

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
- ❑ Masuk Indonesia tahun 1980-an

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 Algoritma

- ❑ Memasak.
- ❑ Jika seseorang ingin mengirim surat kepada kenalanannya di tempat lain, langkah yang harus dilakukan adalah:
 - Menulis surat
 - Surat dimasukkan ke dalam amplop tertutup
 - Amplop ditemplei perangko secukupnya.
 - Pergi ke Kantor Pos terdekat untuk mengirimkannya
- ❑ Dalam bidang komputer, algoritma sangat diperlukan dalam menyelesaikan berbagai masalah pemrograman, terutama dalam komputasi numeris.
- ❑ Tanpa algoritma yang dirancang baik maka proses pemrograman akan menjadi salah, rusak, atau lambat dan tidak efisien

Contoh Bu Tati (Lagi)

- ❑ 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 hal yang “tergantung” pada sesuatu:
 - Ibu Tati mengambil kantong kentang dari rak
 - Ibu Tati mengambil panci dari almari
 - Lakukan persiapan, gunakan celemek, **pakaian tergantung hari**
 - Ibu Tati mengupas kentang
 - Ibu Tati mengembalikan kantong kentang dari rak

Lanjutan - Kondisi

- Hari Sabtu :
 - “Ibu Tati melihat bahwa bajunya tidak berwarna muda karena itu ia tidak memakai celemek” (berarti tidak ada aksi memakai celemek)
- Hari Minggu:
 - “Ibu Tati melihat bahwa bajunya berwarna muda karena itu ia memakai celemek”
- Jadi:
 - Ambil kantong kentang dari rak
 - Ambil panci dari almari
 - Depend on warna baju
 - berwarna muda : Pakai celemek
 - tidak berwarna muda : Tidak pakai celemek
 - Kupas kentang
 - Kembalikan kantong kentang ke rak

Lanjutan - Perulangan

- Karena dapat pesanan, maka kentang yg harus dikupas 500 buah
 - depend on 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.
- ❑ *Definiteness* (pasti): algoritma memiliki instruksi-instruksi yang jelas dan tidak ambigu.
- ❑ *Finiteness* (ada batas): algoritma harus memiliki titik berhenti (stopping role).
- ❑ *Effectiveness* (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

Jenis Proses Algoritma

- ❑ Sequence Process: instruksi dikerjakan secara sekuensial, berurutan.
- ❑ Selection Process: instruksi dikerjakan jika memenuhi kriteria tertentu
- ❑ Iteration Process: instruksi dikerjakan selama memenuhi suatu kondisi tertentu.
- ❑ Concurrent Process: beberapa instruksi dikerjakan secara bersama.

Contoh Algoritma

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

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

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

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:
 - Input a
 - Input b
 - If $a > b$ then kerjakan langkah 4
 - print a
 - print b

Perbandingan

Algoritma	Pseudo-code
Nilai A 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

Bagian Program

- Suatu program pada dasarnya terdiri dari 3 bagian:
 - Input: bisa ada, bisa tidak
 - Proses
 - Output: minimal satu output.

Langkah Pembuatan Program

Mendefinisikan masalah

- ❑ Ini merupakan langkah pertama yang sering dilupakan orang.
- ❑ Menurut hukum Murphy (oleh Henry Ledgard):
 - “Semakin cepat menulis program, akan semakin lama kita dapat menyelesaikannya”.
- ❑ Hal tersebut berlaku untuk permasalahan yang kompleks. Tentukan masalahnya, apa saja yang harus dipecahkan dengan menggunakan komputer, dan apa inputan serta outputnya.

Langkah Pembuatan Program

Menemukan solusi

- Setelah masalah didefinisikan, maka langkah berikutnya adalah menentukan solusi. Jika masalah terlalu kompleks, maka ada baiknya masalah tersebut dipecah menjadi modul-modul kecil agar lebih mudah diselesaikan.
- Contohnya masalah invers matriks, maka kita dapat membagi menjadi beberapa modul:
 - meminta masukkan berupa matriks bujur sangkar
 - mencari invers matriks
 - menampilkan hasil kepada pengguna
- Dengan penggunaan modul tersebut program utama akan menjadi lebih singkat dan mudah dilihat.

Langkah Pembuatan Program

Memilih algoritma

- ▣ Pilihlah algoritma yang benar-benar sesuai dan efisien untuk permasalahan tersebut

Menulis program

- ▣ Pilihlah bahasa yang mudah dipelajari, mudah digunakan, dan lebih baik lagi jika sudah dikuasai, memiliki tingkat kompatibilitas tinggi dengan perangkat keras dan platform lainnya.

Langkah Pembuatan Program

Menguji program

- ▣ Setelah program jadi, silahkan uji program tersebut dengan segala macam kemungkinan yang ada, termasuk error-handlingnya sehingga program tersebut akan benar-benar handal dan layak digunakan.

Menulis dokumentasi

- ▣ Menulis dokumentasi sangat penting agar pada suatu saat jika kita akan melakukan perubahan atau membaca source code yang sudah kita tulis dapat kita ingat-ingat lagi dan kita akan mudah membacanya. Caranya adalah dengan menuliskan komentar-komentar kecil tentang apa maksud kode tersebut, untuk apa, variabel apa saja yang digunakan, untuk apa, dan parameter-parameter yang ada pada suatu prosedur dan fungsi.

Langkah Pembuatan Program

Merawat program

- ❑ Program yang sudah jadi perlu dirawat untuk mencegah munculnya bug yang sebelumnya tidak terdeteksi. Atau mungkin juga pengguna membutuhkan fasilitas baru yang dulu tidak ada

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
 - Output: genap / ganjil / nol
- ❑ Buatlah algoritma 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

- ❑ Menghitung usia berdasarkan tahun (saja) lahir dan tahun (saja) sekarang
 - Input : Tahun lahir (tl), Tahun sekarang (ts)
 - Proses : $\text{Umur} = ts - tl$
 - Output : Cetak Umur
- ❑ Menguji apakah suatu suhu (dalam Celcius) adalah beku, cair, gas
 - Input: suhu dlm celcius (bil bulat)
 - Proses: jika $< 0 = \text{beku}$, $0-100 = \text{cair}$, dan $> 100 = \text{gas}$
 - Output: 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

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

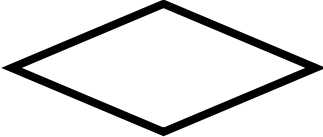


Flowchart

- ❑ Secara garis besar, unsur-unsur pemrograman adalah **Input → Proses → Output**. Semua bahasa pemrograman, pasti mempunyai komponen-komponen sebagai berikut :
 - Input (scanf)
 - Percabangan (if, switch)
 - Perulangan (while, for, for each, loop)
 - Output (printf)

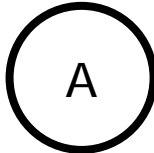

Lambang




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

Lambang (2)

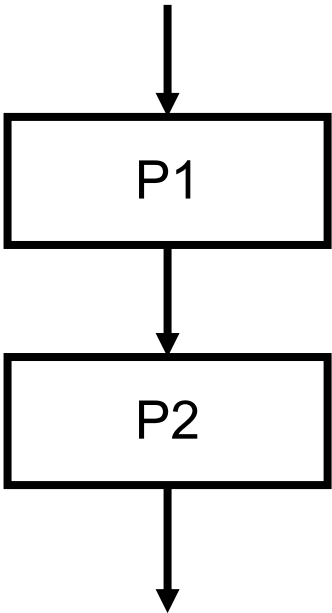
Keterangan	Lambang
Percabangan (<i>Decision</i>)	
Pemberian nilai awal suatu variabel (<i>Preparation</i>)	
Memanggil prosedur/fungsi (<i>Call</i>)	

Lambang (3)

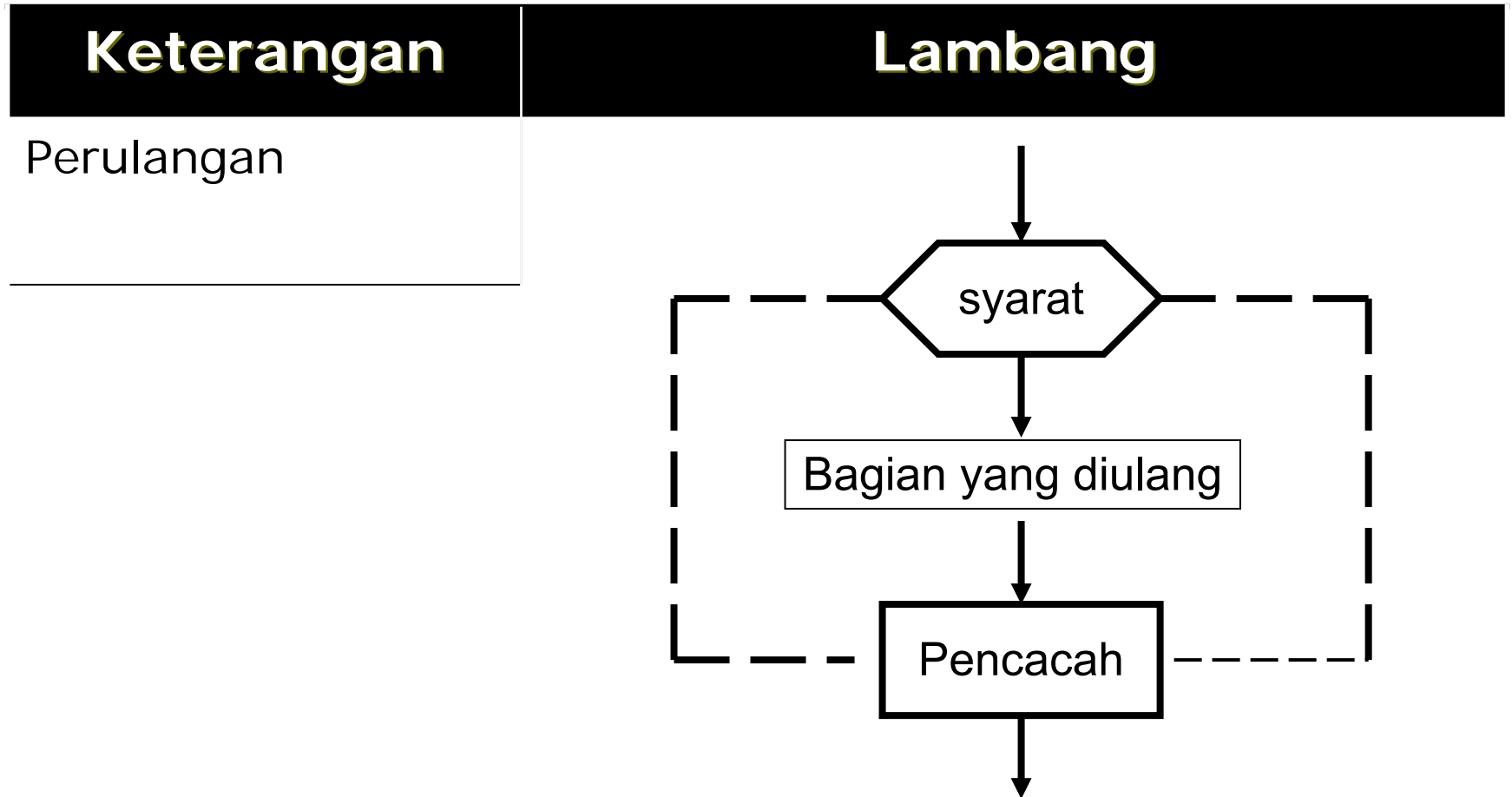
Keterangan	Lambang
<i>Connector (di halaman yg sama)</i>	
<i>Off page Connector (halaman lain)</i>	

Dokumen / Multi dokumen	 
Harddisk	

Lambang (4)

Keterangan	Lambang
<i>Sequence Process</i>	 <pre>graph TD; In(()) --> P1[P1]; P1 --> P2[P2]; P2 --> Out(())</pre> <p>The diagram shows a vertical sequence of two rectangular process boxes. The top box is labeled 'P1' and the bottom box is labeled 'P2'. An arrow points down into the top of the 'P1' box from above. Another arrow points down from the bottom of the 'P2' box to below it. This represents a sequence of two process steps.</p>

Lambang (6)



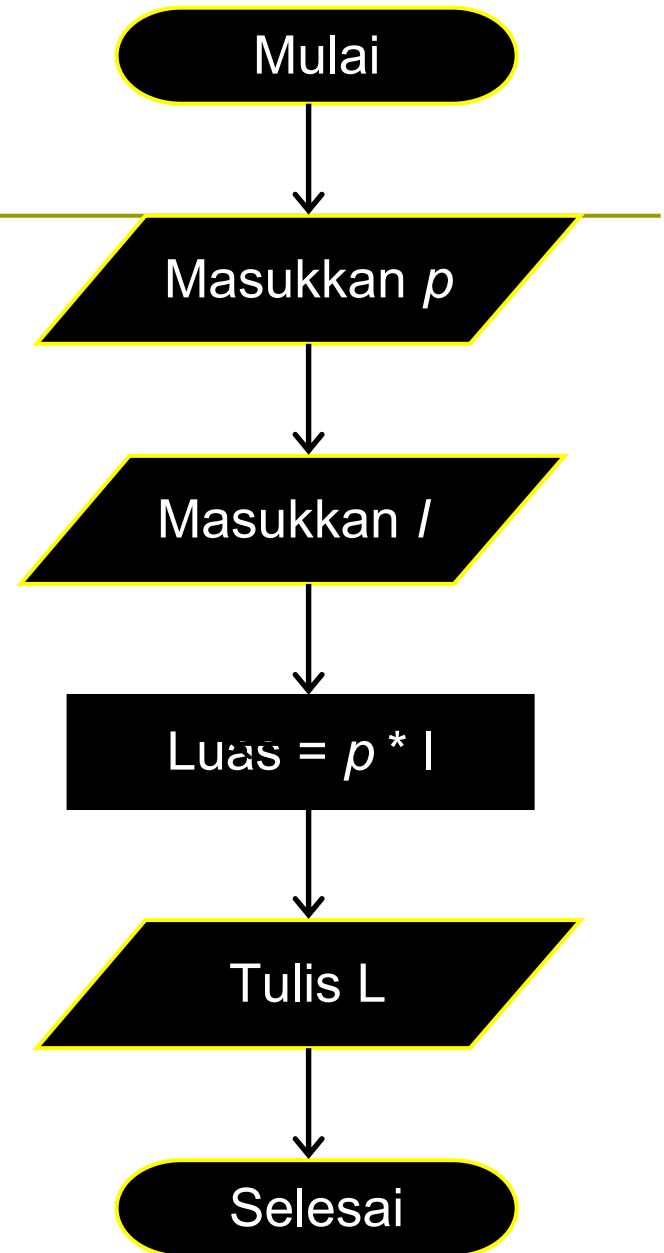
Contoh Flowchart

Problem:

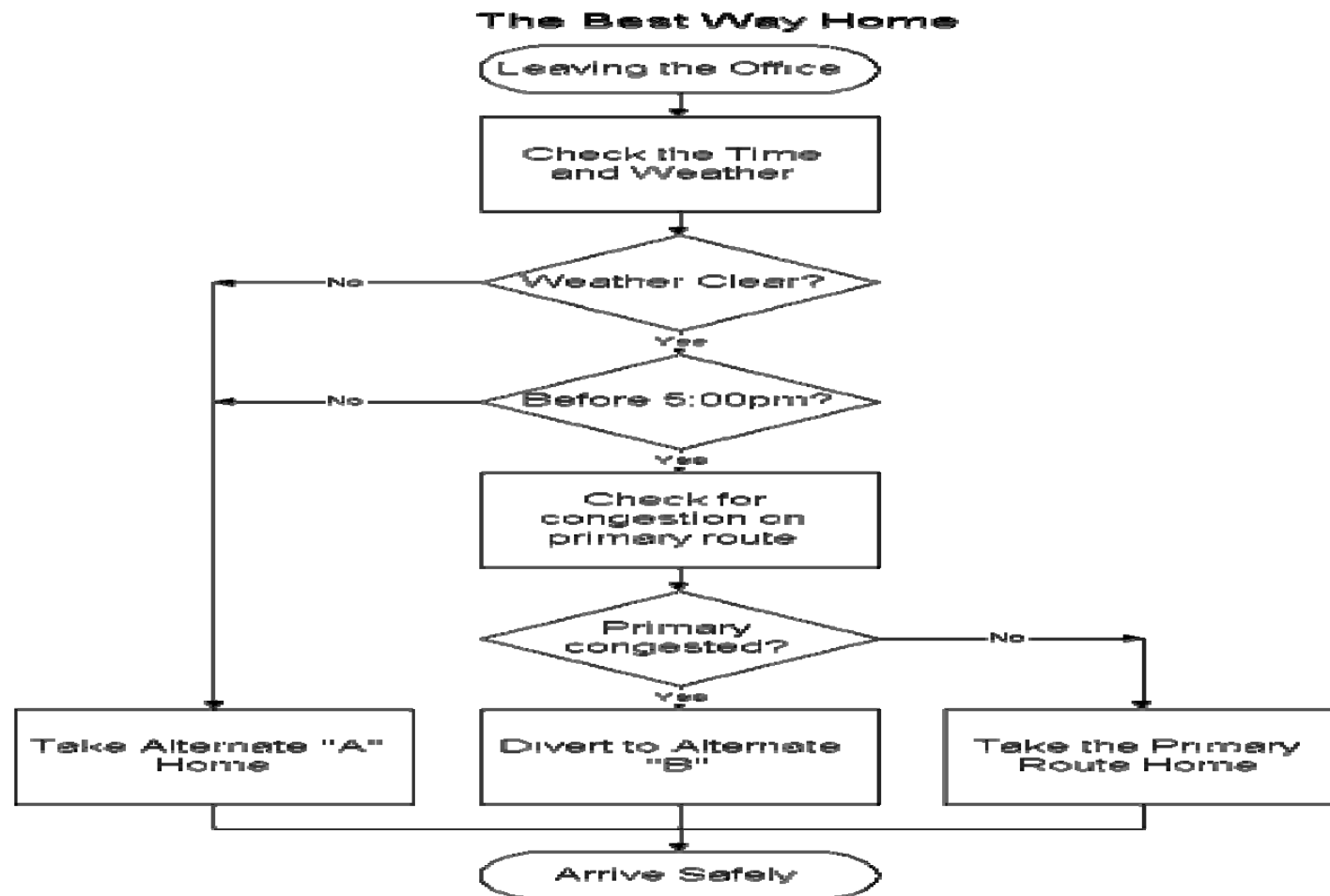
Menghitung
luas persegi panjang

Algoritma:

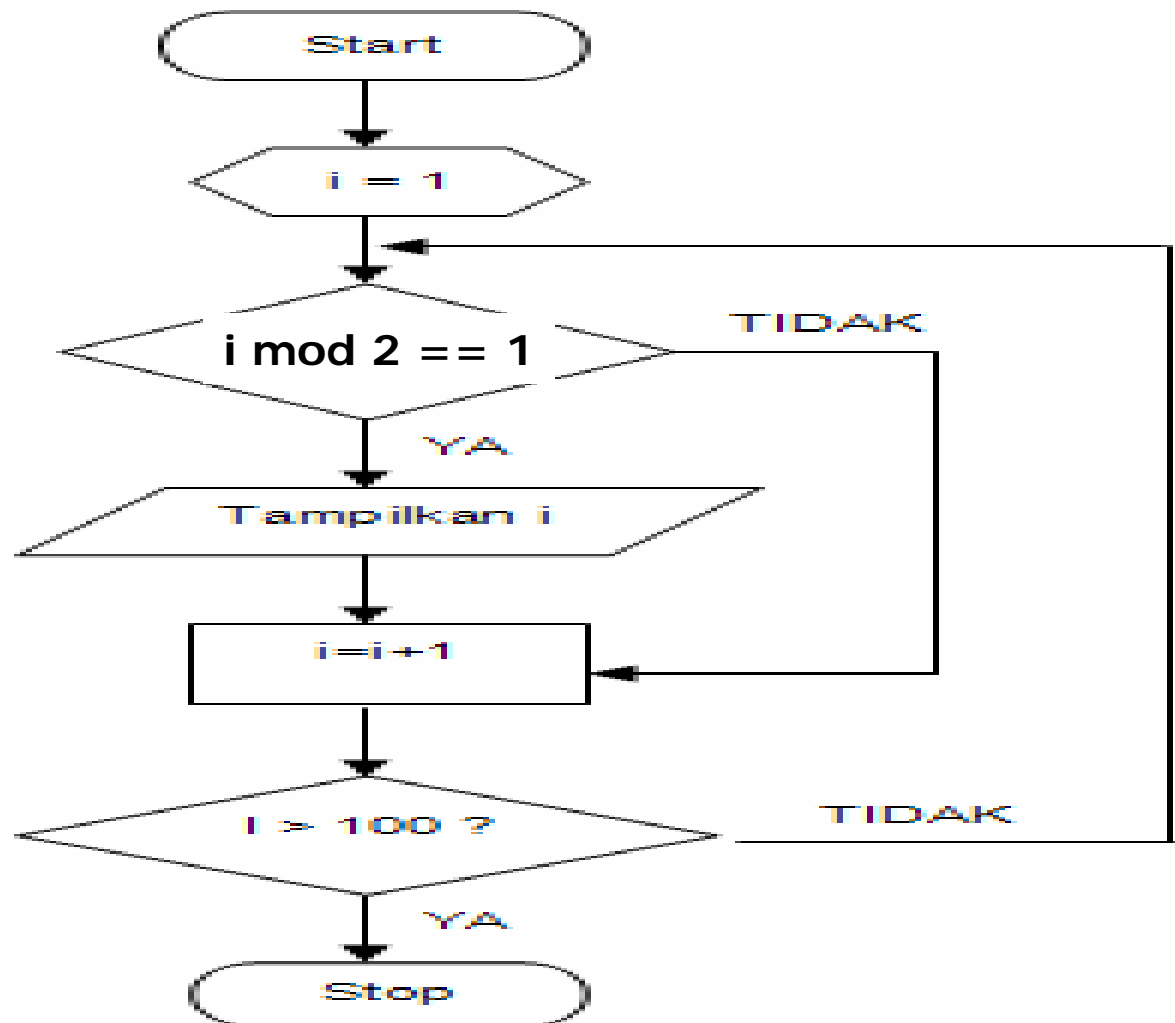
1. Masukkan panjang (p)
2. Masukkan lebar (l)
3. Hitung luas (L), yaitu panjang kali lebar
4. Cetak luas (L)



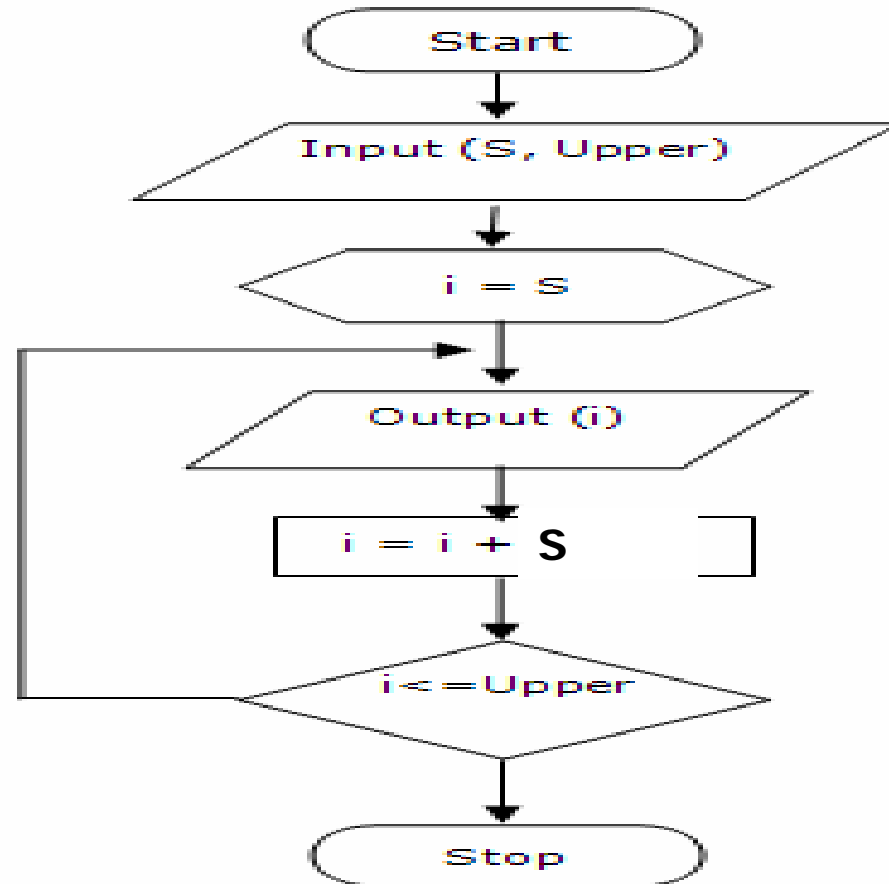
Flowchart : Mencari jalan pulang



Flowchart bilangan ganjil 1 - 100



Flowchart Kelipatan Bilangan

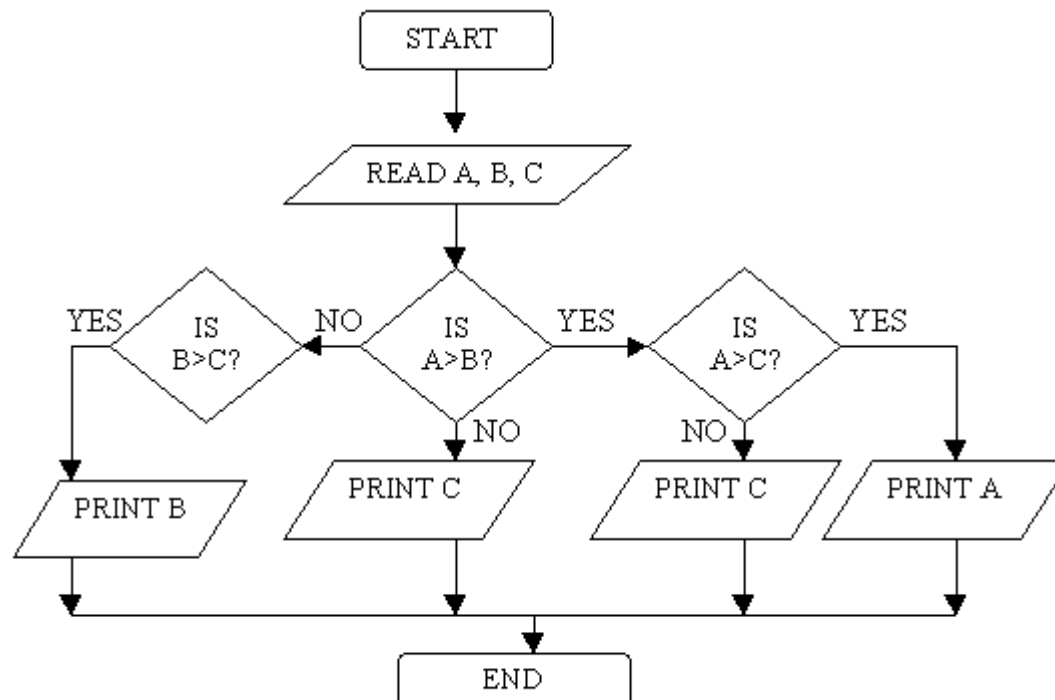


Keterangan :

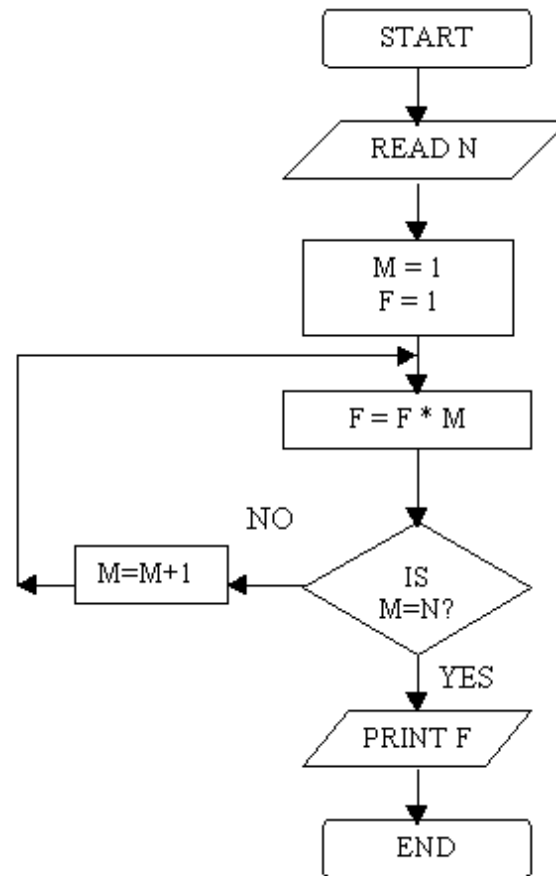
S : kelipatan yang ingin ditampilkan

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

- ▣ Buatlah flowchart untuk semua soal tadi!

NEXT

Translator Bahasa Pemrograman dan C