

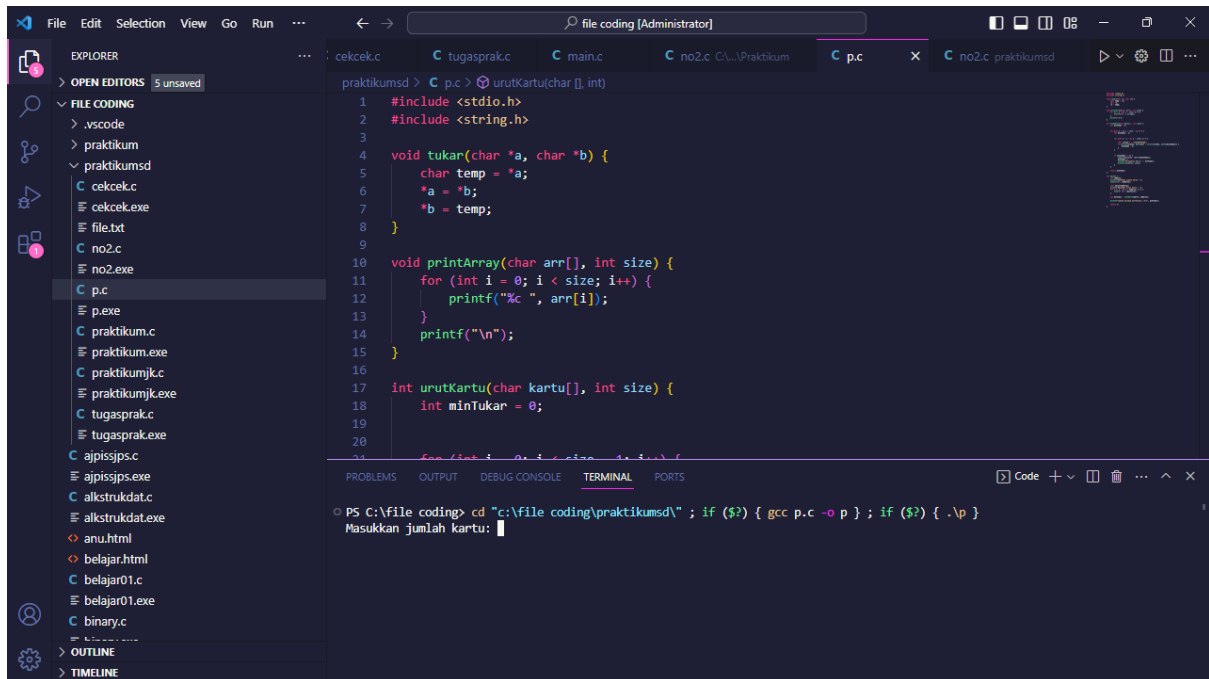
Nama : Muh Aggum Nias putra

NIM :1203230047

Kelas :IF-03-01

Tugas praktikum algoritma dan struktur data

1. Screenshots screen laptop



2. Source code

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

```
#include <string.h>
```

```
void tukar(char *a, char *b) {  
    char temp = *a;  
    *a = *b;  
    *b = temp;  
}
```

```
void printArray(char arr[], int size) {  
    for (int i = 0; i < size; i++) {
```

```

        printf("%c ", arr[i]);
    }
    printf("\n");
}

int urutKartu(char kartu[], int size) {
    int minTukar = 0;

    // Loop untuk setiap elemen di array
    for (int i = 0; i < size - 1; i++) {
        int minIndex = i;

        for (int j = i + 1; j < size; j++) {
            // Membuat urutan "1-10-J-Q-K"
            char urutan[] = "123456789JQK";
            if (strchr(urutan, kartu[j]) < strchr(urutan, kartu[minIndex])) {
                minIndex = j;
            }
        }

        if (minIndex != i) {
            tukar(&kartu[i], &kartu[minIndex]);
            minTukar++;
            printf("Pertukaran ke-%d: ", minTukar);
            printArray(kartu, size);
        }
    }

    return minTukar;
}

```

```
int main() {  
    int noKartu;  
    printf("Masukkan jumlah kartu: ");  
    scanf("%d", &noKartu);  
  
    char kartu[noKartu];  
    printf("Masukkan nilai kartu : ");  
    for (int i = 0; i < noKartu; i++) {  
        scanf(" %c", &kartu[i]);  
    }  
  
    int minTukar =urutKartu(kartu, noKartu);  
  
    printf("Jumlah minimum pertukaran: %d\n", minTukar);  
  
    return 0;  
}
```

3. Output

```

PS C:\file coding> cd "c:\file coding\praktikumsd\" ; if ($?) { gcc p.c -o p } ; if ($?) { .\p }
Masukkan jumlah kartu: 4
Masukkan nilai kartu : 6697
Pertukaran ke-1: 6 6 7 9
Jumlah minimum pertukaran: 1
PS C:\file coding\praktikumsd> cd "c:\file coding\praktikumsd\" ; if ($?) { gcc p.c -o p } ; if ($?) { .\p }
Masukkan jumlah kartu: 5
Masukkan nilai kartu : 3 2 8 7 4
Pertukaran ke-1: 2 3 8 7 4
Pertukaran ke-2: 2 3 4 7 8
Jumlah minimum pertukaran: 2
PS C:\file coding\praktikumsd> cd "c:\file coding\praktikumsd\" ; if ($?) { gcc p.c -o p } ; if ($?) { .\p }
Masukkan jumlah kartu: 6
Masukkan nilai kartu : 10 j k q 3 2
Pertukaran ke-1: 0 1 j k q 3
Pertukaran ke-2: 0 j 1 k q 3
Pertukaran ke-3: 0 j k 1 q 3
Pertukaran ke-4: 0 j k q 1 3
Jumlah minimum pertukaran: 4
PS C:\file coding\praktikumsd>

```

penjelasan code :

ini bertujuan untuk mengurutkan kartu dan menghitung jumlah minimum pertukaran yang diperlukan.

Pertama-tama, program meminta pengguna untuk memasukkan jumlah kartu yang akan diinput. Setelah itu, pengguna diminta untuk memasukkan nilai-nilai kartu tersebut.

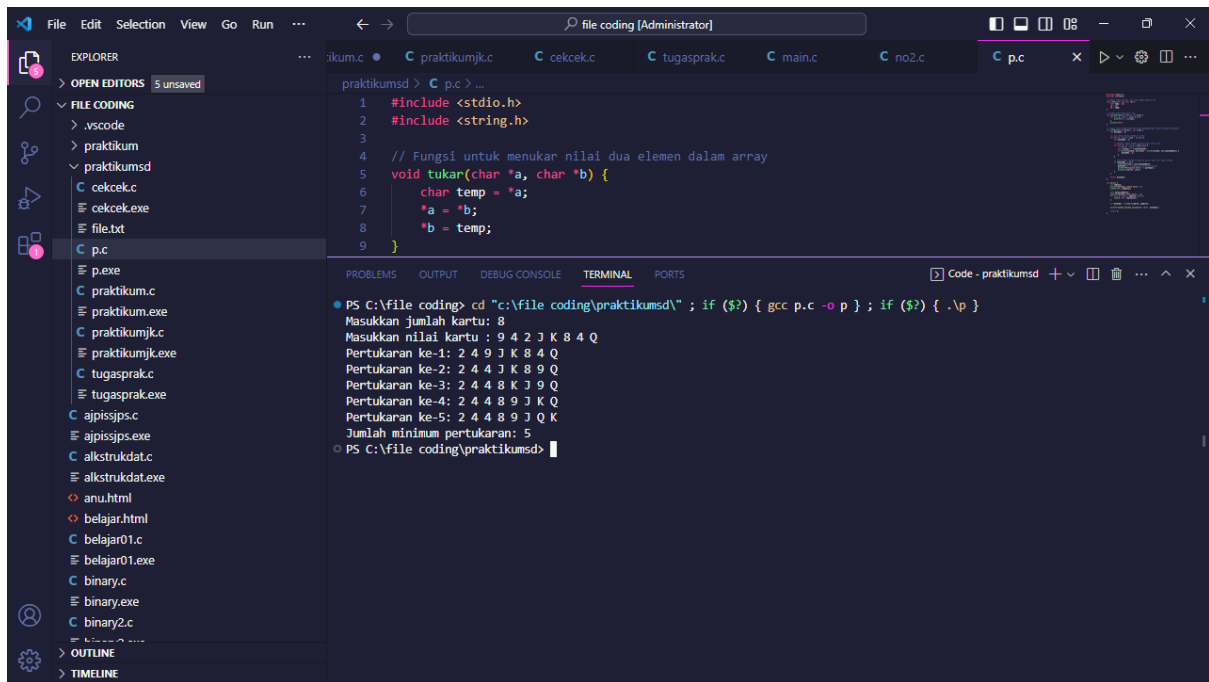
Kemudian, program menggunakan algoritma selection sort yang dimodifikasi untuk mengurutkan kartu sesuai dengan urutan "1-10-J-Q-K". Proses pengurutan ini melibatkan menukar posisi elemen-elemen dalam array kartu sehingga sesuai dengan urutan yang diinginkan.

Setiap kali ada pertukaran elemen, program mencetak pesan yang menunjukkan nomor pertukaran beserta tampilan kartu setelah pertukaran tersebut.

Setelah proses pengurutan selesai, program mencetak jumlah minimum pertukaran yang dilakukan.

Program ini diakhiri dengan mengembalikan nilai 0 sebagai penanda bahwa program telah berakhir dengan sukses.

BONUS :



2. INPUT

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

```
void printBoard(int *chessBoard, int size) {
    for (int i = 0; i < size; i++) {
        for (int j = 0; j < size; j++) {
            printf("%d ", *((chessBoard+i*size) + j));
        }
        printf("\n");
    }
}
```

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

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

```
    for (int k = 0; k < 8; k++) {
        int x = i + moves[k][0];
        int y = j + moves[k][1];
```

```

        if (x >= 0 && x < size && y >= 0 && y < size) {
            *((chessBoard+x*size) + y) = 1;
        }
    }
}

```

```

int main() {
    int i, j;
    scanf("%d %d", &i, &j);

    int size = 8;
    int chessBoard[size][size];

    for (int x = 0; x < size; x++) {
        for (int y = 0; y < size; y++) {
            chessBoard[x][y] = 0;
        }
    }

    koboImaginaryChess(i, j, size, (int *)chessBoard);

    printBoard((int *)chessBoard, size);

    return 0;
}

```

Output :

```
PS C:\file coding> cd "c:\file coding\praktikumsd\" ; if ($?) { gcc no2.c -o no2 } ; if ($?) { .\no2 }
2 2
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:\file coding\praktikumsd> cd "c:\file coding\praktikumsd\" ; if ($?) { gcc no2.c -o no2 } ; if ($?) { .\no2 }
3 7
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:\file coding\praktikumsd> 
```

Penjelasan code :

untuk menampilkan papan catur dan menandai posisi yang dapat dicapai oleh sebuah bidak kuda dari posisi yang ditentukan pengguna. Berikut adalah deskripsi langkah-langkahnya:

1. Di awal program, kita memiliki dua fungsi utama, yaitu printBoard dan koboImaginaryChess, serta fungsi main.
2. Fungsi printBoard bertugas untuk mencetak papan catur ke layar. Papan catur direpresentasikan sebagai matriks integer, di mana setiap elemen matriks menunjukkan apakah posisi tersebut dapat dijangkau oleh bidak kuda atau tidak. Fungsi ini menerima parameter berupa array 2 dimensi chessBoard yang menyimpan informasi posisi, serta ukuran papan catur.
3. Fungsi koboImaginaryChess bertugas untuk menandai posisi yang dapat dicapai oleh bidak kuda dari suatu posisi yang ditentukan. Fungsi ini menerima parameter berupa koordinat i dan j yang merupakan posisi bidak kuda, ukuran papan catur, dan array 2 dimensi chessBoard. Fungsi ini menggunakan aturan gerakan bidak kuda untuk menandai posisi yang dapat dicapai.
4. Di dalam fungsi main, program membaca input dari pengguna berupa koordinat posisi bidak kuda (i dan j). Kemudian, program menginisialisasi papan catur dengan ukuran 8x8 dan mengisi setiap elemennya dengan nilai 0.

5. Setelah itu, program menggunakan fungsi `koboImaginaryChess` untuk menandai posisi yang dapat dicapai oleh bidak kuda dari posisi yang diberikan pengguna.

6. Terakhir, program mencetak papan catur yang telah diubah ke layar menggunakan fungsi `printBoard`.

Program ini membantu pengguna memvisualisasikan posisi-posisi yang dapat dicapai oleh sebuah bidak kuda dari posisi yang ditentukan pada papan catur.