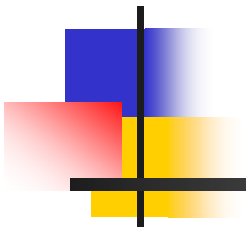


# Sekilas Tentang C





# Objectives

---

- 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 (1)

---

- 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 sendiri dikembangkan pada Laboratorium Bell (USA) sekitar tahun 1972 —→ Dennis Ritchie
- Bahasa C pertama kali digunakan pada komputer Digital Equipment Corporation (DEC) PDP-11 yang menggunakan sistem operasi UNIX.
- Pada tahun 1978, Dennis Ritchie dan Brian W.Kernighan mengeluarkan buku berjudul “The C Programming Language” (menjadi acuan pembuatan berbagai versi C)
- Untuk melaksanakan standarisasi terhadap bahasa C pada tahun 1983, ANSI (American National Standards Institute) membuat C standard ANSI.



## Sejarah bahasa C (2)

---

- 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

---

- 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.

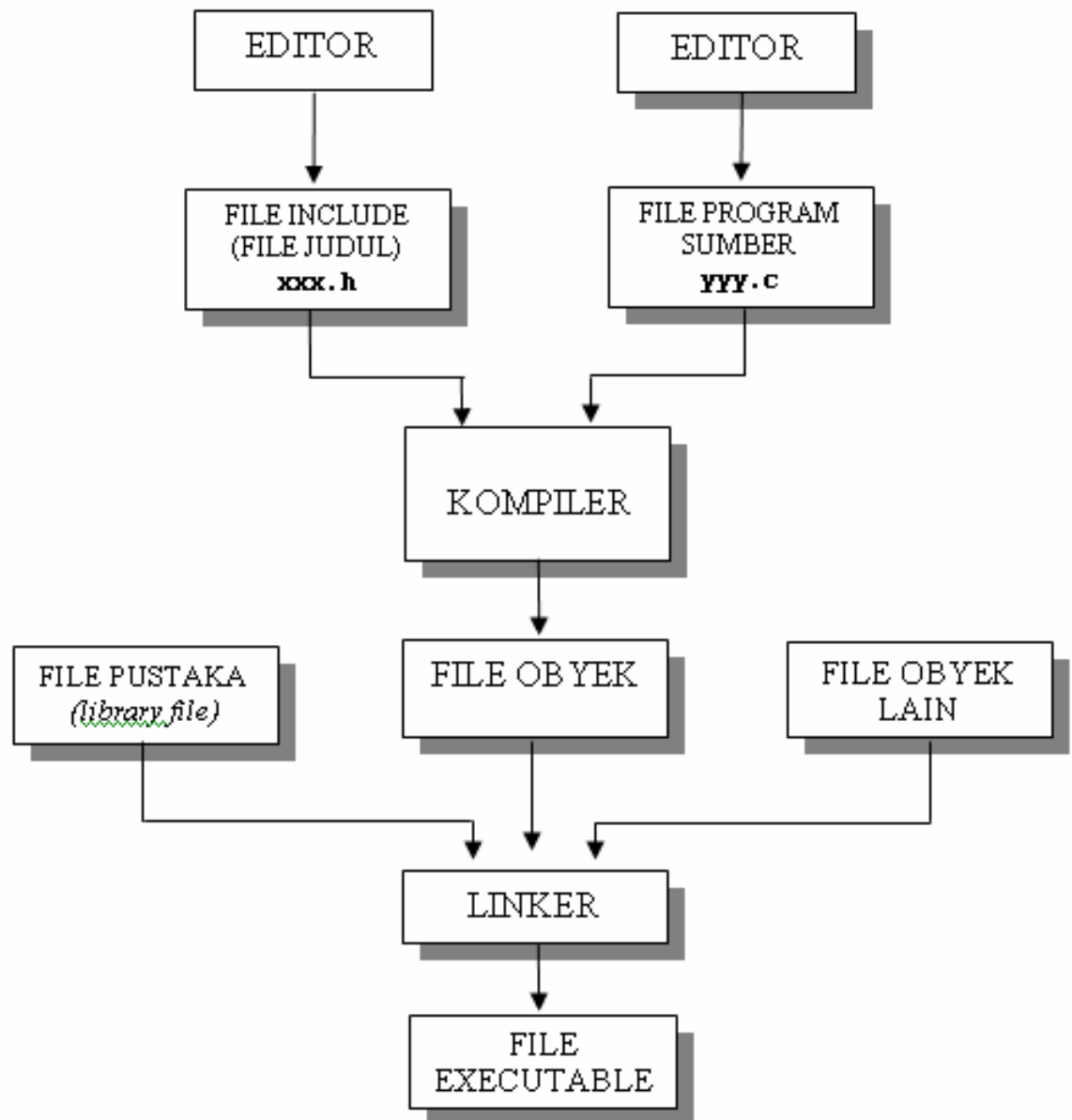
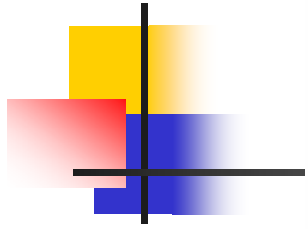
*"Dimana Ada Kemauan Disitu Ada Jalan"*



# 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.



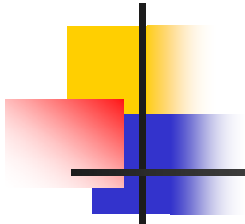




# Struktur penulisan C

---

- 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-statenen;
}
```

**fungsi utama**

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

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



# 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.



## Program yang terpendek dalam C :

---

```
main( ) ← Fungsi utama program c
{ ← awal tubuh fungsi main()
  sekaligus awal eksekusi
  program
} ← akhir tubuh fungsi main()
  sekaligus akhir eksekusi
  program
```



# Pengenalan 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>`



Contoh :

---

```
#include <namafile>
```

```
Main( )
```

```
{
```

```
}
```

# Pengenalan Fungsi printf()

- menampilkan suatu keluaran pada layar/monitor.
- Melibatkan file stdio.h

Contoh : (Menampilkan kalimat “selamat datang di PENS ITS”)

```
#include <stdio.h>
```

```
Main( )
```

```
{
```

```
    printf("selamat datang di PENS ITS");
```

```
}
```

# Fungsi printf() - continued

■ Tanda ‘\’ atau escape sequence dapat digunakan pada fungsi printf:

- \n menyatakan karakter baris baru.
- \" menyatakan karakter petik ganda
- \\ menyatakan karakter backslash ‘\’.
- \t menyatakan karakter tab.

## Contoh:

Untuk menampilkan keluaran berupa:

selamat datang

Di PENS ITS

```
#include <stdio.h>
main( )
{
printf("selamat datang\n");
printf("di PENS ITS\n");
}
```



# Fungsi printf() - continued

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

■ 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' );
}
```



# 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  
}
```