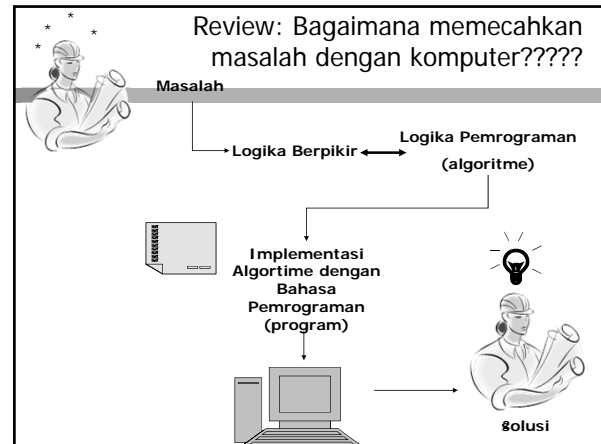


ALGORITME DAN PEMROGRAMAN

Kuliah #3

- Implementasi algoritme
- Struktur program C
- Elemen program



Review: Algoritme dan Program

- Program Komputer:
 - ▣ Instruksi terstruktur yang disusun dan diberikan kepada komputer untuk dilaksanakan dengan menggunakan bahasa pemrograman tertentu, misalnya C, Pascal, Basic, dsb.
 - ▣ Implementasi dari algoritme yang telah disusun sebelumnya.

Bahasa Pemrograman

- Bahasa pemrograman adalah notasi yang digunakan untuk menulis program (komputer) dengan aturan tertentu.
- Bahasa ini dibagi menjadi tiga tingkatan yaitu
 - ▣ bahasa mesin,
 - ▣ bahasa tingkat rendah dan
 - ▣ bahasa tingkat tinggi.

Bahasa mesin (*machine language*)

- Bahasa mesin berupa *micro-instruction* atau *hardwire*
- Programnya sangat panjang dan sulit dipahami
- Sangat tergantung pada arsitektur mesin
- Prosesnya sangat cepat dan tidak perlu interpreter atau penterjemah

Bahasa tingkat rendah (*low level language*)

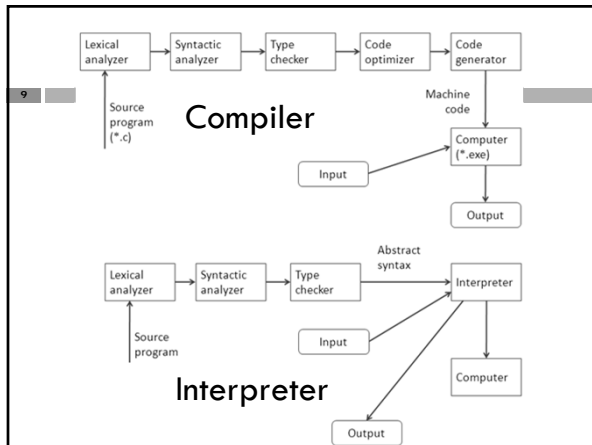
- Bahasa tingkat rendah berupa macroinstruction (*assembly*) sehingga sering disebut sebagai bahasa rakitan (*assembly language*)
- Bahasa tingkat rendah tergantung pada arsitektur mesin
- Programnya panjang dan sulit dipahami walaupun prosesnya cepat
- Jenis bahasa tingkat ini perlu penterjemah berupa assembler

Bahasa tingkat tinggi (high level language)

- Bahasa tingkat tinggi lebih menyerupai bahasa manusia sehingga mudah dipahami
- Tidak tergantung pada arsitektur mesin tetapi memerlukan penterjemah berupa compiler atau interpreter
- Terdiri dari banyak model
- Contoh: C, Java, Pascal, dsb

Compiler dan Interpreter

- Bahasa program dianalisis dan selanjutnya diterjemahkan ke dalam bentuk yang dapat dipahami mesin.
- Proses penterjemahan:
 - Dijalankan oleh komputer (compiler) – real machine → compiling
 - Dijalankan oleh interpreter – software yang mensimulasikan virtual machine dan menjalankan dalam real machine → interpreting



Bahasa C

- Bahasa C disusun berdasarkan dua bahasa terdahulu, yaitu BCPL dan B.
- BCPL dikembangkan tahun 1967 oleh Martin Richards.
- Ken Thompson memodelkan beberapa fitur di dalam bahasa B bersama rekan-rekannya yang menggunakan bahasa BCPL untuk membuat sistem operasi UNIX yang pertama di Bell Laboratories pada tahun 1970 dengan menggunakan komputer DEC PDP-7.
- Bahasa C dikembangkan lebih lanjut dari bahasa B oleh Dennis Ritchi di Bell Laboratories dan pertama kali diimplementasikan dalam komputer DEC PDP-11 pada tahun 1972 yang menggunakan sistem operasi UNIX.
- C pertama kali dipublikasikan oleh Kernighan dan Ritchi pada tahun 1978

Tahapan pemrosesan program C

• Edit (menulis program di komputer) -- coding, dan hasilnya disebut source code	Text editor: NotePad, vi, ...	Turbo C, Borland C, Bloodshed Dev-C++
• Preprocess atau Compile → object module (.obj) • Link → executable file (.exe)	Compiler: Turbo C, Borland C, gcc, ...	
• Load • Execute -- running program	Command prompt	

Struktur program C (contoh 1)

Program menuliskan teks "HELLO" ke layar.

```

/* Program hello.c */
#include <stdio.h>

main()
{
    printf("HELLO\n");
    return 0;
}
    
```

- Komentar**: Tidak diproses oleh compiler
- Compound statement**: Terdiri atas lebih dari satu statement, Diapit oleh kurung kurawal { }
- Fungsi utama (main routine)**: Setiap program C harus mengandung fungsi main
- Fungsi menuliskan ke standard output**: Disebut statement
- Akhir dari fungsi main**: Menunjukkan program berakhir dengan benar

Struktur program C (contoh 2)

13

Program menjumlahkan dua bilangan bulat.

```

/* Program jumlah */
#include <stdio.h>
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).
- Membaca ke variabel a dan b.

Pernyataan penugasan

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

Fungsi menulis output

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

Data

14

- Setiap program umumnya mempunyai data, dan setiap data memiliki tipe tertentu.
- Suatu nilai data 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

15

- 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, ...
 - Eksponensial : 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

16

Data Type	Range	Keyword
Char or Signed Char	-128 to 127	char
Unsigned Char	0 to 255	unsigned char
Int or Signed int	-32768 to 32767	int
Unsigned int	0 to 65535	unsigned
Short int or Signed short int	-128 to 127	short
Unsigned short int	0 to 255	unsigned short
Long int or signed long int	-2147483648 to 2147483647	long int
Unsigned long int	0 to 4294967295	unsigned 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

17

- 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

18

- 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 ratauan, tinggiBadan;
```

Format untuk output dan input

19

Data type	Format
Char or Signed Char or Unsigned Char	%c
Int or Signed int	%d
Unsigned int	%u
Short int or Signed short int	%hd
Unsigned short int	%hu
Long int or signed long int	%ld
Unsigned long int	%lu
Float or Double	%f
Long Double	%lf

Menuliskan output :: printf

20

- Format:


```
printf(ekspresi);
printf("format", ekspresi);
```
- Contoh (apa outputnya?):


```
int a=5; b=10; c=15;
float x=12.56;
printf("Output 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);
```

```
#include <stdio.h>
main()
{
    int a=5, b=10, c=15;
    float x=12.56;
    printf("Output 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);
    return 0;
}
```

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

Membaca input :: scanf

22

- 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

23

```
#include <stdio.h>
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 input 35 200, apa output program tersebut?
Apa sebenarnya yang dilakukan program tersebut?

Latihan 2 :: Menghitung luas segitiga

24

- Masalah

Program menuliskan nilai luas segitiga yang memiliki alas dan tinggi tertentu, dengan format dua digit di belakang koma.
- Perumusan masalah

Input 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

25

□ Algoritme

```

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

```

Latihan 2 :: Menghitung luas segitiga

26

□ Program C

```

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

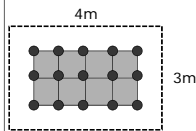
```

Latihan 3 :: Menghitung berat benih jagung

27

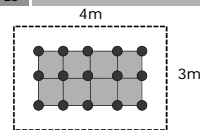
Pak Jalal memiliki kebun berbentuk segiempat dengan panjang dan lebar dalam satuan meter dan selalu berupa bilangan bulat. Kebun akan ditanami jagung dengan jarak tanam masing-masing 0.5 meter membentuk segiempat. Jagung selalu ditanam 1 meter dari batas pinggir kebun, tidak pernah ditanam di batas pinggirnya. 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 x 2 x 0.15)



Analisis Masalah

28



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 = $15 * 2 * 0.15 = 4.50$