

# **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	12
BAB IV : Data Uji	18
BAB V : Lampiran	20

# 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, function.py sebagai file tempat fungsi-fungsi branch and boundnya and puzzle.py sebagai kelas puzzle atau matrixnya.

#### 1. main.py

```
You, 1 second ago | 1 author (You)
1  from puzzle import Matrix
2  from function import puzzleSolve
3  import time
4  from collections import deque
5
6  def header():
7      print(""" \033[36m          You, 1 second ago • Uncommitted changes
8
9
10
11
12
13
14
15
16
17  \033[0m""")
18
19  def selamatDatang():
20      print("\033[31m===== \033[0m")
21      print("\033[34m=====Selamat datang di program solver puzzle===== \033[0m")
22      print("\033[31m===== \033[0m")
23      print()
24      print("\033[33mProgram ini dibuat oleh Febryola Kurnia Putri-13520140 K02\033[0m")
25      print("\033[37mProgram ini dibuat untuk memenuhi tugas mata kuliah IF2211\033[0m")
26      print("\033[31m===== \033[0m")
27
28  def masukanInput(puzzlee):
29      print("Pilih opsi untuk memasukkan puzzle: ")
30      print("1. Masukan dari console")
31      print("2. Masukan dari file")
32      x = int(input("\033[48mMasukkan nomor pilihan : \033[0m"))
33      if (x == 1):
34          print()
35          print("""Masukkan 4 baris puzzle yang dipisahkan dengan spasi
36  contoh : \n1 2 3 4\n5 6 7 8\n9 10 11 12\n13 14 15 0\n""")
37          print("\033[33mMasukkan puzzle \033[0m: ")
38          puzzlee.inputFromConsole()
39      else :
40          print("\033[36mfile testing dimasukkan ke folder test\033[0m")
41          print("contoh masukan: test1.txt")
42          filename = str(input("\033[36mMasukkan nama file \033[0m: "))
43          sebelum = "./test/"
44          puzzlee.readFile(sebelum+filename)
45
```

```

46 #fungsi untuk mencetak matriks
47 def printMatriks(mat):
48     for i in range(4):
49         for j in range(4):
50             print(mat[i][j], end = " ")
51         print()
52
53 #fungsi untuk mencetak path
54 def printPath(res, puzzle):
55     mat = puzzle.mat
56     basis = puzzle.basis
57     path = res.path
58     for i in path:
59         if (i == "u"):
60             print("\n\033[33mLangkah yang dilakukan\033[0m : up")
61             mat[basis[0]][basis[1]] = mat[basis[0] - 1][basis[1]]
62             mat[basis[0] - 1][basis[1]] = 0
63             basis = (basis[0] - 1, basis[1])
64         elif (i == "d"):
65             print("\n\033[33mLangkah yang dilakukan\033[0m : down")
66             mat[basis[0]][basis[1]] = mat[basis[0] + 1][basis[1]]
67             mat[basis[0] + 1][basis[1]] = 0
68             basis = (basis[0] + 1, basis[1])
69         elif (i == "l"):
70             print("\n\033[33mLangkah yang dilakukan\033[0m : left")
71             mat[basis[0]][basis[1]] = mat[basis[0]][basis[1] - 1]
72             mat[basis[0]][basis[1] - 1] = 0
73             basis = (basis[0], basis[1] - 1)
74         elif (i == "r"):
75             print("\n\033[33mLangkah yang dilakukan\033[0m : right")
76             mat[basis[0]][basis[1]] = mat[basis[0]][basis[1] + 1]
77             mat[basis[0]][basis[1] + 1] = 0
78             basis = (basis[0], basis[1] + 1)
79     printMatriks(mat)
80
81 def terimakasih():
82     print("""\033[36m
83
84
85
86
87
88
89
90
91
\033[0m""")

```



```

92
93 def main():
94     calon = []
95     Q = deque()
96     puzzlee = Matrix()
97     masukanInput(puzzlee)
98     print()
99     if (not puzzlee.isInputValid()):
100         print("\033[34mInputan tidak valid, silakan kembali masukkan Puzzle!\033[0m\n")
101         masukanInput(puzzlee)
102     print("Inputan Puzzle : ")
103     printMatriks(puzzlee.mat)
104     if (puzzlee.isSolve()):
105         print("\033[33mPuzzle dapat diselesaikan :))\033[0m\n")
106         now = time.time()
107         Result = puzzleSolve(puzzlee, calon, Q)
108         print("\033[36mjumlah langkah : ", len(Result.path))
109         print("\033[31mjalur pencarian : ", Result.path)
110         print("\033[32mWaktu eksekusi : ", time.time()-now, " detik\033[0m\n")
111         print("\033[33mjumlah simpul yang dibangkitkan : ", Matrix.totalNodes)
112         print()
113         cetak = str(input("\033[35mApakah Anda ingin mencetak matriks langkah \033[0m? (y/n) "))
114         if (cetak == "Y" or cetak == "y"):
115             printPath(Result, puzzlee)
116         else :
117             print("\033[34mPuzzle tidak dapat diselesaikan karena fungsi kurang(i) bernilai ganjil!\033[0m\n")
118
119     lanjut = str(input("\n\033[35mApakah Anda ingin melanjutkan program \033[0m? (y/n) "))
120     if (lanjut == "Y" or lanjut == "y"):
121         print()
122         main()
123     else:
124         terimakasih()
125         exit()
126 if __name__ == "__main__":
127     header()
128     selamatDatang()
129     main()
130     terimakasih()

```

## 2. puzzle.py

```

You, 28 minutes ago | 1 author (You)
1 import copy
2
3 #kelas Matrix
You, 28 minutes ago | 1 author (You)
4 class Matrix:
5     # konstruk isi matrix dan nodes awal
6     box = [[1,2,3,4], [5,6,7,8], [9,10,11,12], [13,14,15,0]]
7     totalNodes = 0
8
9     #fungsi untuk konstruktor
10    def __init__(puzzle):
11        Matrix.totalNodes += 1
12        puzzle.u = None #atas
13        puzzle.l = None #kiri
14        puzzle.d = None #bawah
15        puzzle.r = None #kanan
16        puzzle.path = "" #inisialisasi awal path
17        puzzle.hitungcost = 0 #inisialisasi awal fungsi cost
18        puzzle.mat = [[0 for i in range(4)] for i in range(4)] #inisialisasi awal nilai matrix
19        puzzle.basis = (0,0) #inisialisasi basis
20
21    #fungsi untuk mengenerate simpul jika matrix yang kosong digeser ke atas
22    def aturUp(puzzle):
23        if (puzzle.basis[0] - 1 >= 0):
24            Matrix.totalNodes += 1
25            puzzle.u = copy.deepcopy(puzzle)
26            puzzle.u.mat[puzzle.basis[0]][puzzle.basis[1]] = puzzle.u.mat[puzzle.basis[0] - 1][puzzle.basis[1]]
27            puzzle.u.path += "u"
28            puzzle.u.hitungcost = puzzle.u.hitungCost() + len(puzzle.u.path)
29            puzzle.u.mat[puzzle.basis[0] - 1][puzzle.basis[1]] = 0
30            puzzle.u.basis = (puzzle.basis[0] - 1, puzzle.basis[1])
31
32    #fungsi untuk mengenerate simpul jika Matrix yang kosong digeser ke bawah
33    def aturDown(puzzle):
34        if (puzzle.basis[0] + 1 <= 3):
35            Matrix.totalNodes += 1
36            puzzle.d = copy.deepcopy(puzzle)
37            puzzle.d.mat[puzzle.basis[0]][puzzle.basis[1]] = puzzle.d.mat[puzzle.basis[0] + 1][puzzle.basis[1]]
38            puzzle.d.path += "d"
39            puzzle.d.hitungcost = puzzle.d.hitungCost() + len(puzzle.d.path)
40            puzzle.d.mat[puzzle.basis[0] + 1][puzzle.basis[1]] = 0
41            puzzle.d.basis = (puzzle.basis[0] + 1, puzzle.basis[1])
42
43    #fungsi untuk mengenerate simpul jika matrix yang kosong digeser ke kiri
44    def aturLeft(puzzle):

```

```

44 def aturLeft(puzzle):
45     if (puzzle.basis[1] - 1 >= 0):
46         Matrix.totalNodes += 1
47         puzzle.l = copy.deepcopy(puzzle)
48         puzzle.l.mat[puzzle.basis[0]][puzzle.basis[1]] = puzzle.l.mat[puzzle.basis[0]][puzzle.basis[1] - 1]
49         puzzle.l.path += "l"
50         puzzle.l.hitungcost = puzzle.l.hitungCost() + len(puzzle.l.path)
51         puzzle.l.mat[puzzle.basis[0]][puzzle.basis[1] - 1] = 0
52         puzzle.l.basis = (puzzle.basis[0], puzzle.basis[1] - 1)
53
54 #fungsi untuk generate simpul jika Matrix yang kosong digeser ke kanan
55 def aturRight(puzzle):
56     if (puzzle.basis[1] + 1 <= 3):
57         Matrix.totalNodes += 1
58         puzzle.r = copy.deepcopy(puzzle)
59         puzzle.r.mat[puzzle.basis[0]][puzzle.basis[1]] = puzzle.r.mat[puzzle.basis[0]][puzzle.basis[1] + 1]
60         puzzle.r.mat[puzzle.basis[0]][puzzle.basis[1] + 1] = 0
61         puzzle.r.basis = (puzzle.basis[0], puzzle.basis[1] + 1)
62         puzzle.r.path += "r"
63         puzzle.r.hitungcost = puzzle.r.hitungCost() + len(puzzle.r.path)
64
65 #fungsi untuk menghitung cost tiap simpul
66 def hitungCost(puzzle):
67     hitung = 0
68     for i in range(4):
69         for j in range(4):
70             if (puzzle.mat[i][j] != Matrix.box[i][j]):
71                 hitung += 1
72     return hitung;
73
74 #fungsi untuk menerima input dari console
75 def inputFromConsole(puzzle):
76     for i in range(4):
77         temp = str(input())
78         puzzle.mat[i] = temp.split()
79         for j in range(4):
80             puzzle.mat[i][j] = int(puzzle.mat[i][j])
81             if (puzzle.mat[i][j] == 0):
82                 puzzle.basis = (i,j)
83             if (puzzle.mat[i][j] != Matrix.box[i][j]):
84                 puzzle.hitungcost += 1
85
86 #fungsi untuk mengecek apakah puzzle dapat diselesaikan atau tidak
87 def isSolve(puzzle):
88     hitung = 0
89     temp = []

```



```

90     for i in range(4):
91         for j in range(4):
92             if (puzzle.mat[i][j] == 0):
93                 temp.append(16)
94             else :
95                 temp.append(puzzle.mat[i][j])
96     for i in range(len(temp)):
97         for j in range(i+1, len(temp)):
98             if (temp[i] > temp[j]):
99                 hitung += 1
100     if ((puzzle.basis[0] + puzzle.basis[1]) % 2 == 1):
101         hitung += 1
102     print("\nKurang(i): ", hitung)
103     if (hitung % 2 == 0):
104         return True
105     else:
106         return False
107
108     #fungsi yang mengembalikan true jika matrix yang dimasukkan valid
109     def isInputValid(puzzle):
110         valid = [0 for i in range(16)]
111         for i in range(4):
112             for j in range(4):
113                 if (puzzle.mat[i][j] < 0 or puzzle.mat[i][j] > 15):
114                     return False
115                 else :
116                     valid[puzzle.mat[i][j]] += 1
117         for i in valid:
118             if (i == 0):
119                 return False
120         return True
121
122     #fungsi untuk membaca dari file
123     def readFile(puzzle, filename):
124         f = open(filename, "r")
125         for i in range(4):
126             isi = f.readline()
127             puzzle.mat[i] = isi.split()
128             for j in range(4):
129                 puzzle.mat[i][j] = int(puzzle.mat[i][j])
130                 if (puzzle.mat[i][j] == 0):
131                     puzzle.basis = (i,j)
132                 if (puzzle.mat[i][j] != Matrix.box[i][j]):
133                     puzzle.hitungcost += 1
134         f.close()

```

```

135
136     #fungsi untuk mencetak Matrix
137     def printMatrix(puzzle):
138         for i in puzzle.mat:
139             for j in range(4):
140                 print(i[j], end = " ")
141             print()
142         print("Total Simpul : ", Matrix.totalNodes)
143         print("Basis : ", puzzle.basis[0], puzzle.basis[1])
144         print("Path : ", puzzle.path)
145         print("Cost : ", puzzle.hitungcost)
146
147     #fungsi untuk mencetak basis
148     def printBasis(puzzle):
149         print(puzzle.basis[0] , puzzle.basis[1])
150
151     #fungsi untuk mencetak path
152     def printPath(puzzle):
153         print(puzzle.path)
154
155     #fungsi untuk mencetak cost
156     def printCost(puzzle):
157         print(puzzle.hitungcost)
158
159     #fungsi untuk mencetak total simpul
160     def printTotalNodes(puzzle):
161         print(Matrix.totalNodes)
162

```

### 3. function.py

```

You, 26 minutes ago | 1 author (You)
1  from puzzle import Matrix
2
3  #mengembalikan true jika puzzle A sama dengan puzzle B
4  def apakahSama(puzzleA, puzzleB):
5      for i in range(4):
6          for j in range(4):
7              if (puzzleA[i][j] != puzzleB[i][j]):
8                  return False
9      return True
10
11 #mengembalikan true jika puzzle sesuai target
12 def apakahSelesai(matrix, box):
13     return apakahSama(matrix, box)
14
15 #fungsi untuk sorting queue agar memenuhi priority queue
16 def sortingQueue(Queue):
17     for i in range(len(Queue)):
18         for j in range(i+1, len(Queue)):
19             if (Queue[i].hitungcost > Queue[j].hitungcost):
20                 temp = Queue[i]
21                 Queue[i] = Queue[j]
22                 Queue[j] = temp
23
24 #mengembalikan true jika dalam array dalam puzzle terdapat dalam matrix
25 def apakahContain(puzzle, matrix):
26     for i in puzzle:
27         if (apakahSama(i, matrix)):
28             return True
29     return False
30
31 #fungsi untuk menemukan path puzzle
32 def puzzleSolve(Puzz, calon, Queue):
33     awal = Puzz
34     calon.append(Puzz.mat)
35     end = apakahSelesai(awal.mat, Matrix.box)
36     while(not(end)):
37         #atur simpul atas
38         awal.aturUp()
39         if (awal.u != None):
40             if (apakahContain(calon, awal.u.mat)):
41                 Matrix.totalNodes -= 1
42                 awal.u = None
43             else :
44                 calon.append(awal.u.mat)
45                 Queue.append(awal.u)

```

```

46         if (apakahSelesai(awal.u.mat, Matrix.box)):
47             awal = awal.u
48             break
49
50     #atur bawah
51     awal.aturDown()
52     if (awal.d != None):
53         if (apakahContain(calon, awal.d.mat)):
54             Matrix.totalNodes -= 1
55             awal.d = None
56         else :
57             calon.append(awal.d.mat)
58             Queue.append(awal.d)
59             if (apakahSelesai(awal.d.mat, Matrix.box)):
60                 awal = awal.d
61                 break
62
63     #atur simpul kiri
64     awal.aturLeft()
65     if (awal.l != None):
66         if (apakahContain(calon, awal.l.mat)):
67             Matrix.totalNodes -= 1
68             awal.l = None
69         else :
70             calon.append(awal.l.mat)
71             Queue.append(awal.l)
72             if (apakahSelesai(awal.l.mat, Matrix.box)):
73                 awal = awal.l
74                 break
75
76     #atur kanan
77     awal.aturRight()
78     if (awal.r != None):
79         if (apakahContain(calon, awal.r.mat)):
80             Matrix.totalNodes -= 1
81             awal.r = None
82         else :
83             calon.append(awal.r.mat)
84             Queue.append(awal.r)
85             if (apakahSelesai(awal.r.mat, Matrix.box)):
86                 awal = awal.r
87                 break
88     sortingQueue(Queue)
89     awal = Queue.popleft()
90     return awal

```

## 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

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 : notSolve_1.txt

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

Kurang(i): 59
Puzzle tidak dapat diselesaikan karena fungsi kurang(i) 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
=====

```

```

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 : notSolve_2.txt

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

Kurang(i): 55
Puzzle tidak dapat diselesaikan karena fungsi kurang(i) 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

Inputan Puzzle :

6 5 2 4

9 1 3 8

10 0 7 15

13 14 12 11

Kurang(i): 34

Puzzle dapat diselesaikan :))

jumlah langkah : 17

jalur pencarian : luurdlurddrdldurd

Waktu eksekusi : 9.982694149017334 detik

jumlah simpul yang dibangkitkan : 1163

Apakah Anda ingin mencetak matriks langkah ? (y/n)

```

Langkah yang dilakukan : left
6 5 2 4
9 1 3 8
0 10 7 15
13 14 12 11

Langkah yang dilakukan : up
6 5 2 4
0 1 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : up
0 5 2 4
6 1 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : right
5 0 2 4
6 1 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : down
5 1 2 4
6 0 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : left
5 1 2 4
0 6 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : up
0 1 2 4
5 6 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : right
1 0 2 4
5 6 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : right
1 2 0 4
5 6 3 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : down
1 2 3 4
5 6 0 8
9 10 7 15
13 14 12 11

Langkah yang dilakukan : down
1 2 3 4
5 6 7 8
9 10 0 15
13 14 12 11

Langkah yang dilakukan : right
1 2 3 4
5 6 7 8
9 10 15 0
13 14 12 11

Langkah yang dilakukan : down
1 2 3 4
5 6 7 8
9 10 15 11
13 14 12 0

Langkah yang dilakukan : left
1 2 3 4
5 6 7 8
9 10 15 11
13 14 0 12

Langkah yang dilakukan : up
1 2 3 4
5 6 7 8
9 10 0 11
13 14 15 12

Langkah yang dilakukan : right
1 2 3 4
5 6 7 8
9 10 11 0
13 14 15 12

Langkah yang dilakukan : down
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 0

Apakah Anda ingin melanjutkan program ? (y/n)

```

#### 4. Kasus Uji 4 (Solve)

- input

```

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

```

- output





```
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

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

Kurang(i): 16
Puzzle dapat diselesaikan :))

jumlah langkah : 3
jalur pencarian : drd
Waktu eksekusi : 0.0029921531677246094 detik

jumlah simpul yang dibangkitkan : 1173

Apakah Anda ingin mencetak matriks langkah ? (y/n) y

Langkah yang dilakukan : down
1 2 3 4
5 6 7 8
9 10 0 11
13 14 15 12

Langkah yang dilakukan : right
1 2 3 4
5 6 7 8
9 10 11 0
13 14 15 12

Langkah yang dilakukan : down
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 0

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

## 5. Kasus Uji 5 (Solve)

- input

```

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

```

- 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

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

Kurang(i): 24
Puzzle dapat diselesaikan :))

jumlah langkah : 6
jalur pencarian : dddrrr
Waktu eksekusi : 0.0010120868682861328 detik

jumlah simpul yang dibangkitkan : 12

Apakah Anda ingin mencetak matriks langkah ? (y/n) y

Langkah yang dilakukan : down
1 2 3 4
0 6 7 8
5 10 11 12
9 13 14 15

Langkah yang dilakukan : down
1 2 3 4
5 6 7 8
9 10 11 12
0 13 14 15

Langkah yang dilakukan : right
1 2 3 4
5 6 7 8
9 10 11 12
13 0 14 15

Langkah yang dilakukan : right
1 2 3 4
5 6 7 8
9 10 11 12
13 14 0 15

Langkah yang dilakukan : right
1 2 3 4
5 6 7 8
9 10 11 12
13 14 15 0

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

#### 2. Data Uji Solveable

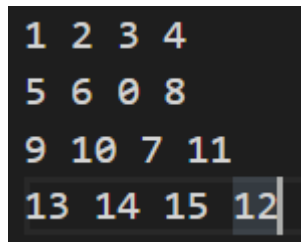
- solve\_1.txt

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

link berkas:

<https://drive.google.com/file/d/1lcJrgfzQqnADR28n6p6yDtULIPRa3blh/view?usp=sharing>

- solve\_2.txt

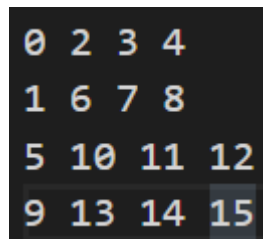


1	2	3	4
5	6	0	8
9	10	7	11
13	14	15	12

link berkas:

[https://drive.google.com/file/d/1-nULvQry5jNWH\\_0j2MVLN4Popk5-6ke\\_/view?usp=sharing](https://drive.google.com/file/d/1-nULvQry5jNWH_0j2MVLN4Popk5-6ke_/view?usp=sharing)

- solve\_3.txt



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

link berkas:

[https://drive.google.com/file/d/1Ktgsn\\_hiOQaZVvqH-ksi4Ly2ntJPYBEK/view?usp=sharing](https://drive.google.com/file/d/1Ktgsn_hiOQaZVvqH-ksi4Ly2ntJPYBEK/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>