

## LAPRAK ASD ARRAY, POINTER, FUNGSI

Nama : Ilham Prakosa

Nim : 1203230027

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

## 1. penjelasan Source Code

1. `#include <stdio.h>`: Ini adalah preprocessor directive yang memasukkan konten dari file header `stdio.h`, yang menyediakan fungsi-fungsi standar untuk input dan output seperti `printf()` dan `scanf()`.
2. `#include <string.h>`: Ini adalah preprocessor directive yang memasukkan konten dari file header `string.h`, yang menyediakan fungsi-fungsi untuk manipulasi string, meskipun dalam program ini, tampaknya tidak digunakan.
3. `void print_cards(int n, char cards[], int swap_count) {`: Ini adalah definisi fungsi `print_cards`. Fungsi ini bertujuan untuk mencetak kartu-kartu yang disimpan dalam array `cards` sebanyak `n` kartu, serta menampilkan jumlah pertukaran yang sudah dilakukan (`swap_count`).
4. `printf("Pertukaran %d: ", swap_count);`: Ini mencetak pesan "Pertukaran <nomor pertukaran>: " ke layar, di mana <nomor pertukaran> digantikan dengan nilai dari `swap_count`.
5. `for (int i = 0; i < n; i++) {`: Ini adalah awal dari loop `for` yang bertujuan untuk mencetak setiap kartu dalam array `cards`.
6. `printf("%c ", cards[i]);`: Ini mencetak kartu yang berada pada indeks `i` dalam array `cards`.
7. `printf("\n");`: Ini mencetak newline (baris baru) setelah selesai mencetak semua kartu.
8. `int min_swaps(int n, char cards[]) {`: Ini adalah definisi fungsi `min_swaps`. Fungsi ini menghitung jumlah minimum pertukaran yang diperlukan untuk mengurutkan kartu-kartu dalam array `cards` secara terurut.
9. `int swaps = 0;`: Ini mendeklarasikan variabel `swaps` yang akan digunakan untuk menghitung jumlah pertukaran yang telah dilakukan.
10. `for (int i = 0; i < n && swaps < 5; i++) {`: Ini adalah awal dari loop `for` yang akan melakukan proses pemilihan dan pertukaran kartu. Loop ini akan berhenti jika sudah dilakukan 5 pertukaran atau jika semua kartu telah diperiksa.
11. `int correct = i;`: Variabel `correct` digunakan untuk menyimpan indeks dari kartu yang dianggap sudah berada di posisi yang benar.
12. `for (int j = i + 1; j < n; j++) {`: Ini adalah awal dari loop `for` yang akan mencari kartu dengan nilai terkecil dari kartu pada indeks `i`.
13. `if (cards[j] < cards[correct]) {`: Ini membandingkan nilai kartu pada indeks `j` dengan nilai kartu pada indeks `correct`, jika nilai `cards[j]` lebih kecil, maka `correct` akan diperbarui menjadi `j`.
14. `if (correct != i) {`: Setelah selesai mencari kartu dengan nilai terkecil, kondisi ini memeriksa apakah kartu pada indeks `i` sudah benar-benar berada di posisi yang benar.
15. `swaps++;`: Jika kartu pada indeks `i` tidak berada di posisi yang benar, maka jumlah `swaps` akan ditambah satu.
16. `char temp = cards[i]; cards[i] = cards[correct]; cards[correct] = temp;`: Ini bertujuan untuk menukar posisi kartu pada indeks `i` dengan kartu pada indeks `correct`.

17. `if (swaps < 5) { print_cards(n, cards, swaps); } else { print_cards(n, cards, 5); }`: Ini memanggil fungsi `print_cards` untuk mencetak status kartu setelah dilakukan pertukaran. Jika jumlah pertukaran masih kurang dari 5, maka jumlah pertukaran yang dicetak sesuai dengan `swaps`, jika sudah mencapai 5 pertukaran, maka yang dicetak adalah 5.
18. `return swaps;`: Fungsi `min_swaps` mengembalikan nilai dari variabel `swaps`, yaitu jumlah total pertukaran yang dilakukan.
19. `int main() {`: Ini adalah awal dari fungsi `main`, yang merupakan titik awal eksekusi program.
20. `scanf("%d", &n);`: Ini meminta input dari pengguna berupa jumlah kartu (`n`).
21. `char cards[n];`: Ini mendeklarasikan array `cards` dengan ukuran sebanyak `n`, yang akan digunakan untuk menyimpan nilai-nilai kartu.
22. `for (int i = 0; i < n; i++) { scanf(" %c", &cards[i]); }`: Ini adalah loop yang meminta input nilai-nilai kartu dari pengguna dan menyimpannya dalam array `cards`.
23. `int swaps = min_swaps(n, cards);`: Ini memanggil fungsi `min_swaps` untuk menghitung jumlah minimum pertukaran yang diperlukan untuk mengurutkan kartu-kartu dalam array `cards`.
24. `printf("%d\n", swaps);`: Ini mencetak jumlah minimum pertukaran yang diperlukan ke layar.
25. `return 0;`: Ini adalah akhir dari fungsi `main`, yang mengindikasikan bahwa program telah berakhir dengan sukses.

## 1.Out put

```

4
PS C:\TugasMahaDewa\output> & .\TGSMT2.11.exe'
4
6 6 9 7
Pertukaran 1: 6 6 7 9
1
PS C:\TugasMahaDewa\output> & .\TGSMT2.11.exe'
5
3 2 8 7 4
Pertukaran 1: 2 3 8 7 4
Pertukaran 2: 2 3 4 7 8
2
PS C:\TugasMahaDewa\output> & .\TGSMT2.11.exe'
6
10 J Q 3 2
Pertukaran 1: 0 1 J Q 3 2
Pertukaran 2: 0 1 2 Q 3 J
Pertukaran 3: 0 1 2 3 Q J
Pertukaran 4: 0 1 2 3 J Q
4

```

```

8
9 4 2 J k 8 4 Q
Pertukaran 1: 2 4 9 J k 8 4 Q
Pertukaran 2: 2 4 4 J k 8 9 Q
Pertukaran 3: 2 4 4 8 k J 9 Q
Pertukaran 4: 2 4 4 8 9 J k Q
Pertukaran 5: 2 4 4 8 9 J Q k
5

```

## 2. Pejelasan Source Code

1. `#include <stdio.h>`: Mendefinisikan preprocessor directive untuk memasukkan file header standar `stdio.h`, yang berisi deklarasi fungsi standar untuk input dan output, seperti `printf()` dan `scanf()`.
2. `#include <stdlib.h>`: Mendefinisikan preprocessor directive untuk memasukkan file header standar `stdlib.h`, yang berisi fungsi-fungsi umum, seperti alokasi memori dan konversi angka.
3. `void koboImaginaryChess(int i, int j, int size, int *chessBoard) {`: Ini adalah deklarasi sebuah fungsi bernama `koboImaginaryChess` yang mengambil empat parameter: dua integer (`i` dan `j`), satu integer `size`, dan satu pointer ke array integer (`chessBoard`). Fungsi ini tidak mengembalikan nilai (`void`).
4. `for (int row = 0; row < size; row++) {`: Perulangan untuk menginisialisasi setiap elemen pada papan catur menjadi 0, dimulai dari baris pertama hingga baris ke- $(size-1)$ .
5. `for (int col = 0; col < size; col++) {`: Perulangan bersarang untuk menginisialisasi setiap elemen dalam setiap baris menjadi 0, dimulai dari kolom pertama hingga kolom ke- $(size-1)$ .
6. `*(chessBoard + row * size + col) = 0;`: Mengatur nilai elemen papan catur ke 0 menggunakan aritmatika pointer.
7. `int moves[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1}, {2, 1}, {1, 2}, {-1, 2}, {-2, 1}};`  
Mendefinisikan sebuah array 2 dimensi yang berisi langkah-langkah mungkin untuk kuda di papan catur.
8. `for (int move = 0; move < 8; move++) {`: Perulangan untuk melakukan langkah pada setiap arah yang mungkin.
9. `int newRow = i + moves[move][0];`: Menghitung posisi baris baru berdasarkan langkah pada arah tertentu.
10. `int newCol = j + moves[move][1];`: Menghitung posisi kolom baru berdasarkan langkah pada arah tertentu.
11. `if (newRow >= 0 && newRow < size && newCol >= 0 && newCol < size) {`: Memeriksa apakah langkah yang dihasilkan masih berada dalam batas papan catur.
12. `*(chessBoard + newRow * size + newCol) = 1;`: Mengatur nilai elemen papan catur yang sesuai dengan langkah yang dihasilkan menjadi 1.
13. `for (int row = 0; row < size; row++) {`: Perulangan untuk mencetak papan catur setelah semua langkah diterapkan.
14. `for (int col = 0; col < size; col++) {`: Perulangan bersarang untuk mencetak setiap elemen dalam setiap baris.
15. `printf("%d ", *(chessBoard + row * size + col));`: Mencetak nilai dari setiap elemen papan catur, dipisahkan oleh spasi.
16. `printf("\n");`: Mencetak newline untuk mengakhiri setiap baris papan catur.
17. `int main() {`: Mendefinisikan fungsi `main()` sebagai titik awal program.

18. `int i, j;`: Deklarasi variabel `i` dan `j` yang akan digunakan untuk menunjukkan posisi awal kuda.
19. `scanf("%d %d", &i, &j);`: Membaca input posisi awal kuda dari pengguna.
20. `int size = 8;`: Deklarasi variabel `size` yang menunjukkan ukuran papan catur (8x8).
21. `int *chessBoard = (int *)malloc(size * size * sizeof(int));`: Mengalokasikan memori dinamis untuk papan catur menggunakan `malloc()`.
22. `koboImaginaryChess(i, j, size, chessBoard);`: Memanggil fungsi `koboImaginaryChess()` untuk menampilkan papan catur dengan langkah-langkah yang mungkin dari posisi awal kuda.
23. `free(chessBoard);`: Membebaskan memori yang dialokasikan untuk papan catur menggunakan `free()`.
24. `return 0;`: Mengembalikan nilai 0 untuk menandakan bahwa program berakhir dengan sukses.

## 2.Out put

```
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:\TugasMahaDewa\output> & .\TGSMT2.12.exe'
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:\TugasMahaDewa\output> & .\TGSMT2.12.exe'
5 4
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 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
PS C:\TugasMahaDewa\output> 
```

