

**MODUL BAHASA PEMROGRAMAN**

**CIO 220**

**MODUL SESI 1**  
**PENDAHULUAN**

**DISUSUN OLEH**  
**KUNDANG K JUMAN**

**UNIVERSITAS ESA UNGGUL**  
**2020**  
**KONSEP DASAR BAHASA PEMOGRAMAN**

## **MODUL SESI 1**

### **Pokok Pembahasan**

- **Konsep Dasar Pemrograman**
- **Tipe – tipe bahasa pemrograman**
- **Sejarah C++**
- **Evolusi bahasa C++**
- **Pengenalan GCC**
- **Perintah – perintah Dasar GCC**
- **Kegunaan Bahasa C++**
- **Konsep dasar bahasa C++**
- **Struktur bahasa C++**
- **Kelebihan dan Kekurangan Bahasa C**
- **Contoh program C++**

### **Konsep Dasar Pemrograman**

- **Dasar pemahaman suatu pemrograman adalah analisa suatu algoritma**
- **Setiap algoritma yang disusun tidak tergantung pada bahasa pemrograman tertentu**
- **Setiap bahasa pemrograman mempunyai karakteristik yang berbeda baik struktur, sintak, maupun kode program**

### **Sejarah Bahasa Pemrograman**

Bahasa pemrograman, atau biasah disebut dengan bahasa komputer / bahasa pemrograman komputer, merupakan instruksi standar untuk memerintah komputer. Bahasa pemrograman ini ialah sebuah himpunan dari aturan sintaks dan semantik yang digunakan untuk mendefinisikan program komputer. Bahasa ini memungkinkan seorang programmer mampu menentukan secara persis data mana yang akan diolah oleh komputer, bagaimana data tersebut akan diteruskan/disimpan, dan jenis langkah apa yang akan dijalankan dalam berbagai macam keadaan (situasi).

Terdapat banyak macam bahasa komputer yang sudah diciptakan dan sudah banyak pula yang mempunyai perubahan sejak komputer pertama diciptakan. Awalnya programmer bekerja dengan intruksi komputer yang sangat primitif yaitu bernama Machine Language atau dalam bahasa

Indonesia adalah Bahasa Mesin. Beberapa tahun kemudian bahasa komputer pun berevolusi lagi. Lahirlah bahasa tingkat tinggi seperti BASIC dan COBOL. Bahasa ini memungkinkan programmer bekerja dengan sejumlah kata atau kalimat tertentu, seperti `Let I=100`. Instruksi ini akan diterjemahkan kembali ke bahasa mesin dengan menggunakan suatu compiler ataupun interpreter.

Sebuah interpreter bekerja dengan menterjemahkan baris program saat dibaca, mengubahnya menjadi bahasa mesin, dan mengeksekusinya (menjalankannya). Sedangkan Compiler akan menghasilkan suatu program yang sangat cepat pada saat dijalankan karena waktu untuk menterjemahkan program telah dihilangkan. Keuntungan dari Compiler adalah menghasilkan program executable yang tidak tergantung kepada program sumber. Sedangkan pada interpreter kita harus menjalankannya dengan menggunakan program sumber.

Pada beberapa tahun, prinsip utama dari programmer komputer adalah menulis sebuah kode (program) yang sependek mungkin, dan dapat dijalankan (dieksekusi komputer) dengan cepat. Serta program yang dibuat harus berukuran kecil karena memori komputer saat itu sangat mahal, dan harus cepat sebab tenaga CPU (pemrosesan) saat itu juga sangat mahal. Beberapa tahun kemudian setelah komputer menjadi semakin kecil, cepat, dan harga dari komputer serta komponennya semakin murah. Prioritas membuat program berukuran kecil dan harus cepat dieksekusi pun berubah. Sekarang biaya tenaga programmer jauh lebih mahal dibandingkan dengan harga komputer yang digunakan oleh programmer. Program yang ditulis dengan baik dan mudah ditangani adalah suatu yang bernilai tinggi. Mudah ditangani artinya adalah pada saat kebutuhan bisnis berubah, program dengan mudah dapat beradaptasi dan dikembangkan tanpa mengeluarkan biaya yang sangat besar.



MODUL 1 PENGENALAN BAHASA C++ ,ALGORITMA PEMROGRAMAN,INTEGRATED DEVELOPMENT EQUIPMENT (IDE) VISUAL C++ dan DASAR-DASAR BAHASA C++ A.

Tujuan Setelah mempelajari bab ini diharapkan mahasiswa akan mampu : ♣ Memahami sejarah perkembangan bahasa C++. ♣ Mengenal dan memahami algoritma program dengan flowchart. ♣ Mengenal dan menggunakan IDE C++ dengan baik. ♣ Mengenal dan memahami penggunaan Identifier, tipe data, variable, assignment, komentar, dan konstanta. ♣ Membuat program sederhana menggunakan bahasa C++.

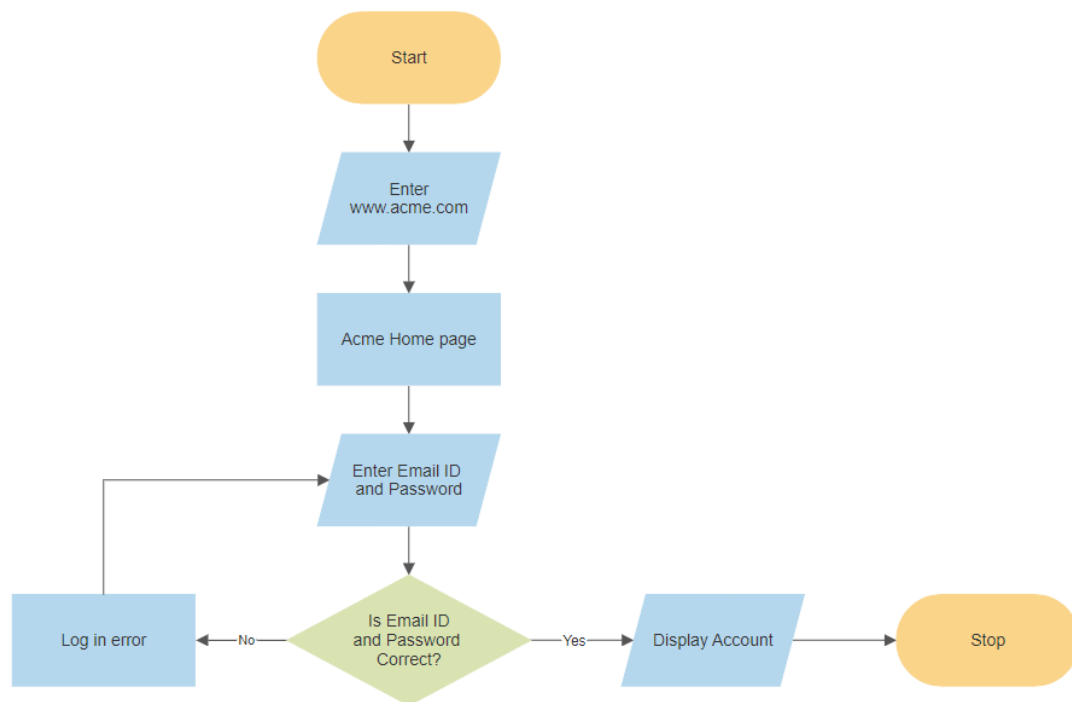
B. Dasar Teori Sejarah Perkembangan Bahasa C++ C++ adalah bahasa pemrograman komputer yang di buat oleh Bjarne Stroustrup, yang merupakan perkembangan dari bahasa C dikembangkan di Bong Labs (Dennis Ritchie) pada awal tahun 1970-an, Bahasa itu diturunkan dari bahasa

sebelumnya, yaitu B, Pada awalnya, bahasa tersebut dirancang sebagai bahasa pemrograman yang dijalankan pada sistem Unix, Pada perkembangannya, versi ANSI (American National Standart Institute). Bahasa pemrograman C menjadi versi dominan, Meskipun versi tersebut sekarang jarang dipakai dalam pengembangan sistem dan jaringan maupun untuk sistem embedded, Bjarne Stroustrup pada Bel labs pertama kali mengembangkan C++ pada awal 1980-an. Untuk mendukung fitur-fitur pada C++, dibangun efisiensi dan sistem support untuk pemrograman tingkat rendah (low level coding). Pada C++ ditambahkan konsep-konsep baru seperti class dengan sifat-sifatnya seperti inheritance dan overloading. Salah satu perbedaan yang paling mendasar dengan bahasa C adalah dukungan terhadap konsep pemrograman berorientasi objek (Object Oriented Programming)

**Algoritma Pemrograman** Algoritma adalah metode dan tahapan sistematis yang digunakan untuk memecahkan suatu permasalahan. Sedangkan pemrograman adalah proses pembuatan program dengan menerapkan algoritma dan struktur data tertentu menggunakan bahasa pemrograman. Struktur data adalah tempat tatanan penyimpanan data yang dibutuhkan program pada komputer. Gambar 1.1 Bagan Struktur Program Kode program yang telah ditulis kemudian akan di-compile dan dieksekusi untuk menjalankannya. Compile adalah menerjemahkan kode program yang ditulis menggunakan bahasa pemrograman, ke bahasa mesin yang dapat dimengerti oleh komputer sehingga komputer mengerti apa yang diperintahkan. Gambar 1.2 Mekanisme Eksekusi Program

**Flowchart** Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah, secara berurutan dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut

Flowchart Flowchart atau diagram alir merupakan sebuah diagram dengan simbol-simbol grafis yang menyatakan aliran algoritma atau proses yang menampilkan langkah-langkah, secara berurutan dengan menghubungkan masing masing langkah tersebut menggunakan tanda panah. Diagram ini bisa memberi solusi selangkah demi selangkah untuk penyelesaian masalah yang ada di dalam proses atau algoritma tersebut.



Fungsi bahasa pemrograman yaitu memerintah komputer untuk mengolah data sesuai dengan alur berpikir yang kita inginkan. Keluaran dari bahasa pemrograman tersebut berupa program/aplikasi. Contohnya adalah program yang digunakan oleh kasir di mal-mal atau swalayan, penggunaan lampu lalu lintas di jalan raya, dll.

Bahasa Pemrograman yang kita kenal ada banyak sekali di belahan dunia, tentang ilmu komputer dan teknologi dewasa ini. Perkembangannya mengikuti tingginya inovasi yang dilakukan dalam dunia teknologi. Contoh bahasa pemrograman yang kita kenal antara lain adalah untuk membuat aplikasi game, antivirus, web, dan teknologi lainnya.

Bahasa pemrograman komputer yang kita kenal antara lain adalah Java, Visual Basic, C++, C, Cobol, PHP, .Net, dan ratusan bahasa lainnya. Namun tentu saja kebutuhan bahasa ini harus disesuaikan dengan fungsi dan perangkat yang menggunakannya.

Secara umum bahasa pemrograman terbagi menjadi 4 kelompok, yaitu:

- Object Oriented Language (Visual dBase, Visual FoxPro, Delphi, Visual C)
- High Level Language (seperti Pascal dan Basic)
- Middle Level Language (seperti bahasa C), dan
- Low Level Language (seperti bahasa Assembly)

C++ diciptakan oleh Bjarne Stroustrup di laboratorium Bell pada awal tahun 1980-an, sebagai pengembangan dari bahasa C dan Simula. Saat ini, C++ merupakan salah satu bahasa yang paling populer untuk pengembangan software berbasis OOP.

Kompiler untuk C++ telah banyak beredar di pasaran. Software developer yang paling diminati adalah Borland Inc. dan Microsoft Corp. Produk dari Borland untuk kompiler C++ adalah Turbo C++, Borland C++, Borland C++ Builder. Sedangkan dari Microsoft adalah Ms. Visual C++.

## 1. STRUKTUR BAHASA C++

Cara terbaik untuk belajar bahasa pemrograman adalah dengan langsung mempraktikannya. Cobalah contoh program berikut :

```
// program  
pertamaku #include  
<iostream.h> int main  
    (  
  
        {  
    cout << "Selamat Belajar C++";  
    return 0;  
    }
```

Program di atas, misalnya dapat disimpan dengan nama *latih1.cpp*. Cara untuk menyimpan dan mengkompile program berbeda-beda, tergantung kompiler yang dipakai.

Ketika di-*run*, maka di layar akan muncul sebuah tulisan “Selamat Belajar C++”. Contoh di atas, adalah sebuah contoh program sederhana menggunakan C++. Namun, penggalan program tersebut telah menyertakan sintak-sintak dasar bahasa C++. Sintak dasar tersebut, akan kita bahas satu per satu:

merupakan sebuah baris komentar. Semua baris, yang ditandai dengan dua buah tanda

slash (/), akan dianggap sebagai baris komentar dan tidak akan berpengaruh pada hasil.

Biasanya, baris komentar dipakai oleh programmer untuk memberikan penjelasan tentang program.

Baris komentar dalam C++, selain ditandai dengan (//) juga dapat ditandai dengan (/\*....\*/).

Perbedaan mendasar dari keduanya adalah :

```
// baris komentar  
/* blok komentar */
```

### **#include <iostream.h>**

pernyataan yang diawali dengan tanda (#) merupakan pernyataan untuk menyertakan preprocessor. Pernyataan ini bukan untuk dieksekusi. **#include <iostream.h>** berarti memerintahkan kompiler untuk menyertakan file header **iostream.h**. Dalam file header ini, terdapat beberapa fungsi standar yang dipakai dalam proses input dan output. Seperti misalnya perintah *cout* yang dipakai dalam program utama.

### **int main ()**

Baris ini menandai dimulainya kompiler akan mengeksekusi program. Atau dengan kata lain, pernyataan **main** sebagai penanda program utama. Adalah suatu keharusan, dimana sebuah program yang ditulis dalam bahasa C++ memiliki sebuah **main**.

**main** diikuti oleh sebuah tanda kurung () karena main merupakan sebuah fungsi. Dalam bahasa C++ sebuah fungsi harus diikuti dengan tanda (), yang nantinya dapat berisi argumen. Dan sintak formalnya, sebuah fungsi dimulai dengan tanda {}, seperti dalam contoh program.

### **cout << "Selamat Belajar C++";**

perintah ini merupakan hal yang akan dieksekusi oleh compiler dan merupakan perintah yang akan dikerjakan. **cout** termasuk dalam file **iostream**. **cout** merupakan perintah untuk menampilkan ke layar.



**Perlu diingat, bahwa setiap pernyataan dalam C++ harus diakhiri dengan tanda semicolon (;) untuk memisahkan antara pernyataan satu dengan pernyataan lainnya.**

**return 0;**

pernyataan return akan menyebabkan fungsi main() menghentikan program dan mengembalikan nilai kepada main. Dalam hal ini, yang dikembalikan adalah nilai 0. Mengenai pengembalian nilai, akan dijelaskan nanti mengenai **Fungsi** dalam C++.

Coba tambahkan sebaris pernyataan lagi, sehingga program contoh di atas akan menjadi seperti berikut:

```
// latihan kedua
#include <iostream.h>
int main ()
{
    cout << "Selamat Belajar C++";
    cout << "di kampusku";
    return 0;
}
```

Maka perintah cout yang kedua akan menampilkan sebuah kalimat lagi di layar, dengan tulisan “di kampusku”.

## **2. VARIABEL**

Berbeda dengan pendeklarasian variabel di bahasa pemrograman lain, dalam C++ sebelum mendeklarasikan variabel, hal pertama yang harus dideklarasikan adalah tipe data yang akan digunakan untuk menampung data.

Format penulisannya adalah :

**Tipe\_data pengenalan = nilai ;**

Sebagai contoh :

**int a;**

**float nomor;**

atau dapat juga pemberian nilai awal untuk variable dilakukan pada saat deklarasi,

contoh :

**int a=10; char s='a';**

Jika hendak mendeklarasikan beberapa variabel sekaligus dengan tipe data yang sama, dapat dilakukan dengan 2 cara, yaitu :

**int a; int b; int c;**

atau dapat disederhanakan dengan deklarasi :

**int a,b,c;**

## 1. KONSTANTA

Konstanta mirip dengan variable, namun memiliki nilai tetap. Konstanta dapat berupa nilai Integer, Float, Karakter dan String.

Pendeklarasian konstanta dapat dilakukan dengan 2 cara :

- ❖ menggunakan (**#define**)  
deklarasi konstanta dengan cara ini, lebih gampang dilakukan karena akan menyertakan **#define** sebagai preprocessor directive.

## 3. VARIABEL

Berbeda dengan pendeklarasian variabel di bahasa pemrograman lain, dalam C++ sebelum mendeklarasikan variabel, hal pertama yang harus dideklarasikan adalah tipe data yang akan digunakan untuk menampung data.

Format penulisannya adalah :

**Tipe\_data pengenalan = nilai ;**

Sebagai contoh :

**int a;**

**float nomor;**

atau dapat juga pemberian nilai awal untuk variable dilakukan pada saat deklarasi,

contoh :

**int a=10; char s='a';**

Jika hendak mendeklarasikan beberapa variabel sekaligus dengan tipe data yang sama, dapat dilakukan dengan 2 cara, yaitu :

**int a; int b; int c;**

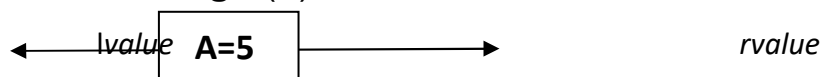
atau dapat disederhanakan dengan deklarasi :

**int a,b,c;**

#### 4. OPERATOR

Dalam C++, terdapat berbagai macam operator yang dapat dimanfaatkan dalam aplikasi.

##### ❖ Operator Assign (=)



artinya memberikan nilai 5 ke dalam variable a. Sebelah kiri tanda = dalam pernyataan di atas, dikenal dengan *lvalue* (left value) dan di sebelah kanan tanda = dikenal dengan *rvalue* (right value). lvalue harus selalu berupa variable, sedangkan rvalue dapat berupa variable, nilai, konstanta, hasil operasi ataupun kombinasinya.

##### ❖ Operator Aritmatika (+, -, \*, /, %)

**Operator (=), akan memberikan nilai ke dalam suatu variabel**

❖ **Operator Aritmatika ( +, -, \*, /, %)**

Operator	Keterangan
+	Penjumlahan
-	Pengurangan
*	Perkalian
/	Pembagian
%	Modulus

Untuk operator %, sama dengan modulus, yaitu untuk mengetahui sisa hasil bagi. Misalnya  $a = 11 \% 3$ , maka variable a akan terisi nilai 2 karena sisa hasil bagi 11 dan 3 adalah 2.

❖ **Operator Majemuk ( +=, -=, \*=, /=, %=, <<=, >>=, &=, |= )** Dalam C++, operasi aritmatika dapat disederhanakan penulisannya dengan format penulisan operator majemuk.

**Misalnya :**

$a += 5$  sama artinya dengan menuliskan  $a = a + 5$   $a *= 5$  sama artinya dengan menuliskan  $a = a * 5$   $a /= 5$  sama artinya dengan menuliskan  $a = a / 5$

$a \% = 5$  sama artinya dengan menuliskan  $a = a \% 5$

*lvalue a = 5 rvalue*

❖ **Operator Penaikan dan Penurunan ( ++ dan -- )**

Operator penaikan (++) akan menaikkan atau menambahkan 1 nilai variable. Sedangkan operator (--) akan menurunkan atau mengurangi 1 nilai variable.

Misalnya :

**$a++$ ;  $a+=1$ ;**

**$a=a+1$ ;**

untuk ketiga pernyataan tersebut, memiliki arti yang sama yaitu menaikkan nilai variable 1. Karakteristik dari operator ini adalah dapat dipakai di awal (++a) atau diakhir (--a) variable. Untuk penggunaan biasa, mungkin tidak akan ditemui perbedaan hasil dari cara penulisannya. Namun untuk beberapa operasi nantinya harus diperhatikan cara peletakan operator ini, karena akan berpengaruh terhadap hasil.

Contoh 1 :

**B=3;**

**A=++B;**

**// A= 4, B=4**

Contoh 2:

**B=3;**

**A=B++;**

**//hasil A=3, B=4**

Dari contoh1, nilai B dinaikkan sebelum dikopi ke variable

A. Sedangkan pada contoh2, nilai B dikopi terlebih dahulu ke variable A baru kemudian dinaikkan.

#### ❖ **Operator Relasional** (==, !=, >, <, >=, <=)

Yang dihasilkan dari operator ini bukan berupa sebuah nilai, namun berupa bilangan bool yaitu benar atau salah.

Operator	Keterangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar dari
<	Kurang dari
>=	Lebih besar dari atau sama dengan
<=	Kurang dari atau sama dengan

Contoh :

**(7==5) hasilnya adalah false (5>4)**

**hasilnya adalah true (5<5)**

**hasilnya adalah false**

#### ❖ **Operator Logika** ( !, &&, || )

Operator logika juga digunakan untuk memberikan nilai atau kondisi **true** dan **false**.

Biasanya operator logika dipakai untuk membandingkan dua kondisi.

Misalnya:

**((5==5) && (3>6))** mengembalikan nilai **false**, karena (true && false)

untuk logika NOT (!), contohnya **!(5==5)** akan mengembalikan nilai **false**, karena !(true).

## ❖ **Operator Kondisional ( ? )**

Format penulisan operator kondisional adalah :

**kondisi ? hasil1 : hasil2**

Jika kondisi benar maka yang dijalankan adalah hasil1 dan jika kondisi salah, maka akan dijalankan hasil2.

Contoh :

**7==5 ? 4 : 3** hasilnya adalah **3**, karena 7 tidak sama

dengan 5

**5>3 ? a : b** hasilnya adalah a, karena 5 lebih

## **Kegunaan Bahasa C++**

- Bahasa C++ dapat digunakan untuk membuat program berupa aplikasi komputer berbasis teks maupun window
- Bahasa C++ juga biasanya digunakan untuk melakukan interfacing ke perangkat keras, karena bahasa C++ lebih dekat dengan bahasa mesin Bahasa C++ dapat digunakan untuk membuat program berupa aplikasi komputer berbasis teks maupun window
- Bahasa C++ juga biasanya digunakan untuk melakukan interfacing ke perangkat keras, karena bahasa C++ lebih dekat dengan bahasa mesin
- Bahasa C++ dapat digunakan untuk membuat program berupa aplikasi komputer berbasis teks maupun window

- Bahasa C++ juga biasanya digunakan untuk melakukan interfacing ke perangkat keras, karena bahasa C++ lebih dekat dengan bahasa mesin Bahasa C++ dapat digunakan untuk membuat program berupa aplikasi komputer berbasis teks maupun window
- Bahasa C++ juga biasanya digunakan untuk melakukan interfacing ke perangkat keras, karena bahasa C++ lebih dekat dengan bahasa mesin

### **Konsep Dasar Bahasa C++**

- Fungsi main()

Harus ada pada program, sebab fungsi inilah yang menjadi titik awal dan akhir eksekusi program. Tanda { diawal fungsi menyatakan awal tubuh fungsi dan juga sebagai awal program dijalankan, sedangkan tanda } di akhir fungsi merupakan akhir bagian isi fungsi dan sekaligus adalah akhir dari eksekusi program.

- Fungsi printf()

Merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga

Dalam bentuk umum :

printf ("kontrol string", daftar pernyataan);

### **Kelebihan Bahasa C**

1. C mempunyai banyak jenis data yang dikenal & operator untuk manipulasi data
2. C menyediakan berbagai struktur data dn pengendalian proses.
3. C mudah dipahami dibandingkan dengan bahasa mesin
4. C mempunyai kecepatan eksekusi yang mendekati kecepatan eksekusi bahasa mesin
5. C memungkinkan manipulasi data dalam bentuk bit maupun byte secara efisien dan dapat memanipulasi alamat dari suatu data
6. C memakai sedikit memori
7. C merupakan salah satu bahasa pemrograman yang terstruktur

