

DIKTAT STRUKTUR DATA



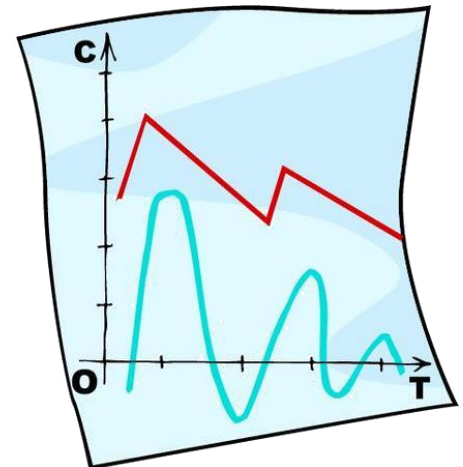
Oleh:
Tim Struktur Data IF

BAB I – PENGENALAN STRUKTUR DATA
ARRAY STATIS

DATA



1. Kumpulan dari nilai-nilai.
2. Bagian terkecil dari suatu informasi
3. Data bersifat **JAMAK**.





STRUKTUR DATA

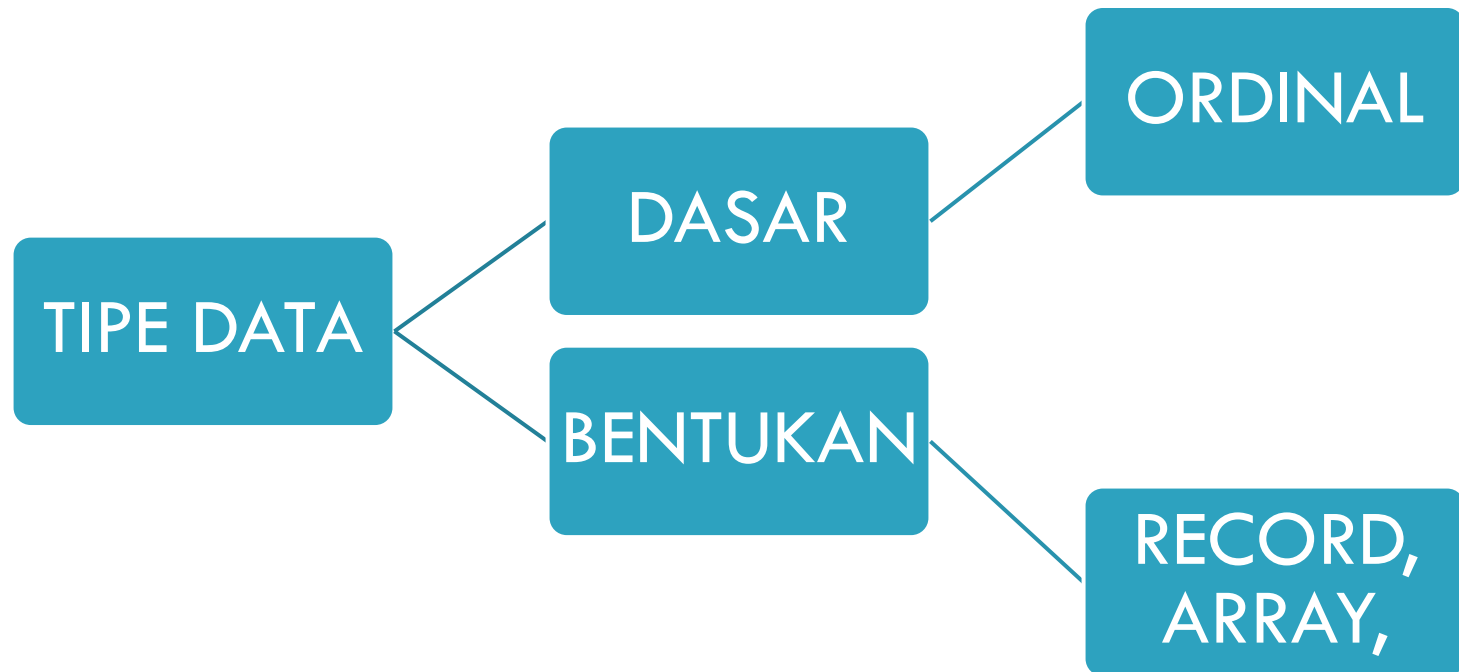
1. Model matematis atau bentuk logik dari suatu organisasi data.
2. Model data harus bisa merefleksikan data di kehidupan nyata.
3. Struktur data harus simpel dalam memproses data yang ada di dalamnya.

TIPE DATA

1. Tipe data dasar
2. Tipe data bentukan.



TIPE DATA





TIPE DATA DASAR

1. Tipe data yang sudah dikenal di kehidupan sehari-hari.
2. Sering disebut sebagai **predefined data type**.
3. Contoh: bilangan logika, bilangan bulat, karakter, bilangan riil, dan string.
4. Bilangan logika, bilangan bulat, karakter → **tipe data ordinal** (bisa dikonversikan ke nilai integer).



TIPE DATA BENTUKAN

1. Tipe yang didefinisikan sendiri oleh pemrogram.
2. Sering disebut sebagai **user-defined type data**.
3. Contoh: tipe dasar yang diberi nama dengan tipe data baru dan **record**.



ALGORITMA TIPE BENTUKAN

1. Tipe dasar yang diberi nama tipe baru

```
type nama_baru : tipedasar
```

2. Record

```
type nama_record : record  
  < elemen1 : tipedata,  
    elemen2 : tipedata >
```




TIPE ARRAY

- Adalah tipe terstruktur yg mempunyai komponen dalam jumlah yang tetap dan setiap komponen mempunyai tipe yang sama.

PENGERTIAN ARRAY STATIS

Sekumpulan data yang bertipe data sama yang bisa diakses lewat indeksnya.



REPRESENTASI ARRAY STATIS



Array statis direpresentasikan di memori secara kontinyu. Contoh: array nama (1:5).

nama

nama(1)
nama(2)
nama(3)
nama(4)
nama(5)





DEKLARASI UMUM (1)

Algoritma:

Kamus :

nama_var_array: array[1..maks_array] of tipedata

Contoh:

Kamus :

nama: array[1..5] of string

DEKLARASI UMUM (2)

Algoritma:

Kamus :

Const

maks_array = ...

nama_var_array: array[1..maks_array] of tipedata

Contoh:

Kamus :

Const

maks_array = 5

nama: array[1..maks_array] of string

DEKLARASI UMUM (3)

Algoritma:

Kamus:

Const

maks_array = ...

Type

nama_type_array=array[1..maks_array] of typedata

nama_var_array:nama_type_array

DEKLARASI UMUM (3)

Contoh:

Kamus:

Const

maks_array = 5

Type

data_nama=array[1..maks_array] of string

nama:data_nama

DEKLARASI ARRAY OF RECORD



Algoritma:

Kamus :

Const

maks_array = ...

Type

nama_record = record

< field_1:tipe_data_1,

field_2:tipe_data_2,

...

field_n:tipe_data_n >

endrecord

nama_type_array=array[1..maks_array] of nama_record

nama_var_array:nama_type_array

DEKLARASI ARRAY OF RECORD



Contoh:

Kamus :

Const

```
maks_array = 5
```

Type

```
data_mahasiswa = record
```

```
< nim, nama: string,  
  nilai    : integer,  
  indeks   : char >
```

```
endrecord
```

```
mahasiswa = array[1..maks_array] of data_mahasiswa
```

```
mhs:mahasiswa
```

OPERASI-OPERASI

1. Penciptaan (create) array statis

Mempersiapkan array untuk diakses/diproses dengan asumsi elemen array diisi dengan angka 0 jika elemen arraynya diisi numerik/bilangan/angka atau diisi dengan karakter " " / " " / ' ' untuk alphanumerik.



OPERASI-OPERASI

Algoritma:

```
Procedure create (Output nama_var_array:nama_type_array)  
{I.S: elemen array diberi harga awal agar siap digunakan}  
{F.S: menghasilkan array yang siap digunakan}
```

Kamus:

indeks: integer

Algoritma:

```
for indeks  $\leftarrow$  1 to maks_array do  
    nama_var_array(indeks)  $\leftarrow$  0 {elemen array numerik}  
endfor
```

EndProcedure

Inisialisasi Array

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

Kamus

i : integer

ALGORITMA:

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

EndProcedure

OPERASI-OPERASI

2. Traversal

Proses mengunjungi setiap elemen array satu persatu dari elemen pertama sampai elemen terakhir.





OPERASI-OPERASI

Proses traversal:

1. Pengisian elemen array dengan data
2. Menampilkan elemen array
3. Penambahan data di array
4. Penyisipan data di indeks tertentu pada array
5. Penghapusan data di indeks tertentu pada array
6. Menentukan nilai maksimum dan minimum
7. Menghitung nilai rata-rata, dsb.

OPERASI-OPERASI

Algoritma umum traversal:

```
Procedure traversal (I/O nama_var_array:nama_type_array)  
{I.S: maksimum array sudah terdefinisi}  
{F.S: menghasilkan array yang sudah diproses}
```

Kamus:

Algoritma:

```
for indeks  $\leftarrow$  1 to maks_array do  
    proses
```

```
endfor
```

```
Terminasi {penutupan yang harus dilakukan setelah proses  
           selesai}
```

EndProcedure

