliling lingkaran.

2. *Flowchart / Diagram Alir*

Flowchart atau diagram alir adalah diagram atau bagan yang digunakan untuk menggambarkan langkah-langkah suatu algoritma atau proses secara visual menggunakan simbol-simbol tertentu dan dihubungkan dengan panah (alur). Setiap simbol memiliki arti khusus, misalnya simbol oval untuk mulai/selesai, persegi panjang untuk proses, jajargenjang untuk input/output, dan belah ketupat untuk percabangan (keputusan). *Flowchart* membantu seseorang memahami alur logika program atau sistem dengan lebih mudah karena ditampilkan dalam bentuk gambar yang terstruktur dan sistematis.

Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmensegmen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian. Kegunaan *flowchart* antara lain, untuk mendesain program dan untuk merepresentasikan program. Maka, flowchart harus dapat merepresentasikan komponen-komponen dalam bahasa pemrograman.

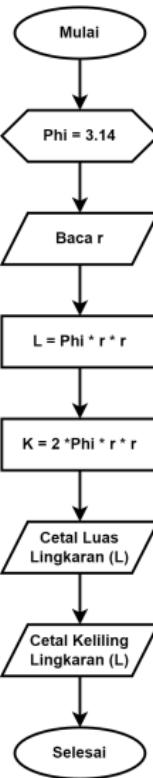
Pada dasarnya notasi dalam flowchart memiliki arti yang berbeda-beda. Berikut adalah notasi yang sering digunakan dalam proses pembuatan *flowchart*.

	Flow Direction symbol Yaitu simbol yang digunakan untuk menghubungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga connecting line.		Simbol Manual Input Simbol untuk pemasukan data secara manual on-line keyboard
	Terminator Symbol Yaitu simbol untuk permulaan (start) atau akhir (stop) dari suatu kegiatan		Simbol Preparation Simbol untuk mempersiapkan penyimpanan yang akan digunakan sebagai tempat pengolahan di dalam storage.
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses dalam lembar / halaman yang sama.		Simbol Predefine Proses Simbol untuk pelaksanaan suatu bagian (sub-program)/procedure
	Connector Symbol Yaitu simbol untuk keluar - masuk atau penyambungan proses pada lembar / halaman yang berbeda.		Simbol Display Simbol yang menyatakan peralatan output yang digunakan yaitu layar, plotter, printer dan sebagainya.
	Processing Symbol Simbol yang menunjukkan pengolahan yang dilakukan oleh komputer		Simbol disk and On-line Storage Simbol yang menyatakan input yang berasal dari disk atau disimpan ke disk.
	Simbol Manual Operation Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh computer		Simbol magnetik tape Unit Simbol yang menyatakan input berasal dari pita magnetik atau output disimpan ke pita magnetik.
	Simbol Decision Simbol pemilihan proses berdasarkan kondisi yang ada.		Simbol Punch Card Simbol yang menyatakan bahwa input berasal dari kartu atau output ditulis ke kartu
	Simbol Input-Output Simbol yang menyatakan proses input dan output tanpa tergantung dengan jenis peralatannya		Simbol Dokumen Simbol yang menyatakan input berasal dari dokumen dalam bentuk kertas atau output dicetak ke kertas.

Gambar 1.1: Notasi *flowchart*

Notasi di atas memiliki jenis dan fungsi yang berbeda-beda. Ada yang berfungsi untuk menghubungkan satu notasi dengan notasi lainnya seperti notasi flow, on-page dan off-page reference. Selain itu ada juga notasi yang berfungsi untuk menunjukkan suatu proses yang sedang berjalan, dan yang terakhir terdapat notasi yang berfungsi untuk memasukan input dan menampilkan output.

Berikut contoh *flowchart* menghitung luas dan keliling lingkaran yang algoritmanya dinotasikan dalam bentuk diagram alir (*flowchart*):



Gambar 1.2: *Flowchart* mencari luas dan keliling lingkaran.

3. Pseudocode

Pseudocode adalah cara menuliskan algoritma menggunakan bahasa yang menyerupai kode program tetapi tidak terikat pada aturan sintaks bahasa pemrograman tertentu. Pseudocode ditulis secara terstruktur dan logis (menggunakan kata-kata seperti *IF*, *THEN*, *ELSE*, *FOR*, *WHILE*) agar mudah dipahami manusia sekaligus mudah diterjemahkan ke dalam bahasa pemrograman seperti Python atau C++. Tujuannya adalah untuk menjelaskan alur penyelesaian masalah secara jelas sebelum diubah menjadi program yang sebenarnya.

Struktur algoritma dibagi ke dalam beberapa bagian, diantaranya:

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

Contoh:

```
Algoritma Luas_persegi_panjang
    {Menghitung sebuah luas persegi panjang apabila panjang
     dan lebar persegi panjang tersebut diberikan}
```

Deklarasi

```
{Definisi nama peubah/variabel}
```

```
float panjang, lebar, luas
```

Deskripsi

```
READ (panjang,lebar)           #bisa juga : INPUT
```

```
luas <- panjang * lebar
```

```
WRITE (Luas)                  #bisa juga : OUTPUT
```

Berikut ini notasi yang sering digunakan dalam pseudocode

1. INPUT: Digunakan untuk menunjukkan proses memasukan suatu isi variabel.
2. OUTPUT: Digunakan untuk menunjukkan proses keluaran yang terjadi.
3. WHILE: Digunakan untuk sebuah perulangan yang memiliki iterasi awali.
4. FOR: Digunakan untuk sebuah perulangan perhitungan iterasi.
5. REPEAT – UNTIL: Digunakan untuk sebuah perulangan yang memiliki kondisi akhir.
6. IF – THEN – ELSE: Digunakan untuk mengambil sebuah keputusan dari beberapa kondisi.

BAGIAN 2: LATIHAN MANDIRI (60%)

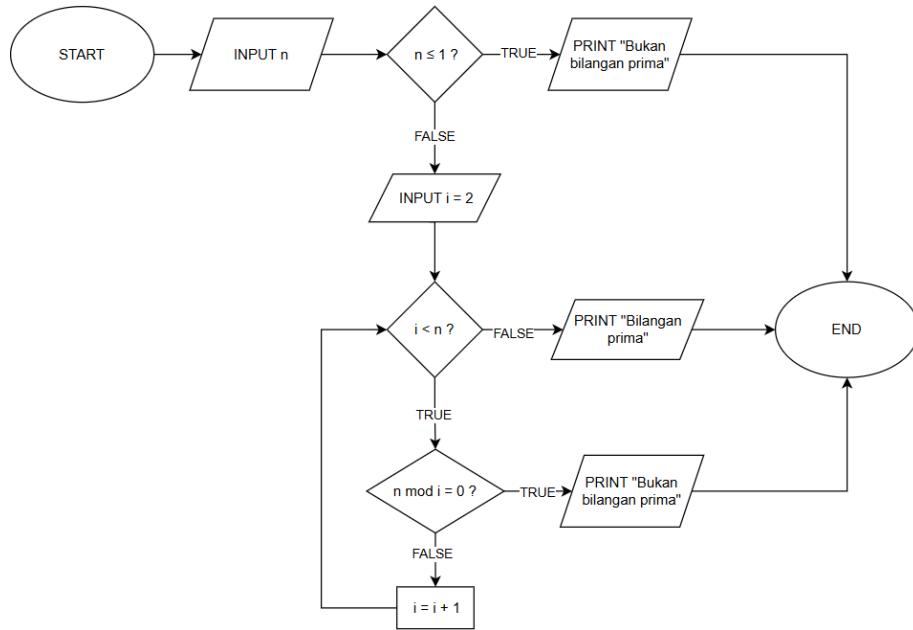
Pada bagian ini anda menuliskan jawaban dari soal-soal Latihan Mandiri yang ada di modul praktikum. Jawaban anda harus disertai dengan source code, penjelasan dan screenshot output.

Link Github: https://github.com/fieliel-star/71251214_raffiel.git

Soal 1

Pseudocode ALGORITMA cekBilanganPrima:

```
START
    INPUT n
    IF n <= 1 THEN
        PRINT "Bukan bilangan prima"
    ELSE
        i = 2
        WHILE i < n DO
            IF n mod i = 0 THEN
                PRINT "Bukan bilangan prima"
                STOP
            ENDIF
            i = i + 1
        ENDWHILE
        PRINT "Bilangan prima"
    ENDIF
END
```



Penjelasan:

Algoritma tersebut dimulai dengan menerima input sebuah bilangan n . Pertama, dilakukan pengecekan apakah $n \leq 1$. Jika kondisi ini benar, maka bilangan tersebut langsung dinyatakan bukan bilangan prima, karena bilangan prima adalah bilangan yang lebih besar dari 1 dan hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri. Jika $n > 1$, maka program menginisialisasi variabel i dengan nilai 2 sebagai pembagi awal.

Selanjutnya, program melakukan perulangan selama $i < n$. Pada setiap perulangan, diperiksa apakah $n \bmod i = 0$. Jika hasilnya 0, berarti n habis dibagi oleh bilangan selain 1 dan dirinya sendiri, sehingga dinyatakan bukan bilangan prima dan proses dihentikan. Jika tidak habis dibagi, maka nilai i ditambah 1 dan pemeriksaan diulang. Apabila seluruh nilai i sudah diperiksa hingga tidak memenuhi kondisi perulangan dan tidak ditemukan pembagi, maka bilangan tersebut dinyatakan sebagai bilangan prima.

Soal 2

Pseudocode Algoritma jenisAkarPersamaanKuadrat:

START

INPUT a, b, c

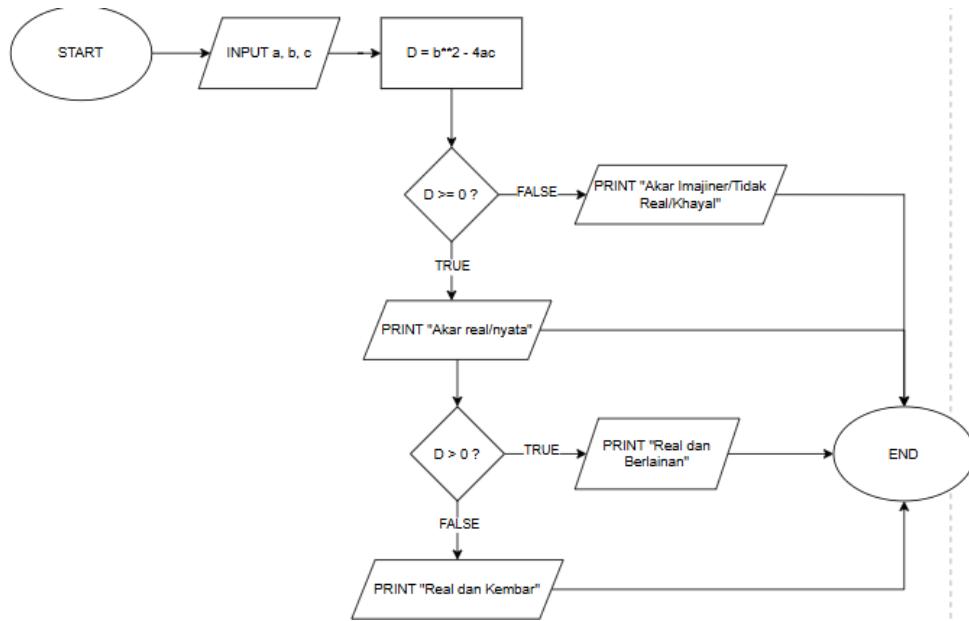
$D = b^2 - 4*a*c$

```

IF D >= 0 THEN
    OUTPUT "Akar Real/Nyata"
    IF D > 0 THEN
        PRINT "Real dan Berlainan"
    ELSE
        PRINT "Real dan Kembar"
    ENDIF
ELSE
    PRINT "Akar Imaginer/Tidak Real/Khayal"
ENDIF
END

```

Flowchart:



Penjelasan:

Pengguna memasukkan nilai a, b, c sebagai koefisien persamaan kuadrat. Setelah itu dilakukan proses perhitungan diskriminan dengan rumus $D = b^2 - 4ac$. Nilai D inilah yang menjadi penentu jenis akar. Selanjutnya terdapat percabangan pertama dengan kondisi $D \geq 0$. Jika kondisi ini salah (FALSE), berarti $D < 0$ sehingga program langsung mencetak "Akar Imaginer/Tidak Real/Khayal", lalu menuju END.

Jika kondisi $D \geq 0$ bernilai benar (TRUE), maka program mencetak “Akar Real/Nyata” karena diskriminan tidak negatif. Setelah itu dilakukan percabangan kedua untuk menentukan jenis akar realnya. Jika $D > 0$, maka dicetak “Real dan Berlainan” karena terdapat dua akar berbeda. Jika tidak (berarti $D = 0$), maka dicetak “Real dan Kembar” karena kedua akarnya sama. Semua jalur akhirnya menuju END, sehingga alur logika sudah runtut dan sesuai teori diskriminan pada persamaan kuadrat.

Soal 3

Pseudocode menghitungFaktorial:

Berikut pseudocode berdasarkan flowchart pada gambar (menghitung faktorial n):

ALGORITMA HitungFaktorial

START

INPUT n

IF n < 0 THEN

 PRINT "Tidak terdefinisi"

ELSE

 hasil = 1

 i = 1

 WHILE i <= n DO

 hasil = hasil * i

 i = i + 1

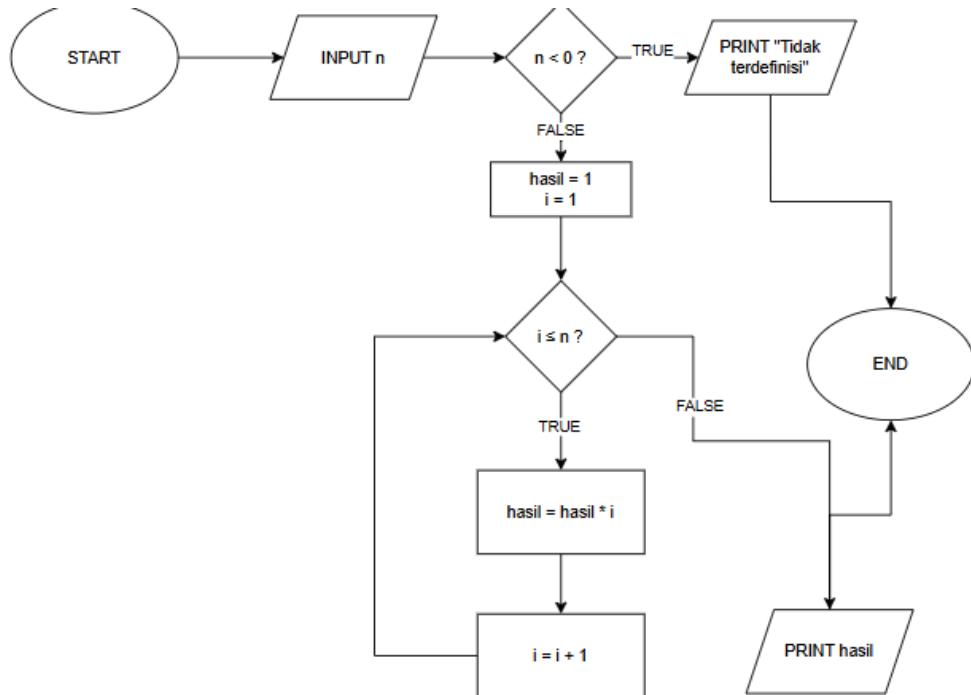
 ENDWHILE

 PRINT hasil

ENDIF

END

Flowchart:



Penjelasan:

Pengguna diminta menginput nilai n. Program kemudian mengecek kondisi $n < 0$. Jika kondisi ini benar (TRUE), maka program langsung menampilkan pesan "Tidak terdefinisi" karena faktorial untuk bilangan negatif memang tidak didefinisikan, kemudian proses berakhir (END). Jika kondisi salah (FALSE), maka program melanjutkan dengan menginisialisasi dua variabel: hasil = 1 (sebagai penampung hasil perkalian) dan i = 1 (sebagai penghitung perulangan).

Selanjutnya dilakukan perulangan dengan mengecek apakah $i \leq n$. Jika benar, maka nilai hasil dikalikan dengan i (hasil = hasil \times i), lalu i ditambah 1. Proses ini terus berulang sampai kondisi $i \leq n$ bernilai salah. Ketika perulangan selesai, program mencetak nilai hasil, yang merupakan nilai faktorial dari n, lalu proses berakhir (END).