2. Penyelesaian Masalah & Pengenalan Kepada Pengaturcaraan

2.0 Penyelesaian masalah dgn komputer

- Dari Bab 1 : Komputer boleh menyelesaikan masalah tertentu jika diberi arahan dgn menulis aturcara.

- Pengaturcaraan **bukan** hanya menulis aturcara.

- Penyelesaian masalah dengan komputer adalah prosedur untuk menentukan penyelesaian kepada sesuatu masalah dan menyatakannya dalam bentuk yang jelas - PALING SUSAH.

- Penyelesaian masalah (PM) memerlukan kemampuan seni & sains. Bidang kejuruteraan & sains menggunakan kaedah kejuruteraan & sains utk menyelesaikan masalah.

- Proses pembangunan perisian yg menggunakan kaedah PM kejuruteraan & sains dinamakan kejuruteraan perisian.

- PM bidang kejuruteraan lwn kejuruteraan perisian (KP).

Langkah^2 PM Kejuruteraan	**Langkah^2 PM KP**
1. Mengenali & memahami masalah	1. Dapatkan spesifikasi keperluan masalah/sistem
2. Mengumpul maklumat	2. Analisa masalah
3. Memilih teori & prinsip yg bersesuaian 4. Membuat andaian^2 yg diperlukan	3. Rekabentuk penyelesaian
5. Selesaikan masalah	4. Pengkodan & pelaksanaan
6. Menguji keputusan	5. Pengujian perisian
	6. Penyenggaraan

2.1 Pembangunan Perisian

- Langkah^2 proses pembangunan perisian:

1. Spesifikasi keperluan
2. Analisa masalah
3. Rekabentuk
4. Pengkodan & pelaksanaan
5. Penentusahan & Pengujian
6. Penyenggaraan

2.1.1 Spesifikasi Keperluan

- Memahami masalah & persekitaran masalah yg hendak diselesaikan.

- Andaian^2 juga perlu dilakukan utk memastikan skop masalah.

Cth masalah mudah : membuat kopi.
1. Memahami masalah : Kopi utk berapa orang, kopi O @ kopi susu @ kedua2nya dll.
2. Persekitaran masalah : Situasi buat kopi di pejabat, rumah & apa kelengkapan yg ada dll.
3. Andaian : Taraf gula yg diambil sama, air yg digunakan belum dimasak dll.

2.1.2 Analisa Masalah

- Mengkaji/menganalisa spesifikasi masalah dgn terperinci. 2 elemen utama yg perlu di kenal pasti:

i. input masalah

Apakah data @ maklumat yg perlu diberi ke dalam sistem aturcara utk penyelesaian masalah tersebut.
Format input juga perlu ditentukan.

cth: kopi, gula, air (format: yg belum dimasak) & krim.

2.1.2 Analisa Masalah (samb ....)

ii. output masalah

Apakah data @ maklumat yg perlu dari aturcara sebagai penyelesaian pada masalah.
Format keluaran perlu ditentukan.

cth : Format 5 cawan kopi susu @ Format 5 cawan kopi 'O' dgn 5 sudu susu dlm bekas berasingan.

ii. proses

Menganalisa apakah proses yg perlu dilalui oleh input utk menghasilkan output. Kekangan^2 & andaian^2 pd PM yg lebih terperinci dilakukan disini.

cth : formula yg digunakan, proses masuk kopi dlm cawan dll.

2.1.3 Rekabentuk

- 2 tugas utama dlm rekabentuk aturcara:
  - memilih struktur yg sesuai utk menyusun & menyimpan data dalam proses penyelesaian masalah.
  - merekabentuk **algoritma** utk memproses input & mengeluarkan output.

- Bagaimana algoritma dibuat dlm proses pembangunan aturcara ? ..... TUNGGU.....

2.1.4 Pengkodan & Pelaksanaan

- Proses melaksanakan algoritma yg direkabentuk ke dlm bahasa pengaturcaraan tertentu (dlm kelas ini bhs C).

- Format input yg telah dikenal pasti akan digunakan bersama algoritma tersebut & keluaran akan dikeluarkan mengikut format yg juga telah dikenal pasti.

- Bagaimana perlaksanaan algoritma dilakukan dlm pengaturcaraan C?.....TUNGGU.....

2.1.5 Penentusahan (verification) & Pengujian

- Utk menentukan aturcara yg dihasilkan betul.

- Pengujian aturcara perlu dilakukan dgn memberi input & memeriksa output sama ada output yg betul dikeluarkan dari aturcara yg ditulis.

- Perlu dilakukan pada semua fasa sebelum ini..... KENAPA?

  Fasa 1 & 2
  Berinteraksi pd orang yg memberi masalah (pakar) yg utk diselesaikan supaya pemahaman masalah tepat & tidak terpesong (pakar masalah kelas ini adalah SAYA).

  Fasa 3
  Pastikan rekabentuk yg terhasil memenuhi spesifikasi keperluan yg telah dihasilkan dr fasa 1 & 2.

  Fasa 4
  Pastikan aturcara selaras dgn algoritma rekabentuk & lengkap.
  Pastikan tiada kesilapan sintaks (ralat yg dikesan semasa mengkompail aturcara) & kesilapan logik (ralat masa larian) pada aturcara. AKAN DIALAMI SEMUA.....

2.1.6 Penyenggaraan

- Proses penyenggaraan perisian amat unik jika dibanding dgn produk kejuruteraan lain.
  Cthnya kereta : jika roda kereta rosak tukar pada roda yg baru.
  jika fungsi utk antara muka perisian word perlu ditukar BAGAIMANA????

- Taraf kesukaran penyenggaran sesuatu perisian bergantung kpd fass 1-5 yg telah ditempuhi oleh perisian tersebut... SAMADA PENULISNYA MENGIKUT PRINSIP KP @ TIDAK.....

2.2 Algoritma & Bagaimana Mengungkap Algoritma

- Sila sambung TUNGGU 1.1.3 disini.

- Untuk menyelesaikan masalah dengan komputer, kita perlu:

  1. Reka algoritma (fasa rekabentuk)
  2. Nyatakan algoritma sebagai aturcara dalam bahasa aturcara yang sesuai (fasa pengkodan & perlaksanaan)
  3. Laksanakan aturcara tersebut di atas komputer (fasa pengkodan & perlaksanaan)

- Algoritma - resipi bagaimana untuk menyelesaikan masalah dengan komputer. Ia menerangkan cara & langkah yang perlu dilakukan untuk menyelesaikan masalah.

- Tanpa algoritma tiada aturcara, tanpa aturcara tiada penyelesaian.

- Algoritma tidak bergantung kepada bahasa pengaturcaraan.

- Apa yg perlu ada dlm algoritma:
  - input
  - output
  - tidak mengelirukan/ringkas/mudah
  - dlm bentuk umum - boleh difahami semua golongan pembaca yg berkeperluan
  - tepat utk menyelesaikan masalah
  - terhingga - ada kesudahan
  - cekap

- Berbagai kaedah mengungkap algoritma :

Kenyataan bahasa	Carta alir	Kod Pseudo
Gambarajah N-S	Jadual aksi	Mesin keadaan terhingga

  ... dsbnya.

2.2.1 Kaedah kod pseudo

- Kod pseudo adalah jujukan langkah yg ditulis menggunakan bhs tabii yg menerangkan pemprosesan yg terlibat dlm penyelesaian masalah.

- Tujuan utama mempersembahkan penyelesaian masalah dalam bentuk yang logik dan sistematik.

- Ada beberapa cara penulisan kod pseudo digunakan, dlm kelas ini 2 jenis akan diperkenalkan:
  - kenyataan bahasa
  - kenyataan bahasa menyerupai bahasa pengaturcaraan

- Setiap langkah dlm penyelesaian masalah disususn satu-persatu secara sistematik. Setiap baris kod pseudo mewakili satu langkah yang dilaksanakan sekali shj.

i. Kaedah kod pseudo - kenyataan bahasa

- Algoritma yg ditulis dlm bahasa tabii & tidak mempunyai peraturan yg khusus.

- Dalam kaedah ini nombor yg diberikan pada jujukan langkah adalah sangat penting.

- Contoh : algoritma untuk menyediakan secawan kopi

1. masak air
2. masukkan kopi dalam cawan
3. jika perlu gula
   3.1. tambahkan gula
4. jika perlu krim
   4.1 tambahkan krim
5. masukkan air panas ke dalam cawan

- Tertib perlaksanaan langkah[2] mengikut nombor yg ditulis.

2.2.1 Kaedah kod pseudo (samb...)

- Sesuai utk masalah yg mudah krn tiada jujukan & struktur yg kompleks.

a) Struktur jujukan:

1. mula
2. langkah_1
3. langkah_2
:
n. langkah_n
n+1. tamat

b) Struktur pemilihan:

n. jika syarat
    n.1. langkah_n.1
    n.2. langkah_n.2
    :
    n.n. langkah_n.n

c) Struktur ulangan:

n. pergi ke langkah sebelum n

b) Struktur lompat:

n. pergi ke langkah selepas n

ii. Kaedah kod pseudo - kenyataan bhs menyerupai bhs pengaturcaraan

- Kod pseudo adalah pengungkapan algoritma dlm bahasa Inggeris mudah yg hampir menyerupai bahasa pengaturcaraan PASCAL - TAPI ia BELUM terikat kpd sebarang bahasa pengaturcaraan !

- Tertib perlaksanaan langkah^2 mengikut aturan langkah^2 itu ditulis.

Contoh algoritma utk memasak air:
    isi air ke dlm cerek
    nyalakan api
    masak air hingga mendidih

Tertib perlaksanaan

2.2.1 Kaedah kod pseudo (samb...)

- Tertib perlaksanaan langkah^2 boleh dikawal dgn menggunakan beberapa struktur kawalan. Kata-kunci^2 digunakan utk menjelaskan struktur^2 kawalan ini.

a) Struktur jujukan:

```
begin
    langkah_1
    langkah_2
    :
    langkah_n
end
```

b) Struktur pemilihan:

```
if  syarat then
      bhg_benar
else
      bhg_takbenar
endif
```

c) Struktur ulangan:

```
repeat
    bhg_yg_perlu diulang
until  syarat
     @
while  syarat   do
    bhg_yg_perlu diulang
endwhile
     @
repeat N kali
    bhg_yg_perlu diulang N x
```

Perhatikan:
i. syarat adalah ungkapan samada benar @ tidak
ii. bhg tertentu teks (blok) diindenkan (diberi jidar) utk menjelaskan kod pseudo
iii. Blok kadangkala dikumpulkan diantara kata kunci ***begin*** & ***end***

- Kod pseudo menyediakan secawan kopi

```
masak air
masukkan kopi dalam cawan
if  perlu_gula then
     tambahkan gula
endif
if  perlu_krim then
     tambahkan krim
endif
masukkan air panas ke dalam cawan
```

2.2.1 Kaedah kod pseudo (samb...)

- Kod pseudo menyediakan secawan kopi

```
masak air
masukkan kopi dalam cawan
if  perlu_gula
    then tambahkan gula
endif
if  perlu_krim
    then tambahkan krim
endif
masukkan air panas ke dalam cawan
```

- Penghalusan berlangkah algoritma

Perhatikan, kod pseudo di atas masih lagi dinyatakan dlm ungkapan umum. Contohnya, *masak air, masukkan kopi* dsbnya. Algoritma ini perlu dihalusi lagi utk menghapuskan kekeliruan dlm melaksanakannya.

Kod pseudo di atas adalah contoh bagaimana fasa pertama algoritma ditulis, bermula dari langkah^2 yg agak umum. Dlm fasa^2 yg seterusnya, langkah^2 umum ini dihalusi sedikit demi sedikit sehingga menghasilkan langkah^2 yg lebih spesifik.

Teknik merekabentuk algoritma sebegini dipanggil **teknik Atas ke Bawah** (Top-down). Proses menghalusi algoritma sedikit demi sedikit dipanggil **Penghalusan berlangkah**.

Contoh: Penghalusan berlangkah ungkapan *masak air*

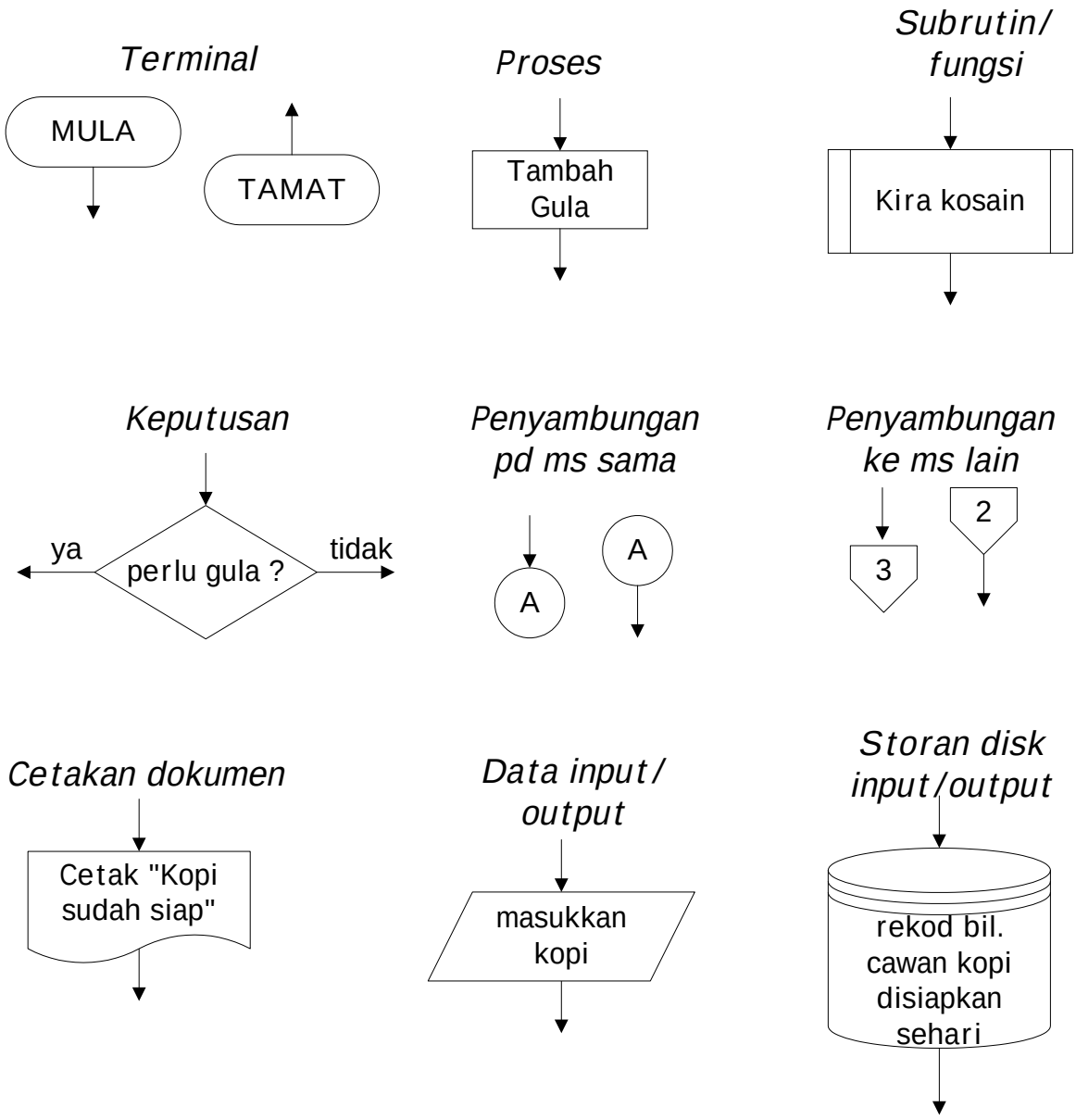
masak air ⟶

```
isi air ke dlm cerek
nyalakan api
repeat
    tunggu
until  air_mendidih
```

2.2.2 Kaedah carta alir

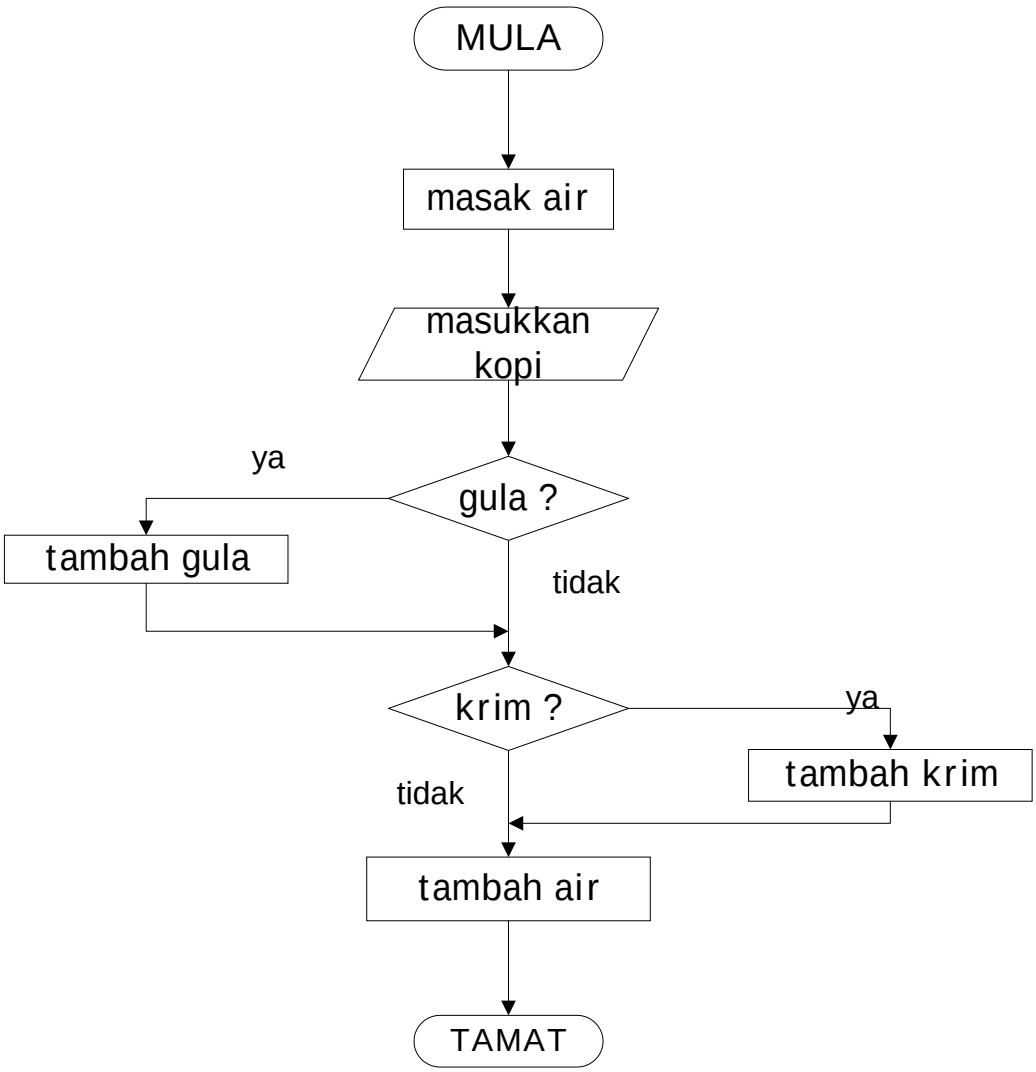
- Carta alir terdiri dari kotak-kotak berbagai bentuk yang menunjukkan operasi yg perlu dilakukan dan anak-panah[2] yg menyatakan jujukan operasi tersebut.

- Simbol carta alir:



2.2.2 Kaedah carta alir (samb...)

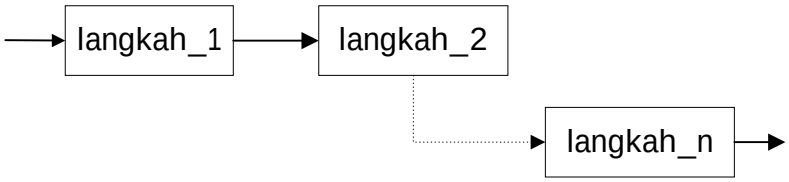
- Carta alir menyediakan secawan kopi



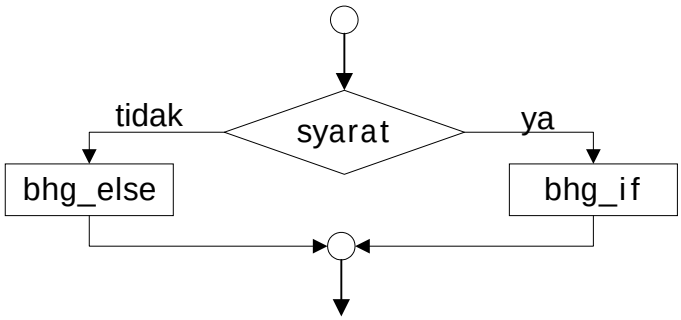
2.3 Carta Alir lwn. Kod Pseudo

Kod Pseudo	Carta Alir

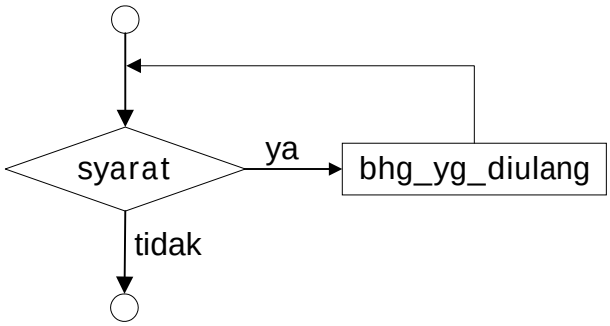
begin
langkah_1
langkah_2
. . . .
. . . .
langkah_n
end



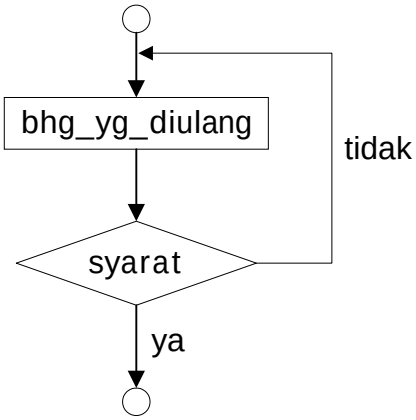
if syarat **then**
bhg_if
else
bhg_else
endif



while syarat **do**
bhg_yg_perlu diulang
endwhile



repeat
bhg_yg_perlu diulang
until syarat



2.4 Contoh PM & Algoritma

Tulis aturcara yg akan menerima input sepasang nombor & mencetak mesej "SAMA" jika dua nombor tersebut mempunyai nilai yg sama, mesej "NOMBOR PERTAMA LEBIH BESAR" jika nombor pertama lebih besar dpd nombor kedua & mesej "NOMBOR KEDUA LEBIH BESAR" jika nombor kedua lebih besar dpd nombor pertama.

Fasa: Spesifikasi keperluan

- Fasa : Analisa Masalah

  Input:
  1. media : papan kekunci
  2. data :
     nombor pertama (nom1) &
     nombor kedua (nom2)

  Output:
  1. media : skrin
  2. data :
     "SAMA" @
     "NOMBOR PERTAMA LEBIH BESAR" @
     "NOMBOR KEDUA LEBIH BESAR"

  Proses:
  1. bandingkan 2 nonbor & tentukan output

- Fasa : Rakabentuk

  i. Kod pseudo I

  1. Mula
  2. Baca nom1, nom2
  3. Jika nom1=nom2
     3.1. cetak mesej "SAMA"
     3.2. pergi ke langkah 6
  4. Jika nom1>nom2
     4.1. cetak mesej "NOMBOR PERTAMA LEBIH BESAR"
     4.2. pergi ke langkah 6
  5. Jika nom1<nom2
     5.1. cetak mesej "NOMBOR KEDUA LEBIH BESAR"
     5.2. pergi ke langkah 6
  6. Tamat

no. adalah penting!!

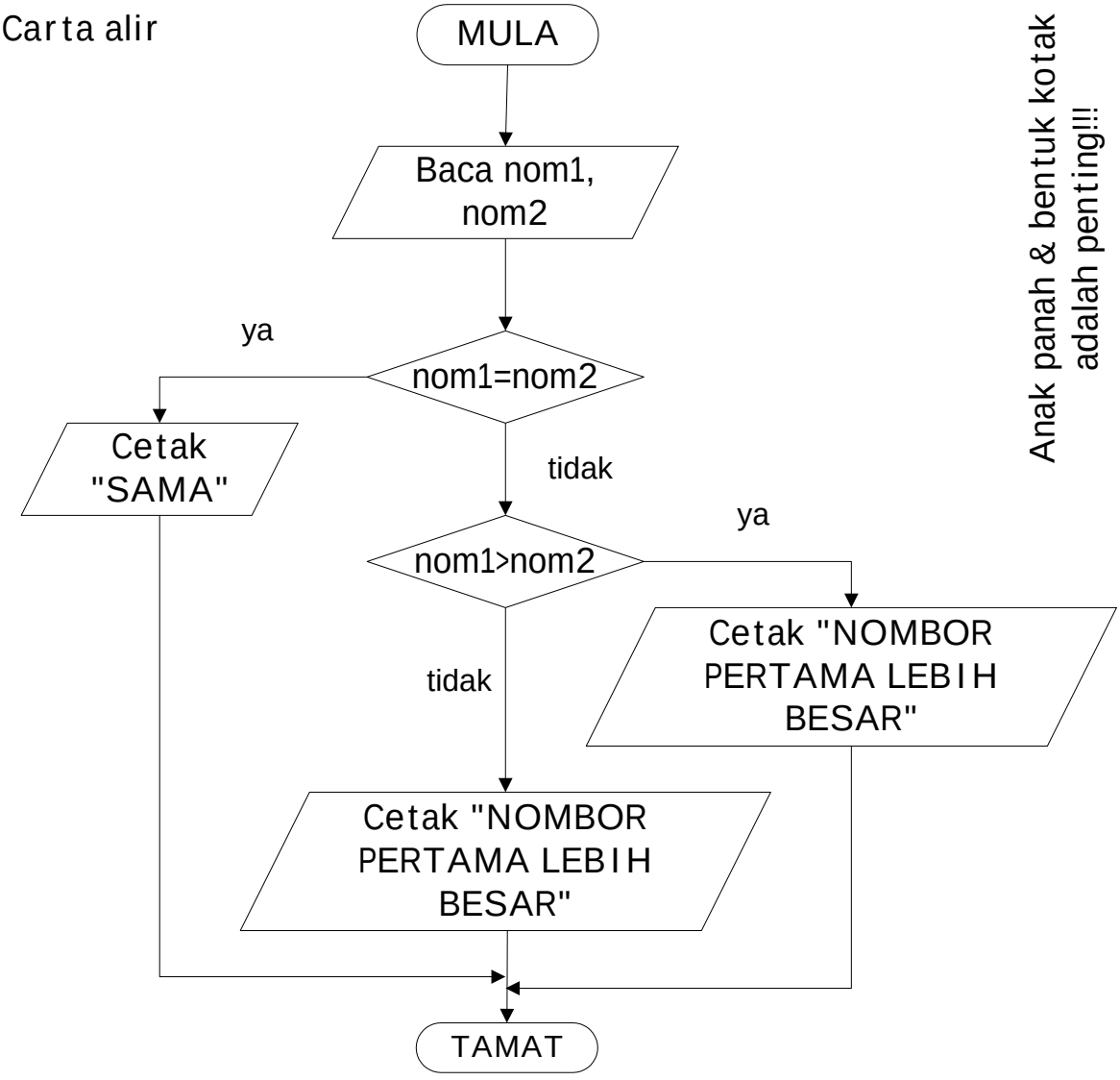
2.4 Contoh PM & Algoritma (samb....)

ii. Kod pseudo I I

```
begin
   Baca nom1, nom2
   if nom1=nom2 then
      cetak mesej "SAMA"
   else
      if nom1>nom2
         cetak mesej "NOMBOR PERTAMA LEBIH BESAR"
      else
         cetak mesej "NOMBOR KEDUA LEBIH BESAR"
      endif
   endif
end
```

Esotan (indent) & penggunaan kata-kunci adalah penting!!!

iii. Carta alir



Anak panah & bentuk kotak adalah penting!!!

2.5 Proses Menyediakan Aturcara

- Langkah[2] penyelesaian masalah menggunakan disiplin KP telah difahami dari bhg 2.1.
  Bagaimana hendak menyerapkan disiplin tersebut dalam proses penyedian aturcara.

- Kajian kes
  Walaupun contoh kajian kes ini mudah, ttp proses ini boleh digunakan utk masalah mudah atau rumit.

Langkah 1 : Definisi Masalah @ Spesifikasi Keperluan Masalah

Daripada jejari yg diberikan kira luas bulatan & ukurlilit bulatan tersebut.

Langkah 2 : Analisa Masalah

Input:
1. media: papan kekunci
2. data:
   radius - dimasukkan oleh pengguna
   π = 3.14159 pemalar

Output
1. media: skrin
2. data:
   luas bulatan
   ukurlilit bulatan

Proses:
formula 1: luas bulatan = πj^2
formula 2: ukurlilit bulatan = $2\pi j$
j-jejari bulatan

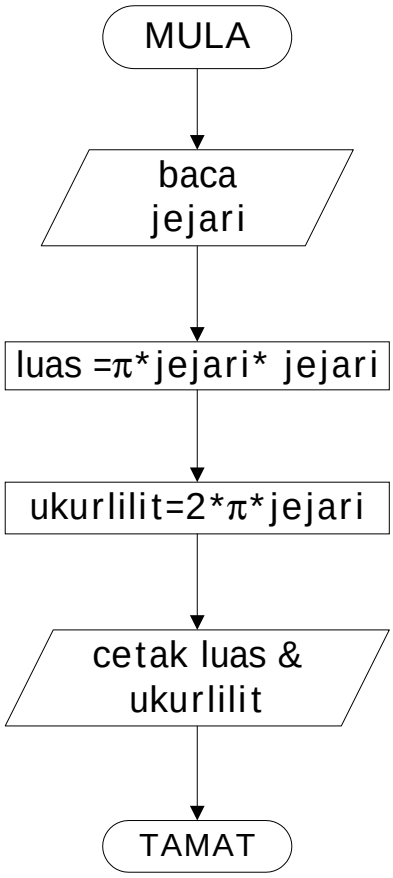
2.5 Proses Menyediakan Aturcara (samb ....)

Langkah 3 : Rekabentuk

Langkah 3.1
Mulakan rekabentuk awal dgn menggunakan kaedah kod pseudo kenyataan bahasa:

1. Baca jejari bulatan
2. Kira luas bulatan
3. Kira ukurlilit bulatan
4. Cetak luas dan ukurlilit bulatan

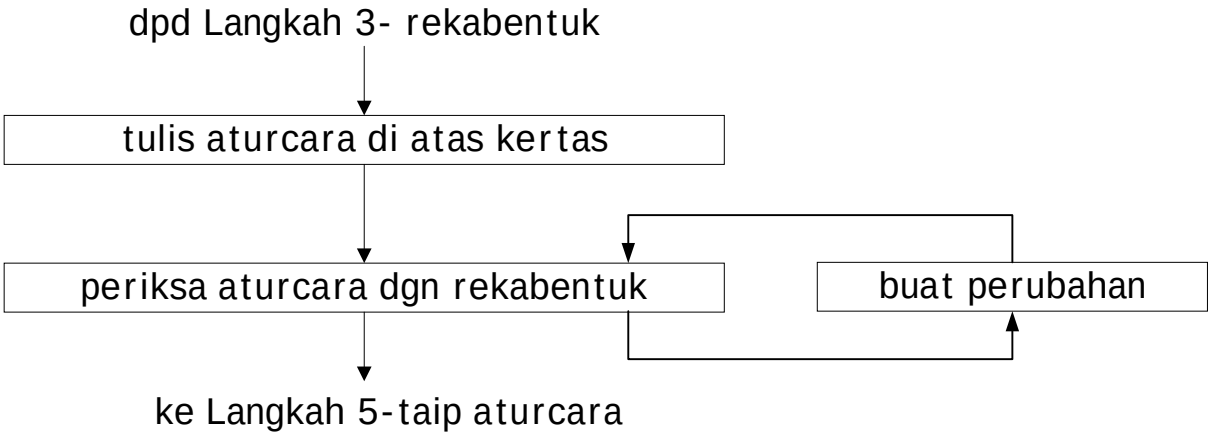
Langkah 3.2
Algoritma ini perlu dihalusi & dilukis menggunakan carta alir.



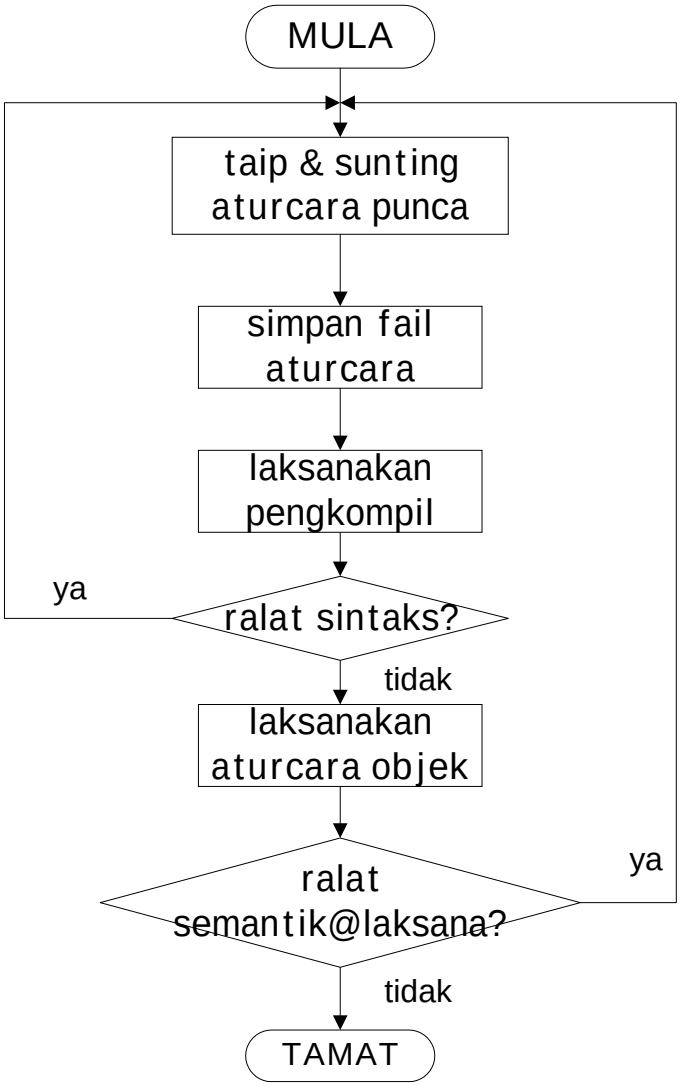
Langkah 3.3
Jika algoritma melibatkan struktur ulangan & pemilihan sila tulis kod pseudo menggunakan kenyataan bhs menyerupai bhs pengaturcaraan.

2.5 Proses Menyediakan Aturcara (samb ....)

Langkah 4 : Tulis Kod
Cadangan strategi penulisan kod :



Langkah 5 & **Langkah 6**: Taip Aturcara & Uji Aturcara
Carta alir taip aturcara & uji aturcara:



guna penyunting fail EDIT, WORD dsbnya

contoh : FAILSAYA.C

contoh : Turbo C, Borland C, Microsoft C dsbnya

C> FAILSAYA.EXE

Latihan:

1. Soalan ulangkaji.

a) Senaraikan fasa/langkah pembangunan perisian.

b) Algoritma bergantung kpd bhs pengaturcaraan. betul @ salah.

c) Fasa perlaksanaan bergantung kpd bhs pengaturcaraan. betul @ salah.

d) Senaraikan 4 kaedah pengungkapan algoritma.

2. Nyatakan pada fasa mana proses^2 berikut dilakukan:

a) Penyataan masalah diperolehi dari pengguna.

b) Kos pseudo ditulis untuk mengira luas segi tiga.

c) Setelah algoritma siap dibina, data contoh digunakan dgn memasukkan data ke dalam algoritma & memastikan output dari algoritma tersebut memberi nilai yg betul.

d) Mengenal pasti input sistem.

e) Menulis arahan^2 bahasa C utk PM.

f) Memastikan ralat masa-larian & ralat sintaks tiada pada kod.

g) Menulis dokumen^2 yg perlu dlm proses PM.

Latihan (samb...)

3. Rekakan algoritma dlm bentuk kod pseudo dan carta alir bagi menyelesaikan masalah[2] berikut:

   a) menentukan samada satu integer itu genap atau ganjil.

   b) mencari nilai rintangan setara bagi 4 perintang yg disambungkan selari.

   c) mencari nilai rintangan setara bagi M perintang yg disambungkan selari.

   d) mencari punca bagi persamaan kuadratic $Ax^2+Bx+C=0$.

4. Tuliskan hasil algoritma berikut dlm bentuk kod pseudo dan carta alir selepas penghalusan berlangkah dilakukan :

   a)
   ```
   if perlu_gula
       then tambahkan gula
   else
       jangan tambah gula
   endif
   ```

   b)
   ```
   isi air ke dlm cerek
   nyalakan api
   repeat
       tunggu
   until air_mendidih
   ```