

MUH RICHARD DANESWARA

IF 03-02

ALSTRUKDAT

1. code

```
#include <stdio.h>

typedef struct kartu {
    int nilai;
} Kartu;

void tukar(Kartu *a, Kartu *b) {
    Kartu temp = *a;
    *a = *b;
    *b = temp;
}

int minLangkah(Kartu *kartu, int n) {
    int langkah = 0;
    for (int i = 0; i < n - 1; i++) {
        int minIdx = i;
        for (int j = i + 1; j < n; j++) {
            if (kartu[j].nilai < kartu[minIdx].nilai) {
                minIdx = j;
            }
        }
        if (minIdx != i) {
            langkah++;
            tukar(&kartu[i], &kartu[minIdx]);
        }
    }
    return langkah;
}

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

    Kartu kartu[n];
    for (int i = 0; i < n; i++) {
        char nilai[3];
        scanf("%s", nilai);
        if (nilai[0] >= '0' && nilai[0] <= '9') {
            kartu[i].nilai = nilai[0] - '0';
        } else {
            switch (nilai[0]) {
                case 'J':
```

```

        kartu[i].nilai = 10;
        break;
    case 'Q':
        kartu[i].nilai = 11;
        break;
    case 'K':
        kartu[i].nilai = 12;
        break;
    }
}
}

int langkah = minLangkah(kartu, n);
printf("%d\n", langkah);

return 0;
}

```

Output

```

PS C:\belajar\.vscode\praktikumalstrukdat> cd "c:\belajar\.vscode\praktikumalstrukdat" ; if ($?) { gcc praktikumarraypointerfungsi
i.c -o praktikumarraypointerfungsi } ; if ($?) { .\praktikumarraypointerfungsi }
4
6 6 9 7
1
PS C:\belajar\.vscode\praktikumalstrukdat> cd "c:\belajar\.vscode\praktikumalstrukdat" ; if ($?) { gcc praktikumarraypointerfungsi
i.c -o praktikumarraypointerfungsi } ; if ($?) { .\praktikumarraypointerfungsi }
5
3 2 8 7 4
2
PS C:\belajar\.vscode\praktikumalstrukdat> cd "c:\belajar\.vscode\praktikumalstrukdat" ; if ($?) { gcc praktikumarraypointerfungsi
i.c -o praktikumarraypointerfungsi } ; if ($?) { .\praktikumarraypointerfungsi }
6
10 J K Q 3 2
4
PS C:\belajar\.vscode\praktikumalstrukdat>

```

Penjelasan

```
if (minIdx != i) {  
    langkah++;  
    tukar(&kartu[i], &kartu[minIdx]);  
}
```

Kondisi ini terjadi ketika kartu minimum (minIdx) tidak berada pada posisi saat ini (i). Dalam hal ini, langkah pertukaran (langkah) akan diincrement dan fungsi tukar akan dipanggil untuk menukar kartu pada posisi i dan minIdx.

```
for (int i = 0; i < n; i++) {  
    char nilai[3];  
    scanf("%s", nilai);  
    if (nilai[0] >= '0' && nilai[0] <= '9') {  
        kartu[i].nilai = nilai[0] - '0';  
    } else {  
        switch (nilai[0]) {  
            case 'J':  
                kartu[i].nilai = 10;  
                break;  
            case 'Q':  
                kartu[i].nilai = 11;  
                break;  
            case 'K':  
                kartu[i].nilai = 12;  
                break;  
        }  
    }  
}
```

Perulangan ini digunakan untuk membaca nilai kartu satu per satu. Di dalam perulangan: nilai[3] digunakan untuk menyimpan string yang mewakili nilai kartu. scanf("%s", nilai) digunakan untuk membaca string dari input pengguna. Jika karakter pertama dari string adalah angka (0 - 9), nilai kartu dikonversi ke integer dan disimpan dalam kartu[i].nilai. Jika karakter pertama bukan angka, nilai kartu dikonversi berdasarkan karakter: J: 10 Q: 11 K: 12

```
• for (int j = i + 1; j < n; j++) {  
•     if (kartu[j].nilai < kartu[minIdx].nilai) {  
•         minIdx = j;  
•     }  
• }  
•
```

Perulangan ini digunakan untuk mencari indeks minimum (minIdx) di sebelah kanan elemen saat ini (i). Di dalam perulangan: Jika nilai kartu pada indeks j lebih kecil daripada nilai kartu pada indeks minIdx, maka minIdx diperbarui menjadi j.

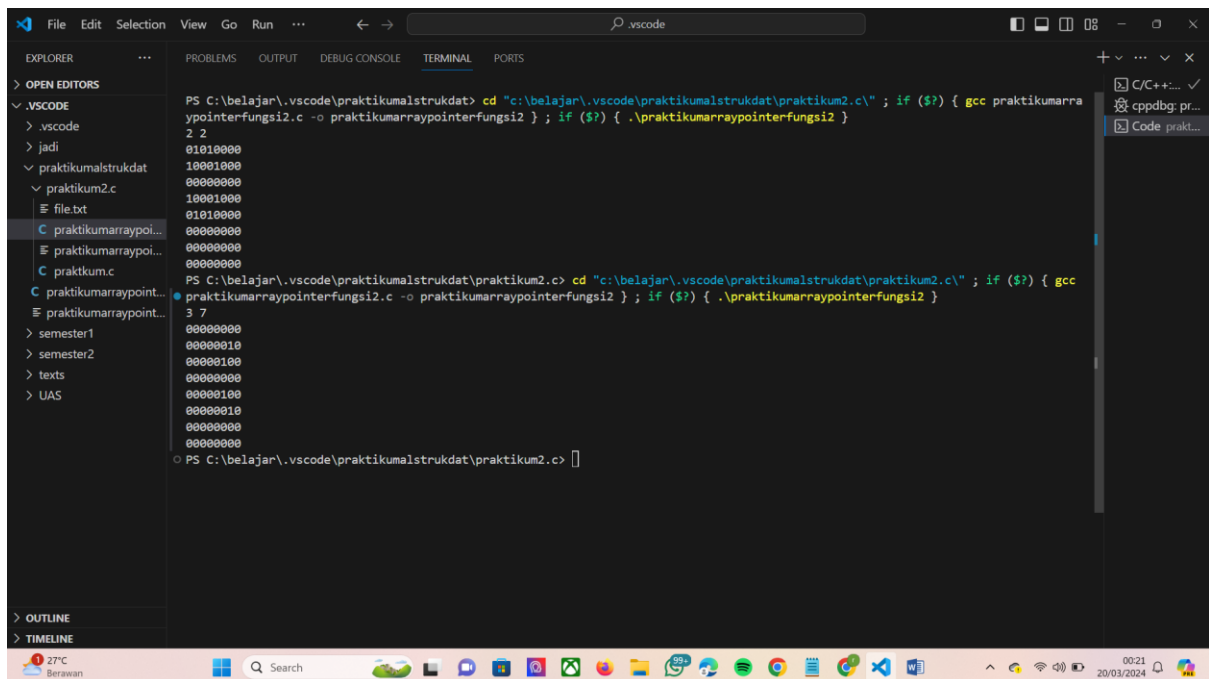
```
Kartu kartu[n];
```

Baris ini mendeklarasikan array kartu dengan tipe Kartu dan ukuran n

2.code

```
#include <stdio.h>
#include <stdlib.h>
int isValidPosition(int x, int y) {
    return (x >= 0 && x < 8 && y >= 0 && y < 8);
}
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    int moves[8][2] = {
        {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
        {-1, -2}, {-1, 2}, {1, -2}, {1, 2}
    };
    int x, y;
    for (int k = 0; k < 8; k++) {
        x = i + moves[k][0];
        y = j + moves[k][1];
        if (isValidPosition(x, y)) {
            chessBoard[x * size + y] = 1;
        }
    }
}
int main() {
    int i, j;
    scanf("%d %d", &i, &j);
    int size = 8;
    int *chessBoard = (int *)malloc(size * size * sizeof(int));
    for (int m = 0; m < size; m++) {
        for (int n = 0; n < size; n++) {
            chessBoard[m * size + n] = 0;
        }
    }
    koboImaginaryChess(i, j, size, chessBoard);
    for (int m = 0; m < size; m++) {
        for (int n = 0; n < size; n++) {
            printf("%d", chessBoard[m * size + n]);
        }
        printf("\n");
    }
    free(chessBoard);
    return 0;
}
```

Output



```
PS C:\belajar\.vscode\praktikumalstrukdat> cd "c:\belajar\.vscode\praktikumalstrukdat\praktikum2\" ; if ($?) { gcc praktikumarraypointerfungsi2.c -o praktikumarraypointerfungsi2 } ; if ($?) { .\praktikumarraypointerfungsi2 }
2 2
01010000
10001000
00000000
10001000
01010000
00000000
00000000
00000000
PS C:\belajar\.vscode\praktikumalstrukdat\praktikum2.c> cd "c:\belajar\.vscode\praktikumalstrukdat\praktikum2\" ; if ($?) { gcc praktikumarraypointerfungsi2.c -o praktikumarraypointerfungsi2 } ; if ($?) { .\praktikumarraypointerfungsi2 }
3 7
00000000
00000100
00000100
00000000
00000100
00000100
00000000
00000000
PS C:\belajar\.vscode\praktikumalstrukdat\praktikum2.c>
```

Penjelasan

```
int isValidPosition(int x, int y) {
    return (x >= 0 && x < 8 && y >= 0 && y < 8);
}
```

Kondisi ini mengecek apakah koordinat x dan y berada dalam rentang yang valid (0-7) pada papan catur 8x8. Fungsi ini mengembalikan nilai 1 (true) jika posisinya valid dan 0 (false) jika tidak.

```
int moves[8][2] = {
    {-2, -1}, {-2, 1}, {2, -1}, {2, 1},
    {-1, -2}, {-1, 2}, {1, -2}, {1, 2}
};
```

Array moves menampung delapan kemungkinan langkah kuda dalam catur (dua langkah dalam satu arah dan satu langkah tegak lurus di delapan arah).

```
for (int k = 0; k < 8; k++) {
    x = i + moves[k][0];
    y = j + moves[k][1];
    if (isValidPosition(x, y)) {
        chessBoard[x * size + y] = 1;
    }
}
```

Perulangan ini iterates through delapan kemungkinan langkah kuda. Di setiap iterasi:

Koordinat baru (x dan y) dihitung dengan menambahkan vektor langkah dari moves ke koordinat awal (i dan j). Fungsi isValidPosition dipanggil untuk memastikan koordinat baru berada dalam papan catur. jika posisinya valid, sel yang sesuai di chessBoard ditandai dengan nilai 1.

```
for (int m = 0; m < size; m++) {  
    for (int n = 0; n < size; n++) {  
        printf("%d", chessBoard[m * size + n]);  
    }  
    printf("\n");  
}
```

Perulangan ini iterates through seluruh elemen chessBoard dan mencetak nilai setiap elemen (0 atau 1) dalam urutan baris-utama. Dan hasilnya menunjukkan visualisasi sel-sel yang terpengaruh pada papan catur.

Komponen penilaian	ya	tidak
Soal 1 sesuai dengan output yang diinginkan		
Soal 2 sesuai dengan output yang diinginkan		
Bonus soal 1 dikerjakan		