

## BAB II

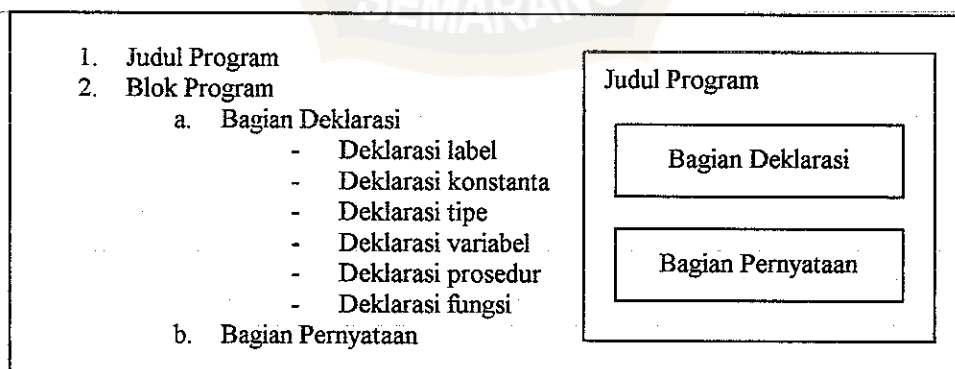
### LANDASAN TEORI

#### 2.1. Bahasa Pemrograman Turbo Pascal

Pascal adalah termasuk bahasa pemrograman tingkat tinggi (*high level language*) yang dirancang oleh Niklaus Wirth dari *Technical University* di Zurich, Switzerland. Nama Pascal diambil dari seorang matematikawan dan filosof terkenal pada abad 17 dari Perancis, Blaise Pascal.

##### 2.1.1. Struktur Program Pascal

Bahasa Pascal memiliki struktur yang diawali dengan judul program (*program heading*), dan secara ringkasnya dapat dilihat pada gambar 2.3.1.



Gambar 2.3.1. Struktur Program Pascal

##### 2.1.2. Tipe Data

Sebuah pengenalan suatu data yang akan dipergunakan di program

Pascal harus disebutkan tipenya. Bentuk umum penulisan deklarasi tipe :

Type *pengenal* = *tipe*;

dengan *pengenal* : nama pengenal yang menyatakan tipe data.

*tipe* : tipe data.

Pascal menyediakan beberapa macam tipe dari data, terdiri dari :

1. Tipe data sederhana.
  - a. Tipe data standar (integer, real, char, string, boolean).
  - b. Tipe data didefinisikan pemakai (tipe skalar, subrange).
2. Tipe data terstruktur (array, record, file, set).
3. Tipe data penunjuk (*Pointer*).

### 2.1.3. Konstanta

Pemrograman Turbo Pascal mengenal adanya deklarasi konstanta yang merupakan sebuah pengenal berisi nilai-nilai yang tetap sesuai dengan masukan atau *input* yang ditentukan. Diawali dengan kata cadangan (*reserve word*) *Const* dan diikuti oleh kumpulan pengenal yang diberi suatu nilai konstanta. Berikut ini adalah bentuk umum penulisannya dalam Turbo Pascal :

Const *pengenal* { : tipe data } = *nilai*;

dengan *pengenal* : nama konstanta.

*nilai* : nilai konstanta.

### 2.1.4. Variabel

Variabel adalah sebuah pengenal yang dapat berubah-ubah nilainya sesuai dengan jenis atau tipe data yang telah ditentukan. Kata cadangan *Var* digunakan sebagai judul dalam bagian deklarasi variabel ini, dan

diikuti oleh satu atau lebih pengenal yang dipisahkan dengan tanda koma (,), diikuti titik dua (:), dan tipe dari datanya serta diakhiri dengan titik koma. Bentuk umum penulisannya dalam Turbo Pascal :

*Var pengenal* : tipe data;

dengan *pengenal* : nama variabel.

### 2.1.5. Ungkapan

Ungkapan disusun dari sejumlah operator dan operand. Kebanyakan operator Pascal bersifat biner, artinya bahwa setiap operator pasti digunakan oleh dua buah operand. Namun ada juga operator yang bersifat tunggal (memerlukan satu operand), dan ditulis sebelum operand, contoh : -F.

Berikut ini adalah tabel yang menyajikan urutan operasi dari ungkapan yang rumit.

Tabel 2.3.1. Urutan Operasi pada Turbo Pascal

Operator	Urutan	Kategori
@, not	Pertama (tertinggi)	Operator Tunggal
*, /, div, mod, and, shl, shr	Kedua	Operator Pengali
+, -, or, xor	Ketiga	Operator Penjumlah
=, <>, <, >, >=, <=, in	Keempat (terendah)	Operator Relasi

Aturan mengenai urutan operasi dari ungkapan pada bahasa pemrograman Pascal antara lain :

- Apabila operand yang terletak di antara dua buah operator dengan urutan yang berbeda, maka yang dikerjakan terlebih dulu sesuai dengan operator yang memiliki urutan lebih tinggi.
- Operand yang terletak di antara dua buah operator yang mempunyai urutan sama dikerjakan menurut operator yang ditulis di sebelah kirinya.
- Ungkapan yang terletak di dalam tanda kurung dikerjakan terlebih dahulu sebelum dianggap sebagai sebuah operand.

#### 2.1.6. Statement

Statement merupakan satuan terkecil dalam suatu program, dan terbagi menjadi dua kelompok, yaitu :

- a. Statement sederhana, adalah suatu statement yang tidak berisi statement lain. Contoh : statement prosedur, statement *goto*.
- b. Statement terstruktur, yaitu statement yang tersusun dari sejumlah statement lain yang tereduksi secara berurutan, secara terkendali atau secara berulang. Contohnya statement *looping*.

#### 2.1.7. Prosedur dan Fungsi

Prosedur (*Procedure*) dan fungsi (*Function*) dipakai untuk menambahkan sekumpulan statement yang seolah terpisah dari program utama, tapi sebetulnya merupakan bagian dari program utama.

a. *Procedure*

Prosedur banyak digunakan pada program yang terstruktur dengan diawali kata cadangan *Procedure*. Prosedur dipanggil dan digunakan di dalam blok program yang lainnya dengan menyebutkan judul prosedurnya. Struktur dalam prosedur sama dengan struktur program, dan dimungkinkan adanya prosedur lain di dalam suatu prosedur (disebut *nested procedure/prosedur tersarang*). Bentuk umum deklarasi prosedur :

Procedure *nama* <(daftar parameter formal)>;

dengan *nama* adalah judul prosedur. Parameter formal ada dua macam yaitu parameter variabel (diawali kata *Var*) dan parameter nilai (tidak diawali kata *Var*). Parameter variabel bersifat membawa perubahan nilai parameter dalam prosedur ke dalam program utama. Sedangkan parameter nilai memiliki sifat kebalikannya, tidak membawa segala bentuk perubahan nilai parameter.

b. *Function*

Blok fungsi hampir sama dengan blok prosedur, hanya fungsi harus dideklarasikan dengan tipenya karena fungsi berfungsi sebagai variabel. Tipe deklarasi ini menunjukkan hasil dari fungsi. Bentuk umumnya :

Function *nama* <daftar parameter formal> : tipe;

### 2.1.8. Rekursi

Rekursi (*Recursion*) adalah proses dari suatu subprogram yang memanggil dirinya sendiri. Rekursi dapat berupa fungsi ataupun prosedur.

Menuliskan fungsi atau prosedur rekursi harus mengandung suatu kondisi akhir dari proses rekursi. Kondisi ini diperlukan agar tidak terjadi proses rekursi yang tidak berujung (*indefinite*) atau berjalan tanpa henti. Pengakhiran proses rekursi dapat dilakukan dengan menggunakan statement penyeleksian kondisi, sehingga jika kondisi telah terpenuhi maka proses rekursi akan berhenti.

### 2.1.9. Grafik

BGI (*Borland Graphics Interface*) adalah piranti grafis yang disediakan oleh perangkat lunak Turbo Pascal. Fasilitas ini memungkinkan membuat obyek-obyek grafis melalui Turbo Pascal. Unit *Graph* merupakan unit standar pada Pascal yang sering digunakan untuk menggambar grafik. Untuk dapat memanfaatkan fasilitas pada unit *Graph*, diperlukan sebuah file GRAPH.TPU, dan satu atau lebih file berekstensi BGI yang bersesuaian dengan penggerak grafik yang dimiliki. Pada fraktal landscape ini, perlu ditambahkan satu unit *Fractal* dengan file FRACTAL.TPU, yang bersesuaian dengan jenis penggerak grafik yang dipakai dan berfungsi sebagai penterjemah warna yang akan digunakan.

Prosedur standar *InitGraph* merupakan sebuah prosedur yang terlebih dahulu harus disebutkan untuk memulai pembuatan grafik pada Turbo Pascal. Bentuk umum penulisan dari prosedur standar *InitGraph* adalah sebagai berikut :

```
InitGraph (var GraphDriver : integer;  
          var GraphMode   : integer;  
          DriverPath      : string);
```

GraphDriver adalah driver pada komputer yang dipakai. Misalkan digunakan Color Graphics Adapter (CGA), maka GraphDriver atau driver grafik yang harus disebutkan adalah CGA atau nilai konstanta 1.

Sedangkan GraphMode adalah mode grafik yang dipergunakan untuk driver yang bersangkutan dengan driver grafik. Bila pada GraphDriver diisi CGA atau 1, maka terdapat empat pilihan konstanta untuk GraphMode yang bersesuaian dengan driver komputer, yaitu 0 (CGAC0), 1 (CGAC1), 2 (CGAC2), 3 (CGAC3), dan 4 (CGAHi).

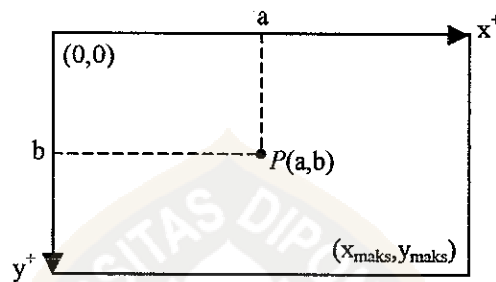
DriverPath menunjukkan direktori letak file driver grafik. Bila file driver grafik terletak pada *current directory* (direktori yang berlaku), maka DriverPath dapat ditulis dengan nilai kosong atau ''.

Untuk mengakhiri penggunaan fasilitas grafik pada Turbo Pascal dan menyebabkan kembalinya keadaan pada mode layar awal sebelum mode grafik, dipergunakan prosedur standar *CloseGraph*.

## 2.2. Sistem Koordinat Layar

Suatu grafik selalu bisa dipecah menjadi elemen yang paling kecil yang disebut *titik*. Sedangkan dalam layar penampil, elemen terkecil itu sering disebut *piksel*. Letak suatu titik atau piksel dapat dibedakan dengan titik atau piksel yang lain berdasarkan lokasi-lokasi titik/piksel tersebut. Sistem koordinat layar dua dimensi dengan sumbu x dan sumbu y positif merupakan salah satu sistem yang dapat dipakai untuk membedakan lokasi atau posisi sembarang titik atau obyek lain.

Pada sistem koordinat layar, koordinat  $(0,0)$  menunjukkan titik kiri atas layar sebagai pusat, dan koordinat  $(x_{maks}, y_{maks})$  menunjukkan titik kanan bawah layar sebagai titik maksimum. Penentuan nilai  $x_{maks}$  dan  $y_{maks}$  pada sistem koordinat ini bersesuaian dengan mode grafik yang dipakai dalam program komputer.



Gambar 2.1.1. Kedudukan Titik  $P$  Dalam Sistem Koordinat Layar

Pada sistem koordinat ini, penentuan nilai  $x$  dan  $y$  untuk kedua sumbu koordinat bisa positif dan bisa negatif, hanya saja yang dapat ditampilkan di layar untuk nilai  $x$  dan  $y$  yang positif.