

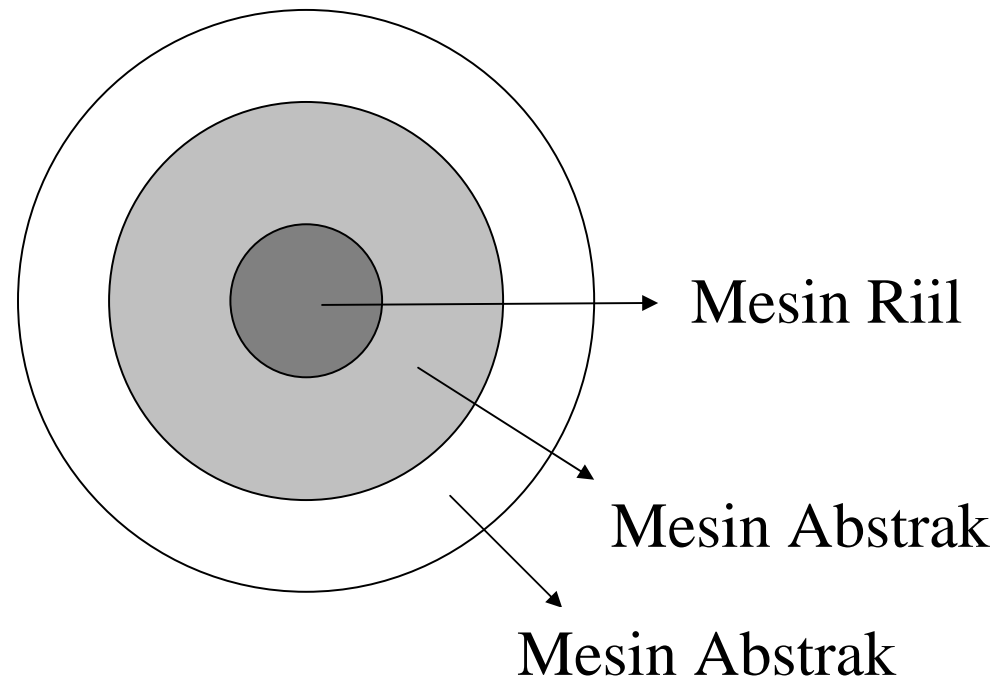


Mesin Karakter dan Mesin Kata

Tim Pengajar IF2030/Algoritma
dan Struktur Data

Mesin

- Mesin:
 - mekanisme yang terdefinisi dan mengerti serta mampu untuk mengeksekusi aksi-aksi primitif yang terdefinisi untuk mesin tersebut
- Mesin abstrak:
 - mesin yang dianggap ada dan diasumsikan mampu melakukan mekanisme yang didefinisikan untuk mesin tersebut
 - Mesin abstrak memodelkan suatu semesta (*universe*) tertentu





Mesin Abstrak

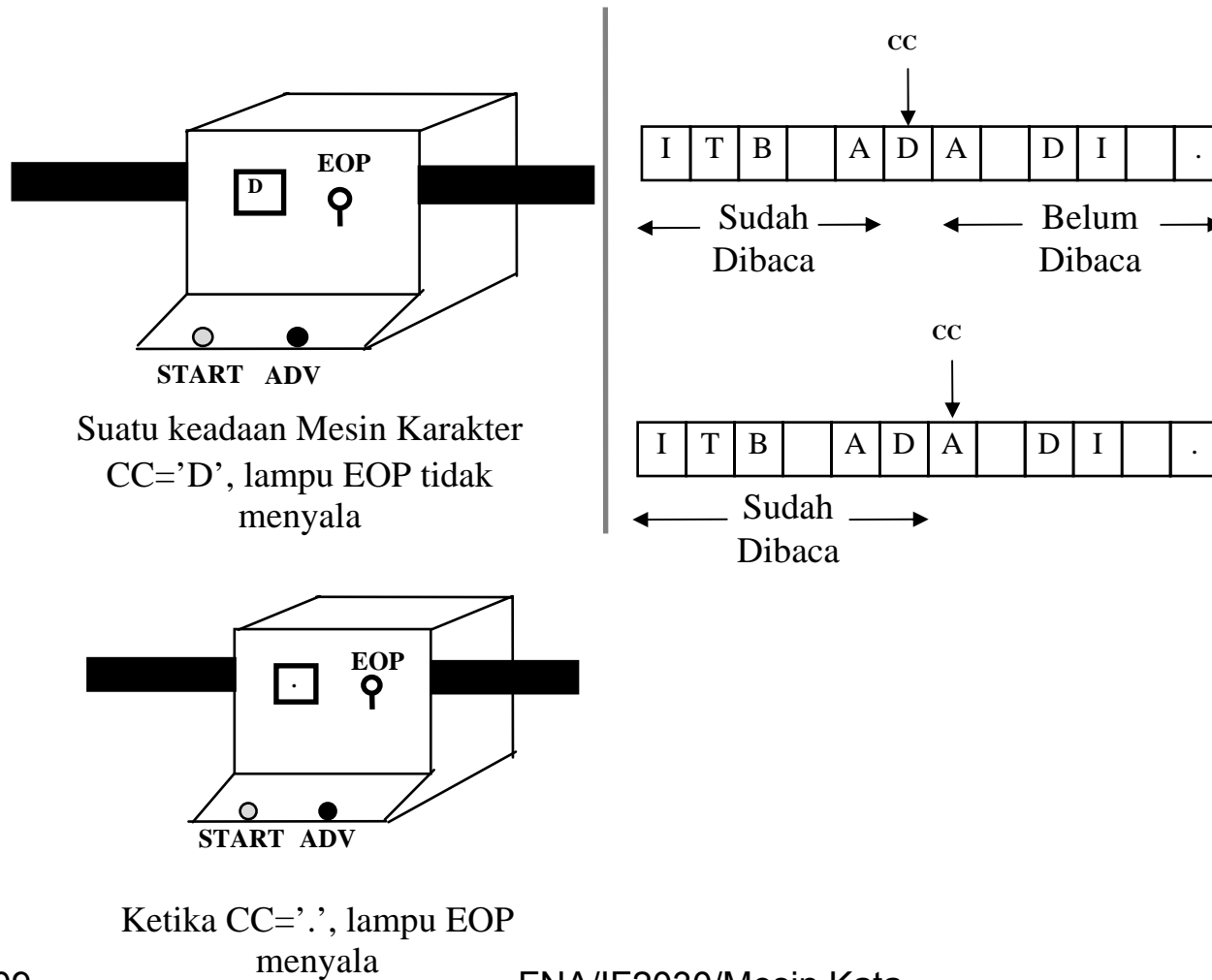
- Mendefinisikan mesin abstrak berarti mendefinisikan:
 - Sekumpulan **state** yang mungkin
 - Sekumpulan **aksi primitif** yang diasumsikan dapat dimengerti dan dieksekusi mesin yang bersangkutan
- Contoh mesin abstrak:
 - mesin gambar
 - mesin integer
 - mesin rekam
 - **mesin karakter**



Mesin Karakter (1)

- Mesin karakter adalah mesin abstrak yang terdiri atas:
 - Pita berisi deret karakter, diakhiri dengan '.' (titik)
 - Pita yang hanya berisi '.' disebut sebagai pita kosong,
 - Tombol **START** dan **ADV**
 - Lampu **EOP** (End Of Pita)
 - “Jendela” yang ukurannya sebesar satu karakter
 - Hanya karakter yang posisinya sedang pada jendela dapat dibaca; karakter lain tidak kelihatan
 - Karakter yang sedang pada jendela dinamakan **CC** (Current Character)
- State mesin karakter ditentukan oleh CC dan EOP

Mesin Karakter (2)



Mesin Karakter (3)

- Primitif untuk mengubah posisi pita

procedure START

```
{ Mesin siap dioperasikan. Pita disiapkan untuk dibaca.  
  Karakter pertama yang ada pada pita posisinya adalah pada  
  jendela  
  I.S. : sembarang  
  F.S. : CC adalah karakter pertama pada pita  
        Jika CC ≠ '.' maka EOP akan padam (false)  
        Jika CC = '.' maka EOP akan menyala (true) }
```

procedure ADV

```
{ Pita dimajukan satu karakter.  
  I.S. : Karakter pada jendela = CC, CC ≠ '.'  
  F.S. : CC adalah karakter berikutnya dari CC yang lama,  
        CC mungkin = '.'  
        Jika CC = '.' maka EOP akan menyala (true) }
```

EOP diwakili oleh boolean, bernilai **true** jika menyala; atau **false** jika tidak menyala. Jika **EOP** menyala, mesin sudah tidak dapat dioperasikan lagi.



Studi Kasus Mesin Karakter (1)

CountHuruf

- Diberikan sebuah mesin karakter dengan pita berisi karakter (mungkin kosong). Buatlah algoritma untuk menghitung banyaknya huruf yang ada pada pita tersebut. **Banyaknya** karakter pada pita kosong adalah nol.

Program COUNTHURUF

{ SKEMA PEMROSESAN DENGAN MARK :
menghitung banyaknya huruf pada pita karakter }

KAMUS

CI : integer

ALGORITMA

CI \leftarrow 0 { Inisialisasi }

START { First Elmt }

while (CC \neq '.') do { not EOP }

CI \leftarrow CI + 1 { Proses }

ADV { Next_Elmt }

{ CC = '.' }

output (CI) { Terminasi }

Studi Kasus Mesin Karakter (2)

Hitung-A



- Diberikan sebuah mesin karakter dengan pita berisi karakter (mungkin kosong), Buatlah algoritma untuk menghitung banyaknya huruf 'A' yang ada pada pita tersebut. Banyaknya karakter 'A' pada pita kosong adalah nol.

```
Program COUNT_A
{ SKEMA PEMROSESAN DENGAN MARK :
menghitung banyaknya huruf A pada pita karakter }
KAMUS
    CI : integer
ALGORITMA
    CI ← 0          { Inisialisasi, CI = 0 }
    START          { First_Elmt }
    while (CC ≠ '.') do    { not EOP }
        depend on CC      { Proses }
            CC = 'A' : CI ← CI + 1
            CC ≠ 'A' : -
        ADV          { Next_Elmt }
    { CC = '.' }
    output (CI)          { Terminasi }
```




Mesin Kata (1)

- Mesin Kata:
 - Mesin abstrak yang berdasarkan mesin karakter
- Diberikan sebuah mesin karakter dengan pita berisi karakter (mungkin kosong), yang diakhiri titik (‘.’)
- Kata:
 - sederetan karakter suksesif pada pita yang merupakan karakter bukan *blank*

Mesin Kata (2)

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

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





Studi Kasus Mesin Kata

Panjang Rata-Rata Kata

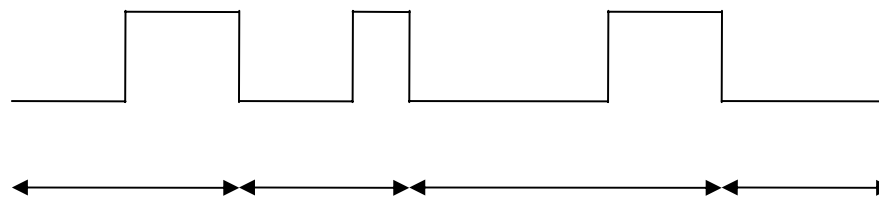
- Diberikan sebuah mesin karakter dengan pita berisi karakter (mungkin kosong), yang diakhiri titik, hitunglah panjang rata-rata kata yang ada pada pita tsb. Panjang kata rata-rata tidak terdefinisi jika pita kosong atau pita tidak mengandung kata (hanya berisi '*blank*' dan titik).



Panjang Rata-Rata Kata

Versi 1

- Akhir dari proses adalah sebuah **boolean**, yang akan berisi true jika kata terakhir telah diakuisisi dan diproses
- Kata diakuisisi mulai dari karakter pertama sesudah akhir kata (atau karakter pertama pita untuk kata pertama)
- Akuisisi kata terakhir menghasilkan 'kata kosong'.

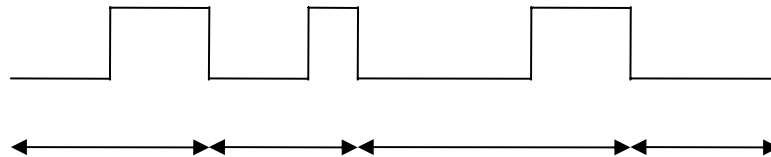


- Diktat Pemrograman Prosedural hlm. 172 s.d. 174

Panjang Rata-Rata Kata Versi 2



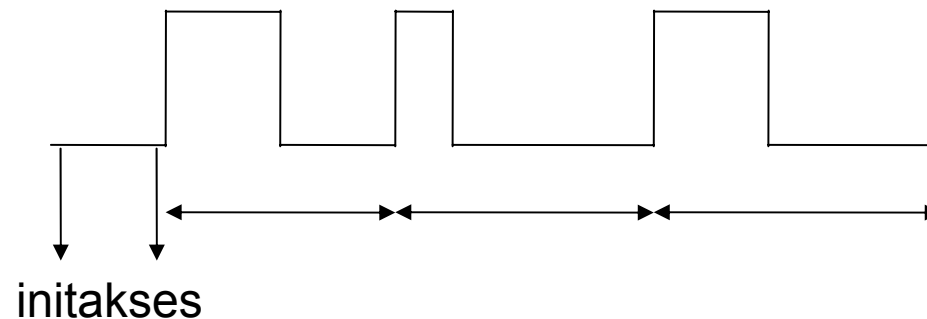
- Akhir dari proses adalah sebuah kata yang 'kosong', yaitu panjangnya NOL. Model akuisisi kata sama dengan Versi 1.



- Diktat Pemrograman Prosedural hlm. 175 s.d. 176

Panjang Rata-Rata 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
- Diktat Pemrograman Prosedural hlm. 177 s.d. 178

Mesin Kata dan Tabel

- Definisi type Kata:

```
type Kata : <TabKata : array [1..NMax] of character,  
                Length : integer >  
{ TabKata adalah tempat penampung/container kata,  
  Length menyatakan panjangnya kata }
```

- Mesin Kata:
 - Adaptasi salah satu versi akuisisi kata pada studi kasus Panjang Rata-Rata Kata



Mesin Kata

Akuisisi Kata - Versi 3 (1)

KAMUS UMUM

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

```
use MSNKAR
```

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

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

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

```
constant NMax : integer = 50
```

```
{ jumlah maksimum karakter suatu kata }
```

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

```
type Kata : < TabKata : array [1..NMax] of character,  
Length : integer >
```

```
{ TabKata adalah tempat penampung/container kata,  
Length menyatakan panjangnya kata }
```




Mesin Kata

Akuisisi Kata - Versi 3 (2)

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

```
procedure Ignore_Blank
```

```
{ Mengabaikan satu atau beberapa BLANK }
```

```
{ I.S. : CC sembarang }
```

```
{ F.S. : CC  $\neq$  BLANK atau CC = MARK }
```

```
procedure INITAKSES
```

```
{ Mengabaikan satu atau beberapa BLANK di awal pita karakter }
```

```
{ I.S. : CC sembarang }
```

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



Mesin Kata

Akuisisi Kata - Versi 3 (3)

```
procedure ADVKATA (output CKata : Kata)
{ I.S. : CC adalah karakter pertama kata yang akan diakuisisi }
{ F.S. : CKata adalah kata terakhir yang sudah diakuisisi,
        CC adalah karakter pertama dari kata berikutnya,
        mungkin MARK }
{ Proses : Akuisisi kata menggunakan procedure SalinKata }

procedure SalinKata (output CKata : Kata)
{ Mengakuisisi kata, menyimpan dalam CKata }
{ I.S. : CC adalah karakter pertama dari kata }
{ F.S. : CKata berisi kata yang sudah diakuisisi,
        jika karakternya melebihi NMax, sisa "kata" dibuang;
        CC = BLANK atau CC = MARK;
        CC adalah karakter sesudah karakter terakhir yang
        diakuisisi }
```

Studi Kasus Mesin Kata Hitung While



- Diberikan suatu pita karakter yang mengandung abjad, blank, dan diakhiri titik, harus dicari kemunculan kata 'while' pada pita tersebut
- Diktat Pemrograman Prosedural hlm. 182 s.d. 185



Fungsi/Prosedur Lain

- Fungsi KataSama
- Hitung “while”
- Palindrom
- Anagram
- Frekuensi kata pertama
- DII.



Mesin Karakter dan Mesin Kata

Dalam Bahasa C

mesinkar.h



```
#ifndef __MESIN_KAR__
#define __MESIN_KAR__

#include "boolean.h"

extern char CC;
extern boolean EOP;

void START();
/* Mesin siap dioperasikan. Pita disiapkan untuk dibaca.
   Karakter pertama yang ada pada pita posisinya adalah pada jendela.
   I.S. : sembarang
   F.S. : CC adalah karakter pertama pada pita
          Jika CC != '.' maka EOP akan padam (false)
          Jika CC = '.' maka EOP akan menyala (true) */

void ADV();
/* Pita dimajukan satu karakter.
   I.S. : Karakter pada jendela = CC, CC != '.'
   F.S. : CC adalah karakter berikutnya dari CC yang lama,
          CC mungkin = '.'
          Jika CC = '.' maka EOP akan menyala (true) */

#endif
```

mesinkar.c



```
#include <stdio.h>
#include "boolean.h"
#include "mesinkar.h"

char CC;
boolean EOP;
FILE *pita;

void START() {
    /* Mesin siap dioperasikan. Pita
       disiapkan untuk dibaca.
       Karakter pertama yang ada pada
       pita posisinya adalah pada
       jendela.
       I.S. : sembarang
       F.S. : CC adalah karakter pertama
              pada pita
              Jika CC != '.' maka EOP
              akan padam (false)
              Jika CC = '.' maka EOP
              akan menyala (true) */

    pita = fopen("pitakar.txt","r");
    ADV();
}
```

```
void ADV() {
    /* Pita dimajukan satu karakter.
       I.S. : Karakter pada jendela =
              CC, CC != '.'
       F.S. : CC adalah karakter
              berikutnya dari CC yang
              lama,
              CC mungkin = '.'
              Jika CC = '.' maka EOP
              akan menyala (true) */
    fscanf(pita,"%c",&CC);
    EOP = (CC=='.' );
    if (EOP)
        fclose(pita);
}
```

mesinkata.h (model akuisisi v.2)



```
#ifndef __MESINKATA_H__
#define __MESINKATA_H__

#include "mesinkar.h"

typedef struct {
    char TabKata[100];
    int Length;
} Kata;

extern Kata KT;

void IgnoreBlank();
/* Mengabaikan satu atau beberapa
   BLANK
   I.S. : CC sembarang
   F.S. : CC != BLANK atau
         CC = MARK */
void STARTKATA();
/* I.S. : CC sembarang */
/* F.S. : KT.Length = 0,
         dan CC = Mark;
         atau KT.Length != 0,
         KT adalah kata yang sudah
         diakuisisi,
         CC karakter pertama
         sesudah karakter terakhir
         kata */
```

```
void ADVKATA();
/* I.S. : KT.Length != 0;
         CC adalah karakter sesudah
         karakter terakhir dari
         kata yang sudah diakuisisi
   F.S. : Jika CC = MARK, maka
         KT.Length = 0;
         atau KT.Length != 0,
         KT adalah kata terakhir
         yang sudah diakuisisi;
         CC karakter pertama
         sesudah karakter terakhir
         kata */

#endif
```


mesinkata.c (model akuisisi v.2)



```
#include "mesinkar.h"
#include "mesinkata.h"

Kata KT;

void IgnoreBlank() {
/* Mengabaikan satu atau beberapa
   BLANK
   I.S. : CC sembarang
   F.S. : CC != BLANK atau
          CC = MARK */
  while (!EOP && CC==' ')
    ADV();
}

void STARTKATA() {
/* I.S. : CC sembarang */
  F.S. : KT.Length = 0,
        dan CC = Mark;
        atau KT.Length != 0,
        KT adalah kata yang sudah
        diakuisisi,
        CC karakter pertama
        sesudah karakter terakhir
        kata */
  START();
  ADVKATA();
}
```

```
void ADVKATA() {
/* I.S. : KT.Length != 0;
        CC adalah karakter sesudah
        karakter terakhir dari
        kata yang sudah diakuisisi
   F.S. : Jika CC = MARK, maka
          KT.Length = 0;
          atau KT.Length != 0,
          KT adalah kata terakhir
          yang sudah diakuisisi;
          CC karakter pertama
          sesudah karakter terakhir
          kata */
  IgnoreBlank();
  KT.Length=0;
  while (!EOP && CC!=' ') {
    KT.TabKata[KT.Length]=CC;
    KT.Length++;
    KT.TabKata[KT.Length]=0;
    ADV();
  }
}
```



mainkata.c

```
#include <stdio.h>
#include "mesinkata.h"

int main() {
    STARTKATA();
    while (KT.Length>0) {
        printf("%s\n",KT.TabKata);
        ADVKATA();
    }
    return 0;
}
```



Cara Kompilasi

```
$ cc -c mesinkar.c  
$ cc -c mesinkata.c  
$ cc -c mainkata.c  
$ cc -o mainkata mesinkar.o mesinkata.o mainkata.o
```

Atau cara lain:

```
$ cc -o mainkata mesinkar.c mesinkata.c mainkata.c
```