Mesin Abstrak dan Mesin Karakter

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

Mesin

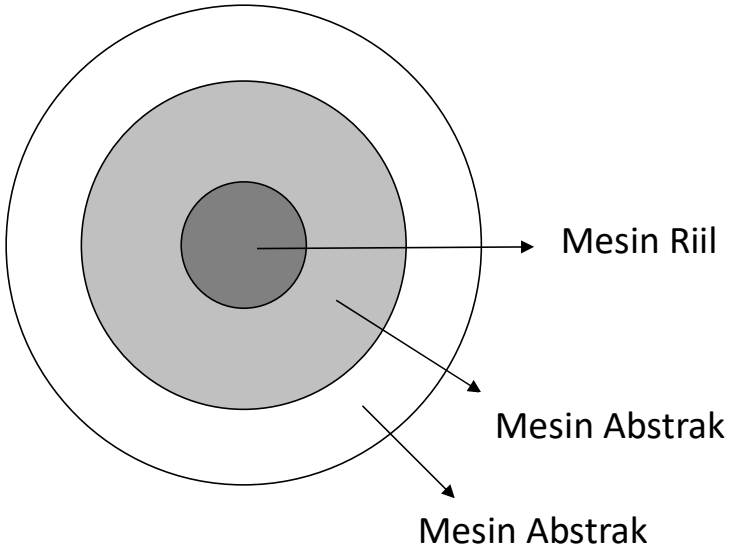
Mesin:

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

Mesin abstrak 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

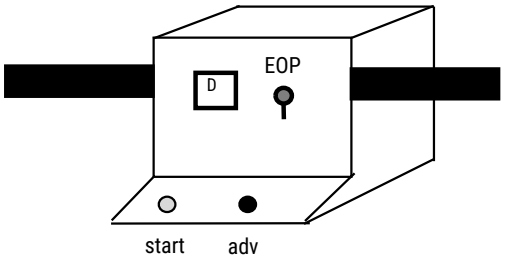
Mesin Karakter

Mesin Karakter (1)

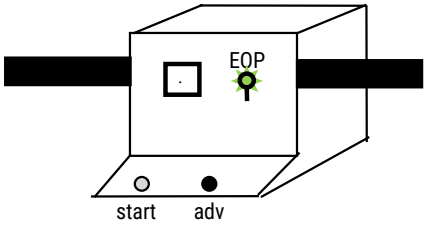
Terdiri atas:

- Pita berisi deret karakter, diakhiri dengan **MARK** berupa '.' (titik)
  - Pita hanya berisi MARK → pita kosong
- Tombol **start** dan **adv**
- "**Jendela**" → ukuran sebesar satu karakter
  - **CC** (*Current Character*) → karakter yang sedang tampak di jendela
- Lampu **EOP** (*End of Pita*)

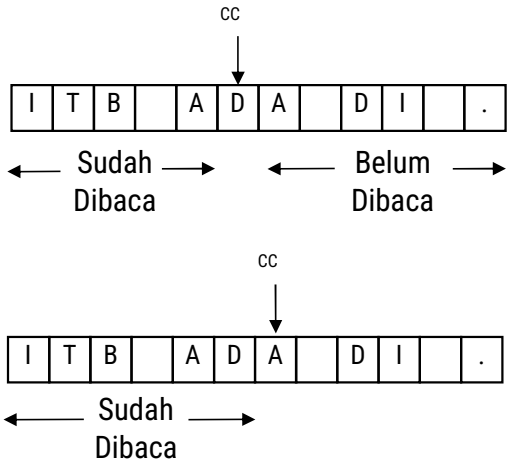
State mesin karakter ditentukan oleh **CC** dan **EOP**



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



Ketika CC='.', lampu EOP menyala



Mesin Karakter (2)

Primitif terkait 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 ≠ MARK maka eop akan padam (false)
        Jika cc = MARK maka eop akan menyala (true) }

procedure adv
{ Pita dimajukan satu karakter.
  I.S.: Karakter pada jendela = cc, cc ≠ MARK
  F.S.: cc adalah karakter berikutnya dari cc yang lama,
        cc mungkin = MARK
        Jika  cc = MARK 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) - CountKarakter

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 CountCharacters
{ SKEMA PEMROSESAN DENGAN MARK:
  menghitung banyaknya huruf pada pita karakter }
KAMUS
    ctr: integer
ALGORITMA
    ctr ← 0                 { Inisialisasi }
    start                   { First Elmt }
    while (cc ≠ MARK) do    { not EOP }
        ctr ← ctr + 1       { Proses }
        adv                 { Next Elmt }
    { cc = MARK }
    output(ctr)             { 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 CountA
{ SKEMA PEMROSESAN DENGAN MARK:
  menghitung banyaknya huruf A pada pita karakter }
KAMUS
    ctr: integer
ALGORITMA
    ctr ← 0                 { Inisialisasi, CI = 0 }
    start                   { First Elmt }
    while (cc ≠ MARK) do    { not EOP }
      if cc = ‘A’ then      { Proses }
        ctr ← ctr + 1
      adv                   { Next Elmt }
    { cc = MARK }
    output(ctr)             { Terminasi }
```

Mesin Karakter

Dalam Bahasa C

mesinkar.h

```
#ifndef __MESIN_KAR__
#define __MESIN_KAR__

#include "boolean.h"

#define MARK '.'

/* State Mesin */
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 != MARK maka eop akan padam (false)
         Jika cc = MARK maka eop akan menyala (true) */
void adv();
/* Pita dimajukan satu karakter.
   I.S.: Karakter pada jendela = cc, cc != MARK
   F.S.: cc adalah karakter berikutnya dari cc yang lama,
         cc mungkin = MARK
         Jika cc = MARK maka eop akan menyala (true) */

#endif
```

mesinkar.c

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

char cc;
boolean eop;

static FILE *pita;
static int retval;

void start() {
/* Mesin siap dioperasikan. Pita disiapkan ... */
  /* Algoritma */
  pita = fopen("pitakar.txt","r");
  adv();
}

void adv() {
/* Pita dimajukan satu karakter. ... */
  /* Algoritma */
  retval = fscanf(pita,"%c",&cc);
  eop = (cc == MARK);
  if (eop) {
    fclose(pita);
  }
}
```

Latihan-1: Hitung-LE

Diberikan sebuah mesin karakter dengan pita berisi karakter (mungkin kosong), Buatlah algoritma untuk menghitung banyaknya pasangan huruf ‘L‘ dan ‘E’ yang ada pada pita tersebut. Banyaknya pasangan huruf ‘L‘ dan ‘E’ pada pita kosong adalah nol.

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

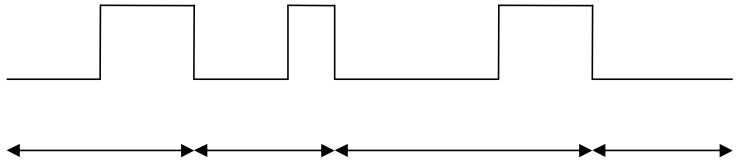
f. Mengandung blank di awal pita

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)



- 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)
    { 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 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 }
```

Mesin Kata dalam Bhs C

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

mesinkata1.h (model akuisisi v.1)

```
#ifndef MESINKATA1_H
#define MESINKATA1_H

#include "mesinkar.h"

#define N_MAX 50
#define BLANK ' '

typedef struct {
  char buffer[N_MAX];
  int length;
} Kata;

/* State Mesin Kata */
extern boolean endKata;
extern Kata currentKata;

void ignoreBlank();
/* Mengabaikan satu atau beberapa BLANK
   I.S.: cc sembarang
   F.S.: cc ≠ BLANK atau cc = MARK */
```

mesinkata1.h (model akuisisi v.1)

```
void 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 */

void 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 */

void 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 */

#endif
```

mesinkata1.c (model akuisisi v.1)

```c
#include "mesinkata1.h"

boolean endKata;
Kata currentKata;

void ignoreBlank() {
/* Mengabaikan satu atau beberapa BLANK
   I.S.: cc sembarang
   F.S.: cc ≠ BLANK */

  /* Kamus Lokal */

  /* Algoritma */
  while (cc == BLANK) {
    adv();
  } /* cc != BLANK */
}
```

mesinkata1.c (model akuisisi v.1)

```
void 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) {
    endKata = true;
  } else /* cc != MARK */ {
    endKata = false;
    salinKata();
  }
}
```

mesinkata1.c (model akuisisi v.1)

```
void 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) {
    endKata = true;
  } else /* cc != MARK */ {
    salinKata();
  }
}
```

mesinkata1.c (model akuisisi v.1)

```
void 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 */
  int i; /* definisi */
  /* Algoritma*/
  i = 0; /* inisialisasi */
  while ((cc != MARK) && (cc != BLANK)) {
    currentKata.buffer[i] = cc;
    adv();
    i++;
  } /* cc = MARK or cc = BLANK */
  currentKata.length = i;
}
```

mainkata.c

```c
#include <stdio.h>
#include "mesinkata1.h"

int main() {
  startKata();
  while (!endKata) {
    for (int i=0;i<currentKata.length;i++) {
      printf("%c",currentKata.buffer[i]);
    }
    printf("\n");
    advKata();
  }
  return 0;
}
```

Cara Kompilasi

```
$ cc -c mesinkar.c
$ cc -c mesinkata1.c
$ cc -c mainkata.c
$ cc -o mainkata mesinkar.o mesinkata1.o mainkata.o
```

Atau cara lain:

```
$ cc -o mainkata mesinkar.c mesinkata1.c mainkata.c
```

Mesin Kata - Model Akuisisi versi 2 dan 3

Model akuisisi versi 2

Seperti versi 1, namun akhir dari proses akuisisi adalah kata ‘kosong’

Model akuisisi versi 3

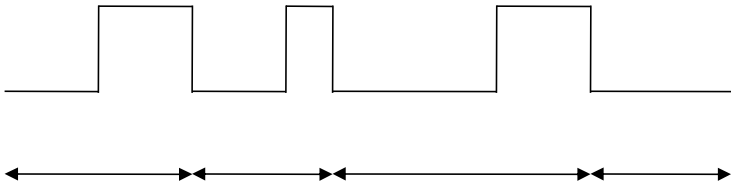
Mengabaikan blank pada awal pita dan memroses sisanya → model akuisisi tanpa mark

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)

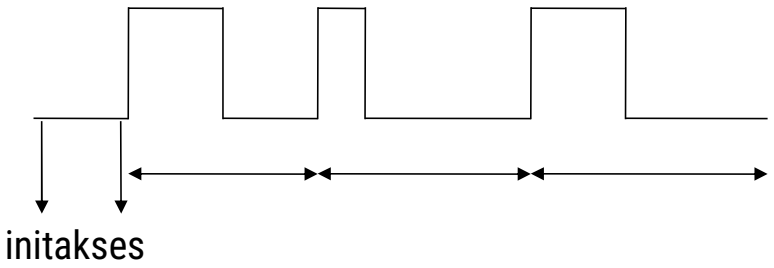
```
ALGORITMA
    { Inisialisasi kataWHILE }
    kataWHILE.buffer[0] ← 'W'
    kataWHILE.buffer[1] ← 'H'
    kataWHILE.buffer[2] ← 'I'
    kataWHILE.buffer[3] ← 'L'
    kataWHILE.buffer[4] ← 'E'
    kataWHILE.length ← 5
    NWHILE ← 0
    startKata
    while currentKata.length ≠ 0 do
        if isKataEqual(kataWHILE, 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 ← 0
    repeat
        currentKata.buffer[i] ← cc
        adv
        i ← i + 1
    until (cc = MARK) or (cc = BLANK)
    { 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 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 }
```

Latihan Soal
Mesin Kata

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

Soal 1: Frekuensi Kata Pertama

Dibaca sebuah pita karakter yang diakhiri titik. Hitunglah frekuensi kemunculan kata pertama dalam pita tersebut.

Andaikata teks selalu dalam bahasa Indonesia, dan kata terpanjang dalam bahasa Indonesia terdiri dari 50 karakter.

Contoh: “aku pergi ke pasar kemudian aku pulang ke rumah supaya aku dapat mandi.” Sehingga frekuensi 'aku' adalah 3/13.

Jika pita karakter kosong, memberikan pesan bahwa pita karakter kosong.

Soal 2: Anagram

Buatlah sebuah program yang membaca sebuah pita karakter, dan menuliskan ke layar ada berapa banyak kata yang ANAGRAM dengan kata pertama pada pita karakter tersebut (tidak termasuk kata pertama).

Kata dikatakan anagram jika memiliki panjang yang sama, terdiri atas huruf yang sama dan masing-masing huruf memiliki jumlah yang sama.

Tuliskanlah sebuah fungsi yang menerima masukan dua buah Kata yang direpresentasi dalam array, memeriksa apakah kedua kata itu anagram dan mengirimkan sebuah harga boolean (true jika kedua kata anagram, false jika tidak).

Contoh:

SEBAB dan BEBAS adalah anagram
BAGUS dan GABUS adalah anagram
SUPER dan PUSER adalah anagram.

Soal 3: Mesin Token

Sebuah pita karakter berisi ekspresi matematika dalam notasi postfix, dan diakhiri dengan karakter titik '.'

Contoh isi pita karakter: 12 3 * 4 8 + -.
Dalam notasi infix: (12 * 3) – (4 + 8)

Setiap rangkaian karakter yang membentuk angka (operan) dan operator (*, /, +, -, ^) disebut sebagai token.

Setiap token dipisahkan oleh 1 atau lebih BLANK (spasi).

Buatlah Mesin Token dengan memodifikasi Mesin Kata (pilih salah satu versi model akuisisi kata).

Buatlah sebuah driver Mesin Token yang menuliskan ekspresi matematika dalam pita karakter ke layar.