

STRUKTUR DATA



**UNIVERSITAS
INABA**

MODUL - Sesi 1
STRUKTUR DATA

M THAMRIN BASRI

STRUKTUR DATA & ARRAY

Pengertian Struktur Data

Struktur Data adalah : suatu koleksi atau kelompok data yang dapat dikarakteristikan oleh organisasi serta operasi yang didefinisikan terhadapnya. Pemakaian Struktur Data yang tepat didalam proses pemrograman, akan menghasilkan Algoritma yang lebih jelas dan tepat sehingga menjadikan program secara keseluruhan lebih sederhana.

Konsep Dasar Tipe Data

Pada garis besarnya, Data dapat dikategorikan menjadi :

1. Type Data Sederhana (TIPE DATA ATOMIK)

Terdiri dari :

a. Data Sederhana Tunggal

Misalnya : Integer, Real/Float, Boolean dan Character

1) Integer

Merupakan Bilangan Bulat dan tidak mengandung pecahan. seperti :

...-3,-2,-1,0,1,2,3,....

Type	Batas nilai	Ukuran Memori
Short int	-32768....32767	2 Byte
Int	- 32768 ... 32767	2 Byte
Long	- 2147483678 ... 2147283647	4 Byte
Unsigned integer	0-65535	2 Byte

2) Real/Float

Type data yang merupakan bilangan pecahan. Jenis Data float ditulis dgn menggunakan titik(koma) desimal. Misalnya : 0.32 4,35 -131.128

Type Real dapat juga ditulis dengan Rumus :

$$M * R^e = X$$

M = Pecahan, R = Radix,

e = Exponen, X = Hasil Bilangan,

Misalnya : $3.2 * 10^{-1} = 0.32$

$4.35 * 10^2 = 435$

Type	Batas nilai	Format
float	3.4E-38 s/d 3.4E+38	unsigned 32 bit
double	1.7E-308 s/d 1.7E+308	unsigned 64 bit
Long double	3.4E-4932 s/d 1.1E+4932	unsigned 80 bit

3) Boolean

Type data yang hanya mempunyai dua bentuk keluaran yaitu nilai **True** dan **False** (Benar dan Salah) yang dinyatakan dengan 1 dan 0, Sehingga satuan data yang terpakai cukup satu bit saja. Operator yang digunakan adalah : And, Or dan Not.

Input			NOT (!)			AND (&&)	OR ()
A	B	C	!A	!B	!C	A&&B&&C	A B C
0	0	0	1	1	1	0	0
0	0	1	1	1	0	0	1
0	1	0	1	0	1	0	1
0	1	1	1	0	0	0	1
1	0	0	0	1	1	0	1
1	0	1	0	1	0	0	1
1	1	0	0	0	1	0	1
1	1	1	0	0	0	1	1

4) Character

Type data yang terdiri dari aksara (simbol) yang meliputi digit numerik, character alfabetik dan special character. Untuk menuliskan tipe char, karakter perlu ditulis di dalam tanda petik tunggal (').

Contoh :

'A' _ karakter berupa huruf A

'1' _ karakter berupa angka 1

'*' _ karakter 3ymbol *

b. Data Sederhana

Majemuk Misalnya :

String

String merupakan type data majemuk yang terbentuk dari kumpulan character sebanyak 256 (default) dengan jangkauan nilai 0 - 255. Kumpulan character yang digunakan untuk membentuk String dinamakan *alfabet*.

Pemberian nilai String diapit dengan tanda petik ganda ("). **Bentuk umum**

penulisan tipe data ini adalah : tipe_data pengenal [panjang] ;

pengenal = nama variabel

panjang = bilangan bulat yg menunjukkan jumlah karakter

Contoh : char nama[15] ;

Fungsi pada Operasi STRING

1) Strcpy()

Digunakan untuk menyalin nilai string.

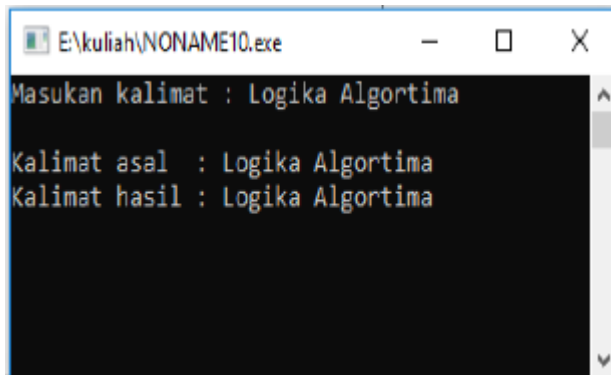
Contoh menggunakan program C++:

```

#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <stdio.h>
main()
{
char asal[100];
char hasil[100];
clrscr();
cout<<"Masukan kalimat : "; gets(asal);
strcpy(hasil,asal);cout<<endl;
cout<<"Kalimat asal : "<<asal<<endl;
cout<<"Kalimat hasil : "<<hasil<<endl;
getch(); }

```

Hasil Tampilan dari Program diatas sebagai berikut :



2) Strcat

Digunakan untuk menggabungkan nilai string.

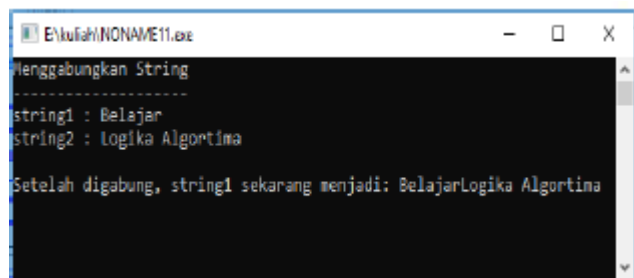
Contoh menggunakan program c++:

```

int main() {
char string1 [] ="Belajar";
char string2 [] ="Logika Algortima";
cout<<"Menggabungkan String"<<endl;
cout<<"----- "<<endl;
cout<<"string1 : "<<string1<<endl;
cout<<"string2 : "<<string2<<endl;
strcat(string1, string2);
cout<<"\nSetelah digabung, string1 sekarang menjadi:
"<<string1<<endl;
getche(); }

```

Hasil Tampilan dari Program diatas sebagai berikut :



```
E:\kuliah\NONAME11.exe
Menggabungkan String
-----
string1 : Belajar
string2 : Logika Algoritma

Setelah digabung, string1 sekarang menjadi: BelajarLogika Algoritma
```

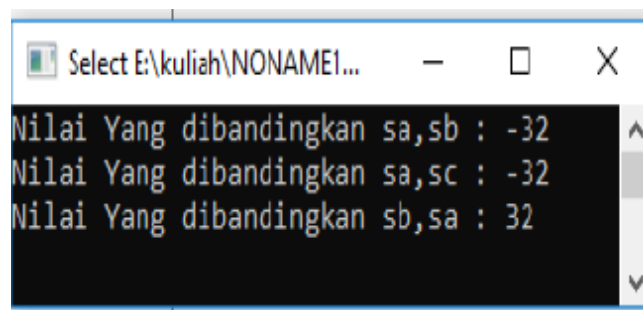
3) Strcmp

Digunakan untuk membandingkan 2 nilai string.

Contoh dalam penggalan program c++:

```
#include <iostream.h>
#include <stdio.h>
#include <conio.h>
main()
{
char sa[]="Logika";
char sb[]="Logika Algoritma";
char sc[]="Logika Algoritma & Pemrograman";
/*Melakukan perbandingan terhadap dua string dan penampilan
nilainya*/
printf("Nilai Yang dibandingkan sa,sb : %d\n",strcmp(sa,sb));
printf("Nilai Yang dibandingkan sa,sc : %d\n",strcmp(sa,sc));
printf("Nilai Yang dibandingkan sb,sa : %d\n",strcmp(sb,sa));
getch();
return 0;
}
```

Hasil Tampilan dari Program diatas sebagai berikut :



```
Select E:\kuliah\NONAME1...
Nilai Yang dibandingkan sa,sb : -32
Nilai Yang dibandingkan sa,sc : -32
Nilai Yang dibandingkan sb,sa : 32
```

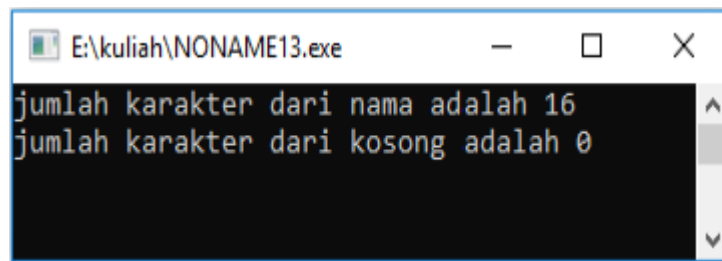
4) Strlen

Digunakan untuk mengetahui panjang nilai string.

Contoh dalam penggalan program c++:

```
#include <iostream.h>
#include <conio.h>
#include <string.h>
main()
{
char nama[50] = "Logika Algoritma";
char kosong[50] = "";
clrscr();
cout << "jumlah karakter dari nama adalah " << strlen(nama) << endl;
cout << "jumlah karakter dari kosong adalah " << strlen(kosong) << endl;
getch();}
```

Hasil Tampilan dari Program diatas sebagai berikut :



5) Strchr

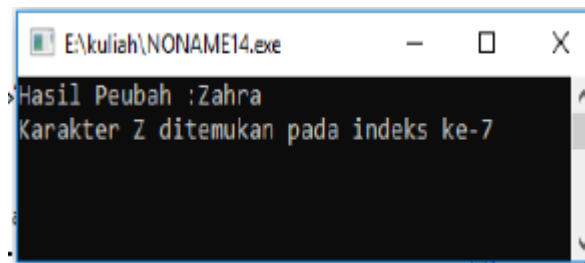
Digunakan untuk mencari nilai karakter dalam string.

Contoh dalam penggalan program C++:

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
int main(void){
char str [100]="Aisyah Zahra";
char karakter='Z';
char *hasil;
hasil=strchr(str,karakter);
printf("Hasil Peubah :%s\n",hasil);
```

```
printf("Karakter %c ditemukan pada indeks ke-%d",karakter,(hasil-
str));
getch();
return 0; }
```

Hasil Tampilan dari Program diatas sebagai berikut :



2. Struktur Data

Terdiri dari :

a. Struktur Data Sederhana

Misalnya Array dan Record

1) Array

Adalah tipe terstruktur yang terdiri dari sejumlah komponen yang mempunyai tipe yang sama. **Jenis Array** dibedakan menjadi 3 jenis: array 1 dimensi, 2 dimensi dan multidimensi

2) Record

Sebuah record merupakan koleksi satuan data yang heterogen, yakni terdiri dari berbagai type. Satuan data sering disebut sebagai field dari record. Field dipanggil dengan namanya masing-masing.

b. Struktur Data Majemuk

Terdiri dari :

a. Linier

Misalnya : Stack, Queue dan Linear Linked List.

b. Non Linier

Misalnya : Pohon (Tree), Pohon Biner (Binary Tree), Pohon Cari Biner (Binary Search Tree), General Tree serta Graph.

Konversi Bilangan

1. Decimal adalah bilangan berbasis sepuluh yang terdiri dari 0, 1, 2, 3, 4, 5, 6, 7, 8, dan 9
2. Hexadecimal adalah bilangan berbasis enam belas yang terdiri dari 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, dan F

Tabel di bawah adalah contoh **konversi bilangan** Decimal, dan Hexadecimal

Decimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Hexadecimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

Contoh KONVERSI ANTAR BILANGAN

Konversi Bilangan Decimal ke Hexadecimal

Contoh $254_{(10)} = \dots\dots\dots_{(16)}$

Caranya dengan membagi bilangan tersebut dengan enam belas sampai bilangan tersebut tidak bisa lagi dibagi enam belas (kurang dari enam belas) dengan mencatat setiap sisa pembagian.

$254 : 16 = 15$ sisa 14 atau E (lihat tabel di atas)

$15 : 16 =$ sisa 15 atau F (lihat tabel di atas)

Jadi $254_{(10)} = FE_{(16)}$ diurutkan dari sisa pembagian terakhir.

Array

1. Pengertian Array

Array / Larik : Struktur Data Sederhana yang dapat didefinisikan sebagai pemesanan alokasi memory sementara pada komputer. Array dapat didefinisikan sebagai suatu himpunan hingga elemen yang teratur dan homogen. *Terurut* : Dapat diartikan bahwa elemen tersebut dapat diidentifikasi sebagai elemen pertama, elemen kedua dan seterusnya sampai elemen ke-n. *Homogen* : Adalah bahwa setiap elemen dari sebuah Array tertentu haruslah mempunyai type data yang sama. Sebuah Array dapat mempunyai elemen yang seluruhnya berupa integer atau character atau String bahkan dapat pula terjadi suatu Array mempunyai elemen berupa Array.

2. Karakteristik Array

- a. Mempunyai batasan dari pemesanan alokasi memory (Bersifat Statis)
- b. Mempunyai Type Data Sama (Bersifat Homogen)
- c. Dapat Diakses Secara Acak

Tiga hal yang harus diketahui dalam mendeklarasikan array :

- a. Type data array
- b. Nama variabel array
- c. Subskrip / index array

3. Jenis Array (yang akan dipelajari) adalah :

- a. Array Dimensi Satu (One Dimensional Array)

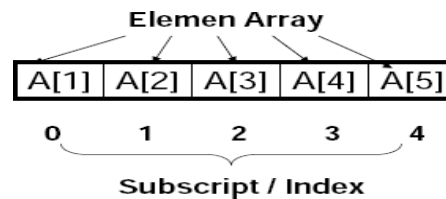
1) Deklarasi Array Dimensi Satu

Dapat disebut juga dengan istilah vektor yang menggambarkan data dalam suatu urutan.

Deklarasi : Type_Data Nama_Variabel [index]

Misalnya : int A[5];

Penggambaran secara Logika :



Contoh :

```
void main()
{ int bil [5];
  clrscr;
  cout<<"Masukkan 5 bilangan genap : "<<endl;
  for (int i = 0; i < 5; i++)
  { cout<<"Bilangan ";
    cout<< i + 1 <<" : ";
    cin>> bil[i];
    cout<<endl;
  }
  cout<<endl;
  cout<<"5 bilangan genap yang dimasukkan "<<endl;
  for (int i = 0; i < 5; i++)
  cout<<" "<<bil[i];
  getch();
}
```

```
D:\VCPP\array1.exe
Masukkan 5 bilangan genap :
Bilangan 1 : 45
Bilangan 2 : 50
Bilangan 3 : 100
Bilangan 4 : 75
Bilangan 5 : 30

5 bilangan genap yang dimasukkan :
 45 50 100 75 30
```

Gambar 1.1 Tampilan Program array dimensi 1

- 2) Rumus untuk menentukan jumlah elemen dalam Array :

$$\sum_{i=1}^n \pi (\text{Elemen Array})$$

Π = Perkalian dari elemen sebelumnya (untuk array dimensi dua & tiga)

Contoh :

Suatu Array A dideklarasikan sbb : `int A[10]`; maka jumlah elemen Array dimensi satu tersebut adalah = 10

- 3) Pemetaan Array Dimensi 1

$$\text{Rumus : } @A[i] = B + (i - 1) * L$$

Dimana :

$@A[i]$: Posisi Array yg dicari

B : Posisi awal index di memory komputer

I : Subkrip atau indeks array yg dicari

L : Ukuran / Besar memory suatu type data

Contoh :

Suatu Array A dideklarasikan sebagai berikut :

`int A[5]`; dengan alamat awal index berada di 0011 (H) dan ukuran memory type data integer = 2. Tentukan berapa alamat array A[3] ?

Rumus : $@A[i] = B + (i - 1) * L$

Diketahui :
 $@A[i] = A[3]$
 $B = 0011 \text{ (H)}$
 $i = 3$
 $L = 2$

Penyelesaian :
 $A[3] = 0011 \text{ (H)} + (3 - 1) * 2$
 $= 0011 \text{ (H)} + 4 \text{ (D)}$
 $= 0011 \text{ (H)} + 4 \text{ (H)}$
 $= 0015 \text{ (H)}$

4 Desimal = 4 Hexa

0	1	2	3	4
A[1]	A[2]	A[3]	A[4]	A[5]
0011	0013	0015	0017	0019

0	1	2	3	4	5	6	7	indeks
								value
21d2	21d4	21d6	21d8	21da	21dc	21de	21e0	alamat

```
#include <stdio.h>

void main() {
    int a[8];
    for (int i=0; i<8; i++) {
        printf("%x\n", &a[i]); %x adalah hexadesimal
    }
}
```

b. Array Dimensi Dua (Two Dimensional Array)

1) Deklarasi array Dimensi Dua

Sering digunakan dalam menterjemahkan matriks pada pemrograman.

Deklarasi : `Type_Data Nama_Variabel [Index1] [index2];`

Misal : `int A[3][2];`

Penggambaran secara Logika :

	0	1
0		
1		
2		

- 2) Menentukan jumlah elemen dalam Array dimensi dua:

$$\sum_{i=1}^n \pi (\text{Elemen array})$$

Π = Perkalian dari elemen sebelumnya (untuk array dimensi dua & tiga)

Contoh :

Suatu Array X dideklarasikan sbb : `int X[4][3]`, maka jumlah elemen Array dimensi dua tersebut adalah : $(4) * (3) = 12$

Terbagi Dua cara pandang (representasi) yang berbeda :

- a) **Secara Kolom Per Kolom (Coloumn Major Order/CMO)**

$$@M[i][j] = M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

- b) **Secara Baris Per Baris (Row Major Order / RMO)**

$$@M[i][j] = M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

Keterangan :

@M[i][j]: Posisi Array yg dicari,

M[0][0] : Posisi alamat awal index

array,I : Baris

j : kolom

L : Ukuran memory type data

K : Banyaknya elemen per kolom

N : Banyaknya elemen per baris

Penggambaran secara logika

Misal : `int M[3][2];` (Array dengan 3 Baris & 2 Kolom)

	0	1
0		
1		
2		

Berdasarkan Cara pandang :

1. Kolom Per Baris (Row Major Order / RMO)

M[0,0]	M[0,1]	M[1,0]	M[1,1]	M[2,0]	M[2,1]
--------	--------	--------	--------	--------	--------

Jumlah elemen per baris = 2

2. Baris Per Kolom (Coloumn Major Order / CMO)

M[0,0]	M[1,0]	M[2,0]	M[0,1]	M[1,1]	M[2,1]
--------	--------	--------	--------	--------	--------

Jumlah elemen per kolom = 3

3) Pemetaan Array Dimensi Dua

Contoh Pemetaan :

Suatu Array X dideklarasikan sebagai berikut : `Float X[4][3]`, dengan alamat index `X[0][0]` berada di `0011(H)` dan ukuran type data float = 4. Tentukan berapa alamat array `X[3][2]` berdasarkan cara pandang baris dan kolom ?

	0	1	2	← index
0	0011 _(H)			
1				
2		?		
3				

↑
index

Gambar 1.2 Contoh Pemetaan array dimensi dua

Secara Baris Per Baris (Row Major Oder / RMO)

$$@M[i][j] = @M[0][0] + \{(i - 1) * N + (j - 1)\} * L$$

$$\begin{aligned} X[3][2] &= 0011_{(H)} + \{(3 - 1) * 3 + (2 - 1)\} * 4 \\ &= 0011_{(H)} + 28_{(D)} \quad 1C_{(H)} \\ &= 0011_{(H)} + 1C_{(H)} \end{aligned}$$

$$= 002D_{(H)}$$

Secara Kolom Per Kolom (Coloumn Major Oder / CMO)

$$@M[i][j] = @M[0][0] + \{(j - 1) * K + (i - 1)\} * L$$

$$\begin{aligned} X[3][2] &= 0011_{(H)} + \{(2 - 1) * 4 + (3 - 1)\} * 4 \\ &= 0011_{(H)} + 24_{(D)} \quad 18_{(H)} \\ &= 0011_{(H)} + 18_{(H)} \end{aligned}$$

$$= 0029_{(H)}$$

Contoh Program Array Dimensi Dua :

```
#include<stdio.h>
#include<conio.h>
main()
{
    int a[3][5];
    for (int i=0;i<3;i++)
    {
        for (int j=0;j<5;j++)
        {
            printf("%x ",&a[j][i]);
        }
        printf("\n");
    }
    getch();
}
```

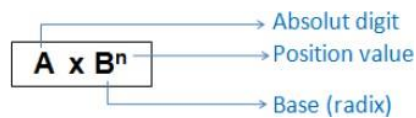


Gambar 1.3 Tampilan Program Array Dimensi Dua

SISTEM BILANGAN

Konsep Dasar Sistem Bilangan

Sistem bilangan adalah suatu cara untuk mewakili besaran dari suatu item fisik. Konsep dasar sistem bilangan dikarakteristikan oleh basis (radix), absolute digit dan posisi (*place*) value, yang dituliskan:



Basis yang digunakan sistem bilangan tergantung dari jumlah nilai bilangan yang dipergunakan.

1. Sistem Bilangan Desimal

Sistem bilangan desimal menggunakan basis 10 (*deca*). Menggunakan 10 macam simbol bilangan berbentuk digit angka: 0,1,2,3,4,5,6,7,8,9. Dasar penulisan:

$$A \times 10^n$$

Bentuk nilai desimal dapat berupa *integer* (bilangan bulat) dan pecahan. Dapat ditulis dalam bentuk eksponensial yaitu ditulis dengan *mantissa* dan *exponent*.

Contoh:

$$1234 = \underbrace{0,1234}_{\text{mantissa}} \times \underbrace{10^4}_{\text{exponent}}$$

Penulisan *base/radix* dituliskan setelah absolut digit, yaitu **A₁₀**, atau A(D).

Dalam hal ini yang dituliskan adalah **A₁₀**. Contoh nilai **4352₁₀** dan **762,15₁₀** dapat diartikan:

$$\begin{array}{rcl}
 4 \times 10^3 & = & 4000 \\
 3 \times 10^2 & = & 300 \\
 5 \times 10^1 & = & 50 \\
 2 \times 10^0 & = & \underline{2}_+ \\
 & & 4352
 \end{array}
 \qquad
 \begin{array}{rcl}
 7 \times 10^2 & = & 700 \\
 6 \times 10^1 & = & 60 \\
 2 \times 10^0 & = & 2 \\
 1 \times 10^{-1} & = & 0,1 \\
 5 \times 10^{-2} & = & \underline{0,05}_+ \\
 & & 762,15
 \end{array}$$

2. Sistem Bilangan Biner

Sistem bilangan biner menggunakan basis 2 (*binary*). Menggunakan 2 macam simbol bilangan berbentuk digit angka: 0 dan 1. Penulisan *base/radix* dituliskan setelah absolut digit, yaitu **A₂** atau A(B). Dalam hal ini yang dituliskan adalah **A₂**. Dasar penulisan:

$$A \times 2^n$$

Contoh penulisan: 1001 0011₂

3. Sistem Bilangan Oktal

Sistem bilangan oktal menggunakan basis 8 (*octal*). Menggunakan 8 macam simbol bilangan berbentuk digit angka: 0,1,2,3,4,5,6,7. Penulisan *base/radix* dituliskan setelah absolut digit, yaitu **A₈** atau A(O). Dalam hal ini yang dituliskan adalah **A₈**. Dituliskan:

$$A \times 8^n$$

Contoh penulisan: 347₈

4. Sistem Bilangan Hexadecimal

Sistem bilangan hexadesimal menggunakan basis 16 (*hexa*). Menggunakan 16 macam simbol bilangan berbentuk digit angka: 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F. Penulisan *base/radix* dituliskan setelah absolut digit, yaitu **A₁₆** atau A_(H). Dalam hal ini yang dituliskan adalah **A₁₆**. Dituliskan:

$$A \times 16^n$$

Contoh penulisan: A78₁₆

Satuan Data

Komputer bekerja atas dasar sistem biner berupa 0 dan 1 yang disebut bit. Bit merupakan satuan data terkecil dalam sistem komputer. Bit-bit dapat digunakan untuk menyusun karakter apa saja. Sebuah karakter dinyatakan dengan 8 bit atau 16 bit.

1. Byte

Byte merupakan satuan yang digunakan untuk menyatakan sebuah karakter pada sistem ASCII atau EBCDIC. 1 Byte = 8 bit.

2. Kilobyte (KB)

Biasa digunakan untuk berkas gambar berukuran kecil. 1 Kilobyte = 1024 byte

3. Megabyte (MB)

Biasa digunakan untuk menyatakan kapasitas RAM dalam PC.

$$1 \text{ MB} = 1024 \text{ KB} = 1.048.576 \text{ byte}$$

4. Gigabyte (GB)

Biasa digunakan untuk menyatakan kapasitas *harddisk* dalam PC.

1 GB = 1024 MB = 1.073.741.824 byte

5. Terabyte (TB)

Biasa digunakan untuk menyatakan kapasitas *harddisk* dalam *mainframe*.

1 TB = 1024 GB = 1.009.511.627.776 byte

6. Petabyte (PB)

1 PB = 1024 TB

Sistem Pengkodean

Sistem yang digunakan untuk mengkodekan karakter bermacam-macam. Data disimpan dalam memori komputer menempati posisi 1 byte, yang menggunakan kombinasi dari digit Biner. Komputer berbeda dalam menggunakan kode biner untuk mewakili sebuah karakter. Ada beberapa kode yang akan dibahas, yaitu BCD, EBCDIC, ASCII dan Unicode

1. BCD (Binary Coded Decimal)

Merupakan kode biner yang digunakan hanya untuk mewakili nilai digit desimal saja. Sebuah karakter BCD dinyatakan dengan 4 bit Karakter yang tersedia sebanyak 10 angka, yaitu angka 0,1,2,3,4,5,6,7,8,9. Digunakan pada komputer generasi pertama.

Tabel 2.1 BCD 4 bit

BCD 4 Bit	Digit Desimal	BCD 4 Bit	Digit Desimal
0000	0	0101	5
0001	1	0110	6
0010	2	0111	7
0011	3	1000	8
0100	4	1001	9

2. EBCDIC (*Extended Binary Coded Decimal Interchange Code*)

EBCDIC dikembangkan oleh IBM, yang diterapkan pada berbagai komputer *mainframe*. Sebuah karakter dinyatakan dengan 8 bit. Karakter yang tersedia sebanyak $2^8 = 256$ karakter. Digunakan pada komputer generasi ketiga.

Tabel 2.2 EBCDIC

HEX	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NUL	DLE	DS		SP	&										0
1	SOH	DC1	DOS						a	j			A	J		1
2	STX	DC2	FS	SYN					b	k	s		B	K	S	2
3	ETX	DC3							c	l	t		C	L	T	3
4	PF	RES	BYP	PN					d	m	u		D	M	U	4
5	HT	NL	LF	RS					e	n	v		E	N	V	5
6	LC	BS	ETB	UC					f	o	w		F	O	W	6
7	DEL	IL	ESC	EOT					g	p	x		G	P	X	7
8	CAN								h	q	y		H	Q	Y	8
9	RLF	EM							i	r	z		I	R	Z	9
A	SMM	CC	SM			!		:								
B	VT					\$	'	#								
C	FF	IFS		DC4	<	"	%	@								
D	CR	IGS	ENQ	NAK	()	.									
E	SO	IRS	ACK		□	:	>	=								
F	SI	IUS	BEL	SUB			?									

3. ASCII (*American Standard Code for Information Interchange*)

ASCII dikembangkan oleh ANSI (*American National Standard Institute*).

Sebuah karakter ASCII dinyatakan dengan 8 bit. Karakter yang tersedia

sebanyak 226 karakter, meliputi huruf, angka, dan spesial karakter, termasuk simbol Yunani dan karakter grafis.

Tabel 2.3 ASCII

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
128	80	À	160	A0	À	192	C0	À	224	E0	À
129	81	Á	161	A1	Á	193	C1	Á	225	E1	Á
130	82	Â	162	A2	Â	194	C2	Â	226	E2	Â
131	83	Ã	163	A3	Ã	195	C3	Ã	227	E3	Ã
132	84	Ä	164	A4	Ä	196	C4	Ä	228	E4	Ä
133	85	Å	165	A5	Å	197	C5	Å	229	E5	Å
134	86	Æ	166	A6	Æ	198	C6	Æ	230	E6	Æ
135	87	Ç	167	A7	Ç	199	C7	Ç	231	E7	Ç
136	88	È	168	A8	È	200	C8	È	232	E8	È
137	89	É	169	A9	É	201	C9	É	233	E9	É
138	8A	Ê	170	AA	Ê	202	CA	Ê	234	EA	Ê
139	8B	Ë	171	AB	Ë	203	CB	Ë	235	EB	Ë
140	8C	Ì	172	AC	Ì	204	CC	Ì	236	EC	Ì
141	8D	Í	173	AD	Í	205	CD	Í	237	ED	Í
142	8E	Î	174	AE	Î	206	CE	Î	238	EE	Î
143	8F	Ï	175	AF	Ï	207	CF	Ï	239	EF	Ï
144	90	Ð	176	B0	Ð	208	D0	Ð	240	F0	Ð
145	91	Ñ	177	B1	Ñ	209	D1	Ñ	241	F1	Ñ
146	92	Ò	178	B2	Ò	210	D2	Ò	242	F2	Ò
147	93	Ó	179	B3	Ó	211	D3	Ó	243	F3	Ó
148	94	Ô	180	B4	Ô	212	D4	Ô	244	F4	Ô
149	95	Õ	181	B5	Õ	213	D5	Õ	245	F5	Õ
150	96	Ö	182	B6	Ö	214	D6	Ö	246	F6	Ö
151	97	Ù	183	B7	Ù	215	D7	Ù	247	F7	Ù
152	98	Ú	184	B8	Ú	216	D8	Ú	248	F8	Ú
153	99	Û	185	B9	Û	217	D9	Û	249	F9	Û
154	9A	Ü	186	BA	Ü	218	DA	Ü	250	FA	Ü
155	9B	Ý	187	BB	Ý	219	DB	Ý	251	FB	Ý
156	9C	Þ	188	BC	Þ	220	DC	Þ	252	FC	Þ
157	9D	ß	189	BD	ß	221	DD	ß	253	FD	ß
158	9E	à	190	BE	à	222	DE	à	254	FE	à
159	9F	á	191	BF	á	223	DF	á	255	FF	á

4. Unicode

Sebuah karakter Unicode dinyatakan dengan 16 bit. Karakter yang tersedia sebanyak 65.536 karakter, meliputi huruf, angka, dan spesial karakter, termasuk simbol Yunani, karakter grafis, simbol Arab dan Cina.

Konversi Bilangan

1. Konversi dari Bilangan Desimal ke Biner

Dengan cara membagi bilangan desimal dengan 2 (basis biner) sampai tidak bisa dibagi lagi. Kemudian sisa pembagian diurutkan dari bawah ke atas dalam format 8 bit. Contoh nilai 89_{10} akan dikonversikan menjadi Biner.

89	
<u>2 :</u>	
44	sisa 1
<u>2 :</u>	
22	sisa 0
<u>2 :</u>	
11	sisa 0
<u>2 :</u>	
5	sisa 1
<u>2 :</u>	
2	sisa 1
<u>2 :</u>	
1	sisa 0

Dituliskan dari bawah ke atas: **1011001**
 Karena penulisan dengan 8 bit, maka
 $89_{10} = \mathbf{0101\ 1001}_2$

2. Konversi dari Bilangan Desimal ke Oktal

Dengan cara membagi bilangan desimal dengan 8 (basis oktal) sampai tidak bisa dibagi lagi. Cara yang digunakan sama dengan bilangan biner. Contoh nilai 147_{10} akan dikonversikan menjadi Oktal.

147	
<u>8 :</u>	
18	sisa 3
<u>8 :</u>	
2	sisa 2

Dituliskan dari bawah ke atas: **223**
 Maka hasilnya menjadi $147_{10} = \mathbf{223}_8$

3. Konversi dari Bilangan Desimal ke Hexadesimal

Dengan cara membagi bilangan desimal dengan 16 (basis hexa) sampai tidak bisa dibagi lagi. Cara yang digunakan sama dengan bilangan biner.

Contoh nilai 123_{10} akan dikonversikan menjadi Hexa

$$\begin{array}{r} 123 \\ \underline{16 :} \\ 7 \end{array} \quad \text{sisanya } 11 \uparrow$$

Sisa 11 dikodekan menjadi B

Maka hasilnya menjadi $123_{10} = 7B_{16}$

4. Konversi dari Bilangan Biner ke Desimal

Dengan cara mengalikan masing-masing bit biner dalam bilangan sesuai dengan *radix* dan *position value*-nya. Contoh bit $11\ 0101_2$ akan dikonversikan menjadi Desimal.

$$\begin{array}{l} 110101_2 \\ \begin{array}{l} \rightarrow 1 \times 2^0 = 1 \\ \rightarrow 0 \times 2^1 = 0 \\ \rightarrow 1 \times 2^2 = 4 \\ \rightarrow 0 \times 2^3 = 0 \\ \rightarrow 1 \times 2^4 = 16 \\ \rightarrow 1 \times 2^5 = 32 \end{array} \end{array}$$

Maka hasil di samping dituliskan:
 $11\ 0101_2 = 53_{10}$

53

5. Konversi dari Bilangan Biner ke Oktal

Dengan cara membagi digit biner tersebut ke dalam tiga digit dari kanan. Ketiga digit tersebut kemudian dikonversikan menjadi decimal. Contoh bit $1010\ 1011_2$ akan dikonversikan menjadi Oktal.

Biner	10	101	011
Desimal	2	5	3

Maka dituliskan menjadi $1010\ 1011_2 = 253_8$

6. Konversi dari Bilangan Biner ke Hexadesimal

Dengan cara membagi digit biner tersebut ke dalam empat digit dari kanan.

Keempat digit tersebut kemudian dikonversikan menjadi decimal. Contoh bit

10101011₂ akan dikonversikan menjadi Hexa.

Biner	1010	1011
Desimal	10	11
Hexa	A	B

Maka dituliskan menjadi **1010 1011**₂ = **AB**₁₆

7. Konversi dari Bilangan Oktal ke Desimal

Dengan cara mengalikan masing-masing bit octal dalam bilangan sesuai dengan

radix dan *position valuenya*. Contoh bit **371**₈ akan dikonversikan menjadi

Desimal.

$$\begin{array}{rcl}
 \text{371}_8 & \rightarrow & 1 \times 8^0 = 1 \\
 & \rightarrow & 7 \times 8^1 = 56 \\
 & \rightarrow & 3 \times 8^2 = \underline{192} + \\
 & & 249
 \end{array}$$

Maka hasil disamping
dituliskan:

$$\text{371}_8 = 249_{10}$$

8. Konversi dari Bilangan Oktal ke Biner

Dengan cara mengkonversikan setiap satu digit octal menjadi tiga digit biner.

Contoh bit **71**₈ akan dikonversikan menjadi Biner.

Oktal	7	1
Biner	111	001

Maka dituliskan menjadi **71**₈ = **0011 1001**₂

9. Konversi dari Bilangan Oktal ke Hexadesimal

Konversi ini tidak dapat dilakukan secara langsung, tetapi harus dikonversikan terlebih dahulu ke Desimal atau Biner. Contoh bit **243₈** akan dikonversikan menjadi Hexa.

Oktal	2	4	3
Biner	010	100	011
	1010		0011
Hexa	A		3

Maka dituliskan menjadi **243₈ = A3₁₆**

10. Konversi dari Bilangan Hexadesimal ke Desimal

Dengan cara mengalikan masing-masing bit hexa dalam bilangan sesuai dengan *radix* dan *position valuenya*. Contoh bit **8F₁₆** akan dikonversikan menjadi Desimal.

$$\begin{array}{l}
 8F_{16} \rightarrow \\
 \quad F \times 16^0 = 15 \\
 \quad 8 \times 16^1 = \underline{128} \\
 \quad \quad \quad 143
 \end{array}$$

Maka hasil disamping dituliskan:

$$8F_{16} = 143_{10}$$

11. Konversi dari Bilangan Hexadesimal ke Biner

Dengan cara mengkonversikan setiap satu digit hexa menjadi empat digit biner.

Contoh bit **8F₁₆** akan dikonversikan menjadi Biner

Hexa	8	F
Biner	1000	1111

Maka dituliskan menjadi **8F₁₆ = 1000 1111₂**

12. Konversi dari Bilangan Hexadesimal ke Oktal

Konversi ini tidak dapat dilakukan secara langsung, tetapi harus dikonversikan terlebih dahulu ke Desimal atau Biner sama dengan konversi dari oktal ke hexa.

Contoh bit **8F**₁₆ akan dikonversikan menjadi Oktal

Hexa	8		F
Biner	1000		1111
	10	001	111
Oktal	2	1	7

Maka dituliskan menjadi **8F**₁₆ = **217**₈

REPRESENTASI DATA

Pengertian Representasi Data

Komputer menggunakan dan memanipulasi data untuk perhitungan aritmatik, pemrosesan data, dan operasi logik. Type data yang digunakan dalam komputer digital diklasifikasikan:

1. Data Numerik: merepresentasikan integer, pecahan, real, dan desimal berkodebiner.
2. Data Logikal: digunakan oleh operasi seperti OR, AND, COMPLEMENT, COMPARE dan SHIFT.
3. Data Bit Tunggal: digunakan oleh operasi seperti SET, CLEAR, dan TEST.
4. Data Alfanyumerik: digunakan untuk manipulasi string oleh instruksi seperti MOVE dan SEARCH

Berikut ini merupakan Ilustrasi Representasi Data :



Gambar 3.1 Ilustrasi Representasi Data

Representasi Integer

Untuk keperluan pengolahan dan penyimpanan data komputer, hanya bilangan biner yang dapat merepresentasikan bilangan Integer direpresentasikan selain oleh nilai bilangannya juga dengan adanya tambahan tanda (Signed Integer). Berikut ini tipe-tipe representasi integer :

1. Representasi Sign and Magnitude

Merepresentasikan bilangan integer negative. Bit yang paling kiri diidentifikasi sebagai tanda (*sign*). Jika bit paling kiri adalah **nol** maka bilangan tersebut **positif**. Jika bit paling kiri adalah **satu** maka bilangan tersebut **negative**.

Contoh:

$$+18_{10} = 00010010_2$$

$$-18_{10} = 10010010_2$$

a. **Penjumlahan** pada Sign-Magnitude mempunyai aturan:

- 1) Sign tidak dijumlahkan, hanya magnitude.
- 2) Buang carry out dari bit yang paling kiri.
- 3) Jumlahkan yang sign-nya sama
- 4) Sign hasil = sign penambah

Contoh penjumlahan 4 bit:

0 0010 (2)	1 1011 (-11)
0 0101 (5)	1 0100 (-4)
+-----	+-----
0 0111 (7)	1 1111 (-15)

b. Pengurangan pada Sign-Magnitude mempunyai aturan:

- 1) Lakukan pengurangan jika sign sama
- 2) Jika sign tidak sama, ubah soal ke

penjumlahan Contoh pengurangan:

0 10100 (20)	1 1011 (-11)
0 00101 (5)	1 0100 (-4)
$\begin{array}{r} 010100 \\ 000101 \\ \hline 011111 \end{array}$	$\begin{array}{r} 11011 \\ 10100 \\ \hline 10111 \end{array}$
0 1111 (15)	1 0111 (-7)

Kelemahan Sign-Magnitude:

Penambahan dan pengurangan memerlukan pertimbangan baik tanda bilangan maupun nilai relatifnya. Ada dua representasi bilangan nol, yaitu

$$+0_{10} = 00000000_2$$

$$-0_{10} = 10000000_2$$

Representasi Komplemen Satu (One's Complement)

Komplemen pada dasarnya merubah bentuk pengurangan menjadi pertambahan. Komplementasi bilangan biner dengan cara mengubah 1 menjadi

0 dan 0 menjadi 1. Contoh:

$$00110110 = 11001001$$

Representasi Komplemen Dua (Two's Complement)

Dibentuk dengan mengambil komplemen satu dari bilangannya dan dengan menambahkan 1 pada posisi paling kanan. Contoh desimal 49 (dalam biner) menjadi bentuk komplemen dua:

```

110001
001110  → bentuk komplemen ke-1
  _____ 1+
001111  → bentuk komplemen ke-2

```

Penjumlahan Biner

Penjumlahan bilangan biner dilakukan sama seperti penjumlahan bilangan-bilangan desimal. Operasi pengurangan, perkalian dan pembagian seperti yang dilakukan pada komputer dan kalkulator digital sesungguhnya menggunakan penjumlahan sebagai operasi dasarnya.

Ada 4 kondisi dalam penjumlahan bilangan biner:

$$0 + 0 = 0$$

$$1 + 0 = 1$$

$$0 + 1 = 1$$

$$1 + 1 = 0 \text{ (carry out 1)}$$

Maksud dari *carry out*, hasilnya tidak bisa memuat lebih dari 1 digit, tetapi disimpan ke dalam kolom sebelah yang lebih tinggi nilainya (digit paling kiri yang diabaikan).

1. Penjumlahan Biner dengan Komplemen Dua

Ada beberapa kasus yang dapat dilakukan dengan komplemen dua:

a. Kasus 1: Dua Bilangan Positif

Penjumlahan dari dua bilangan positif dilakukan secara langsung.

Contoh:

$$\begin{array}{rcl}
 8 & \rightarrow & 1000 \\
 4 & \rightarrow & 0100 \\
 \hline + & & \\
 12 & \rightarrow & 1100
 \end{array}$$

b. Kasus 2: Bilangan Positif dan Negatif yang lebih kecil

Contoh: $8 + (-4)$.

Caranya bilangan -4 akan diubah ke dalam bentuk komplement dua, sehingga biner 4 (0100) menjadi:

$$\begin{array}{rcl}
 \text{Komplemen satu} & : & 1011 \\
 \text{Komplemen dua} & : & 1011 \\
 & & \underline{1} \\
 & & 1100
 \end{array}$$

jadi penjumlahannya adalah:

$$\begin{array}{rcl}
 8 & \rightarrow & 1000 \\
 -4 & \rightarrow & 1100 \\
 \hline + & & \\
 4 & \rightarrow & 1\ 0100 \quad (\text{digit paling kiri diabaikan})
 \end{array}$$

c. Kasus 3: Bilangan Positif dan Negatif yang lebih besar

Contoh: $8 + (-11)$.

Caranya bilangan -11 akan diubah ke dalam bentuk komplement dua, sehingga biner 11 (1101) menjadi:

$$\begin{array}{rcl}
 \text{Komplemen satu} & : & 0010 \\
 \text{Komplemen dua} & : & 0010 \\
 & & \underline{1} \\
 & & 0011
 \end{array}$$

jadi penjumlahannya adalah:

$$\begin{array}{rcl}
 8 & \rightarrow & 1000 \\
 -11 & \rightarrow & 0011 \\
 \hline + & & \\
 -3 & \rightarrow & 1011 \quad (\text{bentuk biner dari } -3)
 \end{array}$$

d. Kasus 4: Dua Bilangan Negatif

Contoh: $-8 + (-7)$

Caranya bilangan -8 dan -7 akan diubah ke dalam bentuk komplement dua,

jadi biner 8 (1000) dan 7 (0111) menjadi:

Komplemen satu	: 0111	dan	1000
Komplemen dua	: 0111		1000
	$\begin{array}{r} 1 \\ \hline \end{array} +$		$\begin{array}{r} 1 \\ \hline \end{array} +$
	1000		1001

jadi penjumlahannya adalah:

-8	→	1000	
-7	→	$\begin{array}{r} 1001 \\ \hline \end{array} +$	
$\frac{-7}{-15} +$	→	1 0001	(digit paling kiri diabaikan)

UNIVERSITAS
INABA