RENCANA PEMBELAJARAN STRUKTUR DATA

Dosen : Sulistyowati, ST., M.Kom.

- **MATA KULIAH : STRUKTUR DATA**
- **KODE/BOBOT/SEMESTER : 16132402 / 3 SKS / 2**
- **PRASYARAT : ALGORITMA & PEMROGRAMAN I**

- **Tujuan Pembelajaran / Learning Objective :**

  Mahasiswa dapat menyusun spesifikasi tipe data abstrak yang meliputi struktur, stack, queue, tree, linked list, pengurutan data, pencarian data, dan notasi polish serta mahasiswa mampu memilih dan menyusun representasi struktur data yang efisien

Rencana Pembelajaran

Minggu ke	Capaian Pembelajaran	Materi (Pokok Bahasan)
1	Mahasiswa mampu mengimplementasikan dalam program konsep Struktur (Record)	Rencana Pembelajaran, Konsep struktur/record, struktur dalam array dan implementasinya dalam program
2	Mahasiswa mampu memahami konsep Pointer dan mengimplementasikan dalam program konsep Single Linked List (Tambah Simpul)	Pointer, Menambah Simpul Depan Single Linked List, Menambah Simpul Tengah Single Linked List, Menambah Simpul Belakang Single Linked List
3	Mahasiswa mampu mengimplementasikan dalam program konsep Single Linked List (Hapus Simpul)	Menghapus Simpul Depan Single Linked List, Menghapus Simpul Tengah Single Linked List, Menghapus Simpul Belakang Single Linked List
4	Mahasiswa mampu mengimplementasikan dalam program konsep Single Linked List (Cari dan Cetak Simpul)	Mencari Data/Simpul di Single Linked List, Mencetak Data/Simpul di Single Linked List
5	Mahasiswa mampu mengimplementasikan dalam program konsep Double Linked List (Tambah Simpul)	Menambah Simpul Depan Double Linked List, Menambah Simpul Tengah Double Linked List, Menambah Simpul Belakang Double Linked List

Rencana Pembelajaran

Minggu ke	Capaian Pembelajaran	Materi (Pokok Bahasan)	
6	Mahasiswa mampu mengimplementasikan dalam program konsep Double Linked List (Hapus Simpul)	Menghapus Simpul Depan Double Linked List, Menghapus Simpul Tengah Double Linked List, Menghapus Simpul Belakang Double Linked List	150
7	Mahasiswa mampu mengimplementasikan dalam program konsep Double Linked List (Cari dan Cetak Simpul)	Mencari Data/Simpul di Double Linked List, Mencetak Data/Simpul di Double Linked List	150
8	UTS (Evaluasi ketercapaian dari pertemuan minggu ke 1 sampai 7)	UTS (Evaluasi ketercapaian dari pertemuan minggu ke 1 sampai 7)	150
9	Mahasiswa mampu mengimplementasikan dalam program konsep Tumpukan (Stack)	Penyajian Stack, Operasi PUSH, Operasi POP	150
10	Mahasiswa mampu mengimplementasikan dalam program konsep Antrian (Queue)	Penyajian Queue, Operasi ENQUEUE, Operasi DEQUEUE	150

Rencana Pembelajaran

Minggu ke	Capaian Pembelajaran	Materi (Pokok Bahasan)
11	Mahasiswa mampu mahami konsep Notasi Polish	Notasi INFIX, Perubahan Notasi NFIX ke POSTFIX, Perubahan Notasi InfoIX ke PREFIX
12	Mahasiswa mampu mengimplementasikan dalam program konsep Notasi Polish (Implementasi Dengan Stack dan Queue)	Implementasi Stack dan Queue pada Perubahan Notasi INFIX ke PREFIX, Implementasi Stack dan Queue pada Perubahan Notasi INFIX ke POSTFIX
13	Mahasiswa mampu mengimplementasikan dalam program konsep Pengurutan (Sorting) - Part I	Insertion Sort, Selection Sort, Bubble Sort
14	Mahasiswa mampu mengimplementasikan dalam program konsep Pengurutan (Sorting) - Part II	Quick Sort, Radix Sort
15	Mahasiswa mampu mengimplementasikan dalam program konsep Pencarian (Searching)	Binary Search, Sequential Search
16	UAS (Evaluasi ketercapaian dari pertemuan minggu ke 9 sampai 15)	UAS (Evaluasi ketercapaian dari pertemuan minggu ke 9 sampai 15)

Pustaka :

1. D. Suryadi H. S., **Pengantar Struktur Data**, Penerbit Gunadarma
2. Loomis, Mary E. S., **Data Management and File Structures**, Prentice Hall International Inc
3. Reynolds, W. Charles, **Program Design and Data Structures in Pascal**, Wadsworth Pub. Co.
4. Insap Santoso, **Struktur Data**, Andi Offset Yogyakarta

STRUKTUR / RECORD

Dosen :
Sulistyowati, ST., M.Kom.

Pendahuluan

- Pada pembahasan sebelumnya, Array merupakan sebuah tipe data terstruktur/bentukan yang berupa sekumpulan data dengan tipe data sama.
- Bagaimana jika kita ingin menyimpan sekumpulan data yang tipe datanya berbeda-beda dalam satu variabel?
- **Struktur/Record** adalah sebuah tipe data terstruktur yang berupa sekumpulan data dengan tipe data berbeda.

Deklarasi Struktur

- Bentuk deklarasi struktur :

```
struct  Nama_struktur
{
        Tipe_data1  Nama_variabel_field1;
        Tipe_data2  Nama_variabel_field2;
        ....
        Tipe_datax  Nama_variabel_fieldx;
 } ;
```

Dimana :

- **Nama_struktur** = tipe data abstrak (yaitu tipe data yang dibuat sendiri oleh user) yg berupa tipe data struktur
- **Nama_variabel_field1,…,Nama_variabel_fieldx** = field / elemen / anggota struktur
- **Tipe_data1,…,Tipe_datax** = tipe data dari masing-masing field

Deklarasi Struktur (con't)

Contoh deklarasi struktur :

```
struct Master_Barang
{
   char Kode_Barang[7];
   char Nama_Barang[25];
   int Jumlah_Barang;
   float Harga_Satuan;
};
Master_Barang Data_Barang,B;
```

Dari contoh diatas, dideklarasikan sebuah tipe data struktur Bernama *Master_Barang,* dengan elemen/field : *Kode_Barang, Nama_Barang, Jumlah_Barang* dan *Harga_Satuan.*

Definisi Variabel Struktur

- Setelah tipe data struktur dideklarasikan, struktur ini dapat digunakan untuk mendefinisikan suatu variabel.

- Bentuk definisi variabel struktur :

  tipe_data_struktur nama_variabel_struktur;

  Contoh :

  ```
  Master_Barang  Data_Barang;
  ```

- Contoh diatas merupakan pendefinisian variabel struktur ➔ *Data_Barang, yang* bertipe data struktur *Master_Barang.* Dengan pendefinisian ini, *Data_Barang* memiliki 4 buah field yaitu :

  1. Kode_Barang
  2. Nama_Barang
  3. Jumlah_Barang
  4. Harga_Satuan

Definisi Variabel Struktur (con't)

- Bentuk lain definisi variabel struktur :

```
struct  Nama_struktur
{
    Tipe_data1  Nama_variabel1;
    Tipe_data2  Nama_variabel2;
    ….
    Tipe_datax  Nama_variabelx;
} nama_variabel_struktur ;
```

ATAU

```
struct
{
    Tipe_data1  Nama_variabel1;
    Tipe_data2  Nama_variabel2;
    ….
    Tipe_datax  Nama_variabelx;
} nama_variabel_struktur ;
```

Definisi Variabel Struktur (con't)

Contoh :

```
struct Master_Barang
{
   char Kode_Barang[7];
   char Nama_Barang[25];
   int Jumlah_Barang;
   float Harga_Satuan;
} Data_Barang;
```

```
struct
{
   char Kode_Barang[7];
   char Nama_Barang[25];
   int Jumlah_Barang;
   float Harga_Satuan;
} Data_Barang;
```

Definisi Variabel Struktur (con't)

- Jika 2 buah atau lebih variabel yang mempunyai tipe data struktur yang sama, dapat ditulis seperti pada contoh berikut :

```
struct Master_Barang
{
    char Kode_Barang[7];
    char Nama_Barang[25];
    int Jumlah_Barang;
    float Harga_Satuan;
}

Master_Barang Data_Barang,DB;
```

Pada contoh ini, variabel Data_Barang dan DB sama-sama bertipe data struktur Master_Barang.

```
struct Master_Barang
{
    char Kode_Barang[7];
    char Nama_Barang[25];
    int Jumlah_Barang;
    float Harga_Satuan;
} Data_Barang,DB;
```

Struktur Dalam Struktur (Nested Structure)

- Suatu struktur juga bisa mengandung struktur yang lain. Contohnya :

```
struct Master_Tanggal
{
   int Tanggal; int Bulan; int Tahun;
};

struct Master_Barang
{
   Master_Tanggal Tanggal_Barang;
   char Kode_Barang[7]; char Nama_Barang[25];
   int Jumlah_Barang; float Harga_Satuan;
} Data_Barang;
```

Struktur Dalam Struktur (con't)

Gambar elemen/field dari variabel Data_Barang :

Mengakses Elemen Struktur

- Bentuk :

  nama_variabel_struktur.nama_variabel_field;

  Tanda titik diberikan diantara nama variabel struktur dan nama variabel field.

  Contoh :

```
Data_Barang.Tanggal_Barang.Tanggal = 5;
Data_Barang.Tanggal_Barang.Bulan = 7;
Data_Barang.Tanggal_Barang.Tahun = 2011;
Data_Barang.Kode_Barang = “KA-991”;
Data_Barang.Nama_Barang = “Sepatu”;
Data_Barang.Jumlah_Satuan = 100;
Data_Barang.Harga_Satuan = 3500;
```

Penugasan Struktur

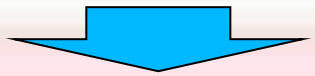
- Jika 2 buah variabel struktur mempunyai tipe sama (mempunyai jumlah dan nama field yang sama), maka pemberian nilai terhadap suatu struktur dapat dilakukan dengan bentuk :

  nama_variabel_struktur1 = nama_variabel_struktur2;

  Contoh :

```
…
Master_Barang  Data_Barang,DB;

DB = Data_Barang;
```



seluruh elemen pada variabel Data2_Barang diisi dengan elemen terkait yang ada pada variabel Data_Barang.

Penugasan Struktur (con't)

- Pernyataan :

```
DB = Data_Barang;
```

merupakan penyederhanaan dari 7 pernyataan berikut :

- `DB.Tanggal_Barang.Tanggal = Data_Barang.Tanggal_Barang.Tanggal`
- `DB.Tanggal_Barang.Bulan = Data_Barang.Tanggal_Barang.Bulan`
- `DB.Tanggal_Barang.Tahun = Data_Barang.Tanggal_Barang.Tahun`
- `DB.Kode_Barang = Data_Barang.Kode_Barang`
- `DB.Nama_Barang = Data_Barang.Nama_Barang`
- `DB.Jumlah_Barang = Data_Barang.Jumlah_Barang`
- `DB.Harga_Satuan = Data_Barang.Harga_Satuan`

Pembandingan Struktur

- Untuk membandingkan isi dari 2 buah variabel struktur tidak bisa dilakukan secara langsung seperti berikut :

```
if (DB == Data_Barang)
    pernyataan;
```

- Sehingga untuk membandingkan 2 buah struktur maka masing-masing field/elemen harus dibandingkan secara sendiri-sendiri, seperti pada contoh berikut :

```
if ((DB.Tanggal_Barang.Tanggal==Data_Barang.Tanggal_Barang.Tanggal) &&
    (DB.Tanggal_Barang.Bualan==Data_Barang.Tanggal_Barang.Bulan) &&
    (DB.Tanggal_Barang.Tahun==Data_Barang.Tanggal_Barang.Tahun) &&
    (DB.Kode_Barang==Data_Barang.Kode_Barang) &&
    (DB.Nama_Barang==DB.Nama_Barang) &&
    (DB.Jumlah_Barang==DB.Jumlah_Barang) &&
    (DB.Harga_Satuan==DB.Harga_Satuan))
```

Struktur Dalam Array

```
main()
{
	int i; char x[2];
	struct tgl
	{
		int tanggal, bulan, tahun;
	};
	struct Master
	{
		char nama[35], alamat[50], npm[15];
		tgl t_lahir;
		char jkel;
		int umur;
	} Data_mhs[10];

	for(i=1; i<=2; i++)
	{
		cout<<"Inputkan Data Mahasiswa ke - "<<i<<endl;
		cout<<“	NAMA			: "; gets(Data_mhs[i].nama);
		cout<<"	NPM			: "; gets(Data_mhs[i].npm);
		cout<<"	ALAMAT			: "; gets(Data_mhs[i].alamat);
		cout<<"	TANGGAL LAHIR		: "; cin>>Data_mhs[i].t_lahir.tanggal;
		cout<<"	BULAN LAHIR		: "; cin>>Data_mhs[i].t_lahir.bulan;
		cout<<"	TAHUN LAHIR		: "; cin>>Data_mhs[i].t_lahir.tahun;
		cout<<"	JENIS KELAMIN		: "; cin>>Data_mhs[i].jkel;
		cout<<"	UMUR			: "; cin>>Data_mhs[i].umur;
		cin.getline(x,sizeof(x));
		cout<<endl;
	}
```

Struktur Dalam Array (lanjutan)

```
        cout<<"========== DATA MAHASISWA ============"<<endl<<endl;
        for (i=1; i<=2; i++)
        {
                cout<<" MAHASISWA KE - "<<i<<endl;
                cout<<" -------------------------"<<endl;
                cout<<"    NAMA                 : "<<Data_mhs[i].nama<<endl;
                cout<<"    NPM                  :  "<<Data_mhs[i].npm<<endl;
                cout<<"    ALAMAT               :  "<<Data_mhs[i].alamat<<endl;
                cout<<"    TANGGAL LAHIR  :  "<<Data_mhs[i].t_lahir.tanggal<<"-”<
                                        Data_mhs[i].t_lahir.bulan<<"-"<<Data_mhs[i].t_lahir.tahun<<endl;
                cout<<"    JENIS KELAMIN   :  "<<Data_mhs[i].jkel<<endl;
                cout<<"    UMUR                 :  "<<Data_mhs[i].umur<<endl<<endl;
        }
        getch();
}
```

Soal-soal Latihan

Buatlah program C++ untuk :

1. Menginputkan dan mencetak data pegawai sebanyak n pegawai. Dimana data pegawai yang disimpan adalah : nama, npp (nomor pokok pegawai), alamat, jenis kelamin, tanggal masuk (tanggal masuk terdiri dari tanggal,bulan dan tahun), status pegawai (meliputi pegawai honorer dan pegawai tetap), gaji.

2. Membuat menu untuk :
   Menginputkan dan mencetak data mahasiswa sebanyak n mahasiswa. Dimana data mahasiswa yang disimpan adalah : nama, npm, jurusan (jurusan yang ada yaitu : T. Informatika, T.Elektro, T.Mesin, T.Sipil, T.Kimia, T.Industri), alamat, jenis kelamin, tempat /tanggal lahir (tanggal lahir meliputi tanggal, bulan, tahun), umur, Indeks Prestasi (IP terdiri dari IPS dan IPK).