

1 D4 - TEKKOM B

EXPRESSION (OPERATOR & OPERAND) & STATEMENT



Nama	:	Septian Bagus Jumanoro
Kelas	:	1 – D4 Teknik Komputer B
NRP	:	322160039
Dosen	:	Ir Sigit Wasista, M.Kom.
Mata Kuliah	:	Praktikum Pemrograman Dasar 1
Hari/Tgl. Praktikum	:	10 September 2021

PERCOBAAN 2

1. JUDUL: EXPRESSION (OPERATOR & OPERAND) & STATEMENT

2. TUJUAN: Mempelajari dan mengamati penggunaan operator aritmatika seperti

+, -, *, / dan % (modulus) dan operator unary seperti – (minus), ++ (increment) dan – (decrement) yang dipadukan dengan tipe variable dasar yaitu int (bilangan bulat) dan float (presisi tunggal), double (presisi ganda) dan char.

3. TEORI: Suatu Expression adalah terdiri atas dua bagian yaitu operator dan operand, yang dikombinasikan menjadi satu. Operator adalah symbol yang digunakan untuk melakukan proses operasi satu atau beberapa operand. Operand adalah bagian yang paling sederhana dari Expression. Operand dapat berupa konstanta seperti 339, 55, 132 dan seterusnya atau berupa variable seperti x, y, jumlah, selisih dan sebagainya. Statement (statement) adalah unsur dasar pembentuk suatu program. Suatu program terdiri dari beberapa statement, dimana komputer akan melakukan tugas tertentu sesuai dengan urutan statement. Ada 3 jenis statement, yaitu: expression statement, compound statement, dan control statement. Suatu expression statement adalah suatu expression yang diikuti dengan tanda titik koma [;] (semi colon). Suatu compound statement (dikenal dengan block statement) adalah dua atau lebih statement yang dikelompokkan menjadi satu dengan cara memberi batas tanda kurung awal dan tanda kurung akhir, sehingga tidak perlu diakhiri dengan tanda titik koma pada akhir dari compound. Control statement adalah statement yang mengendalikan langkah-langkah program, contohnya for loop, while loop, dan if-else (yang akan dilakukan pada percobaan berikutnya). Symbolic Constant adalah suatu nama dimana digunakan untuk menggantikan suatu nilai tertentu, sehingga akan lebih mudah dalam pembacaan suatu program, contohnya #define PI 3.14, untuk membedakan dengan variable maka nama ditulis dengan huruf besar seperti PI.

4. PROGRAM PERCOBAAN

4.1.1 Mengamati tipe variable yang digunakan untuk menyimpan suatu nilai sesuai dengan tipe dari masing masing nilai yang diberikan pada suatu variable, kemudian mencetak isi variable tersebut dan melihat hasilnya. Misalnya bilangan bulat, maka dapat ditampung menggunakan tipe variabel integer yang mempunyai jangkauan dari -32768 sampai 32767, bilangan pecahan dengan presisi tunggal dapat disimpan pada tipe variable dengan tipe float yang mempunyai jangkauan dari 3.4×10^{-38} sampai 3.4×10^{38} (7 angka presisi), bilangan pecahan dengan presisi ganda dapat disimpan pada tipe variable dengan tipe double yang mempunyai jangkauan dari 1.7×10^{-308} sampai dengan 1.7×10^{308} (15 angka presisi), long double mempunyai jangkauan dari 3.4×10^{-4932} sampai dengan 3.4×10^{4932} (19 angka presisi) dan tipe variable char dengan jangkauan nilai dari -128 sampai 127 (satu karakter).

```

/* Nama File : EXPRESS1.C */

void main()
{
    int bulat = 32767;
    float pecahan = 339.1234567;
    double ganda = 3.4567890e+11;
    char karakter = 'W';

    printf("Variabel bilangan bulat      = %d\n\n", bulat);
    printf("Variabel bilangan pecahan    = %d\n\n", pecahan);
    printf("Variabel bilangan pecahan2 = %d\n\n", ganda);
    printf("Variabel bilangan karakter = %d\n\n", karakter);

    getch();
}

```

4.1.2 Mengamati hasil dari suatu persamaan aritmatika, dimana nilai-nilai operandnya dimasukkan melalui keyboard serta hirarki dari operator.

```

/* Nama File : EXPRESS2.C */

void main()
{
    int a, b, c, d, hasil;

    printf("\nMasukkan nilai a b = ");
    scanf("%d %d", &a, &b);
    printf("\nMasukkan nilai c d = ");
    scanf("%d %d", &c, &d);
    printf("a = %d, b = %d, c = %d, d = %d", a, b, c, d);

    hasil = a - b;
    printf("\nHasil dari: a - b adalah %d\n", hasil);

    hasil = c + d;
    printf("\nHasil dari: c + d adalah %d\n", hasil);

    hasil = b * c;
    printf("\nHasil dari: b * c adalah %d\n", hasil);

    hasil = a / c;
    printf("\nHasil dari: a / c adalah %d\n", hasil);

    hasil = a + b * c;
    printf("\nHasil dari: a + b * c adalah %d\n", hasil);

    hasil = a * b + c * d;
    printf("\nHasil dari: a * b + c * d adalah %d\n", hasil);

    getch();
}

```

4.1.3 Mengamati perbedaan penggunaan tipe variable integer dan tipe variable float pada proses pembagian, bila pembagian dilakukan menggunakan tipe integer maka pecahan akan dibuang. Perhatikan hasil dari pembagian isi variable a dibagi dengan 5, isi variable a dibagi dengan isi variable b dan isi variable c dibagi dengan d. Perhatikan juga penggunaan operator unari minus (-).

```
/* Nama File : EXPRESS3.C */

void main()
{
    int a = 12;
    int b = 8;
    int hasil;
    float c = 11.0;
    float d = 7.0;

    printf("6 + a / 5 * b = %d\n\n", 6 + a / 5 * b);
    printf("a / b * b = %d\n\n", a / b * b);
    printf("c / d * d = %f\n\n", c / d * d);
    printf("-a = %d\n", -a);

    getch();
}
```

4.1.4 Untuk menghitung sisa dari hasil bagi maka telah disediakan operator modulus yaitu (%), dimana hasil dari operasi dengan operator ini akan menghasilkan sisa pembagian. Perlu diperhatikan bahwa dikarenakan operator ini adalah sama dengan operator format, sehingga bila operator modulus ingin ditampilkan, maka penulisan operator ini harus ditulis dua kali (%%).

```
/* Nama File : EXPRESS4.C */

void main()
{
    int a = 14, b = 2, c = 3, d = 4;

    printf("a %% b = %d\n\n", a % b);
    printf("a %% c = %d\n\n", a % c);
    printf("a %% d = %d\n\n", a % d);
    printf("a / d * d + a %% d = %d\n", a / d * d + a % d);

    getch();
}
```

4.1.5 Suatu variable dapat menampung isi dari suatu tipe variable yang berbeda asalkan tipe variable tersebut lebih besar atau sama dengan variable sumber. Contoh tipe variabel float dapat menampung isi dari tipe variable integer dan tidak sebaliknya.

```
/* Nama File : EXPRESS5.C */

void main()
{
    float p1 = 123.45678, f2;
    int b1, b2 = -150;

    b1 = p1;          /* konversi pecahan ke bulat */
    printf("%f disimpan ke int meng)asilkan %d\n\n", p1, b1);

    p1 = b2;          /* konversi bulat ke pecahan */
    printf("%d disimpan ke float meng)asilkan %f\n\n", b2, p1);

    p1 = b2 / 100;     /* bulat dibagi bulat */
    printf("%d dibagi 100 meng)asilkan %f\n\n", b2, p1);

    p2 = b2 / 100.0;   /* bulat dibagi pecahan */
    printf("%d dibagi 100 meng)asilkan %f\n\n", b2, p2);

    getch();
}
```

4.1.6 Operator increment (++) dan operator decrement (--) hanya dapat digunakan pada suatu variable, dan tidak dapat digunakan pada suatu konstanta. Penggunaan dari operator ini akan berakibat perubahan pada variable, yaitu ditambah satu (++) atau dikurangi satu (--), contoh $x++$ adalah sama dengan $x=x+1$ dan $x--$ adalah sama dengan $x=x-1$. Sedangkan posisi dari tanda (prefix atau postfix) juga berpengaruh pada hasil suatu expression.

```
/* Nama File : EXPRESS6.C */

void main()
{
    int x, y, z;

    x=80;  y=x++;  z=++x;
    printf("Nilai dari x, y, z adala) %d, %d, %d\n\n", x, y, z);

    y=x--;  z=--x;
    printf("Nilai dari x, y, z adala) %d, %d, %d", x, y, z);

    getch();
}
```

4.1.7 Format tampilan dari suatu variable dengan tipe pecahan dapat digunakan tiga macam format tampilan yaitu menggunakan karakter khusus %e, %f dan %g. Disini diberikan program untuk mencoba hasil yang ditampilkan dari ketiga karakter format tersebut.

```
/* Nama File: EXPRESS7.C */

#include cstdio.)S

void main()
{
    float x;
    printf("Masukkan nilai pecahan yg akan ditampilkan : ");
    scanf("%f", &x);

    printf("format e =S %e\n", x);
    printf("format f =S %f\n", x);
    printf("format g =S %g\n", x);

    getch();
}
```

5. TUGAS-TUGAS (Dikumpulkan 1 minggu setelah praktikum dilaksanakan)

4.5.1 Buatlah program untuk mengkonversikan suatu nilai derajat Fahrenheit

(F) ke derajat Celcius (C) menggunakan formula dibawah ini:

$$C = \frac{5}{9} (F - 32)$$

Contoh tampilan:

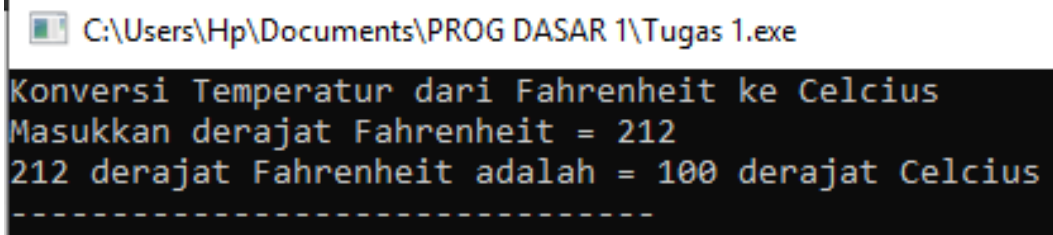
Masukkan derajat Fahrenheit = 212 **J**
212 derajat Fahrenheit adalah = 100 derajat Celcius

Source Code

```
#include <stdio.h>

void main()
{
    int fahrenheit, celcius;
    puts("Konversi Temperatur dari Fahrenheit ke Celcius" );
    printf("Masukkan derajat Fahrenheit = ");
    scanf("%d", &fahrenheit);
    celcius = (fahrenheit - 32) * 5/9;
    printf("%d derajat Fahrenheit adalah = %d derajat Celcius ",
           fahrenheit, celcius);
}
```

Output



```
C:\Users\Hp\Documents\PROG DASAR 1\Tugas 1.exe
Konversi Temperatur dari Fahrenheit ke Celcius
Masukkan derajat Fahrenheit = 212
212 derajat Fahrenheit adalah = 100 derajat Celcius
-----
```

Analisa

Pada program diatas saya menggunakan variabel yang bertipe integer. Awal jalannya program saya menggunakan `scanf("...")` untuk menginput dari keyboard. Setelah itu di masukkan kedalam rumus konversi dari fahrenheit ke celcius $((f - 32) \times \frac{5}{9})$. Lalu diprint dan menghasilkan output.

4.5.2 Buatlah program untuk mengevaluasi formula dibawah ini:

$$y = 3^2 + 6x + 9$$
$$z = \frac{2y^2 + 5x^2}{9y}$$

Contoh tampilan:

Formula: $y = 3^2 + 6x + 9$

Masukkan nilai $x = 2$ **J**

Didapatkan nilai $y = 33$ dan nilai $z = 7.400673$

Source Code

```
#include <stdio.h>


void main()
{
    int x,y;
    float z;

    puts("Formula 1 & 2");
    printf("Masukkan nilai x = ");
    scanf("%d", &x);

    y = 3 * pow(x,2) + 6 * x + 9;
    z = (2 * pow(y,2) + 5 * pow(x,2) ) / (9 * y);

    printf("Didapatkan nilai y = %d dan nilai z = %f ", y, z);
}
```

Output

 C:\Users\Hp\Documents\PROG DASAR 1\Tugas 1.exe

```
Formula 1 & 2
Masukkan nilai x = 2
Didapatkan nilai y = 33 dan nilai z = 7.400673
-----
```


Analisa

Pada program diatas saya menggunakan 2 tipe variabel yaitu integer dan float. Lalu memasukkan syntax scanf agar program dapat menginput dari keyboard. Lalu masuk ke dalam rumus formula 1 dimana x = hasil input dari keyboard. Untuk formula 2 dimana y = hasil operasi formula 1. Lalu program di print dan menghasilkan output ketika program sudah di compile & run.

4.5.3 Buatlah program untuk menghitung keliling dan luas suatu lingkaran, dengan formula:

$$\text{keliling} = 2 r$$

$$\text{luas} = r^2$$

Contoh tampilan:

Masukkan jari-jari lingkaran = 12 **J**

Keliling lingkaran dengan jari-jari 12 adala) = 75.36

Luas Lingkaran dengan jari-jari 12 adala) = 452.16

Source Code


```
#include <stdio.h>

void main()
{
    int r;
    float Pi, k, l;

    Pi = 3.14;
    printf("Masukkan jari-jari lingkaran = ");
    scanf("%d", &r);
    puts("");
    k = 2 * Pi * r;
    l = Pi * pow(r,2);

    printf("Keliling lingkaran dengan jari-jari %d adalah = %.2f", r, k);
    puts("");
    printf("Luas lingkaran dengan jari-jari %d adalah = %.2f", r, l);
}
```

Output

 C:\Users\Hp\Documents\PROG DASAR 1\Tugas 1.exe

```
Masukkan jari-jari lingkaran = 12
```

```
Keliling lingkaran dengan jari-jari 12 adalah = 75.36
```

```
Luas lingkaran dengan jari-jari 12 adalah = 452.16
```

Analisa

Pada program tersebut saya menggunakan 2 tipe variabel yaitu integer dan float. Lanjut mendeskripsikan nilai variabel $\pi = 3,14$. Lalu menggunakan syntax `scanf("...")` untuk menginput angka dari keyboard. Masuk ke rumus yaitu :

- Keliling = $2\pi r$
- Luas = πr^2

Disini untuk perpangkatan saya menggunakan alternatif power (`pow`). Setelah itu di `printf` dan menghasilkan output.

4.5.4 Buatlah program untuk mengkonversi dari jam ke menit.

Contoh tampilan:

```
Program konversi jam ke menit
Masukkan jam dan menit (jj:mm) = 11:07 J
Jam 11:07 adala) setara dengan 667 menit
```

Petunjuk: gunakan fungsi `scanf()` sebagai berikut:

```
scanf("%d:%d",&jam,&menit);
```


Source Code

```
#include <stdio.h>

void main()
{
    int jam, menit, time;

    puts("Program konversi jam ke meinit");
    printf("Masukkan jam dan menit (jj:mm) = ");
    scanf("%d:%d", &jam, &menit);
    time = jam * 60 + menit;
    printf("Jam %d:%d adalah setara dengan %d menit", jam,
    menit, time);
}
```

Output

 C:\Users\Hp\Documents\PROG DASAR 1\Tugas 1.exe

```
Program konversi jam ke meinit
Masukkan jam dan menit (jj:mm) = 11:07
Jam 11:7 adalah setara dengan 667 menit
-----
```

Analisa

Pada program tersebut saya menggunakan tipe variabel integer. Lalu memasukkan syntax scanf dengan petunjuk `scanf("%d:%d", jam, menit)` untuk menginput jam dan menit. Lalu memasukkan rumus dari jam ke menit yaitu $\text{time} = \text{jam} \times 60 + \text{menit}$. Setelah itu Source Code di compile & run, dan muncullah output.

4.5.5 Bagaimana tampilan dari program dibawah ini? Mengapa demikian?

```
/* Nama File: EXPRESS8.C */

void main()
{
    int a, b, c, d;
    double e, f, g, );

    a=75; b=259; c=3315; d=12345;
    e=75.07; f=259.055; g=3315.3310; )=12345.67890;

    printf("a=%7d\nb=%7d\nc=%7d\nd=%7d\n\n", a, b, c, d);
    printf("e=%12.5f\nf=%12.5f\ng=%12.5f\n)=%12.5f\n", e, f, g, ));

    getch();
}
```

Output

```
C:\Users\Hp\Documents\PROG DASAR 1\Tugas 1.exe
a=      75
b=     259
c=    2315
d=   12345

e=    75.07000
f=   259.05500
g=  3315.33100
h= 12345.67890
-----
```

Analisa

Tampilan pada output program tersebut barisan angkanya tersusun rapi layaknya anak tangga. Mengapa demikian? Karena pada source code menggunakan beberapa tipe variabel yaitu Integer dan Double. Dikarenakan integer khusus bilangan bulat, maka pada saat memanggilnya %7d yang artinya mencetak total 7 karakter setelah nilai variabel. Begitu juga dengan tipe variabel double yang sama dengan float yaitu khusus bilangan desimal, maka pada saat memanggilnya %12.5f yang artinya total keseluruhan karakter yang diprint yaitu 12 dan terdapat 5 karakter setelah koma.