

Memahami Struktur dan Element pada Bahasa Pemrograman

Chendra Hadi S.

chendrahadi@yahoo.com

Lisensi Dokumen:

Copyright © 2003 IlmuKomputer.Com

Seluruh dokumen di **IlmuKomputer.Com** dapat digunakan, dimodifikasi dan disebarkan secara bebas untuk tujuan bukan komersial (nonprofit), dengan syarat tidak menghapus atau merubah atribut penulis dan pernyataan copyright yang disertakan dalam setiap dokumen. Tidak diperbolehkan melakukan penulisan ulang, kecuali mendapatkan ijin terlebih dahulu dari **IlmuKomputer.Com**.

Pendahuluan

Sejak dulu hingga sekarang, tentu kita mengetahui bahwa di dunia komputer terdapat beraneka ragam bahasa pemrograman. Karena begitu banyaknya jenis-jenis bahasa pemrograman, bahasa-bahasa tersebut juga dikelompokkan berdasarkan kriteria tertentu. Ada yang mengelompokkannya menjadi 3 level bahasa yaitu: high level (Seperti Pascal dan Basic), middle level (Seperti Bahasa C), dan low level (Seperti Bahasa Assembly). Ada juga yang mengelompokkannya menjadi procedural/functional programming, object oriented programming, dan sebagainya.

Namun pada dasarnya, bahasa-bahasa pemrograman tersebut memiliki bagian-bagian yang serupa. Yang membedakan hanyalah tata bahasa yang digunakan. Seperti halnya belajar bahasa Indonesia, bahasa Jawa, ataupun bahasa asing. Masing-masingnya memiliki bagian-bagian yang sama, yang membedakan hanyalah istilah/symbol yang digunakan dan dialek bahasa tersebut.

Di dalam artikel ini, akan dibahas mengenai bagian-bagian bahasa pemrograman procedural dengan contoh kasus bahasa Pascal dan C. Bahasa pemrograman procedural merupakan bahasa pemrograman yang melibatkan fungsi-fungsi atau prosedur-prosedur sebagai sub program untuk membentuk solusi dari suatu permasalahan. Berbeda halnya dengan bahasa pemrograman yang berorientasi obyek, yang menggunakan pendekatan obyek dalam menyelesaikan suatu persoalan.

Sebelum masuk lebih dalam mengenai struktur dalam pemrograman procedural, kita akan membahas mengenai langkah-langkah sistematis dalam pembuatan suatu program, sebagai berikut:

1. Mendefinisikan permasalahan
2. Membuat rumusan untuk pemecahan masalah
3. Implementasi
4. Menguji coba dan membuat dokumentasi

1. Mendefinisikan Permasalahan

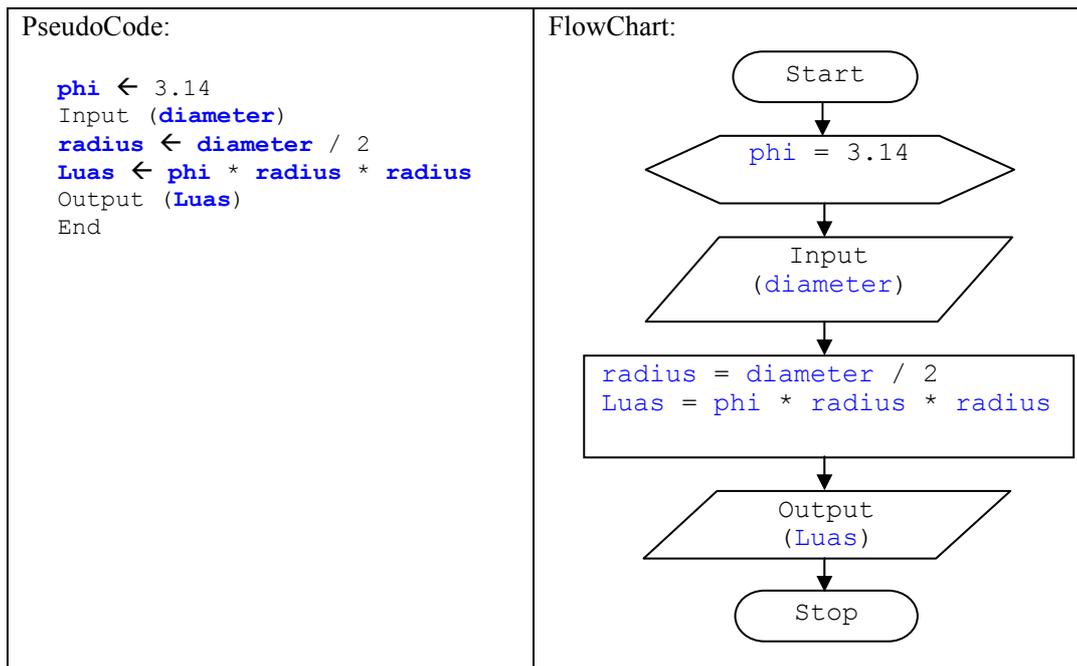
Yang dimaksud mendefinisikan permasalahan yaitu kita harus mengerti dengan baik mengenai permasalahan apa yang ingin diselesaikan. Contoh:

- a. Permasalahan menghitung luas lingkaran, dengan data yang diketahui adalah diameter lingkaran.
- b. Permasalahan menampilkan bilangan dengan kelipatan tertentu dari 0 hingga range tertentu.

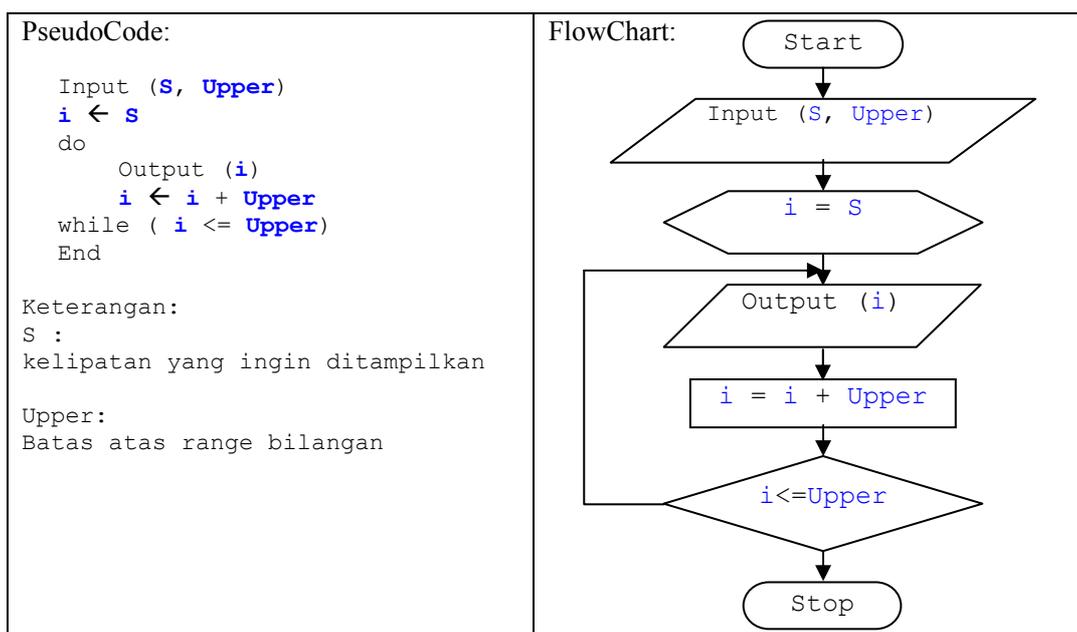
2. Membuat Rumusan untuk Pemecahan Masalah

Setelah kita mengetahui dengan baik mengenai permasalahan yang ingin diselesaikan, langkah selanjutnya yaitu membuat rumusan algoritma untuk memecahkan masalah. Rumusan tersebut dapat disusun dalam bentuk pseudocode ataupun flowchart. Contoh:

- a. Untuk contoh menghitung luas lingkaran



- b. Untuk contoh menampilkan kelipatan



3. Implementasi

Apabila langkah 1 dan 2 belum melibatkan bahasa pemrograman, maka langkah ketiga ini telah mulai melibatkan bahasa pemrograman yang ingin digunakan. Di dalam mengimplementasi algoritma kita akan menentukan bahasa pemrograman apa yang cocok atau ingin kita gunakan. Misalnya Pascal atau Delphi, Basic, dan sebagainya. Implementasi tersebut tentunya mengacu pada algoritma yang telah disusun pada langkah sebelumnya, baik itu variable-variable yang digunakan maupun alur program. Jika program diimplementasikan dengan bahasa pemrograman yang bersifat visual dan event driven (melibatkan desain form dan event-event) seperti Visual Basic atau Delphi, maka perlu pula diperhatikan langkah-langkah berikut:

1. Menambahkan obyek-obyek control pada Form seperti EditBox, ComboBox, Button, dll
2. Mengatur posisi control, properties kontrol (seperti caption, warna, jenis tulisan, dan sebagainya), serta urutan fokus obyek-obyek yang ada pada form.
3. Pemberian nama obyek kontrol yang sesuai. Misalnya untuk input diameter diberi nama txtDiameter.
4. Menentukan event-event kontrol yang berpengaruh pada fungsionalitas program
5. Mulai koding

4. Menguji Coba dan Membuat Dokumentasi

Setelah selesai implementasi, langkah selanjutnya yaitu menguji program tersebut apakah telah berjalan sesuai dengan tujuannya untuk memberi solusi dari suatu permasalahan. Apabila program belum berjalan dengan baik, maka kita perlu mengkaji kembali rumusan/algoritma yang telah dibuat pada langkah kedua, serta memperbaiki implementasi program yang mungkin keliru.

Untuk memudahkan dalam memeriksa kesalahan suatu program ataupun memahami jalannya program, kita juga perlu membuat dokumentasi dari program yang dibuat. Dokumentasi tersebut berisi informasi mulai dari tujuan/fungsi program, algoritma program, hingga cara menggunakannya.

Struktur Bahasa Program Procedural

Secara umum, bahasa pemrograman yang berbasis prosedur terdiri dari blok/sub program. yang memiliki dua bagian utama yaitu:

1. Bagian deklarasi
2. Bagian Statement

1. Bagian Deklarasi

Bagian deklarasi merupakan bagian program untuk mendefinisikan tipe data suatu variable, konstanta, serta fungsi dan prosedur yang akan digunakan pada program. Selain itu, bagian deklarasi dapat juga digunakan untuk memberi nilai awal suatu variable. Dengan kata lain, deklarasi digunakan untuk memperkenalkan suatu nama kepada Compiler program. Berikut contoh deklarasi:

- a. Deklarasi Variable:

<pre>Bahasa Pascal: var i,i2 : Integer; s : String;</pre>	<pre>Bahasa C: int i,i2; char s[100];</pre>
--	---

Penjelasan:

Untuk mendeklarasikan variable pada Pascal, digunakan reserved word **var**, kemudian diikuti dengan nama variable (identifier) yang ingin digunakan, dan kemudian tipe data dari variable tersebut. Sedangkan pada C, deklarasi diawali dengan tipe data variable baru diikuti dengan nama variable (identifier).

Suatu identifier harus diawali oleh karakter bukan angka, tetapi tidak boleh mengandung karakter khusus seperti *, - + / \ = < > . ? & dan sebagainya. Pada bahasa Pascal, identifier tidak bersifat case sensitive, maksudnya, huruf besar ataupun huruf kecil dianggap sama. Sebaliknya pada Bahasa C, identifier bersifat case sensitive, sehingga variable s dan S akan dianggap dua identifier yang berbeda.

- b. Deklarasi Konstanta pada Pascal maupun C:

```
const phi = 3.14;
```

Penjelasan:

Konstanta yaitu nilai yang tetap. Jadi jika mengacu pada contoh di atas, maka nilai phi tidak dapat diubah-ubah dan akan selalu 3.14

- c. Deklarasi Tipe Data Pascal (kiri) dan C (kanan):

<pre>type TDataSiswa = ^DataSiswa; DataSiswa = record Nama : String[30]; Alamat : String[60]; Telp : String[15]; end; type hari = (Sen, Sel, Rab, Kam, Jum, Sab, Mig);</pre>	<pre>struct DataSiswa { char Nama[30]; char Alamat[60]; char Telp[15]; }; enum hari (Sen, Sel, Rab, Kam, Jum, Sab, Mig);</pre>
---	--

Penjelasan:

Tipe Data dapat dikelompokkan menjadi:

1. Tipe Data sederhana

Tipe data sederhana merupakan tipe data yang paling kecil, yang hanya melibatkan satu item data, misalnya tipe data integer, string, real, Boolean, dan sebagainya. Kita dapat juga mendefinisikan sendiri tipe data ini. Tipe data yang didefinisikan sendiri tersebut diistilahkan enumerated data type (pada contoh adalah type hari).

2. Tipe Data terstruktur

Tipe data terstruktur merupakan tipe data yang terdiri dari beberapa item data. Bentuk dari tipe data ini dapat berupa array (terdiri dari item-item yang memiliki tipe data yang sama) ataupun record (terdiri dari item-item yang boleh memiliki tipe data yang berbeda). Pada contoh di atas, DataSiswa termasuk tipe data terstruktur.

3. Tipe Data Pointer

Tipe data pointer digunakan untuk menunjuk pada alamat memory suatu data yang lain. Jadi tipe data pointer pada dasarnya tidak menyimpan nilai data secara langsung, melainkan hanya menyimpan alamat dimana data berada. Untuk contoh pada bahasa Pascal, TDataSiswa merupakan tipe data pointer. Pada Bahasa C, untuk mendeklarasikan pointer untuk tipe data DataSiswa pada variable yang bernama TDataSiswa, dapat dituliskan sebagai berikut:

```
DataSiswa *TDataSiswa;
```

d. Deklarasi Procedure/Function:

Bahasa Pascal:

```
procedure Cetak (kal: String);  
function Tambah (a,b:Integer): Integer;
```

Bahasa C:

```
void Cetak (char *String);  
int Tambah (int a, int b);
```

Penjelasan:

Jika melihat pada contoh deklarasi pada bahasa C, mungkin timbul pertanyaan apa beda prosedur dengan fungsi? Pada Bahasa C, semua sub program dianggap fungsi, berbeda dengan Pascal yang menyertakan reserved word **procedure** dan **function** untuk membedakan antara keduanya.

Sebenarnya, perbedaan utama antara prosedur dan fungsi yaitu: prosedur adalah fungsi yang tidak mengembalikan suatu nilai. Sebaliknya fungsi adalah suatu prosedur yang mengembalikan nilai. Apabila mengacu pada contoh di atas, maka fungsi tambah akan mengembalikan suatu nilai yang bertipe integer, sedangkan prosedur Cetak tidak mengembalikan nilai apa-apa. Pada Bahasa C, procedure pada dasarnya adalah function yang mengembalikan **void** alias tidak mengembalikan nilai apa-apa.

2. Bagian Statement

Bagian statement merupakan bagian program yang berisi perintah yang akan dieksekusi/dijalankan. Pada bahasa Pascal, bagian statement selalu diawali dengan reserved word **begin** dan **end**. Apabila blok statement adalah blok utama program, maka reserved word **end** harus diakhiri dengan tanda titik(.), sebaliknya jika blok statement bukan blok utama program maka reserved word **end** diakhiri dengan tanda titik koma (;). Sebaliknya pada bahasa C, dimulai dari deklarasi variable hingga akhir statement diawali dan diakhiri dengan tanda kurung kurawal { dan }.

Berikut adalah contoh potongan kode untuk implementasi menghitung luas lingkaran dengan Bahasa Pascal (kiri) dan Bahasa C (kanan).

```
uses crt;  
  
const phi = 3.14;  
var diameter, radius,  
    Luas : real;  
begin  
    readln(diameter);  
    radius := diameter / 2;  
    Luas := phi * radius * radius;  
    writeln(Luas);  
end.
```

```
#include <stdio.h>  
  
void main()  
{  
    const phi = 3.14;  
    float diameter, radius, Luas;  
  
    scanf("%f", &diameter);  
    radius = diameter / 2.0;  
    Luas = phi * radius * radius;  
    printf("%f", Luas);  
}
```

Berikut adalah penjelasan baris demi baris dari potongan kode Pascal:

1. **uses** crt;
Reserved word **uses** digunakan untuk meng*include* library ke dalam program. Terdapat cukup banyak Library/Unit standar yang ada pada Pascal, antara lain untuk menangani perhitungan matematika, manipulasi strings, dan sebagainya.
2. **const** phi = 3.14;
Baris kedua ini digunakan untuk mendeklarasikan konstanta phi
3. **var** diameter, radius,
 Luas : real;
Baris ketiga ini digunakan untuk mendeklarasikan variable diameter, radius, dan luas dengan tipe data real (bilangan pecahan)
4. **begin**
Baris ini menandakan blok statement dimulai
5. readln(diameter);
Baris kelima berisi perintah readln yang berfungsi untuk meminta input dari user, dan kemudian disimpan ke variable diameter.
6. radius := diameter / 2;
Baris keenam ini melakukan operasi pembagian diameter dengan dua, kemudian hasilnya disimpan pada variable radius (untuk mendapatkan jari-jari dari diameter).
7. Luas := phi * radius * radius;
Baris ini kembali melakukan operasi matematika yang berfungsi untuk menghitung luas.
8. writeln(Luas);
Baris ini digunakan untuk mencetak isi variable Luas ke layar
9. **end.**
Baris ini menandakan akhir dari blok statement utama.

Berikut adalah penjelasan baris demi baris dari potongan kode Bahasa C untuk contoh di atas.

1. #include <stdio.h>
Baris di awal program ini meng*include* header library stdio ke dalam program. Seperti halnya Pascal, Bahasa C juga memiliki cukup banyak library standar yang dapat digunakan.
2. void main()
Baris kedua ini menandakan awal dari blok statement utama. Pada bahasa C, blok program utama merupakan suatu fungsi/sub program yang diberi nama 'main'.
3. { const phi = 3.14;
Pada awal baris ketiga ini, terdapat tanda kurung kurawal sebagai pembuka blok statement. Kemudian reserved word **const** digunakan untuk mendeklarasikan konstanta phi.
4. float diameter, radius, Luas;
Baris keempat ini digunakan untuk mendeklarasikan variable diameter, radius, dan luas dengan tipe data float (bilangan pecahan)
5. scanf("%f", &diameter);
Baris kelima berisi perintah yang berfungsi untuk meminta input bertipe float dari user, dan kemudian nilainya disimpan ke variable diameter.
6. radius = diameter / 2.0;
7. Luas = phi * radius * radius;
Baris keenam dan ketujuh melakukan operasi matematika untuk menghitung luas lingkaran.
8. printf("%f", Luas);
Baris ini digunakan untuk mencetak isi variable luas yang bertipe float.
9. }
Baris ini menandakan akhir dari blok statement.

Element-Element dalam Bahasa Pemrograman

Ketika kita mempelajari suatu bahasa pemrograman, kita akan menjumpai element-element yang pada dasarnya serupa antara satu bahasa dengan bahasa yang lain. Hal itu dikarenakan element-element tersebut merupakan bagian dari tata bahasa pemrograman yang bersangkutan.

Berikut adalah element-element pada bahasa pemrograman:

1. Aturan Leksikal
2. Tipe data
3. Expression
4. Statement
5. Function dan Procedure

Kita akan membahas satu per satu element-element yang tersebut di atas.

1. Aturan Leksikal

Yang dimaksud aturan leksikal yaitu aturan yang digunakan dalam membentuk suatu deklarasi, definisi, maupun statement hingga menjadi satu program yang utuh. Aturan ini meliputi beberapa element antara lain:

- a. Token
- b. Komentar
- c. Identifier
- d. Keywords (Reserved Words)
- e. Operator

1.a. Token

Token yaitu element terkecil pada bahasa pemrograman yang memiliki arti penting bagi compiler. Yang termasuk token antara lain: identifier, keywords(reserved words), operator, dan sebagainya. Token yang satu dengan yang lain dipisahkan dengan satu atau lebih spasi, tab, baris baru, atau komentar.

1.b. Komentar

Komentar yaitu teks (kumpulan karakter) yang diabaikan oleh Compiler. Komentar sangat berguna untuk memberi catatan mengenai bagian program tertentu sebagai referensi baik bagi programmer itu sendiri maupun orang lain yang membaca kode program tersebut.

Pada bahasa Pascal, teks yang berada di antara kurung kurawal pembuka {dan kurung kurawal tutup } akan dianggap sebagai komentar. Selain itu, dapat pula menggunakan tanda (* sebagai pembuka komentar, dan tanda *) sebagai penutup. Contoh:

```
(* Program mencetak hello world
   Oleh: Saya *)
begin
  { Cetak hello World
    Oleh Saya
  }
  Writeln ('Hello World');
end.
```

Pada bahasa C, teks yang berada di antara tanda /* dan tanda */ akan dianggap sebagai komentar. Dan untuk teks yang ada setelah tanda // juga akan dianggap komentar satu baris. Berikut adalah contoh penggunaan komentar pada bahasa C:

```
/* Program mencetak hello world
   Oleh: Saya */
void main() {
  // Cetak hello World
  // Oleh Saya
  printf("Hello World");
}
```

1.c. Identifier

Identifier merupakan kumpulan karakter yang digunakan sebagai penanda untuk nama variable, nama tipe data, fungsi, prosedur, dan sebagainya. Aturan penulisan identifier pada bahasa Pascal dan bahasa C dapat dikatakan serupa. Yaitu: suatu identifier harus diawali oleh karakter non angka sebagai berikut:

_ a b c d e f g h i j k l m n o p q r s t u v w x y z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Selanjutnya boleh menggunakan karakter angka (0 1 2 3 4 5 6 7 8 9) maupun karakter non angka tersebut di atas, namun tidak boleh menggunakan karakter khusus/spesial seperti + - * / ? ! { } [] dan sebagainya. Berikut adalah contoh-contoh identifier yang benar maupun salah.

_Nama →Benar
No_Telpon →Benar
Bilangan2 →Benar
4data →Salah, karena diawali oleh karakter angka: **4data**
Teks? →Salah, karena mengandung karakter khusus/special: **Teks?**

Tetapi sebagai catatan yang perlu diingat, identifier pada bahasa Pascal bersifat case insensitive (huruf besar dan huruf kecil dianggap sama), sedangkan pada bahasa C, identifier bersifat case sensitive (huruf besar dan huruf kecil dibedakan). Sebagai contoh, identifier **No_Telpon** dan **no_telpon** pada bahasa Pascal dianggap sama, sedangkan pada bahasa C, dianggap sebagai dua identifier yang berbeda.

1.d. Keywords (Reserved Words)

Keywords atau Reserved words merupakan kata-kata yang telah ada/didefinisikan oleh bahasa pemrograman yang bersangkutan. Kata-kata tersebut telah memiliki definisi yang sudah tetap dan tidak dapat diubah. Karena telah memiliki definisi tertentu, maka kata-kata ini tidak dapat digunakan sebagai identifier.

Pada bahasa Pascal, yang termasuk reserved words antara lain:

and	array	asm	begin	case	const	div
do	downto	else	end	file	for	forward
function	goto	if	in	label	mod	nil
not	of	or	packed	procedure		program
record	repeat	set	string	then	to	
type	unit	until	uses	var	while	with

Pada bahasa C, yang termasuk reserved words antara lain:

break	case	char	const	continue	default	do
double	else	enum	float	for	goto	if
inline	int	long	return	short	signed	sizeof
static	struct	switch	typedef	union	unsigned	void
while						

1.e. Operator

Operator digunakan untuk menyatakan suatu perhitungan/operasi. Operator yang digunakan untuk operasi yang melibatkan satu operand disebut unary operator. Jika melibatkan dua operand maka disebut binary operator, dan jika melibatkan tiga operand, operator tersebut disebut ternary operator. Di dalam suatu operasi dapat terdapat banyak operator. Urutan eksekusi dari operator-operator disebut juga operator precedence. Precedence yang lebih rendah akan dieksekusi belakangan, misalnya:

A = 10 + 5 * 2

Karena precedence operator * lebih tinggi daripada operator + maka nilai A adalah 20, diperoleh dari perkalian 5 dan 2, kemudian dijumlahkan dengan 10. Untuk mendahulukan eksekusi precedence yang lebih rendah dapat digunakan tanda (dan) sebagai contoh:

A = (10 + 5) * 2

Variable A akan memiliki nilai 30, diperoleh dari penjumlahan 10 dan 5, kemudian dikalikan 2.

Operator dapat dikategorikan menjadi beberapa jenis antara lain:

1. Arithmetic Operator
2. Assignment Operator
3. Bitwise dan Logical Operator
4. Relational Operator
5. Pointer Operator

Berikut akan dibahas lima kelompok operator tersebut di atas. Sebagai catatan, masih terdapat operator-operator yang belum tercakup pada kelompok tersebut di atas. Yang dibahas di sini hanya bagian-bagian yang sangat umum.

1.e.1. Arithmetic Operator

Yang termasuk di dalam operator arithmetic yaitu operator yang digunakan untuk melakukan operasi aritmatika, seperti:

- penjumlahan: **+** (pada bahasa C, terdapat juga operator **++** sebagai prefix increment. Misalnya operasi: **i++** atau **++i**, akan melakukan increment nilai **i** sebesar 1)
- pengurangan: **-** (seperti penjumlahan, pada bahasa C, juga terdapat operator **--** sebagai prefix decrement)
- perkalian: *****
- pembagian: **/** (pada bahasa Pascal, terdapat juga operator **div** yang digunakan untuk melakukan pembagian bilangan bulat)
- mencari sisa pembagian: Pada bahasa Pascal adalah operator **mod**, sedangkan pada bahasa C menggunakan tanda **%**

Operator-operator yang disebut di atas termasuk binary operator karena melibatkan dua operand. Terdapat pula operator unary, yaitu tanda **-** dan **+** yang digunakan sebagai penanda bilangan negatif atau positif.

1.e.2. Assignment Operator

Operator ini digunakan untuk memberi nilai suatu identifier. Pada bahasa Pascal, digunakan tanda titik dua dan tanda sama dengan **:=** untuk memberi nilai pada variable. Contoh:

```
C := A + B;      atau   C := 4;
```

Pada bahasa C, terdapat beberapa assignment operator yaitu:

- Tanda **=** memiliki fungsi yang sama dengan tanda **:=** pada bahasa Pascal
- Tanda **+=** digunakan untuk melakukan assignment penjumlahan, misalnya terdapat dua operasi sebagai berikut:

```
C = 4;  
C += 3;
```
- Setelah baris pertama dieksekusi, maka **C** bernilai 4. Setelah baris kedua dieksekusi **C** memiliki nilai 7.
- Tanda **-=** digunakan untuk melakukan assignment pengurangan. Cara penggunaannya sama seperti contoh penggunaan tanda **+=** di atas.
- Tanda **<<=** merupakan left shift assignment, digunakan untuk menggeser bit ke kiri.
- Tanda **>>=** merupakan right shift assignment, digunakan untuk menggeser bit ke kanan.

1.e.3. Bitwise dan Logical Operator

Operator ini digunakan untuk melakukan operasi bit dan logika. Yang termasuk di dalam operator ini antara lain:

- Negasi
bahasa Pascal : **NOT** contoh **A := NOT B;**
bahasa C : **!** contoh **A = !B;**

- And
 bahasa Pascal : AND contoh A := A AND B;
 bahasa C : && contoh A = A && B;
- Or
 bahasa Pascal : OR contoh A := B OR C;
 bahasa C : || contoh A = B || C;
- Shift Left
 bahasa Pascal : shl contoh A := B shl C;
 bahasa C : << contoh A = B << C;
- Shift Right
 bahasa Pascal : shr contoh A := B shr C;
 bahasa C : >> contoh A = B >> C;

1.e.4. Relational Operator

Operator relasional digunakan untuk membandingkan nilai dua operand. Sebagai catatan, operand yang dibandingkan harus memiliki tipe data yang sama, kecuali untuk bilangan bulat (bertipe integer) dan bilangan pecahan (bertipe real atau float). Yang termasuk operator relasional yaitu:

- Penanda kesamaan = (Pada bahasa C, penanda kesamaan menggunakan dua tanda sama dengan, yaitu ==)
- Penanda lebih besar >
- Penanda lebih besar atau sama dengan >=
- Penanda lebih kecil <
- Penanda lebih kecil atau sama dengan <=
- Penanda ketidaksamaan. Pada bahasa Pascal menggunakan tanda <> sedangkan pada bahasa C menggunakan tanda !=

1.e.5. Pointer Operator

Operator pointer digunakan untuk melakukan operasi pada operand yang berupa pointer. Pada bahasa Pascal, digunakan tanda ^ sebagai deference pointer. Sedangkan pada bahasa C, deference pointer menggunakan tanda asterisks *.

2. Tipe Data

Tipe data digunakan untuk menentukan jenis nilai yang dapat ditampung oleh suatu variable. Pada suatu bahasa pemrograman umumnya telah menyediakan tipe-tipe data yang sederhana (simple) maupun yang terstruktur. Dan apabila kita membutuhkan tipe data yang belum tersedia, kita dapat mendefinisikan sendiri tipe data baru, yang disebut enumerated type.

Berikut adalah tipe data sederhana (simple) yang terdapat pada bahasa Pascal dan bahasa C standar.

Tipe	Bahasa Pascal	Bahasa C	Jangkauan Nilai
Integer	byte	char (unsigned char)	0..255
	shortint	char (signed char)	-128..127
	integer	int	-32768..32767
	word	short (unsigned short)	0..65535
	longint	long	- 2147483648..2147483647
Real/Pecahan	real	float	3.4 E - 38..3.4 E + 38
	double	double	1.7 E - 308..1.7E + 308
String	string, char(1 karakter)	-	-
Boolean	Boolean	bool (tidak terdapat pada bahasa C standar)	true, false

Sebagai catatan, selain tipe data di atas, masih terdapat tipe-tipe data lain yang disediakan oleh compiler bahasa Pascal ataupun C. Tipe data tersebut biasanya merupakan feature yang ditambahkan oleh Compiler. Misalnya tipe data Boolean (bool) tidak terdapat pada bahasa Ansi C, namun tipe data tersebut dapat ditemui pada MS Visual C++.

Yang dimaksud dengan tipe data terstruktur yaitu tipe data yang dapat menampung lebih dari satu nilai, sbb:

1. Array

Yang dimaksud array yaitu tipe data berindeks yang terdiri dari satu atau lebih elemen/komponen yang memiliki tipe data yang sama. Berikut adalah contoh penggunaan array:

<pre>{bahasa Pascal} var bil : array[0..3] of integer; begin bil[0] := 1; bil[1] := 2; bil[2] := bil[0] + 2; bil[3] := bil[0] + bil[2]; end;</pre>	<pre>/*bahasa C*/ void main() { int bil[4]; bil[0] = 1; bil[1] = 2; bil[2] = bil[0] + 2; bil[3] = bil[0] + bil[2]; }</pre>
---	---

Pertama-tama mari kita perhatikan bagian deklarasi variable **bil** untuk contoh di atas. Pada bahasa Pascal, kita dapat menentukan indeks batas bawah dan batas atas dari suatu array, sedangkan pada bahasa C, indeks batas bawah dari array adalah selalu nol, dan batas atas array adalah jumlah element dikurang satu. Untuk contoh di atas, variable **bil** dapat menampung sebanyak 4 element, dimulai dari element ke-0, ke-1, ke-2, dan ke-3. Untuk mengakses nilai indeks tertentu, digunakan tanda kurung siku [dan]. Pada contoh di atas, dapat dilihat bahwa indeks ke-0 yaitu **bil[0]** diberi nilai 1. Kemudian, indeks ke-1 yaitu **bil[1]** bernilai 2. Selanjutnya **bil[2]** bernilai 3 (penjumlahan nilai **bil[0]** dan angka 2), dan **bil[3]** bernilai 4. Nilai-nilai **bil** di memory akan tampak sebagai berikut:

bil[0]	bil[1]	bil[2]	bil[3]
1	2	3	4

Array pada contoh di atas adalah array berdimensi 1. Kita dapat membuat array berdimensi banyak, Berikut adalah contoh penggunaan array berdimensi 2:

<pre>{bahasa Pascal} var bil : array[0..1,0..1] of integer; begin bil[0,0] := 1; bil[0,1] := 2; bil[1,0] := bil[0,0] + 2; bil[1,1] := bil[0,0] + bil[1,0]; end;</pre>	<pre>/*bahasa C*/ void main() { int bil[2][2]; bil[0][0] = 1; bil[0][1] = 2; bil[1][0] = bil[0][0] + 2; bil[1][1] = bil[0][0] + bil[1][0]; }</pre>
--	---

Nilai-nilai **bil** pada contoh di atas adalah sebagai berikut:

bil[0][0]	bil[0][1]	bil[1][0]	bil[1][1]
1	2	3	4

Atau dalam bentuk matriks 2x2 sbb:

baris/kolom	indek ke-0	indek ke-1
indek ke-0	<code>bil[0][0] = 1</code>	<code>bil[0][1] = 2</code>
indek ke-1	<code>bil[1][0] = 3</code>	<code>bil[1][1] = 4</code>

2. Record (Pascal) atau struct (Bahasa C)

Tipe data ini digunakan untuk merepresentasikan kumpulan (set) elemen/komponen yang memiliki satu jenis atau lebih tipe data. Tiap element disebut juga field atau property atau attribute. Berikut adalah contoh penggunaan record dan struct:

<pre>{bahasa Pascal} type Tsiswa = record no_induk : integer; gender : char; end; var saya : Tsiswa; teman : array[1..3] of Tsiswa; begin saya.no_induk := 10; saya.gender := 'L'; teman[1].no_induk := 1; teman[1].gender := 'L'; teman[2].no_induk := 2; teman[2].gender := 'P'; teman[3].no_induk := 3; teman[3].gender := 'P'; end;</pre>	<pre>/*bahasa C*/ struct TSiswa { int no_induk; char gender; }; void main() { struct TSiswa saya, teman[3]; saya.no_induk = 10; saya.gender = 'L'; teman[0].no_induk = 1; teman[0].gender = 'L'; teman[1].no_induk = 2; teman[1].gender = 'P'; teman[2].no_induk = 3; teman[2].gender = 'P'; }</pre>
--	--

Pada contoh di atas, dideklarasikan tipe data terstruktur yang diberi nama TSiswa. Tipe data tersebut terdiri atas dua elemen yang masing-masing bertipe integer dan char.

Setelah kita membahas mengenai tipe data sederhana dan terstruktur di atas, berikut ini akan dibahas mengenai enumerated data type. Yang dimaksud dari enumerated data type yaitu tipe data yang didefinisikan sendiri oleh programmer. Berikut adalah contoh pada bahasa Pascal dan bahasa C.

<pre>{bahasa Pascal} type Thari = (sen, sel, rab, kam, jum, sab, ming); var hari : Thari; begin hari := sen; end;</pre>	<pre>/*bahasa C*/ enum Thari { sen, sel, rab, kam, jum, sab, ming}; void main() { Thari hari; hari := sen; }</pre>
---	---

3. Expression

Yang dimaksud dengan expression (ekspresi) yaitu suatu pernyataan yang menghasilkan suatu nilai. Expression tersusun dari operator dan operand yang digunakan untuk menghitung atau memberi suatu nilai suatu variable atau identifier.

Expression yang paling sederhana yaitu nama variable. Expression yang lebih kompleks akan melibatkan operator-operator, maupun pemanggilan function atau procedure. Berikut adalah contoh-contoh expression:

- A → hanya berupa nama variable
- 10 → berupa suatu nilai
- A + 3 * 2 → expression menggunakan operator
- Calculate (A, B) → melakukan pemanggilan function bernama Calculate

4. Statement

Seperti yang telah disinggung di atas, statement merupakan bagian program yang berisi perintah yang akan dieksekusi/dijalankan. Karena itu, statement-statement ini menentukan bagaimana jalannya program dan bagaimana suatu nilai variable dimanipulasi/berubah.

Statement dapat dikelompokkan menjadi antara lain:

1. Simple Statement
2. Compound Statement
3. Selection Statement
4. Iteration Statement

4.1. Simple Statement

Yang digolongkan ke dalam simple statement (statement sederhana) yaitu statement yang tidak berisi statement lainnya, sbb:

- Assignment Statement

Yaitu statement yang digunakan untuk memberikan nilai ke suatu variable, contoh pada bahasa Pascal (kiri) dan bahasa C (kanan):

```
a := 10;
b := a * 2;
c := c * b;
Student[i].id := 12;
```

```
a = 10;
b = a * 2;
c = c * b;
Student[i].id = 12;
```

- Statement untuk pemanggilan function atau procedure

Yaitu statement yang memanggil function atau procedure yang telah didefinisikan pada program. Contoh pemanggilan procedure dan fungsi pada bahasa Pascal dan bahasa C (dengan asumsi procedure Calculate, Cetak, dan function GetLength untuk contoh di bawah, telah didefinisikan terlebih dahulu) :

```
Calculate (a,b);
b := GetLength(Student[i].Name);
Cetak (a,b);
```

```
Calculate (a,b);
b = GetLength(Student[i].Name);
Cetak (a,b);
```

- Jump Statement

Yaitu statement yang digunakan untuk melompati statement-statement lain. Yang termasuk dalam kategori statement ini yaitu:

Fungsi	Bahasa Pascal	Bahasa C
Melompat ke statement tertentu	goto	goto
Keluar dari iterative statement	break	break
Melompat ke iterasi selanjutnya pada iterative statement	continue	continue
Keluar dari function/procedure	exit	return

4.2. Compound Statement

Compound statement (kumpulan statement) adalah sekumpulan statement yang terdiri dari statement-statement lain, termasuk juga iteration dan selection statement yang akan dibahas setelah

ini. Pada bahasa Pascal, kumpulan statement diawali oleh keywords **begin** dan ditutup oleh keywords **end**, sedangkan pada bahasa C, kumpulan statement akan diawali dan diakhiri oleh tanda kurung kurawal { dan }. Berikut adalah contoh compound statement:

<pre><i>{bahasa Pascal}</i> if i = 0 then begin <i>{compound statement starts here}</i> write('Nilai i tidak boleh nol'); hasil := -1; <i>{compound statement ends here}</i> end;</pre>	<pre><i>/*bahasa C*/</i> if (i = 0) { <i>//compound statement starts here</i> printf("Nilai i tidak boleh nol"); hasil = -1; <i>//compound statement ends here</i> }</pre>
--	--

Selection Statement melakukan pemeriksaan nilai/kondisi, yang kemudian akan memilih statement mana yang akan dieksekusi. Statement ini terdiri dari 2 jenis yaitu: if..then..else statement dan case/switch statement. Berikut adalah cara penulisan selection statement pada bahasa Pascal dan bahasa C:

Selection Statement	Bahasa Pascal	Bahasa C
If... then	<pre>if (kondisi) then (statement)</pre> <p>Contoh:</p> <pre>if a = 0 then b := a;</pre>	<pre>if (kondisi) (statement)</pre> <p>Contoh:</p> <pre>if (a == 0) b = a;</pre>
if...then... else...	<pre>if (kondisi) then (statement) else (statement)</pre> <p>Contoh:</p> <pre>if a = 0 then b := a else b := -1;</pre>	<pre>if (kondisi) (statement) else (statement)</pre> <p>Contoh:</p> <pre>if (a == 0) b = a; else b = -1;</pre>
if...then... elseif...then... else...	<pre>if (kondisi) then (statement) else if (kondisi) then (statement) else if ... else (statement)</pre> <p>Contoh:</p> <pre>if ((a < 0) and (b < 0)) then c := abs(a)+abs(b) else if ((a=0) and (b>0)) then c := b else if ((a>0) and (b=0)) then c := a else begin c := a + b; a := c; b := c; end;</pre>	<pre>if (kondisi) (statement) else if (kondisi) (statement) else if ... else (statement)</pre> <p>Contoh:</p> <pre>if ((a < 0) && (b < 0)) c = abs(a)+abs(b); else if ((a==0) && (b>0)) c = b else if ((a>0) && (b==0)) c = a else { c = a + b; a = c; b = c; }</pre>

select case/switch	<pre> case of (<i>identifier</i>) <i>value_1</i> : (<i>statement</i>) ... <i>value_n</i> : (<i>statement</i>) end; Contoh: case of <i>usia</i> 5 : write('Balita'); 15 : write('Remaja'); 25 : write('Dewasa'); end; </pre>	<pre> switch(<i>identifier</i>) { case <i>value_1</i> : (<i>statement</i>) case ... case <i>value_n</i> : (<i>statement</i>) } Contoh: switch(<i>usia</i>) { case 5 : printf("Balita");break; case 15: printf("Remaja");break; case 25: printf("Dewasa");break; } </pre>
--------------------	--	---

Sebagai catatan, statement pada tabel di atas dapat berupa berbagai macam statement, baik compound statement, simple statement, maupun selection statement itu sendiri dan iteration statement yang akan dibahas berikut ini.

4.4. Iteration Statement

Iteration statement digunakan untuk melakukan perulangan sekumpulan statement (compound statement). Iteration statement pada bahasa Pascal dan C adalah sebagai berikut:

Iteration Statement	Bahasa Pascal	Bahasa C
Repeat/do loop: Evaluasi kondisi dilakukan di bagian akhir.	<pre> repeat (<i>statement</i>); ... (<i>statement</i>); until (<i>kondisi</i>); Contoh: <i>i</i> := 1; repeat <i>i</i> := <i>i</i> + 2; writeln(<i>i</i>); until <i>i</i> >=10; </pre>	<pre> do (<i>statement</i>) while (<i>kondisi</i>); Contoh: <i>i</i> = 1; do { <i>i</i> = <i>i</i> + 2; printf("%i\n",<i>i</i>); } while (<i>i</i><10); </pre>
while do: Evaluasi kondisi dilakukan di bagian awal	<pre> while (<i>kondisi</i>) do (<i>statement</i>) Contoh: <i>i</i> := 1; while (<i>i</i><10) do begin <i>i</i> := <i>i</i> + 2; writeln(<i>i</i>); end; </pre>	<pre> while (<i>kondisi</i>) do (<i>statement</i>) Contoh: <i>i</i> = 1; while (<i>i</i><10) { <i>i</i> := <i>i</i> + 2; printf("%i\n",<i>i</i>); } </pre>
for..loop: Perulangan dengan increment nilai	<pre> for <i>counter</i> := <i>lower</i> to <i>upper</i> do (<i>statement</i>) Atau for <i>counter</i> := <i>upper</i> downto <i>lower</i> do (<i>statement</i>) Contoh: for <i>i</i> := 1 to 10 do writeln(<i>i</i>); </pre>	<pre> for (<i>init</i>;<i>exp1</i>;<i>exp2</i>) (<i>statement</i>) Contoh: for (<i>i</i>=1;<i>i</i><=10;<i>i</i>++) printf("%i\n",<i>i</i>); </pre>

	<pre>for i := 1 to 10 do begin write(i); for j := 10 downto i do write(j); writeln; end;</pre>	<pre>for (i=1;i<=10;i++) { printf("%i",i); for(j=10;j>=1;j--) printf("%i",j); printf("\n"); }</pre>
--	--	---

5. Function dan Procedure

Procedure dan Function disebut juga subroutine, merupakan blok statement yang dapat dipanggil dari lokasi yang berbeda di dalam program. Yang membedakan antara function dan procedure yaitu: suatu function jika dijalankan/dipanggil akan mengembalikan suatu nilai.

Pada Bahasa C, semua subroutine adalah function. Apabila kita ingin membuat subroutine yang tidak mengembalikan nilai, kita dapat memberi nilai kembalian berupa **void**.

Ketika procedure atau function dipanggil, kita dapat melewati suatu nilai ke dalam function atau procedure tersebut. Nilai yang dilewatkan disebut juga argument atau parameter. Ada dua cara melewati nilai, yaitu:

1. Passing by Value (Dilewatkan secara nilai)
- 2.

Jika di dalam procedure atau function dilakukan perubahan nilai parameter yang dilewatkan secara nilai, maka nilai parameter yang sebenarnya tidak ikut berubah, hal ini dikarenakan parameter yang dilewatkan secara nilai akan dicopy sebagai nilai local di procedure/function yang bersangkutan. Contoh pada bahasa Pascal (kiri) dan C (kanan):

<pre>{bahasa Pascal} uses crt; procedure Ubah(a:integer); begin a := 10; writeln('Ubah menjadi=',a); end; {main program} var bil : integer; begin bil := 1; writeln('Bil sebelum=',bil); Ubah(bil); writeln('Bil sesudah=',bil); end;</pre>	<pre>/*bahasa C*/ #include <stdio.h> void Ubah(int a) { a = 10; printf("Ubah menjadi=%i\n",a); } /*main program*/ void main() { int bil; bil = 1; printf("Bil sebelum=%i\n",bil); Ubah(bil); printf("Bil sesudah=%i\n",bil); }</pre>
---	--

Hasil eksekusi kedua program di atas adalah sama, pada layar akan tampak tulisan:

```
Bil sebelum=1
Ubah menjadi=10
Bil sesudah=1
```

3. Passing by Reference

Jika di dalam procedure atau function dilakukan perubahan nilai parameter yang dilewatkan secara reference, maka nilai parameter yang sebenarnya juga akan berubah. Contoh:

```
{bahasa Pascal}
uses crt;

procedure Ubah(var a:integer);
begin
  a := 10;
  writeln('Ubah menjadi=',a);
end;

{main program}
var bil : integer;
begin
  bil := 1;
  writeln('Bil sebelum=',bil);
  Ubah(bil);
  writeln('Bil sesudah=',bil);
end;
```

```
/*bahasa C*/
#include <stdio.h>

void Ubah(int *a)
{
  *a = 10;
  printf("Ubah menjadi=%i\n",*a);
}

/*main program*/
void main()
{ int bil;
  bil = 1;
  printf("Bil sebelum=%i\n",bil);
  Ubah(&bil);
  printf("Bil sesudah=%i\n",bil);
}
```

Jika dijalankan, maka di layar akan tampak:

```
Bil sebelum=1
Ubah menjadi=10
Bil sesudah=10
```

Penutup

Artikel ini sebenarnya bukan ditujukan untuk mempelajari bahasa pemrograman Pascal ataupun C. Melainkan membahas mengenai struktur dan element yang ada pada bahasa pemrograman secara umum. Dengan memahami elemen-elemen bahasa, kita dapat dengan cepat dan mudah untuk mempelajari berbagai macam bahasa pemrograman. Sebagai catatan akhir, artikel ini dibuat berdasarkan pengalaman penulis ditambah dengan beberapa referensi lainnya seperti MSDN dan online help Delphi.