

BAB III

TIPE DATA ARRAY, SET DAN STRUKTUR DATA RECORD

TUJUAN PRAKTIKUM

1. Mengerti dalam penggunaan Array.
2. Mengerti struktur data dalam bentuk Record.

TEORI PENUNJANG

Turbo Pascal menyediakan beberapa tipe terstruktur seperti array, record, file dan set. Dalam bab ini yang akan dibahas adalah array, set dan record.

Array

Array merupakan kumpulan data yang sejenis dan disimpan dalam memori komputer. Setiap elemen array dibedakan atas indeksinya. Indeks adalah batas bawah dan batas atas indeks yang menyatakan jumlah anggota dalam array. Beberapa indeks dapat dibuat sekaligus dengan simbol pemisah adalah tanda koma (,), yang berarti dimensi array lebih dari satu. Elemen boleh bertipe apa saja tetapi indeks tipenya harus ordinal.

Set

Suatu set (himpunan) adalah suatu kumpulan dari obyek yang mempunyai urutan yang dianggap sebagai suatu kesatuan. Tiap-tiap obyek di dalam set disebut dengan anggota (member) atau elemen himpunan (elemen of the set). Anggota-anggota dari set harus mempunyai tipe yang sama dan disebut dengan base type dan base type harus berbentuk tipe data sederhana yaitu bertipe ordinal (integer,boolean,char,skalar kecuali real).

Sintak : Set of tipe ordinal

Contoh

Type

Jumlah = set of char

Pembentuk set (*set constructor*) ,menunjukkan elemen elemen yang digunakan di dalam ungkapan set. Set construcot terdiri dari satu atau lebih elemen jangkauan dari elemen yang dipisahkan dengan koma dan diletakkan diantara kurung bracket ('[' dan ']').

Contoh :

[2,3,5,7,11] : nilai integer 2,3,5,7, dan 11

```
type
karakter = set of 'a'..'z'
var
hurufku : karakter;
```

Record

Sebuah record berisi beberapa komponen atau fields yang dapat berbeda tipenya. Setiap fields adalah list identifier dipisah koma, diikuti oleh simbol titik dua dan sebuah tipe. Deklarasi tipe data record diawali dengan kata cadangan record dan diikuti oleh suatu daftar field dan diakhiri dengan kata cadangan End.

Sintak : type
 Record
 Daftar field
 End

Contoh

```
Type
Mhs = Record
      Npm : string[8];
      Nama : string[30];
      Nilai : byte
End;
Var
RecordMhs : Mhs;
```

Tiap komponen field dari record dapat dipergunakan dengan cara menuliskan:

```
Pengenal-record.pengenal-field[.pengenal-field]  
RecordMhs.Npm:='51295801';
```

RecordMhs adalah pengenal-field dan Npm adalah pengenal-field.

Statemen With

Statemen with menyebutkan pengenal record, sehingga penggunaan field-field selanjutnya tidak perlu menyebutkan pengenal record.

Contoh :

```
With recordMhs do  
  Begin  
    Npm      := '51295801'  
    Nama     := 'Ipunk';  
    Nilai    := '90';  
  End;
```