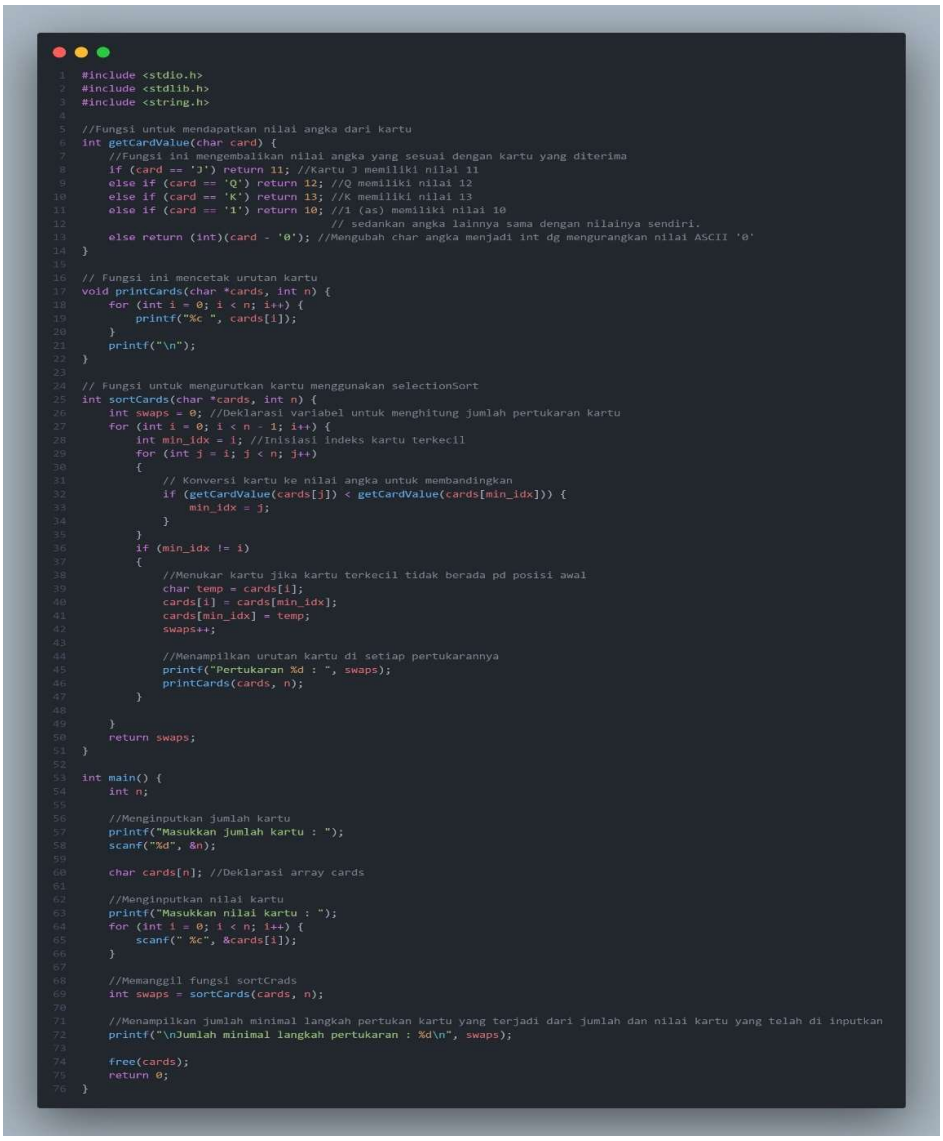
Nama : Hafizhah Nur Zahira
NIM : 1203230024
Kelas : IF 03-02

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

1. Mengurutkan Kartu
    - SS Source Code



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

//Fungsi untuk mendapatkan nilai angka dari kartu
int getCardValue(char card) {
    //Fungsi ini mengembalikan nilai angka yang sesuai dengan kartu yang diterima
    if (card == 'J') return 11; //Kartu J memiliki nilai 11
    else if (card == 'Q') return 12; //Q memiliki nilai 12
    else if (card == 'K') return 13; //K memiliki nilai 13
    else if (card == '1') return 10; //1 (as) memiliki nilai 10
                                     // sedankan angka lainnya sama dengan nilainya sendiri.
    else return (int)(card - '0'); //Mengubah char angka menjadi int dg mengurangkan nilai ASCII '0'
}

// Fungsi ini mencetak urutan kartu
void printCards(char *cards, int n) {
    for (int i = 0; i < n; i++) {
        printf("%c ", cards[i]);
    }
    printf("\n");
}

// Fungsi untuk mengurutkan kartu menggunakan selectionSort
int sortCards(char *cards, int n) {
    int swaps = 0; //Deklarasi variabel untuk menghitung jumlah pertukaran kartu
    for (int i = 0; i < n - 1; i++) {
        int min_idx = i; //Inisiasi indeks kartu terkecil
        for (int j = i; j < n; j++)
        {
            // Konversi kartu ke nilai angka untuk membandingkan
            if (getCardValue(cards[j]) < getCardValue(cards[min_idx])) {
                min_idx = j;
            }
        }
        if (min_idx != i)
        {
            //Menukar kartu jika kartu terkecil tidak berada pd posisi awal
            char temp = cards[i];
            cards[i] = cards[min_idx];
            cards[min_idx] = temp;
            swaps++;

            //Menampilkan urutan kartu di setiap pertukarannya
            printf("Pertukaran %d : ", swaps);
            printCards(cards, n);
        }

    }
    return swaps;
}

int main() {
    int n;

    //Menginputkan jumlah kartu
    printf("Masukkan jumlah kartu : ");
    scanf("%d", &n);

    char cards[n]; //Deklarasi array cards

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

    //Memanggil fungsi sortCrads
    int swaps = sortCards(cards, n);

    //Menampilkan jumlah minimal langkah pertukan kartu yang terjadi dari jumlah dan nilai kartu yang telah di inputkan
    printf("\nJumlah minimal langkah pertukaran : %d\n", swaps);

    free(cards);
    return 0;
}
```

- Penjelasan

```
//Fungsi untuk mendapatkan nilai angka dari kartu

int getCardValue(char card) {
if (card == 'J') return 11;
else if (card == 'Q') return 12;
else if (card == 'K') return 13;
else if (card == '1') return 10;
else return (int)(card - '0');
}
```

Pertama terdapat fungsi getCardsValue. Pada fungsi getCardsValue ini berguna untuk mendapatkan nilai angka dari sebuah kartu. Fungsi ini mengembalikan nilai angka yang sesuai dengan kartu yang diterima sebagai parameter. Kartu J memiliki nilai 11, Q memiliki nilai 12, K memiliki nilai 13, 1 (as) memiliki nilai 10, dan angka lainnya sama dengan nilainya sendiri. Lalu pada return kita mengubah karakter angka menjadi integer dengan mengurangkan nilai ASCII '0'.

```
// Fungsi ini mencetak urutan kartu
void printCards(char *cards, int n) {
    for (int i = 0; i < n; i++) {
        printf("%c ", cards[i]);
    }
    printf("\n");
}
```

Kedua terdapat fungsi printCards. Fungsi printCards ini akan mencetak urutan kartu yang diterima sebagai parameter, dengan panjang length. Setiap kartu dipisahkan oleh spasi.

```
// Fungsi untuk mengurutkan kartu menggunakan selectionSort
int sortCards(char *cards, int n) {
    int swaps = 0; //Deklarasi variabel untuk menghitung jumlah pertukaran
kartu
    for (int i = 0; i < n - 1; i++) {
        int min_idx = i; //Inisiasi indeks kartu terkecil
        for (int j = i; j < n; j++)
        {
            // Konversi kartu ke nilai angka untuk membandingkan
            if (getCardValue(cards[j]) < getCardValue(cards[min_idx])) {
                min_idx = j;
            }
        }
        if (min_idx != i)
        {
            //Menukar kartu jika kartu terkecil tidak berada pd posisi awal
            char temp = cards[i];
            cards[i] = cards[min_idx];
            cards[min_idx] = temp;
            swaps++;
```

```
            //Menampilkan urutan kartu di setiap pertukarannya
            printf("Pertukaran %d : ", swaps);
            printCards(cards, n);
        }

    }
    return swaps;
}
```

Ketiga terdapat fungsi sortCards. Fungsi ini untuk mengurutkan kartu menggunakan algoritma selection sort. Fungsi ini akan mencari kartu terkecil untuk di letakkan di posisi awal. Jika kartu terkecil tidak berada di posisi awal, maka kartu akan menukar posisi. Setelah dirasa posisi sudah bener maka fungsi akan menampilkan urutan kartu dengan memanggil fungsi printCards disetiap pertukarannya.

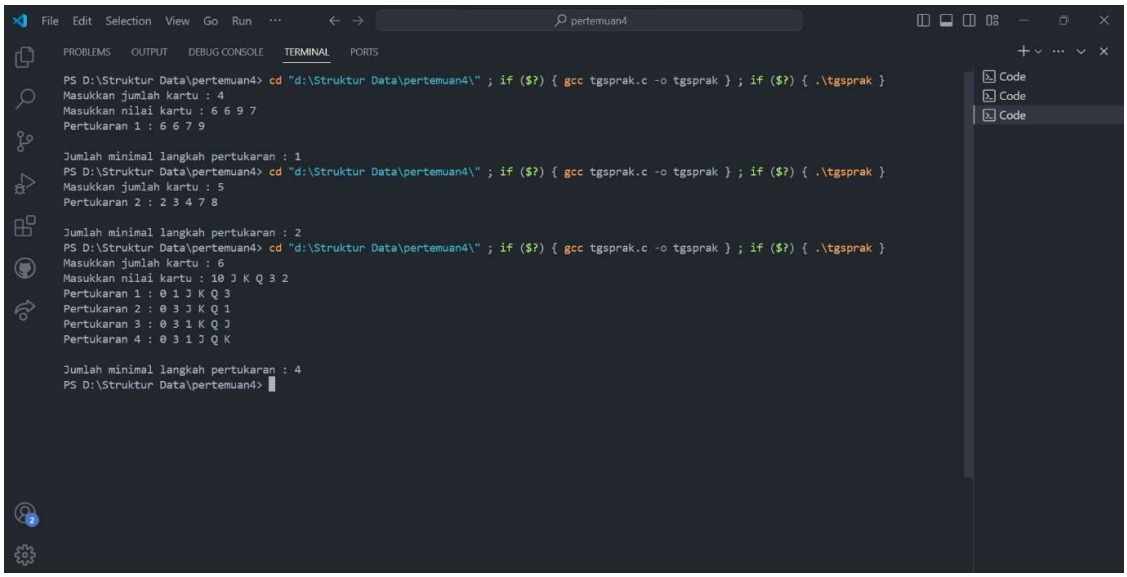
```
int main() {
    int n;

    //Menginputkan jumlah kartu
    printf("Masukkan jumlah kartu : ");
    scanf("%d", &n);

    char cards[n]; //Deklarasi array cards

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

    //Memanggil fungsi sortCrads
    int swaps = sortCards(cards, n);

    //Menampilkan jumlah minimal langkah pertukan kartu yang terjadi dari
jumlah dan nilai kartu yang telah di inputkan
    printf("\nJumlah minimal langkah pertukaran : %d\n", swaps);

    free(cards);
    return 0;
}
```

Selanjutnya yaitu inti program. Pada inti program, user diminta untuk menginputkan jumlah kartu dan menginputkan nilai kartu. Kemudian inputan user tersebut akan diproses dalam fungsi sortCards untuk melakukan pengurutan kartu dan akan menampilkan urutan kartu disetiap pertukarannya. Lalu program juga menghitung dan menampilkan jumlah minimun langkah pertukaran kartunya.

```
    free(cards);
```

Dalam inti program ini terdapat free(cards) yang digunakan untuk membebaskan memori yang telah dialokasikan setelah selesai digunakan.

- SS output



- Output Bonus

2. Perpindahan posisi kuda catur
    - SS Source Code

```c
#include <stdio.h>

//Fungsi untuk mengecek apakah posisi (x, y) valid pada papan catur 8x8
int valid(int x, int y) {
    return (x >= 0 && x < 8 && y >= 0 && y < 8);
}

//Fungsi untuk menandai semua langkah yang mungkin dilakukan oleh kuda pada papan catur
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    //Langkah-langkah yang mungkin dilakukan oleh kuda
    int geserX[] = {2, 1, -1, -2, -2, -1, 1, 2};
    int geserY[] = {1, 2, 2, 1, -1, -2, -2, -1};

    //Menandai setiap langkah yang mungkin dilakukan oleh kuda
    for (int k = 0; k < 8; k++) {
        int lanjutX = i + geserX[k];
        int lanjutY = j + geserY[k];
        if (valid(lanjutX, lanjutY)) {
            *(chessBoard + lanjutX * size + lanjutY) = 1; // Menandai langkah kuda dengan nilai 1
        }
    }
}

//Fungsi untuk menampilkan papan catur
void printBoard(int size, int chessBoard[][size]){
     for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            printf("%d ", chessBoard[x][y]);
        }
        printf("\n");
    }
}

int main() {
    int i, j;

    //Menginputkan posisi i dan j
    printf("Masukkan posisi i dan j : ");
    scanf("%d %d", &i, &j);

    int size = 8; // Ukuran papan catur
    int chessBoard[size][size]; // Array 2D untuk papan catur

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

    //Memanggil Fungsi untuk menandai setiap langkah kuda
    koboImaginaryChess(i, j, size, (int *)chessBoard);

    //Memanggil Fungsi menampilkan papan catur
    printBoard(size, chessBoard);

    return 0;
}
```

- Penjelasan

```c
//Fungsi untuk mengecek apakah posisi (x, y) valid pada papan catur 8x8
int valid(int x, int y) {
return (x >= 0 && x < 8 && y >= 0 && y < 8);
}
```

Pertama terdapat fungsi valid. Fungsi ini digunakan untuk mengecek apakah posisi (x, y) valid pada papan catur 8x8. Posisi dianggap valid jika x dan y berada di dalam rentang 0-7.

```
//Fungsi untuk menandai semua langkah yang mungkin dilakukan oleh kuda pada
papan catur
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    //Langkah-langkah yang mungkin dilakukan oleh kuda
    int geserX[] = {2, 1, -1, -2, -2, -1, 1, 2};
    int geserY[] = {1, 2, 2, 1, -1, -2, -2, -1};

    //Menandai setiap langkah yang mungkin dilakukan oleh kuda
    for (int k = 0; k < 8; k++) {
        int lanjutX = i + geserX[k];
        int lanjutY = j + geserY[k];
        if (valid(lanjutX, lanjutY)) {
            *(chessBoard + lanjutX * size + lanjutY) = 1; // Menandai langkah
kuda dengan nilai 1
        }
    }
}
```

Kedua terdapat void koboImaginaryChess. Fungsi ini untuk menandai semua langkah yang mungkin dilakukan oleh kuda pada papan catur. Fungsi menerima parameter posisi awal kuda (i, j), ukuran papan catur size, dan pointer ke array chessBoard yang merupakan representasi papan catur dengan nilai awal 0. Untuk mengakses elemen pada array chessBoard, digunakan notasi pointer aritmetik ***(chessBoard + lanjutX * size + lanjutY)** untuk menandai posisi yang dicapai oleh bidak kuda.

```
    //Langkah-langkah yang mungkin dilakukan oleh kuda
    int geserX[] = {2, 1, -1, -2, -2, -1, 1, 2};
    int geserY[] = {1, 2, 2, 1, -1, -2, -2, -1};
```

Fungsi ini menggunakan array geserX dan geserY untuk menyimpan langkah yang mungkin dilakukan oleh kuda. Elemen dalam array ini menunjukkan perubahan koordinat X dan Y yang akan dilakukan oleh kuda.

```
void printBoard(int size, int chessBoard[][size]){
     for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            printf("%d ", chessBoard[x][y]);
        }
        printf("\n");
    }
}
```

Ketiga terdapat void printBoard. Fungsi ini digunakan untuk menampilkan isi papan catur dengan ukuran dan nilai nilai.

```c
int main() {
    int i, j;

    //Menginputkan posisi i dan j
    printf("Masukkan posisi i dan j : ");
    scanf("%d %d", &i, &j);

    int size = 8; // Ukuran papan catur
    int chessBoard[size][size]; // Array 2D untuk papan catur

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

    //Memanggil Fungsi untuk menandai setiap langkah kuda
    koboImaginaryChess(i, j, size, (int *)chessBoard);

    //Memanggil Fungsi menampilkan papan catur
    printBoard(size, chessBoard);

    return 0;
}
```

Selanjutnya terdapat inti program. Inti program akan meminta inputan dari pengguna untuk memasukkan posisi awal kuda pada papan catur. Pada inti program ini terdapat perulangan digunakan untuk menginisialisasi seluruh elemen pada array 2d chessBoard dengan nilai 0. Kemudian program akan memanggil fungsi koboImaginaryChess untuk menandai setiap langkah kuda dan memanggil fungsi printBoard yang akan menampilkan papan catur.

- SS Output