



Struktur Data



Objektif

- Mengetahui maksud struktur data dan menjelaskan penggunaannya dalam pemrograman
- Mengetahui operasi yang terkait dengan struktur data dan metode pemrograman paling umum yang terkait dengan struktur tersebut.
- Mengetahui metode dan notasi yang digunakan untuk menspesifikasi apa-apa yang perlu dikerjakan oleh program dan bagaimana program ini melakukan pekerjaan tersebut.



Jenis/Tipe Data (Data Type)

- Terdiri dari
 - Set nilai data
 - Set operasi yang bisa diterapkan pada nilai tersebut



Klasifikasi Jenis Data

- Simple Data Type (Jenis Data Sederhana)
 - Item data individual
- Data Structures / data aggregates (struktur data)
 - Kombinasi dari item data individual
 - Membentuk item data lain



Jenis Data Sederhana

- Numerik, terdiri dari :
 - Numerik integer (bilangan bulat)
 - Numerik real (bilangan riil)
- Karakter, terdiri dari :
 - Alfabet : a .. z, A .. Z
 - Angka : 0 .. 9
 - Simbol khusus : + ? ' ! [] { } ... dll
- Boolean (logika), terdiri dari :
 - True
 - False



Identifier

- Dalam bahasa pemrograman, item data diidentifikasi menurut namanya, bukan menurut alamat lokasinya dalam memori
- *Identifier* akan merupakan **konstanta** jika ia selalu dikaitkan dengan nilai data yang sama
- *Identifier* akan merupakan **variabel** jika nilai datanya yang terkait bisa berubah
- *Literal*, nilai data yang tertera dalam program namun bukan sebagai *identifier*



Deklarasi Data

- Jenis data konstanta dan variabel harus didefinisikan dalam program sehingga :
 - operasi yang tepat dapat dijalankan pada nilai data dan
 - Jumlah ruang penyimpanan yang tepat bisa ditentukan
- Statement untuk mendefinisikan jenis data disebut **declarative statement**
- Beberapa bahasa pemrograman memiliki sintaks pendeklarasian yang berbeda
- Beberapa contoh program (pendeklarasian data) yang akan diberikan ditulis dalam *pseudo-code*



Contoh

Constants

```
pi = 3.141592654
```

Variables

```
i, qty                : integer
harga_satuan, harga_beli : real
status                : boolean
nama                  : character(25)
```



Struktur Data

- Kelompok item data yang terorganisasi yang dianggap sebagai suatu unit
- Disebut juga sebagai jenis data kompleks (*complex data type*) atau *data aggregates*
- Beberapa struktur data :
 - Array (larik)
 - String
 - Record
 - List (daftar)
 - Tree



Array (Larik)

- Set item data yang disusun secara baik menjadi rangkaian dan diacu atau ditunjuk oleh satu *identifier*
- Contoh : Nilai = (56 42 89 65 48)
- Item data individual dalam array bisa ditunjuk secara terpisah dengan menyatakan posisinya dalam array itu
 - Nilai(1) menunjuk 56
 - Nilai(2) menunjuk 42
- Bilangan yang ditulis dalam tanda kurung menandakan **posisi** item individual dalam array (disebut juga *subscript / indeks*)

Array (Larik)

- Variabel bisa digunakan sebagai subscript, misalnya Nilai(*i*).
 - Jika *i* = 2 maka menunjuk ke Nilai(**2**) yaitu 42
 - Jika *i* = 4 maka menunjuk ke Nilai(**4**) yaitu 65
- Item data individual dalam suatu array sering disebut **elemen**
- Matriks
 - Array yang hanya berisi bilangan dan tidak ada data alfabetisnya
- Klasifikasi Array
 - Array 1 dimensi
 - Array multi dimensi

Array Multi Dimensi

- Mempunyai elemen-elemen yang disusun ke dalam baris dan kolom dan digunakan sebagai tabel data
- Contoh : Nilai ujian dari mahasiswa satu kelas untuk beberapa mata kuliah bisa ditempatkan dalam array 2 dimensi

Siswa ke (no. baris)	B. Inggris (kolom 1)	Matematika (kolom 2)
1	$A(1,1) = 56$	$A(1,2) = 44$
2	$A(2,1) = 42$	$A(2,2) = 36$
3	$A(3,1) = 89$	$A(3,2) = 73$
4	$A(4,1) = 65$	$A(4,2) = 86$
5	$A(5,1) = 48$	$A(5,2) = 51$

$$\rightarrow A = \begin{bmatrix} 56 & 44 \\ 42 & 36 \\ 89 & 73 \\ 65 & 86 \\ 48 & 51 \end{bmatrix}$$



Deklarasi Array

■ Array 1 dimensi

Variables

Nilai: array [1..5] of integer

A : array [1..4] of real

■ Array 2 dimensi

Variables

A : array [1..5, 1..2] of integer



Penanganan Array

■ Metode dasar penanganan array :

- ☐ Mencari nilai terbesar
- ☐ Mencari nilai terkecil
- ☐ Menghitung nilai rata-rata
- ☐ Menghitung nilai total
- ☐ Menghitung jumlah nilai di bawah rata-rata

■ Menyortir Array (Sort)

- ☐ Buble sort
- ☐ Straight selection sort

■ Mencari/Meneliti Array (Search)

- ☐ Linear search



Penanganan Array

- Contoh : Nilai ujian mahasiswa akan dibaca dalam array. Kemudian akan ditampilkan nilai terbesar, nilai terkecil, nilai rata-rata, nilai total, dan jumlah nilai di bawah rata-rata.
- Tahapan penanganan array
 - Input nilai data ke dalam array
 - Mengkalkulasi nilai terbesar, terkecil, total, dan rata-rata
 - Mengkalkulasi jumlah nilai di bawah rata-rata
 - Menampilkan hasilnya (output)



String

- Rangkaian karakter yang ditangani sebagai unit data tunggal
- Contoh (string literal) :
 - "ABC, 32fl2. 3h"
 - "Kucing dalam karung"
- Contoh (variabel string) :
 - A = "Universitas"
 - B = "Gunadarma"
- Berada dalam bentuk array karakter 1 dimensi



String

- *Fixed-length* string (String yang panjangnya tetap)
 - Mempunyai jumlah tempat karakter yang tetap yang tersedia (bisa digunakan) untuk penyimpanan data
- *Variable-length* string (String yang panjangnya berubah-ubah)
 - Memberi data sejumlah spasi (ruang) sesuai yang ia perlukan



■ *Fixed-length* string

posisi karakter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
isi	A	N	D	R	I	A	M	R	I		I	N	A			J	O	K	O		D	E	D	I	
komentar	string ke 1					string ke 2					string ke 3					string ke 4					string ke 5				

■ *Variable-length* string

posisi karakter	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
isi	A	N	D	R	I	*	A	M	R	I	*	I	N	A	*	J	O	K	O	*	A	L	*		
komentar	string ke 1					string ke 2					string ke 3					string ke 4					string ke 5			tempat sisa	



Deklarasi String

■ *Fixed-length* string

```
Variables  
    nama : string[5]
```

■ *Variable-length* string

```
Variables  
    nama : string
```



Operasi pada String

■ Concatenation

- Penggabungan dua atau lebih string
- Contoh :

```
A = "Universitas"  
B = "Gunadarma"  
C = A + B
```

```
maka  
C = "UniversitasGunadarma"
```



Operasi pada String [2]

■ Substring

- Mengambil bagian dari suatu string

- Contoh

```
A = "Universitas"  
B = "Gunadarma"  
C = Left(A, 3)  
D = Right(B, 5)  
E = Substr(A, 4, 5)
```

maka

```
C = "Uni"  
D = "darma"  
E = "versi"
```



Record

- Seperti array 1 dimensi
- Terdiri dari serangkaian item data yang terkait
- Item data berurutan yang ada dalam record bisa mempunyai jenis yang berbeda
- Contoh : Mengorganisasikan 3 item data yang berbeda ke dalam struktur data tunggal
 - NIP : string(8)
 - Nilai : real
 - Lulus : boolean



Deklarasi Record

```
mahasiswa      : record
                  NIP      : string(8)
                  Nilai     : real
                  Lulus     : boolean
                end record
```

- Setiap elemen memiliki *identifier* sendiri
- Elemen dari suatu record disebut **field**



- Penunjukan ke setiap field dari suatu record bisa dilakukan dengan :

- ☐ Notasi “**dot**” (titik)

```
Begin
    mahasiswa.NPM := '51292215'
    mahasiswa.Nilai := 90.5
    mahasiswa.Lulus := True
End
```

- ☐ Notasi “**with**”

```
Begin
    with mahasiswa
    do
        NPM := '51292215'
        Nilai := 90.5
        Lulus := True
    end with
End
```

Array Record (Tabel)

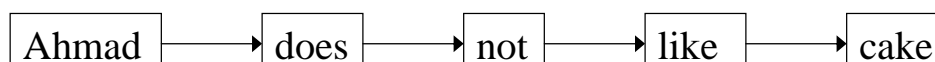
- Kumpulan dua atau lebih record
- Deklarasi Array Record

Variable

```
Mahasiswa : Array [1..5] of record
    NIP      : string(8)
    Nilai    : real
    Lulus    : boolean
End record
```

List

- Memberikan cara yang fleksibel untuk penanganan item data secara urut
- Perubahan terhadap urutan tersebut dapat dicapai (dilakukan) dengan perpindahan data yang minimal dan kehilangan ruang penyimpanan yang sedikit
- Contoh : Kalimat "Ahmad does not like cake" dituliskan sebagai suatu list, seperti berikut :





■ Beberapa istilah

- *DATUM* : item data dalam list
- *POINTER* : penunjuk yang menyambungkan item data satu dengan yang lain
- *NODE* / elemen : elemen dari suatu list yang terbentuk dari datum dan pointer
- *TERMINATOR* : pointer terakhir dari list
- *START POINTER* : menyatakan tempat datum pertama
- *FREE STORAGE POINTER* : menyatakan di mana datum berikutnya bisa mengarah atau menuju



Deklarasi List

- List dapat dideklarasikan sebagai sebuah array record

Variable

kalimat : Array [1..7] of record

Datum : string

Next : integer

End record



<div>Start Pointer</div> <div>Free storage Pointer</div>	Row Number	Datum	Pointer to Next Datum	Comment
	1	“Ahmad”	2	Next datum is in row 2
	2	“does”	3	Next datum is in row 3
	3	“not”	4	Next datum is in row 4
	4	“like”	5	Next datum is in row 5
	5	“cake”	-1	Last datum; -1 is a terminator
	6		7	
	7		8	



Operasi pada List

- *Deletion* : penghapusan elemen suatu list
 - Ketika elemen suatu list dihapus, tempat penyimpanan yang telah dikosongkan dapat digunakan lagi
- *Insertion* : penyisipan elemen ke dalam suatu list
- *Search* : pencarian elemen dalam suatu list



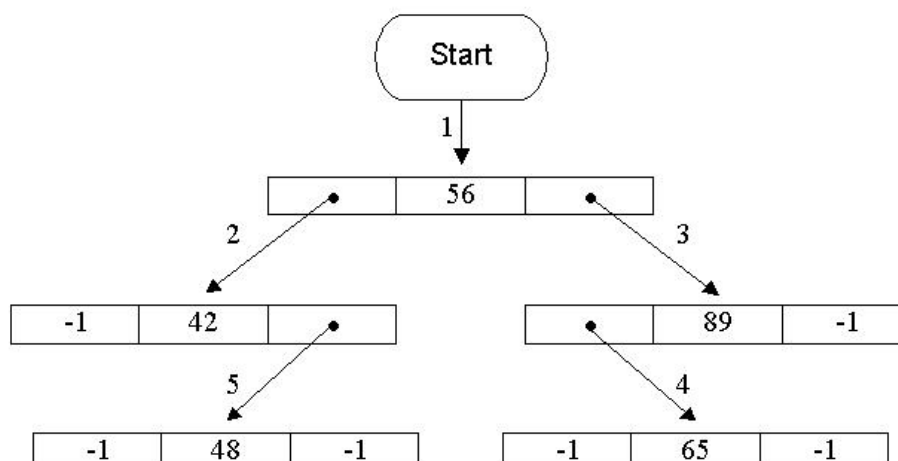
Tree

- Struktur data hirarki
- Dikonstruksi menggunakan aturan preseden untuk item data, misal : menggunakan rangkaian alfabet atau numerik
- Beberapa Istilah :
 - *NODE* : elemen dari suatu tree
 - Setiap *node* memiliki (sedikitnya) dua pointer yaitu *left pointer* dan *right pointer*
 - *ROOT NODE* : datum pertama yang ditempatkan dalam tree
 - *PARENT NODE* : *node* yang memiliki *node* di bawahnya (*sub-node*)
 - *CHILD NODE* : *node* yang berada di bawah *parent*
 - *LEAF NODE* : *node* yang tidak mempunyai *child*

31



- Contoh : bilangan-bilangan ini (56 42 89 65 48) ditempatkan ke dalam tree



- Catatan :
 - Node paling kiri berisi bilangan terkecil
 - Node paling kanan berisi bilangan terbesar



Row Number	Left Pointer	Datum	Right Pointer
1	2	56	3
2	-1	42	5
3	4	89	-1
4	-1	65	-1
5	-1	48	-1