

LAPORAN PRAKTIKUM

Nama : Rafif Muhammad

NIM : 1203230018

Kelas : IF 03-03

A. 1. Source Code

```
1 #include <stdio.h>
2 #include <stdbool.h>
3
4 #define MAX_KARTU 50
5
6 void tukar(int *a, int *b) {
7     int temp = *a;
8     *a = *b;
9     *b = temp;
10 }
11
12 int bubbleSort(int kartu[], int n) {
13     int steps = 0;
14     bool ditukar;
15     for (int i = 0; i < n - 1; i++) {
16         ditukar = false;
17         for (int j = 0; j < n - i - 1; j++) {
18             if (kartu[j] > kartu[j + 1]) {
19                 tukar(&kartu[j], &kartu[j + 1]);
20                 steps++;
21                 printf("pertukaran ke %d: ", steps);
22                 for (int k = 0; k < n; k++) {
23                     printf("%d ", kartu[k]);
24                 }
25                 printf("\n");
26                 ditukar = true;
27             }
28         }
29         if (!ditukar) {
30             break;
31         }
32     }
33     return steps;
34 }
35
36 int main() {
37     int n;
38     int kartu[MAX_KARTU];
39
40     printf("Input jumlah kartu: ");
41     scanf("%d", &n);
42
43     if (n > MAX_KARTU) {
44         printf("Jumlah kartu melebihi batas maksimum.\n");
45         return 1;
46     }
47
48     printf("Input nilai kartu sebanyak %d (pisahkan dengan spasi): ", n);
49     for (int i = 0; i < n; i++) {
50         char card[3];
51         scanf("%s", card);
52         if (card[0] >= '2' && card[0] <= '9') {
53             kartu[i] = card[0] - '0';
54         } else {
55             switch (card[0]) {
56                 case 'J':
57                     kartu[i] = 11;
58                     break;
59                 case 'Q':
60                     kartu[i] = 12;
61                     break;
62                 case 'K':
63                     kartu[i] = 13;
64                     break;
65                 case '1':
66                     if (card[1] == '0') {
67                         kartu[i] = 10;
68                     } else {
69                         printf("Nilai kartu tidak valid.\n");
70                         return 1;
71                     }
72                     break;
73                 default:
74                     printf("Nilai kartu tidak valid.\n");
75                     return 1;
76             }
77         }
78     }
79
80     int steps = bubbleSort(kartu, n);
81     printf("Urutan kartu setelah diurutkan: ");
82     for (int i = 0; i < n; i++) {
83         switch (kartu[i]) {
84             case 10:
85                 printf("10 ");
86                 break;
87             case 11:
88                 printf("J ");
89                 break;
90             case 12:
91                 printf("Q ");
92                 break;
93             case 13:
94                 printf("K ");
95                 break;
96             default:
97                 printf("%d ", kartu[i]);
98                 break;
99         }
100     }
101     printf("\n%d\n", steps);
102
103     return 0;
104 }
```

2. Penjelasan

#include <stdio.h>

untuk fungsi input-output standar

#include <stdbool.h>

untuk mendapatkan tipe data bool

#define MAX_KARTU 50

Mendefinisikan maksimal kartu sebanyak 50

void tukar(int *a, int *b) {

int temp = *a;

***a = *b;**

***b = temp;**

Fungsi prosedur dengan nama tukar yang berfungsi untuk menukar dua variabel yang diberikan

int bubbleSort(int kartu[], int n) {

```

int steps = 0;
bool ditukar;
for (int i = 0; i < n - 1; i++) {
    ditukar = false;
    for (int j = 0; j < n - i - 1; j++) {
        if (kartu[j] > kartu[j + 1]) {
            tukar(&kartu[j], &kartu[j + 1]);
            steps++;
            printf("pertukaran ke %d: ", steps);
            for (int k = 0; k < n; k++) {
                printf("%d ", kartu[k]);
            }
            printf("\n");
            ditukar = true;
        }
    }
    if (!ditukar) {
        break;
    }
    return steps;
}

```

function untuk mengurutkan data dengan cara bubble sort yang dimana untuk mengurutkan data kartu dengan cara mengecek satu satu setiap angka dan menukar apabila angka setelahnya lebih besar dari angka sebelumnya, perulangan tersebut akan berlangsung sampai semua data selesai. Kemudian jumlah pertukaran akan dijumlahkan dan ditampilkan di layar.

```

int main() {
    fungsi utama dalam suatu program
    int n;
    variable n untuk menyimpan jumlah kartu
    int kartu[MAX_KARTU];
    variabel kartu untuk menyimpan nilai dari kartu yang dimasukkan
    printf("Input jumlah kartu: ");
    scanf("%d", &n);
    input jumlah kartu
    if (n > MAX_KARTU) {
        printf("Jumlah kartu melebihi batas maksimum.\n");
        return 1;
    }
    mengecek apakah jumlah kartu melebihi dari jumlah maksimal kartu, jika iya maka akan menampilkan "jumlah kartu melebihi batas maksimum"
    printf("Input nilai kartu sebanyak %d (pisahkan dengan spasi): ", n);
    for (int i = 0; i < n; i++) {
        jika kartu tidak melebihi maka akan menampilkan "Input nilai kartu sebanyak %d (pisahkan dengan spasi): " maka user akan menginput nilai dari setiap kartu
        char card[3];
        scanf("%s", card);
    }
}

```

variabel array dengan nama card, dan fungsi scanf untuk membaca input dari user

```
if (card[0] >= '2' && card[0] <= '9') {
```

```
kartu[i] = card[0] - '0';
```

apabila kartu adalah angka antara 2 dan 9 maka angka tersebut akan langsung diambil dari karakter pertama 'card' dan disimpan dalam array 'kartu'

```
} else {
```

```
switch (card[0]) {
```

```
case 'J':
```

```
kartu[i] = 11;
```

```
break;
```

```
case 'Q':
```

```
kartu[i] = 12;
```

```
break;
```

```
case 'K':
```

```
kartu[i] = 13;
```

```
break;
```

```
case '1':
```

```
if (card[1] == '0') {
```

```
kartu[i] = 10;
```

```
} else {
```

```
printf("Nilai kartu tidak valid.\n");
```

```
return 1;
```

```
}
```

```
break;
```

```
default:
```

```
printf("Nilai kartu tidak valid.\n");
```

```
return 1;
```

```
}
```

Jika kartu adalah J, Q, K maka nilai tersebut akan bernilai sebagai 11, 12, dan 13, Jika kartu adalah 10, maka program memeriksa karakter kedua untuk memastikan bahwa itu benar-benar 10. Jika ya, maka nilai kartu diset sebagai 10. Jika tidak, program mencetak pesan kesalahan dan keluar dengan kode status 1. Jika kartu tidak sesuai dengan data di atas maka akan menampilkan "nilai kartu tidak valid" dan program akan keluar

```
int steps = bubbleSort(kartu, n);
```

fungsi untuk mengurutkan kartu yang dimasukkan player dengan fungsi bubblesort

```
printf("Urutan kartu setelah diurutkan: ");
```

Memberitahu user bahwa program akan menampilkan urutan kartu yang telah diurutkan

```
for (int i = 0; i < n; i++) {
```

memulai loop untuk mencetak setiap kartu dalam urutan yang telah diurutkan

```
switch (kartu[i]) {
```

```
case 10:
```

```
printf("10 ");
```

```
break;
```

```
case 11:
```

```
printf("J ");
```

```

break;
case 12:
printf("Q ");
break;
case 13:
printf("K ");
break;
default:
printf("%d ", kartu[i]); break; }

```

Fungsi untuk mencetak setiap kartu dalam urutan yang telah diurutkan. Kartu dengan nilai 10 akan dicetak sebagai angka 10, J dicetak sebagai "J", Q dicetak sebagai "Q", K dicetak sebagai "K", dan angka lainnya dicetak sesuai dengan nilainya.

```
printf("\n%d\n", steps);
```

mencetak jumlah pengurutan kartu

```
return 0; }
```

mengembalikan nilai 0, menandakan bahwa program selesai

3. Output

```

PS C:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2> cd "c:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2\" ;
if ($?) { gcc praktikum3a.c -o praktikum3a } ; if ($?) { .\praktikum3a }
Input jumlah kartu: 6
Input nilai kartu sebanyak 6 (pisahkan dengan spasi): 10 J K Q 3 2
pertukaran ke 1: 10 11 12 13 3 2
pertukaran ke 2: 10 11 12 3 13 2
pertukaran ke 3: 10 11 12 3 2 13
pertukaran ke 4: 10 11 3 12 2 13
pertukaran ke 5: 10 11 3 2 12 13
pertukaran ke 6: 10 3 11 2 12 13
pertukaran ke 7: 10 3 2 11 12 13
pertukaran ke 8: 3 10 2 11 12 13
pertukaran ke 9: 3 2 10 11 12 13
pertukaran ke 10: 2 3 10 11 12 13
Urutan kartu setelah diurutkan: 2 3 10 J Q K
10
PS C:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2>

```

B. 1. Source Code

```
1 #include <stdio.h>
2
3 int posisi(int i, int j, int size) {
4     return (i >= 0 && i < size && j >= 0 && j < size);
5 }
6
7 void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
8     int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};
9
10    for (int k = 0; k < 8; k++) {
11        int a = i + moves[k][0];
12        int b = j + moves[k][1];
13
14        if (posisi(a, b, size)) {
15            chessBoard[a * size + b] = 1;
16        }
17    }
18 }
19
20 int main() {
21     int i, j;
22     int chessBoard[8][8] = {0};
23
24     printf("Masukkan 2 nilai i dan j (pisahkan dengan spasi) : ");
25     scanf("%d %d", &i, &j);
26
27     koboImaginaryChess(i, j, 8, (int *)chessBoard);
28
29     printf("Hasil :\n");
30     for (int baris = 0; baris < 8; baris++) {
31         for (int kolom = 0; kolom < 8; kolom++) {
32             printf("%d ", chessBoard[baris][kolom]);
33         }
34         printf("\n");
35     }
36
37     return 0;
38 }
39
```

2. Penjelasan

#include <stdio.h>

untuk fungsi input-output standar

int posisi(int i, int j, int size) {

Fungsi bernama posisi yang digunakan untuk memeriksa apakah posisi i dan j benar pada papan catur berukuran size x size

return (i >= 0 && i < size && j >= 0 && j < size);

}

Implementasi fungsi posisi yang berfungsi untuk mengembalikan nilai 1 jika posisi i dan j benar di papan catur, jika salah maka akan mengembalikan nilai 0.

void koboImaginaryChess(int i, int j, int size, int *chessBoard) {

fungsi prosedurr bernama koboimaginarychess yang berfungsi untuk menandai posisi yang dapat dicapai oleh kuda dalam sekali langkah

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

inisialisasi array yang bertujuan untuk memeriksa semua kemungkinan arah kuda berjalan dalam satu langkah

for (int k = 0; k < 8; k++) {

fungsi loop yang digunakan untuk mengiterasi semua kemungkinan pergerakan kuda dalam array move

int a = i + moves[k][0];

untuk menentukan nilai baris baru setelah kuda bergerak

int b = j + moves[k][1];

Untuk menentukan nilai kolom baru setelah kuda bergerak

```

    if (posisi(a, b, size)) {
memeriksa apakah posisi baru yang dihitung masih berada dalam papan catur
        chessBoard[a * size + b] = 1;
ini adalah fungsi dimana untuk menandai posisi baru sebagai posisi yang dapat dicapai
oleh kuda
    }
}
}
}

```

```

int main() {
fungsi utama dalam program
    int i, j;
variabel i dan j yang berfungsi untuk menyimpan posisi awal kuda
    int chessBoard[8][8] = {0};
variabel array dua dimensi yang bernilai 0 disetiap indeks.

```

```

    printf("Masukkan 2 nilai i dan j (pisahkan dengan spasi) : ");
    scanf("%d %d", &i, &j);
input nilai i dan j
    koboImaginaryChess(i, j, 8, (int *)chessBoard);
memanggil fungsi koboimaginarychess untuk menandai posisi yang dapat dicapai oleh
kuda dari posisi yang dimasukkan oleh user
    printf("Hasil:\n");
mencetak "hasil"
    for (int baris = 0; baris < 8; baris++) {
loop for yang digunakan untuk mengiterasi baris pada array chessBoard
        for (int kolom = 0; kolom < 8; kolom++) {
loop for bersarang yang digunakan untuk mengiterasi kolom pada array chessBoard.
            printf("%d ", chessBoard[baris][kolom]);
berfungsi untuk mencetak nilai setiap elemen array
        }
    }
    return 0;
program mengembalikan nilai 0 menandakan bahwa program telah selesai dijalankan

```

3. Output

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2> cd "c:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2\" ;
if ($?) { gcc praktikum3b.c -o praktikum3b } ; if ($?) { .\praktikum3b }
Masukkan 2 nilai i dan j (pisahkan dengan spasi) : 2 2
Hasil :
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 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
PS C:\Users\rafif\Documents\Templat Office Kustom\.vscode\alpro semester 2\praktikum semester 2>

```