



TIB21 – PEMROGRAMAN DASAR

UNIVERSITAS BUNDA MULIA

Capaian Pembelajaran Mata Kuliah

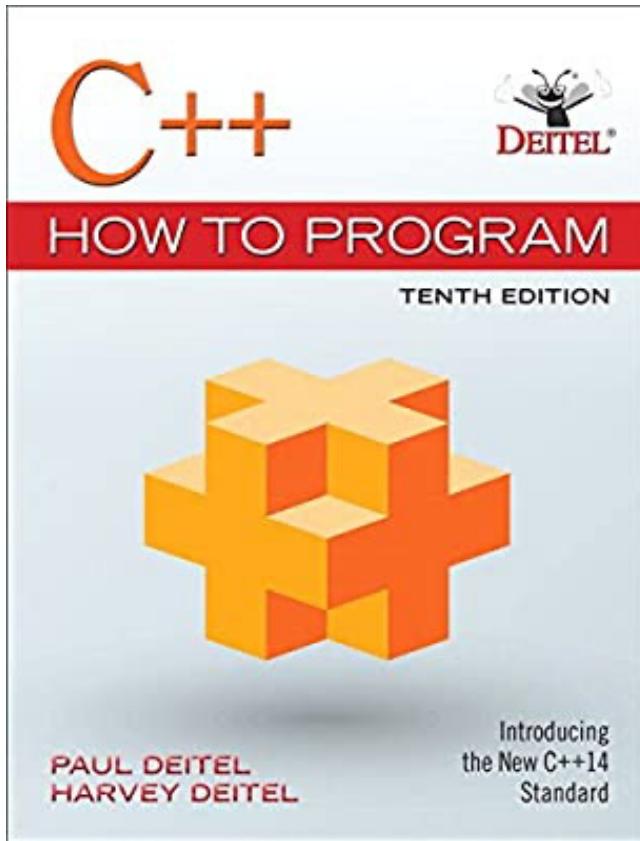
1. Mahasiswa mampu menjelaskan **konstruksi dasar** pemrograman C++, **tipe data** dan **notasi** pemrograman C++ (C2, A2)
2. Mahasiswa mampu mengaplikasikan **program kontrol** dan berbagai macam **fungsi** serta **array** dalam bahasa pemrograman C++ untuk menyelesaikan berbagai macam kasus (C3, A3)
3. Mahasiswa mampu mengaplikasikan **pemrosesan arsip** dalam bahasa pemrograman C++ (C3, A3)



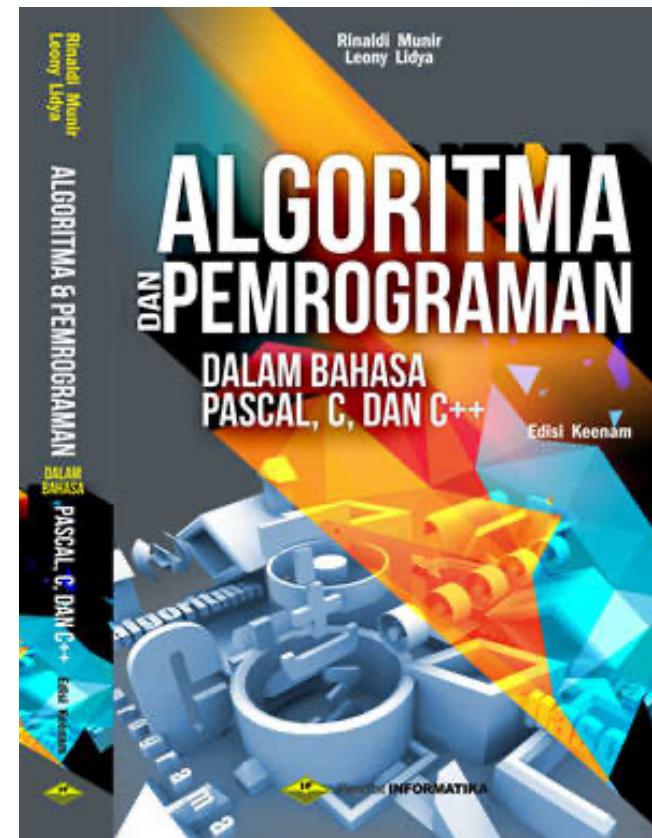
KONSEP BAHASA PEMROGRAMAN C++

Pertemuan ke-1 dan 2

Diadopsi dari sumber :



1

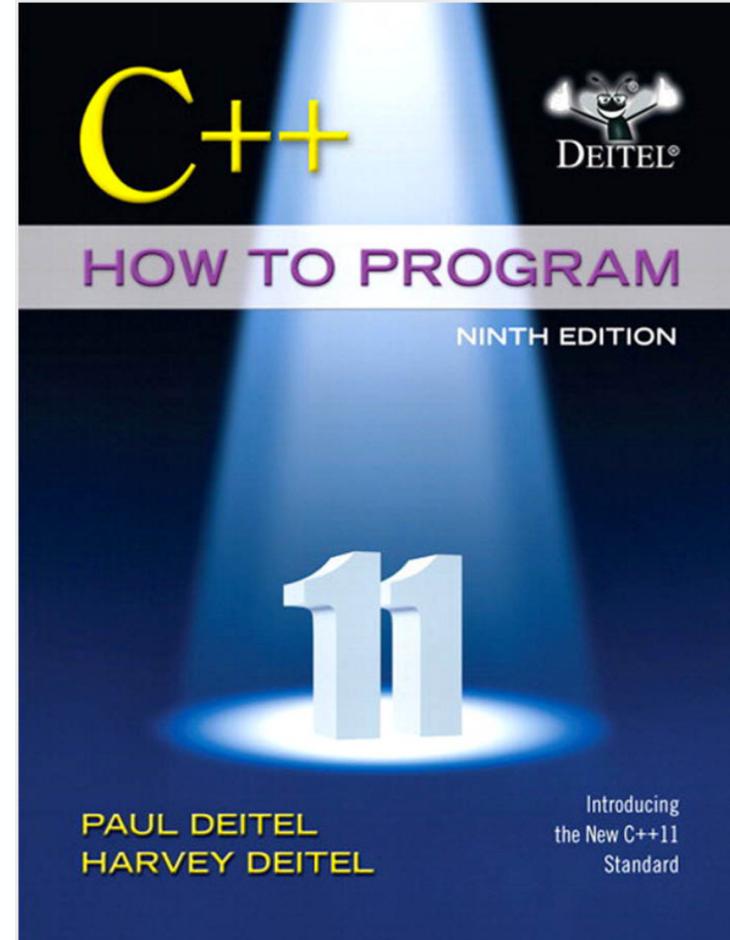
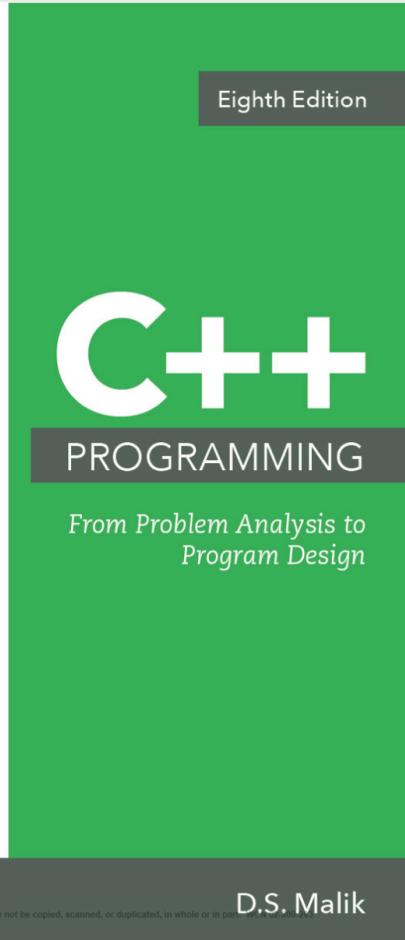


2

Diadopsi dari sumber :

No	Judul	Pengarang	Penerbit	Edisi	Kota	Tahun	Jenis
1	C++ How to Program	Paul Deitel, Harvey Deitel	Pearson Education, Inc.	10	Hoboken, New Jersey	2017	Utama
2	Algoritma dan Pemrograman dalam Bahasa Pascal, C, dan C++	Rinaldi Munir, Leorny Lidya	Penerbit Informatika	6	Bandung	2016	Pendukung

Referensi tambahan selain yang tercantum di RPS



Sub-CPMK

Mahasiswa mampu menjelaskan konsep bahasa pemrograman C++ (C2, A2)

Materi

1. Perbedaan bahasa C dan C++
2. Pengenalan lingkungan pengembangan C++
3. Compiler bahasa C++
4. Pengenalan library bahasa C++
5. Operator bahasa C++
6. Translasi pseudocode ke bahasa C++

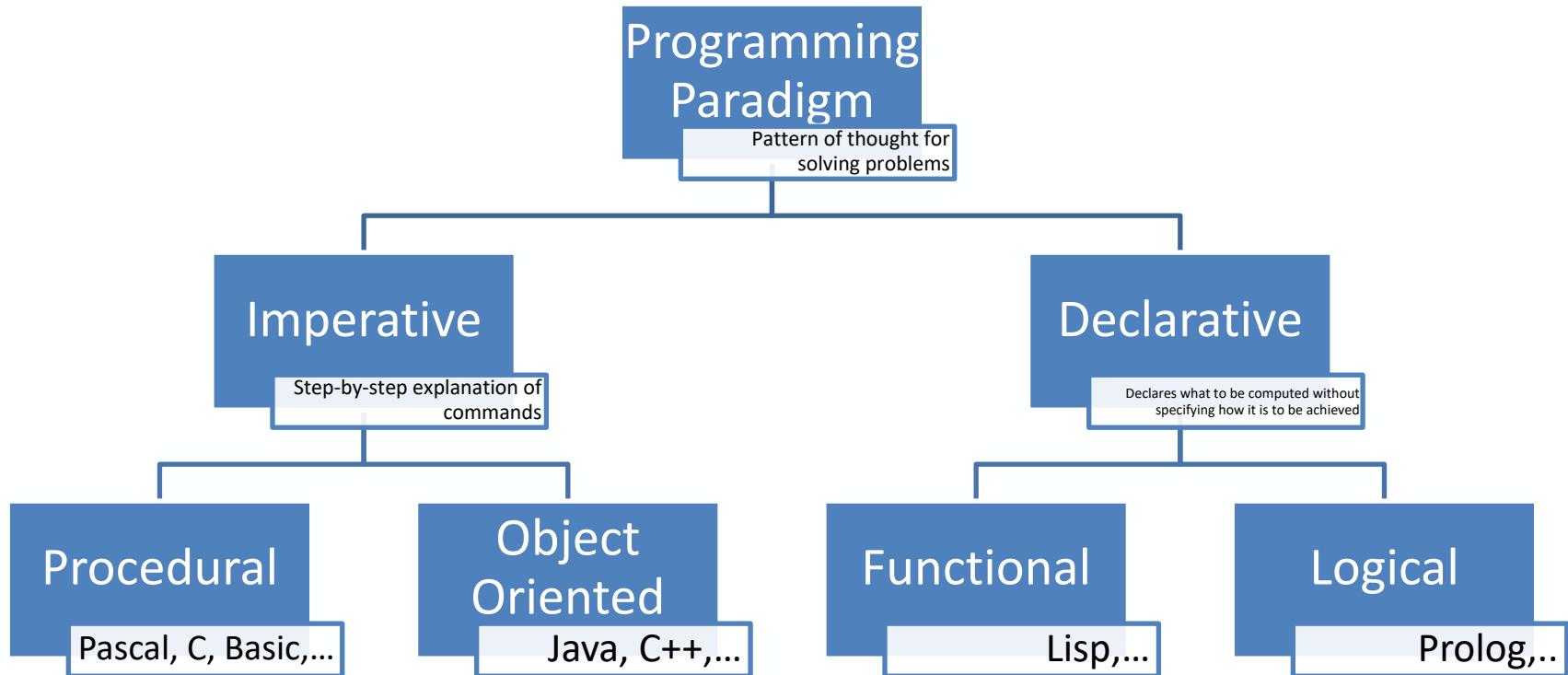
Referensi

1. Referensi 1, Bab 1-2 (hal 1-63), Bab 5 poin 5.11-5.12 (hal 188-193)
2. Referensi 2, Bab 5, hal 81-94



1. Perbedaan bahasa C dan C++

1.1 Paradigma



1.1 Paradigma (Lanj..)



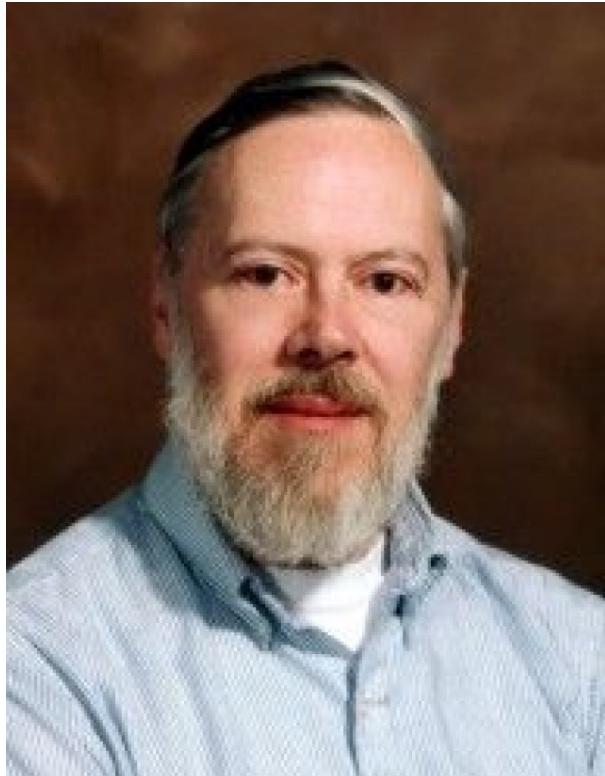
Rank	Language	Type	Score
1	Python ▾	🌐💻⚙️	100.0
2	Java ▾	🌐📱💻	95.3
3	C ▾	📱💻⚙️	94.6
4	C++ ▾	📱💻⚙️	87.0
5	JavaScript ▾	🌐	79.5
6	R ▾	💻	78.6
7	Arduino ▾	⚙️	73.2
8	Go ▾	🌐💻	73.1
9	Swift ▾	📱💻	70.5
10	Matlab ▾	💻	68.4

Top Programming Language 2020 (22 Juli 2020)

Sumber:

<https://spectrum.ieee.org/at-work/tech-careers/top-programming-language-2020>

1.2 Sejarah



Dennis Ritchie

Creator of C



Bjarne Stroustrup

Creator of C++

(<https://www.stroustrup.com>)

1.2 Sejarah (Lanj..)

- Perkembangan Bahasa Pemrograman C++ tidak lepas dari Bahasa Pemrograman C, sebagai pendahulunya.
- Pada sekitar tahun 1972 Bahasa Pemrograman C ditemukan oleh Brian W. Kerninghan dan Dennis M. Ritchie, dan sekitar satu dekade kemudian lahirlah C++, oleh Bjarne Stroustrup dari Laboratorium Bell, AT&T, pada tahun 1983.
- C++ cukup kompatibel dengan bahasa pendahulunya C. Pada mulanya C++ disebut “a better C”. Nama C++ sendiri diberikan oleh Rick Mascitti pada tahun 1983, yang berasal dari operator increment pada bahasa C.
- Keistimewaan yang sangat berarti dari C++ ini adalah karena bahasa ini mendukung pemrograman yang berorientasi objek (OOP / Object Oriented Programming).

1.2 Sejarah (Lanj..)

1.1.1 Pemrograman Berorientasi Objek

Sekilas tentang: **PEMROGRAMAN BERORIENTASI OBJEK**

- Sampai saat ini, program dianggap sebagai sekumpulan *procedure* yang melakukan aksi terhadap data.
- *Procedure* atau *function*, adalah suatu set instruksi khusus yang dieksekusi secara bergantian. Data terpisah dari *procedure*, dan trik pemrogramannya adalah menjaga urutan pemanggilan fungsi, dan data apa yang diubah. Dalam demikian terciptalah program yang terstruktur.

1.2 Sejarah (Lanj..)

1.1.1 Pemrograman Berorientasi Objek

- Ide dari pemrograman terstruktur adalah memecah program yang besar menjadi kecil sehingga lebih mudah dipahami. Program-program lama memaksa pengguna untuk melakukan langkah-perlangkah melalui layar monitor. Sedangkan program modern menyajikan semua pilihan sekaligus dan merespon aksi pengguna.
- OOP berusaha untuk memenuhi kebutuhan itu, menyediakan teknik untuk mengelola kompleksitas, mencatat penggunaan ulang komponen *software*, dsb. Inti dari OOP adalah memperlakukan data dan procedure sebagai sebuah objek berisi entitas dengan identitas dan ciri yang khusus.

1.2 Sejarah (Lanj..)

1.1.2 C++ dan Pemrograman Berorientasi Objek

- Tiga karakteristik utama dari bahasa yang berorientasi objek adalah
 1. Encapsulation
 2. Inheritance
 3. Polymorphism.
- Tiga ciri diatas mendukung *reusability*, yang merupakan salah satu faktor penentu kualitas *software*.

1.2 Sejarah (Lanj..)

1.1.2 C++ dan Pemrograman Berorientasi Objek

- C++ mendukung karakteristik ***encapsulation*** dengan menggunakan konsep **class**. Setelah terbentuk, maka **class** akan bertindak sebagai entitas yang tenkapsulasi.
- Dengan adanya konsep ***inheritance***, maka C++ mendukung ide penggunaan ulang suatu object.
- ***Polymorphisme*** (banyak bentuk) merupakan suatu konsep yang menyatakan sesuatu yang sama dapat memiliki berbagai bentuk dan perilaku yang berbeda.

1.2 Perbandingan

(Sumber: <https://hackr.io/blog/difference-between-c-and-cplusplus>, April 2020)

PARAMETERS	C	C++
Development Area	Embedded devices and system-level code.	Gaming, networking and server-side applications.
Approach	C follows a top-down approach. It begins with a high-level design and ends with a low-level design.	C++ follows a bottom-up approach. Opposed to the top-down approach, the bottom-up approach starts with the low-level design and finishes with the high-level design.
Compatibility with each other.	The C compiler isn't able to execute the C++ code.	C++ is a superset of C so C++ is able to run most C code.
Compatibility with other programming languages	No such feature is offered by C.	It provides compatibility with other generic programming languages.

1.2 Perbandingan (Lanj..)

(Sumber: <https://hackr.io/blog/difference-between-c-and-cplusplus>, April 2020)

PARAMETERS	C	C++
Data Security, Encapsulation, and Information Hiding	It does not support encapsulation. Not good for securing data.	Supports encapsulation as it is object-oriented. Good for securing data.
Variables	No such feature is offered by C.	C++ is capable of hiding variables in a class while offering only a function interface.
Use of Strings	The char[] represents string literals in C.	C++ has a variable type called a string.
Data Types	Supports only built-in and primitive data types.	Along with built-in and primitive data types, it also supports user-defined data types.

1.2 Perbandingan (Lanj..)

(Sumber: <https://hackr.io/blog/difference-between-c-and-cplusplus>, April 2020)

PARAMETERS	C	C++
Default Header File	stdio.h	iostream.h
Exception Handling	Provides no support for exception handling.	Provides try and catch block for the exception handling.
File Extension	Files are saved with .c extension.	Files are saved with .cpp extension.
Functions with Default Arguments	It does not allow using functions with default arguments.	Allows using functions with default arguments.
GUI Programming	It has GTK tool for enabling GUI programming.	It has QT tool for enabling GUI programming.

1.2 Perbandingan (Lanj..)

(Sumber: <https://hackr.io/blog/difference-between-c-and-cplusplus>, April 2020)

PARAMETERS	C	C++
Inheritance	It does not support inheritance.	Supports inheritance as it is object-oriented.
Input and Output Functions	It uses scanf() and printf() for input and output operations.	It uses cin and cout for input and output operations.

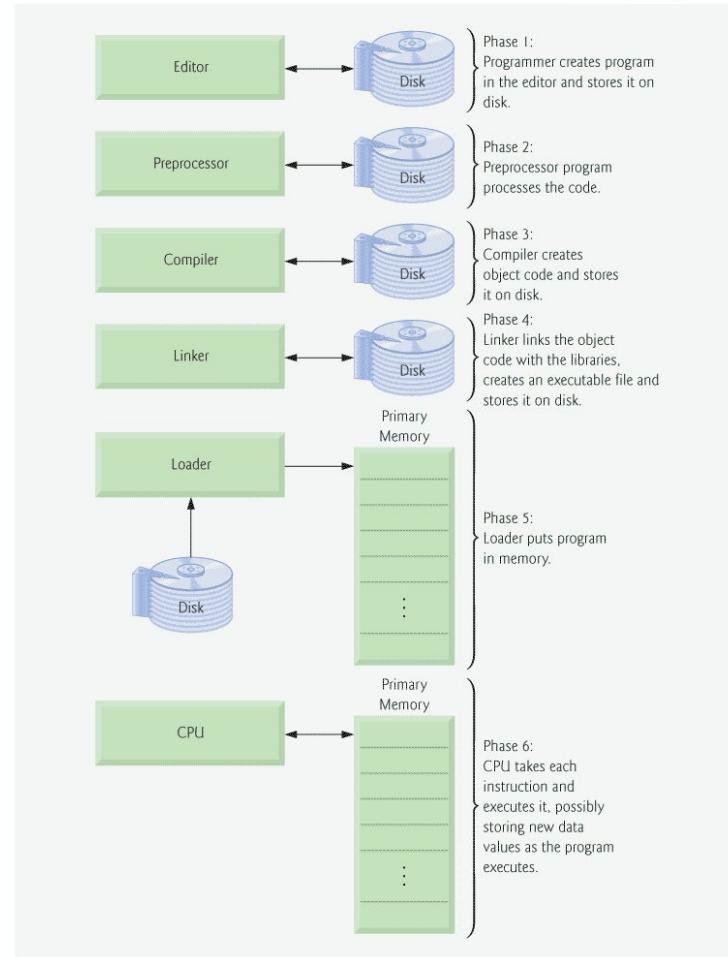


2. Pengenalan lingkungan pengembangan C++

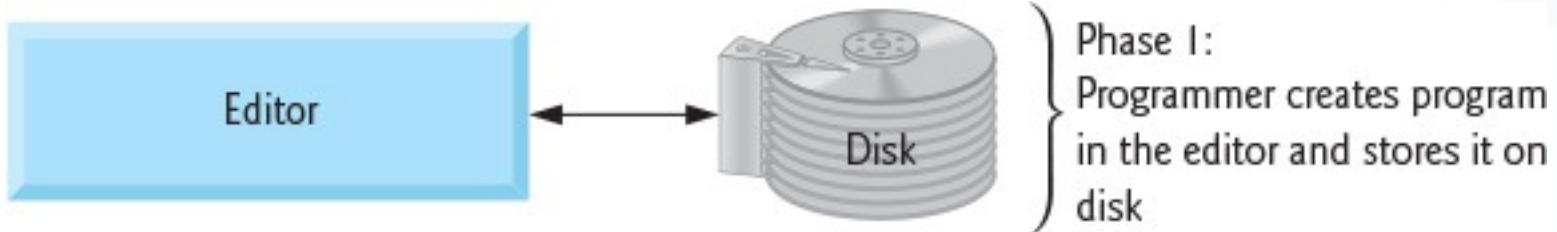
Secara umum, C++ terdiri dari 3 bagian, yaitu *a program development environment* (lingkungan pengembangan program), *the language* dan *C++ standard library*.

2.1 Lingkungan Pengembangan

1. Edit
2. Preprocess
3. Compile
4. Link
5. Load
6. Execute

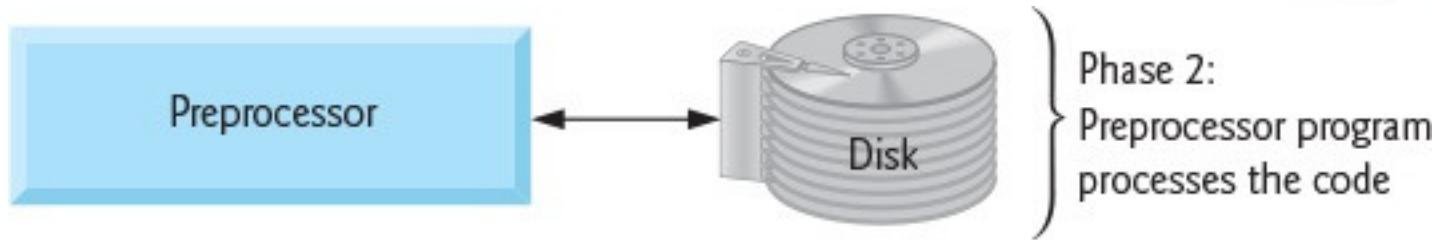


2.1.1 Editing a Program



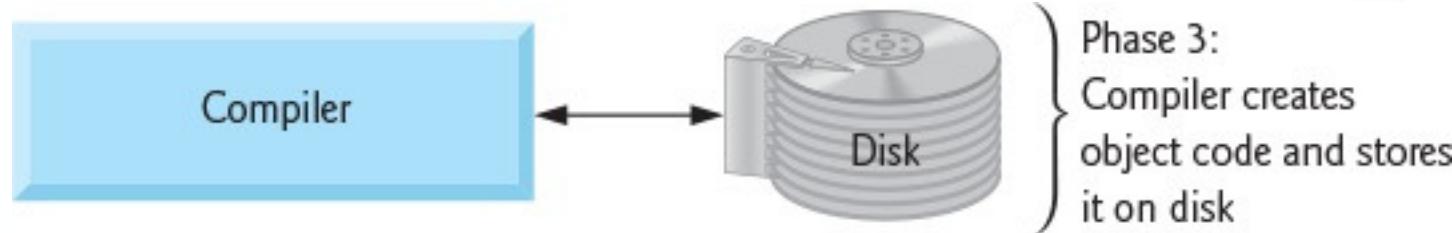
- Fase 1 adalah membuat/mengedit program menggunakan editor. Program C++ (yang dinamakan source code) akan di simpan di komputer (disk).
- Extension dari program C++ adalah .cpp, .cxx, .cc atau .C (huruf besar C)

2.1.2 Preprocessing a C++ Program



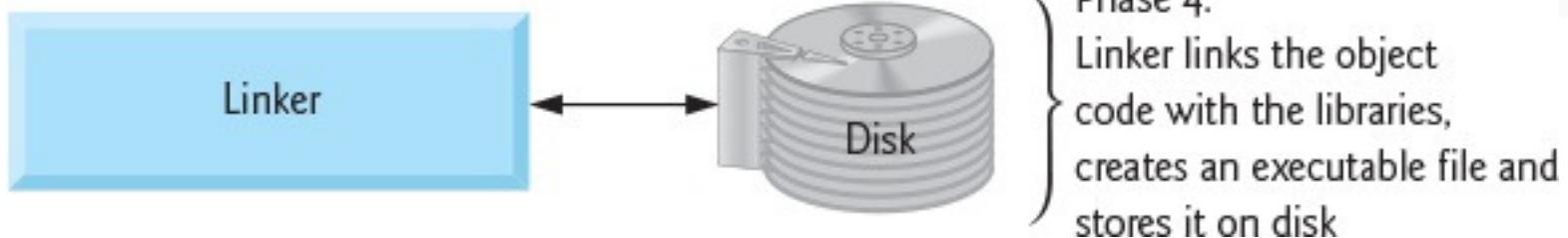
- Pada fase 2, diberikan perintah untuk melakukan kompilasi terhadap program yang sudah ditulis.
- Statement yang dimulai dengan simbol # (misalnya `#include <iostream>`) disebut sebagai preprocessor directive. Statement inilah yang akan diproses oleh **preprocessor**

2.1.3 Compiling a C++ Program



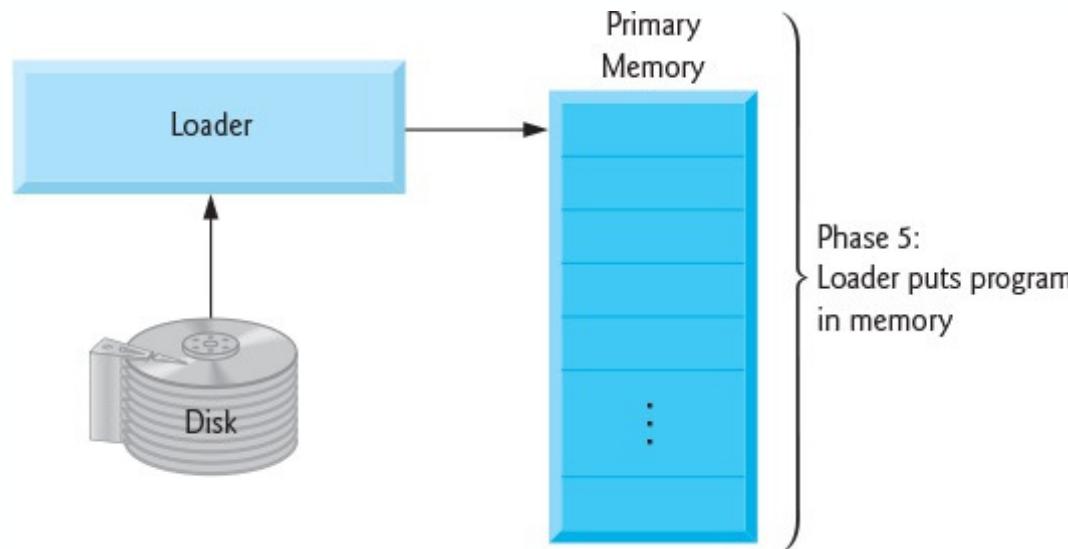
- Setelah pemrosesan preprocessor directive, langkah selanjutnya adalah memeriksa apakah program secara sintaksis sudah sesuai dengan kaidah bahasa pemrograman C++.
- Lalu compiler menerjemahkan program C++ ke dalam bahasa mesin (*machine-learning code*), yang biasa dikenal sebagai *object program*.

2.1.4 Linking



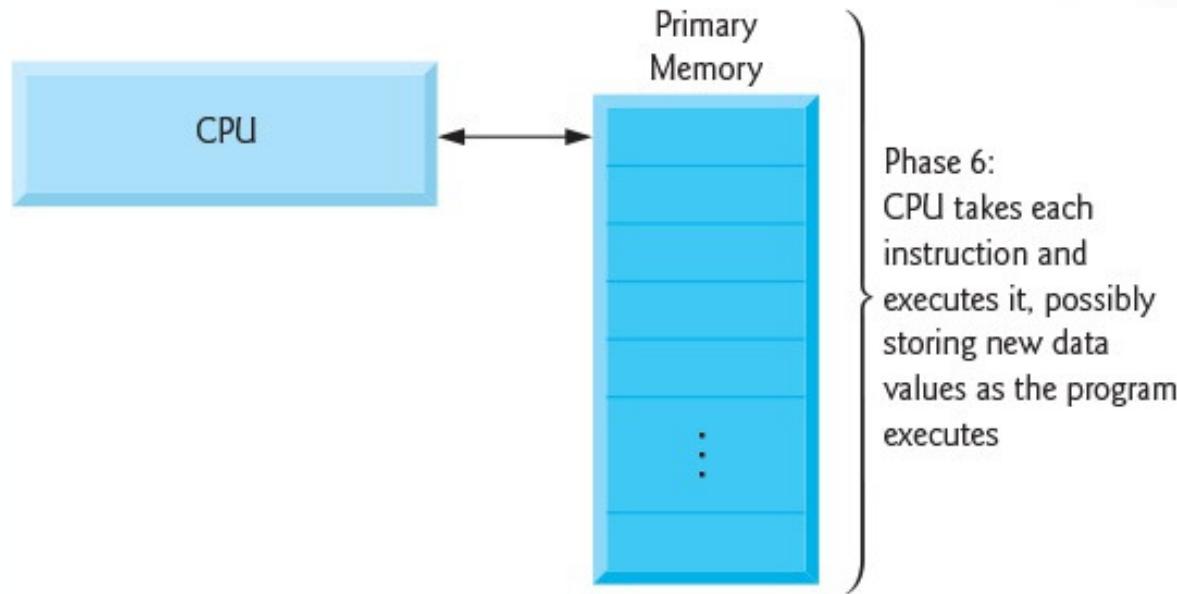
- Menghubungkan program C++ dengan fungsi ataupun standard/private library lainnya jika ada.
- Apabila program yang dikompilasi dan links sudah benar, maka akan dihasilkan executable image. Biasanya ekstensionnya berupa .out

2.1.5 Loading



- Sebelum program dapat dieksekusi, maka program harus ditempatkan di main memory.
- Loader akan mengambil executable image dari disk dan transfer ke memory, termasuk mengambil komponen lainnya yang dibutuhkan

2.1.6 Execution



- Terakhir, komputer (CPU) akan mengeksekusi satu per satu instruksinya dalam satu waktu.
- Beberapa komputer dengan arsitektur modern, dapat melakukannya secara paralel

2.2 Proses Penyelesaian Masalah

1. Analisis masalah dan solusi yang dibutuhkan, serta merancang algoritma yang tepat untuk menyelesaikan masalah tersebut.
2. Mengimplementasikan bahasa pemrograman, misalnya C++, dan verifikasi algoritma
3. Pemeliharaan program dengan cara modifikasi apabila terdapat perubahan.

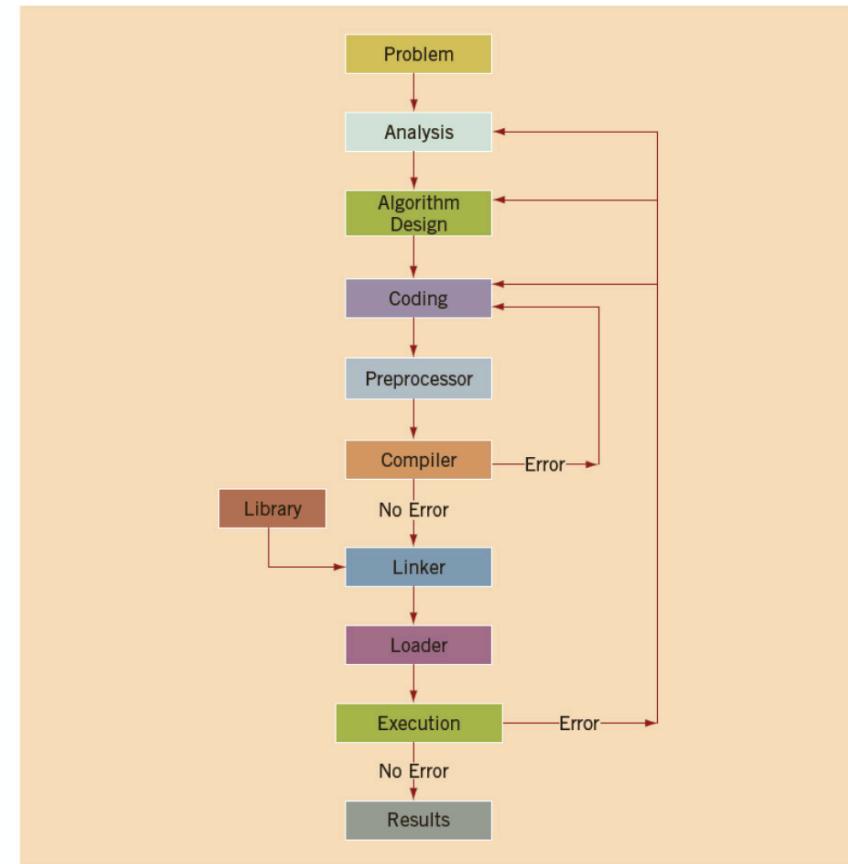


FIGURE 1-3 Problem analysis-coding-execution cycle

- Referensi lainnya: Buku D.S. Malik, hlm. 9-13



3. Compiler bahasa C++

- Pseudo-code di dalam sebuah algoritma bukan kode program sebenarnya sehingga tidak dapat dieksekusi oleh komputer.
- Agar dapat dijalankan oleh komputer, maka algoritma dalam notasi pseudo-code harus diterjemahkan ke dalam bahasa pemrograman dinamakan proses **translasi**.
- Dalam hal ini adalah translasi ke dalam bahasa pemrograman C++, menggunakan compiler.

2.1 Compiler

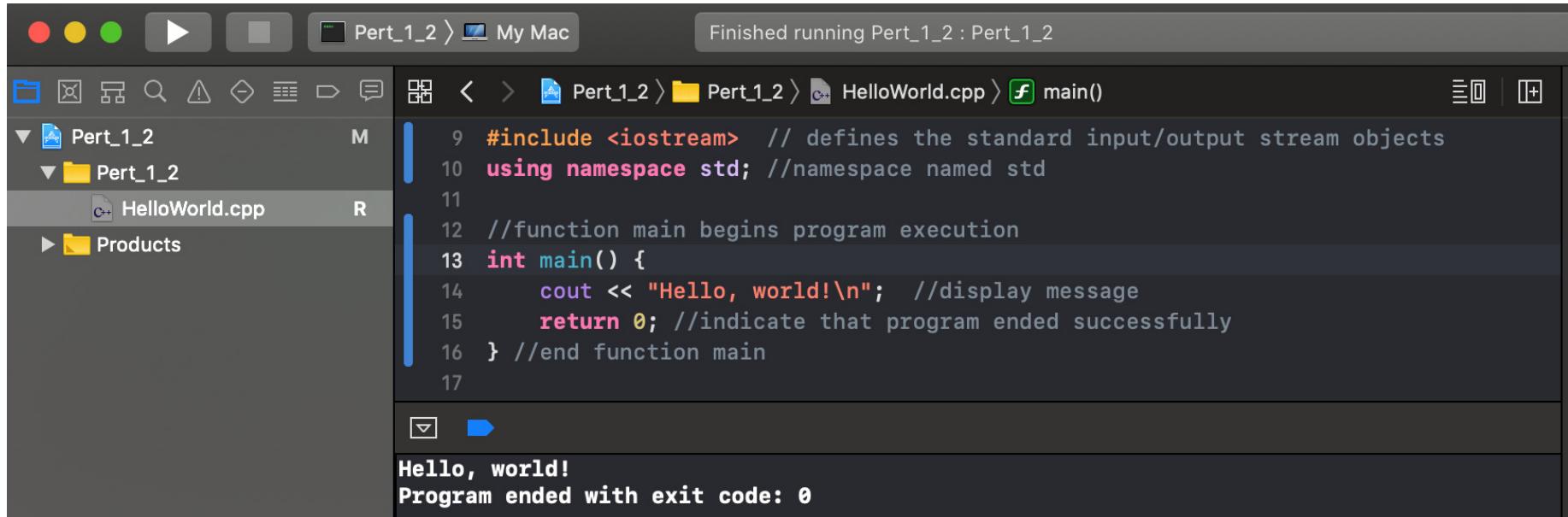


- Ada banyak IDEs (Integrated development environments) yang tersedia.
- Beberapa yang terkenal: Ms. Visual Studio, NetBeans, Eclipse, Apple's Xcode, dsb...

2.1 Compiler (Lanj..)



Xcode



Pert_1_2 > Pert_1_2 > HelloWorld.cpp > main()

```
9 #include <iostream> // defines the standard input/output stream objects
10 using namespace std; //namespace named std
11
12 //function main begins program execution
13 int main() {
14     cout << "Hello, world!\n"; //display message
15     return 0; //indicate that program ended successfully
16 } //end function main
17
```

Finished running Pert_1_2 : Pert_1_2

```
Hello, world!
Program ended with exit code: 0
```

2.1 Compiler (Lanj..)



The screenshot shows a web-based C++ compiler interface. The browser address bar displays 'onlinegdb.com'. The interface includes a toolbar with buttons for Run, Debug, Stop, Share, Save, and Beautify, along with a Language selector set to C++. A file named 'main.cpp' is open in the editor. The code is as follows:

```
1 //*****
2
3 //          Online C++ Compiler.
4 //          Code, Compile, Run and Debug C++ program online.
5 // Write your code in this editor and press "Run" button to compile and execute it.
6
7 ****
8
9 #include <iostream>
10 using namespace std;
11
12 int main()
13 {
14     cout<<"Hello World";
15     return 0;
16 }
```

The output window shows the result of running the program: 'Hello World' followed by the message '...Program finished with exit code 0' and 'Press ENTER to exit console.'

2.2 Demo Penjelasan IDE

<<bagian ini adalah simulasi C++ environment menggunakan compiler tertentu>>

Mencakup:

1. Instalasi
2. Menjalankan aplikasi (Built, Run, Debug, dsb)
3. Pengenalan jendela dari IDE.



4. Pengenalan library bahasa C++

4.1 Standard iostream library header

- `iostream` (`cin`, `cout`)
- `iomanip` (`setw`, `setprecision`)
- `fstream` (`ofstream`, `ifstream`,...)
- ...

4.2 Math Library Function

Function	Header File	Purpose	Parameter(s) Type	Result
abs (x)	<cmath>	Returns the absolute value of its argument: abs (- 7) = 7	int (double)	int (double)
ceil (x)	<cmath>	Returns the smallest whole number that is not less than x: ceil (56.34) = 57.0	double	double
cos (x)	<cmath>	Returns the cosine of angle x: cos (0.0) = 1.0	double (radians)	double
exp (x)	<cmath>	Returns e^x , where e = 2.718: exp (1.0) = 2.71828	double	double

4.2 Math Library Function (Lanj..)

Function	Header File	Purpose	Parameter(s) Type	Result
<code>fabs (x)</code>	<code><cmath></code>	Returns the absolute value of its argument: <code>fabs (-5.67) = 5.67</code>	<code>double</code>	<code>double</code>
<code>floor (x)</code>	<code><cmath></code>	Returns the largest whole number that is not greater than <code>x</code> : <code>floor(45.67) = 45.00</code>	<code>double</code>	<code>double</code>
<code>islower (x)</code>	<code><cctype></code>	Returns <code>true</code> if <code>x</code> is a lowercase letter; otherwise it returns <code>false</code> ; <code>islower('h')</code> is <code>true</code>	<code>int</code>	<code>int</code>
<code>isupper (x)</code>	<code><cctype></code>	Returns <code>true</code> if <code>x</code> is a uppercase letter; otherwise it returns <code>false</code> ; <code>isupper('K')</code> is <code>true</code>	<code>int</code>	<code>int</code>
<code>pow (x, y)</code>	<code><cmath></code>	Returns x^y ; If <code>x</code> is negative, <code>y</code> must be a whole number: <code>pow(0.16, 0.5) = 0.4</code>	<code>double</code>	<code>double</code>
<code>sqrt (x)</code>	<code><cmath></code>	Returns the nonnegative square root of <code>x</code> , <code>x</code> must be nonnegative: <code>sqrt(4.0) = 2.0</code>	<code>double</code>	<code>double</code>
<code>tolower (x)</code>	<code><cctype></code>	Returns the lowercase value of <code>x</code> if <code>x</code> is uppercase; otherwise, returns <code>x</code>	<code>int</code>	<code>int</code>
<code>toupper (x)</code>	<code><cctype></code>	Returns the uppercase value of <code>x</code> if <code>x</code> is lowercase; otherwise, returns <code>x</code>	<code>int</code>	<code>int</code>



5. Operator bahasa C++

atau

**Dasar-dasar yang Perlu
Diketahui**

5.1 Comments

- Komentar digunakan untuk memberikan informasi kepada *user*, bukan *compiler*.
- Tipe:

```
21 //Ini adalah single-line comment,  
22 //sedangkan di bawah ini adalah:  
23 /*  
24     Multiple Line  
25 */
```

5.2 Special Symbols

+	?
-	,
*	<=
/	!=
.	==
;	>=

5.3 Reserved Words (Keywords)

C++ Keywords				
<i>Keywords common to the C and C++ programming languages</i>				
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			
<i>C++-only keywords</i>				
and	and_eq	asm	bitand	bitor
bool	catch	class	compl	const_cast
delete	dynamic_cast	explicit	export	false
friend	inline	mutable	namespace	new
not	not_eq	operator	or	or_eq
private	protected	public	reinterpret_cast	static_cast
template	this	throw	true	try
typeid	typename	using	virtual	wchar_t
xor	xor_eq			
<i>C++11 keywords</i>				
alignas	alignof	char16_t	char32_t	constexpr
decltype	noexcept	nullptr	static_assert	thread_local

5.4 Identifier

- Terdiri dari huruf, angka, dan karakter underscore (_)
- Harus diawali dengan huruf atau underscore
- C++ adalah case sensitive
 - NUMBER tidak sama dengan number
- Dua predefined identifiers adalah cout dan cin

5.4 Identifier (Lanj..)

Contoh illegal identifier:

Illegal Identifier	Reason	A Correct Identifier
employee Salary	There can be no space between employee and Salary .	employeeSalary
Hello!	The exclamation mark cannot be used in an identifier.	Hello
one + two	The symbol + cannot be used in an identifier.	onePlusTwo
2nd	An identifier cannot begin with a digit.	second

5.5 Whitespaces

- Semua program C++ terdiri dari whitespaces
 - termasuk blanks, tabs, and newline characters
- Digunakan untuk memisahkan special symbols, reserved words, dan identifiers
- Ketepatan penggunaan whitespaces sangatlah penting
 - Tujuannya agar program mudah dibaca

5.6 Data Types

- Tipe data: sekumpulan nilai yang dapat dikenakan operasi
- Tiga kategori tipe data:
 - Simple
 - Structured
 - Pointers

5.6 Data Types (Lanj..)

- Tiga kategori simple data:
 - Integral: integers (bilangan bulat)
 - Floating-point: bilangan desimal
 - Enumeration type: tipe data yang didefinisikan oleh user (user-defined)

5.6 Data Types (Lanj..)

Nilai dan alokasi memori untuk tipe data simple:

Data Type	Values	Storage (in bytes)
<code>int</code>	$-2147483648 (= -2^{31})$ to $2147483647 (= 2^{31} - 1)$	4
<code>bool</code>	<code>true</code> and <code>false</code>	1
<code>char</code>	$-128 (= -2^7)$ to $127 (= 2^7 - 1)$	1
<code>long long</code>	$-9223372036854775808 (-2^{63})$ to $9223372036854775807(2^{63} - 1)$	64

5.6 Data Types (Lanj..)

5.6.1 Integral Data Type

- Sembilan kategori dari tipe data Integral:
 1. Char
 2. Short
 3. Int
 4. Long
 5. Bool
 6. Unsigned char
 7. Unsigned short
 8. Unsigned int
 9. Unsigned long

5.6 Data Types (Lanj..)

5.6.1 Integral Data

- Contoh:
 - 6728
 - 0
 - 78
 - +763
- Positive integers tidak menggunakan + sign
- Tidak menggunakan koma
 - Koma hanya digunakan untuk tanda pemisah di list

5.6 Data Types (Lanj..)

5.6.2 Bool Data Type

- Tipe `bool`
 - Dua nilainya: `true` dan `false`
 - Dapat dimanipulasi menggunakan ekspresi logical (Boolean)
- `true` dan `false` disebut sebagai logical values
- `bool`, `true`, dan `false` merupakan reserved words

5.6 Data Types (Lanj..)

5.6.3 Char Data Type

- Merupakan tipe data integral yang terkecil
- Digunakan untuk characters: huruf, angka, and karakter khusus
- Masing-masing karakter diapit oleh tanda petik satu
 - 'A', 'a', '0', '*', '+', '\$', '&'
- Blank space dianggap sebagai karakter dan ditulis ' ', dengan spasi kosong diantara petik satu.

5.6 Data Types (Lanj..)

5.6.2 Floating-Point Data Type

TABLE 2-3 Examples of Decimal Numbers in Scientific and C++ Floating-Point Notations

Decimal Number	Scientific Notation	C++ Floating-Point Notation
75.924	7.5924 * 10 ¹	7.592400E1
0.18	1.8 * 10 ⁻¹	1.800000E-1
0.0000453	4.53 * 10 ⁻⁵	4.530000E-5
-1.482	-1.482 * 10 ⁰	-1.482000E0
7800.0	7.8 * 10 ³	7.800000E3

C++ menggunakan notasi khusus untuk merepresentasikan bilangan real (floating-point notation)

5.6 Data Types (Lanj..)

5.6.2 Floating-Point Data Type

- `float`: merepresentasikan bilangan real
 - Range: -3.4E+38 to 3.4E+38 (empat bytes)
- `double`: merepresentasikan bilangan real
 - Range: -1.7E+308 to 1.7E+308 (delapan bytes)
- Mayoritas untuk compiler baru, tipe data `double` dan `long double` dianggap sama

5.7 Arithmetic Operators and Operator Precedence

- Operator aritmatika dalam C++:
 - + addition
 - - subtraction
 - * multiplication
 - / division
 - % modulus operator
- +, -, *, and / dapat digunakan untuk tipe data integral dan floating-point
- Operators dapat berupa unary atau binary

5.8 Order of Precedence

- Semua operator di dalam () akan dievaluasi terlebih dahulu
- *, /, dan % berada di level yang sama dan dievaluasi terlebih dahulu
- + dan – berada di level yang sama dan dievaluasi kemudian
- Ketika semua operator berada di level yang sama
 - Maka dievaluasi dari kiri ke kanan (associativity)
- $3 * 7 - 6 + 2 * 5 / 4 + 6$ berarti:
 $((3 * 7) - 6) + ((2 * 5) / 4)) + 6$

5.9 String Type

- Programmer-defined type menggunakan ANSI/ISO Standard C++ library
- Diapit dengan tanda petik
- Null: sebuah string yang tidak memiliki karakter
- Setiap karakter menempati posisi tertentu di string
 - Posisi dari karakter pertama adalah 0
- Panjang string adalah jumlah karakter yang ada di dalamnya
 - Contoh: Panjang string "William Jacob" adalah 13

5.9 String Type (Lanj..)

Contoh Lainnya:

String	Position of a Character in the String	Length of the String
"William Jacob"	Position of 'w' is 0. Position of the first 'i' is 1. Position of '' (the space) is 7. Position of 'J' is 8. Position of 'b' is 12.	13
"Mickey"	Position of 'M' is 0. Position of 'i' is 1. Position of 'c' is 2. Position of 'k' is 3. Position of 'e' is 4. Position of 'y' is 5.	6

The length of the following string is 22.

"It is a beautiful day."

5.10 Input

- Data harus di-load ke main memory sebelum dimanipulasi
- Penyimpanan data ke dalam memory dilakukan dengan 2 cara:
 - Instruct computer to allocate memory
 - Include statements to put data into memory

5.11 Allocating Memory with Constants and Variables

- Named constant: lokasi memori dimana content disimpan tidak berubah selama eksekusi
- Sintaks untuk mendeklarasikan konstanta:

```
const dataType identifier = value;
```
- Di C++, `const` merupakan reserved word

Consider the following C++ statements:

```
const double CONVERSION = 2.54;
const int NO_OF_STUDENTS = 20;
const char BLANK = ' ';
const double PAY_RATE = 15.75;
```

5.11 Allocating Memory with Constants and Variables (Lanj..)

- Variable: lokasi memory tempat penyimpanan konten dapat berubah
- Sintaksnya adalah sebagai berikut:

```
dataType identifier, identifier, . . .;
```

Consider the following statements:

```
double amountDue;  
int counter;  
char ch;  
int x, y;  
string name;
```

5.12 Putting Data into Variables

- Cara mendeklarasikan variable:
 - Use C++'s assignment statement
 - Use input (read) statements

5.12 Putting Data into Variables (Lanj..)

5.12.1 Assignment Statement

- Sintaks untuk assignment statement:

```
variable = expression;
```

- Expression dievaluasi, dan nilainya di-assign ke variable di sisi kiri
- Dalam C++, = merupakan operator penugasan (assignment operator)

5.12 Putting Data into Variables (Lanj..)

5.12.1 Assignment Statement

Suppose you have the following variable declarations:

```
int num1, num2;  
double sale;  
char first;  
string str;
```

Now consider the following assignment statements:

```
num1 = 4;  
num2 = 4 * 5 - 11;  
sale = 0.02 * 1000;  
first = 'D';  
str = "It is a sunny day.;"
```

5.12 Putting Data into Variables (Lanj..)

5.12.1 Assignment Statement

- Variabel dapat diinisialisasi ketika dideklasikan:

```
int first=13, second=10;  
char ch=' ';  
double x=12.6;
```

- Semua variable harus dideklarasikan sebelum digunakan
 - Tetapi tidak menutup kemungkinan, dilakukan selama proses deklarasi

5.12 Putting Data into Variables (Lanj..)

5.12.1 Assignment Statement

- C++ memiliki spesial assignment statements yang disebut compound assignments
 $+=$, $-=$, $*=$, $/=$, and $\%=$
- Contoh:

```
x *= y;
```

5.12 Putting Data into Variables (Lanj..)

5.12.2 Input (Read) Statement

- `cin` menggunakan `>>` untuk menerima input

```
cin >> variable >> variable ...;
```

- Stream extraction operator nya adalah `>>`
- Contoh, jika `miles` adalah variable bertipe `double`
`cin >> miles;`
 - Komputer akan menerima nilai dari tipe `double`
 - Simpan nilai tersebut ke dalam variable `miles`

5.12 Putting Data into Variables (Lanj..)

5.12.2 Input (Read) Statement

- `cin` dapat menerima lebih dari satu nilai
- Contohnya, jika `feet` dan `inches` adalah tipe data `int`, statement nya adalah:

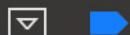
```
cin >> feet >> inches;
```

- Input 2 nilai integer dari keyboard
- Simpan masing-masing ke dalam variable `feet` dan `inches`

5.12 Putting Data into Variables (Lanj..)

5.12.2 Input (Read) Statement

```
9 //Program yang menggunakan "cin" sebagai input
10 #include <iostream>
11 using namespace std;
12
13 int main(){
14     int feet;
15     int inches;
16
17     cout << "Masukkan 2 buah nilai integers, pisahkan dengan spasi: ";
18     cin >> feet >> inches;
19     cout << endl;
20
21     cout << "Feet = " << feet << endl;
22     cout << "Inches = " << inches << endl;
23     return 0;
24 }
```



Masukkan 2 buah nilai integers, pisahkan dengan spasi: 1 2

Feet = 1
Inches = 2

Program ended with exit code: 0

5.13 Variable Initialization

- Ada 2 cara untuk inisialisasi variabel

```
int feet;
```

- Menggunakan *assignment statement*

```
feet = 35;
```

- Menggunakan *read statement*

```
cin >> feet;
```

5.14 Increment & Decrement Operators

- Increment operator: increment variable dengan 1
 - Pre-increment: `++variable`
 - Post-increment: `variable++`
- Decrement operator: decrement variable dengan 1
 - Pre-decrement: `--variable`
 - Post-decrement: `variable-`
- Apakah perbedaan dari keduanya?

```
x = 5;  
y = ++x;
```

```
x = 5;  
y = x++;
```

5.15 Output

- Sintaks cout dan << adalah:

```
cout << expression or manipulator << expression or manipulator...;
```

- Dikenal sebagai output statement
- Stream insertion operator adalah <<
- Expression akan dievaluasi, dan nilainya akan ditampilkan pada screen.

5.15 Output (Lanj..)

- Manipulator digunakan untuk format keluaran
 - Contoh: endl untuk memulai baris baru

```
9 //menampilkan "cout" sebagai output program
10 #include <iostream>
11 using namespace std;
12
13 int main(){
14     cout << 3 / 2 << endl;
15     cout << "Ohai!" << endl;
16     cout << 10 << endl;
17     cout << "3 + 2" << endl;
18     cout << 3 + 2 << endl;
19     cout << "3 + 2 = " << 3 + 2 << endl;
20     cout << 'Z' << endl;
21     cout << 3 + 2 * 4 << endl;
22     cout << "Ohai! \nApa Kabar?" << endl;
23     return 0;
24 }
```

OUTPUT:

```
1
Ohai!
10
3 + 2
5
3 + 2 = 5
Z
11
Ohai!
Apa Kabar?
Program ended with exit code: 0
```

5.15 Output (Lanj..)

- Baris baru menggunakan '\n'
 - Dapat diletakkan di posisi mana pun.

```
cout << "Hello there.";  
cout << "My name is James.;"
```

- **Output:**

Hello there.My name is James.

```
cout << "Hello there.\n";  
cout << "My name is James.;"
```

- **Output :**

Hello there.

My name is James.

5.15 Output (Lanj..)

TABLE 2-4 Commonly Used Escape Sequences

	Escape Sequence	Description
\n	Newline	Cursor moves to the beginning of the next line
\t	Tab	Cursor moves to the next tab stop
\b	Backspace	Cursor moves one space to the left
\r	Return	Cursor moves to the beginning of the current line (not the next line)
\\"	Backslash	Backslash is printed
\'	Single quotation	Single quotation mark is printed
\"	Double quotation	Double quotation mark is printed

5.16 Preprocessor Directives

- C++ memiliki beberapa operations
- Banyak fungsi dan symbol yang dibutuhkan untuk menjalankan program C++ disediakan oleh library.
- Setiap library memiliki nama dan mengacu ke header file.
- Preprocessor directives merupakan commands yang diberikan kepada preprocessor
- Semua preprocessor commands dimulai dengan #
- Pada akhir commands, tidak ada tanda titik koma (;

5.16 Preprocessor Directives (Lanj..)

- Sintaks yang memasukkan sebuah header file:

```
#include <headerFileName>
```

- Contoh:

```
#include <iostream>
```

- Menyebabkan preprocessor memasukkan header file iostream ke dalam program

5.17 namespace and Using cin and cout in a Program

- **cin** dan **cout** dideklarasikan pada header file **iostream**, tetapi di dalam **std** namespace
- Untuk menggunakan **cin** dan **cout** dalam program, gunakan 2 statement berikut:

```
#include <iostream>  
using namespace std;
```

5.18 Using the string Data Type in a Program

- Untuk menggunakan tipe string, gunakan header file string
- Masukkan ke dalam preprocessor directive:
`#include <string>`



6. Translasi pseudocode ke bahasa C++

6.1 Creating a C++ Program

- C++ memiliki 2 bagian utama
 - Preprocessor directives
 - The program
- Preprocessor directives dan program statements membentuk C++ source code (.cpp)
- Compiler generates object code (.obj)
- Executable code dihasilkan dan disimpan ke dalam file dengan ekstensi .exe

6.1 Creating a C++ Program (Lanj)

- C++ program merupakan sekumpulan fungsi, salah satunya adalah function main
- Baru pertama dari fungsi main disebut *heading of the function*:
`int main()`
- Statement berada di dalam kurung kurawal({ and }), yaitu *body of the function*
 - Terdapat 2 jenis statements:
 - Declaration statements
 - Executable statements

```
*****  
// Author: D.S. Malik  
//  
// This program shows where the include statements, using  
// statement, named constants, variable declarations, assignment  
// statements, and input and output statements typically appear.  
*****  
  
#include <iostream>                                //Line 1  
  
using namespace std;                               //Line 2  
  
const int NUMBER = 12;                            //Line 3  
  
int main()                                         //Line 4  
{                                                 //Line 5  
    int firstNum;                                //Line 6  
    int secondNum;                               //Line 7  
  
    firstNum = 18;                                //Line 8  
    cout << "Line 9: firstNum = " << firstNum  
        << endl;                                 //Line 9  
  
    cout << "Line 10: Enter an integer: ";       //Line 10  
    cin >> secondNum;                           //Line 11  
    cout << endl;                               //Line 12  
  
    cout << "Line 13: secondNum = " << secondNum  
        << endl;                                //Line 13  
  
    firstNum = firstNum + NUMBER + 2 * secondNum; //Line 14  
  
    cout << "Line 15: The new value of "  
        << "firstNum = " << firstNum << endl;      //Line 15  
  
    return 0;                                    //Line 16  
}
```

//Line 17

6.1 Creating a C++ Program (Lanj)

Sample Run:

Line 9: firstNum = 18

Line 10: Enter an integer: 15

Line 13: secondNum = 15

Line 15: The new value of firstNum = 60

6.2 Program Style and Form

- Setiap program C++ program memiliki function main
- Kode yang ditulis harus benar secara sintaksis.

6.3 Syntax

- Ketika kompilasi dilakukan, maka kode yang error dapat terdeteksi

```
int x;          //Line 1
int y           //Line 2: error
double z;       //Line 3
```

```
y = w + x; //Line 4: error
```

6.4 Use of Semicolons, Brackets, and Commas

- Semua C++ statements diakhiri dengan titik koma (;)
 - Ini dikenal juga dengan statement terminator
- { dan } bukanlah C++ statements
- Koma digunakan untuk memisahkan item di dalam list.

6.5 Semantics

- Meskipun secara sintaksis sudah benar, namun program belum dapat mengeluarkan hasil yang diinginkan.
- Contohnya:

$2 + 3 * 5$ dan $(2 + 3) * 5$

keduanya secara sintaksis benar, namun memiliki makna yang berbeda.

6.6 Naming Identifiers

- Identifiers dapat berupa self-documenting:
 - CENTIMETERS_PER_INCH
- Hindari penulisan yang sekaligus seperti:
annualsale
- Solusi:
 - Huruf besar untuk setiap kata
 - annualSale
 - Tambahkan underscore (_) untuk memisahkan kata
 - annual_sale

6.7 Prompt Lines

- Prompt lines: executable statements yang menginformasikan user apa yang harus dilakukan.

```
cout << "Please enter a number between 1 and 10 and "
      << "press the return key" << endl;
cin >> num;
```

6.8 Documentation

- Program yang terdokumentasi dengan baik dapat dimengerti dengan mudah, dan mudah dimodifikasi.
- Kita dapat menggunakan comments untuk menulis dokumentasi program
- Comment dapat berupa:
 - Deskripsi program
 - Identitas programmer
 - Menjelaskan statement tertentu secara detail

6.9 Form and Style

- Ada 2 cara untuk mendeklarasikan variable:
 - Cara 1

```
int feet, inch;  
double x, y;
```
 - Cara 2

```
int a,b;double x,y;
```
- Keduanya benar, tetapi cara kedua lebih sulit dibaca.

6.9 Form and Style (Lanj..)

//An improperly formatted C++ program.

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    int num; double height;
    string name;
    cout << "Enter an integer: "; cin >> num; cout << endl;
    cout<<"num: "<<num<<endl;
    cout<<"Enter the first name: "; cin>>name;
    cout<<endl; cout <<"Enter the height: ";
    cin>>height; cout<<endl;

    cout<<"Name: "<<name<<endl;cout<<"Height: "
    <<height; cout <<endl; return 0;
}
```



//A properly formatted C++ program.

```
#include <iostream>
#include <string>

using namespace std;

int main()
{
    int num;
    double height;
    string name;
    cout << "Enter an integer: ";
    cin >> num;
    cout << endl;

    cout << "num: " << num << endl;

    cout << "Enter the first name: ";
    cin >> name;
    cout << endl;

    cout << "Enter the height: ";
    cin >> height;
    cout << endl;

    cout << "Name: " << name << endl;
    cout << "Height: " << height << endl;
}

return 0;
```



6.10 Programming Example: Convert Length

- TUIiskWrite a program that takes as input a given length expressed in feet and inches
 - Convert and output the length in centimeters
- Input: length in feet and inches
- Output: equivalent length in centimeters
- Lengths are given in feet and inches
- Program computes the equivalent length in centimeters
- One inch is equal to 2 . 54 centimeters

6.10 Programming Example: Convert Length (Lanj..)

- Convert the length in feet and inches to all inches:
 - Multiply the number of feet by 12
 - Add given inches
- Use the conversion formula (1 inch = 2.54 centimeters) to find the equivalent length in centimeters

6.10 Programming Example: Convert Length (Lanj..)

- The algorithm is as follows:
 - Get the length in feet and inches
 - Convert the length into total inches
 - Convert total inches into centimeters
 - Output centimeters

6.10 Programming Example:

6.10.1 Variables and Constants

- **Variables**

```
int feet;          //variable to hold given feet
int inches;        //variable to hold given inches
int totalInches;   //variable to hold total inches
double centimeters; //variable to hold length in
                     //centimeters
```

- **Named Constant**

```
const double CENTIMETERS_PER_INCH = 2.54;
const int INCHES_PER_FOOT = 12;
```

6.10 Programming Example:

6.10.2 Main Algorithm

- Prompt user for input
- Get data
- Echo the input (output the input)
- Find length in inches
- Output length in inches
- Convert length to centimeters
- Output length in centimeters

6.10 Programming Example:

6.10.3 Putting It Together

- Program begins with comments
- System resources will be used for I/O
- Use input statements to get data and output statements to print results
- Data comes from keyboard and the output will display on the screen
- The first statement of the program, after comments, is preprocessor directive to include header file iostream

6.10 Programming Example:

6.10.3 Putting It Together

- Two types of memory locations for data manipulation:
 - Named constants
 - Usually put before main
 - Variables
- This program has only one function (`main`), which will contain all the code
- The program needs variables to manipulate data, which are declared in `main`

6.10 Programming Example:

6.10.4 Body of the Function

- The body of the function `main` has the following form:

```
int main ()
```

```
{
```

 declare variables

 statements

```
    return 0;
```

```
}
```

6.10 Programming Example:

6.10.5 Writing a Complete Program

- Begin the program with comments for documentation
- Include header files
- Declare named constants, if any
- Write the definition of the function main

```
using namespace std;

    //Named constants
const double CENTIMETERS_PER_INCH = 2.54;
const int INCHES_PER FOOT = 12;

int main ()
{
    //Declare variables
    int feet, inches;
    int totalInches;
    double centimeter;

    //Statements: Step 1 - Step 7
    cout << "Enter two integers, one for feet and "
        << "one for inches: ";                                //Step 1
    cin >> feet >> inches;                               //Step 2
    cout << endl;
    cout << "The numbers you entered are " << feet
        << " for feet and " << inches
        << " for inches. " << endl;                          //Step 3

    totalInches = INCHES_PER FOOT * feet + inches;        //Step 4

    cout << "The total number of inches = "
        << totalInches << endl;                            //Step 5

    centimeter = CENTIMETERS_PER_INCH * totalInches;     //Step 6

    cout << "The number of centimeters = "
        << centimeter << endl;                           //Step 7

    return 0;
}
```

6.10 Programming Example:

6.10.6 Sample Run

Enter two integers, one for feet, one for inches: **15 7**

The numbers you entered are 15 for feet and 7 for inches.

The total number of inches = 187

The number of centimeters = 474.98

Ringkasan

- Identifier terdiri dari huruf, angka, dan underscores, dan bisa diawali dengan huruf atau underscore
- Operator aritmatika dalam C++ dapat berupa addition (+), subtraction (-), multiplication (*), division (/), and modulus (%)
- Gunakan `cin` dan stream extraction operator `>>` untuk input dari user
- Gunakan `cout` dan stream insertion operator `<<` untuk output ke screen
- Biasanya ekstensi untuk program C++ adalah `.cpp`

Latihan/Review

1. Buatlah sebuah program sapaan dalam notasi algoritma dengan spesifikasi sebagai berikut:
 - Menampilkan tulisan “Halo, siapa namamu?” di layer, lalu
 - Meminta pengguna memasukkan namanya, lalu
 - Menampilkan tulisan “Di kota apa kamu sekarang? Di layer, lalu
 - Meminta pengguna memasukkan nama kotanya sekarang, dan akhirnya `n <kota>` adalah string yang dibaca berdasarkan input user.

Latihan/Review (Lanj..)

2. Tulislah program input-output sederhana untuk menghitung luas bangun geometri (misalnya: bujursangkar, segitiga, trapesium, dsb). Data masukan dibaca dari piranti masukan dan luas bangun ditampilkan sebagai keluaran.
3. Di sebuah toko, pembeli mendapat diskon harga 5% untuk setiap barang yang dibeli. Tulislah algoritma yang membaca harga sebuah barang (dalam Rp) dan potongan diskon (dalam %), lalu hitunglah harga barang setelah didiskon.

Latihan/Review (Lanj..)

4. Buatlah program sederhana dalam notasi algoritma yang membaca temperatur dalam satuan Celcius, lalu mengkonversinya ke satuan Fahrenheit, dan menampilkan hasil konversinya ke layar. Rumus dari Celcius (C) ke Fahrenheit (F) adalah $F = \frac{9}{5} \times C + 32$.
5. Ulangi soal nomor 4 untuk konversi temperature dari Fahrenheit ke Celcius, dengan rumus $C = \frac{5}{9} \times (F - 32)$



Terima Kasih

UNIVERSITAS BUNDA MULIA