

# Mesin Kata

IF2110/IF2111 – Algoritma dan Struktur Data  
Sekolah Teknik Elektro dan Informatika  
Institut Teknologi Bandung

# Mesin Kata (1)

Mesin Kata:

- Mesin abstrak yang bekerja memproses kata berdasarkan mesin karakter
- Diberikan sebuah mesin karakter dengan pita berisi karakter (mungkin kosong), yang diakhiri titik ('.')

# Mesin Kata (2)

Kata:

sederetan karakter suksesif pada pita yang merupakan karakter bukan blank

Definisi type Kata:

```
type Kata: < buffer: array [0..N_MAX-1] of character,  
              length: integer >  
{ buffer adalah tempat penampung/container kata,  
  length menyatakan panjangnya kata }  
{ Deklarasi: K: Kata }  
{ Definisi kata kosong: K.Length = 0 }  
{ Memori array yg dipakai selalu dimulai dari indeks  
  ke-0 }
```

# Mesin Kata (3)

Model-model akuisisi KATA (token) pada pita karakter:

- Versi 1
- Versi 2
- Versi 3

a. Hanya mengandung titik (pita kosong)



b. Hanya mengandung blank diakhiri titik



c. Mengandung blank di awal dan akhir pita



d. Tidak mengandung blank di awal maupun di akhir pita



e. Mengandung blank di akhir pita



f. Mengandung blank di awal pita

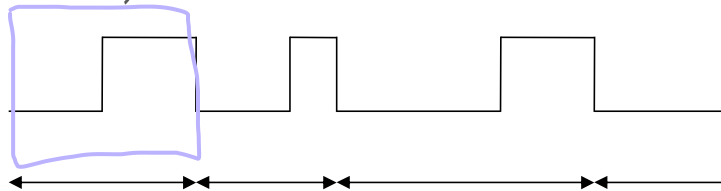


# Mesin Kata

## Model Akuisisi Kata Versi 1

# Model Akuisisi Kata Versi 1

- Kata diakuisisi mulai dari **karakter pertama sesudah akhir kata** (atau karakter pertama pita untuk kata pertama)  
blank/spasinya tetep dibaca



- Akhir dari proses adalah sebuah boolean (**endKata**), yang akan berisi true jika kata terakhir telah diakuisisi dan diproses.

# Model Akuisisi Kata Versi 1 (1)

KAMUS UMUM

```
{ ***** Mesin lain yang dipakai ***** }
```

use MESINKAR

```
{ ***** Konstanta ***** }
```

```
constant MARK: character = '.'
```

```
constant BLANK: character = ' '
```

```
constant N_MAX: integer = 50 { jumlah maksimum karakter suatu kata }
```

```
{ ***** Definisi Type Kata ***** }
```

```
type Kata: < buffer: array [0..N_MAX-1] of character,  
                    length: integer >
```

```
{ buffer adalah tempat penampung/container kata,  
  length menyatakan panjangnya kata }
```

```
{ ***** Definisi State Mesin Kata ***** }
```

```
endKata: boolean { penanda akhir akuisisi kata }
```

```
currentKata : Kata     { kata yang sudah diakuisisi dan akan diproses }
```

# Model Akuisisi Kata Versi 1 (2)

```
{***** Primitif-Primitif Mesin Kata *****}  
procedure ignoreBlank  
{ Mengabaikan satu atau beberapa BLANK }  
{ I.S.: cc sembarang }  
{ F.S.: cc ≠ BLANK atau cc = MARK }  
procedure startKata  
{ I.S.: cc sembarang }  
{ F.S.: endKata = true, dan cc = Mark;  
{  
    atau endKata = false,  
    currentKata adalah kata yang sudah diakuisisi,  
    cc karakter pertama sesudah karakter terakhir kata }  
procedure advKata  
{ I.S.: cc adalah karakter pertama kata yang akan diakuisisi }  
{ F.S.: currentKata adalah kata terakhir yang sudah diakuisisi,  
    cc adalah karakter pertama sesudah karakter terakhir kata }  
{ Proses: Akuisisi kata menggunakan procedure salinKata }  
procedure salinKata  
{ Mengakuisisi kata, menyimpan dalam currentKata }  
{ I.S.: cc adalah karakter pertama dari kata }  
{ F.S.: currentKata berisi kata yang sudah diakuisisi;  
    cc = BLANK atau cc = MARK;  
    cc adalah karakter sesudah karakter terakhir yang diakuisisi }
```



# Model Akuisisi Kata Versi 1 (3)

procedure ignoreBlank

{ *Mengabaikan satu atau beberapa BLANK* }

{ *I.S.: cc sembarang* }

{ *F.S.: cc ≠ BLANK atau cc = MARK* }

KAMUS LOKAL

ALGORITMA

while (cc = BLANK) do

adv

{ *cc ≠ BLANK* }

# Model Akuisisi Kata Versi 1 (4)

```
procedure startKata
{ I.S.: cc sembarang }
{ F.S.: endKata = true, dan cc = Mark; }
{
    atau endKata = false, currentKata adalah kata yang sudah
    diakuisisi,
    cc karakter pertama sesudah karakter terakhir kata }

```

KAMUS LOKAL

ALGORITMA

```
start
ignoreBlank
if (cc = MARK) then
    endKata ← true
else
    endKata ← false
salinKata
```

# Model Akuisisi Kata Versi 1 (5)

procedure advKata

*{ I.S.: cc adalah karakter pertama kata yang akan diakuisisi }  
{ F.S.: currentKata adalah kata terakhir yang sudah diakuisisi,  
cc adalah karakter pertama dari kata berikutnya,  
mungkin MARK }  
{ Proses: Akuisisi kata menggunakan procedure salinKata }*

KAMUS LOKAL

ALGORITMA

```
ignoreBlank
if (cc=MARK) then
    endKata ← true
else
    salinKata
```

# Model Akuisisi Kata Versi 1 (6)

procedure salinKata

*{ Mengakuisisi kata, menyimpan dalam currentKata }*  
*{ I.S.: cc adalah karakter pertama dari kata }*  
*{ F.S.: currentKata berisi kata yang sudah diakuisisi;*  
*cc = BLANK atau cc = MARK;*  
*cc adalah karakter sesudah karakter terakhir yang*  
*diakuisisi }*

KAMUS LOKAL

i: integer

ALGORITMA

i ← 0

repeat

currentKata.buffer[i] ← cc

adv

i ← i + 1

until (cc = MARK) or (cc = BLANK) *disalin per kata terlebih dahulu*

*{ cc = MARK or cc = BLANK }*

currentKata.length ← i

karakter pertama

# Studi Kasus 1 - Panjang Rata-Rata Kata

Diberikan pita berisi karakter (mungkin kosong) yang diakhiri titik, hitunglah panjang rata-rata kata yang ada pada pita tersebut.

Panjang kata rata-rata tidak terdefinisi jika pita kosong atau pita tidak mengandung kata (hanya berisi 'blank' dan titik).

# Panjang Rata-Rata Kata - Model Akuisisi Kata Versi 1 (1)

Program PanjangRataRataKata1

*{ Menghitung panjang rata-rata kata dalam pita karakter }*

*{ Model akuisisi kata versi 1 }*

KAMUS

*{ \*\*\* Mesin yang digunakan \*\*\* }*

USE MesinKata1

nbKata: integer *{ banyaknya kata dalam pita }*

lengthTotal: integer *{ akumulasi panjang kata }*

ALGORITMA

*{ di halaman berikutnya }*

# Panjang Rata-Rata Kata - Model Akuisisi Kata Versi 1 (2)

## ALGORITMA

```
lengthTotal ← 0
nbKata ← 0
startKata
while not endKata do
    lengthTotal ← lengthTotal + currentKata.length
    nbKata ← nbKata + 1
    advKata
{ endKata = true: semua karakter sudah diakuisisi }
if (nbKata ≠ 0) then
    output (lengthTotal/nbKata)
else { nbKata = 0 }
    output ("Pita tidak mengandung kata")
```

## Studi Kasus 2 - Hitung WHILE

Diberikan suatu pita karakter yang mengandung abjad, blank, dan diakhiri titik, harus dicari banyaknya kemunculan kata 'WHILE' pada pita tersebut

*Hint:* dapat memanfaatkan fungsi `isKataEqual`

```
function isKataEqual (k1, k2: Kata) → boolean  
    { Menghasilkan true jika k1 = k2 }
```



# Hitung WHILE - Model Akuisisi Kata Versi 1 (1)

## Program HitungWhile1

*{ Menghitung banyaknya kata WHILE dalam pita karakter }  
{ Model akuisisi kata versi 1 }*

## KAMUS

*{ \*\*\* Mesin yang digunakan \*\*\* }*

USE MesinKata1

kataWHILE: Kata *{ Kata yang menyimpan WHILE }*

nWHILE: integer *{ banyaknya kata WHILE }*

function isKataEqual (k1, k2: Kata) → boolean

*{ Menghasilkan true jika k1 = k2 }*

## ALGORITMA

*{ di halaman berikutnya }*

# Hitung WHILE - Model Akuisisi Kata Versi 1 (2)

## ALGORITMA

```
{ Inisialisasi kataWHILE }
kataWHILE.buffer[0] ← 'W'
kataWHILE.buffer[1] ← 'H'
kataWHILE.buffer[2] ← 'I'
kataWHILE.buffer[3] ← 'L'
kataWHILE.buffer[4] ← 'E'
kataWHILE.length ← 5
nWHILE ← 0
startKata
while not endKata do
    if isKataEqual(kataWHILE, currentKata) then
        nWHILE ← nWHILE + 1
    advKata
{ endKata = true: semua karakter sudah diakuisisi }
output (nWHILE)
```

## Hitung WHILE - Model Akuisisi Kata Versi 1 (3)

Sebagai latihan, realisasikan fungsi `isKataEqual` sebagai berikut.

```
function isKataEqual (k1, k2: Kata) → boolean  
  { Menghasilkan true jika k1 = k2 }
```