

# Algoritma Pemrograman (2)

Departemen Teknik Informatika  
Universitas Hasanuddin

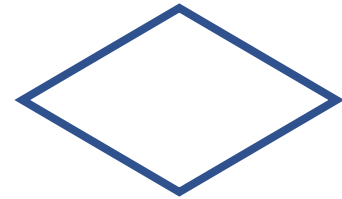
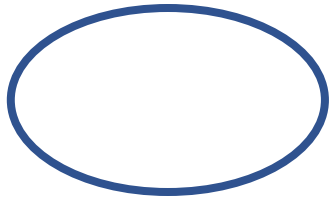
# Learning Outcomes

- Menjelaskan definisi algoritma dan pembuatan algoritma (flowchart)

# Flowchart

- Bagan-bagan yang mempunyai arus yang menggambarkan langkah-langkah penyelesaian suatu masalah.
- Merupakan cara penyajian dari suatu algoritma.
- Ada 2 macam Flowchart :
  - System Flowchart → urutan proses dalam system dengan menunjukkan alat media input, output serta jenis media penyimpanan dalam proses pengolahan data.
  - Program Flowchart → urutan instruksi yang digambarkan dengan symbol tertentu untuk memecahkan masalah dalam suatu program.






# Flowchart Basic Symbols







# Symbol-symbol Flowchart

- *Flow Direction Symbols* (Simbol penghubung alur)
- *Processing Symbols* (Simbol proses).
- *Input-output Symbols* (Simbol input-output)

# Symbol-symbol Flowchart

SIMBOL	NAMA	FUNGSI
	TERMINATOR	Permulaan/akhir program
	GARIS ALIR (FLOW LINE)	Arah aliran program
	PREPARATION	Proses inisialisasi/ pemberian harga awal
	PROSES	Proses perhitungan/ proses pengolahan data
	INPUT/OUTPUT DATA	Proses input/output data, parameter, informasi

# Symbol-symbol Flowchart – *Cont.*

SIMBOL	NAMA	FUNGSI
	PREDEFINED PROCESS (SUB PROGRAM)	Permulaan sub program/ proses menjalankan sub program
	DECISION	Perbandingan pernyataan, penyeleksian data yang memberikan pilihan untuk langkah selanjutnya
	ON PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada satu halaman
	OFF PAGE CONNECTOR	Penghubung bagian-bagian flowchart yang berada pada halaman berbeda

# Pembuatan Flowchart

- Tidak ada kaidah yang baku.
- Flowchart = gambaran hasil analisa suatu masalah
- Flowchart dapat bervariasi antara satu pemrogram dengan pemrogram lainnya.
- Secara garis besar ada 3 bagian utama:
  - Input
  - Proses
  - Output

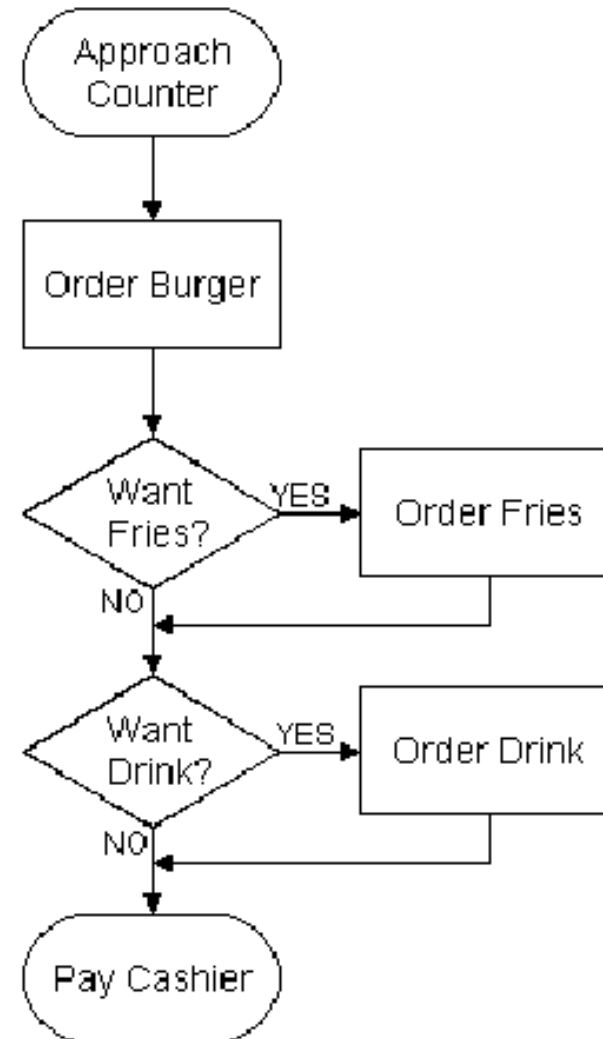


# Pembuatan Flowchart – *cont.*

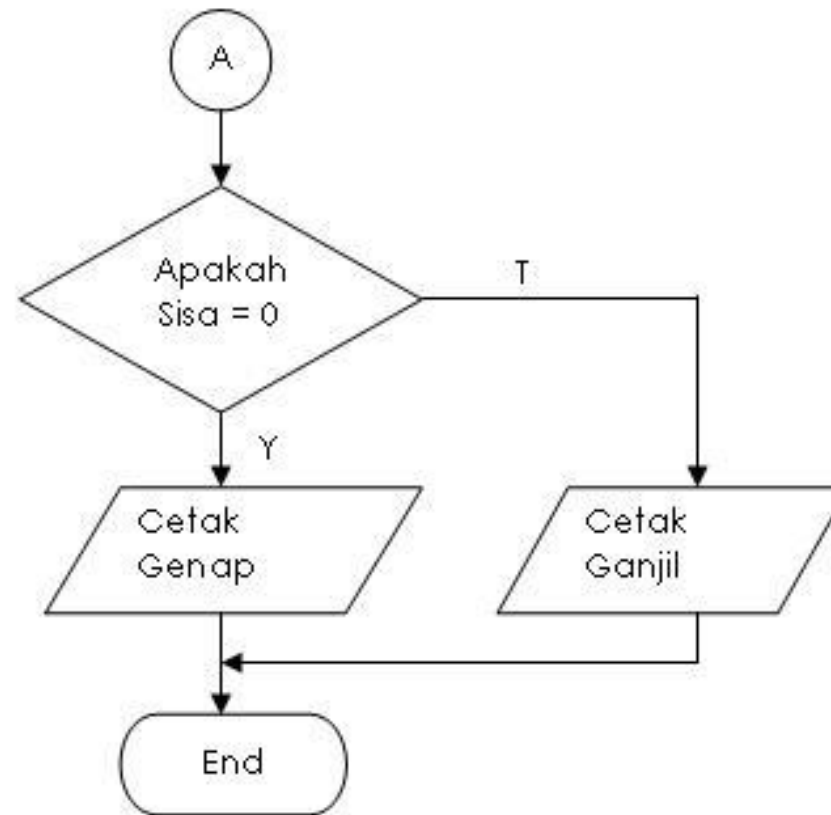
- Hindari pengulangan proses yang tidak perlu dan logika yang berbelit sehingga jalannya proses menjadi singkat.
- Jalannya proses digambarkan dari atas ke bawah dan diberikan tanda panah untuk memperjelas.
- Sebuah flowchart diawali dari satu titik START dan diakhiri dengan END.

## Contoh 1 Flowchart

### Ordering a Burger Standard Process



## Contoh 2 Flowchart



# Pseudocode

- *Outline* dari sebuah program komputer
- Ditulis dalam bahasa Inggris atau Indonesia sederhana
- Kata kunci (*keyword*) digunakan untuk menjelaskan **struktur kendali** (misalnya: “jika”, “ulangi”, “sampai”, “if”, “repeat”, “until”)

# Aturan penulisan algoritma

1. **Bagian Kepala (Header)** → berisi tentang nama algoritma dan keterangan tentang algoritma tersebut
2. **Bagian Deklarasi (Definisi Variabel)** → berisi tentang nama variable, nama tetapan, nama prosedur, nama fungsi dan tipe data yang akan digunakan dalam algoritma
3. **Bagian Deskripsi (rincian langkah)** → berisi tentang langkah-langkah penyelesaian masalah, termasuk perintah baca data, pemberian nilai, penampilan nilai dan lain-lain

# Menulis Pseudocode

Enam operasi dasar komputer:

1. Menerima informasi (Input)
2. Menampilkan Informasi (Output)
3. Melakukan perhitungan aritmetika (Compute)
4. Memberikan nilai ke data (Store)
5. Membandingkan dan Memilih (Compare)
6. Melakukan pengulangan (Loop)

# 1. Menerima Informasi

- Sewaktu komputer menerima informasi atau *input*, maka *statement* yang biasa digunakan adalah “Read”, “Get”, atau “Baca”
- Contoh:
  - Read** Bilangan
  - Get** kode\_pajak
  - Baca nama\_mahasiswa

## 2. Menampilkan Informasi

- Sewaktu komputer menampilkan informasi ataupun *output*, maka *statement* yang biasa digunakan adalah “Print”, “Write”, “Put”, “Output”, “Display” ataupun “Cetak”
- Contoh:
  - Print** “Teknik Informatika”
  - Cetak** “Saya sedang belajar”
  - Output** Average



### 3. Melakukan perhitungan Aritmetika

- Untuk melakukan operasi aritmetika digunakan pseudocode berikut:
  - + untuk penjumlahan (add)
  - Untuk pengurangan (subtract)
  - \* Untuk perkalian (multiply)
  - / Untuk pembagian (divide)
  - () Untuk kurung
- Statement “Compute”, “Calculate” ataupun “Hitung” juga dapat digunakan.
- Contoh:
  - Add** number to total
  - Total = Total + number

## 4. Memberikan nilai ke data

- Ada tiga cara untuk memberikan nilai ke dalam data :
  - Memberikan nilai awal, menggunakan *statement* “Initialize” atau “Set”
  - Memberikan nilai sebagai hasil dari suatu proses, maka tanda “=” digunakan
  - Untuk menyimpan suatu nilai maka *statement* “Save” atau “Store” digunakan
- Contoh:
  - Set** Counter to 0
  - Total = Harga \* Jumlah

## 5. Membandingkan dan memilih

- Salah satu operasi terpenting yang dapat dilakukan komputer adalah membandingkan dan memilih salah satu alternatif solusi.
- Keyword yang digunakan : “IF”, “THEN” dan “ELSE”

- Contoh

**IF** Pilih='1' **THEN**

Discount = 0.1 \* harga

**ELSE**

Discount = 0.2 \* harga

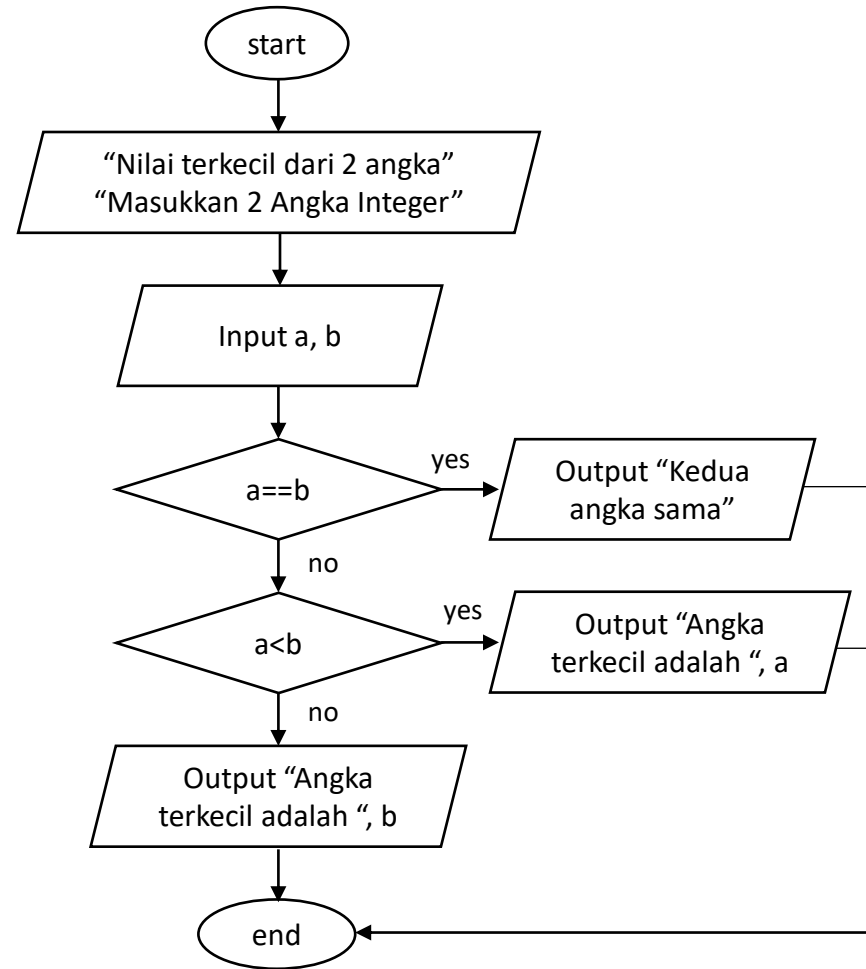
**ENDIF**

## 6. Melakukan pengulangan

- Jika ada beberapa perintah yang harus diulang, maka dapat digunakan *keyword* “DOWHILE” dan “ENDDO”.
- Contoh

```
DOWHILE bil < 10
    cetak bil
    bil = bil +1
ENDDO
```

1. Start
2. Output “Nilai Terkecil dari 2 angka”
3. Output “Masukkan 2 angka integer”
4. Input a dan b
5. If (a==b)
6. Output “Kedua angka sama”
7. Else if (a<b)
8. Output “Angka terkecil adalah “, a
9. Else
10. Output “Angka terkecil adalah “, b
11. End



**Algoritma** Mencari nilai terkecil dari 2 angka

#### Definisi Variabel

integer a, b;

#### Rincian Langkah

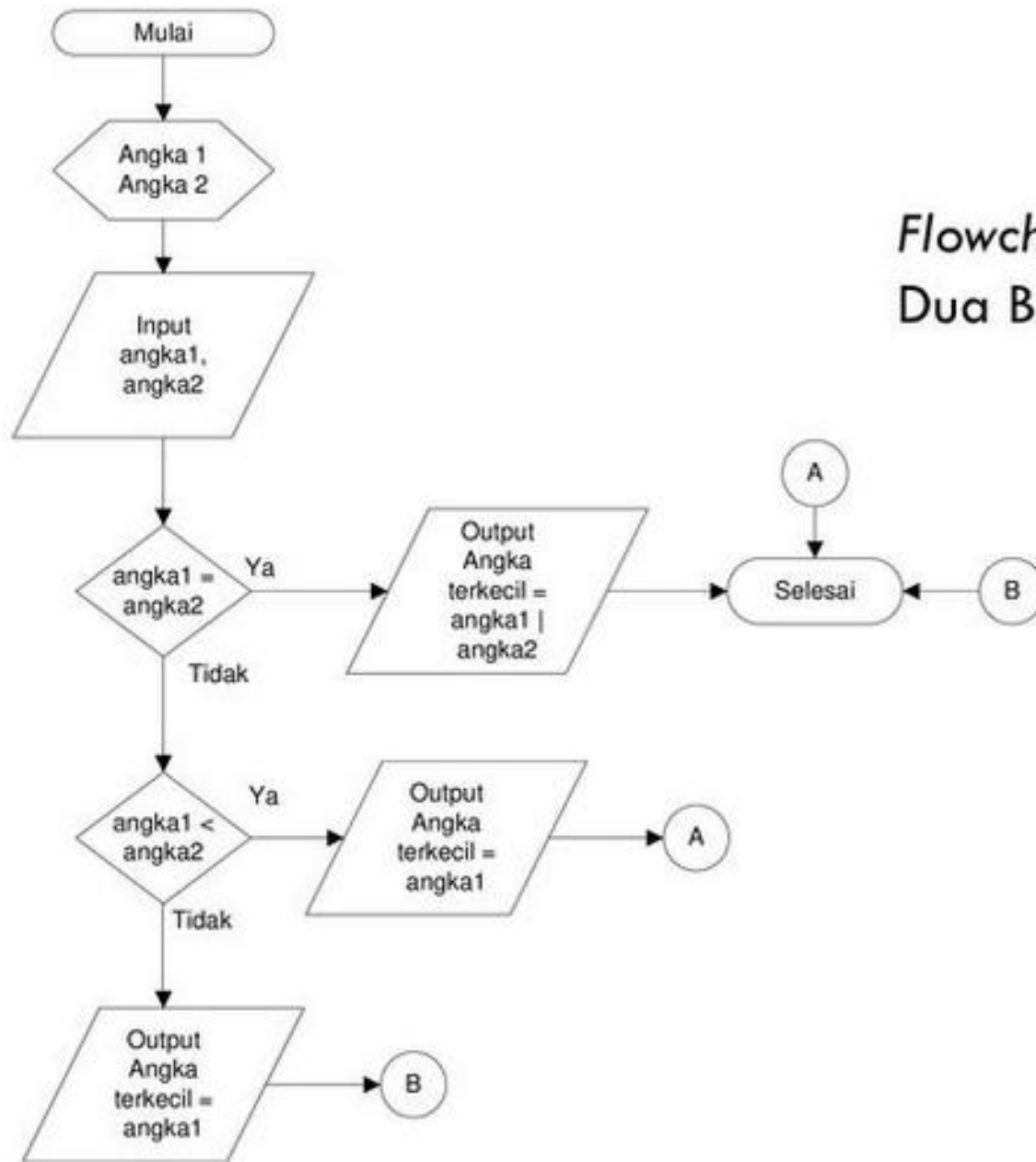
```

print("Nilai Terkecil dari 2 angka");
print("Masukkan 2 angka integer");
{memasukkan angka}
write(a,b);
read(a,b);
{periksa kondisi nilai angka}
if(a==b)
    then write("Kedua angka sama")
else if(a<b)
    then write("Angka terkecil adalah ",a)
    else
        write("Angka terkecil adalah ",b)
end if
  
```

Algoritma  
berbasis konteks  
kalimat

Algoritma  
berbasis diagram

Algoritma  
berbasis pseudo  
code



*Flowchart* untuk Mencari Angka Terkecil dari Dua Bilangan (dengan connector)

# Teorema Terstruktur

- Teorema terstruktur memungkinkan untuk menulis program komputer hanya dengan menggunakan tiga struktur kontrol yaitu:
  1. Sequence
  2. Selection
  3. Repetition

# 1. Sequence

- Sequence merupakan urutan pengerjaan dari perintah / *statement* pertama sampai dengan perintah / *statement* terakhir
- Sequence dapat digunakan untuk menampilkan empat operasi dasar komputer yang telah dibahas sebelumnya yaitu: menerima informasi, menampilkan informasi, melakukan perhitungan aritmetika dan memberikan nilai



# Sequence

- Contoh:

**Cetak** "Jumlah Mahasiswa"

**Set** Jumlah to 49

**Cetak** "Tambahan mahasiswa baru"

**Baca** mhs\_baru

Jumlah = Jumlah + mhs\_baru

**Cetak** "Jumlah Mahasiswa"

**Cetak** jumlah

- Penjelasan

- Urutan pengerjaan adalah mulai dari urutan pertama sampai dengan urutan terakhir, jika mhs\_baru diisi dengan 2, maka jumlah yang tercetak adalah 51

## 2. Selection

- Struktur Kontrol Selection adalah penggambaran sebuah kondisi dan pilihan diantara dua aksi.
- Statement Pertama akan dikerjakan jika kondisi bernilai benar, jika tidak maka akan mengerjakan perintah setelah keyword “else” (jika ada).

# Selection

- Contoh :  
    **IF** Bulan = 2 **THEN**  
        **Cetak** “Februari”  
    **ELSE**  
        **Cetak** “Bukan Februari”
- Penjelasan
  - Tulisan “Februari” akan ditampilkan jika var “Bulan” bernilai 2, jika tidak maka tulisan “Bukan Februari” yang akan ditampilkan

### 3. Repetition

- Beberapa *statement* / perintah dapat diulang dengan menggunakan struktur kontrol *repetition*.
- *Statement* / perintah akan tetap diulang selama kondisi perulangan memenuhi (jika menggunakan DOWHILE – ENDDO)

# Repetition

- Contoh:

Bintang = 0

**DOWHILE** bintang < 5

**Cetak** bintang

    bintang = bintang + 1

**ENDDO**

- Penjelasan:

- Pertama kali bintang akan diisi dengan 0, setelah itu isi dari bintang akan dicetak sebanyak lima kali, sehingga tampilannya akan sebagai berikut:

0 1 2 3 4

Thanks