

LARIK

MINGGU KE: 5

TUJUAN:

- Mahasiswa dapat memahami definisi larik.
- Mahasiswa dapat mendeklarasikan larik.
- Mahasiswa dapat memproses larik.

TEORI PENGANTAR:

Definisi Larik

Larik adalah struktur data yang menyimpan sekumpulan elemen yang bertipe sama, setiap elemen diakses langsung melalui indeksinya. Contoh: $a_1, a_2, \dots, a_n$.

Deklarasi Larik

Sebagai Peubah

DEKLARASI

A : array[1..100] of integer
NamaMhs : array[1..12] of string
NilUjian : array[0..74] of real

Sebagai Tipe Bentukan

DEKLARASI

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

Ukuran Larik sebagai Sebuah Konstanta

DEKLARASI

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

Cara Mengacu Elemen Larik

A[4] {mengacu elemen keempat dari larik A}
NamaMhs[i] ← 'Achmad' {mengisi elemen ke-i dari larik NamaMhs dengan string 'Achmad'}

Pemrosesan Larik

Contoh:

PROGRAM PemrosesanLarik
{Skema pemrosesan Larik secara beruntun}

DEKLARASI

```
const Nmaks = 100    {ukuran maksimum larik}  
type LarikInt : array[1..Nmaks] of integer  
A : LarikInt  
i : integer    {indeks larik}
```

ALGORITMA:

```
Inisialisasi  
i ← 1 {mulai dari elemen pertama}  
while i ≤ Nmaks do  
    pemrosesan terhadap A[i]  
    i ← i + 1    {tinjau elemen berikutnya}  
endwhile  
{i > Nmaks}  
Terminasi
```

Ukuran Efektif Larik

DEKLARASI

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

Inisialisasi Larik

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

DEKLARASI

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

ALGORITMA:

```
for i ← 1 to n do  
    A[i] ← 0  
endfor
```

Program Pemanggil

PROGRAM PemrosesanLarik

{Program untuk mengisi elemen larik dengan nilai 0}

DEKLARASI

```
const Nmaks = 100    {ukuran maksimum larik}  
type LarikInt : array[1..Nmaks] of integer  
A : LarikInt  
i : integer    {indeks larik}  
n : integer    {ukuran efektif larik}  
procedure InisDengan0(output A : LarikInt, input n : integer)  
{Menginisialisasi setiap elemen larik A[1..n] dengan nol.}
```

ALGORITMA:

```
read(n) {tentukan jumlah elemen larik yang akan digunakan, dengan syarat  $1 \leq n \leq N_{maks}$ }  
InisDengan0(A, n)  
{cetak hasil inisialisasi}  
for i  $\leftarrow$  1 to n do  
  write (A[i])  
endfor
```

Mengisi Elemen Larik dengan Pembacaan

Jumlah elemen efektif ditentukan di awal

```
procedure BacaLarik1(output A: LarikInt, input n: integer)  
DEKLARASI  
  i : integer  
ALGORITMA:  
  for i  $\leftarrow$  1 to n do  
    read(A[i])  
  endfor
```

Jika jumlah elemen efektif baru diketahui di akhir pembacaan

```
procedure BacaLarik2(output A: LarikInt, output n: integer)  
DEKLARASI  
  jawab: char  
ALGORITMA:  
  n  $\leftarrow$  0  
  repeat  
    n  $\leftarrow$  n + 1  
    read(A[n])  
    write('Lagi? (y/t)')  
    read(jawab)  
  until jawab='t'
```

atau

```
procedure BacaLarik3(output A: LarikInt, output n: integer)  
DEKLARASI  
  x : integer  
ALGORITMA:  
  n  $\leftarrow$  0  
  read(x)  
  while x  $\neq$  9999 do  
    n  $\leftarrow$  n + 1  
    A[n]  $\leftarrow$  x  
    read(x)  
  endwhile
```

PRAKTIKUM:

1. Buat program input dan cetak elemen larik.
2. Buat program menghitung rata-rata dan mencari nilai maksimum.
3. Buat program untuk menyalin larik.
4. Buat program untuk menguji kesamaan dua buah larik.