

PERTEMUAN 1:

TIPE DATA DAN HIRARKI

A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai jenis-jenis data dan hirarki tipe data. Di modul ini, Anda harus mampu:

- 1.1 Mengetahui macam-macam tipe data dan hirarki serta mampu merepresentasikan kedalam bahasa pemrograman dan membuat aplikasi.

B. URAIAN MATERI

<i>Tujuan Pembelajaran 1.1:</i>

<i>Macam-macam tipe data</i>

DEFINISI DATA

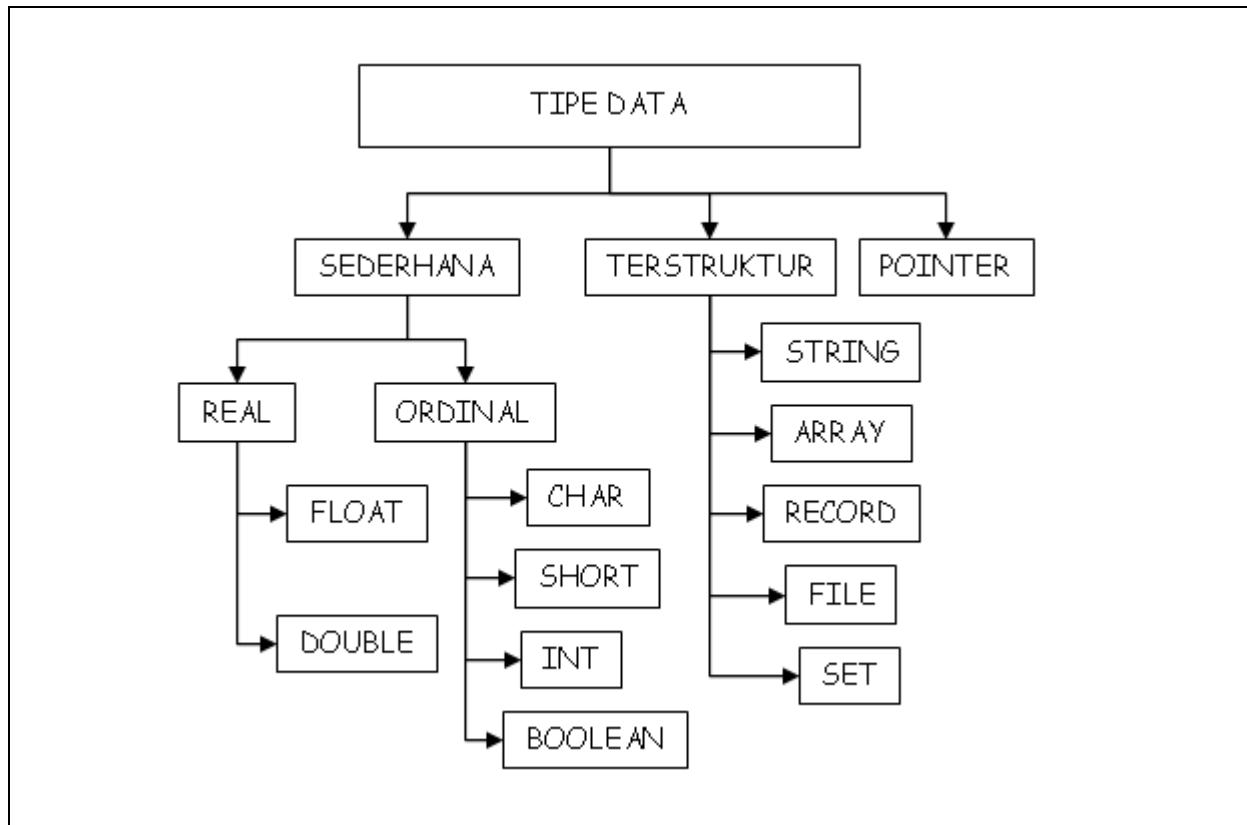
Fakta atau kenyataan yang tercatat mengenai suatu obyek. Pengertian data ini menyiratkan suatu nilai yang bisa dinyatakan dalam bentuk konstanta atau variabel

- ☐ Konstanta menyatakan nilai yang tetap
- ☐ Variabel digunakan dalam program untuk menyatakan nilai yang dapat diubah-ubah selama eksekusi berlangsung

EMPAT ISTILAH TENTANG DATA

- ☐ **TIPE DATA** : macam/isi data didalam suatu variabel
- ☐ **OBYEK DATA** : Himpunan dari elemen, misal : x himpunan bilangan integer
- ☐ **REPRESENTASI DATA** : Suatu mapping dari struktur data d kesuatu himpunan struktur data e, misal : boolean direpresentasikan dalam 0 dan 1
- ☐ **STRUKTUR DATA** : koleksi dari variabel yang dinyatakan dengan sebuah nama, dengan sifat setiap variabel dapat memiliki tipe yang berlainan. Struktur data biasa dipakai untuk mengelompokkan beberapa informasi yang berkaitan menjadi sebuah kesatuan.

HIERARKI TIPE DATA



PERINTAH OUTPUT

Bentuk Umum : `cout<<"Keterangan"<<variabel<<endl;`

```

/*Contoh Program*/
#include<iostream.h>
void main(void)
{
    Bil=90;
    cout<<"Bilangannya yaitu = "<<Bil<<endl;
}
    
```

Akan tampil :

Bilangannya yaitu = 90

PERINTAH INPUT

Bentuk Umum : `cin>>variabel;`

```
/*Contoh Program*/  
#include<iostream.h>  
void main(void)  
{  
    int A;  
    cout<<"Masukan A :";cin>>A;  
}
```

```
/*Contoh Program*/  
#include<iostream.h>  
void main(void)  
{  
    char NAMA[20];  
    cout<<"Masukan Nama :";cin>>NAMA;  
    cout<<"Namanya adalah:"<<NAMA<<endl;  
}
```

Hasilnya :

Masukan Nama :Susi Susanti[enter]
Namanya adalah : Susi

```
/*Contoh Program*/  
#include<iostream.h>  
void main(void)  
{  
    char NAMA[7];  
    cout<<"Masukan Nama :";cin.getline(NAMA,7);  
    cout<<"Namanya adalah : "<<NAMA<<endl;  
}
```

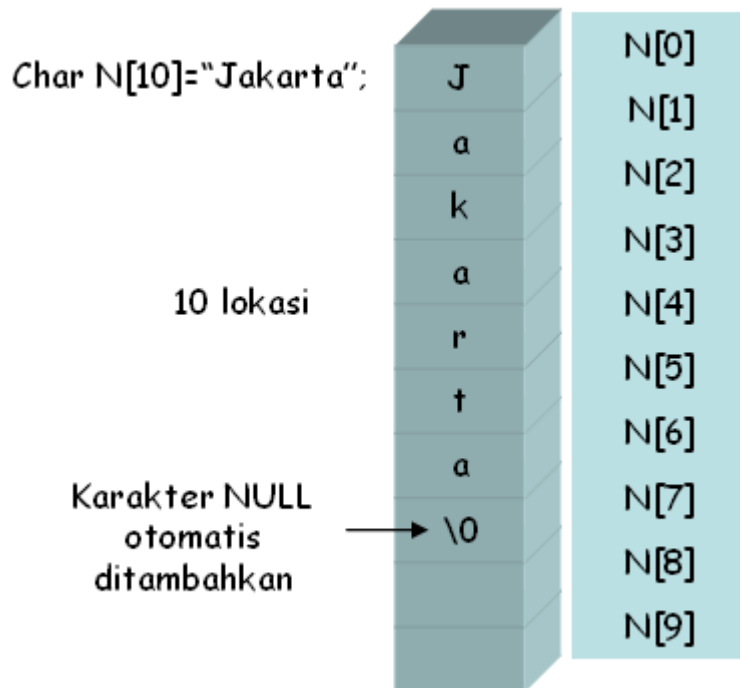
Hasilnya :

Masukan Nama :Susi Susanti[enter]
Namanya adalah : Susi S

TIPE DATA TERSTRUKTUR

1. String

Data yang berisi sederetan karakter dimana banyaknya karakter bisa berubah-ubah sesuai dengan kebutuhan

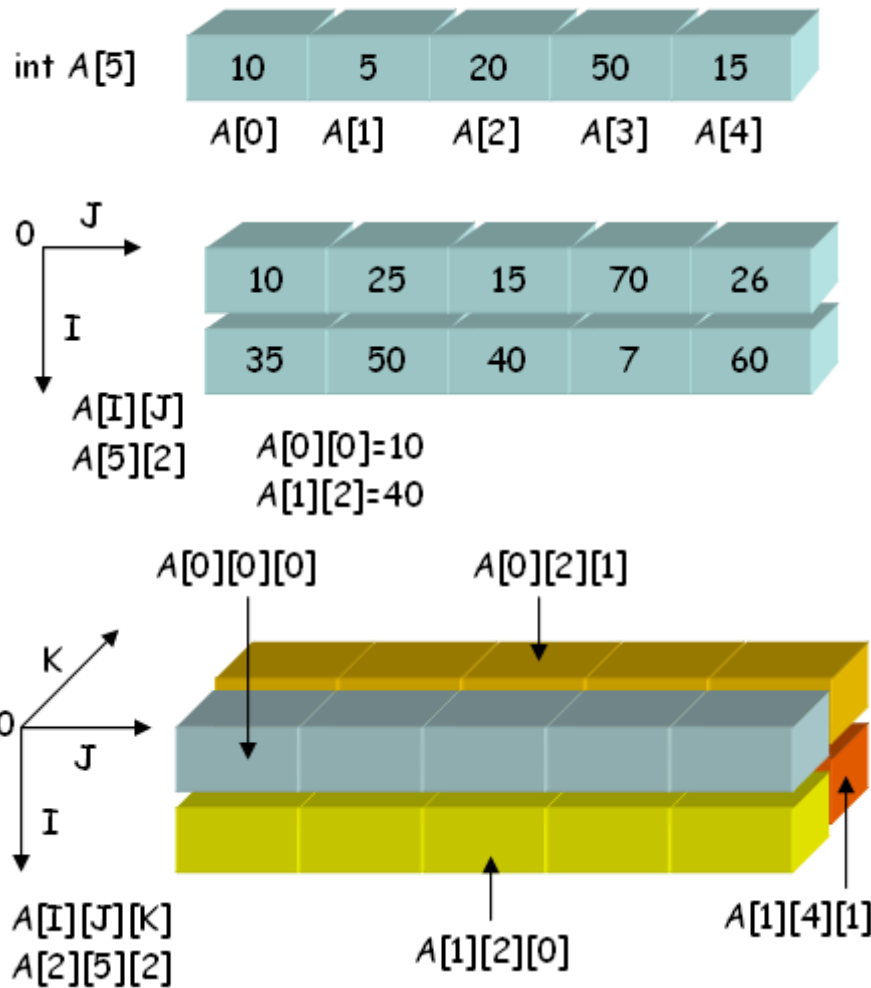


```
/*Contoh Program*/
#include<iostream.h>
void main(void)
{
    int i;
    char N[8]="Jakarta";
    for(i=0;i<=7;i++)
        cout<<"N["<<i<<"]"<<" = "<<N[i]<<endl;
}
```

```
N[0]=J
N[1]=a
N[2]=k
N[3]=a
N[4]=r
N[5]=t
N[6]=a
N[7]=
```

2. Larik(Array)

Array adalah variabel yang mampu menyimpan sejumlah nilai yang bertipe sama.



3. Record/Struktur

Terdiri dari beberapa variabel yang terstruktur dan masing-masing variabel bisa mempunyai tipe yang berbeda.

```
struct mahasiswa
{
    char *nama;
    char *nim;
    int uts,uas;
    float akhir;
    char grade;
}
```

4. Set -> Union

Berbeda dengan struktur, anggota dari union menggunakan secara bersama-sama ruang penyimpanan memori yang sama.

```
#include<iostream.h>
union BilBulat
{
    unsigned int bInt;
    unsigned char cKar[4];
};
void main(void)
{
    BilBulat Bilangan;
    Bilangan.bInt=0x56782233;
    cout<<"bInt : "<<hex<<Bilangan.bInt<<endl;
    cout<<"cKar[0] : "<<hex<<int(Bilangan.cKar[0])<<endl;
    cout<<"cKar[1] : "<<hex<<int(Bilangan.cKar[1])<<endl;
    cout<<"cKar[2] : "<<hex<<int(Bilangan.cKar[2])<<endl;
    cout<<"cKar[3] : "<<hex<<int(Bilangan.cKar[3])<<endl;
}
```

Hasilnya :

bInt : 56782233

cKar[0] : 33

cKar[1] : 22

cKar[2] : 78

cKar[3] : 56

4. Set -> Enumerasi

Merupakan himpunan dari konstanta integer yang diberi nama.

```
#include<iostream.h>
enum Manusia {Pria,Wanita};
enum Benda {Merah=1, Hijau, Biru};
void main(void)
{
    enum Manusia jns_kel;
    enum Benda Warna;

    jns_kel=Pria;cout<<jns_kel<<endl;
    jns_kel=Wanita;cout<<jns_kel<<endl;
    Warna=Merah;cout<<Warna<<endl;
}
```

Hasilnya :

0

1

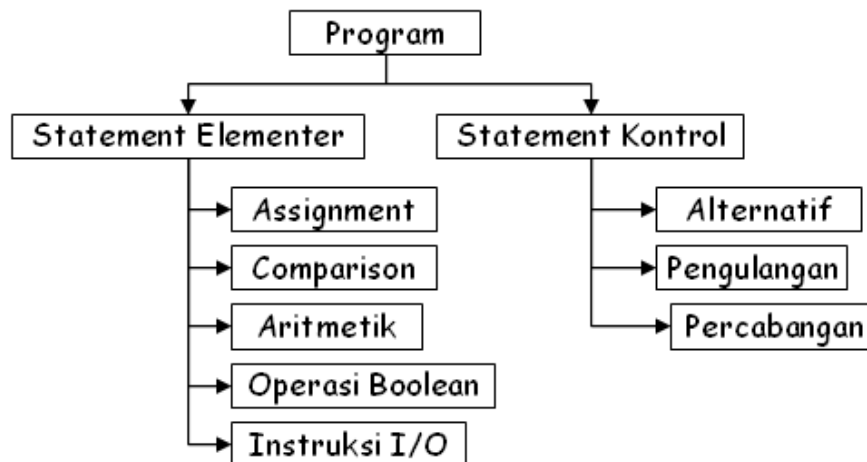
1

5. File

Merupakan organisasi dari sejumlah record sejenis. Masing-masing record dapat terdiri dari satu atau beberapa field dan setiap field terdiri dari satu atau beberapa karakter

PROGRAM

Kumpulan instruksi-instruksi yang ditulis dengan aturan tertentu yang dimengerti oleh komputer untuk melaksanakan suatu tugas.



1. Assignment(penugasan)

Untuk memberikan nilai ke variabel yang telah dideklarasikan
 Bil = 3;

2. Compariason

Untuk keperluan pengambilan keputusan. Diperlukan operator relasi sebagai berikut :
 >, <, >=, <=, ==, !=

Contoh : A = 5 dan B = 2

Kondisi	Nilai
A==B	FALSE
A > B	TRUE
A < B	FALSE
A >= B	TRUE
A <= B	FALSE
A != B	TRUE

Not Kondisi	Nilai
!(A==B)	TRUE
!(A > B)	FALSE
!(A < B)	TRUE
!(A >= B)	FALSE
!(A <= B)	TRUE
!(A != B)	FALSE

$C = (A < B); \text{cout} \ll C;$ Hasilnya = 1

Sehingga :

$!(A == B)$	sama dengan	$A != B$
$!(A > B)$	sama dengan	$A <= B$
$!(A < B)$	sama dengan	$A >= B$
$!(A >= B)$	sama dengan	$A < B$
$!(A <= B)$	sama dengan	$A > B$
$!(A != B)$	sama dengan	$A == B$

3. Arithmetic Statement

Operator : +, -, *, /, %

$C = 7$ dan $D = 3$

$E = C + D$ hasilnya 10

$E = C - D$ hasilnya 4

$E = C * D$ hasilnya 21

$E = C \% D$ hasilnya 1

$E = C / D$ hasilnya ??

4. Operasi Boolean/logika : menghubungkan ungkapan relasi yang hasilnya true atau false.

Operator : && (dan), || (atau), ! (not)

```
#include<iostream.h>
void main(void)
{
    int A=5, B=2, C=6, D=4;
    bool F;
    F=(A>B && C>D);
    cout<<" F = "<<F<<endl;
    F=(A>B && B>D);
    cout<<" F = "<<F<<endl;
    F=(A>B || C>D);
    cout<<" F = "<<F<<endl;
}
```

Hasilnya :

F = 1

F = 0

F = 1

5. Operasi Input/Output, Operator cin, cout, Standard Input : Keyboard, Standard Output : Screen

A. Alternatif : if, if – else, switch

```
if(kondisi)
{
    pernyataan;
}
```

pernyataan akan dilaksanakan jika nilai kondisi sama dengan 1 (TRUE)

```
if(kondisi)
{
    pernyataan 1;
}
else
{
    pernyataan 2;
}
```

pernyataan 1 akan dilaksanakan jika nilai kondisi sama dengan TRUE (1),
pernyataan 2 akan dilaksanakan jika nilai kondisi sama dengan FALSE (0)

```
switch(x)
{
    case 1 :
        pernyataan 1;
        break;
    case 2 :
        pernyataan 2;
        break;
    default :
        pernyataan 3;
}
```

Jika x bernilai 1 :
dikerjakan pernyataan 1
Jika x bernilai 2 :
dikerjakan pernyataan 2
Jika nilai x bukan 1 dan bukan 2 :
dikerjakan pernyataan 3

B. Pengulangan : do – while, while, for

```
x=0;
do
{
    cout<<x<<endl;
    x++;
}while(x<=3);
```

x awal = 0

cout<<x	x = x + 1	while(x<=3)
0	1	TRUE
1	2	TRUE
2	3	TRUE
3	4	FALSE

```
x=3;
while(x>=0)
{
    cout<<x<<endl;
    x--;
}
```

x awal = 3

while(x>=0)	cout<<x	x = x - 1
TRUE	3	2
TRUE	2	1
TRUE	1	0
TRUE	0	-1
FALSE		

```
for(x=0;x<=3;x++)
    cout<<x;
```

x awal = 0

x<=3	cout<<x	x = x + 1
TRUE	0	1
TRUE	1	2
TRUE	2	3
TRUE	3	4
FALSE		

Jumlah loop = $(3 - 0) + 1$

C. Percabangan

diperlukan label sebagai identitas cabang

```
goto label;  
pernyataan 1;  
pernyataan 3;  
label:  
{  
    pernyataan 4;  
}
```

LATIHAN

```
*Soal 1*/  
#include<iostream.h>  
void main(void)  
{  
    int i;  
    i=10;  
    if(i=1)  
        cout<<"Ini Pernyataan dalam if"<<endl;  
    else  
        cout<<"Ini Pernyataan dalam else"<<endl;  
}
```

```
/*Soal 2*/  
#include<iostream.h>  
void main(void)  
{  
    int A=10,B=20;  
    if(A && B)  
        cout<<"Ini Pernyataan dalam if"<<endl;  
    else  
        cout<<"Ini Pernyataan dalam else"<<endl;  
}
```

```
/*Soal 3*/
#include<iostream.h>
void main(void)
{
    int A=0,B=20;
    if(A && B)
        cout<<"Ini Pernyataan dalam if"<<endl;
    else
        cout<<"Ini Pernyataan dalam else"<<endl;
}
```

```
/*Soal 4*/
#include<iostream.h>
void main(void)
{
    int A=0,B=20;
    if(A || B)
        cout<<"Ini Pernyataan dalam if"<<endl;
    else
        cout<<"Ini Pernyataan dalam else"<<endl;
}
```

```
/*Soal 5*/
#include<iostream.h>
void main(void)
{
    int A=5,B=4;
    do
    {
        cout<<" B-- = "<<B--<<endl;
        cout<<" (B|A) = "<<(B|A)<<endl;
    }while((A|B)>=0);
}
```

```

/*Soal 6*/
#include<iostream.h>
void main(void)
{
    int A=5,B=4;
    do
    {
        cout<<" --B = "<<--B<<endl;
        cout<<" (B&A) = "<<(B&A)<<endl;
    }while((A&B)>0);
}

```

```

/*Soal 7*/
#include<iostream.h>
void main(void)
{
    int A=5,B=4;
    while((A&B))
    {
        cout<<" --B = "<<--B<<endl;
        cout<<" (B&A) = "<<(B&A)<<endl;
    }
}

```

```

/*Soal 9*/
#include<iostream.h>
void main(void)
{
    for(int i=0;i<=2;i++)
        for(int j=0;j<=3;j++)
            cout<<i+(++j)<<endl;
}

```

```

/*Soal 10*/
#include<iostream.h>
void main(void)
{
    for(int i=0;i<=2;i++)
        for(int j=0;j<=3;j++)
            cout<<j+(i++)<<endl;
}

```

```
/*Soal 8*/  
#include<iostream.h>  
void main(void)  
{  
    int A=5,B=4;  
    while((A&B)!=0)  
    {  
        cout<<" --B = "<<--B<<endl;  
        cout<<" (B&A) = "<<(B&A)<<endl;  
    }  
}
```

C. DAFTAR PUSTAKA

Buku

1. Esakov, Jeffrey, Tom Weiss, Data Structures An Advanced Approach Using C, Prentice-Hall, Inc. 1989
2. Hariyanto, Bambang, Struktur Data, Informatika Bandung, Pebruari 2000
3. Kadir, Abdul, Pemrograman Dasar Turbo C, Andi Offset, Yogyakarta, 1991
4. Kruse, Robert L. Data Structures & Program Design, Prentice-Hall, Inc. 1987
5. Standish, Thomas A. Data Structures, Algorithms & Software Principles In C, Addison Wesley, 1995