



Muryan Awaludin, M.Kom.



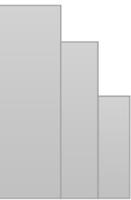
www.muryanawaludin.blogspot.com

www.ilmudesaingrafis.blogspot.com

www.brainflit.com

08562616116 (WA Only)

muryanawaludin1@gmail.com



Tujuan Pembelajaran

1. Memahami **Konsep Algoritma**
2. Memahami **Struktur Algoritma**
3. Memahami Algoritma menggunakan **bahasa natural Algoritma**
4. Memahami **Pseudocode**
5. Memahami **Flowchart dan penggunaan Tool Flowcahrt**
6. Pengenalan **Variabel**
7. Memahami Pengenalan **tipe data**
8. Memahami Pengenalan **operator**

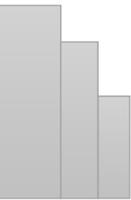
Algoritma Pemrograman

- **Belajar memprogram:** belajar tentang strategi pemecahan masalah, metodologi dan sistematika pemecahan masalah tersebut kemudian menuangkannya dalam suatu notasi yang disepakati bersama



“lebih bersifat pemahaman persoalan, analisis, sintesis”





Algoritma Pemrograman

- Belajar bahasa pemrograman: belajar memakai suatu bahasa, aturan sintaks (tatabahasa), setiap instruksi yang ada dan tata cara pengoperasian kompilator atau interpreter bahasa yang bersangkutan pada mesin tertentu

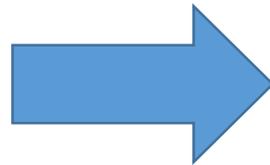
Jadi?

“BELAJAR **MEMPROGRAM**”

TIDAK SAMA DENGAN

“BELAJAR **BAHASA PEMROGRAMAN**”

Sejarah



PERANGKO dari **RUSIA**

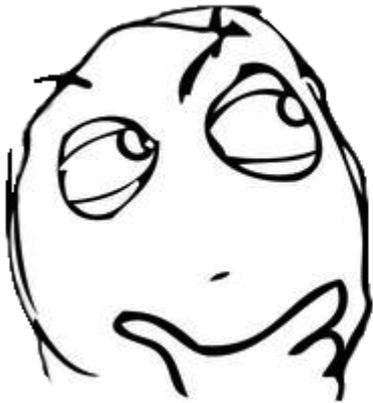


Muhammad bin Musa al-Khwarizmi

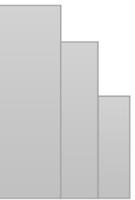
Seorang ilmuwan Islam yang karya karyanya dalam bidang matematika, astronomi, astrologi dan geografi banyak menjadi dasar perkembangan ilmu modern

Definisi Algoritma

- Apa sih **Algoritma** itu?



“Urutan **langkah-langkah logis**
penyelesaian masalah yang disusun
secara sistematis”



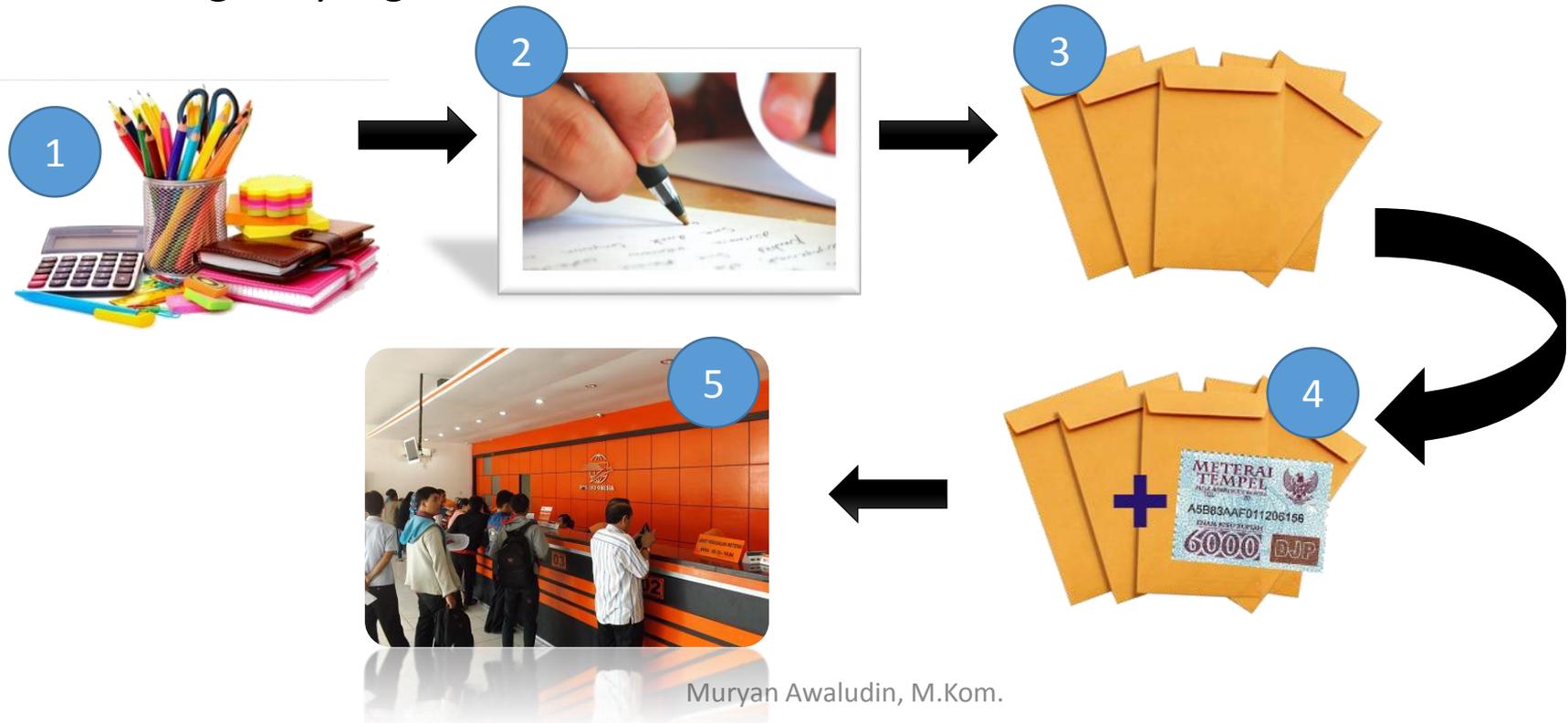
Definisi Algoritma

Algoritma yang dapat menyelesaikan suatu permasalahan dalam waktu yang singkat memiliki tingkat kerumitan yang rendah, sementara algoritma yang membutuhkan waktu lama untuk menyelesaikan suatu masalah membutuhkan tingkat kerumitan yang tinggi

Struktur Algoritma

Perhatikan algoritma sederhana berikut:

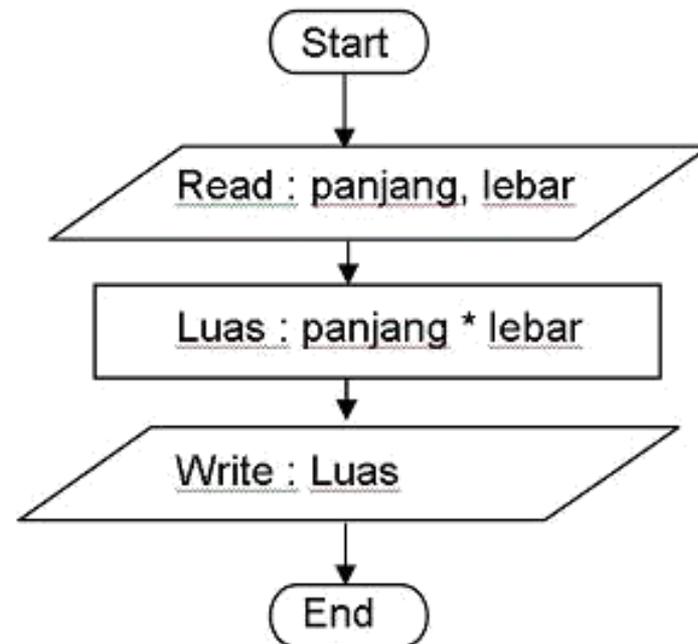
- Jika seseorang ingin mengirim surat kepada kenalnya di tempat lain, langkah yang harus dilakukan adalah:



Struktur Algoritma

Algoritma menghitung luas persegi panjang:

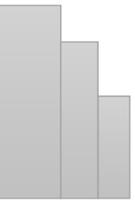
1. Masukkan panjang (P)
2. Masukkan lebar (L)
3. Luas $P * L$
4. Tulis Luas



Keuntungan Pembuatan Algoritma

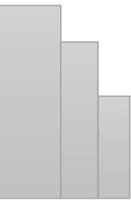
- a) Pembuatan atau **penulisan algoritma tidak tergantung pada bahasa pemrograman manapun**, artinya penulisan algoritma independen dari bahasa pemrograman dan komputer yang melaksanakannya
- b) **Notasi algoritma** dapat **diterjemahkan ke dalam berbagai bahasa pemrograman**
- c) Apapun bahasa pemrogramannya, **output yang akan dikeluarkan sama karena algoritamanya sama**





6 hal yang perlu diperhatikan dalam membuat algoritma

1. Teks algoritma berisi deskripsi **langkah-langkah** penyelesaian masalah.
2. **Tidak ada notasi yang baku** dalam penulisan teks algoritma seperti notasi bahasa pemrograman
3. Setiap orang dapat membuat aturan penulisan dan notasi algoritmik sendiri
4. **Notasi algoritmik bukan notasi bahasa pemrograman**, karena itu pseudocode dalam notasi algoritmik tidak dapat dijalankan oleh komputer
5. Algoritma digunakan untuk membantu kita dalam **mengkonversikan suatu permasalahan ke dalam bahasa pemrograman**
6. **Algoritma merupakan hasil pemikiran konseptual**, supaya dapat dilaksanakan oleh komputer, algoritma harus ditranslasikan ke dalam notasi bahasa pemrograman



Contoh:

Algoritma menghitung luas segitiga

1. Start
2. Baca data alas dan tinggi
3. Luas adalah alas kali tinggi kali 0.5
4. Tampilkan Luas
5. Stop

Sekilas algoritma di samping benar, **namun** apabila dicermati maka algoritma ini mengandung kesalahan yang mendasar, yaitu **tidak ada pembatasan pada nilai data untuk alas dan tinggi**

Hasil Perbaikan

1. Start
2. Baca data alas dan tinggi
3. Periksa data alas dan tinggi, **jika** nilai data alas dan tinggi lebih besar dari nol maka lanjutkan ke langkah ke 4 jika tidak maka stop
4. Luas adalah alas kali tinggi kali 0.5
5. Tampilkan Luas
6. Stop

Contoh:

Algoritma Berangkat Sekolah

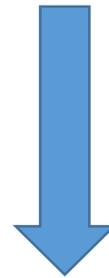
1. Mulai
2. Bangun dari tempat tidur
3. Mandi Pagi
4. Sarapan Pagi
5. Pergi Ke Sekolah
6. Cari Ruang Kelas
7. Masuk kelas untuk Belajar
8. Selesai



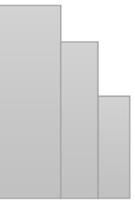
JUST
KIDDING

Beda Algoritma dan Program?

- **Program** → Kumpulan pernyataan komputer
- **Algoritma** → Metode dan tahapan sistematis dalam program
- Program ditulis dengan menggunakan bahasa pemrograman, **jadi** bisa disebut bahwa **program adalah suatu implementasi dari bahasa pemrograman**

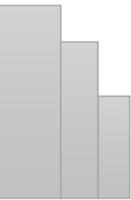


Program = **Algoritma** + Bahasa (Struktur Data)



Penerjemah Bahasa Pemrograman

- Untuk menterjemahkan bahasa pemrograman yang kita tulis maka diperlukan **Compiler** dan **Interpreter**
- **Compiler:** Suatu program yang **menterjemahkan bahasa program** (Source code) **ke dalam bahasa obyek (object code) secara keseluruhan program**
- **Interpreter:** **menganalisis dan mengeksekusi setiap baris dari program secara keseluruhan**. Keuntungan dari interpreter adalah dalam eksekusi yang bisa dilakukan dengan segera. Tanpa melalui tahap kompilasi, untuk alasan ini interpreter digunakan pada saat pembuatan program berskala besar



Perbedaan Compiler dan interpreter

<i>Compiler</i>	<i>Interpreter</i>
Menterjemahkan secara keseluruhan	Menterjemahkan Instruksi per instruksi
Bila terjadi kesalahan kompilasi maka source program harus diperbaiki dan dikompilasi ulang	Bila terjadi kesalahan interpretasi dapat Diperbaiki
Dihasilkan Object program	Tidak dihasilkan obyek program
Dihasilkan Executable program	Tidak dihasilkan Executable program
Proses pekerjaan program lebih cepat	Proses pekerjaan program lebih lambat
Source program tidak dipergunakan hanya bila untuk perbaikan saja	Source program terus dipergunakan
Keamanan dari program lebih terjamin	Keamanan dari program kurang terjamin

Jenis-Jenis Bahasa Pemrograman

- Bahasa Pemrograman Tingkat rendah (Bahasa mesin, Biner)
- Bahasa Pemrograman Tingkat tinggi

Contoh-contoh Bahasa Pemrograman yang ada:

1. Prosedural : Algol, Pascal, Fortran, Basic, Cobol, C
 2. Fungsional : LOGO, APL, LISP
 3. Deklaratif : Prolog
- Object oriented murni : Smalltalk, Eifel, Java, PHP

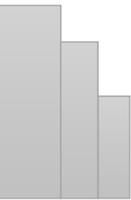


Cara Penulisan Algoritma

1. Structured English (SE)
2. Pseudocode
3. FlowChart

Algoritma





Cara Penulisan Algoritma

1. Structured English (SE)

SE merupakan alat yang cukup baik **untuk menggambarkan suatu algoritma**. Dasar dari SE adalah Bahasa Inggris, namun kita dapat memodifikasi dengan Bahasa Indonesia sehingga kita boleh menyebutnya sebagai Structured Indonesian (SI).

Structured English

PROGRAM OddOrEven:

Read in a number.

Divide it by two.

If there is a remainder, the
number is odd, otherwise
it's even.

END.

“SE atau SI lebih tepat untuk menggambarkan suatu algoritma yang akan dikomunikasikan kepada pemakai perangkat lunak”

Cara Penulisan Algoritma

2. Pseudocode

“Kode yang mirip dengan instruksi kode program sebenarnya”

Pseudocode didasarkan pada bahasa pemrograman yang sesungguhnya seperti BASIC, FORTRAN atau PASCAL

Pseudocode

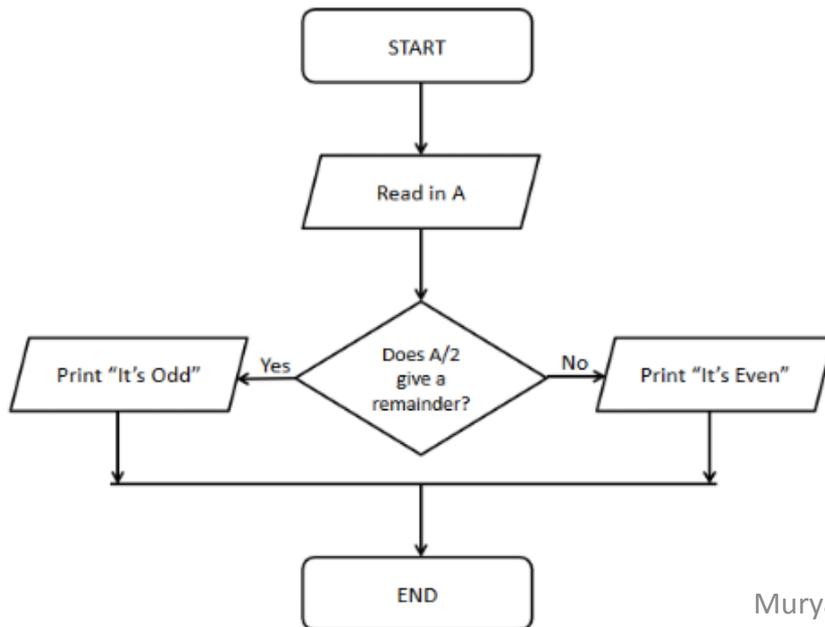
```
PROGRAM OddOrEven:  
  Read A;  
  IF A/2 gives a remainder  
    THEN Print "It's Odd";  
    ELSE Print "It's Even";  
  ENDIF;  
END.
```

“Pseudo berarti imitasi atau tiruan atau menyerupai, sedangkan code menunjuk pada kode program”

Cara Penulisan Algoritma

3. FlowChart

“Skema/bagan (chart) yang menunjukkan aliran (flow) di dalam suatu program secara logika”



Keterangan	Lambang
Mulai/selesai (terminator)	
Aliran data	
Input/Output	
Proses	
Percabangan (Decision)	
Pemberian nilai awal suatu variabel (Preparation)	
Memanggil prosedur/fungsi (Call)	
Connector (di halaman yg sama)	
Off page Connector (halaman lain)	

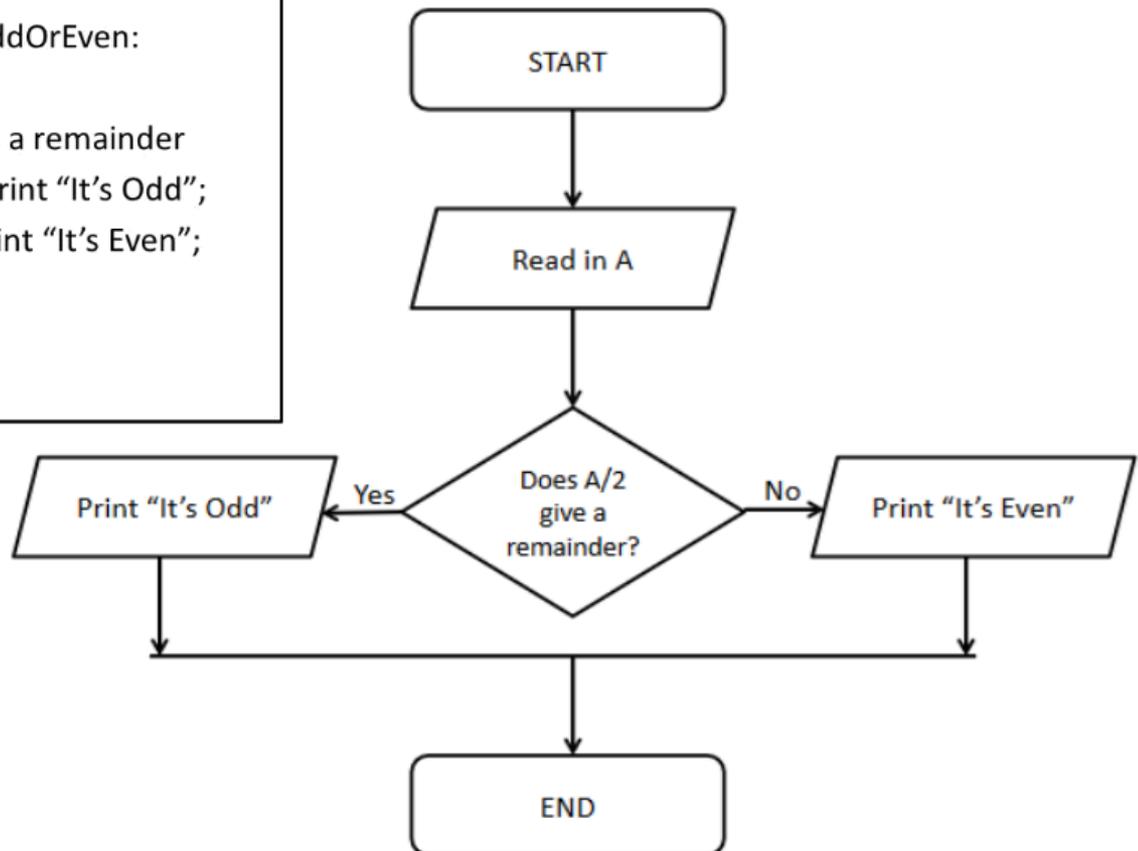
Cara Penulisan Algoritma

Structured English

```
PROGRAM OddOrEven:  
  Read in a number.  
  Divide it by two.  
  If there is a remainder, the  
  number is odd, otherwise  
  it's even.  
END.
```

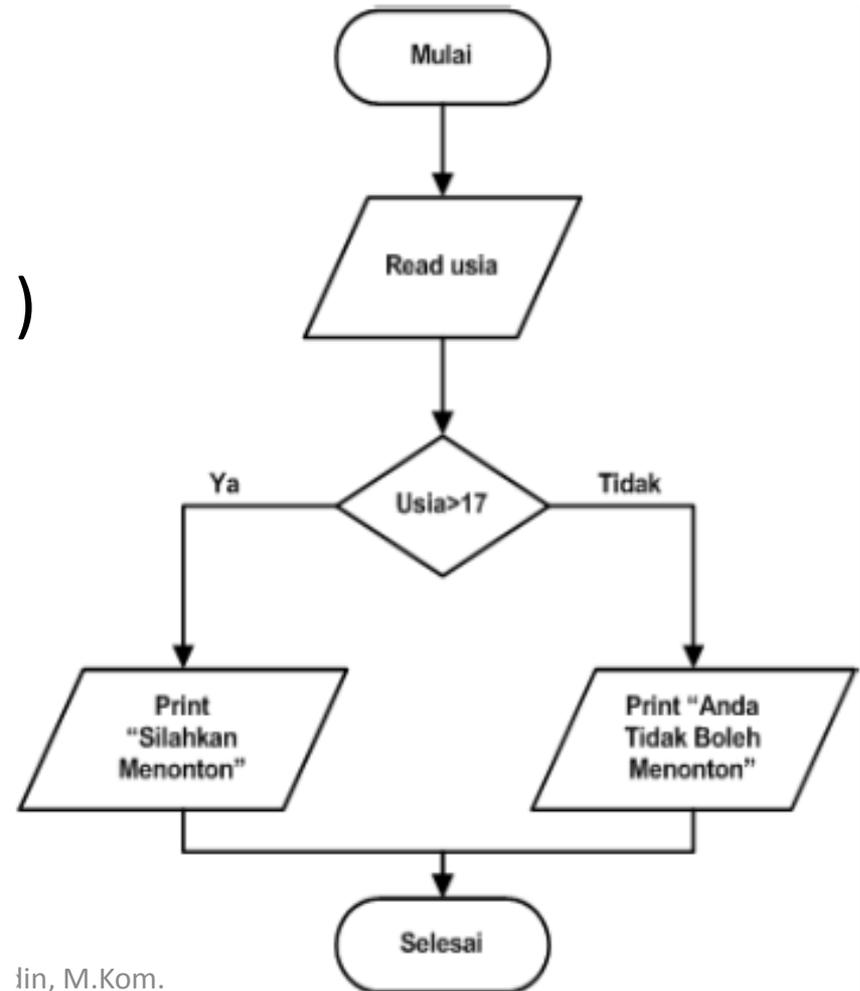
Pseudocode

```
PROGRAM OddOrEven:  
  Read A;  
  IF A/2 gives a remainder  
  THEN Print "It's Odd";  
  ELSE Print "It's Even";  
  ENDIF;  
END.
```



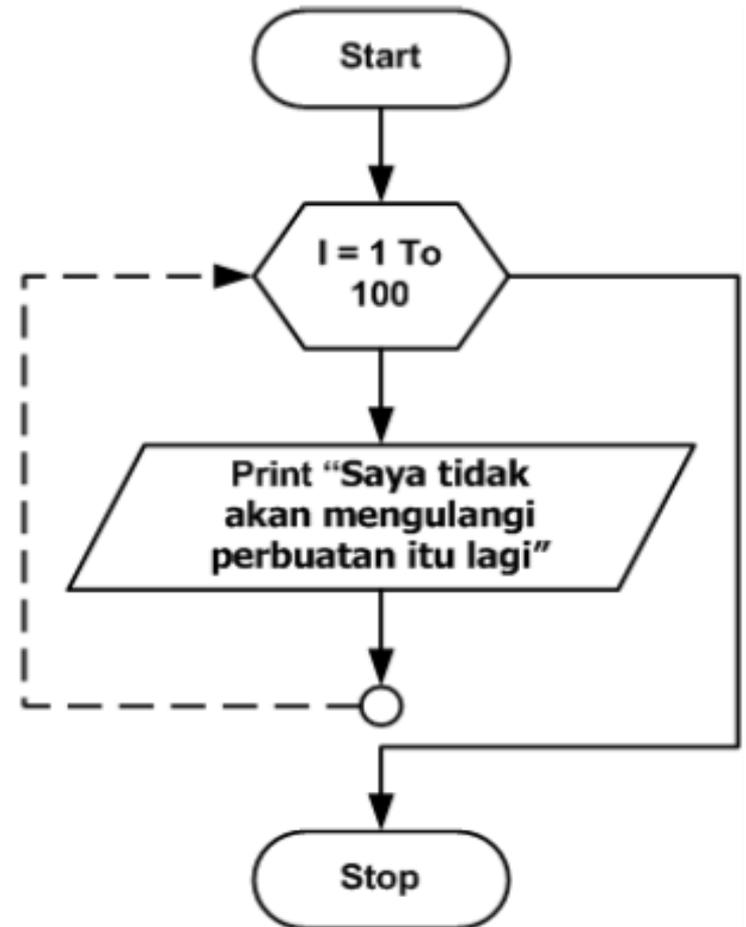
Contoh 1:

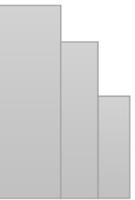
Bagan alir logika (Flowchart) penyelesaian masalah nonton film



Contoh 2:

Bagan alir logika (flowchart) untuk mencetak pernyataan sebanyak 100 kali





Lanjut ke pertemuan berikutnya

