



# Pertemuan - 2

## Array dan Matriks

*Oleh: Boldson Herdianto. S., SKom., MMSI.*

# Tipe Data Sederhana

- ★ Integer : melingkupi himpunan bilangan bulat (... , -3, -2, -1, 0, 1, 2, 3, ...)
- ★ Real : melingkupi himpunan bilangan real, termasuk integer, pecahan, dan irrasional. Untuk penulisan data real dapat menggunakan titik desimal, misalnya 37.125, selain itu dalam bahasa pemrograma digunakan penulisan eksponensial (3.7125E+01 atau 371.25E-01)

# Cont.

- ✦ Boolean atau Logika: jenis data yang memiliki 2 nilai yaitu benar (True) dan salah (False) yang diwujudkan sebagai 1 dan 0.
- ✦ Character: meliputi huruf dalam alfabet serta simbol digit (disebut karakter alphanumeric) atau simbol khusus seperti \$, !, dsb.

# Deklarasi Data dalam Bahasa Pemrograman

## PROGRAM CONTOH

{Program untuk menghitung luas dan keliling lingkaran}

CONST

Pi = 3.14; {Konstanta pi}

VAR

a : Integer; {a adalah jari-jari lingkaran}

l : real; {l adalah luas lingkaran}

k : real; {k adalah keliling lingkaran}

BEGIN

READLN (a);

l := pi \* a \* a;

k := pi \* 2 \* a;

WRITELN ('Luas =', l);

WRITELN ('Keliling =', k)

END.

# Struktur Data : **Array / Larik**

## Tujuan

Membahas struktur data yang paling sederhana dan mudah pengoperasiannya, yaitu array / larik.

## Definisi

struktur data yang mengacu pada sekumpulan elemen yang diakses melalui indeks

**Array A satu dimensi :**  
8 indeks (1 s/d 8) dan data 1, 7, 18 dst.

1	7	18	03	69	24	08	70
1	2	3	4	5	6	7	8

# KELEBIHAN & KEKURANGAN

## Array / Larik

### ☀ KELEBIHAN

- Struktur Data paling mudah
- ✓ - Memori ekonomis, bila semua elemen terisi
- ✓ - Waktu akses sama ke setiap elemen

### ☀ KEKURANGAN

- Boros memori jika banyak elemen yang tidak digunakan
- ✓ - Struktur Data Statis

# KAMUS DATA

**Array / Larik**

## ☀ KAMUS

- ✓ Nama : array [1..400] of string
- ✓ Panjang : array [a...e] of real
- ✓ Type TITIK : record < X : real, Y : real >  
Kurva : array [0 .. 30] of Titik

✓ Catatan : Tipe Data sejenis (homogen)

✓ Indeks data memiliki keterurutan

# PROSES LARIK

**Array / Larik**

Program Proses\_Larik

## KAMUS

Const :  $N = 8$  {jumlah elemen larik}  
Indeks : integer  
A : array [1..N] of integer {deklarasi larik A dengan tipe data integer}

## ALGORITMA

For Indeks  $\leftarrow$  1 to N do

***PROSES LARIK***

Endfor

- ✓ Catatan : Tipe Data sejenis (homogen)
- ✓ Indeks data memiliki keterurutan



# CONTOH PROSES

**Array / Larik**

## ALGORITMA

For Indeks  $\leftarrow$  1 to N do

*PROSES LARIK*

Endfor

- ✓ Mengisi elemen larik dengan 0 (inisialisasi)
- ✓ Mengisi elemen larik dari piranti masukan
- ✓ Mencetak elemen larik ke piranti keluaran

**A[Indeks]=0**

**Input A[Indeks]**

**Print A[Indeks]**

# INISIALISASI

**Array / Larik**

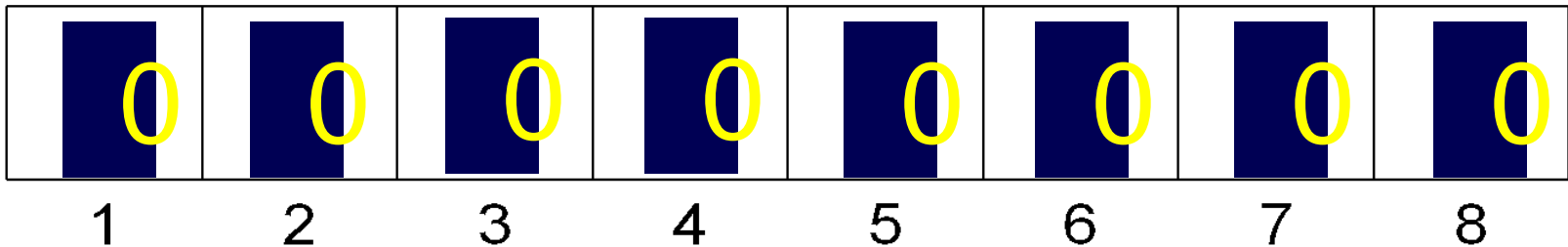
## ALGORITMA

For Indeks  $\leftarrow$  1 to 8 do

***$A[\text{Indeks}] = 0$***

Endfor

Array A satu dimensi :



# INPUT ELEMEN

**Array / Larik**

## ALGORITMA

For Indeks  $\leftarrow$  1 to 8 do

**Input** *A[Indeks]*

Endfor

? 1

? 3

? 5

Array A satu dimensi :

1	3	5	7	2	9	4	7
1	2	3	4	5	6	7	8

# CETAK ELEMEN

**Array / Larik**



## ALGORITMA

```
For Indeks  $\leftarrow$  1 to 8 do  
  Print A[Indeks]  
Endfor
```

Array A satu dimensi

1	3	5	7	2	9	4	7
1	2	3	4	5	6	7	8

# PROSES BENTUK LAIN

**Array / Larik**

## ALGORITMA

For Indeks  $\leftarrow$  1 to 8 do

**Proses Larik**

Endfor



- ❑ Mencari bilangan **maksimum/minimum** pada larik
- ❑ **Menjumlahkan nilai** seluruh elemen larik
- ❑ **Membuat rata-rata** nilai seluruh elemen larik
- ❑ Mencari nilai tertentu pada larik



# Cari Bilangan Maksimum

**Array / Larik**

## ALGORITMA

**Maks = A[1]**

For Indeks  $\leftarrow$  2 to 8 do

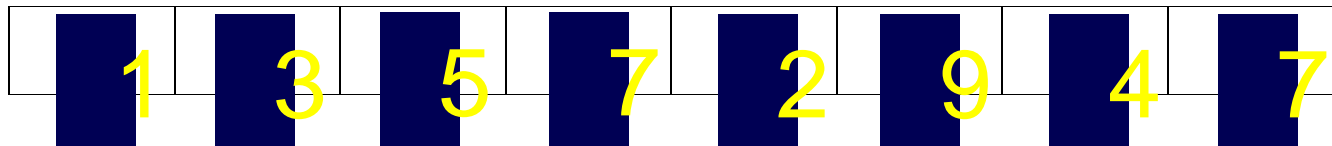
**If A[Indeks] > Maks then**

**Maks = A[Indeks]**

Endfor

**Print Maks**

Array A satu dimensi :





# HITUNG PANJANG

**Array / Larik**

$$\text{Panjang} = \text{UB} - \text{LB} + 1$$

dimana:

UB - upper bound ( indeks terbesar)

LB - lower bound (indeks terkecil)

Contoh : Seorang pedang mobil menggunakan larik untuk menyimpan data penjualan dari tahun 1990 sampai dengan tahun 2001. Berapa panjang (jumlah elemen) larik yang harus disediakan?

$$LB = 1990$$

$$UB = 2001$$

$$\text{Jadi panjang} = UB - LB + 1 = 2001 - 1990 + 1 = 12$$



# PENGALAMATAN

**Array / Larik**

$$\text{LOK(LA[K])} = \text{Awal(LA)} + W(K - \text{LB})$$

di mana:

- LOK(LA[K]) – lokasi elemen dengan indeks K, yang dicari
- K -- Indeks yang dicari
- Awal (LA) -- Lokasi awal dari larik
- W – jumlah byte untuk menyimpan 1 elemen larik
- LB -- lower bound / batas bawah





# PENGALAMATAN

**Array / Larik**

$$\text{LOK(LA[K])} = \text{Awal(LA)} + W(K - \text{LB})$$

Contoh:

Misalkan Awal (JUAL) = 100 dan  $W= 4$ , maka

$$\text{LOK (JUAL[1990])} = 100$$

$$\text{LOK (JUAL[1991])} = 104$$

$$\text{LOK (JUAL[1992])} = 108$$

Berapa lokasi JUAL[2000] ? untuk mendapat lokasi tersebut

$$\begin{aligned}\text{LOK(LA[K])} &= \text{Awal(LA)} + W(K - \text{LB}) \\ &= 100 + 4 * (2000 - 1990) \\ &= 140\end{aligned}$$