



# DASAR PEMROGRAMAN

- **Reserved Words/Keywords**
- **Identifier**
- **Tipe Data**



Yoannita

# Reserved Words/Keywords

- Kata-kata yang dikenal oleh C++ dan mempunyai kegunaan khusus dalam program.
- *Keywords* tidak boleh digunakan sebagai identifier (nama variabel, kelas, method, dll).
- Contoh keywords dalam C++ :

# List of all the C++ keywords

asm	enum	protected	try
auto	explicit *	public	typedef
bool *	export *	register	typeid *
break	extern	reinterpret_	typename *
case	false *	cast *	union
catch	float	return short	unsigned
char	for friend	signed	using *
class const	goto	sizeof	virtual
const_cast	if	static	void
*	inline	static_cast	volatile
continue	int	*	wchar_t *
default	long	struct	while
delete	mutable *	switch	
do	namespace	template	
double	* new	this	
dynamic_c	operator	throw	
ast * else	private	true *	

The 15 keywords that are starred (\*) were added to the language after it's initial release

# IDENTIFIER (pengenal)

- *Identifier adalah suatu pengenal atau pengidentifikasi yang kita deklarasikan agar kompiler dapat mengenalinya. Identifier sendiri dapat berupa nama variabel, konstanta, fungsi, kelas, template, maupun namespace.*
- *Identifier yang berperan sebagai variabel dan konstanta berfungsi untuk menampung sebuah nilai yang digunakan dalam program. Identifikasi ini dilakukan untuk mempermudah proses penanganan data atau nilai, misalnya untuk memasukkan dan menampilkan nilai.*

# Aturan Penamaan Identifier

- a. Identifier dapat berupa satu atau beberapa karakter
  - ✓ Huruf (A s/d Z, a s/d z)
  - ✓ Digit (0 1 2 3 4 5 6 7 8 9 )
  - ✓ Garisbawah (\_)
- b. *Identifier* **HARUS dimulai/diawali** dengan huruf, underscore/ garis bawah (\_) atau tanda dollar (\$).
- c. Untuk selebihnya dapat menggunakan karakter apapun, **kecuali karakter yang digunakan sebagai operator** oleh C++.
- d. Identifier haruslah **bukan merupakan keywords** yang dikenal C++

- Identifier* adalah *case sensitive* (membedakan huruf besar/ kecil)
- Variabel yang didefinisikan dengan awalan \* (tanda bintang) merupakan **pointer** (bukan berisi data, melainkan alamat dari suatu data)

Contoh variabel pointer : `int *pint;`

# Penamaan Identifier

- d. Karakter-karakter di dalam identifier **tidak boleh dipisah dengan spasi**. Hal ini mengingat bahwa nama adalah kesatuan leksikal, maka setiap nama harus ditulis secara utuh.
- Anda dapat mengganti Spasi dengan karakter “\_”. Namun, sebagai pengganti bentuk penulisan tradisional untuk identifier, yang menggunakan huruf kecil dengan garis-bawah sebagai pemisah kata
    - nama\_karyawan
    - Nama\_Karyawan, beberapa identifier dituliskan dengan huruf kapital pada awal kata, seperti :
    - namaKaryawan atau NamaKaryawan

1. Huruf awal untuk **nama variabel dan method** sebaiknya bukan huruf kapital, sesuai dengan standar kaidah penamaan identifier, sehingga dapat dibedakan dgn nama class.

- bilanganGenap                      nama variabel
- bilanganGenap()                    nama method
- BilanganGenap                      nama class

## 2. Berikanlah nama identifier sesuai dengan deskripsinya

<code>int ccount</code>	<b>Bad</b>	<b>Nobody knows what a ccount is</b>
<code>int customerCount</code>	<b>Good</b>	<b>Clear what we're counting</b>
<code>int i</code>	<b>Bad</b>	<b>What does i stand for?*</b>
<code>int index</code>	<b>Good</b>	<b>This variable is indexing something</b>
<code>int _count</code>	<b>Bad</b>	<b>Do not start variable names with underscore</b>
<code>int count</code>	<b>Either</b>	<b>Okay only if obvious what we're counting</b>
<code>int data</code>	<b>Bad</b>	<b>What kind of data?</b>
<code>int value1, value2</code>	<b>Either</b>	<b>Can be hard to differentiate between the two</b>
<code>int numberOfApples</code>	<b>Good</b>	<b>Descriptive</b>
<code>int totalScore</code>	<b>Good</b>	<b>Descriptive</b>
<code>int monstersKilled</code>	<b>Good</b>	<b>Descriptive</b>
<code>int x, y</code>	<b>Either</b>	<b>Okay only in trivial mathematical functions</b>

- \* Note: it is okay to use trivial variable names for variables that have a trivial use, such as loop variables. We will address this topic in the section on flow control.



3. Baris komentar dapat digunakan untuk membantu mendeskripsikan suatu identifier.

Daripada memberi nama sebuah variabel ***numberOfCharsIncludingWhitespaceAndPunctuation***, lebih baik memberi nama yang singkat, dengan bantuan baris komentar untuk menjelaskan variabel tersebut, contoh :

```
// holds number of chars in a piece of text --  
//      including whitespace and punctuation!  
int numberOfChars;
```

# Hungarian notation

- Ada cara yang terkadang digunakan untuk menamai suatu variabel, yaitu Hungarian notation. Pada Hungarian notation, nama variabel selalu dimulai dengan suatu prefix yang mengindikasikan tipe dari variabel tersebut.

```
int nValue;    // the n before Value represents that this is an integer
bool bValue;   // b means boolean
char chValue;  // ch means char
double dValue; // d means double
float fValue;  // f means float
```

## ■ Valid

- Nomor
- kd\_barang
- kodeBarang
- \$barang
- bahASA\_PEmrogramaN
- jmlBarang

## ■ Invalid

- 1\_harga - tidak boleh diawali dengan angka
- %barang - simbol % tidak boleh digunakan
- Nama pelanggan - tidak boleh menggunakan spasi
- Kd+barang - tidak boleh menggunakan tanda operator

# Case Sensitive

- Huruf kecil dan huruf kapital pada identifier tidak dianggap sama
- Identifier alamat, Alamat, dan ALAMAT menyatakan tiga identifier yang berbeda.
- Contoh :

```
#include <iostream>
#include <conio.h>
using namespace std;

void main ()
{
    char nama[4] = "Evi";
    char Nama[4] = "Eva";
    char NAMA[4] = "Evo";

    cout<<"String yang dicetak : "<<Nama;
    _getch();
}
```

# Tipe data dan variabel

- Tipe data digunakan untuk mendefinisikan objek data yang akan dimanipulasi dalam sebuah program.
- Objek data yang akan dimanipulasi disebut variabel atau konstanta
- Variabel adalah objek data yang nilainya dapat diubah-ubah, dan konstanta adalah objek data yang nilainya tidak dapat diubah dan hanya dapat digunakan atau diakses.

# Variabel

- → lokasi penyimpanan yang ada di memori
- Setiap variabel memiliki kemampuan menyimpan suatu informasi sesuai dengan tipe data yang dideklarasikan untuk variabel tersebut.
  - tipe-data nama-variabel;
  - tipe-data nama-variabel = <pengisian nilai variabel>;

Contoh :

```
■ int counter;
```

```
■ int counter = 3;
```

```
■ int panjang, lebar, tinggi;
```

```
■ int panjang = 10, lebar = 5, tinggi = 1;
```

```
■ int panjang, lebar, tinggi = 1; // hanya var tinggi yg diberikan nilai
```

# Type data dasar pada C++

Type data	Ukuran memori	Jangkauan nilai	Jumlah digit presisi
char	1 byte	-128 s/d +127	
short int	2 byte		
int	2 byte	-32768 s/d +32767	
long	4 byte	-2.147.438.648 s/d 2.147.438.647	
float	4 byte	$3.4 \times 10^{-38}$ s/d $3.4 \times 10^{38}$	6-7
double	8 byte	$1.7 \times 10^{-308}$ s/d $1.7 \times 10^{308}$	15-16
long double	10 byte	$3.4 \times 10^{-4932}$ s/d $1.1 \times 10^{4932}$	19

# Type data dasar pada C++

Type	Length	Range	
unsigned char	8 bits(1 byte)	0	to 255
char	8 bits	-128	to 127
enum	16 bits	-32,768	to 32,767
unsigned int	16 bits	0	to 65,535
short int	16 bits	-32,768	to 32,767
int	16 bits	-32,768	to 32,767
unsigned long	32 bits	0	to 4,294,967,295
long	32 bits	-2,147,483,648	to 2,147,483,647
float	32 bits	$3.4 \times 10^{-38}$	to $3.4 \times 10^{+38}$
double	64 bits	$1.7 \times 10^{-308}$	to $1.7 \times 10^{+308}$
long double	80 bits	$3.4 \times 10^{-4932}$	to $1.1 \times 10^{+4932}$



# short, int, long int

- bilangan bulat
- tidak mengandung titik desimal
- deklarasi variabel bertipe int pada C++:

```
int var;
```

```
int monsterKilled;
```

```
int totalScore;
```

```
long hargaBarang;
```

# float, double, long double

- deklarasi variabel pada C++:  
`float var;`  
`double var;`  
`long double var;`
- Nilai bilangan pecahan dapat dinyatakan dalam berbagai bentuk :  
`3.14159 // 3.14159`  
`6.02e23 // 6.02 x 1023`  
`1.6e-19 // 1.6 x 10-19`  
`3.0 // 3.0`

# boolean

- Variabel dengan tipe data ini hanya memiliki 2 nilai, yaitu true (1) dan false (0).
- Untuk mendeklarasikan tipe data boolean, gunakan keyword *bool*.

```
bool statusAktif = true;
```

# string

- String merupakan kumpulan dari karakter-karakter. Salah satu cara mendefinisikan string adalah menggunakan tipe terstruktur array dengan tipe char :

```
char var[banyaknyakarakter];
```

(lih. Slide 23)

# Signed and Unsigned

**Signed** → bilangan positif dan negatif  
**Unsigned** → bilangan positif

- Contoh pendeklarasian :
  - Signed int jmlPoint;
  - Unsigned int umurKaryawan;
- Bila tidak didefinisikan, maka secara default akan bernilai signed
  - Int jml; // serupa dengan signed int jml;

# Lebih lanjut tentang Char

- Char mendeklarasikan tipe data karakter.
- Karakter adalah sembarang huruf, angka, tanda baca tunggal.

- `char letter;` ←
- `char letter = 'A';`

bentuk pendeklarasian tersebut hanya bisa memuat satu karakter (+ karakter NULL)

Bila hanya satu karakter maka gunakan tanda petik satu ‘

- `char nama[10];`
- `char nama[10] = "Palembang";`

# Lebih lanjut tentang Char

- Variabel string tidak lain adalah array bertipe char yang mengandung element berupa karakter NULL yang menyatakan akhir dari string.
- `char nama[10];` merupakan string dengan panjang maksimal 9 karakter, hal ini disebabkan karakter NULL akan ditambahkan pada akhir string.
- `char nama[10] = "Palembang";`

P	A	L	E	M	B	A	N	G	\0
---	---	---	---	---	---	---	---	---	----



karakter ber-ASCII 0  
sebagai pengakhir string

Karakter NULL merupakan ciri khas pengakhir suatu string pada C++.

# String

- Cara 1 : menggunakan tipe terstruktur array dengan tipe char yaitu `char var[banyaknyakarakter];`
- Cara 2 :
  - Definisikan `#include<string>`
  - `string namaMahasiswa;`
  - `cout<<"Input string : ";`
  - `getline(cin, namaMahasiswa);`
  - `cout<<namaMahasiswa;`

# Menentukan Tipe Variabel

- Jika variabel digunakan untuk menyimpan data **bilangan bulat** saja, maka pilihannya adalah tipe bilangan bulat (seperti **int**, **long**).
- Jika variabel hendak dipakai untuk menampung data bilangan pecahan, maka variabel harus didefinisikan bertipe bilangan pecahan (seperti **float**, **double**)



# Menentukan Tipe Variabel

Long atau int?

- Penentuannya didasarkan oleh kemungkinan besarnya nilai yang akan ditampung variabel.
- Misalnya bila dipastikan nilai yang akan ditampung tidak akan melebihi 32,767, tetapi juga tidak kurang dari -32,768 maka pilihannya adalah tipe int

# Menentukan Tipe Variabel

- Pemrosesan aritmatika yang menggunakan tipe seperti int akan **lebih cepat** dibandingkan jika menggunakan tipe long.
- Jika suatu variabel diisi dengan nilai diluar jangkauannya, maka nilai yang tersimpan akan diubah sesuai jangkauannya. Misalnya variabel bertipe int diberikan nilai 75000, maka yang akan tersimpan adalah 9464. sebab nilai positif terbesar pada tipe int adalah 32767. hal ini bekerja sebagaimana speedometer, apabila nilai maksimumnya terlampaui maka akan dimulai dari nilai terendahnya lagi.

Berikan keterangan apakah identifier berikut ini valid atau tidak valid :

- avanZA
- from-to
- \$camry
- v10s
- \_avaLoN
- Roger7
- Good/bad
- 4ever
- NiCKnAMe%
- A valid one
- Valid?
- (yes)
- pemrogramanBerorientasiObjek
- netto|bruto
- NAMA
- PT-1
- #brg
- Nama barang
- 2semester