



# Flowchart

---

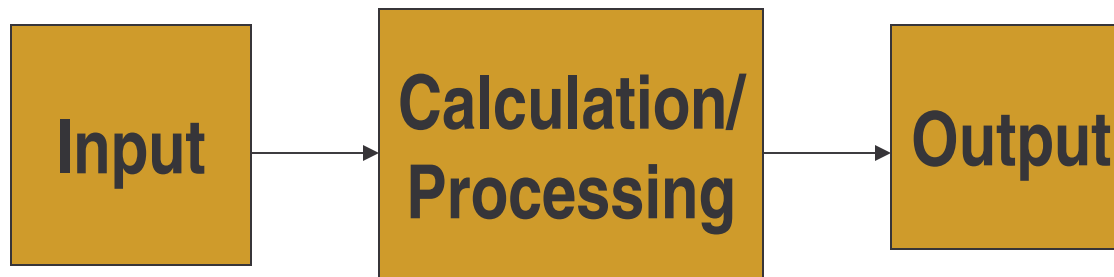
# Membuat sebuah program

- Defining the problem
    - Mendefinisikan apa yang akan dilakukan atau yang dihasilkan oleh program
  - Planning
    - Mendefinisikan langkah-langkah, membuat flowchart
  - Programming
    - Menuliskan program, termasuk perbaikan eror dan testing
  - Documentation
    - Menambahkan komentar dan “membukukan” program
  - Implementation
    - Pengembangan, membuat program semakin baik
-

---

Sebuah program dapat dibagi menjadi 3 bagian:

- ❑ Input data
- ❑ Processing data – Calculation, etc.
- ❑ Output information/results



# Flowchart

---

Suatu diagram yang menggunakan simbol-simbol khusus yang menunjukkan langkah-langkah sebuah program

---

---

# Apakah flowchart?

- Sebuah diagram yang dibangun oleh berbagai macam bentuk simbol yang saling terhubung dengan garis-garis beranak panah.
  - Simbol-simbol tersebut mengindikasikan aksi apa yang akan dilakukan atau keputusan apa yang akan dibuat.
  - Bisa juga digunakan untuk menggambarkan proses yang kompleks ketika terjadi gabungan antara aksi dan pengambilan keputusan.
  - Flowchart membantu dalam planning & programming.
-

---

## Flowchart digunakan untuk menggambarkan:

- Langkah-langkah dalam suatu proses
  - Keputusan yang dibuat dalam proses tersebut
  - Data penting yang digunakan dalam langkah-langkah tersebut, jika data tersebut penting
-

# Flowchart Symbols



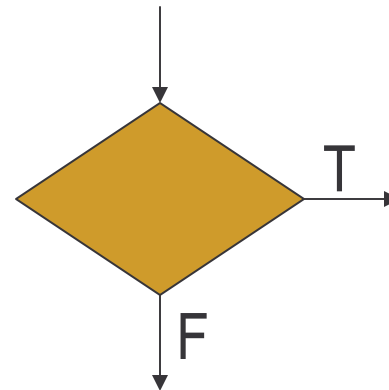
**Terminal symbol**  
(Start & Stop)



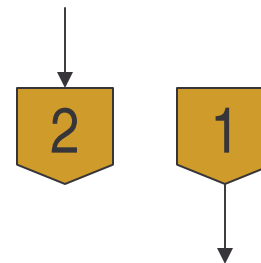
**Processing symbol**  
(Calculate, Assign, Initialize, Declare)



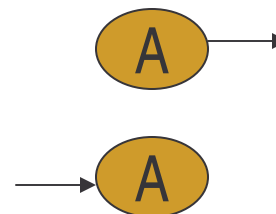
**Input/Output symbol**  
(get, put)



**Decision symbol**  
(Selection, Conditional Statement)



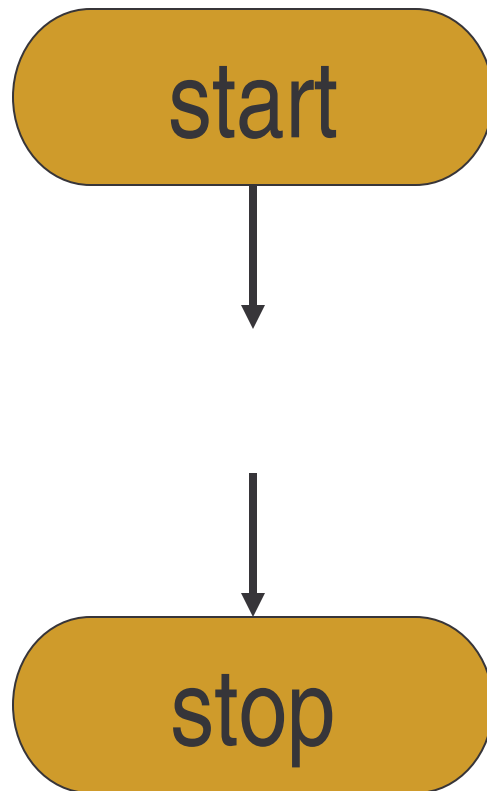
**Page connector**  
(used when run out of room at the bottom)



**Step connector**  
(go from one part to another)

---

# Terminal Symbol

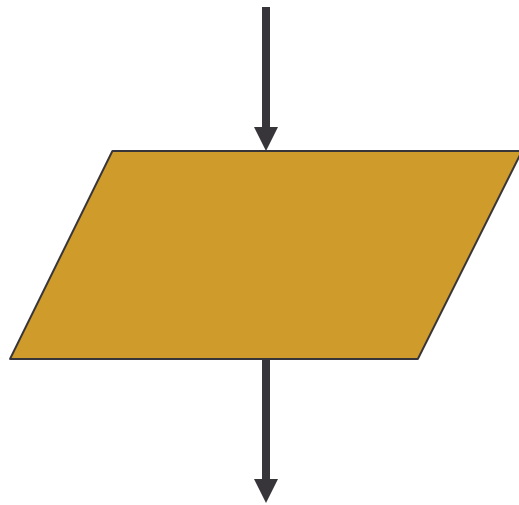


- Only one “start” & one “stop”
  - 1 arrow only
  - Each flowchart starts with one “start” & ends with one “stop”
-



---

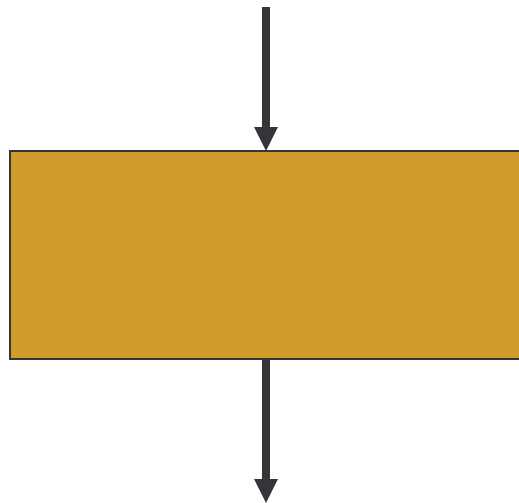
# Input / Output Symbol



- 2 arrows only
  - 1 in & 1 out
- Input Data
  - `get fullname:*`
- Display result
  - `put fullname`

---

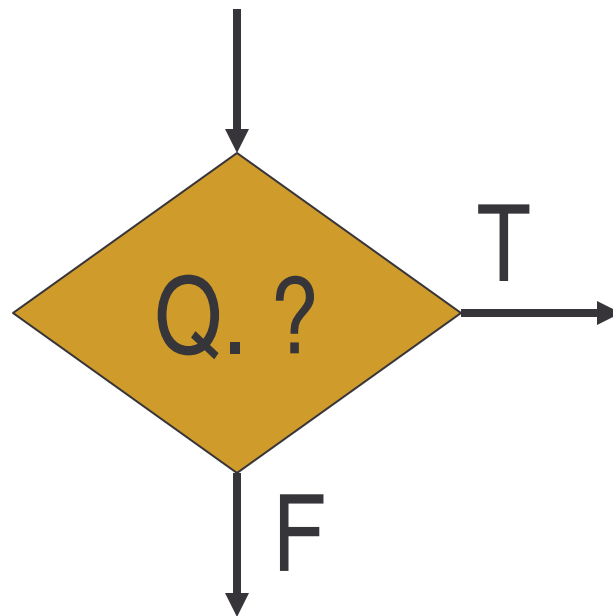
# Processing Symbol



- 2 arrows only
    - 1 in & 1 out
  - Declaration
    - `var sum : real`
  - Initialization
    - `sum := 30.14`
  - Calculation
-

---

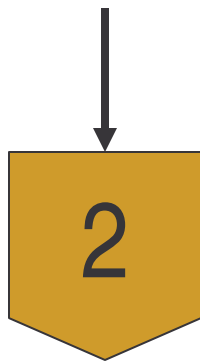
# Decision Symbol



- 3 arrows
    - 1 in & 2 out
  - Question
    - Gunakan kalimat tanya yang dapat dijawab dengan **Yes** atau **No**
  - 2 out going arrows
    - T (True, Yes)
    - F (False, No)
-

---

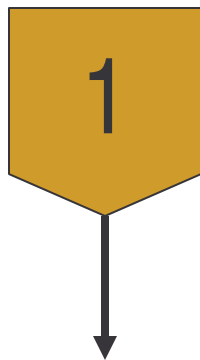
# Page connector



- 1 arrow only



- ☀ Shows the next page

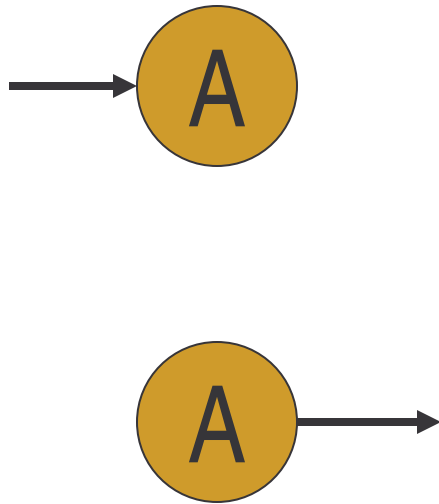


- ☀ Shows the previous page

- ☀ Run out of space
-

---

# Step connector



- 1 arrow only
  - Appear in
    - 1 to 1
    - Many to 1
  - Use letters
  - Go from one step to another
-

---

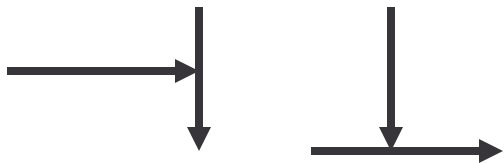
# Arrows



- From Top to Bottom & from Left to Right

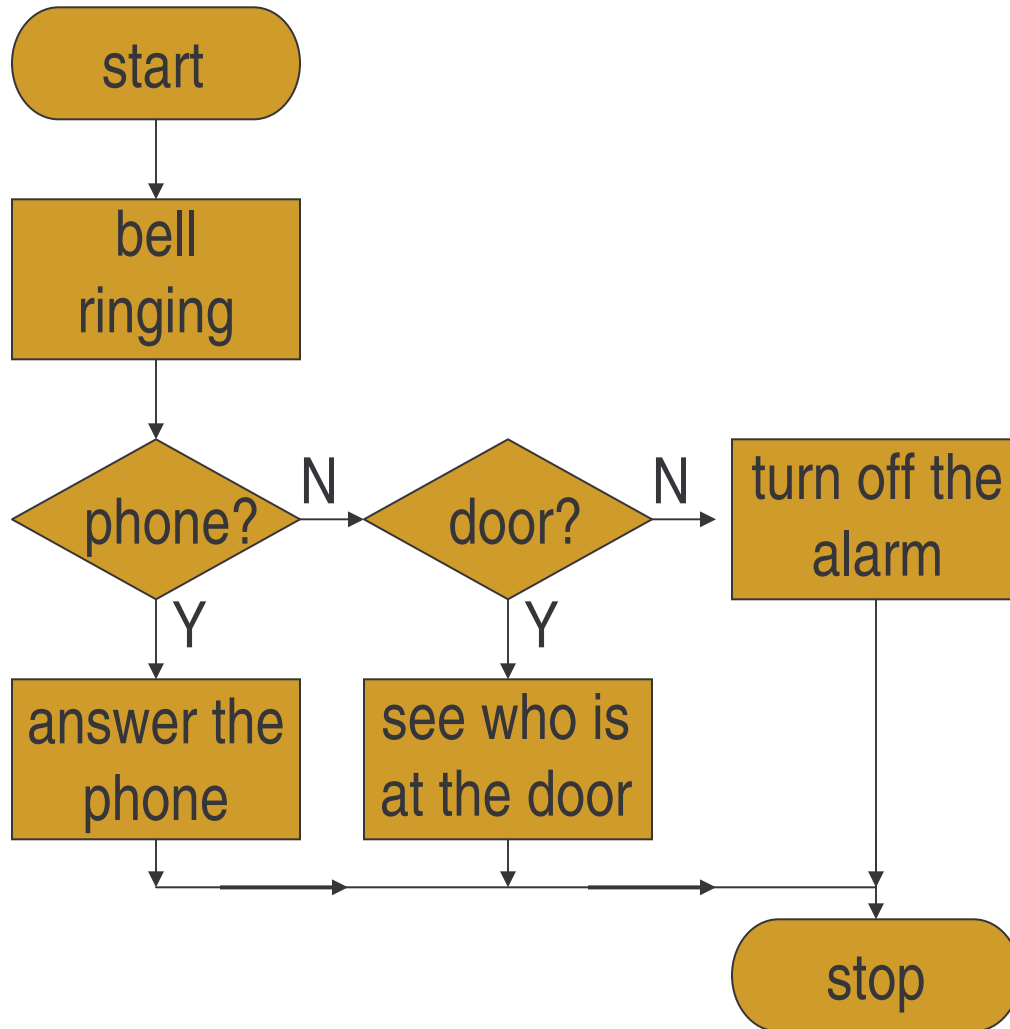


- Make turn



- Join together
-

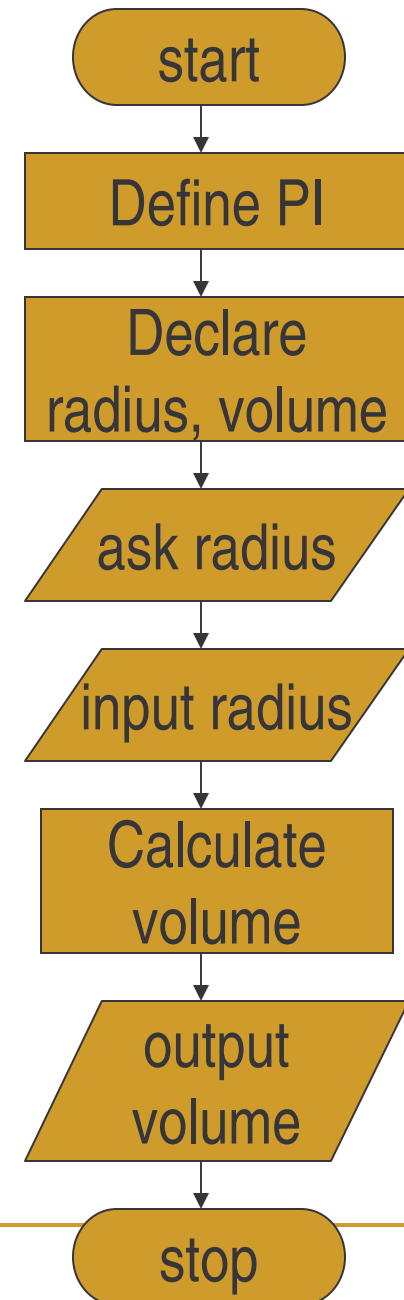
# Latihan 1



- 1.is it the door bell?
- 2.start
- 3.answer the phone
- 4.turn off the alarm
- 5.is it the telephone?
- 6.listen to bell ringing
- 7.stop
- 8.see who is at the door

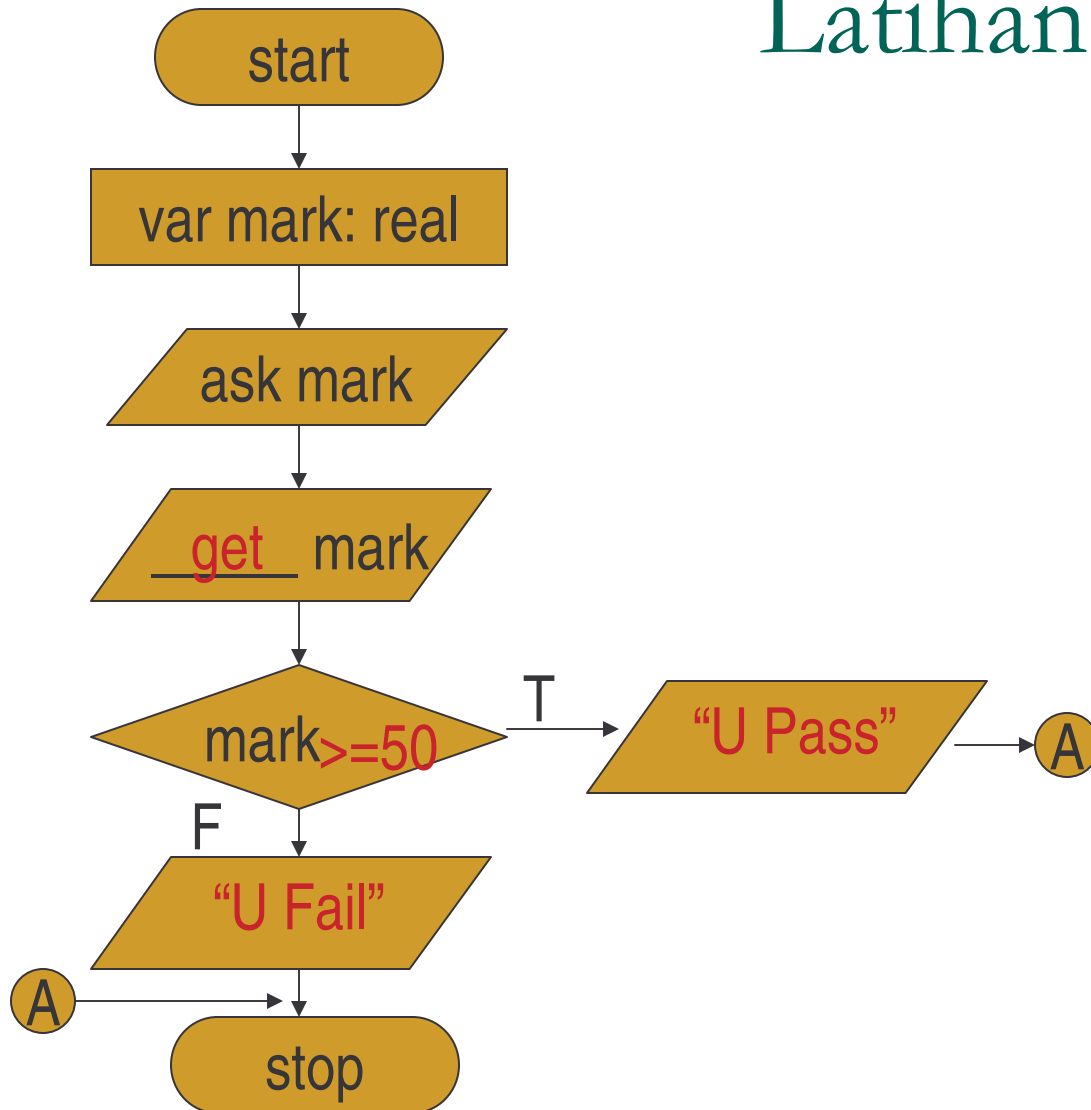
## Latihan 2

```
const PI := 3.14
var radius, volume : real
put "please enter..."..
get radius
volume := (4/3)*PI*radius**3
put ""
put "The volume is", volume
```



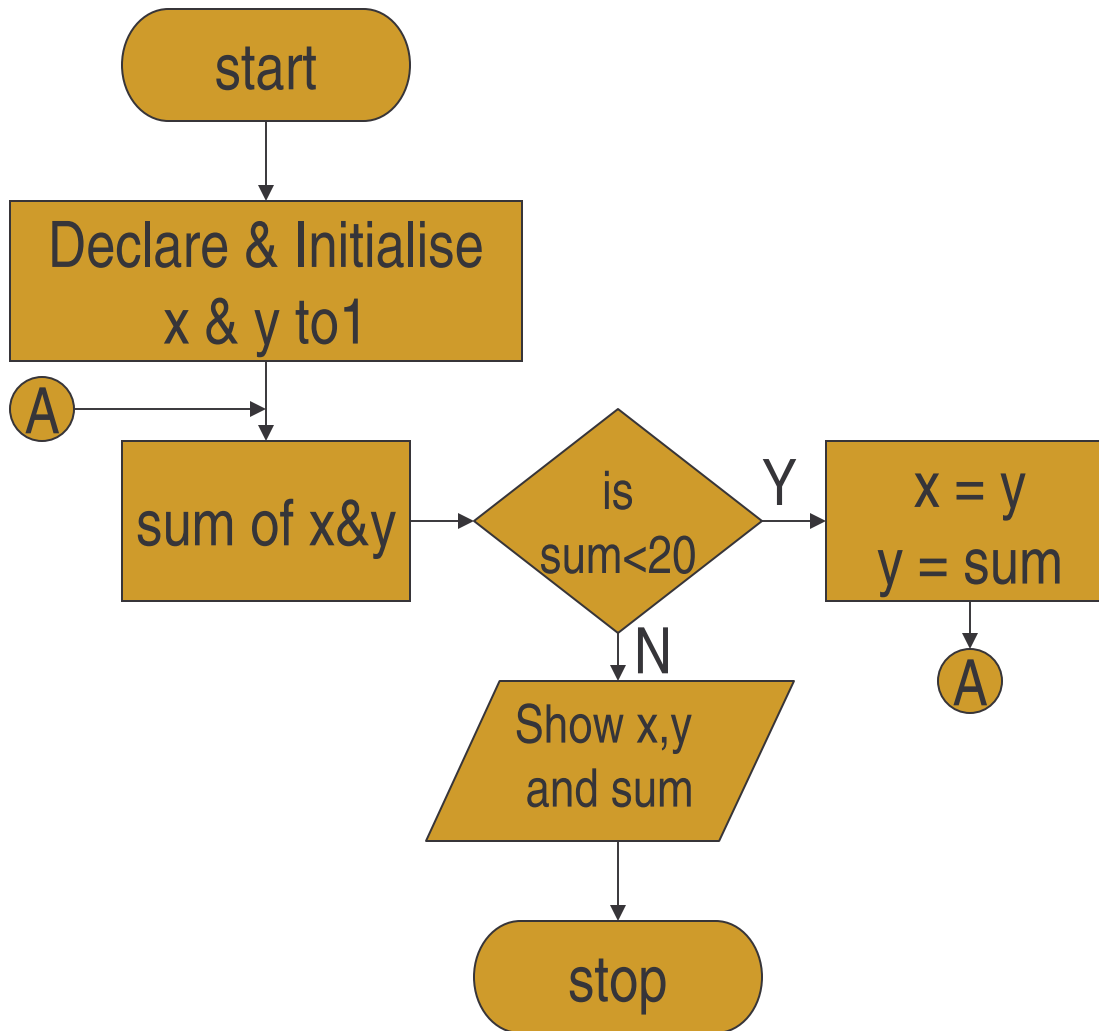


# Latihan 3



- 1.start declare variables
- 2.what is the mark
- 3.ask for mark
- 4.check  $\text{mark} \geq 50$
5. $\geq 50$  show "U Pass"
6. $< 50$  show "U Fail"
- 7.stop

# Latihan 4 dengan Trace Table



**Trace Table**

x	y	sum
1	1	2
1	2	3
2	3	5
3	5	8
5	8	13
8	13	21

---

# Programming

- Merubah langkah-langkah ke instruksi-instruksi / bahasa pemrograman
  - Debug – Mengoreksi eror/kesalahan
    - Kesalahan sintak
      - Kesalahan grammar yang mudah ditemukan
    - Kesalahan logical
      - Program berjalan & bekerja tetapi menghasilkan output yang salah atau tidak sesuai harapan.
      - Dapat ditemukan melalui testing
-

---

# Documentation

- Menambahkan komentar
    - Lebih mudah untuk dimengerti
  - Menggunakan “spaces” dengan bijak
    - Blanks, Tabs, New Lines
    - Easy to trace, follow & understand
  - Pemilihan nama variabel & konstanta
    - Mudah dimengerti, tidak terlalu panjang dan tidak terlalu pendek
-

---

# Implementation

- ❖ Meminta orang lain untuk menjalankan program
  - ❖ Menjadikan lebih user friendly termasuk input & output
  - ❖ Menjadikan lebih efisien (lebih sedikit baris)
  - ❖ Menjadikan waktu eksekusi cepat
  - ❖ Menambahkan tampilan akhir, dll.
-

---

# Ingat

- Sebuah flowchart adalah gambaran suatu proses.
  - Pemilihan model dan kedalaman/detail tergantung pada tujuan flowchart dibuat.
  - Setiap orang yang terlibat dalam proses harus ikut berpartisipasi dalam pembuatan flowchart dan menyetujuinya.
  - Flowchart bersifat dinamis, bisa diubah bila proses/masalah juga berubah.
-

---

**Good Luck!**

---