

Nama : I Made Suta Eka Dharma

Kelas : IF 03-01

NIM : 1203230072

## SOAL LATIHAN PRAKTIKUM ARRAY, POINTER, DAN FUNGSI

### KARTU

#### Source Code

```
1. #include <stdio.h>
2. #include <string.h>
3.
4. int get_card_value(char card) {
5.     if (card == 'J') return 11;
6.     else if (card == 'Q') return 12;
7.     else if (card == 'K') return 13;
8.     else if (card == '1') return 10;
9.     else return card - '0';
10.}
11.
12.int main() {
13.    int n, i, j, min_swaps = 0;
14.    char cards[100];
15.    scanf("%d", &n);
16.    for (i = 0; i < n; i++) {
17.        scanf(" %c", &cards[i]);
18.    }
19.    for (i = 0; i < n; i++) {
20.        int min_idx = i;
21.        for (j = i + 1; j < n; j++) {
22.            if (get_card_value(cards[j]) <
get_card_value(cards[min_idx])) {
23.                min_idx = j;
24.            }
25.        }
26.        if (min_idx != i) {
27.            char temp = cards[i];
28.            cards[i] = cards[min_idx];
29.            cards[min_idx] = temp;
30.            min_swaps++;
31.            printf("pertukaran ke-%d: ", min_swaps);
32.            for (int k = 0; k < n; k++) {
33.                printf("%c ", cards[k]);
34.            }
35.            printf("\n");
```

```

36.     }
37. }
38. printf("%d\n", min_swaps);
39. return 0;
40.}

```

Output :

```

PS C:\Users\Asus\vscode> cd "c:\Users\Asus\vscode\eko\" ; if ($?) { gcc tmb1.c -o tmb1 } ; if ($?) { .\tmb1 }
4
6 6 9 7
pertukaran ke-1: 6 6 7 9
1
PS C:\Users\Asus\vscode\eko> cd "c:\Users\Asus\vscode\eko\" ; if ($?) { gcc tmb1.c -o tmb1 } ; if ($?) { .\tmb1 }
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:\Users\Asus\vscode\eko> cd "c:\Users\Asus\vscode\eko\" ; if ($?) { gcc tmb1.c -o tmb1 } ; if ($?) { .\tmb1 }
6
10 J K Q 3 2
pertukaran ke-1: 0 1 J K Q 3
pertukaran ke-2: 0 3 J K Q 1
pertukaran ke-3: 0 3 1 K Q J
pertukaran ke-4: 0 3 1 J Q K
4
PS C:\Users\Asus\vscode\eko> cd "c:\Users\Asus\vscode\eko\" ; if ($?) { gcc tmb1.c -o tmb1 } ; if ($?) { .\tmb1 }
8
9 4 2 J K 8 4 Q
pertukaran ke-1: 2 4 9 J K 8 4 Q
pertukaran ke-2: 2 4 4 J K 8 9 Q
pertukaran ke-3: 2 4 4 8 K J 9 Q
pertukaran ke-4: 2 4 4 8 9 J K Q
pertukaran ke-5: 2 4 4 8 9 J Q K
5

```

Penjelasan Code :

- Fungsi `get_card_value(char card)`: Fungsi ini mengembalikan nilai numerik dari sebuah kartu. Jika kartu adalah 'J', 'Q', 'K', atau 'A', maka nilai numeriknya diatur sesuai dengan aturan yang telah disebutkan. Jika bukan, maka fungsi ini mengembalikan nilai numerik kartu tersebut dalam bentuk integer. Ini digunakan untuk membandingkan nilai kartu saat pengurutan.
- `main()`: Fungsi utama program.
- Mendeklarasikan variabel-variabel yang diperlukan seperti `n` untuk jumlah kartu, `min_swaps` untuk jumlah pertukaran, dan array `cards` untuk menyimpan kartu-kartu tersebut.
- Menggunakan `scanf` untuk membaca jumlah kartu (`n`) yang akan dimasukkan.
- Menggunakan loop `for` untuk membaca karakter kartu satu per satu dan menyimpannya dalam array `cards`.
- Melakukan pengurutan kartu dengan menggunakan algoritma selection sort.
- Setiap kali ada pertukaran, jumlah pertukaran (`min_swaps`) akan bertambah dan kartu-kartu yang sudah diurutkan akan dicetak.
- Output program berupa jumlah pertukaran yang terjadi.

## CATUR

Source Code :

```
#include <stdio.h>

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Menginisialisasi semua elemen array dengan nilai 0
    for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            *(chessBoard + x * size + y) = 0;
        }
    }

    // Menghitung kemungkinan posisi yang dapat dilalui oleh bidak kuda
    int moves[8][2] = { {-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2},
{1, -2}, {1, 2} };
    for (int k = 0; k < 8; k++) {
        int newX = i + moves[k][0];
        int newY = j + moves[k][1];
        if (newX >= 0 && newX < size && newY >= 0 && newY < size) {
            *(chessBoard + newX * size + newY) = 1;
        }
    }

    // Menampilkan output array
    for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            printf("%d ", *(chessBoard + x * size + y));
        }
        printf("\n");
    }
}

int main() {
    int i, j;
    scanf("%d %d", &i, &j);
    int chessBoard[8][8];
    koboImaginaryChess(i, j, 8, (int *)chessBoard);
    return 0;
}
```

Output :

```
PS C:\Users\Asus\vscode> cd "c:\Users\Asus\vscode\eko\" ; if ($?) { gcc catur.c -o catur } ; if ($?) { .\catur }
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:\Users\Asus\vscode\eko> cd "c:\Users\Asus\vscode\eko\" ; if ($?) { gcc catur.c -o catur } ; if ($?) { .\catur }
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:\Users\Asus\vscode\eko>
```

Penjelasan Code :

- Fungsi ini koboImaginaryChessmengambil empat parameter: idan jmewakili posisi awal ksatria di papan catur, sizemewakili ukuran papan catur, dan chessBoardmerupakan penunjuk ke array bilangan bulat yang mewakili papan catur.
- Fungsi ini menginisialisasi chessBoardarray dengan mengatur semua elemennya ke 0.
- Fungsi ini menghitung kemungkinan posisi yang dapat dicapai ksatria dalam 2 gerakan. Ia menggunakan array movesuntuk menyimpan kemungkinan gerakan ksatria. Setiap gerakan direpresentasikan sebagai sepasang bilangan bulat (dx, dy), dimana dx mewakili perubahan koordinat x ksatria dan dy mewakili perubahan koordinat y.
- Fungsi ini mengulangi setiap kemungkinan gerakan dan menghitung posisi baru ksatria. Jika posisi baru berada dalam batas papan catur, fungsi akan menyetel elemen terkait dalam chessBoardlarik ke 1.
- Fungsi ini mencetak chessBoardarray sebagai matriks 8x8.
- Fungsi tersebut mainmengambil posisi awal ksatria sebagai masukan dari pengguna dan memanggil koboImaginaryChessfungsi dengan parameter yang diberikan.

