



**Kampus
Merdeka**
INDONESIA JAYA

ALGORITMA DAN STRUKTUR DATA

Jemy Arieswanto



MATERI

1. Variable
2. Konstanta
3. Tipe data
4. Assignment Statement
5. Aritmetic Expression dalam Assignment Statement

1. Variabel

Salah satu hal yang paling dasar dalam pemrograman adalah variabel, jika diilustrasikan sebuah variabel seperti sebuah kotak kecil, dimana kita dapat menyimpan barang-barang dalam kotak tersebut untuk digunakan nanti. Konsep variabel diambil dari matematika seperti :

$x = 1$ → menyimpan nilai 1 dalam variabel x

1. Variabel

Dalam pemrograman variabel adalah lokasi atau area atau tempat di dalam memori yang dapat menyimpan data sementara dalam suatu program, dan data tersebut dapat diubah, disimpan atau ditampilkan kapanpun dibutuhkan.

Setiap variabel harus diberi nama, dan nama variabel harus berbeda antara satu variabel dengan variabel yang lain. Masing-masing variabel memiliki alamat sendiri didalam memori komputer, kita cukup menyebutkan nama variabel dimana data di simpan, maka komputer akan dapat menemukan alamat variabel tempat data tersebut tersimpan pada memori.

1. Variabel

Contoh :

Misal terdapat 3 buah variabel yang di beri nama **X**, **Y** dan **Z**, jika dilustraskan didalam memori komputer digambarkan sebagai berikut :

Memori



X



Y



Z

Terlihat dari ilustrasi gambar diatas, variabel **X**, **Y** dan **Z** akan menempati lokasi atau area tertentu didalam memori komputer. Dimana variabel tersebut digunakan untuk menampung nilai yang akan digunakan dalam program.

1. Variabel

Pemberian Nama Variabel

Pemberian nama variabel ditentukan oleh pembuat program sendiri, namun dalam pemberian nama variabel terdapat syarat-syarat seperti berikut :

- 1) Nama variabel tidak boleh sama dengan nama *keyword* dan *function*.
- 2) Nama variabel maksimum 32 karakter.
- 3) Nama variabel harus diawali dengan huruf atau garis bawah (*underscore* _),
karakter berikutnya boleh angka, huruf atau garis bawah
- 4) Nama variabel tidak boleh ada spasi.

1. Variabel

Keyword Dalam ANSI Bahasa C++/C

auto	float	friend	inline
break	for	delete	new
case	go to	class	operator
char	if	asm	private
const	int	while	protected
continue	long	volatile	public
default	register	void	template
do	return	unsigned	this
double	short	union	virtual
else	signed	typedef	struct
enum	size of	switch	static
extern			

1. Variabel

Contoh pemberian nama variabel

_ Contoh Penamaan Variabel

Penamaan Yang benar	Penamaan Yang salah	Keterangan
X	1X	Awalnya bukan huruf
X1		atau garis bawah
luas	luas-1	Mengandung tanda minus
LUAS	Keliling Lingkaran	(-)
KelilingLingkaran	benar/salah	Mengandung spasi
Keliling_Lingkaran	float	Mengandung spesial karakter
KL	switch	
_panjang	keliling-lingkaran	Sama dengan <i>keyword</i>
FLOAT		Sama dengan <i>keyword</i>
		Mengandung tanda minus (-)



Dalam bahasa C++/C penamaan variabel berbeda antara huruf besar dan huruf kecil (*case sensitif*), variabel **nilai** berbeda dengan **NILAI** berbeda dengan **Nilai**, **FLOAT** berbeda dengan **float** yang merupakan *keyword*.

2. Konstanta

Konstanta merupakan nilai numerik/angka atau karakter yang tetap.

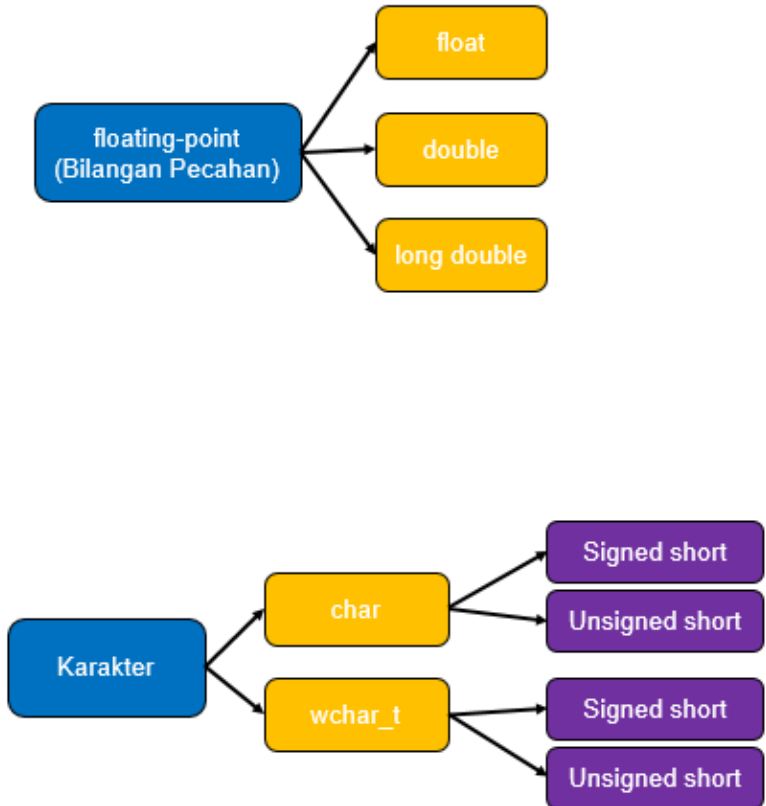
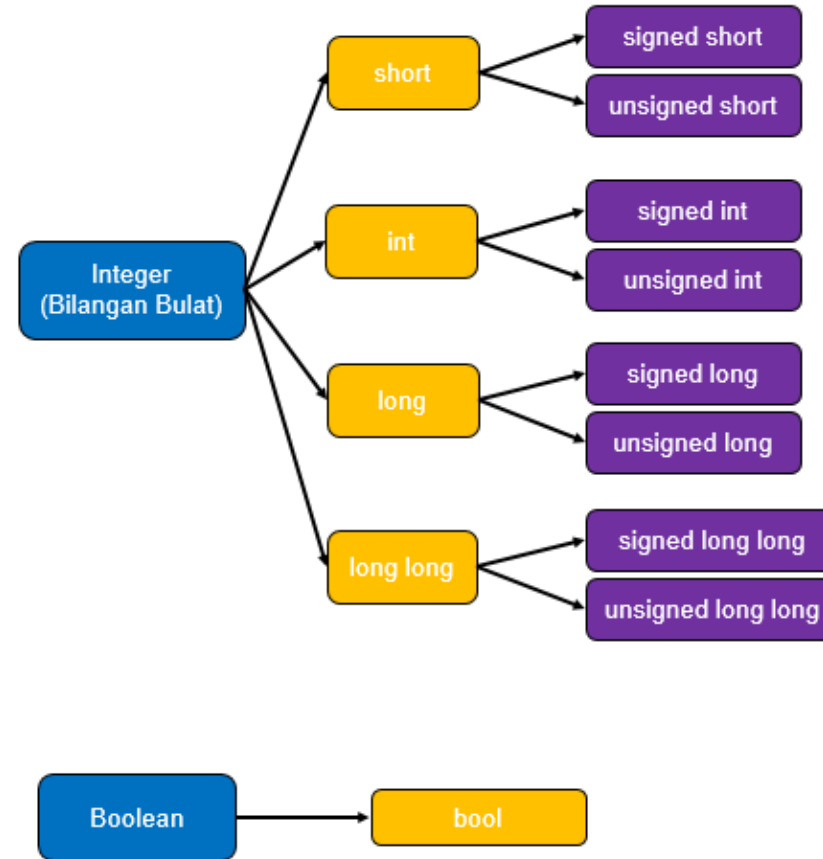
Seperti nilai **PI** yaitu **22/7** atau **3.14159** merupakan nilai konstanta, nilai yang tidak dapat diubah atau nilainya tetap. Untuk mendeklarasikan konstanta menggunakan keyword ***const***.

```
const double PI = 3.14159;
```

Dengan menggunakan keyword ***const***, nilai dari variabel **PI** tidak bisa diubah setelah dideklarasikan.

3. Tipe Data

Tipe data merupakan jenis-jenis data yang dikategorikan berdasarkan sifat dan jenisnya, gambar berikut memperlihatkan kategori tipe data dasar :



3. Tipe Data

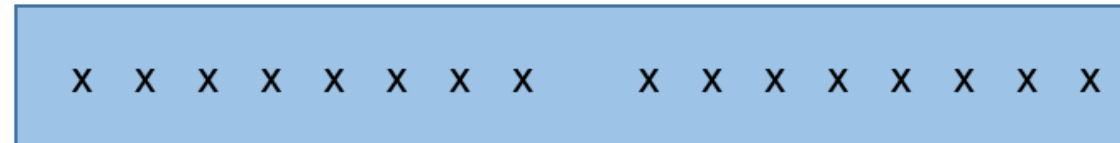
Tipe Data *numeric integer (bilangan bulat)*

Integer adalah jenis tipe data untuk menyimpan nilai numerik/angka dalam bentuk bilangan bulat, yaitu bilangan yang tidak mengandung angka dibelakang koma. Sebagai contoh 4, -17, 50 dan 100 adalah bilangan bulat. Bilangan bulat bisa positif(+), negatif(-) atau 0, tipe data bilangan bulat terbagi menjadi :

3. Tipe Data

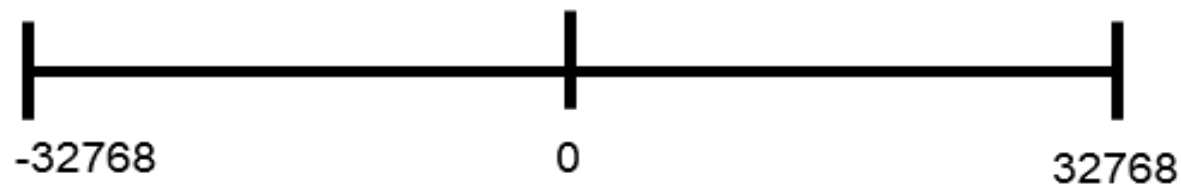
Short

Tipe data short dalam memori komputer menempati area 2 byte (16bit), dengan ilustrasi sebagai berikut :



Ilustrasi bit-bit short integer (16 bit)

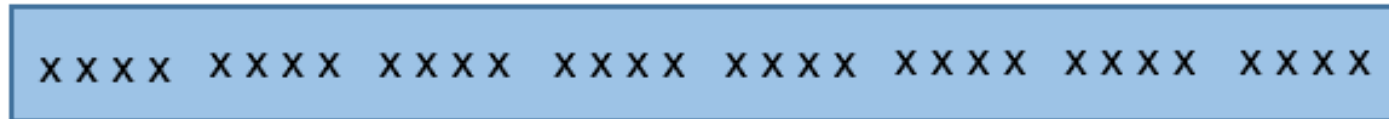
Sedangkan nilai yang dapat ditampung tipe data **short berkisar** -32768 sampai dengan 32768 dengan ilustrasi sebagai berikut :



3. Tipe Data

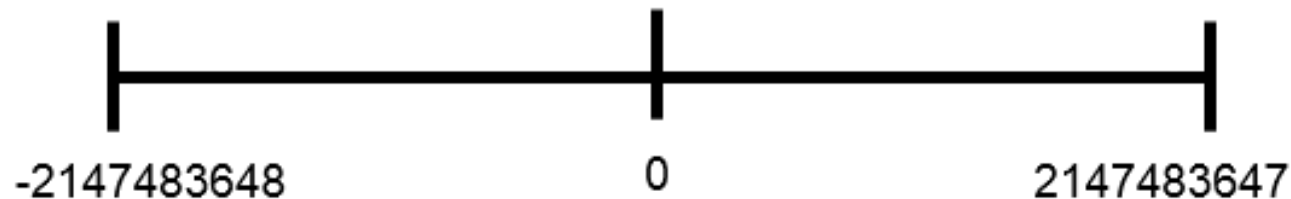
Int

Tipe data **int** dalam memori komputer menempati area 4 byte (32bit), dengan ilustrasi sebagai berikut :



Ilustrasi bit-bit interger (32 bit)

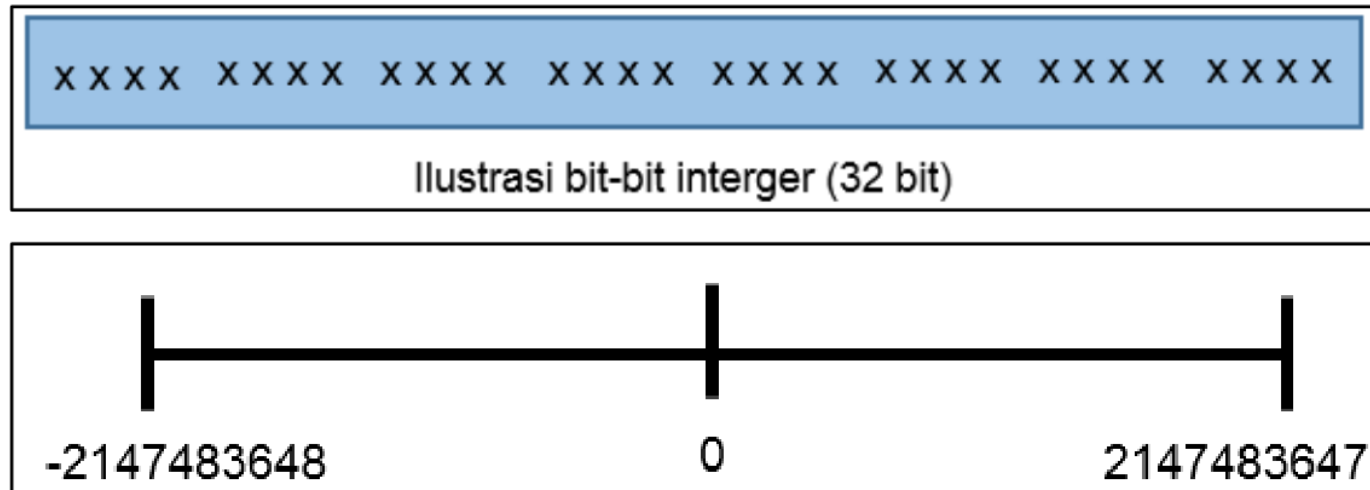
Nilai yang dapat ditampung dalam tipe data **int** berkisar -2147483648 sampai dengan 2147483647



3. Tipe Data

Long

Tipe data **long** dalam memori komputer menempati area 4 byte (32bit) sama seperti tipe data **int**, jangkauan nilai yang dapat ditampung juga sama seperti tipe data **int**.



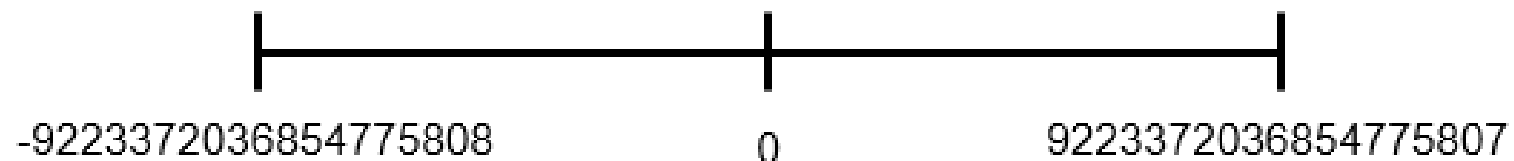
3. Tipe Data long long

Tipe data **long long** dalam memori komputer menempati area 8 byte (64bit), dengan ilustrasi sebagai berikut :

XXXXXXXXXXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX XXXX

Ilustrasi bit-bit long long integer (64bit)

Nilai yang dapat di tampung antara -9223372036854775808 sampai dengan 9223372036854775807.



3. Tipe Data

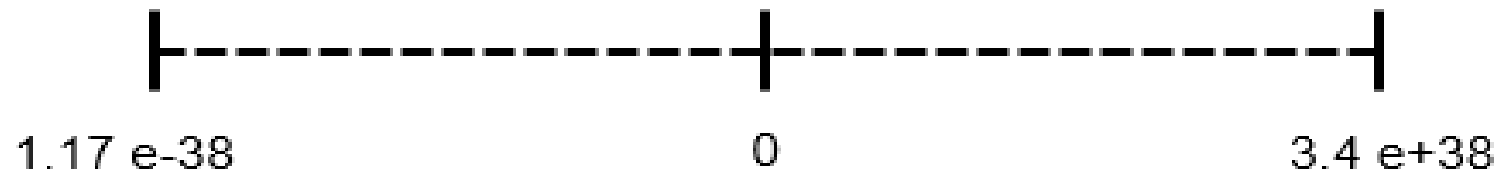
Tipe Data Floating-Point (Pecahan)

Banyak perhitungan yang dilakukan oleh komputer membutuhkan angka yang memiliki bagian fraksional atau angka yang memiliki nilai dibelakang koma (bilangan desimal). Misalnya, ketika menghitung luas lingkaran perlu melibatkan nilai PI dengan nilai berkisar 3.14159. Bahasa pemrograman C++ mendukung angka-angka non-integer(bukan bilangan bulat) seperti nilai PI, bilangan non-integer disebut dengan *floating-point*.

3. Tipe Data

Float (*floating-point single precision*)

kata kunci atau keyword yang digunakan untuk mendeklarasikan *floating-point single precision* adalah **float**. Tipe data **float** disimpan dalam area 4 byte (32bit) dan memiliki ketelitian sampai dengan 6 *digit* dibelakang koma, dengan jangkauan nilai ditampilkan pada gambar berikut



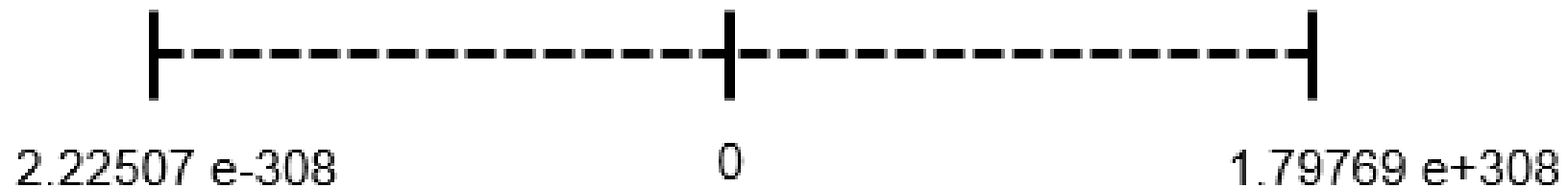
1.35 e-02 artinya $\rightarrow 1.35 * 10^{-2} = 1.35 * 0.01 = 0.0135$

1.35 e+02 artinya $\rightarrow 1.35 * 10^2 = 1.35 * 100 = 135$

3. Tipe Data

Double (*floating-point double precision*)

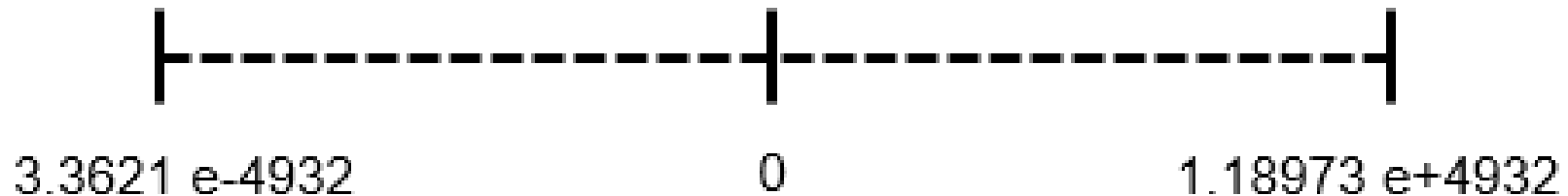
kata kunci atau *keyword* yang digunakan untuk mendeklarasikan *floating-point double precision* adalah **double**. *Tipe data double* disimpan dalam area 8 byte (64bit) dan memiliki ketelitian hingga 15 digit, dengan jangkauan nilai ditampilkan pada gambar berikut :



3. Tipe Data

Long double (floating-point double precision)

kata kunci atau *keyword* yang digunakan untuk mendeklarasikan *floating-point double precision* adalah ***long double***. Tipe data ***long double*** disimpan dalam area 12 byte (96bit), dengan jangkauan nilai ditampilkan pada gambar berikut :



3. Tipe Data

Tipe Data Karakter

Jenis tipe data **karakter** digunakan untuk menyimpan **kode** karakter. Kode karakter merupakan suatu bilangan bulat yang terkait karakter yang diwakilkan. Misalnya, huruf **A** diwakili oleh kode 65 dalam kode **ASCII** (*American Standard Code for Information Interchange*).

3. Tipe Data

ASCII table

Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex	Char	Dec	Oct	Hex
(nul)	0	0000	0x00	(sp)	32	0040	0x20	@	64	0100	0x40	`	96	0140	0x60
(soh)	1	0001	0x01	!	33	0041	0x21	A	65	0101	0x41	a	97	0141	0x61
(stx)	2	0002	0x02	"	34	0042	0x22	B	66	0102	0x42	b	98	0142	0x62
(etx)	3	0003	0x03	#	35	0043	0x23	C	67	0103	0x43	c	99	0143	0x63
(eot)	4	0004	0x04	\$	36	0044	0x24	D	68	0104	0x44	d	100	0144	0x64
(enq)	5	0005	0x05	%	37	0045	0x25	E	69	0105	0x45	e	101	0145	0x65
(ack)	6	0006	0x06	&	38	0046	0x26	F	70	0106	0x46	f	102	0146	0x66
(bel)	7	0007	0x07	'	39	0047	0x27	G	71	0107	0x47	g	103	0147	0x67
(bs)	8	0010	0x08	(40	0050	0x28	H	72	0110	0x48	h	104	0150	0x68
(ht)	9	0011	0x09)	41	0051	0x29	I	73	0111	0x49	i	105	0151	0x69
(nl)	10	0012	0x0a	*	42	0052	0x2a	J	74	0112	0x4a	j	106	0152	0x6a
(vt)	11	0013	0x0b	+	43	0053	0x2b	K	75	0113	0x4b	k	107	0153	0x6b
(np)	12	0014	0x0c	,	44	0054	0x2c	L	76	0114	0x4c	l	108	0154	0x6c
(cr)	13	0015	0x0d	-	45	0055	0x2d	M	77	0115	0x4d	m	109	0155	0x6d
(so)	14	0016	0x0e	.	46	0056	0x2e	N	78	0116	0x4e	n	110	0156	0x6e
(si)	15	0017	0x0f	/	47	0057	0x2f	O	79	0117	0x4f	o	111	0157	0x6f
(dle)	16	0020	0x10	0	48	0060	0x30	P	80	0120	0x50	p	112	0160	0x70
(dc1)	17	0021	0x11	1	49	0061	0x31	Q	81	0121	0x51	q	113	0161	0x71
(dc2)	18	0022	0x12	2	50	0062	0x32	R	82	0122	0x52	r	114	0162	0x72
(dc3)	19	0023	0x13	3	51	0063	0x33	S	83	0123	0x53	s	115	0163	0x73
(dc4)	20	0024	0x14	4	52	0064	0x34	T	84	0124	0x54	t	116	0164	0x74
(nak)	21	0025	0x15	5	53	0065	0x35	U	85	0125	0x55	u	117	0165	0x75
(syn)	22	0026	0x16	6	54	0066	0x36	V	86	0126	0x56	v	118	0166	0x76
(etb)	23	0027	0x17	7	55	0067	0x37	W	87	0127	0x57	w	119	0167	0x77
(can)	24	0030	0x18	8	56	0070	0x38	X	88	0130	0x58	x	120	0170	0x78
(em)	25	0031	0x19	9	57	0071	0x39	Y	89	0131	0x59	y	121	0171	0x79
(sub)	26	0032	0x1a	:	58	0072	0x3a	Z	90	0132	0x5a	z	122	0172	0x7a
(esc)	27	0033	0x1b	;	59	0073	0x3b	[91	0133	0x5b	{	123	0173	0x7b
(fs)	28	0034	0x1c	<	60	0074	0x3c	\	92	0134	0x5c		124	0174	0x7c
(gs)	29	0035	0x1d	=	61	0075	0x3d]	93	0135	0x5d	}	125	0175	0x7d
(rs)	30	0036	0x1e	>	62	0076	0x3e	^	94	0136	0x5e	~	126	0176	0x7e
(us)	31	0037	0x1f	?	63	0077	0x3f	_	95	0137	0x5f	(del)	127	0177	0x7f

3. Tipe Data

Tipe data karakter terbagi menjadi 2, yaitu **char** dan **wchar_t** (*wide character type*) :

1) Char

Tipe data **char** digunakan untuk menyatakan karakter tunggal dalam 1 byte (8 bit) seperti huruf alfabet (baik huruf besar maupun huruf kecil), angka, tanda baca, dan karakter kontrol (seperti *newline* dan *tab*). Dalam bahasa pemrograman c++ **karakter diapit oleh** kutip tunggal('), seperti:

'A'

3. Tipe Data

2) **wchar_t**

Tipe data **wchar_t** (*wide character type*) *terdiri dari* setidaknya 2 byte (16 bit) dan dengan demikian mampu menyimpan karakter **Unicode modern**. **Unicode** adalah kode 16-bit yang juga digunakan dalam Windows NT dan berisi kode untuk sekitar 35.000 karakter dalam 24 bahasa. Tipe data **wchar_t** **biasa digunakan** untuk penggunaan karakter yang besar seperti bahasa china.

3. Tipe Data

Tipe Data Boolean

Hasil perbandingan operator logika **AND** atau **OR** akan menghasilkan nilai **boolean** yaitu **benar** (*true*) atau **salah** (*false*). *Dalam* bahasa pemrograman C++ tipe **boolean** digunakan untuk merepresentasikan nilai **boolean**. Ekspresi nilai boolean akan direpresentasikan sebagai angka 1 jika bernilai **benar** (*true*), dan angka 0 jika bernilai **salah** (*false*).

```
bool benar = true → 1 (satu)
bool salah = false → 0 (nol)
```


3. Tipe Data

Tipe Data Pada C++

Tipe data	Penulisan dengan C++	Jumlah byte	Jangkauan nilai yang dapat ditampung
Character	char atau signed char	1	-128 sampai dengan 127
	unsigned char	1	0 sampai dengan 255
	wchar_t	2	
Integer	signed short atau short	2	-32768 samapi dengan 32768
	Unsigned short	2	0 sampai dengan 65535
	int atau signed int	4	-2147483648 sampai dengan 2147483647
	unsigned int	4	0 sampai dengan 4294967295
	long atau signed long atau signed long int	4	-2147483648 sampai dengan 2147483647

3. Tipe Data

Tipe data	Penulisan dengan C++	Jumlah byte	Jangkauan nilai yang dapat ditampung
	unsigned long atau unsigned long int	4	0 s.d 4294967295
	long long int	8	-9223372036854775808 s.d 9223372036854775807
	unsigned long long int	8	0 sampai dengan 1844644073709551615
Floating point single precision	Float	4	1.17e-38 sampai dengan 3.4e+38
Floating point double precision	double	8	2.2e-308 sampai dengan 1.7e+308
	long double	12	3.4e-4932 sampai dengan 1.18e+4932

3. Tipe Data

Deklarasi Variabel dengan Tipe Data

Sebelum variabel digunakan dalam program, variabel harus dideklarasikan terlebih dahulu agar program dapat mengenali variabel yang akan digunakan dalam program. Deklarasi variabel tidak terlepas dari tipe data yang digunakan, cara mendeklarasikan variabel seperti berikut:

tipedata <spasi> *namavariabel*

Contoh :

```
int nilai  
long long int _investasi  
float rata2  
double keliling  
unsigned int nilai  
char A
```

3. Tipe Data

deklarasi variabel dapat dilakukan dengan beberapa variabel yang dipisahkan dengan koma, seperti berikut:

tipedata <spasi> namavariabel1, namavaribel2,..., namavariabeln

3. Tipe Data

Contoh :

```
int nilai1, nilai2  
long long int _investasi, _HS  
float rata2, nilai_akhir  
double keliling, luas  
unsigned int nilai1, nilai2, nilai3  
char A, B, C
```

Contoh-Contoh penggunaan Variabel dan Tipe data dengan bahasa pemrograman C++

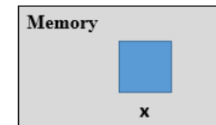
3. Tipe Data

```
1 #include<iostream>
2 using namespace std;
3 int main() {
4     int x; //deklarasi variabel x dengan tipe data int
5     x = 15; //mengisi nilai variabel x dengan nilai 5
6     cout<<x; //mencetak isi variabel x
7
8     return 0;
9 }
```

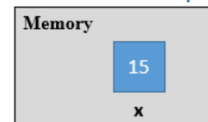
Output : 15



int x; → artinya mendeklarasikan atau menyiapkan sebuah variabel dengan nama **x** bertipe data integer(bilangan bulat).



x = 15; → artinya variabel yang telah dideklarasikan atau disiapkan di isi dengan nilai 15.



cout → dibaca si-out keyword yang berfungsi untuk mencetak.

cout<<x; → artinya mencetak nilai yang tersimpan dalam variabel x yaitu 15.

4. Assignment Statement

Assignment statement atau biasa di terjemahkan menjadi **pernyataan penugasan**, yang di maksud di sini adalah mengisi sebuah variabel dengan suatu nilai. Fomat penulisan atau penggunaan assignment statement adalah sebagai berikut:

```
variabel = ekspresi;
```

Dalam **assignment statement / pernyataan penugasan** nilai ekspresi harus cocok atau sesuai dengan tipe data variabel. Ekpresi di sisi kanan akan dievaluasi dan nilainya akan di berikan ke dalam variabel yang ada pada sisi kiri.

4. Assignment Statement

```
int X;
```

```
X = 8
```

X = 8 → artinya mengisi variabel **X** dengan nilai 8 (**X** diisi 8)

Atau nilai **X** dibuat sama dengan 8.

Ilustrasi digambarkan sebagai berikut:



X

Sebelum intruksi **X=8;** dilaksanakan, variabel **X** sebenarnya sudah ada isinya, tetapi kita tidak mengetahui isinya.



X

Setelah intruksi **X = 8;** dilaksanakan maka isi dari variabel **X** menjadi 8.

4. Assignment Statement

X = Y	Mengisi variabel X dengan isi dari variabel Y . Dimana isi variabel Y tidak berubah.
<pre>int X = 8, Y = 10; X = Y; cout<< X; cout<< Y;</pre>	Tercetak: 10 10
<div><div>NOTE</div><p>Ilustrasi program di atas: sebelum intruksi X = Y dilaksanakan, misal X = 8, Y = 10.</p><div><div>8</div><div>10</div><div>X</div><div>Y</div></div><p>setelah intruksi X = Y dijalankan, maka isi dari variabel X menjadi 10, dan isi dari variabel Y tetap 10 (tidak berubah).</p><div><div>10</div><div>10</div><div>X</div><div>Y</div></div></div>	

4. Assignment Statement

$Z = X + Y;$	Mengisi variabel Z dengan hasil penjumlahan dari variabel X dan Y .
<pre>#include<iostream> main(){ int Z=6, X=3, Y=5; Z = X + Y; }</pre>	<p>Misal nilai variabel Z, X dan Y sebelum intruksi dilaksanakan digambarkan sebagai berikut:</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"><div style="background-color: #4a86e8; color: white; padding: 10px; width: 40px;">6</div>Z</div> <div style="text-align: center;"><div style="background-color: #4a86e8; color: white; padding: 10px; width: 40px;">3</div>X</div> <div style="text-align: center;"><div style="background-color: #4a86e8; color: white; padding: 10px; width: 40px;">5</div>Y</div> </div> <p>Setelah intruksi Z = X+Y dilaksanakan, nilai Z berubah dengan</p> <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"><div style="background-color: #4a86e8; color: white; padding: 10px; width: 40px;">8</div>Z</div> <div style="text-align: center;"><div style="background-color: #4a86e8; color: white; padding: 10px; width: 40px;">3</div>X</div> <div style="text-align: center;"><div style="background-color: #4a86e8; color: white; padding: 10px; width: 40px;">5</div>Y</div> </div>
Tecetak: 8	
<p>Proses pelaksanaan intruksi Z = X+Y dalam komputer:</p> <p>Pertama nilai X dan Y di bawa ke <i>processor</i> (CPU) kemudian dilakukan penambahan nilai X dan Y di dalam CPU. Hasil dari penjumlahan tersebut di isikan ke dalam variabel Z.</p> <p>Isi variabel Z menjadi 8,</p> <p>Sedangkan isi variabel X dan Y tidak berubah.</p>	

4. Assignment Statement

```
X = 4;  
Y = 2;  
X = X + Y;
```

Tercetak: 6

Mengisi variabel **X** dengan hasil penjumlahan nilai **X** dan **Y**.

(**X** di isi dengan nilai **X + Y**)

Sebelum intruksi **X = X + Y** di laksanakan, nilai **X** dan **Y** digambarkan sebagai berikut:



Setelah intruksi **X = X + Y** dilaksanakan, nilai dari variabel **X** berubah menjadi **6**, dan ini dari variabel **Y** tetap **2** (tidak berubah).

4. Assignment Statement

$X = 6;$ $X = X + 1;$	Mengisi variabel X dengan hasil penjumlahan nilai A dengan konstanta 1. (variabel X diisi dengan nilai $A+1$;)
Tercetak: 7	

4. Assignment Statement

1)

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int x = 7;
5      cout<<x+2;
6      cout<<"\n"<<x;
7  }
```

Tercetak : 9
7

2)

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int x = 7;
5      cout << int(x=x+2);
6      cout << x;
7  }
```

Tercetak: 9
9

4. Assignment Statement

3)

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int X = 7, Y = 5;
5      X = Y;
6      cout<<X;
7      cout<<"\n"<<Y;
8  }
```

Tercetak: 5
5

4)

```
1  #include<iostream>
2  using namespace std;
3  int main() {
4      int X = 7, Y = 5;
5      X = Y;
6      Y = X;
7      cout<<X;
8      cout<<"\n"<<Y;
9  }
```

Tercetak: 5
5

4. Assignment Statement

5)

```
#include<iostream>
using namespace std;
int main() {
    int a = 12, b = 75, c=0;
    c = a;
    a = b;
    b = c;

    cout<<"Nilai a = " << a;
    cout<<"\nNilai b = " << b;
    cout<<"\nNilai c = " << c;
}
```

Nilai a = 75

Nilai b = 12

Nilai c = 12

6)

```
#include<iostream>
using namespace std;
int main() {
    int a = 12, b = 3;
    a = a/b;
    b = a*2;

    cout<<"Nilai a = " << a;
    cout<<"\nNilai b = " << b;
}
```

Nilai a = 4

Nilai b = 8

5. Arithmetic Expresion dalam Assignment Statement

Perhatikan assignment statement berikut ini:

```
X = A + B * ( C - D ) ;
```

Disebut aritmatic expression atau pernyataan aritmatik dimana A,B,C dan D disebut operand (yang dioperasikan). Sedangkan +, * ,- dan () disebut aritmatic operator(operator aritmatika).

5. Arithmetic Expresion dalam Assignment Statement

Operator Aritmatika

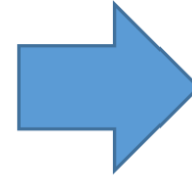
Dalam bahasa C++ dan Java, operator aritmatika yang digunakan adalah sebagai berikut:

Operator	Tingkat Hierarki atau Power
* / %	Kali, Bagi dan Modulus sama kuatnya Kali, bagi dan modulus Lebih didahulukan dari Tambah dan Kurang
+ -	Tambah dan kurang sama derajatnya
()	Yang ada di dalam () adalah satu kesatuan nilai

5. Arithmetic Expresion dalam Assignment Statement

Pembagian

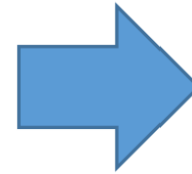
```
int N = 25;  
N = N / 2;  
cout<<N;
```



Tercetak 12

N tipe integer, hanya akan menyimpan bilangan bulat

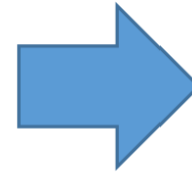
```
float F = 25.0;  
F = F / 2;  
cout<< F;
```



Tercetak 12.500000

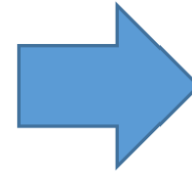
F tipe Float dapat menampung nilai pecahan

```
int N ;  
float F;  
N = 25;  
F = N / 2;  
cout<<F;
```



Tercetak 12.0000000

```
int N ;  
float F;  
N = 25.5;  
F = N / 2;  
cout<<F;
```

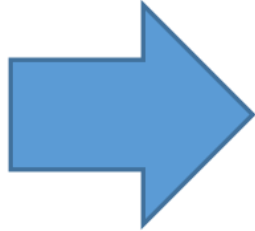


Tercetak 12

5. Arithmetic Expresion dalam Assignment Statement

Modulus(Sisa Pembagian)

```
int N =25 ;  
int K;  
K = N% 2;  
  
cout<<K;
```

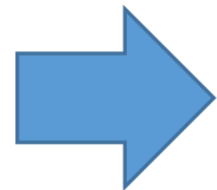


Tercetak 1

Karena 25 dibagi2 = 12
sisanya 1

Operator % biasa disebut modulus hanya berlaku untuk bilangan integer, sehingga % disebut sisa pembagian bilangan integer.

```
int A = 25;  
cout<<A;
```



ERROR

Tipe float tidak dapat
dikenali oleh operator %(
harus menggunakan
fungsi)

5. Arithmetic Expresion dalam Assignment Statement

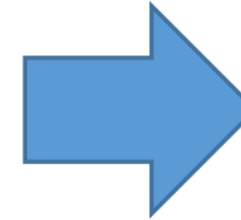
Contoh hasil pembagian modulus:

10 % 2 = 0
10 % 3 = 1
10 % 4 = 2
10 % 5 = 0
8 % 6 = 2
0 % 2 = 0

5. Arithmetic Expresion dalam Assignment Statement

Aritmatika dalam instruksi **cout**

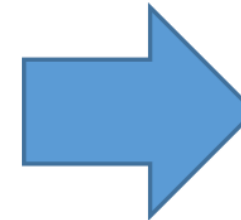
```
int A = 25;  
cout<<A;
```



Akan tercetak : 25

Nilai variable A tetap dan tidak berubah

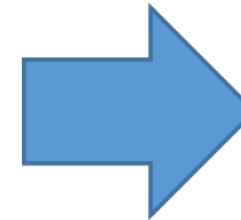
```
int A = 25;  
cout<<A+1;
```



Akan tercetak: 26

Yaitu 25+1. Sedangkan nilai A tetap tidak berubah

```
int A = 25;  
A = A+1;  
cout<< A;
```



Isi variable A ditambah 1 dahulu menjadi 26 kemudian dicetak.

5. Arithmetic Expresion dalam Assignment Statement

Untuk semua instruksi
cout berikut dengan nilai
awal A=25

Int A = 25;	Tercetak	Keterangan
cout << A+1;	26	Nilai A tetap = 25
cout << A+=1;	26	Nilai A ditambah 1 dulu, menjadi 26 kemudian dicetak
cout << A++;	25	Nilai A dicetak dulu, kemudian ditambah 1, sekarang nilai A=26
cout << ++A;	26	Nilai A ditambah 1 dulu, menjadi 26 kemudian dicetak
cout << A--;	25	Nilai A dicetak dulu, kemudian dikurangi 1, sekarang nilai A=24
cout << --A;	24	Nilai A dikurangi 1 dulu, menjadi 24 kemudian dicetak
cout << ++A++;	Error	

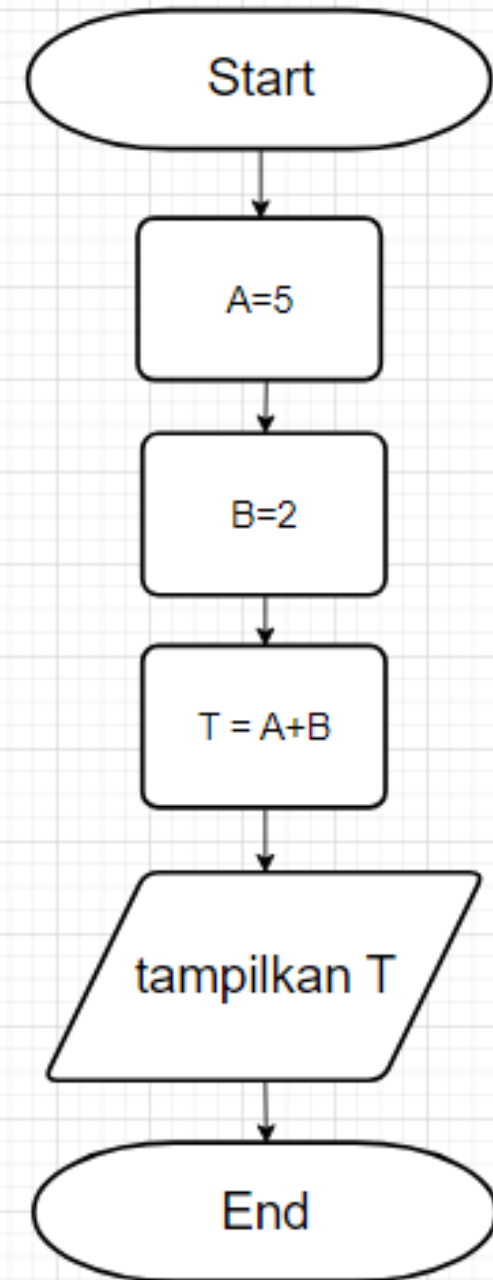
5. Arithmetic Expresion dalam Assignment Statement

Contoh Penulisan Aritmatika Statement

Penulisan dalam amtematika	Penulisan dlam program
$X = A + B / C$	<code>X = A + B / C;</code>
$X = (A + B) / C$	<code>X = (A + B) / C;</code>
$X = \frac{A + B}{C}$	<code>X = (A + B) / C</code>
$X = \frac{A + B}{2C}$	<code>X = (A + B) / (2 * C)</code>
$X = \frac{A^2 - \sqrt{2B}}{4AC}$	<code>X=(A*A – sqrt(2*B))/(4*A*C)</code>

Contoh Source code to Flowchart

```
#include <iostream>
using namespace std;
int main()
{
    int A, B, T;
    A= 5;
    B= 2;
    T = A + B;
    cout<<T;
}
```



Referensi

A.S, R. (2018). *LOGIKA ALGORITMA dan PEMROGRAMAN DASAR*. BANDUNG: MODULA.

Charibaldi, N. (2004). *Modul Kuliah Algoritma Pemrograman II Edisi Kedua*. Yogyakarta.

Davis, S. R. (2014). *C++ For Dummies (7th ed.)*. John Wiley & Sons, Inc.

Deitel, P., & Deitel, H. (2014). *C++ How To Program (9th ed.)*. United State of America: Pearson.

Munir, R. (2005). *Algoritma dan Pemrograman dalam Bahasa Pascal dan C*. Bandung: Penerbit Informatika.

Sjukani, M. (2014). *Algoritma dan Struktur Data 1 dengan C, C++ dan Java (Edisi 9 ed.)*. Jakarta: Mitra Wacana Media.

END OF SLIDE



ALGORITMA & STRUKTUR DATA