## LAPORAN TUGAS PRAKTIKUM

Nama : Fandi Ardiansyah

Kelas : IF-03-03

NIM : 1203230079

KOMPONEN PENILAIAN	YA	TIDAK
Soal 1 sesuai dengan output yang diinginkan		
Soal 2 sesuai dengan output yang diinginkan		
Bonus soal 1 dikerjakan		

# Soal no. 1

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
struct Kartu {
    char nilai;
};
void swap(struct Kartu *a, struct Kartu *b) {
    struct Kartu temp = *a;
    *a = *b;
    *b = temp;
void printArray(struct Kartu arr[], int n) {
    for (int i = 0; i < n; i++) {
        printf("%c ", arr[i].nilai);
    printf("\n");
int minSteps(struct Kartu arr[], int n) {
    int steps = 0;
    for (int i = 0; i < n - 1; i++) {
        for (int j = 0; j < n - i - 1; j++) {
            if (arr[j].nilai > arr[j + 1].nilai) {
                swap(&arr[j], &arr[j + 1]);
                printf("\nPertukaran %d: ", steps);
                printArray(arr, n);
    return steps;
```

```
int main() {
    printf("Masukkan jumlah kartu: ");
    int n;
    scanf("%d", &n);

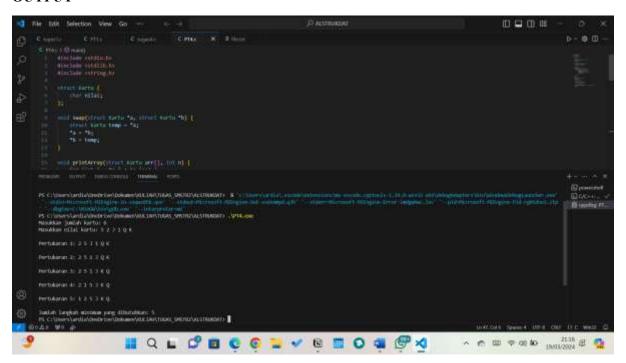
    struct Kartu arr[n];

    printf("Masukkan nilai kartu: ");
    for (int i = 0; i < n; i++) {
        scanf(" %c", &arr[i].nilai);
    }

    int steps = minSteps(arr, n);
    printf("\nJumlah langkah minimum yang dibutuhkan: %d\n", steps);

    return 0;
}</pre>
```

#### **OUTPUT**



Berikut adalah penjelasan mengenai kode di atas baris per baris:

- 1. **#include <stdio.h>**: Ini adalah preprocessor directive yang mengizinkan program untuk menggunakan fungsi-fungsi standar input/output (I/O) dari library stdio.h. Library ini menyediakan fungsi seperti printf dan scanf.
- 2. **#include <stdlib.h>**: Preprocessor directive ini memungkinkan program untuk menggunakan fungsi-fungsi standar dari library stdlib.h. Library ini berisi fungsi-fungsi seperti alokasi memori dinamis (malloc, calloc, realloc) dan konversi string ke angka (atoi, atof).

- 3. **struct Kartu**: Ini adalah definisi dari sebuah struktur bernama Kartu. Struktur ini memiliki satu anggota yaitu nilai dengan tipe data char.
- 4. **void swap(struct Kartu \*a, struct Kartu \*b)**: Ini adalah definisi dari sebuah fungsi bernama swap. Fungsi ini bertujuan untuk menukar dua elemen dari tipe data struct Kartu. Parameter a dan b adalah pointer ke dua elemen yang akan ditukar.
- 5. **void printArray(struct Kartu arr[], int n)**: Ini adalah definisi dari fungsi printArray. Fungsi ini akan mencetak elemen-elemen dari array arr dengan ukuran n.
- 6. int minSteps(struct Kartu arr[], int n): Ini adalah definisi dari fungsi minSteps. Fungsi ini akan mengurutkan array arr menggunakan algoritma Bubble Sort dan menghitung jumlah langkah yang diperlukan untuk mengurutkannya. Fungsi ini mengembalikan jumlah langkah tersebut.
- 7. **int main()**: Ini adalah fungsi utama dari program. Semua kode yang dieksekusi akan berada di dalam fungsi ini.
- 8. **printf("Masukkan jumlah kartu: ");**: Ini mencetak pesan ke layar untuk meminta pengguna memasukkan jumlah kartu.
- 9. **scanf("%d", &n);**: Ini membaca input dari pengguna berupa jumlah kartu dan menyimpannya dalam variabel n.
- 10. **struct Kartu arr[n]**;: Ini mendeklarasikan array arr dengan ukuran sebanyak n elemen, di mana setiap elemen adalah tipe data struct Kartu.
- 11. **printf("Masukkan nilai kartu: ");**: Ini mencetak pesan ke layar untuk meminta pengguna memasukkan nilai-nilai kartu.
- 12. **scanf(" %c", &arr[i].nilai);**: Ini membaca input dari pengguna berupa karakter dan menyimpannya dalam anggota nilai dari setiap elemen array arr.
- 13. **int steps = minSteps(arr, n)**;: Ini memanggil fungsi minSteps untuk mengurutkan array arr dan menghitung jumlah langkah yang diperlukan.
- 14. printf("\nJumlah langkah minimum yang dibutuhkan: %d\n", steps);: Ini mencetak jumlah langkah minimum yang diperlukan untuk mengurutkan kartu-kartu.
- 15. **return 0**;: Ini mengakhiri program dan mengembalikan nilai 0 sebagai tanda bahwa program berjalan dengan sukses.

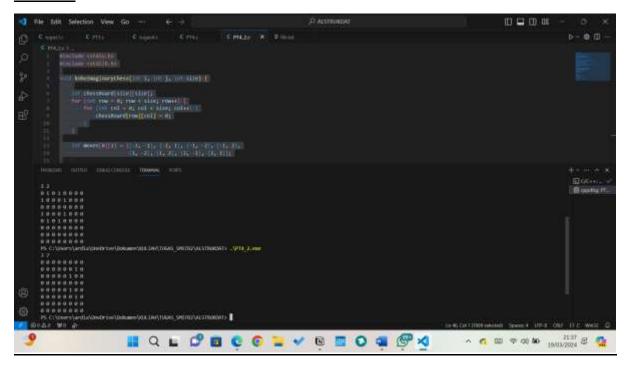
### Soal no. 2

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

void koboImaginaryChess(int i, int j, int size) {
   int chessBoard[size][size];
   for (int row = 0; row < size; row++) {
      for (int col = 0; col < size; col++) {
        chessBoard[row][col] = 0;
    }
}</pre>
```

```
int moves[8][2] = \{\{-2, -1\}, \{-2, 1\}, \{-1, -2\}, \{-1, 2\},
                         \{1, -2\}, \{1, 2\}, \{2, -1\}, \{2, 1\}\};
    for (int k = 0; k < 8; k++) {
        int newRow = i + moves[k][0];
        int newCol = j + moves[k][1];
        if (newRow >= 0 \&\& newRow < size \&\& newCol >= 0 \&\& newCol < size) {
             chessBoard[newRow][newCol] = 1;
    for (int row = 0; row < size; row++) {</pre>
        for (int col = 0; col < size; col++) {</pre>
             printf("%d", chessBoard[row][col]);
             if (col < size - 1) {
                 printf(" ");
        printf("\n");
int main() {
    int i, j;
    scanf("%d %d", &i, &j);
    int size = 8;
    koboImaginaryChess(i, j, size);
    return 0;
```

#### **OUTPUT**



Berikut penjelasan baris per baris untuk kode di atas:

- 1. '#include <stdio.h>': Ini adalah preprocessor directive yang memasukkan header file standar untuk fungsi input-output dalam bahasa C, yang diperlukan untuk fungsi input-output seperti 'printf()' dan 'scanf()'.
- 2. '#include <stdlib.h>': Ini adalah preprocessor directive yang memasukkan header file standar untuk fungsi-fungsi umum dalam bahasa C, seperti alokasi memori, konversi angka, dan fungsi-fungsi lainnya yang didefinisikan dalam library ini.
- 3. 'void koboImaginaryChess(int i, int j, int size) {': Ini adalah definisi fungsi 'koboImaginaryChess()' yang mengambil tiga parameter: 'i', 'j', dan 'size'.
- 4. 'int chessBoard[size][size];': Membuat sebuah matriks dua dimensi 'chessBoard' dengan ukuran 'size' x 'size'.
- 5. 'for (int row = 0; row < size; row++) {': Ini adalah loop for untuk mengiterasi setiap baris dalam papan catur.
- 6. 'for (int col = 0; col < size; col++) {': Ini adalah loop for bersarang untuk mengiterasi setiap kolom dalam papan catur.

- 7. 'chessBoard[row][col] = 0;': Setiap sel dalam papan catur diinisialisasi dengan nilai 0.
- 8. 'int moves[8][2] = {{-2, -1}, {-2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}, {2, -1}, {2, 1}};': Array 'moves' digunakan untuk menyimpan langkah-langkah yang mungkin dari setiap posisi knight dalam bentuk offset baris dan kolom.
- 9. 'for (int k = 0; k < 8; k++) {': Ini adalah loop for untuk mengiterasi setiap langkah yang mungkin.
- 10. 'int newRow = i + moves[k][0];' dan 'int newCol = j + moves[k][1];': Menghitung baris dan kolom baru untuk setiap langkah yang mungkin.
- 11. 'if (newRow >= 0 && newRow < size && newCol >= 0 && newCol < size) {': Memeriksa apakah langkah yang dihasilkan berada di dalam papan catur.
- 12. `chessBoard[newRow][newCol] = 1;`: Jika langkah berada dalam papan catur, maka sel yang sesuai dalam `chessBoard` ditandai sebagai 1.
- 13. Loop-loop yang sama seperti pada poin 5-7 digunakan untuk mencetak papan catur yang telah dimodifikasi ke layar.
- 14. 'int main() {': Fungsi utama dari program dimulai di sini.
- 15. 'int i, j; scanf("%d %d", &i, &j); ': Mengambil input dari pengguna untuk posisi awal knight dalam papan catur.
- 16. 'int size = 8;': Mendefinisikan ukuran papan catur sebagai 8x8.
- 17. `koboImaginaryChess(i, j, size);`: Memanggil fungsi `koboImaginaryChess()` untuk menampilkan papan catur yang dimodifikasi.
- 18. `return 0;`: Mengindikasikan bahwa program telah berhasil dijalankan dan mengembalikan nilai 0 kepada sistem operasi.