

LOGIKA DAN ALGORITMA PEMROGRAMAN

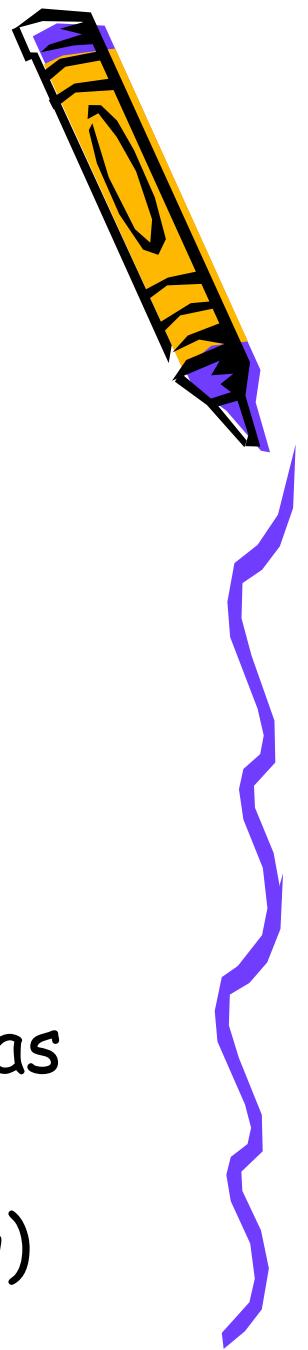
Pertemuan Ke-1



BUDI HARTONO

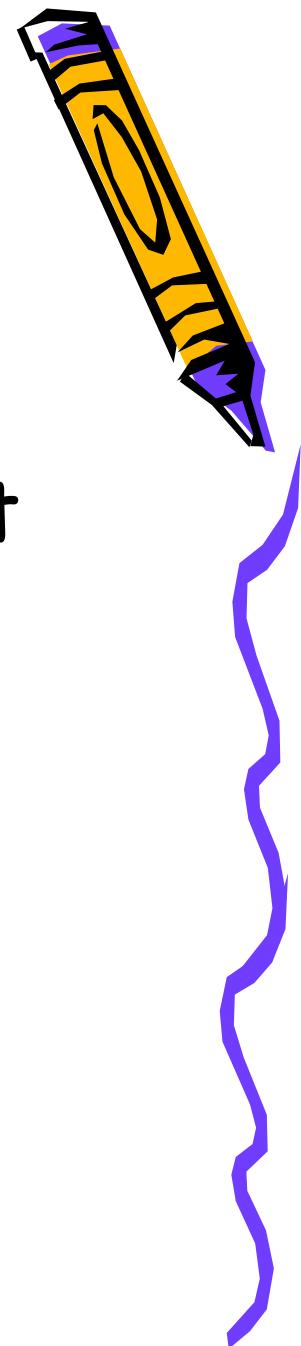
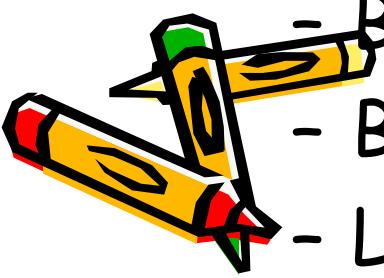
EVALUASI

- Tugas-Tugas → 30%
 - Mid→ 30%
 - UAS → 40%
 - Total Nilai = 100%
 - Syarat Ujian
 - Kehadiran Minimal 75%
 - Kehadiran antara 50% - 74% ada tugas tambahan
- Kehadiran <50 tidak boleh ujian(ulang)



Pengumpulan Tugas

- Lewat E-Mail :
htnbudi19@gmail.ac.id
- Di kumpulkan Langsung lewat print atau cetakan pada saat ujian
- Format
 - Halaman sampul, dll
 - Bab-1 Pendahuluan
 - Bab-2 Landasan Teori
 - Bab-3 Pembahasan
 - Bab-4 Penutup
 - Lampiran2, tambahan, dll



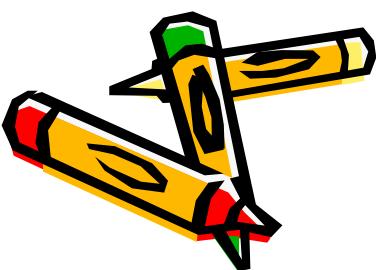
Capaian Pembelajaran

- Diharapkan mahasiswa, mampu mengerti, memahami dan dapat menjelaskan mengenai konsep algortima dalam pemrograman dan dapat mengimplementasikan kedalam bahasa pemrograman pascal atau yang lainnya
- Mahasiswa mampu menerapkan algoritma dalam menyelesaikan setiap permasalahan yang dihadapi baik di dunia kerja maupun dalam kehidupan sehari-hari



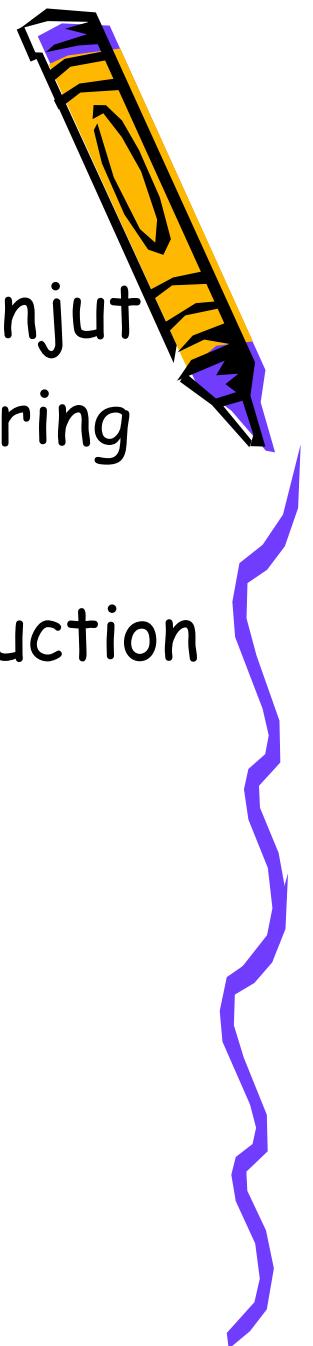
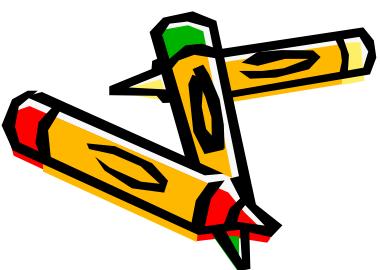
Deskripsi

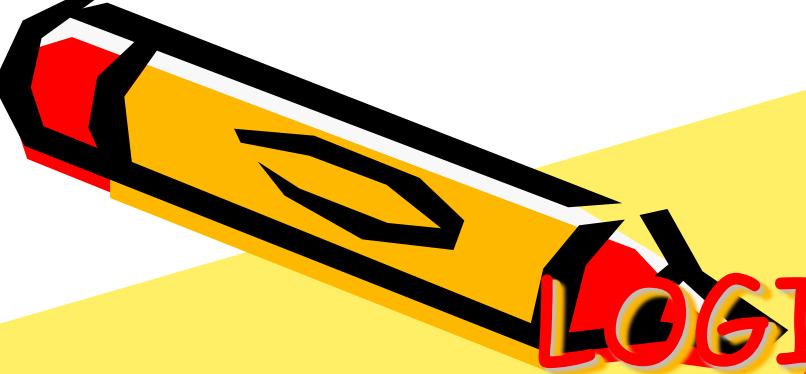
- Matakuliah ini membahas permasalahan logika algoritma dalam pemrograman mulai dari definisi algoritma, struktur dasar pemrograman, software yang digunakan, flowchart program, pengenalan tipe data, percabangan, perulangan, array, fungsi, procedure dan sampai pada logika pengurutan program dengan bahasa pascal atau bahasa lainnya



Kontrak Kuliah

- Pengenalan algoritma
- Pengenalan Pascal
- Input dan output-1
- Input dan output-2
- Kondisi/percabangan
- Percabangan Lanjut
- Perulangan
- Mid
- Perulangan Lanjut
- Manipulasi String
- Goto
- Procedure /Fuction
- Array-1
- Array-2
- Record
- UAS





LOGIKA DAN ALGORITMA PEMROGRAMAN

BUDI HARTONO

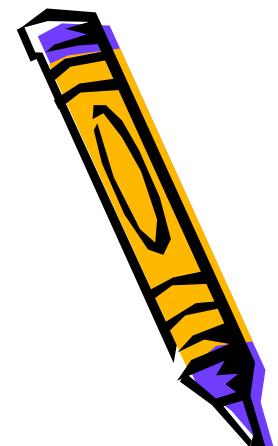


Tujuan Pembelajaran

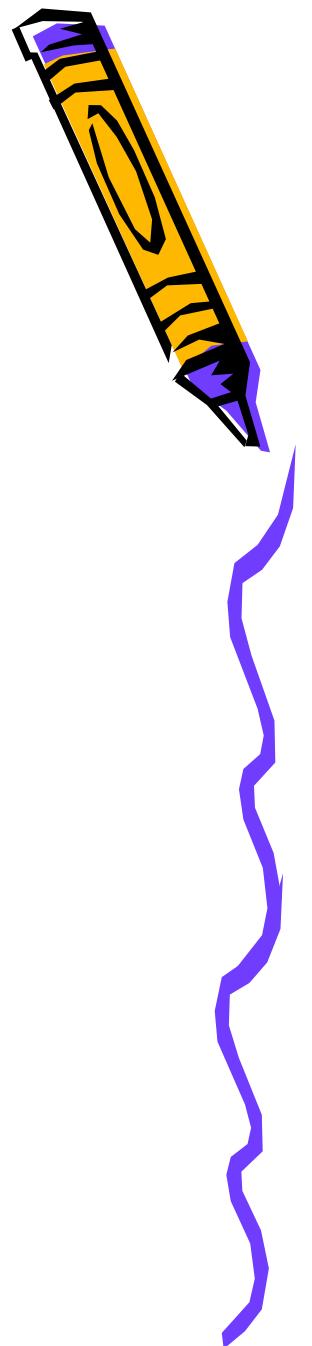
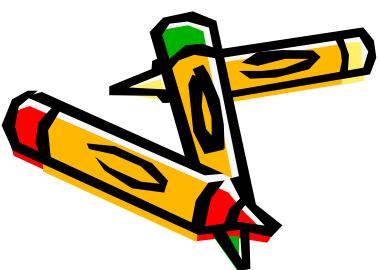
Setelah menyelesaikan pada bab ini,
Mahasiswa diharapkan dapat:

- Mahasiswa mampu memahami pengertian algoritma
- Mahasiswa mampu menuliskan algoritma dengan berbagai cara
- Mahasiswa mampu menguasai teknik dasar pemrograman dengan baik.

Mahasiswa mampu memahami algoritma-algoritma umum dalam pemrograman.



- Apa Itu Logika ?
- Apa Itu Algoritma?
- Apa Itu Program ?
- Jadi Logika Algoritma dan Pemrograman ?



PENGERTIAN DASAR

LOGIKA

Diperkenalkan pertama kali oleh Aristoteles (384-322 SM)



ALGORITMA

Diperkenalkan Oleh Ahli Matematika dari Persia (Uzbekistan) yang bernama Abu Jafar Muhammad Ibnu Musa Al-Khwarizmi (770-840). Di literatur barat dia lebih terkenal dengan sebutan Algorizm. Panggilan inilah yang kemudian dipakai untuk menyebut konsep algorithm yang ditemukannya. Dalam bahasa Indonesia kita kemudian menyebutkannya sebagai algoritma.



Pengertian

Definisi Logika

1. Logika berasal dari bahasa Yunani yaitu LOGOS yang berarti ilmu / Penalaran atau bentuk pemikiran.
2. Logika dapat diartikan ilmu yang memberikan prinsip-prinsip yang harus diikuti agar dapat berfikir valid menurut aturan yang berlaku

Definisi Algoritma

1. Langkah - langkah yang dilakukan agar solusi masalah dapat diperoleh.
2. Suatu prosedur yang merupakan urutan langkah-langkah yg berintegrasi.
3. Suatu metode khusus yang digunakan untuk menyelesaikan suatu masalah yang nyata. (Webster Dictionary)

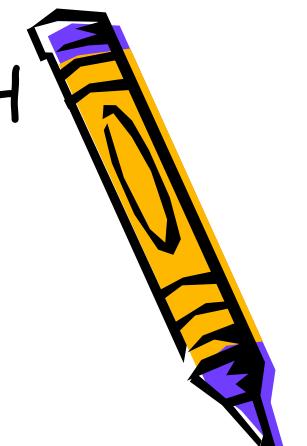
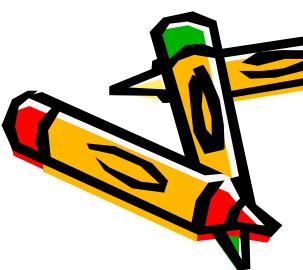
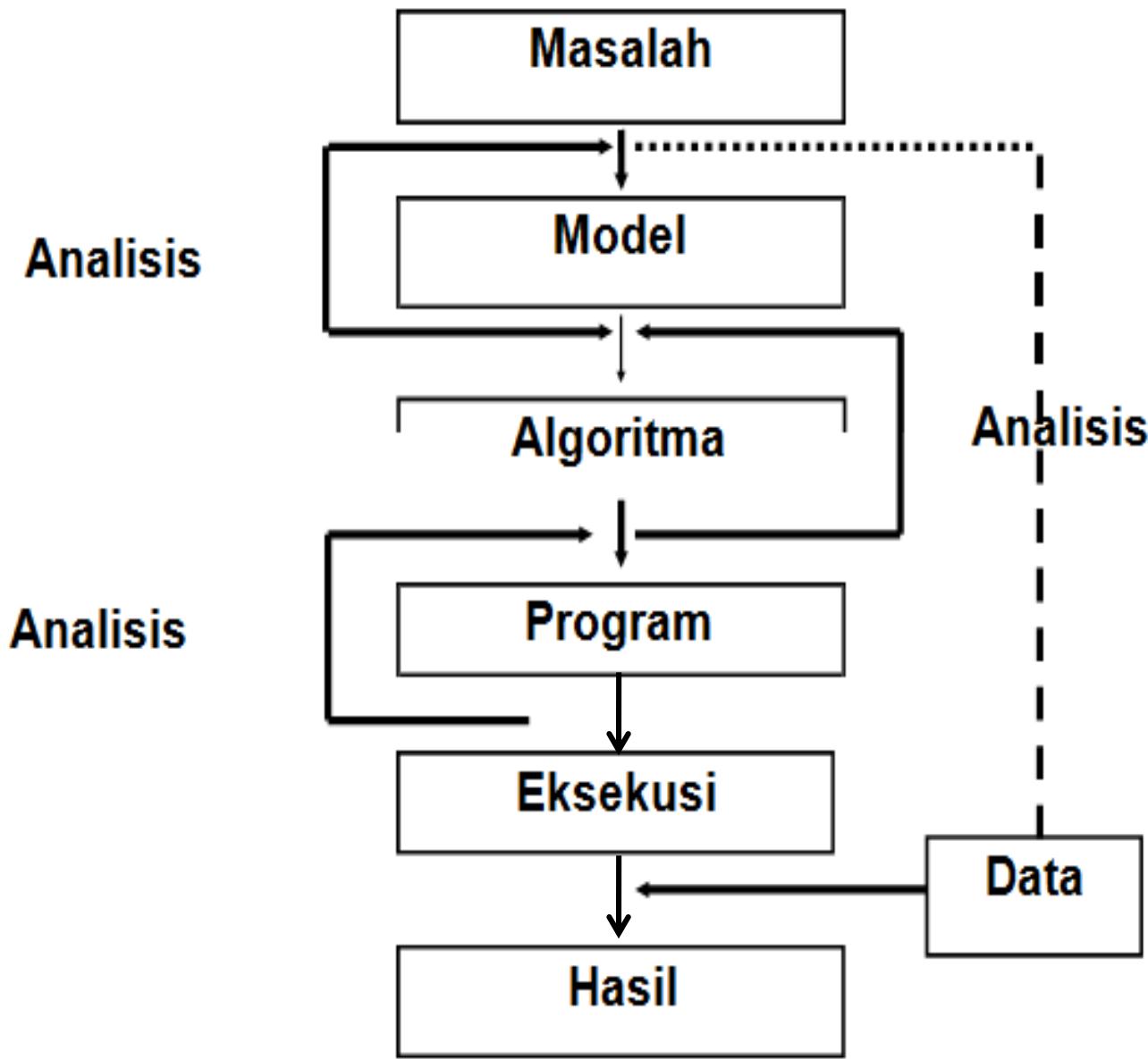


Pengertian Logika dan Algoritma

- Logika dan Algoritma adalah ilmu yang mempelajari cara penyelesaian masalah berdasarkan langkah-langkah terbatas yang logis dan sistematis dengan tujuan tertentu.

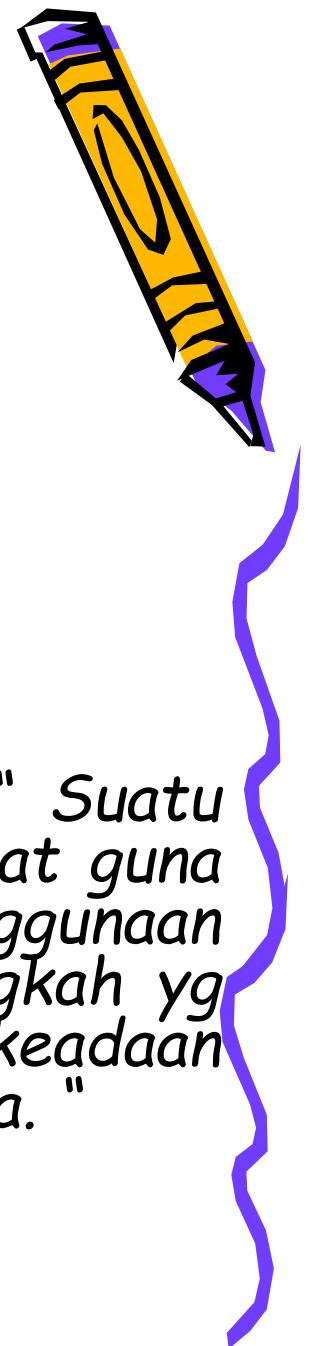
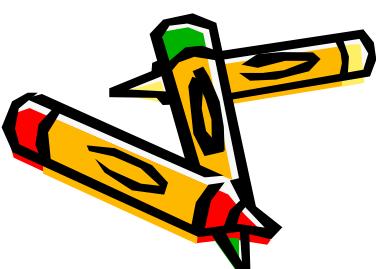


TAHAP PENYELESAIAN MASALAH

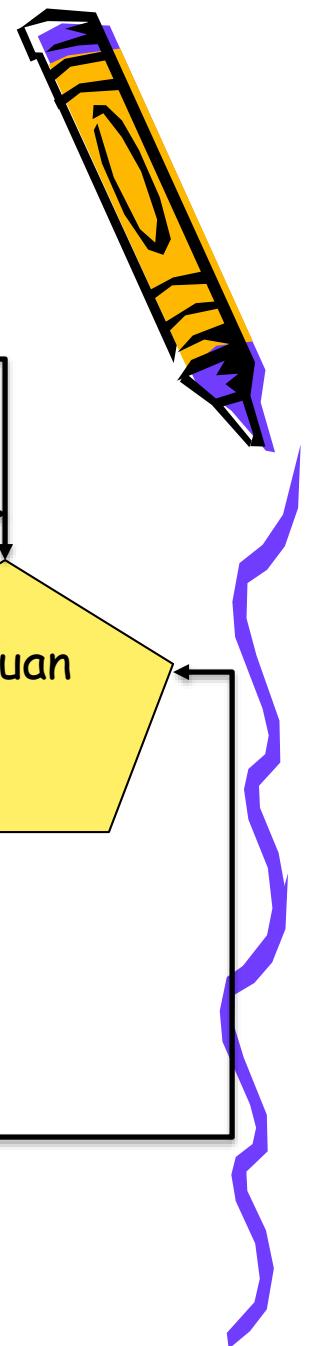
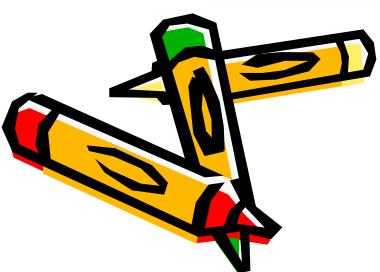
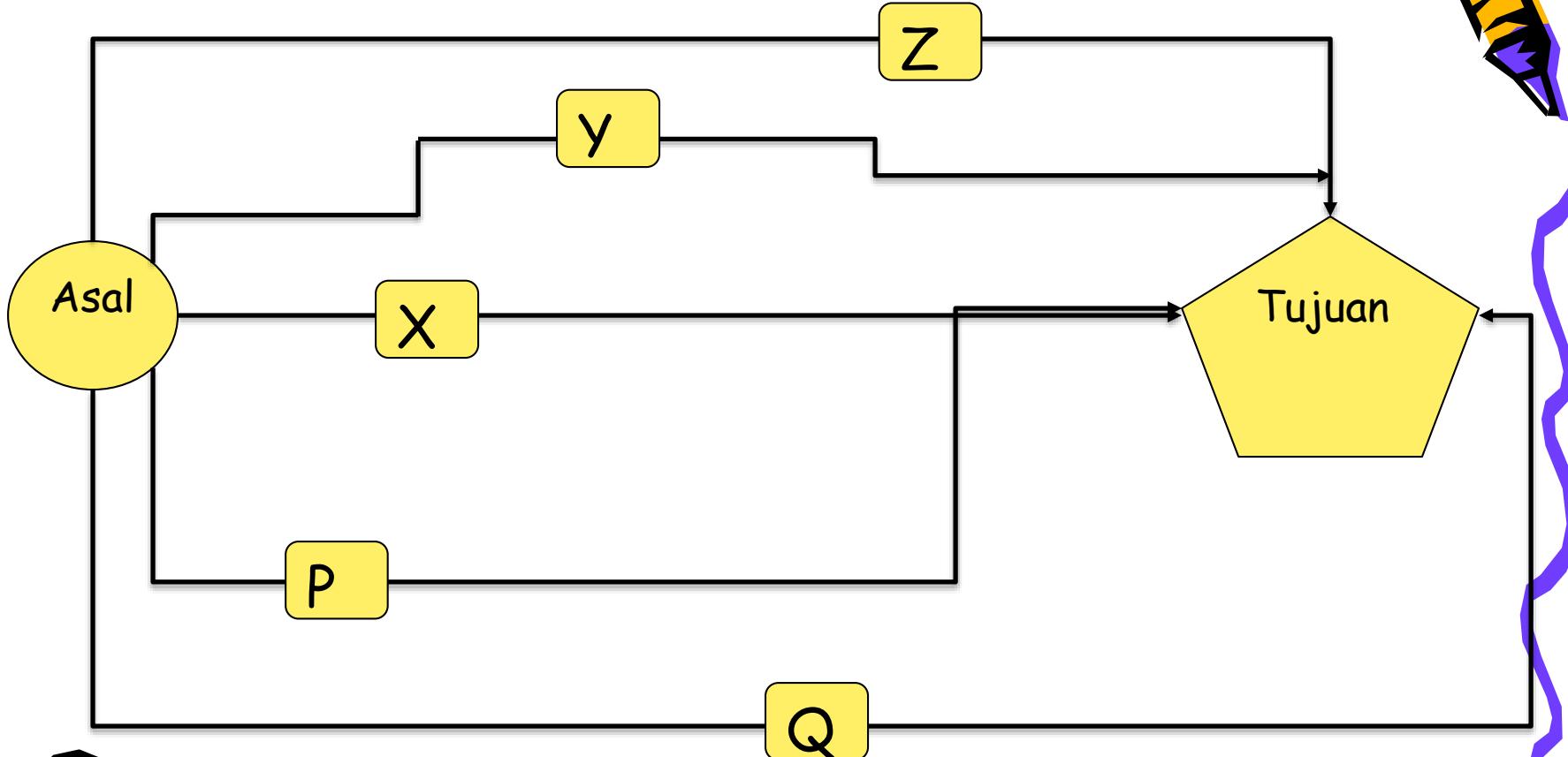


Kriteria Pemilihan Algoritma.

1. Ada Output,
 2. Efektifitas dan Efisiensi,
 3. Jumlah Langkahnya Berhingga,
 4. Berakhir,
 5. Terstruktur,
- ❖ Suatu Algoritma yg terbaik (**The Best**) : " Suatu algoritma harus menghasilkan output yg tepat guna (efektif) dlm waktu yg relatif singkat & penggunaan memori yg relatif sedikit (efesien) dgn langkah yg berhingga & prosedurnya berakhir baik dlm keadaan dip'oleh suatu solusi ataupun tdk ada solusinya. "



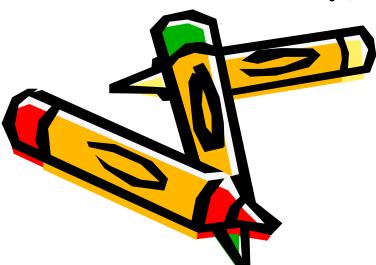
Analogi Algoritma



Contoh :

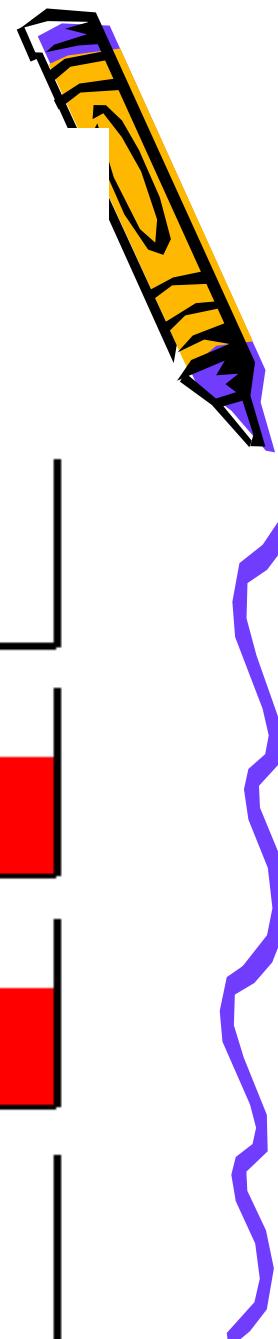
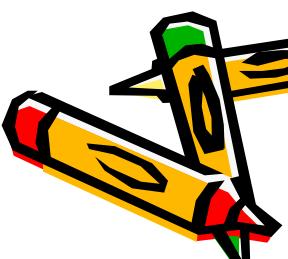
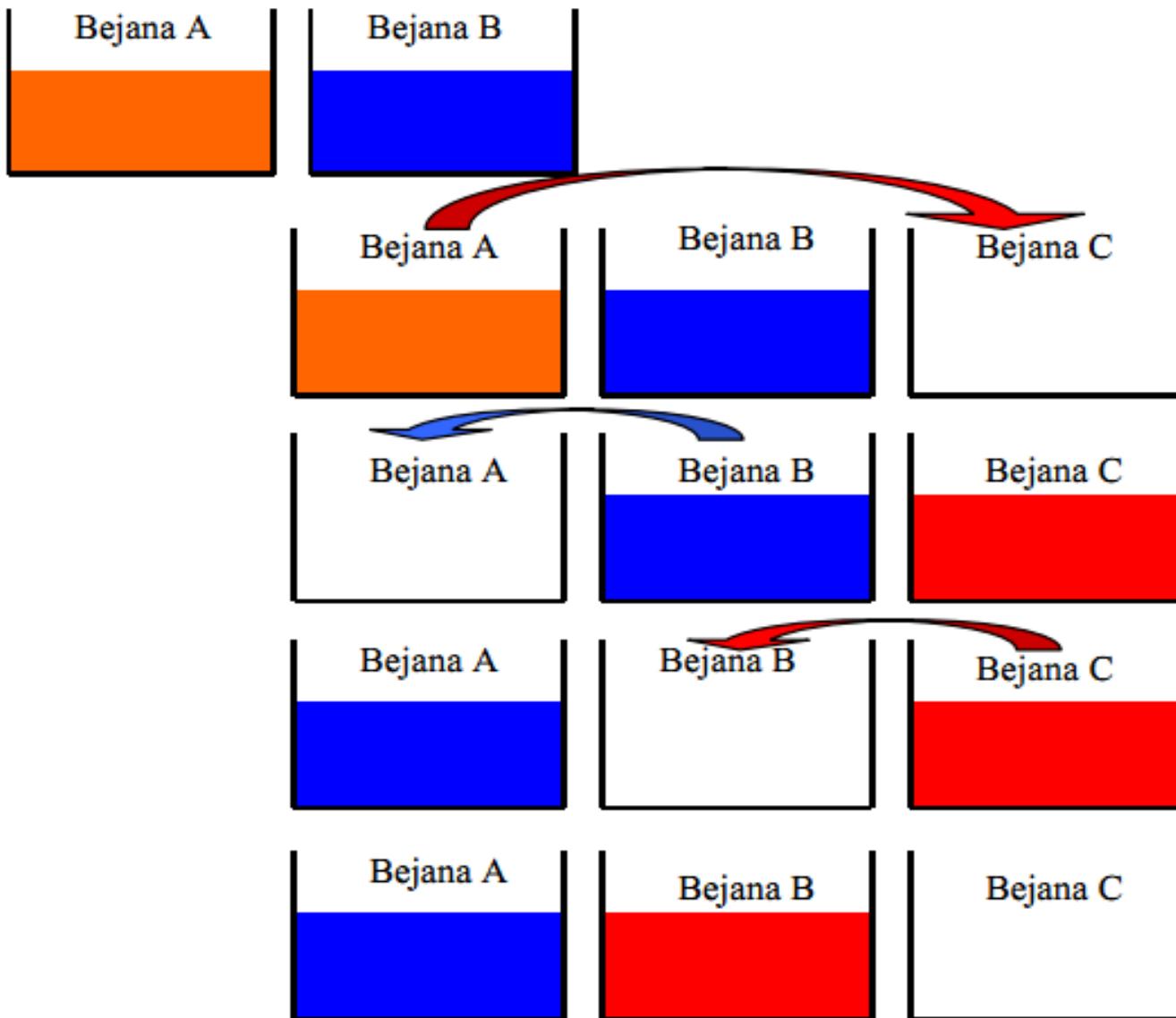
Sebuah prosedur ketika akan mengirimkan surat kepada teman:

1. Tulis surat pada secarik kertas surat
2. Ambil sampul surat atau amplop
3. Masukkan surat ke dalam amplop
4. Tutup amplop surat dengan lem perekat
5. Tulis alamat surat yg dituju, jika tdk ingat, lebih dahulu ambil buku alamat & cari alamat yg dituju, lalu tulis alamat tsb pd amplop surat.
6. Tempelkan perangko pada amplop surat
7. Bawa surat ke kantor pos utk diserahkan pd pegawai pos atau menuju ke bis surat untuk memasukkan surat ke dlm kotak/bis surat.



Contoh Lain adalah

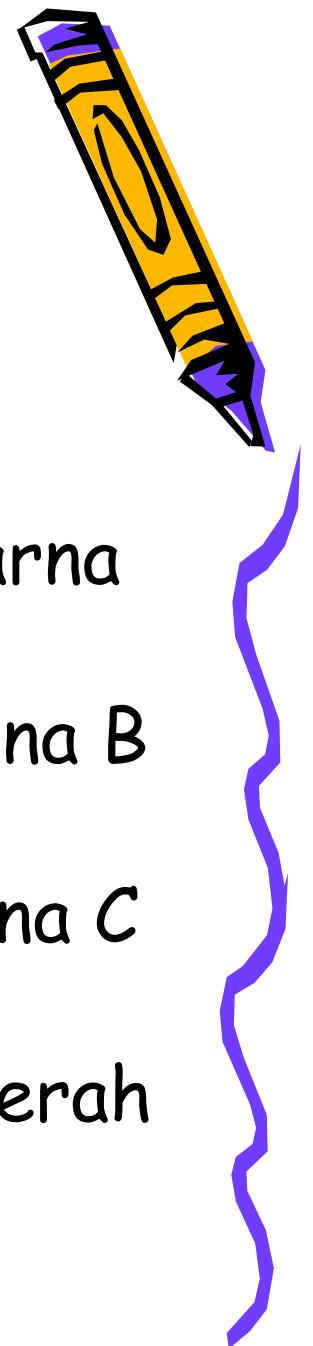
Algoritma tukar isi bejana



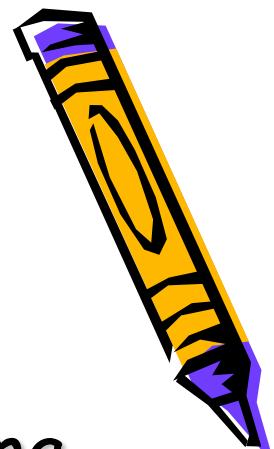
Keterangan :

Untuk memindahkan isi dari Bejana A dan Bejana B, dibutuhkan Bejana C (sebagai tempat penampungan sementara).

- 1) Isi Bejana C (Kosong) dari Bejana A (warna Merah)
- 2) Isi Bejana A (setelah Kosong) dari Bejana B (warna Biru)
- 3) Isi Bejana B (setelah Kosong) dari Bejana C (warna Merah)
- 4) Maka Isi Bejana A = Biru; Bejana B = Merah dan ; Bejana C = Kosong

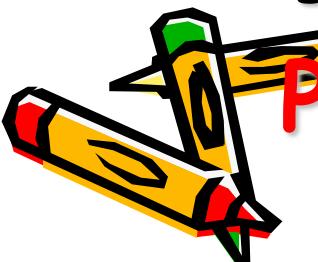


Apa Itu Program?



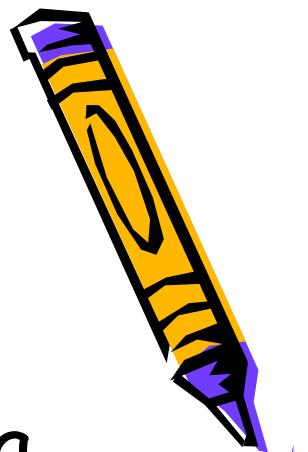
Definisi

- Program adalah formulasi sebuah algoritma dalam bentuk bahasa pemrograman, sehingga siap untuk dijalankan pada mesin komputer.
- Program berisikan kumpulan instruksi-instruksi tersendiri yang biasanya disebut *source code* yang dibuat oleh programmer (pembuat program).
- Program : Realisasi dari Algoritma.

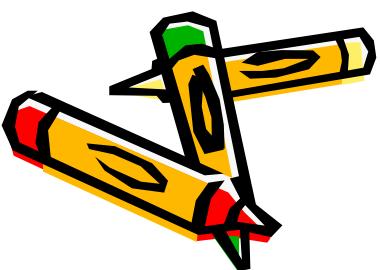


Program = Algoritma + Bahasa

Bahasa Pemrograman



- Bahasa pemrograman adalah bahasa buatan yang digunakan untuk mengedalikan perilaku dari sebuah mesin , biasanya berupa mesin komputer, sehingga dapat digunakan untuk memberitahukan komputer tentang apa yang harusdilakukan.



Klasifikasi Generasi Program



- Generasi ke-1: machine language (bahasa mesin).
 - Bahasa pemrograman ini berupa kode-kode mesin yang hanya bisa dipahami oleh mikroprosesor
- Generasi ke-2: assembly language
 - Pemrograman dengan bahasa assembly (rakitan). Bahasa assembly ini bukan bahasa asli mikroprosesor, meskipun begitu programer tetap harus mengetahui keunikan dari masing-masing mikroprosesor (register dan jenis instruksi).
- Generasi ke-3: procedural programming
 - Pada generasi ini didesain supaya mudah dipahami oleh manusia. Pada generasi ini mulai dikenalkan (FORTRAN, COBOL, ALGOL, BASIC, C, C++, Pascal, Java.)
- Generasi ke-4: 4 GL (fourth-generation language)
 - Generasi ke 4 adalah bahasa query terstruktur (SQL, Structured Query Language) ini menekankan pada basis data (SQL, Progress 4GL, Matlab, PowerBuilder)
- Generasi ke-5: Programming Language Based Object Oriented & Web Development



Pada generasi ini lebih menekankan pada aspek efisiensi dan penggunaan kembali (re-use-able). Generasi ke - 5 ini juga di kenal dengan nama "intelligent programming" (pemrograman kecerdasan) yang menekankan aspek otomatisasi dalam setiap prosesnya

Tingkatan Bahasa



- Bahasa pemrograman tingkat rendah
 - Bahasa mesin atau kode mesin merupakan satu-satunya bahasa yang bisa diolah komputer secara langsung tanpa transformasi sebelumnya (kompilasi).
- Bahasa pemrograman tingkat menengah
 - Bahasa tingkat menengah memberikan satu tingkat abstraksi di atas kode mesin
- Bahasa pemrograman tingkat tinggi
 - Pada tingkat tinggi (high level programming language) muncul pada generasi ke3 bahasa pemrograman , meskipun sudah berkembang sangat jauh dari bahasa pemrograman tingkat tinggi semula.
 - Bahasa pemrograman tingkat Object
 - Oriented dan Visual / Very-High Level bahasa pemrograman tingkat Object - Oriented dan Visual" merupakan "bahasa pemrograman tingkat tinggi".

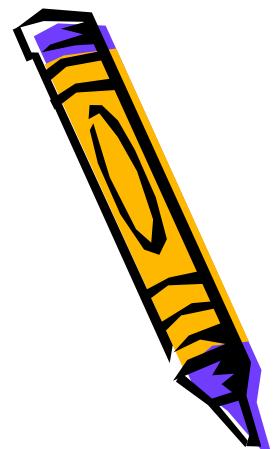


Mengapa Algoritma ?

- Pembuatan atau penulisan algoritma tidak tergantung pada bahasa pemrograman manapun.
- Notasi algoritma dapat diterjemahkan ke dalam berbagai bahasa pemrograman.
- Apapun bahasa pemrogramannya, output yang akan dikeluarkan sama karena algoritmanya sama.



Syarat Algoritma Baik?

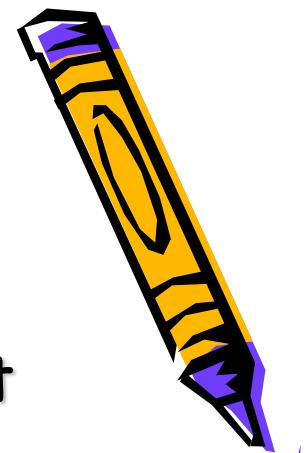


- Tingkat kepercayaannya tinggi (*reability*) Hasil yang diperoleh dari proses harus berakurasi tinggi dan benar.
- Pemrosesan yang efisien (*cost rendah*)
Proses harus diselesaikan secepat mungkin dan frekuensi kalkulasi yang sependek mungkin.
- Sifatnya general
Bukan sesuatu yang hanya untuk menyelesaikan satu kasus saja, tapi juga untuk kasus lain yang lebih general.



● Bisa dikembangkan (*expandable*)

Haruslah sesuatu yang dapat kita kembangkan lebih jauh berdasarkan perubahan requirement yang ada.



● Mudah dimengerti

Siapapun yang melihat, dia akan bisa memahami algoritma Anda. Susah dimengertinya suatu program akan membuat susah di-maintenance (kelola).

● Portabilitas yang tinggi (*portability*)

Bisa dengan mudah diimplementasikan di berbagai platform komputer.



Precise (tepat, betul, teliti)



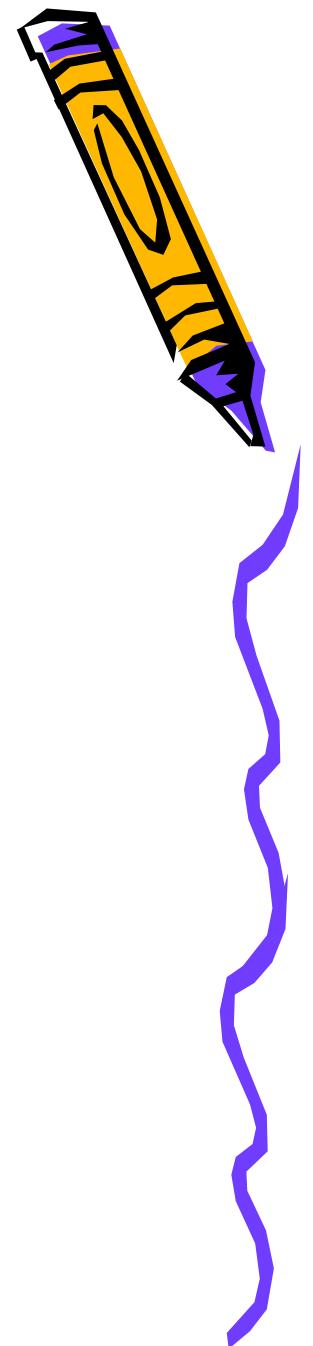
- Efektif

Tidak boleh ada instruksi yang tidak mungkin dikerjakan oleh pemroses yang akan menjalankannya.

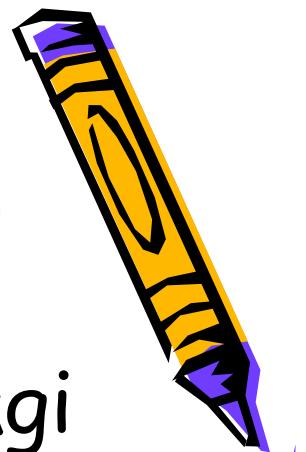
- Harus terminate

Jalannya algoritma harus ada kriteria berhenti.

- Output yang dihasilkan tepat.

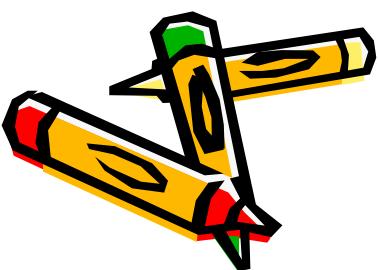


PENYAJIAN ALGORITMA

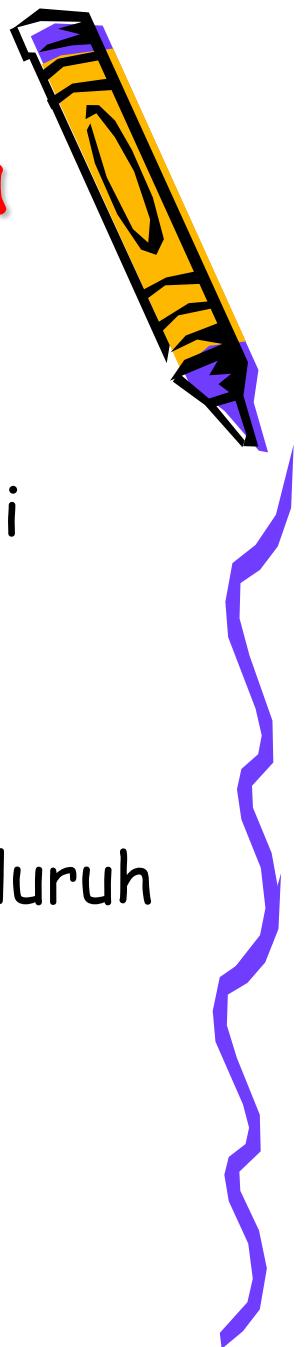


Bentuk penyajian untuk algoritma dibagi menjadi 3 (tiga) bentuk penyajian, yaitu :

- Algoritma dengan struktur Bahasa Indonesia
- Algoritma dengan *Pseudocode*
- Algoritma dengan *Flowchart*

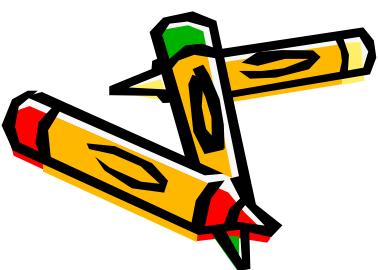


Struktur Bahasa Indonesia



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.



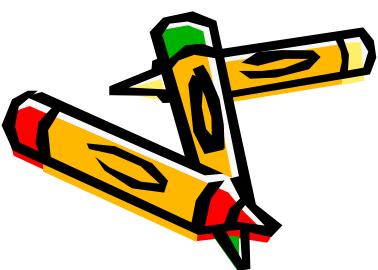
Struktur Bahasa Indonesia



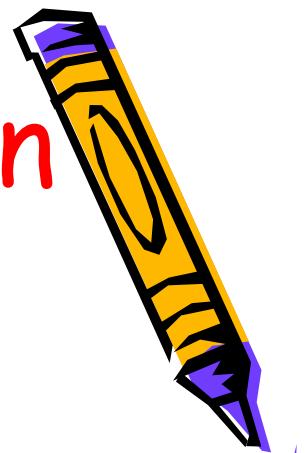
Contoh : Menghitung rata-rata tiga buah data

Algoritma dengan struktur bahasa Indonesia :

- 1) Baca bilangan a, b, dan c
- 2) Jumlahkan ketiga bilangan tersebut
- 3) Bagi jumlah tersebut dengan 3
- 4) Tulis hasilnya



Kelebihan dan Kekurangan



- Kelebihan
 - Mudah dipahami oleh semua orang (sifatnya yang general)
- Kekurangan
 - Timbul masalah pada bahasa yang bias (ambiguitas) atau penerimaan yang berbeda



Pseudo-Code

Pseudocode adalah bentuk informal untuk mendeskripsikan algoritma yang mengikuti struktur bahasa pemrograman tertentu : Tujuan dari penggunaan pseudocode adalah supaya :

1. Lebih mudah di baca oleh manusia
2. Lebih mudah di pahami
3. Lebih mudah dalam menuangkan ide pemikiran

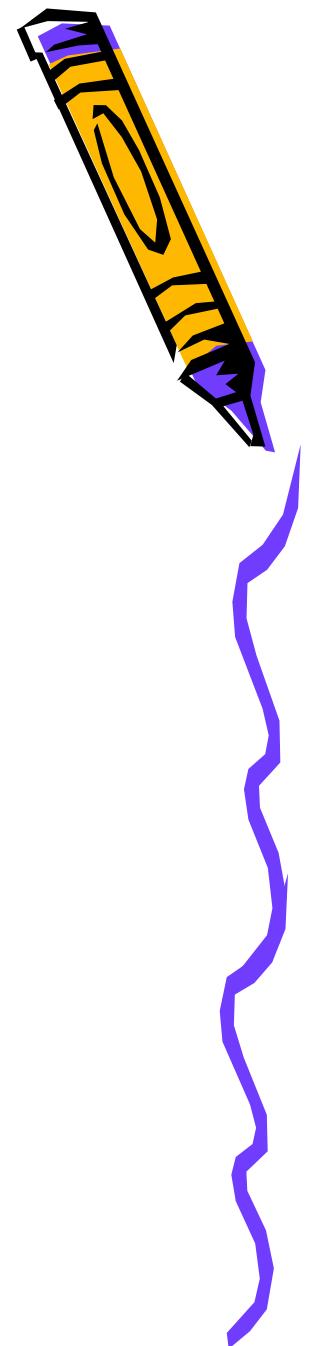


Pseudo-Code

Contoh (1) : Menghitung rata-rata tiga buah data

Algoritma dengan struktur pseudocode :

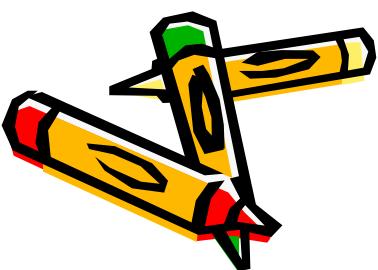
- 1) input (a, b, c)
- 2) $Jml = a+b+c$
- 3) $Rerata = Jml/3$
- 4) Output (Rerata)



Kelebihan dan Kekurangan

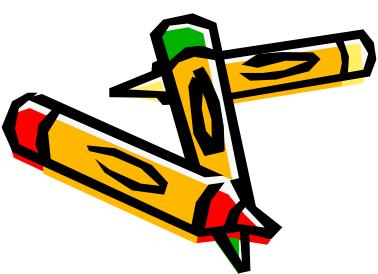


- Kelebihan
 - Bahasanya pasti dan tidak bias
- Kekurangan
 - Hanya bisa dipahami oleh orang-orang tertentu saja, karena adanya kode-kode tertentu

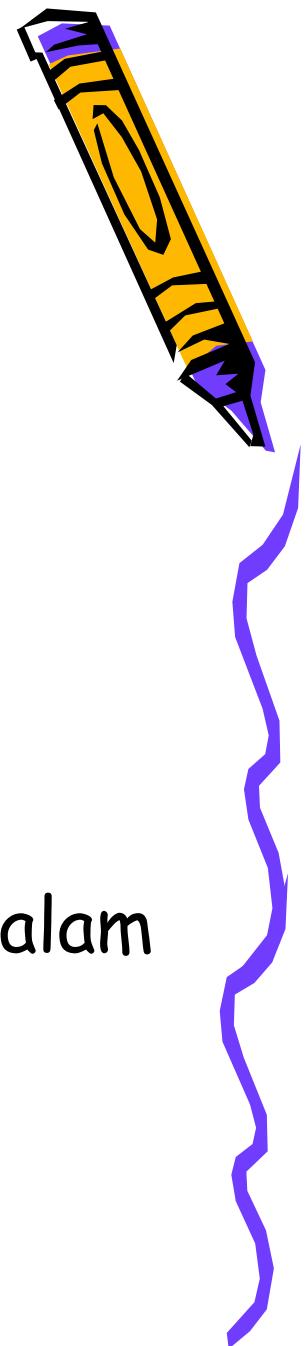


Flowchart

Flowchart adalah penggambaran secara grafik dari langkah-langkah dan urut-urutan prosedur dari suatu program. Flowchart menolong analis dan programmer untuk memecahkan masalah kedalam segmen-semen yang lebih kecil dan menolong dalam menganalisis alternatif-alternatif lain dalam pengoperasian.



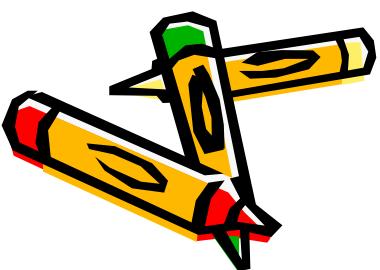
Flowchart



Kegunaan:

- Untuk mendesain program
- Untuk merepresentasikan program

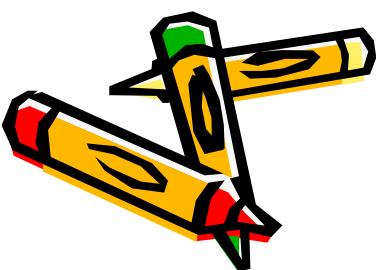
Maka, flowchart harus dapat
Merepresentasikan komponen-komponen dalam
bahasa pemrograman



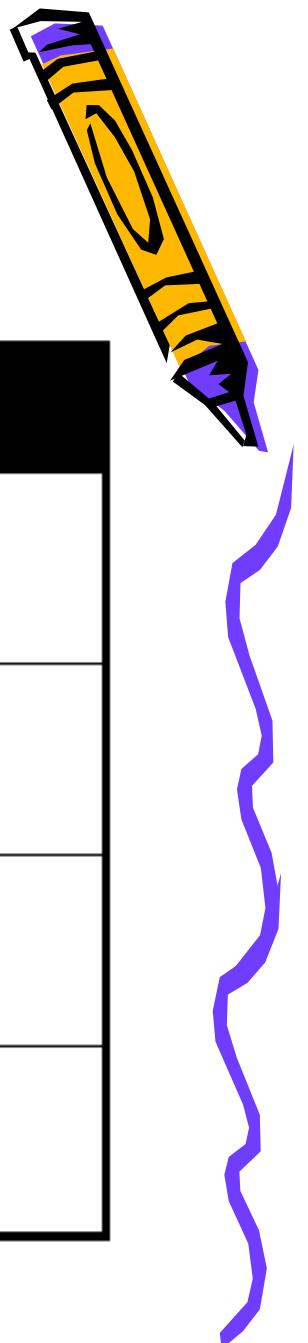
Kelebihan dan Kekurangan



- Kelebihan
 - Lebih mudah dipahami karena menggunakan simbol-simbol yang jelas
- Kekurangan
 - Bilang program terlalu banyak dan berulang akan menyebabkan kesulitan dalam penggambaran flowchartnya.



Lambang

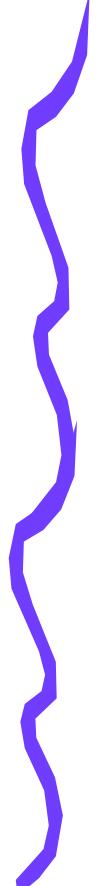
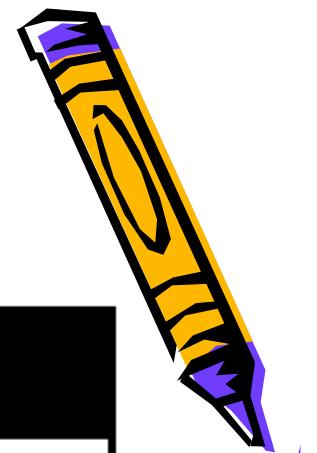


Keterangan	Lambang
Mulai/selesai (terminator)	
Aliran data	
<i>Input/Output</i>	
Proses	



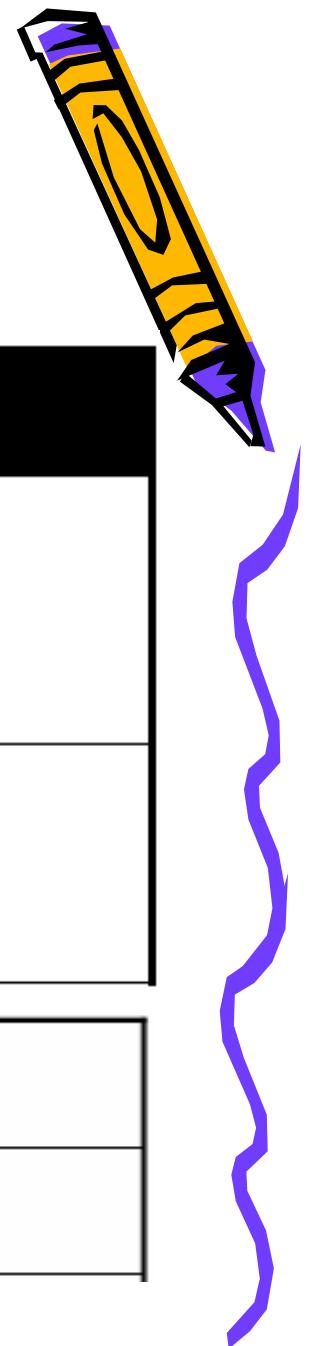
Lambang (2)

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



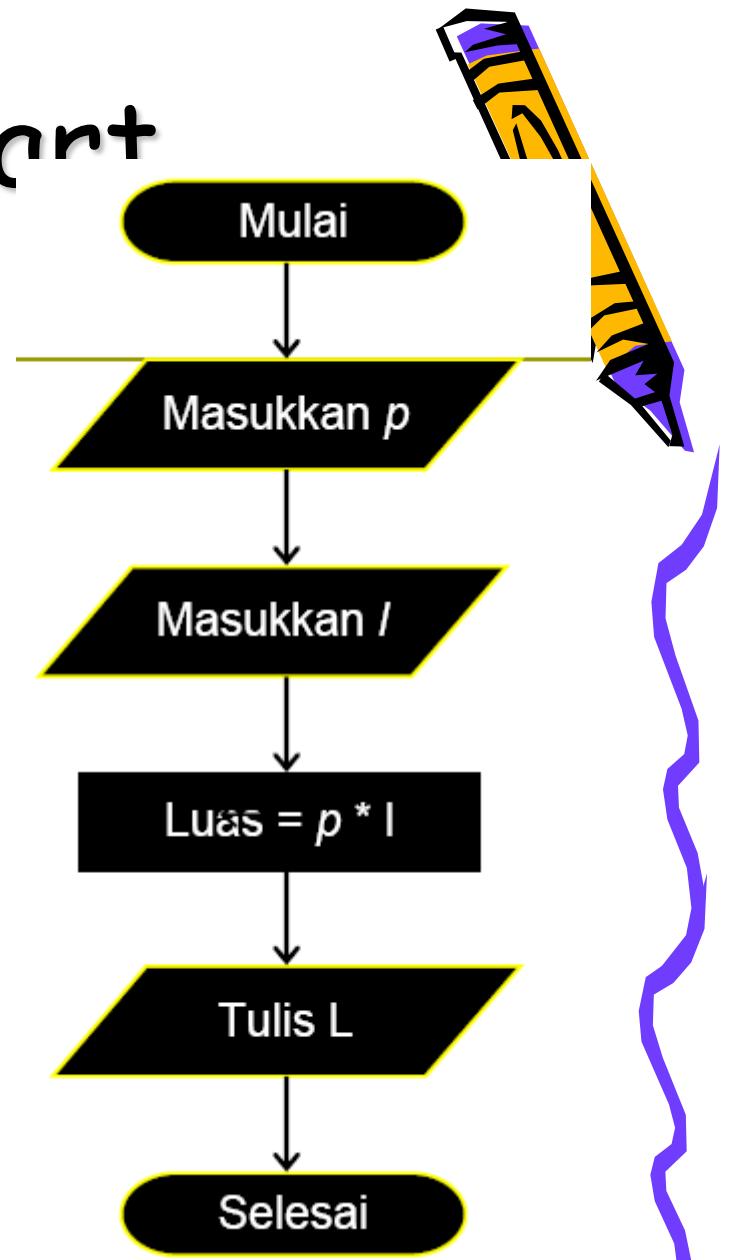
Lambang (3)

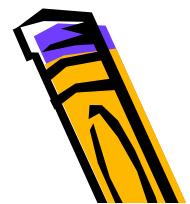
Keterangan	Lambang
<i>Connector (di halaman yg sama)</i>	A
<i>Off page Connector (halaman lain)</i>	B
Dokumen / Multi dokumen	 
Harddisk	



Contoh Flowchart

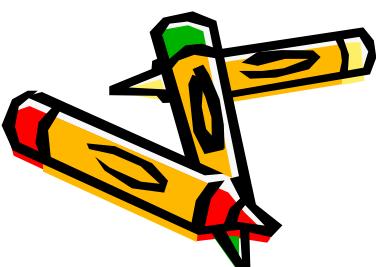
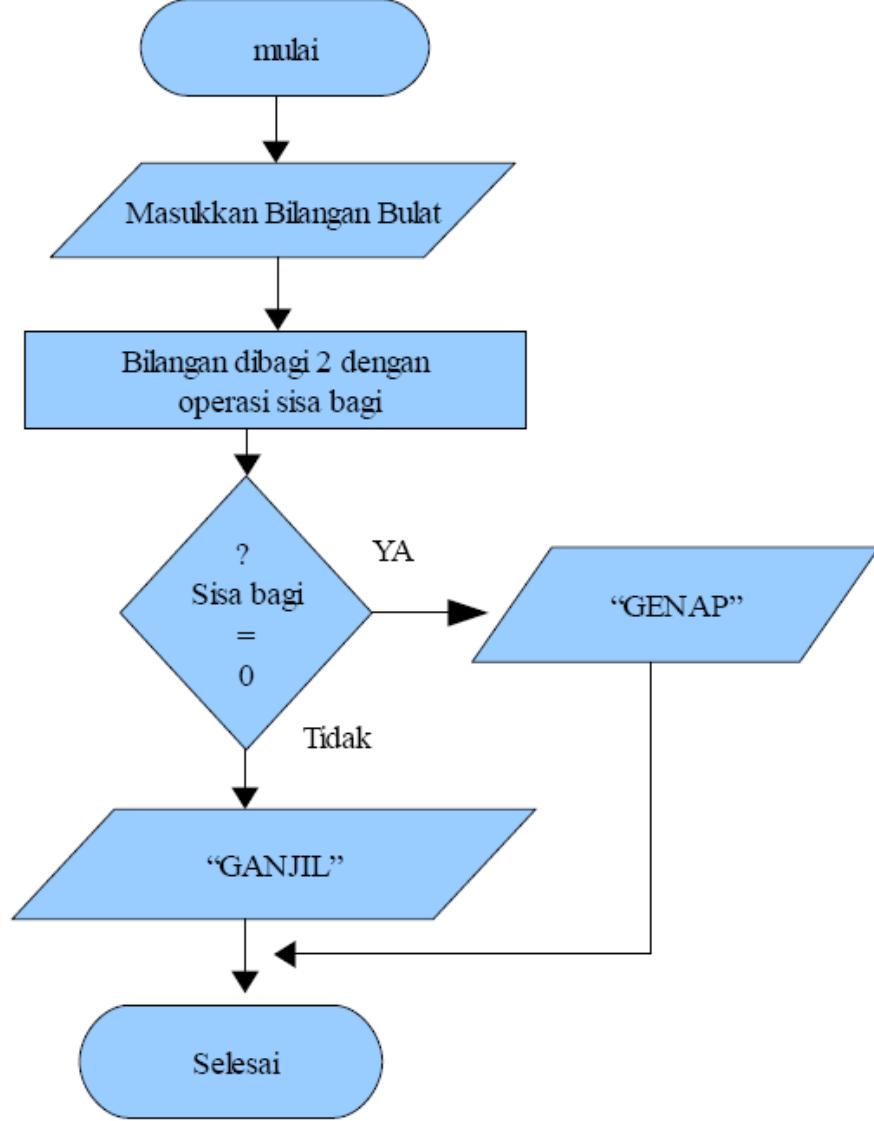
- Problem:
Menghitung
Luas persegi panjang
Algoritma:
 1. Masukkan panjang(p)
 2. Masukkan lebar(l)
 3. Hitung luas(L),
yaitu panjang kali lebar
 4. Cetak luas(L)





Contoh Flowchart

- Problem:
Menentukan
Bilangan ganjil
atau
Genap





STRUKTUR DASAR ALGORITMA

1. Urutan (*sequence*)

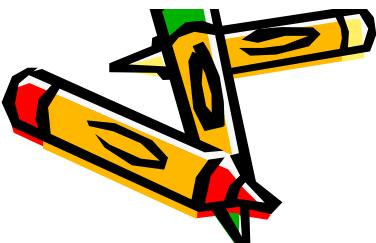
Terdiri dari satu atau lebih instruksi. Tiap instruksi dikerjakan urut sesuai urutan penulisannya.

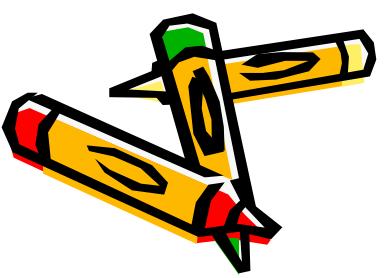
2. Pemilihan (*selection*)

Instruksi dikerjakan bila telah memenuhi kondisi tertentu, contoh : if kondisi then aksi1(*kondisi = true*) else aksi2(*kondisi = false*)

3. Pengulangan (*repetition*)

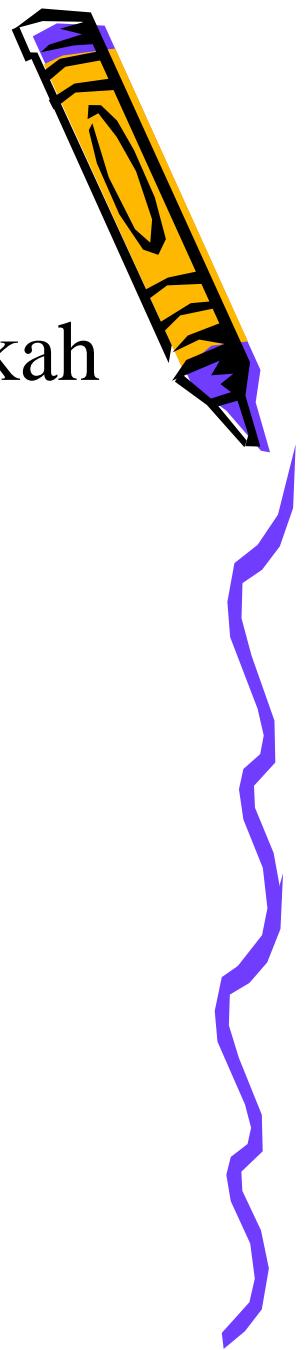
Mengerjakan instruksi yang berulang, contoh : repeat aksi until kondisi (*true = selesai; false = ulang*).



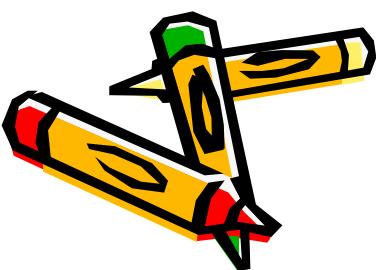


1. Dalam menyusun suatu program, langkah pertama yang harus dilakukan adalah :

- a. Membuat program
- b. Membuat Algoritma
- c. Membeli komputer
- d. Proses
- e. Mempelajari program

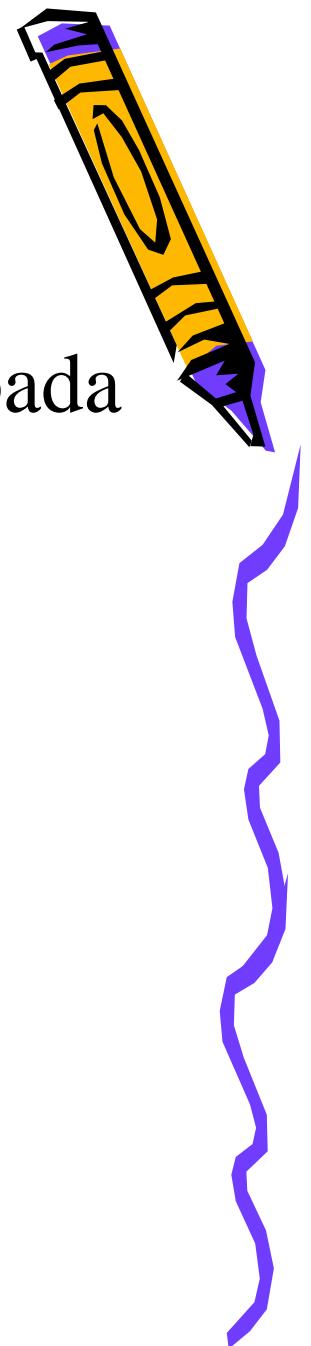
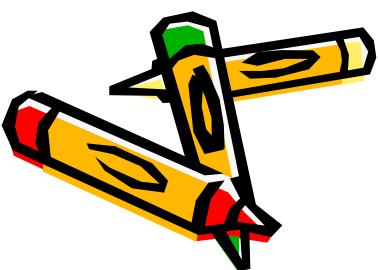


2. Sebuah prosedur langkah demi langkah yang pasti untuk menyelesaikan sebuah masalah di sebut :
- a. Proses
 - b. Program
 - c. Algoritma
 - d. Step
 - e. Diagram



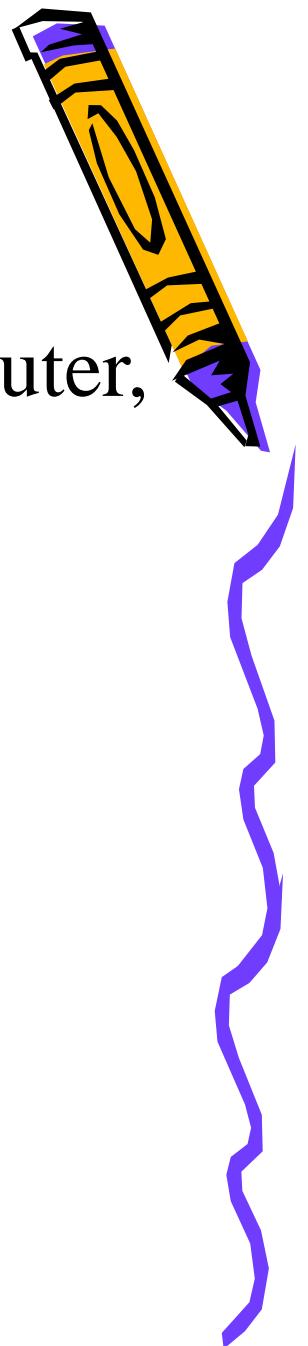
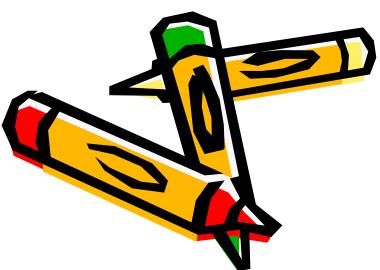
3. Pseudocode yang digunakan pada penulisan algoritma berupa :

- a. Bahasa Inggris
- b. Bahasa Puitis
- c. Bahasa pemograman
- d. Sembarang bahasa asal terstruktur
- e. Bahasa Mesin



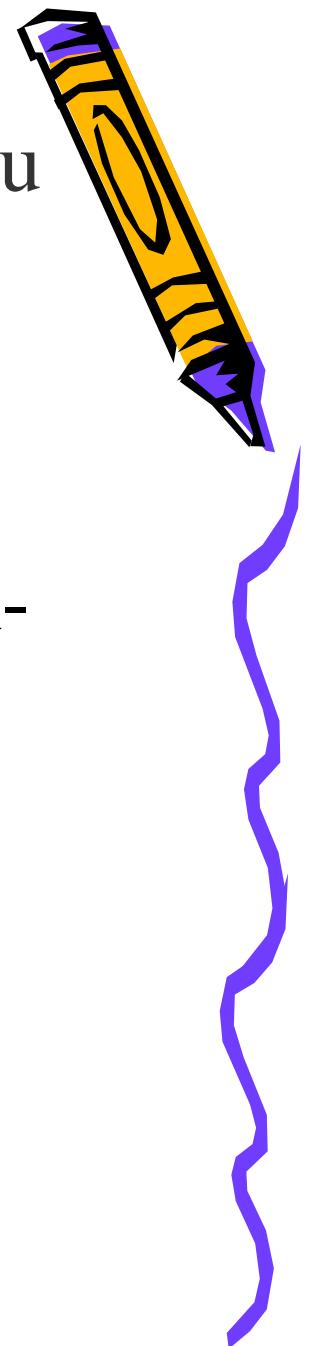
4. Pada pembuatan program komputer, algoritma dibuat :

- a. Sebelum pembuatan program
- b. Pada saat program dibuat
- c. Sesudah pembuatan program
- d. Pada saat verifikasi program
- e. Pada saat di jalankan



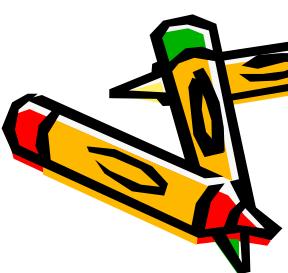
5. Tahapan dalam menyelesaikan suatu masalah adalah :

- a. Masalah-Pseudocode-Flowchart-Program-Eksekusi-Hasil
- b. Masalah-Algoritma-Flowchart-Program-Eksekusi-Hasil
- c. Masalah-Model-Algoritma-Eksekusi-Hasil
- d. Masalah-Model-Algoritma-Program-Eksekusi-Hasil



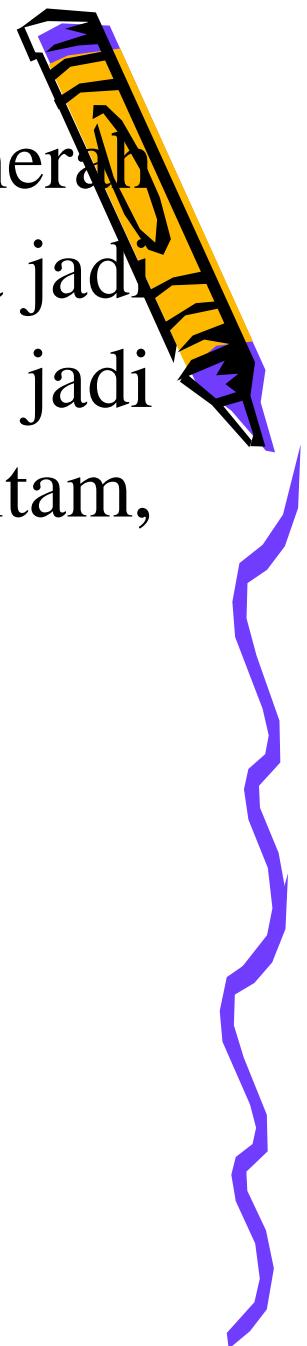
6. Diketahui bahwa kantong P kosong. Kantong Q berissi 10 buah kelereng dan kantong R berisi 15 kelereng. Apabila yang terbawa hanya sebuah kantong dan dikatakan **BUKAN** kantong P yang terbawa, Maka jumlah kelereng yang terbawa adalah :

- a. 10
- b. 15
- c. 10 atau 15
- d. 10 dan 15
- e. Kosong



7. Diberikan algoritma : Apabila warna merah maka jadi hijau. Apabila warna hijau maka jadi putih, selain warna merah dan hijau maka jadi ungu. Jika kondisi input warna adalah hitam, maka warna jadi :

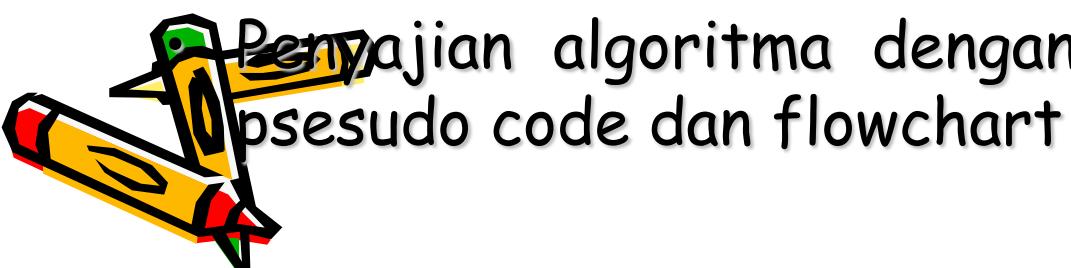
- a. Merah
- b. Ungu
- c. Hijau
- d. Putih
- e. Abu-abu



Penutup



- Logika adalah Suatu prosedur yang merupakan urutan langkah- langkah yg berintegrasi.
- Program adalah berisikan kumpulan instruksi-instruksi tersendiri yang biasanya disebut *source code* yang dibuat oleh programmer (pembuat program).
- Bahasa pemrograman adalah bahasa buatan yang digunakan untuk mengedalikan perilaku dari sebuah mesin , biasanya berupa mesin komputer, sehingga dapat digunakan untuk memberitahukan komputer tentang apa yang harusdilakukan



Penyajian algoritma dengan 3 metode (bahasa, psesudo code dan flowchart


 A large, central word cloud containing the words "thank you" in various languages. The word "thank" is in red and "you" is in green. Other words in the cloud include "danke", "gracias", "merci", "obrigado", "dank je", "teşekkür ederim", "ngiyabonga", "mehran", and "mochchakkeram". The entire word cloud is set against a white background with a decorative border featuring purple wavy lines and yellow/black pencils in the corners.

