

## Soal Latihan 2 Praktikum Algoritma Struktur Data - Pointer dan Fungsi

1. Buatlah sebuah fungsi bernama `greatestOf` yang menerima empat parameter, yaitu bilangan bulat non-negatif  $a$ ,  $b$ ,  $c$ , dan  $d$  kemudian mengembalikan nilai maksimum dari keempat bilangan tersebut dengan jaminan bahwa tidak ada dua dari empat bilangan tersebut yang sama. Solusi terdiri dari 2 baris, yaitu baris pertama berisi bilangan  $a$ ,  $b$ ,  $c$ , dan  $d$  yang dipisahkan tanda spasi. Kemudian dilanjutkan baris berikutnya yang berisi nilai maksimum dari keempat bilangan tersebut.
2. Buatlah sebuah fungsi bernama `arrayMean` bertipe data float yang menerima parameter sebuah array integer sepanjang  $N$  serta ukurannya, dan mengembalikan nilai rata-rata dari array tersebut. Solusi terdiri dari 3 baris, yaitu baris pertama berisi bilangan bulat positif  $N$ . Setelah itu, dilanjutkan oleh baris kedua yang berisi sebanyak  $N$  bilangan yang dipisahkan data spasi. Kemudian diakhiri oleh baris terakhir yang berisi nilai rata-rata dari  $N$  bilangan tersebut dengan presisi hingga 2 angka di belakang koma.
3. Saat sedang gabut, Kobo membayangkan suatu skenario dimana hanya ada satu buah bidak, yaitu kuda (knight), yang berada di dalam papan catur berukuran  $8 \times 8$ . Kobo penasaran dan ingin mengetahui posisi mana saja yang dapat dicapai oleh bidak kuda tersebut dalam sekali jalan apabila bidak tersebut berada pada posisi  $i, j$  dengan rincian  $0 \leq i, j < 8$ . Namun, karena Kobo tidak punya uang untuk membeli papan catur, ia ingin mensimulasikannya pada komputer dengan menggunakan bahasa C. Dalam simulasinya, ia ingin punya array 2D dengan nilai awal 0 di setiap index sebagai bidak caturnya. Kemudian, Kobo ingin memberi nilai 1 pada setiap posisi yang mungkin dilalui oleh bidak kuda tersebut dalam sekali jalan apabila bidak tersebut berada pada posisi  $i, j$ . Bantulah Kobo untuk membuat program tersebut dengan melanjutkan fungsi berikut:

```
void koboImaginaryChess(int i, int j, int size, int *chessBoard){...}
```

Input terdiri dari satu baris yang berisi nilai  $i$  dan  $j$  yang dipisahkan tanda spasi. Output adalah output semua nilai di setiap baris dan kolom dari array berukuran  $8 \times 8$  sesuai soal.

## Jawaban

### 1. Source Code

```
//memanggil header file stdio.h untuk akses fungsi standar input dan output
#include <stdio.h>
//fungsi untuk memilah dan mengembalikan nilai terbesar dari array
int greatestOf(int *arr){
    //sebagai permisalan, nilai terbesar saat ini adalah 0
    int max = 0;
    //melakukan perulangan untuk menyisir selueruh isi array
    for(int a = 0; a < 4; a++){
        //jika nilai dari array panda ideks a lebih besar dari nilai terbesar saat ini,
        // maka nilai array tersebut akan menggantikan nilai terbesar saat ini.
        if(arr[a] > max) max = arr[a];
    }
    //mengembalikan nilai terbesar
    return max;
}

//fungsi utama yang pertama kali akan dieksekusi program
int main(){
    //deklarasi array dengan panjang 4 elemen
    int arr[4];
    //menerima dan menyimpan input pengguna untuk setiap elemen array di atas
    scanf("%d %d %d %d", &arr[0], &arr[1], &arr[2], &arr[3]);
    //menampilkan nilai terbesar dari array yang didapat dari nilai kembalian fungsi greatestOf
    printf("%d", greatestOf(arr));
}
```

### 1. Result

```
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> .\tugas2_soal1
4 5 3 2
5
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> .\tugas2_soal1
101 27 26 998
998
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> █
```

## 2. Source Code

```
//memanggil header file stdio.h untuk akses fungsi standar input dan output
#include <stdio.h>
//fungsi pertama yang pertama kali akan dieksekusi program
int main(){
    //deklarasi variabel len(panjang) dan sum(total)
    int len, sum = 0;
    //menerima dan menyimpan input pengguna berupa angka untuk panjang array
    scanf("%d", &len);
    //deklarasi array dengan panjang yang ditentukan pengguna
    int arr[len];
    //perulangan sebanyak panjang array
    for(int i = 0; i < len; i++){
        //menerima dan menyimpan input untuk array pada indeks i
        scanf(" %d", &arr[i]);
        //menjumlahkan inputan ke variabel total
        sum += arr[i];
    }
    //menampilkan hasil dari pembagian total dengan jumlah yang casting menjadi float
    printf("%.2f", (float) sum/len);
}
```

## 2. Result

```
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> .\tugas2_soal2
4
3 7 1 10
5.25
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> .\tugas2_soal2
11
4 6 10 30 22 11 89 62 78 24 1
30.64
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> █
```

### 3. Source Code

```
//memanggil header file stdio.h untuk akses fungsi standar input dan output
#include <stdio.h>
//prosedur untuk menampilkan titik-titik yang bisa dicapai kuda (knight) sesuai nilai titik [i][j] yang diterima.
void koboImaginaryChess(int i, int j, int size, int *chessBoard){
    //melakukan perulangan sebanyak 8 kali (8 baris)
    for(int x = 0; x < size; x++){
        //melakukan perulangan sebanyak 8 kali (8 kolom)
        for(int y = 0; y < size; y++){
            //menyimpan nilai 0 pada indeks [x][y]
            *(chessBoard + x * size + y) = 0;
            //jika (i - x) adalah 2, maka baris (x) adalah baris dimana titik yang bisa dicapai kuda (knight) berada.
            if((i - x) == 2){
                //jika (j - y) adalah 1 atau (y - j) adalah 1,
                // maka kolom (y) adalah kolom dimana titik yang bisa dicapai kuda (knight) berada.
                if((j - y) == 1 || (y - j) == 1){
                    //menyimpan nilai 1 pada indeks [x][y] sebagai tanda dari titik yang bisa dicapai kuda (knight).
                    *(chessBoard + x * size + y) = 1;
                }
            }
            //jika (i - x) adalah 1, maka baris (x) adalah baris dimana titik yang bisa dicapai kuda (knight) berada.
            if((i - x) == 1){
                //jika (j - y) adalah 2 atau (y - j) adalah 2,
                // maka kolom (y) adalah kolom dimana titik yang bisa dicapai kuda (knight) berada.
                if((j - y) == 2 || (y - j) == 2){
                    //menyimpan nilai 1 pada indeks [x][y] sebagai tanda dari titik yang bisa dicapai kuda (knight).
                    *(chessBoard + x * size + y) = 1;
                }
            }
            //jika (x - i) adalah 1, maka baris (x) adalah baris dimana titik yang bisa dicapai kuda (knight) berada.
            if((x - i) == 1){
                //jika (j - y) adalah 2 atau (y - j) adalah 2,
                // maka kolom (y) adalah kolom dimana titik yang bisa dicapai kuda (knight) berada.
                if((j - y) == 2 || (y - j) == 2){
                    //menyimpan nilai 1 pada indeks [x][y] sebagai tanda dari titik yang bisa dicapai kuda (knight).
                    *(chessBoard + x * size + y) = 1;
                }
            }
            //jika (x - i) adalah 2, maka baris (x) adalah baris dimana titik yang bisa dicapai kuda (knight) berada.
            if((x - i) == 2){
                //jika (j - y) adalah 2 atau (y - j) adalah 2,
                // maka kolom (y) adalah kolom dimana titik yang bisa dicapai kuda (knight) berada.
                if((j - y) == 1 || (y - j) == 1){
                    //menyimpan nilai 1 pada indeks [x][y] sebagai tanda dari titik yang bisa dicapai kuda (knight).
                    *(chessBoard + x * size + y) = 1;
                }
            }
        }
        //menampilkan kotak yang direpresentasikan dengan 0 dan 1
        printf("%d ", *(chessBoard + x * size + y));
    }
    //mencetak baris baru (newline)
    printf("\n");
}

//fungsi utama yang akan pertama kali dieksekusi program
int main(){
    //mendeklarasikan variabel-variabel yang diperlukan
    int i, j, arr[8][8];
    //menerima dan menyimpan inputan dari user ke variabel i dan j sebagai titik posisi kuda (knight) berada.
    scanf("%d %d", &i, &j);
    //memanggil fungsi koboImaginaryChess dengan mengoper variabel-variabel yang diperlukan.
    koboImaginaryChess(i, j, 8, arr[0]);
}
```

### 3. Result

```
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> .\tugas2_soal3
2 2
0 1 0 1 0 0 0 0
1 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
1 0 0 0 1 0 0 0
0 1 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> .\tugas2_soal3
3 7
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
PS D:\KULIAH\ITTS Semester 2\Tugas\Algoritma dan Struktur Data\Pertemuan 06 - 11 April 2023> |
```