
Sesi/Perkuliahan ke: IX

Tujuan Instruksional Khusus :

1. Mahasiswa dapat menjelaskan pengertian tentang record.
2. Mahasiswa dapat membuat deklarasi tipe data record.
3. Mahasiswa dapat membuat contoh program sederhana dengan menggunakan tipe data record.

Pokok Bahasan : RECORD

Deskripsi singkat : Dalam pertemuan ini akan mempelajari tentang Konsep Pendeklarasian pada record, Pengaksesan ke data record dengan Notasi dan With, Tipe data record dengan field tipe record

Referensi :

1. Anonim, "Algoritma & Pemrograman II", Penerbit Gunadarma, Jakarta, 1990
2. Bryon, Goffried, "Programming with PASCAL", Schaum Series, New York, 1986
3. Prather, Ronald E., "Problem Solving Principle : Programming with Pascal", Prentice Hall, New Jersey, 1982
4. Dumairy, Matematika Terapan untuk Bisnis & Ekonomi Press (BPFE Yogyakarta)
5. Yogiarto H.M, Turbo Pascal 5.0 Penerbit Andi Offset, Yogyakarta.
6. Ediman Lukito, Belajar Sendiri Pemrograman dengan Turbo Pascal 7.0
7. Ir. P. Insap Santosa, Turbo Pascal versi 5.0 dan 5.5, Elexmedia Komputindo

RECORD

Record dapat dikatakan sebagai suatu kumpulan data item yang masing-masing mempunyai jenis data berbeda.

Data item yang merupakan elemen record biasanya disebut dengan FIELD.

CARA MENDEKLARASIKAN RECORD

Bentuk umum deklarasi suatu variabel berjenis record adalah sbb :

```
TYPE identifier = RECORD
    Nama_field_1 : jenis;
    Nama_field_2 : jenis;
    .....
    .....
    nama_field_n : jenis;
END;
```

Contoh :

1. VAR nilai : RECORD

```
    Nilai_1 : integer;
    Nilai_2 : integer;
END;
```

2. TYPE date = RECORD

```
    Tanggal : 1..31;
    Bulan : 1...12;
    Tahun : 1900..2000;
END;
```

```
VAR event1,event2 : ARRAY [1..10] OF date;
```

3. TYPE account = RECORD

```
    cust_no      : integer;  
    cust_type    : char;  
    cust_balance : real;  
    END;
```

```
VAR customer : account;
```

MEMPROSES VARIABEL BERJENIS RECORD

Perhatikan deklarasi variabel berikut :

```
TYPE nilai : RECORD  
    Nilai1 : real;  
    Nilai2 : real;  
    END;  
VAR x,y : nilai;
```

Untuk memproses variabel x dan / atau y dilakukan dengan cara menyebutkan field designatornya, yg terdiri dari atas :

Nama_record.nama_field

Pada deklarasi diatas yang dimaksud dengan field designator-nya adalah :

```
x.nilai1  
x.nilai2  
y.nilai1  
y.nilai2
```

Jadi jika ingin membaca variabel x atau y atau keduanya, maka bentuk statement-nya adalah :

```
READ (x.nilai1, x.nilai2, y.nilai1, y.nilai2);
```

Selanjutnya, misal ingin dibuat program sederhana untuk menjumlahkan dua bilangan kompleks a dan b yang hasilnya disimpan di c.

Secara aljabar penjumlahan dua bilangan kompleks adalah sebagai berikut :

$$a = x1 + iy1$$

$$b = x2 + iy2 +$$

$$c = (x1 + x2) + i(y1 + y2)$$

Maka bentuk garis besar programnya adalah sebagai berikut :

Program contoh ;

Type bk = record

 Bag_nyata : integer;

 Bag_imajiner : integer;

End;

Var a,b,c : bk;

Begin

 Read (a.bag_nyata, a.bag_imajiner, b.bag_nyata, b.bag_imajiner);

 c.bag_nyata := a.bag_nyata + b.bag_imajiner;

 c.bag_imajiner := a.bag_imajiner + b.bag_imajiner;

 writeln(c.bag_nyata, ' + ', 'i', c.bag_imajiner);

End.

STATEMENT “WITH”

Selain cara yang telah disebutkan diatas, untuk memproses suatu record dapat digunakan statement WITH.

Dengan statement ini penulisannya akan lebih sederhana.

Bentuk Umum penulisan statement WITH ini adalah :

```
WITH nama_record DO statement
```

Perhatikan deklarasi dibawah ini :

```
TYPE x = RECORD
    No    : integer;
    Kode  : char;
    Juml  : integer;
    Harga : real;
END;
VAR p,q : x;
```

Untuk membaca variabel p dan q di atas dengan memanfaatkan statement WITH bentuknya menjadi :

```
WITH p,q DO read (no, kode, juml, harga);
```

Bandingkan jika digunakan cara sebelumnya :

```
Read(p.no, p.kode, p.juml,p.harga,q.no,q.kode,q.juml,q.harga);
```

CONTOH :

Pernyataan seperti :

```
Data.npm := '22297566'  
Data>Nama := 'Abdul Kadir'  
Data.Fakultas := 'Teknik'
```

Dapat diganti dengan :

```
WITH Data Do  
Begin  
npm := '22297566'  
Nama := 'Abdul Kadir'  
Fakultas := 'Teknik'  
end;
```

Apabila di dalam pernyataan WITH terdapat lebih dari satu record, haruslah pada kedua record tersebut tidak ada field dengan nama yang sama. Sebagai contoh :

```
Type  
BarangX=RECORD  
    Batu:integer;  
    Kayu:real;  
END;
```

```
BarangY=RECORD  
    Batu:string[10];  
    Kayu:char;  
END;
```

```
Var  
brg_X:barangX;  
brg_Y:barangY;
```

Karena kedua variabel record brg_X dan brg_Y memiliki nama field yang sama, Jika misalnya kemudian dikenakan pernyataan :

```
WITH brg_X, brg_Y Do
Begin
  writeln(batu);
  writeln(kayu);
End;
```

dapat menyebabkan hasil tidak seperti yang diharapkan.

Record yang Bervariasi

Yaitu suatu record dengan field yang bisa berubah pada saat program berjalan. Hal yang perlu diperhatikan adalah bahwa beberapa field yang berada dalam record tidak pernah muncul dalam serempak, hanya akan ada satu field yang terpakai dalam satu saat.

Record varian akan memberikan fasilitas untuk menentukan field yang diperlukan pada saat program berjalan (RUN-TIME), berdasarkan keperluannya. Bentuk umum Record Varian :

```
TYPE namarecord = RECORD
  Nama_field_1 : jenis;
  Nama_field_2 : jenis;
  .....
  nama_field_n : jenis;
  Case Tagfield:jenis Of
    nama_tagfield : (Nama_field:jenis);
    nama_tagfield : (Nama_field_1, Nama_field_2:jenis);
    .....
    nama_tagfield : (Nama_field_n:jenis);
  END;
```

Contoh :

```
Type status=(T,P,J);
  gaji=RECORD
    napeg    :string[25];
    nopeg    :string[10];
    bagian   :string[15];
  CASE stat :status OF
    T:(gaji:integer);
    P:(gajiperjam,jmlmax:integer);
    J:(upahperjam,lembur:integer);
  end;
```

Array tipe record

```
type barang=RECORD
  namabrg:string[20];
  jmlbrg:byte;
  hargabrg:real;
  total:real;
end;

var jual:array [1..10] of barang
  i,j:integer;
  tot1:real;

Begin
  l:=1;
  write('Nama barang :');readln(jual[i].nmbrg);
  Repeat
    write('Jumlah barang :');readln(jual[i].jmlbrg);
    write('Harga barang :');readln(jual[i].hrgrg);
```

```

    jual[i].total:=jual[i].jmlbrg* jual[i]. hrg.brg;
    inc (l);
    write('Nama barang :');readln(jual[i].nmbrg);
    until (jual[i].nmbrg='0') or (l=11);
dec(i);
clrscr;
writeln ('-----');
writeln ('No nama barang  jumlah harga  total');
writeln ('-----');
    for j:=1 to l do
    begin
        gotoxy(1,j+3);write (j);
        gotoxy(5,j+3);write(jual[i].nmbrg);
        gotoxy(26,j+3);write(jual[i].jmlbrg:10);
        gotoxy(37,j+3);write(jual[i].hrgbrg:10:2);
        gotoxy(48,j+3);write(jual[i].total:10:2);
        tot1:=tot1 + jual[j].total ;
    end;
writeln ('-----');
writeln('                Total :',tot1:10:2');
writeln ('-----');
end.

```

Array dalam record

```
Type barang= RECORD
    nmbrg:string[20];
    jmlbrg:array[1..3]of byte;
    hrgbrg:real;
    total:real;
end;
var jual:barang;
    tbarang, i:integer;
Begin
    clrscr;
    write('Nama Barang :');readln(jual.nmbrg);
    for i:=1 to 3 do
        begin
            write('Jumlah barang ',i,' : ');readln(jual.jmlbrg[i]);
            tbarang:=tbarang+jual.jmlbrg[i];
        end;
    write('Harga barang :');readln(jual.hrgbrg);
    jual.total:=tbarang*jual.hrgbrg;
    writeln('Total Harga Barang = ', jual.total:10:2);
end.
```

Coba lihat sendiri perbedaan antara array tipe record dan array dalam record dari dua contoh program di atas !