

A. Struktur Program

1. Struktur Kode Program C++

a. Preprocessor directive (pengarah kompilator) { # }
Sebagai pengatur kompilasi

b. Fungsi utama dan tambahan { main () }

Fungsi utama adalah bagian yang pertama kali akan dieksekusi untuk di jalankan, di tandai dengan nama main di ikuti tanda ().

c. Deklarasi

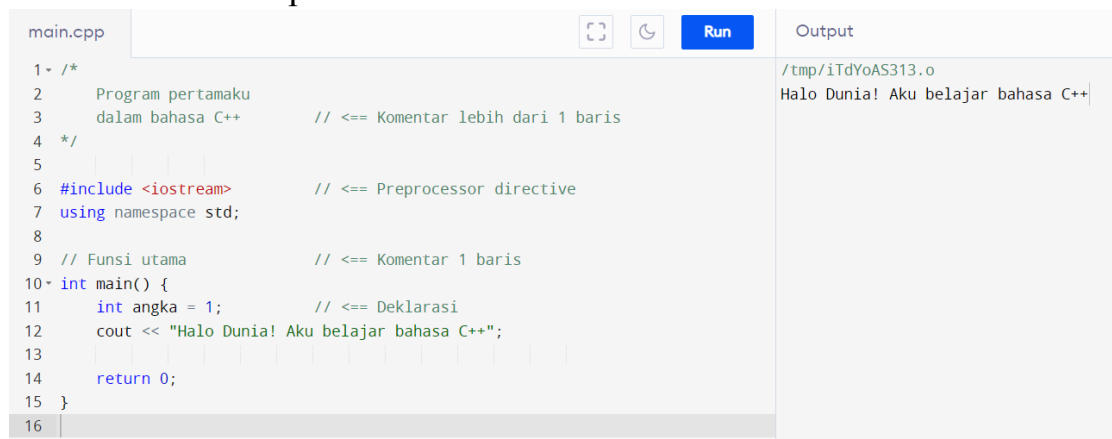
Pernyataan (statement) berisi pengenalan variable, konstanta, fungsi dll.

d. Definisi fungsi

Bagian yang menyertai fungsi. Bagian definisi selalu diawali tanda kurawal buka “{” diakhiri tanda kurawal tutup “}”. Di dalam definisi fungsi ini terdapat satu atau lebih pernyataan (statement). Setiap pernyataan harus diakhiri tanda titik koma (;) untuk mengakhiri pernyataan tersebut sudah selesai.

e. Komentar

Biasanya berisi keterangan tertentu. Komentar tidak akan dieksekusi dan tidak akan di tampilkan saat program di jalankan. Dalam C++ ada dua cara menuliskan komentar, satu baris diawali dengan tanda // dan komentar satu baris atau lebih diapit oleh tanda /* dan */ .



The screenshot shows a code editor with a file named 'main.cpp'. The code is as follows:

```
1 * /*
2   Program pertamaku
3   dalam bahasa C++      // <== Komentar lebih dari 1 baris
4   */
5
6   #include <iostream>    // <== Preprocessor directive
7   using namespace std;
8
9   // Fungsi utama      // <== Komentar 1 baris
10  int main() {
11      int angka = 1;      // <== Deklarasi
12      cout << "Halo Dunia! Aku belajar bahasa C++";
13
14      return 0;
15  }
16
```

On the right side, there is an 'Output' panel showing the result of running the program:

```
/tmp/iTdYoAS313.o
Halo Dunia! Aku belajar bahasa C++
```

```

1  /*
2      Program pertamaku
3      dalam bahasa C++      // <== Komentar lebih dari 1 baris
4  */
5
6  #include <iostream>      // <== Preprocessor directive
7  using namespace std;
8
9  // Fungsi utama      // <== Komentar 1 baris
10 int main() {
11     int angka = 1;      // <== Deklarasi
12     cout << "Halo Dunia! Aku belajar bahasa C++";
13
14     return 0;
15 }
16

```



Fungsi Utama
berikut definisinya

B. Pembuatan Program dengan C++

Buka web <https://www.programiz.com/cpp-programming/online-compiler/>

Coba tuliskan kode ini

main.cpp	Run	Output
<pre> 1 /* Program Pertamaku */ 2 #include <iostream> 3 using namespace std; 4 5 int main() { 6 cout << "Halo Dunia! Aku belajar bahasa C++"; 7 8 return 0; 9 } </pre>		<pre> /tmp/jazTed9V4q.o Halo Dunia! Aku belajar bahasa C++ </pre>

Yang perlu di perhatikan :

1. Spasi, baris baru, dan baris kosong digunakan supaya kode program lebih mudah di baca. Baris kosong akan diabaikan oleh compiler.
2. Bahasa C++ bersifat case sensitive, yaitu penulisan huruf besar dan kecil berpengaruh. Jadi Nama berbeda dengan nama atau NAMA.

C. Aturan Penulisan Bahasa Pemrograman C++

1. Variabel (Tempat menyimpan nilai atau data)
Setiap variable punya nama (identifier) dan nilai (value).

Contoh:

Nama = "Budi"

Harga = 2500

Artinya nama dan harga adalah nama dari variable, sedangkan "Budi" dan 2500 adalah nilai dari masing masing variable.

Ada 5 tipe data dasar dalam C++.

Tipe Data	Kata Kunci	Jenis Data yg disimpan
Bilangan bulat (integer)	Int, short, long	Bilangan bulat tanpa decimal (1 2 3 atau -1 2 3).
Bilangan desimal	Double, float	Bilangan decimal (pecahan) (19,9 atau -19,9).
Huruf (character)	char	Huruf tunggal / 1 karakter, missal 'a' atau 'B'. nilai atau huruf diapit oleh tanda kutip.
Teks	string	Kata/kalimat, contoh "Hello Word" , nilai diapit oleh tanda kutip ganda.
Boolean	bool	Nilai logika benar (true) atau salah (false).

Aturan pemberian nama variable dalam C++ :

- Ditulis menggunakan huruf, angka, atau garis bawah.
- Harus diawali dengan huruf.
- Tidak boleh menggunakan spasi.
- Tidak boleh menggunakan karakter khusus, seperti : ,+,-,*,/,<,>,&,(,) dll.
- Tidak boleh menggunakan kata kunci bahasa pemrograman.
- Bersifat case sensitive, yaitu huruf besar dan kecil di bedakan.

Contoh variable yang benar :

- Namasiswa
- X12
- harga_total
- JenisMotor

Contoh variable yang salah :

- Nama siswa (salah karena menggunakan spasi)
- 12X (salah karena dimulai dengan angka)
- harga.total (salah karena menggunakan karakter titik)
- Jenis Motor (salah karena menggunakan spasi)
- for (salah karena menggunakan kata kunci Bahasa pemrograman)

Berikut contoh penggunaan variable dalam C++/

main.cpp

Run

```
1  /* Belajar Variabel */
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7      string nama = "Adi Jaya";
8      char kelas = 'A';
9      int umur;
10
11     cout << " Namaku " << nama << "\n";
12     cout << "Kelasku " << kelas << "\n";
13     umur = 20;
14     cout << "Umur " << umur;
15
16     return 0;
17 }
```

Hasilnya

main.cpp	Run	Output
<pre>1 /* Belajar Variabel */ 2 3 #include <iostream> 4 using namespace std; 5 6 int main() { 7 string nama = "Adi Jaya"; 8 char kelas = 'A'; 9 int umur; 10 11 cout << " Namaku " << nama << "\n"; 12 cout << "Kelasku " << kelas << "\n"; 13 umur = 20; 14 cout << "Umur " << umur; 15 16 return 0; 17 }</pre>	<pre>/tmp/jazTed9V4q.o Namaku Adi Jaya Kelasku A Umur 20</pre>	

2. Input dan Output

Pada kode pemrograman sebelumnya, kita telah menjumpai perintah “cout <<”. Perintah cout digunakan untuk menampilkan output ke layar. Kita juga menjumpai kode perintah “\n” atau bisa ditulis dengan perintah “endl”, yaitu perintah untuk menambahkan baris baru pada layar.

Input pada program digunakan untuk meminta masukan dari pengguna. Untuk meminta input dari pengguna kata perintah cin diikuti tanda >>.

Coba ketikkan kode di bawah :

```
main.cpp   Run
```

```
1  /* Belajar Input Output */
2
3  #include <iostream>
4  using namespace std;
5
6  int main() {
7      int bil;
8
9      cout << "masukan sembarang bilangan: ";
10     cin >> bil;
11     cout << "bilangan yang dimasukan adalah " << bil;
12
13     return 0;
14 }
```

Hasilnya:

main.cpp   Run	Output
<pre>1 /* Belajar Input Output */ 2 3 #include <iostream> 4 using namespace std; 5 6 int main() { 7 int bil; 8 9 cout << "masukan sembarang bilangan: "; 10 cin >> bil; 11 cout << "bilangan yang dimasukan adalah " << bil; 12 13 return 0; 14 }</pre>	<pre>/tmp/nOmVUh4MQs.o masukan sembarang bilangan: 88 bilangan yang dimasukan adalah 88</pre>

Program di atas meminta memasukkan sembarang bilangan, kemudian menampilkan bilangan yang dimasukkan tersebut.

D. Struktur Dasar Algoritma

Ada 3 struktur dasar yaitu struktur berurutan (sequencing), percabangan (branching) dan pengulangan (looping).

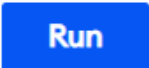
1. Struktur berurutan (sequencing)

Contohnya adalah seperti kode pemrograman seperti yang telah kita buat tadi.

2. Struktur percabangan (branching)

Dalam Bahasa C++ algoritma percabangan di tandai dengan kata kunci “if”, di ikuti dengan kondisi atau persyaratan yang harus dipenuhi.

Contoh kondisi algoritma percabangan :

```
main.cpp   
1- /* Belajar percabangan
2   menentukan nilai mutlak */
3
4 #include <iostream>
5 using namespace std;
6
7- int main() {
8   float bil;
9
10  cout << "Masukan sembarang bilangan: ";
11  cin >> bil;
12  cout << "Nilai mutlak dari " << bil;
13-  if (bil < 0) {
14      bil = bil * (-1);
15  }
16  cout << " adalah " << bil;
17
18  return 0;
19 }
```

Program diatas meminta memasukkan sembarang bilangan dari pengguna, kemudian mengeluarkan nilai mutlak dari bilangan yang dimasukan tersebut. Sesuai definisi nilai mutlak, jika bilangan yang dimasukan kurang dari 0 maka

dikalikan dengan -1. Sebaliknya, jika bilangan yang dimasukkan lebih dari atau sama dengan 0 maka akan langsung ditampilkan.

Output

```
/tmp/n0mvUh4MQs.o  
Masukan sembarang bilangan: 39  
Nilai mutlak dari 39 adalah 39
```

Output

```
/tmp/n0mvUh4MQs.o  
Masukan sembarang bilangan: -44  
Nilai mutlak dari -44 adalah 44
```

3. Struktur pengulangan (looping)

Ada beberapa bentuk pengulangan. Salah satunya pengulangan for dan pengulangan while. Untuk pengulangan For, dengan ketentuan kita harus tahu banyaknya pengulangan yang akan di lakukan

Untuk pengulangan while ini bis akita gunakan jika kita tidak tahu pasti harus berapa kali mengulang.

Contoh Penggunaan pengulangan For pada bilangan :

main.cpp



Run

```
1 ▾ /* Belajar pengulangan For
2     | menampilkan kelipatan bilangan*/
3
4 #include <iostream>
5 using namespace std;
6
7 ▾ int main() {
8     int i, kelipatan, batas;
9
10    cout << "Kelipatan berapa? ";
11    cin >> kelipatan;
12    cout << "Batasnya? ";
13    cin >> batas;
14    cout << "\nBilangan kelipatan " << kelipatan;
15    cout << " kurang dari " << batas << " adalah \n";
16 ▾    for (i=kelipatan ; i< batas ; i+=kelipatan){
17        |    cout << i << endl;
18    }
19    return 0;
20 }
```

Output

```
/tmp/LfaMzuKruU.o
Kelipatan berapa? 4
Batasnya? 35
Bilangan kelipatan 4 kurang dari 35 adalah
4
8
12
16
20
24
28
32
|
```

Program di atas meminta masukan bilangan kelipatan yang akan ditampilkan dan batasnya. Perhatikan bahwa nilai awal counter tidak selalu harus dari 0.

E. Struktur data sederhana (Array)

Array adalah struktur data yang menyimpan sekumpulan elemen yang bersifat sama, setiap elemennya diakses langsung melalui indeksnya. Array dapat dianalogikan seperti lemari yang memiliki banyak kotak penyimpanan atau laci yang diberi nomor berurutan.

Pada variable array kita tidak hanya menentukan tipe dasarnya saja, tetapi juga jumlah elemen array tersebut atau dalam batas atas indeksnya. Nilai indeks awal adalah 0 bukan 1. Untuk mengisi data array kita dapat langsung menentukan pada indeks berapa kita akan isikan demikian juga untuk memanggil atau menampilkan data dari array.

coba ketikkan kode program di bawah ini :



Python



R



Database



Jupyter



Coffee



C



C++



C

JS



Go

PHP

main.cpp

```
1 // Belajar array sederhana
2
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     int nilai [3]; //deklarasi array
8
9     //pengisian array
10    nilai[0] = 17;
11    nilai[1] = 8;
12    nilai[2] = 45;
13
14    //menampilkan isi array
15    cout<< " Isi array: \n";
16    cout<< nilai[0] << endl;
17    cout << nilai[1] << endl;
18    cout << nilai[2] << endl;
19
20    return 0;
21 }
```

```
Output
/tmp/ZxGwFv2101.o
Isi array:
17
8
45
```

Perhatikan pada kode program diatas, saat mendeklarasikan array nilai indeks adalah 3 untuk menunjukkan jumlah elemennya ada 3, sedangkan saat mengisi dan menampilkan, indeks dimulai dari angka 0 bukan dari 1. Penggunaan array sering disertai dengan struktur pengulangan untuk memudahkan dalam mengisi dan menampilkan isi array.

cobalah kode peprogaman di bawah ini :

```
main.cpp
1- /*
2   Belajar array sederhana
3   Mengisi array dengan perulangan lalu hitung nilai rata-rata
4   */
5
6   #include <iostream>
7   using namespace std;
8
9   int main()
10  {
11      float bil [4]; //deklarasi nilai dengan 4 indeks
12      float rata; //nilai rata-rata
13      int i; //counter
14      cout<< " Masukan 4 bilangan!\n";
15      for (i=0; i<4; i++)
16      {
17          cout<< "bilangan ke-" << i+1 <<": ";
18          cin>>bil[i]; //pengisian array
19      }
20      rata= (bil[0] + bil[1] + bil[2] + bil[3])/4; //hitung rata-rata
21
22      //menampilkan rata-rata
23      cout << " rata-ratanya adalah " <<rata;
24
25      return 0;
26  }
27
```

```
Output
/tmp/hpuQo5DnqL.o
Masukan 4 bilangan!
bilangan ke-1: 12
bilangan ke-2: 14
bilangan ke-3: 17
bilangan ke-4: 15
rata-ratanya adalah 14.5
```

Pada program diatas, pengguna diminta memasukkan 4 bilangan yang disimpan ke dalam array. Pernyataan “i+1” pada perintah cout tidak akan menambah nilai counter i, tetapi hanya menampilkan nilai variable i yang ditambah 1. nilai rata-ratanya dihitung dengan menjumlahkan semua bilangan pada array kemudian dibagi 4.