

PERTEMUAN 3 VARIABEL, KONSTANTA DAN TIPE DATA

A. Tujuan Pembelajaran

Setelah mempelajari materi ini, mahasiswa mampu :

1. Membedakan jenis-jenis tipe data dasar dalam pemrograman
2. Menggunakan jenis-jenis tipe data dasar dalam pemrograman
3. Memahami penggunaan variabel dan konstanta dalam pemrograman
4. Mendeklarasikan variabel menggunakan jenis-jenis tipe data dasar.

B. Uraian Materi

Sebuah variabel akan mengalokasikan tempat penyimpanan yang mempunyai nama dalam memori komputer. Setiap variabel akan memiliki tipe tertentu yang akan menentukan ukuran dan letak memori, rentang nilai yang dapat disimpan dan operasi yang dapat diterapkan ke variabel tersebut.

1.1 Variabel

a. Definisi

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 :

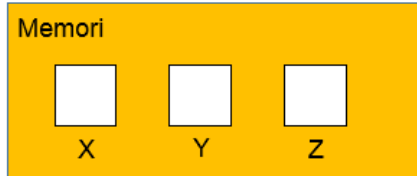
$x = 1 \rightarrow$ menyimpan nilai 1 dalam variabel x

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.

Contoh :

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



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.

b. 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.

Tabel 3. 1. 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			

c. Contoh pemberian nama variabel

Berikut adalah beberapa contoh pemberian nama variabel yang benar, dan contoh pemberian nama variabel yang salah:

Tabel 3. 2. Contoh Penamaan Variabel

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



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*.

1.2 Konstanta

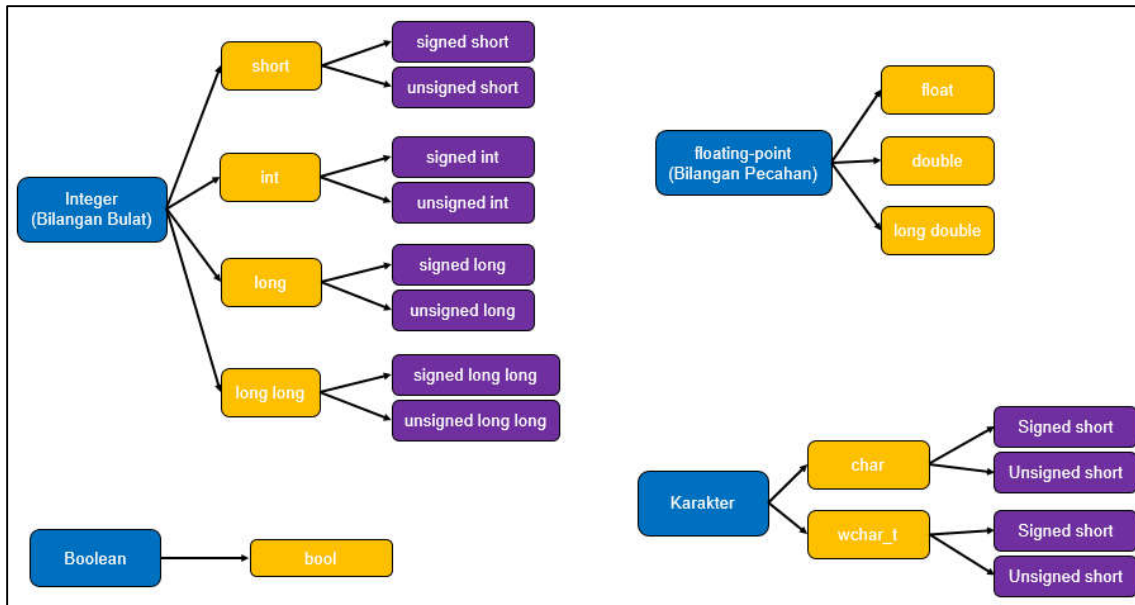
Konstanta merupakan nilai numerik/angka atau karakter yang tetap. Seperti nilai **PI** yaitu 22/7 atau 3.14159 merupakan nilai konstanta, nilai yang tidak dapat diubah atau nilainya tetap. Untuk mendeklarasikan konstanta menggunakan *keyword* **const**.

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

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

1.3 Tipe Data

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



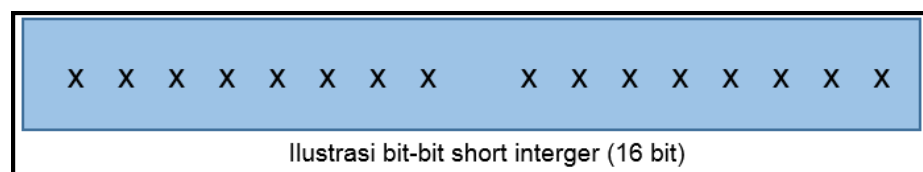
Gambar 3. 1. Kategori Tipe Data Dasar

1.3.1 Tipe Data *numeric integer* (bilangan bulat)

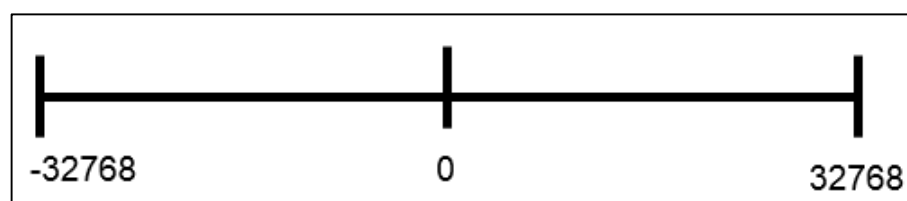
Integer adalah jenis tipe data untuk menyimpan nilai numerik/angka dalam bentuk bilangan bulat, yaitu bilangan yang tidak mengandung angka dibelakang koma. Sebagai contoh 4, -17, 50 dan 100 adalah bilangan bulat. Bilangan bulat bisa positif(+), negatif(-) atau 0, tipe data bilangan bulat terbagi menjadi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

a. Short

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

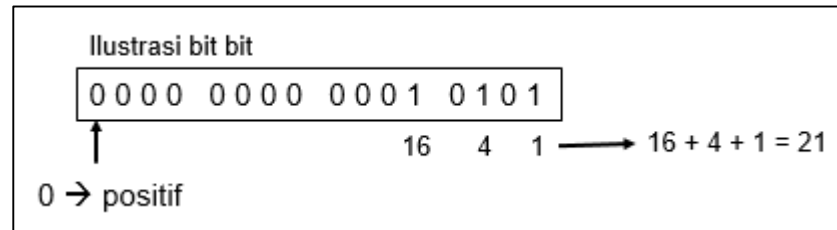


Sedangkan nilai yang dapat ditampung tipe data **short** berkisar -32768 sampai dengan 32768 dengan ilustrasi sebagai berikut :



Gambar 3. 2. Jangkauan Nilai Tipe data short

Tipe data **short** hanya dapat menampung nilai numerik/angka dengan maksimum nilai 32768, sehingga jika ingin menampung nilai yang lebih besar dari 32768 harus menggunakan tipe data yang lain, tipe data yang dapat menampung nilai yang lebih besar.

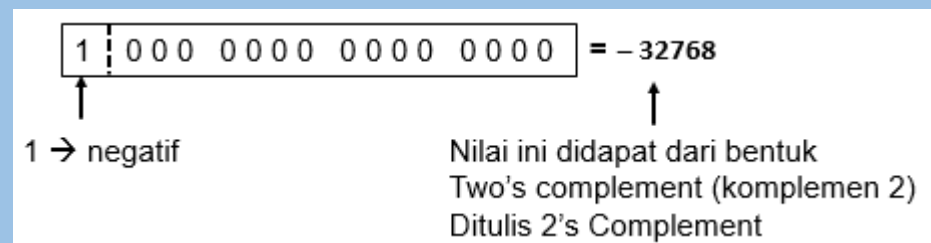


Beberapa contoh ilustrasi bit-bit serta nilai yang tersimpan dalam 2 byte tipe data **short**.

0000 0000 0000 0000 = 0	0111 1111 1111 1111 = 32767
0000 0000 0000 0001 = 1	1000 0000 0000 0000 = -32768
0000 0000 0000 0011 = 3	1000 0000 0000 0001 = -32767
0000 0000 0000 1000 = 8	1111 1111 1111 1111 = -1



Penulisan bit-bit yang dikelompokkan menjadi 4 bit dengan tujuan untuk mempermudah pembacaan bit-bit tersebut.



Cara memperoleh nilai 2's Complement

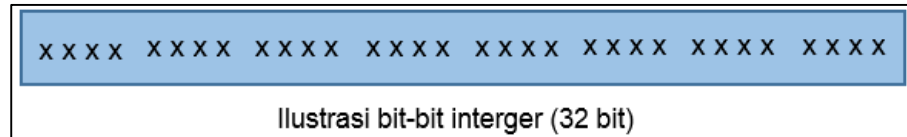
data	1 000 0000 0000 0000
One's complement	0 111 1111 1111 1111
Ditambah 1	<div style="text-align: right;">1 +</div>
2's complement	1 000 0000 0000 0000

One's complement diperoleh dari membalikan nilai bit-bit pada data awal, bit 0 menjadi bit 1 dan sebaliknya.

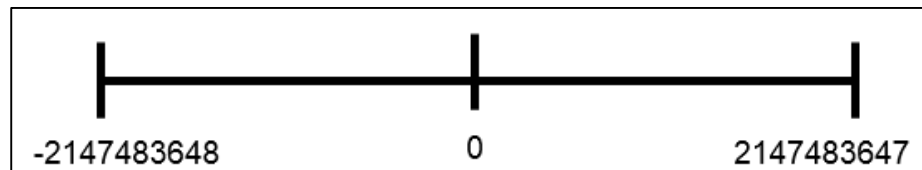
Pada *2's complement* jika dihitung bit-bit yang on, nilainya adalah 32768.

b. Int

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



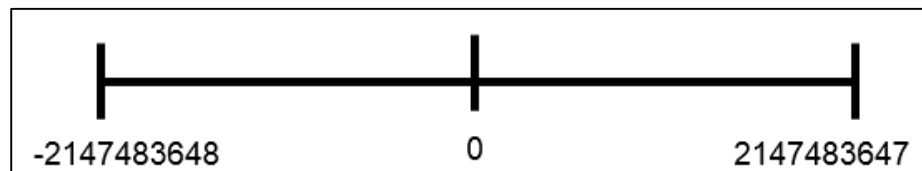
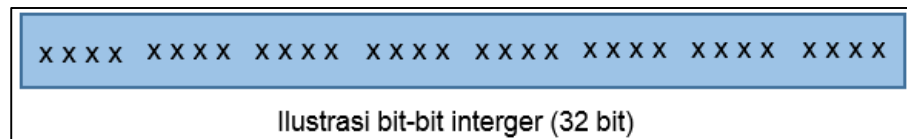
Nilai yang dapat ditampung dalam tipe data **int** berkisar -2147483648 sampai dengan 2147483647



Gambar 3. 3 Jangkauan Nilai Tipe data int

c. Long

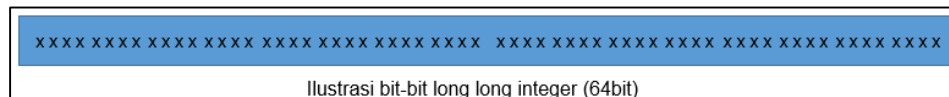
Tipe data **long** dalam memori komputer menempati area 4 byte (32bit) sama seperti tipe data **int**, jangkauan nilai yang dapat ditampung juga sama seperti tipe data **int**.



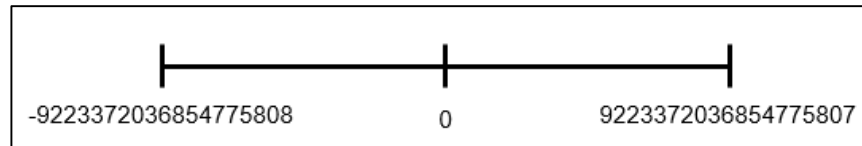
Gambar 3. 4 Jangkauan Nilai Tipe data long

d. long long

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



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



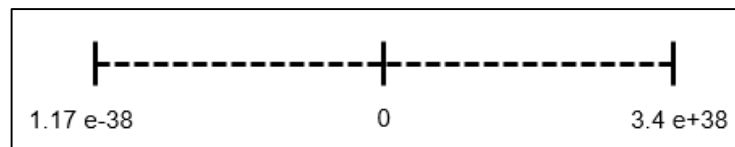
Gambar 3. 5. Jangkauan Nilai Tipe data long long

1.3.2 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-interger disebut dengan *floating-point*.

a. 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 :

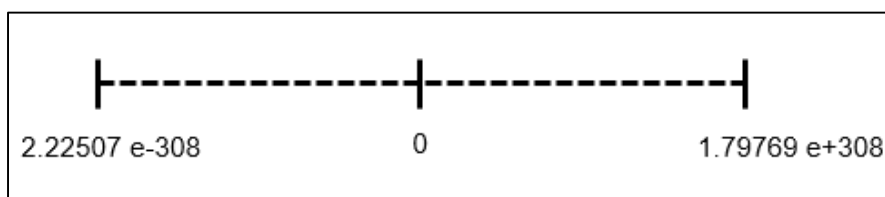


1.35 e-02 artinya $\rightarrow 1.35 * 10^{-2} = 1.35 * 0.01 = 0.0135$

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

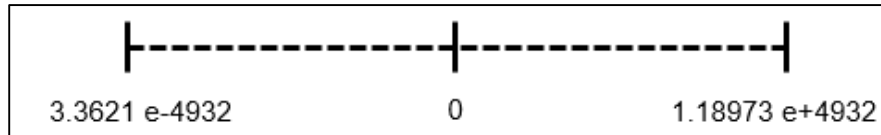
b. Double (*floating-point double precision*)

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 :



c. 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 :



1.3.3 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*).

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

Gambar 3. 6. Kode ASCII

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

a. 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 *new line* dan *tab*). Dalam bahasa pemrograman c++ **karakter** diapit oleh kutip tunggal('), seperti:

```
'A'
```

b. 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.

1.3.4 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)
```

1.3.5 Modifikasi Tipe Data

Terdapat beberapa tipe data modifikasi yang dapat dikenakan terhadap tipe data dasar, seperti **signed** dan **unsigned**. Untuk lebih jelasnya ditunjukkan pada **Tabel 3.3**.

1.3.6 Rangkuman Tipe Data

Tipe data yang digunakan dalam bahasa pemrograman C++ di tampilkan pada tabel berikut :

Tabel 3. 3. 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
	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 18446744073709551615
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

1.4 Deklarasi Variabel dengan Tipe Data

Sebelum variabel digunakan dalam program, variabel harus dideklarasikan terlebih dahulu agar program dapat mengenali variable 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
```

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

tipe data <spasi> namavariabel1, namavariabel2,..., namavariabeln

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++

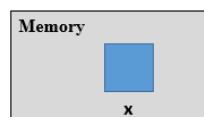
Listing 3.1 : variabel_01.cpp

```
#include<iostream>
using namespace std;
main()
{
    int x; //deklarasi variabel x dengan tipe data int
    x = 15; //mengisi nilai variabel x dengan nilai 5
    cout<<x; //mencetak isi variabel x
}
```

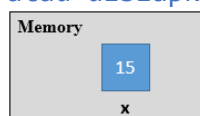
Output : 15



`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.

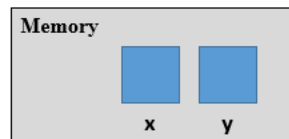
Listing 3.2 : variabel02.cpp

```
#include<iostream>
using namespace std;
main()
{
    int x, y; //deklarasi variabel x dan y dengan tipe data int
    x = 15; //mengisi nilai variabel x dengan nilai 15
    y = 50; //mengisi nilai variabel y dengan nilai 50
    cout<<x; //mencetak isi variabel x
    cout<<y; //mencetak isi variabel y
}
```

Output : 15 50

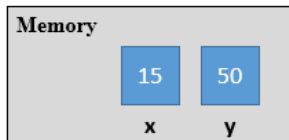


int x, y; → artinya mendeklarasikan atau menyiapkan dua buah variabel dengan nama x dan y bertipe data integer(bilangan bulat).



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

y = 50; → artinya variabel yang telah dideklarasikan atau disiapkan di isi dengan nilai 50



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

cout<<y; → artinya mencetak nilai yang tersimpan dalam variabel y yaitu 50.

Listing 3.3 : variabel03.cpp

```
#include<iostream>
using namespace std;
main()
{
    const double phi = 3.14; //deklarasi variabel phi dengan tipe data int dan konstanta
    cout<<phi; //mencetak isi variabel x
}
```

Output : 3.14



`const double phi = 3.14;` → deklarasi variabel `x` dengan tipe data `double` dengan keyword `const`
`const` → dengan menggunakan keyword `const` artinya nilai yang ada pada variabel `phi` tidak bisa di ubah
`cout<<phi;` → mencetak isi variabel `phi`

Listing 3.4 : variabel04.cpp

```
#include<iostream>
using namespace std;
main()
{
    int nilai_uts, nilai_uas, rata_rata; //deklarasi variabel
    nilai_uts = 80; //mengisi nilai variabel
    nilai_uas = 70; //mengisi nilai variabel
    rata_rata = (nilai_uts+nilai_uas)/2;
    cout<<rata_rata; //mencetak isi variabel x
}
```

Output : 75

Listing 3.5 : variabel05.cpp

```
#include<iostream>
using namespace std;
main()
{
    float f; //deklarasi variabel f dengan tipe data float
    f = 1.5; //mengisi nilai variabel f dengan nilai 1.5
    cout<<f; //mencetak isi variabel f
}
```

Output : 15

Listing 3.5 : variabel06.cpp

```
#include<iostream>
using namespace std;
main()
{
    float f, g, h; //deklarasi variabel f dengan tipe data float
    f = 7; //mengisi nilai variabel f dengan nilai 1.5
    f = 3; //mengisi nilai variabel f dengan nilai 1.5
    h = 7/3;
    cout<<h; //mencetak isi variabel h
}
```

Output : 2.33333

C. Soal Latihan / Tugas

1. Sebutkan jenis-jenis tipe data dasar dalam bahasa pemrograman c++ ?

2. berikan tipe data yang tepat dari nilai-nilai berikut :
 - a. 2000
 - b. 1.5
 - c. -15000000
 - d. 999999999
3. Deklarasikan sebuah variabel yang beri nama **nilai akhir** kemudian isikan suatu nilai ke dalam variabel tersebut ?
4. Berapah nilai minimum dan maksimum yang dapat di tampung dalam tipe-tipe data berikut :
 - a. short
 - b. int
 - c. long double
 - d. char
5. Manakan nama-nama variabel yang benar dan nama variabel yang salah dari variabel-variabel berikut, dan berikan penjelasan.
 - a. 1Nilai
 - b. _kelas
 - c. long
 - d. CONST
 - e. \$dolar

D. Referensi

- Data, R. (t.thn.). *Data, Refsnes*. Dipetik October 22, 2019, dari https://www.w3schools.com/cpp/cpp_booleans.asp
- 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.
- Halterman, R. L. (2019). *Fundamental of C++ Programming*.
- Kirch-Prinz, U., & Kirch-Prinz, U. (2002). *A Complete Guide to Programming in C++*. Sudbury: Jones and Bartlett Publishers.
- S, R. A. (2018). *Logika Algoritma dan Pemrograman Dasar*. Bandung: Modula.
- Sjukani, M. (2014). *Algoritma dan Struktur Data 1 dengan C, C++ dan Java* (Edisi 9 ed.). Jakarta: Mltra Wacana Media.