NAMA : AWRA SANK RAMA

NIM : 1203230014

KELAS : IF.03.01

1.INPUT

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

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

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

// Fungsi untuk mengurutkan kartu dan mengembalikan jumlah minimum pertukaran
int urutKartu(char kartu[], int size) {
    int minTukar = 0;

    // Loop untuk setiap elemen di array
    for (int i = 0; i < size - 1; i++) {
        int minIndex = i;

        // Temukan indeks elemen terkecil dari sisa array
        for (int j = i + 1; j < size; j++) {
            // Membuat urutan "1-10-J-Q-K"
            char urutan[] = "123456789JQK";
            if (strchr(urutan, kartu[j]) < strchr(urutan, kartu[minIndex])) {
                minIndex = j;
            }
        }

        // Jika elemen terkecil tidak di posisi saat ini, tukar mereka
        if (minIndex != i) {
            tukar(&kartu[i], &kartu[minIndex]);
            minTukar++; // Tambahkan jumlah pertukaran
            printf("Pertukaran ke-%d: ", minTukar);
            printArray(kartu, size);
```

```
        }
    }

    return minTukar;
}

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

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

    int minTukar = urutKartu(kartu, noKartu);

    printf("Jumlah minimum pertukaran: %d\n", minTukar);

    return 0;
}
```

OUTPUT :

```
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc latihan.c -o latihan } ; if ($?) { .\latihan }
Masukkan jumlah kartu: 4
Masukkan nilai kartu : 6 6 9 7
Pertukaran ke-1: 6 6 7 9
Jumlah minimum pertukaran: 1
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc latihan.c -o latihan } ; if ($?) { .\latihan }
Masukkan jumlah kartu: 5
Masukkan nilai kartu : 3 2 8 7 4
Pertukaran ke-1: 2 3 8 7 4
Pertukaran ke-2: 2 3 4 7 8
Jumlah minimum pertukaran: 2
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc latihan.c -o latihan } ; if ($?) { .\latihan }
Masukkan jumlah kartu: 6
Masukkan nilai kartu : 10 J K Q 3 2
Pertukaran ke-1: 0 1 J K Q 3
Pertukaran ke-2: 0 1 3 K Q J
Pertukaran ke-3: 0 1 3 J Q K
Jumlah minimum pertukaran: 3
PS D:\CODING> cd "d:\CODING\" ; if ($?) { gcc latihan.c -o latihan } ; if ($?) { .\latihan }
Masukkan jumlah kartu: 8
Masukkan nilai kartu : 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
Jumlah minimum pertukaran: 5
PS D:\CODING>
```

PENJELASAN :

Program yang diberikan adalah sebuah program dalam bahasa C yang bertujuan untuk mengurutkan kartu sesuai dengan urutan "1-10-J-Q-K" dan menghitung jumlah minimum

pertukaran yang diperlukan untuk mengurutkan kartu tersebut. Berikut adalah cara kerja program tersebut:

1. Pertama-tama, program meminta pengguna untuk memasukkan jumlah kartu yang akan diurutkan melalui pernyataan printf("Masukkan jumlah kartu: ") dan scanf("%d", &noKartu);. Nilai ini disimpan dalam variabel noKartu.

2. Selanjutnya, program meminta pengguna untuk memasukkan nilai dari setiap kartu melalui pernyataan printf("Masukkan nilai kartu : ") dan loop for. Setiap nilai kartu disimpan dalam array kartu.

3. Setelah semua kartu dimasukkan, fungsi urutKartu() dipanggil untuk mengurutkan kartu. Fungsi ini mengembalikan jumlah minimum pertukaran yang diperlukan untuk mengurutkan kartu tersebut.

4. Di dalam fungsi urutKartu(), dilakukan pengurutan menggunakan algoritma selection sort. Algoritma ini mencari elemen terkecil dari array yang belum diurutkan dan menukarnya dengan elemen pertama. Proses ini diulangi untuk setiap elemen di array.

5. Saat melakukan perbandingan untuk menemukan elemen terkecil, urutan kartu "1-10-J-Q-K" dibandingkan dengan menggunakan fungsi strchr(). Ini dilakukan untuk menentukan urutan yang benar dari kartu, karena karakter '1' hingga '9' merupakan urutan numerik, kemudian diikuti oleh 'J', 'Q', dan 'K'. Jika elemen yang dibandingkan dalam array kartu memiliki urutan yang tidak sesuai, maka mereka akan ditukar.

6. Setiap kali ada pertukaran elemen, program mencetak langkah pertukaran tersebut dengan memanggil fungsi printArray().

7. Setelah semua kartu diurutkan, jumlah minimum pertukaran yang diperlukan untuk mengurutkan kartu tersebut dicetak ke layar.

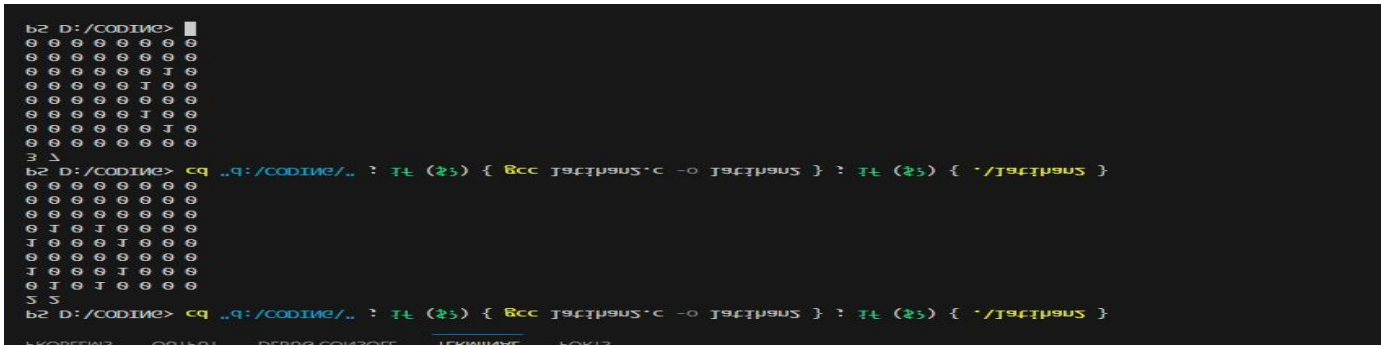
Program ini memiliki kompleksitas waktu O(n^2) karena menggunakan algoritma selection sort. Artinya, waktu eksekusi program akan meningkat secara kuadratik seiring dengan peningkatan jumlah kartu yang dimasukkan.

2.INPUT

```c
#include <stdio.h>

// Fungsi untuk mencetak papan catur
void printBoard(int *chessBoard, int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            printf("%d ", *((chessBoard+i*size) + j));
        }
        printf("\n");
    }
}

// Fungsi untuk menandai posisi yang dapat dicapai oleh bidak kuda
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Gerakan bidak kuda
    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 x = i + moves[k][0];
        int y = j + moves[k][1];

        // Cek apakah posisi berada di dalam papan catur
        if (x >= 0 && x < size && y >= 0 && y < size) {
            *((chessBoard+x*size) + y) = 1;
        }
    }
}

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

    int size = 8;
    int chessBoard[size][size];

    // Inisialisasi papan catur dengan 0
    for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            chessBoard[x][y] = 0;
        }
    }
```

```
    // Menandai posisi yang dapat dicapai oleh bidak kuda
    koboImaginaryChess(i, j, size, (int *)chessBoard);

    // Mencetak papan catur
    printBoard((int *)chessBoard, size);

    return 0;
}
```

OUTPUT :



PENEJLASAN :

Program ini merupakan program dalam bahasa C yang bertujuan untuk menampilkan papan catur serta menandai posisi yang dapat dicapai oleh sebuah bidak kuda yang ditempatkan di koordinat (i, j) yang dimasukkan oleh pengguna.

Berikut adalah cara kerja program tersebut:

1. Pada fungsi printBoard(), papan catur direpresentasikan sebagai matriks dengan ukuran size x size. Fungsi ini mencetak papan catur ke layar dengan mengakses setiap elemen matriks dan mencetak nilainya. Papan catur direpresentasikan dalam bentuk angka, di mana angka 0 menunjukkan sel yang belum dijangkau oleh bidak kuda, dan angka 1 menunjukkan sel yang dapat dijangkau oleh bidak kuda.

2. Pada fungsi koboImaginaryChess(), dilakukan penandaan terhadap sel-sel yang dapat dicapai oleh bidak kuda. Untuk setiap posisi (i, j) yang diberikan, fungsi ini menghitung posisi-posisi yang dapat dicapai oleh bidak kuda dari posisi tersebut. Gerakan bidak kuda diwakili oleh array moves, di mana setiap elemen array adalah pasangan koordinat

pergeseran relatif dari posisi saat ini. Fungsi ini memperbarui nilai sel-sel yang dapat dijangkau oleh bidak kuda menjadi 1.

3. Pada fungsi main(), koordinat (i, j) untuk bidak kuda dimasukkan oleh pengguna menggunakan scanf(). Kemudian, papan catur diinisialisasi sebagai matriks size x size dengan semua elemen diisi dengan nilai 0, yang menandakan bahwa awalnya tidak ada sel yang dapat dijangkau oleh bidak kuda.

4. Setelah itu, fungsi koboImaginaryChess() dipanggil dengan koordinat bidak kuda (i, j) dan papan catur yang telah diinisialisasi. Hal ini akan menandai posisi-posisi yang dapat dicapai oleh bidak kuda di papan catur.

5. Terakhir, papan catur dicetak ke layar menggunakan fungsi printBoard().

Dengan demikian, setelah memasukkan koordinat bidak kuda, program akan mencetak papan catur yang menunjukkan posisi-posisi yang dapat dicapai oleh bidak kuda dari posisi tersebut.

TERIMAKASIH