

Algoritma & Pemrograman #1



Antonius Rachmat C, S.Kom, M.Cs

Algoritma



- ❑ Asal kata Algoritma (algorism - algorithm) berasal dari nama **Abu Ja'far Muhammad ibn Musa Al-Khuwarizmi**
- ❑ Ilmuan Persia yang menulis kitab "al jabar w'al-muqabala" (rules of restoration and reduction – aturan pemugaran dan pengurangan)
- ❑ Tahun 825 M
- ❑ Berasal dari Iran
- ❑ Algoritma masuk Indonesia tahun 1980-an

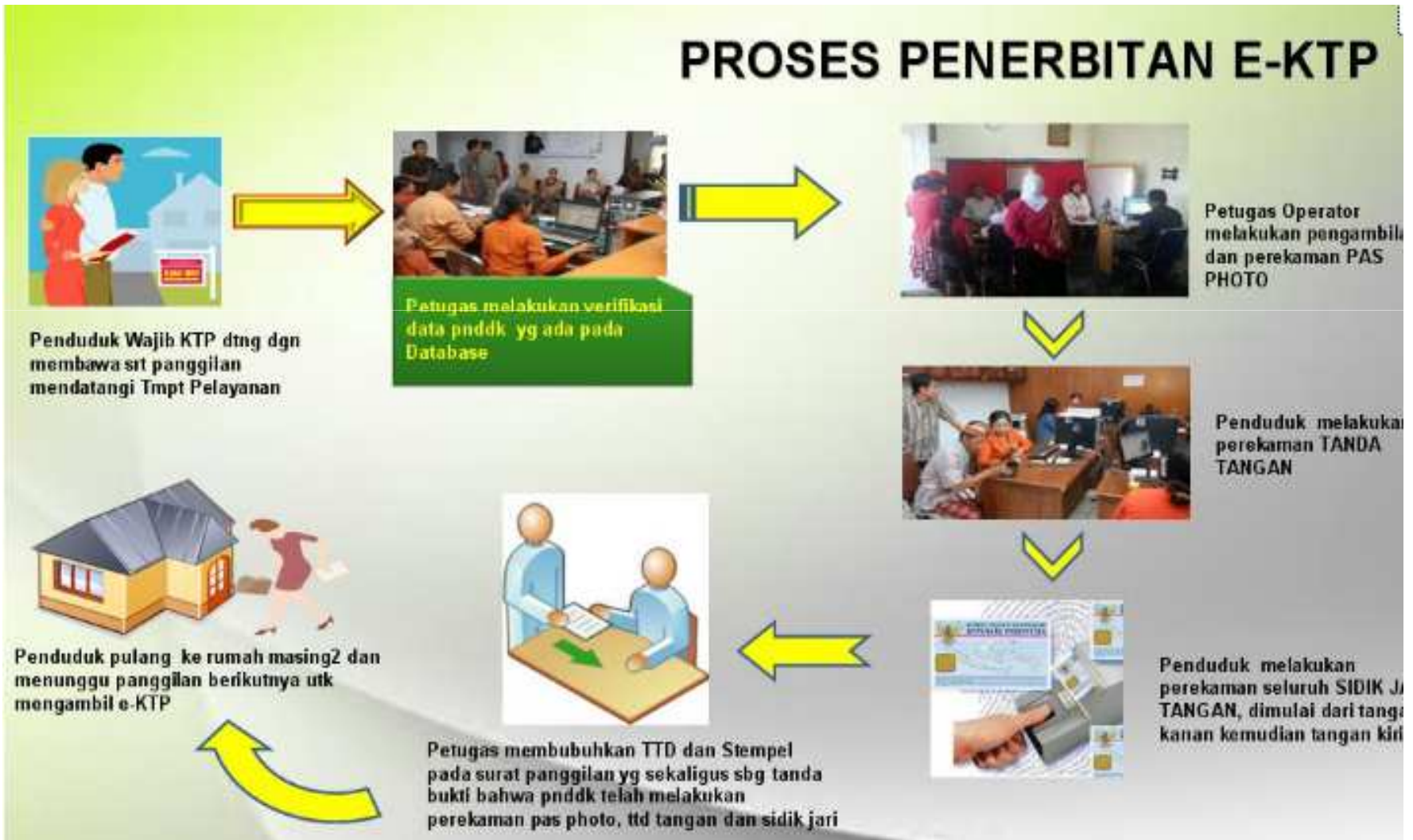
Cara memasak tempe goreng??



- Potong tempe
- Bumbui tempe
- Siapkan minyak
- Goreng tempe
- Hidangkan

Penerbitan e-KTP

PROSES PENERBITAN E-KTP



Pembelajaran Gambar

- ❑ Langkah-langkah harus benar, logis, dan urut
- ❑ Sama-sama melakukan pemecahan masalah
- ❑ Dapat dituliskan/digambarkan dalam langkah-langkah

Algoritma dalam bentuk tulisan

- Resep masakan
- Panduan registrasi
- Panduan pembukaan tabungan
- Panduan instalasi software
- Panduan pemasangan suatu perangkat

- **Semuanya menggunakan bahasa manusia**

Definisi Algoritma

- Algoritma adalah urutan langkah **logis** tertentu untuk memecahkan suatu masalah.
 - Urutan langkah logis, yang berarti algoritma harus mengikuti suatu urutan tertentu, tidak boleh melompat-lompat.

(Dari Microsoft Press Computer and Internet Dictionary 1997, 1998)

- Alur pemikiran dalam menyelesaikan suatu pekerjaan yang dituangkan secara **tertulis**.
 - Alur pikiran, sehingga algoritma seseorang dapat juga berbeda dari algoritma orang lain.
 - Tertulis, yang artinya dapat berupa kalimat, gambar, atau tabel tertentu.

(Dari Algoritma dan Struktur Data dengan C, C++, dan Java oleh Moh Sjukani hal 1)

Contoh Bu Tati



- Langkah Mengupas Kentang untuk Makan Malam
 - Ibu Tati mengambil kantong kentang dari rak
 - Ibu Tati mengambil panci dari almari
 - Ibu Tati mengupas kentang
 - Ibu Tati mengembalikan kantong kentang dari rak
- Ada aksi yang “tergantung” pada sesuatu:
 - Ibu Tati mengambil kantong kentang dari rak
 - Ibu Tati mengambil panci dari almari
 - Lakukan persiapan, gunakan celemek atau tidak **tergantung warna baju** (baju cerah/tidak)
 - Ibu Tati mengupas kentang
 - Ibu Tati mengembalikan kantong kentang dari rak



Lanjutan - Kondisi

- Misal suatu hari:
 - “Ibu Tati melihat bahwa bajunya tidak berwarna cerah karena itu ia **tidak memakai celemek**” (berarti tidak ada aksi memakai celemek)
- Misal hari lainnya:
 - “Ibu Tati melihat bahwa bajunya berwarna cerah karena itu ia **memakai celemek**”
- Jadi:
 - Ambil kantong kentang dari rak
 - Ambil panci dari almari
 - **Tergantung** warna baju
 - berwarna cerah : Pakai celemek
 - tidak berwarna cerah : Tidak pakai celemek
 - Kupas kentang
 - Kembalikan kantong kentang ke rak



Lanjutan - Perulangan

- Karena dapat pesanan, maka kentang yg harus dikupas **500 buah**
 - Tergantung jumlah kentang yang sudah dikupas
 - belum cukup : Kupas 1 kentang
 - cukup : selesai
- Atau:
 - while jumlah kentang terkupas belum cukup do
 - Kupas 1 kentang
- Bisa jadi pada saat mengupas kentang tergantung pada kentangnya, jika rusak / busuk, buang, **tidak dikupas**
- Jadi:
 - while jumlah kentang terkupas belum cukup do
 - Depend on kondisi kentang
 - Busuk : buang dan cari kentang berikutnya, tidak dihitung
 - Tidak Busuk: kupas 1 kentang

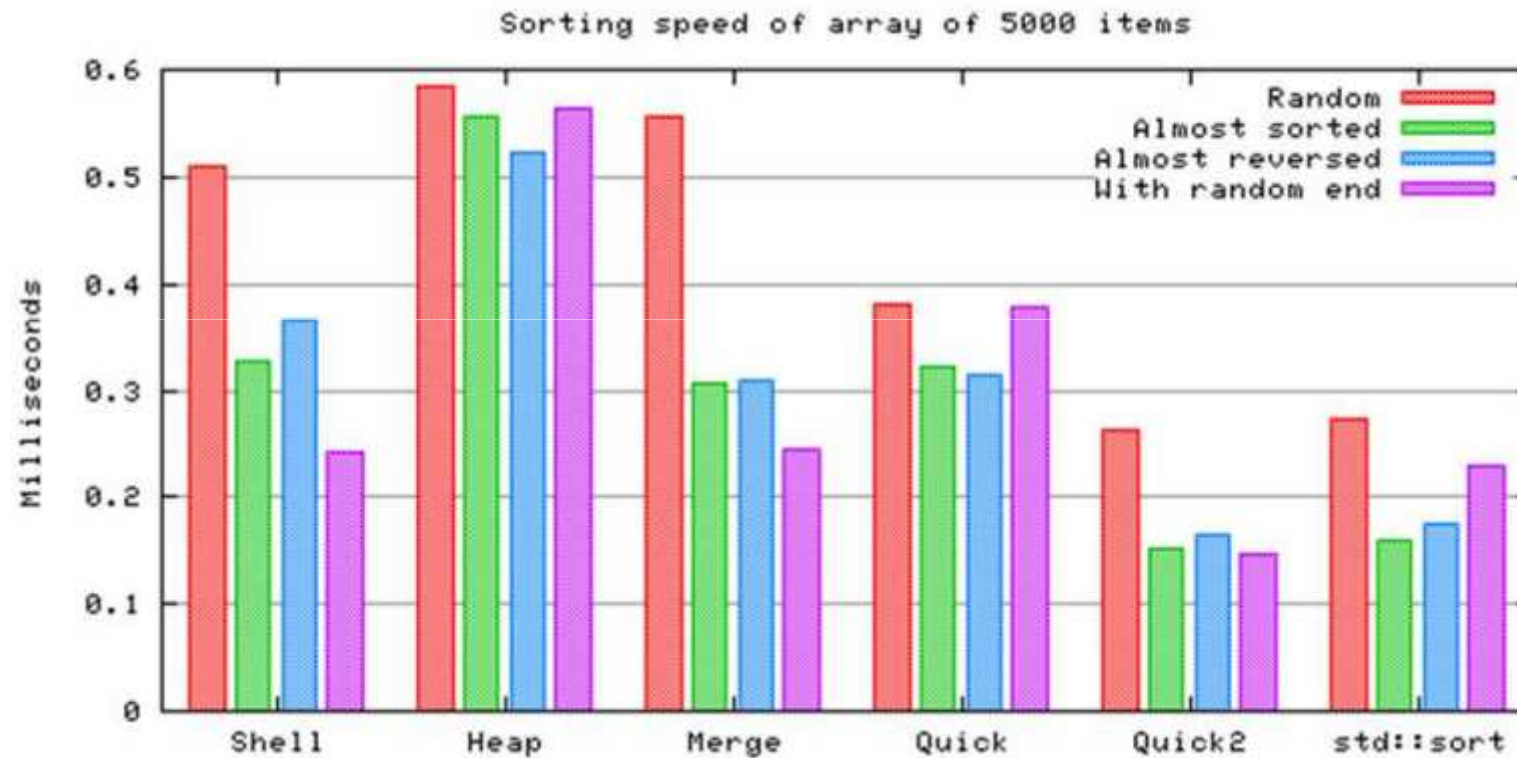
Kriteria Algoritma (Donald E. Knuth)

- ❑ **Input**: algoritma dapat memiliki nol atau lebih inputan dari luar.
- ❑ **Output**: algoritma harus memiliki minimal satu buah output keluaran.
- ❑ **Definite** (pasti): algoritma memiliki instruksi-instruksi yang jelas dan tidak ambigu.
- ❑ **Finite** (ada batas): algoritma harus memiliki titik berhenti (stopping role).
- ❑ **Effective** (tepat dan efisien): algoritma sebisa mungkin harus dapat dilaksanakan dan efektif. Contoh instruksi yang tidak efektif adalah: $A = A + 0$ atau $A = A * 1$

Namun ada beberapa program yang memang dirancang untuk **unterminatable**, contoh Sistem Operasi



Algoritma yang cepat (efisien)



Contoh Algoritma

- Algoritma menghitung luas persegi panjang:
 - Masukkan panjang (P)
 - Masukkan lebar (L)
 - $\text{Luas} \leftarrow P * L$
 - Tulis Luas

- Sifat: Berlaku umum
 - Tidak menggunakan simbol atau sintaks dari suatu bahasa pemrograman
 - Tidak tergantung pada suatu bahasa pemrograman
 - Notasi-notasinya dapat digunakan untuk seluruh bahasa manapun

Contoh algoritma 2

□ Kasus:

- Menukar dua buah bilangan
- $X = 10$ dan $Y = 2$, ditukar menjadi $X = 2$ dan $Y = 10$
- Bagaimana caranya?

□ Jawab 1:

- $X = X + Y$
- $Y = X - Y$
- $X = X - Y$

□ Jawab 2:

- tampung = X
- $X = Y$
- $Y = \text{tampung}$

Hitung $25 \times 34 = ?$

- Grid method
- Long multiplication method
- Karatsuba multiplication

$$25 \times 34 = ?$$

□ Grid method

$$25 \times 34 = 600 + 150 + 80 + 20 = \mathbf{850}$$

	20	5
30	600	150
4	80	20

$$25 \times 34 = ?$$

- Long multiplication method

$$\begin{array}{r} 25 \\ 34 \\ \hline \end{array} \times$$
$$\begin{array}{r} 100 \\ 75 \\ \hline \end{array} +$$
$$850$$

$$25 \times 34 = ?$$

$$\begin{array}{cc} 2 & 5 & \times & 3 & 4 \\ x1 & x2 & & y1 & y2 \end{array}$$

□ Karatsuba multiplication method

1. Hitung $A = x1 * y1$

$$A = 2 * 3 = 6$$

2. Hitung $B = x2 * y2$

$$B = 5 * 4 = 20$$

3. Hitung $C = (x1 + x2) (y1 + y2)$

$$C = (7) * (7) = 49$$

4. Hitung $K = C - A - B$

$$K = 49 - 6 - 20 = 23$$

5. Hitung $H = A * 100 + K * 10 + B$

6. Hasilnya adalah H

$$\begin{aligned} H &= 6 * 100 + 23 * 10 + 20 \\ &= 600 + 230 + 20 \\ &= 850 \end{aligned}$$

Breaking Question! Point Keaktifan

- Masih ingat 5 kriteria algoritma??

Pseudo Code

- ❑ Kode atau tanda yang menyerupai (pseudo) program atau merupakan penjelasan cara menyelesaikan suatu masalah.
- ❑ Pseudo-code sering digunakan oleh manusia untuk menuliskan algoritma
- ❑ Digunakan dengan bahasa manusia

Contoh

- Problem: mencari bilangan terbesar dari dua bilangan yang diinputkan

- Contoh Algoritma:
 - Masukkan bilangan pertama
 - Masukkan bilangan kedua
 - Jika bilangan pertama $>$ bilangan kedua maka kerjakan langkah 4, jika tidak, kerjakan langkah 5.
 - Tampilkan bilangan pertama
 - Tampilkan bilangan kedua

- Contoh Pseudo-code:
 - 1. Input a
 - 2. Input b
 - 3. If $a > b$ then kerjakan langkah 4
 - 4. print a
 - 5. print b

Algoritma vs Pseudocode

Algoritma	Pseudo-code
Nilai A lama ditambah dengan 5	$A \leftarrow A + 5$
Cetak nilai A bila lebih besar dari 10	IF A > 10 THEN PRINT A
Dari dua bilangan A dan B, cari bilangan yang terbesar	IF A > B THEN PRINT A ELSE PRINT B

Pseudocode mengikuti aturan berikut:

- ❑ Komputer dapat menerima informasi (**input**)
- ❑ Komputer dapat mengeluarkan informasi (**output**)
- ❑ Komputer dapat melakukan **operasi aritmetika (proses)**
- ❑ Komputer dapat **memberikan nilai** pada sebuah variabel atau memory (proses)
- ❑ Komputer dapat **membandingkan** dua variabel dan **memilih** satu dari dua tindakan alternatif (proses)
- ❑ Komputer dapat **mengulang** sekumpulan tindakan (proses)

Pseudocode Standar / Umum

- Menerima input: **READ, GET**
- Menampilkan output: **DISPLAY, PRINT, WRITE, SHOW**
- Aritmatika: **+, -, *, /, %, div, sub, mul, min**
- Pemberian nilai: **←**
- Inisialisasi: **INIT, SET**
- Memilih: **IF ... THEN ... ELSE ... , CASE ...**
- Perulangan: **FOR, WHILE**

Jelaskan output dari

```
SET count ← 1
WHILE (count < 6)
    PRINT count
    count ← count + 1
ENDWHILE
```

Output ?

Berapa kali operasi $\text{count} \leftarrow \text{count} + 1$ dilakukan ?

```
READ x
READ y
READ z
IF (x = y)
    PRINT x
ELSE
    IF (y < z)
        PRINT y
    ELSE
        PRINT z
    ENDIF
ENDIF
```

Apa outputnya jika
diinputkan
 $x = 5, y = 10$ dan $z = 15$?

Apa outputnya jika
diinputkan
 $x = 15, y = 20$ dan $z = 5$?

Flowchart

- **Definisi:**

- Bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial

- **Kegunaan:**

- Untuk mendesain program
- Untuk merepresentasikan program





- Maka, *flowchart* harus dapat merepresentasikan komponen-komponen dalam bahasa pemrograman

Pembuatan Flowchart

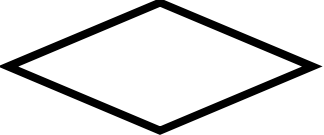


- Sebelum pembuatan program
 - Mempermudah *programmer* dalam menentukan alur logika program

- Sesudah pembuatan program
 - Menjelaskan alur program kepada orang lain

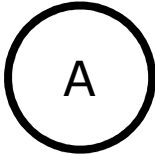




Lambang

Keterangan	Lambang
Mulai/selesai (<i>terminator</i>)	
Aliran data	
<i>Input/Output</i>	
Proses	

Lambang (2)

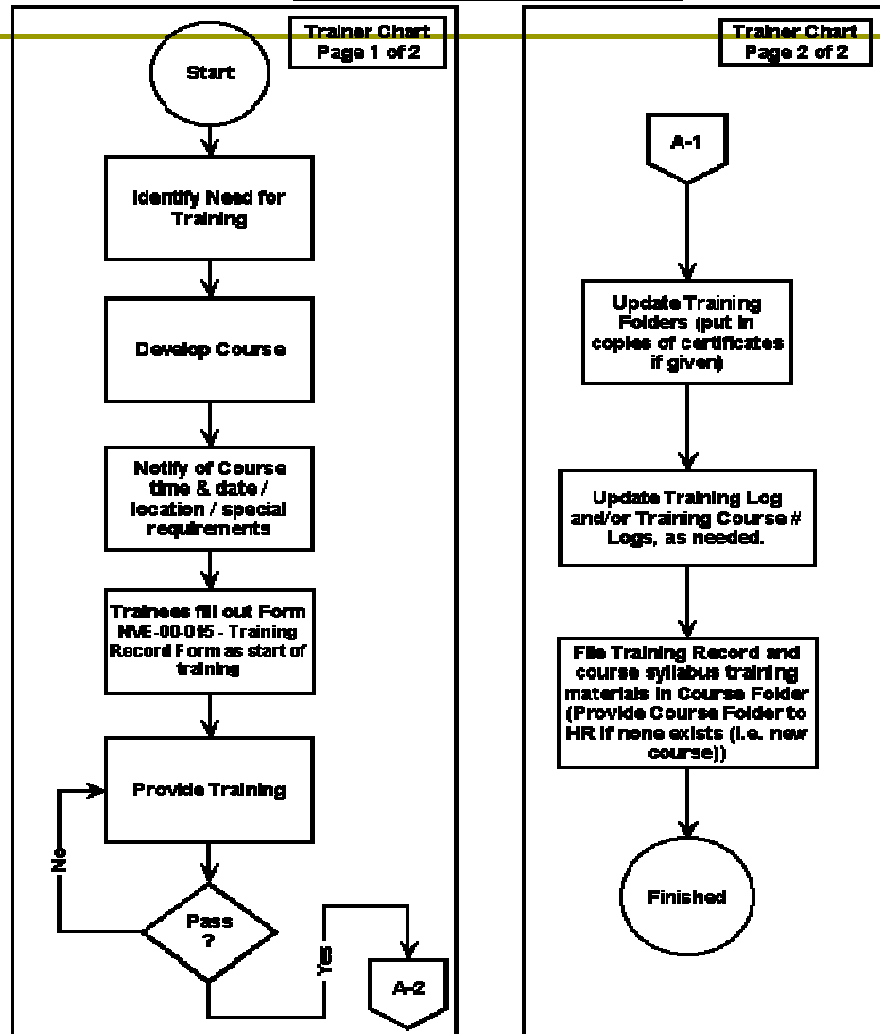
Keterangan	Lambang
Percabangan (<i>Decision</i>)	 A diamond-shaped symbol, representing a decision point in a flowchart.
Pemberian nilai awal suatu variabel (<i>Preparation</i>)	 A hexagonal symbol, representing the preparation of an initial value for a variable.
Memanggil prosedur/fungsi (<i>Call</i>)	 A rectangular symbol with vertical lines on the left and right sides, representing a call to a procedure or function.

Lambang (3)

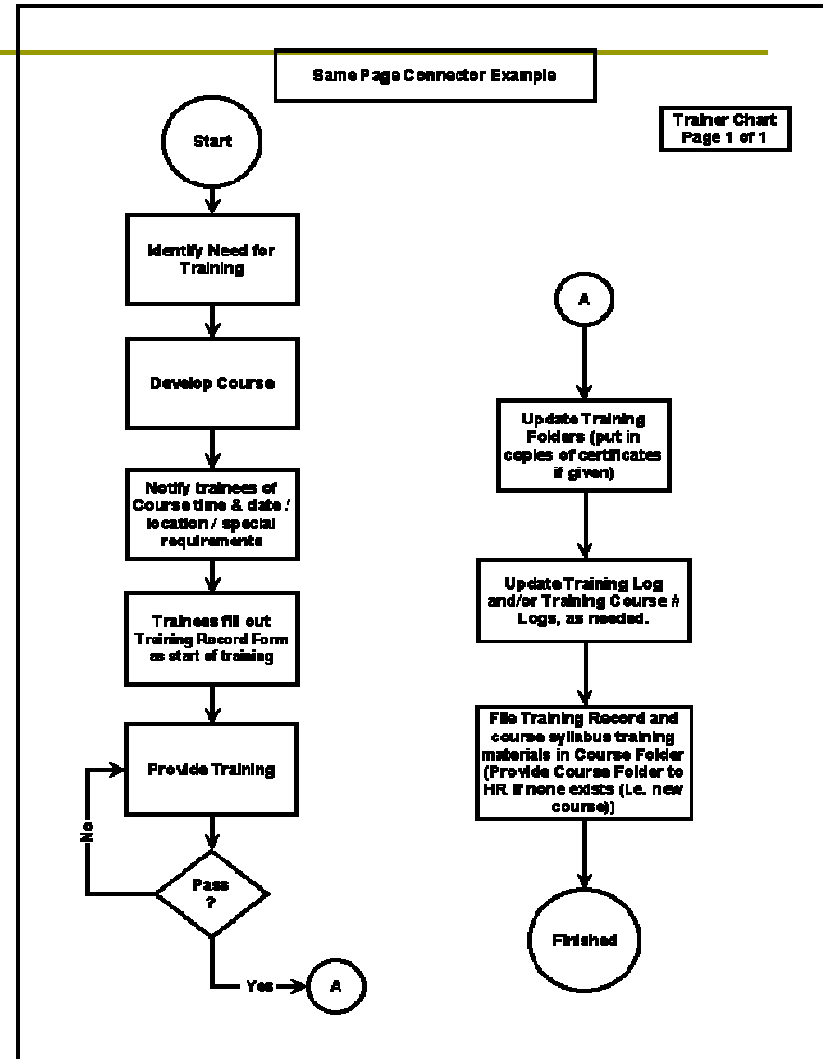
Keterangan	Lambang
<i>Connector (di halaman yg sama)</i>	
<i>Off page Connector (halaman lain)</i>	
Dokumen / Multi dokumen	 
Harddisk	

Page Connector

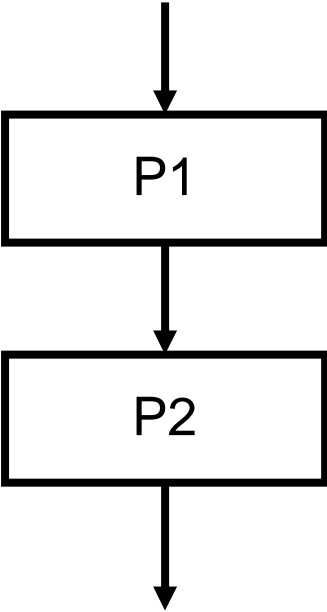
Off Page Connector Example



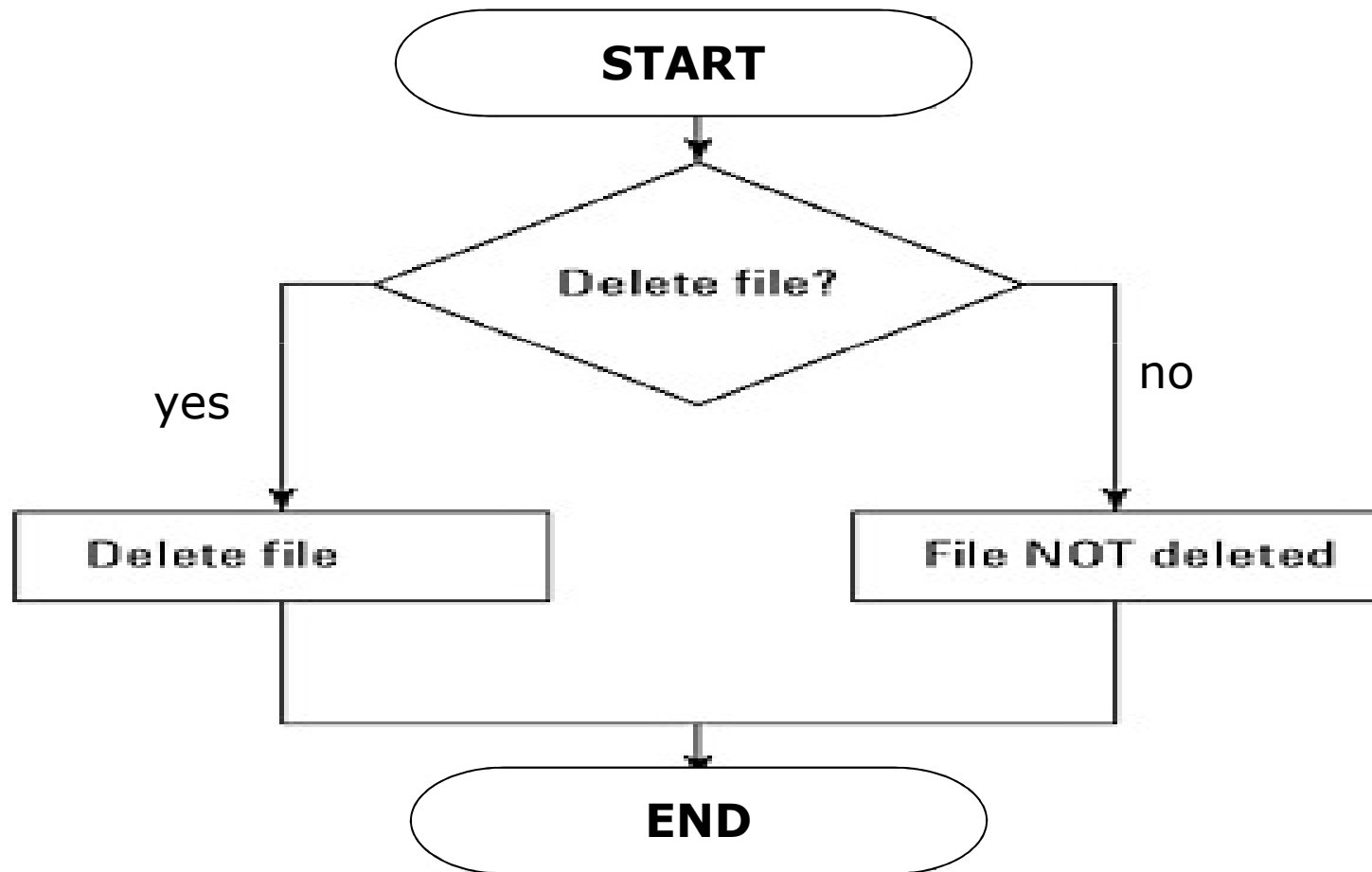
Same Page Connector Example



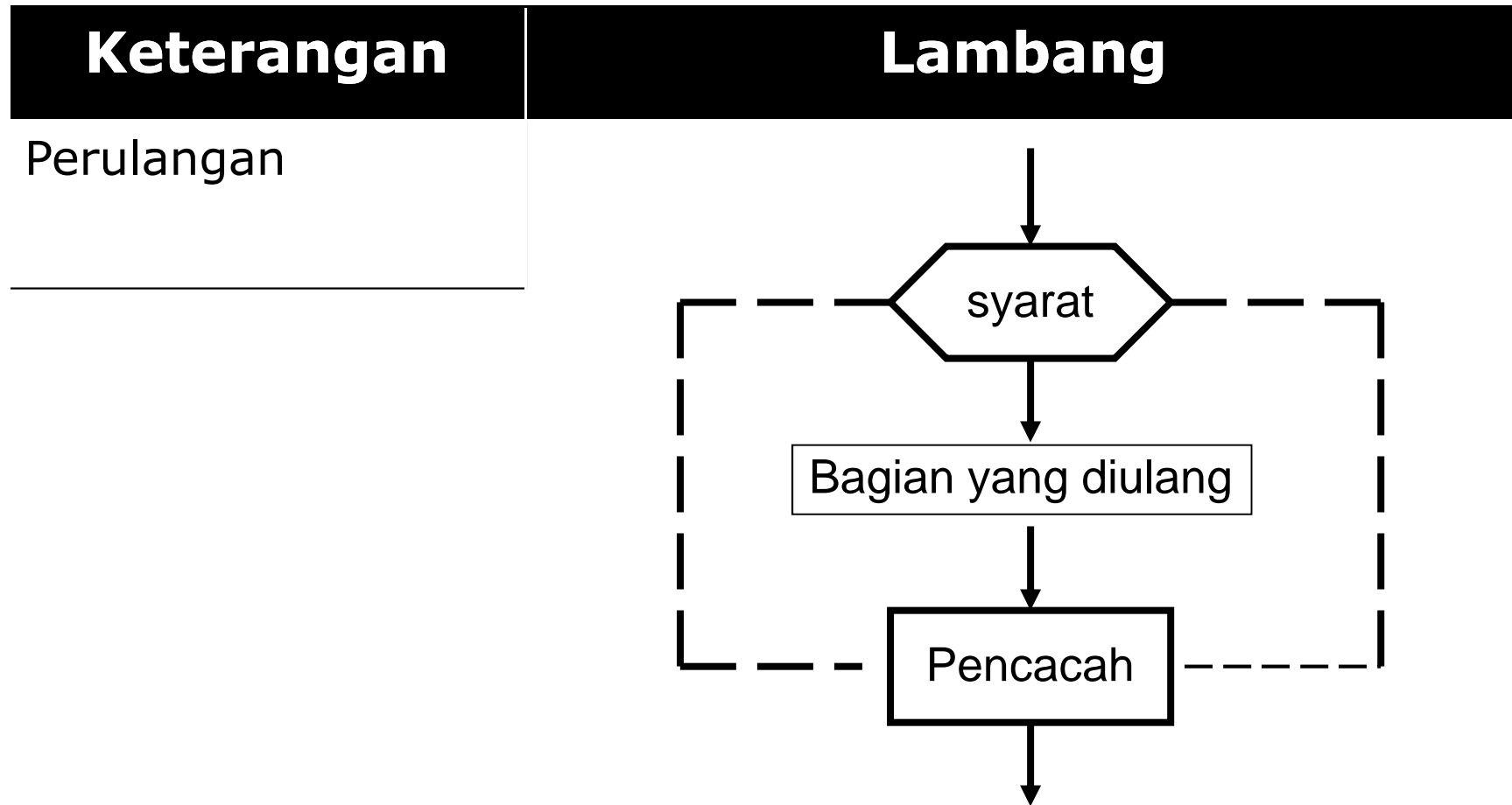
Lambang (4)

Keterangan	Lambang
<i>Sequence Process</i>	 <pre>graph TD; Start(()) --> P1[P1]; P1 --> P2[P2]; P2 --> End(())</pre>

Percabangan



Lambang (6)



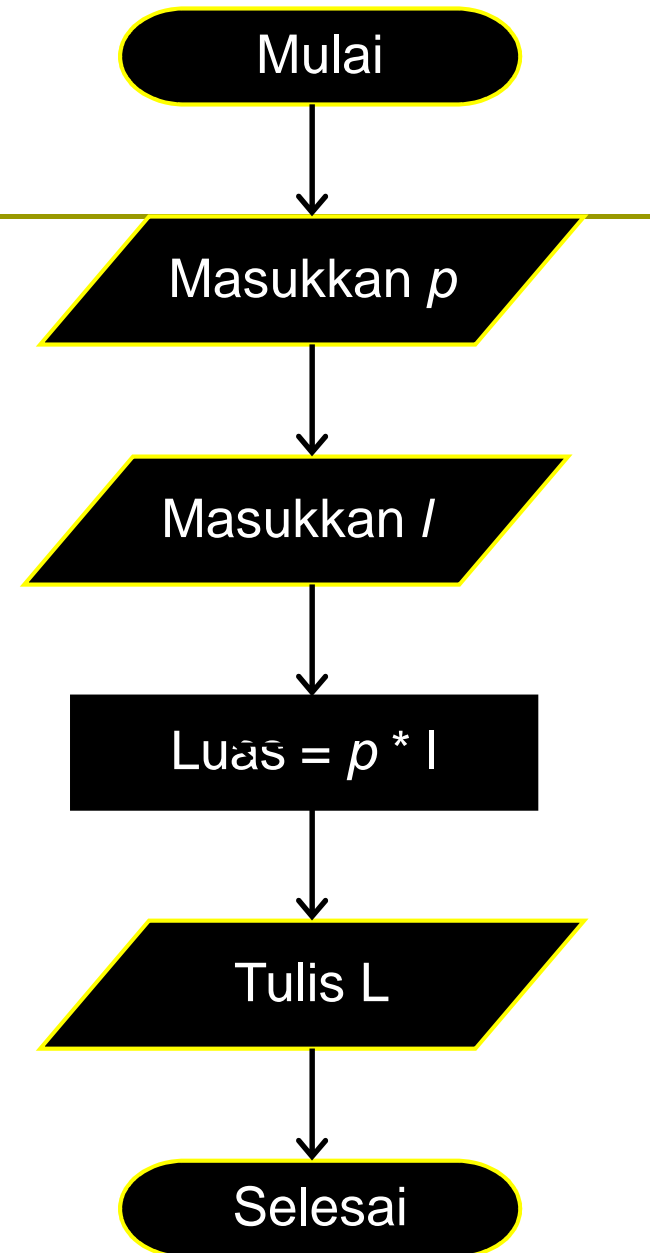
Contoh Flowchart

Problem:

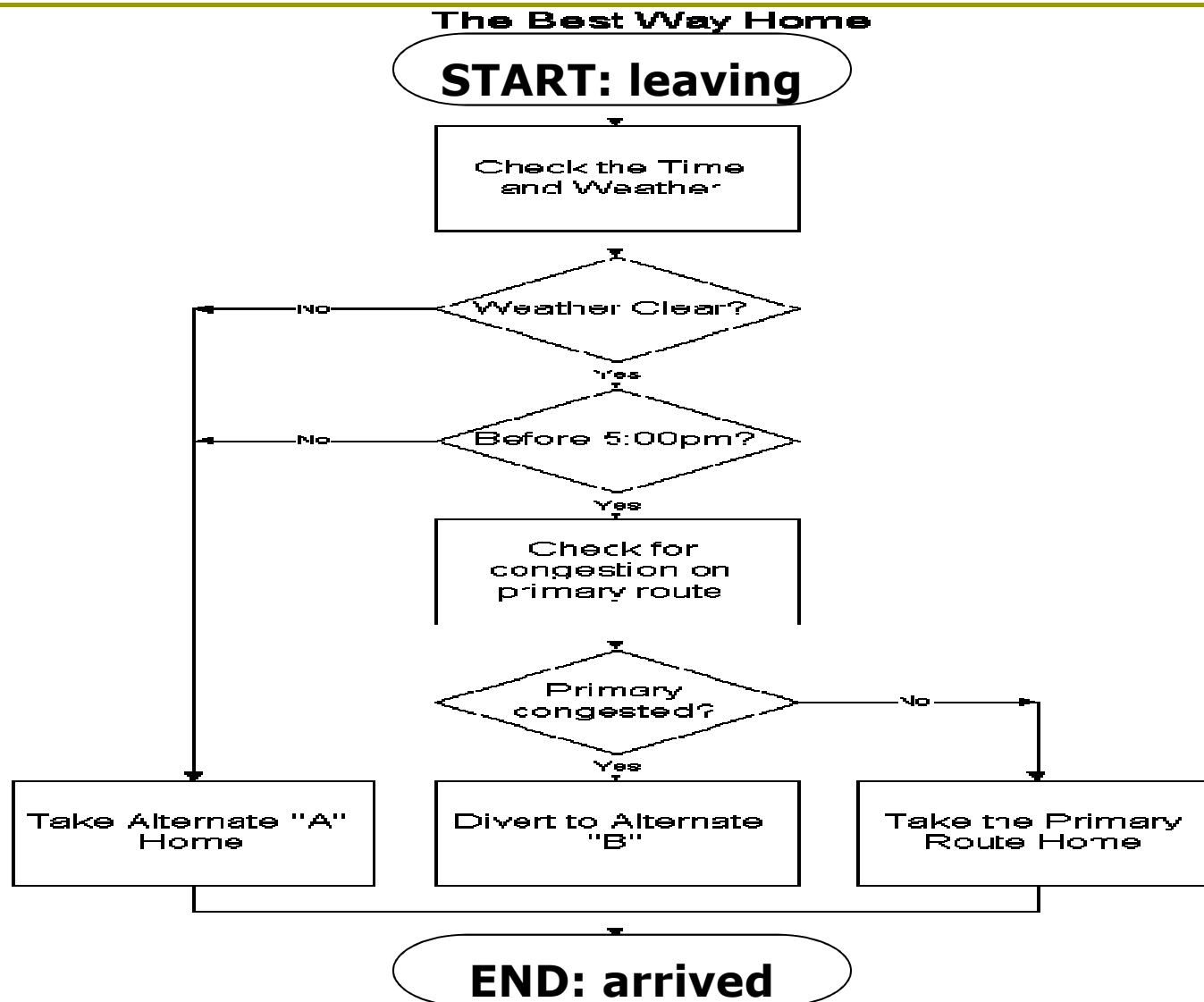
Menghitung
luas persegi panjang

Algoritma:

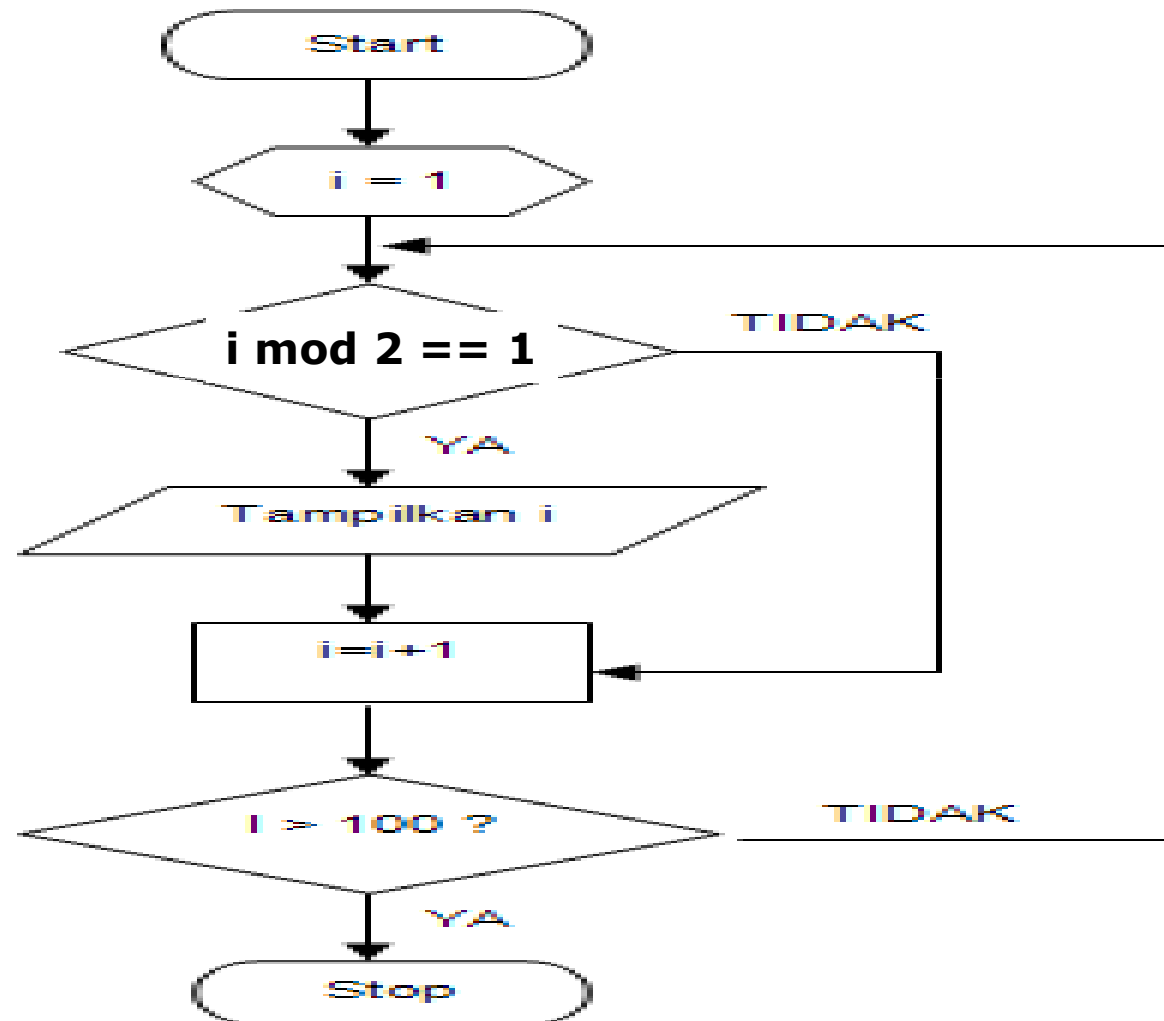
1. Masukkan panjang (p)
2. Masukkan lebar (l)
3. Hitung luas (L), yaitu panjang kali lebar
4. Cetak luas (L)



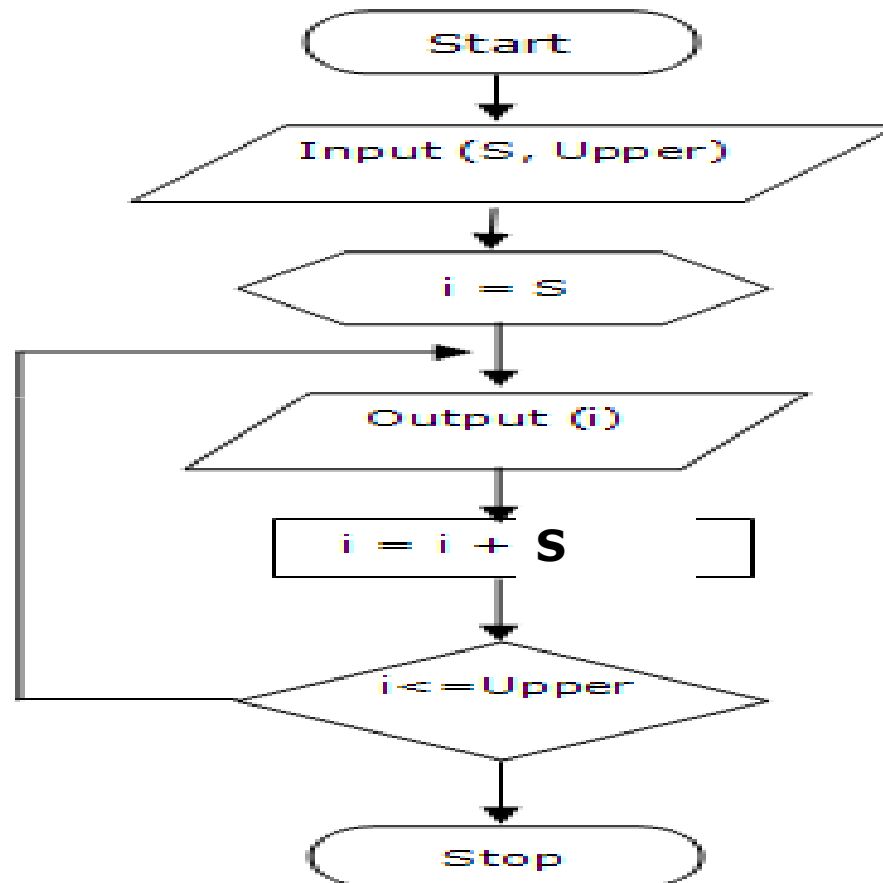
Flowchart : Mencari jalan pulang



Flowchart bilangan ganjil 1 - 100



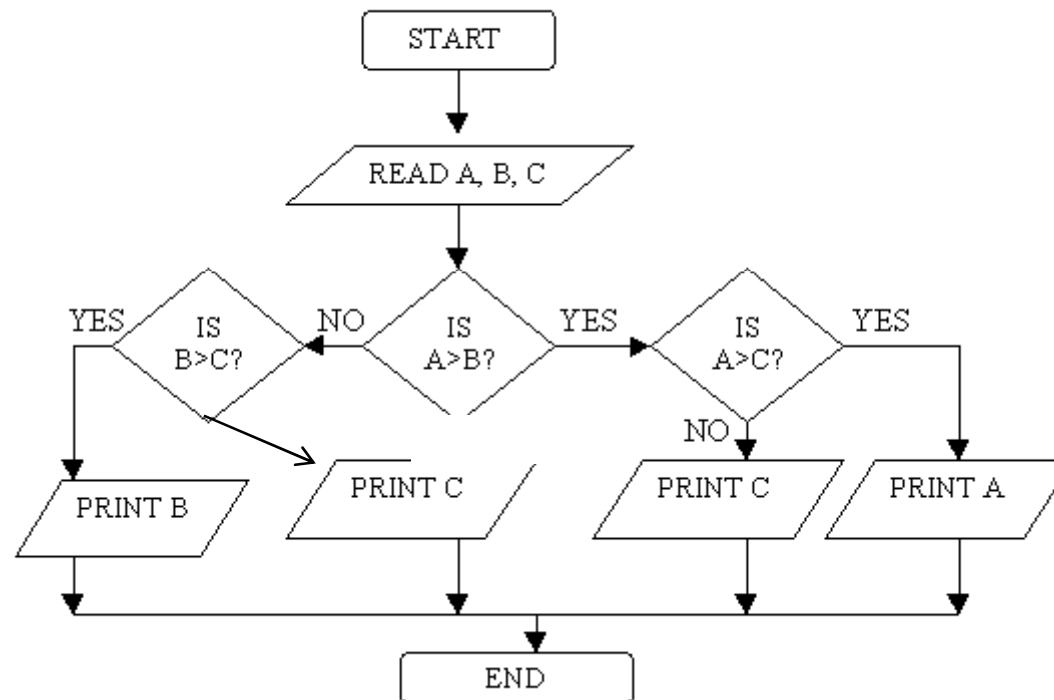
Flowchart Kelipatan Bilangan



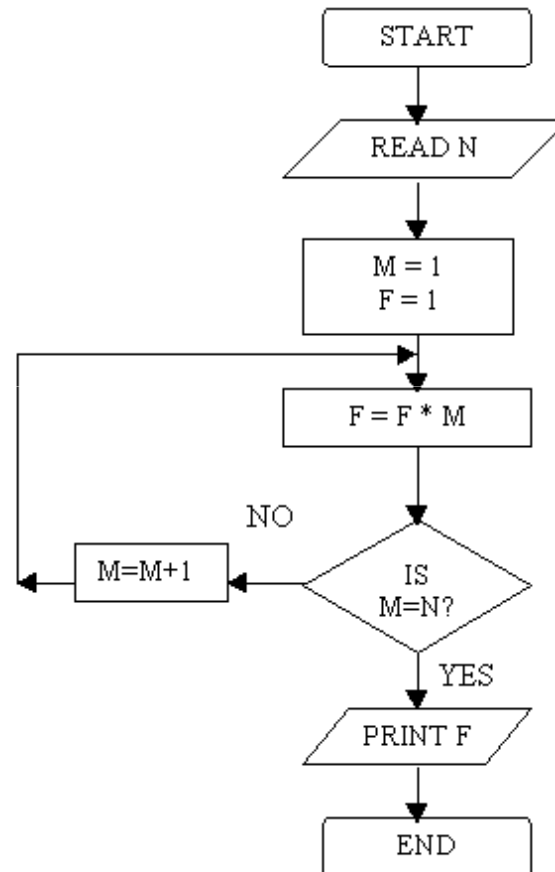
Keterangan :

S : kelipatan yang ingin ditampilkan
Upper : Batas range bilangan

a flowchart to find the largest of three numbers A, B, and C



Draw a flowchart for computing factorial N ($N!$)



Soal-soal

- Buatlah algoritma untuk menghitung konversi suhu.dari Celcius menjadi Reamur dan Farenheit.
 - Input: suhu dalam Celcius
 - Proses: $R = 4/5 * C$ dan $F = 9/5 * C + 32$
 - Output: suhu dalam Reamur dan Farenheit
- Buatlah algoritma untuk mencari sisi miring dari suatu segitiga siku-siku, jika diketahui panjang sisi yang membentuk sudut siku-siku.
 - Input: a dan b, yaitu panjang sisi pembentuk sudut siku-siku
 - Proses: $c = \sqrt{a^2 + b^2}$
 - Ouput: sisi miring (c)

Soal-soal

- Buatlah algoritma untuk menentukan suatu bilangan genap atau ganjil
 - Input: suatu bilangan
 - Ouput: genap / ganjil / nol
- Buatlah algoritma untuk untuk menghitung akar-akar persamaan kuadrat dengan rumus:
- $D = B^2 - 4 * A * C$
 - Jika $D < 0$ maka didapat akar imajiner
 - Jika $D = 0$ maka $X1 = X2$ yang didapat dari $D = -B / (2 * A)$
 - Jika $D > 0$ maka ada dua akar $X1 = -B + \sqrt{D/2 * A}$ dan $X2 = -B - \sqrt{D/2 * A}$

Soal-soal

- ❑ **Mencari 10 bilangan genap pertama dan jumlah totalnya**

- ❑ Menguji apakah suatu suhu (dalam Celcius) adalah beku, cair, gas
 - Input: suhu dlm celcius (bil bulat)
 - Proses: jika $< 0 =$ beku, $0-100 =$ cair, dan $> 100 =$ gas
 - Ouput: beku, cair, gas

Soal - soal

- **Mengetahui bilangan terbesar dari n buah bilangan yg diberikan user**
 - **Input: bilangan2 sebanyak n kali**
 - **Proses: simpan nilai masing2 bil yg diinputkan user, jika bil pertama, langsung catat bahwa bil itu maksimum, kemudian bandingkan dgn bil yg lainnya, jika ada yg lebih besar dari maksimum, jadikan bil itu maksimumnya**
 - **Output: bil maksimum**

Soal

- Buatlah flowchart untuk semua soal tadi!

NEXT

Translator Bahasa Pemrograman dan C