MODUL MATA KULIAH PRAKTIKUM PEMROGRAMAN LANJUT

Modul 3: Tipe Data Bentukan

Dosen Pengampu:
Herryawan Pujiharsono, S.T., M.Eng.
Slamet Indriyanto, S.T., M.T.
Prasetyo Yuliantoro, S.T., M.T.

PROGRAM STUDI S1 TEKNIK TELEKOMUNIKASI
FAKULTAS TEKNIK TELEKOMUNIKASI DAN ELEKTRO
INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2019

I. TUJUAN

1. Mengetahui arti dan fungsi tipe data bentukan
2. Mendefinisikan tipe data baru dari penamaan ulang tipe data dasar
3. Mendefinisikan tipe data bentukan dari tipe data terstruktur (*struct*)
4. Mengkombinasikan tipe data bentukan *struct* dengan larik (*array*)
5. Mengkombinasikan tipe data bentukan *struct* dengan fungsi dan prosedur

II. DASAR TEORI

Dalam membuat program, kadangkala akan dihadapkan dengan struktur data yang tidak sederhana dan apabila hanya ditangani dengan data dasar saja, maka pemrogram akan kesulitan merumuskan komposisinya. Sebagai contoh, program yang akan dibuat melibatkan data tentang mahasiswa, maka untuk variabel mahasiswa akan sulit ditentukan tipe datanya karena pada mahasiswa terdapat beberapa elemen, yaitu nama, nomor induk mahasiswa, jenis kelamin, alamat, dan elemen-elemen lainnya. Tantangan berikutnya adalah bagaimana cara menyimpan data-data mahasiswa tersebut jika jumlah mahasiswa lebih dari satu? Tentunya hal ini akan sangat sulit jika harus diselesaikan dengan tipe data dasar saja. Oleh karena itu, diperlukan adanya suatu tipe data baru yang digunakan untuk menangani kasus di atas, yaitu dengan menggunakan tipe data bentukan.

Tipe data bentukan merupakan suatu tipe data yang dirancang/dibentuk (dan diberi nama) dari beberapa elemen bertipe tertentu yang sudah dikenal. Jadi, di dalam tipe data bentukan akan terdapat elemen dengan tipe data dasar dan dapat juga terdapat tipe data bentukan lain yang telah didefinisikan sebelumnya. Tujuan digunakannya tipe data bentukan adalah supaya perancangan program mendapatkan suatu tipe data dimana seluruh komponennya secara keseluruhan memiliki makna semantik dan di dalamnya terdapat keterkaitan antar komponen. Pada data mahasiswa telah dijabarkan beberapa elemen yang ada, maka dengan menggunakan tipe data bentukan, perancang program dapat mendefinisikan ke dalam program.

Tipe bentukan dapat dibedakan menjadi dua jenis, yaitu tipe data dasar yang diberi nama tipe baru (penamaan ulang tipe data dasar) dan tipe data terstruktur.

A. PENAMAAN ULANG TIPE DATA DASAR

Terkadang pemrogram ingin memberi nama baru terhadap tipe dasar yang sudah dikenal. Tujuannya supaya nama baru tersebut lebih mudah diinterpretasi oleh

yang membaca teks algoritma. Struktur penamaan ulang dalam bentuk pseudocode adalah sebagai berikut.

```
TYPE <Nama_Baru> : <Tipe Data Dasar>
```

Tipe data dasar terdiri dari *integer*, *real*, *char*, dan *string* (Lihat Modul 1). Struktur penamaan ulang tersebut jika ditranslasikan ke dalam struktur Bahasa C++ menjadi sebagai berikut.

```
typedef <Tipe Data Dasar> <Nama_Baru>;
```

Contoh penamaan ulang tipe data dasar *integer* menjadi tipe bernama Bulat dalam bentuk pseudocode dan hasil translasi pada struktur Bahasa C++:

```
TYPE Bulat : integer
```

```
typedef int Bulat;
```

Tipe yang sudah diberi penamaan ulang tersebut selanjutnya dapat digunakan untuk menggantikan nama dari tipe data dasar untuk pendeklarasian variabel. Contoh penggunaan tipe data *integer* dengan nama baru Bulat untuk pendeklarasian variabel dalam bentuk pseudocode dan hasil translasinya pada C++ adalah sebagai berikut.

```
Bil1 : Bulat
```

```
Bulat Bil1;
```

B. PENDEFINISIAN TIPE DATA TERSTRUKTUR

Tipe data terstruktur adalah tipe data yang berupa rekaman (*record*). Rekaman tersebut disusun oleh satu atau lebih *field* dimana tiap *field* adalah variabel untuk menyimpan data dengan tipe dasar tertentu atau tipe data bentukan lain yang sudah didefinisikan sebelumnya.

Field 1	*Field 2*	*Field 3*	...	*Field N*

Dalam bentuk pseudocode, sebuah tipe data terstruktur (*record*) dapat disusun sebagai berikut.

```
TYPE <Nama_Tipe> : RECORD
<
      <Field 1> : <Tipe_Data 1>,
      <Field 2> : <Tipe_Data 2>,
      .....
      <Field N> : <Tipe_Data N>
>
```

Berdasarkan struktur tersebut, struktur *record* hampir seperti larik. Keduanya memiliki persamaan, yaitu sama-sama memiliki elemen. Namun, perbedaannya adalah jika pada larik, elemennya harus memiliki tipe data yang sama dan elemennya diakses tau diidentifikasi menggunakan indeks, sedangkan *record* memiliki elemen dengan tipe

yang dapat berbeda-beda dan elemennya diakses atau diidentifikasi menggunakan *identifier* atau nama variabel. Jika ditranslasikan ke dalam struktur Bahasa C++, tipe data bentukan *record* memiliki struktur sebagai berikut.

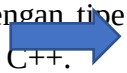
```
typedef struct <Nama_Tipe>
{
    <Tipe_Data 1> <Field 1>;
    <Tipe_Data 2> <Field 2>;
    .....
    <Tipe_Data N> <Field N>;
}
```

Contoh pembuatan tipe data bentukan *record* pada pseudocode dan hasil translasi ke struktur Bahasa C++ dengan nama *record* adalah *Titik* yang mempunyai elemen *x* dan *y* bertipe integer:

```
TYPE Titik : RECORD
<
    x : integer,
    y : integer
>
```

```
typedef struct
{
    Int x;
    Int y;
} Titik;
```

Sama seperti pada penamaan ulang tipe data dasar, setelah tipe bentukan dibuat, selanjutnya tipe data tersebut dapat digunakan untuk mendeklarasikan suatu variabel. Contoh pendeklarasian variabel *P* dengan tipe data bentukan *record* bernama *Titik* dalam bentuk pseudocode dan Bahasa C++.



```
P : Titik
```

```
Titik P;
```

Variabel yang sudah dideklarasikan dengan tipe bentukan *record* tersebut selanjutnya dapat secara otomatis menggunakan elemen-elemen yang ada di dalam tipe bentukan tersebut. Pengaksesan elemen-elemen tersebut pada pseudocode dan Bahasa C++ dilakukan dengan cara yang sama, yaitu sebagai berikut.

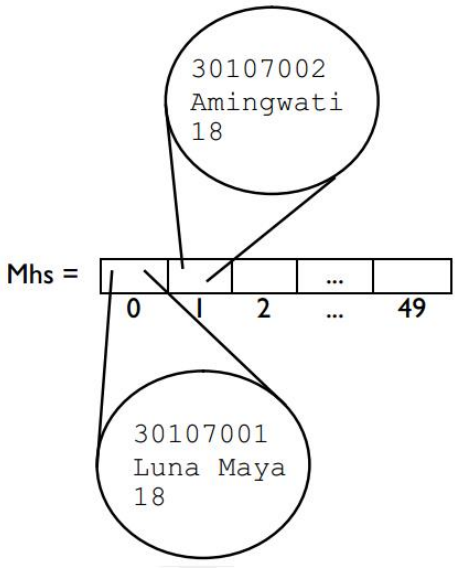
```
<Variabel_Record>.<elemen 1>
<Variabel_Record>.<elemen 2>
.....
<Variabel_Record>.<elemen N>
```

Pengakesan elemen dilakukan satu per satu dan masing-masing elemen dapat dioperasi layaknya variabel biasa. Contoh pengaksesan elemen *x* dan *y* pada tipe bentukan *record* bernama *Titik* melalui variabel *P* adalah sebagai berikut.

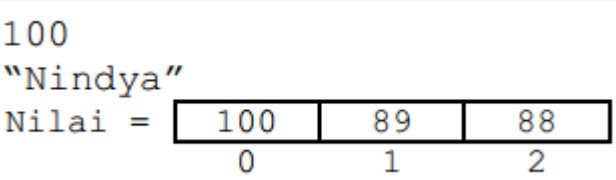
P.x *P.y*

C. KOMBINASI TIPE DATA BENTUKAN DAN LARIK

Sebuah larik dapat dideklarasikan dengan menggunakan tipe data bentukan. Hal ini berarti setiap indeks pada larik tersebut akan berisi elemen-elemen tipe data bentukan tersebut seperti ditunjukkan pada gambar berikut.



Contoh di atas menunjukkan bahwa di dalam sebuah elemen larik dapat diisi dengan suatu nilai yang memiliki tipe data bentukan. Selain itu, kondisi yang sebaliknya, yaitu memasukkan variabel bertipe larik menjadi salah satu atau beberapa elemen di dalam variabel yang memiliki tipe data bentukan, juga dapat dilakukan, seperti ditunjukkan pada gambar berikut (Lihat Latihan 4).



Jika kedua contoh tersebut digabungkan maka akan membentuk larik dari tipe data bentukan yang mengandung larik.

D. TIPE DATA BENTUKAN DALAM SUBPROGRAM (FUNGSI DAN PROSEDUR)

Selain dapat dioperasikan dan dikombinasikan dengan larik, tipe data bentukan juga dapat digunakan pada subprogram dengan struktur fungsi atau prosedur, baik sebagai tipe data untuk *return value* maupun sebagai parameter. Pengaksesan variabel

yang menggunakan tipe data bentukan tersebut pada subprogram pun hampir sama seperti pengaksesan variabel bertipe data bentukan yang dijelaskan sebelumnya.

III. PERCOBAAN

1. Latihan 1- Penamaan ulang tipe data dasar

```
#include <iostream>

using namespace std;

typedef int Bilangan;

int main()
{
    Bilangan B;
    cout << "Masukkan satu bilangan:";
    cin >> B;
    cout << endl << "Bilangan yang ada masukkan adalah " << B << endl;
    return 0;
}
```

- Apa saja <*Tipe Data Dasar*> yang diberi nama baru pada kode program di atas? Apa <*Nama_Baru*> untuk tipe data dasar tersebut?
- Ganti <*Tipe Data Dasar*>-nya menjadi *real* sehingga variabel yang diinputkan bisa berupa pecahan!
- Tambahkan tipe data *Huruf* untuk mendeklarasikan variabel yang digunakan untuk menampung kalimat yang diinputkan oleh user dan kemudian dioutputkan pada layar monitor!

2. Latihan 2 - Pendefinisian tipe data terstruktur

```
#include <iostream>

using namespace std;

typedef struct
{
    int x;
    int y;
} Titik;

int main()
{
    Titik P;
    cout << "Masukkan satu koordinat:" << endl << "x1: ";
    cin >> P.x;
    cout << "y1: ";
    cin >> P.y;
    cout << endl << "Koordinat yang anda masukkan:(" << P.x << "," << P.y << ")" << endl;
    return 0;
}
```

- Apa <*Nama_Tipe*> untuk tipe data bentukan *record* yang dibentuk pada kode program tersebut?
- Apa saja elemen yang ada di dalam tipe bentukan tersebut dan apa tipe datanya?

- Variabel apa yang dideklarasikan dengan tipe bentukan tersebut?
- Tambahkan satu variabel dengan tipe bentukan tersebut dan beri nilai pada masing-masing elemen tersebut!
- Jumlahkan nilainya dengan variabel yang sudah dideklarasikan sebelumnya!

3. Latihan 3 - Kombinasi tipe data bentukan dan larik (Tipe data bentukan di dalam larik)

```
#include <iostream>
using namespace std;
typedef struct
{
    int NIM;
    string nama;
    int umur;
} Mahasiswa;

int main()
{
    Mahasiswa Mhs[10];
    int i;
    for (i=0;i<10;i++)
    {
        cout << "Data ke-" << i+1 << ":" << endl;
        cout << "NIM: ";
        cin >> Mhs[i].NIM;
        cout << "Nama: ";
        cin >> Mhs[i].nama;
        cout << "Umur: ";
        cin >> Mhs[i].umur;
    }
    cout << "|\tNIM \t|\tNAMA \t|\tUMUR \t|\n";
    for (i=0;i<10;i++)
    {
        cout << "|\t" << Mhs[i].NIM << "\t|\t" <<
        Mhs[i].nama << "\t|\t" << Mhs[i].umur << "\t|\n";
    }
    return 0;
}
```

- Apa <*Nama_Tipe*> untuk tipe data bentukan *record* yang dibentuk pada kode program tersebut?
- Apa saja elemen yang ada di dalam tipe bentukan tersebut dan apa tipe datanya?
- Variabel larik apa yang dideklarasikan dengan tipe bentukan tersebut? Berapa ukurannya?
- Jika dipresentasikan dalam bentuk gambar maka setiap indeks larik akan tampak sebagai gambar. Apa artinya?

4. Latihan 4 - Kombinasi tipe data bentukan dan larik (Larik di dalam tipe data bentukan)

```
#include <iostream>

using namespace std;

typedef struct
{
    int NIM;
    char nama[20];
    int nilai[3];
} Mahasiswa;

int main()
{
    Mahasiswa Mhs;
    int j;
    cout << "NIM: ";
    cin >> Mhs.NIM;
    cout << "Nama: ";
    cin >> Mhs.nama;
    for (j=0;j<3;j++)
    {
        cout << "Nilai ke-" << j+1 << ":";
        cin >> Mhs.nilai[j];
    }
    cout << "|\tNIM \t|\tNAMA \t|";
    for (j=0;j<3;j++)
    {
        cout << "\tNilai " << j+1 << "\t|";
    }
    cout << endl;
    cout << "|\t" << Mhs.NIM << "\t|\t" << Mhs.nama << "\t|";
    for (j=0;j<3;j++)
    {
        cout << "\t" << Mhs.nilai[j] << "\t|";
    }
    cout << endl;
    return 0;
}
```

- Apa <*Nama_Tipe*> untuk tipe data bentukan *record* yang dibentuk pada kode program tersebut?
- Apa saja elemen yang ada di dalam tipe bentukan tersebut dan apa tipe datanya? Manakah yang termasuk dalam tipe larik?
- Jika dipresentasikan dalam bentuk gambar maka setiap tipe data bentukan akan tampak sebagai gambar. Apa artinya?

5. Latihan 5 – Tipe data bentukan dalam subprogram (fungsi dan prosedur)

```
#include <iostream>
using namespace std;
typedef struct
{
    int Jam;
    int Menit;
    int Detik;
} Waktu;

Waktu Inisialisasi ()
{
    Waktu W;
    W.Jam = 0;
    W.Menit = 0;
    W.Detik = 0;
    return W;
}

void CetakWaktu (Waktu W)
{
    cout << W.Jam << ":" << W.Menit << ":" << W.Detik << endl;
}

int main ()
{
    Waktu Wkt;
    Wkt = Inisialisasi();
    CetakWaktu(Wkt);
    cout << "Atur Waktu: " << endl << "Jam: ";
    cin >> Wkt.Jam;
    cout << "Menit: ";
    cin >> Wkt.Menit;
    cout << "Detik: ";
    cin >> Wkt.Detik;
    CetakWaktu(Wkt);

    return 0;
}
```

- Dari subprogram pada kode program di atas, manakah yang menggunakan tipe data bentukan? Apa <*Nama_Tipe*>-nya?
- Manakah yang menggunakan tipe data bentukan untuk *return value* dan manakah yang menggunakan tipe data bentukan untuk parameter?
- Bagaimana subprogram tersebut mengoutputkan/mengembalikan nilai dari masing-masing elemen pada tipe data bentukan melalui *return value*?
- Bagaimana subprogram tersebut menginputkan dan mengoutputkan nilai masing-masing elemen yang ada pada tipe data bentukan melalui parameter?

IV. TUGAS

1. Buatlah kode program dengan mengkombinasikan tipe data bentukan dan larik untuk menampilkan output seperti pada tampilan berikut.

```
Data ke-1
NIM : 171010123
Nama : Kiki
Nilai ke-1 : 98
Nilai ke-2 : 98
Nilai ke-3 : 67

Data ke-2
NIM : 171010456
Nama : Hira
Nilai ke-1 : 87
Nilai ke-2 : 67
Nilai ke-3 : 56

Data Mahasiswa
1      Kiki    98    98    67
2      Hira    87    67    56
```

2. Buatlah sebuah database untuk suatu aplikasi tertentu menggunakan tipe data terstruktur dimana database tersebut dapat ditambahkan, diedit, dicari, dihapus, dan dicetak

   Catatan:

   - **Perancangan terdiri dari notasi algoritma (pseudocode atau *flowchart*) dan kode program + hasil program**
   - **Database aplikasi yang dibuat oleh setiap praktikan tidak boleh sama**