

BAB II

MATERI PENUNJANG

2.1. HIMPUNAN

Definisi 2.1.1

Himpunan adalah kumpulan, daftar atau kelas objek-objek yang didefinisikan secara jelas. Objek-objek ini disebut elemen atau anggota.

Penulisan himpunan disajikan dengan membuat daftar anggota-anggotanya diantara tanda $\{$ dan $\}$ atau dengan pembentuk atau perincian himpunan garis vertikal $(|)$ yang artinya mendefinisikan suatu himpunan tertentu dengan menyatakan sifat-sifat yang harus dipenuhi oleh elemen-elemennya.

Jadi jika a menjadi anggota himpunan A maka ditulis $a \in A$, jika tidak $a \notin A$.

Himpunan berhingga adalah himpunan yang anggota-anggotanya berhingga atau dapat dihitung, sedang **himpunan tak berhingga** adalah himpunan yang anggota-anggotanya banyak sekali tak berhingga.

Dan **himpunan kosong** adalah himpunan yang tidak mengandung atau mempunyai anggota, ditulis $\{ \}$ atau ϕ . ♦

Contoh 2.1.1

Himpunan berhingga : $M = \{ x \mid x \text{ adalah hari-hari dalam seminggu} \}$

Himpunan tak berhingga : $N = \{ x \mid x \text{ adalah himpunan bilangan genap} \}$

Himpunan kosong : $B = \{ x \mid x^2 = 4, x \text{ adalah bilangan ganjil} \}$

Definisi 2.1.2

Himpunan A disebut **himpunan bagian** dari **himpunan B** jika setiap anggota A juga menjadi anggota B.

Ditulis $A \subseteq B$. ♦

Contoh 2.1.2

$$A = \{1,3,5\}$$

$$B = \{5,4,3,2,1\}$$

Maka $A \subseteq B$ karena tiap elemen 1,3,5 yang termasuk A juga termasuk B.

Definisi 2.1.3

Himpunan A sama dengan **himpunan B** jika keduanya bersama-sama memiliki anggota yang sama artinya jika setiap anggota yang termasuk A juga termasuk B atau sebaliknya.

Ditulis $A=B$, jika $A \subseteq B$ dan $B \subseteq A$. ♦

Contoh 2.1.3

$$A = \{1,2,3,4\}$$

$$B = \{3,1,4,2\}$$

Maka $A=B$ karena tiap elemen 1,2,3,4 dari A termasuk B dan tiap elemen 3,1,4,2 dari B termasuk A.

Definisi 2.1.4

Penggabungan dua himpunan A dan B adalah himpunan dari semua anggota yang termasuk dalam A atau keduanya.

Ditulis $A \cup B = \{ a \mid a \in A \text{ atau } a \in B \}$. ♦

Definisi 2.1.5

Perpotongan dua himpunan A dan B adalah himpunan dari anggota-anggota yang dimiliki bersama oleh A dan B, yaitu anggota-anggota yang termasuk A dan juga termasuk B.

Ditulis $A \cap B = \{ a \mid a \in A \text{ dan } a \in B \}$. ♦

Definisi 2.1.6

Selisih dua himpunan A dan B adalah himpunan anggota-anggota yang termasuk A tetapi tidak termasuk B.

Ditulis $A - B = \{ a \mid a \in A, a \notin B \}$. ♦

Contoh 2.1.4

$$A = \{1,2,3\}$$

$$B = \{2,3,4,5,6\}$$

$$A \cup B = \{1,2,3,4,5,6\}$$

$$A \cap B = \{2,3\}$$

$$A - B = \{1\}$$

2.2 KARAKTER,STRING

Definisi 2.2.1

Karakter adalah huruf, angka , spasi dan simbol khusus lainnya yang dapat dicetak oleh printer dari komputer. ♦

Definisi 2.2.2

String adalah barisan berhingga dari karakter. ♦

Contoh 2.2.1

A, a, 0, 9, _, +, / adalah karakter.

Ibu, suka_/, 9521 adalah suatu string atas karakter.

Definisi 2.2.3

Panjang dari suatu string w ditulis $|w|$ adalah banyaknya simbol membentuk string w.

String kosong ditulis ϵ adalah string yang tidak mengandung karakter (jumlah karakter nol), jadi $|\epsilon| = 0$.

Ditulis " (tanda petik-tunggal langsung diikuti tanda petik-tunggal, tanpa spasi). ♦

Definisi 2.2.4

Penggabungan string adalah jika ada suatu string diikuti string lainnya sehingga membentuk string baru. ♦

Contoh 2.2.2

St1 berisi Pemrograman

St2 berisi Bahasa

St3 berisi Pascal

Penggabungan St1 + St2 memberikan string Pemrograman Bahasa.

Penggabungan St1 + St3 memberikan string Pemrograman Pascal.

Penggabungan St2 + St3 memberikan string Bahasa Pascal.

Penggabungan St1 + St2 + St3 memberikan string Pemrograman Bahasa Pascal.

Definisi 2.2.5

Prefik dari suatu string adalah sejumlah karakter yang mengawali suatu string. Sedangkan **suffik** adalah sejumlah karakter yang mengakhiri suatu string.

Suatu prefik atau suffik yang merupakan string itu sendiri disebut prefik atau suffik sebenarnya. ♦

Contoh 2.2.3

abcd adalah string dengan panjang empat

Prefik : ϵ , a, ab, abc, abcd

Suffik : ϵ , d, cd, bcd, abcd

Definisi 2.2.6

Misal Σ adalah himpunan suatu karakter. Σ^* adalah himpunan yang memuat semua string dari Σ termasuk ϵ . ♦

Contoh 2.2.4

$$\Sigma = \{0,1\}$$

$$\Sigma^* = \{\epsilon, 0, 1, 00, 01, 110, 11, 000, \dots\}$$

2.3 HEURISTIK

George Polya mendefinisikan heuristik sebagai studi metode dan aturan penemuan. Dalam proses pencarian **heuristik** dinyatakan sebagai aturan untuk melakukan pemilihan cabang penyelesaian dalam ruang keadaan yang paling dapat diharapkan mencapai solusi masalah yang dapat diterima. Heuristik merupakan sebuah cara menerka langkah berikutnya yang harus diambil dalam menyelesaikan suatu masalah berdasarkan informasi yang tersedia, heuristik dapat menjadi dasar algoritma pencarian menuju solusi suboptimal atau gagal menemukan solusi apapun. ♦

Contoh 2.3.1

Ketika kita mencari prefik suatu pola.

Fungsi prefik sebuah pola adalah suatu subprogram yang dikenakan sebagai elemen pernyataan untuk menghitung panjang prefik pada pola yang merupakan suffik pada teks, setelah karakter pola cocok secara maksimal pada teks.

Jika m adalah panjang pola, q adalah panjang maksimum karakter dari string prefik suatu pola yang cocok dengan karakter dalam teks pada shift ke- s .

Dan k adalah panjang prefik suatu pola dan merupakan suffik dari q , maka heuristiknya sebagai berikut :

1. Gunakan pendefinisian awal $\pi[1]=0$ dan $k=0$.
2. Masukkan teks(T) dengan panjang n dan pola(P) dengan panjang m dengan syarat $n \leq m$.
3. Lakukan looping :

Dimulai $q=2$, jika $k > 0$ dan $P[k+1] \neq P[q]$ maka $k = \pi[k]$. Dan jika $P[k+1] = P[q]$ maka $\pi[q] = k$ dan $k = k+1$, looping dihentikan jika $q = m$.

Misal :

T :

b	a	a	b	a	b	a	b	a	b	a	c	a
---	---	---	---	---	---	---	---	---	---	---	---	---

P :

a	b	a	b	a	b	a	b	c
---	---	---	---	---	---	---	---	---

- ◆ Pada shift ke-0 karakter pada prefik suatu pola tidak ada yang cocok dengan teks.

T :

b	a	a	b	a	b	a	b	a	b	a	c	a
---	---	---	---	---	---	---	---	---	---	---	---	---

P :

a	b	a	b	a	b	a	b	c
---	---	---	---	---	---	---	---	---

$s=1$ →

- ◆ Pada shift ke-1 karakter pada prefik suatu pola cocok 1 karakter dengan teks.

T :

b	a	a	b	a	b	a	b	a	b	a	c	a
---	---	---	---	---	---	---	---	---	---	---	---	---

P :

a	b	a	b	a	b	a	b	c
---	---	---	---	---	---	---	---	---

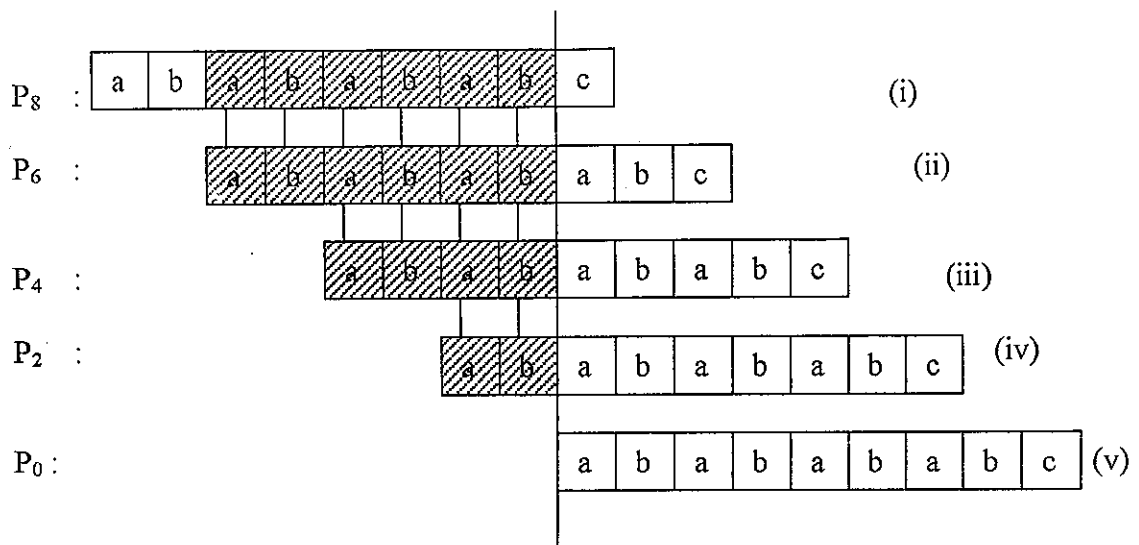
$s=2$ →

- ◆ Pada shift ke-2 karakter pada prefik suatu pola cocok 8 karakter dengan teks.

Dari penambahan shift di atas dapat dilihat bahwa pada shift ke-2 panjang prefik suatu pola yang cocok dengan teks adalah maksimum sehingga $q=8$.

Dari q di atas kita dapat menghitung fungsi prefiknya :

Panjang pola $m=9$, 8 karakter merupakan q .



Gambar 2.1.1

Keterangan gambar 2.1.1 :

Fungsi π dari sebuah pola ($\pi[8]=6, \pi[6]=4, \pi[4]=2, \pi[2]=0$)

- ◆ Dari gambar di atas, karakter prefik suatu pola yang cocok dengan teks adalah pada posisi 1-8, ini yang disebut dengan q(i)
- ◆ Kemudian dari q dibandingkan dengan dirinya sendiri diperoleh karakter 'ababab' posisi 1-6 sehingga $\pi[8]=6$(ii).
- ◆ Dari ababab dibandingkan kembali, diperoleh karakter 'abab' posisi 1-4 sehingga $\pi[6]=4$(iii).
- ◆ Dari abab dibandingkan kembali, diperoleh 'ab' dengan posisi 1-2 sehingga $\pi[4]=2$(iv).

- ◆ Dari ab dibandingkan kembali, disini tidak diperoleh karakter prefik suatu pola yang cocok sehingga $\pi [2]=0\dots\dots(v)$.

Dari uraian maka diperoleh $q=8, k=6,4,2,0$.

2.4 DASAR-DASAR ALGORITMA

Algoritma berasal dari kata algoris dan ritmis, yang pertama kali diungkapkan oleh Abu Ja'far Muhammad Ibnu Musa Al Khowarizmi (825 M) dalam bukunya yang berjudul Al-jabr Wa-al Muqabla.

Di dalam bidang pemrograman, **algoritma** didefinisikan sebagai metode khusus yang tepat dan terdiri dari serangkaian langkah yang terstruktur dan dituliskan secara sistematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer.

Hubungan antara masalah, algoritma, dan solusi dapat digambarkan sebagai berikut :



Gambar 2.1.1. Diagram hubungan masalah, algoritma, dan solusi

Proses dari masalah hingga terbentuk suatu algoritma disebut tahap pemecahan masalah, sedangkan tahap dari algoritma hingga terbentuk suatu solusi disebut dengan tahap implementasi. Solusi yang dimaksud adalah suatu program yang merupakan implementasi dari algoritma yang disusun.

Algoritma yang baik memenuhi beberapa kriteria sebagai berikut :

1. Adanya output

Dalam menyelesaikan suatu permasalahan, algoritma harus memiliki output yang merupakan solusi dari masalah yang sedang diselesaikan.

2. Efektifitas dan Efisien

Suatu algoritma dikatakan efektif jika algoritma tersebut dapat menyelesaikan suatu solusi yang sesuai dengan masalah yang diselesaikan.

Algoritma yang efisien adalah algoritma yang waktu prosesnya lebih singkat dan penggunaan memorinya relatif lebih sedikit.

3. Jumlah langkahnya berhingga

Barisan instruksi yang dibuat dalam suatu urutan tertentu, dimaksudkan agar masalah yang dihadapi dapat diselesaikan. Banyaknya instruksi atau langkah-langkah harus berhingga.

4. Berakhir

Proses didalam mencari penyelesaian suatu masalah harus berhenti atau berakhir. Hasil akhir yang didapat merupakan solusinya atau informasi tidak ditemukannya solusi.

5. Terstruktur

Urutan dari barisan langkah-langkah yang digunakan harus disusun sedemikian rupa agar proses penyelesaiannya tidak berbelit-belit, sehingga memungkinkan waktu prosesnya akan menjadi relatif lebih singkat.

Cara menyatakan algoritma :

1. Dengan bahasa tingkat rendah (natural language)
Yaitu algoritma yang ditulis dengan bahasa ungkapan sehari-hari.
2. Dengan sandi semu (pseudocode)
Yaitu algoritma yang ditulis dengan perintah bahasa pemrograman tertentu ditambah dengan bahasa alamiah.
3. Dengan bagan alir (flowchart)
Yaitu algoritma ditulis dalam bentuk simbol atau diagram.

Contoh 2.4.1

Program untuk menyusun tabel suhu dalam skala Reamur dan Fahrenheit untuk nilai-nilai Celcius mulai dari 5° sampai 100° dengan kenaikan 5°.

1. Dengan bahasa alamiah
Langkah 1 : Mula-mula masukkan nilai $C = 0$
Langkah 2 : Kemudian $C = C + 5$ yang berarti bahwa C yang baru adalah C yang lama ditambah 5, sehingga C yang baru bernilai 5.
Langkah 3 : (Menghitung nilai R dan F)
$$R = C / 5 * 4$$
$$F = C / 5 * 9 + 32$$

Langkah 4 : Mencetak hasil dari R , F dan C yang telah dihitung.
Langkah 5 : Kemudian dicek apakah $C < 100$, jika 'ya' maka kembali ke langkah 2.
Jika 'tidak' maka program selesai.

2. Dengan sandi semu

Langkah 1 : Mula-mula masukkan nilai $C = 0$

Langkah 2 : Kemudian $C = C + 5$ yang berarti bahwa C yang baru adalah

C yang lama ditambah 5, sehingga C yang baru bernilai 5.

Langkah 3 : (Menghitung nilai R dan F)

$$R = C / 5 * 4$$

$$F = C / 5 * 9 + 32$$

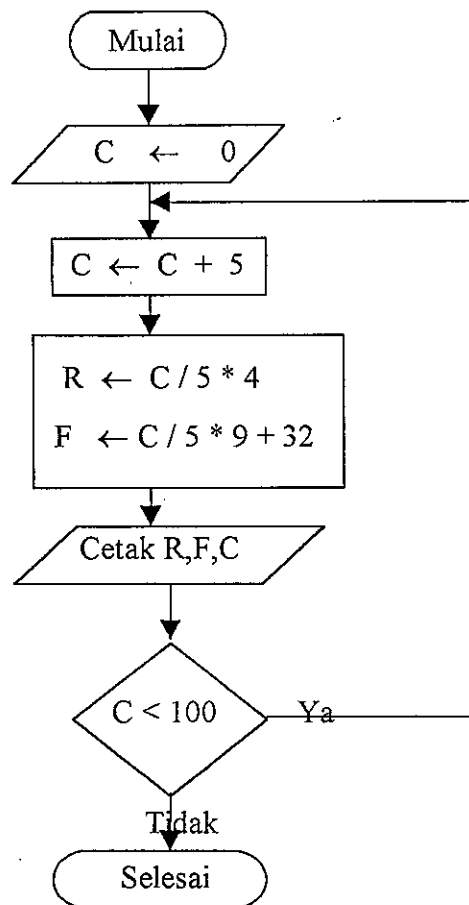
Langkah 4 : Mencetak hasil dari R , F dan C yang telah dihitung.

Langkah 5 : Lakukan test kondisi :

IF $C < 100$ THEN $C = C + 5$

ELSE program selesai

3. Dengan bagan alir



2.5 BAHASA PEMROGRAMAN PASCAL

Sebelum kita membahas bahasa pemrograman Pascal terlebih dahulu akan dibahas pengertian dari program.

Program adalah kata, ekspresi, pernyataan yang disusun menjadi satu kesatuan algoritma berupa urutan langkah untuk menyelesaikan masalah yang merupakan implementasi dari struktur data dengan menggunakan bahasa pemrograman sehingga dapat dieksekusi oleh komputer.

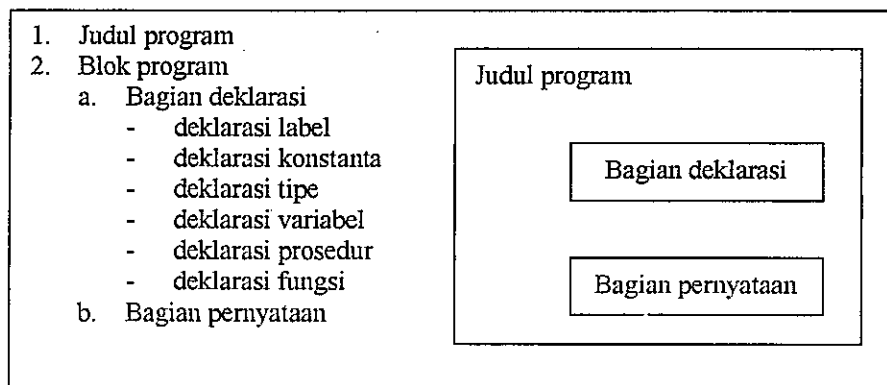
Pascal merupakan bahasa tingkat tinggi yang orientasinya pada segala tujuan, dirancang oleh Niklaus Wirth dan Technical University Switzerland. Turbo Pascal adalah salah satu versi dari Pascal selain UCSD Pascal, MS Pascal, Apple Pascal dan lain sebagainya.

Hal-hal yang perlu diketahui dalam Turbo Pascal antara lain :

2.5.1 Struktur Program Pascal

Secara ringkas, struktur program Pascal dapat dilihat dalam gambar 2.1.2

Gambar 2.1.2 Struktur program Pascal



2.5.2 Tipe Data

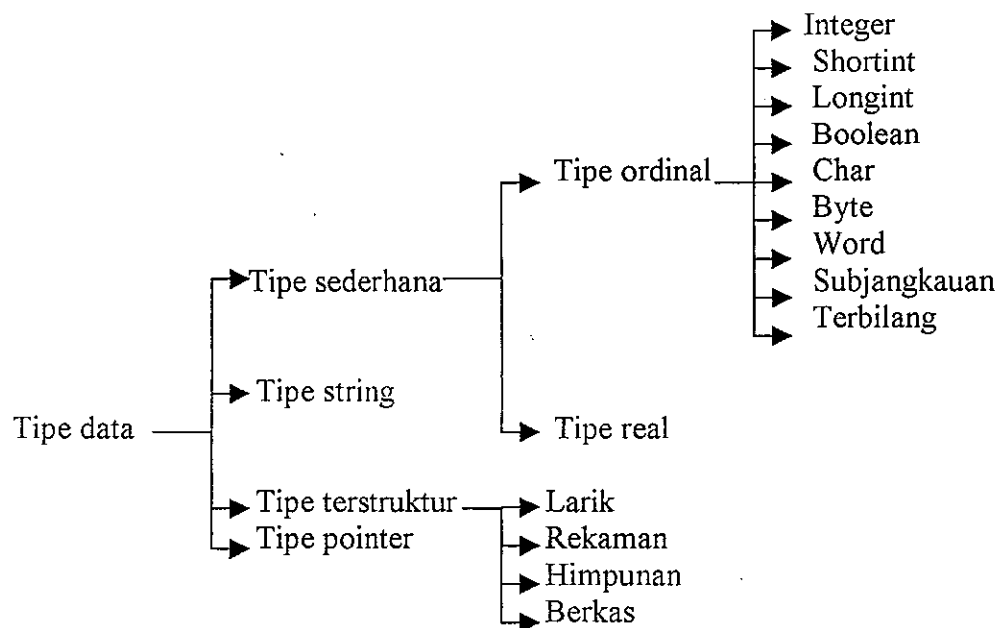
Dalam Pascal semua variabel yang dipakai harus sudah ditentukan tipe datanya. Penentuan tipe data suatu variabel berarti juga penentuan batasan nilai variabel tersebut dan jenis operasi yang bisa dikerjakan atas variabel tersebut.

Bentuk umum dari deklarasi tipe data adalah :

type pengenal = tipe;

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

tipe : tipe data yang berlaku dalam Turbo Pascal.



Gambar 2.1.3. Tipe data dalam Turbo Pascal 7.0

2.5.3 Variabel dan Konstanta

Variabel mewakili suatu nilai data tertentu yang dioperasikan dalam program. Setiap variabel harus dinyatakan tipe datanya.

Bentuk umum deklarasi variabel adalah:

var pengenal = tipe_data ;

dengan *pengenal* : nama variabel yang dideklarasikan

tipe- data : tipe data yang digunakan.

Deklarasi **konstanta** menunjukkan nilai yang tetap dari suatu pengenal dan berlaku pada blok dimana deklarasi tersebut dinyatakan.

Bentuk umum deklarasi konstanta adalah :

const pengenal : *tipe = konstanta*;

dengan *pengenal* : nama konstanta

tipe : tipe data konstanta

konstanta : nilai konstanta

2.5.4 Ungkapan

Ungkapan disusun dari sejumlah operator dan operand. Kebanyakan operator Pascal bersifat biner yang artinya bahwa setiap operator pasti digunakan oleh dua buah operand.

Namun ada juga operator yang bersifat tunggal (*unary*), yaitu yang hanya memerlukan sebuah operand. Operator tunggal selalu ditulis sebelum operand, misalnya $-B$.

Ungkapan yang rumit perlu ditentukan urutan operasinya, sebagaimana tersaji dalam tabel 2.1.4

tabel 2.1.4 Urutan Operasi

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

Ada tiga aturan mengenai urutan operasi :

1. Operand yang terletak diantara dua buah operator yang mempunyai urutan berbeda yang dikerjakan terlebih dahulu sesuai dengan operator yang mempunyai urutan yang lebih tinggi.
2. Operand yang terletak diantara dua buah operator yang mempunyai urutan sama dikerjakan menurut operator yang ditulis disebelah kirinya
3. Ungkapan yang terletak didalam tanda kurung dikerjakan terlebih dahulu sebelum dianggap sebagai sebuah operand.

2.5.5 Statemen

Statemen merupakan satuan terkecil suatu program. Statemen terbagi menjadi dua kelompok, yaitu statemen sederhana dan statemen terstruktur.

Statemen sederhana adalah statemen yang tidak berisi statemen yang lain, terdiri dari statemen pemberian (*assignment statement*), statemen prosedur dan statemen goto.

Statemen terstruktur adalah statemen yang tersusun dari sejumlah statemen lain yang dieksekusi secara berurutan (statemen majemuk dan statemen with), secara terkendali (statemen kendali) atau secara berulang (statemen berulang).

2.5.6 Prosedur dan Fungsi

Prosedur adalah subprogram yang dapat dipanggil dalam program ataupun dalam subprogram lain dan berdiri sendiri sebagai pernyataan. Dan **fungsi** adalah subprogram yang dikenakan sebagai elemen dalam pernyataan penugasan atau suatu ungkapan dan fungsi tidak dapat berdiri sendiri sebagai pernyataan.

Prosedur dan fungsi digunakan untuk menambahkan sekelompok statemen yang seolah-olah terpisah dari program utama, tetapi sesungguhnya merupakan bagian dari program utama.

Prosedur diaktifkan menggunakan statemen prosedur dan fungsi diaktifkan dengan suatu ungkapan yang hasilnya dikembalikan lagi sebagai nilai baru dari ungkapan tersebut.

Prosedur mempunyai struktur yang sama dengan struktur program, dan di dalam prosedur juga dimungkinkan ada prosedur lain yang strukturnya sama. Bentuk ini disebut prosedur tersarang (nested procedure).

Semua deklarasi dalam prosedur (label, konstanta, tipe data, variabel) dikatakan sebagai **deklarasi lokal** jika hanya digunakan dalam prosedur itu saja dan tidak dikenal diluar prosedur. Sedangkan deklarasi dalam program utama bersifat **global** jika digunakan dalam bagian program manapun.

Bentuk umum deklarasi prosedur adalah :

procedure nama <(dafpar)>;

dengan nama : nama prosedur

dafpar : daftar parameter formal

Bentuk umum deklarasi fungsi adalah:

function nama <(dafpar)> : tipe;

dengan nama : nama fungsi

dafpar : daftar parameter formal

tipe : tipe data dari fungsi tersebut.

Deklarasi prosedur dipecah menjadi dua bagian, yaitu nama prosedur dan parameter formal. Parameter formal ada dua macam, yaitu parameter nilai dan parameter variabel.

Parameter nilai adalah parameter yang nilainya tidak dimaksudkan untuk diubah sekluarnya dari subprogram dan **parameter variabel** adalah parameter yang dimaksudkan untuk diubah nilainya ketika keluar dari subprogram.

Secara umum fungsi hampir sama dengan prosedur, dengan sedikit perbedaan bahwa **nama fungsi** sekaligus berfungsi sebagai suatu variabel sehingga dalam deklarasi fungsi harus dinyatakan tipe datanya.

2.6 POLA, TEKS

Pola (P) dan teks (T) adalah penggabungan beberapa string, dengan panjang pola (m) lebih kecil atau sama dengan panjang teks (n).

s adalah pemindahan pola sepanjang teks atau disebut dengan shift, sedangkan s' adalah pemindahan setelah s berjalan.

Ketika kita mempunyai pola, kemudian diberikan suatu teks, maka pencocokan dapat dilakukan dengan mencocokkan pola tersebut pada teks.

q adalah panjang maksimum karakter pola yang berkorespondensi untuk dicocokkan dengan teks sejumlah shift s dan k adalah panjang prefik pola yang merupakan suffik pada teks dan **fungsi prefik** adalah perbandingan karakter pola yang cocok dengan teks (q) dengan dirinya sendiri sejumlah shift s' . Fungsi prefik dapat direpresentasikan dalam array $\pi(q)$.