

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

Nama: Azka Zafran Andiani

Kelas: IF-03-02

NIM: 1203230021

LAPORAN ASD

A. Soal 1

- Source Code:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 typedef struct card
5 {
6     char ABC[2];
7     int value;
8 }card;
9
10 void sortCard(card Data[], int size)
11 {
12     card temp;
13     for(int i = 0; i < size; i++)
14     {
15         int poin = 0;
16         for(int j = 0; j < size - 1; j++)
17         {
18             if(Data[j].value > Data[j+1].value)
19             {
20                 temp = Data[j];
21                 Data[j] = Data[j+1];
22                 Data[j+1] = temp;
23             }
24             else
25             {
26                 poin++;
27             }
28         }
29         if(poin == (size-1))
30         {
31             printf("%d", i);
32             break;
33         }
34         printf("PERTUKARAN %d:\n", i+1);
35         for(int j = 0; j < size; j++)
36         {
37             printf("%s ", Data[j].ABC);
38         }
39         printf("\n");
40     }
41 }
```



```
1  int main()
2  {
3      int n;
4      printf("Berapa kartu yang diinginkan: ");
5      scanf("%d", &n);
6      card deck[n];
7      printf("Masukkan baris kartu anda: ");
8      for(int i = 0; i < n; i++)
9      {
10         scanf(" %s", &deck[i].ABC);
11     }
12     for(int i = 0; i < n; i++)
13     {
14         deck[i].value = atoi(deck[i].ABC);
15         if(deck[i].value == 0)
16         {
17             switch(deck[i].ABC[0])
18             {
19                 case 'J':
20                     deck[i].value = 11;
21                     break;
22                 case 'Q':
23                     deck[i].value = 12;
24                     break;
25                 case 'K':
26                     deck[i].value = 13;
27                     break;
28                 default:
29                     break;
30             }
31         }
32     }
33     sortCard(deck, n);
34
35     return 0;
36 }
```

- Output:

```
Berapa kartu yang diinginkan: 6
Masukkan baris kartu anda: 10 J K Q 2 3
PERTUKARAN 1:
10 J Q 2 3 K
PERTUKARAN 2:
10 J 2 3 Q K
PERTUKARAN 3:
10 2 3 J Q K
PERTUKARAN 4:
2 3 10 J Q K
4
PS C:\Users\Sudarmadji\Documents\C files\tugas alpro\pertemuan#4>
```

- Penjelasan;

Variabel array “deck” dengan tipe data card untuk menyimpan input user berupa baris kartu. Tipe data card disini merupakan tipe data abstrak yang memiliki 2 elemen didalamnya, yaitu elemen “ABC” dengan tipe data string dan elemen “value” dengan tipe data integer.

Setelah user memasukkan baris kartu, program akan menentukan nilai pada setiap kartu. Nilai tersebut sesuai dengan aturan yang diberikan Dimana kartu yang bertuliskan angka, nilainya akan urut seperti normal, kecil ke besar adalah 1-10, sedangkan kartu yang bertuliskan huruf memiliki urutan seperti ini ‘J’ lebih kecil dari ‘Q’ dan ‘Q’ lebih kecil dari ‘K’ (J-Q-K). nilai ini akan dimasukkan ke elemen “value”.

Lalu program akan menjalankan prosedur “sortCard”. Disini program akan mengurutkan baris kartu dari user menggunakan bubble sort (kecil ke besar), program akan mengurutkan baris kartu berdasarkan nilai dari elemen “value” di setiap kartu.

B. Soal 2

- Source Code:

```

1  #include <stdio.h>
2
3  void koboImaginaryChess(int i, int j, int size, int chessBoard[8][8])
4  {
5      if(j-2 >= 0)
6      {
7          if(i+1 < size)
8          {
9              chessBoard[i+1][j-2] = 1;
10             }
11             if(i-1 >= 0)
12             {
13                 chessBoard[i-1][j-2] = 1;
14             }
15         }
16         if(j+2 < size)
17         {
18             if(i+1 < size)
19             {
20                 chessBoard[i+1][j+2] = 1;
21             }
22             if(i-1 >= 0)
23             {
24                 chessBoard[i-1][j+2] = 1;
25             }
26         }
27         if(i-2 >= 0)
28         {
29             if(j+1 < size)
30             {
31                 chessBoard[i-2][j+1] = 1;
32             }
33             if(j-1 >= 0)
34             {
35                 chessBoard[i-2][j-1] = 1;
36             }
37         }
38         if(i+2 < size)
39         {
40             if(j+1 < size)
41             {
42                 chessBoard[i+2][j+1] = 1;
43             }
44             if(j-1 >= 0)
45             {
46                 chessBoard[i+2][j-1] = 1;
47             }
48         }
49     }
50
51     int main()
52     {
53         int COL = 8, ROW = 8;
54         int chessBoard[ROW][COL];
55         for(int i = 0; i < ROW; i++)
56         {
57             for(int j = 0; j < COL; j++)
58             {
59                 chessBoard[i][j] = 0;
60             }
61         }
62         int x, y;
63         printf("Masukkan koordinat x: ");
64         scanf("%d", &x);
65         printf("Masukkan koordinat y: ");
66         scanf("%d", &y);
67         koboImaginaryChess(y, x, ROW, chessBoard);
68         for(int i = 0; i < ROW; i++)
69         {
70             for(int j = 0; j < COL; j++)
71             {
72                 printf("%d ", chessBoard[i][j]);
73             }
74             printf("\n");
75         }
76         return 0;
77     }

```

- Output:

```
Masukkan koordinat x: 4
Masukkan koordinat y: 4
0 0 0 0 0 0 0 0
0 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 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
```

- Penjelasan;

Variabel “chessBoard” merupakan papan catur berupa array int dengan ukuran 8 x 8. Mula-mula program akan menyetel nilai 0 di semua index array. Lalu user akan memasukkan koordinat untuk posisi kuda di papan catur.

Setelah itu program akan menjalankan prosedur “koboImaginaryChess”. Prosedur ini akan menjalankan pengecekan yang dikelompokkan menjadi 4 dengan 2 pengecekan di masing masing kelompok. Yang pertama adalah jika kuda berjalan ke kanan maka gerakan L yang bisa dibuat oleh kuda adalah L mengarah ke atas dan bawah. Yang kedua jika kuda berjalan ke atas maka gerakan L yang bisa dibuat oleh kuda adalah L mengarah ke kanan dan kiri. Untuk pengecekan di arah kiri dan bawah juga melalui pengecekan yang sama. Apabila pengecekan bernilai benar array dengan koordinat (x = kolom, y = baris) yang akan dituju oleh kuda akan bernilai 1, jika sebaliknya maka lanjut ke pengecekan selanjutnya.

Selesai dari pengecekan, program akan menampilkan papan catur dengan posisi dimana saja yang dapat dituju oleh kuda yang disimbolkan dengan angka 1.