

TUGAS KECIL 3 IF 2211
STRATEGI ALGORITMA

Oleh

Mohamad Hilmi Rinaldi 13520149



PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2022

DAFTAR ISI

DAFTAR ISI	2
BAB 1 ALGORITMA <i>BRANCH AND BOUND</i>	3
BAB 2 <i>SCREENSHOT INPUT DAN OUTPUT PROGRAM</i>	4
2.1. input1.txt (Dapat diselesaikan)	4
2.2. input2.txt (Dapat diselesaikan):	6
2.3. input3.txt (Dapat diselesaikan)	8
2.4. input4.txt (Tidak dapat diselesaikan).....	11
2.5. input5.txt (Tidak dapat diselesaikan).....	12
BAB 3 KODE PROGRAM	14
BAB 4 ALAMAT DRIVE	19

BAB 1

ALGORITMA *BRANCH AND BOUND*

Cara kerja program yang dibuat dalam penyelesaian persoalan 15-Puzzle yaitu diawali dengan memasukkan nama file yang ingin dibaca. Lalu, program akan menyelesaikan puzzle masukan dengan algoritma *Branch and Bound*. Adapun algoritma *Branch and Bound* yang diimplementasikan yaitu :


1. Mencari nilai fungsi KURANG(i) untuk semua elemen pada matriks. Fungsi KURANG(i) didefinisikan sebagai banyaknya elemen bernomor j sedemikian sehingga $j < i$ dan $POSISI(j) > POSISI(i)$. Fungsi POSISI(j) dikatakan lebih besar dari POSISI(i) dapat diartikan bahwa baris atau kolom dari elemen j lebih besar dibandingkan dengan elemen i.
2. Mencari nilai X yang didapatkan dari ((baris elemen + kolom elemen) modulo 2).
3. Menghitung total semua fungsi KURANG(i) ditambah dengan nilai X, apabila hasilnya merupakan ganjil maka puzzle tidak dapat diselesaikan dan program berhenti. Sedangkan jika hasilnya merupakan genap, maka artinya puzzle dapat diselesaikan dan lanjut ke langkah 4.
4. Mencari fungsi cost(i) dengan rumus $f(i) + g(i)$. Fungsi f(i) merupakan banyak langkah untuk mencapai *goal state* dan fungsi g(i) merupakan banyak elemen tidak kosong yang tidak berada pada tempat sesuai susunan *goal state*.
5. Masukkan node akar ke dalam antrian PQueue.
6. Mengambil node pada antrian PQueue yang memiliki cost(i) terkecil.
7. Mencari semua hasil pergeseran dari elemen kosong untuk tiap node yang dapat dilakukan. Jika hasil pergeseran tidak sama dengan hasil pergeseran sebelumnya dan memiliki susunan yang sama dengan *goal state*, maka program akan berhenti.
8. Sebaliknya jika hasil pergeseran tidak sama dengan susunan *goal state* maka node hasil pergeseran dimasukkan ke dalam antrian PQueue dan ulangi langkah 6.

BAB 2

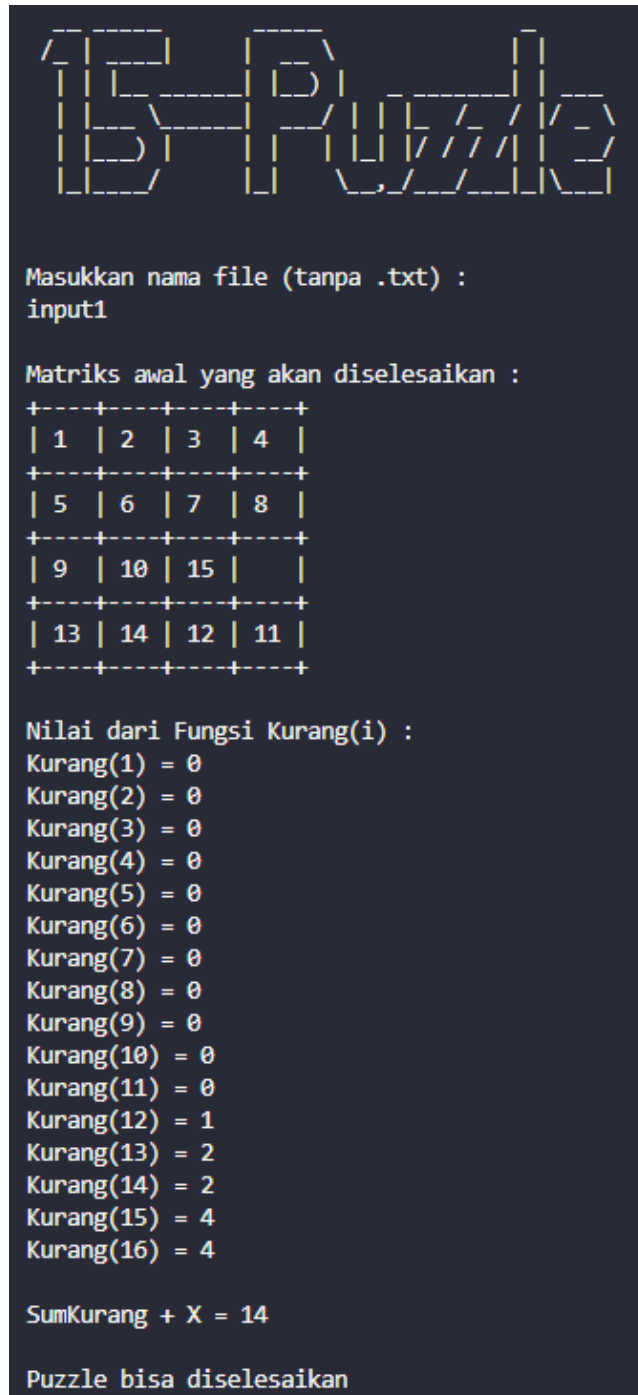
SCREENSHOT INPUT DAN OUTPUT PROGRAM

2.1. input1.txt (Dapat diselesaikan)

Teks yang diuji :

```
test >  input1.txt
```

1	1	2	3	4
2	5	6	7	8
3	9	10	15	0
4	13	14	12	11



Matriks awal :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 15 |   |
+---+---+---+---+
| 13 | 14 | 12 | 11 |
+---+---+---+---+
```

Langkah ke-1 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 15 | 11 |
+---+---+---+---+
| 13 | 14 | 12 |   |
+---+---+---+---+
```

Langkah ke-2 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 15 | 11 |
+---+---+---+---+
| 13 | 14 |   | 12 |
+---+---+---+---+
```

Langkah ke-3 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 |   | 11 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-4 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 11 |   |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-5 :

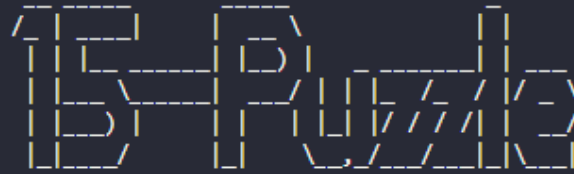
```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 11 | 12 |
+---+---+---+---+
| 13 | 14 | 15 |   |
+---+---+---+---+
```

Waktu eksekusi program : 0.0019948482513427734 detik
Jumlah simpul yang dibangkitkan : 18

2.2. input2.txt (Dapat diselesaikan)

Teks yang diuji :

```
test > ≡ input2.txt
1      1 6 2 4
2      5 3 11 7
3      9 14 10 8
4      13 0 15 12
```



Masukkan nama file (tanpa .txt) :
input2

Matriks awal yang akan diselesaikan :

```
+---+---+---+---+
| 1 | 6 | 2 | 4 |
+---+---+---+---+
| 5 | 3 | 11 | 7 |
+---+---+---+---+
| 9 | 14 | 10 | 8 |
+---+---+---+---+
| 13 |   | 15 | 12 |
+---+---+---+---+
```

Nilai dari Fungsi Kurang(i) :

```
Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 0
Kurang(4) = 1
Kurang(5) = 1
Kurang(6) = 4
Kurang(7) = 0
Kurang(8) = 0
Kurang(9) = 1
Kurang(10) = 1
Kurang(11) = 4
Kurang(12) = 0
Kurang(13) = 1
Kurang(14) = 4
Kurang(15) = 1
Kurang(16) = 2
```

SumKurang + X = 20

Puzzle bisa diselesaikan

Matriks awal :

```
+---+---+---+---+
| 1 | 6 | 2 | 4 |
+---+---+---+---+
| 5 | 3 | 11 | 7 |
+---+---+---+---+
| 9 | 14 | 10 | 8 |
+---+---+---+---+
| 13 |   | 15 | 12 |
+---+---+---+---+
```

Langkah ke-1 :

```
+---+---+---+---+
| 1 | 6 | 2 | 4 |
+---+---+---+---+
| 5 | 3 | 11 | 7 |
+---+---+---+---+
| 9 |   | 10 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-2 :

```
+---+---+---+---+
| 1 | 6 | 2 | 4 |
+---+---+---+---+
| 5 | 3 | 11 | 7 |
+---+---+---+---+
| 9 | 10 |   | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-3 :

```
+---+---+---+---+
| 1 | 6 | 2 | 4 |
+---+---+---+---+
| 5 | 3 |   | 7 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-4 :

```
+---+---+---+---+
| 1 | 6 | 2 | 4 |
+---+---+---+---+
| 5 |   | 3 | 7 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-5 :

```
+---+---+---+---+
| 1 |   | 2 | 4 |
+---+---+---+---+
| 5 | 6 | 3 | 7 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-6 :

```
+---+---+---+---+
| 1 | 2 |   | 4 |
+---+---+---+---+
| 5 | 6 | 3 | 7 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-7 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 |   | 7 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-8 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 |   |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-9 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 11 |   |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```


Langkah ke-10 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 11 | 12 |
+---+---+---+---+
| 13 | 14 | 15 |   |
+---+---+---+---+
```

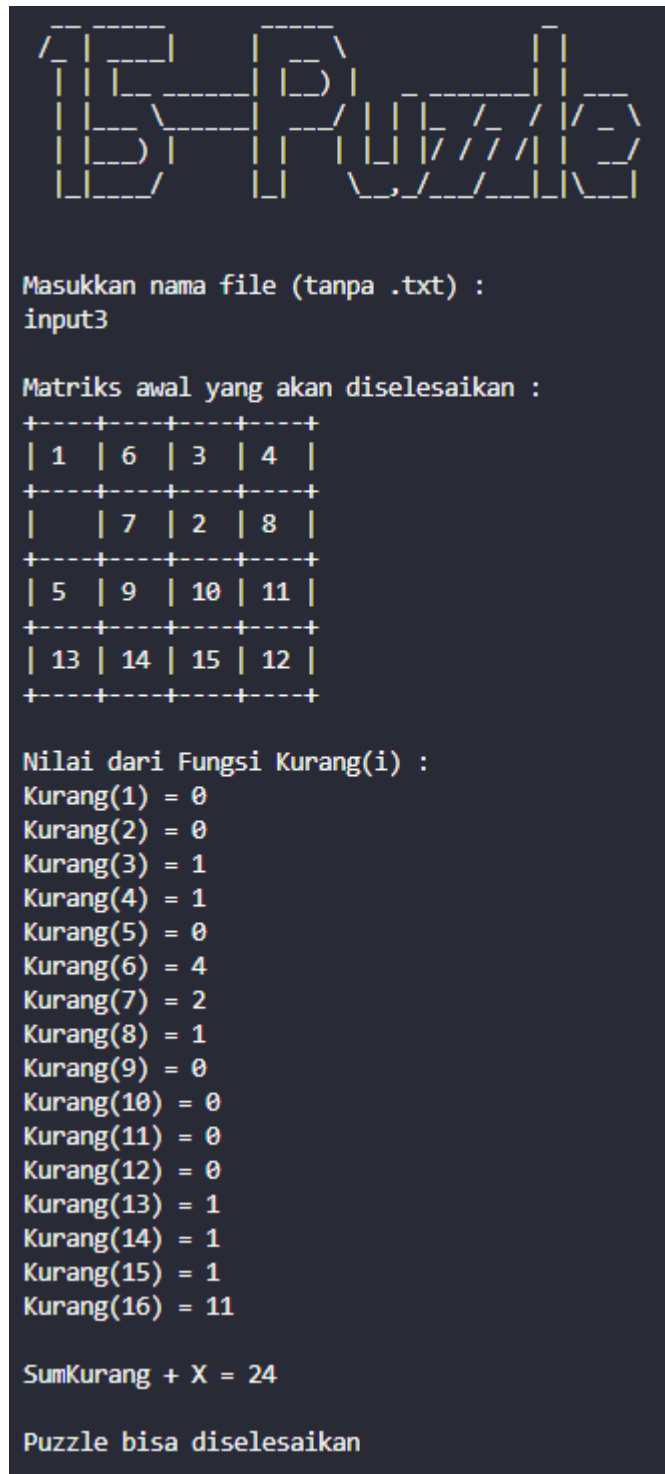
Waktu eksekusi program : 0.002991914749145508 detik
Jumlah simpul yang dibangkitkan : 32

2.3. input3.txt (Dapat diselesaikan)

Teks yang diuji :

```
test >  input3.txt
```

1	1	6	3	4
2	0	7	2	8
3	5	9	10	11
4	13	14	15	12



Langkah ke-1 :

```
+---+---+---+---+
| 1 | 6 | 3 | 4 |
+---+---+---+---+
| 5 | 7 | 2 | 8 |
+---+---+---+---+
|   | 9 | 10 | 11 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-2 :

```
+---+---+---+---+
| 1 | 6 | 3 | 4 |
+---+---+---+---+
| 5 | 7 | 2 | 8 |
+---+---+---+---+
| 9 |   | 10 | 11 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-3 :

```
+---+---+---+---+
| 1 | 6 | 3 | 4 |
+---+---+---+---+
| 5 | 7 | 2 | 8 |
+---+---+---+---+
| 9 | 10 |   | 11 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-4 :

```
+---+---+---+---+
| 1 | 6 | 3 | 4 |
+---+---+---+---+
| 5 | 7 | 2 | 8 |
+---+---+---+---+
| 9 | 10 | 11 |   |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-5 :

```
+---+---+---+---+
| 1 | 6 | 3 | 4 |
+---+---+---+---+
| 5 | 7 | 2 |   |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-6 :

```
+---+---+---+---+
| 1 | 6 | 3 |   |
+---+---+---+---+
| 5 | 7 | 2 | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-7 :

```
+---+---+---+---+
| 1 | 6 |   | 3 |
+---+---+---+---+
| 5 | 7 | 2 | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-8 :

```
+---+---+---+---+
| 1 | 6 | 2 | 3 |
+---+---+---+---+
| 5 | 7 |   | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-9 :

```
+---+---+---+---+
| 1 | 6 | 2 | 3 |
+---+---+---+---+
| 5 |   | 7 | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-10 :

```
+---+---+---+---+
| 1 |   | 2 | 3 |
+---+---+---+---+
| 5 | 6 | 7 | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-11 :

```
+---+---+---+---+
| 1 | 2 |   | 3 |
+---+---+---+---+
| 5 | 6 | 7 | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-12 :

```
+---+---+---+---+
| 1 | 2 | 3 |   |
+---+---+---+---+
| 5 | 6 | 7 | 4 |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-13 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 |   |
+---+---+---+---+
| 9 | 10 | 11 | 8 |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-14 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 11 |   |
+---+---+---+---+
| 13 | 14 | 15 | 12 |
+---+---+---+---+
```

Langkah ke-15 :

```
+---+---+---+---+
| 1 | 2 | 3 | 4 |
+---+---+---+---+
| 5 | 6 | 7 | 8 |
+---+---+---+---+
| 9 | 10 | 11 | 12 |
+---+---+---+---+
| 13 | 14 | 15 |   |
+---+---+---+---+
```

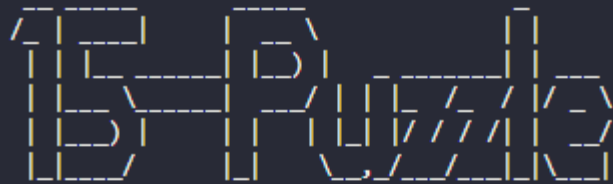
Waktu eksekusi program : 0.030542850494384766 detik
Jumlah simpul yang dibangkitkan : 792

2.4. input4.txt (Tidak dapat diselesaikan)

Teks yang diuji :

```
test > ≡ input4.txt
```

1	1	3	4	15
2	2	0	5	12
3	7	6	11	14
4	8	9	10	13



Masukkan nama file (tanpa .txt) :
input4

Matriks awal yang akan diselesaikan :

1	3	4	15
2	0	5	12
7	6	11	14
8	9	10	13

Nilai dari Fungsi Kurang(i) :

Kurang(1) = 0
Kurang(2) = 0
Kurang(3) = 1
Kurang(4) = 1
Kurang(5) = 0
Kurang(6) = 0
Kurang(7) = 1
Kurang(8) = 0
Kurang(9) = 0
Kurang(10) = 0
Kurang(11) = 3
Kurang(12) = 6
Kurang(13) = 0
Kurang(14) = 4
Kurang(15) = 11
Kurang(16) = 10

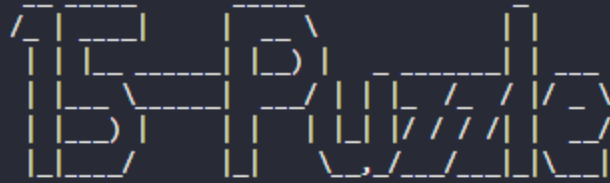
SumKurang + X = 37

Puzzle tidak bisa diselesaikan

2.5. input5.txt (Tidak dapat diselesaikan)

Teks yang diuji :

```
test > ≡ input5.txt
1  2 7 3 4
2  1 9 0 12
3  5 8 6 10
4  13 14 11 15
```



Masukkan nama file (tanpa .txt) :
input5

Matriks awal yang akan diselesaikan :

```
+---+---+---+---+
| 2 | 7 | 3 | 4 |
+---+---+---+---+
| 1 | 9 |   | 12 |
+---+---+---+---+
| 5 | 8 | 6 | 10 |
+---+---+---+---+
| 13 | 14 | 11 | 15 |
+---+---+---+---+
```

Nilai dari Fungsi Kurang(i) :

```
Kurang(1) = 0
Kurang(2) = 1
Kurang(3) = 1
Kurang(4) = 1
Kurang(5) = 0
Kurang(6) = 0
Kurang(7) = 5
Kurang(8) = 1
Kurang(9) = 3
Kurang(10) = 0
Kurang(11) = 0
Kurang(12) = 5
Kurang(13) = 1
Kurang(14) = 1
Kurang(15) = 0
Kurang(16) = 9
```

SumKurang + X = 29

Puzzle tidak bisa diselesaikan

Poin	Ya	Tidak
1. Program berhasil dikompilasi	✓	
2. Program berhasil <i>running</i>	✓	
3. Program dapat menerima input dan menuliskan output	✓	
4. Luaran sudah benar untuk semua data uji	✓	
5. Bonus dibuat		✓

BAB 3

KODE PROGRAM

```
import time
import copy
import heapq as hq

def convert_matriks(nama_file):
    with open("../test/" + nama_file, "r") as f:
        matriks = [[int(num) for num in line.split(' ')] for line in f]
    return matriks

def print_matriks(matriks):
    print("+---+---+---+---+")
    for i in range(4):
        print("", end='| ')
        for j in range(4):
            if(matriks[i][j] < 10 and matriks[i][j]>0):
                print(matriks[i][j],end=' | ')
            elif(matriks[i][j] == 0):
                print(" ",end=' | ')
            else:
                print(matriks[i][j],end=' | ')
        print()
        print("+---+---+---+---+")

def kurang(matriks, i):
    kurang = 0
    x_null,y_null = cari_null(matriks)

    # Jika yang dicari fungsi Kurang(16) maka nilai pada matriks bernilai 0
    if(i == 16):
        i = 0

    x_titik,y_titik = cari_titik(matriks, i)

    # Menset matriks yang bernilai 0 menjadi 16
    matriks[x_null][y_null] = 16

    # Menghitung nilai fungsi kurang
    for y in range(y_titik+1,4):
        if(matriks[x_titik][y_titik] > matriks[x_titik][y]):
            kurang += 1
```

```

    for x in range(x_titik+1,4):
        for y in range(0,4):
            if(matriks[x_titik][y_titik] > matriks[x][y]):
                kurang += 1

    # Menset matriks yang bernilai 16 menjadi 0
    matriks[x_null][y_null] = 0

    return kurang

def total_sumkurang_X(matriks):
    total = 0
    x_null,y_null = cari_null(matriks)

    # Total dari fungsi kurang
    for i in range(0,4):
        for j in range(0,4):
            total += kurang(matriks, matriks[i][j])

    # Mencari nilai X
    X = (x_null + y_null) % 2

    return total + X

def cari_null(matriks):
    for i in range(0,4):
        for j in range(0,4):
            if(matriks[i][j] == 0):
                return i,j

def cari_titik(matriks, x):
    for i in range(0,4):
        for j in range(0,4):
            if(matriks[i][j] == x):
                return i,j

def cost(matriks, target, kedalaman):
    ctr = 0
    for i in range(0,4):
        for j in range(0,4):
            if(matriks[i][j] != 0 and matriks[i][j] != target[i][j]):
                ctr += 1

```

```

    return ctr + kedalaman

def cek_kesamaan_matriks(matriks, target):
    for i in range(0,4):
        for j in range(0,4):
            if(matriks[i][j] != target[i][j]):
                return False
    return True

def cek_matriks_sudah_ada(matriks, list_matriks):
    for i in range(0,len(list_matriks)):
        if(cek_kesamaan_matriks(matriks, list_matriks[i])):
            return True
    return False

def geser_matriks(matriks, x_null, y_null, i):
    if(i == 0 and x_null != 0): # geser ke atas
        swap_matriks(matriks, x_null-1, y_null)
        return True
    elif(i == 1 and x_null != 3): # geser ke bawah
        swap_matriks(matriks, x_null+1, y_null)
        return True
    elif(i == 2 and y_null != 0): # geser ke kiri
        swap_matriks(matriks, x_null, y_null-1)
        return True
    elif(i == 3 and y_null != 3): # geser ke kanan
        swap_matriks(matriks, x_null, y_null+1)
        return True

def swap_matriks(matriks, x, y):
    x_null,y_null = cari_null(matriks)
    matriks_temp = matriks[x][y]
    matriks[x][y] = matriks[x_null][y_null]
    matriks[x_null][y_null] = matriks_temp

def masukan_node(node, list_node, list_node_expand):
    for i in range(len(list_node)):
        list_node_expand.append(list_node[i])
    list_node_expand.append(node)

def print_langkah(list_matriks):
    for i in range(1, len(list_matriks)):
        print("\nLangkah ke-"+str(i)+" :")
        print_matriks(list_matriks[i])

```



```

ketemu = cek_kesamaan_matriks(matriks, matriks_target)

# Algoritma Branch and Bound
while(len(PQueue) != 0 and not ketemu):
    matriks_tmp = hq.heappop(PQueue)
    x_null, y_null = cari_null(matriks_tmp[1])

    # Mengexpand node
    for i in range(0,4):
        children = copy.deepcopy(matriks_tmp[1])
        # Menggeser nilai null matriks (0)
        gerak = geser_matriks(children, x_null, y_null, i)
        if(gerak): # Jika gerakan bisa dilakukan
            if(not cek_matriks_sudah_ada(children, matriks_tmp[2])):
                matriks_solusi = []
                masukan_node(children, matriks_tmp[2], matriks_solusi)
                ctr_node += 1
                # Jika sudah mencapai target maka print langkah untuk
                # mencapai matriks target
                if(cek_kesamaan_matriks(children, matriks_target)):
                    ketemu = True
                    end = time.time()
                    print_langkah(matriks_solusi)
                    print("\nWaktu eksekusi program : " + str((end-start) ) +
                        " detik")
                    print("Jumlah simpul yang dibangkitkan : "+
                        str(ctr_node))
                    break
                # Jika belum mencapai target maka masukkan ke Priority Queue
            else:
                hq.heappush(PQueue, (cost(children, matriks_target,
                    matriks_tmp[3]+1), children, matriks_solusi,
                    matriks_tmp[3]+1))

else: # Jika SumKurang + X = ganjil
    print("\nPuzzle tidak bisa diselesaikan")

```

BAB 4
ALAMAT DRIVE

https://github.com/mhilmirinaldi/Tucil3_13520149