

DASAR PEMROGRAMAN

TIM PENGAJAR PEMROGRAMAN
Departemen Ilmu Komputer IPB

Pertemuan 1

Identitas Mata Kuliah

Nama Mata Kuliah	:	Dasar Pemrograman
Kode Mata Kuliah	:	KOM200
Koordinator	:	Julio Adisantoso (JAS)
Semester	:	Pendek Ganjil 2015/2016
Prasyarat	:	KOM101 - Algoritme
Pengajar	:	JAS, AAS, HKH, VDE
PJ Praktikum	:	HKH
Jadwal Kuliah	:	Senin (15:00-16:40) Selasa (07:00-08:40)
Jadwal Praktikum	:	Ditentukan kemudian

Penentuan Nilai Akhir

- Praktikum, tugas, dan kuis : 10%
- UTS dan UAS Tertulis : 60%
- UTS dan UAS Praktikum : 30%
- Catatan: **Tidak ada ujian perbaikan**

Perangkat Perkuliahan

- Peserta: Mahasiswa Mayor ILKOM
- Bahan Materi: C How to Program - Deitel & Deitel
- Situs latihan dan praktikum: <http://apps.cs.ipb.ac.id/lx>
- Site Material Elektronik (resources) <http://julio.staff.ipb.ac.id>
- Bahasa Pemrograman : C

Tata Tertib

- Kehadiran Paling lambat 15 menit setelah dosen masuk kelas/lab
- Berpakaian sesuai ketentuan TaTib IPB
- Minimum kehadiran 70% masing-masing untuk kuliah dan praktikum (syarat untuk UAS)
- Handphone silakan dinonaktifkan (*silent*), no chatting/BBM/FB/etc saat kuliah maupun praktikum

Kejujuran Akademik

- Setiap **KECURANGAN** akan diberikan imbalan nilai 0 pada mata kuliah ini
 - Menyontek ataupun bekerja sama pada saat ujian
 - Menyalin tugas hasil pekerjaan pihak lain
 - Titip tanda tangan kehadiran
- Imbalan (sanksi) akan diberikan untuk si pelaku maupun yang memberikan kesempatan

Tips Belajar KOM200

- Sehat jasmani dan rohani, serta tidak loyo/ngantuk
- Kuliah dan praktikum dengan sungguh-sungguh dan fokus
- Aktif dan senang "ngoprek" (utak-atik). Banyak berlatih.
- High curiosity, yakni rasa penasaran yang tinggi.
- Mau belajar dan aktif mencari sumber belajar lain

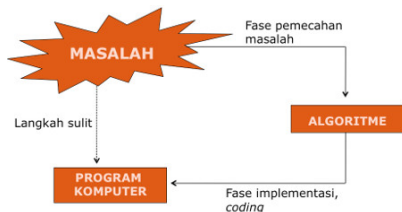


Program Komputer

- Komputer hanyalah sebuah **mesin**
- Komputer bekerja sesuai **perintah** atau **instruksi** yang diberikan
- Perintah atau instruksi yang diberikan kepada komputer untuk memecahkan suatu masalah disebut **program komputer**
- Program komputer ini memandu kerja dari komputer yang secara berstruktur disusun oleh manusia
- Bahasa yang digunakan untuk menyusun program komputer disebut **bahasa pemrograman komputer**

Membuat Program Komputer

- Pemrograman komputer harus dilakukan secara **sistematis**, dimulai dengan merancang atau mendesain langkah-langkah yang tepat yang menjelaskan jawaban dari suatu masalah
- Langkah-langkah yang disusun secara **berstruktur** dan **terurut** untuk menjawab suatu persoalan dengan menggunakan bahasa manusia inilah yang sering disebut dengan **algoritme**



Program C

Contoh Program Sederhana

```
/* C Program to print a sentence. */  
#include <stdio.h>  
int main()  
{  
    printf("Hello world!\n"); // menulis isi dalam tanda kutip  
    return 0;  
}
```

- File program ini disebut sebagai **source code**
- Setiap program C memulai eksekusi dari fungsi main()
- Di dalam main(), ada fungsi printf() yang menuliskan isi yang ada di dalam tanda kutip, yaitu "Hello world!" (\n menuliskan *newline* atau *Enter*)

Program C

Menjumlah Dua Bilangan Bulat

```
/* Menjumlah 2 bilangan bulat yang diinputkan */  
#include <stdio.h>  
int main( )  
{  
    int num1, num2, sum;  
    scanf("%d %d",&num1,&num2); // simpan bil ke variabel num1 dan num2  
    sum=num1+num2; // jumlahkan dan simpan ke sum  
    printf("Jumlah: %d\n",sum); // tulis nilai sum  
    return 0;  
}
```

- User memasukkan 2 bilangan bulat, dan akan disimpan ke variabel num1 dan num2. Dilakukan dengan fungsi scanf().
- Operator + digunakan untuk menjumlah isi variabel num1 dan num2. Operator = untuk menyimpan hasil ke variabel sum.

Tahapan Memproses Program

- Edit (menulis program di komputer) – disebut juga *coding*, dan hasilnya disebut source code

Text editor:
NotePad, vi, ...

- Preprocess atau Compile → object module (obj)
- *Link* → *executable file (exe)*

Compiler: Turbo C, Borland C, gcc, ...

- *Load*: memuat program ke memori
- *Execute*: menjalankan program

Command prompt

Turbo C,

Borland C,

Bloodshed
Dev-C++

CodeBlocks

Geany

Program Testing

- Program testing (pengujian program) merupakan langkah penting untuk menguji benar atau tidaknya program.
- Kesalahan program: (1) Syntax error, (2) Runtime error, (3) Logic error
- Sering tidak menjadi perhatian mahasiswa (cepat puas).
- Buat beberapa kemungkinan input (testcase) untuk mendapatkan kesalahan logika program (sampai tidak ditemukan)
- Khusus dalam automatic grader, output program harus sesuai (persis sama) dengan output yang dihasilkan oleh grader.

Penulisan Program yang Baik

- Menulis (mengetik kode program) adalah pekerjaan yang memerlukan perhatian khusus, karena kode program yang ditulis dengan aturan yang baik akan memudahkan untuk melakukan perubahan maksud, logika, dsb.
- Beberapa hal yang perlu diperhatikan:
 - Komentar atau keterangan
 - Indentation and spacing
 - Nama variabel, beri nama yang bermakna
 - Modular

Keywords

- Character set: sekumpulan alfanumerik (A..Z, a..z, 0..9) dan beberapa karakter khusus yang valid dalam program C.
- White space Characters: blank space, new line, horizontal tab, carriage return, form feed
- Keywords: suatu kata kunci (*reserved word*) yang digunakan dalam pemrograman, yang sudah memiliki makna khusus dan tidak dapat diubah oleh user.
- Contoh: `int sum;`
`int` adalah keyword yang menunjukkan bahwa `sum` bertipe integer (bilangan bulat)

Keywords dalam C (ANSI)

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

- Borland C: asm, far, interrupt, pascal, near, huge, cdecl

Identifiers

- Identifiers: nama yang diberikan ke dalam entitas program C, seperti variabel, fungsi, struktur, dsb.
- Identifiers dibuat dengan nama unik untuk mengidentifikasi entitas tertentu selama program dieksekusi.
- Contoh: `int sum;` (`sum` adalah identifier)
- Aturan nama identifiers:
 - Identifier hanya dapat disusun dari kombinasi huruf (besar dan kecil), angka, dan underscore ”_”.
 - Harus dimulai dengan huruf atau underscore. (underscore tidak disarankan).
 - Tidak ada aturan panjang identifiers, namun yang dikenali oleh kompilator hanya 31 karakter paling depan.

Variabel

- Variabel adalah lokasi memori komputer untuk menyimpan data. Lokasi ini diberikan dengan menggunakan nama unik yang disebut sebagai identifier.
- Nama variabel hanyalah representasi simbolik bagi lokasi memori komputer. Contoh: `sum`, `num1`, `count`, dsb.
- Dalam program C, nama variabel harus dideklarasikan terlebih dahulu sebelum digunakan.
- Format deklarasi: `tipe_data nama_variabel;`
- Variabel bersifat: (1) Lokal, apabila dideklarasikan dan hanya dikenali di dalam suatu fungsi, (2) Global, dideklarasikan di luar fungsi dan dikenali oleh seluruh fungsi yang ada

Konstanta

- Konstanta (*constants*) adalah entitas nilai yang tidak dapat diubah selama eksekusi program.
- Contoh: 1, 2.5, "Pemrograman itu mudah.", dsb.
- Dalam C, konstanta dikelompokkan menjadi:
 - Integer constants
 - Floating-point constants
 - Character constants
 - String constants
 - Enumeration constants (dibahas pada pertemuan lanjut)

Integer constants

- Integer constants (konstanta bilangan bulat) merupakan konstanta numerik (berhubungan dengan angka) tanpa fraksi maupun bagian eksponensial.
- Ada 3 jenis dalam C: (1) bilangan desimal (basis 10), bilangan oktal (basis 8), dan heksadesimal (basis 16).
- Digit bilangan:
 - Desimal: 0 1 2 3 4 5 6 7 8 9
 - Oktal: 0 1 2 3 4 5 6 7
 - Heksadesimal: 0 1 2 3 4 5 6 7 8 9 a b c d e f

Contoh Konstanta Integer

Desimal: 0, -9, 22 dsb

Oktal: 021, 077, 033 dsb

Heksadesimal: 0x7f, 0x2a, 0x521 dsb

Floating-point constants

- Floating point constants (konstanta bilangan riil) merupakan konstanta numerik yang memiliki bentuk fraksi (pecahan) atau eksponensial.
- Contoh: -2.0, 0.0000234, -0.22E-5
- Catatan: E-5 menunjukkan 10^{-5} , sehingga -0.22E-5 = -0.0000022

Character constants

- Character constants (konstanta karakter) adalah konstanta yang diapit oleh tanda kutip tunggal.
- Contoh: 'a', ", '5', 'F', '\n' dsb
- Escape sequences adalah karakter khusus yang tidak dapat dituliskan dalam bentuk konstanta karakter biasa, seperti newline (Enter), tab, dsb.
- Escape sequences: \b (backspace), \f (Form feed), \n (Newline), \r (Return), \t (Horizontal tab), \v (Vertical tab), \\ (Backslash), \' (Single quotation mark), \" (Double quotation mark), \? (Question mark), \0 (Null character)

String constants

- String constants (konstanta string) adalah konstanta yang diapit oleh tanda kutip dobel (double quotes).
- Contoh:

```
"BOGOR"      // string BOGOR  
""           // string kosong (null string)  
"      "     // string dgn 6 spasi  
"x"         // string dgn karakter tunggal  
"Hello world!\n" // string dengan newline
```

Tipe Data

- Tipe data dalam C:
 - Tipe data dasar: (1) Integer, (2) Floating, (3) Character
 - Tipe data turunan: (1) Array, (2) Pointer, (3) Structures, (4) Enumeration

- Sintaks deklarasi variabel:

```
tipe_data nama_variabel;
```

- Contoh:

```
int num1, num2;  
char ch;  
float x;  
double rataan;
```


Tipe Data Integer

- Menggunakan kata kunci `int`

- Contoh:

```
int num1, num2, sum=0;
```

dapat juga ditulis sebagai:

```
int num1;
```

```
int num2;
```

```
int sum=0;
```

- Ukuran memori: 2 bytes (PC lama) atau 4 bytes (32 bit)
- Kisaran nilai (32 bit): -2^{31} sampai dengan $2^{31} - 1$ atau senilai 2147483647

Tipe Data Floating

- Dapat menangani bilangan riil seperti 2.34, -9.382 dsb
- Menggunakan kata kunci `float` untuk single precision (4 bytes atau sampai 6 digit) , atau `double` untuk double precision (8 bytes atau sampai 14 digit)

- Contoh:

```
float var1=22.442e2;  
double var2;
```

Qualifiers

- Qualifiers mengubah makna suatu tipe data dasar menghasilkan tipe data yang baru.
- Ada 4 jenis:
 - Size: mengubah ukuran. Contoh: `long int nilai;`
 - Sign: hanya menangani nilai 0 dan bilangan positif. Contoh: `unsigned int a;`
 - Constant: membuat variabel tersebut tidak dapat diubah nilainya selama eksekusi program. Contoh: `const int p=20;`
 - Volatile (dibahas pada pertemuan lanjut)

Input/Output Integer

Integer

```
#include<stdio.h>
int main()
{
    int c=5;
    printf("Nilai=%d\n",c);
    return 0;
}
```

Output

Nilai=5

Input/Output Integer

Integer

```
#include<stdio.h>
int main() {
    int c;
    scanf("%d",&c);
    printf("Nilai=%d",c);
    return 0;
}
```

Output

```
10
Nilai=10
```

Input/Output Float

Float

```
#include <stdio.h>
int main() {
    float a;
    scanf("%f", &a);
    printf("Nilai=%f dan %.1f", a, a);
    return 0;
}
```

Output

```
23.48
Nilai=23.280000 dan 23.5
```

I/O Format

Data type	Format
Signed Char or Unsigned Char	%c
Signed int	%d
Unsigned int	%u
Signed short int	%hd
Unsigned short int	%hu
Signed long int	%ld
Unsigned long int	%lu
Signed long long int	%lld
Unsigned long long int	%llu
Float	%f
Double	%lf
Long Double	%Lf