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	**<ISI DENGAN NIM ANDA>**
Nama Lengkap	**<ISI DENGAN NAMA LENGKAP ANDA>**
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 adalah urutan langkah-langkah yang disusun secara logis dan teratur untuk menyelesaikan sebuah problem atau masalah. Tujuan dari algoritma sendiri adalah untuk memberikan langkah-langkah logis untuk menyelesaikan masalah dalam bentuk yang mudah dipahami oleh manusia. Contoh dari algoritma adalah resep makanan yaitu algoritma untuk membuat suatu makanan. Adapun karakteristik untuk algoritma yang baik adalah:

- Jelas dan terdefinisi = setiap langkah harus dapat dimengerti dan tidak ada keambiguan dalam penulisan.
- Terbatas (finite) = algoritma harus selesai.
- Efisien = menggunakan langkah seminimal mungkin.
- Input dan output = algoritma yang baik dapat menerima input dan dapat mengeluarkan output

Adapun tiga jenis penulisan algoritma antara lain yaitu :

- Uraian dekriptif
- *Flowchart* / diagram alir
- *Pseudocode*

Uraian Deskriptif

Bentuk ini paling sederhana karena ditulis dengan menggunakan bahasa natural (dalam hal ini menggunakan bahasa Indonesia) dan tidak terkait dengan bahasa pemrograman. Uraian deskriptif sendiri berisi langkah-langkah yang harus dilakukan untuk mendapatkan hasil sesuai dengan masalah atau tujuan. Kelebihan dari uraian deskriptif ini adalah mudah untuk dipahami oleh semua orang (bukan hanya programmer) dan cocok untuk penjelasan yang sederhana. Namun, kekurangan dari cara ini adalah kurang efektif untuk masalah yang kompleks dan membingungkan kalau langkahnya terlalu banyak.

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.

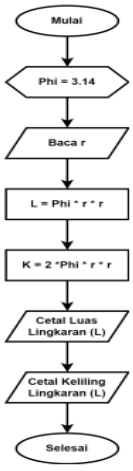
Gambar 3.1: Contoh uraian deskriptif menyelesaikan permasalahan menghitung luas dan keliling suatu lingkaran. *Sumber:* https://l1nq.com/NHwNG

Flowchart (Diagram Alir)

Flowchart adalah cara menuliskan algoritma dalam bentuk gambar atau diagram yang mana setiap langkah diwakili oleh simbol tertentu. Adapun fungsi dari flowchart sendiri yaitu:

- Memvisualisasikan proses
- Meningkatkan komunikasi
- Membantu analisis dan perbaikan proses
- Dokumentasi sistem
- Medukung perencanaan dan desain

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



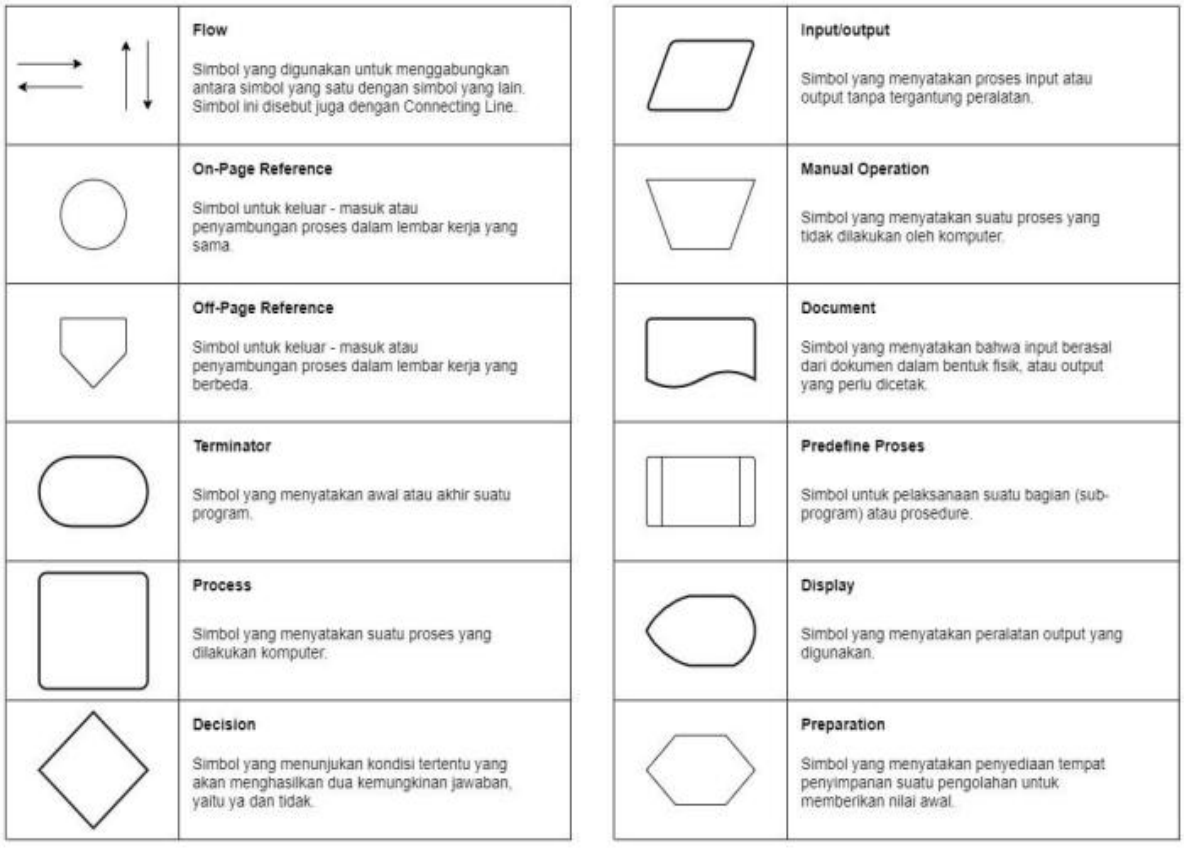
Gambar 3.2: Flowchart mencari luas dan keliling lingkaran. *Sumber:* https://l1nq.com/NHwNG

Secara umum, flowchart memiliki beberapa kegunaan penting:

- Untuk mendesain program, tujuannya supaya kita memiliki gambaran jelas tentang bagimana program akan berjalan sehingga saat membuat code hanya perlu mengikuti alur yang sudah dirancang.
- Untuk memperesentasikan program, hal ini dilakukan dengan tujuan agar orang lain dapat memahami alut program hanya dengan melihat diagram tanpa perlu membaca keseluruhan kode.

Notasi Flowchart

Dalam flowchart, setiap simbol memiki arti yang berbeda. Masing-masing memiliki fungsi tersendiri dan perlu dimengerti agar alur tidak membingungkan.



Gambar 3.3: Notasi flowchart. *Sumber:* https://l1nq.com/NHwNG

Adapun simbol-simbol yang sering digunakan yaitu:

1. Terminator (bentuknya oval) digunakan untuk menandai awal dan ahkir dari sebuah program.
2. Flowline (bentuknya garis) digunakan untuk menghubungkan simbol dengan simbol lain serta menunjukkan arah dari proses.
3. Process (bentuknya persegi) digunakan utnuk memunjukkan adanya sebuah proses dalam program.
4. Input/Output (bentuknya jajar genjang) digunakan untuk menerima input dan menampillkan output.
5. Decision (bentuknya belah ketupat) digunakan untuk percabangan atau pengambilan keputusan berdasatkan kondisi.

Pseudocode

Pseudocode adalah cara menuliskan algoritma yang bentuknya mirip bahasa pemrograman tingkat tinggi seperti C dan Pyhton. Secara umum, pseudocode dibagi menjadi beberapa bagian:

1. Bagian kepala (header), berisi nama algoritma dan penjelasan singkat tentang apa yang dilakukan algoritma tersebut.
2. Bagian deklarasi, berisi definisi variabel yang akan digunakan, termasuk tipe datanya.
3. Bagian deskripsi, berisi langkah-langkah penyelesaian masalah secara urut.

Contoh:

```
program ganjil_genap

deklarasi
var number : integer

algoritma:
INPUT number
IF (number modulus 2 = 0) THEN
        OUTPUT "genap"
ELSE
        OUPUT "ganjil"
```

Gambar 3.4: pseudocode untuk menentukan ganjil dan genap. *Sumber:* https://shorturl.at/Rap6U

Notasi Pseudocode

Adapun notasi yang sering digunakan dalam pseudocode:

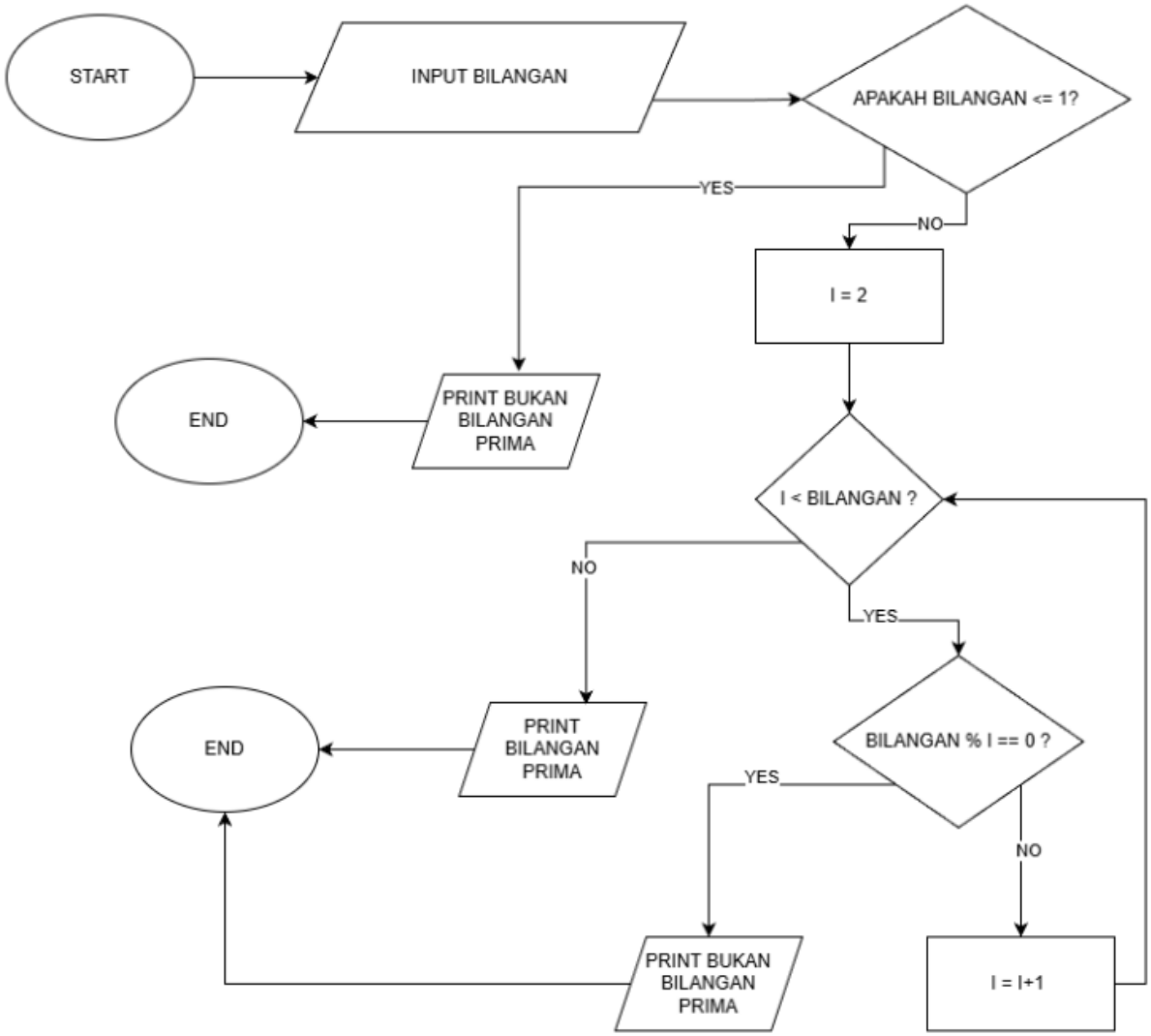
1. Input = digunakan untuk memasukan nilai dari suatu variabel.
2. Output = digunakan untuk memunculkan nilai dari suatu variabel.
3. While = digunakan untuk perulangan yang memiliki iterasi awal.
4. For = digunakan untuk perulangan perhitungan iterasi.
5. Repeat – Until = digunakan untuk sebuah perulangan yang memiliki kondisi ahkir.
6. IF - Then - Else = digunakan untuk mengambil Keputusan dari beberapa kondisi.

BAGIAN 2: LATIHAN MANDIRI (60%)

Link github: https://github.com/angjoshua-sudo/Prak-Alpro-2026.git

SOAL 1

Flowchart:



Pseudocode:

```
PROGRAM MENENTUKAN_BILANGAN_PRIMA

DEKLARASI
variable BILANGAN, I : integer

DESKRIPSI

INPUT BILANGAN
IF (BILANGAN <= 1) THEN
        OUTPUT BUKAN BILANGAN PRIMA

ELSE
        I = 2

        WHILE I < BILANGAN THEN
                IF (BILANGAN % I == 0) THEN
                        OUTPUT BUKAN BILANGAN PRIMA
                ELSE
                        I = I + 1
        ELSE
                OUTPUT BILANGAN PRIMA
```

Penjelasan :

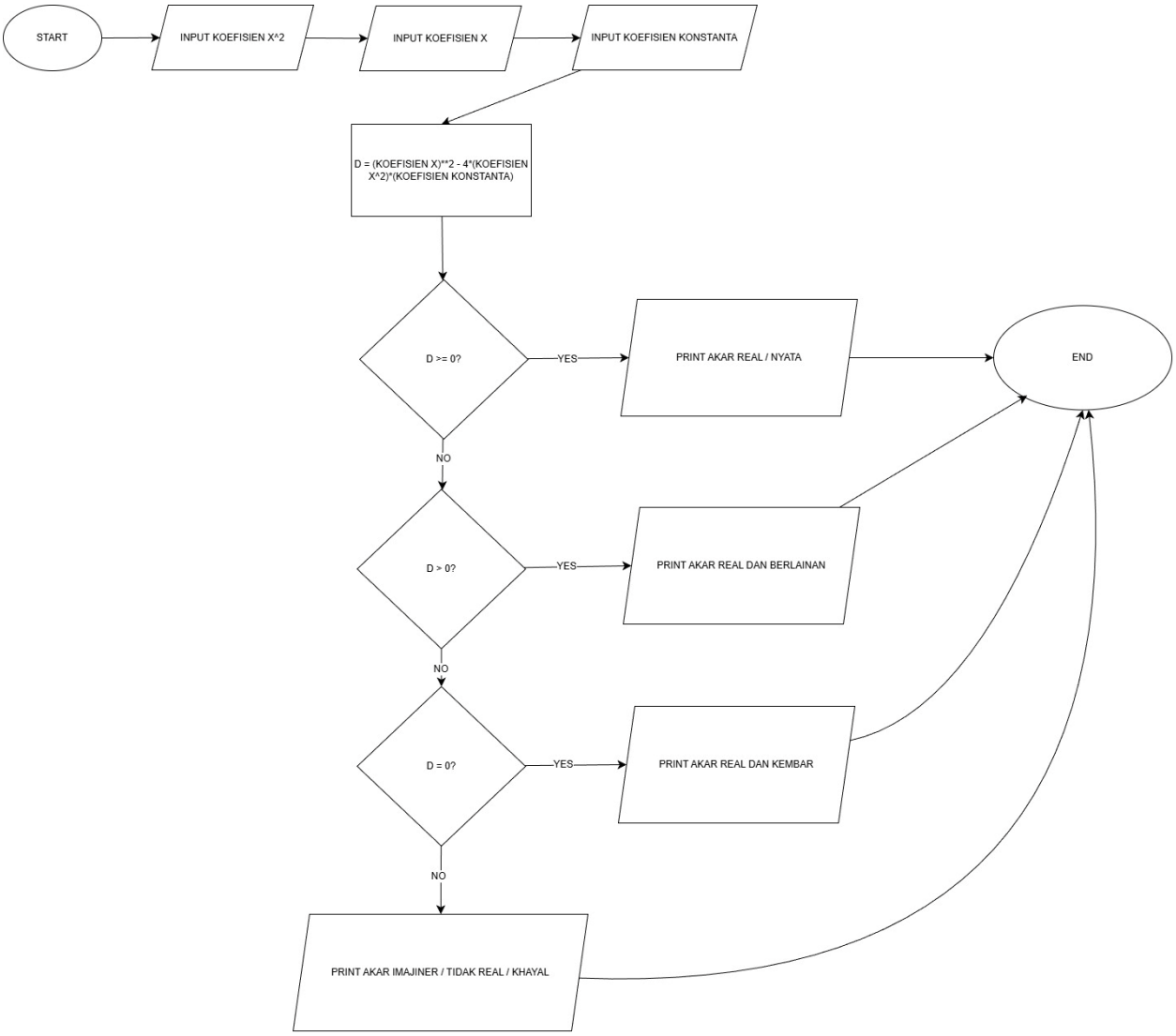
Flowchart dan pseudocode ini bertujuan untuk menentukan **sebuah bilangan prima atau tidak**. Untuk flowchart pertama diawali dengan **START** untuk menunjukkan dimulainya program. Selanjutnya pengguna diminta untuk menginput sebuah angka yang akan di cek dan disimpan ke dalam variabel **BILANGAN** sebagai nilai yang akan diproses.

Setelah nilai berhasil dimasukkan, program melakukan pengecekan awal terhadap kondisi **BILANGAN <= 1**. Pemeriksaan ini dilakukan untuk memastikan bilangan memenuhi syarat awal bilangan prima. Jika kondisinya **YES** maka maka program akan menampilkan output "**BUKAN BILANGAN PRIMA"** dan program selesai. Hal ini karena bilangan prima didefinisikan sebagai bilangan yang lebih besar dari 1 dan hanya memiliki dua faktor, yaitu 1 dan dirinya sendiri. Namun, jika kondisi **NO**, maka program melanjutkan proses dengan menetapkan **I = 2** sebagai pembagi awal.

Setelah itu, program masuk ke tahap perulangan dengan kondisi **I < BILANGAN** untuk menguji apakah terdapat angka yang dapat membagi bilangan tersebut. Di dalam perulangan, dilakukan pengecekan menggunakan operasi modulus yaitu **BILANGAN % I == 0**. Jika kondisi tersebut terpenuhi, berarti bilangan memiliki faktor lain sehingga dinyatakan "**BUKAN BILANGAN PRIMA"** dan program dihentikan. Jika tidak tepenuhi, maka nilai I akan ditambah 1 dan proses diulang kembali. Apabila perulangan selesai tanpa menemukan pembagi yang menghasilkan sisa nol, maka bilangan tersebut akan menghasilkan "**BILANGAN PRIMA"** dan program berahkir dengan simbol **END**.

SOAL 2

Flowchart:



Pseudocode:

```
MENENTUKAN AKAR_PERSAMAAN_KUADRAT


DEKLARASI
variable KOEFISIEN X^2, KOEFISIEN X, KOEFISIEN KONSTANTA : INTEGER

DESKRIPSI
INPUT KOEFISIEN X^2
INPUT KOEFISIEN X
INPUT KOEFISIEN KONSTANTA

D = (KOEFISIEN X)**2 - 4*(KOEFISIEN X^2)*(KOEFISIEN KONSTANTA)

IF D >= 0 THEN
        OUTPUT AKAR REAL / NYATA
ELSE
        IF D > 0 THEN
                OUTPUT AKAR REAL DAN BERLAWANAN
        ELSE
                IF D = 0 THEN
                        OUTPUT AKAR REAL DAN KEMBAR
                ELSE
                        OUTPUT AKAR IMAJINER / TIDAK REAL / KHAYAL
```

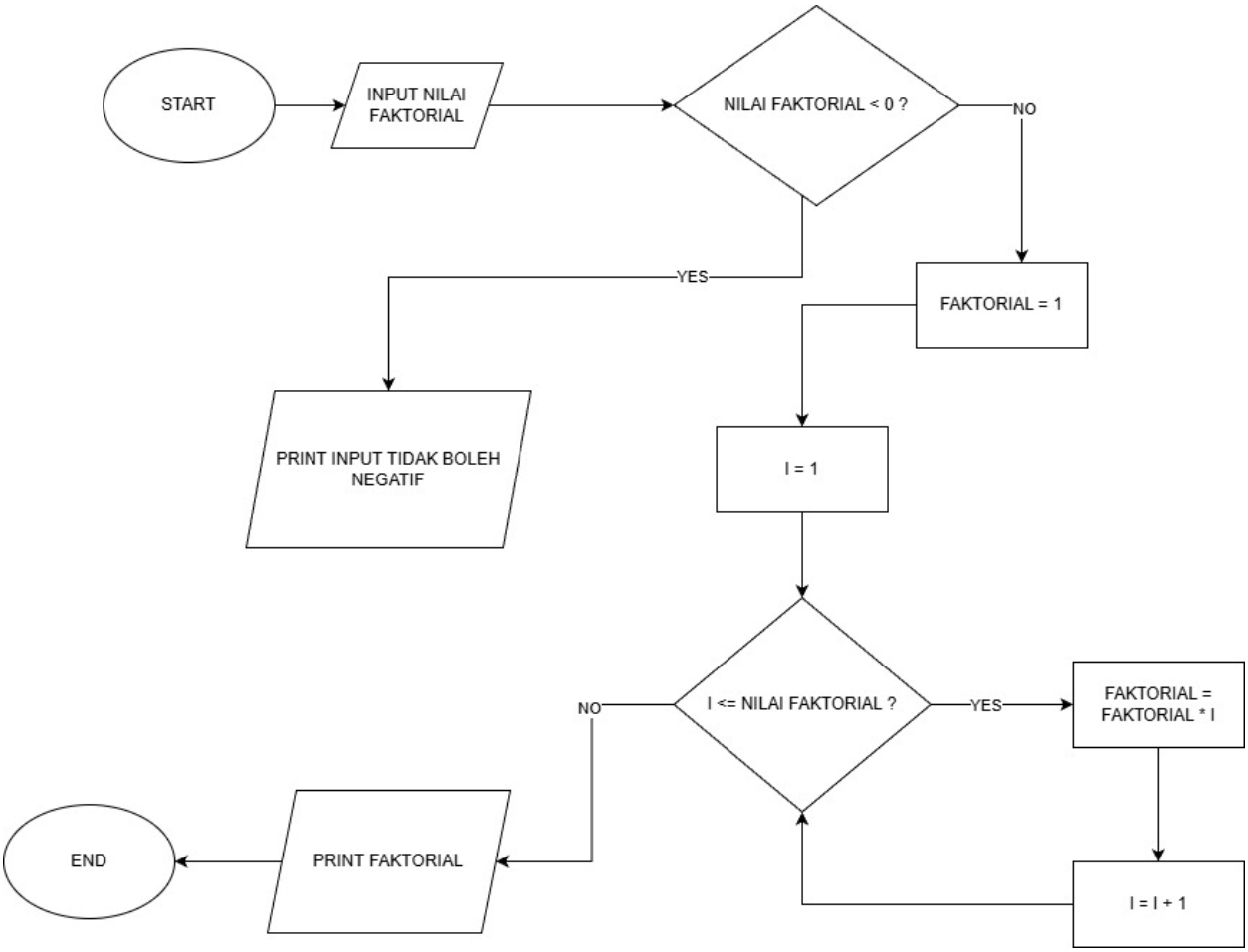
Penjelasan:

Flowchart dan pseudocode ini bertujuan untuk menentukan **jenis akar dari suatu persamaan kuadrat**. Proses pada flowchart diawali dengan simbol **START** yang menandakan dimulainya progam. Selanjutnya, pengguna diminta untuk menginput tiga nilai, yaitu **KOEFISIEN X^2**, **KOEFISIEN X**, dan **KOEFISIEN KONSTANTA**. Ketiga nilai tersebut merupakan komponen utama yang akan di hitung program yang dimasukkan ke rumus **D = (KOEFISIEN X)^2 - 4*(KOEFISIEN X^2)*(KOEFISIEN KONSTANTA)**. Hasil perhitungan ini kemudian menjadi dasar dalam proses pengambilan keputusan melalui beberapa tahap percabangan logika.

Pada tahap pengecekan pertama, program menguji apakah nilai **D >= 0**. Jika kondisi ini terpenuhi (**YES**), maka program akan langsung menampilkan output "**AKAR REAL/NYATA**" dan alur kerja berahkir di simbol **END**. Namun, kondisi tidak terpenuhi maka akan dilanjutkan ke percabangan selanjutnya. Jika nilai **D > 0**, maka program akan menyatakan bahwa akar-akarnya adalah "**AKAR REAL DAN BERLAINAN**". Sementara itu, jika nilai **D = 0**, maka hasilnya adalah "**AKAR REAL DAN KEMBAR"**. Sebaliknya, apabila nilai diskriminan tidak memenuhi kategori di atar (**NO**), makan program akan mengategorikan sebagai **AKAR IMAJINER / TIDAK REAL / KHAYAL**. Setelah salah satu output ditampilkan, seluruh alur proses akan bertemu pada titik terminal **END**.

SOAL 3

Flowchart:



Pseudocode:

```
MENGHITUNG_FAKTORIAL

DEKLARASI
variable NILAI FAKTORIAL, FAKTORIAL, I : integer

DESKRIPSI
INPUT NILAI FAKTORIAL

IF NILAI FAKTORIAL < 0 THEN
        OUTPUT TIDAK BOLEH NEGATIF
ELSE
        FAKTORIAL = 1
        I = 1

        WHILE I <= NILAI FAKTORIAL
                FAKTORIAL = FAKTORIAL*I
                I = I + 1

        OUTPUT FAKTORIAL
```

Penjelasan:

Algoritma ini berfungsi untuk menghitung nilai factorial dari sebuah bilangan bulat yang dimasukkan pengguna. Alur program dimulai dengan simbol **START**, yang kemudian diikuti dengan proses pengambilan input berupa **NILAI FAKTORIAL**. Langkah awal yang dilakukan program adalah valillidasi data, apakah nilai yang dimasukkan kurang dari nol (**NILAI FAKTORIAL < 0 ?**). Jika kondisinya **YES** (bernilai negatif), maka program secara langsung akan menampilkan output "**INPUT TIDAK BOLEH NEGATIF**" dan langsung berhenti. Hal ini dikarenkan dalam matematika, faktorial hanya untuk bilangan positif.

Jika input sudah valid atau bernilai **NO** pada pengecekan negatif, program akan melanjutkan ke menetapkan variabel **FAKTORIAL = 1** dan variabel pencacahan **I = 1**. Setelah itu, program masuk ke dalam struktur perulangan (looping) dengan kondisi **I <= NILAI FAKTORIAL**. Selama kondisi tersebut (**YES**), program akan terus melakukan perhitungan akumulatif dengan mengalikan nilai faktor sebelumnya dengan nilai **I**, lalu menambah nilai **I** sebesar satu (**I = I + 1**) di setiap perulangannya. Perulangan ini akan terus berjalan hingga nilai I melampaui angka input yang diberikan. Ketika kondisi perulangan ahkirnya tidak terpenuhi (**NO**), program akan keluar dari loop, menampilkan hasil ahkir berupa **TAMPILKAN FAKTORIAL**, dan berahkir pada simbol **END**.