



---

# 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	71251184
Nama Lengkap	Matthew Jason Pratama
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 merupakan definisi dari rangkaian langkah-langkah yang logis dan sistematis yang disusun untuk menyelesaikan suatu masalah. Tujuan dari algoritma untuk memberikan petunjuk tentang langkah-langkah logika penyelesaian masalah dalam bentuk yang mudah dipahami manusia sebagai acuan mengembangkan program komputer. Pemahaman sejak dini terhadap algoritma akan mencegah terjadinya kesalahan dalam program komputer yang dikembangkan.

### **Penulisan(Notasi Algoritma)**

Ada tiga macam bentuk notasi algoritma antara lain:

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

## **Uraian Deskriptif**

Contohnya: menyelesaikan permasalahan menghitung luas dan keliling persegi

Hitung\_luas\_dan\_Keliling\_Persegi

Deskripsi:

1. Masukkan panjang sisi persegi ( $r$ )
2. Hitung luas persegi dengan rumus  $L = r^2$
3. Hitung keliling persegi dengan rumus  $= 4r$

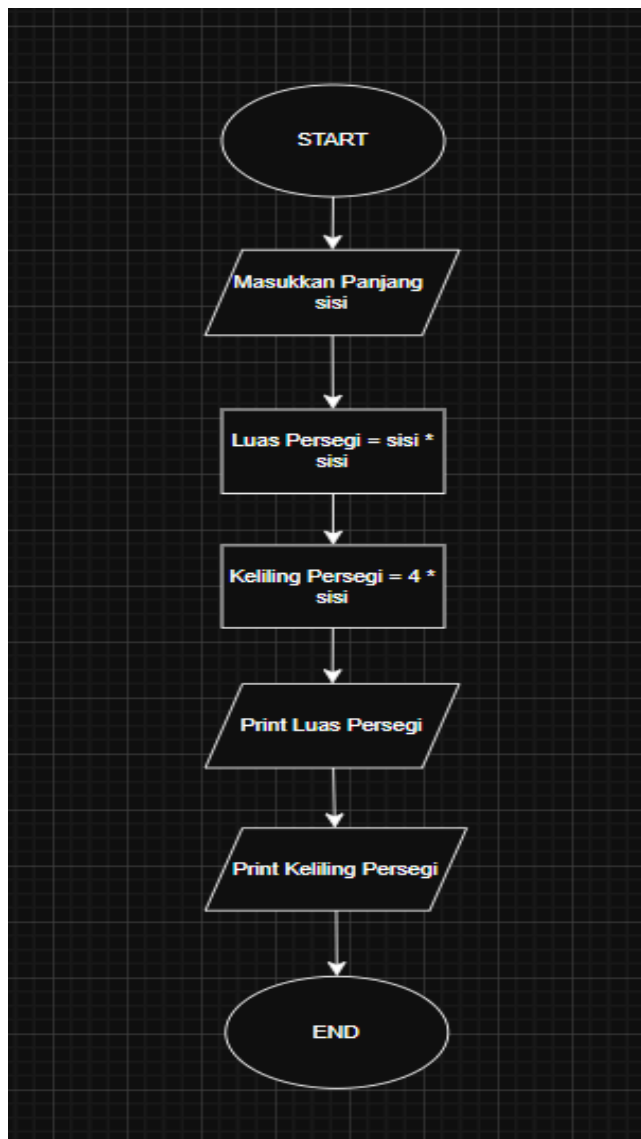
4. Tampilkan luas persegi
5. Tampilkan keliling persegi

### Flowchart / Diagram air

Flowchart atau diagram air adalah gambaran bentuk dari algoritma dalam suatu program, yang menyatakan arah alur program tersebut

Contoh:

Menghitung luas dan keliling persegi yang dibuat dalam bentuk flowchart.



Gambar 1.1 Contoh Flowchart menghitung luas dan keliling persegi

Dengan bantuan *flowchart* kita membantu diri kita dalam analisis dan programmer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menentukan alur jalannya program.

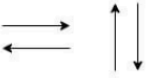









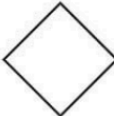
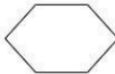
Kegunaan:

1. Mendesain program
2. Mepresentasikan Program

Maka *flowchart* sebaiknya dibuat terlebih dahulu untuk mempresentasikan komponen-komponen dalam bahasa pemrograman.

### **Notasi *Flowchart***

Pada dasarnya notasi *flowchart* memiliki arti yang berbeda-beda. Berikut notasi yang sering digunakan

	<b>Flow</b> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.		<b>Input/output</b> Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	<b>On-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.		<b>Manual Operation</b> Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	<b>Off-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.		<b>Document</b> Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	<b>Terminator</b> Simbol yang menyatakan awal atau akhir suatu program.		<b>Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedur.
	<b>Process</b> Simbol yang menyatakan suatu proses yang dilakukan komputer.		<b>Display</b> Simbol yang menyatakan peralatan output yang digunakan.
	<b>Decision</b> Simbol yang menunjukan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.		<b>Preparation</b> Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Gambar 1.2 Notasi dalam *Flowchart*. sumber : <https://l1nq.com/Te7IN>

Notasi tersebut 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 menunjukan suatu proses yang sedang berjalan dan yang terakhir ada notasi yang menampilkan input dan output.

## **Pseudocode**

Notasi yang menyerupai notasi Bahasa pemrograman tingkat tinggi, misalnya bahasa C dan bahasa Python. Struktur algoritma dibagi ke dalam beberapa bagian diantaranya:

1. Kepala (Header)
2. Deklarasi (Definisi Variabel)
3. Deskripsi (Rincian Langkah)

Contoh:

```
Algoritma Luas_persegi_panjang
{Menghitung sebuah luas persegipanjang
```

```
    apabila panjang dan lebar persegipanjang
    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
```

Gambar 1.3 Contoh *Pseudocode* menghitung luas persegi panjang. sumber : <https://l1nq.com/Te7IN>

## Notasi *Pseudocode*

Ada beberapa notasi yang digunakan dalam pseudocode

1. **INPUT** = Digunakan untuk memasukkan suatu variabel
2. **OUTPUT** = Digunakan untuk menunjukkan hasil dari proses dalam Pseudocode

3. **WHILE** = Digunakan untuk menunjukkan proses keluaran yang terjadi
4. **FOR** = Digunakan untuk sebuah perulangan perhitungan iterasi.
5. **REPEAT - UNTIL** = Digunakan untuk perulangan sampai memenuhi kondisi yang diminta.
6. **IF - THEN - ELSE** = Digunakan untuk mengambil keputusan dari beberapa kondisi

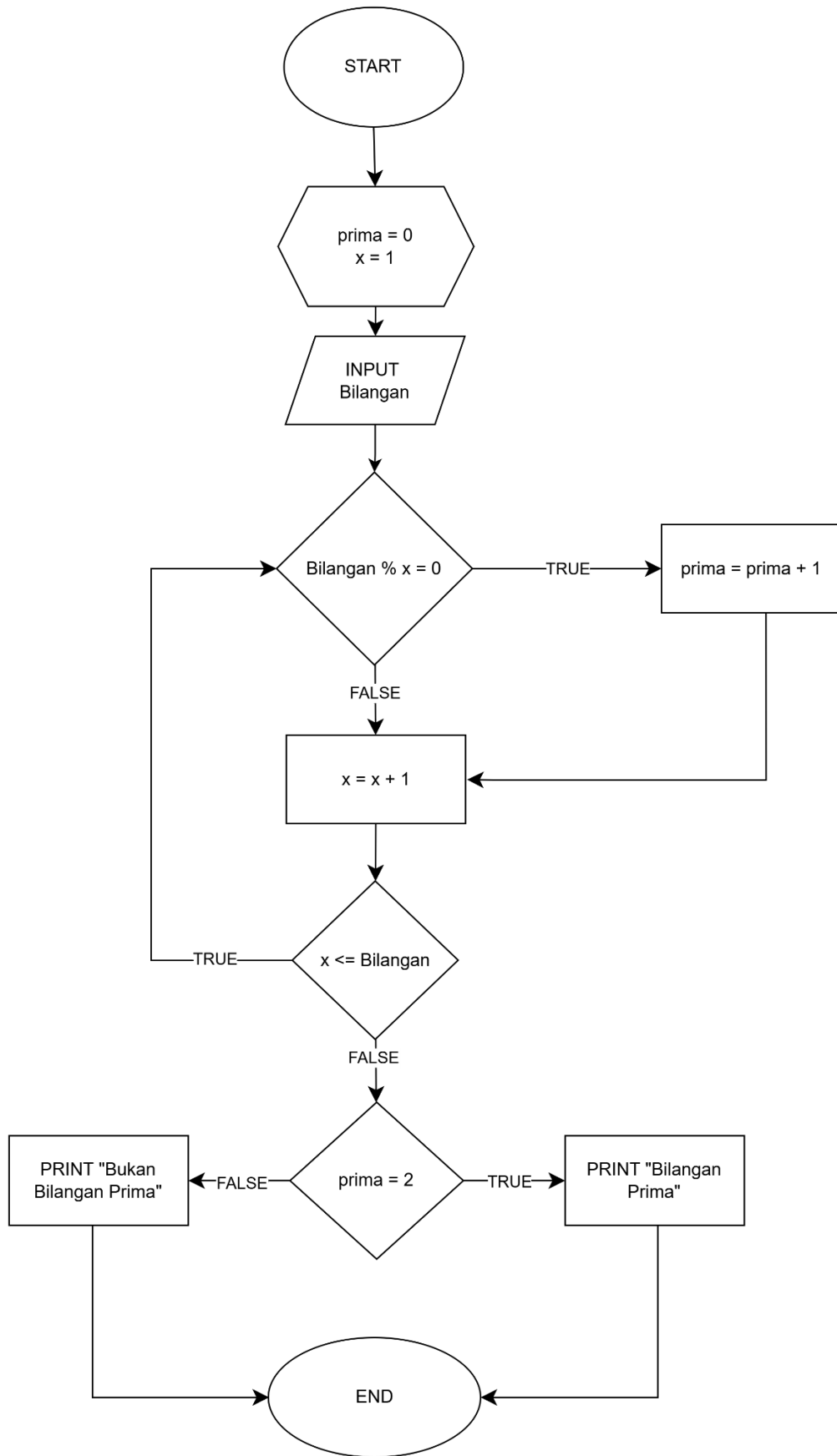
## **BAGIAN 2: LATIHAN MANDIRI (60%)**

Link Github : [https://github.com/Jason-prat/71251184\\_matthew](https://github.com/Jason-prat/71251184_matthew)

### **SOAL 1**

**Source:**





Gambar 2.1 Flowchart Bilangan Prima

```
Program Bilangan Prima

Deklarasi
    prima, x, Bilangan : Integer
Algoritma
    SET prima = 0
    SET x = 1
    INPUT Bilangan
    WHILE
        IF Bilangan MOD x = 0
            SET prima = prima + 1
        ENDIF
        x = x + 1
    ENDWHILE
    IF prima = 2
        PRINT "Bilangan Prima"
    ELSE
        PRINT "Bukan Bilangan Prima"
    ENDIF
    END
```

Gambar 2.2 Pseudocode Bilangan Prima

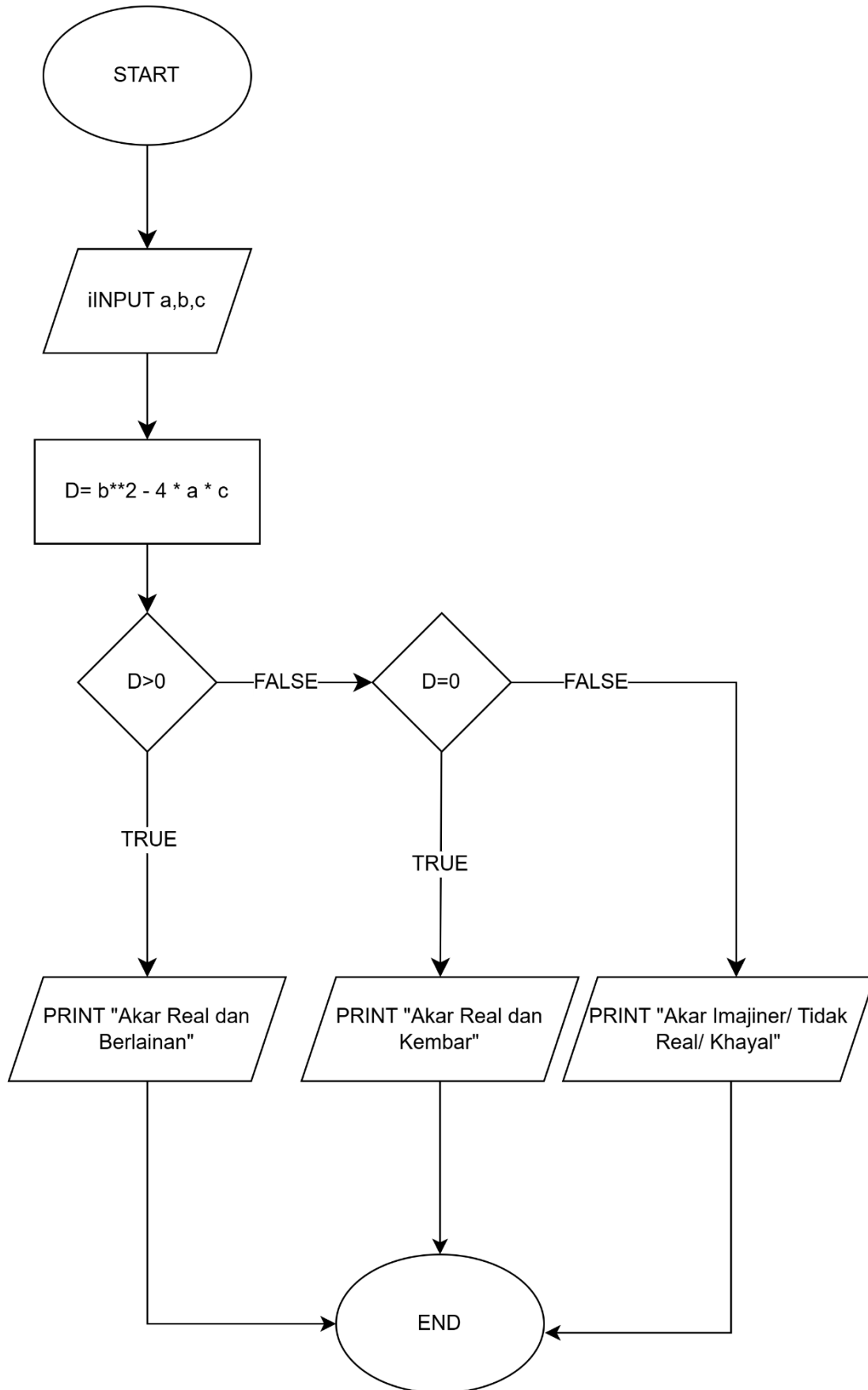
**Penjelasan:**

Flowchart dan pseudocode diatas berfungsi untuk melihat alur program untuk menentukan apakah angka yang dimasukkan merupakan bilangan prima atau bukan. Pertama kita membuat variabel pemfaktoran untuk menentukan apakah bilangan prima, jika pemfaktoran bernilai 2 maka nilai tersebut bilangan prima, jika nilai pemfaktoran lebih dari 2 maka nilai tersebut bukan bilangan prima. Kemudian program akan meminta user memasukkan angka yang ingin dicek, setelah angka dimasukkan akan dicek pertama kali dengan Bilangan = 1 jika ya maka bukan bilangan prima karena angka 1 bukanlah bilangan prima, jika FALSE akan dilanjutkan pengecekan dengan modulo jika habis dibagi pemfaktoran akan ditambah 1 nilainya dan nilai x akan ditambahkan 1, jika tidak nilai dari x akan ditambahkan 1, kemudian nilai x akan di cek apakah lebih kecil sama dengan dari Bilangan, jika ya program akan mengulang keatas untuk mengecek apakah bilangan tersebut masih bisa dibagi sampai habis dengan modulo, sampai

ketika nilai Bilangan lebih kecil dari  $x$  program akan mengecek apakah pemfaktoran nilainya sama dengan 2 jika ya maka bilangan prima jika tidak maka bukan bilangan prima.

## SOAL 2

**Source:**



Gambar 2.3 Flowchart Akar Kuadrat

```
PROGRAM Akar Kuadrat

Deklarasi
    a, b, c : Float
    D : Float
Algoritma
    INPUT a, b, c
    SET D = b**2 - 4 * a * c
    IF D > 0
        PRINT "Akar Real dan Berlainan"
    ELSE
        IF D = 0
            PRINT "Akar Real dan Kembar"
        ELSE
            PRINT "Akar Imajiner/ Tidak Real/ Khayal"
        ENDIF
    ENDIF
END
```

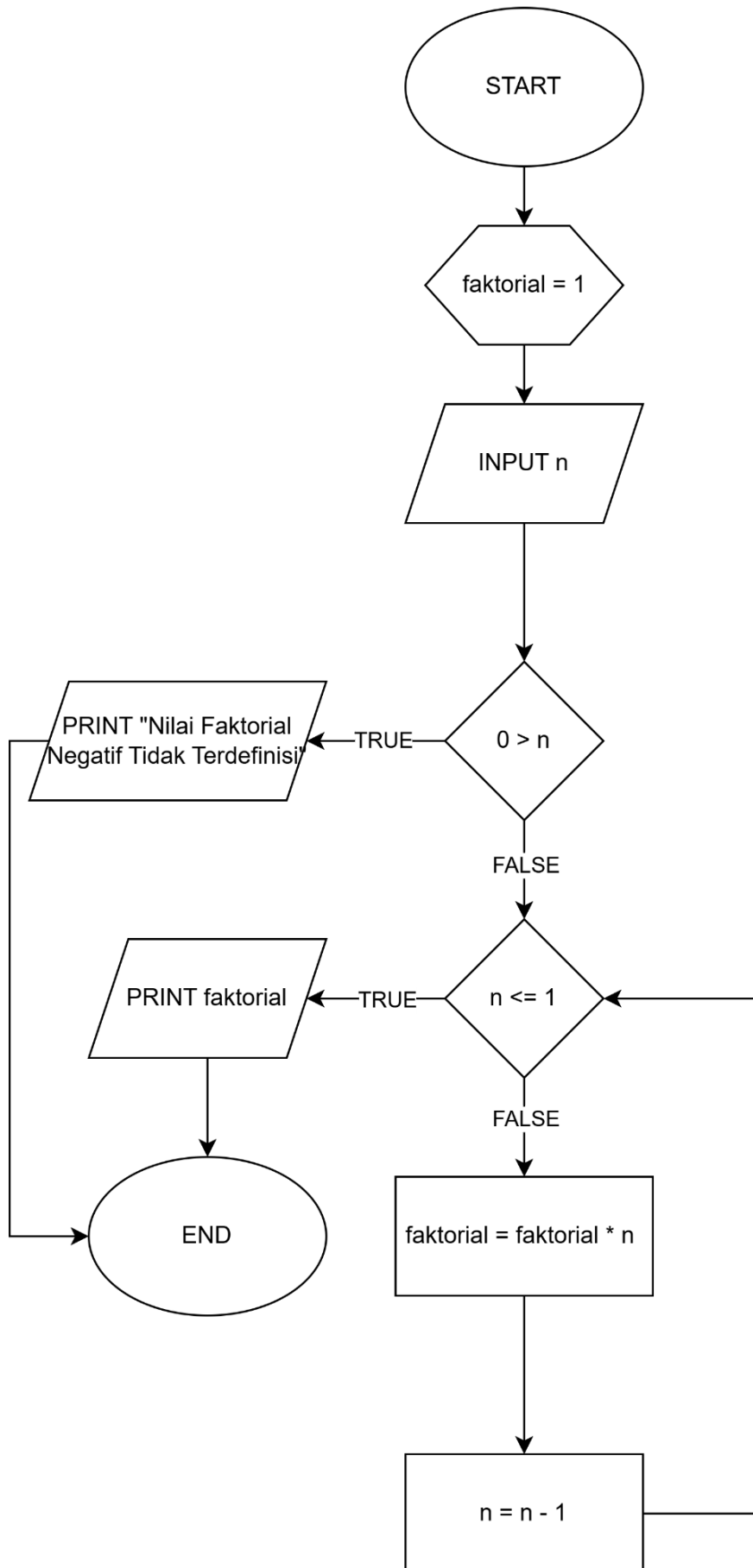
Gambar 2.4 Pseudocode Akar Kuadrat

#### Penjelasan:

Program diatas berfungsi untuk menentukan jenis akar dari persamaan kuadrat, pertama kita meminta user menginput a,b,c kemudian kita menghitung jenis akar persamaan kuadrat menggunakan rumus  $D = b^2 - 4ac$  setelah dihitung program akan berlanjut ke percabangan untuk menentukan jenis akarnya, pertama jika hasil  $D > 0$  maka akan menghasilkan "Akar real dan berlainan", jika  $D = 0$  akan menghasilkan "Akar real dan kembar", dan jika  $D < 0$  akan menghasilkan "Akar imajiner/ tidak nyata/ khayal" program selesai.

### SOAL 3

Source:



Gambar 2.5 Flowchart Faktorial

```
PROGRAM Faktorial

Deklarasi
    faktorial, n : Integer
    SET faktorial = 1

    READ n

    IF n < 0
        PRINT "Nilai Faktorial Negatif Tidak Terdefinisi"
    ELSE
        WHILE n >= 1
            SET faktorial = faktorial * n
            SET n = n - 1
        ENDWHILE
        PRINT faktorial
    ENDIF

END
```

Gambar 2.6 Pseudocode Faktorial

### Penjelasan:

Program di atas berfungsi untuk menghitung hasil faktorial dari nilai yang dimasukkan oleh user. Pertama kita mengetahui faktorial = 1, kemudian kita meminta user menginput nilai faktorial yang ingin dihitung kemudian program akan mengecek apakah nilai yang diinput > 0 karena jika nilai yang di input negatif tidak terdefinisi, jika user menginput lebih dari 0 maka program akan lanjut mengecek apakah nilai sama dengan 1 jika nilai sama dengan 1 maka program akan mengeluarkan 1 karena faktorial dari 1 = 1, jika lebih dari 1 akan lanjut menghitung nilai faktorial menggunakan rumus = (faktorial \* n) dan disimpan di dalam variabel faktorial, kemudian nilai n akan dikurangi 1 dan lanjut terus sampai nilai n = 1 baru kemudian nilai dari faktorial akhir akan di print kepada user.