

PERTEMUAN 3: STRUKTUR / RECORD

A. TUJUAN PEMBELAJARAN

Pada bab ini akan dijelaskan mengenai struktur atau record yang terdapat pada struktur data. Di modul ini, Anda harus mampu:

3.1 Merepresentasikan struktur dan record dalam bahasa pemrograman .

B. URAIAN MATERI

Tujuan Pembelajaran 3.1:

Struktur / Record

Dalam database, data terdiri dari : field, record, dan file.

Field : satuan terkecil dari data

Record : kumpulan dari field

File : kumpulan dari record

Dalam bahasa C, record disebut structure

Mendefinisikan record :

a.

```
struct{  
    char NIM[10];  
    char Nama[20];  
    int Nilai;  
}snilai,enilai;
```

b.

```
typedef struct{  
    char NIM[10];  
    char Nama[20];  
    int Nilai;  
}typenilai;  
typenilai snilai,enilai;
```

c.

```
struct typenilai{  
    char NIM[10];  
    char Nama[20];  
    int Nilai;  
};  
struct typenilai snilai,enilai;
```

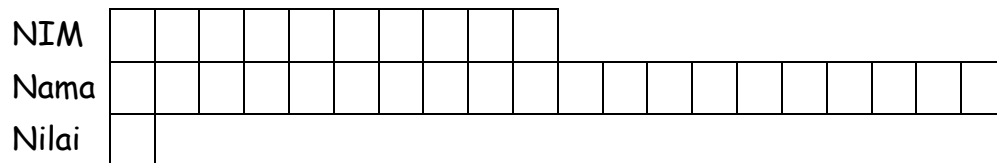
Yang ditulis dengan huruf tebal (struct, char, int) adalah kata kunci yang tidak boleh diganti

snilai, enilai : *structure variable*

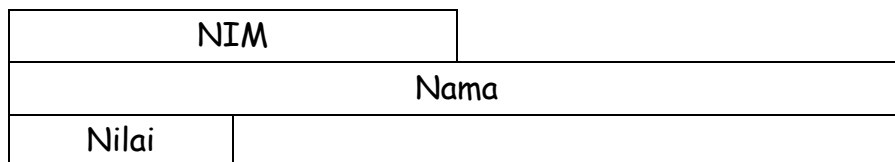
NIM, Nama, Nilai : *member*

typenilai : *structure tag*

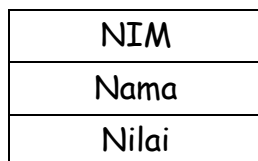
struktur dari snilai maupun enilai dapat diilustrasikan sbb:



atau



atau



Contoh 1.

```
#include<iostream.h>
typedef struct {
    char NIM[10];
    char nama[20];
    int nilai;
}mhs;
mhs mahasiswa;

void main(void)
{
    //Menginput dari kb
    cout<<"Masukan NIM : ";cin>>mahasiswa.NIM;
```

```
cout<<"Masukan Nama : ";cin>>mahasiswa.nama;
cout<<"Masukan Nilai : ";cin>>mahasiswa.nilai;
//Mencetak hasil inputan
cout<<"NIM : "<<mahasiswa.NIM<<endl;
cout<<"Nama : "<<mahasiswa.nama<<endl;
cout<<"Nilai : "<<mahasiswa.nilai<<endl;
}
```

Contoh 2.

```
#include<iostream.h>
void main(void)
{
    typedef struct{
        char FirstName[10];
        char LastName[10];
    }nametype;
    typedef struct{
        char street[30];
        char city[20];
        char state[15];
        char zip[10];
    }addrtype;
    typedef struct{
        nametype name;
        addrtype address;
    }Mhstype;
    Mhstype Mahasiswa;

    //Menginput dari kb
    cout<<"Masukan Nama Depan : "; cin.getline(Mahasiswa.name.FirstName,10);
    cout<<"Masukan Nama Keluarga : "; cin.getline(Mahasiswa.name.LastName,10);
    cout<<"Masukan Alamat : "; cin.getline(Mahasiswa.address.street,30);
    cout<<"Masukan Kota : "; cin.getline(Mahasiswa.address.city,20);
    cout<<"Masukan Propinsi : "; cin.getline(Mahasiswa.address.state,15);
    cout<<"Masukan Kode Pos : "; cin.getline(Mahasiswa.address.zip,10);

    //Mencetak hasil inputan
    cout<<endl;
    cout<<Mahasiswa.name.FirstName<<" ";
    cout<<Mahasiswa.name.LastName<<endl;
    cout<<Mahasiswa.address.street<<endl;
    cout<<Mahasiswa.address.city<<endl;
    cout<<Mahasiswa.address.state<<endl;
    cout<<Mahasiswa.address.zip<<endl;
}
```

Contoh 2 akan terbentuk struktur variabel yang dapat diilustrasikan sbb :

Mahasiswa

name	FirstName
	LastName
address	street
	city
	state
	zip

atau Mahasiswa

name.FirstName
name.LastName
address.street
address.city
address.state
address.zip

atau

Mahasiswa.name.FirstName
Mahasiswa.name.LastName
Mahasiswa.address.street
Mahasiswa.address.city
Mahasiswa.address.state
Mahasiswa.address.zip

Contoh 3

```
#include<iostream.h>
#include<stdlib.h>
#define LEBAR_FN 30 //Lebar String Nama Depan
#define LEBAR_LN 30 //Lebar String Nama Belakang
#define LEBAR_CITY 30 //Lebar String City
#define LEBAR_STR 30 //Lebar String Street
#define LEBAR_STATE 15 //Lebar String State
#define LEBAR_ZIP 10 //Lebar String Zip
#define LEBAR_INCOME 10 //Lebar String Income
void main(void)
{
    typedef struct{
        char FirstName[LEBAR_FN];
        char LastName[LEBAR_LN];
    }strNama;
    typedef struct{
        char street[LEBAR_STR];
        char city[LEBAR_CITY];
        char state[LEBAR_STATE];
        char zip[LEBAR_ZIP];
    }strAlamat;
    typedef struct{
        strNama NamaKeluarga;
        strAlamat AlamatKeluarga;
        int Income;
    }strKeluarga;
```

```

strKeluarga Keluarga[2];
char tmpIncome[LEBAR_INCOME];
/*****Menginput dari kb*****/
for(int i=0;i<=1;i++)
{
    cout<<endl<<"Memasukan Data Keluarga ke-"<<i+1<<endl;
    cout<<"Masukan Nama Depan : ";
    cin.getline(Keluarga[i].NamaKeluarga.FirstName, LEBAR_FN);
    cout<<"Masukan Nama Belakang : ";
    cin.getline(Keluarga[i].NamaKeluarga.LastName, LEBAR_LN);
    cout<<"Masukan Alamat : ";
    cin.getline(Keluarga[i].AlamatKeluarga.street, LEBAR_STR);
    cout<<"Masukan Kota : ";
    cin.getline(Keluarga[i].AlamatKeluarga.city, LEBAR_CITY);
    cout<<"Masukan Propinsi : ";
    cin.getline(Keluarga[i].AlamatKeluarga.state, LEBAR_STATE);
    cout<<"Masukan Kode Pos : ";
    cin.getline(Keluarga[i].AlamatKeluarga.zip, LEBAR_ZIP);
    cout<<"Masukan Penghasilan : ";
    cin.getline(tmpIncome, LEBAR_INCOME);
    Keluarga[i].Income=atoi(tmpIncome);
    cout<<endl;
}

/*****Mencetak hasil inputan*****/
cout<<endl;
for(i=0;i<=1;i++)
{
    cout<<"Keluarga ke-"<<i+1<<endl;
    cout<<Keluarga[i].NamaKeluarga.FirstName<<" ";
    cout<<Keluarga[i].NamaKeluarga.LastName<<endl;
    cout<<Keluarga[i].AlamatKeluarga.street<<endl;
    cout<<Keluarga[i].AlamatKeluarga.city<<endl;
    cout<<Keluarga[i].AlamatKeluarga.state<<endl;
    cout<<Keluarga[i].AlamatKeluarga.zip<<endl;
    cout<<"Penghasilan = "<<Keluarga[i].Income<<endl;
}
}

```

Contoh 3 akan terbentuk struktur variabel yang dapat diilustrasikan sbb :

Keluarga[0]

NamaKeluarga	FirstName
	LastName
AlamatKeluarga	Street
	City
	State
	ZIP
Income	

Keluarga[1]

NamaKeluarga	FirstName
	LastName
AlamatKeluarga	Street
	City
	State
	ZIP
Income	

VARIABEL DINAMIS

Variabel dinamis : dapat disiapkan dan dihapus saat program sedang dijalankan.

Perhatikan program berikut :

```
#include<iostream.h>
#include<stdlib.h>
void main(void)
{
    int *P;
    P=(int *)malloc(sizeof(int));
    *P=25;
    cout<<*P<<endl;
}
```

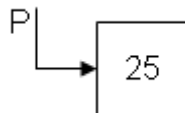
Keterangan :

int *P : menyiapkan sebuah pointer P bertipe int

P = (int*)malloc(sizeof(int)) : menyiapkan suatu area bebas, tanpa nama, selebar sizeof(int), alamatnya dicatat dalam pointer P.

Untuk membebaskan area tersebut dapat digunakan instruksi : free(P)

Program diatas dapat
diilustrasikan sbb :

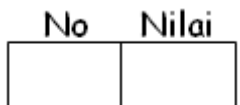


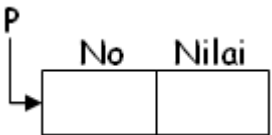
Variabel struktur yang bersifat dinamis :

```
#include<iostream.h>
#include<stdlib.h>
void main(void)
{
    typedef struct{
        int No;
        int Nilai;
    }strNode;
    strNode *P;
    P=(strNode*)malloc(sizeof(strNode));
    P->No=25;
```

```
P->Nilai=87;
cout<<"No = "<<P->No<<endl;
cout<<"Nilai = "<<P->Nilai<<endl;
}
```

Program diatas dapat diilustrasikan sbb :

<pre>typedef struct{ int No; int Nilai; }strNode;</pre>	 <p>A diagram showing a horizontal rectangle divided into two equal sections. The left section is labeled 'No' and the right section is labeled 'Nilai'.</p>
---	--

<pre>P=(strNode*)malloc(sizeof(strNode))</pre>	 <p>A diagram showing a pointer 'P' with an arrow pointing to a horizontal rectangle divided into two sections labeled 'No' and 'Nilai'.</p>
--	--

<pre>P->No=25; P->Nilai=87;</pre>	 <p>A diagram showing a pointer 'P' with an arrow pointing to a horizontal rectangle divided into two sections. The left section is labeled 'No' and contains the value '25'. The right section is labeled 'Nilai' and contains the value '87'.</p>
---	--

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