



## PRAKTIKUM 1

### *Pengenalan Bahasa C++*

#### Tujuan Praktikum :

1. Mengenalkan batang tubuh program bahasa C++
2. Mengetahui aturan pemberian nama
3. Mempraktikkan variabel, konstanta, dan operator
4. Membuat file kerja dengan Bahasa C++
5. Memulai membuat program baru, meng-compile, menyimpan dan menjalankan program yang telah dibuat
6. Memanggil/membuka kembali program yang telah disimpan
7. Membuat program menjadi program yang execute

Instruksi untuk masing-masing percobaan:

1. Buka Aplikasi Dev-C++
2. Klik **File >> New >> Source File**, lalu **Save As** pada **Desktop >> (Folder Baru)**  
**Modul 1 >> Percobaan x.x.cpp**
3. Tuliskan kode program dan pilih **Compile & Run** atau tekan **F9**, perbaiki kode program jika terdapat error.
4. Jalankan program yang telah dibuat dengan cara klik **Run** atau **Ctrl+F10** untuk melihat hasilnya. Cara lain untuk menjalankan program adalah tanpa menggunakan **Dev-C++**, yakni dengan menjalankan file executable (.exe) pada folder penyimpanan file tadi.

#### **1. Dasar Bahasa C++**

Susunan penulisan program dengan C++

```
//*****//  
// deklarasi header file / Preprocessor directive  
// deklarasi fungsi / void main()  
{  
    /* KAMUS */  
    /* ALGORITMA */  
}
```

Lakukan semua percobaan-percobaan dibawah ini dan periksa hasil yang ditampilkan.



## **Percobaan 1.1: Komentar dan Hello**

```
#include <iostream>
using namespace std;
int main(){
    cout << "Hello, Unand." << endl;
    cout << "Hello, " << "Unand ";
    cout << "Padang dan Limau Manis" << endl;
    int n = 66;
    cout<< n << endl;
}
```

- Program diatas memperlihatkan 13 token, yaitu main, (, ), {, int, n, =, 66, :, cout, <<, endl, dan }.
- Token n adalah suatu variable.
- Token 66 adalah suatu konstanta bertipe integer
- Token int, return dan endl adalah suatu keyword
- Token = dan << adalah operator
- Token (, ), {, :, dan } adalah tanda baca
- Baris pertama berisi suatu preprocessor directive yang bukan bagian sebenarnya dari program.

## **Percobaan 1.2: Konstanta, Variable, dan Pencetakan**

```
#include <iostream>
#define phi 3.14
using namespace std;
int main(){
    float jari_jari, luas, keliling;
    jari_jari = 7.0;
    luas = phi * jari_jari * jari_jari;
    keliling = 2 * phi * jari_jari;
    cout << " Luas Lingkaran = " << luas << endl;
    cout << " Keliling Lingkaran = " << keliling;
}
```



## Percobaan 1.3: Tipe Data dan Ukurannya

Tabel 1. Daftar Tipe Data

Tipe data	Byte	Batasan
char	1	Bilangan bulat / ASCII antara -128 s.d. 127
unsigned char	1	Bilangan bulat antara 0 s.d. 255
short	2	Bilangan bulat antara -32.768 s.d. 32.767 ( $-2^{15}$ s.d. $2^{15}-1$ )
Unsigned short	2	Bilangan bulat antara 0 s.d. 65.535 (0 s.d. $2^{16}-1$ )
int	4	Bilangan bulat antara -2.147.483.648 s.d. 2.147.483.647 ( $-2^{31}$ s.d. $2^{31}-1$ )
unsigned int	4	Bilangan bulat antara 0 s.d. $2^{32}-1$
long int	4	Bilangan bulat antara $-2^{31}$ s.d. $2^{31}-1$
Unsigned long int	4	Bilangan bulat antara 0 s.d. $2^{32}-1$
float	4	Bilangan real antara - 3.4 E+38 s.d. 3.4E+38 (7 digit presisi)
double	8	Bilangan real antara -1.7E+308 s.d. 1.7E+308 (15 digit presisi)

```
#include <limits.h>
#include <iostream>
using namespace std;
int main() {
    cout << " Ukuran tipe integer: " << sizeof(int) << endl;
    cout << " Bilangan minimum char: " << CHAR_MIN << endl;
    cout << " Bilangan maximum uchar: " << UCHAR_MAX << endl;
    cout << " Bilangan maximum int : " << INT_MAX;
}
```

## Percobaan 1.4: String

```
#include <iostream>
#include <string.h>
using namespace std;
int main(){
    cout << ("abc\n");
    cout << ("abc\t");
    cout << ("\"Halo\"");
    cout << ("Panjang kata Unand = ") << strlen("Unand");
    cout << strlen("Selamat Pagi. \n") << endl;
    cout << strlen("Selamat Pagi. ") << endl;
    cout << strlen("Selamat") << endl;
    cout << strlen("S") << endl;
    cout << strlen("") << endl;
}
```



## Operator

### Percobaan 1.5: Operator Aritmetika

```
#include <iostream>
using namespace std;
int main () {
    int m = 82, n = 26;
    cout << m << " + " << n << " = " << m+n << endl;
    cout << m << " - " << n << " = " << m-n << endl;
    cout << m << " * " << n << " = " << m*n << endl;
    cout << m << " / " << n << " = " << m/n << endl;
    cout << m << " % " << n << " = " << m%n << endl;
    cout << " - " << m << " = " << -m << endl;
}
```

### Percobaan 1.6: Operator *Increment* dan *Decrement*

```
#include <iostream>
using namespace std;
int main () {
    int m = 44, n = 66;
    cout << "m = " << m << ", n = " << n << endl;
    ++m;
    --n;
    cout << "m = " << m << ", n = " << n << endl;
    m++;
    n--;
    cout << "m = " << m << ", n = " << n << endl;
    cout << endl;

    m = 66;
    n = ++m;
    cout << "m = " << m << ", n = " << n << endl;
    n = m++;
    cout << "m = " << m << ", n = " << n << endl;
    cout << "m = " << m++ << endl;
    cout << "m = " << m << endl;
    cout << "m = " << ++m << endl;
    cout << "m = " << m << endl;
}
```



## Percobaan 1.7: Operator Bitwise

```
#include <iostream>
using namespace std;
int main () {
    // a = 5(00000101), b = 9(00001001)

    int a = 5, b = 9;

    cout << "a = " << a << ", " << " b = " << b << endl;
    cout << "a & b = " << (a & b) << endl; // Hasilnya 00000001
    cout << "a | b = " << (a | b) << endl; // Hasilnya 00001101
    cout << "a ^ b = " << (a ^ b) << endl; // Hasilnya 00001100
    cout << "~(" << a << ") = " << (~a) << endl; // Hasilnya 11111010
    cout << "b << 1" << " = " << (b << 1) << endl; // Hasilnya 00010010
    cout << "b >> 1" << " = " << (b >> 1) << endl; // Hasilnya 00000100
}
```

## Percobaan 1.8: Type Casting

```
#include <iostream>
using namespace std;
int main () {
    int jumlah, nData;
    cout << "Masukkan Jumlah = " << endl;
    cin >> jumlah;
    cout << "Masukkan nData = " << endl;
    cin >> nData;

    float rata;
    cout << "rata = " << (jumlah/nData) << endl;
    rata = float (jumlah) / nData;
    cout << "rata = " << rata;
}
```