

# Bagaimana Menyusun Flowchart ?

Oleh

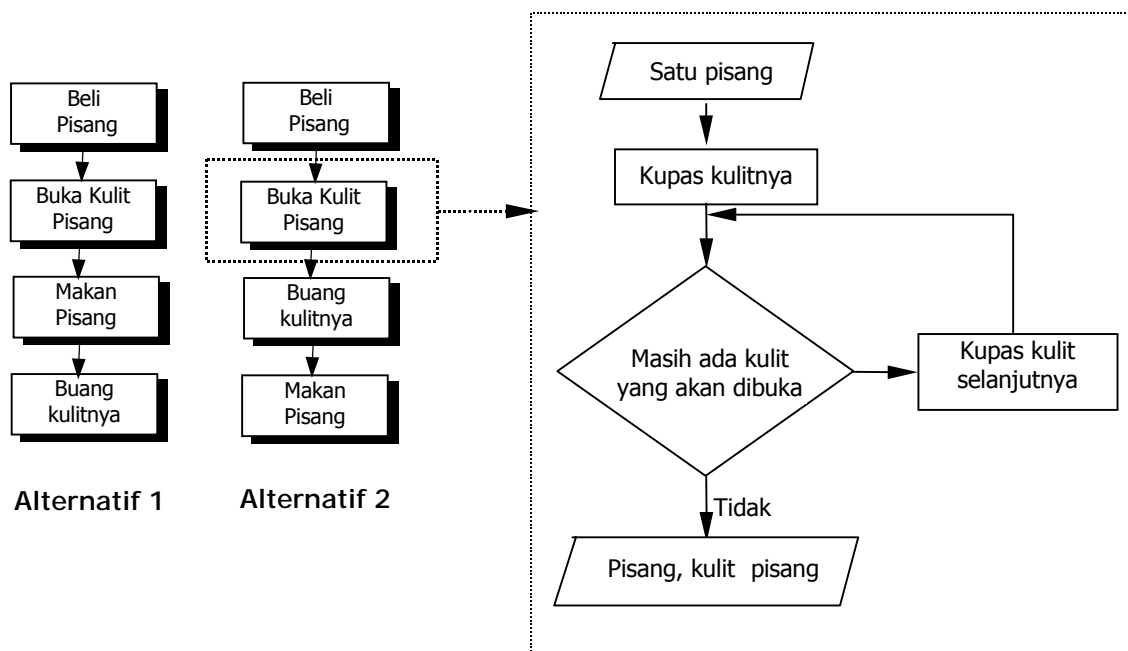
Assisten Praktikum Komputer BPPS HMSFT UNDIP

<http://www.bpps.ourfamily.com>

## Pendahuluan

Penguasaan bagaimana menyusun flowchart atau bagan alir yang sistematis, terarah dan efisien sangat bermanfaat dalam membuat sebuah program. Selain sebagai sarana penyampaian 'bagaimana program tersebut bekerja' kepada orang lain, flowchart biasanya juga sebagai alat bantu pencatatan perkembangan terakhir sebuah program yang akan berguna dalam proses selanjutnya yakni evaluasi, koreksi kesalahan dan pengembangan program.

Flowchart digambarkan dalam bentuk logika urutan langkah yang berhubungan antar satu langkah dengan langkah lainnya berdasarkan runtut waktu. Contoh sederhana cara makan satu pisang yang diterjemahkan dalam sebuah flowchart,



Gambar 1.1 Contoh sederhana bagaimana detail flowchart dibuat

## Input, Proses dan Output

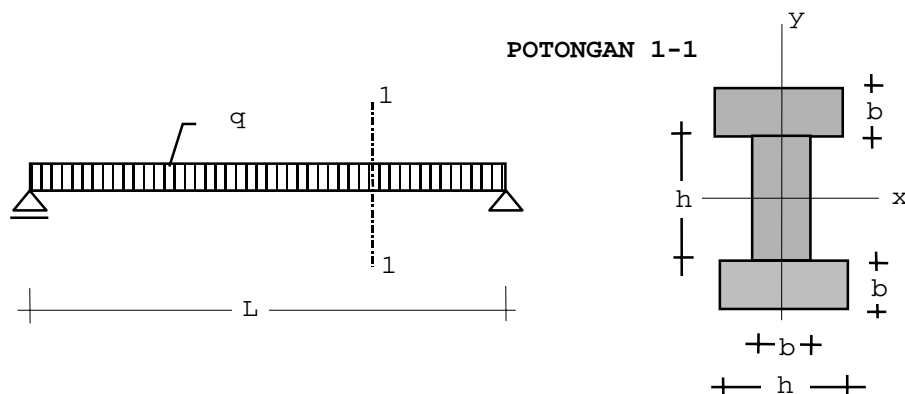
Dalam pembuatan flowchart ada 3 bagian utama yang tidak terpisahkan yang harus didefinisikan terlebih dahulu yakni input, proses dan output. Dalam pendefinisian output-proses-input harus mengacu pada kemudahan user sebagai pengguna akhir dan disarankan untuk tidak perlu terlalu mendetail misalnya bagaimana nanti user interface-nya.



Gambar 1.2 Tiga bagian utama dari flowchart yang tidak terpisahkan

Untuk memperjelas bagaimana cara mendefinisikan output-proses-input dalam sebuah kasus perhitungan, dibawah ini disajikan contoh untuk aplikasi perhitungan gelagar bekisting.

Sebuah gelagar kayu tersusun untuk bekisting bentang  $L$ , dimodelkan seperti gambar dibawah dimuati oleh beban merata sebesar  $q$ . Jika diketahui tegangan  $\sigma_t$  dan  $\tau$  berapakah dimensi balok tersusun yang memenuhi persyaratan kekuatan dan kekakuan ?



Metode yang dipergunakan dalam perhitungan dimensi biasanya adalah metode trial and error begitu juga dalam kasus ini. Mengingat ada dua persyaratan yang harus dipenuhi yakni kekakuan dan kekuatan serta kesulitan yang dihadapi dalam penentuan output dalam bentuk nilai yakni  $b$  dan  $h$  karena dimesi kayu dipasaran sangat bervariasi, maka alternatif

solusinya adalah user mencoba sendiri nilai b dan h dan program akan mengecek apakah persyaratan telah dipenuhi. Pendefinisian dimulai dari output kemudian proses dan yang terakhir adalah input yang diperlukan dalam sebuah flowchart.

DEFINISI	
OUTPUT	Apakah balok penyusun memenuhi persyaratan kekuatan dan kekakuan ? Jawaban Ya/Tidak
PROSES	<p>Cari momen dan gaya geser yang bekerja</p> $M = 1/8. q. L^2$ $V = 1/2 q L$ <p>Cari data untuk balok tersusun</p> $A_{geser} = b.h$ $I_x = 1/6.h.b^3 + 1/12.b.h^3 + 2.b.h.(0,5h+0,5b)^2$ $\bar{y} = b + 0,5h$ $W_x = I_x / \bar{y}$ <p>Chek kekuatan profil</p> <p><u>Chek tegangan lentur balok tersusun</u></p> $\sigma = \frac{M}{W_x} \leq \bar{\sigma}t$ <p><u>Chek tegangan geser balok tersusun</u></p> $\tau = \frac{V}{A_{geser}} \leq \bar{\tau}$ <p>Chek kekakuan</p> <p>Asumsi persyaratan lendutan <math>\delta_{max} = 1/250 L</math></p> $\delta = \frac{5}{384} \frac{qL^4}{EI_x} \leq \delta_{max}$
INPUT	<p>Data input :</p> <p>lebar balok (b), tinggi balok (h), modulus elastisitas (E), beban merata (q), bentang gelagar (L), tegangan lentur ijin (<math>\bar{\sigma}t</math>) dan tegangan geser ijin (<math>\bar{\tau}</math>).</p>

Setelah output-proses-input terdefinisi, langkah selanjutnya dicek kembali dengan mencoba mengisi input, menjalankan perhitungan dengan proses yang telah didefinisikan serta mengecek apakah input yang diperlukan

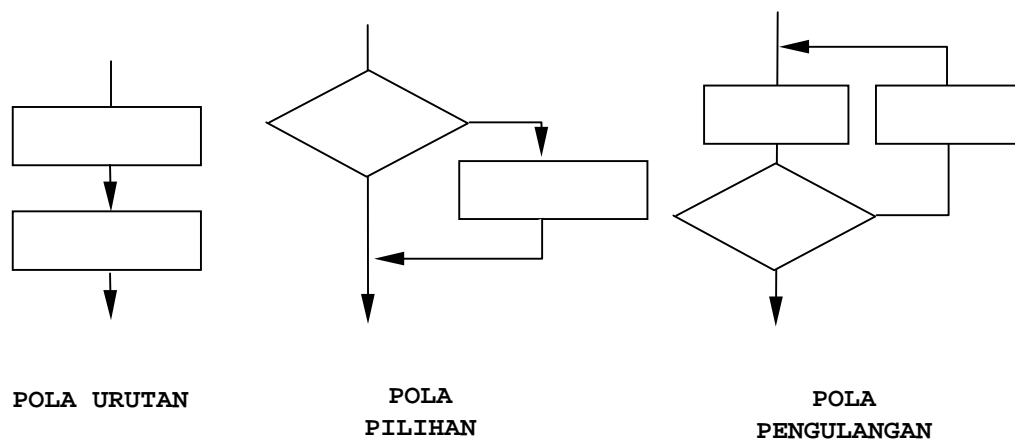
untuk menghasilkan output telah memadai. Setelah melakukan pengecekan, langkah selanjutnya menerjemahkan input-proses-output kedalam sebuah flowchart. Untuk memperjelas bagaimana membuat flowchart ada baiknya dibahas mengenai tiga pola dasar blok flowchart dan logika-logika yang sering digunakan.

### Tiga pola dasar dan simbol dalam flowchart

Secara normal alur kerja dari sebuah program bergerak dari atas ke bawah, namun pergerakan ini dapat dirubah atau dikontrol dengan tiga pola dasar flowchart yang sering digunakan yakni,

1. Pola urutan (sequence)
2. Pola pilihan (selection/multiple condition)
3. Pola pengulangan (iteration/looping)

Pola-pola tersebut jika digambarkan dalam sebuah flowchart dapat dilihat pada gambar 1.3,



Gambar 1.3 Pola-pola dasar sebuah blok flowchart.

- **POLA URUTAN**

Pola urutan memiliki pengertian bahwa seluruh proses dalam suatu block flowchart dilakukan hanya sekali berdasarkan urutan kerja.

- **POLA PILIHAN**

Dengan pola pilihan, suatu blok flowchart dapat direkayasa untuk memiliki alternatif alur kerja sesuai kondisi yang telah ditetapkan. Operator logika yang biasa digunakan untuk

mengekspresikan satu atau lebih kondisi sehingga menghasilkan data logika (boolean) baru adalah **and** dan **or**.

Tabel hasil ekspresi logika **And**

Ekspresi	Hasil
False and False	False
False and True	False
True and False	False
True and True	True

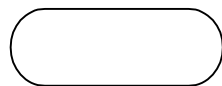
Tabel hasil ekspresi logika **Or**

Ekspresi	Hasil
False or False	False
False or True	True
True or False	True
True or True	True

- **POLA PENGULANGAN**

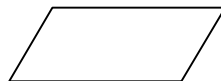
Dengan pola pengulangan, sebuah block flowchart dapat direkayasa untuk melaksanakan perintah secara berulang secara terus menerus hingga suatu batas kondisi yang telah ditetapkan tercapai.

Ketiga pola dasar yang membentuk blok-blok flowchart dapat dikombinasikan menjadi sebuah kesatuan flowchart, adapun simbol-simbol yang sering digunakan dalam flowchart yaitu,



**Terminator**

digunakan untuk memulai/mengakhiri *flowchart*



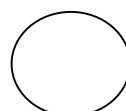
**Input/Output Data**

untuk membaca data dan menulis output



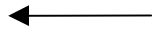
**Process**

simbol yang menyatakan proses pengolahan data



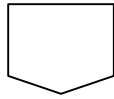
**Connector**

simbol untuk menghubungkan flowchart pada halaman yang sama.



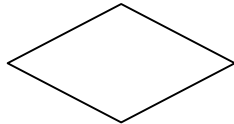
#### Arrow

untuk menentukan arah dari sebuah proses/logika



#### OFF Page Connector

untuk menghubungkan flowchart pada halaman yang berbeda.



#### Decision

untuk menentukan pilihan kondisi yang memenuhi syarat.

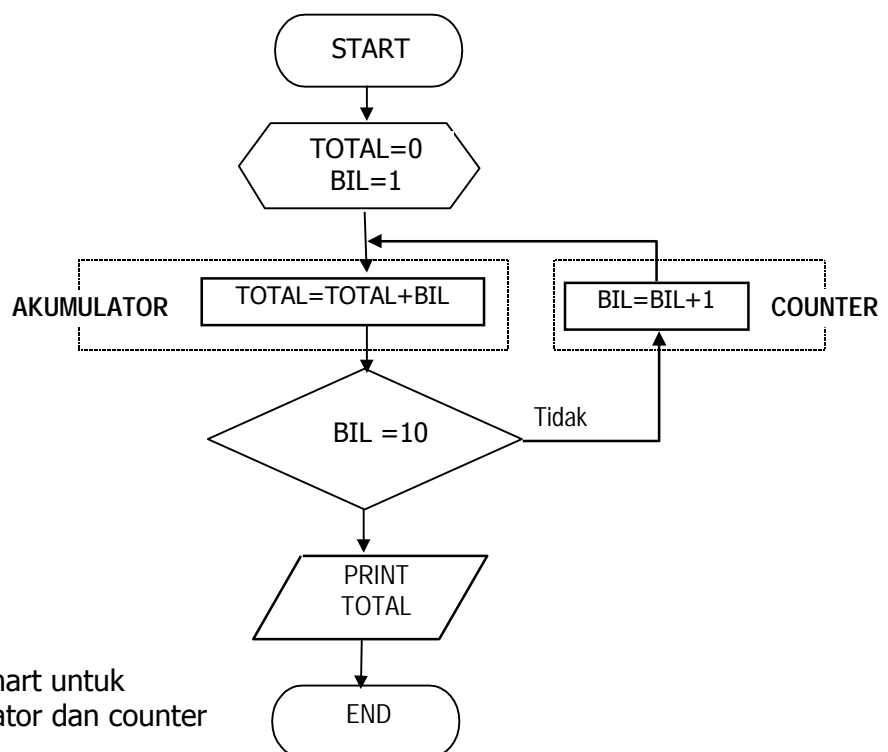


#### Preparation

untuk menentukan nilai awal sebuah variabel

### Logika Akumulator dan Counter

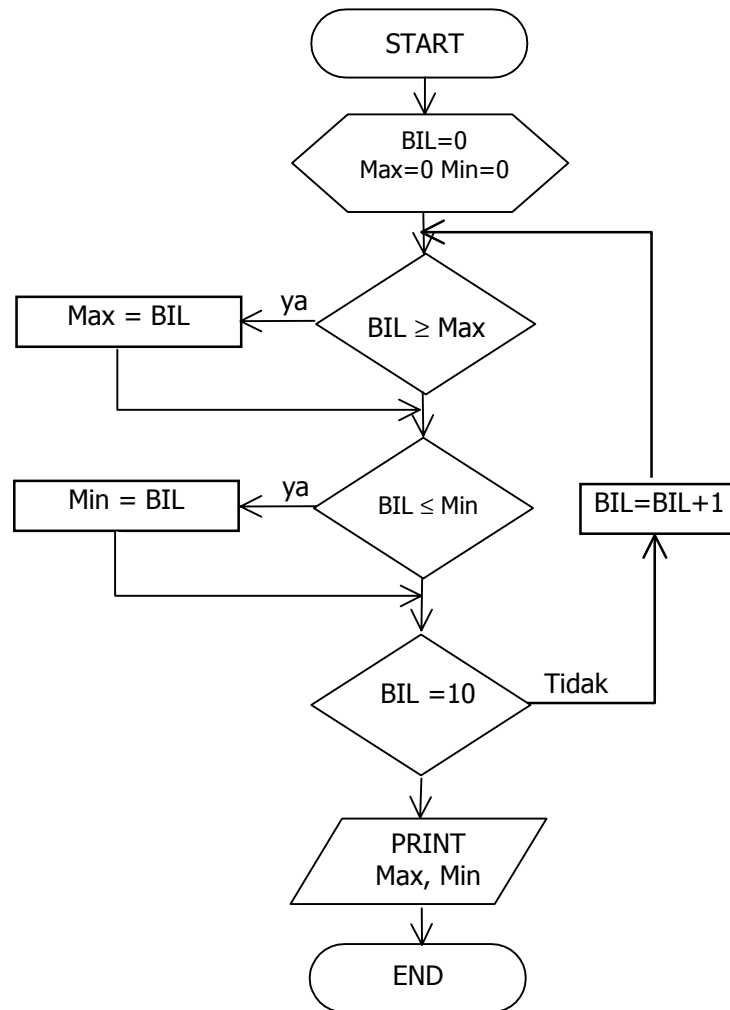
Logika akumulator digunakan jika ditentukan variabel mengakumulasi nilai yang berasal dari variabel lain. Sedangkan counter digunakan jika variabel mengakumulasi nilai berasal dari suatu konstanta. Contoh penggunaannya pada penjumlahan dari bilangan 1 sampai 10 yang akan menghasilkan nilai 55.



**Gambar 1.4**  
Contoh flowchart untuk  
logika akumulator dan counter

## Logika Switching

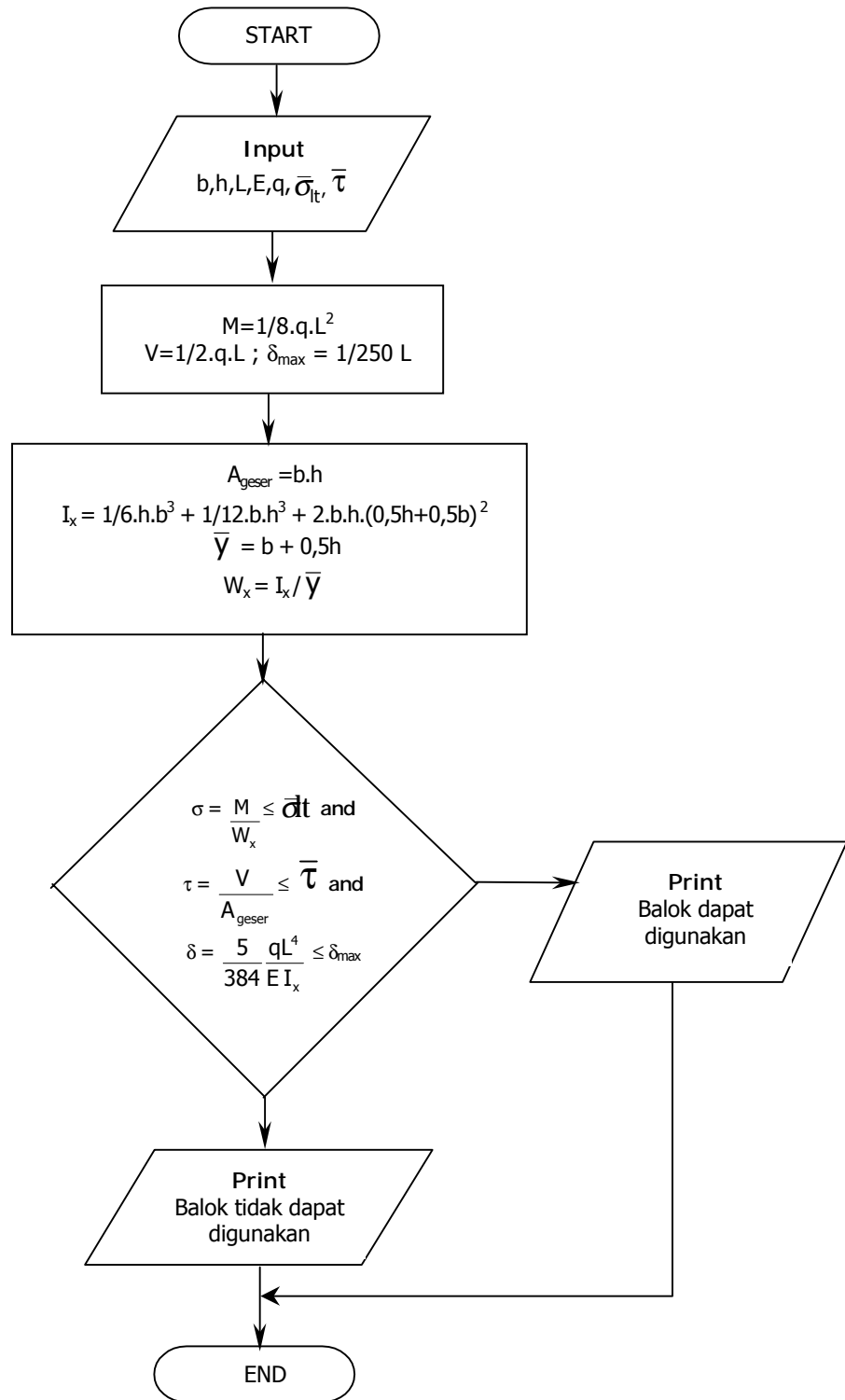
Logika ini akan menjelaskan proses penukaran nilai variabel dengan variabel yang lain. Logika switching ini biasanya digunakan untuk mencari nilai terbesar atau terkecil. Contoh penggunaan logika switching untuk mencari nilai terkecil dan terbesar dari angka 0 sampai dengan 10.



Gambar 1.5 Contoh flowchart untuk logika switching dan counter

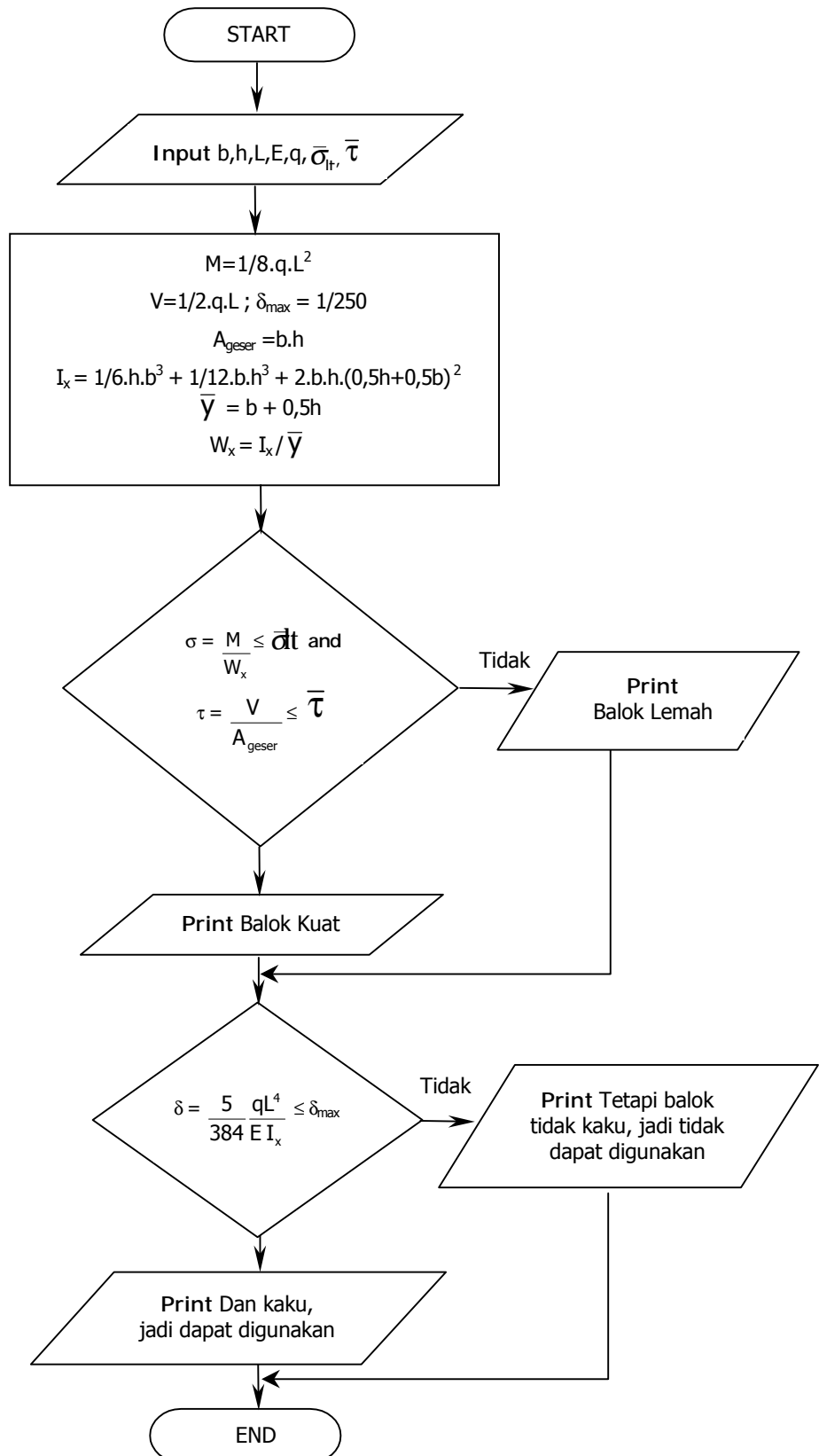
## Menerapkan pendefinisian output-proses-input dalam flowchart

Untuk memperjelas apa dan bagaimana langkah penerjemahan pendefinisian output-proses-input kedalam sebuah flowchart, ada baiknya jika terlebih dahulu disimak alternatif flowchart untuk aplikasi perhitungan gelagar bekisting yang telah disajikan pada bagian sebelumnya.



**Gambar 1.6**  
Flowchart alaternatif 1  
untuk contoh aplikasi pendimensian gelagar bekisting dari kayu





**Gambar 1.7**  
Flowchart alatarnatif 2

untuk contoh aplikasi pendimensian gelagar bekisting dari kayu

Dari gambar 1.6 dan 1.7 tampak bahwa dalam menerjemahkan suatu kasus perhitungan ke dalam sebuah flowchart ada beberapa alternatif flowchart yang dapat dibuat dari pendefinisian output-proses-input program yang sama.

Sehingga disini memang kemampuan imajinasi, kreativitas dan logika programmer akan sangat membantu dalam memilih alur kerja flowchart yang sistematis, terarah dan efisien dari beberapa alternatif yang dapat dibuat.

## **Memvalidasi sistem satuan flowchart**

Sebelum flowchart diterjemahkan kedalam bahasa pemrograman ada baiknya jika flowchart divalidasi dahulu dengan membuat simulasi alur kerja secara manual dengan mengisi data-data. Tujuan dari validasi ini adalah mengecek apakah satuan yang digunakan baik untuk tiap-tiap rumus dalam flowchart maupun untuk data masukan sudah benar.

Biasanya dengan sistem satuan tunggal, user program merasakan program kurang user friendly, sehingga mungkin pada tahap pengembangan program dapat dipertimbangkan penggunaan multi sistem satuan misalnya (kg-cm), (kN-m) dan lain sebagainya.

## **Memperbaiki flowchart**

Dalam dunia pemrograman, Kemampuan untuk dapat memperbaiki alur kerja sebuah flowchart merupakan kemampuan wajib. Dan untuk memperbaiki sebuah flowchart, haruslah terlebih dahulu memiliki kemampuan untuk membaca flowchart dan membandingkannya dengan pendefinisian output-proses-input.

Selama program telah didefinisikan output-proses-inputnya sebenarnya untuk memperbaiki sebuah flowchart, kemampuan penguasaan terhadap disiplin ilmu program yang digunakan secara spesifik misalnya untuk aplikasi metode elemen hingga tidak mutlak diperlukan karena untuk membuat program, programmer bekerja dalam sebuah tim yang memiliki kemampuan masing-masing.

## **Studi Kasus Flowchart**

Setelah membahas tentang apa dan bagaimana flowchart, dibawah ini disajikan pembahasan beberapa kasus tentang aplikasi perhitungan dalam Teknik Sipil untuk didiskusikan dalam kelompok.

### KASUS 1

Buatlah sebuah flowchart untuk menganalisis hasil tes kuat tekan kubus beton dari laboratorium berdasarkan PBI 1971 dengan rumus-rumus sebagai berikut,

	DEFINISI
OUTPUT	Seperti pada lampiran 1
PROSES	<p>Untuk jumlah sample <math>(n) \geq 20</math> maka</p> $\sigma' b_n = \frac{P_n}{A_n \cdot K_n}$ $\sigma' b_m = \frac{\sigma' b_1 + \sigma' b_2 + \sigma' b_3 + \dots + \sigma' b_n}{n}$ $s = \sqrt{\frac{\sum_{n=1}^n (\sigma' b_n - \sigma' b_m)^2}{n - 1}}$ $\sigma' b_k = \sigma' b_m - 1,64s$ <p>Untuk jumlah sample <math>(n) &lt; 20</math> maka</p> $\sigma' b_n = \frac{P_n}{A_n \cdot K_n}$ $\sigma' b_m = \frac{\sigma' b_1 + \sigma' b_2 + \sigma' b_3 + \dots + \sigma' b_n}{n}$ <p><math>P_n</math> = beban maksimum (ton) sample ke-n. <math>A_n</math> = bidang tekan (<math>\text{cm}^2</math>) sample ke-n. <math>K_n</math> = koefisien perbandingan umur untuk 28 hari sample ke-n (<math>\geq 1</math>). <math>\sigma' b_n</math> = kekuatan tekan beton sample ke-n umur 28 hari (<math>\text{kg/cm}^2</math>). <math>\sigma' b_m</math> = kekuatan tekan beton rata-rata (<math>\text{kg/cm}^2</math>). <math>n</math> = Jumlah sample. <math>\sigma' b_k</math> = Kekuatan tekan beton karakteristik (<math>\text{kg/cm}^2</math>).</p>
INPUT	Data untuk tiap sample identifikasi benda uji , tanggal beton cor, umur hari, berat, slump (cm), bidang tekan ( $\text{cm}^2$ ), beban maksimum (ton).