PERTEMUAN 2
NOTASI ALGORITMIK dan BAHASA C++

Algoritma dan Pemrograman I

Algoritma disusun berdasarkan 3 bagian, antara lain :

1. Judul Algoritma
2. Kamus / Deklarasi
3. Algoritma / Deskripsi

Format Syntax Algoritma

JUDUL
ALGORITMA

Nama Algoritma

DEKLARASI/
KAMUS
{Lokal / Umum}
ALGORITMA

```
Type namatipe : tipe [subrange]
Type namatipe : array [min..maks] of tipe
Namavar       : tipe
Namavar       : array [ min..maks] of tipe
Const nama    = nilai
Procedure namaproc
(Input/Output[daftar_nama_parameter:tipe])
```

TUBUH
ALGORITMA/
DESKRIPSI

Notasi Assigment
Notasi Kondisional/Pemilihan
Notasi Pengulangan
Notasi Pemanggilan

URAIAN
ALGORITMA

Kumpulan algoritma masing-masing Procedure ataupun Function yang dipanggil dari Tubuh Algoritma

Algoritma untuk menulis Hello world:

```
Algoritma Hello_world
{ program untuk mencetak "Hello world"}

DEKLARASI
  {tidak ada}

DESKRIPSI
   Write("Hello world")
```

Algoritma untuk menghitung luas segiempat:

```
LuasSegi4
 {algoritma untuk menghitung luas segiempat dengan
  diketahui panjang dan lebarnya}
KAMUS
  Luas,panjang,lebar :integer
ALGORITMA
  panjang  ← 10
  lebar ← 5
  Luas ← panjang * lebar
  write(Luas)
```


Translasi Teks Algoritma ke dalam Teks Program Bahasa C++

Translasi Notasi Kamus

	Algoritma	C++
1		#include <nama_unit>
2	CONST namaconst = nilai	#define namaconst nilai
3	TYPE namatipe : tipedata	Typedef tipedata namatipe;
4	namavar : tipedata namavar : namatipe	tipedata namavar; namatipe namavar;

Contoh :

1		#include <iostream.h>
2	CONST phi = 3.14	#define phi 3.14
3	TYPE jumlah : integer	Typedef int jumlah;
4	n : integer n : jumlah	int n; jumlah n;

Lanjutan Translasi Notasi Pengendalian

	Algoritma	C++

5	IF <kondisi> THEN
	  aksi1
	ELSE
	  aksi2
	ENDIF
		If (kondisi)
		  aksi1;
		else
		  aksi2;
		
		atau
		
		If (kondisi)
		{
		  aksi1;
		}
		else
		{
		  aksi2;
		}

Lanjutan Translasi Notasi Pengendalian

	Algoritma	C++

6	DEPEND ON <ekspresi>
	 <ekspresi 1> : aksi_1
	 <ekspresi 2> : aksi_2
	  :
	 <ekspresi n> : aksi_n
	atau
	CASE namavarcase OF
	  expkonstan 1 : aksi_1
	  expkonstan 2 : aksi_2
	  :
	  expkonstan n : aksi_n
	ELSE
	  aksi_lain
	ENDCASE
		Switch (ekspresi)
		{
		 case nilai1:
		  aksi_1;
		  break;
		 case nilai2:
		  aksi_2;
		  break;
		  :
		 case nilain:
		  aksi_n;
		  break;
		 default: aksi_lain;
		}

Lanjutan Translasi Notasi Pengendalian

	Algoritma	C++

7	[inisialisasi]
	WHILE <*kondisi_ulang*> DO
	daftar_aksi
	{ada aksi thd var kondisi}
	ENDWHILE
		[inisialisasi]
		while (*kondisi_ulang*)
		{
		daftar_aksi;
		/*ada aksi thd var kondisi*/
		}

8	[inisialisasi]
	REPEAT
	daftar_aksi
	{ada aksi thd var kondisi}
	UNTIL <*kondisi_stop*>
		[inisialisasi]
		do
		{
		daftar_aksi;
		/*ada aksi thd var kondisi*/
		}
		while (*kondisi_ulang*);

Lanjutan Translasi Notasi Pengendalian

	Algoritma

9	namavar TRAVERSAL [awal..akhir]
	daftar_aksi;
	
	atau
	
	FOR var ← awal TO/DOWNTO akhir STEP counter DO
	daftar_aksi
	ENDFOR

	C++

	for(awal;kondisiulang;step)
	{
	daftar_aksi;
	}

Contoh Algoritma

```
Algoritma Luas_Lingkaran
   {menghitung luas lingkaran diketahui jari-jarinya}

DEKLARASI
   CONST phi = 3.14
   r : integer
   Luas : real

DESKRIPSI
   r ← 5
   Luas ← phi * r * r
   output(”Jari –jari = ”,r);
   output(”Luas lingkaran  = ”,Luas);
```


Contoh program C++

```
#include <iostream.h>
#define phi 3.14

main ()
{
  int r;
  float Luas;
  r = 5;

  Luas = phi * r * r;

  cout << ”Jari-jari : ” << r;
  cout << ”\nLuas lingkaran : ” << Luas;
}
```

