

TUGAS OTH PRAKTIKUM

Nama : Nabila Amilatul Jannah

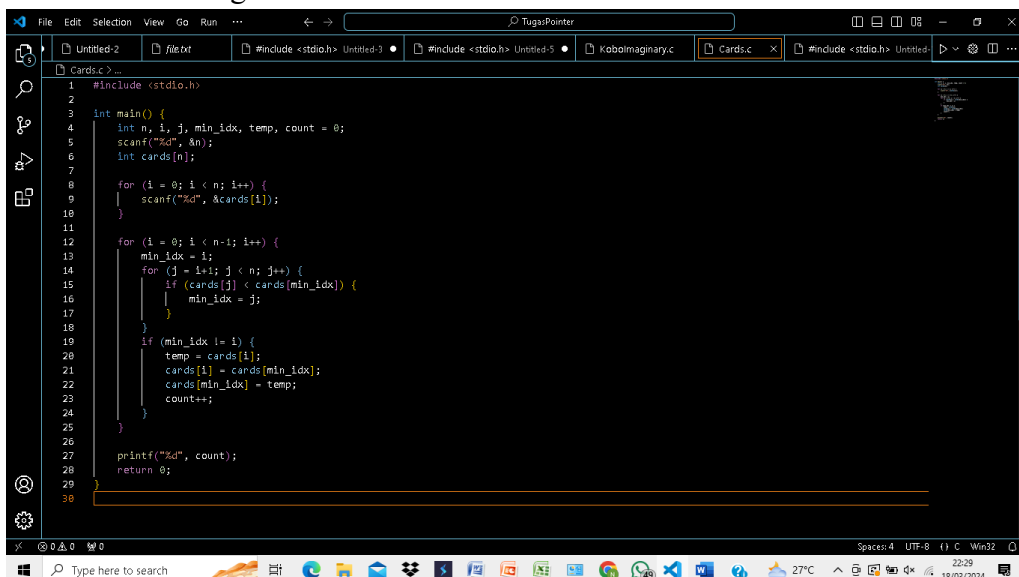
NIM : 1203230103

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		•

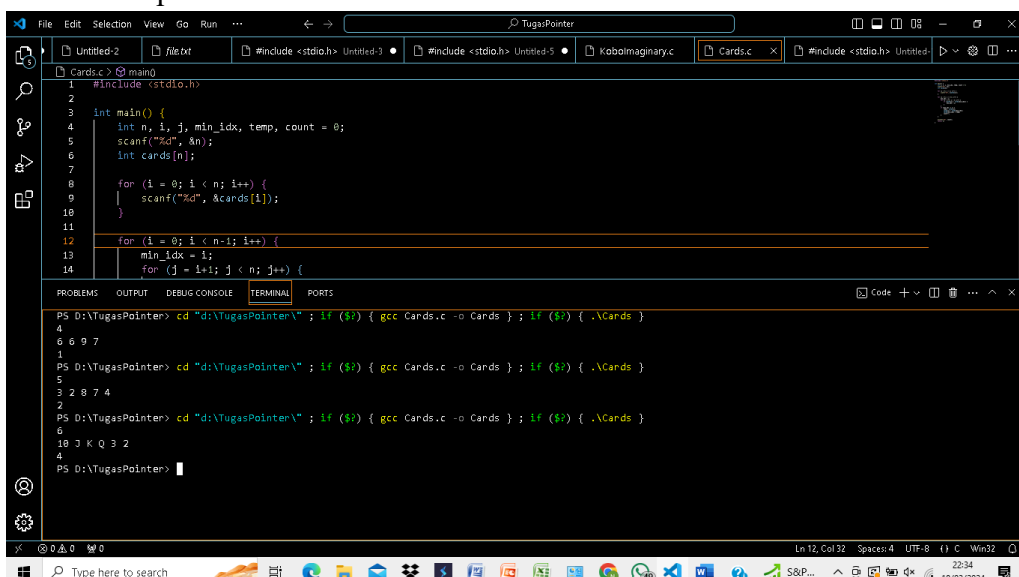
1.

- SS Code Program



```
1 #include <stdio.h>
2
3 int main() {
4     int n, i, j, min_idx, temp, count = 0;
5     scanf("%d", &n);
6     int cards[n];
7
8     for (i = 0; i < n; i++) {
9         scanf("%d", &cards[i]);
10    }
11
12    for (i = 0; i < n-1; i++) {
13        min_idx = i;
14        for (j = i+1; j < n; j++) {
15            if (cards[j] < cards[min_idx]) {
16                min_idx = j;
17            }
18        }
19        if (min_idx != i) {
20            temp = cards[i];
21            cards[i] = cards[min_idx];
22            cards[min_idx] = temp;
23            count++;
24        }
25    }
26
27    printf("%d", count);
28    return 0;
29 }
```

- SS Output



```
PS D:\TugasPointer> cd "d:\TugasPointer\" ; if ($?) { gcc Cards.c -o Cards } ; if ($?) { .\Cards }
6 6 9 7
1
PS D:\TugasPointer> cd "d:\TugasPointer\" ; if ($?) { gcc Cards.c -o Cards } ; if ($?) { .\Cards }
5
3 2 8 7 4
2
PS D:\TugasPointer> cd "d:\TugasPointer\" ; if ($?) { gcc Cards.c -o Cards } ; if ($?) { .\Cards }
10 3 K Q 3 2
4
PS D:\TugasPointer>
```

- Penjelasan Kode Program

```
int n, i, j, min_idx, temp, count = 0;
scanf("%d", &n);
```

‘n’ adalah variabel untuk menyimpan jumlah kartu
 ‘i’, ‘j’ adalah variabel iterasi dalam loop
 ‘min_idx’ variabel untuk menyimpan index terkecil di kartu saat loop
 ‘temp’ variabel sementara untuk membandingkan nilai antara dua kartu
 ‘count’ variabel untuk menyimpan jumlah pertukaran
 Kemudian scanf meminta user memasukkan jumlah kartu dan disimpan dalam variabel n

```
int cards[n];

for (i = 0; i < n; i++) {
    scanf("%d", &cards[i]);
}
```

Program ini membaca nilai dari setiap kartu yang di input oleh user dan menyimpan di dalam array ‘cards’

```
for (i = 0; i < n-1; i++) {
    min_idx = i;
    for (j = i+1; j < n; j++) {
        if (cards[j] < cards[min_idx]) {
            min_idx = j;
        }
    }
    if (min_idx != i) {
        temp = cards[i];
        cards[i] = cards[min_idx];
        cards[min_idx] = temp;
        count++;
    }
}
```

Melakukan pengurutan kartu di array ‘cards’ dengan menggunakan selection sort. Setiap iterasi pertukaran dilakukan dengan mencari kartu terkecil di sisa array yang belum diurutkan dan menukarnya dengan kartu pada posisi saat ini. Variabel ‘count’ digunakan untuk menghitung pertukaran yang dilakukan

```
if (cards[j] < cards[min_idx]) {
    min_idx = j;
}
```

Kode program ini melakukan pemeriksaan apakah nilai kartu pada posisi ke-j lebih kecil dari nilai kartu pada posisi ‘min_idx’. Jika nilai kartu pada posisi ke-j lebih dari

nilai kartu pada 'min_idx', maka 'min_idx' diperbarui menjadi 'j', menunjukkan bahwa kartu pada posisi ke-j adalah kartu terkecil yang ditemukan

```
if (min_idx != i) {  
    temp = cards[i];
```

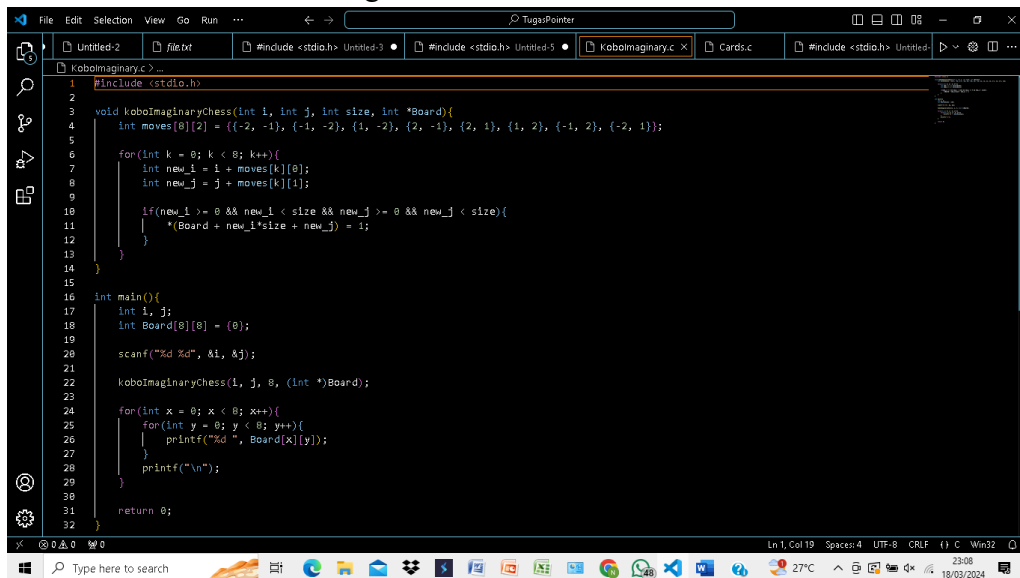
setelah loop kedua selesai, dilakukan pengecekan apakah 'min_idx' tidak sama dengan 'i' yang berarti bahwa nilai terkecil yang ditemukan dalam loop kedua bukanlah nilai pada posisi ke-i. Jika kondisi terpenuhi, maka Langkah pertukaran dilakukan

```
printf("%d", count);  
return 0;
```

program mencetak jumlah pertukaran yang dilakukan agar array menjadi terurut

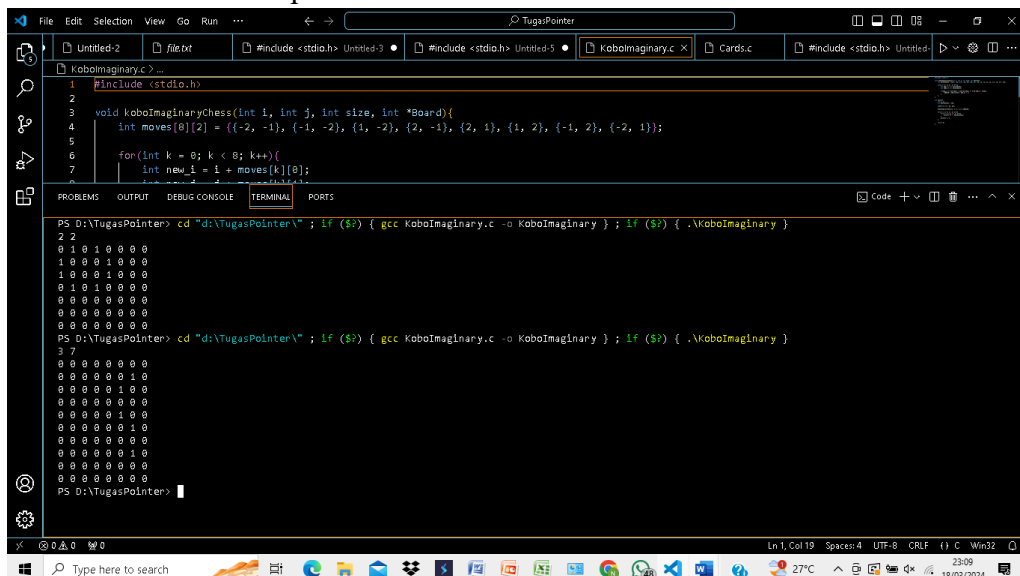
2.

• SS Kode Program



```
1 #include <stdio.h>  
2  
3 void koboImaginaryChess(int i, int j, int size, int *Board){  
4     int moves[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1}, {2, 1}, {1, 2}, {-1, 2}, {-2, 1}};  
5  
6     for(int k = 0; k < 8; k++){  
7         int new_i = i + moves[k][0];  
8         int new_j = j + moves[k][1];  
9  
10        if(new_i >= 0 && new_i < size && new_j >= 0 && new_j < size){  
11            *(Board + new_i*size + new_j) = 1;  
12        }  
13    }  
14 }  
15  
16 int main(){  
17     int i, j;  
18     int Board[8][8] = {0};  
19     scanf("%d %d", &i, &j);  
20  
21     koboImaginaryChess(1, j, 8, (int *)Board);  
22  
23     for(int x = 0; x < 8; x++){  
24         for(int y = 0; y < 8; y++){  
25             printf("%d ", Board[x][y]);  
26         }  
27         printf("\n");  
28     }  
29  
30     return 0;  
31 }  
32
```

• SS Output



```
PS D:\TugasPointer> cd "d:\TugasPointer\" ; If ($?) { gcc KoboImaginary.c -o KoboImaginary } ; If ($?) { .\KoboImaginary }  
2 2  
0 0 0 1 0 0 0 0  
1 0 0 0 1 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  
0 0 0 0 0 0 0 0  
PS D:\TugasPointer> cd "d:\TugasPointer\" ; If ($?) { gcc KoboImaginary.c -o KoboImaginary } ; If ($?) { .\KoboImaginary }  
3 7  
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 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 1 0  
0 0 0 0 0 0 0 0  
PS D:\TugasPointer>
```

- Penjelasan Program

```
void koboImaginaryChess(int i, int j, int size, int *Board){
    int moves[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1}, {2, 1}, {1, 2},
    {-1, 2}, {-2, 1}};
```

koordinat 'i' dan 'j' dari posisi kuda, ukuran papan catur('size') dan pointer ke array papan catur ('Board'). 'int moves [8] [2]' ini mendefinisikan sebuah array dua dimensi dengan 8 baris 2 kolom. Inisialisasi nilai-nilai array di variabel 'moves' mewakili perubahan baris dan kolom yang diperlukan gerakan kuda diatas papan catur seperti membentuk huruf L. Contohnya '{-2, -1}' menunjukkan bahwa kuda bergerak dua baris ke atas dan satu kolom ke kiri dari posisi awal

```
for(int k = 0; k < 8; k++){
    int new_i = i + moves[k][0];
    int new_j = j + moves[k][1];
```

loop for yang akan mengiterasi semua elemen dari array 'moves'. Variabel 'k' digunakan sebagai penghitung loop, dimulai dari 0 dan berakhir pada 7. Variabel 'new_i' dan 'new_j' akan menentukan baris dan kolom baru dimana kuda setelah melakukan gerakan k. Ini dihitung dengan menambahkan nilai array 'moves' ke nilai baris dan kolom saat ini 'i' dan 'j'

```
if(new_i >= 0 && new_i < size && new_j >= 0 && new_j < size){
    *(Board + new_i*size + new_j) = 1;
}
}
```

Ini adalah kondisi memeriksa apakah posisi yang dihasilkan 'new_i' dan 'new_j' berada dalam batas papan catur yang berukuran 'size'x'size' Jika posisi valid, elemen dipapan catur diatur menjadi 1

```
int main(){
    int i, j;
    int Board[8][8] = {0};

    scanf("%d %d", &i, &j);
```

variabel i dan j digunakan untuk menyimpan koordinat awal kuda. Array 'Board' digunakan untuk merepresentasikan papan catur dengan ukuran [8][8]

```
koboImaginaryChess(i, j, 8, (int *)Board);
```

fungsi ini dipanggil dengan parameter koordinat awal kuda, ukuran papan dan pointer array papan catur

```
for(int x = 0; x < 8; x++){
    for(int y = 0; y < 8; y++){
        printf("%d ", Board[x][y]);
```

```
}  
printf("\n");  
}
```

Program ini menggunakan dua loop bersarang. Setiap elemen papan catur dicetak, menunjukkan apakah kuda dapat mencapai posisi tersebut atau tidak. Jika mencapai posisi tersebut maka nilainya 1 sedangkan nilai 0 menunjukkan sebaliknya