BINUS UNIVERSITY ONLINE LEARNING

People
Innovation
Excellence

LECTURE NOTES

COMP6599 – Algorithm and Programming

Minggu 3

Sesi 4

Operator, Operan, dan Aritmatika

LEARNING OUTCOMES

Learning Outcomes (Hasil Pembelajaran) :

Setelah menyelesaikan pembelajaran ini mahasiswa akan mampu :

1. LO 1 : Menjelaskan jenis algoritma dan membuat pemecahan masalahnya.
2. LO 2 : Menerapkan sintaks-sintaks dan fungsi-fungsi bahasa pemrograman C dalam pemecahan masalah.

OUTLINE MATERI (Sub-Topic): Operator, Operan, dan Aritmatika

1. Pengenalan Operator dan Operan
2. Operator Assignment
3. Operator Arimatika
4. Operator Relasional
5. Kondisional ekspresi
6. Operator Logika
7. Operator Bitwise
8. Precedence dan Associative

BINUS UNIVERSITY ONLINE LEARNING

People Innovation Excellence

Operator, Operan, dan Aritmatika

1. Pengenalan Operator dan Operan

   Dalam bahasa C terdapat banyak sekali oparator, yang digunakan untuk memproses operasi perhitungan, seperti *integer* di tambah dengan *integer*. Operan adalah data yang digunakan untuk memanipulasi atau dilakukan operasi.

2. Operasi *assignment*

   Merupakan operasi pemberian nilai ke suatu variabel. Assignment diberikan dengan simbol (=). Operan disebelah kiri operator harus berupa variabel (*L-value*). Operan di sebelah kanan harus berupa ekspresi (*R-value*).

   Berikut contoh dari operasi *assignment:*

   Operand1 = Operand2;

   x = 3; // konstan

   x = y; // jika y bernilai 5 maka x akan bernilai sama dengan y.

3. Operator Arimatika

   Operator aritmatika adalah operator yang digunakan untuk pengolahan aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, dua bilangan (bulat atau pecah), sisa bagi 2 bilangan bulat, penambahan dan pengurangan nilai antar variabel. Dan menghitung hasil sisa bagi dengan menggunakan simbol % yang disebut sebagai modulo.

   Berikut contoh dari operator aritmatika modulo pada bahasa C :

```
angka % 2 == 0 (bilangan genap)
jika
angka % 2 != 0 (bilangan ganjil)
```

Berikut tabel dari operasi aritmatika yang dapat digunakan pada bahasa C :

Symbol	Functionality	Example
+	Addition	x = y + 6;
-	Subtraction	y = x – 5;
*	Multiply	y = y * 3;
/	Division	z = x/y;
%	Modulo	A = 10 % 3;
++	Increment	x++;
--	Decrement	z--;
()	Scope / Priority	x=(2+3)*5

Bila ingin menuliskan sebuah operasi perhitungan penjumlahan secara akumulasi, misalkan **total** yang diperoleh dari penjumlahan **jumlah_belanja**, maka bentuk penulisan programnya adalah **total** = **total** + **jumlah_belanja**. Artinya adalah nilai pada **total** akan terus diakumulasi dengan nilai **jumlah_belanja** yang baru. Perhitungan ini dapat dipersingkat dengan **total** += **jumlah_belanja**. Demikian juga dengan operasi perhitungan lainnya seperti `'/='`, `'-='`, `'*='`, atau `'%='`.

Berikut contoh dari penggabungan operator (*combined operator*) :

Expression	Combined Operator
a = a + b;	a += b;
a = a – b;	a -= b;
a = a * b;	a *= b;
a = a / b;	a /= b;
a = a % b;	a %= b;
a = a ^ b ;	a ^= b;

Increment dan *Decrement* Operators

Selain *combined operator* seperti yang telah dijelaskan diatas, terdapat pula operator singkat untuk menggantikan fungsi `x = x + 1` atau `x += 1`, yaitu dengan menggunakan *increment* atau *decrement* operator.

Operasi penambahan atau pengurangan yang memainkan angka 1 disetiap akumulasinya maka dapat menggunakan *increment* ++ atau *decrement* --. Contoh `x += 1` dapat ditulis menjadi `x++`, operasi `x -= 1` dapat ditulis dengan `x--`.

Perlu diperhatikan bahwa bila menuliskan `++x` tidaklah salah, melainkan hal tersebut mempunyai arti yang berbeda dengan `x++`. Perhatikan contoh dibawah ini:

```
int x = 1;
x++;
printf("x = %d",x);
```

Nilai x yang tercetak diatas adalah 2 karena operasi x++ sudah dijalankan. Perhatikan potongan program berikut ini:

```
int x = 1;
printf("x = %d\n",x++);
printf("Final x = %d", x);
```

Maka *output* yang dihasilkan adalah :

```
x = 1
Final x = 2
```

Nilai x berubah menjadi 2 setelah proses x++ berhasil dieksekusi, tetapi bila mencetak nilai x++ maka nilai pada saat itu belum ditambahkan dengan 1.
Berbeda dengan x++, perhatikan contoh berikut:

```
int x = 1;
printf("x = %d\n",++x);
printf("Final x = %d", x);
```

Maka *output* yang dihasilkan adalah

```
x = 2
Final x = 2
```

Operasi ++x akan langsung dieksekusi yang artinya nilai x langsung ditambahkan satu pada saat itu juga.

4. Operator Relasional

Operasi relasional adalah operator yang digunakan untuk membandingkan 2 nilai sejenis. Kedua nilai tersebut dapat berupa konstanta atau variabel. Jika hasil perbandingan benar maka nilai pengembaliannya akan menjadi 1 (*true*), jika salah maka nilai pengembaliannya akan menjadi 0 (*false*).

Berikut adalah beberapa operator pembanding dan contoh serta hasil dari penggunaan operator tersebut:

Symbol	Functionality
= =	Equality
!=	Not equal
<	Less than
>	Greater than
<=	Less or equal than
>=	Greater or equal than
?:	Conditional assignment

Berikut contoh dari operator relasional pada bahasa C :

```
#include <stdio.h>
int main()
{
    int x = 7, y = 8;
    if(x==y) printf("%d sama dengan %d \n", x, y);
    if(x!=y) printf("%d tidak sama dengan %d \n", x, y);
    if(x<y) printf("%d kurang dari  %d \n", x, y);
    if(x>y) printf("%d lebih besar dari %d \n", x, y);
    if(x<=y) printf("%d lebih kecil sama dengan %d \n", x, y);
    if(x>=y) printf("%d lebih besar sama dengan %d \n", x, y);

getchar();
return 0;
}
```

```
7 tidak sama dengan 8
7 kurang dari  8
7 lebih kecil sama dengan 8
```

5. Kondisional ekspresi

Kondisional ekpresi merupakan operasi bersyarat dimana operator ini bersifat *triadic* (*trinary operator*) yang membutuhkan 3 operan dan dikenal juga sebagai ekspresi bersyarat (*conditional expression*). 3 operan yang diperlukan terdiri dari atas 1 ekspresi yang akan diuji dan memiliki 2 ekspresi pilihan. Kondisional ekpresi menggunakan simbol (:?). Berikut contoh dari kondisional ekspresi pada bahasa C:

```
#include <stdio.h>
int main()
{
    int max, a=3, b = 5;
    max = (a<b)? printf("Max : %d", a) : printf("Max : %d", b);

getchar();
return 0;
}
```

```
Max : 3
```

6. Operator Logika

Operator logika adalah operator yang berkaitan dengan operasi logika, seperti : negasi (ingkaran), konjungsi (dan), disjungsi (atau), nor (*exclusive or*). Berikut adalah tabel operasi logika pada bahasa C :

Operator	Name	Description
!	not	logical negation
&&	and	logical conjunction
||	or	logical disjunction
^	exclusive or	logical exclusion

Dibawah ini merupakan beberapa table kebenaran yang perlu ketahui untuk menghindari *logic eror* pada program yang akan dibuat.

Not (!) operator

p	!p	Example
true	false	!(1>2) is true, because (1>2) is false
false	true	!(1>0) is false, because (1>0) is true

Not operator digunakan untuk membalikkan hasil dari suatu perbandingan. Seperti pada contoh diatas, ketika 1>2 bernilai salah, maka ketika diberikan tanda '!', hasil dari operasi *logic* tersebut adalah benar.

AND (&&) operator

p1	p2	p1 && p2	Example
false	false	false	(2>3) && (5>5) is false Because both (2>3) and (5>5) are false
false	true	false	(2>3) && (6>5) is false Because (2>3) is false
true	false	false	(6>5) && (2>3) is false Because (2>3) is false
true	true	true	(3>2) && (5>=5) is true Because both (3>2) and (5>=5) are true

Operasi AND akan bernilai benar jika dan hanya jika kedua kondisi (atau semua kondisi) harus bernilai benar/**true**. Bila ada satu saja syarat yang idak dipenuhi (salah) maka hasil dari operasi AND bernilai **false**.

OR (||) operator

p1	p2	p1 || p2	Example
false	false	false	(2>3) || (5>5) is false Because both (2>3) and (5>5) are false
false	true	true	(2>3) || (6>5) is true Because (6>5) is true
true	false	true	(6>5) || (2>3) is true Because (6>5) is true
true	true	true	(3>2) || (5>=5) is true Because both (3>2) and (5>=5) are true

Kebalikan dari operasi **AND**, bila ada satu saja kondisi yang benar maka hasil dari operasi **OR** bernilai benar(***true***). Operasi OR bernilai salah jika dan hanya jika kedua kondisi (semua kondisi yang ada) bernilai salah(***false*)**.

7. Operator Bitwise

Operator bitwise adalah operator yang memperlakukan operan-operannya sebagai sebuah nilai tunggal, operator bitwise memperlakukan operan-operannya sebagai kuantitas yang terdiri atas bit-bit. Berikut tabel-tabel operator bitwise yang ada pada bahasa C :

Symbol	Meaning	Example
&	AND	A & B
|	OR	A | B;
^	XOR	A ^ B;
~	Complement	~A;
>>	Shift Right	A >> 3;
<<	Shift Left	B << 2;

Berikut adalah contoh program dari operator bitwise pada bahasa C :

```
#include <stdio.h>
int main()
{
    int a=3, b = 5;
    printf("%d & %d = %d\n",a,b, a&b);
    printf("%d | %d = %d\n",a,b, a|b);
    printf("%d ^ %d = %d\n",a,b, a^b);
    printf("%d << 2 = %d\n",a, a<<2);
    printf("%d >> 2 = %d\n",b, b>>2);

getchar();
return 0;
}
```

```
3 & 5 = 1
3 | 5 = 7
3 ^ 5 = 6
3 << 2 = 12
5 >> 2 = 1
```

8. Presedensi dan asosiatif (*Precedence dan Associative*)

Presedensi operator menunjukkan tingkat atau level operator, misalnya operator * memiliki presedensi yang lebih tinggi dari dibandingkan operator +. Operator dengan presedensi yang lebih tinggi akan dikerjakan terlebih dahulu. Asosiatif menjelaskan cara pengerjaan suatu ekpresi jika terdapat operator dengan tingkat atau level yang sama pada presedensi, misal pengerjaan dilakukan dari kiri ke kanan atau dari kanan ke kiri. Berikut tabel-tabel presedensi dan asosiatif yang ada pada bahasa C :

Precedence	Operator	Associativity
1	:: (unary) ::(binary)	right to left left to right
2	() [] -> .	left to right
3	++ -- + - ! ~ (type) & sizeof	right to left
4	.* ->*	left to right
5	* / %	left to right
6	+ -	left to right
7	<< >>	left to right
8	< <= > >=	left to right
9	== !=	left to right
10	&	left to right
11	^	left to right
12	|	left to right
13	&&	left to right
14	||	left to right
15	?:	right to left
16	= += -= *= /= &= ^= |= <<= >>=	right to left
17	,	left to right

SIMPULAN

1. Pengenalan Operator dan Operan

   Operator adalah proses operasi perhitungan, seperti *integer* di tambah dengan *integer*.

   Operan adalah data yang digunakan untuk memanipulasi atau dilakukan operasi.

2. Operasi *assignment*

   Assignment diberikan dengan simbol (=). Operan disebelah kiri operator harus berupa variabel (*L-value*). Operan di sebelah kanan harus berupa ekspresi (*R-value*).

3. Operator Arimatika

   Operator aritmatika adalah operator yang digunakan untuk pengolahan aritmatika seperti penjumlahan, pengurangan, perkalian, pembagian, dua bilangan (bulat atau pecah), sisa bagi 2 bilangan bulat, penambahan dan pengurangan nilai antar variabel.

4. Operator Relasional

   Operasi relasional adalah operator yang digunakan untuk membandingkan 2 nilai sejenis. Jika hasil perbandingan benar maka nilai pengembaliannya akan menjadi 1 (*true*), jika salah maka nilai pengembaliannya akan menjadi 0 (*false*).

5. Kondisional ekspresi

   Kondisional ekpresi merupakan operasi bersyarat dimana operator ini bersifat *triadic* (*trinary operator*) yang membutuhkan 3 operan. Kondisional ekpresi menggunakan simbol (:?).

6. Operator Logika

   Operator logika adalah operator yang berkaitan dengan operasi logika, seperti : negasi (ingkaran), konjungsi (dan), disjungsi (atau), nor (*exclusive or*).

7. Operator *Bitwise*

   Operator *bitwise* adalah operator yang memperlakukan operan-operannya sebagai kuantitas yang terdiri atas bit-bit.

8. Presedensi dan asosiatif (*Precedence dan Associative*)

   Presedensi operator menunjukkan tingkat atau level operator. Operator dengan presedensi yang lebih tinggi akan dikerjakan terlebih dahulu. Asosiatif menjelaskan cara pengerjaan suatu ekpresi jika terdapat operator dengan tingkat atau level yang sama pada presedensi.

BINUS UNIVERSITY ONLINE LEARNING

People
Innovation
Excellence

DAFTAR PUSTAKA

1. Paul J. Dietel,Harvey M. Deitel,. 2010. C : how to program. PEAPH. New Jersey. ISBN:978-0-13-705966-9 Chapter 2, 3 & 4
2. Manipulating Data with Operators: http://aelinik.free.fr/c/ch06.htm
3. Thompson Susabda Ngoen, 2006. Pengantar Algoritma dengan Bahasa C. Salemba Teknika. ISBN : 979-9549-25-6. Bagian 1, dan 2.