BAB II
P E N U N J A N G

II.1 Dasar Pemrograman Bahasa C.

Bahasa C, merupakan suatu bahasa pemrograman yang sering dipakai dalam pembuatan perangkat lunak. Banyak sekali kelebihan-kelebihan bahasa C, jika dibandingkan dengan bahasa pemrograman lainnya. Pada penulisan tugas akhir kali ini penulis akan menggunakan bahasa C dalam pembuatan rancangan penambahan fasilitas sistem operasi.

II.1.1 Kata-kata kunci dalam Turbo C

Kata-kata kunci adalah kata-kata yang telah digunakan oleh kompiler dan tidak dapat digunakan oleh penulis program sebagai pengenal (misalnya untuk Variabel atau nama fungsi).

Ada 39 buah kata kunci dalam turbo C, yaitu:

asm	doble	int	static
auto	else	interrupt	struct
break	elnum	long	switch
case	extern	near	typedef
cdecl	far	pascal	union
char	float	register	unsigned
const	for	return	void
continue	goto	short	volatile
default	huge	signed	while
do	if	sizeof	

Bahasa C merupakan bahasa yang sensitif terhadap bentuk huruf. Penulisan dalam huruf besar berbeda dengan penulisan huruf kecil. Semua kata kunci harus ditulis dengan huruf kecil. Jika penulisan kata kunci mengandung huruf besar maka akan dianggap bukan kata kunci lagi.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation: (http://eprints.undip.ac.id)

contoh : if berbeda dengan IF

break berbeda dengan Break

II.1.2 Praprosesor #include dan #define

A. Praprosesor #include

Pengarah praprosesor #include dipakai untuk membaca suatu file (kita sebut dengan file judul) yang didalamya berisi deklarasi-deklarasi dan atau definisi-definisi konstanta.

Bentuk umum penulisan :

#include <nama file> atau:

#include "nama file"

Contoh-contoh file judul yang disediakan oleh turbo C dan beberapa fungsi yang terkandung didalamnya :

1. stdio.h

printf()	puts()	gets()	scanf()
rename()	rewind()	fprint()	fopen()
fclose()	fread()	fwrite()	.....dsb

2. conio.h

clrscr()	getch()	getche()	gotoxy()
putch()	wherex()	wherey	.....dsb

3. math.h

sin()	cos()	tan()	asin()
acos()	atan()	exp()	log()
log10()	pow()	sqrt()	.....dsb

4. string.h

strcpy()	strcat()	strchr()	strlen()
strcmp()		.....dsb	

5. stdlib.h

exit()	free()	system()	atof()
atoy()	atol()	abs()	.....dsb

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

B. Praprosesor #define

Praprosesor #define digunakan untuk mendefinisikan suatu konstanta.

contoh : #define PHI 3.14

Pendefinisian di atas mengisyaratkan bahwa nilai 3.14 telah didefinisikan dengan konstanta lain yang bernama PHI, sehingga kita dapat memakai PHI untuk mengadakan perhitungan di dalam program.

II.1.3 Struktur Penulisan Program Bahasa C

Struktur penulisan program bahasa C dapat dilihat sebagai kumpulan dari sebuah atau beberapa fungsi. Fungsi pertama yang harus ada dalam penulisan program bahasa C sudah ditentukan namanya, yaitu bernama main().

Suatu fungsi dalam bahasa C dibuka dengan diawali tanda { dan ditutup dengan tanda } , sedangkan statemen-statemen program dapat ditulis diantara kedua tanda tersebut.

Perhatikan struktur penulisan program berikut:

```
main()
{
  statement program
  statement program
}
```
} fungsi main() yang telah ditentukan namanya

```
Fungsi_satu()
{
  statement program
}
```
} fungsi yang lain yang nama fungsinya dapat dibuat sendiri

```
Fungsi_dua()
{
  statement program
}
```
} fungsi yang lain yang nama fungsinya dapat dibuatsendiri

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

II.1.4 Komentar dalam Program

Agar program yang ditulis mudah dipahami, biasanya pada program disertakan komentar atau keterangan-keterangan mengenai program tersebut. Dalam bahasa C komentar program ditulis dengan diawali tanda /* dan diakhiri dengan tanda */.

contoh :

```
/* contoh komentar program */
```

Contoh lain yang komentarnya lebih dari satu baris :

```
/*      PROGRAM INI DISUSUN OLEH      *
 *      ------------------------      *
 *      Nama     : Sumarno            *
 *      NIM      : J 101 91 0550      *
 *      ------------------------      */
```

II.1.5 Type Data

Bahasa C menyediakan lima macam type data dasar, yaitu :

NO.	Type Data	Kata kunci
1.	Integer (bulat)	int
2.	Pecahan ketepatan tunggal	float
3.	Pecahan ketepatan ganda	double
4.	Karakter	char
5.	Tak Bertype	void

Kelima type data dasar tersebut dapat dikombinasikan dengan 4 pengubah sbb:

- signed
- unsigned
- long
- short

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Untuk lebih jelas lagi perhatikan tabel berikut :

TYPE DATA	LEBAR (BIT)	JANGKAUAN	
		AWAL	AKHIR
int	16	- 32768	32767
signed int	16	- 32768	32767
short int	16	- 32768	32767
signed short int	16	- 32768	32767
unsigned int	16	0	65535
unsigned short int	16	0	65535
long int	32	-2147483648	2147483649
signed long int	32	-2147483648	2147483649
unsigned long int	32	0	4294967296
float	32	3.4E-38	3.4E+38
double	64	1.7E-308	1.7E+308
long double	80	3.4E-4932	3.4E+4932
char	8	-128	127
signed char	8	-128	127
unsigned char	8	0	255

II.1.6 Variabel dan Mendeklarasikan Variabel

A. Pengertian Variabel

Variabel adalah nama-nama yang ditentukan sendiri oleh pembuat program yang digunakan untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat diubah-ubah selama eksekusi program berlangsung.

B. Mendeklarasikan Variabel

Variabel yang akan digunakan dalam program yang ditulis dalam bahasa C haruslah dideklarasikan terlebih dahulu. Pengertian deklarasi disini adalah memesan memori dan menentukan jenis data yang bisa disimpan didalamnya. Bentuk umum pendeklarasian variabel :

```
Typedata    daftar_variabel;
```

daftar_variabel dapat berisi satu nama variabel atau lebih. Jika daftar_variabel lebih dari satu nama variabel

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

maka antara variabel satu dengan variabel yang lain dipisahkan dengan tanda koma.

contoh :

```
int x;     /* Variabel x bertipe integer (bulat) */
float a,b; /* Variabel a dan b dideklarasikan sbg*
            * variabel bertipe pecahan           */
```

II.1.7 Konstanta

A. Pengertian Konstanta

Berbeda dengan Variabel, konstanta adalah suatu nilai yang tidak berubah selama eksekusi program berlangsung.

B. Konstanta Numerik Integer

Konstanta Numerik Integer merupakan konstanta dengan nilai numerik bilangan bulat.

contoh :

-43

3700

C. Konstanta Numerik Pecahan

Konstanta Numerik Pecahan adalah nilai numerik yang dapat mempunyai nilai pecahan dibelakang titik desimal. Konstanta jenis ini dapat juga ditulis dengan notasi saintifik (notasi e).

contoh :

423.45

-20.87

1.2e-06 artinya 1.2×10^{-6}

5.4E+22 artinya 5.4×10^{22}

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

D. Konstanta Karakter dan String.

Konstanta Karakter adalah nilai sebuah karakter atau huruf yang ditulis diantara tanda petik tunggal sedangkan konstanta string adalah nilai satu buah karakter atau lebih yang ditulis diantara tanda petik ganda.

contoh :

'a'	(konstanta karakter huruf a)
"a"	(konstanta string huruf a)
"matematika"	(konstanta string matematika)

E. Konstanta karakter Escape

Konstanta Karakter Escape digunakan untuk menghasilkan sesuatu. Konstanta jenis ini penulisannya diawali dengan tanda \ (back slash).

Dibawah ini tabel konstanta escape dan fungsinya.

Konstanta Escape	Hasil
\a	Bunyi bell
\b	Mundur satu spasi
\f	Ganti halaman baru
\n	Ganti baris baru
\r	Menuju kolom pertama di baris tsb
\t	Tabulasi Horisontal
\v	Tabulasi Vertikal
\'	Karakter petik tunggal
\"	Karakter petik ganda
\\	Karakter garis miring
\0	Karakter tanda NULL (nilai kosong)

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

II.1.8 Operator

Operator merupakan simbol atau karakter yang biasa dilibatkan dalam penulisan program untuk melakukan suatu operasi.

Berikut ini tabel Operator dalam bahasa C :

Operator	Jenjang
() , [] , -> , .	1
! , ~ , ++ , -- , & , (tipe) , sizeof	2
* , / , %	3
+ , -	4
<< , >>	5
< , <= , > , >=	6
== , !=	7
&	8
^	9
¦	10
&&	11
¦¦	12
?¦	13
= , += , -= , *= , /= , %=	14
&=, ^= , ¦= , <<= , >>=	15
,	16

Jenjang menunjukan operator mana yang akan diproses terlebih dahulu. Misalnya operator * akan diproses lebih dulu dibandingkan dengan operator + sebab jenjang operator * lebih tinggi dari pada operator +. Untuk operator yang memiliki jenjang yang sama tinggi, maka operator yang berada diposisi lebih kiri akan dikerjakan terlebih dahulu.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Operator Aritmatika

Operator	Arti	Jenjang
*	Perkalian	3
/	Pembagian	3
%	Sisa Pembagian	3
+	Penjumlahan	4
-	Pengurangan	4

contoh :

A + B / C + D mempunyai arti $A + \frac{B}{C} + D$

A % B * C mempunyai arti (A % B)*C

Operator Penurunan dan Penaikan

Operator	Arti	Jenjang
++	Peningkatan dengan nilai 1	2
--	Penurunan dengan nilai 1	2

contoh :

++A atau A++ mempunyai arti A=A+1

--B atau B-- mempunyai arti B=B-1

Operator Manipulasi Bit

Operator	Arti	Jenjang
~	NOT (Tidak)	2
<<	Geser ke kiri	5
>>	Geser ke kanan	5
&	AND	8
^	XOR	9
ñ	OR	10

Operator ini digunakan untuk mengolah data yang berkaitan dengan bilangan basis dua.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

- Operator ~ (Komplemen)

Operator ini mempunyai sifat membalik nilai setiap bit, artinya bit bernilai 1 akan diubah menjadi bit bernilai 0 demikian juga sebaliknya.

- Operator << dan Operator >>

Operator ini untuk menggeser bit kekiri atau kekanan sebanyak yang diperintahkan.Bit yang kosong akibat dari pergeseran diisi dengan nilai nol.

- Operator & , Operator ^ dan Opertor |

Perhatikan Tabel Operasi berikut :

Operand 1	Operand 2	Hasil		
		&	^	|
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	0	1

Operator (tipe)

Pembagian dua buah bilangan integer yang diterima di variabel pecahan akan tetap bernilai integer. Jika hasil yang diinginkan harus bernilai pecahan maka dapat digunakan operaor (type).

contoh :

```
#include <stdio.h>

main()

{ int x=7,y=3;
  float z;

  z=x/y;
  printf(" NILAI Z = %f\n", z);

  z=(float) x/y;
  printf(" NILAI Z = %f\n", z);
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

```
    z=(float) (x/y);
    printf(" NILAI Z = %f\n", z);
}
```

Hasil program :

```
NILAI Z = 2.000000
NILAI Z = 2.333333
NILAI Z = 2.000000
```

Operator pengerjaan.

Perhatikan tabel Operasi berikut.

Operator	Contoh penggunaan	Arti
+=	A += 1	A = A + 1
-=	A -= B	A = A - B
*=	A *= B	A = A * B
/=	A /= 3	A = A / 3
%=	A %= 2	A = A % 2

Operator hubungan / relasi

Operator hubungan digunakan untuk membandingkan dua buah nilai. Hasil perbandingan akan bernilai satu (jika benar) dan nol (jika salah).

Operator	Jenjang	Arti
<	6	Lebih kecil dari
<=	6	Lebih kecil atau sama dengan
>	6	Lebih besar dari
>=	6	Lebih besar atau sama dengan
==	7	Sama dengan
!=	7	Tidak sama dengan

contoh :

```
#include <stdio.h>

main()
{ int A=5; B=7;
   printf(" %d < %d Hasilnya %d (benar) \n",A,B,A<B);
   printf(" %d > %d Hasilnya %d (salah) \n",A,B,A>B);
}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Hasil program :

```
5 < 7 Hasilnya 1 (benar)
5 > 7 Hasilnya 0 (salah)
```

Operator Logika

Operator logika digunakan untuk membandingkan hasil dari operator-operator hubungan.

Operator	Jenjang	Arti
!	2	tidak
&&	11	dan
||	12	atau

Perhatikan tabel operasi berikut :

X	Y	!X	!Y	X && Y	X || Y
1	1	0	0	1	1
1	0	0	1	0	1
0	1	0	1	0	1
0	0	1	1	0	0

Operator koma (,)

Operator koma digunakan untuk menggabung beberapa ungkapan dengan proses yang berurutan dari ungkapan sebelah kiri koma ke ungkapan sebelah kanan koma.

contoh :

```
x = (b=5, b*2)
  = 10
```

II.2 Operasi masukan dan Pengeluaran.

Memasukan data dan menampilkan hasil merupakan tindakan yang seringkali dilakukan dalam pemrograman. Pada umumnya penampilan hasil ditujukan kepiranti layar, sedangkan pemasukan data biasanya melalui papan ketik.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation.
(http://eprints.undip.ac.id)

II.2.1 Menampilkan hasil atau informasi ke layar.

Menampilkan hasil dalam bahasa C semua prosesnya dilakukan oleh fungsi-fungsi. Fungsi-fungsi ini berada di file judul stdio.h dan conio.h.

A. Tampilan hasil tidak terformat

Pengertian tidak terformat disini adalah lebar dan bentuk dari tampilannya tidak dapat diatur oleh penulis program. Beberapa fungsi pustaka untuk menghasilkan informasi tidak terformat antara lain : putch() (untuk menampilkan nilai karakter) dan puts() (untuk menampilkan nilai string).

contoh :

```
#include <conio.h>
main()
{ char c,s[11];
   c='A';
   strcpy(s,"Ini String");

   puts(s);
   putch(c);
   putch(c);
}
```

Hasil program :

```
Ini String
AA
```

Fungsi puts() secara otomatis akan diakhiri dengan perpindahan baris, sedangkan fungsi putch() tidak.

B. Tampilan Hasil terformat.

Untuk menampilkan hasil terformat dapat digunakan fungsi printf() yang prototypenya berada di file stdio.h. Pengaturan bentuk format dapat dilakukan melalui kode-kode format yang dapat dilihat pada tabel berikut :

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Kode Format	Kegunaan
%c	Format nilai sebuah karakter
%s	Format nilai String
%d	Format nilai desimal integer
%i	Format nilai desimal integer
%u	Nilai desimal integer tak bertanda
%x	Format nilai Heksadesimal integer
%o	Format nilai Oktal integer
%f	Format nilai pecahan
%e	Format nilai pecahan dalam saintifik
%g	Pengganti %f atau %e
%p	Format alamat memori suatu pointer

Untuk menentukan format panjang medan tampilan dapat dituliskan nilai integer setelah tanda %.Nilai integer positif digunakan untuk format rapat kanan , sedangkan nilai negatip digunakan untuk format rapat kiri. Untuk mengatur lebar medan nilai pecahan dapat ditulikan format dengan bentuk :

%m.n	dimana	m = panjang medan. n = jumlah digit pecahan. m,n merupakan bilangan bulat.

Perhatikan contoh berikut :

```
#include <stdio.h>

mean()
{   unsigned int bil=4710;
    char huruf='R';
    float   x=251000.0;

    printf(" %4c \n ",huruf);
    printf(" %-4c \n ",huruf);
    printf("4710 dalam Heksadesimal     =%x\n",bil);
    printf("251000.0 dalam saintifik   =%e\n",x);
    printf("251000.0 dalam format f9.2=%9.2f\n",x);
}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Hasil program :

```
R
R
4710 dalam Heksadesimal    = 1226
251000.0 dalam saintifik   = 2.51000e+05
251000.0 dalam format f9.2= 251000.00
```

II.2.2 Memasukan data dari keyboard

Seperti halnya menampilkan hasil, proses memasukan data pada bahasa C juga dilakukan melalui fungsi-fungsi pustaka. Untuk memasukan data dari keyboard prototype fungsi pustaka berada pada file stdio.h dan conio.h.

A. Memasukan data tidak terformat.

Untuk memasukan data tidak terformat dapat digunakan fungsi-fungsi pustaka sbb :

getche()	Untuk memasukan nilai karakter tanpa diakhiri dengan tombol Enter dan nilai karakter tersebut akan ditampilkan dilayar monitor.
getch()	Mempunyai fungsi yang sama dengan getche() akan nilai karakternya tidak ditampilkan dilayar monitor.
getchar()	Memasukan nilai karakter dengan diakhiri tombol Enter.
gets()	Untuk memasukan nilai bertipe string.

B. Memasukan data terformat.

Untuk memasukan data terformat dapat digunakan fungsi scanf().Kode-kode format untuk memasukan data terformat

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

sama dengan kode-kode format yang digunakan untuk tampilan hasil terformat.

Perhatikan contoh berikut :

```
#include <stdio.h>
#include <conio.h>

main()
{   char huruf,kata[80];
    int  bil;

    printf("Masukan sebuah huruf !");huruf=getchar();
    printf("Masukan sebuah kata !");gets(kata);
    printf("Masukan sebuah bilangan bulat ! ");
    scanf("%d",&bil);

    printf("\n %c adalah  Huruf",huruf);
    printf("\n %s adalah  Kata ",kata);
    printf("\n %d adalah  Bilangan bulat",bil);
}
```

Hasil program :

```
Masukan sebuah huruf ! R
Masukan sebuah kata ! Matematika
Masukan sebuah bilangan bulat ! 126

R adalah Huruf
Matematika adalah Kata
126 adalah Bilangan bulat
```

II.3 Penyeleksian Kondisi.

Penyeleksian kondisi dimaksudkan untuk menyeleksi statement-statement program yang harus dijalankan sesuai dengan kondisinya.

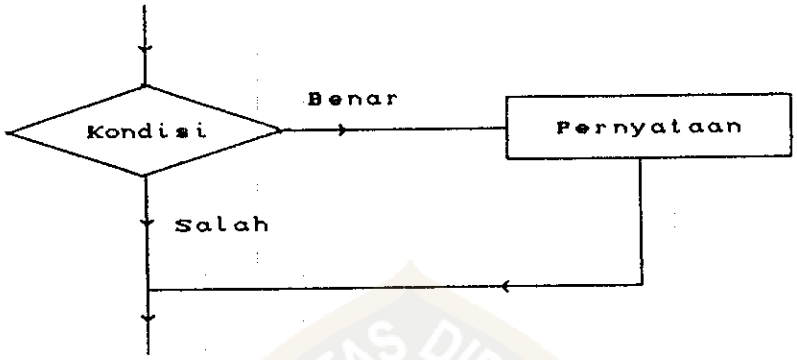
II.3.1 Statemen if dan if-else.

Statemen if mempunyai bentuk penulisan secara umum :

```
if (kondisi)
{
    Blok
    Pernyataan
}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Secara diagram alur dapat digambarkan sbb :



Jika kondisi bernilai 1 (benar) maka blok pernyataan akan diproses, akan tetapi jika kondisi bernilai 0 (salah) maka blok pernyataan tidak akan di proses.

Jika blok pernyataan hanya berisi satu pernyataan, maka tanda { dan tanda } boleh dihilangkan sehingga bentuk penulisan menjadi :

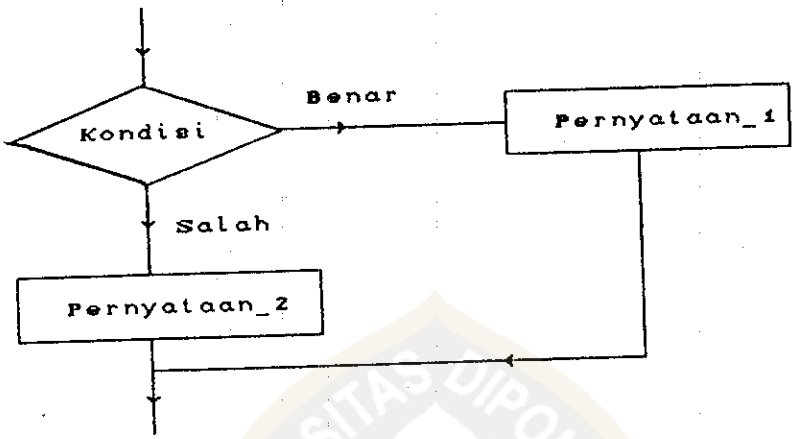
```
if (kondisi) pernyataan;
```

Penulisan Statemen if-else mempunyai bentuk secara umum :

```
if (kondisi)
{
      Blok
   Pernyataan_1
}
else
{
      Blok
   Pernyataan_2
}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more.than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Secara diagram alur dapat digambarkan sbb :



Jika kondisi bernilai 1 (benar), maka blok pernyataan_1 akan diproses dan penyeleksian akan dihentikan , akan tetapi jika kondisi bernilai 0 (salah) maka blok pernyataan_2 yang akan diproses dengan mengabaikan blok pernyataan_1.

Jika blok pernyataan_1 dan blok pernyataan_2 masing-masing hanya berisi satu buah pernyataan maka tanda { dan tanda } boleh tidak ditulis sehingga bentuk penulisan menjadi :

```
if (kondisi) pernyataan_1;
else pernyataan_2;
```

II.3.2 Statement switch

Statement switch digunakan untuk menangani penyeleksian kondisi yang melibatkan sejumlah alternatif.

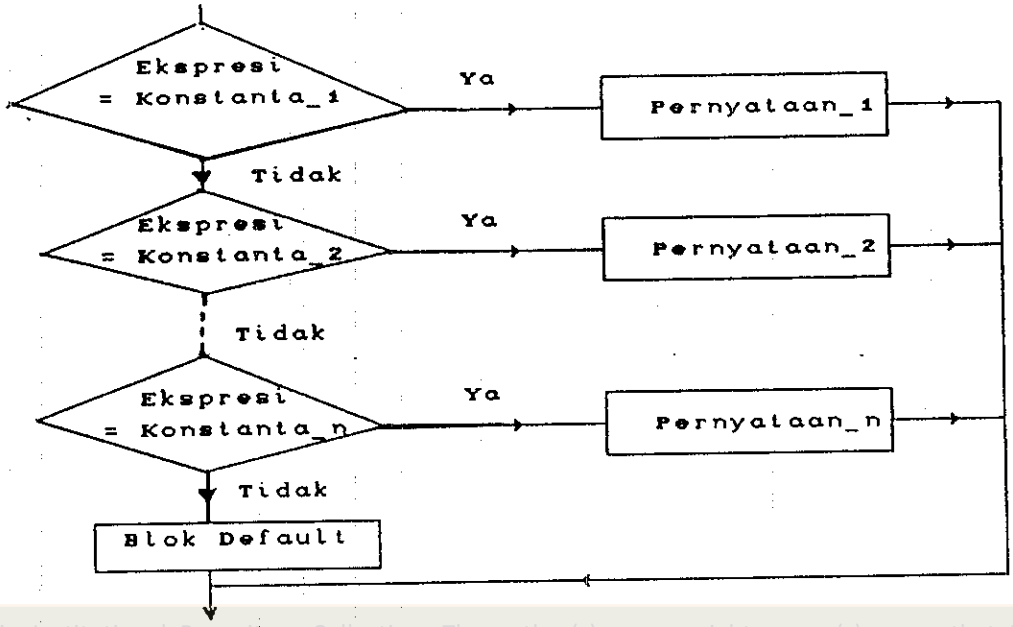
This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Bentuk umum penulisan statement switch :

```
Switch (ekspresi)
{  case konstanta_1 :
       Blok pernyataan_1
     break;
   case konstanta_2 :
       Blok pernyataan_2
     break;
   .
   .
   .
   case konstanta_n  :
       Blok pernyataan_n
     break;
   default :
       Blok default
}
```

Ekspresi berupa ungkapan bertipe integer atau karakter.

Secara diagram alur dapat digambarkan sbb :



This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

II.4 Perulangan Proses.

Perulangan proses dimaksudkan untuk mengulang statement program yang dijalankan.

II.4.1 Statement for

Bentuk umum penulisan statement for :

```
for (awal ; akhir ; tambah)
{
    Blok Pernyataan
}
```

awal : Suatu ungkapan yang memberikan nilai awal pada suatu variabel.(Variabel ini selanjutnya disebut indeks loop).

akhir : Suatu kondisi yang harus dipenuhi agar supaya perulangan proses terus dilakukan.

tambah : Suatu ungkapan yang merubah indeks loop setiap kali perulangan proses.

Jika blok pernyataan hanya terdiri dari satu pernyataan maka tanda { dan tanda } boleh tidak ditulis,sehingga bentuk penulisan menjadi :

```
for (awal ; akhir; peningkatan)
    pernyataan;
```

contoh :

```
#include <stdio.h>
main()
{   int i;
    for (i=1 ; i<=3 ; i++)
       printf("%s \n","MATEMATIKA");
}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Hasil program

```
MATEMATIKA
MATEMATIKA
MATEMATIKA
```

II.4.2 Stetement While

Bentuk umum penulisan statement while :

```
while (kondisi)
{
    Blok Pernyataan
}
```

Perulangan akan terus dilakukan jika kondisi yang diseleksi pada statement while masih bernilai benar dan akan dihentikan jika kondisinya sudah bernilai salah.

Jika blok statement hanya terdiri dari satu pernyataan maka tanda { dan tanda } boleh tidak ditulis,sehingga bentuk penulisan menjadi :

```
while (kondisi) pernyataan;
```

contoh :

```
#include <conio.h>
main()
{   int i;

    i=0;
    while (i<3)
    {   puts("MATEMATIKA");
        i++;
    }
}
```

Hasil Program :

```
MATEMATIKA
MATEMATIKA
MATEMATIKA
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

II.4.3 Statement do-while

Bentuk umum penulisan statement do-while :

```
do
{
    Blok Pernyataan
}while (kondisi)
```

Proses perulangan akan terus dijalankan jika kondisi yang diseleksi masih bernilai benar, dan akan dihentikan jika kondisi sudah bernilai salah.

Jika blok statement hanya terdiri dari satu pernyataan maka tanda { dan tanda } boleh tidak ditulis,sehingga bentuk penulisan menjadi :

```
do pernyataan while (kondisi)
```

contoh :

```
#include <conio.h>
main()
{   int i;
    i=0;

    do
    {    puts ("MATEMATIKA");
         i++;
    } while (i< 5);
}
```

Hasil Program:

```
MATEMATIKA
MATEMATIKA
MATEMATIKA
MATEMATIKA
MATEMATIKA
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

II.5 FUNGSI

Fungsi adalah suatu bagian program yang dirancang untuk melakukan suatu tugas tertentu dan letaknya dipisahkan dari program yang menggunakannya.

II.5.1 Penulisan, pendeklarasian dan hasil balik fungsi.

A. Penulisan fungsi

Bentuk umum penulisan fungsi :

```
type nama_fungsi(parameter)
{
    Tubuh Fungsi
}
```

type menentukan tipe data dari keluaran fungsi. Jika tipe ini tidak dituliskan,maka keluaran fungsi akan dianggap bertipe int.

nama_fungsi merupakan nama yang dibuat sendiri oleh pembuat program.

parameter dimaksudkan sebagai alat komunikasi data yang dikirim dari program ke fungsi yang bersangkutan.Parameter ini disebut sebagai parameter formal.Penulisan parameter formal dapat dilakukan dengan 2 cara, yaitu : cara lama dan cara baru.

Cara lama :

```
int Fg_contoh(A,B,C)
float A
int B
char C
{
    tubuh fungsi
}
```

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Cara baru :

```
int Fg_contoh(float A,int B,char C)
{
   tubuh fungsi
}
```

Pada penulisan parameter cara baru perlu diperhatikan sbb:

```
int Fg_contoh(float A,float B,char C)   (benar)
int Fg_contoh(float A,B,char C)         (salah)
```

Tubuh fungsi : berisi pernyataan-pernyataan yang menjalankan tugas fungsi tersebut.

B Pendeklarasian fungsi

Suatu fungsi yang memberikan nilai keluaran selain tipe int, harus dideklarasikan terlebih dahulu sebelum digunakan, jika tidak maka keluaran fungsi akan dianggap bertipe int.

Bentuk umum penulisan deklarasi fungsi :

```
tipe nama_fungsi();
```

catatan : - tanda titik koma harus ditulis
- parameter formal boleh tidak ditulis

C Hasil balik Fungsi

Untuk fungsi yang akan memberikan hasil balik, dapat dilakukan melalui statement return (didalam tubuh fungsi) diikuti oleh nilai hasil baliknya yang ditulis dalam tanda kurung. Statement return menandakan bahwa proses yang terjadi di dalam fungsi tersebut telah selesai.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

contoh :

```
#include <stdio.h>
long int Faktorial(); /* Deklarasi Fungsi*/

main()
{   int N;
    long int Fac;

    Printf("Ketikan sebuah bilangan ! ");
    scanf("%d",&N);
    Fac=Faktorial(N);
    printf("%d   faktorial = %ld ",Fac);
}

long int Faktorial(int N) /* Definisi fungsi */
{   int i;
    long int F=1;
    if (N<0) return (0)
    else
       for (i=2;i<=N;i++)
       F*=i;
    return(F);
}
```

Hasil program :

```
Ketikan sebuah bilangan ! 5
5 faktorial = 120
```

II.5.2 Ruang lingkup Variabel.

Dalam bahasa C, variabel-variabel yang digunakan dapat diklasifikasi menjadi beberapa bagian :

A. Variabel Lokal

Variabel lokal adalah suatu variabel yang namanya dan nilainya hanya dikenal didalam suatu fungsi. Untuk membuat suatu variabel menjadi variabel lokal, maka variabel tersebut harus dideklarasikan didalam blok fungsi yang bersangkutan.

Variabel lokal secara otomatis akan diciptakan ketika fungsi dipanggil dan akan dihapus nilai yang disimpan, ketika proses sudah meninggalkan fungsi tersebut.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

B. Variabel global

Variabel global adalah variabel yang namanya dan nilainya dikenal disemua fungsi dari program. Variabel global dibuat dengan cara mendeklarassikan diluar fungsi-fungsi yang menggunakanya.

C. Variabel register.

Variabel register adalah variabel yang nilainya di dalam register bukan dalam memori RAM dengan tujuan untuk mempercepat proses pengaksesan.Variabel register hanya bisa diterapkan pada variabel lokal yang bertipe char atau int.Variabel jenis ini biasa digunakan pada indeks loop.Cara membuat variabel register adalah dengan menambahkan kata kunci register didepan variabel yang akan dideklarasikan sebagai variabel register.

II.5.3 Cara pengiriman Parameter

Pengiriman parameter ke suatu fungsi dapat dilakukan dengan dua cara yaitu :

A. Pengiriman parameter secara nilai.

Cara ini mempunnyai karakteristik sbb:

1. Yang dikirimkan ke fungsi adalah nilai datanya.
2. Fungsi yang menerima akan menyimpan di alamat yang berbeda dengan alamat nilai data yang dikirim.
3. Perubahan nilai difungsi tidak akan merubah nilai yang berada di program yang mengirim
4. Pengiriman nilai dapat dilakukan dalam bentuk ungkapan.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation.
(http://eprints.undip.ac.id)

Contoh :

```
#include <stdio.h>
void Secara_nilai();
main()
{   float A=25;
    Secara_nilai(A,A/3);
    prinf("Informassi 2 :\n");
    printf("A=%f dialamat %p\n",A,&A);
    printf("A/3 = %f\n",A/3);
}


void Secara_nilai (float A,float B)
{   A=7;
    prinf("Informassi 1\n:");
    printf("A=%f dialamat %p\n",A,&A);
    printf("A/3 =%f\n",A/3);
}
```

Hasil program :

```
Informassi 1 :
A = 7.000000 dialamat FFCA
A/3 = 8.333333

Informassi 2 :
A = 25.000000 dialamat FFD8
A/3 = 8.333333
```

B. Pengiriman parameter secara acuan.

Cara ini mempunyai karakterisstik sbb:

1. Yang dikirim ke fungsi adalah alamat dari nilai datanya (alamat dari suatu nilai data dapat diperoleh dengan operator &)
2. Fungsi yang menerima kiriman alamat tersebut akan menggunakan alamat yang sama untuk menempatkan nilai datanya.
3. Perubahan nilai di fungsi akan merubah nilai asli yang ada diprogram yang memanggil fungsi tersebut.
4. Pengiriman tidak dapat dilakukan dalam bentuk ungkapan.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)

Contoh :

```
#include <stdio.h>
void Tambah();
main()
{ int A=2,B=3,C;

   Tambah(&A,&B);
   printf("%d ditambah %d adalah %d\n",A,B,C);
}

void Tambah(int *A,int *B, Int *C)
{   *C=*A+*B;
    return;
}
```

Hasil program :

```
2 ditambah 3 adalah 5
```

Catatan :

Suatu fungsi yang akan menerima masukan berupa alamat dari suatu data maka parameter formalnya harus dideklarasikan bertipe pointer.

This document is Undip Institutional Repository Collection. The author(s) or copyright owner(s) agree that UNDIP-IR may, without changing the content, translate the submission to any medium or format for the purpose of preservation. The author(s) or copyright owner(s) also agree that UNDIP-IR may keep more than one copy of this submission for purpose of security, back-up and preservation:
(http://eprints.undip.ac.id)