

Laporan Tugas Kecil IF2211

Strategi Algoritma

Penyelesaian Persoalan 15-Puzzle dengan Algoritma Branch and Bound



oleh
Febryola Kurnia Putri
13520140
K02

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

Daftar Isi

DAFTAR ISI	1
BAB I : Algoritma Branch and Bound	2
BAB II : Source Code	3
BAB III : Test Case	11
BAB IV : Data Uji	22
BAB V : Lampiran	24

BAB I

Algoritma Branch and Bound

Algoritma Branch and Bound diterapkan untuk menyelesaikan persoalan 15-Puzzle ini. Penggambaran algoritmanya yaitu dengan struktur pohon/tree. Pohon pada program ini menggambarkan keadaan dari setiap simpul. Setiap simpul pada pohon memiliki elemen puzzle, basis dari matrix (koordinat puzzle yang kosong), (matrix) yang berfungsi untuk menyimpan pola puzzle sementara setiap simpul, path (jalur yang ditempuh untuk mencapai simpul digambarkan dengan u (up), l(left), d(down), dan r(right), function cost nya yang digunakan untuk menghitung cost yang dibutuhkan untuk setiap simpulnya, serta cabang pohon untuk membangkitkan simpul-simpul anaknya. ubin kosong direpresentasikan dengan angka 0. Algoritma Branch and Bound yang digunakan pada persoalan 15-puzzle ini secara umum sebagai berikut:

1. Masukkan simpul akar ke dalam antrian Queue. Jika simpul akar merupakan simpul solusi atau goal node-nya maka berarti solusi telah ditemukan. Hentikan pencarian.
2. Jika antrian Queue kosong, maka hentikan pencarian.
3. Jika antrian Queue tidak kosong, pilih dari antrian Queue simpul i yang mempunyai nilai cost function yang paling kecil. Jika terdapat beberapa simpul i yang memenuhi, pilih satu secara sembarang.
4. Jika simpul i adalah simpul solusi, maka berarti solusi sudah ditemukan. Persoalan ini menggunakan pendekatan least cost search, maka 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 simpul i tidak mempunyai anak, maka kembali ke langkah 2.
6. Untuk setiap anak j dari simpul i , hitung cost function untuk simpul j , dan masukkan semua anak-anak tersebut ke dalam antrian Queue.
7. Kembali ke langkah 2.

Algoritma ini menggunakan prinsip least cost search pada pencarian simpul anak pada tiap langkahnya. Least cost search digunakan untuk menentukan simpul anak mana yang akan ditelusuri selanjutnya. Tiap pembangkitan simpul anak, akan dihitung nilai costnya dan dimasukkan ke dalam antrian Queue pemrosesan simpul hidup. Queue yang digunakan adalah Priority Queue dengan prioritas utama adalah simpul anak dengan nilai cost terkecil. Dengan demikian simpul anak yang dipilih yang memiliki cost terkecil dari simpul anak yang lain pada antrian pemrosesan. Cost yang digunakan adalah sebagai berikut :

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

$c(i)$ = cost untuk simpul ke i

$f(i)$ = cost untuk mencapai simpul ke i dari simpul akar (depth)

$g(i)$ = cost untuk mencapai simpul tujuan dari simpul ke i . Dalam kasus ini adalah jumlah susunan ubin pada simpul anak yang tidak terdapat pada susunan akhir.

BAB II

Source Code

Program 15-Puzzle ini memiliki 3 file di dalamnya, yaitu main.py sebagai main program, functionFile.py sebagai file tempat fungsi-fungsi branch and boundnya dan pengolahan console serta file and puzzle15.py sebagai kelas puzzle dan nodenya.

1. main.py

```
1  #import dependency yang dibutuhkan
2  from puzzle15 import Node, Puzzle
3  import functionFile
4  import numpy as np
5  import time
6
7  #fungsi untuk mencetak header
8  def header():
9      print(""" \033[36m
10 | | | | |
11 | | | | |
12 | | | | |
13 | | | | |
14 | | | | |
15 | | | | |
16 | | | | |
17 | | | | |
18 | | | | |
19 | | | | |
20 | | | | |
21 | | | | |
22 | | | | |
23 | | | | |
24 | | | | |
25 | | | | |
26 | | | | |
27 | | | | |
28 | | | | |
29 | | | | |
30 | | | | |
31 | | | | |
32 | | | | |
33 | | | | |
34 | | | | |
35 | | | | |
36 | | | | |
37 | | | | |
38 | | | | |
39 | | | | |
40 | | | | |
41 | | | | |
42 | | | | |
    \033[0m""")
21 #fungsi untuk mencetak welcoming message
22 def selamatDatang():
23     print("\033[31m===== \033[0m")
24     print("\033[34m=====Selamat datang di program solver puzzle===== \033[0m")
25     print("\033[31m===== \033[0m")
26     print()
27     print("\033[33mProgram ini dibuat oleh Febryola Kurnia Putri-13520140 K02 \033[0m")
28     print("\033[37mProgram ini dibuat untuk memenuhi tugas mata kuliah IF2211 \033[0m")
29     print("\033[31m===== \033[0m")
30
31 #fungsi untuk mencetak terima kasih
32 def terimakasih():
33     print(""" \033[36m
34 | | | | |
35 | | | | |
36 | | | | |
37 | | | | |
38 | | | | |
39 | | | | |
40 | | | | |
41 | | | | |
42 | | | | |
    \033[0m""")
```

```

44 #fungsi untuk menginputkan puzzle
45 def masukanPuzzle():
46     print("\n\033[32mPilih opsi untuk memasukkan puzzle: \033[0m")
47     print("1. Masukan dari console")
48     print("2. Masukan dari file")
49     x = int(input("\033[48mMasukkan nomor pilihan : \033[0m"))
50     if (x == 1):
51         print()
52         print("""Masukkan 4 baris puzzle yang dipisahkan dengan spasi
53 contoh : \n1 2 3 4\n5 6 7 8\n9 10 11 12\n13 14 15 0\n""")
54         print("\033[33mMasukkan puzzle \033[0m: ")
55         puzzle = functionFile.consoleToPuzzle()
56     else :
57         print("\033[36mfile testing dimasukkan ke folder test\033[0m")
58         print("contoh masukan: test1.txt")
59         filename = str(input("\033[36mMasukkan nama file \033[0m: "))
60         sebelum = "./test/"
61         puzzle = functionFile.fileToPuzzle(sebelum+filename)
62     return puzzle
63
64 #fungsi utama
65 def main():
66     #puzzle awal
67     puzzleAwal = np.array([1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,0]).reshape(4,4)
68     #inputkan puzzle
69     puzzle = masukanPuzzle()
70     print("\033[35m\n\n===== \033[33mInisialisasi Puzzle\033[0m \033[35m===== \033[0m")
71     #inisialisasi puzzle
72     a,b = functionFile.cekApakahKosong(puzzle)
73     #jika puzzle kosong
74     print("\033[31mUbin Kosong Berada Pada Koordinat (" +str(a)+"," +str(b)+")\033[0m")
75     #mencetak cost awal
76     print("\033[31mCost : " +str(functionFile.hitungCost(0,puzzle,puzzleAwal))+ "\033[0m")
77     #mencetak puzzle
78     functionFile.cetakPuzzle(puzzle)
79     print("\033[33m===== \033[0m")
80     print("\033[35mBerikut Fungsi Kurang setiap Box\n\033[0m")
81     functionFile.cetakFungsiKurang(puzzle)
82     print("\033[31mSigma Kurang(i) + X = " + str(functionFile.result(puzzle))+ "\033[0m")
83     perpindahan = ("right", "down", "left", "up")
84     print("\n\033[33m===== \033[0m")
85
86     if(functionFile.solvable(puzzle)): #jika puzzle solvable
87         Queue = Puzzle() #inisialisasi queue
88         print("\033[32mPuzzle dapat diselesaikan :))\033[0m\n")
89
89     simpul_yang_dibangkitkan = 0 #inisialisasi simpul yang dibangkitkan
90     print("Puzzle Awal")
91     simpul = Node(puzzle) #inisialisasi simpul awal
92
93     start = time.time() #waktu mulai
94     Queue.enqueue((functionFile.hitungCost(0,simpul.puzzle,puzzleAwal),simpul,"",0))
95     #enqueue simpul awal
96     Queue_temp = Queue.dequeue()
97     simpul = Queue_temp[1] #inisialisasi simpul awal
98     New_puzzle = simpul.puzzle #inisialisasi puzzle awal
99     Move_balik = "" #inisialisasi move balik
100     next_step = Queue_temp[3] + 1 #inisialisasi next step
101     simpul_yang_dibangkitkan += 1 #simpul yang dibangkitkan
102
103     while(not functionFile.apakahSama(New_puzzle, puzzleAwal)): #jika puzzle tidak sama dengan puzzle awal
104         for mov in perpindahan: #mengulang perpindahan
105             if(mov != Move_balik): #jika move tidak sama dengan move balik
106                 after_move = functionFile.move(New_puzzle,mov) #pindahkan puzzle
107                 if(not functionFile.apakahSama(after_move,New_puzzle)): #jika puzzle tidak sama dengan puzzle awal

```

```

108         new_simpul = Node(after_move) #inisialisasi simpul baru
109         new_simpul.parent = simpul #inisialisasi parent
110         new_simpul.dalam = simpul.dalam + 1 #inisialisasi dalam
111         simpul_yang_dibangkitkan += 1 #simpul yang dibangkitkan
112         Queue.enqueue((functionFile.hitungCost(next_step,new_simpul.puzzle,puzzleAwal),new_simpul,mov,next_step))
113     #jika sudah tidak ada simpul yang dapat dibangkitkan
114     Queue_temp = Queue.dequeue()
115     simpul = Queue_temp[1]
116     New_puzzle = simpul.puzzle
117     Move= Queue_temp[2]
118     Move_balik = functionFile.move_mirror(Move)
119     next_step = Queue_temp[3] + 1
120
121     #jika sudah sama dengan puzzle awal
122     end = time.time() #waktu selesai
123     functionFile.cetakPuzzle(puzzle) #cetak puzzle
124     functionFile.cetakPath(simpul) #cetak path
125     waktu = end - start #waktu yang ditempuh
126     print("\n\033[32mJumlah Langkah = " +str(next_step-1)+"\033[0m") #jumlah langkah
127     print("\033[33mWaktu Eksekusi = " +str(waktu) + " seconds" + "\033[0m") #waktu eksekusi
128     print("\033[34mJumlah Simpul yang Dibangkitkan = " + str(simpul_yang_dibangkitkan)+ "\033[0m")
129 else: #jika puzzle tidak solvable
130     print("\033[34mPuzzle tidak dapat diselesaikan karena sigma Kurang(i) + X bernilai ganjil!\033[0m\n")
131
132 lanjut = str(input("\n\033[35mApakah Anda ingin melanjutkan program \033[0m? (y/n) "))
133 #jika lanjut
134 if (lanjut == "Y" or lanjut == "y"):
135     print()
136     main() #panggil fungsi utama
137 else:
138     terimakasih() #terimakasih
139     exit() #keluar
140
141 if __name__ == "__main__": #jika program dijalankan
142     header() #cetak header
143     selamatDatang() #cetak selamat datang
144     main() #panggil fungsi utama
145     terimakasih() #terimakasih

```

2. puzzle15.py

```

1  # class utama yang diperlukan
2
3  #class Node untuk simpul dan menyimpan matrix dan parent
You, 7 minutes ago | 1 author (You)
4  class Node:
5      #constructor
6      def __init__(self, data=None):
7          self.puzzle = data      You, 7 minutes ago • update semuaa ...
8          self.parent = None
9          self.dalam = 0
10
11 #class untuk mencari solusi puzzle
You, 7 minutes ago | 1 author (You)
12 class Puzzle(object):
13     #constructor
14     def __init__(self):
15         self.queue = []
16
17     #method join untuk menggabungkan 2 matrix
18     def __str__(self):
19         return '\n'.join([str(i) for i in self.queue])
20
21     #method enqueue untuk menambahkan simpul baru ke queue
22     def enqueue(self, data):
23         self.queue.append(data)
24
25     #method dequeue untuk menghapus simpul terakhir dari queue
26     def dequeue(self):
27         index = 0
28         #mencari simpul terakhir
29         for i in range(len(self.queue)):
30             if(self.queue[i][0] < self.queue[index][0]):
31                 index = i
32         #menghapus simpul terakhir
33         item = self.queue[index]
34         del self.queue[index]
35         #mengembalikan simpul terakhir
36         return item
37

```

3. functionFile.py

```

1  #import dependency yang dibutuhkan
2  import numpy as np
3
4  #fungsi untuk mengeksport console ke puzzle
5  def consoleToPuzzle():
6      puzzle = [[0] * 4 for _ in range(4)]
7      for i in range(4):
8          lines = list(map(int, input().split()))
9          for j in range(4):
10             puzzle[i][j] = lines[j]
11     puzzles = np.reshape(puzzle,(4,4)).astype('int32')
12     #Mengembalikan puzzle
13     return puzzles
14
15     #fungsi untuk membaca file
16     def readFile(filename):
17         f = open(filename,"r")
18         return f
19
20     #fungsi untuk mengubah file ke puzzle
21     def fileToPuzzle(filename):
22         text = readFile(filename)
23         temp = text.read().split()
24         puzzle = np.reshape(temp,(4,4)).astype('int32')
25         return puzzle
26
27     #fungsi untuk mencetak path
28     def cetakPath(simpul):
29         if(simpul.parent == None):
30             return
31         cetakPath(simpul.parent)
32         print("\033[35m\n=====\\033[0m")
33         print("\033[31mLangkah ke - "+str(simpul.dalam)+" : "+"\\033[0m")
34         #cetak puzzle
35         cetakPuzzle(simpul.puzzle)
36
37     #fungsi untuk mencetak puzzle
38     def cetakPuzzle(puzzle):
39         print("\033[34m=====\\033[0m")
40         for i in range(4):
41             for j in range(4):
42                 print("\033[34m\\033[0m ",end="")
43                 print(puzzle[i][j], end="")
44                 if(puzzle[i][j] < 10):
45                     print(" ", end="")

```



```

46     print("\033[34m\033[0m")
47     if(i != 3):
48         print("\033[34m\033[0m")
49     print("\033[34m\033[0m")
50
51     #fungsi untuk mengecek apakah puzzle kosong
52     def cekApakahKosong(puzzle):
53         for i in range(4):
54             for j in range(4):
55                 if(puzzle[i][j] == 0):
56                     row = i
57                     column = j
58             return row,column
59
60     #fungsi untuk mengubah angka 0 menjadi posisi yang seharusnya
61     def ubahAngka0(puzzle):
62         x,y = cekApakahKosong(puzzle)
63         puzzle[x][y] = 16
64         return puzzle
65
66     #fungsi untuk mencari nilai x
67     def temukanX(puzzle):
68         puzzle_temp = puzzle.copy()
69         x,y = cekApakahKosong(puzzle_temp)
70         sum = x+y
71         return (sum % 2)
72
73     #fungsi untuk menghitung cost
74     def hitungCost(dalam,puzzle,mat):
75         hitung = 0
76         for i in range(4):
77             for j in range(4):
78                 if((puzzle[i][j] != mat[i][j]) and puzzle[i][j] != 0):
79                     hitung += 1
80         return (dalam+hitung)
81
82     #fungsi untuk mencari nilai fungsi kurang setiap ubin
83     def fungsiKurang(puzzle):
84         hitung = 0
85         puzzle_temp = puzzle.copy()
86         puzzle_temp = ubahAngka0(puzzle_temp)
87         puzzle_temp = np.reshape(puzzle_temp,(16,))
88         for i in range(16):
89             temp = puzzle_temp[i]
90             for j in range(i,16):

```

```

91         if(temp > puzzle_temp[j]):
92             hitung += 1
93     return hitung
94
95     #fungsi untuk mencetak fungsi kurang setiap ubin
96     def cetakFungsiKurang (parameter) puzzle: Any
97         puzzle_temp = puzzle.copy()
98         puzzle_temp = ubahAngka0(puzzle_temp)
99         puzzle_temp = np.reshape(puzzle_temp,(16,))
100         for i in range(16):
101             hitung = 0
102             temp = puzzle_temp[i]
103             for j in range(i,16):
104                 if(temp > puzzle_temp[j]):
105                     hitung += 1
106             print("Fungsi Kurang("+ str(temp)+") = "+ str(hitung))
107
108     #fungsi untuk mencari nilai sum
109     def result(puzzle):
110         sum = temukanX(puzzle)
111         sum += fungsiKurang(puzzle)
112         return sum
113
114     #fungsi untuk menentukan apakah puzzle dapat diselesaikan
115     def solvable(puzzle):
116         sum = result(puzzle)
117         if(sum % 2 == 0):
118             return True
119         else:
120             return False
121
122     #fungsi untuk menukar posisi puzzle
123     def swap(puzzle,row,column):
124         x,y = cekApakahKosong(puzzle)
125         puzzle[x][y] = puzzle[row][column]
126         puzzle[row][column] = 0
127         return puzzle
128
129     #fungsi untuk mencari nilai pindah
130     def move(puzzle,pindah):
131         puzzle_temp = puzzle.copy()
132         x,y = cekApakahKosong(puzzle_temp)
133         if(pindah == "left"):
134             if(y != 0):
135                 y -= 1
136         elif(pindah == "right"):

```

```

136     elif(pindah == "right"):
137         if(y != 3):
138             y += 1
139     elif(pindah == "up"):
140         if(x != 0):
141             x -= 1
142     elif(pindah == "down"):
143         if(x != 3):
144             x += 1
145     puzzle_temp = swap(puzzle_temp,x,y)
146     return puzzle_temp
147
148     #fungsi untuk mencari lawan dari move
149     def move_mirror(pindah):
150         if(pindah == "left"):
151             return "right"
152         elif(pindah == "right"):
153             return "left"
154         elif(pindah == "up"):
155             return "down"
156         elif(pindah == "down"):
157             return "up"
158
159     #fungsi untuk mengecek apakah kedua puzzle sama
160     def apakahSama(puzzle,mat):
161         for i in range(4):
162             for j in range(4):
163                 if(puzzle[i][j] != mat[i][j]):
164                     return False
165         return True
166
167

```

BAB III

Test Case

1. Kasus Uji 1 (unSolve)

- input

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

- output

```
15-Puzzle

=====Selamat datang di program solver puzzle=====

Program ini dibuat oleh Febryola Kurnia Putri-13520140 K02
Program ini dibuat untuk memenuhi tugas mata kuliah IF2211

===== Inisialisasi Puzzle =====
Ubin Kosong Berada Pada Koordinat (1,3)
Cost : 14



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



=====
```

```
Berikut Fungsi Kurang setiap Box

Fungsi Kurang(12) = 11
Fungsi Kurang(6) = 5
Fungsi Kurang(1) = 0
Fungsi Kurang(7) = 4
Fungsi Kurang(13) = 8
Fungsi Kurang(10) = 6
Fungsi Kurang(8) = 4
Fungsi Kurang(16) = 8
Fungsi Kurang(4) = 2
Fungsi Kurang(9) = 3
Fungsi Kurang(5) = 2
Fungsi Kurang(15) = 4
Fungsi Kurang(2) = 0
Fungsi Kurang(14) = 2
Fungsi Kurang(3) = 0
Fungsi Kurang(11) = 0
Sigma Kurang(i) + X = 59

=====
Puzzle tidak dapat diselesaikan karena sigma Kurang(i) + X bernilai ganjil!

Apakah Anda ingin melanjutkan program ? (y/n) █
```

2. Kasus Uji 2 (unSolve)

- input

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

- output

```
15-Puzzle

=====
=====Selamat datang di program solver puzzle=====
=====

Program ini dibuat oleh Febryola Kurnia Putri-13520140 K02
Program ini dibuat untuk memenuhi tugas mata kuliah IF2211
=====
```

```

===== Inisialisasi Puzzle =====
Ubin Kosong Berada Pada Koordinat (2,3)
Cost : 13



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



=====
Berikut Fungsi Kurang setiap Box

Fungsi Kurang(13) = 12
Fungsi Kurang(2) = 1
Fungsi Kurang(6) = 4
Fungsi Kurang(11) = 8
Fungsi Kurang(12) = 8
Fungsi Kurang(1) = 0
Fungsi Kurang(4) = 1
Fungsi Kurang(10) = 5
Fungsi Kurang(15) = 6
Fungsi Kurang(7) = 2
Fungsi Kurang(3) = 0
Fungsi Kurang(16) = 4
Fungsi Kurang(8) = 1
Fungsi Kurang(14) = 2
Fungsi Kurang(5) = 0
Fungsi Kurang(9) = 0
Sigma Kurang(i) + X = 55

=====
Puzzle tidak dapat diselesaikan karena sigma Kurang(i) + X bernilai ganjil!

Apakah Anda ingin melanjutkan program ? (y/n) █

```

3. Kasus Uji 3 (Solve)

- input

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

- output

15-Puzzle

=====Selamat datang di program solver puzzle=====

Program ini dibuat oleh Febryola Kurnia Putri-13520140 K02
Program ini dibuat untuk memenuhi tugas mata kuliah IF2211

Pilih opsi untuk memasukkan puzzle:

1. Masukan dari console
2. Masukan dari file

Masukkan nomor pilihan : 2

file testing dimasukkan ke folder test

contoh masukan: test1.txt

Masukkan nama file : solve_1.txt

===== Inisialisasi Puzzle =====

Ubin Kosong Berada Pada Koordinat (2,1)

Cost : 11

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

=====

Berikut Fungsi Kurang setiap Box

=====

Berikut Fungsi Kurang setiap Box

Fungsi Kurang(6) = 5

Fungsi Kurang(5) = 4

Fungsi Kurang(2) = 1

Fungsi Kurang(4) = 2

Fungsi Kurang(9) = 4

Fungsi Kurang(1) = 0

Fungsi Kurang(3) = 0

Fungsi Kurang(8) = 1

Fungsi Kurang(10) = 1

Fungsi Kurang(16) = 6

Fungsi Kurang(7) = 0

Fungsi Kurang(15) = 4

Fungsi Kurang(13) = 2

Fungsi Kurang(14) = 2

Fungsi Kurang(12) = 1

Fungsi Kurang(11) = 0

Sigma Kurang(1) + X = 34

=====

Puzzle dapat diselesaikan :))

Puzzle Awal

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

=====

Langkah ke - 1 :

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

=====

Langkah ke - 2 :

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

=====

Langkah ke - 3 :

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

=====

Langkah ke - 4 :

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

=====

Langkah ke - 5 :

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

=====

Langkah ke - 6 :

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

=====

Langkah ke - 7 :

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

=====

Langkah ke - 8 :

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

=====

Langkah ke - 9 :

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

Tugas Kecil 3 STIMA - 15-Puzzle
IF2211 Strategi Algoritma 2021/2022

15


```
Langkah ke - 10 :


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



=====
Langkah ke - 11 :


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



=====
Langkah ke - 12 :


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



=====
Langkah ke - 13 :


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



=====
Langkah ke - 14 :


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



=====
Langkah ke - 15 :


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



=====
Langkah ke - 16 :


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



=====
Langkah ke - 17 :


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



Jumlah Langkah = 17
Waktu Eksekusi = 0.11599206924438477 seconds
Jumlah Simpul yang Dibangkitkan = 1229

Apakah Anda ingin melanjutkan program ? (y/n) █
```

4. Kasus Uji 4 (Solve)

- input

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

- output

```

=====
15-Puzzle
=====Selamat datang di program solver puzzle=====
=====
Program ini dibuat oleh Febryola Kurnia Putri-13520140 K02
Program ini dibuat untuk memenuhi tugas mata kuliah IF2211
=====

Pilih opsi untuk memasukkan puzzle:
1. Masukan dari console
2. Masukan dari file
Masukkan nomor pilihan : 2
file testing dimasukkan ke folder test
contoh masukan: test1.txt
Masukkan nama file : solve_2.txt

===== Inisialisasi Puzzle =====
Ubin Kosong Berada Pada Koordinat (1,1)
Cost : 8



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



=====
Berikut Fungsi Kurang setiap Box

Fungsi Kurang(1) = 0
Fungsi Kurang(6) = 4
Fungsi Kurang(2) = 0
Fungsi Kurang(4) = 1
Fungsi Kurang(5) = 1
Fungsi Kurang(16) = 10
Fungsi Kurang(3) = 0
Fungsi Kurang(8) = 1
Fungsi Kurang(9) = 1
Fungsi Kurang(7) = 0
Fungsi Kurang(15) = 5
Fungsi Kurang(11) = 1
Fungsi Kurang(13) = 2
Fungsi Kurang(14) = 2
Fungsi Kurang(10) = 0
Fungsi Kurang(12) = 0
Sigma Kurang(i) + X = 28

=====
Puzzle dapat diselesaikan :))

```

Puzzle Awal

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

=====

Langkah ke - 1 :

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

=====

Langkah ke - 2 :

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

=====

Langkah ke - 3 :

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

=====

Langkah ke - 4 :

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

=====

Langkah ke - 5 :

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

=====

Langkah ke - 6 :

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

=====

Langkah ke - 7 :

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

=====

Langkah ke - 8 :

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

=====

Langkah ke - 9 :

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

=====

Langkah ke - 10 :

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

=====

Langkah ke - 11 :

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

```

Langkah ke - 12 :


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



=====
Langkah ke - 13 :


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



=====
Langkah ke - 14 :


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



=====
Langkah ke - 15 :


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



=====
Langkah ke - 16 :


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



=====
Langkah ke - 17 :


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



=====
Langkah ke - 18 :


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



Jumlah Langkah = 18
Waktu Eksekusi = 1.2877891063690186 seconds
Jumlah Simpul yang Dibangkitkan = 9178

Apakah Anda ingin melanjutkan program ? (y/n)

```

5. Kasus Uji 5 (Solve)

- input

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

- output

15-Puzzle

=====Selamat datang di program solver puzzle=====

Program ini dibuat oleh Febryola Kurnia Putri-13520140 K02
Program ini dibuat untuk memenuhi tugas mata kuliah IF2211

Pilih opsi untuk memasukkan puzzle:
1. Masukan dari console
2. Masukan dari file
Masukkan nomor pilihan : 2
file testing dimasukkan ke folder test
contoh masukan: test1.txt
Masukkan nama file : solve_3.txt

===== Inisialisasi Puzzle =====
Ubin Kosong Berada Pada Koordinat (2,0)
Cost : 10

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

===== Berikut Fungsi Kurang setiap Box

===== Berikut Fungsi Kurang setiap Box

Fungsi Kurang(5) = 4
Fungsi Kurang(1) = 0
Fungsi Kurang(3) = 1
Fungsi Kurang(4) = 1
Fungsi Kurang(9) = 4
Fungsi Kurang(2) = 0
Fungsi Kurang(7) = 1
Fungsi Kurang(8) = 1
Fungsi Kurang(16) = 7
Fungsi Kurang(6) = 0
Fungsi Kurang(15) = 5
Fungsi Kurang(11) = 1
Fungsi Kurang(13) = 2
Fungsi Kurang(10) = 0
Fungsi Kurang(14) = 1
Fungsi Kurang(12) = 0
Sigma Kurang(i) + X = 28

===== Puzzle dapat diselesaikan :))

Puzzle Awal

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

===== Langkah ke - 1 :

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

===== Langkah ke - 2 :

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

===== Langkah ke - 3 :

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

===== Langkah ke - 4 :

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

===== Langkah ke - 5 :

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

Langkah ke - 6 :

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

=====

Langkah ke - 7 :

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

=====

Langkah ke - 8 :

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

=====

Langkah ke - 9 :

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

=====

Langkah ke - 10 :

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

Jumlah Langkah = 10
Waktu Eksekusi = 0.002001047134399414 seconds
Jumlah Simpul yang Dibangkitkan = 24

Apakah Anda ingin melanjutkan program ? (y/n)

BAB IV

Data Uji

link berkas semua data uji:

https://drive.google.com/drive/folders/1S6RVgaSAXSS_PxdMTofN5Ynp73PP7fA?usp=sharing

1. Data Uji Unsolveable

- notSolve_1.txt

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

link berkas:

<https://drive.google.com/file/d/14KlyfcE9IghYswxN1MtGLljqgnLx1laE/view?usp=sharing>

- notSolve_2.txt

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

link berkas:

https://drive.google.com/file/d/1bpRsg-5FKS3qsmRNnZ_AwNksXlrsdFVQ/view?usp=sharing

2. Data Uji Solveable

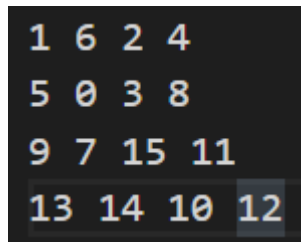
- solve_1.txt

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

link berkas:

<https://drive.google.com/file/d/1Uu9gGNgMOGhFxnZ28YItBPDPcmzNNHqu/view?usp=sharing>

- solve_2.txt

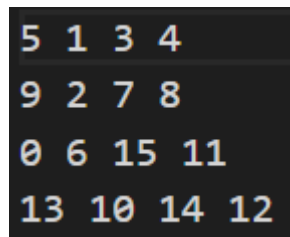


1	6	2	4
5	0	3	8
9	7	15	11
13	14	10	12

link berkas:

<https://drive.google.com/file/d/1as-mezNqPzu3jQwmI2NwkuIvwteYmFgL/view?usp=sharing>

- solve_3.txt



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

link berkas:

https://drive.google.com/file/d/1VYikvQx_XkbeJFPrZ-dOxVLIxIWBEEdCa/view?usp=sharing

Bab V

Lampiran

No	Poin	Ya	Tidak
1	Program berhasil dikompilasi	V	
2	Program berhasil running	V	
3	Program dapat menerima input dan mengeluarkan output	V	
4	Luaran sudah benar untuk semua data uji	V	
5	Bonus dibuat		V

Alamat github : <https://github.com/febryola/tucil3stima>