PERTEMUAN 4
STRUKTUR

A. TUJUAN PEMBELAJARAN

Setelah menyelesaikan pertemuan ini, mahasiswa mampu mempraktekkan:

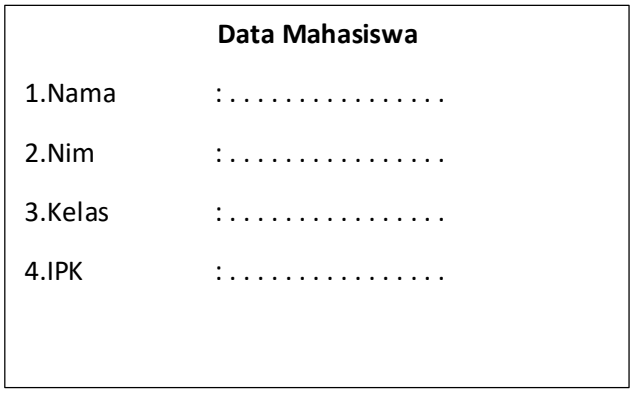
1. Pengertian Struktur
2. Deklarasi Struktur
3. Struktur di dalam struktur
4. Menggunakan Struktur dalam Array
5. Struktur bit-field

B. URAIAN MATERI

1. Pengertian Struktur

Struktur merupakan Kumpulan dari variabel - variabel atau elemen data yang digabungkan menjadi satu kesatuan data, Ataupun pengeleompokan dari sejumlah data atau variabel-variabel sejumlah data dengan tipe yang berbeda yang berkumpul dalam satu nama yang sama. Struktur biasanya difungsikan untuk pengelompokan beberapa informasi yang berkaitan dengan sebuah record.Maka struktur berguna untuk menggabungkan sejumlah datadengan tipe yang berbeda.

Sedangkan array berguna untuk menyimpan data dengan tipe yang sama misalnya int atau char. Tetapi dalam kehidupan nyata kita akan menghadapikebutuhan untuk menyimpan berbagai jenis data dalam sebuah array, Seperti pada datamahassiswa di universitas berisi informasi dariberbagai jenis data. Perhatikan contoh berikut :



Gambar 4.7 contoh data mahasiswa

2. Deklarasi Struktur

Sebagai contoh untuk membuat strukturpada gambar 4.1 data mahasiswa adalah sebagai berikut :

```
struct   mahasiswa
 {
char  nama [20]; //terdapat spasi 20 karakter
 int nim;
char kelas;
floatipk;
 } ;
typedefstruct  mahasiswa  mhs;
```

Struktur data adalah sekelompok elemen data yang dikelompokkan bersama di bawah nama. Item data ini, juga dikenal sebagai anggota, dapat memiliki jenis dan panjang yang berbeda. Struktur data dapat dideklarasikan dalam C ++ menggunakan sintaks berikut:

structtype_nama {

member_type1 member_name1;

member_type2 member_name2;

member_type3 member_name3;

.

} nama objek;

Di mana type_name adalah nama untuk tipe struktur, nama-objek bisa menjadi satu set pengenal yang valid untuk objek dari tipe struktur ini. Terlampir dalam kurung kurawal {} adalah daftar anggota data, masing-masing ditentukan dengan jenis dan pengenal yang valid sebagai nama.

Sebagai contoh:

```
struct produk {
  int berat;
  double harga;
} ;

produkaple;
produkpisang, melon;
```

Ini menjelaskan jenis struktur, yang disebut produk, dan mendefinisikannya dengan dua anggota: berat dan harga, masing-masing dengan tipe fundamental yang berbeda. Deklarasi ini membuat jenis baru (produk), yang kemudian digunakan untuk mendeklarasikan tiga objek (variabel) jenis ini: apel, pisang, dan melon. Harap dicatat bahwa setelah produk dinyatakan, itu digunakan seperti jenis lainnya.

Tepat di akhir definisi struktur, dan sebelum titik koma terakhir (;), kolom object_names opsional dapat digunakan untuk langsung mendeklarasikan objek dengan tipe struktur. Misalnya, objek struktur apel, pisang, dan melon dapat dideklarasikan saat menentukan tipe struktur data:

```
struct product {
  int berat;
  double harga;
} apel, pisang, melon;
```

Dalam kasus ini, di mana nama objek ditentukan, nama tipe (produk) menjadi opsional: struktur memerlukan nama tipe atau setidaknya satu nama dalam nama objek, tetapi tidak harus keduanya.

Penting untuk membuat perbedaan yang jelas antara apa nama jenis struktur (produk) dan apa yang menjadi objek dari jenis ini (apel, pisang dan melon). Banyak objek (seperti apel, pisang, dan melon) dapat dinyatakan dari suatu tipe struktur (produk).

Setelah ketiga objek dengan jenis tekstur tertentu (apel, pisang, dan melon) telah dinyatakan, anggota dapat langsung diakses. Sintaksnya cukup dengan menyisipkan titik (.) Antara nama objek dan nama anggota. Misalnya, kita dapat bekerja dengan salah satu elemen ini seolah-olah itu adalah variabel standar dari tipenya masing-masing:

```
apple.berat
apple.harga
banana.berat
banana.harga
melon.berat
melon.harga
```

Masing-masing memiliki tipe data yang sesuai dengan anggota yang mereka rujuk: apple.berat, banana.berat dan melon.hargabertipe int, sedangkan apple.harga, banana.harga dan melon.harga adalah tipe double .

Berikut adalah contoh nyata dengan tipe struktur beraksi:

```
// example about structures
#include <iostream>
#include <string>
#include <sstream>
using namespace std;

struct movies_t {
  string title;
  int year;
} mine, yours;

void printmovie (movies_t movie);

int main ()
{
  string mystr;

  mine.title = "2001 A Space Odyssey";
  mine.year = 1968;

  cout << "Enter title: ";
  getline (cin,yours.title);
  cout << "Enter year: ";
  getline (cin,mystr);
  stringstream(mystr) >> yours.year;
```

```
  cout << "My favorite movie is:\n ";
  printmovie (mine);
  cout << "And yours is:\n ";
  printmovie (yours);
  return 0;
}

void printmovie (movies_t movie)
{
  cout << movie.title;
  cout << " (" << movie.year << ")\n";
}
```

Contoh tersebut menunjukkan bagaimana anggota suatu objek berperilaku seperti variabel biasa. Misalnya, anggota yours.year adalah variabel valid tipe int, dan mine.title adalah variabel valid tipe string.

Tapi objek milikku dan milikmu juga variabel dengan tipe (tipe film_t). Misalnya, keduanya diteruskan ke fungsi printmovie seolah-olah keduanya adalah variabel sederhana. Oleh karena itu, salah satu ciri struktur data adalah kemampuannya untuk mengacu pada dua anggotanya secara terpisah atau pada keseluruhan struktur secara keseluruhan. Dalam kedua kasus, gunakan pengenal yang sama: nama struktur.

3. Struktur dalam struktur

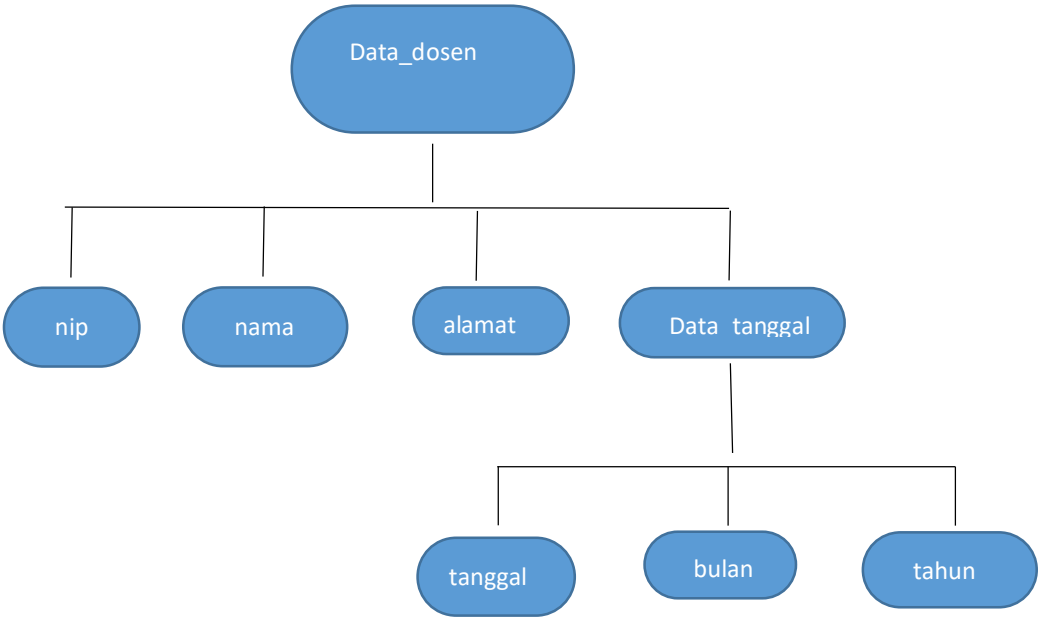
Suatu struktur dapat mengandung struktur lainya untuk dijadikan field di dalamnya. Sebagai gambaran struturdata_dosen mempunyai sebuah field struktur tanggal lahir yang berisi fieldtanggal,bulan,dan tahun. Bentuk umum

pendeklarasiannya adalah sebagai berikut :

```
4   typedef struct // deklarasi struct pertama
5   {
6       int tanggal;
7       int bulan;
8       int tahun;
9   }data_tanggal;
10  typedef struct // deklarasi struct dalam struct
11  {
12      long int nip;
13      char nama[30];
14      char alamat[50];
15      data_tanggal tanggal_lahir;
16  }data_dosen;
```

Gambar 4.8 Deklarasi structur dalam struktur

Pada contoh ini, terdapat pendeklarasian struktur bernama data_tanggal sekaligus pendefinisian variabel struktur bernama data_dosengambar di bawah ini menunjukan anggota dari variable data_dosen.



Untuk lebih jelasnya Sebagai contoh untuk kasus data_dosen adalah sebagai berikut :

```
#include <cstdlib>
#include <iostream>
usingnamespacestd;
typedefstruct // deklarasi struct pertama
{
 int tanggal;
 int bulan;
 int tahun;
}data_tanggal;
typedefstruct // deklarasi struct dalam struct
{
 long int nip;
 char nama[30];
 char alamat[50];
 data_tanggaltanggal_lahir;
}data_dosen;
data_dosen dosen;
int main(int argc,char*argv[])
{                    // input
 cout<<"NIP              : "; cin>>dosen.nip;
 cout<<"Nama  :";fflush(stdin); cin.get(dosen.nama,30);
 cout<<"Alamat           :";fflush(stdin); cin.get(dosen.alamat,50);
 cout<<"Tanggal lahir  : "; cin>>dosen.tanggal_lahir.tanggal;
```

```
cout<<"Bulan lahir      : "; cin>>dosen.tanggal_lahir.bulan;
cout<<"tahun lahir      : "; cin>>dosen.tanggal_lahir.tahun;
cout<<endl<<endl;
cout<<"NIP              : "<<dosen.nip<<endl;
cout<<"Nama             : "<<dosen.nama<<endl;
cout<<"Alamat: "<<dosen.alamat<<endl;
cout<<"Tanggal Lahir:"<<dosen.tanggal_lahir.tanggal<<endl;
cout<<"Bulan Lahir    : "<<dosen.tanggal_lahir.bulan<<endl;
cout<<"Tahun Lahir    : "<<dosen.tanggal_lahir.tahun<<endl;
cout<<endl<<endl;
return EXIT_SUCCESS;
}
```

4. Struktur dalam array

Pada Subpertemuan sebelumnya telah dipelajari bagaimana cara untuk mendeklarasikan sebuah variable dengan tipe data struktur yang dibuat. Tapi dari cara pendeklarasian tersebut timbul suatu permasalahan yaitu bagaimana kalau deklarasi variable struktur nya sangat banyak? Permaslahan tersebut dapat dipecahkan dengan cara mendeklarasikan variabel strucknya menggunakan variabel bertipearray.

Contoh kasusnya adalah sebuah struktur yang menampung 100 data dosen. Untuk memecahkannya buatlah struktur data_dosen yang nantinya akan digunakan untuk mendeklarasikan variable dosen yang bertipe array. Untuk lebih jelasnya buatlah program berikut ini :

```
#include <cstdlib>
#include <iostream>
using namespace std;
typedef struct
{
    long int nip;
    char nama[30];
    char alamat[50];
}data_dosen;
// Array of struct
data_dosen dosen[10];
int main(int argc,char*argv[])
{
int i,j;
cout<<"berapa data dosen ?";
cin>>j;
cout<<endl;
for(i=0;i<j;i++)
{   cout<<"Data dosen ke -" <<i+1<<endl;
    cout<<"====================================="<<end
l;
    cout<<"NIP          : "; cin>>dosen[i].nip;
    cout<<"Nama         : ";fflush(stdin); cin.get(dosen[i].nama,30);
    cout<<"Alamat       :";fflush(stdin); cin.get(dosen[i].alamat,50);
    cout<<endl<<endl;
}
```

```
    return EXIT_SUCCESS;
    }
```

5. Struktur Bit-Field

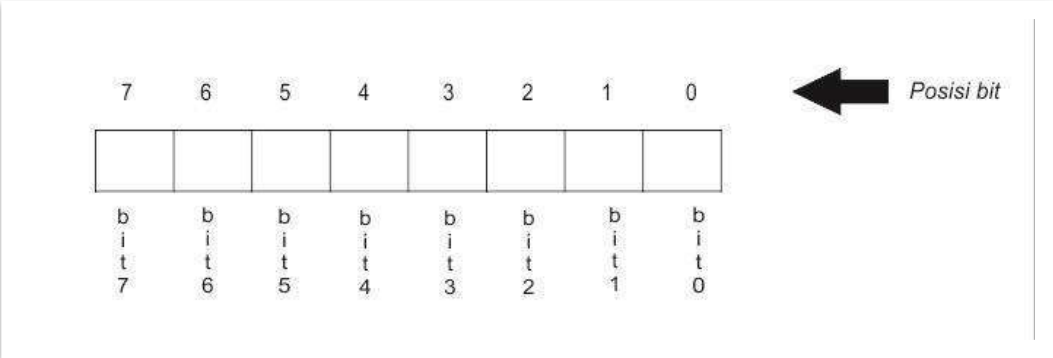
Suatu bit atau beberapa bit dalam sebuah data berukuran satu byte atau dua byte dapat diakses dengan mudah melalui bit-field. Dengan cara ini, suatu bit atau beberapa dapat diakses tanpa melibatkan operator permanipulasi bit (seperti & dan I). selain itu, satu atau dua byte dapat dipakai untuk menyimpan sejumlah informasi.

Contoh berikut memberikan gambaran bit-bit dalam sebuah byte, berdasarkan struktur bit-field.

```
typedef struct
{
    unsigned bit0: 1;
    unsigned bit1: 1;
    unsigned bit2: 1;
    unsigned bit3: 1;
    unsigned bit4: 1;
    unsigned bit5: 1;
    unsigned bit6: 1;
    unsigned bit7: 1;

}info_bit;
```

Pada deklarasi seperti diatas, titik dua (:) menyatakan panjangnya bit-field. Apabila disajikan dalam gambar, struktur bit-fielddiatas dapat digambarkan seperti berikut



Untuk lebih jelasnya lihatlah contoh program berikut ini :

```
#include<iostream>
#include<cstdlib>

usingnamespacestd;

int main( int argc,char*argv[])
{
typedefstruct
{
        unsigned bit0: 1;
        unsigned bit1: 1;
        unsigned bit2: 1;
        unsigned bit3: 1;
        unsigned bit4: 1;
        unsigned bit5: 1;
        unsigned bit6: 1;
```

```
        unsigned bit7: 1;

}info_bit;

union ubyte
{
        unsignedcharbyte;
        info_bit bit;
};
ubyteascii;
int nilai;
cout<<"Masukan Nilai ASCII antara 0 sampai dengan 255 = ";
cin>>nilai;

ascii.byte=nilai;
cout<<"76543210 <---------- posisi bit"<<endl;
cout<<ascii.bit.bit7
        <<ascii.bit.bit6
        <<ascii.bit.bit5
        <<ascii.bit.bit4
        <<ascii.bit.bit3
        <<ascii.bit.bit2
        <<ascii.bit.bit1
        <<ascii.bit.bit0
        <<endl;
```

```
return 0;
}
```

Perlu diketahui , suatu variabel yang dideklarasikan sebagai bit-field tidak bisa diisi secara langsung dengan suatu nilai. Pada program diatas, struktur info_bit berbagi memori dengan variabel bertipeunsignedchar.

C. SOAL LATIHAN/TUGAS

Latihan	Petunjuk Pengerjaan Tugas
Latihan 4	1. Jelaskan menurut anda apa itu struktur ! 2. Jelaskan perbedaan array dengan struktur ! 3. Jelaskan menurut andamaanfaat menggunakan struktur ! 4. Buatlah program struktur dalam struktur yang terdiri dari data mahasiswa ! 5. Buatlah Program menghitung luas dan keliling segitiga dengan menggunakan struktur !.

D. REFERENSI

Bachtiar,Adam Mukharil. (2018).*Pemrograman C dan C++.* Bandung

Kadir, Abdul.(2013).*From zero to a pro pemrograman c++.Yogyakarta*

Moh.Sjukani . 2014. ALGORITMA (Algoritma & Struktur Data 1) dengan *C, C++ dan Java, Edisi 9, Mitra Wacana Media.*