

# *Sekilas Tentang C*

Dasar Pemrograman & Algoritma

# Objectives

---

Setelah menyelesaikan bab ini, anda diharapkan dapat:

- Menjelaskan sejarah dan ruang lingkup pemakaian bahasa C
- Menjelaskan kelebihan dan kekurangan bahasa C
- Menjelaskan proses kompilasi dan linking program C
- Menjelaskan struktur penulisan bahasa C dan menjelaskan komponen-komponen program dalam contoh aplikasi sederhana

# Sejarah bahasa C

---

- Akar dari bahasa C adalah bahasa BCPL → dikembangkan oleh Martin Richards (1967).
- Ken Thompson mengembangkan bahasa yang disebut dengan B pada tahun 1970.
- Bahasa C pertama kali digunakan pada komputer Digital Equipment Corporation (DEC) PDP-11 yang menggunakan sistem operasi UNIX.
- Standar bahasa C yang asli adalah standar dari UNIX.
- ANSI (*American National Standards Institute*) membentuk suatu komite (*ANSI committee X3J11*) pada tahun 1983 yang kemudian menetapkan standar ANSI untuk bahasa C.

## Sejarah bahasa C - continued

---

- Bahasa C digolongkan bahasa tingkat menengah.
- Lebih sulit dari Pascal dan Basic (bahasa tingkat tinggi).
- C mengkombinasikan antara bahasa tingkat rendah (mesin / assembly) dengan bahasa tingkat tinggi.
- Tujuan → kemudahan membuat program dari bahasa tingkat tinggi dikombinasikan dengan kecepatan eksekusi dari bahasa tingkat rendah.

# Kelebihan bahasa C

---

- Tersedia hampir di semua jenis komputer.
- Kodenya bersifat portabel. Aplikasi yang ditulis dengan C untuk suatu komputer tertentu dapat digunakan di komputer lain hanya dengan sedikit modifikasi.
- Struktur bahasa yang baik → mudah dipelajari, memudahkan pembuatan program, memudahkan pelacakan kesalahan program.
- Dibandingkan bahasa mesin atau assembly, C jauh lebih mudah dipahami. C berorientasi pada permasalahan, bukan pada mesin.
- Dapat memanipulasi data dalam bentuk bit atau byte.
- Memungkinkan untuk memanipulasi alamat dari suatu data atau pointer.

## Kelemahan Bahasa C

---

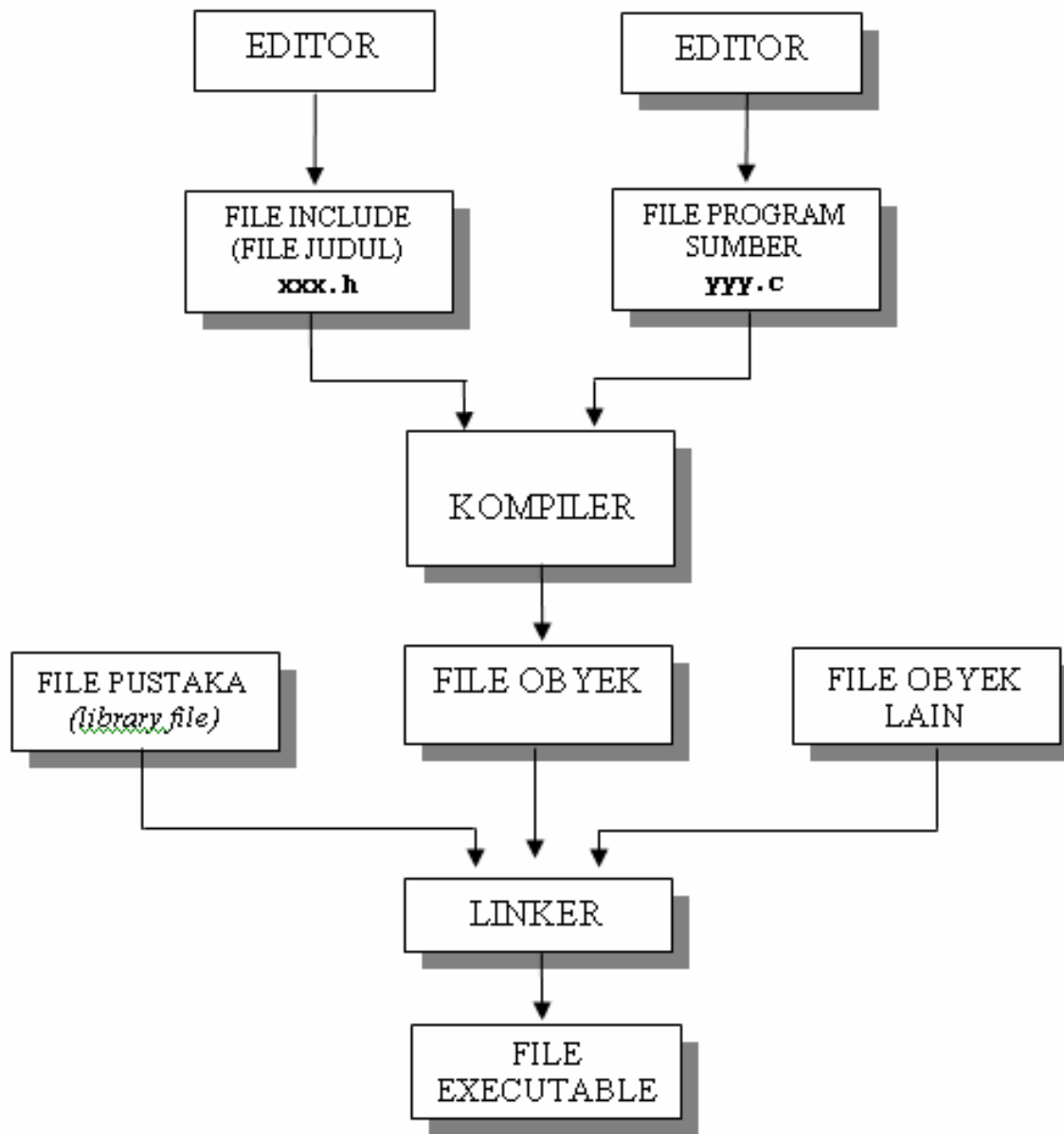
- Banyaknya operator serta fleksibilitas penulisan program kadang-kadang membingungkan pemakai.
- Para pemrogram C tingkat pemula umumnya belum pernah mengenal pointer dan tidak terbiasa menggunakannya → kemampuan C justru terletak pada pointer.

*"Dimana Ada Kemauan Disitu Ada Jalan"*

# Compiler dan Linker

---

- Sebelum program C dapat dijalankan, harus di compile dan di linking terlebih dahulu.
- Compile adalah proses menterjemahkan seluruh program ke dalam bahasa mesin sekaligus.
- Compile dapat berhasil jika dalam program tidak ada kesalahan kaidah sama sekali.
- List program asli dapat dirahasiakan.
- Linking adalah proses untuk menggabungkan beberapa file program hasil kompilasi.



# Interpreter

---

- Biasa digunakan pada bahasa Basic.
- Menerjemahkan baris per baris instruksi pada program untuk setiap saat.
- Proses pengujian program cepat (+).
- Running program menjadi lambat (-).
- Program asli tidak dapat dirahasiakan (-).

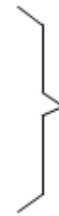
# Struktur penulisan C

---

- Program C tersusun atas sejumlah fungsi.
- Minimal dalam satu program bahasa C ada satu fungsi → fungsi main().
- Setiap fungsi terdiri dari satu atau beberapa pernyataan.
- Penulisan suatu fungsi diawali dengan '{' dan diakhiri dengan '}'.
- C disebut bahasa terstruktur → strukturnya menggunakan fungsi-fungsi sebagai program-program bagian (*subroutine*).

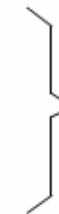
---

```
main()  
{  
    statemen-statement;  
}
```



**fungsi utama**

```
fungsi_fungsi_lain()  
{  
    statemen-statement;  
}
```



**fungsi-fungsi lain yang  
ditulis oleh pemrogram**

## Preprocessor directive (`#include`)

---

- *#include* merupakan salah satu jenis pengarah praprosesor (*preprocessor directive*).
- Dipakai untuk membaca file yang di antaranya berisi deklarasi fungsi dan definisi konstanta.
- Beberapa file judul disediakan dalam C → nama filenya diakhiri dengan ekstensi **.h**.
- `#include <stdio.h>` menyatakan pada kompiler agar membaca file bernama *stdio.h* saat pelaksanaan kompilasi.
- Bentuk umum `#include`: `#include "namafile"`

## Fungsi main()

---

- Merupakan fungsi istimewa.
- Harus ada pada program C.
- Fungsi ini menjadi titik awal dan titik akhir eksekusi program.
- Biasa ditempatkan pada posisi paling atas dari program → memudahkan untuk mencari program utama.

## Fungsi printf()

---

- Digunakan untuk menampilkan suatu keluaran pada layar/monitor.
- Bentuk umum:  
`printf("string kontrol", daftar argumen);`
- String kontrol dapat berupa:
  - %d untuk menampilkan bilangan bulat (integer)
  - %f untuk menampilkan bilangan titik-mengambang (pecahan)
  - %c untuk menampilkan sebuah karakter
  - %s untuk menampilkan sebuah string

## Fungsi printf() - *continued*

---

- **Contoh:**

```
#include <stdio.h>
main( )
{
    printf( "No      : %d\n", 10 );
    printf( "Nama   : %s\n", "Ali" );
    printf( "Nilai  : %f\n", 80.5 );
    printf( "Huruf  : %c\n", 'A' );
}
```

## Fungsi printf() - *continued*

---

- Untuk menampilkan kalimat “Hello World” digunakan perintah: `printf( "Hello World" );`
- Tanda ‘\’ dapat digunakan pada fungsi printf:
  - \” menyatakan karakter petik ganda.
  - \\ menyatakan karakter backslash ‘\’.
  - \t menyatakan karakter tab.

# Komentar dalam program

---

- Digunakan untuk keperluan dokumentasi
- Dimulai dengan tanda “/\*” dan diakhiri dengan tanda “\*/”
- Untuk komentar yang hanya satu baris ditulis dengan diawali tanda “//”
- Contoh:

```
/*Tanda ini adalah komentar  
tidak masuk dalam eksekusi program */  
#include <stdio.h>
```

```
main()  
{  
    printf( "Coba\n" ); //komentar satu baris  
}
```