



MODUL #06

PEMROGRAMAN TERSTRUKTUR

TIPE DATA DALAM C++

NUR ALAMSYAH

TIPE DATA DALAM C++

Tipe data adalah klasifikasi variabel untuk menentukan bentuk keadaan data yang akan disimpan dalam memori dan jenis operasi apa yang dapat dilakukan oleh variabel tersebut.

Macam-macam Tipe Data dalam C++

variabel memiliki **3 jenis tipe data** yaitu Variabel Numerik, Variabel Text dan Boolean.

1. Variabel Numerik : Memungkinkan kita untuk menyimpan data dalam bentuk bilangan.

Variabel numerik dapat dibagi menjadi beberapa bagian yaitu : Bilangan Bulat (*Integer* `int`), Bilangan Desimal Berpresisi Tunggal (*Floating Point* `float`). Bilangan Desimal Berpresisi Ganda (*Double Precision* `double`).

2. Variabel Text : memungkinkan kita untuk menyimpan data dalam bentuk Karakter. Bahasa Pemrograman C++ menyediakan beberapa tipe data dasar seperti `char` (*Character* atau Karakter Tunggal) dan `string` (Kumpulan Karakter).

3. Boolean : pada bahasa pemrograman C++ ditulis dengan penulisan `bool` merupakan jenis memori yang dapat mewakili satu nilai dari dua pilihan yaitu `1` (*True*) atau `0` (*False*).

Tipe data yang umum digunakan :

Tipe Data	Bentuk Penulisan
<i>Boolean</i>	<code>bool</code>
<i>Character</i>	<code>char</code>
<i>Integer</i>	<code>int</code>
<i>Floating Point</i>	<code>float</code>
<i>Double Floating Point</i>	<code>double</code>
<i>String</i>	<code>string</code>
<i>Valueless</i>	<code>void</code>

Dari tipe data di atas bisa kita modifikasi menggunakan **Modifiers Type** untuk memberikan fungsi lebih kepada Variabel.

Modifiers Type	Fungsi
<code>signed</code>	Untuk membuat Tipe Data Numerik menjadi Negatif
<code>unsigned</code>	Untuk membuat Tipe Data Numerik menjadi Positif
<code>short</code>	Pemesanan Jarak dan Ukuran memori lebih Kecil
<code>long</code>	Pemesanan Jarak dan Ukuran memori lebih Besar

Ketika kita sedang membuat variabel, kita sebenarnya sedang memesan memori untuk penyimpanan kepada komputer sebesar tipe data yang digunakan karena setiap tipe data memiliki besar memori yang berbeda.

Di bawah ini adalah tabel keterangan detail mengenai masing-masing Tipe Data termasuk Tipe Data yang menggunakan Tipe Modifikasi. keterangan cara penulisan dan mengenai seberapa besar memori yang akan digunakan untuk penyimpanan.

Daftar Lengkap Tipe data C++

Tipe Data	Bentuk Penulisan	Ukuran Memori
<i>Integer</i>	<code>int</code>	4 byte = 32 bit
	<code>signed int</code>	4 byte = 32 bit
	<code>signed short int</code>	2 byte = 16 bit
	<code>signed long int</code>	4 byte = 32 bit
	<code>signed long long int</code>	4 byte = 32 bit
	<code>unsigned int</code>	4 byte = 32 bit
	<code>unsigned short int</code>	2 byte = 16 bit
	<code>unsigned long int</code>	4 byte = 32 bit
	<code>unsigned long long int</code>	4 byte = 32 bit

<i>Floating Point</i>	<code>float</code>	4 byte = 32 bit
<i>Double Floating Point</i>	<code>double</code>	8 byte = 64 bit
	<code>long double</code>	12 byte = 96 bit
<i>Boolean</i>	<code>bool</code>	1 byte = 8 bit
<i>Character</i>	<code>char</code>	1 byte = 8 bit
(C++ 11)	<code>char16_t</code>	2 byte = 16 bit
(C++ 11)	<code>char32_t</code>	4 byte = 32 bit
	<code>wchar_t</code>	2 byte = 16 bit
	<code>unsigned char</code>	1 byte = 8 bit
	<code>signed char</code>	1 byte = 8 bit
<i>String</i>	<code>string</code>	4 byte = 32 bit
<i>Pointer Null (C++ 11)</i>	<code>decltype(nullptr)</code>	4 byte = 32 bit
<i>Valueless</i>	<code>void</code>	1 byte = 8 bit

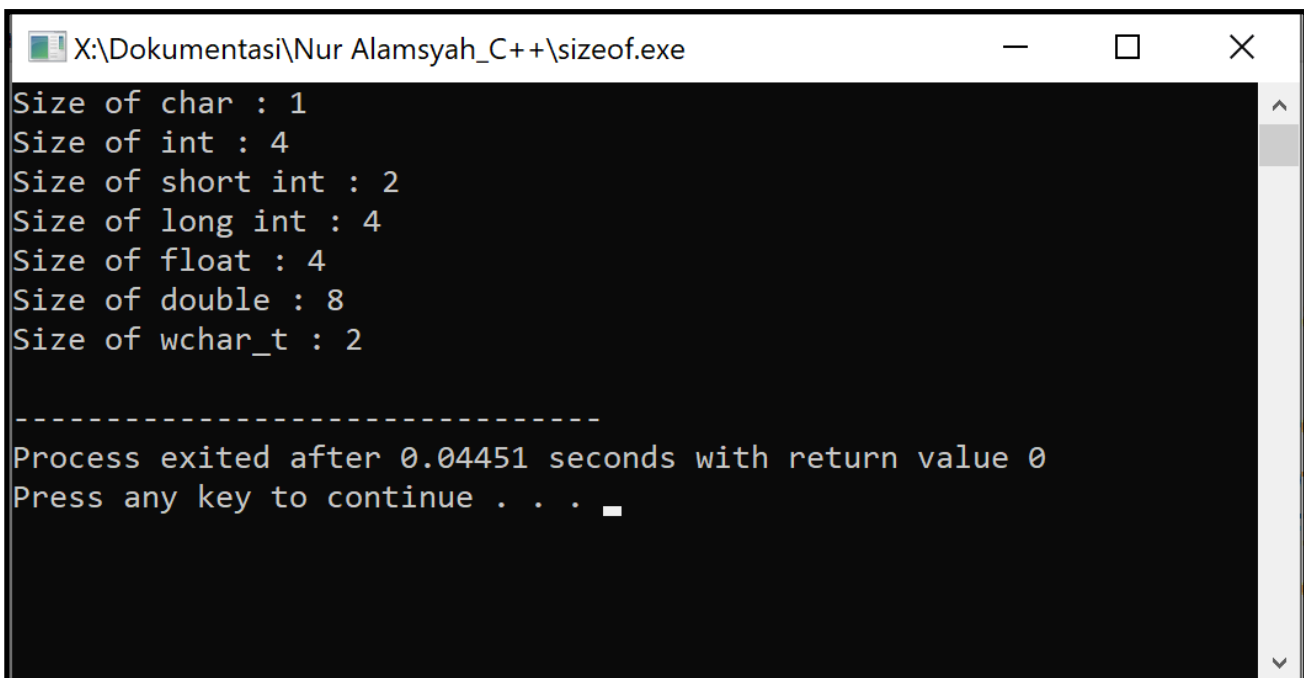
Tetapi Untuk ukuran memori mungkin bisa berbeda-beda pada setiap komputer yang anda gunakan. Untuk melihat pasti ukuran memori setiap tipe data pada komputer anda, anda bisa menggunakan `function sizeof()` untuk memeriksa besar memori yang dipakai.

Function `sizeof()` biasanya digunakan untuk mengetahui besar memori dari tipe data, variabel, obyek dan semua deklarasi yang dapat memakan memori pada komputer.

Praktikkan: Buat file baru bernama “sizeof.cpp” dan simpan difolder

```
sizeof.cpp
1  #include<iostream>
2  using namespace std;
3
4  int main(){
5      cout<<"Size of char : "<<sizeof(char)<<endl;
6      cout<<"Size of int : "<<sizeof(int)<<endl;
7      cout<<"Size of short int : "<<sizeof(short int)<<endl;
8      cout<<"Size of long int : "<<sizeof(long int)<<endl;
9      cout<<"Size of float : "<<sizeof(float)<<endl;
10     cout<<"Size of double : "<<sizeof(double)<<endl;
11     cout<<"Size of wchar_t : "<<sizeof(wchar_t)<<endl;
12     return 0;
13 }
```

Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\sizeof.exe
Size of char : 1
Size of int : 4
Size of short int : 2
Size of long int : 4
Size of float : 4
Size of double : 8
Size of wchar_t : 2

-----
Process exited after 0.04451 seconds with return value 0
Press any key to continue . . .
```

Fungsi dan Cara penggunaan macam-macam dasar Tipe Data

Boolean `bool`

Boolean adalah salah satu tipe data yang dapat memiliki nilai dari dua pilihan yaitu **True 1** atau **False 0** . Tipe data ini biasanya digunakan untuk memberikan dan memastikan kebenaran dari sebuah operasi atau kondisi program.

Bentuk penulisan

```
1 bool daftar_variabel = inisialisasi;
```

Contoh penulisan

```
1 bool pilihan=0; // Contoh mendirikan variable  
   Boolean dengan inisial
```

Praktikkan: Buat file baru bernama “`bool.cpp`” dan simpan difolder

```
bool.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int angka; //Tipe data Integer untuk menyimpan data bersifat bilangan bulat
7      bool hasil = true; //Tipe data Boolean yang mempunyai nilai awal true ( 1 )
8
9      cout<<"Masukan angka = ";cin>>angka; //Input Bilangan bulat ke variabel "angka"
10
11     hasil = angka > 10;
12     //variabel "hasil" akan menyimpan nilai kondisi dari variabel "angka" > 10.
13     cout<<hasil; //jika kondisi True maka akan tertulis angka 1 atau 0 jika kondisi False.
14     return 0;
15 }
```

Jika kita inputkan nilai lebih dari 10 misal angka 22 maka akan menghasilkan angka 1 (true) dan jika kurang dari 10 misal angka 3, maka akan menghasilkan angka 0 (false).

Hasil Tampilan:

```
X:\Dokumentasi\Nur Alamsyah_C++\bool.exe
Masukan angka = 22
1
-----
Process exited after 3.916 seconds with return value 0
Press any key to continue . . .
```

```
X:\Dokumentasi\Nur Alamsyah_C++\bool.exe
Masukan angka = 3
0
-----
Process exited after 3.358 seconds with return value 0
Press any key to continue . . .
```

Character **char**

Char adalah salah satu tipe data sembarang huruf, angka, dan simbol. Yang memungkinkan kita untuk memesan memori yang dapat menyimpan nilai dalam bentuk karakter tunggal seperti 'a', '\$' dan lain-lain.

Bentuk penulisan

```
1 char nama_variabel = inisialisasi ;
```

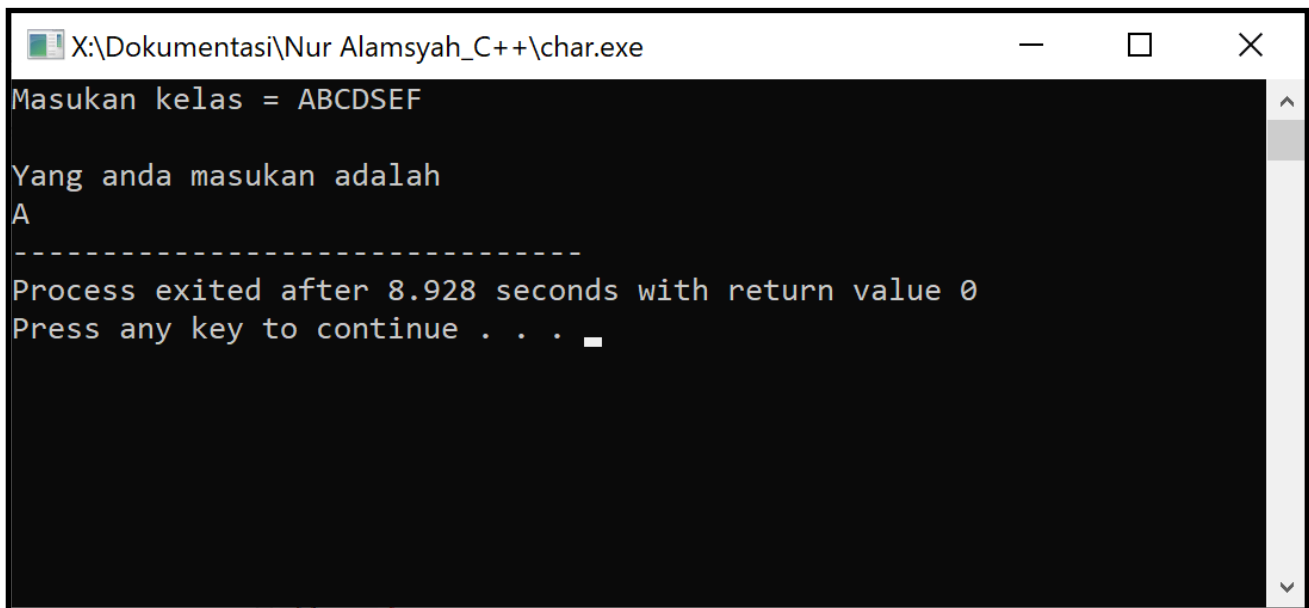
Contoh penulisan

```
1 char grade = 'a' ;
```

Praktikkan: Buat file baru bernama “char.cpp**” dan simpan difolder**

```
char.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      char x='A'; //Variabel Char hanya dapat memuat satu karakter.
6
7      cout<<"Masukan kelas = ";
8      cin>>x; //Jika anda memasukan karakter lebih dari satu maka variabel "x" hanya akan mengambil satu karakter pertama.
9      cout<<endl<<"Yang anda masukan adalah "<<endl<<x;
10
11     return 0;
12 }
```


Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\char.exe
Masukan kelas = ABCDSEF
Yang anda masukan adalah
A
-----
Process exited after 8.928 seconds with return value 0
Press any key to continue . . .
```

Integer `int`

Integer adalah salah satu tipe data numerik yang memungkinkan kita untuk menyimpan data dalam bentuk bilangan bulat.

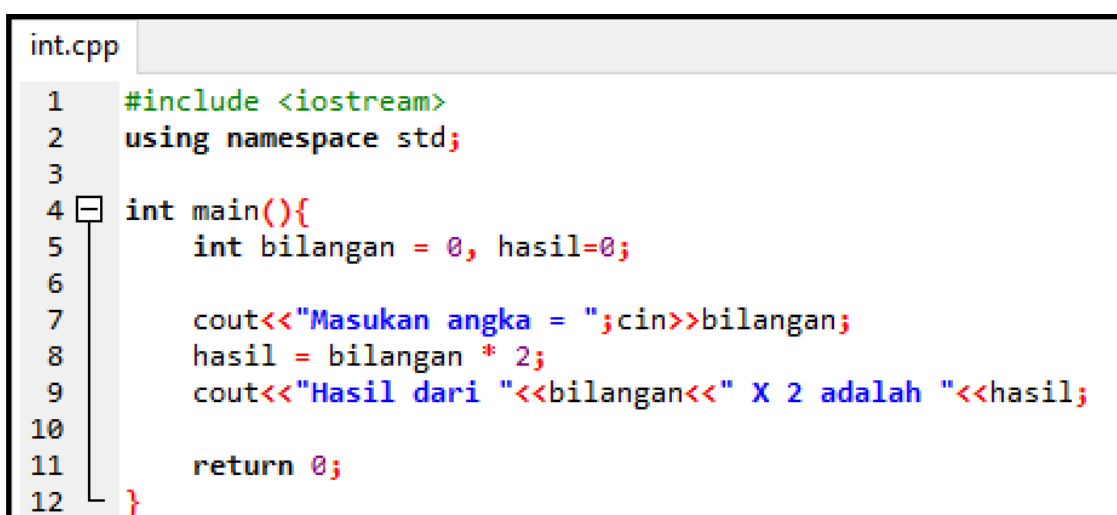
Bentuk penulisan

```
1 int daftar_variabel = inisialisasi;
```

Contoh penulisan

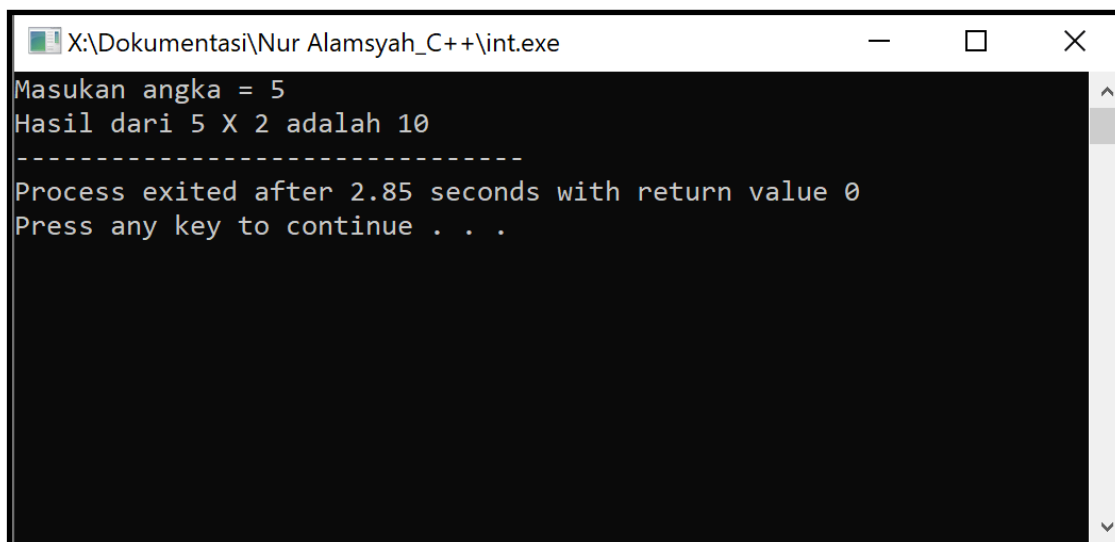
```
1 int bilangan = 10;
```

Praktikkan: Buat file baru bernama `int.cpp` dan simpan di folder



```
int.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      int bilangan = 0, hasil=0;
6
7      cout<<"Masukan angka = ";cin>>bilangan;
8      hasil = bilangan * 2;
9      cout<<"Hasil dari "<<bilangan<<" X 2 adalah "<<hasil;
10
11     return 0;
12 }
```


Hasil Tampilan:



Float Point **float**

Float adalah salah satu tipe data Numerik yang memungkinkan untuk menyimpan nilai dalam memori bersifat bilangan pecahan atau real, maupun eksponensial.

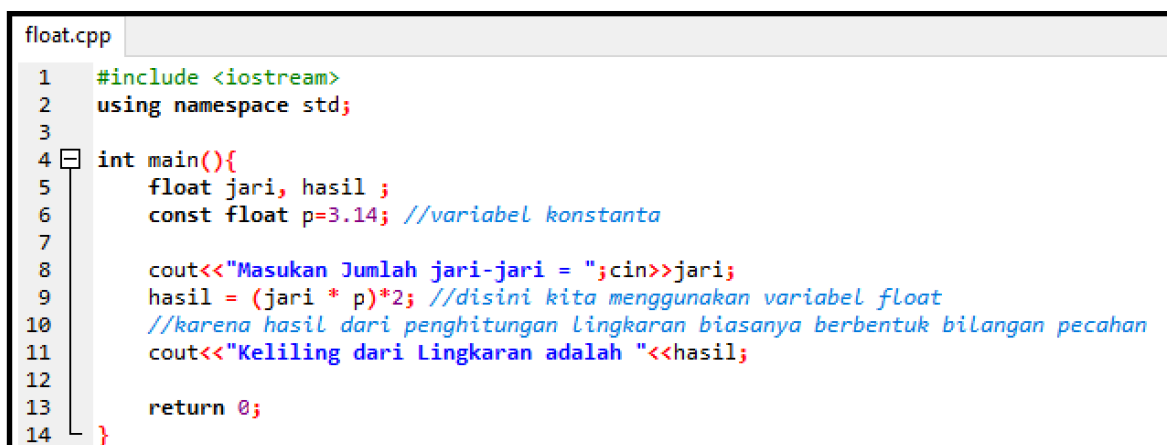
Bentuk penulisan

```
1 float daftar_variabel = inisialisasi;
```

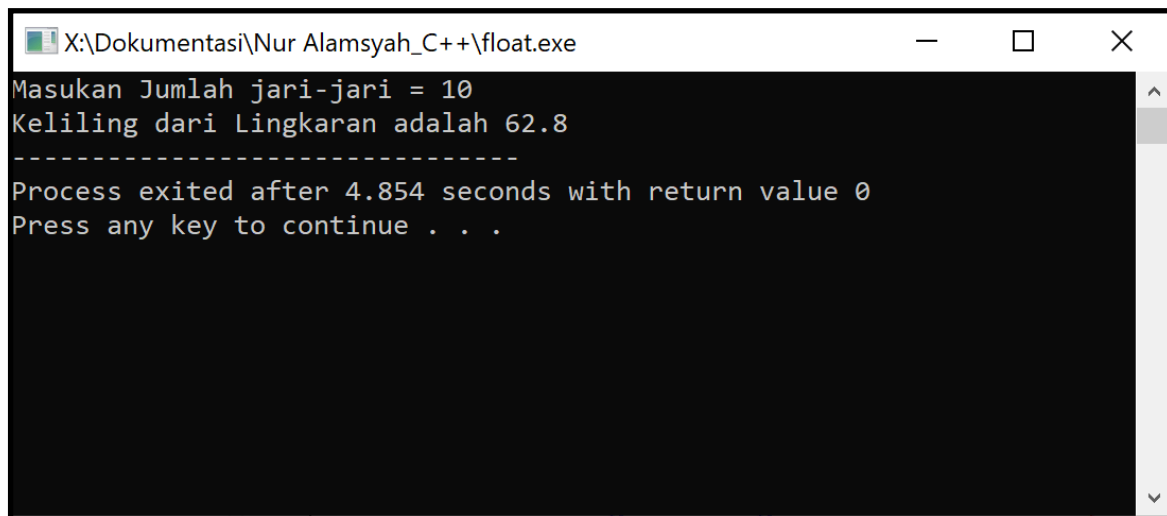
Contoh penulisan

```
1 float p=3,14;
```

Praktikkan: Buat file baru bernama “float.cpp**” dan simpan difolder**



Hasil Tampilan:



Double Floating Point `double`

`Double` sama seperti `float` adalah salah satu tipe data yang bersifat menyatakan bilangan pecahan atau real, maupun eksponensial. Bedanya adalah penyimpanan angka maksimal lebih besar daripada `float`, otomatis `double` juga akan membutuhkan memori yang lebih besar.

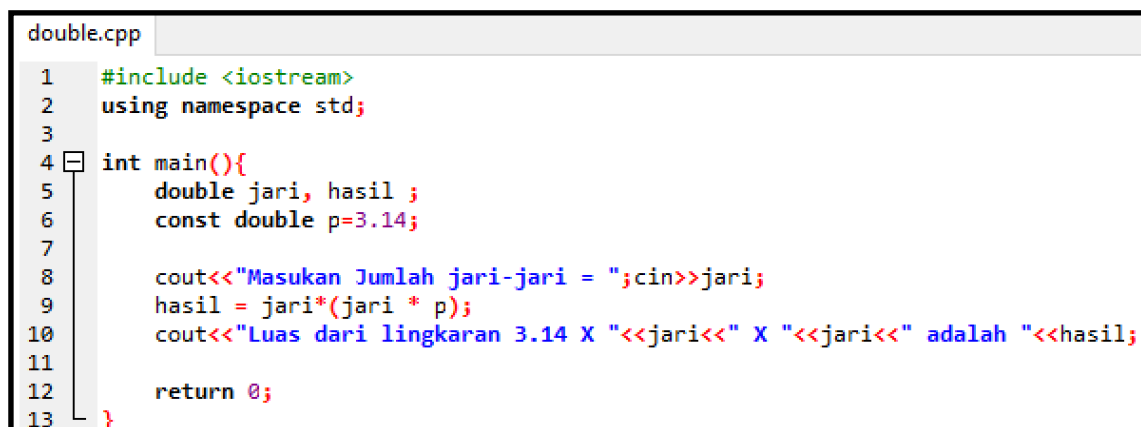
Bentuk penulisan

```
1 double daftar_variabel = inisialisasi;
```

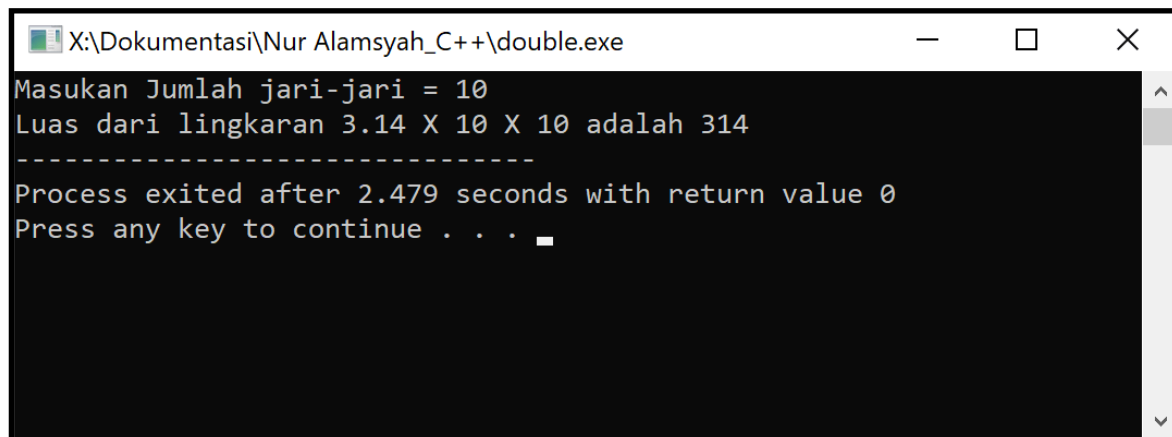
Contoh penulisan

```
1 double angka = 3.14;
```

Praktikkan: Buat file baru bernama `"double.cpp"` dan simpan difolder



Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\double.exe
Masukan Jumlah jari-jari = 10
Luas dari lingkaran 3.14 X 10 X 10 adalah 314
-----
Process exited after 2.479 seconds with return value 0
Press any key to continue . . .
```

String **string**

Merupakan tipe data *Text* yang memungkinkan kita menyimpan nilai dengan bentuk *text*, kumpulan dari karakter.

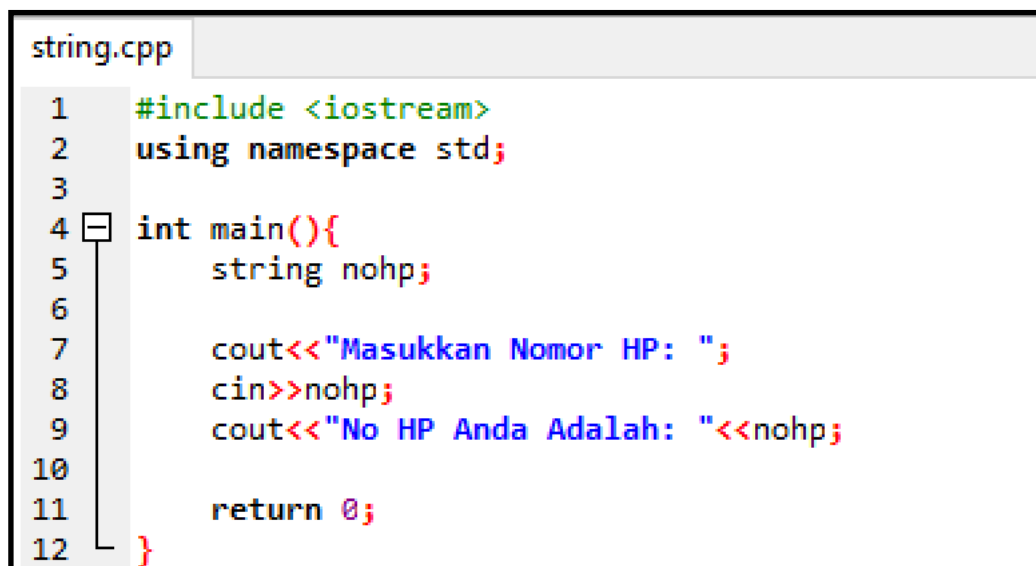
Bentuk penulisan

```
1 string daftar_variabel = inisialisasi;
```

Bentuk penulisan

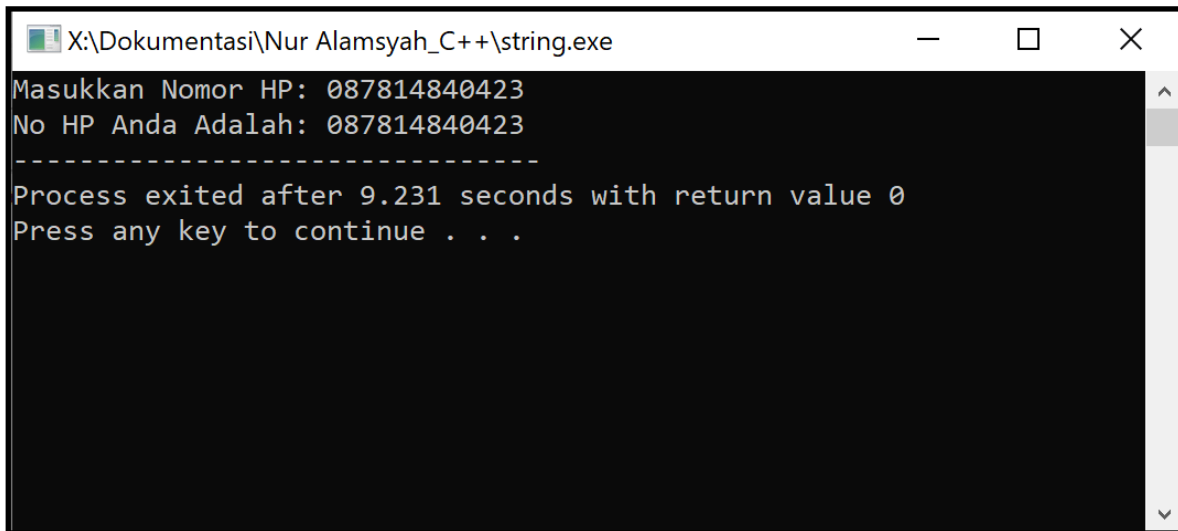
```
1 string kalimat = "Selamat Belajar Di fti uniska";
```

Praktikkan: Buat file baru bernama “string.cpp**” dan simpan difolder**



```
string.cpp
1  #include <iostream>
2  using namespace std;
3
4  int main(){
5      string nohp;
6
7      cout<<"Masukkan Nomor HP: ";
8      cin>>nohp;
9      cout<<"No HP Anda Adalah: "<<nohp;
10
11     return 0;
12 }
```

Hasil Tampilan:



```
X:\Dokumentasi\Nur Alamsyah_C++\string.exe
Masukkan Nomor HP: 087814840423
No HP Anda Adalah: 087814840423
-----
Process exited after 9.231 seconds with return value 0
Press any key to continue . . .
```