

Nama : Haidar Fulca Kurniawan
NIM : 1203230077
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		

TUGAS PRAKTIKUM ALGORITMA STRUKTUR DATA

SOAL 1

- Source Code

```
• #include <stdio.h>
• #include <string.h>
•
• int huruf(char kartu) {
•     if (kartu == 'J' && kartu == 'j') {
•         return 11;
•     } else if (kartu == 'Q' && kartu == 'q') {
•         return 12;
•     } else if (kartu == 'K' && kartu == 'k') {
•         return 13;
•     } else if (kartu == '1') {
•         return 10;
•     } else {
•         return (kartu - '0');
•     }
• }
•
• int selection_sort(char kartu[], int n) {
•     int tukar = 0;
•
```

```

•   for (int i = 0; i < n; i++) {
•       int minim = i;
•
•       for (int j = i + 1; j < n; j++) {
•           if (huruf(kartu[j]) < huruf(kartu[minim])) {
•               minim = j;
•           }
•       }
•       if (minim != i) {
•           char temp = kartu[i];
•           kartu[i] = kartu[minim];
•           kartu[minim] = temp;
•
•           printf("pertukaran %d: ",tukar + 1);
•           for (int k = 0; k < n; k++) {
•               printf("%c ", kartu[k]);
•           }
•           printf("\n");
•
•           tukar++;
•       }
•   } return tukar;
• }
•
• int main() {
•     int n;
•     int swap = 0;
•     char remi[1000];
•
•     printf("Masukkan jumlah kartu: ");
•     scanf("%d", &n);
•
•     printf("Masukkan nilai kartu: ");
•     for (int i = 0; i < n; i++) {
•         scanf(" %c", &remi[i]);
•     }
•
•     swap = selection_sort(remi, n);
•
•     printf("Jumlah minimum iterasi pertukaran kartu: %d\n", swap);
•
•     return 0;
• }
•

```

• Output

```
PS C:\Users\ASUS\Downloads\tugas_alpro> cd "c:\Users\ASUS\Downloads\tugas_alpro\semester 2 praktikum\tgs 3\" ; if ($?) { gcc asd1.c -o asd1 } ;  
if ($?) { .\asd1 }  
Masukkan jumlah kartu: 4  
pertukaran 1: 6 6 7 9  
Jumlah minimum iterasi pertukaran kartu: 1  
PS C:\Users\ASUS\Downloads\tugas_alpro\semester 2 praktikum\tgs 3> cd "c:\Users\ASUS\Downloads\tugas_alpro\semester 2 praktikum\tgs 3\" ; if ($?  
) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }  
Masukkan jumlah kartu: 5  
Masukkan nilai kartu: 3 2 8 7 4  
pertukaran 1: 2 3 8 7 4  
pertukaran 2: 2 3 4 7 8  
Jumlah minimum iterasi pertukaran kartu: 2  
PS C:\Users\ASUS\Downloads\tugas_alpro\semester 2 praktikum\tgs 3> cd "c:\Users\ASUS\Downloads\tugas_alpro\semester 2 praktikum\tgs 3\" ; if ($?  
) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }  
Masukkan jumlah kartu: 8  
Masukkan nilai kartu: 9 4 2 J K 8 4 Q  
pertukaran 1: 2 4 9 J K 8 4 Q  
pertukaran 2: 2 4 4 J K 8 9 Q  
pertukaran 3: 2 4 4 8 K J 9 Q  
pertukaran 4: 2 4 4 8 9 J K Q  
pertukaran 5: 2 4 4 8 9 J Q K  
Jumlah minimum iterasi pertukaran kartu: 5  
PS C:\Users\ASUS\Downloads\tugas_alpro\semester 2 praktikum\tgs 3> |
```

• Penjelasan

1. Program Code C ini berguna untuk mengurutkan sebuah kartu remi dari nilai yang terkecil hingga terbesar. Kartu 1 atau AS bernilai 10, kartu J bernilai 11, kartu Q bernilai 12 dan kartu K bernilai 13.

2. Pada **Main Program** pertama tama di deklarasikan 3 variabel, yaitu **n** yang berfungsi untuk menyimpan jumlah kartu yang dimasukkan, **swap** berfungsi untuk menyimpan value jumlah iterasi yang terjadi dari fungsi **selection_sort** dan **remi** yang berfungsi sebagai array penyimpanan nilai dari kartu kartunya.

3. kemudian selanjutnya dimintai inputan jumlah kartu dan nilai kartu kartunya dengan command **printf** dan **scanf** sebagai cara menyimpannya pada variabel. Pada input an nilai kartu, digunakan **for** loop untuk menyimpan pada array **remi**.

4. selanjutnya pemanggilan fungsi **selection_sort** dengan parameter **remi** dan **n** dilakukan serta value nya disimpan pada variabel **swap**.

- Fungsi **Selection Sort**, fungsi ini berguna untuk mengurutkan nilai nilai kartu dari yang terkecil ke yang terbesar. Cara kerjanya ialah, pertama tama dideklarasikan sebuah variabel **tukar**. Lalu dibuatlah sebuah **nested loop** yang di dalamnya ada variabel **minim** bervalue **i**. kemudian pada loop yang kedua, digunakanlah sebuah logika jika **huruf** (*fungsi yang menggunakan if else untuk mengubah 1 J Q K menjadi sebuah nilai angka*) ke **j** lebih kecil dari huruf ke **minim**, maka value **minim** dirubah menjadi **j**. selanjutnya jika **minim** tidak sama dengan **i** lagi maka dilakukan penukaran value pada array **kartu** dengan menggunakan variabel **temp** sebagai pihak ke 3. Kemudian jika sudah dilakukan pertukaran, setiap iterasi akan di tampilkan dengan command **printf**. Lalu jika **if** terpenuhi maka **tukar** akan bertambah 1. Setelah semua program berjalan fungsi akan dikembalikan pada value **tukar** dan value tersebut ialah jumlah iterasi yang diperlukan untuk mengurutkan nilai nilai dari kartu.

5. Yang terakhir, akan di tampilkan jumlah iterasi yang sudah disimpan pada variabel **swap**.

SOAL 2

- **Source Code**

```
• #include <stdio.h>
•
• void papan_catur(int size, int chessBoard[size][size]) {
•     for (int i = 0; i < size; i++) {
•         for (int j = 0; j < size; j++) {
•             chessBoard[i][j] = 0;
•         }
•     }
• }
•
• void KoboImaginaryChess(int i, int j, int size, int chessBoard[size][size]) {
•     int gerak[8][2] = {{-2, -1}, {-1, -2}, {1, -2}, {2, -1},
•                         {2, 1}, {1, 2}, {-1, 2}, {-2, 1}};
•
•     for (int k = 0; k < 8; k++) {
•         int i_baru = i + gerak[k][0];
•         int j_baru = j + gerak[k][1];
•
•         if (i_baru >= 1 && i_baru <= size && j_baru >= 1 && j_baru <= size) {
•             chessBoard[i_baru - 1][j_baru - 1] = 1;
•         }
•     }
• }
•
• int main() {
•     int i, j;
•     int size = 8;
•     int catur[size][size];
•
•     papan_catur(size, catur);
•
•     printf("Papan catur awal:\n");
•     for (i = 0; i < size; i++) {
•         for (j = 0; j < size; j++) {
•             printf("%d ", catur[i][j]);
•         }
•         printf("\n");
•     }
•
•     printf("Masukkan posisi bidak kuda : ");
•     scanf("%d %d", &i, &j);
• }
```

```

•     KoboImaginaryChess(i, j, size, catur);
•
•     printf("Hasil papan catur setelah pergerakan kuda:\n");
•     for (i = 0; i < size; i++) {
•         for (j = 0; j < size; j++) {
•             printf("%d ", catur[i][j]);
•         }
•         printf("\n");
•     }
•
•     return 0;
• }
•
•

```

• Output

```

PS C:\Users\ASUS\Downloads\tugas alpro\semester 2 praktikum\tgs 3> cd "c:\Users\ASUS\Downloads\tugas alpro\semester 2 praktikum\tgs 3\" ; if ($?) { gcc asd2.c -o asd2 } ; if ($?) { .\asd2 }
Papan catur awal:
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 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
Masukkan posisi bidak kuda : 4 4
Hasil papan catur setelah pergerakan kuda:
0 0 0 0 0 0 0 0
0 0 1 0 1 0 0 0
0 1 0 0 0 1 0 0
0 1 0 0 0 1 0 0
0 0 0 0 0 0 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
PS C:\Users\ASUS\Downloads\tugas alpro\semester 2 praktikum\tgs 3>

```

• Penjelasan

1. Program Code C ini berfungsi untuk memberikan kemungkinan mana saja yang dapat dilalui oleh bidak kuda pada sebuah papan catur dengan berupa tanda angka '1'.
2. Pada **Main Program** pertama tama dideklarasikan 4 variabel, yaitu **i** dan **j** sebagai **looping** dan tempat menyimpan inputan posisi bidak kuda, **size** yang bervalue 8 digunakan sebagai batas serta **catur** yang berguna untuk array dari papan catur itu sendiri.
3. Selanjutnya dipanggil sebuah fungsi **papan_catur** dengan parameter **size** dan **catur**.
 - Fungsi **papan_catur**, fungsi ini berguna untuk membuat sebuah array papan catur yang isi setiap indeks nya ialah 0. Fungsi ini menggunakan **nested loop** dalam metodenya.

4. Setelah setiap indeks dari papan catur berupa 0, ditampilkanlah papan catur tersebut dengan command `printf` dan metode `nested loop`.

5. Lalu akan dimintalah sebuah inputan posisi koordinat kuda dan akan disimpan pada variabel `i` serta `j`.

6. Selanjutnya dipanggil sebuah fungsi `KoboImaginaryChess` dengan parameter `i`, `j`, `size` dan `catur`.

- Fungsi `KoboImaginaryChess`, fungsi ini berguna untuk memberikan kemungkinan mana saja yang dapat dilalui oleh pion kuda. Cara kerjanya, pertama dideklarasikan sebuah variabel `gerak` yang isinya ialah kemungkinan gerak pion kuda (*bisa 2 atas 1 kiri, 2 atas 1 kanan, 1 atas 2 kiri dan lainnya*). Kemudian dibuatlah `for loop` yang berguna untuk mencari 8 kemungkinan (*karena ada 8 tempat yang dapat dilalui oleh pion kuda*) pion kuda dapat bergerak sesuai dengan posisi kuda yang telah diinputkan. Setelah koordinat kemungkinan gerak ditemukan diceklah koordinat tersebut menggunakan `if` (*dengan jika `i` dan `j` berada diantara 1-8*), apakah koordinat masih berada di dalam papan catur. Jika masih, maka value di indeks koordinat tersebut akan dirubah menjadi 1. Dengan catatan indeks koordinat dikurangi 1, karena range dari indeks tersebut masihlah 1-8 sedangkan array nya ialah 0-7.

7. kemudian yang terakhir, papan catur yang sudah ada kemungkinan gerak pion kuda di tampilkan dengan command `printf` dan metode `nested loop`.