

BAB 1

TIPE DATA

Struktur data adalah suatu koleksi atau kelompok data yang dapat dikarakterisasikan oleh organisasi serta operasi yang didefinisikan terhadapnya.

Data secara umum dapat dikategorikan :

- Tipe data sederhana atau data sederhana
 1. Tunggal : Integer, Real, Boolean, Karakter
 2. Majemuk : String

- Struktur Data
 1. Sederhana : Array dan Record
 2. Majemuk terdiri atas
 - Linier : Linier Linked List, Stack, Queue
 - Non Linier : Binary Tree, Binary Search Tree, General Tree, Tree, Graf

♣ INTEGER

Anggota dari himpunan bilangan :

{..., -(n+1), -n, ..., -2, -1, 0, 1, 2, ..., n, n+1, ...}

Operasi dasar yaitu : penjumlahan, pengurangan, perkalian, pembagian dan perpangkatan

Pembagian Integer (DIV)

Hasil pembagian integer DIV adalah sebuah integer (menghilangkan bagian pecahan dari hasil pembagian)

Contoh : $27 \text{ DIV } 4 = 6$

Selain itu terdapat **operasi MOD (Modulo)** adalah sisa dari pembagian

Contoh : $27 \text{ MOD } 4 = 3$

Operator yang bekerja terhadap sepasang integer (operand) disebut **Binary Operator**. Sedangkan operator yang hanya bekerja terhadap satu operand saja disebut **Unary Operator**.

Contoh dari unary operator adalah negasi.

♣ REAL

Data numerik yang bukan termasuk integer, digolongkan dalam jenis data real. Ditulis menggunakan titik desimal (atau koma desimal). Dimasukkan ke dalam memori komputer memakai sistem floating point, disebut Scientific Notation.

Penyajianya terdiri dari : mantissa (pecahan) dan eksponen.

Contoh :

Di dalam sistem desimal, $123000 = 0.123 * 10^6$
di sini 0.123 adalah mantissa atau pecahan, sedangkan 6 adalah eksponennya.

Secara umum suatu bilangan real X dituliskan $M * R^E$
di sini : M dijadikan pecahan, R adalah radixnya dan E merupakan eksponennya.

♣ BOOLEAN

Disebut juga jenis data **logical**. Anggota { **true** atau **false**}.

A. Operator Logika, yaitu : AND, OR, NOT

- Operator AND akan menghasilkan nilai true, jika kedua operand bernilai true.
- Operator OR akan menghasilkan nilai true, jika salah satu operand bernilai true
- Operator NOT merupakan “precedence” dari operator AND dan OR.

Dalam suatu ekspresi yang tidak menggunakan tanda kurung, operator **NOT** harus dievaluasi sebelum operator AND dan OR.

B. Operator Relasional, yaitu : >, <, >=, <=, <> dan =

Contoh : $6 < 8 = \text{True}$

$9 < 8 = \text{False}$

♣ KARAKTER

Elemen dari suatu himpunan yang terdiri atas bilangan, abjad dan simbol khusus.

$\{0,1,\dots,8,9, A, B, \dots, Y,Z, +, -, *, \sqrt{}, \dots\}$

♣ STRING

Barisan hingga karakter yang dibentuk oleh suatu kumpulan dari karakter.

Karakter yang digunakan untuk membentuk suatu string disebut alfabet. Dalam penulisannya, suatu string berada dalam tanda “aphostrophe”.

Contoh :

Misal diberikan himpunan alfabet $A = \{C,D,1\}$.

String yang dapat dibentuk dari alfabet di atas di antaranya : ‘CD1’, ‘CDD’, ‘DDC’, ‘CDC1’,... dan sebagainya, termasuk “null string” atau “empty string”

Himpunan tak hingga dari string yang dibentuk oleh alfabet A disebut **VOCABULARY**, Notasi : V_A atau A^*

Jika suatu string dibentuk dari alfabet $\{0,1\}$, maka string yang terbentuk disebut dengan “**Bit String**”.

OPERASI	Operator
Jumlah karakter dalam string	LENGTH
Gabungan 2 buah string	CONCAT
Sub bagian dari string	SUBSTR
Menyisipkan string ke dalam string yang lain	INSERT
Menghapus karakter dalam string	DELETE

LENGTH

Nilai dari operasi ini adalah suatu integer yang menunjukkan panjang dari suatu string .

Notasi : $LENGTH(S) = N$ (integer)
di sini $S = \text{String}$, $N = \text{integer}$

Contoh :

- Jika diberikan string $S = 'a_1a_2 \dots a_N'$
Maka $LENGTH(S) = N$
- Jika diberikan string $S = \text{"SISTEMINFORMASI"}$
Maka $LENGTH(S) = 15$
- Jika diberikan string $S = \text{"SISTEM INFORMASI"}$
Maka $LENGTH(S) = 16$
- Jika diberikan string $S = \text{"ABCD20"}$
Maka $LENGTH(S) = 6$

CONCAT

Operasi ini bekerja terhadap dua string dan hasilnya merupakan resultan dari kedua string tersebut.

Jika S_1 dan S_2 masing-masing adalah suatu string, maka bentuk operasi CONCATENATION dinotasikan dengan : $CONCAT(S_1, S_2)$.

Contoh :

Misal $S_1 = 'a_1a_2 \dots a_N'$ dan $S_2 = 'b_1b_2 \dots b_M'$
Maka $CONCAT(S_1, S_2) = 'a_1a_2 \dots a_Nb_1b_2 \dots b_M'$

String $S_1 = \text{"UNIVERSITAS"}$

String $S_2 = \text{"GUNADARMA"}$

$CONCAT(S_1, S_2) = \text{"UNIVERSITASGUNADARMA"}$

$LENGTH(CONCAT(S_1, S_2)) = 20$

$LENGTH(S_1) + LENGTH(S_2) = LENGTH(CONCAT(S_1, S_2))$

$$\begin{array}{rccccccc} 11 & & + & & 9 & & = & & 20 \\ & & & & 20 & & = & & 20 \end{array}$$

SUBSTR

Operasi ini adalah operasi membentuk string baru, yang merupakan bagian dari string yang diketahui.

Notasi : SUBSTR(S, i, j)
di sini : S = string yang diketahui
i dan j = integer
i = posisi awal substring $1 \leq i \leq \text{LENGTH}(S)$
j = banyak karakter yang diambil
 $0 \leq j \leq \text{LENGTH}(S)$ dan $0 \leq i+j-1 \leq \text{LENGTH}(S)$

Contoh :

Diberikan S = 'a₁a₂ ... a_N' ; i = 2 ; j = 4
Maka SUBSTR(S,i,j) = SUBSTR(S,2,4) = 'a₂a₃a₄a₅'

- String S = "UNIVERSITASGUNADARMA"
SUBSTR(S,i, j) , i = 7 j = 8
SUBSTR(S,4,8) = "SITASGUN"
- String S = "UNIVERSITAS"
SUBSTR(S,4,5) = "VERSI"
LENGTH(SUBSTR(S,4,5)) = 5
- String S = "GUNADARMA"
SUBSTR(S,3,4) = "NADA"
LENGTH(SUBSTR(S,3,4)) = 4

Catatan :

1. LENGTH(SUBSTR(S,i,j)) = j
2. SUBSTR(CONCAT(S₁,S₂),1,LENGTH(S₁)) = S₁
3. SUBSTR(CONCAT(S₁,S₂),LENGTH(S₁)+1,LENGTH(S₂)) = S₂

INSERT

Operasi ini adalah untuk menyisipkan suatu string ke dalam string lain.

Bentuk umumnya adalah :

INSERT(S₁,S₂,i). S₁ dan S₂ masing-masing adalah suatu string dan i adalah posisi awal S₂ pada S₁.

Contoh :

Misalkan : S₁ = 'a₁a₂ ... a_N'
S₂ = 'b₁b₂ ... b_M'
INSERT(S₁, S₂,3) = 'a₁a₂b₁b₂ ... b_Ma₃a₄... a_N'

String S₁ = "UNIVERSITAS"

String S₂ = "GUNADARMA"

INSERT(S₁,S₂,4) = "UNIGUNADARMAVERSITAS"

INSERT(S₂,S₁,4) = "GUNUNIVERSITASADARMA"

DELETE

Operasi ini digunakan untuk menghapus sebagian karakter dalam suatu string.

Bentuk umumnya adalah :

DELETE(S,i,j) → menghapuskan sebagian karakter dalam string S, mulai dari posisi *i* dengan panjang *j*.

Contoh :

Diberikan string $S = 'a_1 a_2 \dots a_N'$

DELETE(S,3,4) = $'a_1 a_2 a_7 a_8 \dots a_N'$

- String S = "UNIVERSITAS"
i = 4, j = 5
DELETE(S,i,j) = "UNITAS"
DELETE(S,j,i) = "UNIVTAS"
- String S = "GUNADARMA"
DELETE(S, 3, 5) = "GUMA"
DELETE(S, 5, 3) = "GUNAMA"

DEKLARASI DALAM BAHASA PEMROGRAMAN

♣ PASCAL

Var Count : integer;
Switch : boolean;
Betha : char;
Alamat : packed array [1..25] of char;

♣ COBOL

DATA DIVISION
01 Count PICTURE S999.
01 Flda PICTURE X.
88 Switch VALUE 'Y'.
01 Betha PICTURE X.
01 Alamat PICTURE X(25).

MAPPING KE STORAGE

♣ INTEGER

Bentuk mapping ke storage dari integer dapat dilakukan dengan beberapa cara, yaitu :

1. Skema Sign and Magnitude
2. Skema One's Complement
3. Skema Two's Complement

♥ SKEMA SIGN AND MAGNITUDE

Cara ini merupakan bentuk konvensional yang digunakan manusia untuk menyatakan suatu bilangan dalam bentuk biner. Di sini representasi bilangan positif dan negatif hanya dibedakan dengan tanda saja. Biasanya tanda positif atau negatif ditunjukkan oleh digit terdepan dari bentuk binernya, untuk representasi dengan jumlah digit tertentu.

Contoh :

$$\begin{aligned} +5 &\rightarrow +101 \text{ atau } 5 \rightarrow 101 \\ -5 &\rightarrow -101 \end{aligned}$$

Catatan : tanda (+) biasanya diabaikan

♥ SKEMA TWO'S COMPLEMENT

Jika x bilangan bulat non negatif maka x' bilangan binary negatif dari x sedemikian sehingga $x + x' = R$

$$R = 2^N$$

N = jumlah digit maksimum

$$x' = R - x$$

Contoh :

Bila $N = 4$, maka $R = 2^4 = 16$

$$x = 5 \rightarrow 0101$$

$$x' = R - x$$

$$= 16 - 5 = 11 \rightarrow 1011 \text{ (-5)}$$

♥ SKEMA ONE'S COMPLEMENT

Jika x bilangan bulat non negatif maka x' bilangan binary negatif dari x sedemikian sehingga $x + x' = R$

$$R = 2^N - 1$$

N = jumlah digit maksimum

$$x' = R - x$$

Contoh :

Bila $N = 4$, maka $R = 2^4 - 1 = 15$

$$x = 5 \rightarrow 0101$$

$$x' = R - x$$

$$= 15 - 5 = 10 \rightarrow 1010 \text{ (-5)}$$

Catatan

Untuk $R = 2^N$ dan $R = 2^N - 1$, bilangan bulat yang dapat disimpan dalam storage untuk ke-2 cara ini adalah :

$$\boxed{2^{(N-1)} - 1}$$

Untuk $R = 2^4$, bilangan bulat terbesar = $2^3 - 1$, maka $r = 2^4$ merepresentasikan bilangan dari -7 sampai dengan +7

INTEGER	SIGN & MAGNITUDE	TWO'S COMPLEMENT	ONE'S COMPLEMENT
-7	-111	1001	1000
-6	-110	1010	1001
-5	-101	1011	1010
-4	-100	1100	1011
-3	-011	1101	1100
-2	-010	1110	1101
-1	-001	1111	1110
0	000	0000	0000
1	001	0001	0001
2	010	0010	0010
3	011	0011	0011
4	100	0100	0100
5	101	0101	0101
6	110	0110	0110
7	111	0111	0111

♣ KARAKTER

Ada banyak skema yang digunakan untuk merepresentasikan karakter dalam storage. Pada umumnya skema yang paling banyak digunakan adalah :

1. Extended Binary Coded Decimal Interchange (EBCDIC)
Digunakan kode 8 bit untuk menyatakan sebuah karakter. Jika dihitung, kemungkinan kombinasi seluruhnya : $2^8 = 256$.
2. American Standard Code for Information Interchange (ASCII)
Digunakan kode 7 bit untuk menyatakan sebuah karakter. Jika dihitung, kemungkinan kombinasi seluruhnya : $2^7 = 128$.

♣ STRING

Untuk mengetahui bentuk mapping pada storage dari suatu string, perlu diketahui beberapa hal yang menyangkut ruang untuk string yang bersangkutan antara lain :

- letak posisi awal (start) dan posisi akhir (terminal)
- suatu pointer yang menunjukkan lokasi pada storage

Ada tiga cara yang umum digunakan untuk mapping suatu string ke dalam storage.

Misal diberikan dua string, yaitu :

$S_1 = \text{'ABCDEFGH'}$ dan $S_2 = \text{'BCD'}$

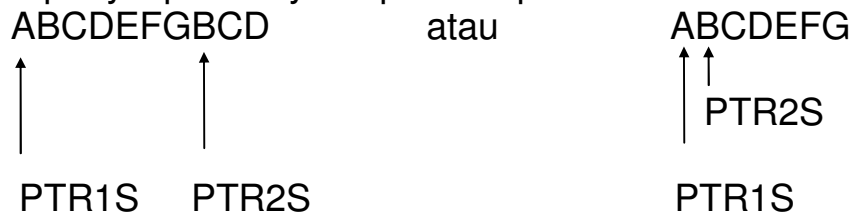
❖ CARA 1

Menggunakan tabel informasi :

- nama string (NAME)
- alamat awal (START)
- panjang string (LENGTH)

NAME	START	LENGTH
STRING1	PTR1S	7
STRING2	PTR2S	3

Format penyimpanannya dapat berupa :



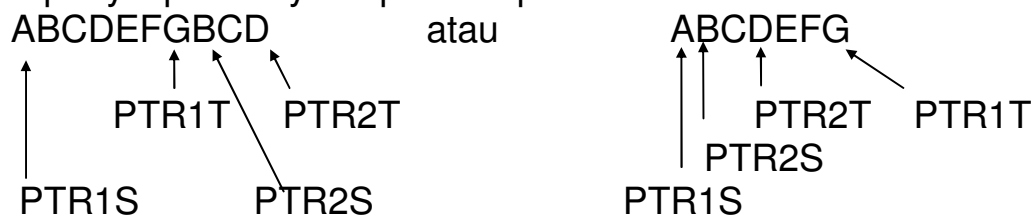
❖ CARA 2

Menggunakan tabel informasi :

- nama string (NAME)
- alamat awal (START)
- alamat akhir (TERM)

NAME	START	TERM
STRING1	PTR1S	PTR1T
STRING2	PTR2S	PTR2T

Format penyimpanannya dapat berupa :



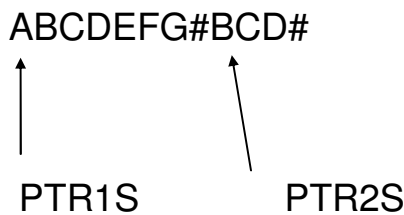
❖ CARA 3

Menggunakan tabel informasi :

- nama string (NAME)
- alamat awal (START)
- suatu tanda yang menunjukkan batas string

NAME	START
STRING1	PTR1S
STRING2	PTR2S

Penyimpanannya :



Cara lain yaitu : 1. Packed
2. Unpacked

Suatu string yang direpresentasikan dalam bentuk packed terbagi atas beberapa word. Banyaknya karakter untuk masing-masing word tergantung dari kode yang digunakan oleh mesin (bit-nya).

Secara umum jumlah word yang digunakan untuk merepresentasikan string S dalam storage dengan K karakter per word adalah :

$$\left\lceil \frac{\text{LENGTH}(S)}{K} \right\rceil$$

Contoh :

Misal diberikan string S = "UniversitasGunadarma", direpresentasikan dalam 4 karakter per word dalam bentuk packed. Maka secara fisik dapat digambarkan :

Univ	ersi	tasG	unad	arma
------	------	------	------	------

Jumlah word : 5
Jumlah karakter/word : 4

Sedangkan cara unpacked, setiap word terdiri hanya satu karakter, berarti jumlah word yang diperlukan untuk merepresentasikan suatu string S adalah : LENGTH(S)

Contoh :

Diberikan string S = "Gunadarma". Representasinya dalam bentuk unpacked adalah : LENGTH(S) = 9

G	u	n	a	d	a	r	m	a
---	---	---	---	---	---	---	---	---