

Pengumuman

- Ada sekitar 50 mahasiswa yang belum bergabung di: <http://www.facebook.com/groups/algor.ipb>
- Ada koreksi *testcase* pada soal 1.2.05 Fungsi Matematika.
- Masih ada sekitar 150 mahasiswa yang sama sekali belum mendapatkan nilai di Kontes PR.
- Harap segera mengerjakannya sebagai latihan awal membuat program yang sangat mudah.
- Soal PR dapat bertambah sebelum beralih ke kelompok soal berikutnya.

ALGORITME DAN PEMROGRAMAN

Kuliah #2

- Struktur program C
- Tipe data
- Masukan / Keluaran

Bahasa Pemrograman

3

- Bahasa pemrograman adalah notasi yang digunakan untuk menulis program (komputer) dengan aturan tertentu.
 - **bahasa mesin:** biner; tergantung pada prosesor yang digunakan; tidak perlu penerjemahan
 - **bahasa tingkat rendah:** kode-kode (mis. MOV, ADD, DIV); tergantung pada prosesor yang digunakan; diterjemahkan ke bahasa mesin dengan **assembler**
 - **bahasa tingkat tinggi:** dekat ke bahasa manusia; diterjemahkan ke bahasa mesin dengan **compiler** (menghasilkan *executable*) atau **interpreter** (tidak menghasilkan *executable*)

Contoh: mencetak huruf A

4

- | | |
|--|---|
| <ul style="list-style-type: none"> □ Bahasa mesin: <pre>1011 0100 0000 0010 1011 0010 0100 0001 1100 1101 0010 0001 1100 1101 0010 0000</pre> □ Bahasa <i>assembly</i> <pre>MOV AH, 02 MOV DL, 41H INT 21H INT 20H</pre> | <ul style="list-style-type: none"> □ Bahasa C <pre>#include <stdio.h> int main() { printf("A"); return 0; }</pre> |
|--|---|

Bahasa C

5

- Bahasa C disusun berdasarkan dua bahasa terdahulu, yaitu BCPL dan B.
- C pertama kali dipublikasikan oleh Kernighan dan Ritchi pada tahun 1978
- Standar C (ISO/IEC 9899:1999)
 - ▣ Mata kuliah ini menggunakan C99
 - ▣ Standar terbaru C11

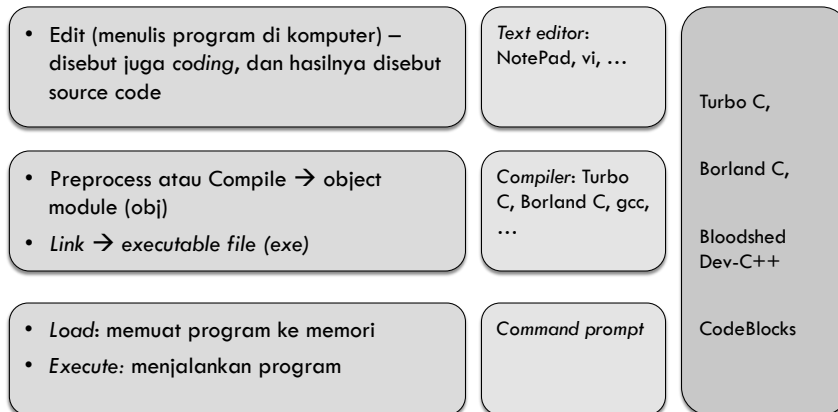
Perbedaan C99 dengan Standar Sebelumnya

6

- Komentar satu baris dengan `//` seperti C++
- *Inline function*
- Tipe data baru:
 - ▣ `long long int`: bilangan bulat sangat besar
 - ▣ `complex`: bilangan kompleks (pakai `complex.h`)
 - ▣ `bool`: true/false (pakai `stdbool.h`)
- *Variable-length array* (di bab *array*)

Tahapan pembuatan dan pemrosesan program C

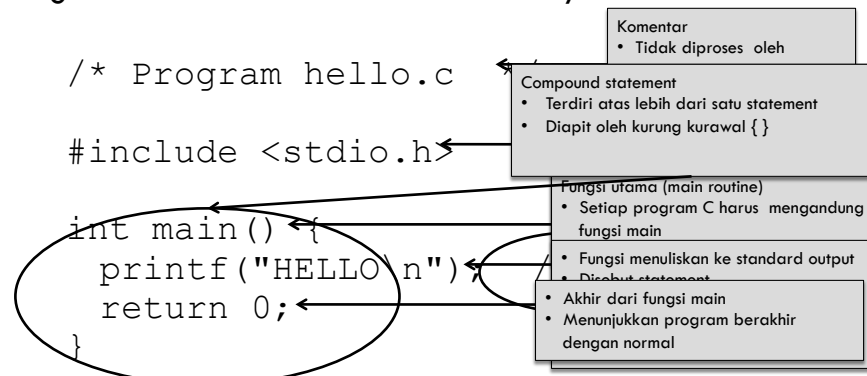
7



Struktur program C (contoh 1)

8

Program menuliskan teks "HELLO" ke layar.



Struktur program C (contoh 2)

9

Program menjumlahkan dua bilangan bulat.

```

/* Program jumlah */
#include <stdio.h>
int main(){
    int a, b, jumlah;
    scanf("%d %d", &a, &b);
    jumlah = a + b;
    printf("%d\n", jumlah);
    return 0;
}

```

Deklarasi variabel

Fungsi membaca data

- Dua jenis argumen, yaitu format dan variabel
- Format menunjukkan tipe data yang dibaca, dalam hal ini adalah dua nilai desimal bulat (%d %d).
- Variabel yang menerima nilai data

Pernyataan penugasan

- Memberikan nilai a+b (sebelah kanan tanda =) ke variabel jumlah (sebelah kiri tanda =)

Fungsi menulis keluaran

- Ada dua argumen: format dan variabel (atau ekspresi)
- Mirip fungsi scanf, tetapi argumen kedua adalah ekspresi

Data

10

- Setiap program umumnya mempunyai data, dan setiap data memiliki nilai dan tipe tertentu.
- Suatu nilai di dalam program dituliskan dalam bentuk *literal constant* (*literal*: hanya berupa nilai, *constant*: tidak berubah).
- Setiap literal mempunyai tipe, misalnya: 3 bertipe *integer*, 4.15 bertipe *floating point*.
- Nilai *literal* bersifat *nonaddressable*, yaitu tidak memiliki alamat dalam memori komputer.

Literal Constant

11

- Literal integer constant – bilangan bulat
 - ▣ Desimal : 24, 103, -5, ...
 - ▣ Oktal : 024, 0103, ...
 - ▣ Heksadesimal : 0x24, 0x103, ...
- Literal floating point constant – bilangan riil
 - ▣ Desimal floating point : 3.14, -90.254, ...
 - ▣ Ekspensial : 1.0E-3
- Literal character constant – kode ASCII
 - ▣ Printable character : 'a', '4', '0', '*', ...
 - ▣ Escape sequence : '\n', '\t', '\r', '\0', '\a', '\\', '\"'
- Literal string constant – beberapa character
 - ▣ Contoh : "ipb", "", "5", "a", "HELLO\n", ...

Tipe data

12

- Jenis bilangan: bulat/real/kompleks
- Rentang bilangan: *short, long, dll*
 - ▣ Nilai minimum dan maksimum tergantung pada *compiler* yang digunakan (lihat *limits.h*)
 - Rentang nilai tipe **int** berbeda-beda
 - Patokan rentang: $\text{short} \leq \text{int} \leq \text{long} \leq \text{long long}$
 - ▣ Rentang pada slide ini berdasarkan nilai di LX
 - ▣ Rentang lebih besar: lebih fleksibel, tapi lebih boros memori
- Dukungan terhadap nilai negatif: *signed – unsigned*

	Data Type	Range	Keyword
	Signed Char	-128 to 127	char
	Unsigned Char	0 to 255	unsigned char
13	Signed int	-2147483648 to 2147483647	int
	Unsigned int	0 to 4294967295	unsigned
	Signed short int	-32768 to 32767	short
	Unsigned short int	0 to 65535	unsigned short
	Signed long int	-2147483648 to 2147483647	long int
	Unsigned long int	0 to 4294967295	unsigned long int
	Signed long long int	-9223372036854775808 to 9223372036854775807	long long int
	Unsigned long long int	0 to 18446744073709551615	unsigned long long int
	Float	3.4 e-38 to 3.4 e+38	Float
	Double	1.7e-308 to 1.7e+308	double
	Long Double	3.4 e-4932 to 3.4 e+4932	long double

Variabel

14

- *Literal constant* disimpan ke dalam suatu variabel, agar dapat diakses di dalam program.
- Variabel merupakan suatu identifier, suatu identitas yang dibuat sendiri dengan aturan:
 - ▣ Terdiri atas satu atau lebih karakter
 - ▣ Dimulai dengan huruf, dan dapat diikuti oleh *alphanumeric* atau *underscore* (`_`)
 - ▣ Dapat dimulai dengan *underscore*, tetapi umumnya digunakan oleh *library C*
- Contoh penamaan *identifier*:
 - ▣ Benar : `n`, `x1`, `jumNegatif`, ...
 - ▣ Salah : `1x`, `jumlah bilangan`, ...

Deklarasi Variabel

15

- Setiap variabel yang digunakan dalam program C harus dideklarasikan dengan menentukan tipe variabel yang bersangkutan.
- Tipe variabel menunjukkan tipe data yang disimpan.
- Format:


```
keyword v1, v2, ..., vn;
```
- Contoh:


```
int jumlah;
int n, tahun;
double rataa, tinggiBadan;
```

Format untuk keluaran dan masukan

16

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

Menuliskan keluaran :: printf

17

- Format:

```
printf(ekspresi);
printf("format", ekspresi);
```

- Contoh (apa keluarannya?):

```
int a=5; b=10; c=15;
float x=12.56;
printf("Keluaran Program\n");
printf("%d-%d=%d\n", b, a, b-a);
printf("Nilai x adalah %.2f\n", x);
printf("a=%d\nb=%d\nc=%d", a, b, c);
```

```
1 #include <stdio.h>
2
3 int main () {
4     int a=5, b=10, c=15;
5     float x=12.56;
6     printf("Output Program\n");
7     printf("%d-%d=%d\n", b, a, b-a);
8     printf("Nilai x adalah %.2f\n", x);
9     printf("a=%d\nb=%d\nc=%d", a, b, c);
10    return 0;
11 }
```

```
Output Program
10-5=5
Nilai x adalah 12.56
a=5
b=10
c=15_
```

Nilai di Luar Rentang atau Beda Tipe?

19

```
#include <stdio.h>
int main () {
    short int x=5.3, y=40000;
    float z=y;
    printf("Keluaran Program\n");
    printf("%d+%d=%d\n", x, y, x+y);
    printf("Nilai z adalah %.2f\n", z);
    printf("x=%hd\ny=%hd\nz=%f", x, y, z);
    return 0;
}
```

Keluaran? Lakukan tracing. Gunakan project di Code::Blocks agar debugging aktif.

Membaca masukan :: scanf

20

□ Format:

```
scanf("format", &variabel);
```

□ Contoh

(bagaimana contoh data yang dibaca?):

```
int a, b;
float x;
scanf("%d", &a);
scanf("%d%d%f", &a, &b, &x);
```

Latihan 1

21

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

Jika diberi masukan **35 200**, apa keluaran program tersebut? Apa sebenarnya yang dilakukan program tersebut? Trace-lah program tersebut.

Latihan 2 :: Menghitung luas segitiga

22

- Masalah
Program menuliskan nilai luas segitiga yang memiliki alas dan tinggi tertentu, dengan format dua digit di belakang koma.
- Perumusan masalah
Masukan program adalah dua nilai *floating point*, yaitu alas dan tinggi. Misalkan alas = a , dan tinggi = t , maka dapat dihitung luas = $0.5 a t$

Latihan 2 :: Menghitung luas segitiga

23

□ Algoritme

```
procedure luasSegitiga {
    read a, t
    luas = 0.5*a*t
    print luas
}
```

Latihan 2 :: Menghitung luas segitiga

24

□ Program C

```
/* Program luas segitiga */
#include <stdio.h>
int main() {
    float a, t, luas;
    scanf("%f %f", &a, &t);
    luas = 0.5*a*t;
    printf("%.2f\n", luas);
    return 0;
}
```

Trace-lah program ini.

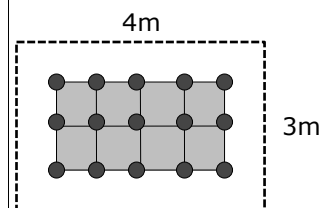
Latihan 3 ::

Menghitung berat benih jagung

25

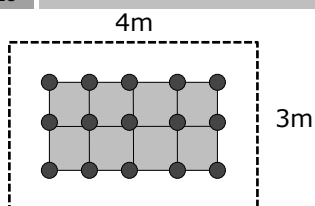
Pak Jalal memiliki kebun berbentuk segiempat dengan panjang dan lebar dalam satuan meter dan selalu berupa bilangan bulat. Kebun akan ditanami jagung dengan jarak tanam masing-masing 0.5 meter membentuk segiempat. Jagung selalu ditanam 1 meter dari batas pinggir kebun, tidak pernah ditanam di batas pinggirnya. Di setiap lubang tanam, selalu dimasukkan dua biji benih jagung. Setiap benih memiliki berat yang sama, yaitu 0.15 gram. Bantulah Pak Jalal menghitung berapa berat benih jagung yang dibutuhkan, dengan membuat program C. Berat dituliskan dalam satuan gram dengan dua digit di belakang koma.

Contoh, jika panjang=4m, dan lebar=3m, maka dibutuhkan benih jagung seberat 4.50 gram ($15 \times 2 \times 0.15$)



Analisis Masalah

26



1. Panjang yang digunakan: $p-2 = 4-2 = 2$
2. Lebar yang digunakan: $l-2 = 3-2=1$
3. Tempat biji yang ditanam :
 $p \rightarrow (2 : 0.5) + 1 = 5$
 $l \rightarrow (1 : 0.5) + 1 = 3$
4. Banyaknya lubang tanam: $5 * 3 = 15$
5. Berat total biji = $15 * 2 * 0.15 = 4.50$