



---

# 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                | 71251182                      |
| Nama Lengkap       | Michael Dylan                 |
| Minggu ke / Materi | 03 / Flowchart dan Pseudocode |

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

## **BAGIAN 1: MATERI MINGGU 1 (40%)**

### **3.3.1 Algoritma**

Algoritma adalah urutan langkah-langkah yang tersusun rapi dan logis untuk menyelesaikan suatu masalah. Tujuannya adalah memberi panduan langkah-langkah penyelesaian masalah yang mudah dimengerti oleh manusia, sehingga bisa dijadikan acuan saat membuat program komputer. Dengan memahami algoritma, kita bisa mengurangi kemungkinan terjadi kesalahan logika saat membuat program.

Penulisan algoritma dapat disajikan dalam tiga bentuk notasi, yaitu:

1. Uraian deskriptif,
2. Flowchart (diagram alir),
3. Pseudocode.

### **3.3.2 Uraian Deskriptif**

Algoritma ini digunakan untuk menghitung luas dan keliling sebuah lingkaran berdasarkan nilai jari-jari yang diketahui. Dengan algoritma ini, kita bisa mendapatkan hasil perhitungan secara sistematis dan mudah dipahami.

Langkah-langkah:

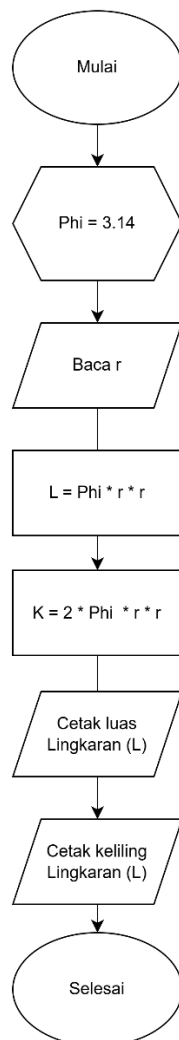
1. Masukkan nilai jari-jari lingkaran.
2. Hitung luas lingkaran dengan rumus  $L = \pi \times r^2$ .
3. Hitung keliling lingkaran dengan rumus  $K = 2 \times \pi \times r$ .
4. Tampilkan hasil luas lingkaran.
5. Tampilkan hasil keliling lingkaran.

Dari langkah-langkah di atas, untuk menghitung luas dan keliling lingkaran diperlukan nilai jari-jari sebagai input. Setelah itu, luas dan keliling dihitung menggunakan rumus yang sesuai, lalu hasilnya ditampilkan sebagai output. Proses ini memudahkan perhitungan secara sistematis dan terstruktur.

### 3.3.3 Flowchart / Diagram Alir

Flowchart adalah gambaran dari langkah-langkah algoritma yang dibuat dalam bentuk diagram alir. Flowchart digunakan untuk menunjukkan urutan proses dan arah jalannya suatu program agar lebih mudah dipahami.

Contoh: Flowchart dapat digunakan untuk menggambarkan proses menghitung luas dan keliling lingkaran, di mana langkah-langkah perhitungannya disajikan dalam bentuk diagram alir. Dengan flowchart, alur perhitungan dapat dilihat secara visual dari awal sampai akhir. Seperti dibawah ini.



Gambar 3.1: Flowchart mencari luas dan keliling lingkaran. Sumber <https://app.diagrams.net/>

Flowchart membantu analis dan programmer dalam memecahkan suatu masalah menjadi bagian-bagian yang lebih kecil sehingga lebih mudah dipahami atau di analisis. Selain itu, flowchart juga berguna untuk melihat berbagai kemungkinan solusi dalam proses pengolahan program.

















Kegunaan flowchart:

1. Digunakan untuk merancang atau mendesain program.
2. Digunakan untuk menggambarkan alur kerja suatu program.

Oleh karena itu, flowchart harus mampu merepresentasikan komponen-komponen yang ada dalam bahasa pemrograman agar mudah diterjemahkan ke dalam kode program.

### 3.3.4 Notasi Flowchart

Pada dasarnya, setiap simbol atau notasi dalam flowchart memiliki arti yang berbeda-beda. Notasi-notasi tersebut digunakan untuk menunjukkan jenis proses atau langkah tertentu dalam sebuah alur program. Seperti dibawah ini.

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

Gambar 3.2: Notasi Flowchart beserta fungsinya. Sumber

<https://www.caturyogam.info/2022/02/flowchart-diagram-alur.html>

Gambar di atas merupakan notasi dalam flowchart yang memiliki jenis dan fungsi yang berbeda-beda. Beberapa notasi digunakan untuk menghubungkan satu simbol dengan simbol lainnya, seperti flow, on-page reference, dan off-page reference. Selain itu, ada notasi yang berfungsi untuk menunjukkan proses yang sedang berlangsung, serta notasi yang digunakan untuk memasukkan input dan menampilkan output.

### 3.3.5 Pseudocode

Notasi ini menyerupai bentuk penulisan pada bahasa pemrograman tingkat tinggi seperti C dan Python. Struktur algoritma umumnya dibagi menjadi beberapa bagian, yaitu:

1. Bagian kepala (header), berisi nama atau judul algoritma.
2. Bagian deklarasi, berisi pendefinisian variabel yang digunakan.
3. Bagian deskripsi, berisi langkah-langkah atau proses yang dilakukan.

Contoh di bawah ini:

```
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
```

Pseudocode di atas digunakan untuk menghitung luas suatu persegi panjang jika nilai panjang dan lebarnya diketahui. Pada bagian deklarasi, ditentukan variabel yang digunakan yaitu panjang, lebar, dan luas dengan tipe data float. Selanjutnya, pada bagian deskripsi, program membaca nilai panjang dan lebar sebagai input. Setelah itu, luas dihitung dengan mengalikan panjang dan lebar. Terakhir, hasil perhitungan luas ditampilkan sebagai output.

### 3.3.6 Notasi Pseudocode

Notasi-notasi dalam pseudocode digunakan sebagai simbol atau kata kunci untuk menggambarkan proses yang terjadi dalam suatu algoritma. Dengan adanya notasi ini, algoritma dapat dituliskan secara terstruktur sehingga mudah dibaca, dipahami, dan diterjemahkan ke dalam bahasa pemrograman apa pun. Pseudocode tidak terikat pada satu bahasa tertentu, tetapi lebih menekankan pada alur logika penyelesaian masalah.

1. INPUT digunakan untuk menunjukkan proses memasukkan data ke dalam program. Data ini bisa berasal dari pengguna, file, atau sumber lain, dan akan disimpan ke dalam variabel untuk diproses lebih lanjut. Tanpa adanya proses input, program tidak memiliki data awal yang dapat diolah.
2. OUTPUT digunakan untuk menampilkan hasil dari proses yang telah dilakukan oleh program. Output dapat berupa nilai, informasi, atau pesan yang ditampilkan ke layar, disimpan ke file, atau dikirim ke perangkat lain. Output berfungsi sebagai bentuk informasi akhir yang dibutuhkan oleh pengguna.
3. WHILE digunakan untuk melakukan perulangan selama kondisi yang diberikan bernilai benar. Pada perulangan ini, kondisi akan dicek di awal. Jika kondisi masih benar, maka perintah di dalam WHILE akan dijalankan secara terus-menerus sampai kondisi tersebut menjadi salah.
4. FOR digunakan untuk perulangan dengan jumlah iterasi yang sudah ditentukan sejak awal. Biasanya digunakan ketika kita sudah mengetahui berapa kali suatu proses harus diulang, misalnya untuk menghitung data dalam jumlah tertentu atau melakukan proses secara berurutan.
5. REPEAT – UNTIL digunakan untuk perulangan yang memiliki kondisi di akhir. Artinya, perintah di dalam REPEAT akan dijalankan terlebih dahulu, baru kemudian kondisi pada UNTIL diperiksa. Perulangan akan terus dilakukan sampai kondisi pada UNTIL bernilai benar.
6. IF – THEN – ELSE digunakan untuk pengambilan keputusan berdasarkan suatu kondisi tertentu. Jika kondisi yang diuji bernilai benar, maka perintah pada bagian IF akan dijalankan. Jika kondisi bernilai salah, maka perintah pada bagian ELSE yang akan dijalankan. Struktur ini sangat penting dalam algoritma karena memungkinkan program memilih tindakan yang berbeda sesuai kondisi yang terjadi.

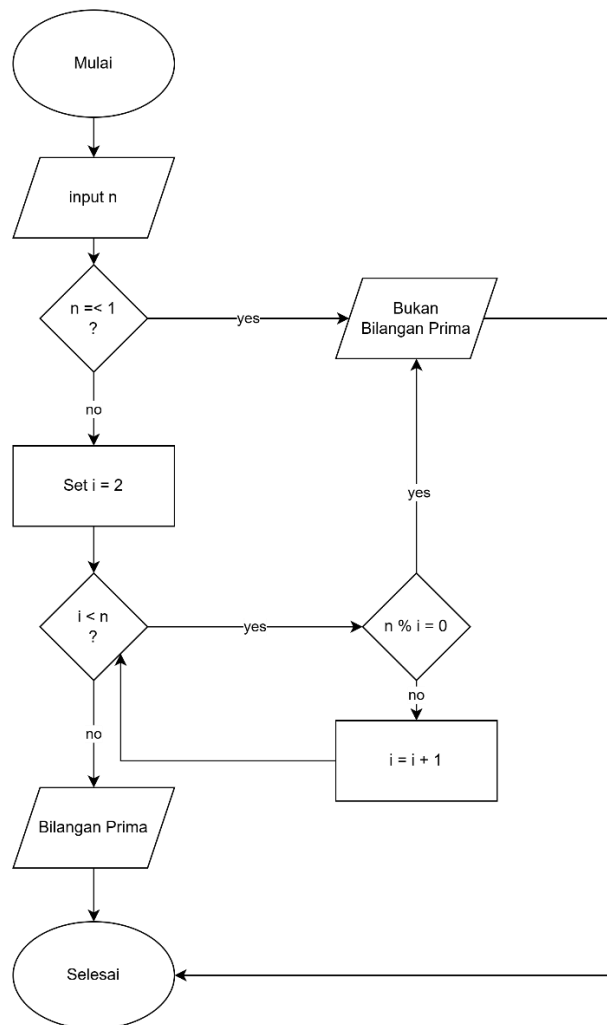
## BAGIAN 2: LATIHAN MANDIRI (60%)

Link Github = [https://github.com/MichaelDylan-ti/71251182\\_michaeldylan/tree/main](https://github.com/MichaelDylan-ti/71251182_michaeldylan/tree/main)

### LATIHAN 3.1

Buatlah algoritma untuk menentukan apakah suatu bilangan merupakan bilangan prima atau bukan. Algoritma yang dibuat disajikan dalam flowchart dan pseudocode.

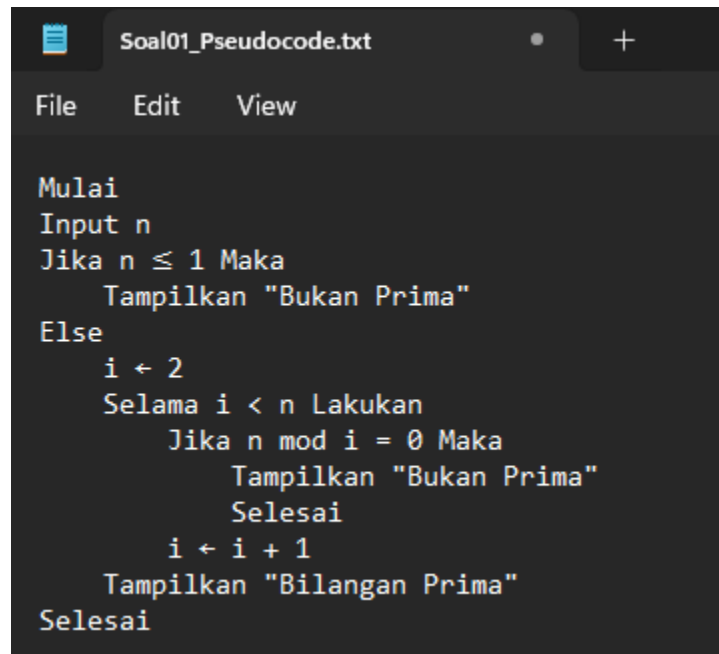
#### Flowchart



Gambar 3.3: Flowchart Latihan 3.1. Sumber <https://app.diagrams.net/>

Flowchart mengecek apakah bilangan  $n$  prima. Langkahnya: input  $n$ , jika  $n \leq 1 \rightarrow$  langsung Bukan Prima. Jika tidak, mulai dari  $i = 2$  cek  $i < n$ . Jika  $n$  habis dibagi  $i \rightarrow$  Bukan Prima, kalau tidak  $\rightarrow i$  naik 1 dan ulang. Jika semua  $i$  dicek tanpa habis dibagi  $\rightarrow$  Bilangan Prima.

### Pseudocode



```
Mulai
Input n
Jika  $n \leq 1$  Maka
    Tampilkan "Bukan Prima"
Else
     $i \leftarrow 2$ 
    Selama  $i < n$  Lakukan
        Jika  $n \bmod i = 0$  Maka
            Tampilkan "Bukan Prima"
        Selesai
         $i \leftarrow i + 1$ 
    Tampilkan "Bilangan Prima"
Selesai
```

Gambar 3.4: Pseudocode Latihan 3.1. Sumber NotePad

Program mulai dengan input  $n$ . Jika  $n \leq 1$ , jelas bukan prima. Kalau lebih dari 1, dicek satu per satu dari 2 sampai  $n-1$ . Kalau ketemu pembagi selain 1 dan  $n$  sendiri  $\rightarrow$  bukan prima. Kalau tidak ketemu  $\rightarrow$  bilangan prima.

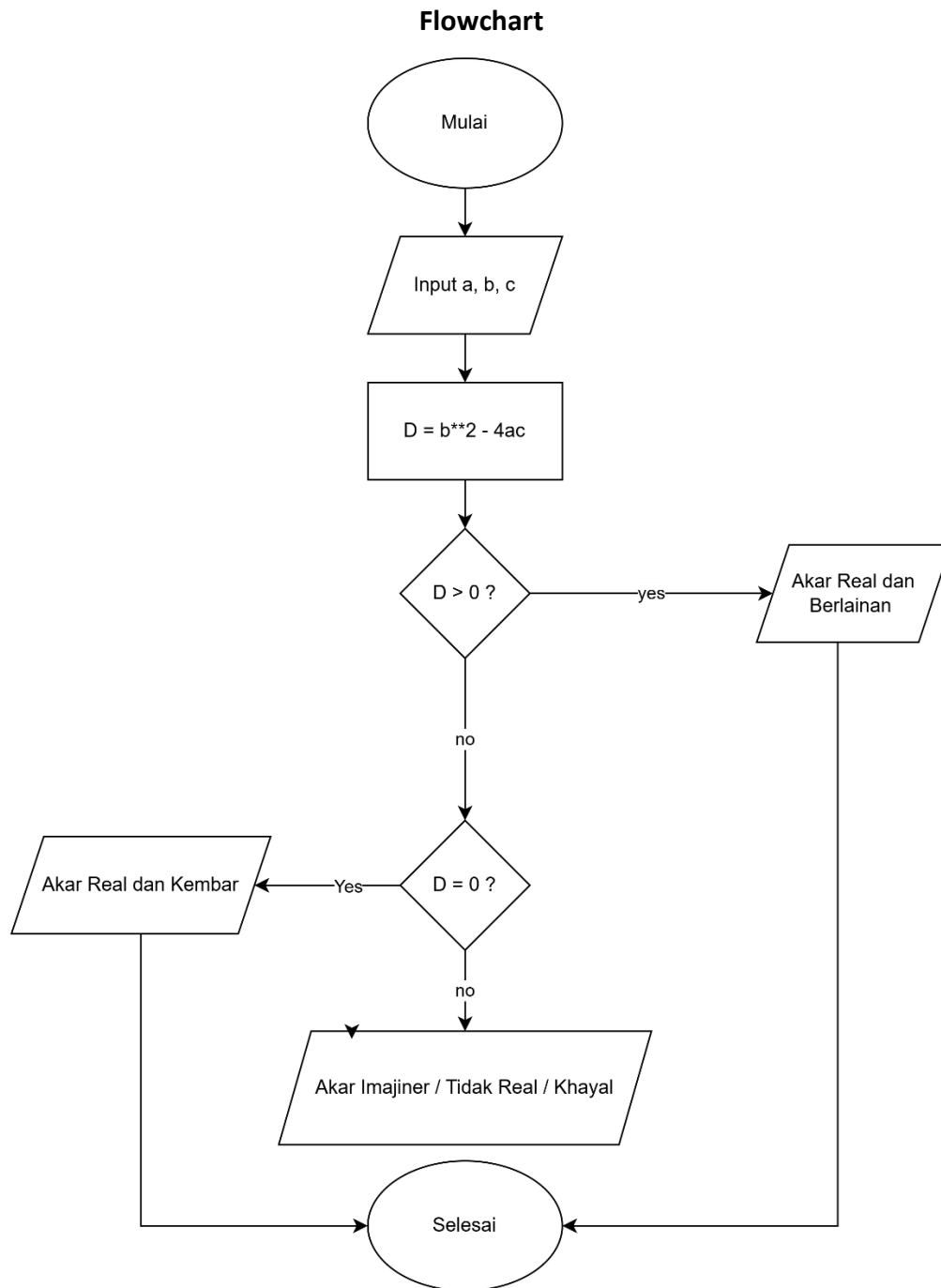
### LATIHAN 3.2

Buatlah algoritma untuk menentukan jenis akar dari suatu persamaan kuadrat. Jenis akar persamaan kuadrat sebagai berikut:

1.  $D \geq 0$ ; Akar Real/Nyata.
2.  $D > 0$ ; Akar Real dan Berlainan.
3.  $D = 0$ ; Akar Real dan Kembar.
4.  $D < 0$ ; Akar Imajiner/Tidak Real/Khayal.
5. Selesai.

Algoritma yang dibuat disajikan dalam flowchart dan pseudocode

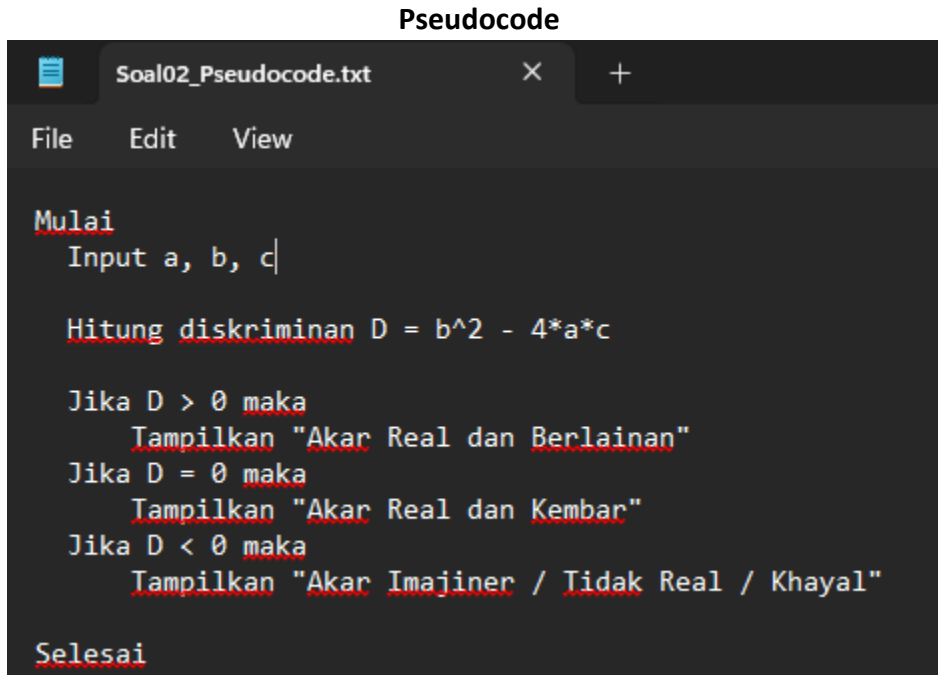




Gambar 3.5: Pseudocode Latihan 3.2. Sumber <https://app.diagrams.net/>

Flowchart dimulai dengan Mulai, kemudian meminta pengguna untuk input nilai a, b, dan c dari persamaan kuadrat. Selanjutnya, sistem menghitung diskriminan  $D = b^2 - 4ac$ . Dari sini, alurnya logis: jika  $D > 0$ , tampilkan "Akar Real dan Berlainan"; jika  $D = 0$ , tampilkan "Akar Real dan Kembar"; jika  $D < 0$ , tampilkan "Akar Imajiner/Tidak Real/Khayal".

**Pseudocode**



```
Mulai
  Input a, b, c

  Hitung diskriminan D = b^2 - 4*a*c

  Jika D > 0 maka
    Tampilkan "Akar Real dan Berlainan"
  Jika D = 0 maka
    Tampilkan "Akar Real dan Kembar"
  Jika D < 0 maka
    Tampilkan "Akar Imajiner / Tidak Real / Khayal"

Selesai
```

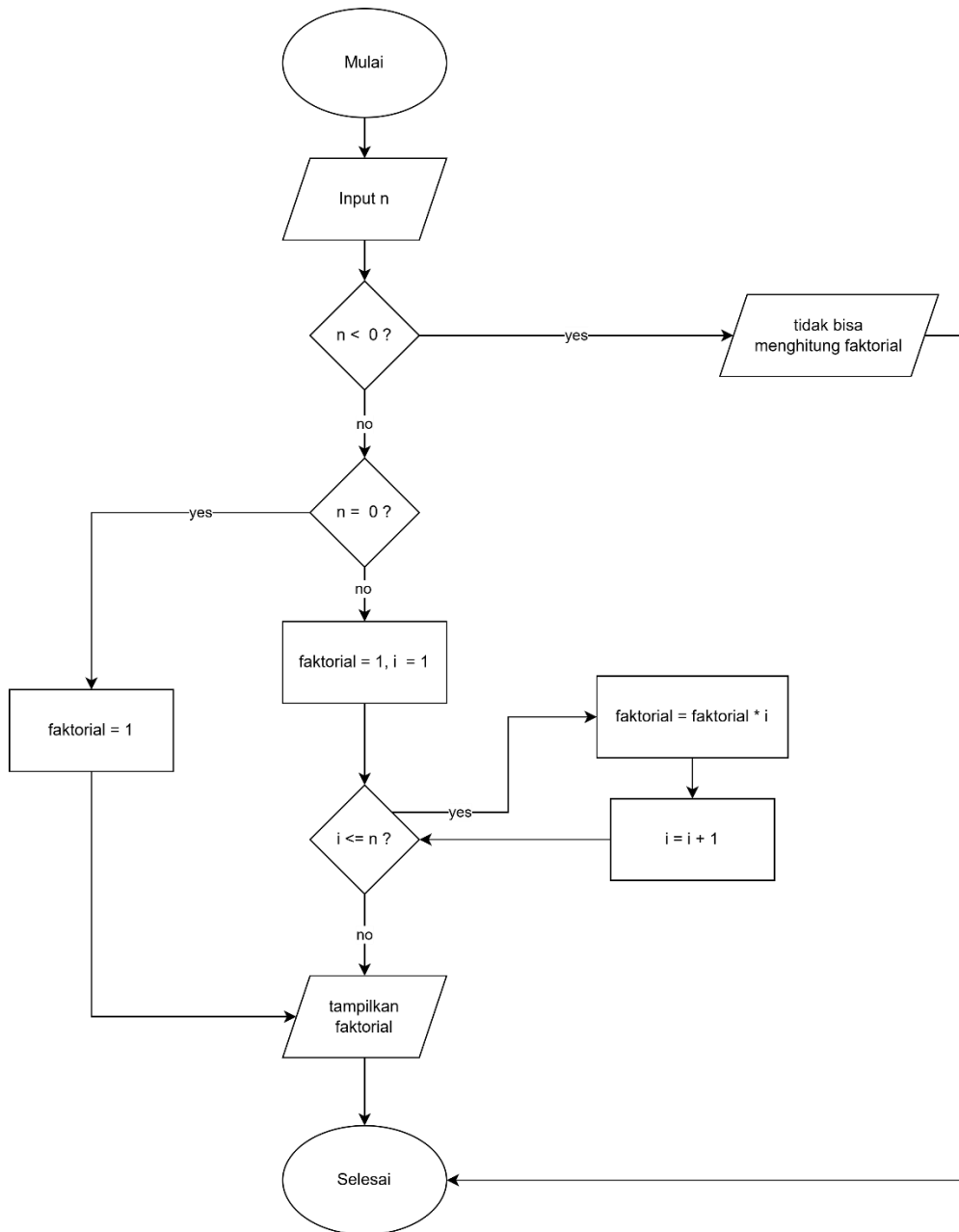
Gambar 3.6: Pseudocode Latihan 3.2. Sumber NotePad

Pseudocode-nya mengikuti logika yang sama: mulai dengan input a, b, c, hitung D, lalu gunakan percabangan sederhana untuk menentukan jenis akar. Kondisi  $D > 0$ ,  $D = 0$ , dan  $D < 0$  langsung menentukan output masing-masing.

### LATIHAN 3.3

Buatlah algoritma untuk menghitung nilai faktorial dari suatu bilangan. Algoritma yang dibuat disajikan dalam flowchart dan pseudocode.

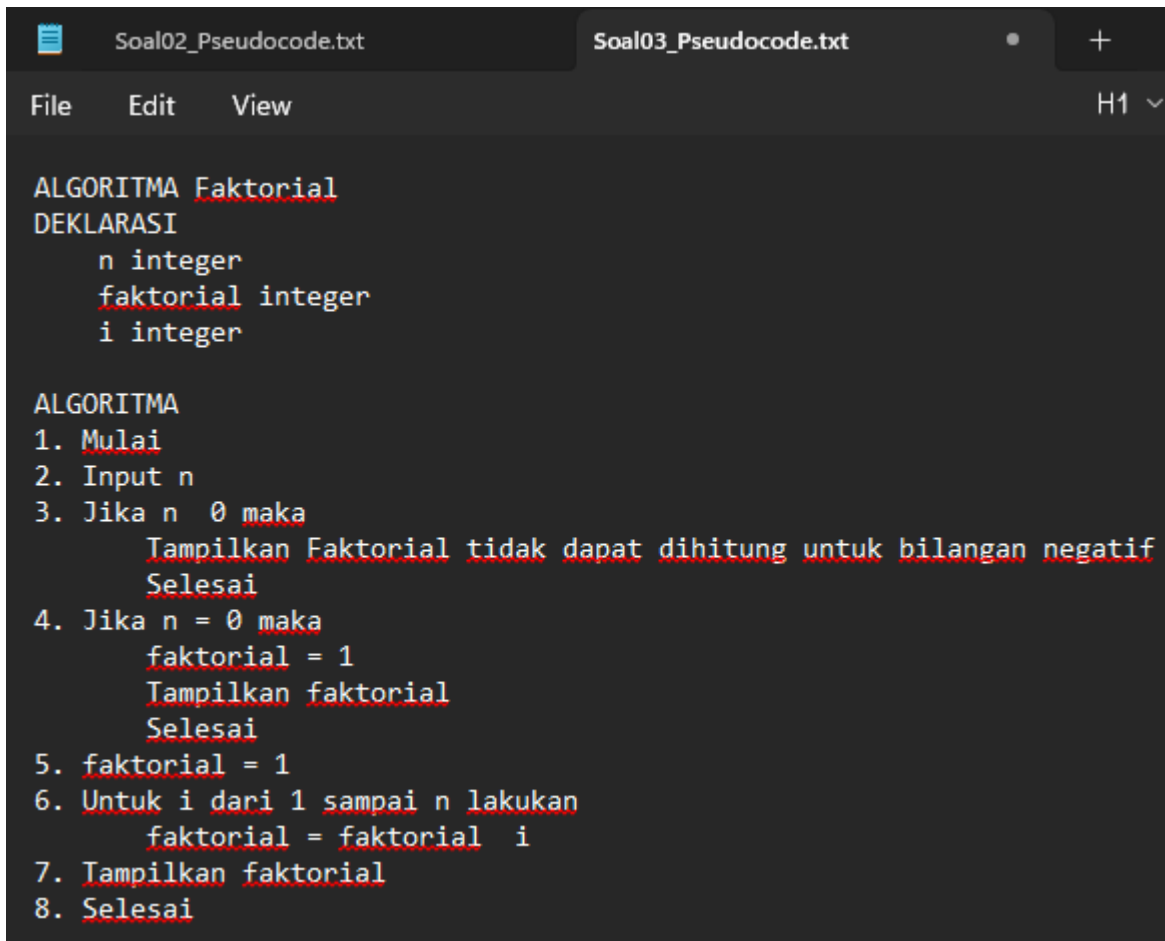
### Flowchart



Gambar 3.7: Pseudocode Latihan 3.3. Sumber <https://app.diagrams.net/>

Flowchart faktorial mulai dengan input bilangan  $n$ . Jika  $n$  negatif, langsung tampil pesan “tidak bisa dihitung”. Jika  $n$  nol, hasil faktorial 1 dan selesai. Untuk bilangan positif, algoritma menginisialisasi faktorial = 1 lalu mengalikan faktorial dengan setiap angka dari 1 sampai  $n$  menggunakan loop. Setelah loop selesai, hasil faktorial ditampilkan.

## Pseudocode



```
Soal02_Pseudocode.txt  Soal03_Pseudocode.txt
File Edit View H1 v

ALGORITMA Faktorial
DEKLARASI
    n integer
    faktorial integer
    i integer

ALGORITMA
1. Mulai
2. Input n
3. Jika n < 0 maka
    Tampilkan Faktorial tidak dapat dihitung untuk bilangan negatif
    Selesai
4. Jika n = 0 maka
    faktorial = 1
    Tampilkan faktorial
    Selesai
5. faktorial = 1
6. Untuk i dari 1 sampai n lakukan
    faktorial = faktorial * i
7. Tampilkan faktorial
8. Selesai
```

Gambar 3.8: Pseudocode Latihan 3.3. Sumber NotePad

Pseudocode-nya sama saja, tapi ditulis seperti resep masak: mulai → input → cek negatif/0 → kalau positif pakai loop dari 1 sampai n → kalikan → tampilkan. Yang bikin mahasiswa asdos kadang bingung itu cuma di loop dan perkalian berulang, karena kalau salah indeks atau urutan, hasilnya bisa salah. Jadi intinya, flowchart buat lihat alur, pseudocode buat bikin kode nyata tapi keduanya sama-sama ngasih “rute jelas” dari mulai sampai selesai.