

Operator

Bahasa C

Baharuddin Aziz

Semester I 2019/2020

Tujuan Perkuliahan

- Mengetahui konsep operator
- Memahami penerapan operator pada bahasa C

Operator

Operator

Apa itu Operator?

Operator: suatu simbol yang digunakan untuk melakukan operasi tertentu.

Contoh:

Menjumlahkan nilai dari variabel x dan y, maka perlu menggunakan operator penjumlahan (+).

$$x + y$$

Operator pada Bahasa C

Ada enam jenis kelompok operator pada bahasa C:

1. Operator Aritmatika;
2. Operator Penugasan;
3. Operator Perbandingan;
4. Operator Logika;
5. Operator *Bitwise*;
6. Operator Lain-lain.

Apa saja perbedaan dari semua jenis operator ini?

Operator Aritmatika

Operator Aritmatika

- **Operator aritmatika:**
operator yang digunakan untuk melakukan operasi aritmatika.

Nama Operator	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa bagi	⋈

Contoh Program: operator.c

```
#include <stdio.h>

void main(){

    int a, b, c;

    printf("Inputkan nilai a: ");
    scanf("%i", &a);
    printf("Inputkan nilai b: ");
    scanf("%i", &b);

    c = a + b;

    printf("Hasil a + b: %i", c);
}
```

```
D:\>gcc -o operator.exe operator.c
D:\>operator
Inputkan nilai a: 17
Inputkan nilai b: 29
Hasil a + b: 46
```


Contoh Program: operator1.c

```
#include <stdio.h>

void main(){

    int a, b;

    printf("Inputkan nilai a: ");
    scanf("%i", &a);
    printf("Inputkan nilai b: ");
    scanf("%i", &b);

    printf("Hasil a + b: %i\n", a + b);
    printf("Hasil a - b: %i\n", a - b);
    printf("Hasil a * b: %i\n", a * b);
    printf("Hasil a / b: %i\n", a / b);
    printf("Hasil a % b: %i\n", a % b);
}
```

```
D:\>gcc -o operator1.exe operator1.c

D:\>operator1
Inputkan nilai a: 7
Inputkan nilai b: 3
Hasil a + b: 10
Hasil a - b: 4
Hasil a * b: 21
Hasil a / b: 2
Hasil a b: 1

D:\>
```

Contoh Program: operator1.c

```
#include <stdio.h>

void main(){

    int a, b;

    printf("Inputkan nilai a: ");
    scanf("%i", &a);
    printf("Inputkan nilai b: ");
    scanf("%i", &b);

    printf("Hasil a + b: %i\n", a + b);
    printf("Hasil a - b: %i\n", a - b);
    printf("Hasil a * b: %i\n", a * b);
    printf("Hasil a / b: %i\n", a / b);
    printf("Hasil a % b: %i\n", a % b);
}
```

- Pada operasi pembagian,
 - $7/3$ hasilnya adalah 3
- *Mengapa demikian?*
 - *Bukankah seharusnya 2.333...?*
- Hal tersebut karena operasi dilakukan pada tipe data *integer*.
- ***Bila diubah programnya menggunakan tipe data float, maka...***

Contoh Program: operator2.c

```
#include <stdio.h>

void main(){

    float a, b;

    printf("Inputkan nilai a: ");
    scanf("%i", &a);
    printf("Inputkan nilai b: ");
    scanf("%i", &b);

    printf("Hasil a / b: %f\n", a / b);
}
```

- Pada operasi pembagian,
 - 7/3 hasilnya adalah 3
- *Mengapa demikian?*
 - *Bukankah seharusnya 2.333...?*
- Hal tersebut karena operasi dilakukan pada tipe data integer.
- ***Bila diubah programnya menggunakan tipe data float, maka...***

Contoh Program: operator2.c

```
#include <stdio.h>

void main(){

    float a, b;

    printf("Inputkan nilai a: ");
    scanf("%i", &a);
    printf("Inputkan nilai b: ");
    scanf("%i", &b);

    printf("Hasil a / b: %f\n", a / b);
}
```

```
D:\>gcc -o operator2.exe operator2.c
D:\>operator2
Inputkan nilai a: 7
Inputkan nilai b: 3
Hasil a / b: 2.333333
```

Operator Penugasan

Operator Penugasan

- **Operator penugasan** (*assignment operator*): operator untuk memberikan tugas pada variabel
- Biasanya untuk mengisi nilai.

Nama Operator	Simbol
Pengisian Nilai	=
Pengisian <u>dan</u> Penambahan	+=
Pengisian <u>dan</u> Pengurangan	-=
Pengisian <u>dan</u> Perkalian	*=
Pengisian <u>dan</u> Pembagian	/=
Pengisian <u>dan</u> Sisa Bagi	%=

Nama Operator	Simbol
Pengisian <u>dan</u> <i>shift left</i>	<<=
Pengisian <u>dan</u> <i>shift right</i>	>>=
Pengisian <u>dan</u> <i>bitwise AND</i>	&=
Pengisian <u>dan</u> <i>bitwise OR</i>	=
Pengisian <u>dan</u> <i>bitwise XOR</i>	^=

Contoh Program: operator3.c

```
#include <stdio.h>

void main(){
    int a, b;

    // pengisian nilai dengan operator =
    a = 5;
    b = 10;
    // pengisian sekaligus penambahan
    b += a; // ini sama seperti b = b + a
    printf("Hasil b += a adalah %d\n", b);
    // pengisian sekaligus pengurangan
    b -= a; // ini sama seperti b = b - a
    printf("Hasil b -= a adalah %d\n", b);
    // pengisian sekaligus perkalian
    b *= a; // ini sama seperti b = b * a
    printf("Hasil b *= a adalah %d\n", b);
    :
```

```

    // pengisian sekaligus pembagian
    b /= a; // ini sama seperti b = b / a
    printf("Hasil b /= a adalah %d\n", b);
    // pengisian sekaligus sisa bagi
    b %= a; // ini sama seperti b = b % a
    printf("Hasil b %= a adalah %d\n", b);
}
```

```
D:\>gcc -o operator3.exe operator3.c

D:\>operator3
Hasil b += a adalah 15
Hasil b -= a adalah 10
Hasil b *= a adalah 50
Hasil b /= a adalah 10
Hasil b = a adalah 0
```

Penjelasan Program: `operator3.c`

- Pada program tersebut, variabel `b` diisi ulang dengan operator penugasan.
- Sebagai contoh, operasi:

`b += a`

sama seperti operasi:

`b = b + a`

artinya: program akan mengisi nilai untuk `b` dengan *nilai `b` ditambah nilai `a`*.

Operator Pemandangan

Operator Perbandingan

- **Operator perbandingan:**
operator untuk membandingkan dua buah nilai.
- Operator ini juga dikenal dengan operator relasi.
- Nilai yang dihasilkan dari operasi perbandingan akan berupa **true** dan **false**.
 - Pada bahasa C, nilai true sama dengan 1 dan false sama dengan 0.

Nama Operator	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama dengan	==

Nama Operator	Simbol
Tidak Sama dengan	!=
Lebih Besar Sama dengan	>=
Lebih Kecil Sama dengan	<=

Contoh Program: operator4.c

```
#include <stdio.h>

void main(){
    int a = 4;
    int b = 3;

    printf("a = %d\n", a);
    printf("b = %d\n", b);
    // operator pembanding
    printf("a > b = %d\n", a > b);
    printf("a < b = %d\n", a < b);
    printf("a >= b = %d\n", a >= b);
    printf("a <= b = %d\n", a <= b);
    printf("a == b = %d\n", a == b);
    printf("a != b = %d\n", a != b);
}
```

```
D:\>gcc -o operator4.exe operator4.c

D:\>operator4
a = 4
b = 3
a > b = 1
a < b = 0
a >= b = 1
a <= b = 0
a == b = 0
a != b = 1
```

Operator Logika

Operator Logika [1/3]

Jika pernah belajar logika matematika, seharusnya tidak asing dengan:

Nama Operator	Simbol
Logika AND	& &
Logika OR	
Negasi/Kebalikan	!

Operator Logika [2/3]

Operator Logika digunakan untuk membuat operasi logika.

Misalnya:

- Pernyataan 1: Budi seorang programmer
- Pernyataan 2: Budi menggunakan Windows

Jika ditanya, apakah Budi programmer yang menggunakan Windows?

Cek dulu kebenarannya:

- Pernyataan 1: Budi seorang programmer = `true`.
- Pernyataan 2: Budi menggunakan Windows = `true`.

Apa Budi programmer dan menggunakan Windows?

`Pernyataan 1 && Pernyataan 2 = true`

Operator Logika [3/3]

Bingung? Coba cek tabel kebenaran untuk logika AND.

Pernyataan 1	Pernyataan 2	Logika AND
true	true	true
true	false	false
false	true	false
false	false	false

Contoh Program: operator5.c

```
#include <stdio.h>

void main(){
    int a = 1; // true
    int b = 0; // false

    printf("a = %d\n", a);
    printf("b = %d\n", b);
    // logika AND
    printf("a && b = %d\n", a && b);
    // logika OR
    printf("a || b = %d\n", a || b);
    // logika NOT
    printf("!a = %d\n", !a);
}
```

```
D:\>gcc -o operator5.exe operator5.c
D:\>operator5
a = 1
b = 0
a && b = 0
a || b = 1
!a = 0
```


Operator Bitwise

Operator Bitwise

- **Operator bitwise:**
operator untuk operasi berdasarkan bit (biner) dari suatu nilai.
- Misalkan ada nilai 6 dan 3. Nilai 6 dan 3 kemudian diubah ke dalam bentuk bilangan biner. Sehingga akan menjadi seperti ini:
$$6 = 0110$$
$$3 = 0011$$
- Operator *bitwise* akan melakukan operasi berdasarkan biner-biner tersebut.

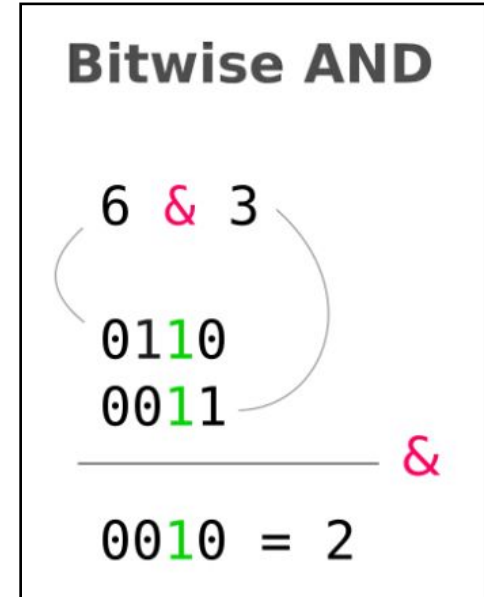
Operator Bitwise

(AND atau &)

Bitwise: AND

- **Bitwise AND:**
operasi bit berdasarkan logika AND.
- Perhatikan bilangan biner untuk nilai 6 dan 3.
 - Apabila melakukan operasi AND, maka akan menghasilkan bilangan biner baru.
 - Kemudian biner yang dihasilkan tersebut diubah kembali ke dalam bilangan desimal.
 - Maka hasilnya adalah 2.

<https://www.petanikode.com>



Contoh Program: operator6.c

```
#include <stdio.h>

void main(){
    int a = 6;
    int b = 3;

    printf("a & b = %d\n", a & b);
}
```

```
D:\>gcc -o operator6.exe operator6.c
```

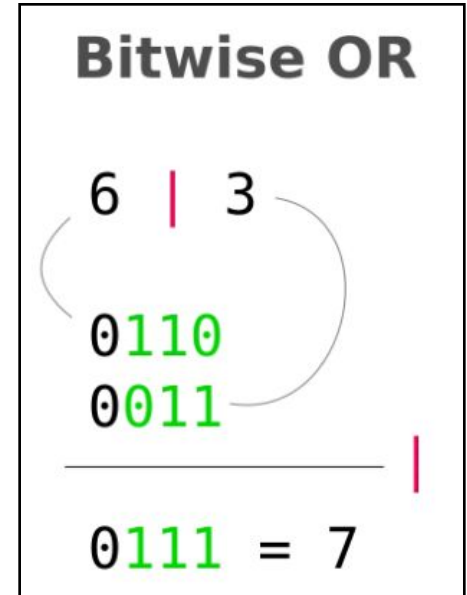
```
D:\>operator6
a & b = 2
```

Operator Bitwise

(OR atau |)

Bitwise: OR

- **Bitwise OR:**
menghasilkan nilai *false* atau 0 saat keduanya *false*.



Contoh Program: operator7.c

```
#include <stdio.h>

void main(){
    int a = 6;
    int b = 3;

    printf("a | b = %d\n", a | b);
}
```

```
D:\>gcc -o operator7.exe operator7.c
```

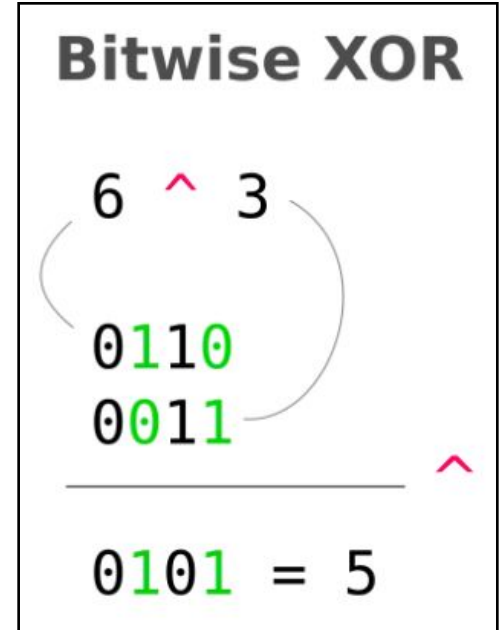
```
D:\>operator7
a | b = 7
```


Operator Bitwise

(XOR atau ^)

Bitwise: XOR

- **Bitwise XOR** (*Exclusive OR*):
menghasilkan nilai 1 saat kedua nilai tidak sama.



Contoh Program: operator8.c

```
#include <stdio.h>

void main(){
    int a = 6;
    int b = 3;

    printf("a ^ b = %d\n", a ^ b);
}
```

```
D:\>gcc -o operator8.exe operator8.c
```

```
D:\>operator8
a ^ b = 5
```

Operator Bitwise

(NOT atau ~)

Bitwise: XOR

- **Bitwise NOT** (komplemen): menghasilkan nilai biner yang terbalik dari biner aslinya.
- Kemudian direpresentasikan dengan komplemen dua.

<https://www.petanikode.com>

Bitwise NOT

6 → 0110
~6 → 1001 = -7

bit untuk simbol negatif dalam komplemen dua

Komplemen Dua

1000	=	-8
1001	=	-7
1010	=	-6
1100	=	-4
1101	=	-3
1110	=	-2
1111	=	-1
0000	=	0
0001	=	1
0010	=	2
0011	=	3
0100	=	4
0101	=	5
0110	=	6
0111	=	7

Contoh Program: operator9.c

```
#include <stdio.h>

void main(){
    int a = 6;

    printf("~a = %d\n", ~a);
}
```

```
D:\>gcc -o operator9.exe operator9.c
```

```
D:\>operator9
~a = -7
```

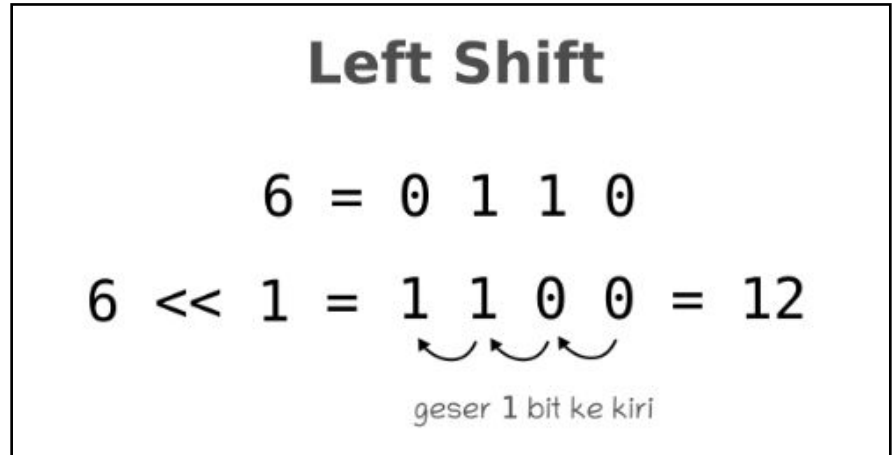
Operator Bitwise

(Left Shift atau \ll)

Bitwise: Left Shift

<https://www.petanikode.com>

- **Bitwise left shift:**
menghasilkan nilai biner yang digeser ke **kiri**.



Contoh Program: operator10.c

```
#include <stdio.h>

void main(){
    int a = 6;

    printf("a << 1 = %d\n", a << 1);
}
```

```
D:\>gcc -o operator10.exe operator10.c
```

```
D:\>operator10
a << 1 = 12
```

Operator Bitwise

(Right Shift atau >>)

Bitwise: Right Shift

- Bitwise right shift sama seperti left shift.
- Perbedaanya terletak pada arahnya.
- Right shift akan menggeser bit ke arah **kanan**.

<https://www.petanikode.com>



Contoh Program: operator11.c

```
#include <stdio.h>

void main(){
    int a = 6;

    printf("a >> 1 = %d\n", a >> 1);
}
```

```
D:\>gcc -o operator11.exe operator11.c
```

```
D:\>operator11
a >> 1 = 3
```

Operator Lain-lain

Operator Lain-lain

Beberapa operator lain yang perlu diketahui.

Nama Operator	Simbol	Keterangan
Alamat memori	&	untuk mengambil alamat memori
Pointer	*	untuk membuat pointer
Ternary	? :	untuk membuat kondisi
Increment	++	untuk menambah 1
Decrement	--	untuk mengurangi 1

Alamat Memori & Pointer

- Operator `&` jika digunakan pada satu variabel, berfungsi mengambil alamat memori dari variabel tersebut.
- Operator `*` berfungsi untuk membuat sebuah pointer.

Contoh Program: operator12.c

```
#include <stdio.h>

void main(){
    // membuat pointer
    int *a = 5;

    // ambil alamat memori dari pointer a
    printf("alamat memori var a %x\n", &a);
}
```

```
D:\>gcc -o operator12.exe operator12.c
operator12.c: In function 'main':
operator12.c:5:14: warning: initialization makes pointer
from integer without a cast [-Wint-conversion]
    int *a = 5;
              ^
D:\>operator12
alamat memori var a 60ff0c
```


Ternary

- Operator *ternary* untuk membuat **kondisi**.
- Cara kerjanya seperti percabangan if/else.

Contoh Program: operator13.c

```
#include <stdio.h>

void main(){
    int a = 4;

    printf("a > 1 %s", a > 1 ? "benar":"salah");
}
```

```
D:\>gcc -o operator13.exe operator13.c
```

```
D:\>operator13
a > 1 benar
```

Increment & Decrement

- Operator *increment* dan *decrement* untuk menambah dan mengurangi nilai dengan 1.
- Operator *increment* dan *decrement* dapat ditaruh di depan maupun di belakang variabel atau nilai.
- Operator ini biasanya digunakan pada blok perulangan.

Contoh Program: operator14.c

```
#include <stdio.h>

void main(){
    int a = 4;

    a++; // increment a
    printf("a++ = %d\n", a);

    ++a; // increment a
    printf("++a = %d\n", a);

    a--; // decrement a
    printf("a-- = %d\n", a);

    --a; // decrement a
    printf("--a = %d\n", a);
}
```

```
D:\>gcc -o operator14.exe operator14.c

D:\>operator14
a++ = 5
++a = 6
a-- = 5
--a = 4
```

Pertanyaan?