

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

PENUGASAN PRAKTIKUM STRUKTUR DATA PERTEMUAN 2

1. **sizeof()** adalah sebuah operator untuk mengetahui jumlah memori (byte) yang diperlukan oleh suatu tipe data pada bahasa C. Gunakan **sizeof()** untuk mengetahui ukuran memori pada berbagai tipe data pada bahasa pemrograman C, seperti **char**, **int**, **float**, **double**. Catat hasil yang Anda dapatkan. Bandingkan dengan hasil yang didapatkan oleh teman-teman Anda. Diskusikan apakah yang menyebabkan hasil yang didapatkan berbeda-beda.

Jawab:

Source Code:

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    printf("%lu\n", sizeof(char));
    printf("%lu\n", sizeof(int));
    printf("%lu\n", sizeof(float));
    printf("%lu", sizeof(double));

    return 0;
}
```

Output:

```
PS G:\My Drive\00. Tingkat 2\STRUKTUR DATA> cd "G:\My Drive\00. Tingkat 2\STRUKTUR DATA\Pertemuan 2\"; if ($?) { gcc 222212738_Pertemuan2_Penugasan1.c -o 222212738_Pertemuan2_Penugasan1 } ; if ($?) { .\222212738_Pertemuan2_Penugasan1 }
1
4
4
8
```

Penjelasan:

Di PC yang saya gunakan, sizeof() dari char sebesar 1 byte, integer sebesar 4 byte, float sebesar 4 byte, dan double sebesar 8 byte. Hal ini tentunya bisa saja berbeda antar komputer, bergantung dari sistem yang digunakan (32/64 bit) maupun compiler yang digunakan.

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

2. Ketik program berikut pada IDE Anda, lalu simpan.

```
/* Aturan Scope pada Bahasa C */
#include <stdio.h>

int main(int argc, char const *argv[])
{
    {
        int x = 10, y = 20; //(1)
        {
            printf("x = %d, y = %d\n", x, y); //print var x
            pada (1) dan var y pada (1)
            {
                int y = 40; //(2)
                x++; //increment var x pada (1)
                y++; //increment var y pada (2)
                printf("x = %d, y = %d\n", x, y); //print
                var x pada (1) dan var y pada (2)
            }
            printf("y = %d, y = %d\n", x, y); //print var x
            pada (1) dan var y pada (1)
        }
    }

    return 0;
}
```

Jawab:

Output:

```
PS G:\My Drive\00. Tingkat 2\STRUKTUR DATA\Pertemuan 2> cd "g:\My Drive\00. Tingkat 2\STRUKTUR DATA\Pertemuan 2\" ; if ($?) { gcc 222212738_Pertemuan2_Penugasan2.c -o 222212738_Pertemuan2_Penugasan2 } ; if ($?) { .\222212738_Pertemuan2_Penugasan2 }
x = 10, y = 20
x = 11, y = 41
y = 11, y = 20
```

Penjelasan:

Perbedaan tersebut terjadi karena adanya *block scope* yang ditandai dengan { ... }. Pada perintah print yang pertama, compiler akan mencetak sesuai dengan value pada variable

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

(1). Namun variable y diinisialisasi lagi pada *block scope* selanjutnya (2), sehingga pada *block scope* tersebut, jika dipanggil variable y maka akan menggunakan variable y pada (2) walaupun sudah pernah diinisiasi sebelumnya. Sehingga, pada *block scope* tersebut meng-*increment* variable x pada (1) dan variable y pada (2). Lalu pada perintah print yang ketiga, maka variable x yang digunakan yaitu pada (1) yang sudah di-*increment* sebelumnya dan variable y sesuai pada (1).

3. Buatlah program untuk menginput nilai elemen-elemen Matriks A berukuran 3x4 dan mencetak Matriks A tersebut. Contoh output sebagai berikut.

1	3	4	5
2	4	6	8
3	5	7	9

Jawab:

Source Code:

```
#include <stdio.h>

int main(int argc, char const *argv[])
{
    int matriks[3][4];

    for (int i = 0; i < 3; i++)
    {
        for (int j = 0; j < 4; j++)
        {
            printf("Masukkan nilai element matriks ke %d, %d: ", i + 1, j + 1);
            scanf("%d", &matriks[i][j]);
        }
    }

    printf("\nHasil matriks:\n");
    for (int i = 0; i < 3; i++)
    {
```

Nama : Muh. Nur Afrizal

NIM : 222212738

Kelas / No : 2KS1 / 25

```
        for (int j = 0; j < 4; j++)
        {
            printf("%d ", matriks[i][j]);
        }
        printf("\n");
    }

    return 0;
}
```

Output:

```
PS G:\My Drive\00. Tingkat 2\STRUKTUR DATA> cd "g:\My Drive\00. Tingkat 2\STRUKTUR DATA\Pertemuan 2\" ; if ($?) { gcc 222212738_Pertemuan2_Penugasan3.c -o 222212738_Pertemuan2_Penugasan3 } ; if ($?) { .\222212738_Pertemuan2_Penugasan3 }
Masukkan nilai element matriks ke 1, 1: 1
Masukkan nilai element matriks ke 1, 2: 3
Masukkan nilai element matriks ke 1, 3: 4
Masukkan nilai element matriks ke 1, 4: 5
Masukkan nilai element matriks ke 2, 1: 2
Masukkan nilai element matriks ke 2, 2: 4
Masukkan nilai element matriks ke 2, 3: 6
Masukkan nilai element matriks ke 2, 4: 8
Masukkan nilai element matriks ke 3, 1: 3
Masukkan nilai element matriks ke 3, 2: 5
Masukkan nilai element matriks ke 3, 3: 7
Masukkan nilai element matriks ke 3, 4: 9

Hasil matriks:
1 3 4 5
2 4 6 8
3 5 7 9
```