

Penyelesaian Persoalan 15-Puzzle dengan Algoritma *Branch and Bound*

LAPORAN TUGAS KECIL 3

Diajukan Untuk Memenuhi Tugas Kecil IF2211 Strategi Algoritma

Semester II tahun 2021/2022

Disusun oleh

Patrick Amadeus Irawan (13520109)



TEKNIK INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2022

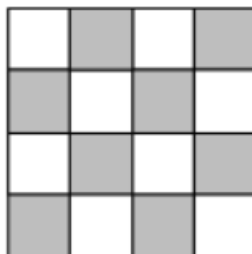
BAB I

Penjelasan Algoritma Program

Algoritma *Branch and Bound* merupakan jenis algoritma yang digunakan untuk mencari solusi optimal untuk persoalan berbasis kombinatorial, diskrit, dan lainnya dengan memanfaatkan optimisasi secara matematis. Optimisasi ini dilakukan dengan memanfaatkan fungsi objektif tertentu yang tidak melanggar *constraint* persoalan dimana fungsi ini membangkitkan simpul dengan pohon ruang status untuk menentukan batas (*bound*) dan membandingkan nilai solusi terbaik.

Persoalan N-Puzzle (dalam implementasi Tugas Kecil kali ini 15-Puzzle) merupakan salah satu persoalan yang dapat diselesaikan dengan algoritma ini. Berikut deskripsi langkah-langkah dari implementasi algoritma untuk menemukan solusi puzzle :

1. Mencari penjumlahan dari **Kurang(i)** untuk $i = 1, 2, 3, \dots, 15$. Dimana **Kurang(i)** didefinisikan sebagai banyaknya ubin bernomor j sedemikian sehingga $j < i$ dan **Posisi(j) > Posisi(i)**. **Posisi(i)** sendiri diartikan sebagai posisi ubin bernomor i pada susunan yang diperiksa.
2. Menambahkan hasil dari tahap (1) dengan :



- a. 1, apabila sel kosong pada posisi awal berada pada sel yang diarsir
 - b. 0, apabila sel kosong pada posisi awal berada pada sel yang tidak diarsir
3. Dari hasil yang didapatkan pada tahap (2) terdapat dua kemungkinan :
 - a. Apabila hasil ganjil, persoalan 15-Puzzle bersangkutan tidak dapat diselesaikan
 - b. Apabila hasil genap, persoalan 15-Puzzle bersangkutan dapat diselesaikan dan lanjut ke tahap (4)
4. Masukkan persoalan ke dalam antrian prioritas Q :
 - a. Jika simpul akar merupakan solusi, persoalan selesai.
 - b. Jika bukan merupakan solusi, lanjut ke tahap (5)

5. Masukkan setiap anak simpul (berdasarkan arah pergeseran ubin yang memungkinkan) ke dalam antrian prioritas Q sesuai dengan **cost** masing-masing. **Cost** disini mengikuti rumus

$$c(P) = f(P) + g(P)$$

$c(P)$ = ongkos untuk simpul P

$f(P)$ = panjang lintasan dari simpul akar ke simpul P

$g(P)$ = taksiran panjang lintasan terpendek dari P ke simpul solusi pada upapohon yang akarnya adalah P

Taksiran untuk $g(P)$ adalah jumlah ubin tidak kosong yang tidak terdapat pada susunan simpul solusi.

6. Kembali ke tahap 4.

BAB II

Implementasi Program

Bahasa Pemrograman : Python

src.py (function logik utama pembangun solusi)

```
1 import time
2 from copy import deepcopy as dc
3 from datetime import datetime as dt
4 import os
5 import heapq as hq
6
7 def readFile(filename) -> list:
8     # I.S. filename terdefinisi, format benar
9     # F.S. mengembalikan representasi struktur data matrix dari bacaan file
10     with open(filename) as f:
11         return [list(map(int, line.strip('\n').split())) for line in f]
12
13 def printMatrix(m) -> None:
14     # I.S. m terdefinisi
15     # F.S. menampilkan matrix m
16     print("-"*21)
17     for i in m:
18         print("|",end="")
19         for j in i:
20             if j == 0:
21                 print(" ",end="|")
22             else:
23                 if j < 10:
24                     print(" 0%d " % j, end="|")
25                 else:
26                     print(" %d " % j, end="|")
27         print()
28     print("-"*5 + "+" + "-"*4 + "+" + "-"*4 + "+" + "-"*5)
29
30 def move(m, x,y, dirs):
31     # I.S. m, x, y, dirs terdefinisi
32     # F.S. mengembalikan matrix m yang telah ditempuh dengan perpindahan dari (x,y) terhadap arah dirs
33     if dirs == "u": # up
34         m[x][y], m[x-1][y] = m[x-1][y], m[x][y]
35     elif dirs == "r":
36         m[x][y], m[x][y+1] = m[x][y+1], m[x][y]
37     elif dirs == "d":
38         m[x][y], m[x+1][y] = m[x+1][y], m[x][y]
39     else:
40         m[x][y], m[x][y-1] = m[x][y-1], m[x][y]
41     return m
42
43 def moveType(x,y):
44     # I.S. x dan y terdefinisi
45     # F.S. mengembalikan string yang menunjukkan arah perpindahan yang memungkinkan dari koordinat (x,y)
46     if x > 0 and x < 3:
47         if y == 0:
48             return "urd"
49         elif y == 3:
50             return "uld"
51         else:
52             return "urd1"
53     elif x == 0:
54         if y == 0:
55             return "rd"
56         elif y == 3:
57             return "ld"
58         else:
59             return "rd1"
60     else:
61         if y == 0:
62             return "ur"
63         elif y == 3:
64             return "ul"
65         else:
66             return "ur1"
```

```

1 def posisi(num,m) -> int:
2     # I.S. num dan m terdefinisi
3     # F.S. mengembalikan posisi dari num pada m, berupa angka dari 0..15
4     pos = 0
5     for i in m:
6         for j in i:
7             if j == num:
8                 return pos
9             pos += 1
10
11 def koordinat(m, num) -> (int,int):
12     # I.S. m terdefinisi, num terdefinisi
13     # F.S. mengembalikan koordinat (i,j) dari num
14     for i in range(len(m)):
15         for j in range(len(m)):
16             if m[i][j] == num:
17                 return i,j
18
19 def getX(m) -> int:
20     # I.S. m terdefinisi, selalu mengandung 0
21     # F.S. mengembalikan 1 apabila posisi 0 di matrix m (i,j) memenuhi (i+j) mod 2 != 0
22     for i in range(len(m)):
23         for j in range(len(m)):
24             if m[i][j] == 0:
25                 return int((i+j) % 2 != 0)
26
27
28 def getCost(m):
29     # I.S. m terdefinisi
30     # F.S. mengembalikan nilai cost dari m, cost didapatkan dengan pendekatan banyak ubin non-kosong
31     # yang berada pada susunan solusi akhir
32     correct_num = 1
33     count = 0
34     for i in m:
35         for j in i:
36             if j != correct_num and j != 0:
37                 count += 1
38             correct_num += 1
39     return count
40
41
42 def isSolution(m):
43     # I.S. m terdefinisi
44     # F.S. mengembalikan True apabila m merupakan solusi, False apabila tidak
45     num = 1
46     for i in m:
47         for j in i:
48             if j != num and j != 0:
49                 return False
50             num += 1
51     return True

```

```

1 def Kurang(m,i):
2 # I.S. m dan i terdefinisi
3 # F.S. mengembalikan nilai KURANG(i), banyaknya ubin bernomor j
4 # sedemikian sehingga j < i dan POSISI(j) > POSISI(i).
5 count = 0
6 for arr in m:
7     for j in arr:
8         if j == 0:
9             j = 16
10            if i == 0:
11                if j != i and j < 16 and posisi(j,m) > posisi(i,m):
12                    count += 1
13            else:
14                if j!=i and j < i and posisi(j,m) > posisi(i,m):
15                    count += 1
16        return count
17
18 def sigmaKurang(m):
19 # I.S. m terdefinisi
20 # F.S. mengembalikan nilai total KURANG(i) untuk i pada range [0..15]
21 sigma = 0
22 for i in range(16):
23     sigma += Kurang(m,i)
24 return sigma
25
26
27 def getX(m):
28 # I.S. m terdefinisi, selalu mengandung 0
29 # F.S. mengembalikan True apabila (i+j) mod 2 != 0 , sebaliknya False
30 for i in range(len(m)):
31     for j in range(len(m)):
32         if m[i][j] == 0:
33             return int((i+j) % 2 != 0)
34
35
36 def isSolvable(m):
37 # I.S. m terdefinisi
38 # F.S. mengembalikan True apabila puzzle terdefinisi memiliki solusi, False apabila tidak
39 return (sigmaKurang(m) + getX(m)) % 2 == 0
40
41 def printInitInfo(m):
42 # I.S. m terdefinisi
43 # F.S. menampilkan informasi awal, mengenai solvable / tidak
44 print("0 refers to block 16")
45 for i in range(16):
46     print(i,":",Kurang(m, i))
47     print("X :",getX(m))
48     print("-----")
49     print("SIGMA KURANG + X = ", sigmaKurang(m) + getX(m))
50     print()
51
52
53 def gatherSolvedPath(m,routes):
54 # I.S. m terdefinisi, routes terdefinisi yang merupakan map dari pasangan (child , parent) dari tiap puzzle nodes
55 # F.S. mengembalikan list of puzzle yang merupakan rute solusi unik yang tercapai
56 result = [(m,'None')]
57 temp = routes[str(m)]
58 while str(temp[0]) != 'root':
59     result.append(temp)
60     temp = routes[str(temp[0])]
61 return result
62
63
64 def printSolvedPath(result):
65 # I.S. result terdefinisi, berupa list berisi puzzle yang merupakan rute solusi unik yang tercapai
66 # F.S. mencetak rute solusi unik yang tercapai
67 for idx in range(len(result)-1,-1,-1):
68     printMatrix(result[idx][0])
69     if idx == 0:
70         print("PROBLEM SOLVED!!")
71     else:
72         move = ''
73         if result[idx][1] == 'u':
74             move = 'DOWN'
75         elif result[idx][1] == 'd':
76             move = 'UP'
77         elif result[idx][1] == 'r':
78             move = 'LEFT'
79         else:
80             move = 'RIGHT'
81         print("STEP {}: MOVE {}".format(len(result)- idx , move))
82

```

gui.py (visualisasi graphical user interface dengan pemanfaatan lib **pygame**)

```
1 import pygame as pg, sys, os
2 import time
3
4 def squash_array(arr):
5     res = []
6     for i in arr:
7         for j in i:
8             res.append(j)
9     return res
10
11 class SlidePuzzle:
12     def __init__(self, grid, tile_size, margin):
13         self.grid, self.tile_size, self.margin = grid, tile_size, margin
14         self.len_tile = grid*grid
15         self.tile = [(x,y) for y in range(grid) for x in range(grid)]
16         self.tile_pos = {(x,y) : (x*(tile_size + margin) + margin, y * (tile_size + margin) + margin)
17                           for y in range(grid) for x in range(grid)}
18         self.font = pg.font.Font(None, 120)
19
20     def draw(self, screen, arr):
21         for i in range(self.len_tile):
22             x, y = self.tile_pos[self.tile[i]]
23             bg = (204, 228, 240)
24             if arr[i] == 0:
25                 bg = (232, 180, 116)
26             pg.draw.rect(screen, bg, (x, y, self.tile_size, self.tile_size))
27             font = (0, 0, 0)
28             if arr[i] == 0:
29                 font = (232, 180, 116)
30             text = self.font.render(str(arr[i]), 2, font)
31             screen.blit(text, (x, y))
32
33
34 def gui(sol):
35     pg.init()
36     os.environ['SDL_VIDEO_CENTERED'] = '1'
37     pg.display.set_caption('15 Puzzle')
38     screen = pg.display.set_mode((450, 450))
39     fpsclock = pg.time.Clock()
40     program = SlidePuzzle(4, 100, 5)
41
42     idx = 0
43
44     timer = pg.USEREVENT + 1
45     arr = squash_array(sol[idx])
46     pg.time.set_timer(timer, 2000)
47     while True:
48         for event in pg.event.get():
49             if event.type == pg.QUIT : pg.quit(); sys.exit()
50
51             if event.type == timer and idx < len(sol):
52                 arr = squash_array(sol[idx])
53                 idx += 1
54             screen.fill((232, 180, 116))
55             program.draw(screen, arr)
56             pg.display.flip()
57
```

main.py (program driver utama)

```
1  from src import *
2  from gui import *
3
4  # ===== FILENAME VALIDATION SECTION ===== #
5  filename = input("Enter your filename (with extension): ")
6
7  path = os.path.abspath(os.path.join(os.path.dirname( __file__ ), '..', 'test'))
8  # validate filename
9  while filename not in os.listdir(path):
10     print("File not found!")
11     filename = input("Enter your filename (with extension): ")
12
13
14  m = readFile(path+'\\'+filename)
15
16
17
18  # ===== MAIN PROGRAM SECTION ===== #
19  printInitInfo(m)
20
21  if isSolvable(m):
22     print("The Problem is solvable!")
23     nodes = 0
24     routes = {str(m) : ('root',None)}
25     root_cost = getCost(m)
26
27     # Init heap
28     l = [(root_cost,m)]
29     hq.heapify(l)
30     print("Please wait...")
31
32     # ===== SOLUTION FINDING SECTION ===== #
33     # begin timer
34     start = dt.now()
35     while not isSolution(m):
36         x,y = koordinat(m,0)
37         for dirs in moveType(x,y):
38             nm = dc(m)
39             nm = move(nm, x, y, dirs)
40
41             # enqueue new problem for direction dirs, with its cost consist of sum of current cost and root
42             if str(nm) not in routes:
43                 hq.heappush(l, (getCost(nm) + root_cost, nm))
44                 nodes += 1
45                 routes[str(nm)] = (m,dirs)
46
47             # get the least cost puzzle
48             root_cost,m = hq.heappop(l)
49
50     end = dt.now()
51     # end timer
52
53
54     # Gather and print the solved path
55     result = gatherSolvedPath(m,routes)
56     print()
57     choice = input("Do you want to launch graphical visualization? (y/n) ")
58     while choice not in ['y','n']:
59         print("Invalid input!")
60         choice = input("Do you want to launch graphical visualization? (y/n) ")
61
62     print("\nTotal Node(s) Needed to Solve : %d" % nodes)
63     print("ALGORITHM RUNTIME: %.5f s" % (float((end-start).total_seconds())))
64
65     if choice == 'y':
66         result = [i[0] for i in result][::-1]
67         gui(result)
68     else:
69         printSolvedPath(result)
70
71
72     # Information about the solution
73 else:
74     print("THIS PROBLEM IS NOT SOLVABLE")
75
```


BAB III

Hasil Percobaan

NB : Hasil percobaan ditampilkan dalam mode terminal untuk mempermudah *screenshot* setiap step dan menghemat waktu. Contoh visualisasi GUI ada di bagian akhir bab 3

1. File tcY1.txt

Simpul yang dibangkitkan : 52

Total Steps : 8

Runtime Algoritma Utama : 0.00 s

```
-----+-----+-----+-----+
| 05 | 01 | 02 | 04 |
-----+-----+-----+-----+
| 09 | 06 | 03 | 08 |
-----+-----+-----+-----+
|   | 10 | 07 | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
STEP 1: MOVE DOWN
-----+-----+-----+-----+
| 05 | 01 | 02 | 04 |
-----+-----+-----+-----+
|   | 06 | 03 | 08 |
-----+-----+-----+-----+
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
STEP 2: MOVE DOWN
-----+-----+-----+-----+
|   | 01 | 02 | 04 |
-----+-----+-----+-----+
| 05 | 06 | 03 | 08 |
-----+-----+-----+-----+
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
```

```
STEP 3: MOVE LEFT
-----+-----+-----+-----+
| 01 |   | 02 | 04 |
-----+-----+-----+-----+
| 05 | 06 | 03 | 08 |
-----+-----+-----+-----+
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
STEP 4: MOVE LEFT
-----+-----+-----+-----+
| 01 | 02 |   | 04 |
-----+-----+-----+-----+
| 05 | 06 | 03 | 08 |
-----+-----+-----+-----+
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
STEP 5: MOVE UP
-----+-----+-----+-----+
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----+
| 05 | 06 |   | 08 |
-----+-----+-----+-----+
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
```

```
STEP 6: MOVE UP
-----+-----+-----+-----+
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----+
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----+
| 09 | 10 |   | 11 |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
STEP 7: MOVE LEFT
-----+-----+-----+-----+
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----+
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----+
| 09 | 10 | 11 |   |
-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----+
STEP 8: MOVE UP
-----+-----+-----+-----+
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----+
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----+
| 09 | 10 | 11 | 12 |
-----+-----+-----+-----+
| 13 | 14 | 15 |   |
-----+-----+-----+-----+
PROBLEM SOLVED!!
Total Node(s) Needed to Solve : 52
ALGORITHM RUNTIME: 0.00000 s
```

2. File tcY2.txt

Simpul yang dibangkitkan : 30485

Total Steps : 21

Runtime Algoritma : 0.5731 s

```

-----
| 02 | 03 | 04 | 11 |
-----+-----+-----+-----
| 01 | 05 | 10 | 08 |
-----+-----+-----+-----
| 09 | 06 | 12 | 15 |
-----+-----+-----+-----
| 13 | 14 |   | 07 |
-----+-----+-----+-----
STEP 1: MOVE LEFT
-----
| 02 | 03 | 04 | 11 |
-----+-----+-----+-----
| 01 | 05 | 10 | 08 |
-----+-----+-----+-----
| 09 | 06 | 12 | 15 |
-----+-----+-----+-----
| 13 | 14 | 07 |   |
-----+-----+-----+-----
STEP 2: MOVE DOWN
-----
| 02 | 03 | 04 | 11 |
-----+-----+-----+-----
| 01 | 05 | 10 | 08 |
-----+-----+-----+-----
| 09 | 06 | 12 |   |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 3: MOVE RIGHT
-----
| 02 | 03 | 04 | 11 |
-----+-----+-----+-----
| 01 | 05 | 10 | 08 |
-----+-----+-----+-----
| 09 | 06 |   | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----

```

```

STEP 4: MOVE DOWN
-----
| 02 | 03 | 04 | 11 |
-----+-----+-----+-----
| 01 | 05 |   | 08 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 5: MOVE LEFT
-----
| 02 | 03 | 04 | 11 |
-----+-----+-----+-----
| 01 | 05 | 08 |   |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 6: MOVE DOWN
-----
| 02 | 03 | 04 |   |
-----+-----+-----+-----
| 01 | 05 | 08 | 11 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 7: MOVE RIGHT
-----
| 02 | 03 |   | 04 |
-----+-----+-----+-----
| 01 | 05 | 08 | 11 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----

```

```

STEP 8: MOVE RIGHT
-----
| 02 |   | 03 | 04 |
-----+-----+-----+-----
| 01 | 05 | 08 | 11 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 9: MOVE RIGHT
-----
|   | 02 | 03 | 04 |
-----+-----+-----+-----
| 01 | 05 | 08 | 11 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 10: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
|   | 05 | 08 | 11 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 11: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 |   | 08 | 11 |
-----+-----+-----+-----
| 09 | 06 | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----

```

```

STEP 12: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 |   | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 13: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 |   | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 15 |
-----+-----+-----+-----
STEP 14: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 07 | 12 |
-----+-----+-----+-----
| 13 | 14 |   | 15 |
-----+-----+-----+-----
STEP 15: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 07 | 12 |
-----+-----+-----+-----
| 13 | 14 | 15 |   |
-----+-----+-----+-----

```

```

STEP 16: MOVE DOWN
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 07 |   |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 17: MOVE DOWN
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 |   |
-----+-----+-----+-----
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 18: MOVE RIGHT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 |   | 08 |
-----+-----+-----+-----
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 19: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----
| 09 | 10 |   | 11 |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----

```

```

STEP 20: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----
| 09 | 10 | 11 |   |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 21: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----
| 09 | 10 | 11 | 12 |
-----+-----+-----+-----
| 13 | 14 | 15 |   |
-----+-----+-----+-----
PROBLEM SOLVED!!
Total Node(s) Needed to Solve : 30485
ALGORITHM RUNTIME: 0.57310 s

```

3. File tcY3.txt

Simpul yang dibangkitkan : 1685878

Total Steps : 31

Runtime Algoritma : 49.69622 s

```

-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 14 | 13 | 10 |
-----+-----+-----+-----
|   | 07 | 08 | 12 |
-----+-----+-----+-----
STEP 1: MOVE DOWN
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
|   | 14 | 13 | 10 |
-----+-----+-----+-----
| 09 | 07 | 08 | 12 |
-----+-----+-----+-----
STEP 2: MOVE LEFT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 |   | 13 | 10 |
-----+-----+-----+-----
| 09 | 07 | 08 | 12 |
-----+-----+-----+-----
STEP 3: MOVE LEFT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 | 13 |   | 10 |
-----+-----+-----+-----
| 09 | 07 | 08 | 12 |
-----+-----+-----+-----
STEP 4: MOVE LEFT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 | 13 | 10 |   |
-----+-----+-----+-----
| 09 | 07 | 08 | 12 |
-----+-----+-----+-----

```

```

STEP 5: MOVE UP
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 | 13 | 10 | 12 |
-----+-----+-----+-----
| 09 | 07 | 08 |   |
-----+-----+-----+-----
STEP 6: MOVE RIGHT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 | 13 | 10 | 12 |
-----+-----+-----+-----
| 09 | 07 |   | 08 |
-----+-----+-----+-----
STEP 7: MOVE RIGHT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 | 13 | 10 | 12 |
-----+-----+-----+-----
| 09 |   | 07 | 08 |
-----+-----+-----+-----
STEP 8: MOVE DOWN
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 14 |   | 10 | 12 |
-----+-----+-----+-----
| 09 | 13 | 07 | 08 |
-----+-----+-----+-----
STEP 9: MOVE RIGHT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
|   | 14 | 10 | 12 |
-----+-----+-----+-----
| 09 | 13 | 07 | 08 |
-----+-----+-----+-----

```

```

STEP 10: MOVE UP
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 14 | 10 | 12 |
-----+-----+-----+-----
|   | 13 | 07 | 08 |
-----+-----+-----+-----
STEP 11: MOVE LEFT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 14 | 10 | 12 |
-----+-----+-----+-----
| 13 |   | 07 | 08 |
-----+-----+-----+-----
STEP 12: MOVE DOWN
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 |   | 10 | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 08 |
-----+-----+-----+-----
STEP 13: MOVE LEFT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 10 |   | 12 |
-----+-----+-----+-----
| 13 | 14 | 07 | 08 |
-----+-----+-----+-----
STEP 14: MOVE LEFT
-----+-----+-----+-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 10 | 12 |   |
-----+-----+-----+-----
| 13 | 14 | 07 | 08 |
-----+-----+-----+-----

```

```

STEP 15: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 10 | 12 | 08 |
-----+-----+-----+-----
| 13 | 14 | 07 |   |
-----+-----+-----+-----
STEP 16: MOVE RIGHT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 10 | 12 | 08 |
-----+-----+-----+-----
| 13 | 14 |   | 07 |
-----+-----+-----+-----
STEP 17: MOVE DOWN
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 10 |   | 08 |
-----+-----+-----+-----
| 13 | 14 | 12 | 07 |
-----+-----+-----+-----
STEP 18: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 | 15 |
-----+-----+-----+-----
| 09 | 10 | 08 |   |
-----+-----+-----+-----
| 13 | 14 | 12 | 07 |
-----+-----+-----+-----
STEP 19: MOVE DOWN
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 11 |   |
-----+-----+-----+-----
| 09 | 10 | 08 | 15 |
-----+-----+-----+-----
| 13 | 14 | 12 | 07 |
-----+-----+-----+-----

```

```

STEP 20: MOVE RIGHT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 |   | 11 |
-----+-----+-----+-----
| 09 | 10 | 08 | 15 |
-----+-----+-----+-----
| 13 | 14 | 12 | 07 |
-----+-----+-----+-----
STEP 21: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 |   | 15 |
-----+-----+-----+-----
| 13 | 14 | 12 | 07 |
-----+-----+-----+-----
STEP 22: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 15 |   |
-----+-----+-----+-----
| 13 | 14 | 12 | 07 |
-----+-----+-----+-----
STEP 23: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 15 | 07 |
-----+-----+-----+-----
| 13 | 14 | 12 |   |
-----+-----+-----+-----
STEP 24: MOVE RIGHT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 15 | 07 |
-----+-----+-----+-----
| 13 | 14 |   | 12 |
-----+-----+-----+-----

```

```

STEP 25: MOVE DOWN
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 |   | 07 |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 26: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 08 | 11 |
-----+-----+-----+-----
| 09 | 10 | 07 |   |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 27: MOVE DOWN
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 |   | 08 |
-----+-----+-----+-----
| 09 | 10 | 07 | 11 |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 29: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----
| 09 | 10 |   | 11 |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----
STEP 30: MOVE LEFT
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----
| 09 | 10 | 11 |   |
-----+-----+-----+-----
| 13 | 14 | 15 | 12 |
-----+-----+-----+-----

```

```

STEP 31: MOVE UP
-----
| 01 | 02 | 03 | 04 |
-----+-----+-----+-----
| 05 | 06 | 07 | 08 |
-----+-----+-----+-----
| 09 | 10 | 11 | 12 |
-----+-----+-----+-----
| 13 | 14 | 15 |   |
-----+-----+-----+-----
PROBLEM SOLVED!!
Total Node(s) Needed to Solve : 1685878
ALGORITHM RUNTIME: 49.69622 s

```

4. File tcX1.txt
(Kasus Puzzle not Solvable)

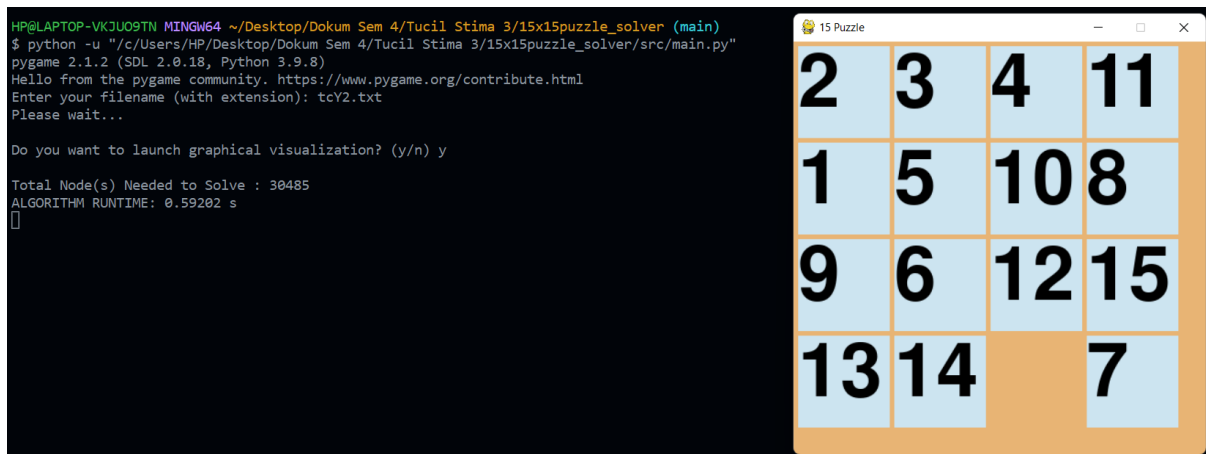
```
Enter your filename (with extension): tcX1.txt
THIS PROBLEM IS NOT SOLVABLE
```

5. File tcX2.txt
(Kasus Puzzle not Solvable)

```
Enter your filename (with extension): tcX2.txt
THIS PROBLEM IS NOT SOLVABLE
```

Contoh Visualisasi menggunakan GUI (PyGame)

Inisialisasi



Tahap selanjutnya...



(setiap 0.5 detik berganti hingga solusi akhir)

....



LAMPIRAN

1. Alamat Repo Github
https://github.com/patrickamadeus/15puzzle_solver

2. Cek List

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