

Record dan Array of Record



Akmal, S.Si, MT

Mata Kuliah : Struktur Data

Tujuan

- ❑ Mahasiswa dapat menjelaskan pengertian record dan bisa menerangkan operasi dasar menggunakan record dan Array or Record dengan benar dalam mendefinisikan suatu ADT (Abstrak Data Type)
- ❑ Mahasiswa dapat : Mengoperasikan dan membuat program dengan algoritma yang benar menggunakan record dan array of record.

Pokok Bahasan

- Record
- Array of Record

Record / Structure → ADT

- ❑ Record adalah suatu tipe data bentukan yang merupakan kumpulan dari atribut-atribut suatu objek.
- ❑ Pada record tipe elemen bisa berbeda-beda tidak seperti array yang mengharuskan mempunyai tipe elemen yang sama.
- ❑ Pendefinisian data dengan menggunakan tipe ini merupakan suatu cara untuk membentuk Abstrak Data Type (ADT)
 - ADT adalah definisi dari TYPE (sekumpulan objek) dan sekumpulan operasi dasar (PRIMITIF) dari TYPE tersebut

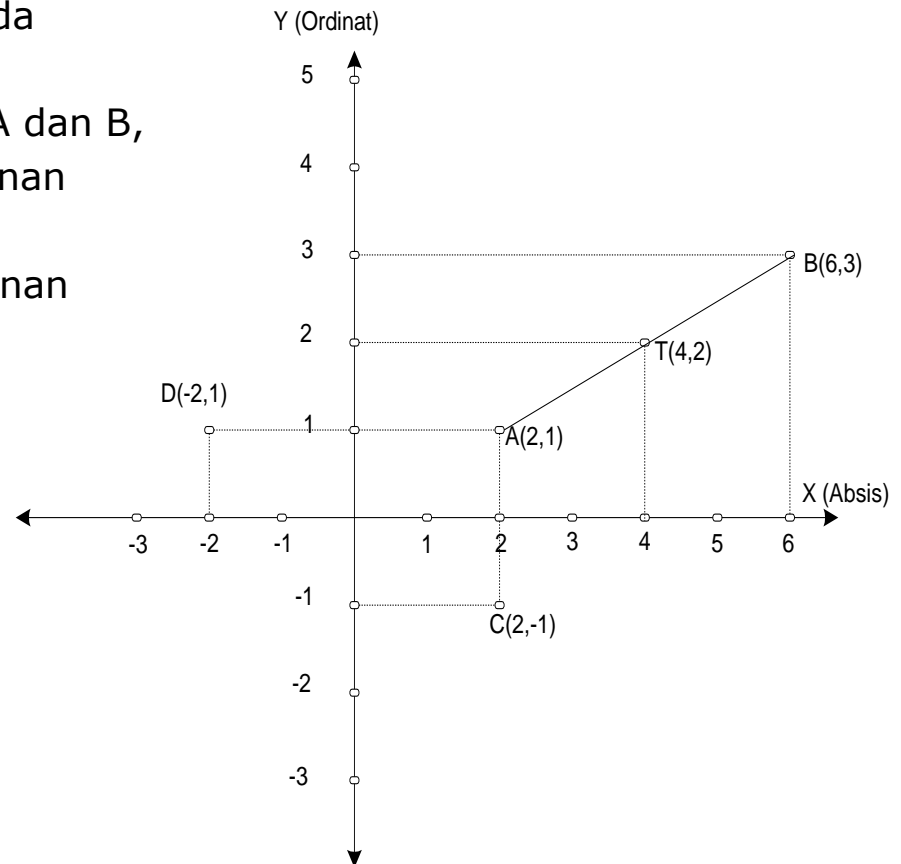
Deklarasi record / structure

```
struct namaRecord {  
    <type Atribut 1> Atribut1;  
    <type Atribut 2> Atribut2;  
    . . . . .  
    <type Atribut n> Atributn;  
};
```

- Cara mengakses variabel di dalam record/struct adalah dengan operator (.).
- Misalkan terdapat nama record/struct *mhs*, dan variabel yang akan diakses di dalamnya adalah *npm*, maka cara mengaksesnya adalah "*mhs.npm*" yang artinya kita mengakses *npm* yang merupakan satu atribut dari data *mhs*.

Contoh:

- Akan dibuat sebuah program untuk mengelola koordinat kartesius yang memiliki koordinat (absis, ordinat).
- Program akan dipecah menjadi input, proses dan output.
- Struktur data yang digunakan adalah tipe record.
- A, B, C, D, T adalah titik-titik yang ada dalam koordinat kartesian
- T merupakan titik tengah dari 2 titik A dan B,
- C merupakan titik hasil dari pencerminan titik A terhadap sumbu X
- D merupakan titik hasil dari pencerminan titik A terhadap sumbu Y



```
struct koordinat {  
    float absis;  
    float ordinat;  
};  
  
void getPoint( koordinat& ttk){  
    cout<<"Masukkan absis    = ";cin >> ttk.absis;  
    cout<<"Masukkan ordinat = ";cin >> ttk.ordinat;  
}  
  
void printPoint( koordinat ttk){  
    cout<< "("<<ttk.absis<<","<<    ttk.ordinat<<")"<<endl;  
}  
  
main(){  
    koordinat a,b,c;  
    cout << "Input Titik a "<<endl; getPoint(a);  
    cout << "Input Titik b "<<endl; getPoint(b);  
    cout << "Titik a = "; printPoint(a);  
    cout << "Titik b = "; printPoint(b);  
}
```

Array of Record

- Elemen array dapat juga digunakan untuk data yang bertipe terstruktur (record).

Caranya :

1. buat tipe nama record / structure
2. buat nama alias tipe array yang elemennya record/struct
3. deklarasikan variable-variabel yang mengacu pada nama tipe alias array tsb.

Contoh deklarasi array of record

```
struct mahasiswa {  
    char NPM[8] ;  
    char nama[20];           // atau string nama  
    int nilai;  
};  
  
typedef  mahasiswa LarikMhs[10];  
  
LarikMhs  mhs;
```

- Misalkan akan dibuat 3 buah fungsi untuk menentukan ukuran dari array, input data dan pencetakan data dengan header sbb :
 void banyakData(int& n);
 void inputMahasiswa (LarikMhs& Mhs, int n);
 void cetakMahasiswa (LarikMhs Mhs, int n);

```

void banyakData(int& n);
void inputMahasiswa (LarikMhs& Mhs, int n);
void cetakMahasiswa(LarikMhs Mhs, int n);
main(){
    LarikMhs mhs;
    int n;
    banyakData(n);
    inputMahasiswa(mhs,n);
    cetakMahasiswa(mhs,n);
    getch();
}

void banyakData(int& n){                // Input banyak data
    cout<<"Banyak data : "; cin>>n;
}

void inputMahasiswa (LarikMhs& Mhs, int n) {
    for (int i=0;i<n;i++) {
        cout<<"masukan data mahasiswa ke- "<<(i+1)<<endl;
        cout<<"NPM      : "; cin>>Mhs[i].NPM;
        cout<<"Nama      : "; cin>>Mhs[i].nama;                // bisa dg getline()
        cout<<"Nilai     : "; cin>>Mhs[i].nilai;
    }
}

void cetakMahasiswa(LarikMhs Mhs, int n){
    cout<<"PENCETAKAN DATA MAHASISWA  "<<endl;
    for (int i=0;i<n;i++) {
        cout<<Mhs[i].NPM<<Mhs[i].nama<<Mhs[i].nilai<<endl;
    }
}

```