

Nama : Khansa Nailah Anjani

Nim : 1203230038

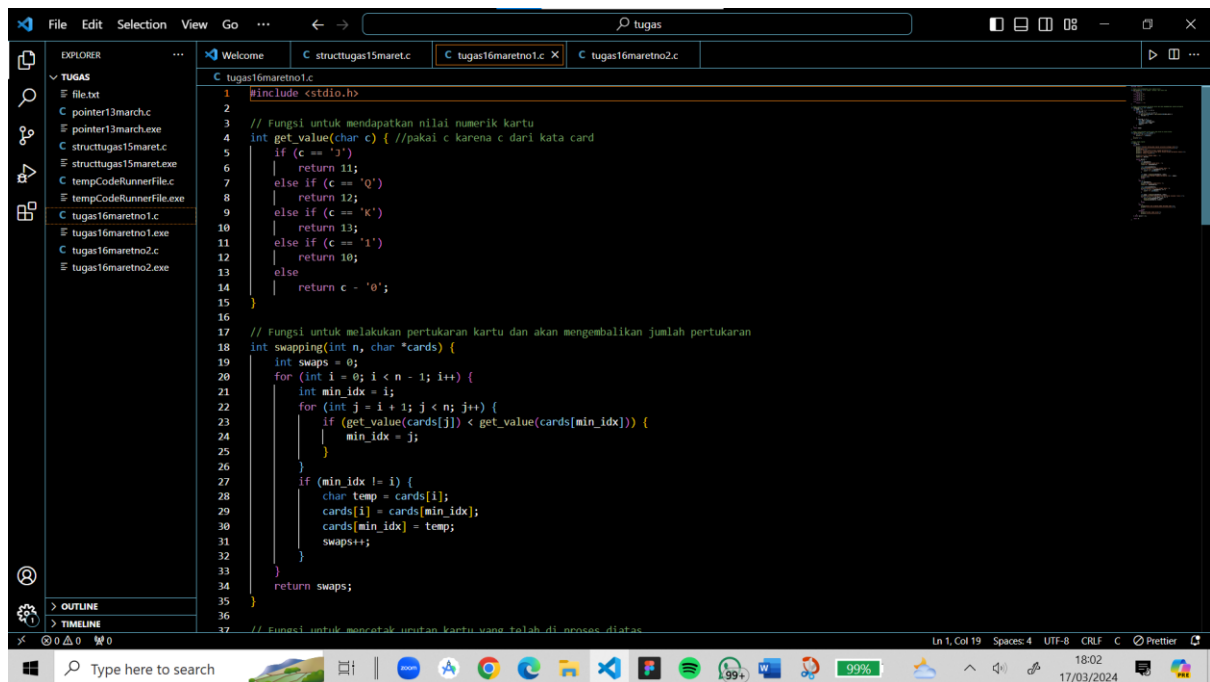
Kelas : IF-03-02

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

NO 1

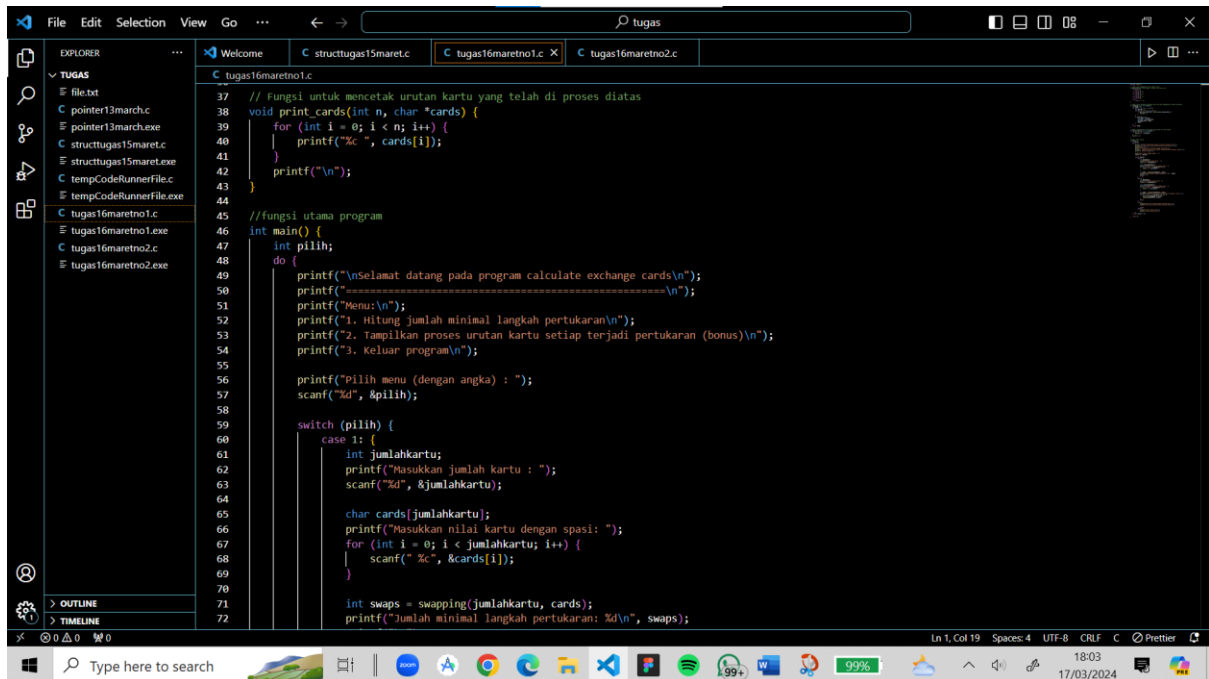
-SS CODE

Baris 1-36



```
1 #include <stdio.h>
2
3 // Fungsi untuk mendapatkan nilai numerik kartu
4 int get_value(char c) { //pakai c karena c dari kata card
5     if (c == 'J')
6         return 11;
7     else if (c == 'Q')
8         return 12;
9     else if (c == 'K')
10        return 13;
11    else if (c == '1')
12        return 10;
13    else
14        return c - '0';
15}
16
17 // Fungsi untuk melakukan pertukaran kartu dan akan mengembalikan jumlah pertukaran
18 int swapping(int n, char *cards) {
19     int swaps = 0;
20     for (int i = 0; i < n - 1; i++) {
21         int min_idx = i;
22         for (int j = i + 1; j < n; j++) {
23             if (get_value(cards[j]) < get_value(cards[min_idx])) {
24                 min_idx = j;
25             }
26         }
27         if (min_idx != i) {
28             char temp = cards[i];
29             cards[i] = cards[min_idx];
30             cards[min_idx] = temp;
31             swaps++;
32         }
33     }
34     return swaps;
35 }
36
37 // Fungsi untuk mencetak urutan kartu yang telah di proses diatas
```

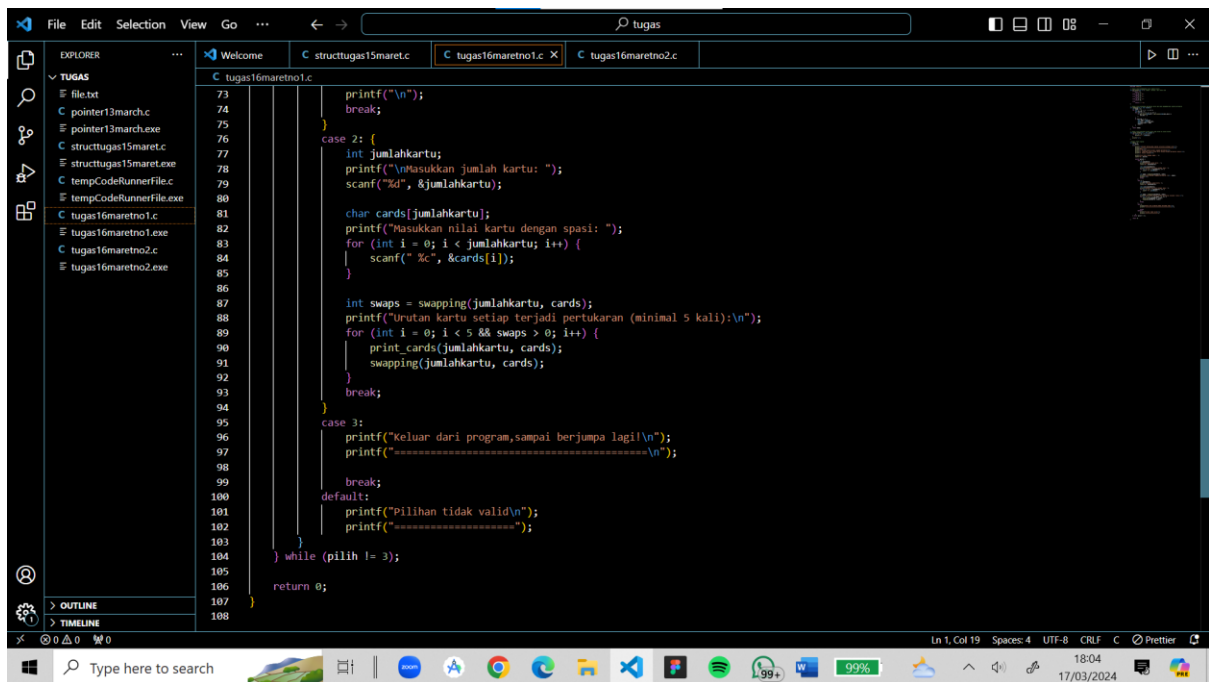
Baris 37-72



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The active file is `tugas16maretno1.c`. The code is in C and implements a card game program. The code is as follows:

```
37 // Fungsi untuk mencetak urutan kartu yang telah di proses diatas
38 void print_cards(int n, char *cards) {
39     for (int i = 0; i < n; i++) {
40         printf("%c ", cards[i]);
41     }
42     printf("\n");
43 }
44
45 //fungsi utama program
46 int main() {
47     int pilih;
48     do {
49         printf("\nSelamat datang pada program calculate exchange cards\n");
50         printf("===== \n");
51         printf("Menu:\n");
52         printf("1. Hitung jumlah minimal langkah pertukaran\n");
53         printf("2. Tampilkan proses urutan kartu setiap terjadi pertukaran (bonus)\n");
54         printf("3. Keluar program\n");
55
56         printf("Pilih menu (dengan angka) : ");
57         scanf("%d", &pilih);
58
59         switch (pilih) {
60             case 1: {
61                 int jumlahkartu;
62                 printf("Masukkan jumlah kartu : ");
63                 scanf("%d", &jumlahkartu);
64
65                 char cards[jumlahkartu];
66                 printf("Masukkan nilai kartu dengan spasi: ");
67                 for (int i = 0; i < jumlahkartu; i++) {
68                     scanf("%c", &cards[i]);
69                 }
70
71                 int swaps = swapping(jumlahkartu, cards);
72                 printf("Jumlah minimal langkah pertukaran: %d\n", swaps);
```

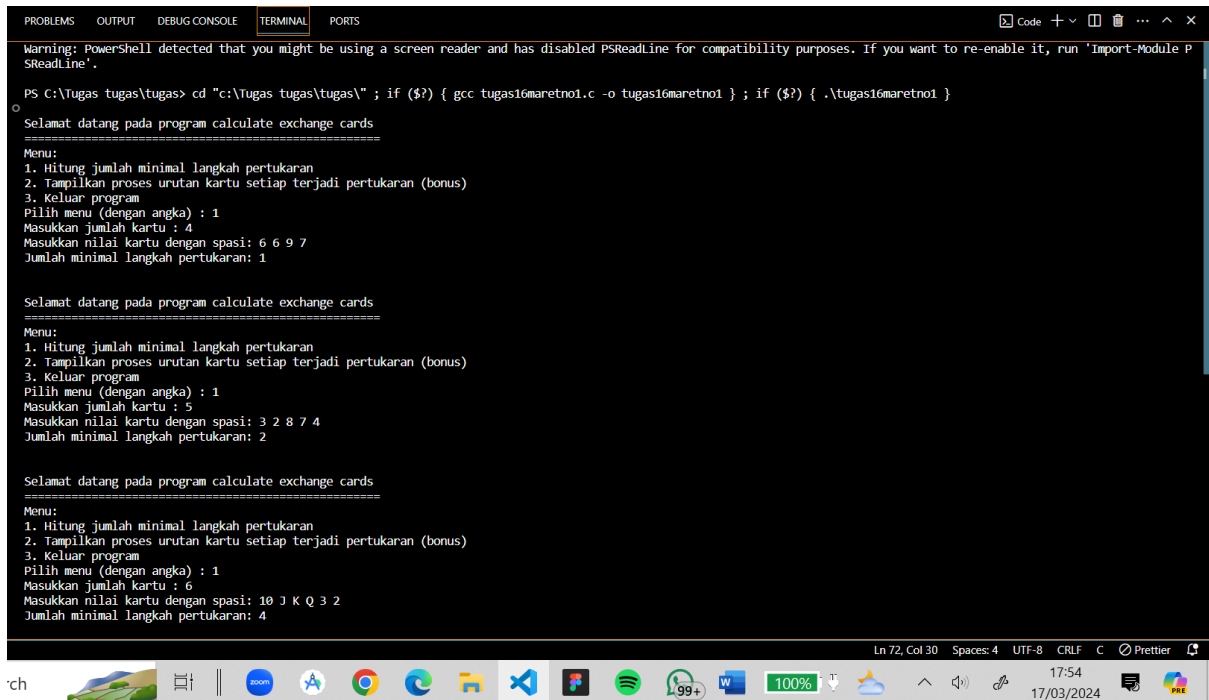
Baris 73-108



The screenshot shows the Visual Studio Code editor with the file explorer on the left. The active file is `tugas16maretno1.c`. The code is in C and implements a card game program. The code is as follows:

```
73         printf("\n");
74         break;
75     }
76     case 2: {
77         int jumlahkartu;
78         printf("Masukkan jumlah kartu : ");
79         scanf("%d", &jumlahkartu);
80
81         char cards[jumlahkartu];
82         printf("Masukkan nilai kartu dengan spasi: ");
83         for (int i = 0; i < jumlahkartu; i++) {
84             scanf("%c", &cards[i]);
85         }
86
87         int swaps = swapping(jumlahkartu, cards);
88         printf("Urutan kartu setiap terjadi pertukaran (minimal 5 kali):\n");
89         for (int i = 0; i < 5 && swaps > 0; i++) {
90             print_cards(jumlahkartu, cards);
91             swapping(jumlahkartu, cards);
92         }
93         break;
94     }
95     case 3: {
96         printf("Keluar dari program,sampai berjumpa lagi!\n");
97         printf("===== \n");
98         break;
99     }
100     default: {
101         printf("Pilihan tidak valid\n");
102         printf("===== ");
103     }
104 } while (pilih != 3);
105
106 return 0;
107
108 }
```

-SS OUTPUT (ada 3)



```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

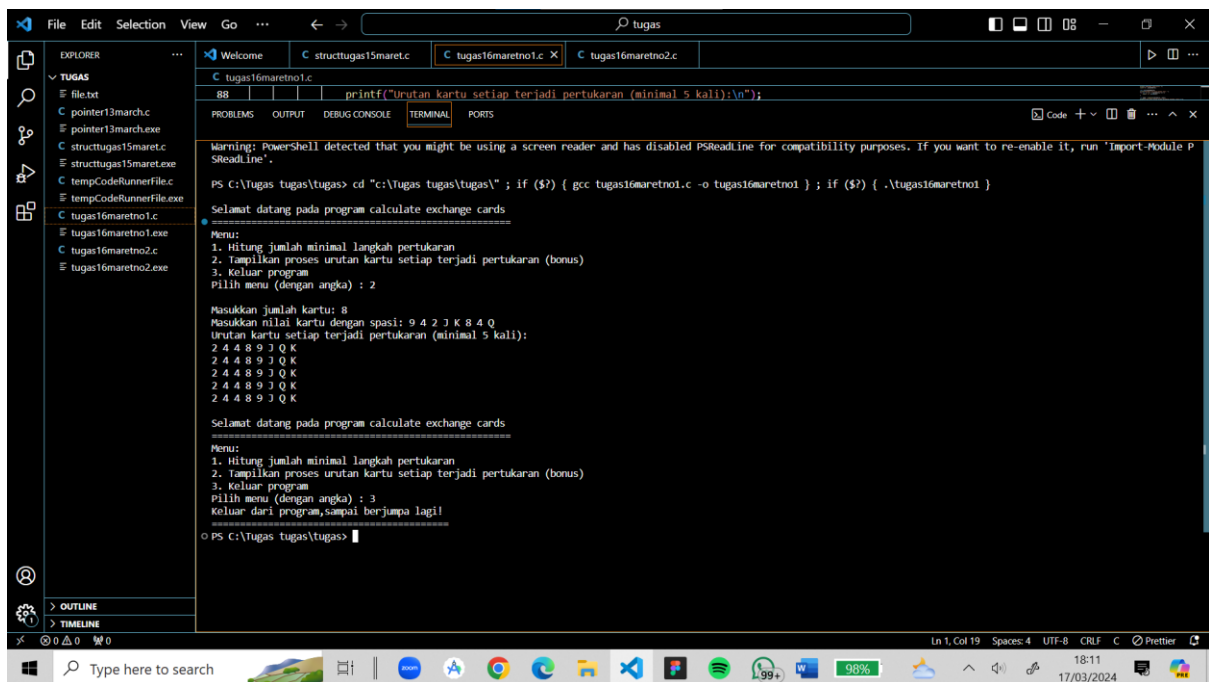
PS C:\Tugas tugas\tugas> cd "c:\Tugas tugas\tugas\"; if ($?) { gcc tugas16maretno1.c -o tugas16maretno1 }; if ($?) { .\tugas16maretno1 }

Selamat datang pada program calculate exchange cards
=====
Menu:
1. Hitung jumlah minimal langkah pertukaran
2. Tampilkan proses urutan kartu setiap terjadi pertukaran (bonus)
3. Keluar program
Pilih menu (dengan angka) : 1
Masukkan jumlah kartu : 4
Masukkan nilai kartu dengan spasi: 6 6 9 7
Jumlah minimal langkah pertukaran: 1

Selamat datang pada program calculate exchange cards
=====
Menu:
1. Hitung jumlah minimal langkah pertukaran
2. Tampilkan proses urutan kartu setiap terjadi pertukaran (bonus)
3. Keluar program
Pilih menu (dengan angka) : 1
Masukkan jumlah kartu : 5
Masukkan nilai kartu dengan spasi: 3 2 8 7 4
Jumlah minimal langkah pertukaran: 2

Selamat datang pada program calculate exchange cards
=====
Menu:
1. Hitung jumlah minimal langkah pertukaran
2. Tampilkan proses urutan kartu setiap terjadi pertukaran (bonus)
3. Keluar program
Pilih menu (dengan angka) : 1
Masukkan jumlah kartu : 6
Masukkan nilai kartu dengan spasi: 10 J K Q 3 2
Jumlah minimal langkah pertukaran: 4
```

-SS OUTPUT (BONUS)



```
File Edit Selection View Go ...
Welcome C:structugas15maretc C:tugas16maretno1.c C:tugas16maretno2.c
C:tugas16maretno1.c
88 printf("urutan kartu setiap terjadi pertukaran (minimal 5 kali):\n");
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module PSReadLine'.

PS C:\Tugas tugas\tugas> cd "c:\Tugas tugas\tugas\"; if ($?) { gcc tugas16maretno1.c -o tugas16maretno1 }; if ($?) { .\tugas16maretno1 }

Selamat datang pada program calculate exchange cards
=====
Menu:
1. Hitung jumlah minimal langkah pertukaran
2. Tampilkan proses urutan kartu setiap terjadi pertukaran (bonus)
3. Keluar program
Pilih menu (dengan angka) : 2
Masukkan jumlah kartu: 8
Masukkan nilai kartu dengan spasi: 9 4 2 J K 8 4 Q
Urutan kartu setiap terjadi pertukaran (minimal 5 kali):
2 4 8 9 J Q K
2 4 8 9 J Q K
2 4 8 9 J Q K
2 4 8 9 J Q K
2 4 8 9 J Q K

Selamat datang pada program calculate exchange cards
=====
Menu:
1. Hitung jumlah minimal langkah pertukaran
2. Tampilkan proses urutan kartu setiap terjadi pertukaran (bonus)
3. Keluar program
Pilih menu (dengan angka) : 3
Keluar dari program, sampai berjumpa lagi!

PS C:\Tugas tugas\tugas>
```

PENJELASAN CODE NOMOR 1

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

Menggunakan preprocessor directive untuk menyertakan library yaitu stdio.h, yang akan kita perlukan untuk menggunakan fungsi input-output standar seperti printf dan scanf.

```

int get_value(char c) { //pakai c karena c dari kata card
    if (c == 'J')
        return 11;
    else if (c == 'Q')
        return 12;
    else if (c == 'K')
        return 13;
    else if (c == '1')
        return 10;
    else
        return c - '0';
}

```

Menggunakan fungsi Bernama `get_value` yang mengambil karakter `c` dan nantinya akan mengembalikan nilai numerik sesuai dengan nilai kartu, kemudian kita menggunakan struktur `if-else` yang digunakan untuk mengecek karakter input dan mengembalikan nilai numerik yang sesuai, contohnya jika karakter numerik J, Q, K atau 1 maka nilai numerik tersebut akan langsung dikembalikan dan jika bukan, maka nilai numerik akan dihitung dengan mengurangi kode ASCII karakter tersebut dengan kode ASCII karakter O.

```

int swapping(int n, char *cards) {
    int swaps = 0;

```

Baris pertama diatas adalah kode yang mendeklarasikan fungsi dari `swapping` Dimana berguna untuk menerima jumlah kartu (`n`) dan terdapat array dengan nama (`cards`) yang berisi karakter kartu kemudian dibaris bawah terdapat code dari inisialisasi sebuah variable (`swaps`) yang akan digunakan untuk menghitung jumlah pertukaran yang dilakukan nantinya.

```

for (int i = 0; i < n - 1; i++) {
    int min_idx = i;
    for (int j = i + 1; j < n; j++) {
        if (get_value(cards[j]) < get_value(cards[min_idx])) {
            min_idx = j;
        }
    }
    if (min_idx != i) {
        char temp = cards[i];
        cards[i] = cards[min_idx];
        cards[min_idx] = temp;
        swaps++;
    }
}
return swaps;
}

```

Code diatas adalah beberapa code dengan fungsi untuk melakukan pertukaran kartu dan akan mengembalikan jumlah pertukaran, disini saya menggunakan looping `for` dan `if`, Dimana `for` pertama untuk melakukan dimulai dari kartu pertama hingga kartu kedua terakhir kemudian kita inisialisasi indeks yang Bernama `min_idx` dengan `i` yang nantinya akan menandakan indeks kartu terkecil pada

iterasi saat ini kemudian looping kedua untuk mencari kartu terkecil yang belum di proses selanjutnya pada if pertama digunakan untuk membandingkan nilai kartu yang belum di proses dengan nilai kartu terkecil yang ditemukan sebelumnya, untuk if kedua jika terdapat kartu terkecil yang tidak berada pada posisi saat ini maka akan dilakukan pertukaran kartu antara kartu saat ini dengan kartu terkecil variable swaps juga di increment karena pertukaran dilakukan. (Simpelnya kita mempunyai daftar kartu yang ingin diurutkan dan daftar2 tersebut kita simpan di dalam sebuah array Bernama cards, kemudian kita akan menggunakan metode seleksi untuk mengurutkan kartu dengan memilih kartu dengan nilai terkecil dan menukar posisinya, kita menggunakan variable temp untuk tempat penampungan sementara saat melakukan proses pengurutan kartu dan pertukaran kartu, kemudian nantinya akan mengembalikan jumlah pertukaran yang dilakukan dalam proses pengurutan.

```
void print_cards(int n, char *cards) {
    for (int i = 0; i < n; i++) {
        printf("%c ", cards[i]);
    }
    printf("\n");
}
```

Disini bisa kita lihat bahwa fungsi Bernama print_cards adalah sebuah fungsi yang digunakan untuk mencetak urutan kartu yang diberikan dalam bentuk array karakter ke layar Dimana fungsi ini memiliki 2 parameter yaitu n (menyatakan jumlah kartu) dan cards (pointer ke array karakter yang berisi urutan kartu yang akan dicetak). untuk Langkah jalannya seperti ini pertama fungsi akan melakukan iterasi melalui setiap elemen dalam array, dimulai dari indeks 0 hingga n-1, kemudian di setiap iterasi nya fungsi printf akan digunakan untuk mencetak karakter cards ke layar dan setelah selesai mencetak seluruh kartu fungsi akan mencetak newline sehingga baris selanjutnya dari output akan dimulai pada baris baru dan membuat tampilan lebih rapi.

```
int main() {
    int pilih;
    do {
        printf("\nSelamat datang pada program calculate exchange cards\n");
        printf("===== \n");
        printf("Menu: \n");
        printf("1. Hitung jumlah minimal langkah pertukaran\n");
        printf("2. Tampilkan proses urutan kartu setiap terjadi pertukaran\n");
        printf("3. Keluar program\n");

        printf("Pilih menu (dengan angka) : ");
        scanf("%d", &pilih);

        switch (pilih) {
            case 1: {
                int jumlahkartu;
                printf("Masukkan jumlah kartu : ");
                scanf("%d", &jumlahkartu);
```

```

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

        int swaps = swapping(jumlahkartu, cards);
        printf("Jumlah minimal langkah pertukaran: %d\n", swaps);
        printf("\n");
        break;
    }
    case 2: {
        int jumlahkartu;
        printf("\nMasukkan jumlah kartu: ");
        scanf("%d", &jumlahkartu);

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

        int swaps = swapping(jumlahkartu, cards);
        printf("Urutan kartu setiap terjadi pertukaran (minimal 5
kali):\n");

        for (int i = 0; i < 5 && swaps > 0; i++) {
            print_cards(jumlahkartu, cards);
            swapping(jumlahkartu, cards);
        }
        break;
    }
    case 3:
        printf("Keluar dari program,sampai berjumpa lagi!\n");
        printf("=====\n");

        break;
    default:
        printf("Pilihan tidak valid\n");
        printf("=====");
    }
} while (pilih != 3);

return 0;
}

```

Disinilah fungsi utama dari program di mainkan,dimulai dari fungsi main() yang didalamnya terdapat sebuah loop do-while yang berarti program akan menjalankan blok kode didalamnya minimal

sekali, kemudian akan terus berputar selama kondisi bagian bawah loop while (pilih != 3) terpenuhi kondisi ini digunakan untuk keluar dari program yang telah dijalankan oleh pengguna. Di dalam loop, terdapat beberapa langkah:

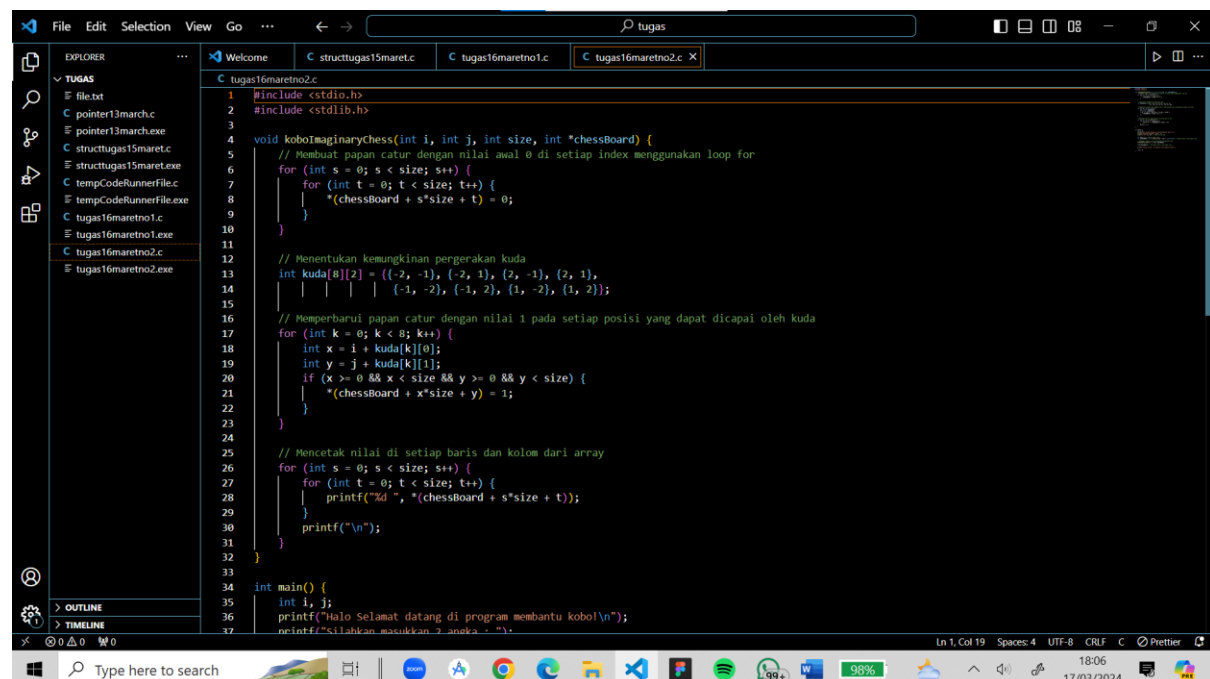
- Program mencetak pesan selamat datang dan daftar menu yang tersedia.
- Pengguna diminta untuk memilih menu dengan memasukkan nomor pilihan.
- Switch-case digunakan untuk menangani berbagai pilihan menu:
 - Jika pengguna memilih menu 1, program meminta pengguna untuk memasukkan jumlah kartu dan nilai kartu, kemudian menghitung jumlah minimal langkah pertukaran yang diperlukan untuk mengurutkan kartu-kartu tersebut.
 - Jika pengguna memilih menu 2, program juga meminta pengguna untuk memasukkan jumlah kartu dan nilai kartu, kemudian selain menghitung jumlah minimal langkah pertukaran, program juga menampilkan urutan kartu setiap kali terjadi pertukaran, dengan batasan minimal 5 kali pertukaran.
 - Jika pengguna memilih menu 3, program mencetak pesan keluar dan keluar dari loop, sehingga mengakhiri program.
 - Jika pengguna memilih menu lain yang tidak valid, program mencetak pesan kesalahan.
 - Setelah menangani pilihan pengguna, program kembali ke awal loop dan menampilkan menu lagi.

Kemudian setelah keluar dari proses loop, program akan mengembalikan nilai 0 dan menandakan bahwa program telah berjalan secara sukses dan program selesai.

NO 2

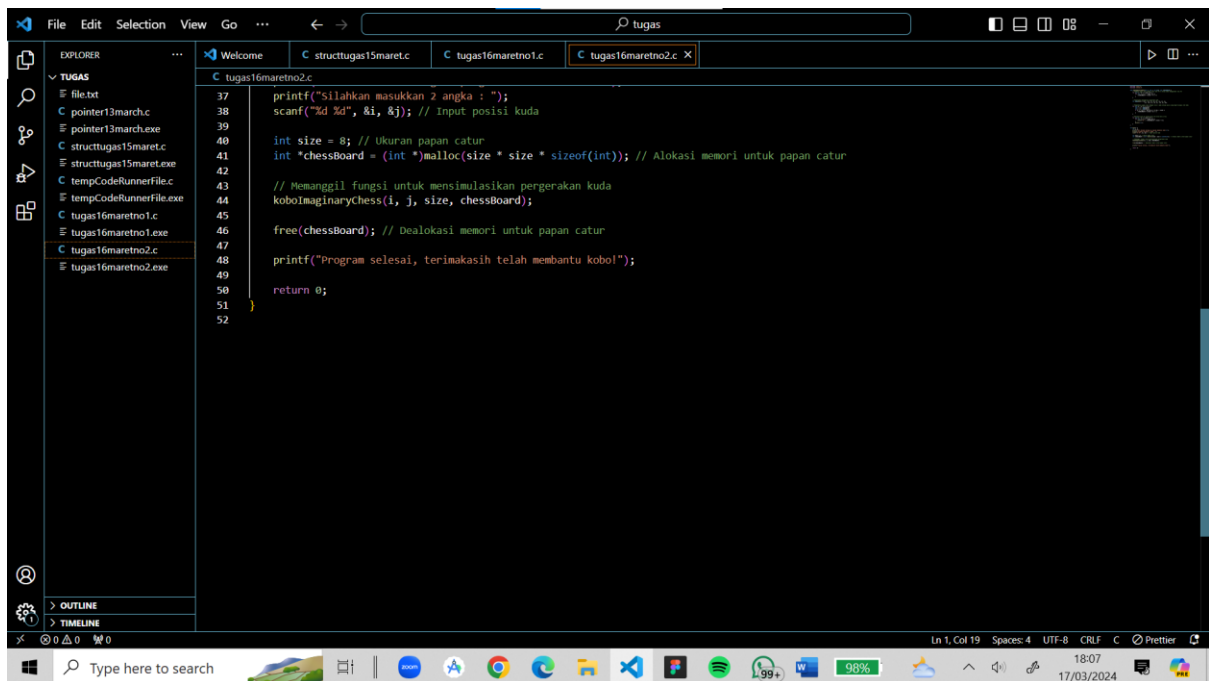
-SS CODE

Baris 1-36



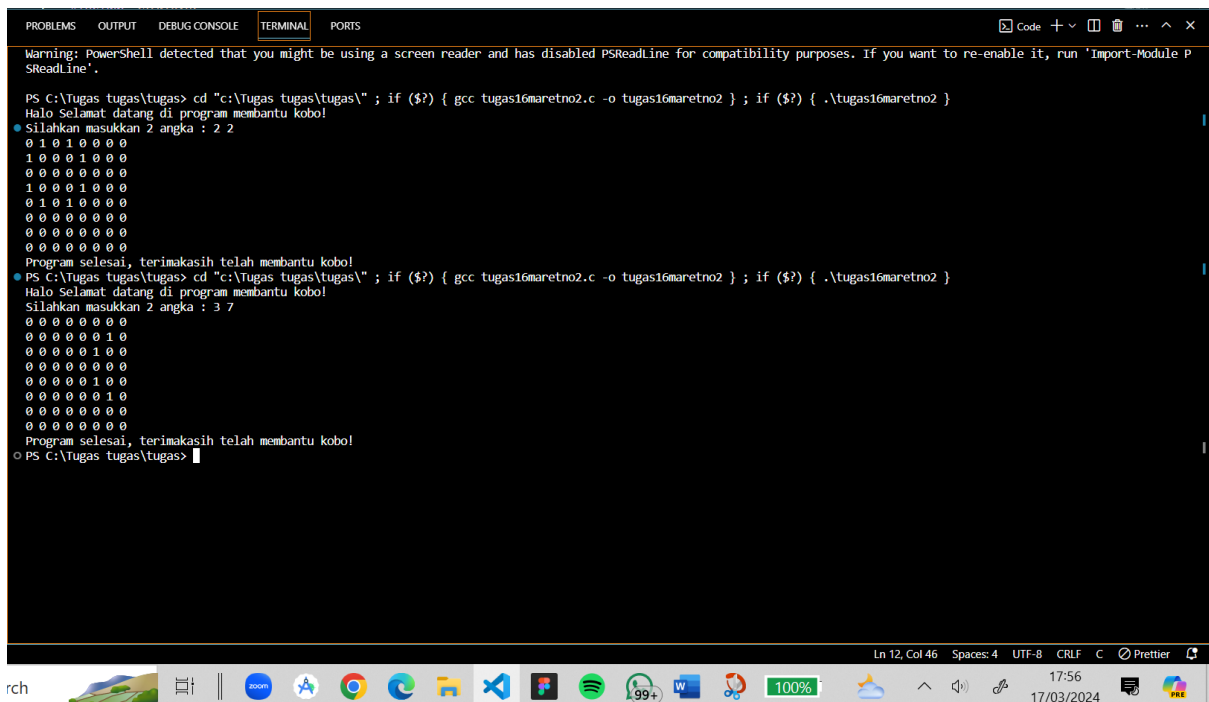
```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 void kobaImaginaryChess(int i, int j, int size, int *chessBoard) {
5     // Membuat papan catur dengan nilai awal 0 di setiap index menggunakan loop for
6     for (int s = 0; s < size; s++) {
7         for (int t = 0; t < size; t++) {
8             *(chessBoard + s*size + t) = 0;
9         }
10    }
11
12    // Menentukan kemungkinan pergerakan kuda
13    int kuda[8][2] = {{-2, -1}, {-2, 1}, {2, -1}, {2, 1},
14                     {-1, -2}, {-1, 2}, {1, -2}, {1, 2}};
15
16    // Memperbarui papan catur dengan nilai 1 pada setiap posisi yang dapat dicapai oleh kuda
17    for (int k = 0; k < 8; k++) {
18        int x = i + kuda[k][0];
19        int y = j + kuda[k][1];
20        if (x >= 0 && x < size && y >= 0 && y < size) {
21            *(chessBoard + x*size + y) = 1;
22        }
23    }
24
25    // Mencetak nilai di setiap baris dan kolom dari array
26    for (int s = 0; s < size; s++) {
27        for (int t = 0; t < size; t++) {
28            printf("%d ", *(chessBoard + s*size + t));
29        }
30        printf("\n");
31    }
32 }
33
34 int main() {
35     int i, j;
36     printf("Halo Selamat datang di program membantu koba!\n");
37     printf("Silahkan masukkan 2 angka < " >>
```

Baris 37-52



```
37 printf("Silahkan masukkan 2 angka : ");
38 scanf("%d %d", &i, &j); // Input posisi kuda
39
40 int size = 8; // Ukuran papan catur
41 int *chessBoard = (int *)malloc(size * size * sizeof(int)); // Alokasi memori untuk papan catur
42
43 // Memanggil fungsi untuk mensimulasikan pergerakan kuda
44 koboImaginaryChess(i, j, size, chessBoard);
45
46 free(chessBoard); // Dealokasi memori untuk papan catur
47
48 printf("Program selesai, terimakasih telah membantu kobo!");
49
50 return 0;
51
52 }
```

-SS OUTPUT (ada 2)



```
Warning: PowerShell detected that you might be using a screen reader and has disabled PSReadLine for compatibility purposes. If you want to re-enable it, run 'Import-Module P
SReadLine'.

PS C:\Tugas tugas\tugas> cd "c:\Tugas tugas\tugas\"; if ($?) { gcc tugas16maretno2.c -o tugas16maretno2 }; if ($?) { .\tugas16maretno2 }
Halo Selamat datang di program membantu kobo!
• Silahkan masukkan 2 angka : 2 2
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
0 0 0 0
Program selesai, terimakasih telah membantu kobo!
• PS C:\Tugas tugas\tugas> cd "c:\Tugas tugas\tugas\"; if ($?) { gcc tugas16maretno2.c -o tugas16maretno2 }; if ($?) { .\tugas16maretno2 }
Halo Selamat datang di program membantu kobo!
Silahkan masukkan 2 angka : 3 7
0 0 0 0
0 0 0 1
0 0 0 0
0 0 0 0
0 0 0 1
0 0 0 0
0 0 0 0
0 0 0 0
Program selesai, terimakasih telah membantu kobo!
PS C:\Tugas tugas\tugas>
```

PENJELASAN CODE NOMOR 2

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

Menggunakan dua jenis header file yang di gunakan untuk fungsi input dan output pada stdio.h kemudian di alokasi memori ke stdlib.h


```
void koboImaginaryChess(int i, int j, int size, int *chessBoard) {
    // Membuat papan catur dengan nilai awal 0 di setiap index menggunakan
    loop for
    for (int s = 0; s < size; s++) {
        for (int t = 0; t < size; t++) {
            *(chessBoard + s*size + t) = 0;
        }
    }
}
```

Mendeklarasi fungsi koboimaginarychess dengan parameter i dan j, size sebagai ukuran dari papan catur dan chessboard sebagai array yang mempresentasikan papan catur. Kemudian kita menggunakan loop nested yang digunakan untuk menginisialisasikan semua elemen papan catur menjadi 0 dengan cara mengakses melalui pointer bernama chessboard.

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

Digunakan untuk menyimpan semua kemungkinan pergerakan dari kuda, setiap elemen array memiliki dua nilai perubahan posisi yaitu x dan y.

```
for (int k = 0; k < 8; k++) {
    int x = i + kuda[k][0];
    int y = j + kuda[k][1];
    if (x >= 0 && x < size && y >= 0 && y < size) {
        *(chessBoard + x*size + y) = 1;
    }
}
```

Loop for diatas digunakan untuk menghitung semua posisi yang mungkin dari pergerakan kuda berdasarkan dari koordinat awal yaitu i dan j, jika posisi yang dihasilkan berada dalam rentang papan catur nilai pada papan catur di posisi tersebut akan menjadi 1.

```
for (int s = 0; s < size; s++) {
    for (int t = 0; t < size; t++) {
        printf("%d ", *(chessBoard + s*size + t));
    }
    printf("\n");
}
```

Loop nested digunakan untuk mencetak nilai nilai papan catur setelah diinisialisasi dan diperbarui. Setiap baris dari papan catur dipisahkan dengan newline yang telah diberikan agar tampilan output tetap rapi.

```

int main() {
    int i, j;
    printf("Halo Selamat datang di program membantu kobo!\n");
    printf("Silahkan masukkan 2 angka : ");
    scanf("%d %d", &i, &j); // Input posisi kuda

    int size = 8; // Ukuran papan catur
    int *chessBoard = (int *)malloc(size * size * sizeof(int)); // Alokasi
memori untuk papan catur

    // Memanggil fungsi untuk mensimulasikan pergerakan kuda
    koboImaginaryChess(i, j, size, chessBoard);

    free(chessBoard); // Dealokasi memori untuk papan catur

    printf("Program selesai, terimakasih telah membantu kobo!");

    return 0;
}

```

Kode diatas adalah fungsi utamanya dimainkan jadi pada kode diatas program akan menampilkan beberapa kalimat dari printf dan kemudian user akan menginput di tempat nya scanf (Dimana user akan menginput posisi kuda) kemudian akan melakukan proses inisialisasi ukuran papan catur dan mengalokasikan memori dan setelah itu memanggil fungsi guna mensimulasikan pergerakan kuda setelah itu proses dealokasi memori papan catur dan program selesai (saat program selesai akan memberi output yang ada pada kalimat printf.s