

LAPORAN
PRAKTIKUM PEMROGRAMAN BERORIENTASI OBJEK

OPERATOR

Untuk Memenuhi Salah Satu Tugas
Mata Kuliah Praktikum Pemrograman Berorientasi Objek
Dosen Pengampu: Suheri, S.T., M.Cs.



Disusun Oleh:
Rezky Yustisio Hadi Pratama (NIM:3201816021)

PROGRAM STUDI D3 TEKNIK INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI PONTIANAK

2019

KATA PENGANTAR

Puji dan syukur saya panjatkan kepada Allah SWT atas rahmat dan karunia-nya sehingga Laporan Tugas Mandiri yang berjudul “Instalasi Java dan IDE NetBeans” dapat terselesaikan dengan baik. Laporan Tugas Mandiri ini merupakan salah satu tugas yang diberikan oleh dosen pengampu mata kuliah Praktikum Pemrograman 3 kepada mahasiswa Program Studi D3 Teknik Informatika Jurusan Teknik Elektro sebagai salah satu bagian dari komponen penilaian akademis.

Laporan Tugas Mandiri ini membahas terkait Operator pada bahasa pemrograman Java. Demikian Laporan Tugas Mandiri ini saya buat, semoga bermanfaat.

Pontianak, 27 Maret 2019

Penyusun,

(Rezky Yustisio Hadi Pratama)

OPERATOR

1. PERCOBAAN 1: (OPERASI ARITMATIKA)

1.1. Program Operasi Aritmatika.

```
12 public class OperatorAritmatika {
13     public static void main(String[] args){
14         int a, b;
15         char c;
16         a = 10;
17         b = 2;
18         c = 'a';
19         System.out.println("Penjumlahan");
20         System.out.println("a + b = " + (a + b));
21         System.out.println("a + c = " + (a + c));
22         System.out.println("Pengurangan");
23         System.out.println("a - b = " + (a - b));
24         System.out.println("a - c = " + (a - c));
25         System.out.println("Perkalian");
26         System.out.println("a * b = " + (a * b));
27         System.out.println("a * c = " + (a * c));
28         System.out.println("Pembagian");
29         System.out.println("a / b = " + (a / b));
30         System.out.println("a / c = " + (a / c));
31         System.out.println("Modulus");
32         System.out.println("a % b = " + (a % b));
33         System.out.println("a % c = " + (a % c));
34     }
35 }
36
```

Source Code Program 1.1.

```
run:
Penjumlahan
a + b = 12
a + c = 107
Pengurangan
a - b = 8
a - c = -87
Perkalian
a * b = 20
a * c = 970
Pembagian
a / b = 5
a / c = 0
Modulus
a % b = 0
a % c = 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output Program 1.1.

- 1.2. Program di modifikasi dengan mengganti nilai variabel a,b dan c dengan nilai yang lain. Adapun nilai variabel **a = 2**, **b = 4**, dan **c = 'c'**.

```
12 public class OperatorAritmatika {
13     public static void main(String[] args){
14         int a, b;
15         char c;
16         a = 2;
17         b = 4;
18         c = 'c';
19         System.out.println("Penjumlahan");
20         System.out.println("a + b = " + (a + b));
21         System.out.println("a + c = " + (a + c));
22         System.out.println("Pengurangan");
23         System.out.println("a - b = " + (a - b));
24         System.out.println("a - c = " + (a - c));
25         System.out.println("Perkalian");
26         System.out.println("a * b = " + (a * b));
27         System.out.println("a * c = " + (a * c));
28         System.out.println("Pembagian");
29         System.out.println("a / b = " + (a / b));
30         System.out.println("a / c = " + (a / c));
31         System.out.println("Modulus");
32         System.out.println("a % b = " + (a % b));
33         System.out.println("a % c = " + (a % c));
34     }
35 }
```

Source Code Program 1.2.

```
run:
Penjumlahan
a + b = 6
a + c = 101
Pengurangan
a - b = -2
a - c = -97
Perkalian
a * b = 8
a * c = 198
Pembagian
a / b = 0
a / c = 0
Modulus
a % b = 2
a % c = 2
BUILD SUCCESSFUL (total time: 1 second)
```

Output Program 1.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan suatu operasi perhitungan pada variabel a, b dan c. Yang dimana perhitungan tersebut antara lain pertambahan, pengurangan, perkalian, pembagian dan modulus.

Berikut ini adalah tabel operator aritmatika beserta tipe data hasil perhitungan aritmatika.

OPERATOR	DESKRIPSI	TIPE DATA HASIL
+	Penjumlahan	<ul style="list-style-type: none"> - Bilangan Bulat jika semua Operan bertipe data Bilangan Bulat - Bilangan Bulat jika salah satu Bilangan Bulat dan yang satunya lagi char atau sebaliknya - Bilangan Pecahan jika salah satu atau semua Operan bertipe data Bilangan Pecahan
-	Pengurangan	
*	Perkalian	
/	Pembagian	
%	Modulus (Sisa Bagi)	

Bagaimana dengan variabel c yang berisikan sebuah karakter, kenapa bisa di lakukan perhitungan ? karena karakter tersebut akan di ubah ke dalam bentuk **ASCII** yang tadinya adalah 'c' (tipe data char) akan berubah menjadi angka 99. Sehingga operasi perhitungan bisa di lakukan.

Berikut ini adalah tabel ASCII.

Dec	Hex	Name	Char	Ctrl-char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	0	Null	NUL	CTRL-@	32	20	Space	64	40	@	96	60	`
1	1	Start of heading	SOH	CTRL-A	33	21	!	65	41	A	97	61	a
2	2	Start of text	STX	CTRL-B	34	22	"	66	42	B	98	62	b
3	3	End of text	ETX	CTRL-C	35	23	#	67	43	C	99	63	c
4	4	End of xmit	EOT	CTRL-D	36	24	\$	68	44	D	100	64	d
5	5	Enquiry	ENQ	CTRL-E	37	25	%	69	45	E	101	65	e
6	6	Acknowledge	ACK	CTRL-F	38	26	&	70	46	F	102	66	f
7	7	Bell	BEL	CTRL-G	39	27	'	71	47	G	103	67	g
8	8	Backspace	BS	CTRL-H	40	28	(72	48	H	104	68	h
9	9	Horizontal tab	HT	CTRL-I	41	29)	73	49	I	105	69	i
10	0A	Line feed	LF	CTRL-J	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	VT	CTRL-K	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	FF	CTRL-L	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage feed	CR	CTRL-M	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	SO	CTRL-N	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	SI	CTRL-O	47	2F	/	79	4F	O	111	6F	o
16	10	Data line escape	DLE	CTRL-P	48	30	0	80	50	P	112	70	p
17	11	Device control 1	DC1	CTRL-Q	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	DC2	CTRL-R	50	32	2	82	52	R	114	72	r
19	13	Device control 3	DC3	CTRL-S	51	33	3	83	53	S	115	73	s
20	14	Device control 4	DC4	CTRL-T	52	34	4	84	54	T	116	74	t
21	15	Neg acknowledge	NAK	CTRL-U	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	SYN	CTRL-V	54	36	6	86	56	V	118	76	v
23	17	End of xmit block	ETB	CTRL-W	55	37	7	87	57	W	119	77	w
24	18	Cancel	CAN	CTRL-X	56	38	8	88	58	X	120	78	x
25	19	End of medium	EM	CTRL-Y	57	39	9	89	59	Y	121	79	y
26	1A	Substitute	SUB	CTRL-Z	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	ESC	CTRL-[59	3B	;	91	5B	[123	7B	{
28	1C	File separator	FS	CTRL-\	60	3C	<	92	5C	\	124	7C	
29	1D	Group separator	GS	CTRL-]	61	3D	=	93	5D]	125	7D	}
30	1E	Record separator	RS	CTRL-^	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	US	CTRL-~	63	3F	?	95	5F	_	127	7F	DEL

2. PERCOBAAN 2: (OPERASI INCREMENT DAN DECREMENT)

2.1. Program Operasi Increment(Kenaikan) dan Decrement(Pengurangan)

```
12 public class OperatorIncDec {
13     public static void main(String args[]){
14         int x;
15         x = 10;
16         System.out.println("Perintah Increment");
17         System.out.println("++x hasilnya: " + ++x);
18         System.out.println("Nilai x: " + x);
19         System.out.println("x++ hasilnya: " + x++);
20         System.out.println("Nilai x: " + x);
21         System.out.println("Perintah Decrement");
22         System.out.println("--x hasilnya: " + --x);
23         System.out.println("Nilai x: " + x);
24         System.out.println("x-- hasilnya: " + x--);
25         System.out.println("Nilai x: " + x);
26     }
27 }
28
```

Source Code Program 2.1.

```
run:
Perintah Increment
++x hasilnya: 11
Nilai x: 11
x++ hasilnya: 11
Nilai x: 12
Perintah Decrement
--x hasilnya: 11
Nilai x: 11
x-- hasilnya: 11
Nilai x: 10
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output Program 2.1.

2.2. Program di modifikasi dengan mengganti nilai variabel x dengan nilai yang lain. Adapun nilai variabel **x = 5**.

```

12 public class OperatorIncDec {
13     public static void main(String args[]){
14         int x;
15         x = 5;
16         System.out.println("Perintah Increment");
17         System.out.println("++x hasilnya: " + ++x);
18         System.out.println("Nilai x: " + x);
19         System.out.println("x++ hasilnya: " + x++);
20         System.out.println("Nilai x: " + x);
21         System.out.println("Perintah Decrement");
22         System.out.println("--x hasilnya: " + --x);
23         System.out.println("Nilai x: " + x);
24         System.out.println("x-- hasilnya: " + x--);
25         System.out.println("Nilai x: " + x);
26     }
27 }

```

Source Code Program 2.2

```

run:
Perintah Increment
++x hasilnya: 6
Nilai x: 6
x++ hasilnya: 6
Nilai x: 7
Perintah Decrement
--x hasilnya: 6
Nilai x: 6
x-- hasilnya: 6
Nilai x: 5
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 2.2

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan suatu instruksi Increment dan Decrement. Increment adalah Operator yang di gunakan untuk melakukan operasi menaikan nilai dari sebuah variabel(bertipe data bilangan bulat, pecahan, atau char) dengan menambahkannya dengan nilai 1. Sedangkan decrement mirip dengan increment, namun decrement mengurangi nilai variabel dengan nilai 1. Masing-masing instruksi tersebut menggunakan 2 tipe, yaitu **PRE** dan **POST**.

Berikut penggunaan tipe **PRE** dan **POST** dengan variabel **x = 5**.

OPERATOR	DESKRIPSI	PRE	POST
++	Increment	hasil = ++x; hasil = 6, x = 6	hasil = x++; hasil = 6, x = 7
--	Decrement	hasil = --x; hasil = 6, x = 6	hasil = x--; hasil = 6, x = 5

- ✓ Tipe PRE, suatu nilai akan di increment/decrement terlebih dahulu, baru kemudian nilai tersebut di gunakan pada operasi yang lain.
- ✓ Tipe POST, suatu nilai digunakan terlebih dahulu, baru kemudian nilai tersebut di increment/decrement.

3. PERCOBAAN 3: (OPERASI MAJEMUK)

3.1. Program Operasi Majemuk.

```

12 public class OperatorMajemuk {
13     public static void main(String args[]){
14         int x;
15         x = 10;
16         System.out.println("x += 2: " + (x += 2));
17         System.out.println("x -= 2: " + (x -= 2));
18         System.out.println("x *= 2: " + (x *= 2));
19         System.out.println("x /= 2: " + (x /= 2));
20         System.out.println("x %= 2: " + (x %= 2));
21     }
22 }
23

```

Source Code Program 3.1.

```

run:
x += 2: 12
x -= 2: 10
x *= 2: 20
x /= 2: 10
x %= 2: 0
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 3.1.

- 3.2. Program di modifikasi dengan mengganti nilai variabel **x** dan angka **2** dengan nilai yang lain. Adapun nilai variabel **x = 15** dan **angka 2** di ganti dengan **angka 4**.


```

12 public class OperatorMajemuk {
13     public static void main(String args[]) {
14         int x;
15         x = 15;
16         System.out.println("x += 4: " + (x += 4));
17         System.out.println("x -= 4: " + (x -= 4));
18         System.out.println("x *= 4: " + (x *= 4));
19         System.out.println("x /= 4: " + (x /= 4));
20         System.out.println("x %= 4: " + (x %= 4));
21     }
22 }

```

Source Code Program 3.2.

```

run:
x += 4: 19
x -= 4: 15
x *= 4: 60
x /= 4: 15
x %= 4: 3
BUILD SUCCESSFUL (total time: 0 seconds)

```

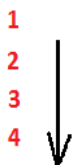
Output Program 3.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan suatu operasi perhitungan terhadap variabel x. Operasi perhitungan tersebut menggunakan Operator majemuk. Operator Majemuk adalah gabungan operator yang digunakan untuk melakukan suatu operasi tertentu. Terdapat 4 Operator Majemuk, antara lain += (pertambahan), -= (pengurangan), *= (perkalian), /= (pembagian), dan %= (modulus).

Berikut ini adalah tabel Operasi Majemuk untuk penyelesaian perhitungan variabel **x = 15** pada program.

OPERATOR	DESKRIPSI	PENYELESAIAN
+=	Penambahan diikuti dengan Penugasan	X += 4 sama dengan X = X + 4, hasilnya X = 19
-=	Pengurangan diikuti dengan penugasan	X -= 4 sama dengan X = X - 4, hasilnya X = 15
*=	Perkalian diikuti dengan Penugasan	X *= 4 sama dengan X = X * 4, hasilnya X = 60
/=	Pembagian diikuti dengan Penugasan	X /= 4 sama dengan X = X / 4, hasilnya X = 15
%=	Modulus diikuti dengan Penugasan	X %= 4 sama dengan X = X % 4, hasilnya X = 3



4. PERCOBAAN 4: (OPERATOR RELASIONAL)

4.1. Program Operator Relasional.

```
12 public class OperatorRelasional {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 2;
16         y = 3;
17         z = 2;
18         System.out.println("Operator ==");
19         System.out.println("x == y? " + (x == y));
20         System.out.println("x == z? " + (x == z));
21         System.out.println("Operator !=");
22         System.out.println("x != y? " + (x != y));
23         System.out.println("x != z? " + (x != z));
24         System.out.println("Operator >");
25         System.out.println("x > y? " + (x > y));
26         System.out.println("x > z? " + (x > z));
27         System.out.println("Operator <");
28         System.out.println("x < y? " + (x < y));
29         System.out.println("x < z? " + (x < z));
30         System.out.println("Operator >=");
31         System.out.println("x >= y? " + (x >= y));
32         System.out.println("x >= z? " + (x >= z));
33         System.out.println("Operator <=");
34         System.out.println("x <= y? " + (x <= y));
35         System.out.println("x <= z? " + (x <= z));
36     }
37 }
38
```

Source Code Program 4.1.

```

run:
Operator ==
x == y? false
x == z? true
Operator !=
x != y? true
x != z? false
Operator >
x > y? false
x > z? false
Operator <
x < y? true
x < z? false
Operator >=
x >= y? false
x >= z? true
Operator <=
x <= y? true
x <= z? true
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 4.1.

4.2. Program di modifikasi dengan mengganti nilai variabel x, y, dan z dengan nilai yang lain. Adapun nilai variabel **x = 4**, **y = 2**, dan **z = 4**.

```

12 public class OperatorRelasional {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 4;
16         y = 2;
17         z = 4;
18         System.out.println("Operator ==");
19         System.out.println("x == y? " + (x == y));
20         System.out.println("x == z? " + (x == z));
21         System.out.println("Operator !=");
22         System.out.println("x != y? " + (x != y));
23         System.out.println("x != z? " + (x != z));
24         System.out.println("Operator >");
25         System.out.println("x > y? " + (x > y));
26         System.out.println("x > z? " + (x > z));
27         System.out.println("Operator <");
28         System.out.println("x < y? " + (x < y));
29         System.out.println("x < z? " + (x < z));
30         System.out.println("Operator >=");
31         System.out.println("x >= y? " + (x >= y));
32         System.out.println("x >= z? " + (x >= z));
33         System.out.println("Operator <=");
34         System.out.println("x <= y? " + (x <= y));
35         System.out.println("x <= z? " + (x <= z));
36     }
37 }

```

Source Code Program 4.2.

```

run:
Operator ==
x == y? false
x == z? true
Operator !=
x != y? true
x != z? false
Operator >
x > y? true
x > z? false
Operator <
x < y? false
x < z? false
Operator >=
x >= y? true
x >= z? true
Operator <=
x <= y? false
x <= z? true
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 4.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan suatu operasi perbandingan antara dua buah operan yang menghasilkan nilai yang bertipe data logika(true atau false). Operasi perbandingan ini disebut Operator Relasional. Di dalam Operator Relasional terdapat 5 macam operator, antara lain == (sama dengan), != (tidak sama dengan), > (lebih besar dari), < (lebih kecil dari), >= (lebih besar dari sama dengan), dan <= (lebih kecil dari sama dengan). Jika hasil perbandingan tersebut benar, maka akan menghasilkan nilai **true**. Dan jika hasil perbandingan tersebut salah, maka akan menghasilkan nilai **false**.

Berikut ini adalah tabel Operasi Relasional untuk penyelesaian operasi perbandingan variabel **x = 4**, **y = 2**, dan **z = 4**.

OPERATOR	DESKRIPSI	PENYELESAIAN
==	Sama dengan	X == Y, hasilnya false ; X == Z, hasilnya true ;
!=	Tidak sama dengan	X != Y, hasilnya true ; X != Z, hasilnya false ;
>	Lebih besar dari	X > Y, hasilnya true ; X > Z, hasilnya false ;
<	Lebih kecil dari	X < Y, hasilnya false ; X < Z, hasilnya false ;
>=	Lebih besar dari atau sama dengan	X >= Y, hasilnya true ; X >= Z, hasilnya true ;
<=	Lebih kecil dari atau sama dengan	X <= Y, hasilnya false ; X <= Z, hasilnya true ;

5. PERCOBAAN 5: (OPERATOR LOGIKA AND)

5.1. Program Operator Logika AND.

```
12 public class OperatorLogikaAND {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 2; y = 2; z = 3;
16         System.out.println("Nilai Awal");
17         System.out.println("x = " + x);
18         System.out.println("y = " + y);
19         System.out.println("z = " + z);
20         System.out.println("Operasi Logika &&");
21         System.out.println("(x==y) && (++x==z) => " + ((x==y) && (++x==z)));
22         System.out.println("x = " + x);
23         System.out.println("y = " + y);
24         System.out.println("z = " + z);
25         System.out.println("(x==y) && (++x==z) => " + ((x==y) && (++x==z)));
26         System.out.println("x = " + x);
27         System.out.println("y = " + y);
28         System.out.println("z = " + z);
29
30         x = 2; y = 2; z = 3;
31         System.out.println("Nilai Awal");
32         System.out.println("x = " + x);
33         System.out.println("y = " + y);
34         System.out.println("z = " + z);
35         System.out.println("Operasi Logika &");
36         System.out.println("(x==y) & (++x==z) => " + ((x==y) & (++x==z)));
37         System.out.println("x = " + x);
38         System.out.println("y = " + y);
39         System.out.println("z = " + z);
40         System.out.println("(x==y) & (++x==z) => " + ((x==y) & (++x==z)));
41         System.out.println("x = " + x);
42         System.out.println("y = " + y);
43         System.out.println("z = " + z);
44     }
45 }
```

Source Code Program 5.1.

```

run:
Nilai Awal
x = 2
y = 2
z = 3
Operasi Logika &&
(x==y) && (++x==z) => true
x = 3
y = 2
z = 3
(x==y) && (++x==z) => false
x = 3
y = 2
z = 3
Nilai Awal
x = 2
y = 2
z = 3
Operasi Logika &
(x==y) & (++x==z) => true
x = 3
y = 2
z = 3
(x==y) & (++x==z) => false
x = 4
y = 2
z = 3
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 5.1.

5.2. Program di modifikasi dengan mengganti nilai variabel x, y dan z dengan nilai yang lain. Adapun nilai variabel **x = 3**, **y = 3**, dan **z = 4**.

```

12 public class OperatorLogikaAND {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 3; y = 3; z = 4;
16         System.out.println("Nilai Awal");
17         System.out.println("x = " + x);
18         System.out.println("y = " + y);
19         System.out.println("z = " + z);
20         System.out.println("Operasi Logika &&");
21         System.out.println("(x==y) && (++x==z) => " + ((x==y) && (++x==z)));
22         System.out.println("x = " + x);
23         System.out.println("y = " + y);
24         System.out.println("z = " + z);
25         System.out.println("(x==y) && (++x==z) => " + ((x==y) && (++x==z)));
26         System.out.println("x = " + x);
27         System.out.println("y = " + y);
28         System.out.println("z = " + z);

```

```

29      x = 3; y = 3; z = 4;
30      System.out.println("Nilai Awal");
31      System.out.println("x = " + x);
32      System.out.println("y = " + y);
33      System.out.println("z = " + z);
34      System.out.println("Operasi Logika &");
35      System.out.println("(x==y) & (++x==z) => " + ((x==y) & (++x==z)));
36      System.out.println("x = " + x);
37      System.out.println("y = " + y);
38      System.out.println("z = " + z);
39      System.out.println("(x==y) & (++x==z) => " + ((x==y) & (++x==z)));
40      System.out.println("x = " + x);
41      System.out.println("y = " + y);
42      System.out.println("z = " + z);
43  }
44  }

```

Source Code Program 5.2.

```

run:
Nilai Awal
x = 3
y = 3
z = 4
Operasi Logika &&
(x==y) && (++x==z) => true
x = 4
y = 3
z = 4
(x==y) && (++x==z) => false
x = 4
y = 3
z = 4
Nilai Awal
x = 3
y = 3
z = 4
Operasi Logika &
(x==y) & (++x==z) => true
x = 4
y = 3
z = 4
(x==y) & (++x==z) => false
x = 5
y = 3
z = 4
BUILD SUCCESSFUL (total time: 1 second)

```

Output Program 5.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan suatu operasi perbandingan logika AND (Short Circuit) dan Logika AND (No Short Circuit) yang nantinya hasil dari operasi tersebut adalah nilai true atau false. Operator logika AND (Short Circuit) menggunakan simbol **&&** (ganda). Operasi Logika AND (Short Circuit) digunakan untuk melakukan operasi perbandingan terhadap nilai logika (true atau false) yang hanya akan mengecek nilai operan pertama saja, jika operan pertama bernilai **false** dan hanya akan menghasilkan nilai **true** jika dan hanya jika semua operan bernilai **true**. Sedangkan operator logika AND (No Short Circuit) menggunakan simbol **&** (tunggal). Operator logika AND (No Short Circuit) digunakan untuk melakukan operasi perbandingan terhadap nilai logika (true atau false) dari semua operan yang terlibat di dalamnya dan hanya akan menghasilkan nilai **true** jika dan hanya jika semua operan bernilai **true**. Perbedaan dari kedua Operator tersebut adalah jika Operator Short Circuit hanya mengecek operan pertama saja. Jika operan pertama **true**, maka hasilnya sudah pasti **true**. Untuk operan selanjutnya tidak akan di cek lagi. Sedangkan Operator Non Short Circuit mirip dengan Operator Short Circuit, bedanya adalah Non Short Circuit tetap mengecek operan selanjutnya walaupun hasil operan pertama telah di dapatkan. Kemudian terdapat juga operator Increment dengan tipe **PRE** pada variabel **x**.

Berikut ini adalah tabel kebenaran Logika AND untuk penyelesaian operasi perbandingan pada program dengan variabel **x = 3, y = 3, dan z =**

4.

OPERAN1	OPERAN2	CEK OPERAN1	CEK OPERAN2	OPERAN1&& OPERAN2	OPERAN1 & OPERAN2	DESKRIPSI
(x==y)	(++x==z)	true	true	true	-	Operasi perbandingan untuk baris 21 dan 25
(x==y)	(++x==z)	false	false	false	-	
(x==y)	(++x==z)	true	true	-	true	Operasi perbandingan untuk baris 35 dan 39
(x==y)	(++x==z)	false	false	-	false	

6. PERCOBAAN 6: (OPERATOR LOGIKA OR)

6.1. Program Operator Logika OR.

```
12 public class OperatorLogikaOR {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 2; y = 2; z = 3;
16         System.out.println("Nilai Awal");
17         System.out.println("x = " + x);
18         System.out.println("y = " + y);
19         System.out.println("z = " + z);
20         System.out.println("Operasi Logika ||");
21         System.out.println("(x==y) || (++x==z) => " + ((x==y) || (++x==z)));
22         System.out.println("x = " + x);
23         System.out.println("y = " + y);
24         System.out.println("z = " + z);
25         System.out.println("(x==z) || (++x==z) => " + ((x==z) || (++x==z)));
26         System.out.println("x = " + x);
27         System.out.println("y = " + y);
28         System.out.println("z = " + z);
29
30         x = 2; y = 2; z = 3;
31         System.out.println("Nilai Awal");
32         System.out.println("x = " + x);
33         System.out.println("y = " + y);
34         System.out.println("z = " + z);
35         System.out.println("Operasi Logika |");
36         System.out.println("(x==y) | (++x==z) => " + ((x==y) | (++x==z)));
37         System.out.println("x = " + x);
38         System.out.println("y = " + y);
39         System.out.println("z = " + z);
40         System.out.println("(x==z) | (++x==z) => " + ((x==z) | (++x==z)));
41         System.out.println("x = " + x);
42         System.out.println("y = " + y);
43         System.out.println("z = " + z);
44     }
45 }
```

Source Code Program 6.1.

```

run:
Nilai Awal
x = 2
y = 2
z = 3
Operasi Logika ||
(x==y) || (++x==z) => true
x = 2
y = 2
z = 3
(x==z) || (++x==z) => true
x = 3
y = 2
z = 3
Nilai Awal
x = 2
y = 2
z = 3
Operasi Logika |
(x==y) | (++x==z) => true
x = 3
y = 2
z = 3
(x==z) | (++x==z) => true
x = 4
y = 2
z = 3
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 6.1.

6.2. Program di modifikasi dengan mengganti nilai variabel x, y, dan z dengan nilai yang lain. Adapun nilai variabel **x = 8**, **y = 8**, dan **z = 9**

```

12 public class OperatorLogikaOR {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 4; y = 8; z = 9;
16         System.out.println("Nilai Awal");
17         System.out.println("x = " + x);
18         System.out.println("y = " + y);
19         System.out.println("z = " + z);
20         System.out.println("Operasi Logika ||");
21         System.out.println("(x==y) || (++x==z) => " + ((x==y) || (++x==z)));
22         System.out.println("x = " + x);
23         System.out.println("y = " + y);
24         System.out.println("z = " + z);
25         System.out.println("(x==z) || (++x==z) => " + ((x==z) || (++x==z)));
26         System.out.println("x = " + x);
27         System.out.println("y = " + y);
28         System.out.println("z = " + z);

```

```

29      x = 4; y = 8; z = 9;
30      System.out.println("Nilai Awal");
31      System.out.println("x = " + x);
32      System.out.println("y = " + y);
33      System.out.println("z = " + z);
34      System.out.println("Operasi Logika |");
35      System.out.println("(x==y) | (++x==z) => " + ((x==y) | (++x==z)));
36      System.out.println("x = " + x);
37      System.out.println("y = " + y);
38      System.out.println("z = " + z);
39      System.out.println("(x==z) | (++x==z) => " + ((x==z) | (++x==z)));
40      System.out.println("x = " + x);
41      System.out.println("y = " + y);
42      System.out.println("z = " + z);
43  }
44  }

```

Source Code Program 6.2.

```

run:
Nilai Awal
x = 4
y = 8
z = 9
Operasi Logika ||
(x==y) || (++x==z) => false
x = 5
y = 8
z = 9
(x==z) || (++x==z) => false
x = 6
y = 8
z = 9
Nilai Awal
x = 4
y = 8
z = 9
Operasi Logika |
(x==y) | (++x==z) => false
x = 5
y = 8
z = 9
(x==z) | (++x==z) => false
x = 6
y = 8
z = 9
BUILD SUCCESSFUL (total time: 4 seconds)

```

Output Program 6.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan suatu operasi perbandingan logika OR (Short Circuit) dan Logika OR (No Short Circuit) yang nantinya hasil dari operasi tersebut adalah nilai true atau false. Operator logika OR (Short Circuit) menggunakan simbol `||` (ganda). Operasi logika OR digunakan untuk melakukan operasi perbandingan terhadap nilai logika (true atau false) yang hanya akan mengecek nilai operan pertama saja. Jika operan pertama bernilai true, maka hasil operasinya adalah true. Dan hanya akan menghasilkan nilai false jika dan hanya jika semua operan bernilai false. Sedangkan operator logika OR (No Short Circuit) menggunakan simbol `|` (tunggal). Operator logika OR (No Short Circuit) digunakan untuk melakukan operasi perbandingan terhadap nilai logika (true atau false) dari semua operan yang terlibat di dalamnya dan hanya akan menghasilkan nilai false jika dan hanya jika semua operan bernilai false. Perbedaan dari kedua Operator tersebut adalah jika Operator Short Circuit hanya mengecek operan pertama saja. Jika operan pertama **true**, maka hasilnya sudah pasti **true**. Untuk operan selanjutnya tidak akan di cek lagi. Sedangkan Operator Non Short Circuit mirip dengan Operator Short Circuit, bedanya adalah Non Short Circuit tetap mengecek operan selanjutnya walaupun hasil operan pertama telah di dapatkan. Kemudian terdapat juga operator Increment dengan tipe **PRE** pada variabel **x**.

Berikut ini adalah tabel kebenaran Logika AND untuk penyelesaian operasi perbandingan pada program dengan variabel **x = 4, y = 8, dan z = 9**.

OPERAN1	OPERAN2	CEK OPERAN1	CEK OPERAN2	OPERAN1 OPERAN2	OPERAN1 OPERAN2	DESKRIPSI
(x==y)	(++x==z)	false	false	false	-	Operasi perbandingan untuk baris 21 dan 25
(x==y)	(++x==z)	false	false	false	-	
(x==y)	(++x==z)	false	false	-	false	Operasi perbandingan untuk baris 35 dan 39
(x==y)	(++x==z)	false	false	-	false	

7. PERCOBAAN 7: (OPERATOR LOGIKA NOT)

7.1. Program Operator Logika NOT.

```

12 public class OperatorLogikaNOT {
13     public static void main(String args[]){
14         boolean status;
15         status = true;
16         System.out.println("Nilai Awal");
17         System.out.println("Variabel status bernilai: " + status);
18         System.out.println("Operator NOT (!)");
19         System.out.println("!status = " + !status);
20     }
21 }
22

```

Source Code Program 7.1.

```

run:
Nilai Awal
Variabel status bernilai: true
Operator NOT (!)
!status = false
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 7.1.

7.2. Program di modifikasi dengan mengganti nilai variabel status dengan nilai yang lain. Adapun nilai variabel **status = false;**

```

12 public class OperatorLogikaNOT {
13     public static void main(String args[]){
14         boolean status;
15         status = false;
16         System.out.println("Nilai Awal");
17         System.out.println("Variabel status bernilai: " + status);
18         System.out.println("Operator NOT (!)");
19         System.out.println("!status = " + !status);
20     }
21 }

```

Source Code Program 7.2.

```

run:
Nilai Awal
Variabel status bernilai: false
Operator NOT (!)
!status = true
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 7.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi pembalikan atau inversi terhadap nilai logika (true atau false). Operator pembalikan menggunakan operator **!** (tanda seru). Pada program, nilai dari variabel **status** adalah **false**. Ketika variabel **status** di lakukan pembalikan dengan menambahkan **!** (tanda seru) di depan nama variabelnya. Maka nilai dari variabel **status** berubah menjadi **true**.

Berikut ini adalah tabel kebenaran Logika AND untuk penyelesaian operasi perbandingan pada program dengan variabel **status = false**.

status	!status
false	true

8. PERCOBAAN 8: (OPERATOR BITWISE COMPLEMENT)

8.1. Program Operator Bitwise Complement.

```
12 public class OperatorBitwiseComplement {
13     public static void main(String args[]){
14         int x = 0;
15         int y = -3;
16         System.out.println("Operator Bitwise - Complement");
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x = " + x);
19         System.out.println("Nilai y = " + y);
20         System.out.println("Operasi: x = ~x");
21         System.out.println("Nilai x = " + ~x);
22         System.out.println("Operasi: y = ~y");
23         System.out.println("Nilai y = " + ~y);
24     }
25 }
26
```

Source Code Program 8.1.

```

run:
Operator Bitwise - Complement
Nilai Awal
Nilai x = 0
Nilai y = -3
Operasi: x = ~x
Nilai x = -1
Operasi: y = ~y
Nilai y = 2
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 8.1.

8.2. Program di modifikasi dengan mengganti nilai variabel x dan y dengan nilai yang lain. Adapun nilai variabel x = 1 dan y = -2.

```

12 public class OperatorBitwiseComplement {
13     public static void main(String args[]) {
14         int x = 1;
15         int y = -2;
16         System.out.println("Operator Bitwise - Complement");
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x = " + x);
19         System.out.println("Nilai y = " + y);
20         System.out.println("Operasi: x = ~x");
21         System.out.println("Nilai x = " + ~x);
22         System.out.println("Operasi: y = ~y");
23         System.out.println("Nilai y = " + ~y);
24     }
25 }

```

Source Code Program 8.2.

```

run:
Operator Bitwise - Complement
Nilai Awal
Nilai x = 1
Nilai y = -2
Operasi: x = ~x
Nilai x = -2
Operasi: y = ~y
Nilai y = 1
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 8.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi complement (dalam bentuk sistem bilangan biner) pada nilai operannya. Dalam operasi complement menggunakan rumus $\sim x = -x-1$.

Berikut adalah ketentuan sistem bilangan biner operasi complement.

BIT	~
0	1
1	0

Penempatan operan untuk nilai positif atau negatif.

- ✓ Jika operan bernilai positif, maka operasi complement dilakukan terhadap Unsigned Biner.
- ✓ Jika operan bernilai negatif, maka operasi complement dilakukan terhadap Komplemen-2.

Berikut adalah tabel berisikan operasi complement bilangan biner variabel $x = 1$ dan $y = -2$.

DESIMAL	UNSIGNED BINER	KOMPLEMEN-2	DESIMAL	SIGNED BINER
0	0000 0000	1111 1111	-1	1000 0001
+1	0000 0001	1111 1110	-2	1000 0010

9. PERCOBAAN 9: (OPERATOR BITWISE AND)

9.1. Program Operator Bitwise AND.

```
12 public class OperatorBitwiseAND {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 3; y = 2; z = 4;
16         System.out.println("Operator Bitwise - AND");
17         System.out.println("Nilai x = " + x);
18         System.out.println("Nilai y = " + y);
19         System.out.println("Nilai z = " + z);
20         System.out.println("Hasil Operasi Bitwise: x & y = " + (x & y));
21         System.out.println("Hasil Operasi Bitwise: x & z = " + (x & z));
22         System.out.println("Hasil Operasi Bitwise: y & z = " + (y & z));
23     }
24 }
25
```

Source Code Program 9.1.


```

run:
Operator Bitwise - AND
Nilai x = 3
Nilai y = 2
Nilai z = 4
Hasil Operasi Bitwise: x & y = 2
Hasil Operasi Bitwise: x & z = 0
Hasil Operasi Bitwise: y & z = 0
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 9.1.

9.2. Program di modifikasi dengan mengganti nilai variabel x, y dan z dengan nilai yang lain. Adapun nilai variabel **x = 6**, **y = 4**, dan **z = 1**.

```

12 public class OperatorBitwiseAND {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 6; y = 4; z = 1;
16         System.out.println("Operator Bitwise - AND");
17         System.out.println("Nilai x = " + x);
18         System.out.println("Nilai y = " + y);
19         System.out.println("Nilai z = " + z);
20         System.out.println("Hasil Operasi Bitwise: x & y = " + (x & y));
21         System.out.println("Hasil Operasi Bitwise: x & z = " + (x & z));
22         System.out.println("Hasil Operasi Bitwise: y & z = " + (y & z));
23     }
24 }

```

Source Code Program 9.2.

```

run:
Operator Bitwise - AND
Nilai x = 6
Nilai y = 4
Nilai z = 1
Hasil Operasi Bitwise: x & y = 4
Hasil Operasi Bitwise: x & z = 0
Hasil Operasi Bitwise: y & z = 0
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 9.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi AND (dalam bentuk sistem bilangan biner) pada nilai operannya. Dalam pengoperasiannya, bilangan-bilangan pada operan akan di ubah menjadi bilangan biner, yang

kemudian operan-operan tersebut akan di lakukan operasi AND setiap bit-bitnya sesuai dengan sistem bilangan biner operasi AND.

Berikut adalah ketentuan sistem bilangan biner operasi AND.

BIT1	BIT 2	BIT1 & BIT2
1	1	1
1	0	0
0	1	0
0	0	0

Penempatan operan untuk nilai positif atau negatif.

- ✓ Jika operan bernilai positif, maka operasi AND dilakukan terhadap Unsigned Biner.
- ✓ Jika operan bernilai negatif, maka operasi AND dilakukan terhadap Komplemen-2.

Selanjutnya hasil dari operasi AND tersebut akan menghasilkan bilangan biner. Lalu bilangan biner tersebut di ubah menjadi bilangan desimal yaitu hasilnya.

Berikut adalah tabel operasi AND bilangan biner variabel **x = 6**, **y = 4**, dan **z = 1**.

OPERAN 1	UNSIGNED BINER	OPERAN 2	UNSIGNED BINER	BINER OPERAN1 & OPERAN 2	OPERAN1 & OPERAN2
0	0000 0000	1	0000 0001	0000	0
6	0000 0110	4	0000 0100	0100	4
6	0000 0110	1	0000 0001	0000	0
4	0000 0100	1	0000 0001	0000	0

10. PERCOBAAN 10: (OPERATOR BITWISE OR)

10.1. Program Bitwise OR

```

12 public class OperatorBitwiseOR {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 3; y = 2; z = 4;
16         System.out.println("Operator Bitwise - OR");
17         System.out.println("Nilai x = " + x);
18         System.out.println("Nilai y = " + y);
19         System.out.println("Nilai z = " + z);
20         System.out.println("Hasil Operasi Bitwise: x | y = " + (x | y));
21         System.out.println("Hasil Operasi Bitwise: x | z = " + (x | z));
22         System.out.println("Hasil Operasi Bitwise: y | z = " + (y | z));
23     }
24 }
25

```

Source Code Program 10.1.

```

run:
Operator Bitwise - OR
Nilai x = 3
Nilai y = 2
Nilai z = 4
Hasil Operasi Bitwise: x | y = 3
Hasil Operasi Bitwise: x | z = 7
Hasil Operasi Bitwise: y | z = 6
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 10.1.

10.2. Program di modifikasi dengan mengganti variabel x, y dan z dengan nilai yang lain. Adapun nilai variabel **x = 42**, **y = 30**, dan **z = 30**.

```

12 public class OperatorBitwiseOR {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 42; y = 30; z = 30;
16         System.out.println("Operator Bitwise - OR");
17         System.out.println("Nilai x = " + x);
18         System.out.println("Nilai y = " + y);
19         System.out.println("Nilai z = " + z);
20         System.out.println("Hasil Operasi Bitwise: x | y = " + (x | y));
21         System.out.println("Hasil Operasi Bitwise: x | z = " + (x | z));
22         System.out.println("Hasil Operasi Bitwise: y | z = " + (y | z));
23     }
24 }

```

Source Code Program 10.2.

```

run:
Operator Bitwise - OR
Nilai x = 42
Nilai y = 30
Nilai z = 30
Hasil Operasi Bitwise: x | y = 62
Hasil Operasi Bitwise: x | z = 62
Hasil Operasi Bitwise: y | z = 30
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 10.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi OR (dalam bentuk sistem bilangan biner) pada nilai operannya. Dalam pengoperasiannya, bilangan-bilangan pada operan akan di ubah menjadi bilangan biner, yang kemudian operan-operan tersebut akan di lakukan operasi OR setiap bit-bitnya sesuai dengan sistem bilangan biner operasi OR.

Berikut adalah ketentuan sistem bilangan biner operasi OR.

BIT1	BIT 2	BIT1 BIT2
1	1	1
1	0	1
0	1	1
0	0	0

Penempatan operan untuk nilai positif atau negatif.

- ✓ Jika operan bernilai positif, maka operasi OR dilakukan terhadap Unsigned Biner.
- ✓ Jika operan bernilai negatif, maka operasi OR dilakukan terhadap Komplemen-2.

Selanjutnya hasil dari operasi OR tersebut akan menghasilkan bilangan biner. Lalu bilangan biner tersebut di ubah menjadi bilangan desimal yaitu hasilnya.

Berikut adalah tabel operasi OR bilangan biner variabel **x = 42**, **y = 30**, dan **z = 30**.

OPERAN 1	UNSIGNED BINER	OPERAN 2	UNSIGNED BINER	BINER OPERAN1 OPERAN 2	OPERAN1 OPERAN2
0	0000....0000	1	0000....0001	0001	1
42	0000....0101010	30	0000....011110	0111110	62
42	0000....0101010	30	0000....011110	0111110	62
30	0000....011110	30	0000....011110	011110	30

11. PERCOBAAN 11: (OPERATOR BITWISE XOR)

11.1. Program Operator Bitwise XOR.

```

12 public class OperatorBitwiseXOR {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 3; y = 2; z = 4;
16         System.out.println("Operator Bitwise - XOR");
17         System.out.println("Nilai x = " + x);
18         System.out.println("Nilai y = " + y);
19         System.out.println("Nilai z = " + z);
20         System.out.println("Hasil Operasi Bitwise: x ^ y = " + (x ^ y));
21         System.out.println("Hasil Operasi Bitwise: x ^ z = " + (x ^ z));
22         System.out.println("Hasil Operasi Bitwise: y ^ z = " + (y ^ z));
23     }
24 }
25

```

Source Code Program 11.1.

```

run:
Operator Bitwise - XOR
Nilai x = 3
Nilai y = 2
Nilai z = 4
Hasil Operasi Bitwise: x ^ y = 1
Hasil Operasi Bitwise: x ^ z = 7
Hasil Operasi Bitwise: y ^ z = 6
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 11.1.

11.2. Program di modifikasi dengan mengganti nilai variabel x, y, dan z dengan nilai yang lain.

```

12 public class OperatorBitwiseXOR {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 2; y = 9; z = 4;
16         System.out.println("Operator Bitwise - XOR");
17         System.out.println("Nilai x = " + x);
18         System.out.println("Nilai y = " + y);
19         System.out.println("Nilai z = " + z);
20         System.out.println("Hasil Operasi Bitwise: x ^ y = " + (x ^ y));
21         System.out.println("Hasil Operasi Bitwise: x ^ z = " + (x ^ z));
22         System.out.println("Hasil Operasi Bitwise: y ^ z = " + (y ^ z));
23     }
24 }

```

Source Code Program 11.2.

```

run:
Operator Bitwise - XOR
Nilai x = 2
Nilai y = 9
Nilai z = 4
Hasil Operasi Bitwise: x ^ y = 11
Hasil Operasi Bitwise: x ^ z = 6
Hasil Operasi Bitwise: y ^ z = 13
BUILD SUCCESSFUL (total time: 2 seconds)

```

Output Program 11.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi XOR (dalam bentuk sistem bilangan biner) pada nilai operannya. Dalam pengoperasiannya, bilangan-bilangan pada operan akan di ubah menjadi bilangan biner, yang kemudian operan-operan tersebut akan di lakukan operasi XOR setiap bit-bitnya sesuai dengan sistem bilangan biner operasi XOR.

Berikut adalah ketentuan sistem bilangan biner operasi XOR.

BIT1	BIT 2	BIT1 ^ BIT2
1	1	0
1	0	1
0	1	1
0	0	0

Selanjutnya hasil dari operasi XOR tersebut akan menghasilkan bilangan biner. Lalu bilangan biner tersebut di ubah menjadi bilangan desimal yaitu hasilnya.

Berikut adalah tabel operasi XOR bilangan biner variabel $x = 2$, $y = 9$, dan $z = 4$.

OPERAN 1	UNSGNED BINER	OPERAN 2	UNSIGNED BINER	BINER OPERAN1 ^ OPERAN 2	OPERAN1 ^ OPERAN2
0	0000 0000	1	0000 0001	0000	0
2	0000 0010	9	0000 01001	01011	11
2	0000 0010	4	0000 0100	00110	6
9	0000 1001	4	0000 0100	01101	13

12. PERCOBAAN 12: (OPERATOR BITWISE SHIFT)

12.1. Program Operator Bitwise Shift.

```

12 public class OperatorSHIFT {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 9; y = 2; z = -1;
16         System.out.println("Operator SHIFT");
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x = " + x);
19         System.out.println("Nilai y = " + y);
20         System.out.println("Nilai z = " + z);
21         System.out.println("Hasil Operasi Right Shift: x >> 1 = " + (x >> 1));
22         System.out.println("Hasil Operasi Left Shift: y << 1 = " + (y << 1));
23         System.out.println("Hasil Operasi Unsigned Right Shift: z >>> 30 = " + (z >>> 30));
24     }
25 }

```

Source Code Program 12.1.

```

run:
Operator SHIFT
Nilai Awal
Nilai x = 9
Nilai y = 2
Nilai z = -1
Hasil Operasi Right Shift: x >> 1 = 4
Hasil Operasi Left Shift: y << 1 = 4
Hasil Operasi Unsigned Right Shift: z >>> 30 = 3
BUILD SUCCESSFUL (total time: 2 seconds)

```

Output Program 12.1.

12.2. Program di modifikasi dengan mengganti nilai variabel x, y, dan z dengan nilai yang lain.

```

12 public class OperatorSHIFT {
13     public static void main(String args[]){
14         int x, y, z;
15         x = 7; y = 3; z = 1;
16         System.out.println("Operator SHIFT");
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x = " + x);
19         System.out.println("Nilai y = " + y);
20         System.out.println("Nilai z = " + z);
21         System.out.println("Hasil Operasi Right Shift: x >> 1 = " + (x >> 1));
22         System.out.println("Hasil Operasi Left Shift: y << 1 = " + (y << 1));
23         System.out.println("Hasil Operasi Unsigned Right Shift: z >>> 30 = " + (z >>> 30));
24     }
25 }

```

Source Code Program 12.2.

```

run:
Operator SHIFT
Nilai Awal
Nilai x = 7
Nilai y = 3
Nilai z = 1
Hasil Operasi Right Shift: x >> 1 = 3
Hasil Operasi Left Shift: y << 1 = 6
Hasil Operasi Unsigned Right Shift: z >>> 30 = 0
BUILD SUCCESSFUL (total time: 1 second)

```

Output Program 12.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi pergeseran bit (dalam bentuk sistem bilangan biner). Pada program terdapat 3 pergeseran bit yaitu **RIGHT SHIFT(>>)**, **LEFT SHIFT(<<)**, dan **UNSIGNED RIGHT SHIFT(>>>)**.

Adapun fungsi-fungsi dari operasi pergeseran bit pada program tersebut antara lain:

- ✓ **RIGHT SHIFT(>>)** adalah operator yang digunakan untuk melakukan operasi pergeseran bit ke kanan (dalam bentuk sistem bilangan biner) pada nilai operannya yang mana signed bit akan diisi dengan nilai signed bit yang sebelumnya.

OPERAN1	BINER	OPERAN2 (JUMLAH BIT PERGESERAN)	OPERAN1 >> OPERAN2
7	0000 0111	1	3

- ✓ **LEFT SHIFT(<<)** adalah operator yang digunakan untuk melakukan operasi pergeseran bit ke kiri (dalam bentuk sistem bilangan biner) pada nilai operannya yang mana signed bit akan diisi dengan nilai signed bit yang sebelumnya.

OPERAN1	BINER	OPERAN2 (JUMLAH BIT PERGESERAN)	OPERAN1 << OPERAN2
3	0000 0011	1	6

- ✓ **UNSIGNED RIGHT SHIFT(>>>)** adalah operator yang digunakan untuk melakukan operasi pergeseran bit ke kanan (dalam bentuk sistem bilangan biner) pada nilai operannya yang mana signed bit akan diisi dengan nilai 0.

OPERAN1	SIGNED BINER	OPERAN2 (JUMLAH BIT PERGESERAN)	OPERAN1 >>> OPERAN2
1	0 000 0001	30	0

13. PERCOBAAN 13: (OPERATOR KONKATENASI STRING)

13.1. Program Operator Konkatenasi String.

```

12 public class OperatorKonkatenasiString {
13     public static void main(String args[]){
14         System.out.println("Operator Konkatenasi String (+)");
15         System.out.println("\"Andi\" + \"Budiman\" = " + ("Andi" + "Budiman"));
16         System.out.println("\"Andi \" + \"Wijaya\" = " + ("Andi " + "Wijaya"));
17         System.out.println("\"Nilai:\" + 3 = " + ("Nilai:" + 3));
18         System.out.println("\"3\" + 3 = " + ("3" + 3));
19         System.out.println("3 + \"4\" = " + (3 + "4"));
20     }
21 }
22

```

Source Code Program 13.1.

```

run:
Operator Konkatenasi String (+)
"Andi" + "Budiman" = AndiBudiman
"Andi " + "Wijaya" = Andi Wijaya
"Nilai:" + 3 = Nilai:3
"3" + 3 = 33
3 + "4" = 34
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 13.1.

13.2. Program di modifikasi dengan mengganti nilai String atau angka yang digunakan dengan nilai yang lain. Adapun nilai angka **3** pada baris 17 sampai baris 19 di ubah jadi angka **6**, dan nilai angka 4 pada baris 19 di ubah jadi angka **5**.

```
12 public class OperatorKonkatenasiString {
13     public static void main(String args[]){
14         System.out.println("Operator Konkatenasi String (+)");
15         System.out.println("\"Andi\" + \"Budiman\" = " + ("Andi" + "Budiman"));
16         System.out.println("\"Andi \" + \"Wijaya\" = " + ("Andi " + "Wijaya"));
17         System.out.println("\"Nilai:\" + 6 = " + ("Nilai:" + 6));
18         System.out.println("\"6\" + 6 = " + ("6" + 6));
19         System.out.println("6 + \"5\" = " + (5 + "6"));
20     }
21 }
```

Source Code Program 13.2.

```
run:
Operator Konkatenasi String (+)
"Andi" + "Budiman" = AndiBudiman
"Andi " + "Wijaya" = Andi Wijaya
"Nilai:" + 6 = Nilai:6
"6" + 6 = 66
6 + "5" = 56
BUILD SUCCESSFUL (total time: 0 seconds)
```

Output Program 13.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah operator yang digunakan untuk melakukan operasi penggabungan antara:

- ✓ Nilai bertipe data String dengan nilai bertipe data String.
- ✓ Nilai bertipe data String dengan nilai bertipe data primitif (byte, short, int, long, float, double, char, boolean).

14. PERCOBAAN 14: (OPERATOR KONDISIONAL)

14.1. Program Operator Kondisional

```

12 public class OperatorKondisional {
13     public static void main(String args[]){
14         int x, y, z;
15         String hasil;
16         x = 3; y = 2; z = 3;
17         hasil = "Belum Diketahui";
18         System.out.println("Nilai Awal");
19         System.out.println("x = " + x);
20         System.out.println("y = " + y);
21         System.out.println("z = " + z);
22         System.out.println("hasil = " + hasil);
23         System.out.println("Operator Kondisional");
24         System.out.println("Contoh 1:");
25         System.out.println("hasil = (x==y) ? \"Benar\" : \"Salah\""); hasil = (x==y) ? "Benar" : "Salah";
26         System.out.println("Nilai variabel hasil = " + hasil); System.out.println("Contoh 2:");
27         System.out.println("hasil = (x==z) ? \"Benar\" : \"Salah\""); hasil = (x==z) ? "Benar" : "Salah";
28         System.out.println("Nilai variabel hasil = " + hasil);
29     }
30 }

```

Source Code Program 14.1.

```

run:
Nilai Awal
x = 3
y = 2
z = 3
hasil = Belum Diketahui
Operator Kondisional
Contoh 1:
hasil = (x==y) ? "Benar" : "Salah"
Nilai variabel hasil = Salah
Contoh 2:
hasil = (x==z) ? "Benar" : "Salah"
Nilai variabel hasil = Benar
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 14.1.

14.2. Program di modifikasi dengan mengganti nilai variabel x, y, dan z dengan nilai yang lain serta ganti pula operasi relasional atau logika. Adapun nilai variabel x = 5, y = 2, dan z = 2. Sedangkan operasi relasionalnya adalah > (lebih dari).

```

12 public class OperatorKondisional {
13     public static void main(String args[]){
14         int x, y, z;
15         String hasil;
16         x = 5; y = 2; z = 2;
17         hasil = "Belum Diketahui";
18         System.out.println("Nilai Awal");
19         System.out.println("x = " + x);
20         System.out.println("y = " + y);
21         System.out.println("z = " + z);
22         System.out.println("hasil = " + hasil);
23         System.out.println("Operator Kondisional");
24         System.out.println("Contoh 1:");
25         System.out.println("hasil = (x > y) ? \"Benar\" : \"Salah\""); hasil = (x > y) ? "Benar" : "Salah";
26         System.out.println("Nilai variabel hasil = " + hasil); System.out.println("Contoh 2:");
27         System.out.println("hasil = (x > z) ? \"Benar\" : \"Salah\""); hasil = (x > z) ? "Benar" : "Salah";
28         System.out.println("Nilai variabel hasil = " + hasil);
29     }
30 }

```

Source Code Program 14.2.

```

run:
Nilai Awal
x = 5
y = 2
z = 2
hasil = Belum Diketahui
Operator Kondisional
Contoh 1:
hasil = (x > y) ? "Benar" : "Salah"
Nilai variabel hasil = Benar
Contoh 2:
hasil = (x > z) ? "Benar" : "Salah"
Nilai variabel hasil = Benar
BUILD SUCCESSFUL (total time: 1 second)

```

Output Program 14.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan penyerdehanaan struktur kontrol percabangan if-then-else. Untuk aturan penulisan penyerdehaannya seperti berikut ini.

variabel = Operan1 ? Operan2 : Operan3

15. PERCOBAAN 15: (OPERATOR CASTING)

15.1. Program Operator Casting

```

12 public class OperatorCasting {
13     public static void main(String args[]){
14         int x = 10;
15         int y = 128;
16         byte a = 0;
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x (bertipe int)  = " + x);
19         System.out.println("Nilai y (bertipe int)  = " + y);
20         System.out.println("Nilai a (bertipe byte) = " + a);
21         System.out.println("Operator Casting Tipe Data");
22         System.out.println("Contoh 1:");
23         System.out.println("a = (byte) x");
24         a = (byte) x;
25         System.out.println("Nilai a = " + a);
26         System.out.println("Contoh 2:");
27         System.out.println("a = (byte) y");
28         a = (byte) y;
29         System.out.println("Nilai a = " + a);
30     }
31 }

```

Source Code Program 15.1.

```

run:
Nilai Awal
Nilai x (bertipe int)  = 10
Nilai y (bertipe int)  = 128
Nilai a (bertipe byte) = 0
Operator Casting Tipe Data
Contoh 1:
a = (byte) x
Nilai a = 10
Contoh 2:
a = (byte) y
Nilai a = -128
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 15.1.

15.2. Program di modifikasi dengan mengganti nilai variabel x, y, dan a dengan nilai yang lain. Adapun nilai variabel **a = 20**, **y = 121**, dan **a = 2**.

```

12 public class OperatorCasting {
13     public static void main(String args[]){
14         int x = 20;
15         int y = 121;
16         byte a = 2;
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x (bertipe int) = " + x);
19         System.out.println("Nilai y (bertipe int) = " + y);
20         System.out.println("Nilai a (bertipe byte) = " + a);
21         System.out.println("Operator Casting Tipe Data");
22         System.out.println("Contoh 1:");
23         System.out.println("a = (byte) x");
24         a = (byte) x;
25         System.out.println("Nilai a = " + a);
26         System.out.println("Contoh 2:");
27         System.out.println("a = (byte) y");
28         a = (byte) y;
29         System.out.println("Nilai a = " + a);
30     }
31 }

```

Source Code Program 15.2.

```

run:
Nilai Awal
Nilai x (bertipe int) = 20
Nilai y (bertipe int) = 121
Nilai a (bertipe byte) = 2
Operator Casting Tipe Data
Contoh 1:
a = (byte) x
Nilai a = 20
Contoh 2:
a = (byte) y
Nilai a = 121
BUILD SUCCESSFUL (total time: 0 seconds)

```

Output Program 15.2.

ANALISA HASIL PROGRAM

Hasil program tersebut adalah melakukan operasi konversi tipe data dari sebuah nilai ke nilai dengan tipe data yang lainnya. Pada program nilai **x** tipe datanya **int**, kemudian tipe data **x** di ubah menjadi tipe data **byte** dengan menggunakan perintah **(tipe data) nama variabel**.

15.3. Program di modifikasi dengan menambahkan potongan program berikut ini:

a = x;

a = y;

```
12 public class OperatorCasting {
13     public static void main(String args[]){
14         int x = 20;
15         int y = 121;
16         byte a = 2;
17         System.out.println("Nilai Awal");
18         System.out.println("Nilai x (bertipe int) = " + x);
19         System.out.println("Nilai y (bertipe int) = " + y);
20         System.out.println("Nilai a (bertipe byte) = " + a);
21         System.out.println("Operator Casting Tipe Data");
22         System.out.println("Contoh 1:");
23         System.out.println("a = (byte) x");
24         a = x;
25         System.out.println("Nilai a = " + a);
26         System.out.println("Contoh 2:");
27         System.out.println("a = (byte) y");
28         a = y;
29         System.out.println("Nilai a = " + a);
30     }
31 }
```

Source Code Program 15.3.

```
run:
Nilai Awal
Nilai x (bertipe int) = 20
Nilai y (bertipe int) = 121
Nilai a (bertipe byte) = 2
Operator Casting Tipe Data
Contoh 1:
a = (byte) x
Exception in thread "main" java.lang.RuntimeException
    at Laporan2.OperatorCasting.main(OperatorCast
C:\Users\Rezky Yustisio\AppData\Local\NetBeans\Cache\
BUILD FAILED (total time: 3 seconds)
```

Output Program 15.3.

ANALISA HASIL PROGRAM

Pada saat program di jalankan terjadi error di karenakan variabel tipe data tertentu tidak bisa langsung di masukkan ke dalam variabel dengan tipe data berbeda yang telah di definisikan sebelumnya. Cara yang benar adalah dengan cara mengcasting tipe datanya dengan menggunakan perintah **(tipe data) nama variabel**. Sehingga menjadi seperti ini, variabel **a = (byte) x;**

KESIMPULAN

Operator pada bahasa pemrograman, khususnya bahasa pemrograman java sangat membantu para pengembang aplikasi dalam melakukan proses perhitungan, perbandingan maupun pengkonversian tipe data.

DAFTAR PUSTAKA

Suheri. ____, *Operator*, [pptx]