



---

# 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	71251230
Nama Lengkap	Okky Alexander
Minggu ke / Materi	03 / Flowchart dan Pseudocode

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

## MATERI ALGORITMA

Algoritma adalah urutan langkah-langkah yang logis dan disusun secara rapi untuk menyelesaikan sebuah masalah. Tujuannya adalah sebagai panduan atau "resep" yang mudah dipahami manusia sebelum mulai membuat program di komputer. Dengan memahami alurnya dari awal, kita bisa menghindari kesalahan logika saat aplikasi atau program tersebut dibuat.

### Cara Menulis Algoritma (Notasi)

Ada tiga cara umum untuk menggambarkan sebuah algoritma:

1. Uraian Deskriptif: Menjelaskan langkah-langkah menggunakan bahasa sehari-hari (seperti menulis instruksi biasa).
2. Flowchart (Diagram Alir): Menggunakan simbol atau gambar untuk menunjukkan alur prosesnya.
3. Pseudocode: Penulisan yang bentuknya mirip kode komputer, tapi masih menggunakan bahasa manusia agar lebih mudah dibaca.

## MATERI URAIAN DESKRIPTIF

Contoh: Menghitung Luas dan Keliling Lingkaran

Bayangkan kita ingin membuat panduan sederhana untuk mencari luas dan keliling lingkaran. Langkah-langkahnya adalah sebagai berikut:

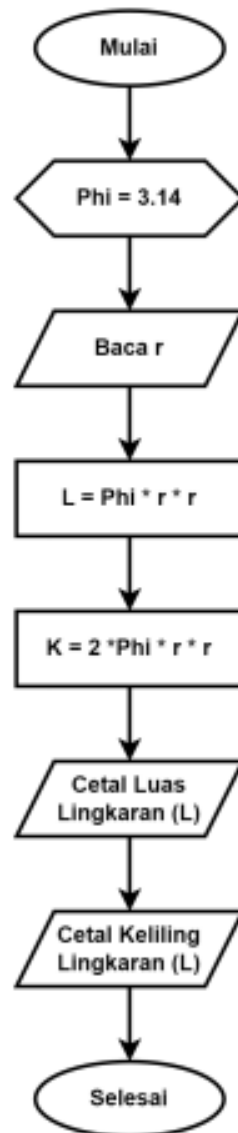
Langkah-langkahnya:

1. Input Data: Pertama, cari tahu dulu berapa ukuran jari-jari ( $r$ ) lingkarannya.
2. Cari Luas: Hitung luasnya menggunakan rumus  $L = p * r ** 2$ .
3. Cari Keliling: Hitung kelilingnya dengan rumus  $K = 2 * p * r$ .
4. Hasil Akhir: Informasikan berapa hasil luas yang sudah dihitung tadi.
5. Selesai: Informasikan juga hasil kelilingnya.

## MATERI FLOWCHART

Sesuai namanya, flowchart adalah bagan yang menggunakan simbol-simbol tertentu untuk menunjukkan ke mana arah alur logika program kita berjalan, mulai dari start sampai selesai. Contoh

Flowchart: Menghitung luas dan keliling lingkaran yang algoritmanya dinotasikan dalam bentuk diagram alir (flowchart).



Gambar 1.1: Flowchart mencari luas dan keliling lingkaran.

Secara ringkas, Flowchart bertindak sebagai "peta navigasi" bagi para analis dan *programmer*. Daripada langsung menghadapi masalah besar yang rumit, flowchart membaginya menjadi potongan-potongan kecil yang lebih sederhana agar lebih mudah dikelola. Selain itu, flowchart sangat berguna untuk membandingkan beberapa cara penyelesaian (alternatif) sebelum kita benar-benar menulis kode program.

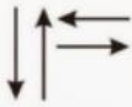




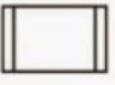







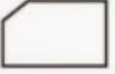
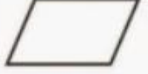

## Kegunaan Utama Flowchart

1. Mendesain Program: Menjadi sketsa atau rancangan awal sebelum aplikasi dibuat.
2. Merepresentasikan Program: Menjadi dokumentasi visual agar orang lain (atau tim lain) paham bagaimana logika program tersebut bekerja tanpa harus membaca ribuan baris kode.

## MATERI NOTASI FLOWCHART

Setiap bentuk atau simbol dalam *flowchart* itu punya "tugas" khusus. Jadi, kita tidak bisa asal pilih gambar kotak atau bulat, karena setiap bentuk mewakili perintah tertentu dalam pemrograman.

Berikut adalah penjelasan sederhana mengenai notasi (simbol) yang paling sering muncul:

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

Gambar 1.2: Notasi Flowchart.

Notasi-notasi tersebut punya jenis dan kegunaan yang bervariasi. Ada simbol yang tugasnya menyambungkan satu alur ke alur lainnya, seperti tanda panah (*flow*), serta penyambung di dalam satu halaman (*on-page*) maupun antar halaman (*off-page*). Selain itu, terdapat notasi yang dipakai untuk menggambarkan suatu proses yang sedang dikerjakan, dan yang terakhir adalah notasi khusus untuk bagian memasukkan data (*input*) serta menampilkan hasil (*output*).

## MATERI PSEUDOCODE

Notasi yang tampilannya mirip dengan bahasa pemrograman tingkat tinggi, seperti bahasa C atau Python. Secara umum, strukturnya terbagi menjadi tiga bagian utama, yaitu bagian kepala (*header*) sebagai judul, bagian deklarasi untuk mendefinisikan variabel, serta bagian deskripsi yang berisi rincian langkah-langkah penyelesaiannya.

Contoh Pseudocode :

```
1 ▾ Algoritma Luas_persegi_panjang
2   {Menghitung sebuah luas persegipanjang
3   apabila panjang dan lebar persegipanjang
4   tersebut diberikan}
5 ▾ Deklarasi
6   {Definisi nama peubah/variabel}
7   float panjang, lebar, luas
8 ▾ Deskripsi
9   READ (panjang,lebar) #bisa juga : INPUT
10  luas <- panjang * lebar
11  WRITE (Luas) #bisa juga : OUTPUT
```

Gambar 1.3: Contoh Pseudocode.

## MATERI NOTASI PSEUDOCODE

Dalam penulisan *pseudocode*, terdapat beberapa notasi standar yang sering digunakan untuk menggambarkan logika program. Notasi **INPUT** berfungsi untuk menerima data ke dalam variabel, sedangkan **OUTPUT** digunakan untuk menampilkan hasil prosesnya. Untuk urusan pengulangan, terdapat tiga pilihan: **WHILE** untuk perulangan yang mengecek kondisi di awal, **FOR** untuk perulangan dengan jumlah hitungan yang sudah pasti, serta **REPEAT – UNTIL** untuk perulangan yang mengecek kondisi di

bagian akhir. Terakhir, terdapat notasi **IF – THEN – ELSE** yang berfungsi untuk menentukan alur program berdasarkan pilihan atau kondisi tertentu.

**INPUT:** Perintah untuk mengambil data dari pengguna atau sumber luar dan menyimpannya ke dalam variabel agar bisa diproses.

**OUTPUT:** Perintah untuk mengeluarkan atau menampilkan data yang sudah diproses agar bisa dilihat oleh pengguna.

**WHILE:** Digunakan untuk mengulang sebuah perintah selama syarat atau kondisi tertentu masih terpenuhi. Pengecekan dilakukan sebelum langkah pengulangan dimulai.

**FOR:** Digunakan saat kita sudah tahu persis berapa kali sebuah perintah harus diulang (misalnya mengulang sebanyak 10 kali).

**REPEAT – UNTIL:** Mirip dengan pengulangan lainnya, namun perintah dijalankan terlebih dahulu, baru kemudian kondisinya dicek di akhir.

**IF – THEN – ELSE:** Logika pemilihan untuk menangani berbagai situasi. Jika suatu kondisi benar, maka jalankan perintah A, jika salah maka jalankan perintah B.

Source Code : [https://github.com/tuangkeman/71251230\\_Okky.git](https://github.com/tuangkeman/71251230_Okky.git)

### **LATIHAN 3.1**

START

INPUT angka

IF angka < 2 THEN

PRINT "Bukan Bilangan Prima"

ELSE

n = 2

WHILE angka % n != 0

n = n + 1

END WHILE

IF n == angka THEN

PRINT "Bilangan Prima"

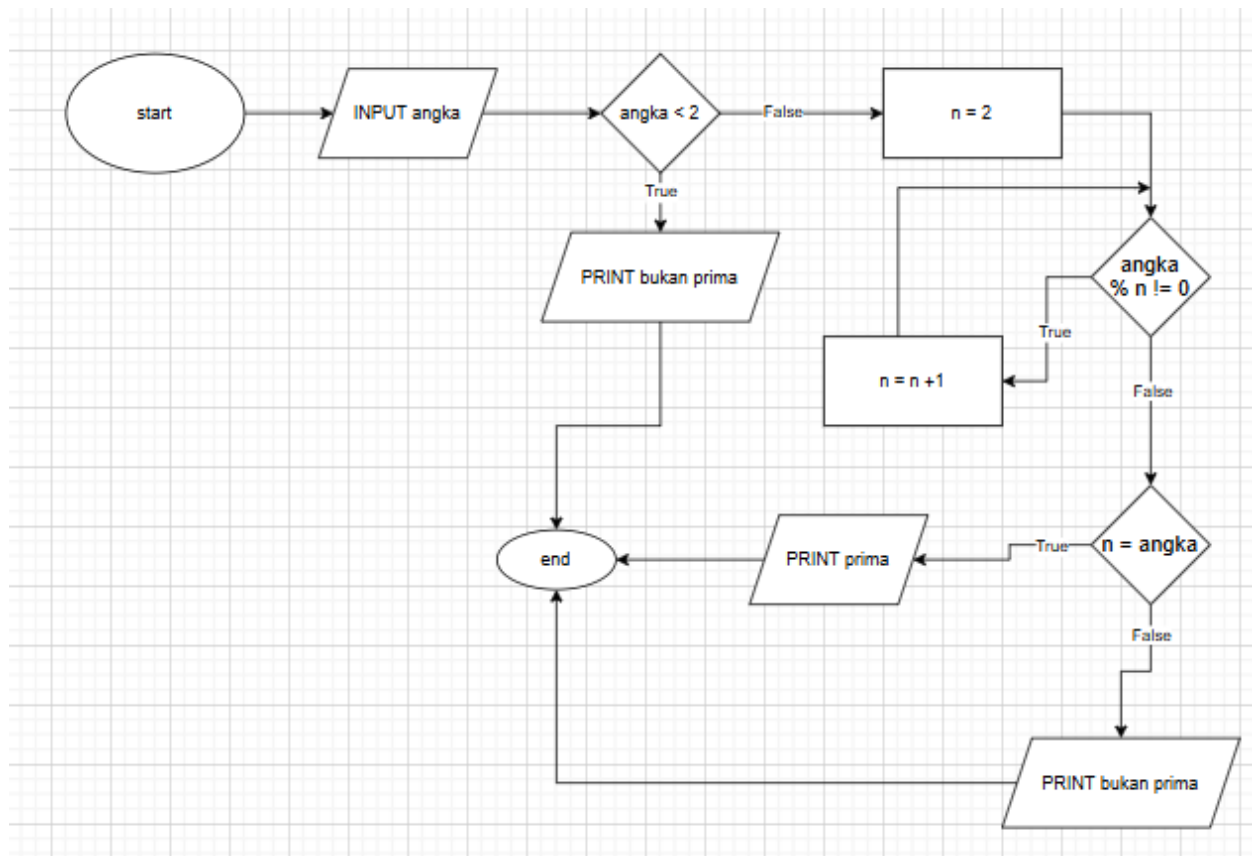
ELSE

PRINT "Bukan Bilangan Prima"

END IF

END IF

END



## LATIHAN 3.2

START

INPUT a, b, c

SET  $D = (b * b) - (4 * a * c)$

IF  $(D > 0)$  THEN

PRINT "Akar Real dan Berlainan"

ELSE IF  $(D = 0)$  THEN

PRINT "Akar Real dan Kembar"

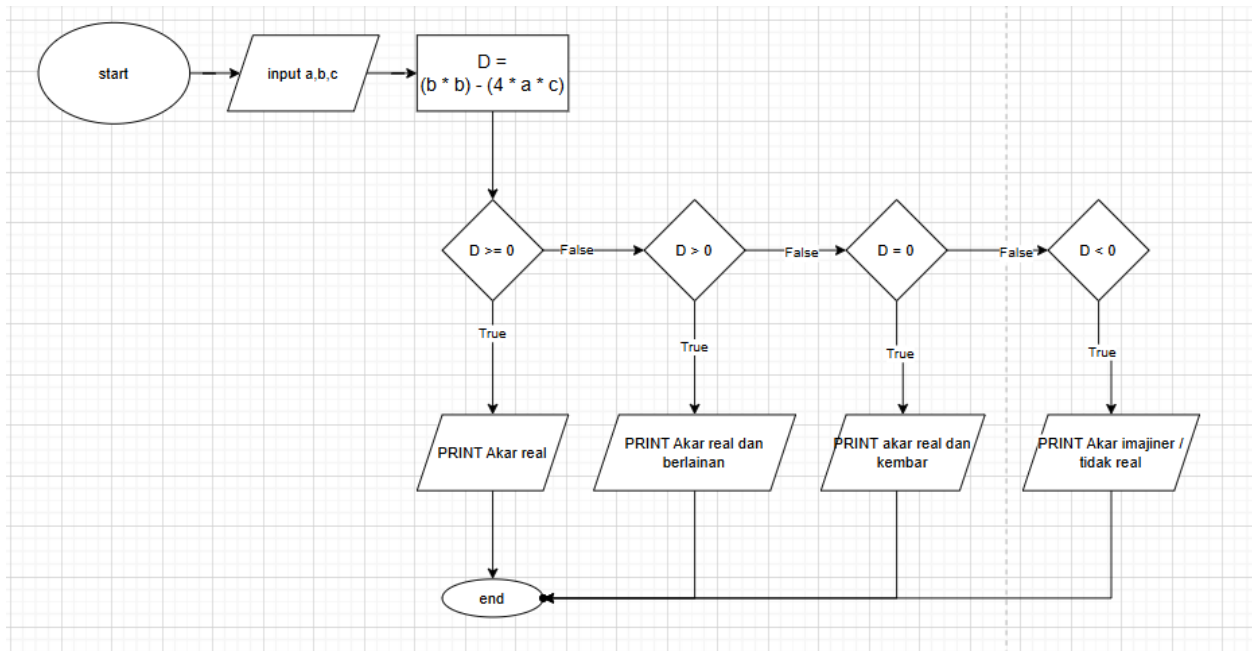
ELSE

PRINT "Akar Imajiner / Tidak Real / Khayal"

END IF

END





### LATIHAN 3.3

START

INPUT n

SET hasil = 1

SET i = 1

IF (n < 0) THEN

PRINT "Input harus bilangan positif"

ELSE

WHILE (i <= n)

SET hasil = hasil \* i

SET i = i + 1

END WHILE

PRINT hasil

END IF

END

