


# Array 1

# Pengantar









- ▶ Sebuah peubah hanya menyimpan sebuah nilai.
- ▶ Ia tidak dapat menyimpan beberapa buah nilai yang bertipe sejenis.
- ▶ Sekumpulan data yang bertipe sama dapat disimpan secara beruntun di dalam memori komputer, setiap elemen data diacu dengan menggunakan indeks.
- ▶ Indeks menyatakan posisi data relatif di dalam kumpulan.
- ▶ Struktur penyimpanan data seperti ini dinamakan *larik (array)*.

# Mengenai Array

- ▶ **Array** adalah struktur data yang menyimpan sekumpulan elemen yang bertipe sama.
  - ▶ Setiap elemen diakses langsung melalui indeksnya.
  - ▶ Indeks array haruslah tipe data yang menyatakan keterurutan, misalnya integer atau karakter.
- 

# Ilustrasi Array

- ▶ Misalkan kita ingin menyimpan data tinggi badan 8 orang siswa (dalam cm).
- ▶ Kita definisikan sebuah array bernama A dengan delapan buah elemen.
- ▶ Tinggi siswa pertama, 158 disimpan dalam A[1], tinggi siswa kedua, 157 disimpan dalam A[2], dst.

A		
1		158
2		157
3		162
4		169
5		172
6		155
7		170
8		163

# Mendeklarasikan Array

- ▶ Array adalah struktur yang statis, artinya jumlah elemen array harus sudah diketahui sebelum program dieksekusi.
- ▶ Jumlah elemen array tidak dapat diubah, ditambah, atau dikurangi selama pelaksanaan program.
- ▶ Deklarasi array:
  - Mendefinisikan banyaknya elemen array (ukuran array)
    - Komputer mengalokasikan sejumlah lokasi memori sebanyak elemen array yang bersangkutan.
  - Mendefinisikan tipe elemen array
    - Menetapkan tipe nilai yang dapat disimpan oleh array.

# Contoh Array sebagai Peubah

Misalkan:

- ▶ A adalah *array* yang berukuran 50 buah elemen yang bertipe *integer*. Indeks array dimulai dari 1.
- ▶ NamaMhs adalah *array* yang berukuran 10 buah elemen yang bertipe *string*. Indeks *array* dimulai dari 1.
- ▶ NilUjian adalah peubah *array* yang berukuran 75 buah elemen yang bertipe *real*. Indeks *array* dimulai dari 0.

## DEKLARASI

```
A           : array[1..50] of integer
NamaMhs     : array[1..10] of string
NilUjian    : array[0..74] of real
```

# Contoh Array sebagai Tipe Bentuk

- Misalkan `ArrayInt` didefinisikan sebagai nama sebuah tipe baru untuk *array* yang bertipe *integer*.
- Ukuran *array* adalah 100 buah elemen.
- Kita bisa mendeklarasikan sebuah peubah yang bertipe `ArrayInt`, misalnya peubah `A`.

## DEKLARASI

```
type ArrayInt : array[1..100] of integer {nama tipe baru}  
A : ArrayInt {A adalah sebuah peubah array integer dengan  
100 elemen}
```

# Contoh Array sebagai sebuah Konstanta

- ▶ Misalkan `ArrayInt` dideklarasikan sebagai tipe bentukan untuk *array* yang bertipe *integer*.
- ▶ Ukuran maksimum *array* adalah 100 buah elemen.
- ▶ Ukuran maksimum *array* dinyatakan sebagai konstanta.
- ▶ Kemudian kita bisa mendeklarasikan sebuah peubah yang bertipe `ArrayInt`, misalnya peubah A.

## DEKLARASI

```
const Nmaks = 100 {ukuran maksimum elemen array}  
type ArrayInt : array[1..Nmaks] of integer  
A : ArrayInt {A adalah sebuah peubah array integer  
dengan 100 elemen}
```



# Pendeklarasian Array yang salah

DEKLARASI

A : array[1..N] of integer

- Pendeklarasian *array* di atas tidak diperbolehkan karena nilai  $N$  tidak diketahui di awal program.
- Ingat bahwa *Array* adalah struktur statis.

# Cara Mengacu Elemen Array

- ▶ Elemen array diacu melalui indeksnya.
- ▶ Contoh:

<code>A[4]</code>	<i>{mengacu elemen keempat dari array A}</i>
<code>NamaMhs[2]</code>	<i>{mengacu elemen kedua dari array NamaMhs}</i>
<code>A[i]</code>	<i>{mengacu elemen ke-i dari array A, asalkan nilai i sudah terdefinisi}</i>
<code>NamaMhs[i+1]</code>	<i>{mengacu elemen ke-i+1 dari array NamaMhs, asalkan nilai i sudah terdefinisi}</i>

# Cara Memanipulasi atau Menggunakan Elemen Array

## ► Contoh :

`A[4] ← 10`

*{mengisi elemen keempat dari array A dengan nilai 10}*

`NamaMhs[i] ← 'Achmad'`

*{mengisi elemen ke-i dari array NamaMhs dengan string 'Achmad'}*

`read(A[i])`

*{membaca elemen ke-i dari array A}*

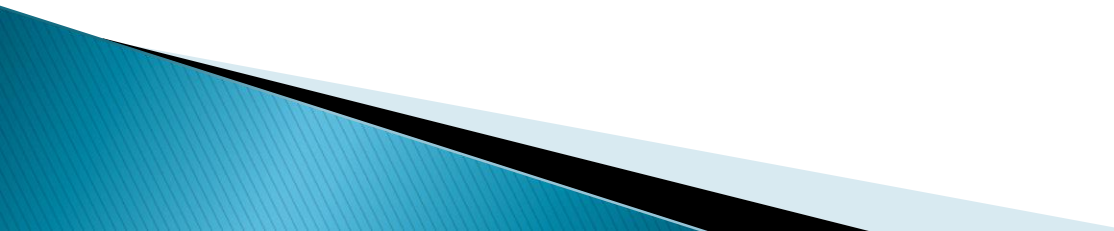
`if A[i] < 10 then`

`A[i] ← A[i] + 10`

`else`

`... {pernyataan lainnya}`

# Pemrosesan Array

- ▶ Elemen array tersusun di memori secara beruntun (sekuensial).
  - ▶ Elemen array diproses secara beruntun melalui indeksnya yang terurut.
  - ▶ Memproses array artinya mengunjungi (traversal) setiap elemen array dan memanipulasi nilai di dalamnya.
  - ▶ Kunjungan dimulai dari elemen pertama array, berturut-turut pada elemen berikutnya, sampai elemen terakhir dicapai, yaitu elemen dengan indeks terbesar.
- 

# Skema Umum Pemrosesan Array

PROGRAM PemrosesanArray

{Skema pemrosesan array secara beruntun}

DEKLARASI

const Nmaks = 100 {ukuran maksimum array}

type ArrayInt : array[1..Nmaks] of integer

A : ArrayInt

I : integer {indeks array}

ALGORITMA

*inisialisasi*

i ← 1 {mulai dari elemen pertama}

while i ≤ Nmaks do

*pemrosesan terhadap A[i]*

i ← i + 1 {tinjauan elemen berikutnya}

endwhile

{i > Nmaks}

*Terminasi*

Merupakan aksi spesifik bergantung pada persoalan yang akan dipecahkan. Misalnya pengisian nilai, pembacaan, penulisan, komputasi, atau manipulasi lainnya.

# Skema Umum Pemrosesan Array (2)

```
PROGRAM PemrosesanArray  
{Skema pemrosesan array secara beruntun}
```

```
DEKLARASI
```

```
const Nmaks      = 100 {ukuran maksimum array}  
type ArrayInt   : array[1..Nmaks] of integer
```

```
A : ArrayInt  
I : integer      {indeks array}
```

```
ALGORITMA
```

```
for i  $\leftarrow$  1 to Nmaks do  
    pemrosesan terhadap A[i]  
endfor
```

Untuk memproses array, lebih baik menggunakan **struktur FOR** karena jumlah elemen array sudah diketahui di awal.

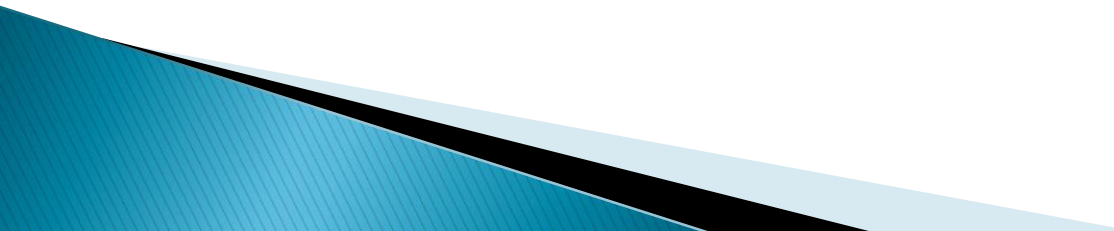
# Ukuran Efektif Array

- ▶ Ukuran Efektif *Array* adalah banyaknya elemen *array* yang digunakan.
- ▶ Misalkan ukuran maksimum *array* yang disajikan adalah 100, dan ukuran efektif *array* yang digunakan kita simpan dalam peubah *n*.

## DEKLARASI

```
const Nmaks = 100 {ukuran maksimum array}
type ArrayInt : array[1..Nmaks] of integer
A             : arrayInt
n             : integer {mencatat ukuran array
                        yang digunakan}
```

# Menginisialisasi Array

- ▶ Menginisialisasi Array adalah memberikan nilai awal untuk seluruh elemen array atau mungkin sebagian saja.
  - ▶ Inisialisasi array mungkin diperlukan, misalnya “mengosongkan” (memberikan nilai nol atau karakter kosong(*null*)) elemen array sebelum dipakai untuk proses tertentu, bergantung pada permasalahan yang akan dipecahkan.
- 



# Menginisialisasi Elemen Array dengan nilai 0

```
procedure InisDengan0(output A : ArrayInt, input n: integer)  
    {menginisialisasi setiap elemen array A[1..N] dengan nol}  
    {K. Awal : n adalah jumlah elemen efektif array, nilainya  
    terdefinisi}  
    {K. Akhir : seluruh elemen array A bernilai nol}
```

DEKLARASI

```
i : integer {pencatat indeks array}
```

ALGORITMA

```
for i  $\leftarrow$  1 to n do  
    A[i]  $\leftarrow$  0  
endfor
```

# Contoh Program Pemanggil

## Tipe Data Array

PROGRAM PemrosesanArray

{Program untuk mengisi elemen array dengan nilai 0}

DEKLARASI

const Nmaks = 1 {ukuran maksimum array}

type ArrayInt : array[1..Nmaks] of integer

A : ArrayInt

i : integer {indeks array}

n : integer {ukuran efektif array}

Procedure InisDengan0(output A : ArrayInt, input n : integer)

ALGORITMA:

read(n) {tentukan jumlah elemen yang akan digunakan dengan syarat  $1 \leq n \leq Nmaks$ }

InisDengan0(A, n)

{cetak hasil inisialisasi}

for i  $\leftarrow$  1 to n do

write(A[i])

endfor

# Hasil inisialisasi array A dengan 0

A	
1	0
2	0
3	0
:	0
:	0
	0
:	0
n	0

# Menginisialisasi Elemen Array dengan nilai i

```
procedure InisDengan(output A : ArrayInt, input n: integer)  
    {menginisialisasi setiap elemen array A[1..N] dengan nilai i =  
    1, 2, 3, ... n}  
    {K. Awal : n adalah jumlah elemen efektif array, nilainya  
    terdefinisi}  
    {K. Akhir : A[1] = 1, A[2] = 2, ... A[n] = n}
```

DEKLARASI

```
i : integer {pencatat indeks array}
```

ALGORITMA

```
for i  $\leftarrow$  1 to n do  
    A[i]  $\leftarrow$  i  
endfor
```

# Hasil inisialisasi array A dengan nilai i

A	
1	1
2	2
3	3
:	
:	
:	
:	
n	n

# Mengisi Elemen Array dengan Pembacaan

Versi 1 : Jika jumlah elemen efektif ditentukan di awal :

```
procedure BacaArray1(output A : ArrayInt, input n: integer)  
    {mengisi setiap elemen array A[1..N] dengan pembacaan}  
    {K. Awal : n adalah jumlah elemen efektif array, nilainya  
    terdefinisi}  
    {K. Akhir : setelah pembacaan, seluruh elemen array A  
    berisi nilai-nilai yang dibaca dari piranti masukan}
```

DEKLARASI

```
    i : integer {pencatat indeks array}
```

ALGORITMA

```
    for i  $\leftarrow$  1 to n do  
        read(A[i])  
    endfor
```

# Mengisi Elemen Array dengan Pembacaan

Versi 2 : Jika jumlah elemen efektif baru diketahui di akhir pembacaan

```
procedure BacaArray2(output A : ArrayInt, output n: integer)  
    {mengisi setiap elemen array A[1..N] dengan pembacaan}  
    {K. Awal : sembarang  
    {K. Akhir : sebanyak n buah elemen array A berisi nilai-nilai  
    yang dibaca; n berisi jumlah elemen array yang diisi.
```

DEKLARASI

```
    jawab : char
```

ALGORITMA

```
    n  $\leftarrow$  0
```

```
    repeat
```

```
        n  $\leftarrow$  n + 1
```

```
        read(A[n])
```

```
        write('Lagi? (y/t)')
```

```
        read(jawab)
```

```
    until jawab='t'
```

# Mencetak Elemen Array

```
procedure CetakArray(input A : ArrayInt, input n : integer)  
    {mencetak elemen-elemen array A[1..n]}  
    {K. Awal : n sudah berisi jumlah elemen array yang dipakai.  
    Elemen-elemen array A sudah terdefinisi}  
    {K. Akhir : elemen-elemen array A tercetak }
```

DEKLARASI

```
i : integer {pencatat indeks array}
```

ALGORITMA

```
for i  $\leftarrow$  1 to n do  
    write (A[i])  
endfor
```



# Menghitung Nilai Rata-Rata

```
procedure HitungRataRata(input A : ArrayInt, input n : integer,  
output u : real)
```

```
    {menghitung nilai rata-rata elemen array A}
```

```
    {K. Awal : Elemen-elemen array A sudah terdefinisi; n adalah jumlah elemen  
    array, nilainya sudah terdefinisi}
```

```
    {K. Akhir : u berisi nilai rata-rata seluruh elemen }
```

DEKLARASI

```
    i : integer           {pencatat indeks array}
```

```
    jumlah : real        {jumlah total seluruh nilai}
```

ALGORITMA

```
    i  $\leftarrow$  1           {dimulai dari elemen pertama}
```

```
    jumlah  $\leftarrow$  0      {jumlah total nilai mula-mula}
```

```
    for i  $\leftarrow$  1 to n do
```

```
        jumlah  $\leftarrow$  jumlah + A[i]
```

```
    endfor
```

```
    U  $\leftarrow$  jumlah/N
```

# Pemanggil Prosedur Menghitung Nilai Rata-Rata

PROGRAM Rerata

{Program utama untuk menghitung nilai rata-rata seluruh elemen array}

DEKLARASI

const Nmaks = 100 {ukuran maksimum array}

type ArrayInt : array[1..Nmaks] of integer

A : ArrayInt

n : integer

u : real

procedure BacaArray1(output A: ArrayInt, input n : integer)

*{mengisi elemen array A[1..n] dengan pembacaan}*

procedure HitungRataRata(input A: ArrayInt, input n: integer, output

u : real) *{menghitung nilai rata-rata array A}*

ALGORITMA:

read(n) *{tentukan jumlah elemen array yang akan digunakan}*

BacaArray1(A, n) *{baca elemen-elemen array}*

HitungRataRata(A, n, u) *{hitung nilai rata-rata}*

write(u)

# Kapan Menggunakan Array?



- ✓ *... jika kita mempunyai sejumlah data yang bertipe sama yang perlu disimpan sementara untuk kemudian diproses.*
- ✓ *Dengan array, kita dapat menghindari penggunaan nama-nama peubah yang banyak.*

# Contoh Menghitung Nilai Rata-Rata tanpa Array

PROGRAM HitungRerata\_TanpaArray

{Program yang membaca 6 buah data,  
mencetaknya, dan menghitung rata-  
ratanya. Tanpa menggunakan array}

DEKLARASI

x1, x2, x3, x4, x5, x6 : integer

u : real

ALGORITMA:

{Baca 6 buah nilai integer, simpan  
di x1, x2,..., x6}

read(x1)

read(x2)

read(x3)

read(x4)

read(x5)

read(x6)

{cetak nilai x1, x2,..., x6}

write(x1)

write(x2)

write(x3)

write(x4)

write(x5)

write(x6)

{hitung nilai rata-rata}

$u \leftarrow (x1+x2+x3+x4+x5+x6) / 6$

write(u)

# Contoh Menghitung Nilai Rata-Rata dengan Array

PROGRAM HitungRerata\_DenganArray

{Program yang membaca 6 buah data,  
mencetaknya, dan menghitung rata-  
ratanya}

DEKLARASI

x : array[1..6] of integer

i, jumlah : integer

u : real

ALGORITMA:

{Baca 6 buah nilai integer, simpan  
di x[i]= 1, 2,..., 6}

for i  $\leftarrow$  1 to 6 do

read(x[i])

endfor

{cetak nilai x[i]=1,2,...,6}

for i  $\leftarrow$  1 to 6 do

write(x[i])

endfor

{hitung nilai rata-rata}

jumlah  $\leftarrow$  0

for i  $\leftarrow$  1 to 6 do

jumlah  $\leftarrow$  jumlah + x[i]

endfor

u  $\leftarrow$  jumlah/6

write(u)

# Mencari Nilai Maksimum Array

```
procedure CariMaks(input A: ArrayInt, input n : integer, output  
maks : integer)
```

```
{mencari elemen terbesar di dalam array A[1..n]}
```

```
{K. awal : n sudah berisi ukuran efektif array, seluruh elemen  
array A[1..n] sudah terdefinisi}
```

```
{K. akhir: maks berisi elemen array yang bernilai maksimum}
```

DEKLARASI

```
    i : integer           {pencatat indeks array}
```

ALGORITMA:

```
    maks ← A[1]           {nilai maksimum sementara}
```

```
    for i ← 2 to n do
```

```
        if A[i] > maks then
```

```
            maks ← A[i]
```

```
        endif
```

```
    endfor
```



# Latihan Soal

Buatlah algoritma untuk mencari nilai minimum dalam array!  
(Lihat algoritma mencari nilai maksimum sebagai referensi)

# Penyelesaian Mencari Nilai Minimum Array

```
procedure CariMin(input A: ArrayInt, input n : integer, output  
min : integer)
```

```
{mencari elemen terbesar di dalam array A[1..n]}
```

```
{K. awal : n sudah berisi ukuran efektif array, seluruh elemen  
array A[1..n] sudah terdefinisi}
```

```
{K. akhir: maks berisi elemen array yang bernilai terkecil}
```

DEKLARASI

```
i : integer           {pencatat indeks array}
```

ALGORITMA:

```
min ← A[1]           {nilai minimum sementara}
```

```
for i ← 2 to n do
```

```
    if A[i] < min then
```

```
        min ← A[i]
```

```
    endif
```

```
endfor
```



# Menyalin Array

```
procedure SalinArray(input A : ArrayInt, input n :  
                    integer, output B : ArrayInt)  
{Menyalin (copy) array A[1..n] ke dalam array B[1..n]}  
{K. Awal : Array A[1..n] sudah terdefinisi}  
{K. Akhir : B[1..n] adalah array yang elemen-elemennya  
sama dengan array A[1..n]}
```

DEKLARASI

i : integer

ALGORITMA

```
for i  $\leftarrow$  1 to n do  
    B[i]  $\leftarrow$  A[i]  
endfor
```

# Menguji Kesamaan Dua Buah Array

```
function ArraySama(input A,B : ArrayInt, input n : integer) → boolean  
{Memberika nilai true jika array A[1..n] dan B[1..n] sama, atau false  
jika tidak sama}
```

DEKLARASI

```
  i : integer  
  sama : boolean
```

ALGORITMA:

```
  i ← 1  
  sama ← true  
  while (i ≤ n) and (sama) do  
    if A[i] = B[i] then  
      i ← i + 1 {tinjau elemen berikutnya}  
    else {A[i] ≠ B[i], maka dapat disimpulkan A dan B tidak sama}  
      sama ← false  
    endif  
  endwhile  
  {i > n or not sama}  
  return sama
```

# Translasi Notasi Algoritmik Array ke dalam Bahasa C\*

» » \*bahasa C memulai indeks  
array dari 0

# Sebagai Peubah

```
/* DEKLARASI */
```

```
int A[101];
```

— Elemen array A adalah A[0], A[1], A[2], ..., A[100]

```
char NamaMhs[13][25]; /* panjang string = 25 karakter */
```

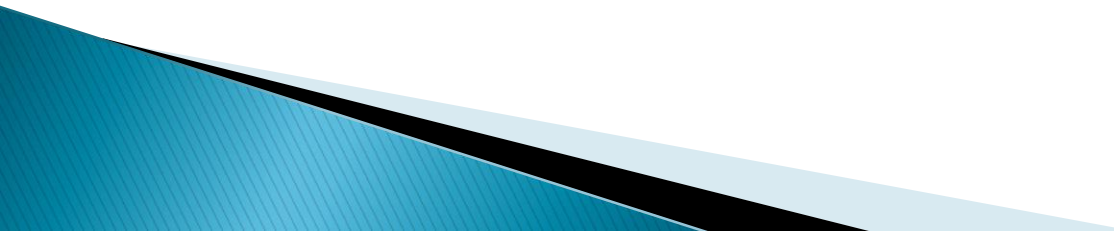
```
float NilUjian[75]
```

Array NamaMhs terdiri atas 13 buah elemen, setiap elemen bertipe string yang panjangnya 25 karakter.

Elemen array NamaMhs adalah NamaMhs[0], NamaMhs[1], NamaMhs[2], ..., NamaMhs[12].

# Cara Mengacu Elemen Array

<code>A[4]</code>	<i>/* mengacu elemen keempat dari array A */</i>
<code>NamaMhs[2]</code>	<i>/* mengacu elemen kedua dari array NamaMhs */</i>
<code>A[i]</code>	<i>/* mengacu elemen ke-i dari array A, asalkan nilai i sudah terdefinisi */</i>
<code>NilUjian[i+1]</code>	<i>/* mengacu elemen ke-i + 1 dari array NilUjian asalkan nilai i sudah terdefinisi */</i>



# Sebagai Tipe Bentukan

```
/* DEKLARASI */  
typedef int ArrayInt[101];  
ArrayInt A; /* A adalah sebuah array integer  
              dengan 100 elemen */  
  
typedef char String[25];  
String NamaMhs[13];
```

# Mendeklarasikan Ukuran Array sebagai Konstanta

```
/* DEKLARASI */  
#define Nmaks 100; /* ukuran maksimum  
                    array */  
typedef int ArrayInt[Nmaks+1];  
ArrayInt A;        /* A adalah sebuah  
                    array integer dengan  
                    100 elemen */
```

# Program Rerata

- » Program utama untuk membaca elemen-elemen array, mencetak elemen-elemen array, dan menghitung nilai rata-ratanya



## *deklarasi tipe bentukan, deklarasi prosedur yang digunakan...*

```
/* Program Rerata */
/* Program utama untuk membaca elemen-elemen array, mencetak
elemen-elemen array, dan menghitung nilai rata-ratanya */

/* Deklarasi prosedur yang digunakan */
#include <stdio.h>

#define Nmaks 100 /* ukuran maksimum array */
typedef int ArrayInt[101];

void BacaArray(ArrayInt A, int n);
/* mengisi elemen-elemen array A[1..n] dengan pembacaan */

void CetakArray(ArrayInt A, int n);
/* mencetak elemen-elemen array[1..n] */

void HitungRataRata(ArrayInt A, int n, float *u);
/* menghitung nilai rata-rata seluruh elemen array A. */
```

## *... program utama...*

```
/* PROGRAM UTAMA */
main()
{
    /* DEKLARASI */
    ArrayInt A;
    int k;    /* indeks array */
    int n;    /* banyaknya elemen array yang dipakai */
    float u; /* nilai rata-rata */

    /*ALGORITMA:*/
    printf("Baca jumlah data (n): "); scanf("%d", &n);

    printf("Baca data : \n");
    BacaArray(A, n);

    printf("Cetak data: \n");
    CetakArray(A, n);

    printf("Hitung Rata-rata: \n");
    HitungRataRata(A, n, &u);
    printf("Rata-rata data = %.2f \n", u);
    system("PAUSE");
}
```

## *... Definisi Prosedur BacaArray...*

```
void BacaArray(ArrayInt A, int n)
/* Mengisi elemen-elemen array A[1..n] dengan pembacaan */
/* K. Awal : n adalah ukuran efektif array, nilainya
   terdefinisi */
/* K. Akhir : setelah pembacaan, seluruh elemen array A
   berisi nilai-nilai yang dibaca dari piranti masukan. */
{
    /* DEKLARASI */
    int i; /* pencatat indeks array */
    /* ALGORITMA: */
    for (i=1; i<=n; i++)
    {
        printf("Masukkan nilai A[%d] : ", i);
        scanf("%d", &A[i]);
    }
}
```

## *... Definisi Prosedur CetakArray...*

```
void CetakArray(ArrayInt A, int n)
/* Mencetak elemen-elemen array A[1..n] */
/* K. Awal : n sudah berisi jumlah elemen array yang dipakai.
   Elemen-elemen array A sudah terdefinisi */
/* K. Akhir : elemen-elemen array A tercetak */
{
    /* DEKLARASI */
    int i; /* pencatat indeks array */
    /* ALGORITMA: */
    for (i=1; i<=n; i++)
        printf("A[%d] = %d \n", i, A[i]);
}
```

### *... Definisi Prosedur HitungRataRata.*

```
void HitungRataRata(ArrayInt A, int n, float *u)
/* Menghitung nilai rata-rata seluruh elemen array A */
/* K. Awal : elemen-elemen array A sudah terdefinisi; n
   adalah jumlah elemen array, nilainya sudah terdefinisi */
/* K. Akhir : u berisi nilai rata-rata seluruh elemen */

{
  /* DEKLARASI */
  int i;          /* pencatat indeks array */
  float jumlah; /* jumlah total seluruh nilai */

  /* ALGORITMA: */
  i = 1;          /* dimulai dari elemen pertama */
  jumlah = 0;     /* jumlah total nilai mula-mula */
  for (i=1; i<=n; i++)
    jumlah = jumlah + A[i];
  /* endfor */

  *u = jumlah/n;
}
```