

BAB II

TEORI PENUNJANG

2.1 Persamaan-persamaan Umum Perpindahan Kalor

Perpindahan kalor adalah suatu ilmu untuk meramalkan perpindahan energi yang terjadi karena adanya perbedaan suhu diantara benda. Perpindahan kalor yang terjadi dapat dibedakan menjadi tiga macam yaitu : konduksi, konveksi dan radiasi.

2.1.1 Persamaan Perpindahan Kalor Konduksi

Hukum dasar yang menyatakan hubungan antara laju panas (heat flow) dan gradien temperatur (temperature gradient) didasarkan atas observasi ilmuan matematika Perancis yang bernama Joseph Fourier. Eksperimen ini dilakukan pada benda padat yang homogen dan isotropic (material/ bahan) mempunyai konduktivitas termal yang sama dan tidak bergantung arahnya. Dari eksperimen ini lahirlah hukum Fourier untuk perpindahan panas konduksi sebagai berikut :

$$q(r,t) = -k \nabla T(r,t) \frac{W}{m^2} \quad \dots (2.1.1)$$

dimana : $q(r,t)$ = Heat Flux Vector

= Laju panas per unit waktu per unit luas dari permukaan
isothermal dalam arah penurunan temperatur (W/m^2)

k = Konduktivitas termal material ($\text{W/m}^\circ\text{C}$)

$$\nabla = \left(\frac{\partial}{\partial x} + \frac{\partial}{\partial y} + \frac{\partial}{\partial z} \right)$$

= Operator vektorial tiga dimensi

Dalam koordinat rectangular, persamaan dapat ditulis sbb:

$$q(x, y, z, t) = -\hat{i}k \frac{\partial T}{\partial x} - \hat{j}k \frac{\partial T}{\partial y} - \hat{k}k \frac{\partial T}{\partial z}$$

dimana : $\hat{i}, \hat{j}, \hat{k}$ = unit vector dalam arah x,y,z

Tiga komponen fluks panas pada masing-masing arah adalah sbb:

$$q_x = -k \frac{\partial T}{\partial x} \quad q_y = -k \frac{\partial T}{\partial y} \quad q_z = -k \frac{\partial T}{\partial z}$$

$$\left[\begin{array}{l} \text{Laju - Panas - yang} \\ \text{memasuki - permukaan} \\ \text{batas - V} \end{array} \right] + \left[\begin{array}{l} \text{Laju - Energi - yang} \\ \text{dibangkitkan - dalam - V} \end{array} \right] = \left[\begin{array}{l} \text{Laju - penyimpanan} \\ \text{Energi - dalam - V} \end{array} \right] \quad \dots (2.1.2)$$

➤ Laju Panas yang memasuki permukaan batas V (Q_c)

$$Q_c = -\int_A \hat{n} q dA = -\int_V \nabla q dV \quad \dots (2.1.3)$$

dimana : A = luasan permukaan dari elemen volume V

$\hat{n}$ = vector unit normal kearah luar dA

q = vector fluks panas pada dA

➤ Laju panas yang dibangkitkan dalam V (Q_g)

$$Q_g = \int_V g(r, t) dV \quad \dots (2.1.4)$$

dimana : $g(r,t)$ = laju panas yang dibangkitkan dalam medium

= panas yang dibangkitkan persatuan waktu persatuan luas

➤ Laju panas yang disimpan dalam volume (Q_s)

$$Q_s = \int_V \rho C_p \frac{\partial T(r,t)}{\partial t} dV \quad \dots (2.1.5)$$

dimana : ρ = massa jenis bahan

C_p = panas jenis bahan

substitusi pers.(2.1.3),(2.1.4) dan (2.1.5) ke pers. (2.1.2) menghasilkan :

$$\int_V \left[-\nabla q(r,t) + g(r,t) - \rho C_p \frac{\partial T(r,t)}{\partial t} \right] dV = 0 \quad \dots (2.1.6)$$

jika elemen volume dibuat sangat kecil mendekati titik maka pers. adalah :

$$-\nabla q(r,t) + g(r,t) = \rho C_p \frac{\partial T(r,t)}{\partial t} \quad \dots (2.1.7)$$

substitusi $q(r,t)$ dari pers. (2.1.1) ke pers.(2.1.7) menghasilkan :

$$\nabla [k \nabla T(r,t)] + g(r,t) = \rho C_p \frac{\partial T(r,t)}{\partial t} \quad \dots (2.1.8)$$

untuk k dan C_p konstan pers. menjadi :

$$\nabla^2 T(r,t) + \frac{1}{k} g(r,t) = \frac{1}{\alpha} \frac{\partial T(r,t)}{\partial t} \quad \dots (2.1.9a)$$

dimana : $\alpha = \frac{k}{\rho C_p}$ = diffusitas termal

Pada medium yang konduktivitas suhunya konstan dan tidak ada laju panas yang dibangkitkan maka persamaan (2.1.9a) dapat dituliskan sbb :

$$\nabla^2 T(r,t) = \frac{1}{\alpha} \frac{\partial T(r,t)}{\partial t} \quad \dots (2.1.9b)$$

Dalam sistem koordinat rectangular (x,y,z) dari persamaan (2.1.8) ;

BAB II TEORI PENUNJANG

$$\frac{\partial}{\partial x} \left(k \frac{\partial T}{\partial x} \right) + \frac{\partial}{\partial y} \left(k \frac{\partial T}{\partial y} \right) + \frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) + g = \rho C_p \frac{\partial T}{\partial t}$$

untuk kasus perpindahan kalor steady state dua dimensi tanpa pembangkitan kalor

berlaku persamaan :

$$k \frac{\partial^2 T}{\partial x^2} + k \frac{\partial^2 T}{\partial y^2} = 0 \rightarrow \frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} = 0$$

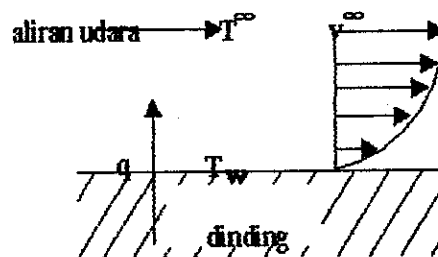
dimana : $g(r,t) = 0$ (tanpa pembangkit kalor)

$$\frac{\partial}{\partial z} \left(k \frac{\partial T}{\partial z} \right) = 0 \quad (\text{tidak ada aliran dalam arah } z)$$

$$\rho C_p \frac{\partial T}{\partial t} = 0 \quad (\text{karena steady state})$$

2.1.2 Persamaan Perpindahan Panas Konveksi

Jika suatu permukaan benda yang panas dikenai aliran udara yang lebih dingin seperti terlihat pada gambar, maka akan terjadi laju perpindahan kalor secara konveksi dari permukaan plat ke udara.



Gbr.2.1.2. Perpindahan kalor konveksi

Hukum Newton tentang pendinginan menyatakan :

$$q = h A (T_w - T_\infty)$$

dimana : q = laju perpindahan kalor konveksi

h = koefisien perpindahan kalor konveksi

A = luas permukaan konveksi

T_w = suhu permukaan benda

T_∞ = suhu arus bebas

2.1.3 Persamaan Radiasi Termal

Radiasi termal adalah suatu mekanisme perpindahan energi melalui daerah-daerah hampa. Mekanisme ini sangat berpengaruh apabila dua buah benda mempunyai beda suhu yang sangat besar. Dalam termodinamika telah ditunjukkan bahwa radiator (penyinaran) ideal atau benda hitam (black body), memancarkan energi dengan laju yang sebanding dengan pangkat empat suhu absolut benda itu dan berbanding lurus dengan luas permukaan.

$$q_{\text{pancaran}} = \sigma \cdot A \cdot T^4$$

dimana : σ = konstanta Stefan Boltzmann

$$= 5,669 \times 10^{-8} \text{ W/m}^2 \cdot \text{°K}^4$$

A = luas permukaan radiasi

T = suhu mutlak (K)

2.2 Metode Finite Difference

2.2.1 Pendekatan beda hingga dari derivatif menggunakan deret Taylors

Penyajian beda hingga dari suatu derivatif didapat dengan menurunkan fungsi

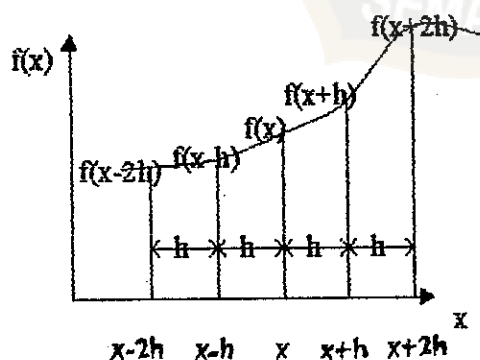
$F(x,y)$ pada $x = x_0$, $y = y_0$ terhadap x

$$\frac{\partial F}{\partial x} = \lim_{\Delta x \rightarrow 0} \frac{F(x_0 + \Delta x, y_0) - F(x_0, y_0)}{\Delta x} \quad \dots (2.2.1)$$

Ekspansi deret Taylor dari fungsi $f(x+h)$ dan $f(x-h)$ pada titik x seperti yang ditunjukkan gambar dibawah ini adalah sbb:

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!} f''(x) + \frac{h^3}{3!} f'''(x) + \dots \quad \dots (2.2.2)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2!} f''(x) - \frac{h^3}{3!} f'''(x) + \dots \quad \dots (2.2.3)$$



2.2.1.1 Turunan Pertama

Turunan orde satu maju dan mundur dari $f(x)$ dengan menyelesaikan pers.

(2.2.2) dan (2.2.3) untuk $f'(x)$:

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(x) - \frac{h^2}{6} f'''(x) + \dots \quad \text{maju} \quad \dots (2.2.4)$$

$$f'(x) = \frac{f(x) - f(x-h)}{h} + \frac{h}{2} f''(x) - \frac{h^2}{6} f'''(x) + \dots \quad \text{mundur} \dots (2.2.5)$$

substitusi persamaan (2.2.4) dan (2.2.5) untuk pendekatan central difference orde I menghasilkan :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f'''(x) - \dots \quad \text{central} \quad \dots (2.2.6)$$

ketiga persamaan terakhir dapat ditulis lagi sbb :

$$f'(x) = \frac{f(x+h) - f(x)}{h} + O(h) \quad \text{maju} \quad \dots (2.2.7a)$$

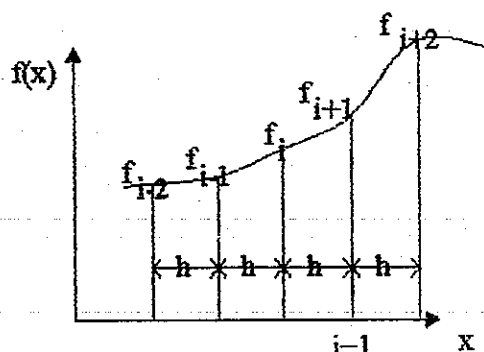
$$f'(x) = \frac{f(x) - f(x-h)}{h} + O(h) \quad \text{mundur} \quad \dots (2.2.7b)$$

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + O(h^2) \quad \text{pusat} \quad \dots (2.2.7c)$$

dimana : $O(h)$ = kesalahan order ke-h

$O(h^2)$ = kesalahan order ke- h^2

Untuk notasi dibawah ini :



BAB II TEORI PENUNJANG

Persamaan menjadi :

$$f'_i = \frac{f_{i+1} - f_i}{h} + O(h) \quad \text{maju} \quad \dots (2.2.8a)$$

$$f'_i = \frac{f_i - f_{i-1}}{h} + O(h) \quad \text{mundur} \quad \dots (2.2.8b)$$

$$f'_i = \frac{f_{i+1} - f_{i-1}}{2h} + O(h^2) \quad \text{central} \quad \dots (2.2.8c)$$

2.2.1.2 Turunan Kedua

Penyajian finite difference untuk turunan kedua $f''(x)$ dari fungsi $f(x)$ adalah dengan menentukan ekspansi deret Taylor pada fungsi $f(x+2h)$ dan $f(x-2h)$:

$$f(x+2h) = f(x) + 2hf'(x) + 2h^2 f''(x) + \frac{4}{3} h^3 f'''(x) + \dots \quad \dots (2.2.9a)$$

$$f(x-2h) = f(x) - 2hf'(x) + 2h^2 f''(x) - \frac{4}{3} h^3 f'''(x) + \dots \quad \dots (2.2.9b)$$

dari eliminasi pers.(2.2.2) dan(2.2.9a) diperoleh :

$$f''(x) = \frac{f(x) + f(x+2h) - 2f(x+h)}{h^2} - hf'''(x) \quad \dots (2.2.10a)$$

dari eliminasi pers.(2.2.3) dan(2.2.9b) diperoleh :

$$f''(x) = \frac{f(x-2h) + f(x) - 2f(x-h)}{h^2} + hf'''(x) \quad \dots (2.2.10b)$$

dari eliminasi $f'(x)$ antara pers. (2.2.2) dan(2.2.3) menghasilkan :

$$f''(x) = \frac{f(x-h) + f(x+h) - 2f(x)}{h^2} - \frac{1}{12} h^2 f'''(x) \quad \dots (2.2.10a)$$

Pers.(2.2.10a) s/d (2.2.10c) dapat ditulis lagi dalam notasi “i” sbb:

$$f''_i = \frac{f_i - 2f_{i+1} + f_{i+2}}{h^2} + O(h) \quad \text{maju} \quad \dots (2.2.11a)$$

BAB II TEORI PENUNJANG

$$f_i'' = \frac{f_{i-2} - 2f_{i-1} + f_i}{h^2} + O(h) \quad \text{mundur} \quad \dots (2.2.11b)$$

$$f_i'' = \frac{f_{i-1} - 2f_i + f_{i+1}}{h^2} + O(h^2) \quad \text{central} \quad \dots (2.2.11c)$$

dimana : $f_i'' = \left. \frac{d^2 f(x)}{dx^2} \right|$



2.3 Penyelesaian Persamaan Linier Simultan dengan Eliminasi Gauss-Seidel

2.3.1 Sistem Persamaan Linier Simultan Orde N

Bentuk umum dari suatu persamaan linier simultan orde N adalah :

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2$$

$$\dots$$

$$\dots$$

$$a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n$$

Dalam bentuk matriks, penulisan persamaan linier simultan diatas dapat disederhanakan menjadi :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \dots \\ b_n \end{bmatrix}$$

Persoalan yang sering dihadapi adalah mencari harga-harga : $x_1, x_2, \dots, x_n$ yang memenuhi persamaan linier simultan tersebut diatas. Dua hal khusus yang harus diperhatikan adalah :

1. Sistem persamaan linier simultan mempunyai penyelesaian tunggal bila matrik A adalah non singular yaitu $\text{Det } A \neq 0$
2. Sistem persamaan linier simultan mempunyai penyelesaian yang tak tentu atau tak mungkin diselesaikan bila matrik A adalah singular yaitu $\text{Det } A = 0$

2.3.2 Eliminasi Gauss-Seidel

Eliminasi Gauss-Seidel dapat digunakan untuk menyelesaikan matrik yang banyak mempunyai elemen berharga nol, atau yang elemen-elemen yang tidak bernilai nol terserak (*sparse*).

Persamaan simultan dengan N elemen yang dituliskan diatas dapat dibentuk sebagai berikut :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} & | & b_1 \\ a_{21} & a_{22} & \dots & \dots & a_{2n} & | & b_2 \\ \dots & \dots & \dots & \dots & \dots & | & \dots \\ \dots & \dots & \dots & \dots & \dots & | & \dots \\ a_{n1} & a_{n2} & \dots & \dots & a_{nn} & | & b_n \end{bmatrix}$$

Sedangkan langkah-langkah penyelesaian secara eliminasi ini adalah :

1. Jadikanlah semua koefisien untuk penyelesaian elemen x_1 pada semua persamaan kecuali pada persamaan baris 1 sama dengan nol, dengan cara semua unsure elemen baris pertama dikalikan dengan $-\frac{a_{21}}{a_{11}}$ lalu dijumlahkan dengan unsur-unsur elemen pada baris kedua. Setelah itu mengalikan semua unsur baris ke 1 tersebut dengan $-\frac{a_{31}}{a_{11}}$, kemudian menjumlahkan dengan unsure-unsur elemen baris ke 3.

Proses ini diteruskan sampai baris ke n. hasil dari proses ini adalah :

$$\begin{bmatrix} a_{11} & a_{12} & \dots & \dots & a_{1n} & | & b_1 \\ 0 & a_{22}' & \dots & \dots & a_{2n}' & | & b_2' \\ \dots & \dots & \dots & \dots & \dots & | & \dots \\ \dots & \dots & \dots & \dots & \dots & | & \dots \\ 0 & a_{n2}' & \dots & \dots & a_{nn}' & | & b_n' \end{bmatrix}$$

2. Langkah ke 2 adalah mengalikan semua koefisien elemen x_2 pada semua persamaan kecuali pada persamaan ke 2 sama dengan nol. Dengan cara mengalikan semua unsure elemen baris ke 2 dengan $-\frac{a_{32}}{a_{22}}$ lalu dijumlahkan dengan unsure-unsur baris ke 3. Berikutnya mengalikan lagi semua unsure-unsur baris ke 2 dengan $-\frac{a_{42}}{a_{22}}$ lalu dijumlahkan dengan unsure-unsur baris ke 4. Proses ini diteruskan sampai baris ke n . Hasil dari proses tersebut adalah :

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & \dots & a_{1n} & b_1 \\ 0 & a_{22}' & \dots & a_{2n}' & b_2' \\ 0 & 0 & a_{33}'' & \dots & b_3'' \\ \dots & & & & \dots \\ 0 & 0 & \dots & a_{n3}'' & a_{nn}'' & b_n'' \end{array} \right]$$

3. Langkah selanjutnya sama dengan langkah-langkah sebelumnya yaitu mengalikan semua koefisien elemen x_n berikutnya untuk semua persamaan kecuali pada persamaan baris ke n ini sama dengan nol. Sehingga hasil akhir dari proses secara keseluruhan adalah :

$$\left[\begin{array}{cccc|c} a_{11} & a_{12} & a_{13} & \dots & a_{1n} & b_1 \\ 0 & a_{22}' & a_{23}' & \dots & a_{2n}' & b_2' \\ 0 & 0 & a_{33}'' & \dots & a_{3n}'' & b_3'' \\ \dots & & & & & \dots \\ \dots & & & & & \dots \\ 0 \dots 0 & 0 & \dots & & a_{nn}'' & b_n'' \end{array} \right]$$

matrik ini dinamakan matrik U atau $[\mu]$ atau disebut Upper Right. Proses-proses ini secara keseluruhan untuk menghasilkan matrik U dapat dituliskan sebagai berikut :

$$A_{ij} = A_{ij} - \frac{A_{ik}}{A_{kk}} \cdot A_{kj}$$

BAB II TEORI PENUNJANG

Untuk :

$$k = 1, 2, 3, \dots, n-1$$

$$i = k+1, k+2, \dots, n$$

$$j = k, k+1, \dots, n+1$$

n = jumlah persamaan

4. Untuk menentukan nilai dari x_i yang dicari secara keseluruhan dapat dituliskan dengan bentuk :

$$x_i = \frac{1}{A_{ii}} \left(a_{i,n+1} - \sum_{r=i+1}^n A_{ir} \cdot x_r \right)$$

Untuk :

$$i = 1, 2, 3, \dots, n$$

$$r = i+1, \dots, n$$

n = jumlah persamaan

2.4 Algoritma

Algoritma adalah himpunan instruksi yang menjelaskan langkah-langkah yang teratur dan berhingga untuk menyelesaikan suatu masalah.

Kriteria algoritma yang baik adalah :

1. *Adanya input* : suatu algoritma sekurang-kurangnya mempunyai sejumlah satu atau lebih input yang disuplai secara eksternal.
2. *Adanya output* : input suatu algoritma harus memberikan output sekurang-kurangnya sejumlah satu input.
3. *Kepastian* : tiap instruksi dalam algoritma harus jelas dan tidak berarti ganda

BAB II TEORI PENUNJANG

1. *Keberhinggaan* : apabila instruksi dalam algoritma ditelusuri, maka algoritma berhenti setelah sejumlah berhingga langkah-langkah.
2. *Keefektifan* : tiap instruksi harus langsung mengarah ke sasaran yang dituju.

Ada beberapa cara untuk menyatakan algoritma yaitu :

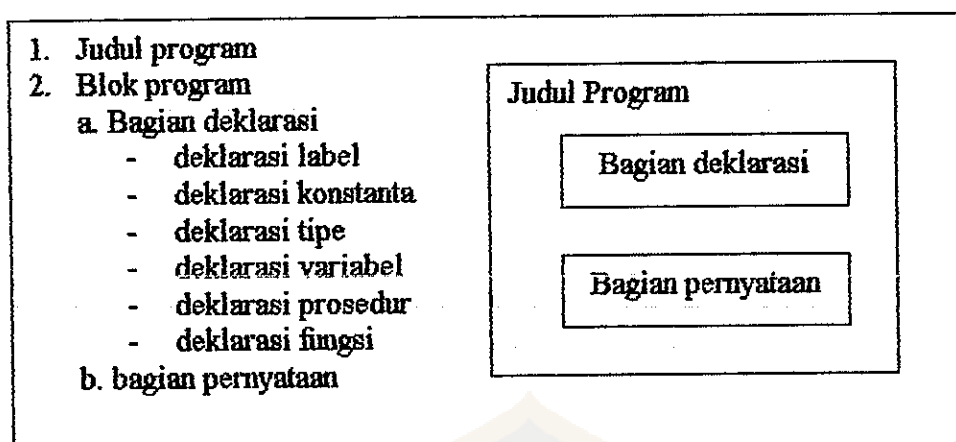
1. Dengan bahasa alamiah : algoritma ditulis dengan ungkapan sehari-hari.
2. Dengan sandi semu (Pseudocode) : algoritma ditulis dengan menggunakan perintah bahasa pemrograman tertentu ditambah dengan bahasa alamiah.
3. Dengan bagan alir (flowchart) : algoritma ditulis dalam bentuk symbol/ diagram.

2.5 Pemrograman Turbo Pascal 7.0

Pascal adalah bahasa tingkat tinggi yang orientasinya pada segala tujuan, dirancang oleh Niklaus Wirth dari 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 diantaranya :

2.5.1 Struktur Program Pascal

Secara ringkas, struktur program Pascal dapat dilihat dalam gambar 2.5.1



Gambar 2.5.1 Struktur program Pascal

2.5.2 Tipe Data

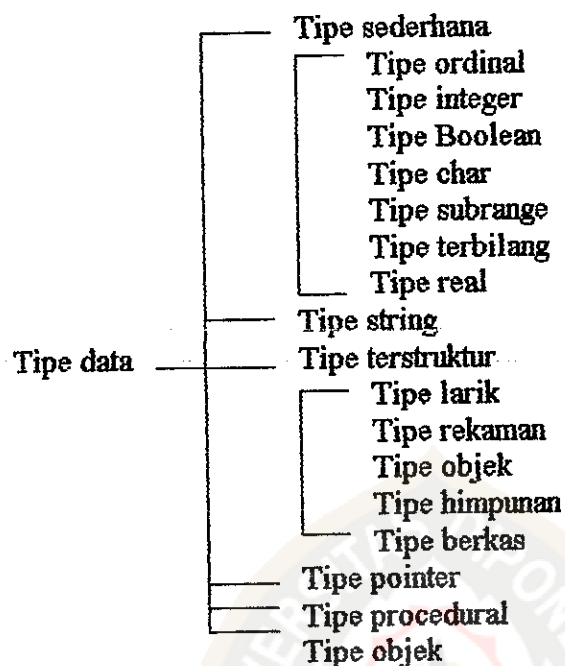
Dalam Pascal semua variabel yang dipakai harus sudah ditentukan tipe datanya. Penentuan tipe data suatu variabel, berarti juga penentuan batasan nilai variabel tersebut dan jenis operasi yang dikerjakan atas variabel tersebut.

Bentuk umum dari deklarasi tipe data adalah :

Type pengenalan = type;

Dengan *pengenal* : nama pengenalan yang menyatakan tipe data.

tipe : tipe data yang berlaku dalam Turbo Pascal.



Gambar 2.5.2 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 pengenalan : *tipe_data*;

Dengan *pengenal* : nama variabel yang dideklarasikan.

tipe_data : tipe data yang digunakan.

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

Bentuk umum deklarasi konstanta adalah :

Const pengenalan = nilai;

Dengan *pengenal* : nama konstanta.

nilai : 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 2.6.1.

Tabel 2.6.1. 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 :

- Operand yang terletak di antara dua buah operator yang mempunyai urutan berbeda dikerjakan terlebih dahulu sesuai dengan operator yang mempunyai urutan yang 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.

2.5.5. 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, secara terkendali atau secara berulang.

2.5.6 Prosedur dan Fungsi

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 bersama parameter aktual dan fungsi diaktifkan dengan suatu ungkapan yang hasilnya dikembalikan lagi sebagai nilai baru dari ungkapan tersebut.

Prosedur memiliki struktur yang sama dengan struktur program, dan di dalam prosedur juga dimungkinkan ada prosedur lain yang strukturnya sama. Bentuk ini disebut

BAB II TEORI PENUNJANG

prosedur tersarang (*nested procedure*). Semua deklarasi dalam prosedur merupakan deklarasi lokal.

Bentuk umum deklarasi prosedur adalah :

Procedure *nama* <(*dafpar*)>;

Dengan *nama* : nama prosedur.

dafpar : nama parameter formal.

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 tidak diawali dengan kata baku *var* dan parameter variabel adalah parameter yang diawali dengan *var*.

Sifat parameter nilai adalah bahwa meskipun nilai parameter dalam prosedur mengalami perubahan, tetapi perubahannya tidak dibawa ke dalam program utama. Sedangkan sifat parameter variabel yaitu jika nilai parameter dalam prosedur mengalami perubahan, maka perubahan itu dibawa ke dalam program utama.

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.

Bentuk umum deklarasi fungsi adalah :

function *nama* <(*dafpar*)> : *tipe*;

dengan *nama* : nama fungsi.

dafpar : daftar parameter formal

tipe : tipe data dari fungsi tersebut.

2.5.7. Rekursif

Suatu objek dikatakan rekursif apabila sebagian objek itu berisi atau didefinisikan sebagai dirinya sendiri.

Algoritma rekursif bias berjalan apabila persoalan yang diselesaikan, atau nilai fungsi yang dihitung, atau struktur data yang diproses sudah dinyatakan dalam bentuk rekursif. Secara umum, program rekursif dapat dinyatakan sebagai komposisi $\mathcal{P}$ dari statemen dasar S_i (yang tidak berisi $\mathcal{P}$) dan $\mathcal{P}$ itu sendiri.

$$P = \mathcal{P} [S_i, P] \quad (2.5.8.1)$$

Piranti yang perlu dan cukup untuk menyatakan program rekursif adalah prosedur. Jika prosedur P berisi acuan eksplisit ke dirinya sendiri., maka disebut dengan rekursif langsung, sedangkan jika P berisi acuan ke prosedur lain Q , maka disebut rekursif tidak langsung. Dalam hal ini sifat rekursif menjadi tidak begitu terlihat.

Dalam prosedur rekursif juga dimungkinkan adanya komputasi yang tidak berhingga, sehingga diperlukan kondisi untuk menghentikan proses eksekusi program. Persyaratan dasarnya jelas bahwa pemanggilan rekursif ke prosedur P harus memperhatikan B yang pada suatu ketika harus bernilai salah. Dengan demikian, algoritma rekursif dapat juga dinyatakan dengan

$$P = \text{if } B \text{ then } \mathcal{P} [S_i, P] \quad (2.5.8.2)$$

$$\text{atau} \quad P = \mathcal{P} [S_i, \text{if } B \text{ then } P] \quad (2.5.8.3)$$