

NAMA : BENEDICTA EKA WIJAYANTI
NIM : 1203230059
KELAS : IF 03 02

NOMOR 1

1. SOURCE CODE

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

// menukar 2 elemen array
void tukar(char *a, char *b)
{
    char temp = *a;
    *a = *b;
    *b = temp;
}

// mencetak array
void cetakArr(char arr[], int size)
{
    for (int i = 0; i < size; i++)
    {
        printf("%c ", arr[i]);
    }
    printf("\n");
}

// mengurutkan kartu
int urutkanKartu(char card[], int size)
{
    int minTukar = 0;

    for (int i = 0; i < size - 1; i++)
    {
        int minIndex = i;

        // menentukan indeks terkecil
        for (int j = i + 1; j < size; j++)
        {
            // membuat urutan
            char urutan[] = "123456789JQK";
            if (strchr(urutan, card[j]) < strchr(urutan, card[minIndex]))
            {
                minIndex = j;
            }
        }
    }
}
```

```

    }
}

// menukar kartu jika tidak ada indeks terkecil
if (minIndex != i)
{
    tukar(&card[i], &card[minIndex]);
    minTukar++;
    printf("Pertukaran ke-%d: ", minTukar);
    cetakArr(card, size);
}
}

return minTukar;
}

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

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

    int minTukar = urutkanKartu(card, noKartu);

    printf("Jumlah minimum pertukaran: %d\n", minTukar);

    return 0;
}

```

2. OUTPUT

```

PS C:\C> cd "c:\C\" ; if ($?) { gcc prak_asd_arraypointer.c -o prak
_asd_arraypointer } ; if ($?) { .\prak_asd_arraypointer }
Masukkan jumlah kartu: 4
Masukkan nilai kartu : 6 6 9 7
Pertukaran ke-1: 6 6 7 9
Jumlah minimum pertukaran: 1
PS C:\C>

```

```

PS C:\C> cd "c:\C\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan jumlah kartu: 8
Masukkan nilai kartu : 9 4 2 J K 8 4 Q
Pertukaran ke-1: 2 4 9 J K 8 4 Q
Pertukaran ke-2: 2 4 4 J K 8 9 Q
Pertukaran ke-3: 2 4 4 8 K J 9 Q
Pertukaran ke-4: 2 4 4 8 9 J K Q
Pertukaran ke-5: 2 4 4 8 9 J Q K
Jumlah minimum pertukaran: 5
PS C:\C>

```

3. PENJELASAN

Program ini untuk mengurutkan kartu dan menampilkan jumlah minimum penukaran ke layar. Terdapat fungsi **tukar** untuk menukar 2 elemen array, terdapat juga **2 pointer** yakni **a** dan **b** yang akan menunjukkan elemen array mana yang ingin ditukar. **temp** berfungsi sebagai tempat penyimpanan sementara.

Selanjutkan fungsi **cetakArr** yang digunakan untuk mencetak elemen array **arr** dan memiliki **2 parameter** yaitu **arr (pointer)** dan **size (ukuran/panjang array)**. Menggunakan perulangan **for loop** untuk mencetak elemen array tersebut.

- **int i = 0;** untuk mendeklarasi variabel **i** dengan tipe data **integer** dan di inisialisasi dengan **nilai 0**.
- **i < size** looping akan terus berjalan selama nilai **i** kurang dari **size**.
- **for (int i = 0; i < size, i++)** artinya perulangan akan dilakukan selama **i** kurang dari **size** dan pada setiap iterasi, nilai **i** akan diincrement/ditambah 1.

Fungsi **urutkanKartu** untuk mengurutkan array **card** dan mengembalikan jumlah minimum pertukarannya. Fungsi ini memiliki 2 parameter yaitu **card (pointer array)** dan **size (ukuran array)**. Mendeklarasikan variabel bernama **minTukar** dengan tipe data integer lalu diinisialisasi dengan nilai 0. Variabel ini digunakan untuk **menghitung jumlah minimum pertukaran**, setiap fungsi **urutkanKartu** menukar 2 elemen **card** maka nilai **minTukar** akan menambah 1.

Terdapat perulangan pertama **for** yang akan diulang sebanyak **size - 1** kali dan perulangan **for** yang lain berguna untuk mencari indeks terkecil dari sisa array yang nantinya akan disimpan di variabel **minIndex**. Perulangan yang kedua menggunakan **strchr** untuk mencari posisi pertama karakter **card[j]** dan **card[minIndex]** muncul dalam string **urutan**. String **urutan** memiliki isi urutan kartu yang valid.

```

if (strchr(urutan, card[j]) < strchr(urutan, card[minIndex]))
{
    minIndex = j;
}

```

Apabila **card[j]** berada di posisi lebih kecil dari pada **card[minIndex]** maka **minIndex** akan diubah menjadi **j**. Contoh : jika **urutan** adalah **'123456789JQK'** dan **card[j] = '5' & card[minIndex] = '3'**. Maka

- **strchr(urutan, '5')** mengembalikan pointer ke lokasi 5 pada string **urutan**
- **strchr(urutan, '3')** mengembalikan pointer ke lokasi 3

Karena lokasi 5 **lebih awal** dari pada lokasi 3 dalam string **urutan**, maka **strchr (urutan, '5') < strchr (urutan, '3')** dan akan bernilai **true**.

Selanjutnya adalah penjelasan kode **if (minIndex != i)**. Kode inti menggunakan selection sort, dimana berfungsi untuk menukar elemen array jika indeks terkecil dalam **minIndex** tidak sama dengan **i**. Jika **minIndex** tidak sama dengan **i**, maka kode didalam **if** akan dijalankan dan berarti ada indeks terkecil yang ditemukan.

Fungsi tukar pada **tukar(&card[i], &card[minIndex])** berguna untuk menukar elemen array **i** dan **minIndex**. Variabel **minTukar++** ditambah 1/ diincrement untuk menghitung **jumlah penukaran yang dilakukan**. Setelah itu akan dicetak menggunakan printf. Fungsi **cetakArr** untuk mencetak **card**.

Terakhir adalah fungsi utama dalam program ini, yaitu berada di **main** yang akan membaca input dari user, menjalankan fungsi pengurutan kartu dan mencetak output.

- **int noKartu** -> mendeklarasikan variabel noKartu dengan tipe data integer (untuk menyimpan jumlah kartu)
- **printf("Masukkan")** -> mencetak pesan kelayar (user akan menginputkan jumlah kartu disini)
- **scanf("%d", noKartu)** -> inputan dari user akan dibaca disini dan disimpan di noKartu
- **char card[noKartu]** -> artinya card bertipe data array dengan ukuran noKartu
- **printf("Masukkan nilai kartu : ")** -> mencetak pesan kelayar (user memasukkan nilai kartu sebanyak jumlah kartu yang diinput sebelumnya)
- **for (int i = 0,)** -> perulangan ini akan membaca nilai kartu yang diinput user satu persatu dan disimpan di card
- **int minTukar =** -> memanggil fungsi **urutkanKartu**, fungsi ini mengembalikan jumlah minimum penukaran
- **printf ("Jumlah minimum")** -> mencetak pesan kelayar supaya user mengetahui berapa kali minimum penukaran kartunya
- **return 0;** -> program akan mengembalikan nilai fungsi main ke 0 yang menandakan program selesai berjalan (sukses).

NOMOR 2

1. SOURCE CODE

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

// untuk mengecek posisi di papan catur
int isValid(int x, int y)
{
    return (x >= 0 && x < 8 && y >= 0 && y < 8);
}
```

```

}

// simulasi gerakan kuda di posisi i dan j
void koboImaginaryChess(int i, int j, int size, int *papanCatur)
{
    // kudaX dan kudaY punya 8 kemungkinan perubahan gerak
    int kudaX[] = {2, 1, -1, -2, -2, -1, 1, 2};
    int kudaY[] = {1, 2, 2, 1, -1, -2, -2, -1};

    // nandain posisi kuda kalau valid bernilai 1
    for (int k = 0; k < 8; k++)
    {
        int x = i + kudaX[k];
        int y = j + kudaY[k];
        if (isValid(x, y))
        {
            papanCatur[x * size + y] = 1; // Menandai posisi sebagai dapat
dicapai
        }
    }
}

int main()
{
    // inisialisasi papan catur
    int papanCatur[8][8] = {0};

    int i, j;
    printf("Masukkan nilai i dan j : ");
    scanf("%d %d", &i, &j); // user masukin posisi awal kuda

    // panggil fungsi untuk simulasi gerakan kuda
    koboImaginaryChess(i, j, 8, (int *)papanCatur);

    // mencetak papan catur hasil dari simulasi
    for (int a = 0; a < 8; a++) // a = row, b = col
    {
        for (int b = 0; b < 8; b++)
        {
            printf("%d ", papanCatur[a][b]);
        }
        printf("\n");
    }

    return 0;
}

```

2. OUTPUT

```
PS C:\C> cd "c:\C\" ; if ($?) { gcc prak_asd_arraypointer.c -o prak
_asd_arraypointer } ; if ($?) { .\prak_asd_arraypointer }
Masukkan nilai i dan j : 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:\C> 
```

```
PS C:\C> cd "c:\C\" ; if ($?) { gcc tempCodeRunnerFile.c -o tempCod
eRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan nilai i dan j : 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:\C> 
```

3. PENJELASAN

Program ini digunakan untuk simulasi pergerakan kuda di papan catur dan memberikan tanda posisi yang dapat dicapai oleh kuda. Terdapat fungsi **isValid** yang berguna untuk mengecek posisi **x** dan **y** dipapan catur valid atau tidak. Fungsi ini akan mengembalikan **true** apabila **x** dan **y** berada diantara 0 dan 7 (batas papan caturnya 8x8) dan mengembalikan **false** jika **x** dan **y** berada diluar batas papan catur.

Selanjutnya adalah fungsi **kobolmaginaryChess** yang berguna untuk mensimulasikan pergerakan kuda dari posisi **i** dan **j** pada papan catur. Fungsi ini memiliki 4 parameter yakni :

- 1) **i** -> posisi awal baris/vertical kuda
- 2) **j** -> posisi awal kolom/horizontal kuda
- 3) **size** -> ukuran papan catur
- 4) **papanCatur** -> pointer yang digunakan untuk menyimpan informasi posisi yang dicapai kuda

Terdapat **8 kemungkinan** perubahan koordinat baris/vertical pada **kudaX** dan **8 kemungkinan** perubahan kolom/horizontal pada **kudaY**. Pada perulangan **for** dilakukan **8 kali perulangan** untuk memeriksa kemungkinan Gerakan kuda. **for (int k = 0; k < 8; k++)** mendefinisikan perulangan yang berjalan 8 kali dari **k = 0 sampai k = 7**.

Didalam perulangan **int x = i + kudaX[k]** menunjukkan perhitungan posisi baris/vertical baru untuk kemungkinan pergerakan kuda. **i** merupakan posisi awal kuda, **kudaX[k]** menyimpan 8 kemungkinan pergerakan kuda sesuai. Lalu tanda penjumlahan (+) digunakan untuk menggabungkan baris awal (i) dengan perubahan **kudaX** untuk

mendapatkan baris baru yang mungkin terbentuk. Demikian pula dengan perulangan **int y = j + kudaY[k]**, dimana **j** merupakan posisi awal kuda (kolom).

Kode **if (isValid(x,y))** berguna untuk memeriksa koordinat **x** dan **y** valid/tidak. **isValid** akan mengembalikan **true** jika koordinat berada didalam batas catur dan mengembalikan **false** jika koordinat berada diluar batas catur. **papanCatur[x * size + y] = 1**; artinya untuk menandai posisi yang dapat dicapai kuda. **papanCatur** adalah array 2 dimensi. Selanjutnya ada rumus **x * size + y**, yang berguna untuk mengkonversi koordinat **x** dan **y** ke **papanCatur**. Lalu kode akan mengubah nilai **papanCatur[x*size+y]** menjadi **1** jika posisi dapat dicapai oleh kuda (pada kondisi ini **isValid** akan mengembalikan nilai **true**).

Selanjutnya adalah fungsi utama, yaitu **main**. Pertama-tama kita mendeklarasi ukuran **papanCatur** dengan ukuran 8x8 dan inisialisasi semua elemen dengan **nilai 0** (untuk menunjukkan posisi yang tidak dapat dicapai/diluar batas papan catur). Setelah itu user akan meng inputkan posisi awal kuda (**i dan j**). Dilanjutkan dengan memanggil fungsi **kobolmaginaryChess** dan mencetak papan catur hasil dari simulasi dengan melakukan perulangan sebanyak 2 kali.

Perulangan pertama untuk **baris(row)** dengan variabel **a** dan rentang **0-7** dan perulangan kedua untuk **kolom(column/col)** dengan variabel **b** dan rentang **0-7**. Kedua perulangan ini berjalan sebanyak 8 kali dan kode akan mencetak 8 baris x 8 kolom