BAB 2

ARRAY, RECORD DAN *OBJECT ORIENTED PROGRAMMING* (OOP)

TUJUAN PRAKTIKUM

1. Praktikan mempelajari penggunaan record, teknik *Object Oriented Programming* (OOP)
2. Praktikan mengetahui type struktur data tradisional yaitu tipe data array yang selanjutnya akan digantikan dengan type data pointer.
3. Praktikan mengetahui perbedaan antara record dan unit.

TEORI PENUNJANG

Di dalam Pascal, semua variabel yang akan dipakai harus sudah ditentukan tipe datanya. Dengan menentukan tipe data suatu variabel, secara otomatis kita menentukan batasan nilai variabel tersebut dan jenis operasi yang bisa dilaksanakan atas variabel tersebut.

2.1 Variabel

Variabel adalah suatu lokasi di memori yang disiapkan oleh pemrogram dan diberi nama yang khas untuk menampung suatu nilai dan atau mengambil kembali nilai tersebut.

Bentuk umum dari deklarasi tipe data adalah :

Type [nama_variabel] = [tipe data];

Keterangan:
[nama_variabel] : Nama variabel yang menyatakan tipe data
[tipe_data] : Tipe data yang berlaku dalam Turbo Pascal.

Ada berbagai macam tipe data yang dikenali Pascal. Beberapa diantaranya yaitu *Array* dan *Record*.

2.1.1 Tipe Array

Larik (Array) adalah tipe terstruktur yang mempunyai komponen dalam jumlah yang tetap dan setiap komponen mempunyai tipe data yang sama. Posisi masing-masing komponen dalam array dinyatakan dengan nomer indeks. Array mempunyai tipe indeks yang merupakan batasan nilai. Tipe indeks ini dapat berbentuk satu dimensi, dua dimensi hingga n dimensi.

Bentuk umum dari deklarasi tipe array adalah :

Type [nama_variabel] = **Array** [tipe_index] **of** tipe;

Keterangan:
[nama_variabel] : Nama variabel yang akan dideklarasikan.
[tipe_index] : Tipe data untuk nomor index.
Tipe : Tipe Data dari komponen Array.

Contoh Pendeklarasian :

Type Matrik = **Array[1..3,1..3] of Byte;**

2.1.2 Tipe Record

Sama halnya dengan Array, *Record* merupakan kumpulan dari data yang terstruktur. Perbedaannya adalah bahwa dalam array semua elemennya harus bertipe sama, sedangkan pada *record* setiap elemen bisa mempunyai tipe data yang berbeda satu sama lainnya.

Dalam penerapannya *record* lebih banyak digunakan selain itu penggunaan *record* juga dapat digabungkan/dikombinasikan dengan penggunaan array.

Bentuk umum dari deklarasi tipe *Record* adalah :

```
Type [nama_variabel] = Record
                       [field1] : tipe1;
                       [field2] : tipe2;
                       ……
                       [fieldN] : tipeN;
                       End;
```

Keterangan:

[nama_variabel] : Nama variabel yang akan dideklarasikan.

[field1]...[fieldN] : Nama *Field* yang akan digunakan.

[tipe1]...[tipeN] : Sembarang Tipe Data untuk tiap *field*.

Contoh Pendeklarasian :

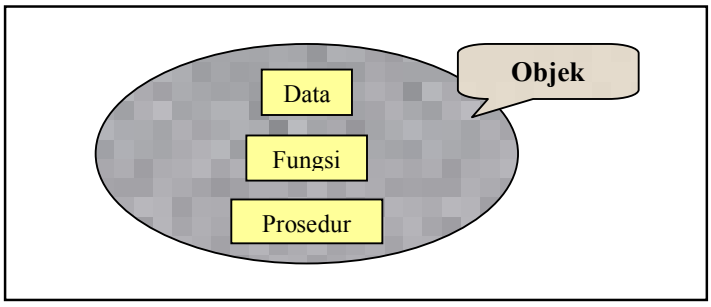
```
Type  Mahasiswa = Record
                    Nama : String[25];
                    NPM : String[8];
                    Nilai : 0..100;
                  End;
```

2.1.3 Object Oriented Programming (OOP)

Selain tipe data *Array* dan *Record*, Pascal juga telah melengkapi dirinya dengan suatu tipe data baru yang penggunaannya semakin banyak di pakai akhir-akhir ini. Semenjak Turbo Pascal 5.5 dan Turbo Pascal 6.0, Pascal telah menambahkan Teknik OOP di dalam struktur programnya.

OOP adalah pemrograman yang berorientasi pada objek. Ide dasar pada **pemrograman berorientasi objek (OOP)** adalah mengkombinasikan data dan fungsi menjadi satu kesatuan unit.

Unit ini dikenal dengan nama **objek** (*object*). Secara garis besar, objek meliputi data (variabel, konstanta, dan lain-lain.), prosedur, fungsi dan hubungannya satu dengan yang lain, yang pada akhirnya menghasilkan elemen tipe baru. Dengan kata lain objek adalah pembentukan struktur data baru.



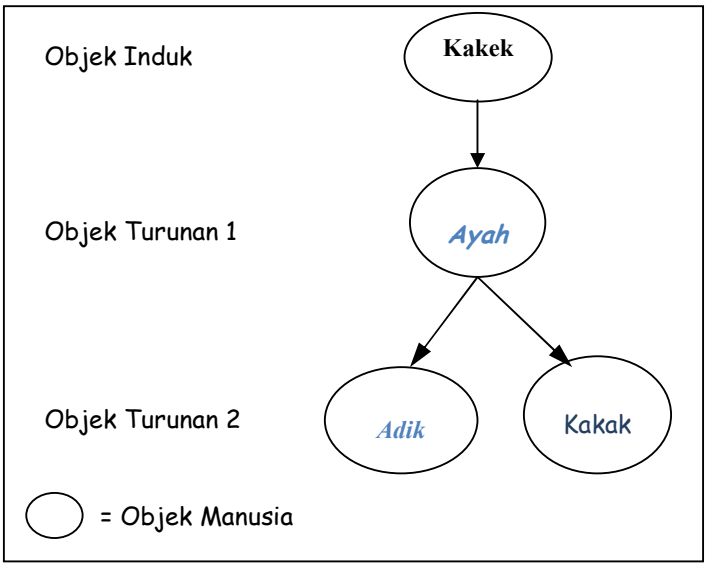
Gambar 2.1 Gambaran umum sebuah objek

2.1.3.1 Karakteristik OOP

Setidaknya ada 3 (tiga) karakteristik/prinsip dasar yang dikenal di dalam pemrograman yang berorientasi objek. Ketiga prinsip itu adalah:

1. ***Encapsulation*** **(pengkapsulan)**, adalah langkah dimana dilakukan pengkombinasian data dengan berbagai prosedur dan fungsi yang berhubungan dengannya. Berbagai prosedur dan fungsi yang dikaitkan dengan datanya itu dikenal dengan istilah ***Method*** **(metode)**. Hasil dari pengkombinasian yang dilakukan inilah yang disebut *Object* (objek), yang merupakan tipe data baru. Contoh dari *encapsulation* dapat dilihat dari pendefinisian objek mahasiswa yang dimana didalamnya terdapat beberapa atribut / *field* seperti nama, npm, alamat dan mungkin *field-field* yang lainnya. Dengan menggunakan OOP cara pandang terhadap *field-field* tersebut tidak hanya dilihat dari variabel yang digunakan melainkan melihat kepada keseluruhan objek mahasiswa yang sedang dipakai.

2. ***Inheritance*** **(penurunan sifat)**, merupakan ciri khas pemrograman OOP, yang tidak terdapat pada pemrograman terstruktur gaya lama. Langkah ini bertujuan membentuk objek baru yang memiliki sifat sama atau mirip dengan objek yang sudah ada sebelumnya. Dalam hal ini dikenal adanya **objek induk (*ansestor*)** dan **objek anak (*descendant*)**. Sifat objek induk diturunkan pada objek anaknya, dan pada gilirannya, masing-masing objek anak dapat diturunkan lagi menjadi objek anak lainnya, demikian seterusnya. Setiap terjadi perubahan pada objek induknya, perubahan itu akan dibawa terus sampai objek anaknya. Susunan objek induk dengan objek anaknya disebut sebagai Hirarki Objek. Gambar 1.2 memberikan gambaran yang jelas mengenai konsep *inheritance* (penurunan sifat) di lihat dari objek manusia.



Gambar 2.2 Konsep Dasar *Inheritance*

3. ***Polymorphism***, merupakan suatu konsep yang menyatakan sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda. Istilah ini sendiri berasal dari Bahasa Yunani, yang berarti "mempunyai banyak bentuk". Contoh penerapan *polymorphism* pada suatu program ialah pada fungsi-fungsi standar yang disediakan oleh *compiler* seperti contohnya untuk melakukan operasi penambahan satu terhadap suatu variabel (*increment*) digunakan fungsi **inc(),** contoh lainnya seperti penggunaan fungsi-fungsi untuk aritmatika seperti **sin()**, **cos()**, dan masih banyak yang lainnya.

Tiga karakteristik di atas mendukung *reusability*, yang merupakan salah satu faktor dari kualitas perangkat lunak. ***Reusability*** adalah suatu sifat yang memperkenankan suatu objek yang telah dibuat untuk dapat didistribusikan ke pemrogram lain untuk dipakai pada program-program mereka. Mereka dapat dengan mudah memodifikasi atau bahkan menambahkan objek tersebut dengan sesuatu yang baru melalui konsep *inheritance*.

2.1.3.2 OOP Dalam Turbo Pascal 7.1

Setiap bahasa yang mendukung penuh penggunaan OOP di dalam pembuatan programnya memiliki pendeklarasian yang berbeda-beda. Struktur program yang menggunakan teknik OOP di dalam Pascal dapat dituliskan sebagai berikut :

```
Type
 { Buat Sebuah Objek Baru }
 < nama_objek >        =  Object ( objek_induk )
                             < variabel >;
                             < fungsi / prosedur >;
                             < fungsi / prosedur >; Virtual;
                          End;
Var
 { Deklarasikan Objek yang telah di buat }
 < nama_variabel >  : < nama_objek >;
```

Untuk membuat suatu objek di dalam Pascal kita harus melakukannya di dalam blok *type* dengan menggunakan statemen *Object*, ini karena setiap objek yang tercipta pada pascal, merupakan suatu pembentukan tipe data baru. Statemen Virtual diatas sifatnya *optional* yang maksudnya digunakan untuk suatu fungsi/prosedur yang tidak nyata (maya).

Jika kita ingin menyatakan objek sebagai objek turunan maka setelah kata kunci *object* kita beri tanda kurung diikuti dengan objek induknya. Setelah objek dibuat, maka pada bagian **Var** objek tersebut dapat kita definisikan kepada variabel lain. Baru pada program utama dapat kita gunakan objek yang telah kita buat tersebut.

Contoh Pendeklarasian :

```
Type
     Kakek   =  Object
                   Nama, Sakit : String;
                   Procedure Penyakit;
                End;
     Bapak   =  Object(Kakek)
                End;
     Anak    =  Object(Bapak)
                End;
```

Contoh pendeklarasian di atas menjelaskan tentang pembentukan tiga buah objek yang dua diantaranya merupakan objek turunan dari objek-objek sebelumnya. Objek Kakek memiliki dua buah variabel yang bertipe string dan satu buah procedure penyakit. Objek Bapak meskipun tidak memiliki variabel ataupun procedure sendiri, telah diwarisi variabel-variabel dan prosedur oleh objek Kakek. Begitu pula dengan objek Anak, objek ini telah diwarisi variabel-variabel dan prosedur oleh objek Bapak yang sebelumnya juga telah diwarisi oleh objek Kakek.

2.1.3.3 Array of Record dan Array of Object

Sebuah *Record* atau *Object* dapat di kombinasikan dengan array. Array ini selanjutnya menjadikan sebuah *Object* atau *Record* terdiri dari beberapa komponen yang sama. Hal ini biasanya digunakan untuk menciptakan data *record* atau objek yang bertipe sama.

Pendeklarasian *Array of Record* atau *Array of Object* ini dapat dilakukan pada bagian *Type* ataupun *Var*.

Contoh Pendeklarasian :

```
Var
    {Mahasiswa merupakan Type data Record}
    DataMhs     =  Array[1..1000] of  Mahasiswa;
    {Anak merupakan Type data Object}
    DataAnak    =  Array[1..1000] of  Anak;
```

Array of Record dan *Array of Object* biasanya digunakan jika kita akan menampung data bertipe sama pada suatu variabel.

LAPORAN PENDAHULUAN

1. Jelaskan pengertian Array.
2. Jelaskan pengertian Record.
3. Apa yang dimaksud dengan:
    a. Object
    b. Method
    c. Encapsulation
    d. Polymorphisme
    e. Inheritance
4. Bagaimana cara pendefinisian Array, Record & OOP dalam Pascal serta perbedaan antara Record dan OOP.
5. Buatlah contoh penggunaan Array, Record & OOP dalam sebuah program.
6. Buatlah program sederhana dengan memanfaatkan salah satu jenis tipe data di atas.

LAPORAN AKHIR

1. Jelaskan apa yang dimaksud dengan Array?
2. Sebutkan dan jelaskan jenis-jenis Array.
3. Jelaskan apa yang dimaksud dengan Record?
4. Jelaskan apa yang dimaksud dengan OOP?
5. Buatlah masing-masing algoritma dari Array, Record dan OOP.
6. Buatlah suatu aplikasi sederhana menggunakan Array
7. Buatlah suatu aplikasi sederhana menggunakan Record
8. Buatlah suatu aplikasi sederhana menggunakan OOP pada Pascal dengan tambahan menggunakan Unit.