

The background features three decorative blue circles of varying sizes, each composed of concentric circles in different shades of blue. These circles are positioned in the top right and bottom right corners. Two thin, light blue lines intersect at the top left, forming a large 'V' shape that frames the text.

BAHASA PEMROGRAMAN C

Bahasa Pemrograman C

A. Pengenalan

Bahasa C diciptakan oleh Dennis Ritchie tahun 1972 di Bell Laboratories.

Kelebihan Bahasa C:

- Bahasa C tersedia hampir di semua jenis computer.
 - Kode bahasa C sifatnya adalah portable dan fleksibel untuk semua jenis computer.
 - Bahasa C hanya menyediakan sedikit kata-kata kunci. hanya terdapat 32 kata kunci.
 - Proses executable program bahasa C lebih cepat
 - Dukungan pustaka yang banyak.
 - C adalah bahasa yang terstruktur
 - Bahasa C termasuk bahasa tingkat menengah
- penempatan ini hanya menegaskan bahwa c bukan bahasa pemrograman yang berorientasi pada mesin. yang merupakan ciri bahasa tingkat rendah. melainkan berorientasi pada obyek tetapi dapat diinterpretasikan oleh mesin dengan cepat. secepat bahasa mesin. inilah salah satu kelebihan c yaitu memiliki kemudahan dalam menyusun programnya semudah bahasa tingkat tinggi namun dalam mengesekusi program secepat bahasa tingkat rendah.

Kekurangan Bahasa C:

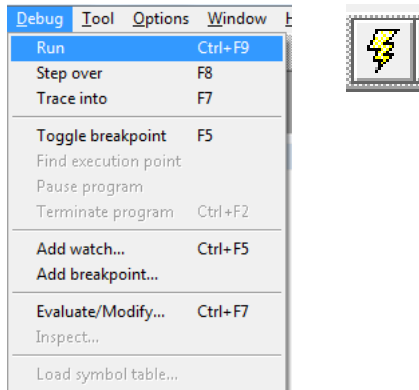
- Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
- Bagi pemula pada umumnya akan kesulitan menggunakan pointer.

B. MENGKOMPILASI PROGRAM

Suatu source program C baru dapat dijalankan setelah melalui tahap kompilasi dan penggabungan. Tahap kompilasi dimaksudkan untuk memeriksa source-program sesuai dengan kaidah-kaidah yang berlaku di dalam bahasa pemrograman C. Tahap kompilasi akan menghasilkan *relocatable object file*. File-file objek tersebut kemudian digabung dengan perpustakaan-fungsi yang sesuai. untuk menghasilkan suatu *executable-program*.

Shortcut yang digunakan untuk mengcompile:

- ALT + F9 → dipakai untuk melakukan pengecekan jika ada error pada program yang telah kita buat.
- CTRL + F9 → dipakai untuk menjalankan program yang telah kita buat atau bisa juga dengan mengklik tombol debug pada tool bar.



C. Struktur Bahasa Pemrograman C

```
<preprocessor directive>
{
    <statement>;
    <statement>;
}
```

Ketikkanlah program sederhana berikut ini:

```
#include <stdio.h>

void main ()
{
    printf ("hello ^^");
}
```

Kemudian compile. Apa hasilnya?

Penjelasan:

1. Header File

adalah berkas yang berisi prototype fungsi, definisi konstanta, dan definisi variable. Fungsi adalah kumpulan code C yang diberi nama dan ketika nama tersebut dipanggil maka kumpulan kode tersebut dijalankan.

Contoh :

stdio.h

math.h
conio.h

2. Preprosesor Directive (#include)

Preprosesor directive adalah bagian yang berisi pengikutsertaan file atau berkas-berkas fungsi maupun pendefinisian konstanta.

Contoh:

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

```
#include phi 3.14
```

3. Void

artinya fungsi yang mengikutinya tidak memiliki nilai kembalian (return).

4. Main ()

Fungsi main () adalah fungsi yang pertama kali dijalankan ketika program dieksekusi. tanpa fungsi main suatu program tidak dapat dieksekusi namun dapat dikompilasi.

5. Statement

Statement adalah instruksi atau perintah kepada suatu program ketika program itu dieksekusi untuk menjalankan suatu aksi. Setiap statement diakhiri dengan titik-koma (;).

D. Kata Kunci (Keyword)

Kata kunci-kata kunci yang terdapat di C, sebagai berikut:

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



E. IDENTIFIER

Identifier atau nama pengenal adalah nama yang ditentukan sendiri oleh pemrogram yang digunakan untuk menyimpan nilai, misalnya nama variable, nama konstanta, nama suatu elemen (misalnya: nama fungsi, nama tipe data, dll). Identifier punya ketentuan sebagai berikut :

1. Maksimum 32 karakter (bila lebih dari 32 karakter maka yang diperhatikan hanya 32 karakter pertama saja).
2. Case sensitive: membedakan huruf besar dan huruf kecilnya.
3. Karakter pertama harus karakter atau underscore (_) . selebihnya boleh angka.
4. Tidak boleh mengandung spasi atau blank.
5. Tidak boleh menggunakan kata yang sama dengan kata kunci dan fungsi.

VARIABEL

Variabel adalah identifier yang nilainya dapat berubah atau diubah selama program berjalan (dieksekusi). Pengubahnya adalah user atau proses.

- Deklarasi variabel (tipe_data nama_variabel;)

Variabel yang akan digunakan dalam program haruslah dideklarasikan terlebih dahulu. Pengertian deklarasi di sini berarti memesan memori dan menentukan jenis data yang bisa disimpan di dalamnya.

Contoh :

```
int a, b, c;
```

- Inisialisasi variabel (tipe_data nama_variabel = nilai;)

```
int a=15, b=7, c=0;
```

KONSTANTA

Konstanta adalah identifier yang nilainya tetap selama program berjalan/ dieksekusi. Cara untuk mengubahnya hanya melalui source codenya saja seperti halnya variabel, konstanta juga memiliki tipe. Penulisan konstanta mempunyai aturan tersendiri, sesuai dengan tipe masing-masing.



1. Konstanta karakter misalnya ditulis dengan diawali dan diakhiri dengan tanda petik tunggal, contohnya : 'A' dan '@'.
2. Konstanta integer ditulis dengan tanda mengandung pemisah ribuan dan tidak mengandung bagian pecahan, contohnya : -1 dan 32767.
3. Konstanta real (*float* dan *double*) bisa mengandung pecahan (dengan tanda berupa titik) dan nilainya bisa ditulis dalam bentuk eksponensial (menggunakan tanda e), contohnya : 27.5f (untuk tipe *float*) atau 27.5 (untuk tipe *double*) dan 2.1e+5 (maksudnya 2,1 x 10⁵).
4. Konstanta string merupakan deretan karakter yang diawali dan diakhiri dengan tanda petik-ganda ("), contohnya : "Pemrograman Dasar C".

Contoh :

```
#define phi 3.14
#define max_data 50
```

F. Tipe Data Dasar

Data merupakan suatu nilai yang bisa dinyatakan dalam bentuk konstanta atau variabel. Konstanta menyatakan nilai yang tetap, sedangkan variabel menyatakan nilai yang dapat diubah-ubah selama eksekusi berlangsung.

Ukuran Memori untuk Tipe Data

Tipe Data	Ukuran Memori	Kawasan
unsigned char	8 bits	0 s/d 255
char	8 bits	-128 s/d 127
short int	16 bits	-32.768 s/d 32.767
unsigned int	32 bits	0 s/d 4.294.967.295
int	32 bits	-2.147.483.648 s/d 2.147.483.647
unsigned long	32 bits	0 s/d 4.294.967.295
enum	16 bits	-2147483.648 to 2.147.483.648
long	32 bits	-2.147.483.648 s/d 2.147.483.647
float	32 bits	3,4 x 10 ⁻³⁸ s/d 3,4 x 10 ⁺³⁸
double	64 bits	1.7 x 10 ⁻³⁰⁸ to 1.7 x 10 ⁺³⁰⁸
long double	80 bits	3.4 x 10 ⁻⁴⁹³² to 3.4 x 10 ⁺⁴⁹³²
near (pointer)	32 bits	not applicable
far (pointer)	32 bits	not applicable

Untuk mengetahui ukuran memory bisa dipakai fungsi sizeof (<tipe_data>).

Catatan:

Ukuran dan kawasan dari masing-masing tipe data adalah bergantung pada jenis mesin yang digunakan (misalnya mesin 16 bit bisa jadi memberikan hasil berbeda dengan mesin 32 bit).

Untuk menampilkan hasil output dibutuhkan kode format, berikut adalah daftar kode format:

Kode format	Kegunaan
%c	Menampilkan sebuah karakter
%s	Menampilkan nilai string
%d	Menampilkan nilai decimal integer
%i	Menampilkan nilai decimal integer
%u	Menampilkan nilai decimal integer tidak bertanda (unsigned integer)
%ld	Menampilkan nilai decimal long integer
%lu	Menampilkan nilai decimal long integer tak bertanda
%li	Menampilkan nilai decimal long integer
%hu	Menampilkan nilai decimal short integer tak bertanda
%hi	Menampilkan nilai decimal short integer
%x	Menampilkan nilai heksa decimal integer
%o	Menampilkan nilai okta integer
%f	Menampilkan nilai pecahan / float
%e	Menampilkan nilai float scientific
%g	Sebagai pengganti %f atau %e tergantung yang terpendek
%lf	Menampilkan nilai pecahan double
%le	Menampilkan nilai pecahan double
%lg	Menampilkan nilai pecahan double
%p	Menampilkan suatu alamat memory untuk pointer



Contoh:

```
#include <stdio.h>

void main()
{
    int a;
    a=sizeof (long);
    printf ("ukuran tipe data long adalah %i bytes",a);
}
```

1. Coba hilangkan %i nya, apa hasilnya?
2. Coba ganti a menjadi ukuran tipe data yang lain, misalnya float dan char!

G. Karakter Escape

Karakter escape adalah karakter yang diawali dengan tanda backslash (/), yang masing-masing memiliki makna tertentu. Berikut adalah daftar karakter:

KARAKTER ESCAPE	ARTI
\a	Bunyi bel (speaker komputer)
\b	Mundur satu spasi (backspace)
\f	Ganti halaman (form feed)
\n	Ganti baris baru (new line)
\r	Ke kolom pertama baris yang sama (carriage return)
\t	Tab horizontal
\v	Tab vertical
\0	Nilai kosong (NULL)
\'	Karakter petik tunggal
\"	Karakter petik ganda
\\	Garis miring terbalik (backslash)
\?	Karakter tanda tanya
\DDD	Menyatakan sebuah karakter yang nilai ASCII nya sama dengan nilai octal DDD
\xHH	Menyatakan sebuah karakter yang nilai ASCII nya sama dengan nilai heksadesimal HH



H. STATEMENT

Apa itu statement?

Macam-macam statement:

1. Statement Kosong/ Empty Statement/ Null Statement

Statement kosong adalah statement yang hanya terdiri dari pengakhiran titik koma (;) saja, sehingga tidak ada tindakan yang akan dilakukan. Statement kosong digunakan untuk memberikan jarak waktu atau delay.

Contoh :

```
for (int i=0; i<1000; i++);
```

2. Statement Ungkapan/ Expression Statement

Statement Ungkapan adalah statement yang dibentuk dari suatu ungkapan yang diakhiri dengan titik koma (;).

Contoh :

```
luas = panjang * lebar;
scanf("%i",&luas);
printf("nilai luas persegi adalah %i",luas);
luas--;
```

3. Statement Kendali/ Control Statement

Statement kendali adalah statement yang dipakai untuk mengendalikan proses dari program, yaitu penyeleksian kondisi/ percabangan (if, case, switch) dan lompatan /perulangan (for, while, do-while, break, continue).

Contoh :

```
if (N<0)
printf("nilai N negatif");
else
printf("nilai N positif");
```

4. Statement Jamak/ Compound Statement/ Block Statement

Statement jamak adalah statement yang terdiri dari gabungan beberapa statement tunggal yang ditulis diantara tanda kurung kurawal (“{” dan “}”).

Contoh :

```
{
scanf("%f",&Panjang);
scanf("%f",&Lebar);
Luas=Panjang*Lebar;
printf("Luas = %f",Luas);
}
```



LATIHAN 1:

```
#include <stdio.h>

void main ()
{
    printf ("1.karakter escape\t2.karakter escape");
}
```

1. Coba ganti \t dengan \n atau \b!
2. Tambahkan //printf (“memberikan comment”); di bawah print (“1.karakter escape\t2.karakter escape”);

LATIHAN 2:

```
#include <stdio.h>

void main ()
{
    int a=36000;
    printf ("%i",a);
}
```

1. Jika program di atas di jalankan, bagaimanakah tampilannya? Mengapa demikian?
2. Coba ganti **int** menjadi **long int** dan **%i** menjadi **%li**. Bagaimana tampilannya?

LATIHAN 3:

```
#include <stdio.h>
void main()
{
    int angka=12345;
    printf ("%d\n", angka);
    printf ("%8d\n", angka);
    printf ("%11d\n", angka);
    printf ("%08d\n", angka);
    printf ("% -8d***\n", angka);
}
```



Perhatikan setiap perbedaan yang dihasilkan!

LATIHAN 4:

```
#include <stdio.h>
void main()
{
    float pecahan=12.3456789;
    printf("%f\n%.2f\n%.3f", pecahan, pecahan, pecahan);
}
```

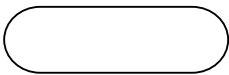
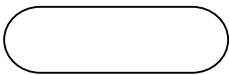
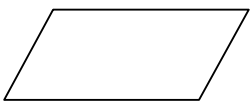
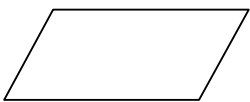
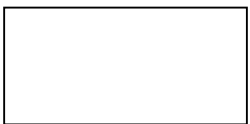
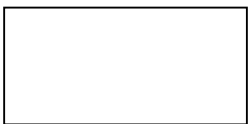
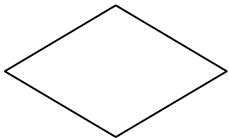
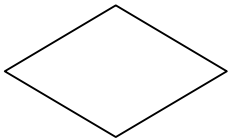
Adakah perbedaan hasil format yang dihasilkan???

LATIHAN 5:

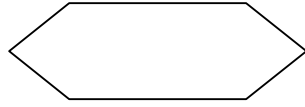
```
#include <stdio.h>
void main()
{
    char huruf='s';
    printf("Huruf anda : %c...", huruf);
    printf("\nHuruf anda : %5c...", huruf);
    printf("\nHuruf anda : %-5c...", huruf);
}
```

Perhatikan setiap perbedaan yang dihasilkan! Coba bandingkan dengan latihan 3!

I. FLOWCHART

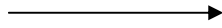
-  1.  Dipakai untuk menunjukkan awal dimulai/diakhirinya suatu proses
-  2.  Dipakai untuk memberikan inputan/ output
-  3.  Dipakai untuk menuliskan proses dalam suatu program
-  4.  Dipakai untuk menuliskan jika adanya percabangan, seperti if, case

5.



Dipakai untuk menuliskan jika adanya perulangan atau lompatan (for, while, do-while, break, continue)

6.



Dipakai untuk menunjukkan alur jalannya program

DAFTAR PUSTAKA

<http://opensource.telkomspeedy.com/forum/viewtopic.php?id=4088>

<http://buletin.melsa.net.id/okt/1020/bahasa-c.html>

<http://lecturer.ukdw.ac.id/anton/download/algoTI2.pdf>

modul-praktikum-pki-ukdw(2008)