

Source code 1

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

// Fungsi untuk mengonversi kartu menjadi angka (untuk mempermudah perbandingan)
int convertCardValue(char card) {
    if (card >= '2' && card <= '9')
        return card - '0';
    else if (card == '1' || card == 'T')
        return 10;
    else if (card == 'J')
        return 11;
    else if (card == 'Q')
        return 12;
    else if (card == 'K')
        return 13;
    return -1; // Kartu invalid
}

// Fungsi untuk menukar dua nilai dalam array
void swap(int *a, int *b) {
    int temp = *a;
    *a = *b;
    *b = temp;
}

// Fungsi untuk mencetak array
void printArray(int *arr, int size) {
    for (int i = 0; i < size; i++) {
        printf("%d", arr[i]);
    }
    printf("\n");
}

// Fungsi untuk mengurutkan kartu menggunakan algoritma Bubble Sort
int bubbleSort(int *cards, int n) {
    int steps = 0;
```

```

for (int i = 0; i < n - 1; i++) {
    for (int j = 0; j < n - i - 1; j++) {
        if (cards[j] > cards[j + 1]) {
            // Swap cards
            swap(&cards[j], &cards[j + 1]);
            steps++;

            // Menampilkan urutan setiap perulangan
            printf("Langkah %d: ", steps);
            printArray(cards, n);
        }
    }
}
return steps;
}

int main() {
    int N;
    scanf("%d", &N);

    // Membaca kartu-kartu yang dimiliki Refan
    int *cards = (int*)malloc(N * sizeof(int));
    char input[6]; // Mengubah ukuran array input untuk menangani string kartu (5 karakter) dan
    terminasi null
    scanf("%s", input);

    // Memproses input kartu dan mengonversi menjadi nilai numerik
    for (int i = 0; i < N; i++) {
        cards[i] = convertCardValue(input[i]);
    }

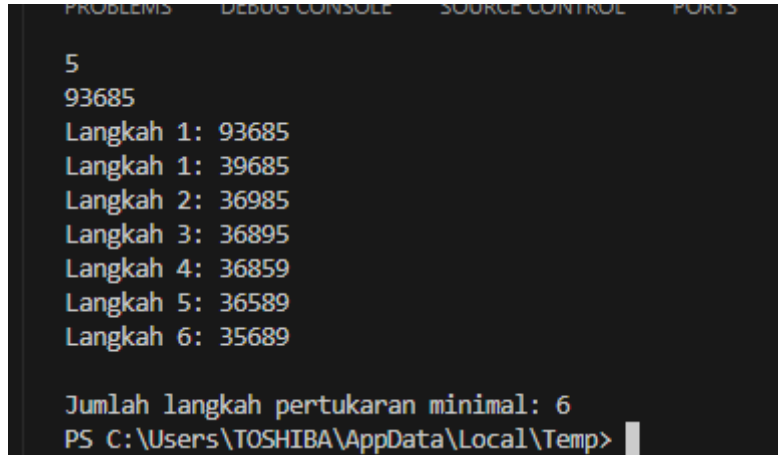
    // Mengurutkan kartu menggunakan Bubble Sort dan menghitung jumlah langkah pertukaran
    printf("Langkah 1: ");
    printArray(cards, N);
    int minSteps = bubbleSort(cards, N);
    printf("\nJumlah langkah pertukaran minimal: %d\n", minSteps);

    free(cards); // Membebaskan memori yang dialokasikan untuk array kartu

```

```
    return 0;
}
```

OUTPUT



```
5
93685
Langkah 1: 93685
Langkah 1: 39685
Langkah 2: 36985
Langkah 3: 36895
Langkah 4: 36859
Langkah 5: 36589
Langkah 6: 35689

Jumlah langkah pertukaran minimal: 6
PS C:\Users\TOSHIBA\AppData\Local\Temp>
```

PENJELASAN CODE

1. `#include <stdio.h>`: Mendefinisikan header file standar input-output untuk fungsi-fungsi input-output standar.
2. `#include <stdlib.h>`: Mendefinisikan header file standar library untuk alokasi memori dinamis dan fungsi-fungsi umum lainnya.
3. `#include <string.h>`: Mendefinisikan header file standar library untuk fungsi-fungsi manipulasi string.
4. `int convertCardValue(char card) { ... }`: Mendefinisikan fungsi `convertCardValue` untuk mengonversi karakter kartu menjadi nilai numerik.
5. `void swap(int *a, int *b) { ... }`: Mendefinisikan fungsi `swap` untuk menukar dua nilai dalam array.
6. `void printArray(int *arr, int size) { ... }`: Mendefinisikan fungsi `printArray` untuk mencetak isi array ke layar.
7. `int bubbleSort(int *cards, int n) { ... }`: Mendefinisikan fungsi `bubbleSort` untuk mengurutkan array kartu menggunakan algoritma Bubble Sort.
8. `int main() { ... }`: Fungsi utama dari program.
9. `int N; scanf("%d", &N);`: Membaca jumlah kartu yang akan diurutkan.
10. `int *cards = (int*)malloc(N * sizeof(int));`: Mengalokasikan memori untuk menyimpan nilai kartu.
11. `char input[6]; scanf("%s", input);`: Membaca string input yang merupakan representasi kartu-kartu.
12. `for (int i = 0; i < N; i++) { ... }`: Memproses string input dan mengonversi setiap karakter kartu menjadi nilai numerik.

13. `printf("Langkah 1: "); printArray(cards, N);`: Mencetak array kartu sebelum diurutkan.
14. `int minSteps = bubbleSort(cards, N);`: Memanggil fungsi `bubbleSort` untuk mengurutkan array kartu.
15. `printf("\nJumlah langkah pertukaran minimal: %d\n", minSteps);`: Mencetak jumlah langkah pertukaran minimal yang dilakukan untuk mengurutkan kartu.
16. `free(cards);`: Membuang memori yang telah dialokasikan untuk array kartu sebelumnya.

Source Code 2

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

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

```
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {  
    // Array untuk menyimpan pergerakan kuda  
    int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};  
  
    // Melakukan pengecekan dan menandai posisi yang dapat dicapai oleh kuda  
    for (int k = 0; k < 8; k++) {  
        int x = i + moves[k][0];  
        int y = j + moves[k][1];  
        if (x >= 0 && x < size && y >= 0 && y < size) {  
            chessBoard[x * size + y] = 1;  
        }  
    }  
  
    // Menampilkan papan catur setelah simulasi pergerakan kuda  
    printf("Papan catur setelah simulasi pergerakan kuda:\n");  
    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 size = 8;
int *chessBoard = (int*)malloc(size * size * sizeof(int));

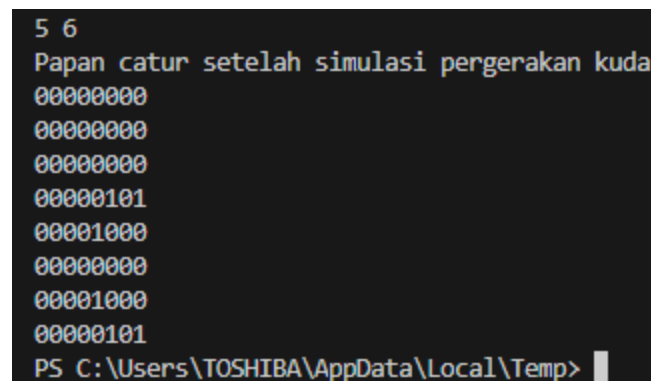
// Menginisialisasi papan catur dengan nilai
for (int k = 0; k < size * size; k++) {
    chessBoard[k] = 0;
}

// Memanggil fungsi koboImaginaryChess untuk mensimulasikan pergerakan kuda
koboImaginaryChess(i, j, size, chessBoard);

free(chessBoard); // Membebaskan memori yang dialokasikan untuk papan
return 0;
}

```

OUTPUT



```

5 6
Papan catur setelah simulasi pergerakan kuda
00000000
00000000
00000000
00000101
00001000
00000000
00001000
00000101
PS C:\Users\TOSHIBA\AppData\Local\Temp>

```

PENJELASAN

1. `#include <stdio.h>`: Mendefinisikan header file standar input-output.
2. `#include <stdlib.h>`: Mendefinisikan header file standar library untuk alokasi memori dinamis.
3. `void koboImaginaryChess(int i, int j, int size, int *chessBoard) { ... }`: Deklarasi fungsi `koboImaginaryChess` untuk mensimulasikan pergerakan kuda pada papan catur.
4. `int main() { ... }`: Fungsi utama dari program.
5. `int i, j; scanf("%d %d", &i, &j);`: Membaca input koordinat awal kuda dari pengguna.
6. `int size = 8;`: Mendefinisikan ukuran papan catur (8x8).
7. `int *chessBoard = (int*)malloc(size * size * sizeof(int));`: Mengalokasikan memori untuk papan catur menggunakan `malloc`.
8. `for (int k = 0; k < size * size; k++) { chessBoard[k] = 0; }`: Menginisialisasi papan catur dengan nilai 0.

9. `koboImaginaryChess(i, j, size, chessBoard);`: Memanggil fungsi `koboImaginaryChess` untuk mensimulasikan pergerakan kuda.
10. `free(chessBoard);`: Membebaskan memori yang dialokasikan untuk papan catur setelah digunakan.

NAMA : FARREL AHNAF KHAYLA PRAPTAMA

NIM : 1203230100

KLS : IF 03-03