

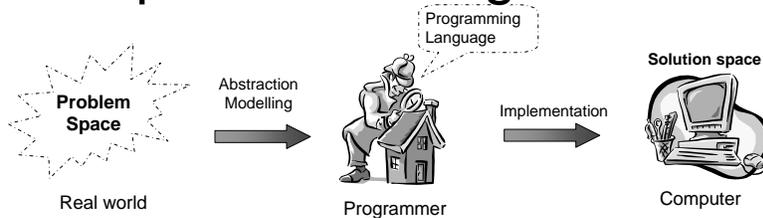
Pemrograman dan Bahasa Pemrograman

Algoritma dan Pemrograman

Tahar Agastani

Teknik Informatika UIN - 2008

Apakah Pemrograman?



- **Pemrograman** adalah proses penulisan, pengujian, dan pemeliharaan (*maintainance*) program (*source code*)
- **Program** adalah implementasi teknis Algoritma yang ditulis dalam bahasa pemrograman tertentu sehingga dapat dilaksanakan oleh komputer.
- Dalam *software engineering*, pemrograman adalah salah satu fase proses pengembangan software yang sering disebut *fase implementasi*.

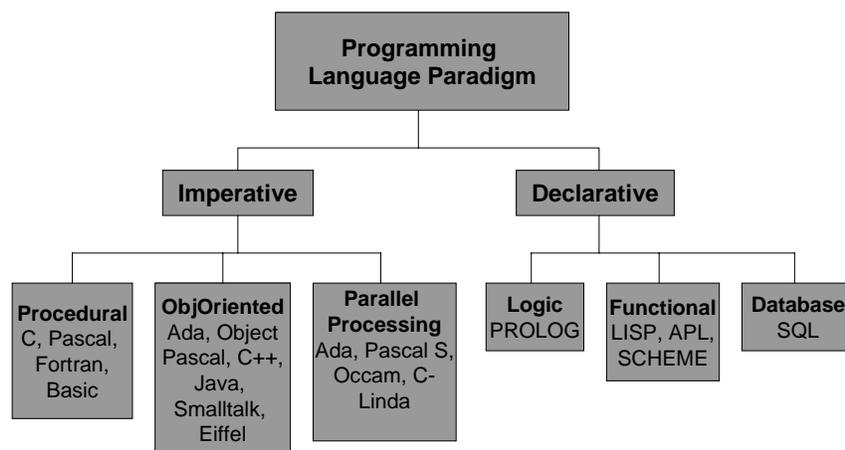
Paradigma Pemrograman

- Definisi:
 - Gaya (style) pemrograman yang berkaitan dengan pemecahan masalah yang dirumuskan dalam bahasa pemrograman.
- Paradigma pemrograman dikelompokkan menjadi *imperative* dan *declarative programming*.
 - *Imperative program*: menyebutkan secara eksplisit algoritma untuk mencapai suatu tujuan (**how to create it**)
 - *Declarative program*: menyebutkan secara eksplisit tujuan yang ingin dicapai (pernyataan) dan meninggalkan implementasi algoritmanya (**what something is like**)

TI - Algoritma dan Pemrograman

3

Paradigma Pemrograman



TI - Algoritma dan Pemrograman

4

Pemrograman Prosedural dan Berorientasi Objek

- Pemrograman Prosedural
 - Berdasarkan urutan-urutan, sekuensial
 - Program adalah suatu rangkaian prosedur untuk memanipulasi data. Prosedur merupakan kumpulan instruksi yang dikerjakan secara berurutan.
 - Harus mengingat prosedur mana yang sudah dipanggil dan apa yang sudah diubah.
- Pemrograman Berorientasi Objek
 - Pemrograman berdasarkan prinsip objek, dimana objek memiliki data/variabel/property dan method/event/prosedur yang dapat dimanipulasi

TI - Algoritma dan Pemrograman

5

Perkembangan Bahasa Pemrograman

- Bahasa Mesin
 - Bahasa yang berisi kode-kode mesin yang hanya dapat diinterpretasikan langsung oleh mesin komputer
 - *Keuntungan*: Eksekusi cepat
 - *Kerugian*: Sangat sulit dipelajari manusia
- Bahasa Assembly
 - Bahasa simbol dari bahasa mesin
 - *Kelebihan*: Eksekusi cepat, masih bisa dipelajari daripada bahasa mesin, file kecil
 - *Kekurangan*: Tetap sulit dipelajari, program sangat panjang.

TI - Algoritma dan Pemrograman

6

- Bahasa Tingkat Tinggi (3rd GL)
 - Bahasa pemrograman yang lebih tinggi dari bahasa assembly
 - *Keuntungan:*
 - Mudah dipelajari
 - Mendekati permasalahan yang akan dipecahkan
 - Kode program pendek
 - *Kerugian:* Eksekusi lambat
 - Contoh: C, C++, Pascal, Basic, dan java
- Bahasa yang berorientasi pada masalah spesifik (4th GL)
 - Digunakan langsung untuk memecahkan masalah tertentu
 - Contoh: SQL untuk database

TI - Algoritma dan Pemrograman

7

Translator

- Assembler
 - Source code: bahasa assembly, Object code: bahasa mesin
- Interpreter
 - Source code: bahasa scripting seperti php, ASP, javascript, Perl.
 - Tidak menghasilkan object code hanya translasi internal
- Kompiler
 - Source code: bahasa program spt C, C++, pascal
 - Object code: bahasa assembly

TI - Algoritma dan Pemrograman

8

Langkah-langkah Pemrograman

1. Mendefinisikan masalah
 - Ini merupakan langkah pertama yang sering dilupakan orang.
2. Menentukan solusi
 - Bila terlalu kompleks dipecah jadi modul2
3. Memilih algoritma
 - Pilihlah algoritma yang benar-benar sesuai dan efisien untuk permasalahan tersebut
4. Menulis program
 - Pilihlah bahasa yang mudah dipelajari, mudah digunakan
5. Menguji program
6. Menulis dokumentasi
7. Memelihara program

Ukuran Kualitas Program

- Sebuah program harus dapat mengerjakan tugasnya dengan benar.
- Dapat melakukan tugasnya secara cepat.
- Tidak menggunakan banyak sumber daya (*processor time, memory, disk capacity, network capacity*).
- Program mudah dibaca dan dimengerti.
- Adanya kesalahan tidak mempengaruhi bagian program (*error locality*).
- Mudah pemeliharaannya.
- Sebuah program harus independent dan tidak bergantung pada program lain.
- Mempunyai dokumentasi yang baik.
- Proses pembuatan program selesai tepat waktu.

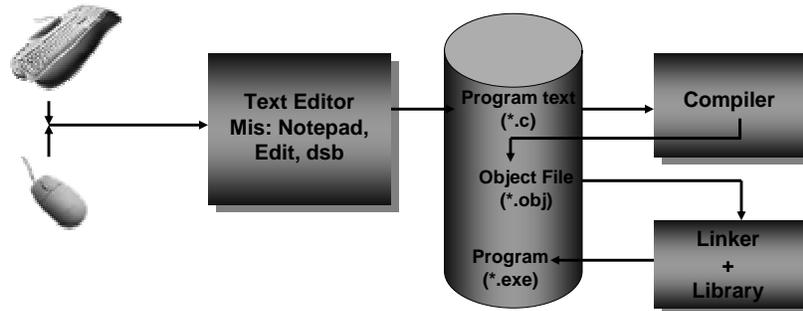
Pengantar Bahasa C

- Mengapa C
 - **Flexibility** : mendekati *low level language* namun mudah dimengerti.
 - **Portability** : dipakai mulai dari komputer mikro sampai superkomputer.
 - Bahasa yang banyak digunakan dalam ilmu komputer untuk membuat O/S (*Operating System*) dan program aplikasi, dll.
 - Didukung oleh banyak pustaka (*libraries*).

▪ Sejarah Bahasa C

- Akar dari bahasa C adalah dari bahasa BCPL yang dikembangkan oleh *Martin Richards* pada tahun 1967.
- Kemudian *Ken Thompson* mengembangkan bahasa ini menjadi bahasa B pada tahun 1970.
- Selanjutnya *Dennis Ritchi* mengembangkan dari bahasa B menjadi bahasa C di *Bell Laboratories Inc. (AT&T Bell Laboratories)* sekitar tahun 1970-an.
- Bahasa C pertama kali digunakan di komputer *Digital Equipment Corporation PDP-11* yang menggunakan sistem operasi UNIX.
- Kepopuleran bahasa C membuat versi-versi dari bahasa ini banyak dibuat untuk komputer mikro (PC).
- Pada tahun 1983, ANSI (*American National Standards Institute*) membentuk suatu komite (ANSI Committee X3J11) yang menetapkan standar ANSI untuk bahasa C.

Penanganan text program dalam komputer



TI - Algoritma dan Pemrograman

13

Struktur Bahasa C

- C adalah salah satu bahasa pemrograman yang terstruktur.
- Bahasa C terdiri dari fungsi-fungsi.
- Tidak ada perbedaan antara prosedur dengan fungsi.
- Setiap program C mempunyai satu fungsi dengan nama “**main**” (program utama).
- Program akan dieksekusi dimulai dari statement pertama pada fungsi “main” tersebut.
- Huruf besar dengan huruf kecil diartikan berbeda (*case-sensitive*).
- Setiap perintah (*statement*) diakhiri dengan semi-colon (titik koma ;).

TI - Algoritma dan Pemrograman

14

Struktur Bahasa C

- Format penulisan fungsi secara umum:

```
#include <header_file>
deklarasi global_variabel;
return_type function_name(parameter)
{
    deklarasi lokal_variabel;
    statements;
    return (value);
}
```

- Fungsi bisa diletakkan di atas atau di bawah fungsi “main”.
- Fungsi yang tidak mengembalikan **nilai tipe**-nya (*return-type*) adalah “**void**”.

TI - Algoritma dan Pemrograman

15

Struktur Bahasa C

- Format penulisan fungsi yang tidak mengembalikan nilai:

```
void function_name (parameter_list)
{
    deklarasi lokal_variabel;
    statements;
}
```

- Contoh Program C:

```
1 /* Prog_1.C-Menampilkan "Hello World !" di monitor */
2 #include <stdio.h>
3 void main()
4 {
5     printf ( "Hello World !");
6 }
```

TI - Algoritma dan Pemrograman

16

Struktur Bahasa C

- Apabila program tersebut di atas dikompilasi dan dijalankan akan menghasilkan tampilan di monitor sbb:

```
Hello World !
```

Keterangan program:

- Penomoran di depan setiap baris program C sebenarnya tidak ada, ini hanya membantu keterangan.
- Baris 1: `/* Prog_1.C-Menampilkan "Hello World !" di monitor */` - merupakan komentar, yang digunakan agar program lebih mudah dibaca dan dimengerti. Komentar tidak ikut dikompilasi. Setiap komentar diapit oleh tanda `/*` dan `*/`

TI - Algoritma dan Pemrograman

17

Struktur Bahasa C

Keterangan program:

- Baris 2: `#include <stdio.h>` - memasukkan header file yang mempunyai nama **stdio.h**, dimana dalam header file tersebut terdapat perintah **printf()**, yang digunakan untuk menampilkan semua konstanta string yang diapit oleh tanda `"` dan `"`. Dalam program ini menampilkan kalimat: Hello World !.
- Baris 3: `void main()` - mendeklarasikan fungsi utama tanpa mengembalikan nilai karena tipe fungsinya void.
- Baris 4: `{` - menyatakan awal kumpulan perintah dari fungsi utama.
- Baris 5: `printf ("Hello World !");` - perintah yang digunakan menampilkan kalimat: Hello World ! di monitor.
- Baris 6: `}` - menyatakan akhir kumpulan perintah dari fungsi utama.

TI - Algoritma dan Pemrograman

18