

NAMA : GALANG UBAIDILLAH

NIM : 1203230056

KELAS : IF 03-03

Tugas Algoritma Struktur Data - Array, Pointer, dan Fungsi

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 // Function to compare card values
6 int compare(const void *a, const void *b) {
7     char cardA = *(char *)a;
8     char cardB = *(char *)b;
9     char order[] = "123456789JQK";
10
11     int indexA = strchr(order, cardA) - order;
12     int indexB = strchr(order, cardB) - order;
13
14     return indexA - indexB;
15 }
16
17 // Function to perform selection sort and display exchanges
18 int sortCards(int n, char cards[]) {
19     int steps = 0;
20     for (int i = 0; i < n - 1; i++) {
21         int minIndex = i;
22         for (int j = i + 1; j < n; j++) {
23             if (compare(&cards[j], &cards[minIndex]) < 0) {
24                 minIndex = j;
25             }
26         }
27         if (minIndex != i) {
28             char temp = cards[i];
29             cards[i] = cards[minIndex];
30             cards[minIndex] = temp;
31             steps++;
32             // Display the exchange
33             printf("Pertukaran %d: ", steps);
34             for (int k = 0; k < n; k++) {
35                 printf("%c ", cards[k]);
36             }
37             printf("\n");
38         }
39     }
40     return steps;
41 }
42
43 int main() {
44     int n;
45     printf("Masukkan jumlah kartu: ");
46     scanf("%d", &n);
47
48     char cards[n];
49     printf("Masukkan nilai kartu (1-10, J, Q, K dipisahkan dengan spasi): ");
50     for (int i = 0; i < n; i++) {
51         scanf(" %c", &cards[i]);
52     }
53
54     int steps = sortCards(n, cards);
55
56     printf("\nJumlah langkah untuk menyortir kartu: %d\n", steps);
57     printf("Urutan kartu setelah disortir: ");
58     for (int i = 0; i < n; i++) {
59         printf("%c ", cards[i]);
60     }
61     printf("\n");
62
63     return 0;
64 }
```

Gambar 1 code poker

NAMA : GALANG UBAIDILLAH

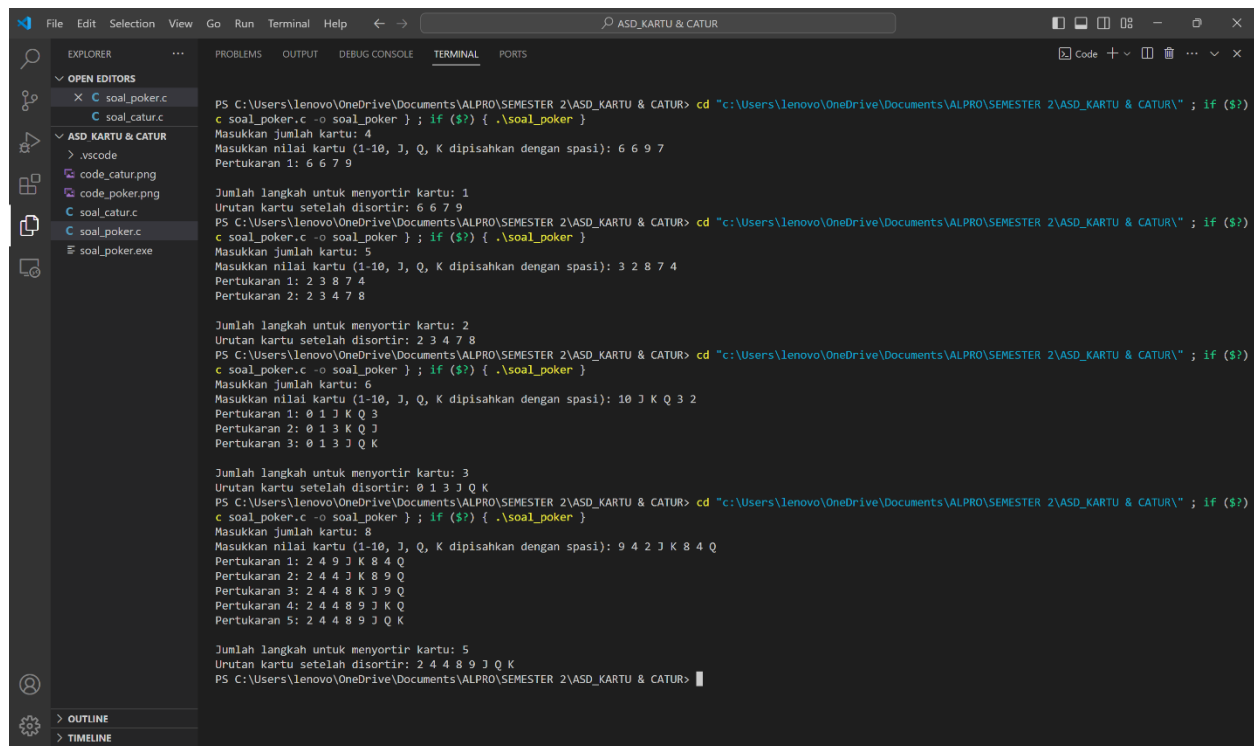
NIM : 1203230056

KELAS : IF 03-03

Penjelasan Kode :

1. *compare(const void *a, const void *b):* Fungsi ini digunakan sebagai pembanding dalam pengurutan kartu. Fungsi ini mengambil dua pointer void (a dan b), kemudian mengonversinya menjadi karakter dan membandingkan urutan kartu berdasarkan nilai numeriknya yang telah ditentukan sebelumnya dalam string order. Fungsi strchr digunakan untuk mencari indeks karakter dalam string order.
2. *sortCards(int n, char cards[]):* Fungsi ini melakukan pengurutan kartu menggunakan algoritma selection sort. Pertama, fungsi ini mengiterasi melalui setiap kartu dalam array cards, mencari kartu terkecil dalam sisa kartu yang belum diurutkan, dan menukar posisi kartu tersebut dengan kartu pada indeks pertama dari sisa kartu yang belum diurutkan. Fungsi juga mencetak setiap pertukaran kartu dan menghitung jumlah langkah yang diperlukan untuk menyortir kartu.
3. *main():* Fungsi utama dari program ini. Pertama, program meminta pengguna untuk memasukkan jumlah kartu yang akan diurutkan. Kemudian, program menerima input nilai kartu dari pengguna dan menyimpannya dalam array cards. Selanjutnya, program memanggil fungsi sortCards untuk melakukan pengurutan kartu dan mencetak jumlah langkah yang diperlukan beserta urutan kartu setelah disortir.

Output Program Poker :



```
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR> cd "c:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR\" ; if ($?) {
c soal_poker.c -o soal_poker } ; if ($?) { .\soal_poker }
Masukkan jumlah kartu: 4
Masukkan nilai kartu (1-10, J, Q, K dipisahkan dengan spasi): 6 6 9 7
Pertukaran 1: 6 6 9 7

Jumlah langkah untuk menyortir kartu: 1
Urutan kartu setelah disortir: 6 6 7 9
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR> cd "c:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR\" ; if ($?) {
c soal_poker.c -o soal_poker } ; if ($?) { .\soal_poker }
Masukkan jumlah kartu: 5
Masukkan nilai kartu (1-10, J, Q, K dipisahkan dengan spasi): 3 2 8 7 4
Pertukaran 1: 2 3 8 7 4
Pertukaran 2: 2 3 4 7 8

Jumlah langkah untuk menyortir kartu: 2
Urutan kartu setelah disortir: 2 3 4 7 8
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR> cd "c:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR\" ; if ($?) {
c soal_poker.c -o soal_poker } ; if ($?) { .\soal_poker }
Masukkan jumlah kartu: 6
Masukkan nilai kartu (1-10, J, Q, K dipisahkan dengan spasi): 10 J K Q 3 2
Pertukaran 1: 0 1 J K Q 3
Pertukaran 2: 0 1 3 K Q J
Pertukaran 3: 0 1 3 J Q K

Jumlah langkah untuk menyortir kartu: 3
Urutan kartu setelah disortir: 0 1 3 J Q K
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR> cd "c:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR\" ; if ($?) {
c soal_poker.c -o soal_poker } ; if ($?) { .\soal_poker }
Masukkan jumlah kartu: 8
Masukkan nilai kartu (1-10, J, Q, K dipisahkan dengan spasi): 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 langkah untuk menyortir kartu: 5
Urutan kartu setelah disortir: 2 4 4 8 9 J Q K
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR>
```

Gambar 2 output poker

NAMA : GALANG UBAIDILLAH

NIM : 1203230056

KELAS : IF 03-03

NAMA : GALANG UBAIDILLAH

NIM : 1203230056

KELAS : IF 03-03

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 // Fungsi untuk mengecek apakah suatu posisi valid di dalam papan catur //
5 int isValidPosition(int x, int y) {
6     return (x >= 0 && x < 8 && y >= 0 && y < 8);
7 }
8
9 // Fungsi untuk memberi nilai 1 pada posisi yang dapat dicapai oleh bidak kuda //
10 void markPosition(int i, int j, int *chessBoard) {
11     if (isValidPosition(i, j)) {
12         chessBoard[i * 8 + j] = 1;
13     }
14 }
15
16 // Fungsi untuk mensimulasikan pergerakan bidak kuda dalam sekali jalan //
17 void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
18     // Posisi yang mungkin dicapai oleh bidak kuda dalam sekali jalan //
19     int dx[] = {-2, -2, -1, -1, 1, 1, 2, 2};
20     int dy[] = {-1, 1, -2, 2, -2, 2, -1, 1};
21
22     // Menandai setiap posisi yang mungkin dicapai oleh bidak kuda //
23     for (int k = 0; k < 8; k++) {
24         int ni = i + dx[k];
25         int nj = j + dy[k];
26         markPosition(ni, nj, chessBoard);
27     }
28 }
29
30 int main() {
31     // Mendefinisikan array untuk papan catur berukuran 8x8 //
32     int chessBoard[64] = {0};
33
34     // Menerima input posisi bidak kuda
35     int i, j;
36     scanf("%d %d", &i, &j);
37
38     // Mensimulasikan pergerakan bidak kuda //
39     koboImaginaryChess(i, j, 8, chessBoard);
40
41     // Mencetak papan catur setelah simulasi //
42     for (int row = 0; row < 8; row++) {
43         for (int col = 0; col < 8; col++) {
44             printf("%d", chessBoard[row * 8 + col]);
45         }
46         printf("\n");
47     }
48
49     return 0;
50 }
```

Gambar 3 code catur

Penjelasan Kode :

1. *isValidPosition(int x, int y)*: Fungsi ini digunakan untuk memeriksa apakah suatu posisi (x, y) valid di dalam papan catur berukuran 8x8. Posisi valid adalah jika x dan y berada di antara 0 dan 7 (inklusif).

2. *markPosition(int i, int j, int *chessBoard)*: Fungsi ini memberi nilai 1 pada posisi yang dapat dicapai oleh bidak kuda dari posisi (i, j). Jika posisi tersebut valid, maka nilai pada array `chessBoard` pada indeks yang sesuai dengan posisi tersebut akan diubah menjadi 1.

NAMA : GALANG UBAIDILLAH

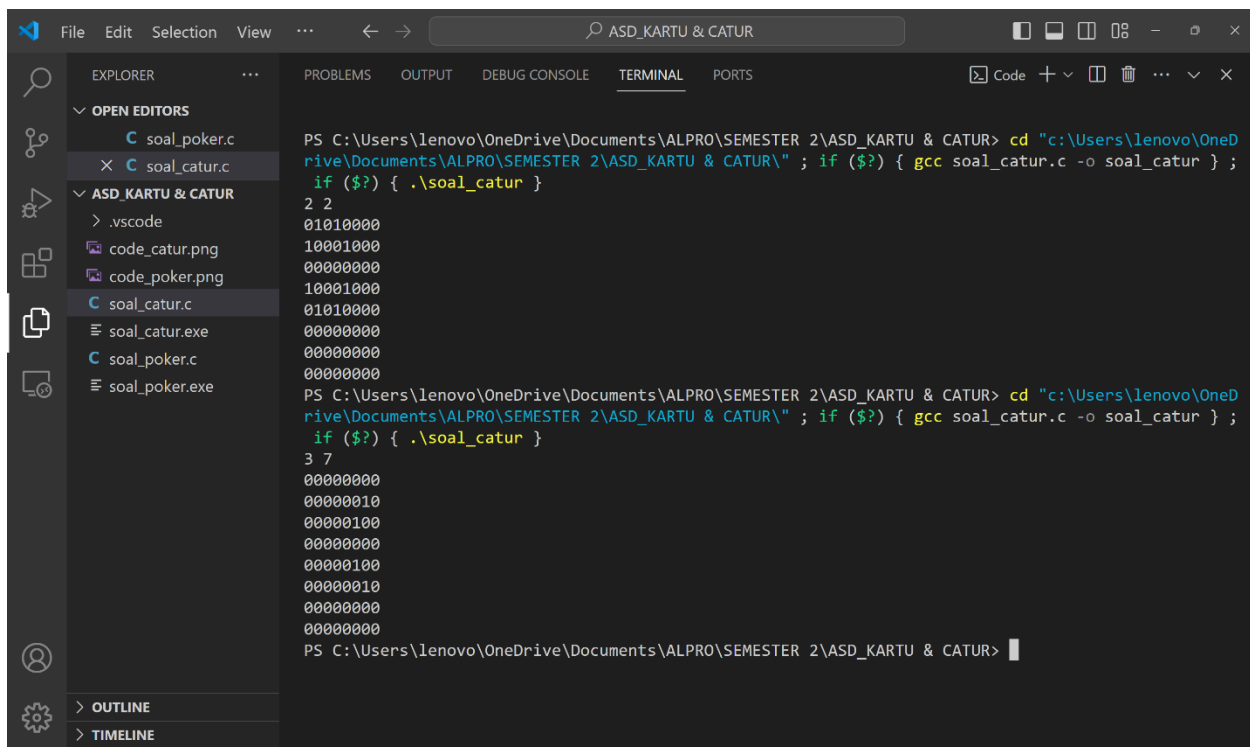
NIM : 1203230056

KELAS : IF 03-03

3. *kobolmaginaryChess(int i, int j, int size, int *chessBoard): Fungsi ini mensimulasikan pergerakan bidak kuda dalam sekali jalan dari posisi (i, j) di papan catur berukuran 8x8. Fungsi ini menggunakan array `dx` dan `dy` untuk menentukan posisi yang mungkin dicapai oleh bidak kuda, kemudian memanggil fungsi `markPosition` untuk menandai setiap posisi yang dapat dicapai.*

4. *main(): Fungsi utama dari program ini. Pertama, program mendefinisikan array `chessBoard` yang merepresentasikan papan catur. Kemudian, program menerima input posisi bidak kuda dari pengguna. Setelah itu, program memanggil fungsi `kobolmaginaryChess` untuk mensimulasikan pergerakan bidak kuda, dan mencetak papan catur setelah simulasi tersebut.*

Output Program Catur :



```
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR> cd "c:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR\" ; if ($?) { gcc soal_catur.c -o soal_catur } ; if ($?) { .\soal_catur }
2 2
01010000
10001000
00000000
10001000
01010000
00000000
00000000
00000000
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR> cd "c:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR\" ; if ($?) { gcc soal_catur.c -o soal_catur } ; if ($?) { .\soal_catur }
3 7
00000000
00000010
00000100
00000000
00000100
00000010
00000000
00000000
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR>
```

Gambar 4 output catur