Algoritma dan Pemrograman Lanjut

Pertemuan Ke-3
Record/Struct dan Array Of Record

Disusun Oleh :
Wilis Kaswidjanti, S.Si.,M.Kom.

Jurusan Teknik Informatika
Fakultas Teknologi Industri
Universitas Pembangunan Nasional "Veteran"
Yogyakarta

Algoritma dan Pemrograman Lanjut

Judul Materi : Record/Struct dan Array Of Record

Deskripsi Materi : Materi ini membahas tipe data terstruktur record/struct dan penggunaannya dengan tipe data terstruktur array

Tujuan Instruksional Khusus :

1. Mendefinisikan dan menggunakan tipe data record/struktur
2. Mendeskripsikan tipe data record

Referensi :

- Buku Teks

  Munir, Rinaldi (2005), *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*, Buku 2, Edisi Ketiga, Penerbit Informatika Bandung.

  Charibaldi, N. (2004), *Modul Kuliah Algoritma Pemrograman II*, Edisi Kedua, Yogyakarta

- Buku Acuan/Referensi

  Brassard, Gilles (1999), *Fundamentals of algorithma*, PrinteceHall.

  Jarne, Stroustrup B. (1997), *C++ Programming language*, AT &T.

  Kristanto, Andri (2003), *Algoritma pemrograman C++*, Graha Ilmu.

  Schildt,Herbert (2000), *The Complete Reference C++*, McGraw-Hill.

  Sedgewick, R. (2000), *Algoritma Third edition In C part 5*, Addison Wesley.

RECORD/STRUCT DAN ARRAY OF RECORD

PENDAHULUAN

Seperti halnya Array, Record/Struct mempunyai sejumlah elemen yang disebut field. Kalau semua elemen array harus mempunyai tipe data yang sama, maka tiap-tiap elemen pada Record/Struct dapat memiliki tipe data yang berbeda.

ISI

A. RECORD/STRUCT

Bentuk umum deklarasi Record/Struct

Algoritma :

```
namaVar : record
            < namaField1 : tipeField1 ,
              namaField2 : tipeField2 ,
              … …
              namaFieldn : tipeFieldn ,
            >
```

Atau namavar dapat dipisah dari deklarasi tiperecordnya, sehingga menjadi

```
Type namaTipeRecord : record
          <  namaField1 : tipeField1 ,
             namaField2 : tipeField2 ,
              … …
             namaFieldn : tipeFieldn ,
          >
  namaVar :  namaTipeRecord
```

Bahasa C++ :

```
struct namaTipeStruct
       {
         tipeField1  namaField1;
         tipeField2  namaField2;
         … …
         tipeFieldn  namaFieldn;
        } namaVar;
```

Atau namavar dapat dipisah dari deklarasi tiperecordnya, sehingga menjadi :

```
typedef  struct
        {
         tipefield1 namafield1;
         tipefield2 namafield2;
         … …
         tipefieldn namafieldn;
       } namatipestruct;
 namatipestruct  namavar;
```

Catatan : namaVar bisa lebih dari satu

Contoh record/struct :

Algoritma :

Deklarasi

Type Mahasiswa : Record < NIM : integer,
Nama : string,
KodeMK : string,
NilaiHuruf : char >

Mhs1,Mhs2 : Mahasiswa

Bahasa C++ :

```
//deklarasi
  typedef  struct  {  int NIM;
```

```
                        char Nama[20];
                        char KodeMK[10];
                        char NilaiHuruf;
                    } Mahasiswa;
    Mahasiswa  Mhs1,Mhs2;
```

Cara mengakses elemen record/structur :

namavar.namafield

Contoh Program :

```
#include<iostream.h>
#include<string.h>
typedef struct
       {  int tanggal, bulan, tahun;
       } data_tanggal;
typedef struct
           {  char nama[30];
              data_tanggal tgl_lahir;
           } data_rekan;
data_rekan info_rekan;
main()
{
   strcpy(info_rekan.nama,"Hanif");
   info_rekan.tgl_lahir.tanggal = 30;
   info_rekan.tgl_lahir.bulan = 4;
   info_rekan.tgl_lahir.tahun = 2002;
   cout<<"Nama  : "<<info_rekan.nama;
   cout<<"\nTanggal lahir :";
   cout<<info_rekan.tgl_lahir.tanggal;
   cout<<"-"<<info_rekan.tgl_lahir.bulan;
   cout<<"-"<<info_rekan.tgl_lahir.tahun;
}
```

Contoh Variasi program Cara Mendeklarasikan Structur :

```
#include<iostream.h>
#include<string.h>
main()
{
   struct data_tanggal
    {  int tanggal, bulan, tahun; };
   struct data_rekan
    {  char nama[30];
       struct data_tanggal  tgl_lahir;
    };
   struct data_rekan info_rekan;
   strcpy(info_rekan.nama,"Hanif");
   info_rekan.tgl_lahir.tanggal = 30;
   info_rekan.tgl_lahir.bulan = 4;
   info_rekan.tgl_lahir.tahun = 2002;
   cout<<"Nama   : "<<info_rekan.nama;
   cout<<"\nTanggal lahir :";
   cout<<info_rekan.tgl_lahir.tanggal;
   cout<<"-"<<info_rekan.tgl_lahir.bulan;
   cout<<"-"<<info_rekan.tgl_lahir.tahun;
}
```

B. ARRAY OF RECORD

Algoritma :

```
namaVar : record
            < namaField1 : tipeField1 ,
              namaField2 : tipeField2 ,
              …….
              namaFieldn : tipeFieldn ,
             >
namaVar : array[rangeindex] of  namaTipeRecord
```

Bahasa C++ :

```
struct namaTipeStruct
         {
           tipeField1  namaField1;
           tipeField2  namaField2;
            … …
           tipeFieldn  namaFieldn;
          } namaVar;
namaTipeStruct  namaVar[ukuran];
```

PENUTUP

Record merupakan suatu tipe data terstruktur yang dapat menampung data field bertipe berbeda. Tipe ini dapat dipadukan dengan tipe data terstruktur lainnya seperti array.

SOAL-SOAL

Buat program untuk menginput dan menampilkan 5 data mahasiswa matakuliah Algoritma dan Pemrograman 2 dengan field-field NoMhs, Nama, Kelas, NilaiAngka dan NilaiHuruf, dengan ketentuan NilaiHuruf tidak diinputkan tetapi berasal dari NilaiAngka. Range NilaiHuruf : $0 \leq E < 20$; $20 \leq D < 40$; $40 \leq C < 60$; $60 \leq B < 75$; $75 \leq A \leq 100$