

Nama: Jonathan Ferdinand Mayon

Kelas: IF 03 03

NIM: 1203230087

1. SS Source Code Nomer 1

```
1 #include <stdio.h>
2 #include <stdbool.h>
3
4 // Fungsi untuk menukar nilai dua kartu
5 void tukarKartu(char *kartu1, char *kartu2) {
6     char temp = *kartu1;
7     *kartu1 = *kartu2;
8     *kartu2 = temp;
9 }
10
11 // Fungsi untuk menghitung jumlah minimal langkah pertukaran
12 void hitungLangkahPertukaran(int N, char *kartu) {
13     int langkah = 0;
14     int i, j;
15
16     // Algoritma sorting bubble
17     for (i = 0; i < N-1; i++) {
18         for (j = 0; j < N-i-1; j++) {
19             // Mengonversi karakter J, Q, K menjadi angka
20             int angka1 = (kartu[j] >= '2' && kartu[j] <= '9') ? (kartu[j] - '0') :
21                 (kartu[j] == 'J' ? 11 : (kartu[j] == 'Q' ? 12 : 13));
22             int angka2 = (kartu[j+1] >= '2' && kartu[j+1] <= '9') ? (kartu[j+1] - '0') :
23                 (kartu[j+1] == 'J' ? 11 : (kartu[j+1] == 'Q' ? 12 : 13));
24
25             // Menukar posisi jika kartu kedua lebih kecil dari kartu pertama
26             if (angka1 > angka2) {
27                 tukarKartu(&kartu[j], &kartu[j+1]);
28                 langkah++;
29
30                 // Mencetak urutan kartu setiap terjadi pertukaran
31                 printf("Langkah %d: ", langkah);
32                 for (int k = 0; k < N; k++) {
33                     printf("%c ", kartu[k]);
34                 }
35                 printf("\n");
36             }
37         }
38     }
39
40     printf("Jumlah minimal langkah pertukaran: %d\n", langkah);
41 }
42
43 int main() {
44     int N;
45     printf("Masukkan jumlah kartu: ");
46     scanf("%d", &N);
47
48     char kartu[N];
49     printf("Masukkan nilai kartu (pisahkan dengan spasi): ");
50     for (int i = 0; i < N; i++) {
51         scanf(" %c", &kartu[i]);
52     }
53
54     // Menghitung jumlah minimal Langkah pertukaran
55     hitungLangkahPertukaran(N, kartu);
56
57     return 0;
58 }
59
```

2. Penjelasan Program Nomer 1

```
#include <stdio.h>
#include <stdbool.h>
```

Ini adalah header file standar yang digunakan dalam bahasa C. <stdio.h> digunakan untuk fungsi input-output standar seperti printf dan scanf, sedangkan <stdbool.h> digunakan untuk tipe data boolean dan nilai-nilai konstan true dan false.

```
// Fungsi untuk menukar nilai dua kartu
void tukarKartu(char *kartu1, char *kartu2) {
    char temp = *kartu1;
    *kartu1 = *kartu2;
    *kartu2 = temp;
}
```

Ini adalah definisi fungsi tukarKartu yang bertugas untuk menukar nilai dua kartu. Fungsi ini menerima dua parameter, yaitu pointer ke dua kartu yang akan ditukar. Di dalam fungsi, nilai dari kedua kartu ditukar menggunakan variabel sementara temp.

```
// Fungsi untuk menghitung jumlah minimal langkah pertukaran
void hitungLangkahPertukaran(int N, char *kartu) {
    int langkah = 0;
    int i, j;

    // Algoritma sorting bubble
    for (i = 0; i < N-1; i++) {
        for (j = 0; j < N-i-1; j++) {
            // Mengonversi karakter J, Q, K menjadi angka
            int angka1 = (kartu[j] >= '2' && kartu[j] <= '9') ? (kartu[j] - '0') :
                (kartu[j] == 'J' ? 11 : (kartu[j] == 'Q' ? 12 : 13));
            int angka2 = (kartu[j+1] >= '2' && kartu[j+1] <= '9') ? (kartu[j+1] - '0') :
                (kartu[j+1] == 'J' ? 11 : (kartu[j+1] == 'Q' ? 12 : 13));

            // Menukar posisi jika kartu kedua lebih kecil dari kartu pertama
            if (angka1 > angka2) {
                tukarKartu(&kartu[j], &kartu[j+1]);
                langkah++;

                // Mencetak urutan kartu setiap terjadi pertukaran
                printf("Langkah %d: ", langkah);
                for (int k = 0; k < N; k++) {
                    printf("%c ", kartu[k]);
                }
                printf("\n");
            }
        }
    }

    printf("Jumlah minimal langkah pertukaran: %d\n", langkah);
}
```

Ini adalah definisi fungsi `hitungLangkahPertukaran` yang bertugas untuk menghitung jumlah minimal langkah pertukaran untuk mengurutkan kartu. Fungsi ini menerima dua parameter, yaitu jumlah kartu (`N`) dan array kartu (`kartu`). Fungsi ini menggunakan algoritma sorting bubble untuk mengurutkan kartu. Dalam proses pengurutan, karakter 'J', 'Q', dan 'K' dikonversi menjadi angka agar dapat dibandingkan secara numerik. Setiap terjadi pertukaran, urutan kartu dicetak ke layar. Pada akhirnya, fungsi ini mencetak jumlah minimal langkah pertukaran ke layar.

```
int main() {
    int N;
    printf("Masukkan jumlah kartu: ");
    scanf("%d", &N);

    char kartu[N];
    printf("Masukkan nilai kartu (pisahkan dengan spasi): ");
    for (int i = 0; i < N; i++) {
        scanf(" %c", &kartu[i]);
    }

    // Menghitung jumlah minimal langkah pertukaran
    hitungLangkahPertukaran(N, kartu);

    return 0;
}
```

Ini adalah fungsi `main` yang merupakan entry point dari program. Pada baris 54-55, pengguna diminta untuk memasukkan jumlah kartu. Pada baris 58-63, pengguna diminta untuk memasukkan nilai kartu satu per satu. Pada baris 66, fungsi `hitungLangkahPertukaran` dipanggil untuk menghitung jumlah minimal langkah pertukaran untuk mengurutkan kartu. Pada akhirnya, program mengembalikan nilai 0 sebagai tanda bahwa program telah berakhir dengan sukses.

Semoga penjelasan ini membantu Anda memahami program tersebut! Jika ada pertanyaan lebih lanjut, jangan ragu untuk bertanya.

3. Output Nomer 1

```
PS C:\project> cd "c:\project\Kuliah Tel-U\kuliah sem 2\ASD PRAKTIKUM" ; if ($?) { gcc prakasd-oth-1.c -o prakasd-oth-1 } ; if ($?) { .\pra
Masukkan jumlah kartu: 7
Masukkan nilai kartu (pisahkan dengan spasi): 4 6 K Q J 2 8
Langkah 1: 4 6 Q K J 2 8
Langkah 2: 4 6 Q J K 2 8
Langkah 3: 4 6 Q J 2 K 8
Langkah 4: 4 6 Q J 2 8 K
Langkah 5: 4 6 J Q 2 8 K
Langkah 6: 4 6 J 2 Q 8 K
Langkah 7: 4 6 J 2 8 Q K
Langkah 8: 4 6 2 J 8 Q K
Langkah 9: 4 6 2 8 J Q K
Langkah 10: 4 2 6 8 J Q K
Langkah 11: 2 4 6 8 J Q K
Jumlah minimal langkah pertukaran: 11
PS C:\project\Kuliah Tel-U\kuliah sem 2\ASD PRAKTIKUM>
```

1. SS Source Code Nomer 2

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  // Fungsi untuk menampilkan papan catur
5  void displayChessBoard(int *chessBoard) {
6      printf("Chess Board:\n");
7      for (int i = 0; i < 8; i++) {
8          for (int j = 0; j < 8; j++) {
9              printf("%d ", *(chessBoard + i * 8 + j));
10         }
11         printf("\n");
12     }
13 }
14
15 // Fungsi untuk menentukan posisi yang dapat dicapai oleh bidak kuda
16 void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
17     int moves[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1}, {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};
18
19     // Mengatur ulang papan catur dengan nilai 0
20     for (int k = 0; k < size * size; k++) {
21         *(chessBoard + k) = 0;
22     }
23
24     // Menentukan posisi yang dapat dicapai oleh bidak kuda
25     for (int k = 0; k < 8; k++) {
26         int new_i = i + moves[k][0];
27         int new_j = j + moves[k][1];
28         if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
29             *(chessBoard + new_i * 8 + new_j) = 1;
30         }
31     }
32 }
33
34 int main() {
35     int i, j;
36     printf("Masukkan nilai i dan j : ");
37     scanf("%d %d", &i, &j);
38
39     int *chessBoard = (int *)malloc(8 * 8 * sizeof(int));
40
41     // Menjalankan fungsi untuk menentukan posisi yang dapat dicapai oleh bidak kuda
42     koboImaginaryChess(i, j, 8, chessBoard);
43
44     // Menampilkan papan catur
45     displayChessBoard(chessBoard);
46
47     free(chessBoard);
48     return 0;
49 }
50
```

2. Penjelasan Program Nomer 2

```
#include <stdio.h>
#include <stdlib.h>
```

Ini adalah header file standar yang digunakan dalam bahasa C. <stdio.h> digunakan untuk fungsi input-output standar seperti printf dan scanf, sedangkan <stdlib.h> digunakan untuk alokasi memori dinamis menggunakan fungsi malloc.

```
void displayChessBoard(int *chessBoard) {
    printf("Chess Board:\n");
    for (int i = 0; i < 8; i++) {
        for (int j = 0; j < 8; j++) {
            printf("%d ", *(chessBoard + i * 8 + j));
        }
        printf("\n");
    }
}
```

Ini adalah definisi fungsi displayChessBoard yang bertugas untuk menampilkan papan catur ke layar. Fungsi ini menerima parameter chessBoard, yaitu array 2 dimensi yang merepresentasikan papan catur. Fungsi ini menggunakan nested loop untuk mencetak setiap elemen dari chessBoard ke layar.

```
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}};

    // Mengatur ulang papan catur dengan nilai 0
    for (int k = 0; k < size * size; k++) {
        *(chessBoard + k) = 0;
    }

    // Menentukan posisi yang dapat dicapai oleh bidak kuda
    for (int k = 0; k < 8; k++) {
        int new_i = i + moves[k][0];
        int new_j = j + moves[k][1];
        if (new_i >= 0 && new_i < size && new_j >= 0 && new_j < size) {
            *(chessBoard + new_i * 8 + new_j) = 1;
        }
    }
}
```

Ini adalah definisi fungsi koboImaginaryChess yang bertugas untuk menentukan posisi yang dapat dicapai oleh bidak kuda dalam sekali jalan. Fungsi ini menerima parameter i dan j, yaitu posisi awal bidak kuda, size, yaitu ukuran papan catur, dan chessBoard, yaitu array 2 dimensi yang merepresentasikan papan catur. Fungsi ini menggunakan array moves untuk menyimpan langkah-langkah yang mungkin dilakukan oleh bidak kuda. Kemudian, fungsi ini menggunakan nested loop untuk menentukan posisi yang dapat dicapai oleh bidak kuda berdasarkan posisi awalnya.

```

int main() {
    int i, j;
    printf("Masukkan nilai i dan j : ");
    scanf("%d %d", &i, &j);

    int *chessBoard = (int *)malloc(8 * 8 * sizeof(int));

    // Menjalankan fungsi untuk menentukan posisi yang dapat dicapai oleh bidak kuda
    koboImaginaryChess(i, j, 8, chessBoard);

    // Menampilkan papan catur
    displayChessBoard(chessBoard);

    free(chessBoard);
    return 0;
}

```

Ini adalah fungsi main yang merupakan entry point dari program. Pada baris 38-39, pengguna diminta untuk memasukkan nilai i dan j yang merupakan posisi awal bidak kuda. Pada baris 41, memori dialokasikan untuk array chessBoard menggunakan fungsi malloc. Pada baris 44, fungsi koboImaginaryChess dipanggil untuk menentukan posisi yang dapat dicapai oleh bidak kuda. Pada baris 47, papan catur ditampilkan ke layar menggunakan fungsi displayChessBoard. Pada akhirnya, memori yang dialokasikan untuk chessBoard dibebaskan kembali menggunakan fungsi free.

3. Output Nomer 2

```

Masukkan nilai i dan j : 2 2
Chess Board:
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:\project\Kuliah Tel-U\kuliah sem 2\ASD PRAKTIKUM> cd "c:\project\Kuliah Tel-U\kuliah sem 2\ASD PRAK
Masukkan nilai i dan j : 3 7
Chess Board:
0 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 0 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 0
PS C:\project\Kuliah Tel-U\kuliah sem 2\ASD PRAKTIKUM>

```