

NAMA : SURYA ADI ARGA WIDANA

NIM : 1203230101

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

KELAS : IF 03-03

Gambar 2 output poker

NAMA : SURYA ADI ARGA WIDANA

NIM : 1203230101

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 :

- 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).*
- 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.*
- koboImaginaryChess(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.*

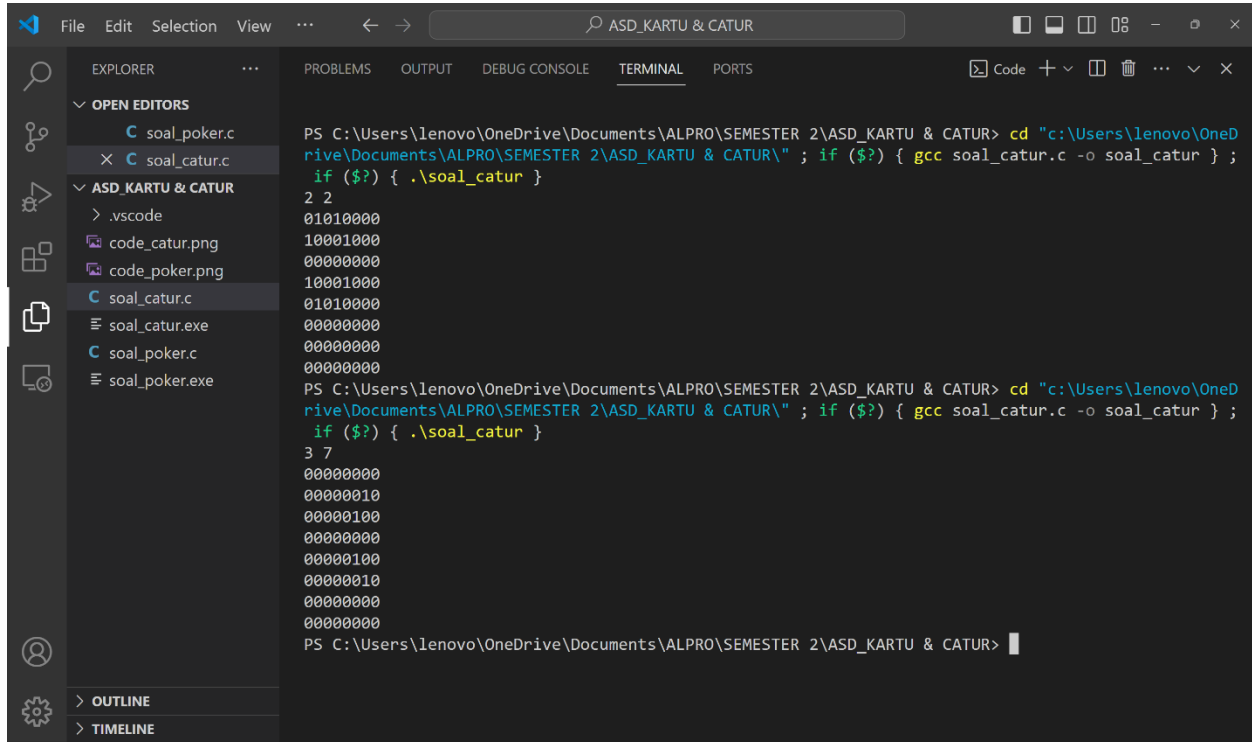
NAMA : SURYA ADI ARG A WIDANA

NIM : 1203230101

KELAS : IF 03-03

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 ``kobilmaginaryChess`` 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
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
00000000
PS C:\Users\lenovo\OneDrive\Documents\ALPRO\SEMESTER 2\ASD_KARTU & CATUR>
```

Gambar 4 output catur