

BAB II

KEGIATAN PEMBELAJARAN

A. Rencana Belajar Peserta

Modul ini diperuntukkan bagi mahasiswa di Jurusan Pendidikan Teknik Elektro sebagai bahan suplemen untuk membantu mahasiswa dalam mempelajari dan memahami bahasa pemrograman, khususnya bahasa C++. Bahasa pemrograman C++ atau selanjutnya disebut C++ merupakan bahasa standar yang biasa digunakan untuk kalangan perguruan tinggi, baik di dalam maupun di luar negeri. C++ mempunyai keunggulan yang hampir tidak dimiliki oleh bahasa pemrograman lain, antara lain *bahasa yang sangat populer saat ini, memiliki portabilitas tinggi, fleksibel, memiliki sedikit kata kunci, dan yang penting adalah mendukung terhadap pemrograman berorientasi objek.*

1. Kompetensi : Menguasai bahasa Pemrograman C++ dengan baik

2. Sub Kompetensi :

- a. Mengetahui seluk beluk sejarah perkembangan bahasa C++
- b. Menjelaskan struktur bahasa C++ dengan tepat
- c. Menggunakan system operator dalam C++
- d. Menjelaskan bentuk percabangan dalam C++
- e. Menjelaskan bentuk perulangan dalam C++
- f. Menjelaskan konsep array dalam C++
- g. Menjelaskan fungsi dalam C++

3. Indikator pencapaian pembelajaran :

- a. Dapat menjelaskan perkembangan C++
- b. Dapat menjelaskan struktur bahasa C++
- c. Dapat menggunakan operator dalam C++
- d. Dapat menggunakan bentuk percabangan dalam C++
- e. Dapat menggunakan bentuk perulangan dalam C++

f. Dapat menggunakan array dalam C++

g. Dapat membuat fungsi dalam C++

4. Jenis kegiatan pembelajaran :

a. Kegiatan pembelajaran dirancang untuk pembelajaran di kelas yang terdiri atas penyampaian kompetensi, sub kompetensi, tujuan belajar, indikator dan materi pembelajaran serta melakukan evaluasi pembelajaran.

b. Kegiatan pembelajaran dilakukan melalui pendekatan ceramah, diskusi, tanya jawab dan penugasan.

c. Proses kegiatan pembelajaran dilakukan secara teori di kelas dan dipraktikkan di laboratorium komputer.

B. Kegiatan Pembelajaran

Kegiatan belajar disusun dalam serangkaian pengalaman belajar yang merupakan satu kesatuan aktifitas belajar dalam rangka mempermudah kegiatan belajar untuk menguasai kompetensi dalam modul ini, yakni terdiri atas materi: (1) Sejarah perkembangan Bahasa C++, (2) struktur bahasa C++; (3) system operator dalam C++; (4) Percabangan dalam C++; (5) Perulangan dalam C++; (6) Array dalam C++; (7) fungsi dalam C++

Keg. Pembelajaran 1 : Perkembangan Sejarah Bahasa C++

1. Tujuan Kegiatan Pembelajaran

Setelah mempelajari materi kegiatan pembelajaran ini mahasiswa akan dapat :

1) Mengetahui secara baik sejarah perkembangan bahasa pemrograman C sampai menjadi C++.

2) Mengetahui kekuatan bahasa C secara benar, mengapa menggunakan C

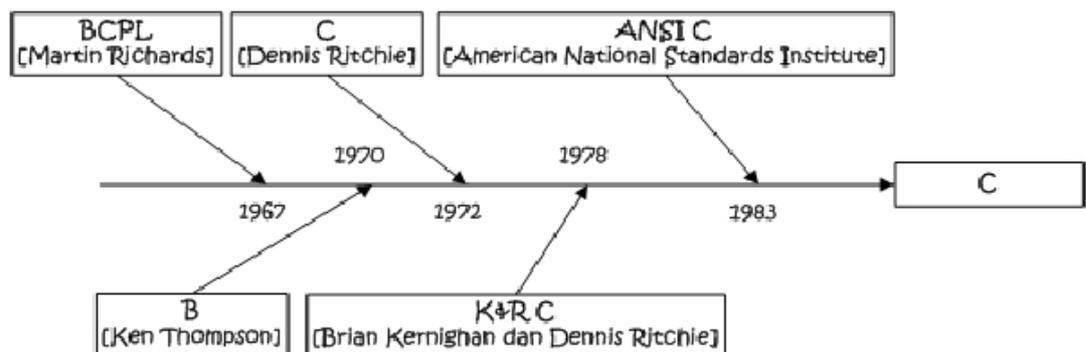
3) Mengetahui struktur bahasa C dengan baik

d. Menjelaskan Konsep Kompilasi dan Eksekusi Program dalam C++

2. Uraian Materi

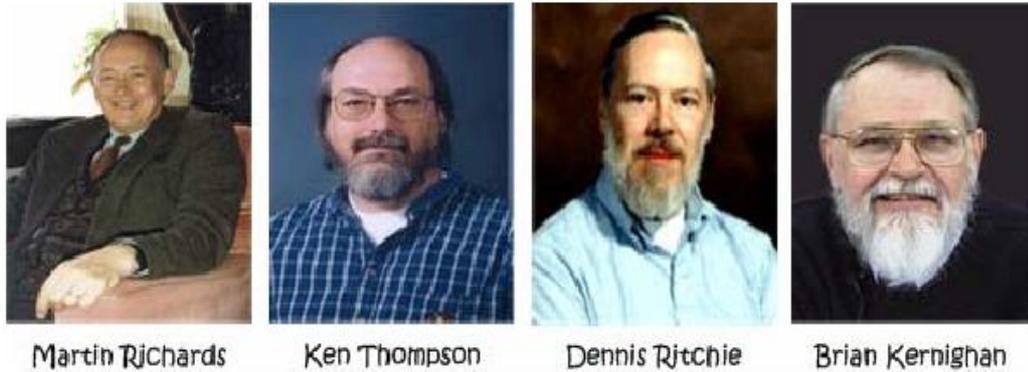
a. Sejarah Bahasa C sampai ke C++

Berbicara tentang C++ dan C sebagai bahasa pendahulunya, C merupakan termasuk bahasa pemrograman tingkat menengah. C merupakan bahasa pemrograman terstruktur yang membagi program ke dalam sejumlah blok (sub program). Tujuannya adalah untuk memudahkan dalam pembuatan dan pengembangan program. Program yang ditulis dengan C mudah sekali dipindahkan dari satu jenis mesin ke mesin lain. Sejarah perkembangan dan latar belakang munculnya bahasa C adalah seperti dalam Gambar 1.



Gambar 1: Sejarah Bahasa C

Boleh dikatakan bahwa akar dari bahasa C adalah bahasa BCPL yang dikembangkan oleh *Martin Richards* pada tahun 1967. Kemudian berdasar pada bahasa BCPL ini *Ken Thompson* yang bekerja di Bell Telephone Laboratories (Bell Labs) mengembangkan bahasa B pada tahun 1970. Saat itu bahasa B telah berhasil diimplementasikan di komputer DEC PDP-7 dengan operating system (OS) UNIX. Pada tahun 1972, peneliti lain di Bell Labs bernama *Dennis Ritchie* menyempurnakannya menjadi bahasa C.



Gambar 2: Tokoh Yang Berjasa Dalam Pengembangan Bahasa C

Pada tahun 1978, *Dennis Ritchie* bersama dengan *Brian Kernighan* mempublikasikan buku yang kemudian menjadi legenda dalam sejarah perkembangan bahasa C, yang berjudul *The C Programming Language*. Buku ini diterbitkan oleh Prentice Hall, dan pada saat ini telah diterjemahkan dalam berbagai bahasa di dunia. Boleh dikatakan bahwa buku ini adalah buku yang paling banyak direfer orang dan dijadikan buku panduan tentang pemrograman bahasa C sampai saat ini. Teknik dan gaya penulisan bahasa C yang merujuk kepada buku ini kemudian terkenal dengan sebutan *K&R C* atau *Classic C* atau *Common C*.

Seiring dengan berkembang pesatnya bahasa C, banyak vendor mengembangkan kompiler C menurut versi masing-masing. Hal ini menggerakkan ANSI (*American National Standards Institute*) pada tahun 1983 untuk membuat suatu komite yang kemudian diberi nama *X3J11*, yang bertujuan untuk membuat definisi standar bahasa C yang lebih modern dan komprehensif, dengan memperbaiki *syntax* dan *grammar* bahasa C. Usaha ini berhasil diselesaikan 5 tahun kemudian, yaitu ditandai dengan lahirnya standard ANSI untuk bahasa C yang kemudian terkenal dengan sebutan *ANSI C* pada tahun 1988.

b. Mengapa Memakai Bahasa C

Sampai saat ini, bahasa C telah berhasil digunakan untuk mengembangkan berbagai jenis permasalahan pemrograman, dari level operating system (unix, linux, ms dos, dsb), aplikasi perkantoran (text editor, word processor, spreadsheet, dsb), bahkan sampai pengembangan sistem pakar (*expert system*). Kompiler C juga telah tersedia di semua jenis platform komputer, mulai dari Macintosh, UNIX, PC, Micro PC, sampai super komputer.

Keistimewaan C++ adalah karena bahasa ini mendukung OOP (Object Oriented Programming). Tujuan utama pembuatan C++ adalah untuk meningkatkan produktivitas pemrogram dalam membuat aplikasi. Kebanyakan pakar setuju bahwa OOP dan C++ mampu mengurangi kompleksitas terutama program yang terdiri dari 10.000 baris lebih, bahkan dapat meningkatkan produktivitas 2x lipat dari C, Pascal dan Basic.

C bisa disebut bahasa pemrograman tingkat menengah (*middle level programming language*). Arti tingkat (level) disini adalah kemampuan mengakses fungsi-fungsi dan perintah-perintah dasar bahasa mesin/hardware (*machine basic instruction set*). Semakin tinggi tingkat bahasa pemrograman (misalnya: java), semakin mudahnya bahasa pemrograman dipahami manusia, namun membawa pengaruh semakin berkurang kemampuan untuk mengakses langsung instruksi dasar bahasa mesin. Demikian juga sebaliknya dengan bahasa pemrograman tingkat rendah (misalnya: assembler), yang semakin sulit dipahami manusia dan hanya berisi perintah untuk mengakses bahasa mesin. Dalam perspektif mudahnya dipahami manusia, C bisa digolongkan dalam bahasa tingkat tinggi, namun C juga menyediakan kemampuan yang ada pada bahasa tingkat rendah, misalnya operasi bit, operasi byte, pengaksesan memori, dsb.

Beberapa alasan mengapa memakai bahasa C adalah terangkum dibawah.

- 1) *C adalah bahasa pemrograman yang paling populer saat ini.* Dengan banyaknya programmer bahasa C, membawa pengaruh semakin mudahnya kita menemukan pemecahan masalah yang kita dapatkan ketika menulis program dalam bahasa C. Pengaruh positif lain adalah semakin banyaknya kompiler yang dikembangkan untuk berbagai platform (berpengaruh ke portabilitas).
- 2) *C adalah bahasa pemrograman yang memiliki portabilitas tinggi.* Program C yang kita tulis untuk satu jenis platform, bisa kita kompilasi dan jalankan di platform lain dengan tanpa ataupun hanya sedikit perubahan. Ini bisa diwujudkan dengan adanya standarisasi ANSI untuk C.
- 3) *C adalah bahasa pemrograman dengan kata kunci (keyword) sedikit.* Kata kunci disini adalah merupakan fungsi ataupun kata dasar yang disediakan oleh kompiler suatu bahasa pemrograman. Hal ini membawa pengaruh semakin mudahnya kita menulis program dengan C. Pengaruh lain dari sedikitnya kata kunci ini adalah proses eksekusi program C yang sangat cepat. C hanya menyediakan 32 kata kunci seperti terangkum pada Tabel 1 dibawah:

Tabel 1: Kata kunci dalam C

auto	break	case	char	const	continue	default
do	double	else	enum	extern	float	for
goto	if	int	long	register	return	short
signed	sizeof	static	struct	switch	typedef	union
unsigned	void	volatile	while			

- 4) *C* adalah bahasa pemrograman yang fleksibel. Dengan menguasai bahasa *C*, kita bisa menulis dan mengembangkan berbagai jenis program mulai dari operating system, word processor, graphic processor, spreadsheets, ataupun kompiler untuk suatu bahasa pemrograman.
- 5) *C* adalah bahasa pemrograman yang bersifat modular. Program *C* ditulis dalam *routine* yang biasa dipanggil dengan fungsi. Fungsi-fungsi yang telah kita buat, bisa kita gunakan kembali (*reuse*) dalam program ataupun aplikasi lain.

c. Struktur Program Bahasa C

Program bahasa *C* adalah suatu program terdiri dari satu atau lebih fungsi-fungsi. Fungsi utama dan harus ada pada program *C* yang kita buat adalah fungsi **main()**. Fungsi `main()` ini adalah fungsi pertama yang akan diproses pada saat program di kompilasi dan dijalankan, sehingga bisa disebut sebagai fungsi yang mengontrol fungsi-fungsi lain. Karena struktur program *C* terdiri dari fungsi-fungsi lain sebagai program bagian (*subroutine*), maka bahasa *C* biasa disebut sebagai *bahasa pemrograman terstruktur*. Cara penulisan fungsi pada program bahasa *C* adalah dengan memberi nama fungsi dan kemudian dibuka dengan *kurung kurawal buka* (`{`) dan ditutup dengan *kurung kurawal tutup* (`}`).

Fungsi-fungsi lain selain fungsi utama bisa dituliskan setelah atau sebelum fungsi utama dengan deskripsi prototype fungsi pada bagian awal program. Bisa juga dituliskan pada file lain yang apabila kita ingin memakai atau memanggil fungsi dalam file lain tersebut, kita harus menuliskan header filenya, dengan *preprocessor directive* **#include**.

Sampai saat ini, bahasa *C* masih digunakan untuk melakukan pengembangan-pengembangan program dan sistem-sistem operasi, diantaranya sistem operasi Windows. Alasan itulah yang menjadikan bahasa

C sangat populer di dunia pemrograman, khususnya di industri perangkat lunak. Kelemahan dari bahasa C adalah masih tergolong susah untuk dipelajari karena masih bersifat prosedural murni. Untuk membentuk satu objek, kita harus melakukan penulisan kode yang banyak. Untuk mengatasi masalah tersebut, pada tahun 1983, seorang doktor bernama Bjarne Stroustrup menciptakan bahasa baru yaitu bahasa C++ yang merupakan bahasa *hybrid* dari bahasa C. Bahasa C++ didasarkan atas bahasa C sehingga kita dapat melakukan kompilasi program-program yang ditulis dalam bahasa C dengan menggunakan kompiler C++.

Pada mulanya C++ disebut "a better C". Nama C++ sendiri diberikan oleh Rick Mascitti pada musim panas 1983. Adapun tanda ++ berasal dari nama operator kenaikan pada bahasa C. Keistimewaan dari bahasa C++ adalah dapat mendukung pemrograman berorientasi objek atau dikenal dengan istilah *Object Oriented Programming* (OOP).

e. C dan C++

Bahasa C dan C++ digolongkan ke dalam bahasa tingkat menengah (*middle level language*). Seorang profesor yang bernama Niklaus Wirth di Politeknik Zurich (Swiss) mengembangkan bahasa tingkat tinggi (*high level language*) yang disebut bahasa Pascal untuk mengajarkan kepada mahasiswanya.

Sebagai bahan pertimbangan, berikut pada Tabel 1 dijelaskan pengelompokan tingkatan dari bahasa pemrograman.

Tabel 2 : Tingkatan Bahasa berdasarkan levelnya

Bahasa Tingkat Tinggi	Ada Modula-2 Pascal COBOL FORTRAN BASIC
Bahasa Tingkat Menengah	Java C++ C FORTH
Bahasa Tingkat Rendah	Macro-Assembler Assembler

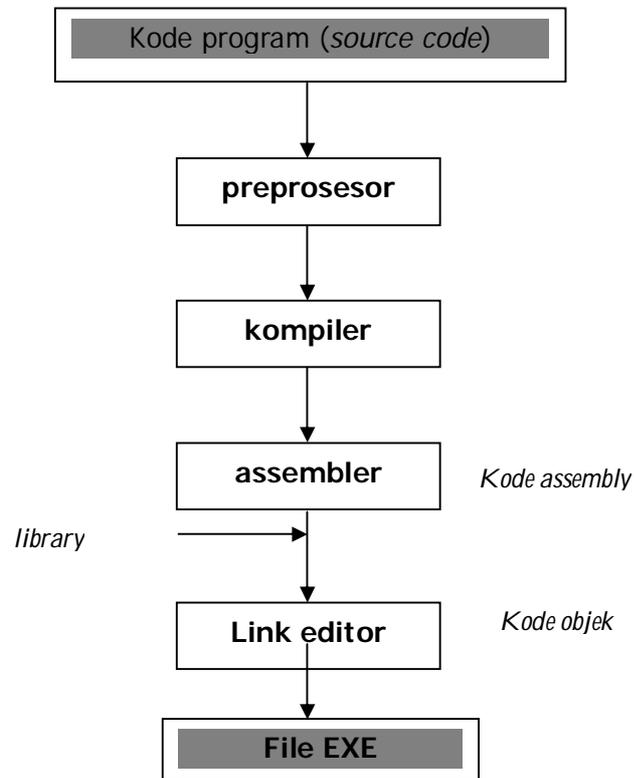
Berdasarkan Tabel 1 di atas, dapat dilihat bahwa bahasa pemrograman yang terdapat pada bagian atas merupakan bahasa pemrograman yang paling mudah untuk dipahami. Sebagai contoh, bahasa C adalah bahasa yang lebih sulit dibandingkan C++ dan C++ adalah lebih sulit dibandingkan dengan bahasa Java, dan seterusnya. Menurut Bjarne Stroustrup (pencipta C++), alasan mengapa C diambil sebagai bahasa dasar dari pembentukan bahasa C++ adalah sebagai berikut:

1. Dapat dihubungkan dengan bahasa tingkat rendah
2. Berjalan dimanapun dan untuk masalah apapun
3. Berjalan mulus dalam sistem operasi UNIX

f. Konsep Kompilasi dan Eksekusi Program

Sebelum melangkah lebih jauh dengan pembentukan program C++, maka sebaiknya kita mengetahui terlebih dahulu konsep dari sebuah kompilasi dan eksekusi program di dalam C/C++. Berikut ini gambar yang mengilustrasikan proses kompilasi dan eksekusi program di dalam C/C++. Perhatikan gambar 3 berikut ini yang menggambarkan proses

eksekusi program sejak dari kode program sampai dengan file EXE yang siap dijalankan dan C++.



Gambar 3 : Ilustrasi proses kompilasi dan eksekusi program di dalam C++

1) Preprocessor

Langkah awal adalah memasukkan kode ke dalam bagian preprosesor, yaitu diawali dengan tanda # (*pound*) dan menghasilkan file yang akan dilewatkan ke dalam kompiler. Beberapa preprosesor tersebut adalah

- #include
- #define
- #ifdef
- Dan lain-lain

2) Kompiler C++

Kompiler akan menerjemahkan kode program yang telah dilewatkan oleh preproesor ke dalam bahasa *assembly*.

3) Assembler

Assembler menerima keluaran dari kompiler C++ dan akan membuat sebuah kode objek. Jika dalam kode program kita tidak menggunakan fungsi-fungsi yang terdapat pada library lain, maka kode objek ini akan langsung dieksekusi menjadi file EXE.

4) Link Editor

Bagian ini dikerjakan jika kode program yang kita buat menggunakan fungsi-fungsi luar yang disimpan dalam suatu library lain. *Link Editor* akan mengkombinasikan kode objek dan library yang ada untuk menjadikan sebuah file EXE.

g. Program C++

Program C++ dapat dibuat menggunakan sebarang editor teks maupun editor sekaligus compilernya. Program utama berekstensi (.CPP). Pada saat kompilasi program utama bersama dengan file header (.h) akan diterjemahkan oleh compiler menjadi file obyek (.OBJ). Selanjutnya file obyek ini bersama-sama dengan file obyek lain dan file library (.LIB) dikaitkan menjadi satu oleh linker. Hasilnya adalah file (.EXE) executable.

Compiler C++ yang telah beredar di pasaran antara lain Microsoft C/C++ dan Visual C++. Keduanya dari Microsoft. Sementara Borland international juga mengeluarkan Turbo C++ dan Borland C++.

h. C++ Klasik dan C++ Modern

Ada perbedaan antara kompiler C++ lama (klasik) dan C++ modern (C++ yang telah memenuhi standar ANSI/ISO). Pada C++ lama masih menggunakan namespace global, sedangkan di C++ modern menggunakan namespace **std**. Hal inilah yang menyebabkan terdapatnya perbedaan penulisan antara C++ lama dan standar.

Seperti yang kita lihat bahwa pada C++ standar penulisan file *header* **iostream** sudah tidak diikuti lagi dengan ekstensi **.h**. Untuk menuliskan kode kedua, diperlukan kompiler C++ yang telah mendukung semua fitur yang terdapat dalam C++ standar atau dilengkapi dengan IDE (Integrated Development Environment).

i. Fungsi main()

Program C++ memang tidak akan pernah lepas dari suatu fungsi/function. Hal ini karena merupakan ciri OOP. Sebuah program C++ minimal memiliki satu fungsi yaitu `main()`. Fungsi ini merupakan awal program utama. Tulisan `main()` merupakan nama fungsi, sedangkan bagian yang diapit dengan `{` dan `}` disebut blok (tubuh fungsi). Dalam hal ini `{` merupakan tanda awal blok dan `}` adalah tanda akhir blok. Seperti halnya dalam Pascal, `{` dalam Pascal identik dengan `BEGIN`, sedangkan `}` identik dengan `END`. Perintah `void` bermakna bahwa fungsi `main()` tidak mengembalikan nilai/value.

j. Object Oriented Programming (OOP)

Ide dasar OOP adalah mengkombinasikan data dan fungsi untuk mengakses data menjadi sebuah kesatuan unit. Unit ini dikenal dengan obyek. Sebagai gambaran untuk mempermudah memahaminya, obyek sebenarnya dapat mencerminkan pola kerja manusia sehari-hari. Sebuah

obyek dapat diibaratkan sebagai departemen di dalam sebuah perusahaan bisnis, misalnya departemen penjualan; akunting; atau personalia.

Pembagian departemen dalam perusahaan merupakan upaya untuk memudahkan pengoperasian perusahaan. Sebagai gambaran, jika Anda seorang manajer penjualan di kantor pusat ingin mengetahui data para salesmen di kantor cabang, apa yang Anda lakukan? Langkah yang Anda tempuh pasti bukan datang ke kantor cabang dan mencari data-data tersebut. Untuk memudahkan tugas Anda cukup Anda menyuruh sekretaris untuk meminta informasi. Masalah bagaimana dan siapa yang mencarinya bukanlah urusan Anda. Analogi dengan hal itu, kalau seseorang bermaksud menggunakan obyek, ia cukup mengirim pesan ke obyek dan obyek itu sendiri yang akan menanganinya.

3. Rangkuman

Bahasa C untuk pertamakalinya dikenalkan pada tahun 1972 oleh *Dennis Ritchie*. Pada tahun 1978, *Dennis Ritchie* bersama dengan *Brian Kernighan* mempublikasikan buku yang kemudian menjadi legenda dalam sejarah perkembangan bahasa C, yang berjudul *The C Programming Language*. Buku ini diterbitkan oleh Prentice Hall, dan pada saat ini telah diterjemahkan dalam berbagai bahasa di dunia.

Bahasa C++ merupakan bahasa pemrograman yang cukup populer di kalangan Perguruan Tinggi hampir di seluruh dunia. C++ memiliki keunggulan yang hampir tidak dimiliki oleh bahasa pemrograman lainnya, antara lain karena C++ memiliki portabilitas yang tinggi, merupakan bahasa induk dari berbagai bahasa pemrograman lainnya, fleksibilitas yang tinggi, sedikit kata kunci, dan mendukung terhadap bahasa pemrograman berorientasi objek.

4. Latihan

- 1) Jelaskan dengan singkat sejarah bahasa C berdasarkan tahun periodik
- 2) Jelaskan dengan singkat proses proses eksekusi program dalam bahasa C++
- 3) Apa yang dimaksud dengan bahasa berbasis fungsi
- 4) Bahasa C++ termasuk bahasa tingkat menengah, coba jelaskan apa maksudnya
- 5) Apa yang dimaksud dengan bahasa berorientasi pada objek