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
I:=1;
write('Nama barang :');readln(jual[i].nmbrg);
   Repeat
   write('Jumlah barang :');readln(jual[i].jmlbrg);
   write('Harga barang :');readln(jual[i].hrgbrg);
   jual[i].total:=jual[i].jmlbrg* jual[i]. jual[i].hrg.brg;
   inc (I);
   write('Nama barang :');readln(jual[i].nmbrg);
   until (jual[i].nmbrg='0') or (I=11);
dec(i);
clrscr;
writeln ('------------------------------------------------------------');
writeln ('No  nama barang      jumlah     harga      total');
writeln ('------------------------------------------------------------');
     for j:=1 to I 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:=1to 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 !

SET (HIMPUNAN)

Set adalah kumpulan dari nilai-nilai yang memiliki kesamaan sifat, yaitu tipe data yang sama dan urutan penulisannya tidak diperhatikan. Setiap obyek dalam himpunan disebut dengan anggota atau elemen himpunan. Contoh :

1. Huruf vokal dalam alfabet
2. kendaraan bermotor
3. binatang menyusui
4. bilangan genap 1 sampai 100
5. dan sebagainya

Set termasuk ke dalam tipe data terstruktur, terdiri dari sejumlah elemen yang bertipe sama dan harus bertipe data ordinal yang memiliki nilai terletak antara 0 sampai 255.

Berbeda dengan tipe data array, tipe data SET tidak mengenal elemen pertama, elemen kedua dan seterusnya. Pada tipe data SET tidak terdapat elemen yang bernilai sama.

Mendeklarasikan SET (Himpunan)

Dapat dideklarasikan dalam deklarasi tipe data :

Type variabel_set = **SET OF** tipe_data;

Dapat pula dideklarasikan dalam deklarasi var data :

Var variabel_set = **SET OF** tipe_data;

Contoh :

```
Type
kata          = set of char; {tipe dasar}
huruf besar   ='A'..'Z';  {subrange}
Perintah      = set of hurufbesar ;
Hari          = (senin,selasa,rabu,kamis,jumat,sabtu,minggu); {enumerasi}
Hari2         = set of Hari;
Pilihan       = set of 1..7 {subrange dalam bentuk bilangan};
```

```
Var
hurufkecil   : set of 'a'..'z';
kapital              : hurufbesar;
hariseminggu         : hari2;
Kalimat      : kata;
```

Dalam keadaan awal, suatu variabel yang bertipe data SET belum terisi. Variabel tersebut merupakan suatu **variabel kosong**, dinyatakan dengan tanda **[ ]**.

Contoh program memberikan nilai untuk tipe data SET :

```
Program SET1;
    type
      Hari          = (senin,selasa,rabu,kamis,jumat,sabtu,minggu); {enumerasi}
      Hari2         = set of Hari;
    var
      hariseminggu:hari2;
begin
   hariseminggu:=[selasa, kamis,sabtu];
end.
```

Operasi SET (Himpunan)

1. Operasi Penjumlahan

Disebut juga operasi penggabungan (UNION), Operatornya (+).
Contoh :
A:= ['B','F','J','L'];
A:= A + ['a','B','b','c','d','e','f','j','k','l'];
maka nilai A sekarang adalah ['B','F','J','L','a','b','c','d','e','f','j','k','l']

2. Operasi Pengurangan

Disebut juga operasi selisih SET, operatornya (-).

Misal :
SET1 - SET2, menghasilkan elemen yang merupakan anggota SET1 yang bukan anggota SET2.

Contoh :
A:=[1,2,3,4,5];
B:=[4,5,6,7];
W:=A-B;
Maka nilai W adalah [1,2,3]

2. Operasi Perkalian

Disebut juga operasi Intersection (irisan), operatornya (*).
Misal :
SET1 - SET2, menghasilkan elemen yang merupakan anggota SET1 dan sekaligus merupakan anggota SET2.

Contoh :
1. A:=[1,2,3,4,5];
2. D:=[1,2,3,4,5];

B:=[1,3,5,6,7];
C:=A*B;
Maka nilai C adalah [1,3,5]

E:=[6,7,8,9,10];
F:=A*B;
Maka nilai F adalah [ ]

Perbandingan dalam SET (Himpunan)

OPERATOR	NAMA OPERATOR	KETERANGAN
=	Sama dengan	Bernilai TRUE, bila kumpulan anggota variabel SET mempunyai anggota yang sama.
< >	Tidak sama	Bernilai TRUE, bila kumpulan anggota variabel SET mempunyai anggota yang tidak sama.
< =	Anggota dari	Bernilai TRUE, bila semua elemen dari himpunan 1 terdapat pada himpunan 2.
> =	Mempunyai anggota	Bernilai TRUE, bila semua elemen dari himpunan 2 terdapat pada himpunan 1.
IN	Terdapat di dalam	Bernilai TRUE, bila elemen merupakan anggota himpunan.

Contoh :

1. [3,4,5]=[4,5,3] hasil TRUE
2. [c,d]=[d,g,c] hasil FALSE
3. [2,4]< >[2,3] hasil TRUE
4. f in [‘a’..’z’] hasil TRUE
5. [2,1] < = [1,2,3] hasil TRUE
6. [5,6,7] > = [7,6] hasil TRUE
7. 3 in [4,5,3,2] hasil TRUE
8. [2,1] > = [1,2] hasil TRUE

Contoh Program perbandingan SET :

```
Program Set1;
Uses crt;
Type kbil=set of byte;
var  kbil1,kabil2:kabil;
     a,b:byte;
begin
  clrscr;
  write(‘Tentukan batas bawah :’);readln(a); {Input Himpunan Pertama}
  write(‘Tentukan batas Atas   :’);readln(b);
  Kbil1:=[a..b];
  write(‘Tentukan batas bawah :’);readln(a); {Input Himpunan Kedua}
  write(‘Tentukan batas Atas   :’);readln(b);
  Kbil2:=[a..b];
  if kbil1= kbil2 then writeln(‘Himpunan 1 sama dengan Himpunan 2’);
  if kbil1 < > kbil2 then writeln(‘Himpunan 1 tidak sama dengan Himpunan 2’);
  if kbil1< =  kbil2 then writeln(‘Himpunan 1 anggota dari Himpunan 2’);
  if kbil1> =  kbil2 then writeln(‘Himpunan 1 mempunyai anggota Himpunan 2’);
end.
```

Contoh Program operator IN :

```
Program set2;
uses crt;
type karakter= set of char;
var  kapital : karakter;
     hidup : karakter;
     jumlahh,jumlahk : integer;
     kalimat : string;
```

```
    i:integer;
begin
  clrscr;
  kapital:=['A'..'Z'];
  hidup:=['A','E','I','O','U','a','e','i','o','u'];
  jumlahh:=0;
  jumlahk:=0;
  writeln('Masukkan suatu kalimat :');
  readlne (kalimat);
  for I:=1 to length(kalimat) do
  begin
  if kalimat[i] in kapital then inc(jumlahk);
  if kalimat[i] in hidup then inc(jumlahh);
  end;
  writeln('Jumlah kapital =',jumlahk);
  writeln('Jumlah huruf hidup =',jumlahh);
end.
```

Aplikasi Penggunaan SET dalam Program :

```
Program bacakarakter;
uses crt;
type karakter=set of char;
var pilihan:char;

function bacakar(pesan:string;pilihansah:karakter):char;
var ch:char;
begin
  write(pesan);
  repeat
    ch:=upcase(readkey);
  until(ch in pilihansah);
  writeln(ch);
  bacakar:=ch;
end;
{Program utama}
Begin
   clrscr;
   repeat
   Pilihan:=bacakar('Anda :(T)ulis Halo, (B)el,(S)elesai',['T','B','S']);
   Case pilihan of
        'T':writeln('Hallo...');
        'B':writeln('bel'+ #7);
        'S':writeln('Terimakasih');
   end;
   Until pilihan ='S';
end.
```