

Laporan Praktikum 2

Dasar – Dasar Pemrograman
Operator Pemrograman



Muhammad Azhar Rasyad
0110217029
Teknik Informatika
Kelas Pagi
Semester 1

Sekolah Tinggi Teknologi Terpadu Nurul Fikri
Tahun 2017

Operator Pemrograman

Operator dalam bahasa C++ dibagi menjadi 7, yaitu sebagai berikut :

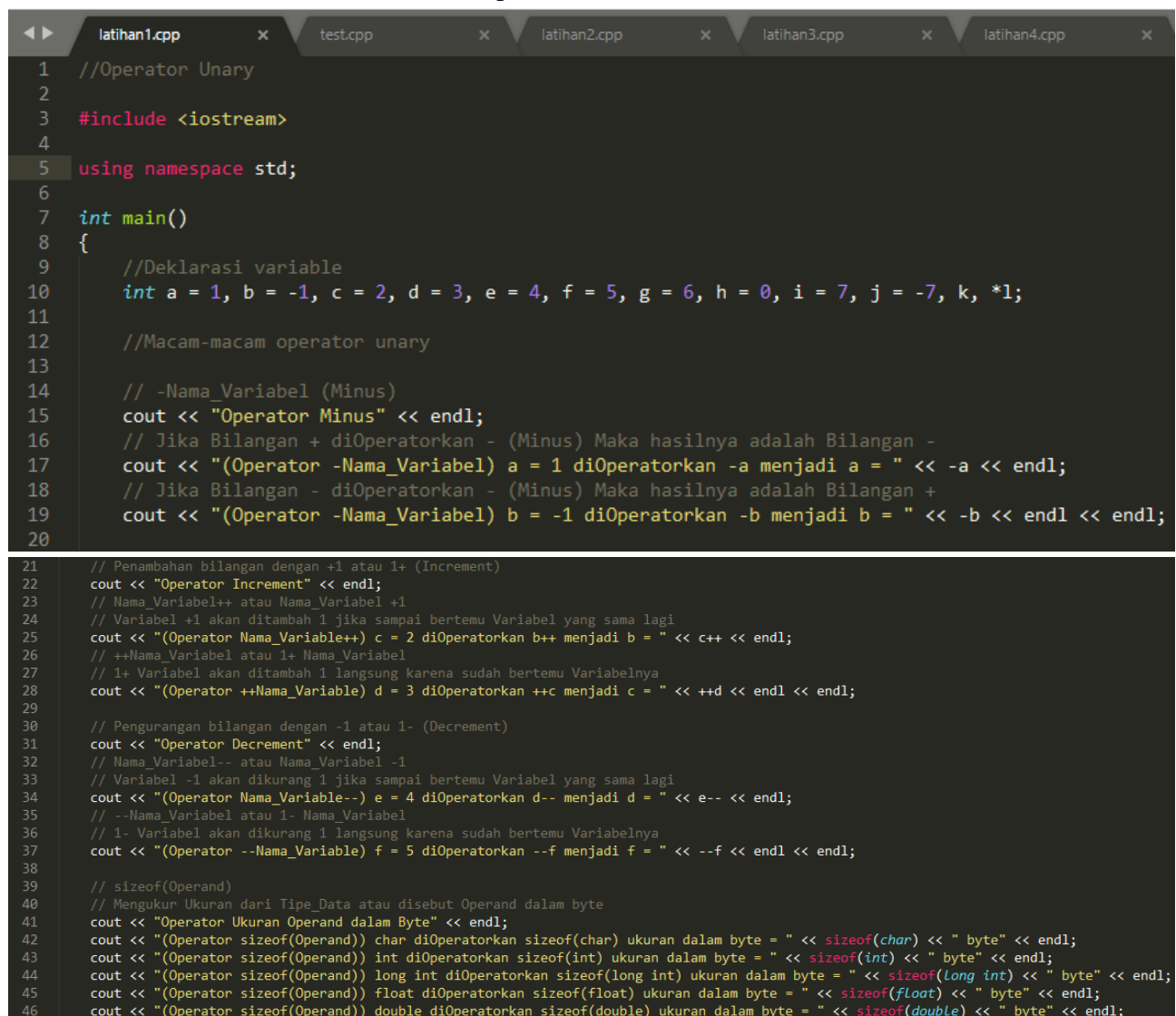
1. Operator Unary
2. Operator Binary
3. Operator Relasional
4. Operator Logika
5. Operator Bitwise
6. Operator Penugasan
7. Operator Ternary

Operand = Objek yang akan dioperasikan oleh operator.

Berikut adalah contoh program dari masing – masing operator dalam bentuk screenshot dan penjelasannya dapat dilihat dalam screenshot tersebut :

1. **Operator Unary** = Pengoperasian 1 operand dengan tipe data angka.

Gambar berikut adalah contoh code dari operator tersebut :



```
1 //Operator Unary
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Deklarasi variable
10    int a = 1, b = -1, c = 2, d = 3, e = 4, f = 5, g = 6, h = 0, i = 7, j = -7, k, *l;
11
12    //Macam-macam operator unary
13
14    // -Nama_Variabel (Minus)
15    cout << "Operator Minus" << endl;
16    // Jika Bilangan + diOperatorkan - (Minus) Maka hasilnya adalah Bilangan -
17    cout << "(Operator -Nama_Variabel) a = 1 diOperatorkan -a menjadi a = " << -a << endl;
18    // Jika Bilangan - diOperatorkan - (Minus) Maka hasilnya adalah Bilangan +
19    cout << "(Operator -Nama_Variabel) b = -1 diOperatorkan -b menjadi b = " << -b << endl << endl;
20
21    // Penambahan bilangan dengan +1 atau 1+ (Increment)
22    cout << "Operator Increment" << endl;
23    // Nama_Variabel++ atau Nama_Variabel +1
24    // Variabel +1 akan ditambah 1 jika sampai bertemu Variabel yang sama lagi
25    cout << "(Operator Nama_Variabel++) c = 2 diOperatorkan b++ menjadi b = " << c++ << endl;
26    // ++Nama_Variabel atau 1+ Nama_Variabel
27    // 1+ Variabel akan ditambah 1 langsung karena sudah bertemu Variabelnya
28    cout << "(Operator ++Nama_Variabel) d = 3 diOperatorkan ++c menjadi c = " << ++d << endl << endl;
29
30    // Pengurangan bilangan dengan -1 atau 1- (Decrement)
31    cout << "Operator Decrement" << endl;
32    // Nama_Variabel-- atau Nama_Variabel -1
33    // Variabel -1 akan dikurang 1 jika sampai bertemu Variabel yang sama lagi
34    cout << "(Operator Nama_Variabel--) e = 4 diOperatorkan d-- menjadi d = " << e-- << endl;
35    // --Nama_Variabel atau 1- Nama_Variabel
36    // 1- Variabel akan dikurang 1 langsung karena sudah bertemu Variabelnya
37    cout << "(Operator --Nama_Variabel) f = 5 diOperatorkan --f menjadi f = " << --f << endl << endl;
38
39    // sizeof(Operand)
40    // Mengukur Ukuran dari Tipe_Data atau disebut Operand dalam byte
41    cout << "Operator Ukuran Operand dalam Byte" << endl;
42    cout << "(Operator sizeof(Operand)) char diOperatorkan sizeof(char) ukuran dalam byte = " << sizeof(char) << " byte" << endl;
43    cout << "(Operator sizeof(Operand)) int diOperatorkan sizeof(int) ukuran dalam byte = " << sizeof(int) << " byte" << endl;
44    cout << "(Operator sizeof(Operand)) long int diOperatorkan sizeof(long int) ukuran dalam byte = " << sizeof(long int) << " byte" << endl;
45    cout << "(Operator sizeof(Operand)) float diOperatorkan sizeof(float) ukuran dalam byte = " << sizeof(float) << " byte" << endl;
46    cout << "(Operator sizeof(Operand)) double diOperatorkan sizeof(double) ukuran dalam byte = " << sizeof(double) << " byte" << endl;
```

```

47     cout << "(Operator sizeof(Operand)) long double diOperatorkan sizeof(long double) ukuran dalam byte = " << sizeof(long double) << " byte" << endl << endl;
48
49     // !Nama_Variabel (Unary NOT)
50     cout << "Operator Unary NOT" << endl;
51     // Jika bilangan bukan 0 maka akan berubah menjadi 0
52     cout << "(Operator !Nama_Variabel) g = 6 diOperatorkan !g menjadi g = " << !g << endl;
53     // Jika bilangan 0 maka akan berubah menjadi 1
54     cout << "(Operator !Nama_Variabel) h = 0 diOperatorkan !h menjadi h = " << !h << endl << endl;
55
56     // ~Nama_Variabel (Bitwise NOT)
57     cout << "Operator Bitwise NOT" << endl;
58     // Jika bilangan + maka bilangan +1 lalu dikalikan -1
59     cout << "(Operator ~Nama_Variabel) i = 7 diOperatorkan ~i menjadi i = " << ~i << endl;
60     // Jika bilangan - maka bilangan +1 lalu dikalikan -1
61     cout << "(Operator ~Nama_Variabel) j = -7 diOperatorkan ~j menjadi j = " << ~j << endl << endl;
62
63     // &Nama_Variabel (Alamat Memori Operand)
64     cout << "Operator Alamat Memori" << endl;
65     // Menghasilkan alamat memori sebuah operand
66     cout << "(Operator &Nama_Variabel) k diOperatorkan &k menjadi k = " << &k << endl << endl;
67
68     // *Nama_Variabel
69     cout << "Operator Nilai Pointer" << endl;
70     // Menghasilkan
71     cout << "(Operator Nama_Variabel) *l diOperatorkan l menjadi l = " << *l << endl;
72 }

```

Compiler dari code tersebut akan seperti gambar berikut :

```

Administrator: C:\Windows\system32\cmd.exe
(Operator Nama_Variable++) c = 2 diOperatorkan b++ menjadi b = 2
(Operator ++Nama_Variable) d = 3 diOperatorkan ++c menjadi c = 4

Operator Decrement
(Operator Nama_Variable--) e = 4 diOperatorkan d-- menjadi d = 4
(Operator --Nama_Variable) f = 5 diOperatorkan --f menjadi f = 4

Operator Ukuran Operand dalam Byte
(Operator sizeof(Operand)) char diOperatorkan sizeof(char) ukuran dalam byte = 1 byte
(Operator sizeof(Operand)) int diOperatorkan sizeof(int) ukuran dalam byte = 4 byte
(Operator sizeof(Operand)) long int diOperatorkan sizeof(long int) ukuran dalam byte = 4 byte
(Operator sizeof(Operand)) float diOperatorkan sizeof(float) ukuran dalam byte = 4 byte
(Operator sizeof(Operand)) double diOperatorkan sizeof(double) ukuran dalam byte = 8 byte
(Operator sizeof(Operand)) long double diOperatorkan sizeof(long double) ukuran dalam byte = 12 byte

Operator Unary NOT
(Operator !Nama_Variabel) g = 6 diOperatorkan !g menjadi g = 0
(Operator !Nama_Variabel) h = 0 diOperatorkan !h menjadi h = 1

Operator Bitwise NOT
(Operator ~Nama_Variabel) i = 7 diOperatorkan ~i menjadi i = -8
(Operator ~Nama_Variabel) j = -7 diOperatorkan ~j menjadi j = 6

Operator Alamat Memori
(Operator &Nama_Variabel) k diOperatorkan &k menjadi k = 0x6afed0

Operator Nilai Pointer
(Operator Nama_Variable) *l diOperatorkan l menjadi l = 0x6afec0
F:\STIT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>

```

2. Operator Binary = Pengoperasian 2 operand dengan tipe data angka.

Gambar berikut adalah contoh code dari operator tersebut :

```
latihan1.cpp x test.cpp x latihan2.cpp x latihan3.cpp
1 //Operator Binary
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Macam-Macam Operasi binary
10
11     // Operasi Tambah Operand1 + Operand2
12     cout << "Operasi Pertambahan" << endl;
13     cout << "Hasil tambah 1 + 2 = " << int(1) + int(2) << endl;
14     cout << "Hasil tambah 1 + -2 = " << int(1) + int(-2) << endl;
15     cout << "Hasil tambah -1 + -2 = " << int(-1) + int(-2) << endl << endl;
16
17     // Operasi Kurang Operand1 - Operand2
18     cout << "Operasi Pengurangan" << endl;
19     cout << "Hasil kurang 1 - 2 = " << int(1) - int(2) << endl;
20     cout << "Hasil kurang -1 - 2 = " << int(-1) - int(2) << endl;
21     cout << "Hasil kurang -1 - -2 = " << int(-1) - int(-2) << endl << endl;
22
23     // Operasi Kali Operand1 * Operand2
24     cout << "Operasi Perkalian" << endl;
25     cout << "Hasil kali 1 * 2 = " << int(1) * int(2) << endl;
26     cout << "Hasil kali -1 * 2 = " << int(-1) * int(2) << endl;
27     cout << "Hasil kali 1 * -2 = " << int(1) * int(-2) << endl;
28     cout << "Hasil kali -1 * -2 = " << int(-1) * int(-2) << endl << endl;
29
30     // Operasi Bagi Operand1 / Operand2
31     cout << "Operasi Pembagian" << endl;
32     cout << "Hasil bagi 2 / 2 = " << float(2) / float(2) << endl;
33     cout << "Hasil bagi -2 / 2 = " << float(-2) / float(2) << endl;
34     cout << "Hasil bagi -2 / -2 = " << float(-2) / float(-2) << endl;
35     cout << "Hasil bagi 1 / 2 = " << float(1) / float(2) << endl << endl;
36
37     // Operasi Modulo Operand1 % Operand2
38     // Hasil bagi yang tidak memiliki nilai koma
39     cout << "Operasi Pemoduloan" << endl;
40     // Jika hasil bagi habis dibagi maka nilainya menjadi 0
41     cout << "Operasi Pemoduloan yang habis dibagi" << endl;
42     cout << "Hasil modulo 1 % 1 = " << int(1) % int(1) << endl;
43     cout << "Hasil modulo -1 % 1 = " << int(-1) % int(1) << endl;
44     cout << "Hasil modulo -1 % -1 = " << int(-1) % int(-1) << endl;
45     // Jika tidak bisa dibagi maka nilainya menjadi nilai yang dibagi bukan pembagi
46     cout << "Operasi Pemoduloan yang tidak bisa dibagi" << endl;
47     cout << "Hasil modulo 2 % 3 = " << int(2) % int(3) << endl;
48     cout << "Hasil modulo -2 % 3 = " << int(-2) % int(3) << endl;
49     cout << "Hasil modulo -2 % -3 = " << int(-2) % int(-3) << endl;
50     // Jika hasil bagi tidak sampai habis maka nilainya menjadi sisa dari hasil yang tidak bisa dibagi
51     cout << "Operasi Pemoduloan yang bisa dibagi tetapi tidak habis" << endl;
52     cout << "Hasil modulo 3 % 2 = " << int(3) % int(2) << endl;
53     cout << "Hasil modulo -3 % 2 = " << int(-3) % int(2) << endl;
54     cout << "Hasil modulo -3 % -2 = " << int(-3) % int(-2) << endl;
55 }
```

Compiler dari code tersebut akan seperti gambar berikut :

```
Administrator: C:\Windows\system32\cmd.exe

Operasi Pertambahan
Hasil tambah 1 + 2 = 3
Hasil tambah 1 + -2 = -1
Hasil tambah -1 + -2 = -3

Operasi Pengurangan
Hasil kurang 1 - 2 = -1
Hasil kurang -1 - 2 = -3
Hasil kurang -1 - -2 = 1

Operasi Perkalian
Hasil kali 1 * 2 = 2
Hasil kali -1 * 2 = -2
Hasil kali 1 * -2 = -2
Hasil kali -1 * -2 = 2

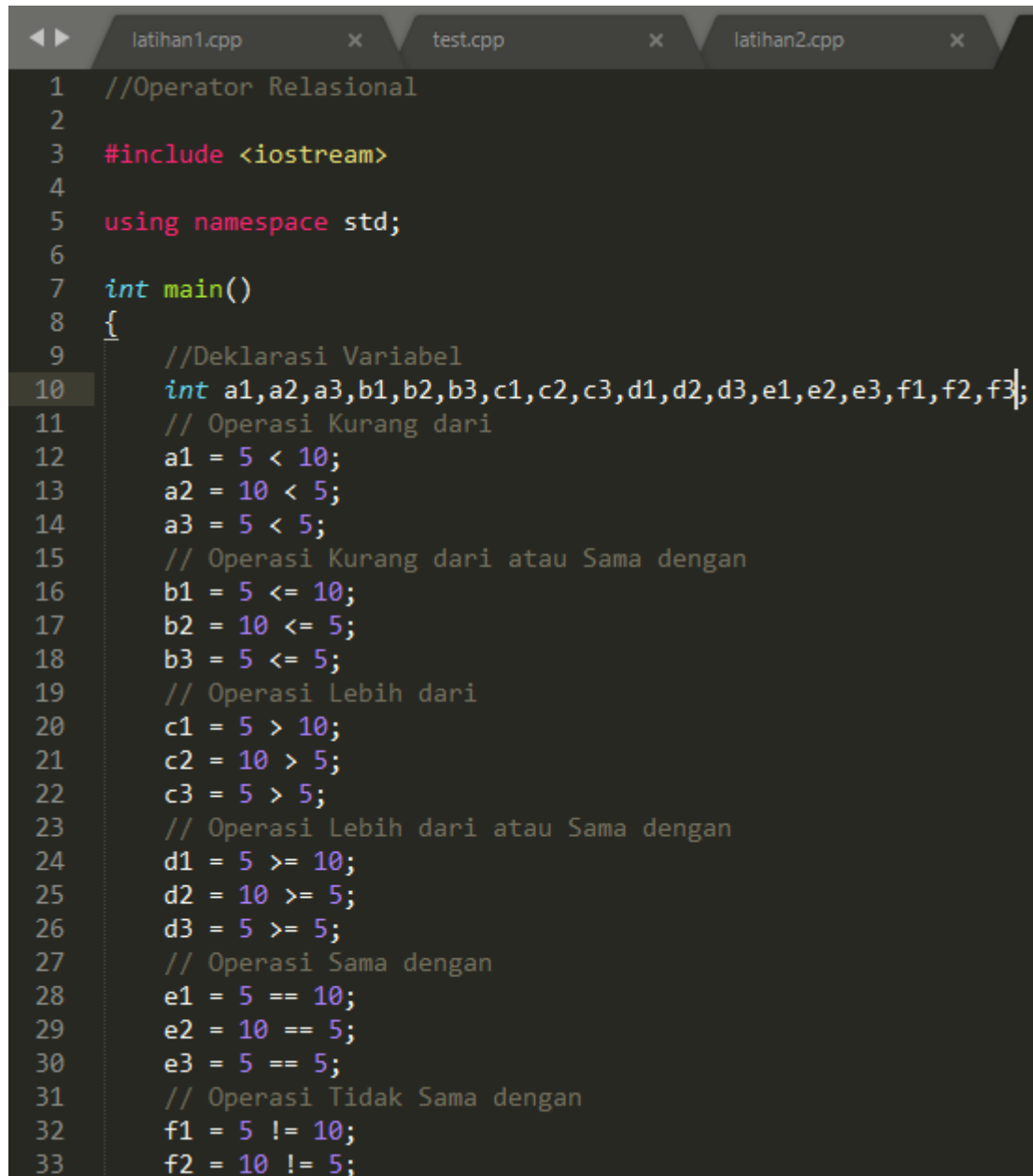
Operasi Pembagian
Hasil bagi 2 / 2 = 1
Hasil bagi -2 / 2 = -1
Hasil bagi -2 / -2 = 1
Hasil bagi 1 / 2 = 0.5

Operasi Pemoduloan
Operasi Pemoduloan yang habis dibagi
Hasil modulo 1 % 1 = 0
Hasil modulo -1 % 1 = 0
Hasil modulo -1 % -1 = 0
Operasi Pemoduloan yang tidak bisa dibagi
Hasil modulo 2 % 3 = 2
Hasil modulo -2 % 3 = -2
Hasil modulo -2 % -3 = -2
Operasi Pemoduloan yang bisa dibagi tetapi tidak habis
Hasil modulo 3 % 2 = 1
Hasil modulo -3 % 2 = -1
Hasil modulo -3 % -2 = -1

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>
```

3. **Operator Relasional** = Perbandingan antar operand dengan tipe data angka atau huruf.

Gambar berikut adalah contoh code dari operator tersebut :

A screenshot of a C++ code editor with three tabs: 'latihan1.cpp', 'test.cpp', and 'latihan2.cpp'. The 'test.cpp' tab is active, showing a C++ program that demonstrates various relational operators. The code includes a comment at the top, a header inclusion, namespace usage, and a main function containing a series of variable declarations and assignments using operators like less than, less than or equal to, greater than, greater than or equal to, equal to, and not equal to. Line numbers 1 through 33 are visible on the left side of the code.

```
1 //Operator Relasional
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Deklarasi Variabel
10    int a1,a2,a3,b1,b2,b3,c1,c2,c3,d1,d2,d3,e1,e2,e3,f1,f2,f3;
11    // Operasi Kurang dari
12    a1 = 5 < 10;
13    a2 = 10 < 5;
14    a3 = 5 < 5;
15    // Operasi Kurang dari atau Sama dengan
16    b1 = 5 <= 10;
17    b2 = 10 <= 5;
18    b3 = 5 <= 5;
19    // Operasi Lebih dari
20    c1 = 5 > 10;
21    c2 = 10 > 5;
22    c3 = 5 > 5;
23    // Operasi Lebih dari atau Sama dengan
24    d1 = 5 >= 10;
25    d2 = 10 >= 5;
26    d3 = 5 >= 5;
27    // Operasi Sama dengan
28    e1 = 5 == 10;
29    e2 = 10 == 5;
30    e3 = 5 == 5;
31    // Operasi Tidak Sama dengan
32    f1 = 5 != 10;
33    f2 = 10 != 5;
```

```

34     f3 = 5 != 5;
35
36     //Macam-Macam operator relasional
37
38     // Jika Relasi benar maka yang keluar adalah nilai 1 (True)
39     // Jika Relasi salah maka yang keluar adalah nilai 0 (False)
40
41     // Kurang dari <
42     // Operand1 < Operand2
43     cout << "Operasi kurang dari ( < )" << endl;
44     cout << "Relasi 5 < 10 = " << a1 << endl;
45     cout << "Relasi 10 < 5 = " << a2 << endl;
46     cout << "Relasi 5 < 5 = " << a3 << endl << endl;
47
48     // Kurang dari atau Sama dengan <=
49     // Operand1 <= Operand2
50     cout << "Operasi kurang dari atau sama dengan ( < )" << endl;
51     cout << "Relasi 5 <= 10 = " << b1 << endl;
52     cout << "Relasi 10 <= 5 = " << b2 << endl;
53     cout << "Relasi 5 <= 5 = " << b3 << endl << endl;
54
55     // Lebih dari >
56     // Operand1 > Operand2
57     cout << "Operasi lebih dari ( > )" << endl;
58     cout << "Relasi 5 > 10 = " << c1 << endl;
59     cout << "Relasi 10 > 5 = " << c2 << endl;
60     cout << "Relasi 5 > 5 = " << c3 << endl << endl;
61
62     // Lebih dari atau Sama dengan >=
63     // Operand1 >= Operand2
64     cout << "Operasi lebih dari atau sama dengan ( > )" << endl;
65     cout << "Relasi 5 >= 10 = " << d1 << endl;
66     cout << "Relasi 10 >= 5 = " << d2 << endl;
67     cout << "Relasi 5 >= 5 = " << d3 << endl << endl;
68
69     // Sama dengan ==
70     // Operand1 == Operand2
71     cout << "Operasi sama dengan ( == )" << endl;
72     cout << "Relasi 5 == 10 = " << e1 << endl;
73     cout << "Relasi 10 == 5 = " << e2 << endl;
74     cout << "Relasi 5 == 5 = " << e3 << endl << endl;
75
76     // Tidak Sama dengan !=
77     // Operand1 != Operand2
78     cout << "Operasi tidak sama dengan ( != )" << endl;
79     cout << "Relasi 5 != 10 = " << f1 << endl;
80     cout << "Relasi 10 != 5 = " << f2 << endl;
81     cout << "Relasi 5 != 5 = " << f3 << endl << endl;
82 }

```

Compiler dari code tersebut akan seperti gambar berikut :

Administrator: C:\Windows\system32\cmd.exe

Operasi kurang dari (<)

Relasi $5 < 10 = 1$

Relasi $10 < 5 = 0$

Relasi $5 < 5 = 0$

Operasi kurang dari atau sama dengan (<=)

Relasi $5 <= 10 = 1$

Relasi $10 <= 5 = 0$

Relasi $5 <= 5 = 1$

Operasi lebih dari (>)

Relasi $5 > 10 = 0$

Relasi $10 > 5 = 1$

Relasi $5 > 5 = 0$

Operasi lebih dari atau sama dengan (>=)

Relasi $5 >= 10 = 0$

Relasi $10 >= 5 = 1$

Relasi $5 >= 5 = 1$

Operasi sama dengan (==)

Relasi $5 == 10 = 0$

Relasi $10 == 5 = 0$

Relasi $5 == 5 = 1$

Operasi tidak sama dengan (!=)

Relasi $5 != 10 = 1$

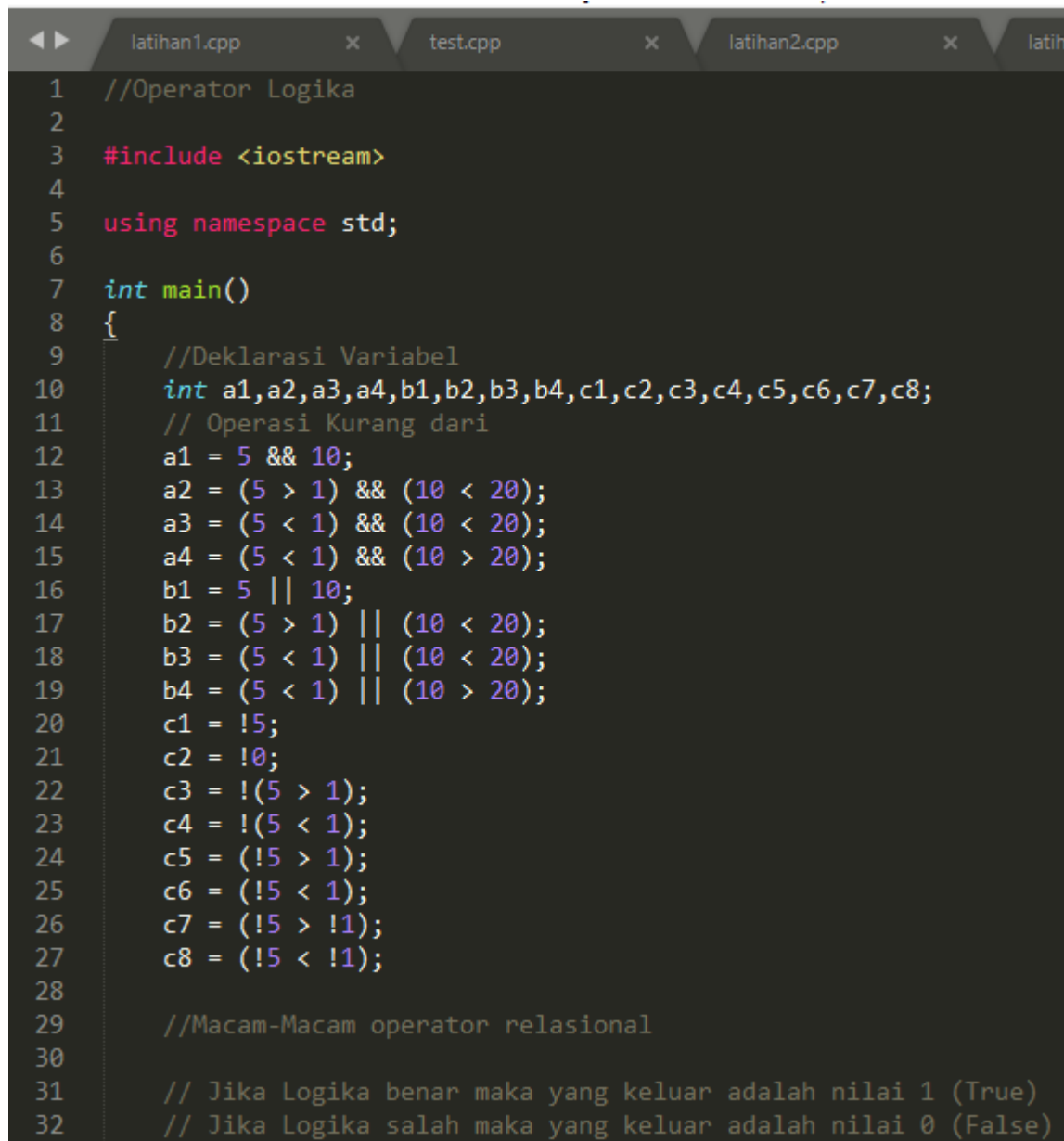
Relasi $10 != 5 = 1$

Relasi $5 != 5 = 0$

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>

4. **Operator Logika** = Perbandingan antar operand dengan tipe data angka atau huruf untuk menentukan suatu kondisi secara logika.

Gambar berikut adalah contoh code dari operator tersebut :



```
1 //Operator Logika
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Deklarasi Variabel
10    int a1,a2,a3,a4,b1,b2,b3,b4,c1,c2,c3,c4,c5,c6,c7,c8;
11    // Operasi Kurang dari
12    a1 = 5 && 10;
13    a2 = (5 > 1) && (10 < 20);
14    a3 = (5 < 1) && (10 < 20);
15    a4 = (5 < 1) && (10 > 20);
16    b1 = 5 || 10;
17    b2 = (5 > 1) || (10 < 20);
18    b3 = (5 < 1) || (10 < 20);
19    b4 = (5 < 1) || (10 > 20);
20    c1 = !5;
21    c2 = !0;
22    c3 = !(5 > 1);
23    c4 = !(5 < 1);
24    c5 = (!5 > 1);
25    c6 = (!5 < 1);
26    c7 = (!5 > !1);
27    c8 = (!5 < !1);
28
29    //Macam-Macam operator relasional
30
31    // Jika Logika benar maka yang keluar adalah nilai 1 (True)
32    // Jika Logika salah maka yang keluar adalah nilai 0 (False)
```

```

33
34 // Operasi AND ( && )
35 // Operand1 && Operand2
36 // Jika kedua operand bernilai 1 (True) maka hasilnya 1 (True)
37 // Jika hanya salah satu 1 (True) maka hasilnya 0 (False)
38 // Jika keduanya 0 (False) maka hasilnya 0 (False)
39 cout << "Operasi AND ( && )" << endl;
40 cout << "Logika 5 && 10 = " << a1 << endl;
41 cout << "Logika (5 > 1) && (10 < 20) = " << a2 << endl;
42 cout << "Logika (5 < 1) && (10 < 20) = " << a3 << endl;
43 cout << "Logika (5 < 1) && (10 > 20) = " << a4 << endl << endl;
44
45 // Operasi OR ( || )
46 // Operand1 || Operand2
47 // Jika kedua operand bernilai 1 (True) maka hasilnya 1 (True)
48 // Jika hanya salah satu 1 (True) maka hasilnya 1 (True)
49 // Jika keduanya 0 (False) maka hasilnya 0 (False)
50 cout << "Operasi OR ( || )" << endl;
51 cout << "Logika 5 && 10 = " << b1 << endl;
52 cout << "Logika (5 > 1) && (10 < 20) = " << b2 << endl;
53 cout << "Logika (5 < 1) && (10 < 20) = " << b3 << endl;
54 cout << "Logika (5 < 1) && (10 > 20) = " << b4 << endl << endl;
55
56 // Operasi NOT ( ! )
57 // !Operand
58 // Jika operand 1 (True) dioperasikan ! (NOT) maka hasilnya 0 (False)
59 // Jika operand 0 (False) dioperasikan ! (NOT) maka hasilnya 1 (True)
60 cout << "Operasi NOT ( ! )" << endl;
61 cout << "Logika !5 = " << c1 << endl;
62 cout << "Logika !0 = " << c2 << endl;
63 cout << "Logika !(5 > 1) = " << c3 << endl;
64 cout << "Logika !(5 < 1) = " << c4 << endl;
65 cout << "Logika (!5 > 1) = " << c5 << endl;
66 cout << "Logika (!5 < 1) = " << c6 << endl;
67 cout << "Logika (!5 > !1) = " << c7 << endl;
68 cout << "Logika (!5 < !1) = " << c8 << endl;
69 }

```

Compiler dari code tersebut akan seperti gambar berikut :

```
Administrator: C:\Windows\system32\cmd.exe
Operasi AND ( && )
Logika 5 && 10 = 1
Logika (5 > 1) && (10 < 20) = 1
Logika (5 < 1) && (10 < 20) = 0
Logika (5 < 1) && (10 > 20) = 0

Operasi OR ( || )
Logika 5 && 10 = 1
Logika (5 > 1) && (10 < 20) = 1
Logika (5 < 1) && (10 < 20) = 1
Logika (5 < 1) && (10 > 20) = 0

Operasi NOT ( ! )
Logika !5 = 0
Logika !0 = 1
Logika !(5 > 1) = 0
Logika !(5 < 1) = 1
Logika (!5 > 1) = 0
Logika (!5 < 1) = 1
Logika (!5 > !1) = 0
Logika (!5 < !1) = 0

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>_
```

5. Operator Bitwise = Memanipulasi bit dengan tipe data angka.

Gambar berikut adalah contoh code dari operator tersebut :

```
latihan1.cpp x latihan2.cpp x latihan3.cpp x latihan4.cpp x latihan5.cpp x latihan6.cpp x
1 //Operator Bitwise
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Deklarasi variable
10    int a1,a2,a3,a4,a5,b1,b2,b3,b4,b5,c1,c2,c3,c4,c5,c6,d1,d2,d3,d4,d5,d6,e1,e2,e3,e4,e5,e6,f1,f2,f3,f4,f5,f6,f7,f8,f9;
11
12    // Operasi Shift Kiri
13    a1 = 1 << 1;
14    a2 = 9 << 3;
15    a3 = 20 << 4;
16    a4 = 0 << 10;
17    a5 = 5 << 0;
18    // Operasi Shift Kanan
19    b1 = 1 >> 1;
20    b2 = 72 >> 3;
21    b3 = 320 >> 4;
22    b4 = 10 >> 5;
23    b5 = 5 >> 0;
24    // Operasi Bit AND
25    c1 = 1 & 17;
26    c2 = 15 & 20;
27    c3 = 4 & 0;
28    c4 = 0 & 7;
29    c5 = 3 & -2;
30    c6 = -1 & 3;
31    // Operasi Bit XOR
32    d1 = 5 ^ 13;
33    d2 = 15 ^ 20;
34
35    d3 = 4 ^ 0;
36    d4 = 0 ^ 7;
37    d5 = 4 ^ -2;
38    d6 = -1 ^ 8;
39    // Operasi Bit OR
40    e1 = 1 | 1;
41    e2 = 15 | 20;
42    e3 = 4 | 0;
43    e4 = 0 | 7;
44    e5 = 4 | -2;
45    e6 = -1 | 8;
46    // Operasi Bit NOT
47    f1 = ~0;
48    f2 = ~1;
49    f3 = ~2;
50    f4 = ~3;
51    f5 = ~0;
52    f6 = ~1;
53    f7 = ~2;
54    f8 = ~3;
55    f9 = ~4;
56
57    //Macam-Macam operator bitwise
58
59    // Operasi Geser Bit ke Kiri atau Shift Kiri ( << )
60    // Menggeser bit ke kiri dari bit 1 yang paling kiri sebanyak bilangan yang dimanipulasi
61    // Jika digeser ke paling kiri maka bit yang sebelumnya bernilai bit 1 diganti dengan bit 0 dan bit 1 berpindah sesuai nilai manipulasi
62    // Nilainya akan bertambah jika digeser ke kiri
63    // Hasil bitnya akan dihitung ulang setelah digeser ke kiri
64    // Bilangan negatif tidak bisa digeser kiri
65    cout << "Operasi Shift Kiri ( << )" << endl;
66    cout << "Bitwise 1 << 1 = " << a1 << " (0000 0001 << 1 = 0000 0010)" << endl;
67    cout << "Bitwise 9 << 3 = " << a2 << " (0000 1001 << 3 = 0100 1000)" << endl;
```

```

67 cout << "Bitwise 20 << 4 = " << a3 << " (0001 0100 << 4 = 0001 0100 0000)" << endl;
68 cout << "Bitwise 0 << 10 = " << a4 << " (0000 0000 << 10 = 0000 0000)" << endl;
69 cout << "Bitwise 5 << 0 = " << a5 << " (0000 0101 << 0 = 0000 0101)" << endl << endl;
70
71 // Operasi Geser Bit ke Kanan atau Shift Kanan( >> )
72 // Menggeser bit ke kanan dari bit 1 yang paling kanan sebanyak bilangan yang dimanipulasi
73 // Jika digeser ke paling kanan maka bitnya terhapus meskipun itu bit 1 karena berpindah sesuai nilai manipulasi
74 // Nilainya akan berkurang jika digeser ke kanan
75 // Hasil bitnya akan dihitung ulang setelah digeser ke kanan
76 // Bilangan negatif tidak bisa digeser kanan
77 cout << "Operasi Shift Kanan ( >> )" << endl;
78 cout << "Bitwise 1 >> 1 = " << b1 << " (0000 0001 >> 1 = 0000 0000)" << endl;
79 cout << "Bitwise 72 >> 3 = " << b2 << " (0100 1000 >> 3 = 0000 1001)" << endl;
80 cout << "Bitwise 320 >> 4 = " << b3 << " (0001 0100 0000 >> 4 = 0001 0100)" << endl;
81 cout << "Bitwise 10 >> 5 = " << b4 << " (0000 1010 >> 10 = 0000 0000)" << endl;
82 cout << "Bitwise 5 >> 0 = " << b5 << " (0000 0101 >> 0 = 0000 0101)" << endl << endl;
83
84 // Operasi Bit AND ( & )
85 // Jika kedua nilainya bit 1 maka hasilnya bit 1
86 // Jika salah satu bit 1 dan satunya lagi bit 0 maka hasilnya bit 0
87 // Jika kedua nilainya bit 0 maka hasilnya bit 0
88 cout << "Operasi Bit AND ( & )" << endl;
89 cout << "Bitwise 1 & 1 = " << c1 << " (0000 0001 & 0000 0001 = 0000 0001)" << endl;
90 cout << "Bitwise 15 & 20 = " << c2 << " (0000 1111 & 0001 0100 = 0000 0100)" << endl;
91 cout << "Bitwise 4 & 0 = " << c3 << " (0000 0100 & 0000 0000 = 0000 0000)" << endl;
92 cout << "Bitwise 0 & 7 = " << c4 << " (0000 0000 & 0000 0111 = 0000 0000)" << endl;
93 cout << "Bitwise 3 & -2 = " << c5 << " (0000 0011 & -1111 1110 = 0000 0010)" << endl;
94 cout << "Bitwise -1 & 3 = " << c6 << " (-1111 1111 & 0000 0011 = 0000 0011)" << endl << endl;
95
96 // Operasi Bit XOR ( ^ )
97 // Jika kedua nilainya bit 1 maka hasilnya bit 0
98 // Jika salah satu bit 1 dan salah satunya lagi bit 0 maka hasilnya 1
99 // Jika kedua nilainya bit 0 maka hasilnya bit 0

```

```

100 cout << "Operasi Bit XOR ( ^ )" << endl;
101 cout << "Bitwise 5 ^ 13 = " << d1 << " (0000 0101 & 0000 1101 = 0000 1000)" << endl;
102 cout << "Bitwise 15 ^ 20 = " << d2 << " (0000 1111 & 0001 0100 = 0001 1011)" << endl;
103 cout << "Bitwise 4 ^ 0 = " << d3 << " (0000 0100 & 0000 0000 = 0000 0100)" << endl;
104 cout << "Bitwise 0 ^ 7 = " << d4 << " (0000 0000 & 0000 0111 = 0000 0000)" << endl;
105 cout << "Bitwise 4 ^ -2 = " << d5 << " (0000 0100 & -1111 1110 = -1111 1110)" << endl;
106 cout << "Bitwise -1 ^ 8 = " << d6 << " (-1111 1111 & 0000 1000 = -1111 1001)" << endl << endl;
107
108 // Operasi Bit OR ( | )
109 // Jika kedua nilainya bit 1 maka hasilnya bit 1
110 // Jika salah satu bit 1 dan satunya lagi bit 0 maka hasilnya bit 1
111 // Jika kedua nilainya bit 0 maka hasilnya bit 0
112 cout << "Operasi Bit OR ( | )" << endl;
113 cout << "Bitwise 1 | 1 = " << e1 << " (0000 0001 & 0000 0001 = 0000 0001)" << endl;
114 cout << "Bitwise 15 | 20 = " << e2 << " (0000 1111 & 0001 0100 = 0001 1111)" << endl;
115 cout << "Bitwise 4 | 0 = " << e3 << " (0000 0100 & 0000 0000 = 0000 0100)" << endl;
116 cout << "Bitwise 0 | 7 = " << e4 << " (0000 0000 & 0000 0111 = 0000 0111)" << endl;
117 cout << "Bitwise 4 | -2 = " << e5 << " (0000 0100 & -1111 1110 = -1111 1110)" << endl;
118 cout << "Bitwise -1 | 8 = " << e6 << " (-1111 1111 & 0000 1000 = -1111 1111)" << endl << endl;

```

```

120 // Operasi Bit NOT ( ~ )
121 // Nilai bilangan +1 kemudian dikonversi kebalikanya
122 // Jika nilainya bit 1 maka hasilnya bit 0
123 // Jika nilainya bit 0 maka hasilnya bit 1
124 cout << "Operasi Bit NOT ( ~ )" << endl;
125 cout << "Bitwise ~0 = " << f1 << " (0000 0000 = -1111 1111)" << endl;
126 cout << "Bitwise ~1 = " << f2 << " (0000 0001 = -1111 1110)" << endl;
127 cout << "Bitwise ~2 = " << f3 << " (0000 0010 = -1111 1101)" << endl;
128 cout << "Bitwise ~3 = " << f4 << " (0000 0011 = -1111 1100)" << endl;
129 cout << "Bitwise ~-0 = " << f5 << " (0000 0000 = -1111 1111)" << endl;
130 cout << "Bitwise ~-1 = " << f6 << " (-1111 1111 = 0000 0000)" << endl;
131 cout << "Bitwise ~-2 = " << f7 << " (-1111 1110 = 0000 0001)" << endl;
132 cout << "Bitwise ~-3 = " << f8 << " (-1111 1101 = 0000 0010)" << endl;
133 cout << "Bitwise ~-4 = " << f9 << " (-1111 1100 = 0000 0011)" << endl << endl;
134 }

```

Compiler dari code tersebut akan seperti gambar berikut :

C:\> Administrator: C:\Windows\system32\cmd.exe

```
Operasi Shift Kiri ( << )
Bitwise 1 << 1 = 2 (0000 0001 << 1 = 0000 0010)
Bitwise 9 << 3 = 72 (0000 1001 << 3 = 0100 1000)
Bitwise 20 << 4 = 320 (0001 0100 << 4 = 0001 0100 0000)
Bitwise 0 << 10 = 0 (0000 0000 << 10 = 0000 0000)
Bitwise 5 << 0 = 5 (0000 0101 << 0 = 0000 0101)

Operasi Shift Kanan ( >> )
Bitwise 1 >> 1 = 0 (0000 0001 >> 1 = 0000 0000)
Bitwise 72 >> 3 = 9 (0100 1000 >> 3 = 0000 1001)
Bitwise 320 >> 4 = 20 (0001 0100 0000 >> 4 = 0001 0100)
Bitwise 10 >> 5 = 0 (0000 1010 >> 10 = 0000 0000)
Bitwise 5 >> 0 = 5 (0000 0101 >> 0 = 0000 0101)

Operasi Bit AND ( & )
Bitwise 1 & 1 = 1 (0000 0001 & 0000 0001 = 0000 0001)
Bitwise 15 & 20 = 4 (0000 1111 & 0001 0100 = 0000 0100)
Bitwise 4 & 0 = 0 (0000 0100 & 0000 0000 = 0000 0000)
Bitwise 0 & 7 = 0 (0000 0000 & 0000 0111 = 0000 0000)
Bitwise 3 & -2 = 2 (0000 0011 & -1111 1110 = 0000 0010)
Bitwise -1 & 3 = 3 (-1111 1111 & 0000 0011 = 0000 0011)

Operasi Bit XOR ( ^ )
Bitwise 5 ^ 13 = 8 (0000 0101 & 0000 1101 = 0000 1000)
Bitwise 15 ^ 20 = 27 (0000 1111 & 0001 0100 = 0001 1011)
Bitwise 4 ^ 0 = 4 (0000 0100 & 0000 0000 = 0000 0100)
Bitwise 0 ^ 7 = 7 (0000 0000 & 0000 0111 = 0000 0000)
Bitwise 4 ^ -2 = -6 (0000 0100 & -1111 1110 = -1111 1110)
Bitwise -1 ^ 8 = -9 (-1111 1111 & 0000 1000 = -1111 1001)

Operasi Bit OR ( | )
Bitwise 1 | 1 = 1 (0000 0001 & 0000 0001 = 0000 0001)
Bitwise 15 | 20 = 31 (0000 1111 & 0001 0100 = 0001 1111)
Bitwise 4 | 0 = 4 (0000 0100 & 0000 0000 = 0000 0100)
Bitwise 0 | 7 = 7 (0000 0000 & 0000 0111 = 0000 0111)
Bitwise 4 | -2 = -2 (0000 0100 & -1111 1110 = -1111 1110)
Bitwise -1 | 8 = -1 (-1111 1111 & 0000 1000 = -1111 1111)

Operasi Bit NOT ( ~ )
Bitwise ~0 = -1 (0000 0000 = -1111 1111)
Bitwise ~1 = -2 (0000 0001 = -1111 1110)
Bitwise ~2 = -3 (0000 0010 = -1111 1101)
Bitwise ~3 = -4 (0000 0011 = -1111 1100)
Bitwise ~-0 = -1 (0000 0000 = -1111 1111)
Bitwise ~-1 = 0 (-1111 1111 = 0000 0000)
Bitwise ~-2 = 1 (-1111 1110 = 0000 0001)
Bitwise ~-3 = 2 (-1111 1101 = 0000 0010)
Bitwise ~-4 = 3 (-1111 1100 = 0000 0011)
```

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>_

6. **Operator Penugasan** = Memendekkan 2 buah operator yang dijadikan 1 buah operator.

Gambar berikut adalah contoh code dari operator tersebut :

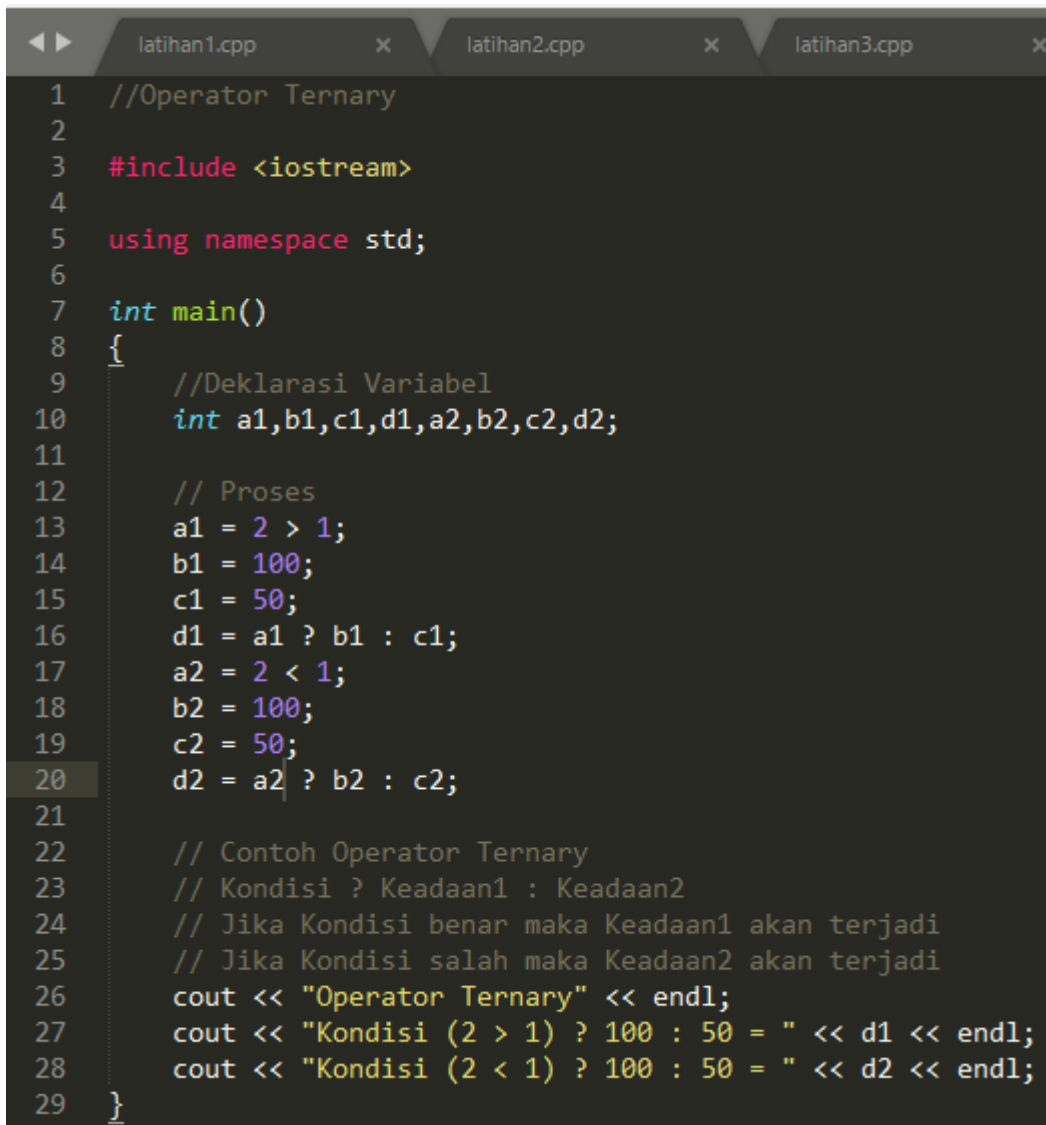
```
latihan1.cpp x latihan2.cpp x latihan3.cpp x latihan4.cpp x latihan5.cp
1 //Operator Penugasan
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Deklarasi variable
10    int a = 1, b = 1, c = 1, d = 1, e = 1, f = 1, g = 1, h = 1, i = 1, j = 1;
11
12    a += 5;
13    b -= 5;
14    c *= 5;
15    d /= 5;
16    e %= 5;
17    f <<= 5;
18    g >>= 5;
19    h &= 5;
20    i |= 5;
21    j ^= 5;
22
23    //Macam-Macam operator penugasan
24
25    // Operator penugasan mengoperasikan 1 operand dengan operator aritmatika atau bitwise
26    cout << "Operator Penugasan" << endl;
27    // Tambah +=
28    cout << "1 += 5 = " << a << endl;
29    // Kurang -=
30    cout << "1 -= 5 = " << b << endl;
31    // Kali *=
32    cout << "1 *= 5 = " << c << endl;
33    // Bagi /=
34    cout << "1 /= 5 = " << d << endl;
35    // Modulo %=
36    cout << "1 %= 5 = " << e << endl;
37    // Shift Kiri <<=
38    cout << "1 <<= 5 = " << f << endl;
39    // Shift Kanan >>=
40    cout << "1 >>= 5 = " << g << endl;
41    // AND &=
42    cout << "1 &= 5 = " << h << endl;
43    // OR |=
44    cout << "1 |= 5 = " << i << endl;
45    // XOR ^=
46    cout << "1 ^= 5 = " << j << endl;
47 }
```

Compiler dari code tersebut akan seperti gambar berikut :

```
Administrator: C:\Windows\system32\cmd.exe
Operator Penugasan
1 += 5 = 6
1 -= 5 = -4
1 *= 5 = 5
1 /= 5 = 0
1 %= 5 = 1
1 <<= 5 = 32
1 >>= 5 = 0
1 &= 5 = 1
1 |= 5 = 5
1 ^= 5 = 4
F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>
```

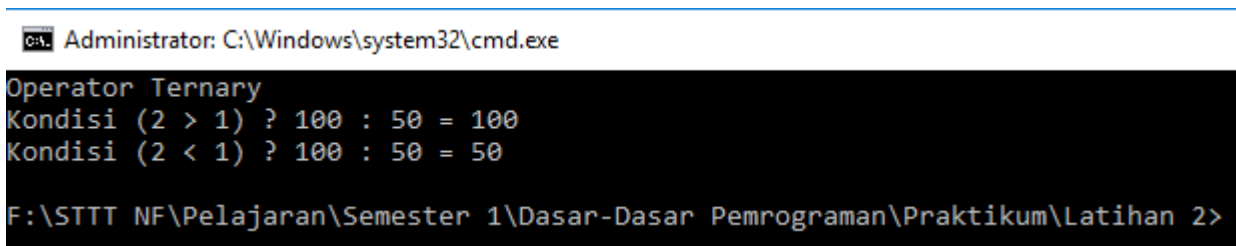

7. **Operator Ternary** = Melibatkan 3 operand dengan tipe angka atau huruf dan adanya kondisi, keadaan1, keadaan2.

Gambar berikut adalah contoh code dari operator tersebut :



```
1 //Operator Ternary
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main()
8 {
9     //Deklarasi Variabel
10    int a1,b1,c1,d1,a2,b2,c2,d2;
11
12    // Proses
13    a1 = 2 > 1;
14    b1 = 100;
15    c1 = 50;
16    d1 = a1 ? b1 : c1;
17    a2 = 2 < 1;
18    b2 = 100;
19    c2 = 50;
20    d2 = a2 ? b2 : c2;
21
22    // Contoh Operator Ternary
23    // Kondisi ? Keadaan1 : Keadaan2
24    // Jika Kondisi benar maka Keadaan1 akan terjadi
25    // Jika Kondisi salah maka Keadaan2 akan terjadi
26    cout << "Operator Ternary" << endl;
27    cout << "Kondisi (2 > 1) ? 100 : 50 = " << d1 << endl;
28    cout << "Kondisi (2 < 1) ? 100 : 50 = " << d2 << endl;
29 }
```

Compiler dari code tersebut akan seperti gambar berikut :



```
C:\> Administrator: C:\Windows\system32\cmd.exe

Operator Ternary
Kondisi (2 > 1) ? 100 : 50 = 100
Kondisi (2 < 1) ? 100 : 50 = 50

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Praktikum\Latihan 2>
```

Tugas

1. Buatlah program mengubah nilai suatu variable dengan menggunakan operator unary.
Dengan ketentuan :

a=15;
b=a;

Keluarannya harus

a=14;
b=15;

Jawaban

Source Code No 1

```
no1.cpp x no2.cpp x no3.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main ()
6  {
7      //Deklarasi
8      int a,b;
9
10     //Input
11     a = 15;
12     b = a;
13     cout << "a = " << a << endl;
14     cout << "b = a" << endl << endl;
15
16     //Proses dan Output
17     cout << "Keluarannya harus " << endl << endl;
18     cout << "a = " << --a << endl;
19     cout << "b = " << b << endl;
20 }
```

Hasil Compile Source Code No 1

```
Administrator: C:\Windows\system32\cmd.exe
a = 15
b = a

Keluarannya harus

a = 14
b = 15

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Tugas\Tugas Praktikum 2>_
```

2. Tuliskan program dibawah ini

Melakukan konversi valuta asing misalnya mata uang dollar US ke mata uang Rupiah, dimana satu dolar sama dengan 11.090

Input = uang dalam US dollar

Proses = $\text{uang_rupiah} = \text{uang_dollar} * 11090$

Output = hasil konversi menjadi uang rupiah

Jawaban

Source Code No 2

```
no1.cpp  no2.cpp  no3.cpp
1  #include <iostream>
2
3  using namespace std;
4
5  int main ()
6  {
7      //Deklarasi
8      int rupiah = 11090, dollar, konversi;
9
10     //Input
11     cout << "Masukkan Jumlah US $ ($1 = Rp11.090) = $";
12     cin >> dollar;
13
14     //Proses
15     konversi = dollar * rupiah;
16
17     //Output
18     cout << endl << "Hasil konversi menjadi uang rupiah = Rp" << konversi;
19 }
```

Hasil Compile Source Code No 2

Masukkan jumlah dollar yang ingin dikonversikan

```
C:\Windows\system32\cmd.exe - "F:\STTT NF\Pelajaran\Sei
Masukkan Jumlah US $ ($1 = Rp11.090) = $ _
```

Hasil dollar akan berubah menjadi rupiah dengan cara dikonversi

```
Administrator: C:\Windows\system32\cmd.exe
Masukkan Jumlah US $ ($1 = Rp11.090) = $10
Hasil konversi menjadi uang rupiah = Rp110900
F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Tugas\Tugas Praktikum 2>
```

3. Nyatakan dalam bentuk program

Mencari nilai akar x1 dan x2, dengan input a, b, c dari keyboard

$$D = b * b - 4 * a * c$$

Dan

$$X1 = -b + \text{akar}(D) / 2a$$

$$X2 = -b - \text{akar}(D) / 2a$$

Tuliskan sesuai dengan prioritas operatornya sehingga hasil perhitungan menjadi benar

Jawaban

Source Code No 3

```
no1.cpp x no2.cpp x no3.cpp x
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      //Deklarasi
9      int a,b,c,e;
10     float x1,x2,d;
11
12     //Input
13     cout << "Rumus D = (b * b) - (4 * a * c)" << endl << endl;
14     cout << "Masukkan nilai a = "; cin >> a;
15     cout << "Masukkan nilai b = "; cin >> b;
16     cout << "Masukkan nilai c = "; cin >> c;
17
18     //Proses
19     d = (b * b) - (4 * a * c);
20     x1 = -b + (sqrt(d))/(2*a);
21     x2 = -b - (sqrt(d))/(2*a);
22
23     //Output
24     cout << endl << "Nilai D = " << d << endl;
25     cout << "Nilai X1 = " << x1 << endl;
26     cout << "Nilai X2 = " << x2 << endl;
27 }
```

Hasil Compile Source Code No 3

Masukkan nilai a

```
C:\Windows\system32\cmd.exe - "F:\STTT
Rumus D = (b * b) - (4 * a * c)
Masukkan nilai a = _
```

Masukkan nilai b

```
C:\Windows\system32\cmd.exe - "F:\STTT
Rumus D = (b * b) - (4 * a * c)
Masukkan nilai a = 1
Masukkan nilai b = _
```

Masukkan nilai c

```
C:\Windows\system32\cmd.exe - "F:\STTT
Rumus D = (b * b) - (4 * a * c)
Masukkan nilai a = 1
Masukkan nilai b = 2
Masukkan nilai c = _
```

Hasilnya akan terlihat nilai D, X1, dan X2

```
Administrator: C:\Windows\system32\cmd.exe
Rumus D = (b * b) - (4 * a * c)
Masukkan nilai a = 1
Masukkan nilai b = 2
Masukkan nilai c = -3

Nilai D = 16
Nilai X1 = 0
Nilai X2 = -4

F:\STTT NF\Pelajaran\Semester 1\Dasar-Dasar Pemrograman\Tugas\Tugas Praktikum 2>_
```