



# Mesin Abstrak

Tim Pengajar KU1071  
Sem. 1 2008-2009

# Tujuan Perkuliahan

- Mahasiswa memahami konsep mesin abstrak (kasus: Mesin Gambar dan Mesin Karakter)
- Mahasiswa memahami primitif-primitif yang terdefinisi pada mesin abstrak tsb.
- Mahasiswa dapat membuat program yang memanfaatkan mesin abstrak melalui primitif-primitifnya

# Konsep Mesin Abstrak

- Mesin adalah mekanisme yang terdefinisi dan mengerti serta mampu untuk mengeksekusi aksi-aksi primitif yang terdefinisi untuk mesin tersebut
- Mesin abstrak adalah mesin yang dianggap ada, dan diasumsikan mampu melakukan mekanisme yang didefinisikan untuk mesin tersebut

# Konsep Mesin Abstrak (lanjutan)

- Mesin abstrak memodelkan suatu semesta (*universe*) tertentu
- Pemodelan mesin abstrak dilakukan agar dapat melakukan pemecahan masalah secara bertahap
- Mendefinisikan mesin abstrak berarti mendefinisikan:
  - sekumpulan **state** yang mungkin
  - **aksi primitif** yang bisa dilakukannya

# Contoh Mesin Abstrak

- Pada diktat diberikan contoh-contoh mesin abstrak, yaitu:
  - Mesin gambar
  - Mesin karakter
  - Mesin integer
  - Mesin rekam
- Pada sesi kuliah ini diajarkan tentang mesin gambar dan mesin karakter (mesin yang lain silahkan dipelajari sendiri)

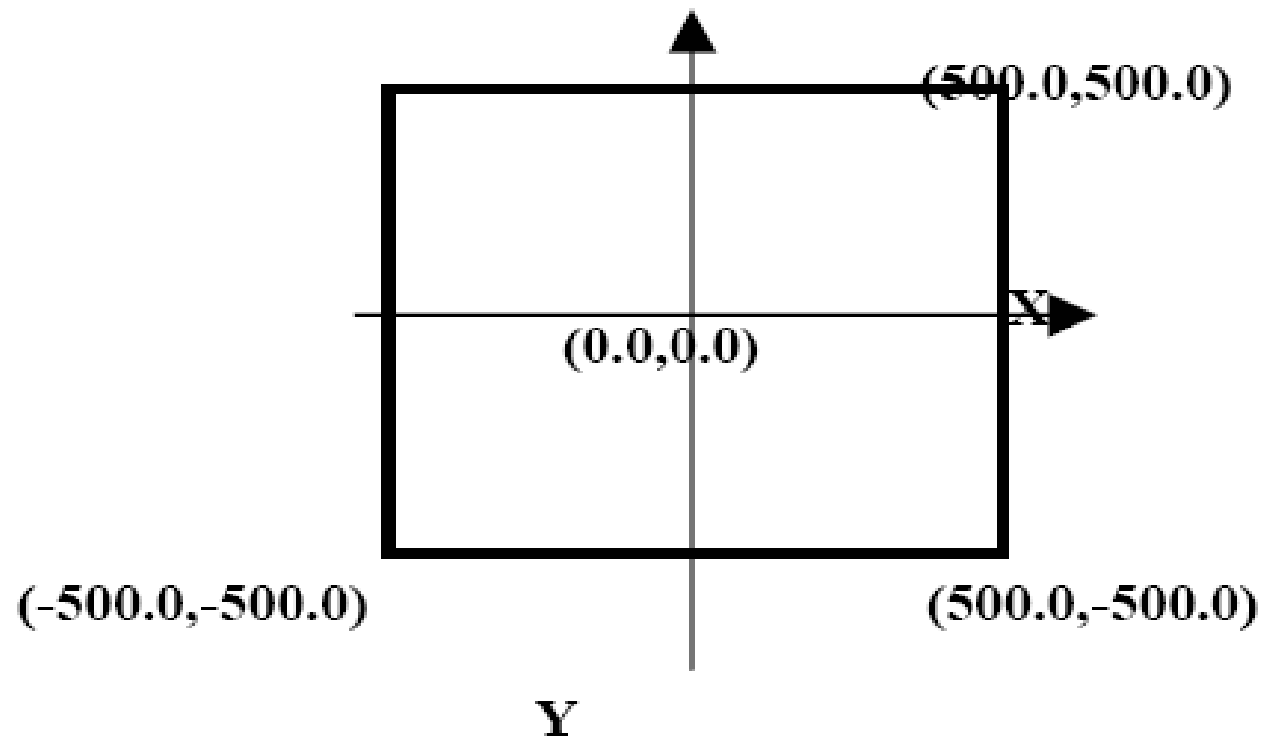
# Mesin Gambar

- Definisi: Mesin Gambar adalah mesin abstrak yang terdiri dari bidang gambar dan pena. Keduanya menyatakan keadaan (*state*) dari mesin
- Bidang Gambar: sekumpulan titik yang membentuk sebuah permukaan/area terbatas yang posisinya dikenali melalui koordinat kartesian dengan  $(0.0, 0.0)$  di tengah bidang gambar

# Mesin Gambar (lanjutan)

- Keadaan bidang gambar dinyatakan oleh titik-titik koordinat yang hitam atau putih.
- Pena mempunyai 3 atribut, yaitu:
  - Posisi penulisan (**Pen**) bisa on atau off. Hanya pada kondisi on pena dapat menghitamkan titik di bidang gambar
  - Posisi pena di bidang gambar (**PosXY**) yang bertipe **Point: <x:real,y:real>**
  - Arah pena (**Dir**) dinyatakan dalam derajat [-360..360]

# Mesin Gambar





# Primitif Mesin Gambar

- Mesin gambar memiliki sejumlah primitif yang terdefinisi (lihat diktat hal. 151-152)
  - Clear, Restart
  - Move, ToPos,
  - Up, Down
  - Right, Left, SetDir

# Studi Kasus: Menggambar Topi

## **Program** GambarTOPI

```
{ I.S: Pen=sembarang, PosXY= sembarang, Dir=sembarang }  
{ F.S: Pen=sembarang, PosXY= sembarang, Dir=sembarang }  
{ Proses: sebuah gambar topi tergambar di bidang gambar }
```

## **KAMUS**

### procedure TOPI

```
{ I.S: Pen=on, PosXY=<x,y>, Dir=60.0 }  
{ F.S: Pen=on, PosXY=<x+15.0,0.0>, Dir=-60.0  
      dan topi tergambar }
```

## **ALGORITMA**

```
Restart  
Down  
SetDir(60.0)  
TOPI
```

# Menggambar Topi (lanjutan)

**procedure** TOPI

```
{ I.S: Pen=on, PosXY=<x,y>, Dir=60.0 }  
{ F.S: Pen=on, PosXY=<x+15.0,y>, Dir=-60.0  
      dan topi tergambar }
```

**KAMUS LOKAL****ALGORITMA**

```
Move(15.0)  
Right(120.0)  
Move(15.0)
```

# Menggambar Barisan Topi

## **Program** BarisanTopi

```
{ I.S: Pen=sembarang, PosXY= sembarang, Dir=sembarang }  
{ F.S: 5 buah gambar topi tergambar di bidang gambar }
```

## **KAMUS**

```
sisi: real>0.0 { panjang sisi topi }  
arah: real { arah sisi miring topi }  
type point: <x:real,y:real>  
awal: point { titik awal penggambaran topi pertama }  
procedure TopiBis(input L:real)  
{ I.S: Pen=on, PosXY=<x,y>, Dir=arah }  
{ F.S: Pen=on, PosXY=<x + L cos(arah),y + L sin(arah)>, Dir=arah-120.0  
      dan sebuah topi tergambar }
```

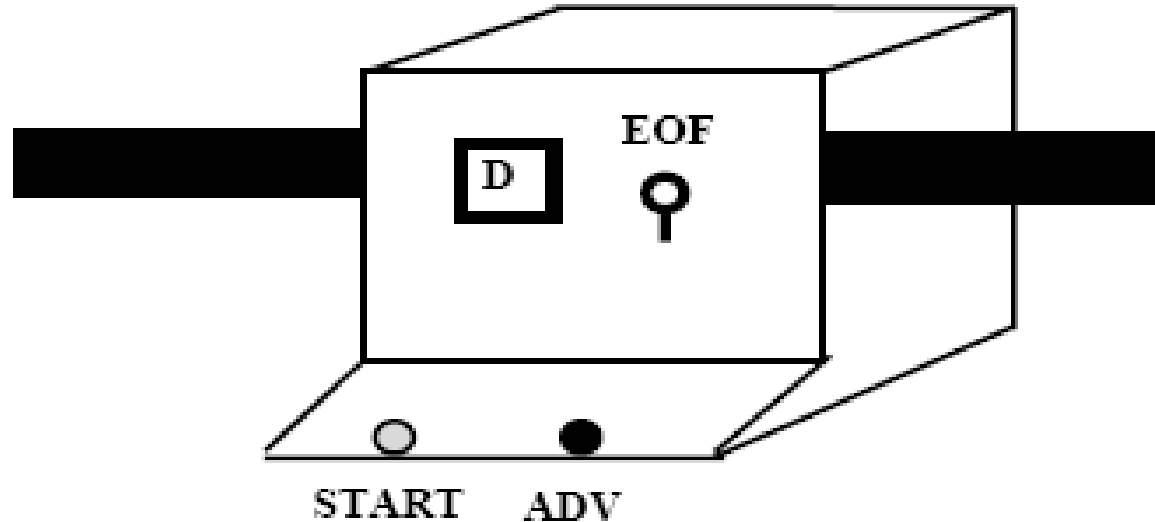
## **ALGORITMA**

```
Clear  
input(sisi,awal,arah)  
SetDir(arah)  
ToPos(awal)  
Down  
repeat 5 times  
    TopiBis(sisi)  
    Left(120.0)
```

# Mesin Karakter

- Definisi: Mesin karakter adalah mesin abstrak yang terdiri dari:
  - Pita berisi deret karakter yang diakhiri “.” (titik). Pita kosong hanya berisi titik saja.
  - Tombol **START**, **ADV**
  - Sebuah lampu **EOP** (End of Pita)
  - Jendela/display sebesar 1 karakter yang dinamakan **CC** (*current character*)

# Mesin Karakter

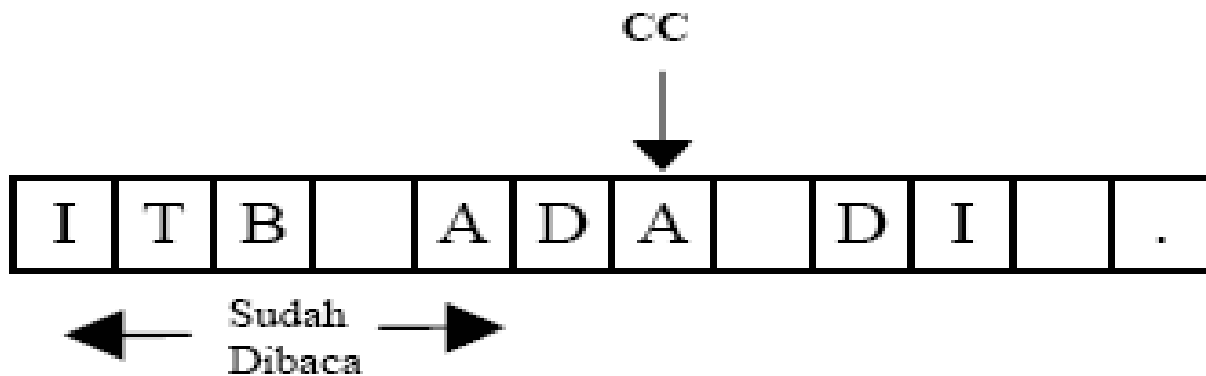
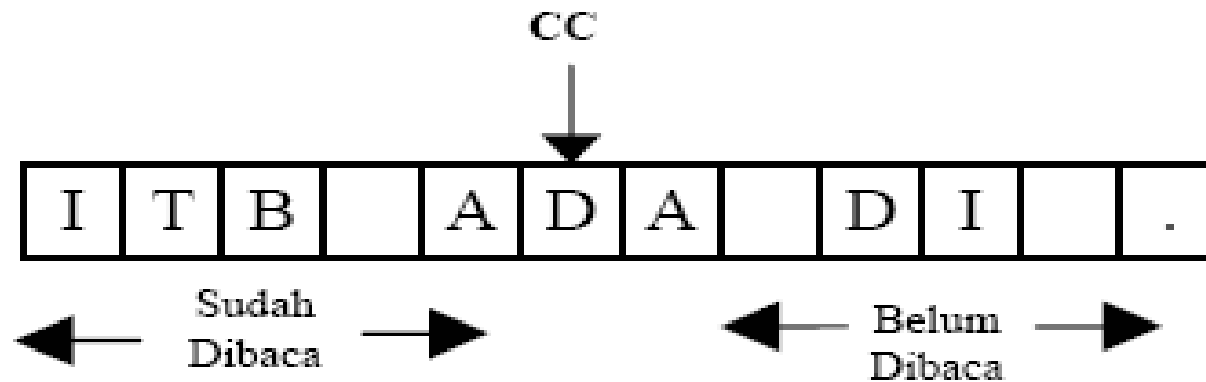


Suatu keadaan Mesin Karakter  
 $CC = 'D'$ , lampu EOP tidak menyala

# Mesin Karakter (lanjutan)

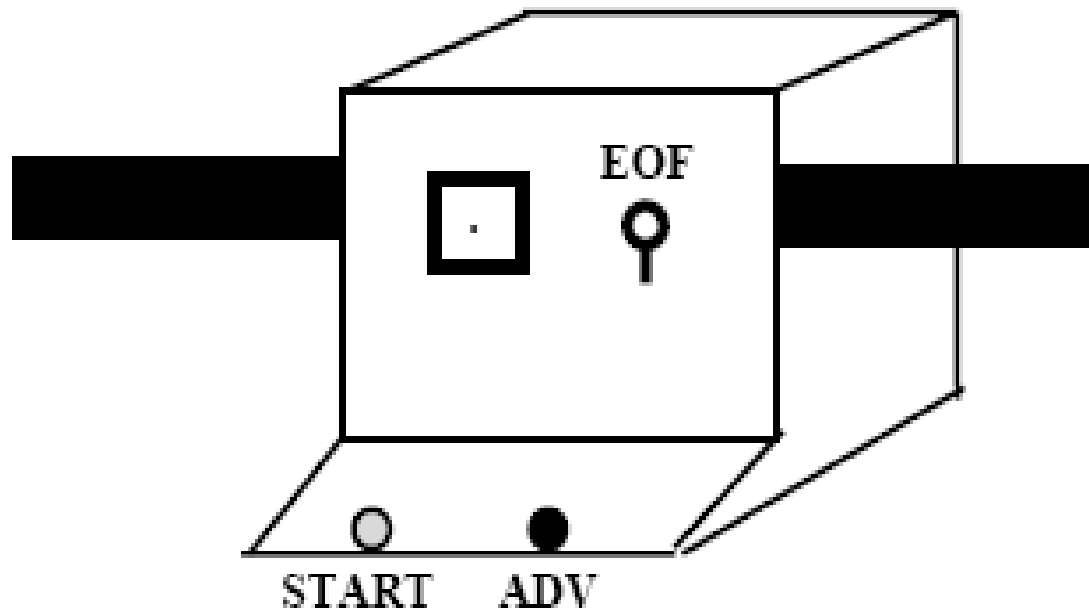
- Hanya karakter yang posisinya di jendela yang bisa dibaca
- Tombol START menyebabkan mesin siap dioperasikan, pita karakter siap dibaca
- Tombol ADV menyebabkan pita karakter maju 1 karakter
- Lampu EOP akan menyala jika karakter di jendela adalah titik

# Pita Karakter





# Mesin Karakter



Ketika  $CC = '.'$ , lampu EOP menyala

# Primitif Mesin Karakter

- Mesin karakter memiliki beberapa primitif yang terdefinisi (lihat diktat hal. 160)
  - START menjadikan mesin siap dioperasikan, pita siap dibaca, posisi karakter pertama pita pada jendela sehingga CC bisa digunakan/dibaca
  - ADV untuk memajukan pita 1 karakter. Karakter berikutnya akan berada di jendela. Bila CC adalah titik maka EOP menyala (true)

# Menghitung Banyaknya Huruf

## **Program** COUNTHURUF

{ Skema pemrosesan dengan MARK: menghitung banyaknya huruf pada pita karakter dan menampilkan hasilnya }

## **KAMUS**

N: integer { banyaknya huruf }

## **ALGORITMA**

N ← 0	{ inisialisasi }
<b>START</b>	{ first Elmt }
<u>while</u> (CC ≠ '.') <u>do</u>	{ atau <u>while</u> ( <u>not</u> EOP) <u>do</u> }
N ← N + 1	{ proses }
<b>ADV</b>	{ next Elmt }
{ CC = '.' }	{ EOP }
<u>output</u> (N)	{ terminasi }

# Banyaknya Huruf A

**Program** COUNT\_A

{ Skema pemrosesan dengan MARK: menghitung banyaknya huruf A pada pita karakter dan menampilkan hasilnya }

**KAMUS**

N: integer { banyaknya huruf }

**ALGORITMA**

N  $\leftarrow$  0 { inisialisasi }

**START** { first Elmt }

while (not **EOP**) do

depend on CC { proses }

        CC = 'A': N  $\leftarrow$  N + 1

        CC  $\neq$  'A': -

**ADV** { next Elmt }

{ EOP } { EOP }

output (N) { terminasi }

# Banyaknya Pasangan AN

**Program** COUNT\_AN

{ Skema pemrosesan dengan MARK: menghitung banyaknya pasangan huruf A dan N pada pita karakter dan menampilkan hasilnya }

**KAMUS**

AN : integer { banyaknya pasangan huruf AN}  
CC1: character { penampung karakter sebelum CC }

**ALGORITMA**

```
AN ← 0                      { inisialisasi }
CC1 ← ' '                   { inisialisasi }
START                      { first Elmt }
while (not EOP) do
  if (CC1='A') and (CC='N') then { proses }
    AN ← AN + 1
  CC1 ← CC                  { nilai CC disimpan }
ADV                        { next Elmt }
{ EOP }                     { EOP }
output(N)                   { terminasi }
```

**Program** COUNT\_1

{menghitung kemunculan angka 1}

# Menghitung kemunculan 1

## KAMUS

pjgstr1 : integer { panjang karakter 1 utk setiap kemunculan }  
jmlgroup : integer { jumlah kemunculan group karakter 1 }  
CC1 : character {huruf sebelumnya }

## ALGORITMA

pjgstr1 ← 0 { inisialisasi }  
jmlgroup ← 0 { inisialisasi }  
CC1 ← '0' { inisialisasi }  
**START** { first Elmt }

while (not **EOP**) do

if (CC = '1') then

if (CC1='0') then { proses }

            jmlgroup ← jmlgroup + 1

            pjgstr1 ← 0

            pjgstr1 ← pjgstr1 + 1

else if (CC1='1')

output(pjgstr1)

    CC1 ← CC { nilai CC disimpan }

**ADV** { next Elmt }

{ EOP } { EOP }

if (pjgstr1 > 0) then

output(pjgstr1)

output('Banyaknya kelompok deretan 1 =',jmlgroup) { terminasi }

# Latihan (Mesin Gambar)

- Buat prosedur untuk menggambar bujur sangkar dengan titik pusat P dengan panjang sisi L  
procedure DrawSquare(input P:point,input L:real)
- Buat program untuk menggambar 10 buah bujur sangkar dengan pusat di  $\langle 50.0, 50.0 \rangle$  dengan panjang sisi 10.0, 20.0, 30.0, 40.0, ..., 100.0. Gunakan prosedur DrawSquare di atas.

# Latihan (Mesin Karakter)

- Buatlah program untuk menghitung banyaknya huruf hidup (vokal) yang terdapat pada pita karakter.

Catatan: bisa juga ditambahkan fungsi **is\_vokal** untuk melengkapi program tsb.

function is\_vokal(ch:character) → boolean



# Latihan

- Buatlah program yang melakukan perbandingan mana yang lebih besar banyaknya huruf hidup (vokal) dibandingkan dengan banyaknya huruf mati (konsonan) yang terdapat pada pita karakter. Perhatikan: pada pita karakter bisa terdapat karakter non alfabet.

Catatan: bisa menggunakan fungsi **is\_vokal** sebelumnya dan tambahkan fungsi **is\_konsonan**.

function is\_konsonan(ch:character) → boolean

# Latihan

- Bila diasumsikan pita karakter hanya berisi karakter alfabet, buatlah program untuk menghitung banyaknya perubahan dari huruf hidup (vokal) ke huruf mati (konsonan). Bisa menggunakan fungsi **is\_vokal** dan **is\_konsonan** sebelumnya.

Contoh:

Pita karakter: ainunnaaaiim.

Hasilnya: 3

# Translasi ke Pascal

# MesinKar

(\* mesinkar.pas : realisasi primitif-ptimitif mesin karakter \*)

(\* Tanggal : 17 November 2008 \*)

(\* Oleh : Fazat Nur Azizah \*)

unit mesinkar;

interface (\* Kamus \*)

const MARK = '.'; (\* konstanta MARK \*)

PITA\_KARAKTER = 'pitakar.txt'; (\* definisi pita \*)

var CC : char; (\* definisi states \*)

f : text;

procedure START;

(\* mesin siap dioperasikan, pita disiapkan untuk dibaca, karakter pertama \*)

(\* yang ada pada pita posisinya adalah pada jendela I.S : CC sembarang F.S : CC karakter pertama pada pita; jika CC <> MARK maka EOP akan padam (false) jika CC = MARK maka EOP akan menyala (true) \*)

function EOP : boolean;(\* true jika end of pita (cc = mark) \*)

procedure ADV;(\* memajukan satu karakter

I.S : karakter pada jendela = CC, CC <> MARK

F.S : CC adalah karakter berikut dari CC yang lama, CC mungkin MARK. Jika CC = MARK maka EOP akan menyala (true) \*)

## Implementation

```
procedure START;  
begin  
    assign (f, PITA_KARAKTER);  
    reset (f);  
    read (f, CC);  
end;  
  
function EOP : boolean;  
begin  
    EOP := (CC = MARK);  
end;  
  
procedure ADV;  
begin  
    read(f, CC);  
    if (CC = MARK) then  
        close (f);  
end;  
  
begin  
end.
```