

Konsep Pemrograman

1. Pengenalan Pemrograman C

Umi Sa'adah

Entin Martiana Kusumaningtyas

Tri Hadiah Muliawati



Politeknik Elektronika Negeri Surabaya
Departemen Teknik Informatika dan Komputer

Overview

- Sejarah dan Ruang Lingkup C
- Proses Kompilasi dan Linking dalam C
- Struktur Penulisan Program C

Sejarah C

- Akar dari bahasa C adalah bahasa BCPL yang dikembangkan oleh Martin Richards pada tahun 1967.
- BCPL memberikan ide kepada Ken Thompson yang kemudian mengembangkan bahasa yang disebut dengan B pada tahun 1970.
- Selanjutnya dari bahasa B ke bahasa C oleh Dennis Ritchie sekitar tahun 1970-an di Bell Telephone Laboratories Inc. (sekarang adalah AT&T Bell Laboratories).
- Bahasa C pertama kali digunakan pada komputer Digital Equipment Corporation PDP-11 yang menggunakan sistem operasi UNIX.



Sejarah C

- C adalah bahasa yang standar, artinya suatu program yang ditulis dengan versi bahasa C tertentu akan dapat dikompilasi dengan versi bahasa C yang lain dengan sedikit modifikasi.
- Standar bahasa C yang asli adalah standar dari UNIX.
- Patokan dari standar UNIX ini diambilkan dari buku yang ditulis oleh Brian Kernighan dan Dennis Ritchie berjudul "*The C Programming Language*", diterbitkan oleh Prentice-Hall tahun 1978.
- Deskripsi C dari Kernighan dan Ritchie ini kemudian dikenal secara umum sebagai "K&R C".
- Kepopuleran bahasa C membuat muncul banyak varian dari kompiler C, sehingga dilakukan standarisasi bahasa oleh ANSI (*American National Standards Institute*) dengan membentuk suatu komite (*ANSI committee X3J11*) pada tahun 1983 yang kemudian menetapkan standar ANSI untuk bahasa C.
- Standar ANSI menetapkan sebanyak 32 buah kata-kata kunci (*keywords*).
- C yang mengadopsi standar dari ANSI dikenal dengan nama ANSI C



Ruang Lingkup Bahasa C

- Banyak pemrograman yang dibuat dengan bahasa C seperti assembler, interpreter, program paket, sistem operasi, editor, kompiler, program bantu, Word Star, Dbase, aplikasi untuk bisnis, matematika, dan game, bahkan ada pula yang menerapkannya untuk kecerdasan buatan.
- Dalam beberapa literatur bahasa C digolongkan sebagai bahasa tingkat menengah.
- Pada kenyataannya bahasa C mengkombinasikan kelebihan dari bahasa tingkat tinggi dan bahasa tingkat rendah.
- Kemudahan dalam membuat program yang ditawarkan pada bahasa tingkat tinggi dan kecepatan eksekusi dari bahasa tingkat rendah merupakan tujuan diwujudkannya bahasa



Proses Kompilasi & Linking

- Agar suatu program dalam bahasa pemrograman dapat dimengerti oleh komputer, diperlukan sebuah penerjemah/translator yang akan menerjemahkan ke dalam kode mesin.
- Adapun penerjemah yang digunakan bisa berupa INTERPRETER atau KOMPIILER
- INTERPRETER ? contoh : BASIC
 - menerjemahkan baris per baris instruksi untuk setiap saat
 - Keuntungan pemakaian interpreter, penyusunan program relatif lebih cepat dan bisa langsung diuji sekalipun masih ada beberapa kesalahan secara kaidah dalam program.
 - Kelemahan:
 - kecepatannya menjadi lambat sebab sebelum suatu instruksi dijalankan selalu harus diterjemahkan terlebih dahulu.
 - saat program dieksekusi, interpreter juga harus berada dalam memori, sehingga memori selalu digunakan baik untuk source program maupun interpreter.
 - Program sumber (*source program*) yaitu program aslinya tidak dapat dirahasiakan (orang lain selalu bisa melihatnya).

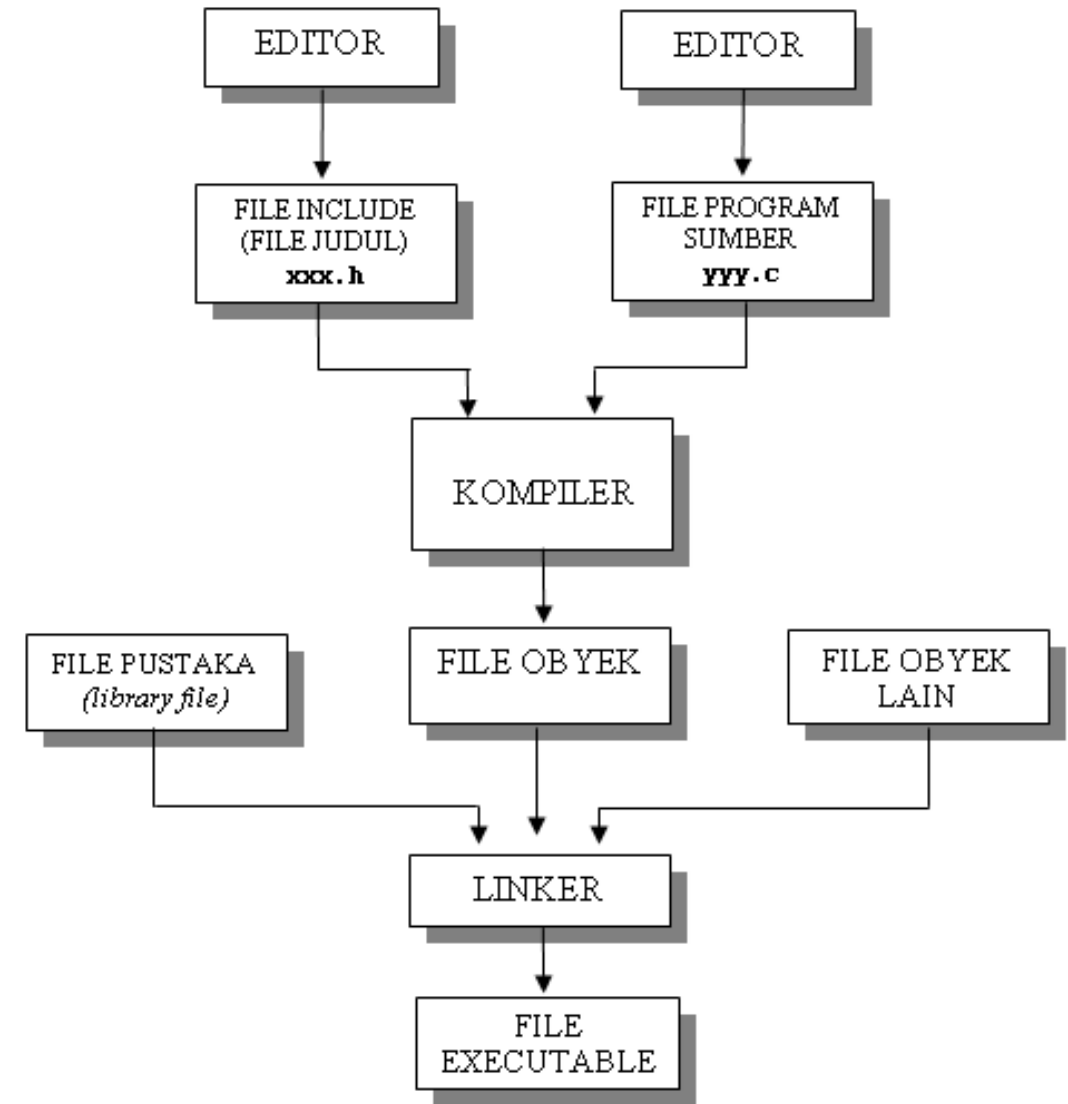
Proses Kompilasi & Linking

- KOMPILER [?] C, PASCAL, JAVA

- Menerjemahkan seluruh instruksi dalam program sekaligus.
- Proses pengkompilasian ini cukup dilakukan sekali saja, selanjutnya hasil penerjemahan (setelah melalui tahapan yang lain) bisa dijalankan secara langsung, tanpa tergantung lagi oleh program sumber maupun kompilernya.
- Keuntungan :
 - Proses eksekusi dapat berjalan dengan cepat, sebab tak ada lagi proses penerjemahan.
 - Program sumber bisa dirahasiakan, sebab yang dieksekusi adalah program yang sudah dalam bentuk kode mesin.
- Kelemahan :
 - proses pembuatan dan pengujian membutuhkan waktu relatif lebih lama, sebab ada waktu untuk mengkompilasi (menerjemahkan) dan ada pula waktu melakukan proses *linking*. Perlu pula diketahui, program akan berhasil dikompilasi hanya jika program tak mengandung kesalahan secara kaidah sama sekali.

Proses Kompilasi & Linking

1. Program C ditulis dengan menggunakan editor dan disimpan dalam file yang disebut file program sumber/source code (dengan ciri utama memiliki ekstensi **.c**).
2. File *include* (umumnya memiliki ekstensi **.h**, misalnya **stdio.h**, atau biasa disebut dengan file judul (*header file*)) berisi kode yang akan dilibatkan dalam program



Gambar 1.1 Proses Kompilasi-Linking dari program C

Proses Kompilasi & Linking

3. Source code dan file include dikompilasi oleh kompiler menjadi kode obyek. Kode obyek berbentuk kode mesin, oleh karena itu tidak dapat dibaca oleh pemrogram tetapi kode ini sendiri juga belum bisa dipahami komputer.
4. Kode obyek bersama-sama dengan kode obyek yang lain (kalau ada) dan isi file pustaka (*library file*, yaitu file yang berisi rutin untuk melaksanakan tugas tertentu. File ini disediakan oleh pembuat kompiler, biasanya memiliki ekstensi **.lib**) perlu dikaitkan (*linking*) dengan menggunakan *linker*, menghasilkan *executable file* (program yang dapat dijalankan/dieksekusi secara langsung dalam lingkungan sistem operasi).
5. Program hasil *linker* ini disimpan dalam file yang disebut file *executable*, yang biasanya berekstensi **.exe**.



Struktur Penulisan Program C

- Program C pada hakekatnya tersusun atas sejumlah blok fungsi, dan sebuah program minimal mengandung sebuah fungsi.
- Fungsi pertama yang harus ada dalam program C dan sudah ditentukan namanya adalah *main()*.
- Setiap fungsi terdiri atas satu atau beberapa pernyataan, yang secara keseluruhan dimaksudkan untuk melaksanakan tugas khusus.
- Bagian pernyataan fungsi (sering disebut tubuh fungsi) diawali dengan tanda kurung kurawal buka ({) dan diakhiri dengan tanda kurung kurawal tutup (}) yang mengisyaratkan awal dan akhir definisi fungsi.
- Di antara kurung kurawal itu dapat dituliskan statemen-statemen program C.



Struktur Penulisan Program C

- Berikut ini adalah struktur dari program C

```
#include <.... .h> //preprocessor directive sesuai fungsi standar yg digunakan
main()                //fungsi utama
{
    statemen-statemen;
}

fungsi_fungsi_lain()
{
    statemen-statemen;    //user defined function
}
```

- Bahasa C dikatakan sebagai bahasa pemrograman terstruktur karena strukturnya menggunakan fungsi-fungsi sebagai program-program bagiannya (*subroutine*)
- Fungsi-fungsi selain main() dapat ditulis setelah fungsi utama atau diletakkan di file pustaka (*library*).
- Jika diletakkan di file pustaka dan akan dipakai di suatu program, maka nama file judulnya (*header file*) harus dilibatkan menggunakan *preprocessor directive* berupa `#include`.

Struktur Penulisan Program C

Fungsi *printf()* merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga. Untuk menampilkan tulisan

Selamat belajar bahasa C

misalnya, pernyataan yang diperlukan berupa:

```
printf("Selamat belajar bahasa C");
```

- Pernyataan di atas berupa pemanggilan fungsi *printf()* dengan argumen atau parameter berupa string.
- Dalam C suatu konstanta string ditulis dengan diawali dan diakhiri tanda petik-ganda (").
- Perlu juga diketahui pernyataan dalam C selalu diakhiri dengan tanda titik koma (;).
- Tanda titik koma dipakai sebagai tanda pemberhentian sebuah pernyataan dan bukanlah sebagai pemisah antara dua pernyataan.



Struktur Penulisan Program C

- Tanda \ pada string yang dilewatkan sebagai argumen *printf()* mempunyai makna yang khusus.
- Tanda ini bisa digunakan untuk menyatakan karakter khusus seperti karakter baris-baru ataupun karakter *backslash* (miring kiri).
- Jadi karakter seperti \n sebenarnya menyatakan sebuah karakter. Contoh karakter yang ditulis dengan diawali tanda \ adalah:

\"	menyatakan karakter petik-ganda
\\	menyatakan karakter backslash
\t	menyatakan karakter tab

Struktur Penulisan Program C

Dalam bentuk yang lebih umum, format *printf()*

```
printf("string kontrol", daftar argumen);
```

dengan string kontrol dapat berupa satu atau sejumlah karakter yang akan ditampilkan ataupun berupa penentu format yang akan mengatur penampilan dari argumen yang terletak pada daftar argumen. Mengenai penentu format di antaranya berupa:

- %d untuk menampilkan bilangan bulat (integer)
- %f untuk menampilkan bilangan titik-mengambang (pecahan)
- %c untuk menampilkan sebuah karakter
- %s untuk menampilkan sebuah string



Contoh

```
#include <stdio.h>

main() {
    printf("Saya sedang belajar Pemrograman");
}
```

Contoh

```
#include <stdio.h>

main() {
    int bilangan = 5;

    printf("Nilai bilangan : %d", bilangan);
}
```

Latihan

1. Apakah keluaran dari program di bawah ini :

```
#include <stdio.h>

main() {
    printf("The black dog was big. ");
    printf("The cow jumped over the moon.\n");
}
```

2. Gunakan pernyataan *printf()* untuk menampilkan (di layar) nilai dari sebuah variabel (misalkan namanya = **sum**) yang bertipe integer.

3. Gunakan pernyataan *printf()* untuk menampilkan (di layar) string “Welcome” yang diikuti dengan sebuah perintah ganti baris.

4. Gunakan pernyataan *printf()* untuk menampilkan (di layar) sebuah karakter dari variabel yang bertipe karakter (misalkan namanya = **letter**).

5. Gunakan pernyataan *printf()* untuk menampilkan (di layar) nilai dari sebuah variabel float (misalkan namanya = **discount**).



Referensi

1. Brian W. Kerninghan, Dennis M. Ritchie (2012): The C Programming Language : Ansi C Version 2 Edition, PHI Learning
2. Byron Gottfried (2010) : Programming with C, Tata McGraw - Hill Education
3. Kochan Stephen (2004) : Programming in C, 3rd Edition, Sams
4. K. N. King (2008) : C Programming: A Modern Approach, 2nd Edition, W. W. Norton & Company
5. Abdul Kadir (2012) : Algoritma & Pemrograman Menggunakan C & C++, Andi Publisher, Yogyakarta
6. https://www.nmu.edu/Webb/ArchivedHTML/MathComputerScience/c_programming/cstart.htm
7. <http://www.gdsw.at/languages/c/programming-bbrowne/>
8. <http://www.programiz.com/c-programming>
9. <http://www.cprogramming.com/tutorial/c-tutorial.html>



bridge to the future

<http://www.eepis-its.edu>