



---

# 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	71251189
Nama Lengkap	Alicia Luna Santoso
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%)

Pada bagian ini, tuliskan kembali semua materi yang telah anda pelajari minggu ini. Sesuaikan penjelasan anda dengan urutan materi yang telah diberikan di saat praktikum. Penjelasan anda harus dilengkapi dengan contoh, gambar/ilustrasi, contoh program (source code) dan outputnya. Idealnya sekitar 5-6 halaman.

Link Github : [https://github.com/alc357/71251189\\_alicia.git](https://github.com/alc357/71251189_alicia.git)

## ALGORITMA

Algoritma adalah rangkaian terurut atau langkah-langkah logis dan sistematis yang dibuat untuk menyelesaikan masalah. Tujuan dibuatnya algoritma adalah memberikan petunjuk berupa langkah-langkah logika penyelesaian masalah dalam bentuk yang mudah dipahami nalar manusia sebagai acuan yang membantu dalam pengembangan program computer. Pemahaman terhadap algoritma akan mencegah terjadinya kesalahan logika pada program komputer yang dikembangkan. Penulisan (**Notasi Algoritma**) ada tiga macam yaitu uraian deskriptif, *Flowchart* (diagram alir), dan *Pseudocode*.

## URAIAN DESKRIFTIF

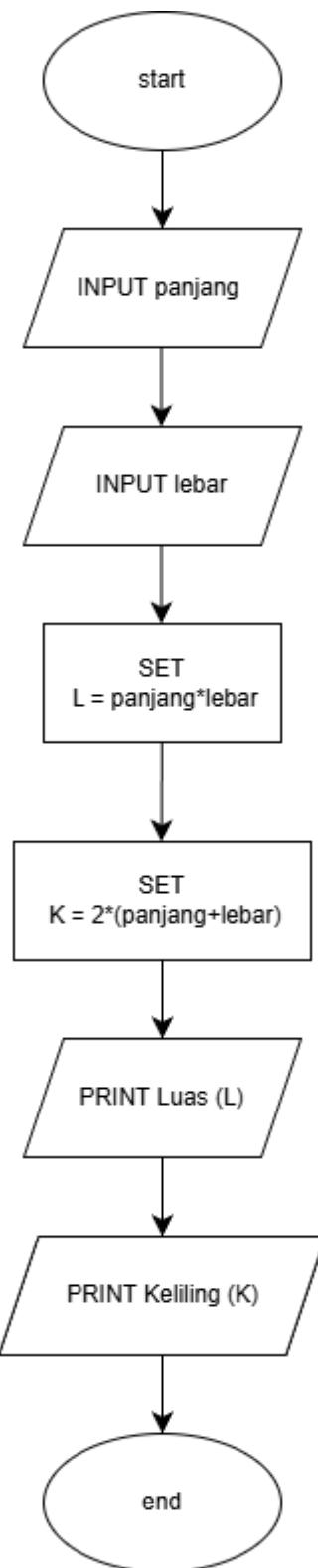
Contoh : Menyelesaikan permasalahan menghitung luas dan keliling persegi panjang  
Algoritma Hitung\_Luas\_Keliling\_PersegiPanjang.

1. Masukkan panjang persegi panjang (p).
2. Masukkan lebar persegi panjang (l).
3. Hitung luas persegi panjang dengan rumus  $L = p * l$ .
4. Hitung keliling persegi panjang dengan rumus  $K = 2 * (p + l)$ .
5. Tampilkan luas persegi panjang.
6. Tampilkan keliling persegi panjang.

## FLOWCHART / DIAGRAM ALIR

Flowchart merupakan gambar dalam bentuk diagram alir dari algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

Contoh : Menghitung luas dan keliling persegi panjang yang algoritmanya dinotasikan dalam bentuk diagram alir (*flowchart*)



Gambar 1.1 : Contoh pembuatan flowchart

Flowchart menolong analisis dan programmer dalam memecahkan masalah ke dalam segmen-segmen yang lebih kecil dan menolong dalam menganalisis alternatif – alternatif lain dalam proses pengoperasian.

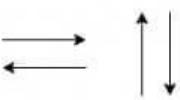
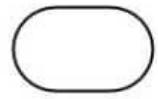
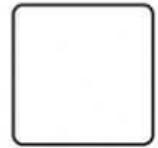
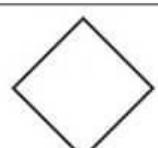
Kegunaan :

1. Mendesain program
2. Merepresentasikan program

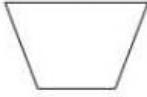
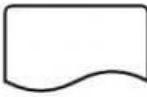
Flowchart harus dapat menggambarkan komponen-komponen dalam bahasa pemrograman

## NOTASI FLOWCHART

Berikut merupakan notasi yang digunakan dalam *flowchart* beserta dengan artinya.

	<b>Flow</b> Simbol yang digunakan untuk menggabungkan antara simbol yang satu dengan simbol yang lain. Simbol ini disebut juga dengan Connecting Line.
	<b>On-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang sama.
	<b>Off-Page Reference</b> Simbol untuk keluar - masuk atau penyambungan proses dalam lembar kerja yang berbeda.
	<b>Terminator</b> Simbol yang menyatakan awal atau akhir suatu program.
	<b>Process</b> Simbol yang menyatakan suatu proses yang dilakukan komputer.
	<b>Decision</b> Simbol yang menunjukkan kondisi tertentu yang akan menghasilkan dua kemungkinan jawaban, yaitu ya dan tidak.

Gambar 1.2 : Notasi dalam *flowchart* beserta dengan artinya 1

	<b>Input/output</b> Simbol yang menyatakan proses input atau output tanpa tergantung peralatan.
	<b>Manual Operation</b> Simbol yang menyatakan suatu proses yang tidak dilakukan oleh komputer.
	<b>Document</b> Simbol yang menyatakan bahwa input berasal dari dokumen dalam bentuk fisik, atau output yang perlu dicetak.
	<b>Predefine Proses</b> Simbol untuk pelaksanaan suatu bagian (sub-program) atau prosedure.
	<b>Display</b> Simbol yang menyatakan peralatan output yang digunakan.
	<b>Preparation</b> Simbol yang menyatakan penyediaan tempat penyimpanan suatu pengolahan untuk memberikan nilai awal.

Gambar 1.3 : Notasi *flowchart* beserta dengan artinya 2

Setiap notasi memiliki fungsi yang berbeda. Secara garis besar dapat dibedakan menjadi

1. Notasi yang berfungsi untuk menghubungkan satu notasi dengan notasi lainnya seperti notasi flow, on-page, dan of page reference,
2. Notasi yang berfungsi untuk menunjukkan proses yang sedang berjalan,
3. Notasi yang berfungsi untuk memasukan (INPUT) dan menampilkan hasil (OUTPUT).

## PSEUDOCODE

Pseudocode adalah notasi yang menyerupai Bahasa pemrograman Tingkat tinggi seperti Bahasa C dan Python. Struktur algoritmanya dapat dibagi ke dalam beberapa bagian yaitu :

1. Bagian kepala (Header)
2. Bagian Deklarasi (definisi variabel)
3. Bagian Deskripsi (ringkah langkah)

Contohnya :

```
Algoritma_Luas_segitiga
{Menghitung sebuah luas segitiga
apabila alas dan tinggi segitiga
tersebut diberikan}
```

Deklarasi

```
{Definisi nama peubah/variabel}
float alas, tinggi, luas
```

Deskripsi

```
READ (alas,tinggi)      #bisa juga : INPUT
luas <- 0.5 * alas * tinggi
WRITE (luas)           #bisa juga : OUTPUT
```

## NOTASI PSEUDOCODE

Contoh – contoh 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 perulangan yang memiliki kondisi akhir.
6. **IF-THEN-ELSE** : digunakan untuk mengambil sebuah Keputusan dari beberapa kondisi.

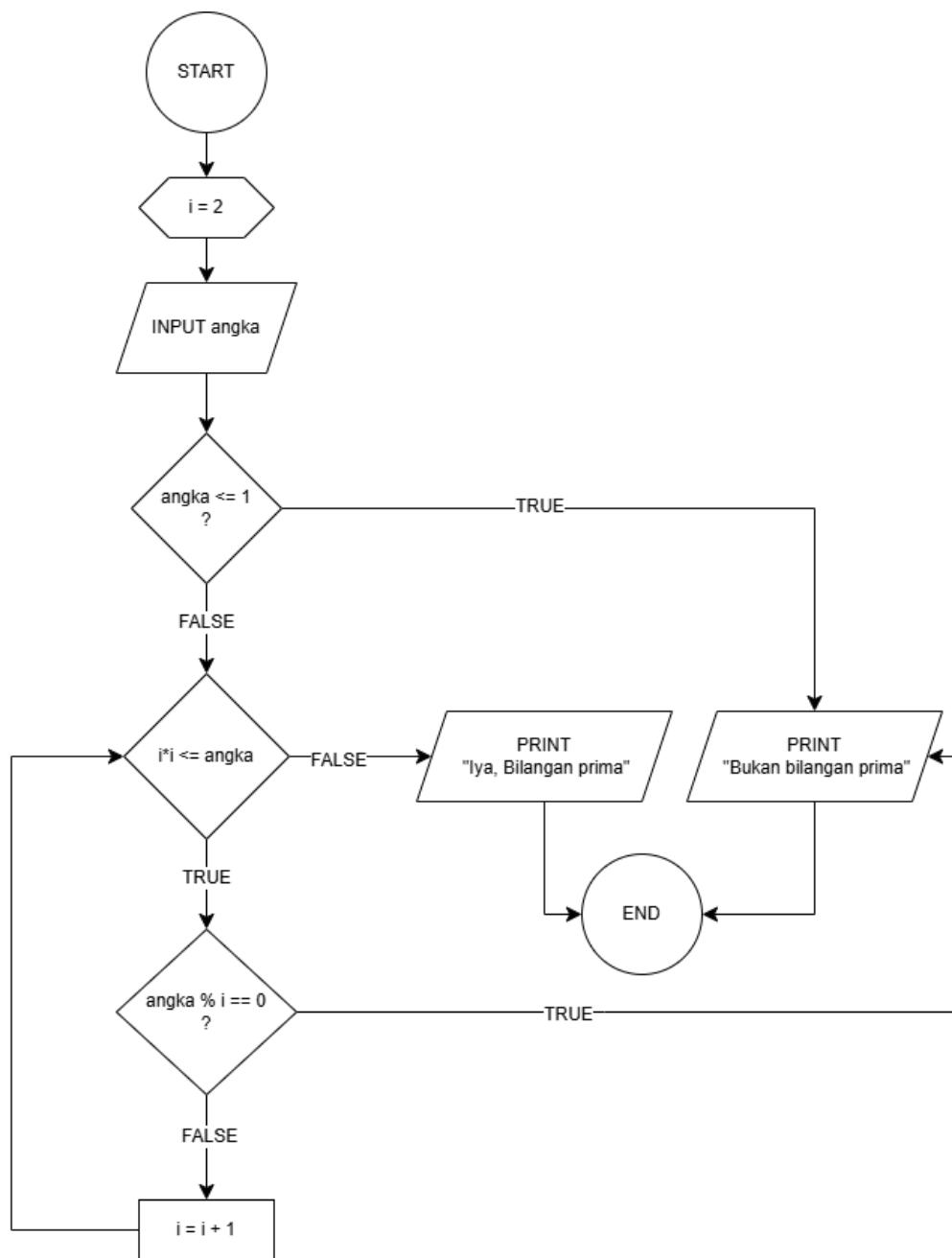
Sumber : Modul 03–Flowchart dan Pseudocode ([Module 03-Flowchart dan Pseudocode](#))

## 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/alc357/71251189\\_alicia.git](https://github.com/alc357/71251189_alicia.git)

### SOAL 1



Gambar 2.1 : Flowchart menentukan bilangan prima

### **Pseudocode**

```
Algoritma_Menentukan_Bilangan_Prima
Deklarasi
    Angka : integer
    i : integer
Deskripsi
    START
    INPUT angka
    i ← 2
    IF angka <= 1 THEN
        OUTPUT "Bukan bilangan prima"
    ELSE
        WHILE i*i <= angka DO
            IF angka mod i = 0 THEN
                OUTPUT "Bukan bilangan prima"
            END
            ENDIF
            i ← i + 1
        ENDWHILE
        OUTPUT "Iya, Bilangan prima"
    ENDIF
END
```

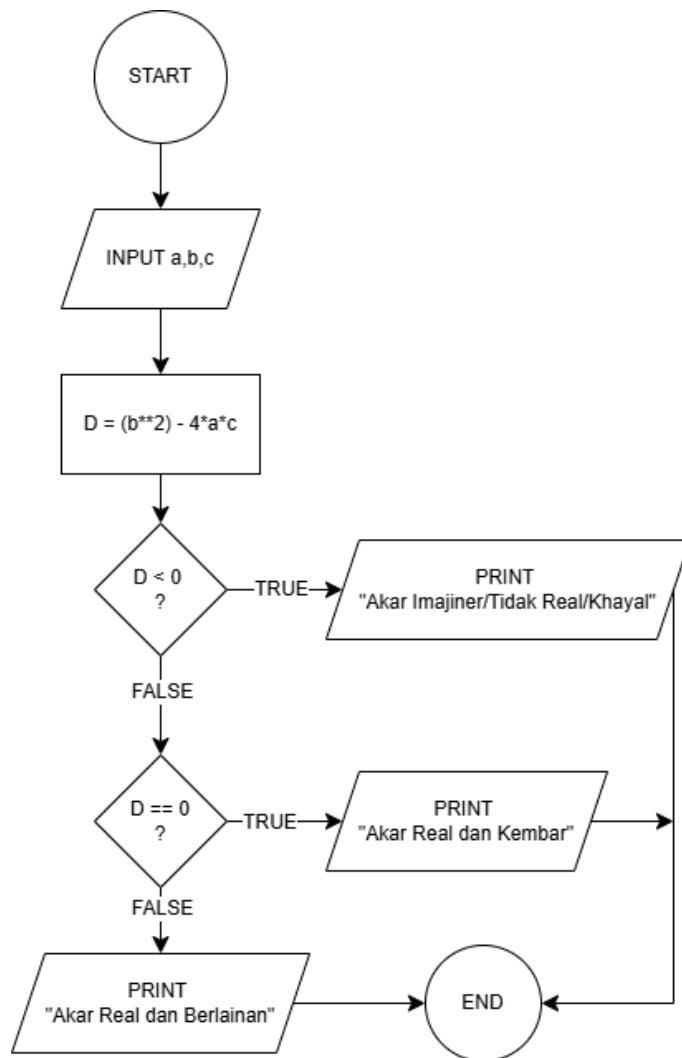
### **Penjelasan :**

1. Pseudocode ini berfungsi menentukan suatu angka merupakan bilangan prima atau bukan sehingga bernama Algoritma\_Menentukan\_Bilangan\_Prima
2. Kemudian, kita deklarasikan jenis data tiap variabel
3. Start berfungsi untuk memulai program
4. Kita inputkan angka yang nantinya akan di cek dan juga inisialisasi variabel i dengan nilai 2. Variabel i digunakan sebagai pembagi awal. Mengapa 2 ? karena bilangan prima tidak perlu dicek pembagi 1
5. Kemudian angka akan dicek apakah  $\leq 1$ ? Jika ya, maka bilangan tersebut bukan bilangan prima karena bilangan prima lebih besar dari 1. Jika tidak proses dilanjutkan.
6. Kita akan melakukan perulangan dengan kondisi  $i^2 \leq \text{angka}$ . Perulangan dilakukan untuk mengecek apakah terdapat pembagi selain 1 dan bilangan itu sendiri. Caranya akan di cek

apakah angka mod i = 0. Jika ya berarti angka dapat dibagi oleh bilangan lain selain 1 dan bilangan itu sendiri sehingga bukan bilangan prima. Namun jika tidak, maka nilai i akan ditambah 1 untuk melanjutkan ke pembagi berikutnya

7. Jika perulangan selesai tanpa menemukan pembagi lain, maka bilangan adalah bilangan prima.
8. Algoritma diakhiri (END)

## SOAL 2



Gambar 2.2 : Flowchart menentukan Jenis Akar

### Algoritma\_Jenis\_Akar

Deklarasi

a, b, c, D : real

Deskripsi

    START

    INPUT a, b, c

    D  $\leftarrow$  (b $^{**}$ 2) - (4\*a\*c)

    IF D < 0 THEN

        OUTPUT "Akar Imaginer/Tidak Real/Khayal"

    ELSE

        IF D = 0 THEN

            OUTPUT "Akar Real dan Kembar"

        ELSE

            OUTPUT "Akar Real dan Berlainan"

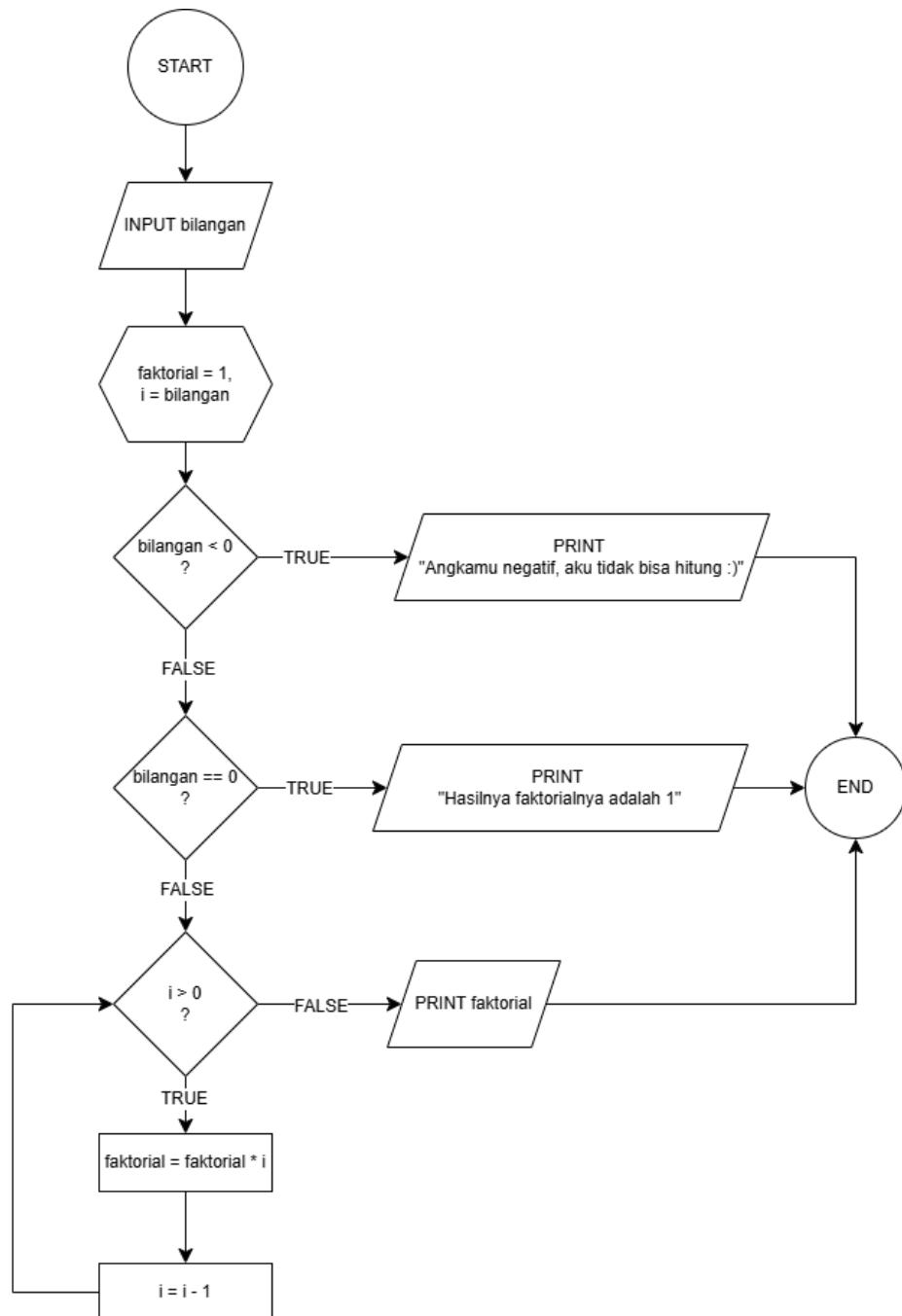
        ENDIF

    ENDIF

### **Penjelasan :**

1. Pseudocode ini berfungsi menentukan jenis akar sehingga diberi nama Algoritma\_Jenis\_Akar
2. Kemudian kita deklarasikan jenis data pada setiap variabel
3. Untuk memulai maka gunakan START
4. Untuk mencari diskriminan kita memerlukan INPUT nilai dari a, b, dan c pada persamaan kuadrat
5. Kemudian, kita masukan rumus diskriminan dalam variabel D.  $D = b^{**}2 - 4*a*c$
6. Dalam menentukan jenis akar maka perlu dibuat beberapa kondisi sesuai ketentuan yang juga sudah tertera di soal.
7. Kemudian akhiri program (END)

### SOAL 3



Gambar 2.3 : Flowchart menentukan nilai bilangan Faktorial

**Pseudocode:**

```
Algoritma_Nilai_Faktorial  
Deklarasi  
    Bilangan,i,faktorial : integer  
Deskripsi  
    INPUT bilangan  
    faktorial ← 1  
    i ← bilangan  
    IF bilangan < 0 THEN  
        OUTPUT "Angkamu negatif, aku tidak bisa hitung :)"  
    ELSE IF bilangan = 0 THEN  
        OUTPUT "Hasil faktorialnya adalah 1"  
    ELSE  
        WHILE i > 0 DO  
            faktorial ← faktorial * i  
            i ← i-1  
        ENDWHILE  
        OUTPUT faktorial  
    ENDIF  
END
```

**Langkah-langkah :**

1. Pseudocode ini berfungsi menentukan nilai faktorial suatu angka maka pseudocode ini bernama Algoritma\_Nilai\_Faktorial
2. Kemudian, start untuk memulai program
3. INPUT nilai yang ingin kita hitung faktorial dan inisialisasi variabel faktorial dengan nilai 1 karena faktorial adalah operasi perkalian. Dan i dengan nilai bilangan untuk operand dalam perulangan
4. Kemudian, kita cek apakah bilangan negatif dengan bilangan < 0 . Jika bilangan negatif maka bilangan tidak dapat terdefinisi
5. Jika lebih dari 0 maka kita cek, apakah bilangan adalah 0. Jika iya, maka nilainya adalah satu. Jika tidak kita lanjut ke proses berikutnya
6. Kemudian lakukan pengecekan perulangan selama i > 0 maka program akan mengalikan faktorial dengan i kemudian mengurangi i dengan 1 hingga kedaan menjadi false.

7. Ketika perulangan sudah selesai, program akan menampilkan variabel faktorial sebagai hasil perhitungan.
8. Program akan selesai (END).