

LAPORAN TUGAS KECIL 3 IF2211 Strategi Algoritma

Semester II tahun 2021/2022

Penyelesaian Persoalan 15-Puzzle dengan Algoritma Branch and Bound



Steven Gianmarg Haposan Siahaan
13520145

Program Studi Teknik Informatika
Sekolah Tinggi Elektro dan Informatika Institut
Teknologi Bandung
2022

Daftar Isi

Daftar Isi	2
BAB I	3
Landasan Teori	3
1.1 Branch and Bound	3
1.2 15-Puzzle	3
BAB II	4
Program dan Spesifikasi Tugas	4
2.1 Spesifikasi Tugas	4
2.2 Implementasi Branch and Bound pada 15-Puzzle	5
2.3 Source Code	6
2.4 Penjelasan Fungsi-fungsi dan Kelas yang terdapat pada Program	11
2.5 Screenshoot Hasil Program	12
BAB III	49
Penutup	49
3.1 Saran	49
3.2 Kesimpulan	49
DAFTAR PUSTAKA	50
Lampiran	51

BAB I

Landasan Teori

1.1 Branch and Bound

Branch and bound adalah paradigma desain algoritma yang umumnya digunakan untuk memecahkan masalah optimasi kombinatorial. Masalah-masalah ini biasanya eksponensial dalam hal kompleksitas waktu dan mungkin perlu mengeksplorasi semua kemungkinan permutasi dalam kasus terburuk. Teknik Algoritma Cabang dan Terikat memecahkan masalah ini dengan relatif cepat.

1. Masukkan simpul akar ke dalam antrian Q. Jika simpul akar adalah simpul solusi (goal node), maka solusi telah ditemukan. Jika hanya satu solusi yang diinginkan, maka stop.
2. Jika Q kosong, Stop.
3. Jika Q tidak kosong, pilih dari antrian Q simpul i yang mempunyai nilai 'cost' $\hat{c}(i)$ paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih satu secara sembarang.
4. Jika simpul i adalah simpul solusi, berarti solusi sudah ditemukan. Jika satu solusi yang diinginkan, maka stop. Pada persoalan optimasi dengan pendekatan least cost search, periksa cost semua simpul hidup. Jika cost nya lebih besar dari cost simpul solusi, maka matikan simpul tersebut.
5. Jika simpul i bukan simpul solusi, maka bangkitkan semua anak-anaknya. Jika i tidak mempunyai anak, kembali ke langkah 2.
6. Untuk setiap anak j dari simpul i , hitung $\hat{c}(j)$, dan masukkan semua anak-anak tersebut ke dalam Q.
7. Kembali ke langkah 2.

1.2 15-Puzzle

15-Puzzle adalah teka-teki geser yang memiliki 15 ubin persegi bernomor 1–15 dalam bingkai dengan tinggi 4 ubin dan lebar 4 ubin, menyisakan satu ubin kosong(pada program ini digunakan angka 16). posisi ubin. Ubin di baris atau kolom yang sama dari posisi terbuka dapat dipindahkan dengan menggesernya secara horizontal(kanan atau kiri) atau vertikal(atas atau bawah). Tujuan dari teka-teki adalah untuk menempatkan ubin dalam urutan numerik. Dinamakan berdasarkan jumlah ubin dalam bingkai, teka-teki 15 juga dapat disebut teka-teki 16, mengacu pada total kapasitas ubinnya.

Teka-teki n adalah masalah klasik untuk algoritma pemodelan yang melibatkan heuristik. Heuristik yang umum digunakan untuk masalah ini termasuk menghitung jumlah ubin yang salah tempat dan menemukan jumlah jarak taksi antara setiap blok dan posisinya dalam konfigurasi tujuan. Perhatikan bahwa keduanya dapat diterima, yaitu mereka tidak pernah melebihi-lebihkan jumlah gerakan yang tersisa, yang memastikan optimalitas untuk algoritme pencarian tertentu

BAB II

Program dan Spesifikasi Tugas

2.1 Spesifikasi Tugas

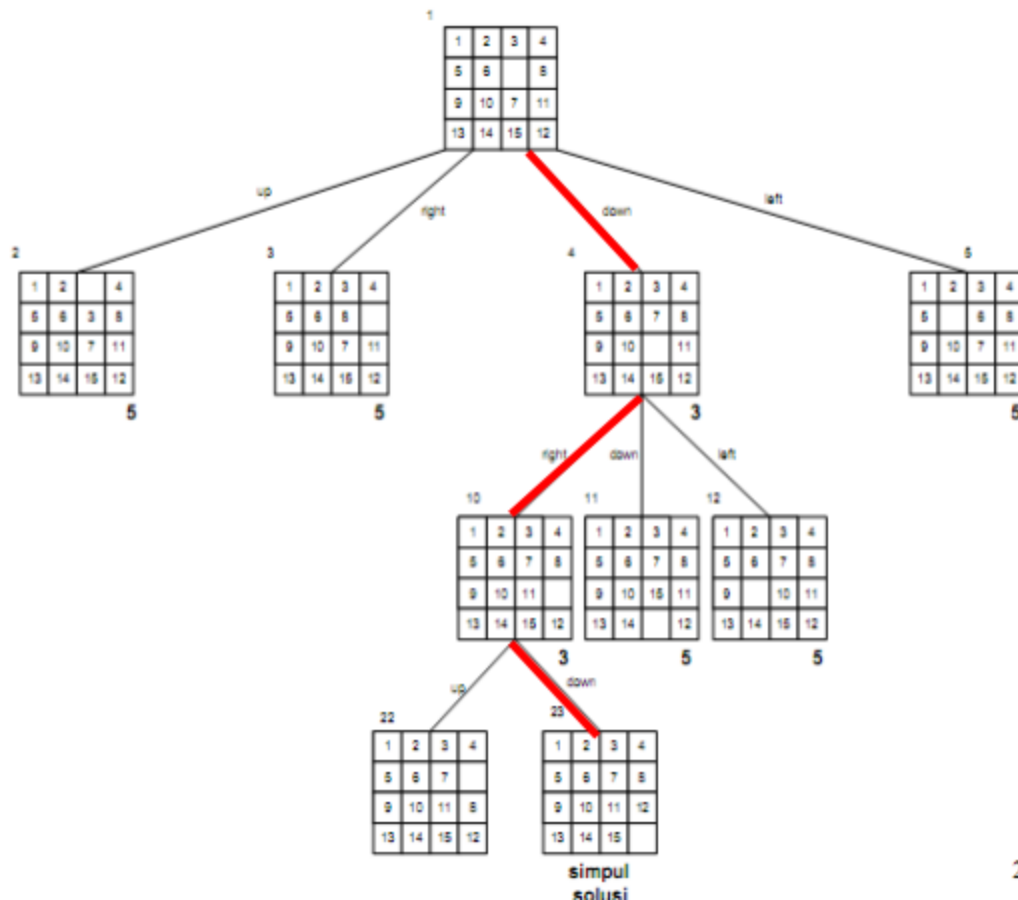
Buatlah program dalam Java/Python untuk menyelesaikan persoalan 15-Puzzle dengan menggunakan Algoritma Branch and Bound seperti pada materi kuliah. Nilai bound tiap simpul adalah penjumlahan cost yang diperlukan untuk sampai suatu simpul x dari akar, dengan taksiran cost simpul x untuk sampai ke goal. Taksiran cost yang digunakan adalah jumlah ubin tidak kosong yang tidak berada pada tempat sesuai susunan akhir (goal state). Untuk semua instansiasi persoalan 15-puzzle, susunan akhir yang diinginkan sesuai dengan Gambar 1.

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

Gambar 2.1 Susunan Akhir Persoalan 15-Puzzle

Masukan: matriks yang merepresentasikan posisi awal suatu instansiasi persoalan 15-puzzle. Posisi awal 15-puzzle dibangkitkan secara acak oleh program dan/atau dimasukkan dari file teks. Program harus dapat menentukan apakah posisi awal suatu masukan dapat diselesaikan hingga mencapai susunan akhir, dengan mengimplementasikan fungsi Kurang(i) dan posisi ubin kosong di kondisi awal (X), seperti pada materi kuliah. Jika posisi awal tidak bisa mencapai susunan akhir, program akan menampilkan pesan tidak bisa diselesaikan,. Jika dapat diselesaikan, program dapat menampilkan urutan matriks rute (path) aksi yang dilakukan dari posisi awal ke susunan akhir. Sebagai contoh pada Gambar 2, matriks yang ditampilkan ke layar adalah matriks pada simpul 1, simpul 4, simpul 10 dan simpul 23. Luanan:

1. Matriks posisi awal 15-puzzle.
2. Nilai dari fungsi Kurang (i) untuk setiap ubin tidak kosong pada posisi awal (nilai ini tetap dikeluarkan, baik persoalan bisa diselesaikan atau tidak bisa diselesaikan).
3. Nilai dari $\sum_{i=1}^{16} \text{KURANG}(i) + X$
4. Jika persoalan tidak dapat diselesaikan (berdasarkan hasil butir 2) keluar pesan.
5. Jika persoalan dapat diselesaikan (berdasarkan hasil butir 2), menampilkan urutan matriks dari posisi awal ke posisi akhir seperti pada penjelasan sebelumnya.
6. Waktu eksekusi program (diluar baca input/tuis output)
7. Jumlah simpul yang dibangkitkan di dalam pohon ruang status pencarian.



Gambar 2.2 Contoh Pohon Ruang Status Persoalan 15-Puzzle

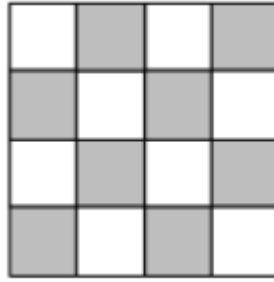
Data Uji: Buatlah 5 buah instansiasi persoalan 15-puzzle, dengan 2 kasus tidak dapat diselesaikan dan 3 kasus yang dapat diselesaikan. Instansiasi persoalan juga disertakan dalam pengumpulan Tugil 3.

Bonus: Program dibuat secara interaktif dengan graphical user interface (GUI), dapat menampilkan gambar 15-puzzle grafis yang menarik seperti di bawah ini, dan dapat menampilkan pergeseran ubin.

2.2 Implementasi Branch and Bound pada 15-Puzzle

Penjelasan singkat program :

1. Program menerima matriks yang berukuran 4x4(dari file masukan eksternal/input manual).
2. Matriks tersebut akan di baca dan akan di cek apakah dapat di selesaikan atau tidak.
3. Pengecekan dilakukan dengan mengecek nilai dari fungsi kurang(i) keseluruhan dengan ditambah X ,dimana $\sum_{i=1}^{16} KURANG(i) + X$
4. Fungsi kurang(i) adalah banyaknya ubin bernomor j sedemikian sehingga $j < i$ dan $POSISI(j) > POSISI(i)$,sedangkan X adalah nilai yang muncul akibat dari posisi blank pada puzzle, jika blank puzzle berada pada daerah yang diarsir maka X akan bernilai 1 dan jika tidak maka bernilai 0. Pada gambar ini tergambarakan jika blank berada pada $(index\ row + index\ col) \% 2 == 0$ maka X bernilai 0, jika $(index\ row + index\ col) \% 2 \neq 0$ maka X bernilai 1 .



Gambar 2.3 Posisi Blank pada Puzzle

5. Jika nilai dari $\sum_{i=1}^{16} \text{KURANG}(i) + X$ adalah genap maka puzzle dapat diselesaikan. Jika bernilai ganjil, maka tidak dapat diselesaikan.
6. Jika dapat diselesaikan, maka matriks tersebut dijadikan sebagai elemen simpul akar dari tree Branch and Bound. Lalu setelah itu akan dibangkitkan simpul lainnya yang merupakan child dari simpul akar saat ini, dengan menggunakan sistem priority queue(antrian).Antrian ini akan memprioritaskan nilai fungsi cost terkecil.
7. Simpul-simpul yang dibangkitkan tersebut berdasarkan arah pergeseran ubin kosong(blank) atau dalam program ini ditandai dengan 16. Pergerakan yang diatur dalam program ini secara berurutan adalah up,right,down,left. Pada proses pergerakan ini harus berhati-hati agar pergerakan tidak bolak-balik(naik lalu turun, atau kanan lalu kiri, karena akan sama saja balik ke posisi awal). Selain itu juga harus diperhatikan, apakah posisi blank(yang ditandai dengan 16) masih berada di dalam index matrix yang aman untuk digeser sesuai pergerakan yang di mau.
8. Semua simpul yang telah dibangkitkan akan di masukkan ke dalam priority queue dengan prioritas adalah nilai cost yang lebih kecil.
9. Prinsipnya tetap sama seperti queue pada biasanya yakni First In First Out, namun dengan adanya tambahan parameter cost sebagai prioritas.
10. Melakukan aksi pop elemen, apabila matriks simpul yang dipop dari PrioQueue maka puzzle sudah terselesaikan. Jika berbeda,ulangi Langkah ke -6.
11. Untuk setiap simpul yang di pop akan di masukkan ke dalam visited agar tidak terjadi pemeriksaan terhadap simpula yang sudah dilewati(menghindari pembangkitan simpul dengan matriks duplikat.

$$\hat{c}(i) = \hat{f}(i) + \hat{g}(i)$$

$\hat{c}(i)$ = ongkos untuk simpul i

$\hat{f}(i)$ = ongkos mencapai simpul i dari akar

$\hat{g}(i)$ = ongkos mencapai simpul tujuan dari simpul i .

Gambar 2. Pendekatan fungsi cost untuk persoalan

2.3 Source Code

```
import copy
import time
import numpy as np
import queue
# Jika menginginkan ukuran puzzle lebih besar
N=4 #Ubah angka pada N= sesuai ukuran puzzle,pada tugas ini diminta 15-Puzzle
final = [[1, 2, 3, 4], [5, 6, 7, 8], [9, 10, 11, 12], [13, 14, 15, 16]] #Puzzle akhir
```

```

matrix=[[0 for i in range(N)] for i in range (N)] #Membuat array yang berisikan 0 sebanyak 16
elemen

def getBlankIndex(matrix):
    for i in range(N):
        for j in range(N):
            if(matrix[i][j]==16):
                return [i,j]
def kurang(matrix):
    index=getBlankIndex(matrix)
    mat1 = []
    # ubah jadi 1d
    for a in matrix:
        for b in a:
            mat1.append(b)
    matrix = mat1
    kurang=[]
    pos=0
    sigma_kurang=0
    for i in range(N*N):
        tmp=matrix[i]
        count_kurang=0
        for j in range(i+1,16):
            if(tmp > matrix[j] and tmp!=0):
                sigma_kurang+=1
    if((index[0]+index[1])%2==0):
        sigma_kurang=sigma_kurang+0
    else :
        sigma_kurang=sigma_kurang+1
    return (sigma_kurang)
def display_kurang(matrix):
    index=getBlankIndex(matrix)
    mat1 = []
    # ubah jadi 1d
    for a in matrix:
        for b in a:
            mat1.append(b)
    matrix = mat1
    kurang=[]
    pos=0
    for i in range(N*N):
        tmp=matrix[i]
        count_kurang=0
        for j in range(i+1,16):
            if(tmp > matrix[j] and tmp!=0):
                count_kurang += 1
        kurang+=[[tmp,count_kurang]]
    kurang.sort() #sort fungsi kurang ke-i
    for k in range(N*N):
        print("Nilai dari fungsi Kurang(" + str(kurang[k][0])+"):" ,kurang[k][1])

```

```

def cost(matrix): #tingkat start dari 1
    count = 0
    for i in range(N):
        for j in range(N):
            if((matrix[i][j] != final[i][j]) and matrix[i][j] != 16):
                count += 1
    return (count)
def punyasolusi(matrix):
    if(kurang(matrix)%2==0):
        return True
    else:
        return False
# queue
class Node:
    def __init__(self, parent, matrix, cost, blank, level,command):
        #parent
        self.parent = parent
        #matrix
        self.matrix = matrix
        # fungsi cost
        self.cost = cost
        # posisi block kosong
        self.blank = blank
        # level nodenya
        self.level = level
        # arah geraknya
        self.command= command
    #priority queue
    def __lt__(self, other): # prioritas cost lalu level
        return(self.cost+self.level <= other.cost + other.level)
        # if self.cost == other.cost:
        #     return self.level < other.level
        # return self.cost < other.cost
def isOnMatrix(blank):
    return 0 <= blank[0] < N and 0 <= blank[1] < N
def display(matrix):
    print('\n'.join(['\t'.join([str(cell) for cell in j]) for j in matrix]))
def print_path(root):
    if root is None:
        return
    print_path(root.parent)
    print("-----")
    display(root.matrix)
    print("Command : ",root.command,"\nLevel : " ,root.level)
    print("-----")
    print()
def solve(matrix,blank):
    #up, right, down, left
    #pergerakan blank space
    # untuk row -1 berarti geser keatas,1 geser kebawah,0 diam

```



```

# untuk col -1 berarti geser kekiri,1 geser kekanan,0 diam
#[[up],[right],[down],[left]]-> dengan susunan [row,col]
move=[[-1,0],[0,1],[1,0],[0,-1]]
visited = set()
nodes = queue.PriorityQueue()
visited.add(tuple(np.reshape(matrix,16)))
totalnodes=1
# create root node
rootcost = cost(matrix) # root cost
root = Node(None,matrix, rootcost, blank, 0,"Posisi Awal")
nodes.put(root)
if(punyasolusi(matrix)):
    print("Susunan Matrix Awal: ")
    print("-----")
    display(matrix)
    print("-----")
    print()
    print("Nilai dari fungsi Kurang (i) + X pada posisi awal :",kurang(matrix))
    display_kurang(matrix)
    print("Puzzle Dapat Diselesaikan\n")
    start = time.time()
    while not nodes.empty() :
        current = nodes.get()
        if (current.cost) == 0:
            stop = time.time()
            print("Terdapat "+ str(current.level)+" Langkah Untuk Menyelesaikan Puzzle
dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)")
            print("")
            print("Berikut Langkah Menyelesaikan : ")
            print_path(current)
            #return current.level, stop - start, len(visited)
            print("Level Akhir: ",current.level)
            print("Waktu yang dibutuhkan: ",stop - start)
            print("Jumlah Simpul yang dibangkitkan:", totalnodes)
            print("-----FINISH-----")
            break
        for i in range(0,4): #Ada 4 move
            #posisi 16(blank) berdasarkan array arah gerak row dan col
            child_blank = [current.blank[0] +move[i][0],current.blank[1] + move[i][1]]
            dir=["Up","Right","Down","Left"]
            if isOnMatrix(child_blank):
                mat = copy.deepcopy(current.matrix)
                #swap element 16(blank) dengan elemen yang berada pada kiri atau kanan
                #atau atas atau bawah(sesuai arah gerak)
                temp = mat[child_blank[0]][child_blank[1]]
                mat[child_blank[0]][child_blank[1]] = 16
                mat[current.blank[0]][current.blank[1]] = temp

                if tuple(np.reshape(mat,16)) not in visited:
                    visited.add(tuple(np.reshape(mat,16)))

```

```

        totalnodes+=1
        child_cost = cost(mat)
        child = Node(current, mat, child_cost, child_blank, current.level +
1,dir[i])
        nodes.put(child)
    else:
        print("Susunan Matrix Awal: ")
        print("-----")
        display(matrix)
        print("-----")
        print()
        print("Nilai dari fungsi Kurang (i) + X pada posisi awal :",kurang(matrix))
        display_kurang(matrix)
        print("Puzzle Tidak Dapat Diselesaikan")
def bacaFile(namaFolder,namaFile):
    with open((namaFolder+"/"+namaFile+ str(".txt")), 'r') as f:
        matrix = [[int(i) for i in line.split(' ')] for line in f]
    return matrix
def writeFile(namaFolder,namaFile):
    with open("testcase/tc10.txt", 'w') as f:
        for i in range (N) :
            row=[0 for k in range(N)]
            for j in range(N):
                elemen=int(input("Elemen Matriks Baris ke- "+str(i+1)+" Kolom ke- " +
str(j+1)+ " : " ))
                if (j==3) :
                    f.write(str(elemen))
                else :
                    f.write(str(elemen)+' ')
            if(i!=3):
                f.write('\n')
        matrix=bacaFile("testcase", "tc10")
    return matrix
def main() :
    print("Cara Memasukkan Puzzle: ")
    print("1. From File")
    print("2. Input Manual")
    cara=int(input("Pilihlah cara memasukkan puzzle : "))
    if(cara==1):
        matrix=bacaFile("testcase",str(input("Masukkan Nama File (tanpa format (.txt)) :
")))
        #Jika ingin memasukkan file input dari folder lain
        #matrix=bacaFile(str(input("Masukkan Nama Folder: ")),str(input("Masukkan Nama File
(tanpa format (.txt)) : ")))
    if(cara==2):
        print("Anda akan menuliskan puzzle di file pilihan anda.")
        matrix = writeFile("testcase",str(input("Masukkan Nama File (tanpa format (.txt)) :
")))
        #Jika ingin memasukkan file input dari folder lain

```

```

        #matrix=bacaFile(str(input("Masukkan Nama Folder: ")),str(input("Masukkan Nama File
(tanpa format (.txt)) : ")))
        blank=getBlankIndex(matrix)
        solve(matrix,blank)
        lanjut=str(input("Apakah ingin menyelesaikan puzzle lainnya? Y/N : " ))
        if(lanjut=="Y"):
            main()
        else :
            print("Terimakasih telah bermain game 15-Puzzle ini")
main()

```

2.4 Penjelasan Fungsi-fungsi dan Kelas yang terdapat pada Program

1. Kelas Node

Kelas Node ini adalah kelas yang akan menyimpan setiap Node yang diperiksa. Class node ini memiliki beberapa atribut, yakni parent,matrix,cost,blank,level,command. Pada inisialisasi, setiap atribut akan di inisialisasi sesuai dengan parameter awalnya. Pada node ini juga akan membuat fungsi lt, yang berguna sebagai priority untuk masuk ke dalam queue(antrian). Pada program ini,yang menjadi prioritas adalah fungsi cost + level terkecil.

2. Fungsi getBlankIndex(matrix)

Fungsi ini memiliki parameter matrix (array 2d), dengan tujuan untuk mencari posisi blank pada puzzle, atau pada program ini digunakan "16" untuk menggambarkan blank pada puzzle. Fungsi ini akan mengembalikan posisi blank dengan [i,j], i berarti row blank dan j berarti column blank.

3. Fungsi kurang(matrix)

Fungsi ini memiliki parameter berupa matrix (array 2d) dan akan mengembalikan nilai sigma dari fungsi kurang(i) +X.

4. Fungsi display_kurang(matrix)

Fungsi ini memiliki parameter berupa matrix (array 2d) dan akan melakukan print fungsi kurang(i) untuk setiap i yang berada pada puzzle.

5. Fungsi cost(matrix)

Fungsi ini memiliki parameter berupa matrix (array 2d) dan akan menghitung cost dari puzzle, namun untuk blank tidak perlu dihitung. Fungsi cost sendiri adalah fungsi yang menunjukkan posisi ubin yang belum sesuai posisi yang diinginkan pada matrix final (hasil akhir),namun blank tidak perlu dihitung. Fungsi ini akan mereturn nilai dari fungsi cost dari suatu matrix (array 2d).

6. Fungsi punyasolusi(matrix)

Fungsi ini memiliki parameter matrix (array 2d) dan akan mengembalikan true jika fungsi kurang dari matrix (array 2d) tersebut merupakan bilangan genap dan false jika merupakan bilangan ganjil.

7. Fungsi isOnMatrix(blank)

Fungsi ini memiliki parameter berupa blank,dimana blank sendiri adalah array yang berisikan [row,col] yang menunjukkan posisi dari blank. Fungsi ini akan memeriksa apakah blank masih berada di dalam range row col dari matrix atau tidak (dalam kasus program ini adalah 4x4).Fungsi ini akan mereturn true jika blank masih berada di dalam matrix. Fungsi ini akan digunakan saat memeriksa apakah pergerakan masih valid (masih berada di dalam matrix).

8. Fungsi display(matrix)

Fungsi display memiliki parameter matrix (array 2d) fungsi ini akan melakukan print terhadap setiap elemen matrix.

9. Fungsi print_path(root)

Fungsi ini memiliki parameter berupa root dan akan melakukan print terhadap setiap matrix yang menjadi rute untuk ke solusi(goal). Pada fungsi ini juga akan ditampilkan level serta command yang dilakukan.

10. Fungsi solve (matrix,blank)

Fungsi ini memiliki parameter berupa matrix (array 2d) serta blank (array) fungsi ini akan menyelesaikan puzzle dengan cara branch and bound.

11. Fungsi bacaFile (namaFolder,namaFile)

Fungsi ini berparameter namaFile yang akan mengembalikan matrix yang tersimpan di dalam file yang memiliki nama adalah namaFile. Setiap line pada file tersebut akan dibaca dan di masukkan ke dalam matrix. Pada program ini juga disediakan jika ingin membaca file dari folder yang berbeda.

12. Fungsi writeFile(namaFolder,namaFile)

Fungsi ini berparameter namaFolder dan namaFile, fungsi ini akan membantu saat pengguna menuliskan puzzle secara manual ke dalam suatu file tertentu.

13. Fungsi main ()

Fungsi ini adalah fungsi utama yang akan menjalankan program dengan memanggil fungsi-fungsi diatas. Pada fungsi ini akan melakukan print-print untuk mempercantik tampilan. Pada Main ini, juga terdapat 2 pilihan cara menginputkan puzzle,yakni melalui file dan input manual. Selain itu terdapat pilihan untuk melakukan pencarian kembali.

2.5 Screenshoot Hasil Program

a. Input dari File : testcase/tc1.txt

```
Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc1
Susunan Matrix Awal:
-----
15      2      1      12
8        5      6      11
4        9     10      7
3       14     13     16
-----

Nilai dari fungsi Kurang (i) +X pada posisi awal : 45
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 1
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 1
Nilai dari fungsi Kurang(5): 2
Nilai dari fungsi Kurang(6): 2
Nilai dari fungsi Kurang(7): 1
Nilai dari fungsi Kurang(8): 5
Nilai dari fungsi Kurang(9): 2
Nilai dari fungsi Kurang(10): 2
Nilai dari fungsi Kurang(11): 5
Nilai dari fungsi Kurang(12): 9
Nilai dari fungsi Kurang(13): 0
Nilai dari fungsi Kurang(14): 1
Nilai dari fungsi Kurang(15): 14
Nilai dari fungsi Kurang(16): 0
Puzzle Tidak Dapat Diselesaikan
Apakah ingin menyelesaikan puzzle lainnya? Y/N : █
```

b. Input dari File : testcase/tc2.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc2
Susunan Matrix Awal:
-----
1      3      4      15
2      16     5      12
7      6      11     14
8      9      10     13
-----

Nilai dari fungsi Kurang (i) +X pada posisi awal : 37
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 1
Nilai dari fungsi Kurang(4): 1
Nilai dari fungsi Kurang(5): 0
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 1
Nilai dari fungsi Kurang(8): 0
Nilai dari fungsi Kurang(9): 0
Nilai dari fungsi Kurang(10): 0
Nilai dari fungsi Kurang(11): 3
Nilai dari fungsi Kurang(12): 6
Nilai dari fungsi Kurang(13): 0
Nilai dari fungsi Kurang(14): 4
Nilai dari fungsi Kurang(15): 11
Nilai dari fungsi Kurang(16): 10
Puzzle Tidak Dapat Diselesaikan
Apakah ingin menyelesaikan puzzle lainnya? Y/N : 

```

c. Input dari File : testcase/tc3.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc3
Susunan Matrix Awal:
-----
1      2      3      4
5      6      7      16
9      10     12     8
11     13     14     15
-----

Nilai dari fungsi Kurang (i) + X pada posisi awal : 12
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 0
Nilai dari fungsi Kurang(5): 0
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 0
Nilai dari fungsi Kurang(8): 0
Nilai dari fungsi Kurang(9): 1
Nilai dari fungsi Kurang(10): 1
Nilai dari fungsi Kurang(11): 0
Nilai dari fungsi Kurang(12): 2
Nilai dari fungsi Kurang(13): 0
Nilai dari fungsi Kurang(14): 0
Nilai dari fungsi Kurang(15): 0
Nilai dari fungsi Kurang(16): 8
Puzzle Dapat Diselesaikan

```

Terdapat 16 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)

Berikut Langkah Menyelesaikan :

```
-----  
1      2      3      4  
5      6      7      16  
9      10     12      8  
11     13     14     15
```

Command : Posisi Awal
Level : 0

```
-----  
1      2      3      4  
5      6      7      8  
9      10     12     16  
11     13     14     15
```

Command : Down
Level : 1

```
-----  
1      2      3      4  
5      6      7      8  
9      10     16     12  
11     13     14     15
```

Command : Left
Level : 2

```
-----  
1      2      3      4  
5      6      7      8  
9      10     14     12  
11     13     16     15
```

Command : Down
Level : 3

```
-----  
1      2      3      4  
5      6      7      8  
9      10     14     12  
11     16     13     15
```

Command : Left
Level : 4

```
-----  
1      2      3      4  
5      6      7      8  
9      10     14     12  
16     11     13     15
```

Command : Left
Level : 5

```
-----  
1      2      3      4  
5      6      7      8  
16     10     14     12  
9      11     13     15
```

Command : Up
Level : 6

```
-----  
1      2      3      4  
5      6      7      8  
10     16     14     12  
9      11     13     15
```

Command : Right
Level : 7

```
-----  
1      2      3      4  
5      6      7      8  
10     11     14     12  
9      16     13     15
```

Command : Down
Level : 8

```
-----  
1      2      3      4  
5      6      7      8  
10     11     14     12  
9      13     16     15  
Command : Right  
Level : 9  
-----  
  
-----  
1      2      3      4  
5      6      7      8  
10     11     16     12  
9      13     14     15  
Command : Up  
Level : 10  
-----  
  
-----  
1      2      3      4  
5      6      7      8  
10     16     11     12  
9      13     14     15  
Command : Left  
Level : 11  
-----
```

```

-----
1      2      3      4
5      6      7      8
16     10     11     12
9      13     14     15
Command : Left
Level : 12
-----

-----
1      2      3      4
5      6      7      8
9      10     11     12
16     13     14     15
Command : Down
Level : 13
-----

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     16     14     15
Command : Right
Level : 14
-----

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     16     15
Command : Right
Level : 15
-----

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16
Command : Right
Level : 16
-----

Level Akhir: 16
Waktu yang dibutuhkan: 0.15311288833618164
Jumlah Simpul yang dibangkitkan: 1508
-----FINISH-----
Apakah ingin menyelesaikan puzzle lainnya? Y/N : 

```

d. Input dari File : testcase/tc4.txt

Cara Memasukkan Puzzle:

1. From File

2. Input Manual

Pilihlah cara memasukkan puzzle : 1

Masukkan Nama File (tanpa format (.txt)) : tc4

Susunan Matrix Awal:

```
-----  
1      2      3      4  
5      6      16     8  
9      10     7      11  
13     14     15     12  
-----
```

Nilai dari fungsi Kurang (i) + X pada posisi awal : 16

Nilai dari fungsi Kurang(1): 0

Nilai dari fungsi Kurang(2): 0

Nilai dari fungsi Kurang(3): 0

Nilai dari fungsi Kurang(4): 0

Nilai dari fungsi Kurang(5): 0

Nilai dari fungsi Kurang(6): 0

Nilai dari fungsi Kurang(7): 0

Nilai dari fungsi Kurang(8): 1

Nilai dari fungsi Kurang(9): 1

Nilai dari fungsi Kurang(10): 1

Nilai dari fungsi Kurang(11): 0

Nilai dari fungsi Kurang(12): 0

Nilai dari fungsi Kurang(13): 1

Nilai dari fungsi Kurang(14): 1

Nilai dari fungsi Kurang(15): 1

Nilai dari fungsi Kurang(16): 9

Puzzle Dapat Diselesaikan

Terdapat 3 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)

Berikut Langkah Menyelesaikan :

```
-----  
1      2      3      4  
5      6      16     8  
9      10     7      11  
13     14     15     12  
-----
```

Command : Posisi Awal

Level : 0

```
-----  
1      2      3      4  
5      6      7      8  
9      10     16     11  
13     14     15     12  
-----
```

Command : Down

Level : 1

```
-----  
1      2      3      4  
5      6      7      8  
9      10     11     16  
13     14     15     12  
-----
```

Command : Right

Level : 2

```

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16
Command : Down
Level : 3
-----

Level Akhir: 3
Waktu yang dibutuhkan: 0.00113677978515625
Jumlah Simpul yang dibangkitkan: 10
-----FINISH-----
Apakah ingin menyelesaikan puzzle lainnya? Y/N : 

```

- e. Input dari File : testcase/tc5.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc5
Susunan Matrix Awal:
-----
1      2      3      4
5      6      16     12
9      10     8      7
13     14     11     15
-----

Nilai dari fungsi Kurang (i) + X pada posisi awal : 22
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 0
Nilai dari fungsi Kurang(5): 0
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 0
Nilai dari fungsi Kurang(8): 1
Nilai dari fungsi Kurang(9): 2
Nilai dari fungsi Kurang(10): 2
Nilai dari fungsi Kurang(11): 0
Nilai dari fungsi Kurang(12): 5
Nilai dari fungsi Kurang(13): 1
Nilai dari fungsi Kurang(14): 1
Nilai dari fungsi Kurang(15): 0
Nilai dari fungsi Kurang(16): 9
Puzzle Dapat Diselesaikan

```

Terdapat 7 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)

Berikut Langkah Menyelesaikan :

```
-----  
1      2      3      4  
5      6      16     12  
9      10     8      7  
13     14     11     15
```

Command : Posisi Awal

Level : 0

```
-----  
1      2      3      4  
5      6      8      12  
9      10     16     7  
13     14     11     15
```

Command : Down

Level : 1

```
-----  
1      2      3      4  
5      6      8      12  
9      10     7      16  
13     14     11     15
```

Command : Right

Level : 2

```
-----  
1      2      3      4  
5      6      8      16  
9      10     7      12  
13     14     11     15
```

Command : Up

Level : 3

```
-----  
1      2      3      4  
5      6      16     8  
9      10     7      12  
13     14     11     15
```

Command : Left

Level : 4

```
-----  
1      2      3      4  
5      6      7      8  
9      10     16     12  
13     14     11     15
```

Command : Down

Level : 5

```

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     16     15
Command : Down
Level : 6
-----

1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16
Command : Right
Level : 7
-----

Level Akhir: 7
Waktu yang dibutuhkan: 0.003448963165283203
Jumlah Simpul yang dibangkitkan: 32
-----FINISH-----
Apakah ingin menyelesaikan puzzle lainnya? Y/N : 

```

f. Input dari File : testcase/tc6.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc6
Susunan Matrix Awal:
-----
1      2      3      4
5      6      7      8
16     13     9      10
11     12     15     14
-----

Nilai dari fungsi Kurang (i) + X pada posisi awal : 12
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 0
Nilai dari fungsi Kurang(5): 0
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 0
Nilai dari fungsi Kurang(8): 0
Nilai dari fungsi Kurang(9): 0
Nilai dari fungsi Kurang(10): 0
Nilai dari fungsi Kurang(11): 0
Nilai dari fungsi Kurang(12): 0
Nilai dari fungsi Kurang(13): 4
Nilai dari fungsi Kurang(14): 0
Nilai dari fungsi Kurang(15): 1
Nilai dari fungsi Kurang(16): 7
Puzzle Dapat Diselesaikan

```

Terdapat 22 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)

Berikut Langkah Menyelesaikan :

```
-----  
1      2      3      4  
5      6      7      8  
16     13     9      10  
11     12     15     14  
Command : Posisi Awal  
Level : 0  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
13     16     9      10  
11     12     15     14  
Command : Right  
Level : 1  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
13     9      16     10  
11     12     15     14  
Command : Right  
Level : 2  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
13     9      15     10  
11     12     16     14  
Command : Down  
Level : 3  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
13     9      15     10  
11     12     14     16  
Command : Right  
Level : 4  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
13     9      15     16  
11     12     14     10  
Command : Up  
Level : 5  
-----
```

-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	16	15	13	9	14	15
11	12	14	10	16	11	12	10
Command : Left				Command : Left			
Level : 6				Level : 9			
-----				-----			
-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	14	15	16	9	14	15
11	12	16	10	13	11	12	10
Command : Down				Command : Up			
Level : 7				Level : 10			
-----				-----			
-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	14	15	9	16	14	15
11	16	12	10	13	11	12	10
Command : Left				Command : Right			
Level : 8				Level : 11			
-----				-----			

```

-----
1      2      3      4
5      6      7      8
9      14     16     15
13     11     12     10
Command : Right
Level : 12
-----

-----
1      2      3      4
5      6      7      8
9      14     12     15
13     11     16     10
Command : Down
Level : 13
-----

-----
1      2      3      4
5      6      7      8
9      14     12     15
13     11     10     16
Command : Right
Level : 14
-----

-----
1      2      3      4
5      6      7      8
9      14     12     16
13     11     10     15
Command : Up
Level : 15
-----

-----
1      2      3      4
5      6      7      8
9      14     16     12
13     11     10     15
Command : Left
Level : 16
-----

-----
1      2      3      4
5      6      7      8
9      14     10     12
13     11     16     15
Command : Down
Level : 17
-----

```

```
-----  
1      2      3      4  
5      6      7      8  
9      14     10     12  
13     16     11     15
```

```
Command : Left  
Level : 18  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
9      16     10     12  
13     14     11     15
```

```
Command : Up  
Level : 19  
-----
```

```
-----  
1      2      3      4  
5      6      7      8  
9      10     16     12  
13     14     11     15
```

```
Command : Right  
Level : 20  
-----
```



```

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     16     15
Command : Down
Level : 21
-----

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16
Command : Right
Level : 22
-----

Level Akhir: 22
Waktu yang dibutuhkan: 2.3222756385803223
Jumlah Simpul yang dibangkitkan: 26576
-----FINISH-----
Apakah ingin menyelesaikan puzzle lainnya? Y/N : █

```

g. Input dari File : testcase/tc7.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc7
Susunan Matrix Awal:
-----
1      2      5      4
3      6      16     7
9      10     8      11
13     14     15     12
-----

Nilai dari fungsi Kurang (i) + X pada posisi awal : 18
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 1
Nilai dari fungsi Kurang(5): 2
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 0
Nilai dari fungsi Kurang(8): 0
Nilai dari fungsi Kurang(9): 1
Nilai dari fungsi Kurang(10): 1
Nilai dari fungsi Kurang(11): 0
Nilai dari fungsi Kurang(12): 0
Nilai dari fungsi Kurang(13): 1
Nilai dari fungsi Kurang(14): 1
Nilai dari fungsi Kurang(15): 1
Nilai dari fungsi Kurang(16): 9
Puzzle Dapat Diselesaikan

```

Berikut Langkah Menyelesaikan :

```
-----  
1      2      5      4  
3      6      16     7  
9      10     8      11  
13     14     15     12
```

Command : Posisi Awal

Level : 0

```
-----  
1      2      16     4  
3      6      5      7  
9      10     8      11  
13     14     15     12
```

Command : Up

Level : 1

```
-----  
1      2      4      16  
3      6      5      7  
9      10     8      11  
13     14     15     12
```

Command : Right

Level : 2

```
-----  
1      2      4      7  
3      6      5      16  
9      10     8      11  
13     14     15     12
```

Command : Down

Level : 3

```
-----  
1      2      4      7  
3      6      5      11  
9      10     8      16  
13     14     15     12
```

Command : Down

Level : 4

```
-----  
1      2      4      7  
3      6      5      11  
9      10     16     8  
13     14     15     12
```

Command : Left

Level : 5

1	2	4	7
3	6	5	11
9	16	10	8
13	14	15	12
Command : Left			
Level : 6			

1	2	4	7
3	16	5	11
9	6	10	8
13	14	15	12
Command : Up			
Level : 7			

1	2	4	7
16	3	5	11
9	6	10	8
13	14	15	12
Command : Left			
Level : 8			

16	2	4	7
1	3	5	11
9	6	10	8
13	14	15	12
Command : Up			
Level : 9			

2	16	4	7
1	3	5	11
9	6	10	8
13	14	15	12
Command : Right			
Level : 10			

2	3	4	7
1	16	5	11
9	6	10	8
13	14	15	12
Command : Down			
Level : 11			

```

-----
2      3      4      7
1      5      16     11
9      6      10     8
13     14     15     12
Command : Right
Level : 12
-----

-----
2      3      4      7
1      5      11     16
9      6      10     8
13     14     15     12
Command : Right
Level : 13
-----

-----
2      3      4      16
1      5      11     7
9      6      10     8
13     14     15     12
Command : Up
Level : 14
-----

```

```

-----
2      3      16     4
1      5      11     7
9      6      10     8
13     14     15     12
Command : Left
Level : 15
-----

-----
2      16     3      4
1      5      11     7
9      6      10     8
13     14     15     12
Command : Left
Level : 16
-----

-----
16     2      3      4
1      5      11     7
9      6      10     8
13     14     15     12
Command : Left
Level : 17
-----

```

1	2	3	4
16	5	11	7
9	6	10	8
13	14	15	12
Command : Down			
Level : 18			

1	2	3	4
5	16	11	7
9	6	10	8
13	14	15	12
Command : Right			
Level : 19			

1	2	3	4
5	6	11	7
9	16	10	8
13	14	15	12
Command : Down			
Level : 20			

1	2	3	4
5	6	11	7
9	10	16	8
13	14	15	12
Command : Right			
Level : 21			

1	2	3	4
5	6	16	7
9	10	11	8
13	14	15	12
Command : Up			
Level : 22			

1	2	3	4
5	6	7	16
9	10	11	8
13	14	15	12
Command : Right			
Level : 23			

```

-----
1      2      3      4
5      6      7      8
9      10     11     16
13     14     15     12
Command : Down
Level : 24
-----

1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16
Command : Down
Level : 25
-----

Level Akhir: 25
Waktu yang dibutuhkan: 29.8370304107666
Jumlah Simpul yang dibangkitkan: 283816
-----FINISH-----
Apakah ingin menyelesaikan puzzle lainnya? Y/N : █

```

h. Input dari File : testcase/tc8.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc8
Susunan Matrix Awal:
-----
2      3      15     4
1      5      8      16
13     7      6      14
10     9      12     11
-----

Nilai dari fungsi Kurang (i) + X pada posisi awal : 38
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 1
Nilai dari fungsi Kurang(3): 1
Nilai dari fungsi Kurang(4): 1
Nilai dari fungsi Kurang(5): 0
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 1
Nilai dari fungsi Kurang(8): 2
Nilai dari fungsi Kurang(9): 0
Nilai dari fungsi Kurang(10): 1
Nilai dari fungsi Kurang(11): 0
Nilai dari fungsi Kurang(12): 1
Nilai dari fungsi Kurang(13): 6
Nilai dari fungsi Kurang(14): 4
Nilai dari fungsi Kurang(15): 12
Nilai dari fungsi Kurang(16): 8
Puzzle Dapat Diselesaikan

```

Terdapat 24 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)

Berikut Langkah Menyelesaikan :

```
-----  
2      3      15     4  
1      5      8      16  
13     7      6      14  
10     9      12     11
```

Command : Posisi Awal

Level : 0

```
-----  
2      3      15     4  
1      5      16     8  
13     7      6      14  
10     9      12     11
```

Command : Left

Level : 1

```
-----  
2      3      16     4  
1      5      15     8  
13     7      6      14  
10     9      12     11
```

Command : Up

Level : 2

2	16	3	4	1	2	3	4
1	5	15	8	5	16	15	8
13	7	6	14	13	7	6	14
10	9	12	11	10	9	12	11
Command : Left				Command : Right			
Level : 3				Level : 6			
16	2	3	4	1	2	3	4
1	5	15	8	5	7	15	8
13	7	6	14	13	16	6	14
10	9	12	11	10	9	12	11
Command : Left				Command : Down			
Level : 4				Level : 7			
1	2	3	4	1	2	3	4
16	5	15	8	5	7	15	8
13	7	6	14	13	6	16	14
10	9	12	11	10	9	12	11
Command : Down				Command : Right			
Level : 5				Level : 8			


```

-----
1      2      3      4
5      7      15     8
13     6      14     16
10     9      12     11
Command : Right
Level : 9
-----

-----
1      2      3      4
5      7      15     8
13     6      14     11
10     9      12     16
Command : Down
Level : 10
-----

-----
1      2      3      4
5      7      15     8
13     6      14     11
10     9      16     12
Command : Left
Level : 11
-----

-----
1      2      3      4
5      7      15     8
13     6      16     11
10     9      14     12
Command : Up
Level : 12
-----

-----
1      2      3      4
5      7      16     8
13     6      15     11
10     9      14     12
Command : Up
Level : 13
-----

-----
1      2      3      4
5      7      15     8
13     6      14     11
10     9      16     12
Command : Left
Level : 14
-----

```

-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	16	15	11	16	9	15	11
10	9	14	12	13	10	14	12
Command : Down				Command : Up			
Level : 15				Level : 18			
-----				-----			
-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	15	11	9	16	15	11
10	16	14	12	13	10	14	12
Command : Down				Command : Right			
Level : 16				Level : 19			
-----				-----			
-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	15	11	9	10	15	11
16	10	14	12	13	16	14	12
Command : Left				Command : Down			
Level : 17				Level : 20			
-----				-----			

```

-----
1      2      3      4
5      6      7      8
9      10     15     11
13     14     16     12
Command : Right
Level : 21
-----

-----
1      2      3      4
5      6      7      8
9      10     16     11
13     14     15     12
Command : Up
Level : 22
-----

-----
1      2      3      4
5      6      7      8
9      10     11     16
13     14     15     12
Command : Right
Level : 23
-----

```

```

-----
1      2      3      4
5      6      7      8
9      10     11     12
13     14     15     16
Command : Down
Level : 24
-----

Level Akhir: 24
Waktu yang dibutuhkan: 2.7331361770629883
Jumlah Simpul yang dibangkitkan: 28021
-----FINISH-----
Apakah ingin menyelesaikan puzzle lainnya? Y/N : 

```

- i. Input dari File : testcase/tc9.txt

```

Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 1
Masukkan Nama File (tanpa format (.txt)) : tc9
Susunan Matrix Awal:
-----
16     1      2      3
5      6      7      8
4      9      10     11
13     14     15     12
-----

Nilai dari fungsi Kurang (i) + X pada posisi awal : 22
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 0
Nilai dari fungsi Kurang(5): 1
Nilai dari fungsi Kurang(6): 1
Nilai dari fungsi Kurang(7): 1
Nilai dari fungsi Kurang(8): 1
Nilai dari fungsi Kurang(9): 0
Nilai dari fungsi Kurang(10): 0
Nilai dari fungsi Kurang(11): 0
Nilai dari fungsi Kurang(12): 0
Nilai dari fungsi Kurang(13): 1
Nilai dari fungsi Kurang(14): 1
Nilai dari fungsi Kurang(15): 1
Nilai dari fungsi Kurang(16): 15
Puzzle Dapat Diselesaikan

```

```

Terdapat 20 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)
Berikut Langkah Menyelesaikan :
-----
16      1      2      3
5       6      7      8
4       9     10     11
13      14     15     12
Command : Posisi Awal
Level : 0
-----

5       1      2      3
16      6      7      8
4       9     10     11
13      14     15     12
Command : Down
Level : 1
-----

5       1      2      3
4       6      7      8
16      9     10     11
13      14     15     12
Command : Down
Level : 2
-----

```

5	1	2	3	16	1	2	3
4	6	7	8	5	4	7	8
9	16	10	11	9	6	10	11
13	14	15	12	13	14	15	12
Command : Right				Command : Up			
Level : 3				Level : 6			

5	1	2	3	1	16	2	3
4	16	7	8	5	4	7	8
9	6	10	11	9	6	10	11
13	14	15	12	13	14	15	12
Command : Up				Command : Right			
Level : 4				Level : 7			

5	1	2	3	1	2	16	3
16	4	7	8	5	4	7	8
9	6	10	11	9	6	10	11
13	14	15	12	13	14	15	12
Command : Left				Command : Right			
Level : 5				Level : 8			

1	2	7	3
5	4	16	8
9	6	10	11
13	14	15	12
Command : Down			
Level : 9			

1	2	7	3
5	16	4	8
9	6	10	11
13	14	15	12
Command : Left			
Level : 10			

1	2	7	3
5	6	4	8
9	16	10	11
13	14	15	12
Command : Down			
Level : 11			

1	2	7	3
5	6	4	8
9	10	16	11
13	14	15	12
Command : Right			
Level : 12			

1	2	7	3
5	6	4	8
9	10	11	16
13	14	15	12
Command : Right			
Level : 13			

1	2	7	3
5	6	4	16
9	10	11	8
13	14	15	12
Command : Up			
Level : 14			

```

-----
1      2      7      3
5      6      16     4
9      10     11     8
13     14     15     12
Command : Left
Level : 15
-----

-----
1      2      16     3
5      6      7      4
9      10     11     8
13     14     15     12
Command : Up
Level : 16
-----

-----
1      2      3      16
5      6      7      4
9      10     11     8
13     14     15     12
Command : Right
Level : 17
-----

```



```
-----  
1      2      3      4  
5      6      7      16  
9      10     11      8  
13     14     15     12
```

Command : Down

Level : 18

```
-----  
1      2      3      4  
5      6      7      8  
9      10     11     16  
13     14     15     12
```

Command : Down

Level : 19

```
-----  
1      2      3      4  
5      6      7      8  
9      10     11     12  
13     14     15     16
```

Command : Down

Level : 20

Level Akhir: 20

Waktu yang dibutuhkan: 0.49399304389953613

Jumlah Simpul yang dibangkitkan: 5963

-----FINISH-----

Apakah ingin menyelesaikan puzzle lainnya? Y/N : ☐

- j. Input manual dengan menuliskan pada file : testcase/tc10

```
Cara Memasukkan Puzzle:
1. From File
2. Input Manual
Pilihlah cara memasukkan puzzle : 2
Anda akan menuliskan puzzle di file pilihan anda
Masukkan Nama File (tanpa format (.txt)) : tc10
Elemen Matriks Baris ke- 1 Kolom ke-1 : 1
Elemen Matriks Baris ke- 1 Kolom ke-2 : 2
Elemen Matriks Baris ke- 1 Kolom ke-3 : 3
Elemen Matriks Baris ke- 1 Kolom ke-4 : 4
Elemen Matriks Baris ke- 2 Kolom ke-1 : 5
Elemen Matriks Baris ke- 2 Kolom ke-2 : 8
Elemen Matriks Baris ke- 2 Kolom ke-3 : 9
Elemen Matriks Baris ke- 2 Kolom ke-4 : 10
Elemen Matriks Baris ke- 3 Kolom ke-1 : 11
Elemen Matriks Baris ke- 3 Kolom ke-2 : 6
Elemen Matriks Baris ke- 3 Kolom ke-3 : 7
Elemen Matriks Baris ke- 3 Kolom ke-4 : 12
Elemen Matriks Baris ke- 4 Kolom ke-1 : 13
Elemen Matriks Baris ke- 4 Kolom ke-2 : 14
Elemen Matriks Baris ke- 4 Kolom ke-3 : 15
Elemen Matriks Baris ke- 4 Kolom ke-4 : 16
Susunan Matrix Awal:
-----
1      2      3      4
5      8      9      10
11     6      7      12
13     14     15     16
-----
```

```
Nilai dari fungsi Kurang (i) + X pada posisi awal : 8
Nilai dari fungsi Kurang(1): 0
Nilai dari fungsi Kurang(2): 0
Nilai dari fungsi Kurang(3): 0
Nilai dari fungsi Kurang(4): 0
Nilai dari fungsi Kurang(5): 0
Nilai dari fungsi Kurang(6): 0
Nilai dari fungsi Kurang(7): 0
Nilai dari fungsi Kurang(8): 2
Nilai dari fungsi Kurang(9): 2
Nilai dari fungsi Kurang(10): 2
Nilai dari fungsi Kurang(11): 2
Nilai dari fungsi Kurang(12): 0
Nilai dari fungsi Kurang(13): 0
Nilai dari fungsi Kurang(14): 0
Nilai dari fungsi Kurang(15): 0
Nilai dari fungsi Kurang(16): 0
Puzzle Dapat Diselesaikan

Terdapat 24 Langkah Untuk Menyelesaikan Puzzle dari Posisi Awal(Posisi Awal tidak dihitung sebagai langkah)
```

Berikut Langkah Menyelesaikan

```
-----  
1      2      3      4  
5      8      9      10  
11     6      7      12  
13     14     15     16
```

Command : Posisi Awal
Level : 0

```
-----  
1      2      3      4  
5      8      9      10  
11     6      7      16  
13     14     15     12
```

Command : Up
Level : 1

```
-----  
1      2      3      4  
5      8      9      10  
11     6      16     7  
13     14     15     12
```

Command : Left
Level : 2

```
-----  
1      2      3      4  
5      8      16     10  
11     6      9      7  
13     14     15     12
```

Command : Up
Level : 3

```
-----  
1      2      3      4  
5      16     8      10  
11     6      9      7  
13     14     15     12
```

Command : Left
Level : 4

```
-----  
1      2      3      4  
5      6      8      10  
11     16     9      7  
13     14     15     12
```

Command : Down
Level : 5

```

-----
1      2      3      4
5      6      8      10
16     11     9      7
13     14     15     12
Command : Left
Level : 6
-----

-----
1      2      3      4
5      6      8      10
13     11     9      7
16     14     15     12
Command : Down
Level : 7
-----

-----
1      2      3      4
5      6      8      10
13     11     9      7
14     16     15     12
Command : Right
Level : 8
-----

-----
1      2      3      4
5      6      8      10
13     16     9      7
14     11     15     12
Command : Up
Level : 9
-----

-----
1      2      3      4
5      6      8      10
13     9      16     7
14     11     15     12
Command : Right
Level : 10
-----

-----
1      2      3      4
5      6      8      10
13     9      7      16
14     11     15     12
Command : Right
Level : 11
-----

```

```

-----
1      2      3      4
5      6      8      16
13     9      7      10
14     11     15     12
Command : Up
Level : 12
-----

-----
1      2      3      4
5      6      16     8
13     9      7      10
14     11     15     12
Command : Left
Level : 13
-----

-----
1      2      3      4
5      6      7      8
13     9      16     10
14     11     15     12
Command : Down
Level : 14
-----

-----
1      2      3      4
5      6      7      8
13     9      10     16
14     11     15     12
Command : Right
Level : 15
-----

-----
1      2      3      4
5      6      7      8
13     9      10     12
14     11     15     16
Command : Down
Level : 16
-----

-----
1      2      3      4
5      6      7      8
13     9      16     10
14     11     15     12
Command : Left
Level : 17
-----

```

-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	10	12	9	16	10	12
14	16	11	15	13	14	11	15
Command : Left				Command : Right			
Level : 18				Level : 21			
-----				-----			
-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
13	9	10	12	9	10	16	12
16	14	11	15	13	14	11	15
Command : Left				Command : Right			
Level : 19				Level : 22			
-----				-----			
-----				-----			
1	2	3	4	1	2	3	4
5	6	7	8	5	6	7	8
16	9	10	12	9	10	11	12
13	14	11	15	13	14	16	15
Command : Up				Command : Down			
Level : 20				Level : 23			
-----				-----			

```
-----  
1      2      3      4  
5      6      7      8  
9      10     11     12  
13     14     15     16
```

Command : Right

Level : 24

```
-----  
Level Akhir: 24
```

Waktu yang dibutuhkan: 7.799945831298828

Jumlah Simpul yang dibangkitkan: 90690

```
-----FINISH-----
```

Apakah ingin menyelesaikan puzzle lainnya? Y/N :

BAB III

Penutup

3.1 Saran

Tidak ada saran untuk tugas kecil 3 ini, spesifikasi sudah cukup, waktu pengerjaan sudah lebih dari cukup. Terimakasih saya ucapkan kepada tim pengajar, yakni dosen dan juga para asisten yang sudah bekerja dengan sangat baik.

3.2 Kesimpulan

Algoritma ini cukup efektif dalam menyelesaikan 15-puzzle. Hanya saja diperlukan ide yang lebih efisien saat melakukan pencocokan antara state yang sudah diperiksa (yang berada di visited) dengan state yang akan diperiksa agar dapat menghemat waktu program, mungkin bisa saja dengan melakukan konversi dari matrix menjadi integer yang di ubah ke dalam bentuk 4 bit (nilai 1 elemen kurang dari 16). Namun ide tersebut belum sempat untuk direalisasikan untuk mempercepat proses pencarian. Namun pada program ini sudah dilakukan upaya optimasi pencarian dengan menggunakan set() yang merupakan bawaan dari python untuk melakukan pencarian.

DAFTAR PUSTAKA

“Algoritma Branch and Bound.” Informatika, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian1.pdf>. Accessed 28 Maret 2022.

“Algoritma Branch and Bound.” Informatika, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian2.pdf>. Accessed 28 Maret 2022.

“Algoritma Branch and Bound.” Informatika, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian3.pdf>. Accessed 28 Maret 2022.

“Algoritma Branch and Bound.” Informatika, <https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2020-2021/Algoritma-Branch-and-Bound-2021-Bagian4.pdf>. Accessed 28 Maret 2022.

Lampiran

Link Repository : https://github.com/StevenSiahaann/Tucil3_13520145

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