

STRUKTUR DATA

Pertemuan 1

Aturan Perkuliahan

- Kehadiran Min 80%
- Quiz+Tugas 30%
- UTS 30%
- UAS 40%

Struktur Data adalah : Cara menyimpan atau merepresentasikan **data** di dalam komputer agar bisa dipakai secara efisien.

Data adalah representasi dari fakta dunia nyata.

Fakta atau keterangan tentang kenyataan yang disimpan, direkam atau direpresentasikan dalam bentuk tulisan, suara, gambar, sinyal atau simbol.



Dalam teknik pemrograman, struktur data berarti tata letak data yang berisi kolom-kolom data, baik itu kolom yang tampak oleh pengguna (user) ataupun kolom yang hanya digunakan untuk keperluan pemrograman yang tidak tampak oleh pengguna. Setiap baris dari kumpulan kolom-kolom tersebut dinamakan catatan (record). Lebar kolom untuk data dapat berubah dan bervariasi. Ada kolom yang lebarnya berubah secara dinamis sesuai masukan dari pengguna dan juga ada kolom yang lebarnya tetap.

Dengan sifatnya ini, sebuah struktur data dapat diterapkan untuk pengolahan database, misalnya untuk keperluan data keuangan, atau untuk pengolah kata (word processor) yang kolomnya berubah secara dinamis. Contoh struktur data dapat dilihat pada file-file spreadsheet, database, pengolahan kata, gambar yang dikompres, dan pemampatan file (kompres) dengan teknik tertentu yang memanfaatkan struktur data.

Secara garis besar type data dapat dikategorikan menjadi:

Type data sederhana.

- Type data sederhana tunggal, misalnya Integer, real, boolean dan karakter.
- Type data sederhana majemuk, misalnya String

Struktur Data, meliputi:

- Struktur data sederhana, misalnya array dan record.
- Struktur data majemuk, yang terdiri dari:

Linier : Stack, Queue, serta List dan Multilist

Non Linier : Pohon Biner dan Graph

Pemakaian struktur data yang tepat didalam proses pemrograman akan menghasilkan algoritma yang lebih jelas dan tepat, sehingga menjadikan program secara keseluruhan lebih efisien dan sederhana.

Struktur data yang standar yang biasanya digunakan dibidang informatika adalah:

- * List linier (Linked List) dan variasinya
- * Multilist
- * Stack (Tumpukan)
- * Queue (Antrian)
- * Tree (Pohon)
- * Graph (Graf)

TYPE DATA SEDERHANA (Dalam Program C/C++)

1. INTEGER

Merupakan Bilangan Bulat dan tidak mengandung pecahan. seperti : ...-3,-2,-1,0,1,2,3,....

Type data Integer

Type	Range	(Byte)
short Int	128 ... 127	1
integer	32768..32767	2
longInt	214783648..2147483647	4

2 . REAL / FLOATING POINT

- Type data yang merupakan bilangan pecahan.
- Merupakan Bilangan Bulat dan tidak mengandung Jenis Data Real/float ditulis dgn menggunakan pecahan. seperti : ...-3,-2,-1,0,1,2,3,....
- 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,

$$\begin{aligned} \text{Misalnya :} \quad & 3.2 * 10^{-1} = 0.32 \\ & 4.35 * 10^2 = 435 \end{aligned}$$

''' Type data REAL

Type	Range	(Byte)
Float	3.4×10^{-38} s/ d $3.4 \times 10^{+38}$	4
Double	1.7×10^{-308} s/ d $1.7 \times 10^{+308}$	8
Long Double	3.4×10^{-4932} s/ d $1.1 \times 10^{+4932}$	10

3. BOOLEAN ATAU LOGICAL

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, Not, Xor

INPUT			NOT(!)			AND(&&)	OR(II)
A	B	C	!A	!B	!C	A&&B&&C	A I B I 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

True False jika and maka hasilnya False

True True jika and maka hasilnya True

True False jika or maka hasilnya True

True True jika or maka hasilnya True

4. CHARACTER

Type data yang terdiri dari aksara (simbol) yang meliputi digit numerik, character alfabetik dan spesial character.

Untuk menuliskan tipe char, karakter perlu ditulis di dalam tanda petik tunggal (')

Contoh :

'A' karakter berupa huruf A

'5' karakter berupa angka 5

'*' karakter simbol *

5. 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 :

`char pengenal[panjang];`

pengenal = nama variabel

panjang = bilangan bulat yg menunjukkan jumlah karakter

Contoh : `char nama[15] ;`

Operator Dalam Bahasa C/C++

Operator Aritmatika	Keterangan
pow	Pangkat
sqrt	Mengitung akar
%	Sisa hasil bagi (modulus)
* , /	Perkalian,Pembagian
+ , -	Penjumlahan, Pengurangan

Operator Dalam Bahasa C/C++

Operator Pemberian Nilai Aritmatika	Keterangan
* =	Perkalian
/ =	Pembagian
% =	sis hasil bagi
+ =	Penjumlahan
- =	pengurangan

Operator Dalam Bahasa C/C++

Operator unary	Keterangan
+	Tanda plus
-	Tanda minus

Operator penambahan pengurangan	Keterangan
++	Penambahan
--	pengurangan

Operator Dalam Bahasa C/C++

Operator Relasi	Keterangan
=	Sama dengan (assingnmet)
!=	Tidak sama dengan
>	Lebih besar
<	Lebih kecil
==	Sama dengan (bukan assignment)
>=	Lebih besar atau sama dengan
<=	Lebih kecil atau sama dengan

Operator Bittwise	Keterangan
~	NOT
<<	Shift Left
>>	Shift Right
&	AND
^	XOR
	OR

⇒ SEKILAS TENTANG STRUKTUR DATA SEDERHANA

1. **Array(Larik)**

Larik adalah struktur data statik yang menyimpan sekumpulan elemen yang bertipe sama. Setiap elemen diakses langsung melalui indeksinya. Indeks larik harus tipe data yang menyatakan keterurutan misalnya integer atau karakter. Banyaknya elemen larik harus sudah diketahui sebelum program dieksekusi. Tipe elemen larik dapat berupa tipe sederhana, tipe terstruktur, atau tipe larik lain. Nama lain array adalah Larik, tabel, atau vektor. [ARRAY-1-2.pptx](#)

2. **Struktur (struct)**

Struktur adalah sebuah kumpulan variabel-variabel yang diwakili dengan sebuah nama. Setiap variabel dapat mempunyai tipe data yang sama atau berbeda-beda. Contohnya adalah objek mahasiswa terdiri dari nama, nim, jurusan, jenis kelamin dan tanggal lahir. Contoh lain, Objek Karyawan terdiri dari elemen-elemen nama karyawan, gaji pokok, gaji lembur, tunjangan anak, tunjangan suami istri, Ppn dsb. Elemen-elemen tersebut disebut dengan **field/atribut**. [STRUKTUR-1-3.pptx](#)

- Disusun oleh satu atau lebih field. Tiap field menyimpan data dari tipe dasar tertentu atau dari tipe bentukan lain yang sudah didefinisikan sebelumnya. Nama rekaman ditentukan oleh pemrogram.
- Rekaman disebut juga tipe terstruktur.
- Bermanfaat untuk mengelompokkan sejumlah data dengan tipe data yang berlainan.
- Contoh :

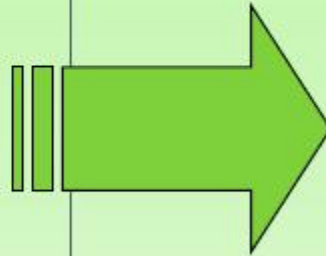
```
struct data_pegawai
{
int nip;
char nama[25];
char alamat[40];
}
struct data_pegawai datpeg;
```

TIPE DATA ARRAY

Pertemuan 1

Bagian 2

APA ITU ARRAY?



- Merupakan tipe terstruktur yang terdiri dari sejumlah komponen-komponen yang mempunyai tipe yang sama.
- Suatu array mempunyai jumlah komponen yang banyaknya tetap.
- Banyaknya komponen dalam suatu array ditunjukkan oleh suatu indek untuk membedakan variabel yang satu dengan variabel lainnya.

Konsep Dasar Array

- Setiap data yang terdapat dalam array tersebut menempati alamat memori yang berbeda disebut **elemen array**.
- Untuk mengakses nilai dari suatu elemen array, akan digunakan indeks dari array tersebut.
- Sangat perlu diperhatikan bahwa dalam bahasa C/C++, indeks array selalu dimulai dari angka 0, bukan 1.
- Hal ini berbeda dengan bahasa pemrograman lainnya (misalnya bahasa Pascal) dimana indeks awal array dapat ditentukan sendiri sesuai dengan keinginan kita.
- Untuk mendeklarasikan suatu array dalam bahasa C adalah dengan menggunakan tanda [] (*bracket*).

Bentuk Deklarasi :

```
tipe_data nama_array[elemen-1][elemen-2]...;
```

Contoh :

```
float Nil_Akhir[6]; //array dimensi satu
```


Dimensi Array (C/C++)

- Dimensi Satu
- Dimensi Dua
- Multidimensi

Array Dimensi Satu

Suatu array dapat digambarkan sebagai kotak panjang yang berisi kotak-kotak kecil didalam kotak panjang tersebut.



Inisialisasi array berdimensi satu adalah memberikan nilai awal terhadap suatu variabel. Bentuk pendefenisian suatu array dapat dilihat pada contoh berikut ini :

```
tipe_data nama_array [banyak_element] = { nilai array };  
float nilai[5] = {56, 66, 87, 98, 78}
```

```

//-----//
// Array Dimensi Satu //
//-----//

#include<stdio.h>
#include<conio.h>

int nilai[5]={56, 66, 87, 98, 78};

int main()
{
    int i;

    for(i=0;i<5;i++)
    {
        printf("Nilai Array Index ke - %d", i);
        printf(" = %d\n", nilai[i]);
    }
    getch();
    return 0;
}

```

Output :

```

Nilai Array Index ke - 0 = 56
Nilai Array Index ke - 1 = 66
Nilai Array Index ke - 2 = 87
Nilai Array Index ke - 3 = 98
Nilai Array Index ke - 4 = 78

```

Array Dimensi Dua

Array tidak hanya dapat dibentuk dalam dimensi satu, juga dapat dibentuk dalam dimensi dua. Bentuk penulisan array berdimensi dua adalah sebagai berikut :

```
tipe_data nama_array [elemen_1][elemen_2] = { nilai array };  
float nilai[2][5] = {2, 3, 4, 5, 2},{4, 2, 6, 2, 7};
```

```

//-----//
// Array Berdimensi Dua //
//-----//

#include<stdio.h>
#include<conio.h>

int nilai[2][5]={{2, 3, 4, 5, 2},
                {4, 2, 6, 2, 7}
                };

int main()
{
    int i, j;

    for(i=0;i<2;i++)
    {
        for(j=0;j<5;j++)
        {
            printf("%d", nilai[i][j]);
            printf(" ");
        }
        printf("\n");
    }

    getch();
    return 0;
}

```

Output :

2	3	4	5	2
4	2	6	2	7

```

#include<stdio.h>
#include<conio.h>

int matrik [4][4];

int main()
{
    int i, j;

    for(i=0;i<4;i++)
    {
        for(j=0;j<4;j++)
        {
            printf("Masukkan angka pada baris ke - %d", i);
            printf(" kolom ke - %d", j);printf(" = ");scanf("%d", &matrik[i][j]);
        }
        printf("\n");
    }
    printf("Ini matrik ordo [4,4] 4 baris 4 kolom");
    {
        printf("\n");
    }
    for(i=0;i<4;i++)
    {
        for(j=0;j<4;j++)
        {
            printf("%d", matrik[i][j]);printf(" ");
        }
        printf("\n");
    }
    getch();
    return 0;
}

```

```

Masukkan angka pada baris ke - 0 kolom ke - 0 = 3
Masukkan angka pada baris ke - 0 kolom ke - 1 = 5
Masukkan angka pada baris ke - 0 kolom ke - 2 = 7
Masukkan angka pada baris ke - 0 kolom ke - 3 = 8

Masukkan angka pada baris ke - 1 kolom ke - 0 = 2
Masukkan angka pada baris ke - 1 kolom ke - 1 = 4
Masukkan angka pada baris ke - 1 kolom ke - 2 = 6
Masukkan angka pada baris ke - 1 kolom ke - 3 = 8

Masukkan angka pada baris ke - 2 kolom ke - 0 = 1
Masukkan angka pada baris ke - 2 kolom ke - 1 = 2
Masukkan angka pada baris ke - 2 kolom ke - 2 = 3
Masukkan angka pada baris ke - 2 kolom ke - 3 = 4

Masukkan angka pada baris ke - 3 kolom ke - 0 = 9
Masukkan angka pada baris ke - 3 kolom ke - 1 = 8
Masukkan angka pada baris ke - 3 kolom ke - 2 = 7
Masukkan angka pada baris ke - 3 kolom ke - 3 = 6

Ini matrik ordo [4,4] 4 baris 4 kolom
3 5 7 8
2 4 6 8
1 2 3 4
9 8 7 6

```


TIPE DATA STRUKTUR

Pertemuan 1

Bagian 3

Struktur adalah sebuah kumpulan variabel-variabel yang diwakili dengan sebuah nama. Setiap variabel dapat mempunyai tipe data yang sama atau berbeda-beda.

Contohnya adalah objek mahasiswa terdiri dari nama, nim, jurusan, jenis kelamin dan tanggal lahir.

Contoh lain, Objek Karyawan terdiri dari elemen-elemen nama karyawan, gaji pokok, gaji lembur, tunjangan anak, tunjangan suami istri, Ppn dsb.

Elemen-elemen tersebut disebut dengan **field/atribut**.

Mendeklarasikan tipe data struktur

```
struct nama_struktur
{
  tipe_data1 daftar_field-1;
  tipe_data2 daftar_field-2;
  ...
  tipe_data_n daftar_field-n;
};
```

Mendefinisikan tipe data struktur

```
nama_struktur var_struktur1, var_struktur2, ...;
```

Contoh deklarasi dan definisi sebuah struktur :

```
struct data_mhs
{
  char nim[8], nama[20];
  int nil_ujian; char hm;
}; /*deklarasi struktur*/
struct data_mhs mhs; /*definisi struktur */
```

atau bisa juga dibuat dengan cara sbb :

```
struct data_mhs
{
  char nim[8], nama[20];
  int nil_ujian;
  char hm;
} mhs;
```

Mengakses tipe data struktur :

```
var_struktur.field
```

Contoh :

```
mhs.nil_ujian=85;
```

```
strcpy(mhs.nama,"Taufik Hidayat");
```

```
scanf("%s",&mhs.nim);
```

```

#include <stdio.h> //contoh menggunakan struktur
                #include <conio.h>

                struct data_mhs
                {
                char nim[10], nama[20];
                int nil_ujian;
                } mhs;

                main()
                {
                clrscr();
                printf("nomor induk mahasiswa : ");
                scanf("%s",&mhs.nim); fflush(stdin);
                printf("nama mahasiswa      : ");
                scanf("%s",&mhs.nama); fflush(stdin);
                printf("nilai ujian        : ");
                scanf("%d",&mhs.nil_ujian); fflush(stdin);

                if (mhs.nil_ujian>=80)
                puts("ANDA DINYATAKAN LULUS");
                else
                puts("ANDA DINYATAKAN GAGAL"); getch();
                }

```

Kadang-kadang dalam suatu struktur terdapat struktur yang lain (struktur bersarang). Sebagai contoh, perhatikan bentuk struktur dari objek karyawan berikut ini :

Karyawan

nama

Alamat

Tanggal Lahir :

Tanggal

Bulan

Tahun

```
struct data_kelahiran
```

```
{
```

```
int tgl,tahun;
```

```
char bulan[30];
```

```
};
```

```
struct karyawan
```

```
{
```

```
char nama[20];
```

```
char alamat[30];
```

```
data_kelahiran tanggal;
```

```
} data_kary;
```



```

#include <stdio.h>
#include <conio.h>
struct data_kelahiran
{ int tgl,tahun; char bulan[30]; };
struct karyawan
{ char nama[20]; char alamat[30];
  data_kelahiran tanggal; } data_kary;
main()
{ clrscr();
  printf("nama karyawan : ");
  scanf("%s",&data_kary.nama); fflush(stdin);
  printf("alamat      : ");
  scanf("%s",&data_kary.alamat); fflush(stdin);
  printf("tanggal lahir : ");
  scanf("%d",&data_kary.tanggal.tgl); fflush(stdin);
  printf("bulan lahir   : ");
  scanf("%d",&data_kary.tanggal.bulan); fflush(stdin);
  printf("tahun lahir   : ");
  scanf("%d",&data_kary.tanggal.tahun); fflush(stdin);

  getch();
}

```

Struktur berbentuk array

Tipe struktur dapat juga berbentuk sebuah array. Pada prinsipnya array pada struktur sama dengan tipe array biasa. Sebagai contoh struktur yang berbentuk array berikut ini :

```
struct data_mhs
{
  char nim[10], nama[20];
  int nil_ujian;
} mhs[10];
```

Definisi di atas mempunyai arti variabel struktur mhs memiliki 10 paket data_mhs yang terdiri dari :

mhs[0].nim mhs[0].nama mhs[0].nil_ujian
... mhs[9].nim mhs[9].nama mhs[9].nil_ujian

```

#include <stdio.h>
#include <conio.h>
struct data_mhs
{char nim[10], nama[20]; int nil_ujian;} mhs[100];
main()
{int i,jml_mhs;
 clrscr(); printf("jumlah mahasiswa : ");
 scanf("%d",&jml_mhs); fflush(stdin);
 /* proses memasukan data nilai mahasiswa */
 for(i=0;i<=jml_mhs;i++)
 {printf("data mahasiswa ke-%d\n",i+1);
 puts("-----");
 printf("nomor induk mahasiswa : ");
 scanf("%s",&mhs[i].nim); fflush(stdin);
 printf("nama mahasiswa : ");
 scanf("%s",&mhs[i].nama); fflush(stdin);
 printf("nilai ujian : ");
 scanf("%d",&mhs[i].nil_ujian); fflush(stdin);
 }getch();
}

```

Struktur sebagai parameter

Selain variabel, nilai, konstanta, struktur pun dapat dilewatkan ke dalam sebuah fungsi.

Perhatikan contoh di bawah ini :

```
#include <stdio.h>
#include <conio.h>
#include <string.h>
struct data
{char nim[5],nama[20];} mhs;
void contoh(data x);
main()
{ strcpy(mhs.nim,"001"); strcpy(mhs.nama,"ABC");
  contoh(mhs);
  getch();
}

void contoh(data x)
{
  printf("%s %s",x.nim, x.nama);
}
```