

Tugas praktikum 4 - OTH

Nama: Moch. Rama Maulana

NIM: 1203232085

Kelas: IF-03-01

Komponen Penilaian	Ya	Tidak
Soal 1 sesuai dengan output yang diinginkan	V	
Soal 2 sesuai dengan output yang diinginkan	V	
Bonus soal 1 dikerjakan	V	

Soal 1

Code dan penjelasan:

```
#include <stdio.h>

// Fungsi untuk konversi kartu huruf menjadi angka
int konversiKartuAngka(char *kartu) {
    int kartuAngka;

    // Cek angka 1 jika kartu ini bernilai 1 dan setelahnya TIDAK bernilai 0
    if (kartu[0] == '1' && kartu[1] != '0') kartuAngka = 1;
    // Cek angka 10 jika kartu ini bernilai 1 dan setelahnya bernilai 0
    else if (kartu[0] == '1' && kartu[1] == '0') kartuAngka = 10;
    // Cek angka 11, 12, dan 13 berdasarkan huruf dari kartu ini
    else if (kartu[0] == 'J') kartuAngka = 11;
    else if (kartu[0] == 'Q') kartuAngka = 12;
    else if (kartu[0] == 'K') kartuAngka = 13;
    // Konversikan kartu ini menjadi integer
    else kartuAngka = kartu[0] - '0';

    return kartuAngka;
}

// Fungsi untuk mencari jumlah minimal pertukaran agar kartu menjadiurut
void sortKartu(char *kartu, int *kartuAngka, int jumlah) {
    // Inisialisasi jumlah pertukaran
    int jumlahTukar = 0;

    // Gunakan algoritma selection sort karena hanya algoritma ini yang sesuai
    // dengan contoh output
    for (int i = 0; i < jumlah; i++) {
        // Inisialisasi index dengan nilai terkecil untuk setiap iterasi
        int minIndex = i;

        // Cek apakah ada index lain dengan value lebih besar dari index awal
        for (int j = i; j < jumlah; j++) {
```

```

        if (kartuAngka[j] < kartuAngka[minIndex]) {
            minIndex = j;
        }
    }

    // Cek apakah index terkecil sama dengan index perulangan, jika sama
    maka akan dilewati agar output bisa efisien
    if (minIndex != i) {
        // Melakukan pertukaran array kartu bentuk angka
        int tempA = kartuAngka[i];
        kartuAngka[i] = kartuAngka[minIndex];
        kartuAngka[minIndex] = tempA;

        // Melakukan pertukaran kartu bentuk huruf
        int tempB = kartu[i];
        kartu[i] = kartu[minIndex];
        kartu[minIndex] = tempB;

        // Menampilkan hasil dari pertukaran untuk setiap iterasi
        // ++jumlahTukar untuk menambahkan jumlahTukar yang sekarang dengan 1,
        ++ di awal agar jumlahTukar ditambahkan
        // pada iterasi saat ini juga, kalau menggunakan ++ di akhir maka
        jumlahTukar ditambahkan di iterasi berikutnya
        printf("Pertukaran ke-%d: ", ++jumlahTukar);
        for (int i = 0; i < jumlah; i++) {
            // Cek apakah isi dari kartu tersebut adalah 10 karena diantara
            semua pilihan kartu hanya angka 10 yang 2 digit
            if (kartuAngka[i] == 10) {
                printf("%d ", kartuAngka[i]);
            } else {
                printf("%c ", kartu[i]);
            }
        }
        printf("\n");
    }
}

// Menampilkan jumlah pertukaran yang DIBUTUHKAN agar kartu bisa urut
printf("%d", jumlahTukar);
}

int main(int argc, char const *argv[]) {
    // Inisialisasi jumlah kartu
    int jumlah;
    scanf("%d", &jumlah);

    // Array kartu
    char kartu[jumlah];

```

```
// Array kartu yang dikonversi ke angka
int kartuAngka[jumlah];

// Lakukan input char sebanyak jumlah yang dimasukkan
for (int i = 0; i < jumlah; i++) {
    scanf(" %s", &kartu[i]);
    // Masukkan kartu huruf yang diubah menjadi kartu angka agar bisa
    disorting
    kartuAngka[i] = konversiKartuAngka(&kartu[i]);
}

// Jalankan fungsi sortKartu
sortKartu(kartu, kartuAngka, jumlah);

return 0;
}
```

Output:

```
4
6 6 9 7
Pertukaran ke-1: 6 6 7 9
1
PS C:\laragon\www\tugas-asd-praktikum-4>
```

```
5
3 2 8 7 4
Pertukaran ke-1: 2 3 8 7 4
Pertukaran ke-2: 2 3 4 7 8
2
PS C:\laragon\www\tugas-asd-praktikum-4>
```

```
6
10 J K Q 3 2
Pertukaran ke-1: 2 J K Q 3 10
Pertukaran ke-2: 2 3 K Q J 10
Pertukaran ke-3: 2 3 10 Q J K
Pertukaran ke-4: 2 3 10 J Q K
4
PS C:\laragon\www\tugas-asd-praktikum-4>
```

Soal 2

Code dan penejelasan:

```
#include <stdio.h>
```

```

// Fungsi untuk mencari kemungkinan perpindahan bidak kuda dalam pola L
void koboImaginaryChess(int baris, int kolom, int size, int *papan) {
    // Inisialisasi algoritma perhitungan kemungkinan perpindahan
    int x[8] = { 2, 1, -1, -2, -2, -1, 1, 2 };
    int y[8] = { 1, 2, 2, 1, -1, -2, -2, -1 };

    // Algoritma untuk mencari cell yang kemungkinan bisa ditempati
    for (int i = 0; i < size; i++) {
        // Logika dari operasi ini adalah menjumlahkan baris dan kolom yang
        // dipilih dengan array x dan y sesuai dengan index iterasi
        // agar membentuk pola L yang bisa dilakukan oleh bidak kuda
        int kemungkinanX = baris + x[i];
        int kemungkinanY = kolom + y[i];

        // Cek apakah posisi kemungkinan bidak kuda yang baru masih berada di
        // dalam papan 8x8, jika iya maka nilainya diubah menjadi 1
        if (kemungkinanX >= 0 && kemungkinanX < size && kemungkinanY >= 0 &&
            kemungkinanY < size) {
            // Parameter papan berisi alamat pertama dari array papan 8x8, disini
            // kita membaca array papan 2 dimensi sebagai 1 dimensi
            // dengan rumus "(baris yang dicari * jumlah kolom) + kolom yang
            // dicari" sehingga jika posisi kuda adalah baris 2 kolom 2,
            // maka kemungkinan pertamanya berada pada index "(4 * 8) + 3 = 35"
            // yang dimana index 35 dalam array papan 8x8 adalah [4][3]
            // yang nilainya kemudian akan diubah menjadi angka 1
            papan[(kemungkinanX * size) + kemungkinanY] = 1;
        }
    }
}

int main() {
    // Inisialisasi papan catur 8x8 dengan nilai awal 0 untuk setiap cellnya
    int papan[8][8] = { {0} };

    // Inisialisasi posisi awal bidak kuda yang akan dipindahkan
    int posisi[2];
    for (int i = 0; i < 2; i++) {
        scanf(" %d", &posisi[i]);
    }

    // Jalankan fungsi koboImaginaryChess
    // Posisi[0] merupakan baris, posisi[1] merupakan kolom, dan &papan[0][0]
    // merupakan alamat dari isi papan yang pertama
    koboImaginaryChess(posisi[0], posisi[1], 8, &papan[0][0]);

    // Menampilkan papan 8x8 dengan kemungkinan posisi baru bidak kuda yang
    // nilainya telah diganti dengan 1

```

```

for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        printf("%d ", papan[i][j]);
    }
    printf("\n");
}

return 0;
}

```

Output:

```

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 C:\laragon\www\tugas-asd-praktikum-4>

```

```

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 C:\laragon\www\tugas-asd-praktikum-4>

```