

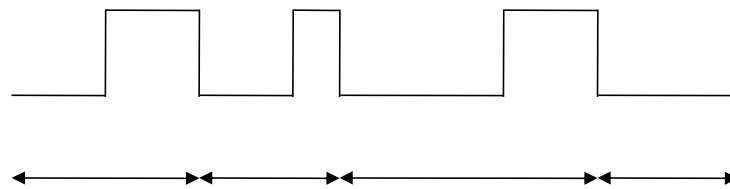
# Mesin Kata

## Model Akuisisi Kata Versi 2

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

# Model Akuisisi Kata Versi 2

Kata diakuisisi mulai dari karakter pertama sesudah akhir kata (atau karakter pertama pita untuk kata pertama) → sama dengan versi 1



Akhir dari proses adalah sebuah kata yang '**kosong**', yaitu panjangnya = 0. Digunakan panjang kata untuk menentukan apakah proses pembacaan pita karakter sudah selesai/belum.

# Model Akuisisi Kata Versi 2 (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 ***** }
```

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

# Model Akuisisi Kata Versi 2 (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.: currentKata.length = 0, dan cc = Mark; }  
{  
    atau currentKata.length ≠ 0, 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 dari kata berikutnya,  
    mungkin MARK }  
{ 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 2 (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 }

procedure startKata

{ I.S.: cc sembarang }

{ F.S.: currentKata.length = 0, dan cc = Mark; }

{  
  atau currentKata.length ≠ 0, currentKata adalah kata yang sudah  
  diakuisisi, cc karakter pertama sesudah karakter terakhir  
  kata }

KAMUS LOKAL

-

ALGORITMA

  start

  ignoreBlank

  salinKata

# Model Akuisisi Kata Versi 2 (4)

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  
salinKata

# Model Akuisisi Kata Versi 2 (5)

```
{***** Primitif-Primitif Mesin Kata *****}
```

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

while (cc ≠ MARK) and (cc ≠ BLANK) do

    currentKata.buffer[i] ← cc

    adv

    i ← i + 1

{ cc = MARK or cc = BLANK }

currentKata.length ← i

# 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 tsb. 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 2 (1)

## Program PanjangRataRataKata2

```
{ Menghitung panjang rata-rata kata dalam pita karakter }  
{ Model akuisisi kata versi 2 }
```

## KAMUS

```
{ *** Mesin yang digunakan *** }  
USE MesinKata2
```

```
nbKata: integer { banyaknya kata dalam pita }  
lengthTotal: integer { akumulasi panjang kata }
```

## ALGORITMA

```
{ di halaman berikutnya }
```

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

## ALGORITMA

```
lengthTotal ← 0
nbKata ← 0
startKata
while currentKata.length ≠ 0 do
    lengthTotal ← lengthTotal + currentKata.length
    nbKata ← nbKata + 1
    advKata
{ currentKata.length = 0: cc mencapai MARK }
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

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

## Program HitungWhile2

```
{ Menghitung banyaknya kata WHILE dalam pita karakter }  
{ Model akuisisi kata versi 2 }
```

## KAMUS

```
{ *** Mesin yang digunakan *** }  
USE MesinKata2
```

```
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 2 (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 currentKata.length ≠ 0 do
    if isKataEqual(kataWHILE, currentKata) then
        nWHILE ← nWHILE + 1
    advKata
{ currentKata.length = 0: cc sampai pada MARK }
output (nWHILE)
```

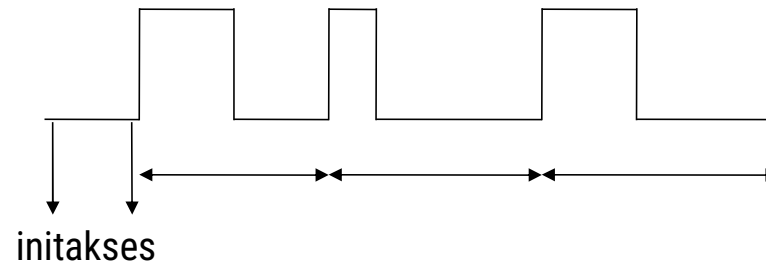
```
function isKataEqual (k1, k2: Kata) → boolean
{ sama seperti pada model akuisisi kata versi 1 }
```

# Mesin Kata

## Model Akuisisi Kata Versi 3

# Model Akuisisi Kata Versi 3

Mengabaikan BLANK pada awal pita dan memproses sisanya



Model akuisisi kata **TANPA MARK**, artinya kata yang diakuisisi tidak pernah merupakan kata 'kosong'

Model akuisisi ini mengharuskan adanya suatu prosedur **initAkses**, yang memposisikan cc pada karakter pertama kata pertama

# Model Akuisisi Kata Versi 3 (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 ***** }
```

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



# Model Akuisisi Kata Versi 3 (2)

```
{***** Primitif-Primitif Mesin Kata *****}  
procedure ignoreBlank  
{ Mengabaikan satu atau beberapa BLANK }  
{ I.S.: cc sembarang }  
{ F.S.: cc ≠ BLANK atau cc = MARK }  
procedure initAkses  
{ Mengabaikan satu atau beberapa BLANK pada awal pita }  
{ I.S.: cc sembarang }  
{ F.S.: cc = MARK; atau cc = karakter pertama dari kata yang akan diakuisisi }  
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 }  
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 3 (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 }

procedure initAkses

{ Mengabaikan satu atau beberapa BLANK pada awal pita }

{ I.S.: cc sembarang }

{ F.S.: cc = MARK; atau cc = karakter pertama dari kata yang akan diakuisisi }

KAMUS LOKAL

-

ALGORITMA

  start

  ignoreBlank

# Model Akuisisi Kata Versi 3 (4)

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

salinKata

ignoreBlank

# Model Akuisisi Kata Versi 3 (5)

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  $\leftarrow$  0

repeat

    currentKata.buffer[i]  $\leftarrow$  cc

    adv

    i  $\leftarrow$  i + 1

until (cc = MARK) or (cc = BLANK)

{ cc = MARK or cc = BLANK }

currentKata.length  $\leftarrow$  i

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

# Panjang Rata-Rata Kata - Model Akuisisi Kata Versi 3 (1)

## Program PanjangRataRataKata3

```
{ Menghitung panjang rata-rata kata dalam pita karakter }  
{ Model akuisisi kata versi 3 }
```

## KAMUS

```
{ *** Mesin yang digunakan *** }  
USE MesinKata3
```

```
nbKata: integer { banyaknya kata dalam pita }  
lengthTotal: integer { akumulasi panjang kata }
```

## ALGORITMA

```
{ di halaman berikutnya }
```

## Panjang Rata-Rata Kata - Model Akuisisi Kata Versi 3 (2)

### ALGORITMA

```
initAkses
lengthTotal ← 0
nbKata ← 0
while cc ≠ MARK do
    advKata
    lengthTotal ← lengthTotal + currentKata.length
    nbKata ← nbKata + 1
{ cc mencapai MARK }
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



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

## Program HitungWhile3

```
{ Menghitung banyaknya kata WHILE dalam pita karakter }  
{ Model akuisisi kata versi 3 }
```

## KAMUS

```
{ *** Mesin yang digunakan *** }  
USE MesinKata3
```

```
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 3 (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
initAkses
nWHILE ← 0
while cc ≠ MARK do
    ADVKATA
    if isKataEqual(kataWHILE, currentKata) then
        nWHILE ← nWHILE + 1
{ cc = MARK: mencapai akhir pita }
output (nWHILE)
```

```
function isKataEqual (k1, k2: Kata) → boolean
{ sama seperti pada model akuisisi versi 1 }
```