

BUKU AJAR DASAR PEMROGRAMAN C++



**ALFIAN MA'ARIF
PROGRAM STUDI TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS AHMAD DAHLAN
2020**

BUKU AJAR DAN PRAKTIK

DASAR PEMROGRAMAN C++

Penulis: Alfian Ma'arif

ISBN:

Desain sampul:

Proofreader:

Penerbit : UAD Press

Kantor : Lembaga Penerbitan dan Publikasi Ilmiah

Universitas Ahmad Dahlan

Kampus 4, Jln. Ringroad Selatan Yogyakarta

Telp. 0274-379418 Ext. 4902

Website: lppi.uad.ac.id

Cetakan pertama, bulan tahun cetak

Hak Cipta © pada Penulis

Hak Cipta dilindungi undang-undang

Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apa pun, baik secara elektronis maupun mekanis, termasuk memfotokopi, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis.

PRAKATA

Pertama penulis mengucapkan syukur Alhamdulillah kepada Allah SWT sehingga buku ini dapat terselesaikan dengan baik. Penulis mengucapkan terima kasih kepada berbagai pihak yang telah membantu dalam penyelesaian buku ini. Secara khusus penulis ingin menyampaikan ucapan terima kasih kepada kedua orang tua, Bapak Muhtadi dan Ibu Tri Puji Astuti. Lalu penulis juga mengucapkan terima kasih kepada Pak Hatib Rahmawan selaku kepala UAD Press, Bapak Nuryono Satya Widodo selaku kaprodi teknik elektro dan seluruh staf dosen teknik elektro UAD.

Buku berjudul Dasar Pemrograman C++, dikhususkan bagi mahasiswa teknik elektro sebagai pendamping mata kuliah Dasar Komputer dan Pemrograman, Pemrograman Lanjut dan Praktikum Dasar Pemrograman.

Buku ini merupakan buku karya kedua dari penulis sehingga masih banyak kesalahan dalam penulisan. Oleh karena itu penulis memohon maaf dan mohon pembaca dapat memberikan masukan dengan cara mengirimkan email kepada penulis ke alamat alfianmaarif@ee.uad.ac.id atau alfianmaarif@gmail.com

Demikian yang dapat penulis sampaikan. Semoga buku ini dapat bermanfaat bagi para pembaca. Terima kasih.

Yogyakarta, 17 Februari 2022

Penulis

DAFTAR ISI

PRAKATA	3
DAFTAR ISI	4
BAB 1 PENDAHULUAN BAHASA C++	9
Belajar C++.....	9
Pengenalan bahasa C++.....	9
Apa itu C++?	9
Mengapa menggunakan Bahasa C++?	9
Memulai Bahasa C++	10
C++ IDE	10
Langkah Pertama Program C++	11
Aturan Penamaan File C++	12
Sintak C++.....	13
Mempelajari Namespace	14
Keluaran C++ (Menampilkan Tulisan/Teks).....	14
Baris Baru C++	15
Komentar C++	16
Komentar baris tunggal (//).....	17
Komentar Multi Baris (/...*/)	17
BAB 2 VARIABEL, INPUT & TIPE DATA	25
Variabel C++	25
Mendeklarasikan variabel C++.....	25
Menampilkan Variabel	27

Menjumlahkan Variabel	27
Mendefinisikan Banyak Variabel	28
Variabel Pengenal C++	28
Konstanta C++	29
Masukan C++	30
Membuat Kalkulator Sederhana	31
Tipe Data	33
Tipe Data Angka.....	34
float dengan double.....	36
Angka Ilmiah	36
Boolean	36
Karakter	37
String.....	38
Aturan Penamaan Variabel	39
BAB 3 OPERATOR	40
Operator Aritmetika.....	40
Operator Penugasan	44
Operator Perbandingan	49
Operator Logika.....	53
BAB 4 STRING & MATH.....	55
Penggabungan String	55
Menambahkan String.....	56
Menambahkan Angka dan String	56
Panjang String.....	57

Mengakses String.....	58
Mengubah Karakter String	58
String Input Pengguna	59
Menghilangkan Namespace.....	60
Math.....	61
Max & Min	61
Pustaka Math	62
Pustaka Math Lainnya	62
Nilai Absolut atau Mutlak	62
Logaritma.....	63
Sudut.....	63
Akar dan Pangkat.....	63
BAB 5 IF ELSE & SWITCH CASE	68
Kondisi dan Pernyataan if.....	68
Pernyataan if	68
Pernyataan else	72
Pernyataan else if.....	73
Singkatan If ... Else (Operator Ternary)	75
Pernyataan Switch	77
Kata Kunci break	79
Kata Kunci default.....	79
BAB 6 PERULANGAN WHILE, DO-WHILE, FOR	81
Perulangan While.....	81
Perulangan Do While.....	82

Perulangan For.....	83
BREAK.....	86
CONTINUE.....	87
Perintah Break dan Continue pada Perulangan While.....	87
Perulangan Turun.....	89
Diagram Alir Perulangan	91
BAB 7 ARRAY (KUMPULAN DATA).....	93
Akses Elemen Array	93
Ubah Elemen Array	94
Perulangan dengan Array	95
BAB 8 REFERENSI & POINTER.....	98
Membuat Referensi.....	98
Membuat Pointer	99
Dereferensi.....	101
Modifikasi Pointer	102
BAB 9 FUNGSI.....	103
Membuat Fungsi	103
Memanggil Fungsi	104
Deklarasi Fungsi dan Definisi	105
Parameter Fungsi	107
Parameter Fungsi Default	108
Beberapa Parameter Fungsi	109
Fungsi dengan Nilai Balik	110
Referensi pada Fungsi.....	111

Melebihi Beban Fungsi.....	112
BAB 10 PEMROGRAMAN BERORIENTASI OBJEK	115
Pemrograman Berorientasi Objek.....	115
Kelas (Class) dan Objek (Object)	115
Membuat Kelas (Class).....	116
Membuat Objek (Object).....	117
Membuat Beberapa Objek	118
Metode pada Kelas	120
Menambahkan Parameter Fungsi pada Kelas	121
Konstruktor	122
Parameter Konstruktor	123
Bekerja dengan Files.....	125
Membaca File	127
DAFTAR PUSTAKA	128
Tentang Penulis	129
Tentang Teknik Elektro UAD	129
Tentang UAD Press	130
RINGKASAN BUKU	131

BAB 1 PENDAHULUAN BAHASA C++

Belajar C++

Bahasa C++ adalah salah satu bahasa pemrograman yang populer.

Bahasa C++ digunakan untuk membuat program atau aplikasi komputer.

Pengenalan bahasa C++

Apa itu C++?

Bahasa C++ adalah bahasa yang dapat digunakan untuk membuat aplikasi berkinerja tinggi.

Bahasa C++ dikembangkan oleh Bjarne Stroustrup, sebagai pengembangan dari bahasa C.

Bahasa C++ memberikan programmer kemampuan tinggi dalam mengendalikan sistem dan memori.

Bahasa C++ telah diperbarui tiga kali yaitu pada tahun 2011, 2014 dan 2017 dengan nama C++11, C++14 dan C++17.

Mengapa menggunakan Bahasa C++?

C++ adalah salah satu dari bahasa pemrograman yang paling populer di dunia.

C++ dapat ditemukan pada sistem operasi saat ini, sistem antar muka dan sistem embedded.

C++ portabel dan dapat digunakan untuk pengembangan aplikasi yang dapat diadaptasi oleh semua platform.

C++ merupakan bahasa pemrograman yang menyenangkan dan mudah untuk dipelajari.

Bahasa C++ memiliki kedekatan dengan bahasa C# dan Java, hal tersebut membuat programmer mudah dalam mengonversi ke bahasa C++ atau sebaliknya.

Memulai Bahasa C++

Untuk memulai menggunakan C++, ada dua hal yang diperlukan yaitu

1. Sebuah teks editor (Editor) seperti notepad untuk menulis kode C++
2. Sebuah kompiler (Compiler) seperti GCC untuk menerjemahkan kode C++ ke bahasa yang dipahami oleh komputer (Bahasa Mesin)

Ada banyak teks editor dan kompiler salah satunya yaitu menggunakan Integrated Development Environment (IDE).

C++ IDE

IDE adalah kependekan dari Integrated Development Environment. Sebuah IDE digunakan untuk memodifikasi dan mengompilasi kode.

IDE yang populer di antaranya Code Blocks, Eclipse dan Visual Studio. IDE tersebut gratis dan dapat digunakan untuk mengedit dan men-debug kode C++.

Catatan: Selain IDE dalam bentuk offline, juga ada IDE dalam bentuk online atau Kompiler Online menggunakan website. Namun fungsinya terbatas.

Code blocks sangat baik untuk memulai belajar bahasa pemrograman C++. Versi terakhir codeblocks dapat ditemui pada website resminya. Lakukan instalasi code blocks sesuai spesifikasi prosesor komputer dan instalasi **mingw-setup.exe** yang akan meng-install teks editor beserta kompilernya.

Langkah Pertama Program C++

Untuk membuat program C++ yang pertama, klik pada **File > New > Empty File**.

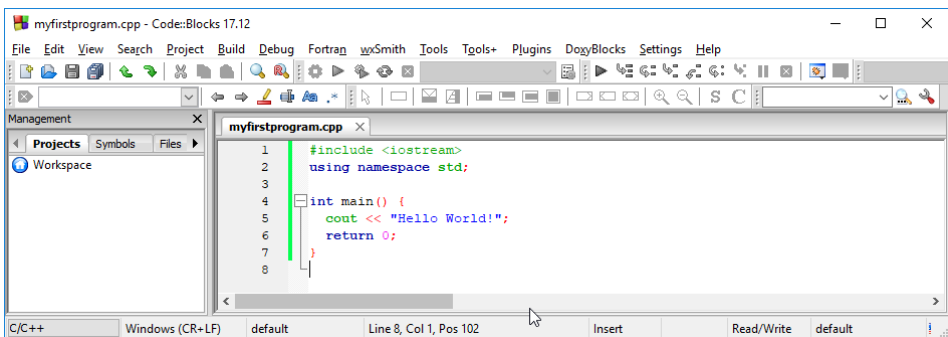
Tuliskan kode C++ dan simpan dengan nama program1.cpp (**File > Save File as**).

```
program1_1.cpp

#include <iostream>
using namespace std;

int main()
{
    cout << "Halo Dunia!";
    return 0;
}
```

Tampilan pada code blocks akan terlihat seperti Gambar 1.1.



Gambar 1.1.

Lalu klik menu **build > build and run** untuk menjalankan dan mengeksekusi program. Hasil akan terlihat seperti pada Gambar 1.2.

```
Hello World!  
Process returned 0 (0x0) execution time : 0.011 s  
Press any key to continue.
```

Gambar 1.2.

Selamat! Anda sudah berhasil menulis dan mengeksekusi program pertama C++.

Aturan Penamaan File C++

Dalam memberikan nama File C++ (dan file program yang lainnya) terdapat beberapa aturan yang harus diperhatikan. Jika aturan ini dilanggar maka akan terjadi kesalahan atau error program. Hal-Hal yang harus diperhatikan adalah

1. Tidak boleh ada spasi pada nama File C++.
2. Tidak boleh ada simbol khusus seperti %, \$, @, #, !, ?, >, <, koma (,), titik (.) dan sebagainya.
3. Spasi dapat digantikan dengan simbol garis bawah atau underscore (_).
4. Nama file tidak boleh dimulai dengan angka.
5. Setelah huruf, nama file boleh dikombinasikan dengan angka.
6. Sebaiknya nama file menggunakan huruf kecil semua.
7. Jika program merupakan file bahasa C++ maka diakhiri dengan .cpp
8. Jika program merupakan file bahasa C maka diakhiri dengan .c

Contoh penamaan File C++ yang benar

program_1_1.cpp

program1_1.cpp

programSatu.cpp

Contoh penamaan File C++ yang salah dan menyebabkan error

Program 1.1.cpp

1_1_program.cpp

Sintak C++

Perhatikan kode pada program1_1.cpp

program1_1.cpp
<pre>#include <iostream> using namespace std; int main() { cout << "Halo Dunia!"; return 0; }</pre>

Penjelasan program

Baris 1: `#include<iostream>` adalah library atau pustaka yang mengizinkan bekerja dengan perintah input dan output, seperti perintah `cout` pada baris lima.

Baris 2: `using namespace std` berarti dapat menggunakan nama untuk objek dan variabel dari pustaka standar.

Baris 3: Spasi kosong, C++ menghiraukan spasi kosong.

Baris 4: `int main()` adalah fungsi yang selalu muncul pada program C++. Fungsi tersebut dinamakan dengan program utama. Program dalam tanda `{ }` akan dieksekusi.

Baris 5: `cout` dapat dibaca dengan kata “see-out” adalah objek yang digunakan bersama dengan operator (`<<`) untuk mengeluarkan atau

mencetak tulisan (teks). Pada contoh tersebut akan mengeluarkan tulisan “Hello World”.

Catatan: Setiap baris program pada C++ diakhiri dengan titik koma (;).

Baris 6: return 0 berfungsi untuk mengakhiri fungsi utama.

Baris 7: Jangan lupa untuk menambahkan kurung kurawal penutup } untuk mengakhiri fungsi program utama.

Mempelajari Namespace

Pada program C++ terkadang ditemui program tanpa library namespace standar. Perintah using namespace std dapat diganti dengan kata std diikuti dengan operator :: seperti pada program berikut.

```
program1_2.cpp

#include <iostream>

int main() {
    std::cout << "Halo Dunia!";
    return 0;
}
```

Programmer bebas untuk menggunakan standar pustaka namespace atau tidak.

Keluaran C++ (Menampilkan Tulisan/Teks)

Perintah cout beserta dengan operator <<, digunakan untuk menampilkan tulisan teks atau nilai sebagai

```
program1_3.cpp

#include <iostream>
using namespace std;
int main() {
```

```
cout << "Halo Dunia!";  
return 0;  
}
```

Programmer dapat menambahkan objek cout sesuai yang diinginkan. Akan tetapi, catat bahwa perintah tersebut tidak menambah baris baru pada akhir dari keluaran:

program1_4.cpp

```
#include <iostream>  
using namespace std;  
int main() {  
    cout << "Halo Dunia!";  
    cout << "Saya Belajar C++";  
    return 0;  
}
```

Baris Baru C++

Untuk menambahkan baris baru, dapat digunakan karakter `\n`.

program1_5.cpp

```
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << "Halo Dunia! \n";  
    cout << "Saya Belajar C++";  
    return 0;  
}
```

Tips: Dua buah karakter `\n` akan membuat baris kosong.

program1_6.cpp

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

int main()
{
    cout << "Halo Dunia! \n\n";
    cout << "Saya Belajar C++";
    return 0;
}
```

Cara lain untuk menambahkan baris baru adalah dengan perintah manipulasi **endl** sebagai:

program1_7.cpp

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

int main()
{
    cout << "Halo Dunia!" << endl;
    cout << "Saya Belajar C++";
    return 0;
}
```

Komentar C++

Komentar dapat digunakan untuk menjelaskan kode C++ dan untuk membuat kode lebih mudah dipahami. Komentar juga dapat digunakan

untuk menghindari eksekusi kode program yang diuji coba. Komentar dapat berupa komentar baris tunggal maupun komentar multi baris.

Komentar baris tunggal (//).

Tulisan antara tanda // dan akhir baris akan dihiraukan oleh kompiler (tidak akan dieksekusi).

Contoh berikut adalah komentar baris tunggal sebelum baris program

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

int main(){
    // Ini adalah komentar
    cout << "Halo Dunia!";
}
```

Contoh komentar baris program di akhir program

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

int main(){
    cout << "Halo Dunia!"; // Ini adalah komentar
}
```

Komentar Multi Baris (/*...*/)

Komentar multi baris dimulai dengan tanda /* dan diakhiri dengan tanda */. Tulisan dan teks apapun antara tanda /* dan */ akan dihiraukan oleh kompiler.

```
#include <iostream>
using namespace std;
int main(){
/* Kode berikut akan menampilkan tulisan Halo Dunia!
pada layar cmd/ Terminal */
cout << "Halo Dunia!";
}
```

Algoritme

Sebelum mempelajari bahasa pemrograman lebih lanjut, hal yang perlu dipelajari terlebih dahulu adalah algoritme. Bahasa pemrograman hanya sebagai alat bantu dalam membuat suatu program. Ada banyak sekali jenis bahasa pemrograman selain bahasa C++. Intinya setiap bahasa pemrograman memiliki materi yang sama dan algoritma yang sama, hanya memang memiliki perintah yang berbeda-beda, namun tetap memiliki fungsi yang sama. Oleh karena itu penting untuk mempelajari tentang algoritma.

Jika seseorang programmer telah menguasai 1 bahasa pemrograman dan algoritma nya, maka akan lebih mudah untuk mempelajari bahasa pemrograman yang lain.

Algoritma adalah urutan atau langkah-langkah untuk perhitungan atau menyelesaikan masalah yang ditulis secara berurutan.

Algoritma pemrograman adalah urutan atau langkah-langkah untuk menyelesaikan masalah pemrograman komputer

Dalam pemrograman hal yang terpenting untuk dipahami adalah logika dalam berpikir bagaimana cara memecahkan dan menyelesaikan masalah pemrograman yang akan dibuat.

Algoritma dapat dituliskan dalam bentuk teks maupun gambar. Beberapa bentuk algoritma adalah tulisan bebas, pseudocode dan diagram alir.

Soal latihan membuat algoritma

Soal 1

Bagaimana cara mendapatkan air dengan volume 4 liter dari gelas dengan ukuran 2 liter, 3 liter dan 5 liter? Sebutkan 2 cara!

Jawab:

Ada dua cara

Masukkan 2 liter air pada gelas 5 liter sebanyak dua kali sehingga diperoleh air dengan volume 4 liter

Masukkan air ke dalam gelas 5 liter dan 2 liter sampai penuh. Tuang air di dalam gelas 5 liter ke gelas 3 liter sehingga menyisakan air 2 liter. Tuang air dalam gelas 2 liter ke gelas tersebut.

Soal 2

Bagaimana cara mendapatkan air dengan volume 4 liter dari gelas dengan ukuran 3 liter dan 5 liter?

Soal 3

Diketahui sebuah program sebagai berikut

```
x=1;
```

```
y=2;
```

```
x=y;
```

```
z=x;
```

Berapa nilai z?

Soal 4

Diketahui sebuah program sebagai berikut

```
x=1;
```

```
x++;
```

```
x=x+1;
```

```
x+=1
```

Berapa nilai x?

Soal 5

Diketahui sebuah program sebagai berikut

```
x=5;
```

```
x--;
```

```
x=x*2;
```

```
x-=1
```

Berapa nilai x?

Soal 6

Diketahui sebuah program sebagai berikut

```
x=1;
```

```
y=2;
```

```
z=x+y;
```

```
z=3;
```

```
x=y+z;
```

```
y=x+y+z;
```

Berapa nilai x, y dan z?

Pseudocode

Contoh pseudocode

```
program hitung_luas_persegi_panjang
```

```
Deklarasi:
```

```
var panjang, lebar, luas: integer;
```

```
Algoritma:
```

```
input(panjang);
```

```
input(lebar);
```

```
luas <-- panjang * lebar;
```

```
print(luas);
```

```
program hitung_luas_lingkaran
```

```
Deklarasi:
```

```
var phi: float;
```

```
var r, luas: integer;
```

```
algoritma:
```

```
phi <-- 3.14;
```

```
input(r); {diinput user}
```

```
luas <-- phi * r *r;
```

```
output(luas);
```

```
program_ganjil_genap
```

```
Input bilangan
```

```
If bilangan mod 2 = 0 then,
```

```
“Output Genap”
```

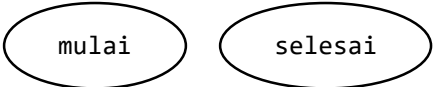

```
Else
```

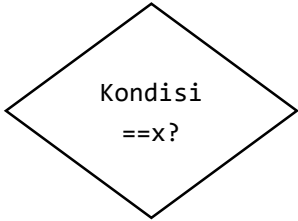

```
“Output Ganjil”
```

```
End if
```

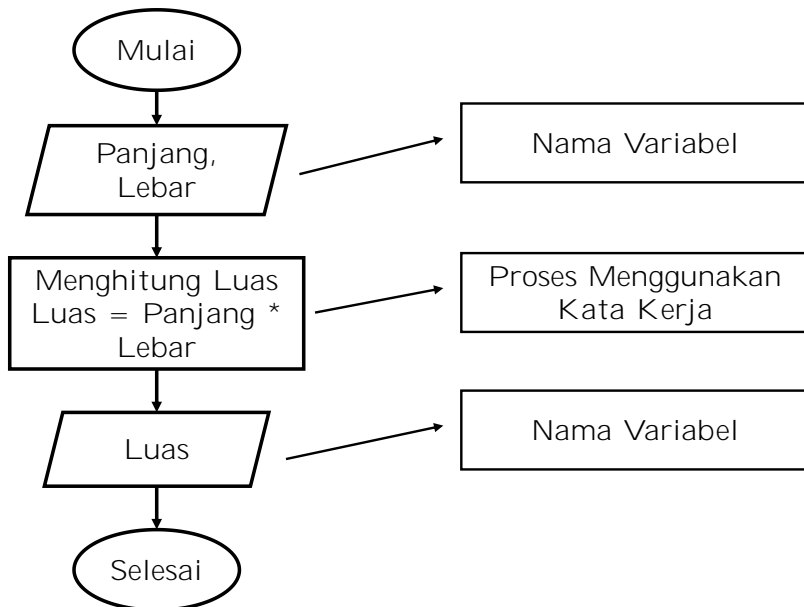
Diagram Alir

Bentuk blok diagram alir ditunjukkan pada tabel berikut

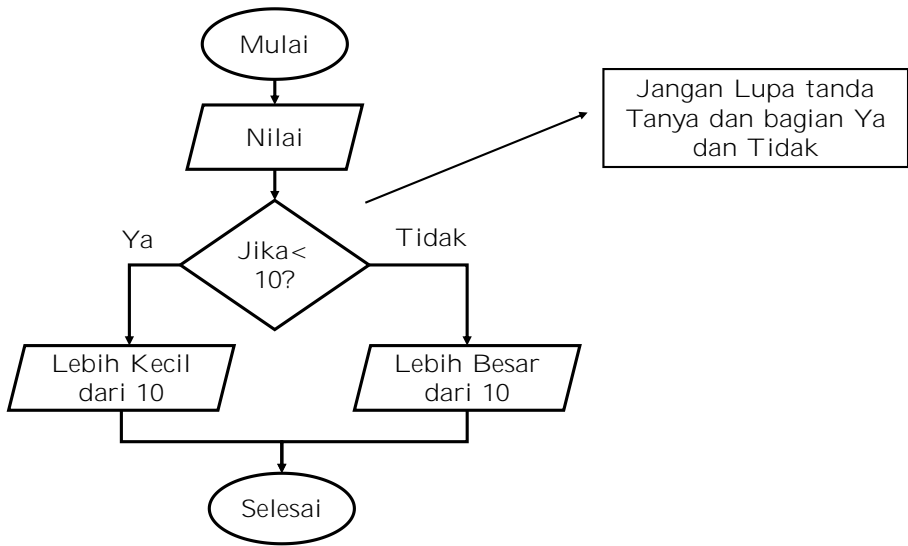
Bentuk Blok	Fungsi
 Oval	Berfungsi sebagai awal dan akhir program (mulai dan selesai)
 Jajar Genjang	Berfungsi sebagai masukan dan keluaran program

 <p>Kondisi ==x?</p> <p>Belah Ketupat</p>	<p>Berfungsi sebagai pernyataan kondisi. Pada belah ketupat terdapat bagian kondisi dengan tanda tanya dan panah Ya dan Tidak.</p>
 <p>Proses</p> <p>Persegi Panjang</p>	<p>Berfungsi sebagai perintah proses. Biasanya menggunakan kata kerja seperti menghitung, membaca, mengirim.</p>

Contoh Diagram Alir untuk menghitung dan menampilkan luas persegi panjang.



Contoh diagram alir untuk menentukan nilai kurang dan lebih dari 10.



Perbandingan antara diagram alir dan pseudocode.

Diagram alir

- Lebih susah dibuat, kebanyakan dibuat setelah program jadi
- Visualisasi bagus, mudah dipahami dan jelas
- Ada aturan baku
- Lebih sederhana

Pseudocode

- Mendekati bahasa pemrograman asli dan lebih mudah dibuat (hanya menggunakan teks)
- Kurang familier
- Terkadang susah dipahami
- Kompleks

BAB 2 VARIABEL, INPUT & TIPE DATA

Variabel C++

Variabel adalah sebuah kotak atau kontainer yang berfungsi untuk menyimpan nilai.

Dalam C++, ada berbagai jenis variabel (didefinisikan dengan kata kunci yang berbeda) sebagai contoh:

int – menyimpan bilangan integer (semua angka) tanpa desimal seperti 123 atau -123

double – menyimpan angka desimal seperti 19,999 atau -19,99

char – menyimpan karakter tunggal seperti 'a' atau 'B'. Nilai char dikurung dengan petik tunggal.

string – menyimpan teks atau tulisan seperti tulisan "Halo Dunia". Nilai string dikurung menggunakan tanda petik dubel.

bool – menyimpan dua kondisi yaitu benar dan salah.

Mendeklarasikan variabel C++

Untuk membuat variabel, programmer harus menuliskan jenis data dan mengisinya dengan suatu nilai seperti

Sintak
<code>tipe_data nama_variabel = nilai;</code>

Dengan variabel **tipe** adalah tipe data variabel C++ seperti int, double, char, string, bool dan **nama_variabel** adalah nama dari variabel seperti x atau namaSaya. Tanda sama dengan (=) digunakan untuk mengisi nilai pada suatu variabel.

Untuk membuat variabel yang dapat menyimpan suatu nilai, perhatikan pada contoh berikut. Untuk membuat variabel dengan nama angkaSaya dengan tipe data int dan memberi nilai 15 dapat dilakukan dengan cara berikut.

Contoh
<pre>int angkaSaya = 15; cout << angkaSaya;</pre>

Programmer juga dapat membuat variabel tanpa memberi nilai langsung, dan memberi nilai variabel tersebut nanti seperti contoh berikut.

Contoh
<pre>int angkaSaya; angkaSaya = 15; cout << angkaSaya;</pre>

Catat bahwa jika nilai baru dimasukkan pada variabel, nilai tersebut akan menghapus nilai yang sebelumnya dan mengganti dengan nilai yang baru dimasukkan.

Contoh
<pre>int angkaSaya = 15; // angkaSaya bernilai 15 angkaSaya = 10; // Sekarang angkaSaya bernilai 10 cout << angkaSaya; // Menampilkan nilai angkaSaya sama dengan 10</pre>

Contoh lain penggunaan tipe data dan variabel

Contoh

```
int angkaSaya = 5; // Integer (Semua angka tanpa
desimal)
double angkaDesimalSaya = 5.99; // Floating point number
(dengan angka desimal)
char karakterSaya = 'D'; // Karakter
string teksSaya = "Hello"; // Tulisan (teks)
bool booleanSaya = true; // Boolean (Benar (true) or
salah (false))
```

Menampilkan Variabel

Perintah objek `cout` digunakan bersama dengan operator `<<` untuk menampilkan variabel.

Untuk menggabungkan teks dan variabel, pisahkan dengan operator `<<` seperti berikut.

program_2_1.cpp

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

int usiaSaya = 35;

int main()
{
    cout << "Saya berusia " << usiaSaya << " tahun.";
}
```

Menjumlahkan Variabel

Untuk menjumlahkan sebuah variabel dengan variabel yang lain, dapat digunakan operator `+` sebagai:

```
program_2_2.cpp
```

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

int main()
{
    int x = 5;
    int y = 6;
    int sum = x + y;
    cout << sum;
}
```

Mendefinisikan Banyak Variabel

Untuk mendefinisikan banyak variabel dengan jenis tipe data yang sama, penggunaan tanda koma untuk memisahkan antar variabel.

```
program_2_3.cpp
```

```
#include <iostream>
using namespace std;
int x = 5, y = 6, z = 50;

int main()
{
    cout << x + y + z;
}
```

Variabel Pengenal C++

Semua variabel C++ harus didefinisikan dengan nama yang unik. Nama yang unik ini dinamakan dengan Pengenal. Tanda pengenal dapat berupa

nama pendek seperti x dan y atau lebih spesifik seperti umur, jumlah dan volume. Direkomendasikan menggunakan nama yang mendeskripsikan jenis variabel agar dapat mudah dipahami dan lebih mudah dilakukan perbaikan kode program.

```
// Bagus
int menitPerJam = 60;

// OK, tapi tidak mudah untuk dipahami apa itu arti m
int m = 60;
```

Konstanta C++

Ketika programmer tidak ingin mengubah nilai suatu variabel, dapat digunakan perintah const. Perintah tersebut akan mendeskripsikan variabel sebagai konstanta yang berarti tidak variabel yang tidak dapat diubah dan hanya bisa dibaca.

program_2_4.cpp

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

const int angkaSaya = 15;
// variabel angkaSaya selalu bernilai 15

int main()
{
    angkaSaya = 10;
    //error: variabel 'angkaSaya' hanya bisa dibaca
}
```

Programmer harus selalu mendeklarasikan variabel sebagai konstan ketika tidak ingin nilainya untuk berganti contohnya sebagai berikut.

program_2_5.cpp

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

const int menitPerJam = 60;
const float PI = 3.14;

int main()
{
    cout << "Dalam satu jam ada " << menitPerJam << "menit";
    cout << "\n";
    cout << "PI bernilai " << PI;
}
```

Masukan C++

Pada materi sebelumnya telah dipelajari perintah **cout** yang digunakan untuk menampilkan nilai. Selanjutnya akan dipelajari perintah **cin** yang digunakan untuk mendapatkan masukan dari programmer. Perintah input **cin** adalah variabel yang membaca data dari keyboard komputer/laptop dengan tanda operator (>>). Pada contoh berikut, programmer dapat memasukkan angka yang disimpan dalam variabel x. Lalu menampilkan nilai x pada terminal.

program_2_6.cpp

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

```
int x;

int main()
{
    cout << "Tuliskan sebuah angka: ";
    // Tuliskan sebuah angka dan tekan Enter

    cin >> x;
    // Mendapatkan nilai dari keyboard

    cout << "Angka tersebut adalah: " << x;
    // Menampilkan x
}
```

Perlu diketahui

`cout` dibaca dengan ejaan “see-out”. Berfungsi sebagai keluaran/ output dan menggunakan operator (<<).

`cin` dibaca dengan ejaan “see-in”. Berfungsi sebagai input/ masukan dan menggunakan operator (>>).

Membuat Kalkulator Sederhana

Pada contoh ini, programmer harus memasukkan dua buah angka. Lalu, menampilkan hasil dengan menjumlahkan dua buah angka tersebut.

```
program_2_7.cpp
```

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

```

int x, y, sum;

int main()
{
    cout << "Tuliskan sebuah angka: ";
    cin >> x;
    cout << "Tuliskan angka yang lain: ";
    cin >> y;
    sum = x + y;
    cout << "Hasil: " << sum;
}

```

Latihan

Lengkapi program berikut untuk menampilkan nilai setiap variabel pada cmd/terminal. Lengkapi bagian “___” dengan perintah yang tepat.

program_2_8.cpp

```

#include <iostream>
using namespace std;

int angkaSaya = 5;
double angkaDesimalSaya = 5.99;
char karakterSaya = 'D';
string teksSaya = "Hello";
bool booleanSaya = true;

int main()
{
    cout<< "Variabel angkaSaya adalah "<< angkaSaya<<"\n";
    cout<< "Variabel angkaDesimalSaya adalah "<<
    angkaDesimalSaya << "\n";
}

```



```

    cout<< "Variabel karakterSaya adalah "<< karakterSaya
<< "\n";
    cout<< "Variabel teksSaya adalah "<< teksSaya << "\n";
    cout<<"Variabel booleanSaya adalah "<<booleanSaya<<
"\n";
}

```

Tipe Data

Pada materi sebelumnya telah dipelajari tentang jenis-jenis tipe data dalam bahasa C++.

Contoh

```

int angkaSaya = 5;           // Angka bilangan bulat
float desimalSaya = 5.99;    // Angka desimal
double DesimalSaya = 9.98;   // Angka desimal
char karakterSaya = 'D';     // Karakter
bool booleanSaya = true;     // Boolean
string kalimatSaya = "Hello"; // Kalimat

```

Tipe data menentukan ukuran dan jenis data yang akan disimpan dalam variabel: Integer (Int) berukuran 4 byte untuk menyimpan data angka tanpa desimal. Float berukuran 4 byte untuk menyimpan bilangan pecahan, berisi satu atau lebih desimal. Cukup untuk menyimpan 7 angka desimal. Double berukuran 8 byte untuk menyimpan bilangan pecahan, berisi satu atau lebih desimal. Cukup untuk menyimpan 15 angka desimal. Boolean berukuran 1 byte untuk menyimpan nilai true (1) atau false (0). Char berukuran 1 byte untuk menyimpan karakter/ huruf/ angka atau data ASCII.

Tipe Data	Ukuran	Deskripsi
int	4 bytes	Menyimpan bilangan bulat, tanpa desimal
float	4 bytes	Menyimpan bilangan pecahan, berisi satu atau lebih desimal. Cukup untuk menyimpan 7 angka desimal
double	8 bytes	Menyimpan bilangan pecahan, berisi satu atau lebih desimal. Cukup untuk menyimpan 15 angka desimal
boolean	1 byte	Menyimpan nilai benar atau salah
char	1 byte	Menyimpan satu karakter / huruf / angka, atau nilai ASCII

Latihan

Lengkapi jenis tipe data yang sesuai dengan variabel pada bagian garis bawah yang kosong.

Latihan Jenis Tipe Data
(____) panjang = 9; (____) desimal = 8.99; (____) huruf = 'A'; (____) boolean = false; (____) kalimat = "Halo Dunia";

Tipe Data Angka

Gunakan int ketika perlu menyimpan bilangan bulat tanpa desimal, seperti 35 atau 1000, dan float atau double saat membutuhkan bilangan floating point (dengan desimal), seperti 9,99 atau 3,14515.

Contoh Program Mendefinisikan Tipe Data int

Contoh

```
#include <iostream>
using namespace std;
int myNum = 1000;
int main(){
    cout << myNum;
}
```

Contoh Program Mendefinisikan Tipe Data flot

Contoh

```
#include <iostream>
using namespace std;
float myNum = 5.75;
int main(){
    cout << myNum;
}
```

Contoh Program Mendefinisikan Tipe Data double

Contoh

```
#include <iostream>
using namespace std;
double myNum = 19.99;
int main(){
    cout << myNum;
}
```

float dengan double

Ketepatan nilai floating point menunjukkan berapa digit yang dapat dimiliki nilai setelah koma desimal. Ketepatan tipe data float hanya enam atau tujuh digit desimal, sedangkan variabel double memiliki presisi sekitar 15 digit. Oleh karena itu, lebih aman menggunakan double untuk sebagian besar kalkulasi.

Angka Ilmiah

Bilangan floating point juga bisa menjadi bilangan ilmiah dengan "e" untuk menunjukkan pangkat 10:

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

float f1 = 35e3;
double d1 = 12E4;

int main()
{
    cout << f1;
    cout << d1;
}
```

Boolean

Tipe data boolean dideklarasikan dengan kata kunci bool dan hanya dapat mengambil nilai true atau false. Ketika nilainya dikembalikan, true = 1 dan false = 0.

program

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

bool isCodingFun = true;
bool isFishTasty = false;

int main()
{
    cout << isCodingFun; // Outputs 1 (true)
    cout << isFishTasty; // Outputs 0 (false)
}
```

Karakter

Tipe data char digunakan untuk menyimpan satu karakter. Karakter harus diapit oleh tanda kutip tunggal, seperti 'A' atau 'c':

program

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

char nilaiSaya= 'B';

int main()
{
    cout << nilaiSaya;
}
```

Alternatifnya, Programmer dapat menggunakan nilai ASCII untuk menampilkan karakter tertentu:

program
<pre>#include <iostream> using namespace std; char a = 65, b = 66, c = 67; int main() { cout << a; cout << b; cout << c; }</pre>

String

Jenis string digunakan untuk menyimpan urutan karakter (teks). Ini bukan tipe bawaan, tetapi berperilaku seperti tipe dalam penggunaan paling dasar. Nilai string harus diapit tanda kutip ganda:

program
<pre>string greeting = "Hello"; cout << greeting;</pre>

Untuk menggunakan string, Anda harus menyertakan file header tambahan dalam kode sumber, pustaka <string>:

program
<pre>#include <string> // Memasukkan Pustaka String</pre>

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

// Create a string variable
string greeting = "Hello";

int main()
{
// Output string value
cout << greeting;
}
```

Aturan Penamaan Variabel

Ada beberapa hal yang harus diperhatikan dalam memberikan nama variabel. Jika penamaan variabel tidak sesuai maka akan terjadi kesalahan atau akan menyebabkan error pada program. Hal-Hal yang harus diperhatikan adalah

1. Nama variabel dapat berupa kombinasi huruf dan angka.
2. Nama variabel boleh kombinasi antara huruf besar dan huruf kecil. Namun untuk huruf awal sebaiknya huruf kecil.
3. Tidak boleh ada spasi pada nama variabel. Spasi dapat digantikan dengan simbol garis bawah atau under score (_).
4. Tidak boleh ada simbol khusus seperti @, #, \$, %, ^, !, ~, &, *, (,), tanda hubung (-) dan sebagainya.

Contoh nama variabel yang benar

luasPersegi, panjang1, lebar1, volume_lingkaran

Contoh nama variabel yang salah

Luas Persegi, 1pajang, 1lebar, !volume_lingkaran

BAB 3 OPERATOR

Operator digunakan untuk melakukan operasi pada variabel dan nilai. Pada program berikut, operator + berfungsi untuk menambahkan dua buah nilai.

Contoh
<pre>int x = 100 + 50;</pre>

Meskipun operator + sering digunakan untuk menambahkan dua data, operator + dapat pula digunakan untuk menambahkan variabel dengan nilai, atau variabel dengan variabel yang lain.

Contoh
<pre>int angka1 = 100 + 50; //hasil = 150 (100 + 50) int angka2 = angka1 + 250; //hasil = 400 (150 + 250) int angka3 = angka2 + angka2; //hasil = 800 (400 + 400)</pre>

C++ membagi operator menjadi beberapa grup berikut:

- Operator aritmetika
- Operator penugasan
- Operator perbandingan
- Operator logika
- Operator bitwise

Operator Aritmetika

Operator Penjumlahan + berfungsi menambahkan dua nilai seperti $x+y$

Operator Pengurangan - berfungsi mengurangi dua nilai seperti $x-y$

Operator Perkalian * berfungsi mengalikan dua nilai seperti $x*y$

Operator Pembagian / berfungsi membagi satu nilai dengan nilai lain x/y