Tugas Kecil 1 IF2211 Strategi Algoritma:

Implementasi Penyelesaian Puzzle-15 dengan Algoritma Branch & Bounce

Dibuat oleh:

Rahmat Rafid Akbar 13520090

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

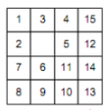
2022

DAFTAR ISI

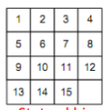
A. Algoritma Penyelesaian Puzzle-15

➢ PART 1 : CHECKING

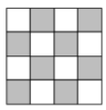
1. Kondisi awal papan puzzle (board) dicek kemungkinannya mencapai GOAL_STATE dengan menggunakan Bounding:
   - Teorema : Status tujuan hanya dapat dicapai dari status awal jika $\sum_{i=1}^{16} KURANG(i) + X$ bernilai genap.
   - X=1 jika sel kosong pada posisi awal ada pada sel yg diarsir



State awal



State akhir



   Nilai dari $KURANG(i)$ untuk setiap $i \in \{1..16\}$ diambil dari penentuan posisi angka i ($POSISI(i)$) terhadap angka j ($POSISI(j)$), yaitu: *(i dan j merupakan indeks posisi)*

$$KURANG(i) = \sum_{POSISI(j) = POSISI(i)+1}^{16} (x_j)$$

   dengan $x_j = \{0,1\}$. Note : $x_j = 1$ saat $(POSISI(j) > POSISI(i)) \wedge (j < i)$

   ✓ Jika nilai Bounding_Function genap, maka lanjutkan ke langkah 2,

   ✓ Jika nilai Bounding_Function ganjil, tampilkan pesan bahwa GOAL_STATE tidak mungkin tercapai.

➢ PART 2 : B&B Progress

2. Kondisi awal papan puzzle (board) yang merupkan simpul akar dimasukkan kedalam antrian prioritas pengecekan (PQ) dan merupakan simpul hidup dengan prioritas terendah (nilai 0).
3. Keluarkan elemen pertama antrian. Elemen ini menjadi simpul anak dari simpul sebelumnya jika bukan root.
   ✓ Jika simpul ini ($current\_board$) merupakan GOAL_STATE maka tampilkan path yang telah dilalui oleh simpul ini ($current\_board.prev\_path$) dan simpul ini merupakan simpul daun,
   ✓ Jika bukan, maka lanjutkan ke langkah 4.
4. Lakukan ekspansi $current\_board$ dengan semua kemungkinan simpul anaknya ($child\_board$) berdasarkan piroritas urutan pergeseran balok kosong (*blank-block*) : Atas, Kanan, Bawah, dan Kiri. Jangan lupa untuk menambahkan arah pergeseran baru ke dalam path yang dilalui setiap simpul anak.
5. Untuk setiap simpul anak dari langkah 4, dihitung nilai $cost\_total$ nya dengan rumus:

$$cost\_\_total = f\_cost + g\_cost$$

$$f\_cost = child\_depth \quad \& \quad g\_cost = \sum_{POSISI(i)=1}^{16} FIT(i)$$

   Dengan $FIT(i) = \{0,1\}$. Note : $FIT(i) = 1$ saat $i = POSISI(i)$

6. Setiap simpul anak dari langkah 5 dimasukkan ke dalam antiran prioritas pengecekan (PQ) dengan elemen berupa tuple $(cost\_total, child\_board)$. Lalu dilakukan pengurutan berdasarkan nilai $cost\_total$ secara menurun.
7. Berdasarkan kondisi antrian prioritas PQ:
   ✓ PQ tidak kosong, ulangi langkah ke 3 hingga dicapai GOAL_STATE,
   ✓ PQ kosong, tampilkan pesan bahwa GOAL_STATE tidak mungkin tercapai.

   ---> Sudah ditangani oleh Bounding_Function

B. Source Code

1. Board.py

Board.py > Board > G_cost

```python
class Board():
    ### Atribut Member
    # f_cost : private
    # blankX : private
    # blankY : private
    # prev_path : private

    ### CTOR
    def __init__(self, mat):
        self.__f_cost = 0
        self.__mat  = mat
        self.__blankX, self.__blankY = self.searchBLANK()
        self.__prev_path = ["none"]
    ### CCTOR
    # Di python memakai fungsi deepcopy()

    ### SETTER & GETTER
    # Matrix
    def setMatrix(self, mat):
        self.__mat = mat
    def getMatrix(self):
        return self.__mat

    # f_cost
    def setf_cost(self, f_cost):
        self.__f_cost = f_cost
    def getf_cost(self):
        return self.__f_cost

    # blank number
    def setBlank(self, x, y):
        self.__blankX = x
        self.__blankY = y
    def getBlank(self):
        return self.__blankX, self.__blankY

    # previous path
    def getprev_path(self):
        return self.__prev_path

    ### COST MODIFIER
    def incrementf_cost(self):
        self.__f_cost += 1

    def G_cost(self):
        sum = 0
        for i in range(4):
            for j in range(4):
                pos = (i)*4 + (j+1)
                if self.__mat[i][j] != 16:
                    if pos != self.__mat[i][j]:
                        sum += 1
        return sum

    def Total_cost(self):
        return self.__f_cost + self.G_cost()

    ### PREV PATH MODIFIER
    def continue_path(self, curr):
        self.__prev_path.append(curr)


    ### Matrix MODIFIER
    # blank finder
    def searchBLANK(self):
        x = -1
        y = -1
        for i in range(4):
            for j in range(4):
                if self.__mat[i][j] == 16:
                    x = i
                    y = j
        return x, y       Gleen, 9 hours ago • Bo

    # OUTPUT
    def printMatrix(self):
        print("/--.--.--.--\\")
        for i in range(4):
            print("|", end="")
            for j in range(4):
                bil = self.__mat[i][j]
                if bil != 16:
                    if bil < 10:
                        print(f" {bil}", end="")
                    else:
                        print(bil, end="")
                else:
                    print("  ", end="")
                if j!=3:
                    print("  ",end="")
                else:
                    print("|")
        print("\\--.--.--.--/")
```

```python
    # MOVE
    def move(self, dir):
        x = self.__blankX
        y = self.__blankY
        # print(x,y)
        xn = -1
        yn = -1
        if dir == "u":
            xn = x-1
            yn = y
            # print(xn, yn)
        elif dir == "r":
            xn = x
            yn = y+1
        elif dir == "d":
            xn = x+1
            yn = y
        elif dir == "l":
            xn = x
            yn = y-1
        temp = self.__mat[xn][yn]
        self.__mat[xn][yn] = 16
        self.__mat[x][y] = temp
        self.setBlank(xn,yn)



    ### CHECKING FUNCTION
    def KURANGI(self):
        KURANGI = [0 for i in range(16)]
        for i in range(4):
            for j in range(4):
                # pos1 dan pos2 dalam rentang 1<= x <=16
                pos1 = (i)*4 + (j+1)
                temp = self.__mat[i][j]
                # Cek selanjutnya : J < I dan POSISI(J) > POSISI(I)
                for k in range(4):
                    for l in range(4):
                        pos2 = (k)*4 + (l+1)
                        if pos2 > pos1:
                            if self.__mat[k][l] < temp:
                                KURANGI[temp-1] += 1
        return KURANGI
```

```python
    def BOUNDING(self):
        KURANGI = self.KURANGI()
        sum = 0
        for i in range(16):
            sum += KURANGI[i]

        # Jika blank ada pada daerah diarsir
        if (self.__blankX+self.__blankY)%2 == 1:
            sum += 1
        return sum, sum%2==0        # int, bool  = {jumlah KURANG(i)+X, apakah genap}


    def printKURANGI(self):
        KURANGI = self.KURANGI()
        print("Tabel Nilai KURANGI: ")
        print("|----------|")
        print("| ANGKA|  N |")
        print("|----------|")
        for i in range(16):
            n = KURANGI[i]
            if i+1 < 10:
                if n < 10:
                    print(f"|   {i+1}  |  {KURANGI[i]} |")
                else:
                    print(f"|   {i+1}  | {KURANGI[i]} |")
            else:
                if n < 10:
                    print(f"|  {i+1}  |  {KURANGI[i]} |")
                else:
                    print(f"|  {i+1}  | {KURANGI[i]} |")
        print("|----------|")

    def printPATH(self):
        for i in self.__prev_path:
            if i=="none":
                continue
            else:
                print(f"{i} ->",end=" ")
        print("GOAL")

    ### COMPARATOR (YA allah debug ini dari malem sampe 13 jam g tidur T_T)
    def __lt__(self, other):
        return True
    def __le__(self, other):
        return True
    def __gt__(self, other):
        return True
    def __ge__(self, other):
        return True
```

2. Other.py

```python
from copy import deepcopy
import random

def readFile(filename):
    filename = "test/" + filename +".txt"
    with open(filename, 'r') as f:
        mat = [[int(num) for num in line.split(' ')] for line in f if line.strip() != "" ]
    return mat

def readManual():
    print("Masukkan matrix 16 angka (dalam bentuk kotak dg spasi): ")
    mat = [[0 for j in range(4)] for i in range(4)]
    for i in range(4):
        word = input()
        mat[i] = [int(x) for x in word.split()]
    return mat

def bigRandomize():
    ran = random.sample(range(1,17),16)
    mat = [[0 for j in range(4)] for i in range(4)]
    for i in range(4):
        for j in range(4):
            mat[i][j] = int(ran[4*i + j])
    return mat
```

```python
def simpleRandomize():
    ran = random.sample(range(12,17),5)
    mat = [[0 for j in range(4)] for i in range(4)]
    for i in range(4):
        for j in range(4):
            pos = (i*4 + j)
            if pos < 11:
                mat[i][j] = pos+1
            else:
                mat[i][j] = int(ran[pos-11])
    return mat

def isGOAL_STATE(mat):
    for i in range(4):
        for j in range(4):
            pos = (i)*4 + (j+1)
            if (pos != mat[i][j]):
                return False
    return True

# Fungsi untuk menentukan simpul yang diexpand dan yang dimasukkan ke dalam antrian
def moveProgress(thisBoard, prev_Path, nextDirection, PQ, node_created):
    thisBoard.incrementf_cost()
    thisBoard.prev_path = deepcopy(prev_Path)
    thisBoard.continue_path(nextDirection)
    this_cost = thisBoard.Total_cost()
    # PQ.append( (this_cost, thisBoard) )                    # LIST IMPLEMENT
    PQ.put( (this_cost, thisBoard) )                         # Prio IMPLEMENT
    node_created.count = node_created.count + 1
```

3. Main.py

```python
from queue import PriorityQueue
from traceback import print_tb
from types import SimpleNamespace
from Board import *
from Others import *
import time
import msvcrt

def inputPilihan():
    looping = True
    time.sleep(0.8)
    print("(1) for file input        (3) for random input")
    print("(2) for manual input      (99) exit the program")
    print()
    op = int(input("Choose a number (1) or (2) or (3) or (99) : "))
    print()
    while not(op==1 or op==2 or op==3 or op==99):
        print("YOUR INPUT NOT VALID..!! Please reinput a number within range..!!")
        op = int(input("Choose a number (1) or (2) or (3) or (99) : "))
        print()
    if op == 1:
        file = input("Now, please input your filename (without extension) : ")
        Mtr = readFile(file)
    elif op == 2:
        Mtr = readManual()
    elif op == 3:
        Mtr = simpleRandomize()
    elif op ==99:
        print("Thanks a lot for using our porgram...!!  />..<\\")
        print("Have a nice day..!!")
        print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
        Mtr = []

    return Mtr

print("""
                                   <------>        ____ _______
     ______                        __           /_  |  ____|
    |  __  \  --->  ->  --->      |  |  ___   /__  |  |
    |  |  |  |     <---  <-  <---  |  |/ __ \ /__ |  |  |____
    |  |_/ _/__    __ ____________|  | |__|/     |++|_____  \\
    |   __/ |  |  |  |__   /|__   /|  |  __/__    |++|     |  |
    |  |    |  |  |  | /  /   /  / |  |\______|   |  |    /  |
    |  |    |   ++   |/  /_  /  /_ |  |___________|__|___/   /
    |__|     \_____//_____|/_____||_________________|______/
""")
time.sleep(1)
print("WELCOME to Puzzle-15 BOARD GAME...!!!")
print()
time.sleep(1.2)
print("This game will be solved by our automatic system,..")
time.sleep(1)
print("You only need to input the Initiate BOARD configuration,..")
time.sleep(1)
print("There are 3 ways to do it,.. let's check it out..!")
time.sleep(1)
print()
print("Press anykey to continue..!" )
trash = msvcrt.getch()
time.sleep(0.3)
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
time.sleep(0.6)
print("--> 1st Way : BOARD Configuration inputed by text file (.txt)")
print("              You can input the file name that stored at folder \"test\"")
print("              example : \"in1\" for in1.txt text file")
print()
time.sleep(1)
print("--> 2nd Way : BOARD Configuration manually inputed")
print("              You can input the matrix with 4x4 size that separated by space")
time.sleep(0.5)
print("""               example : 1  2  3  4
                          5  6  7  8
                          9 10 11 12
                         13 14 15 16 """)
print()
time.sleep(1)
print("--> 3rd Way : BOARD Configuration generated randomly by our system")
print("              Note : BOARD will be showed later")
print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

while True:
    Mtr = inputPilihan()
    if Mtr==[]:
        break

    board = Board(Mtr)
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
    print("--> HERE IS YOUR INITIAL STATE of PUZZLE-15 <--")
    board.printMatrix()
    print()
    time.sleep(0.8)
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
    print("--> HERE IS THE CHECKING TABLE of BOUND <--")
    board.printKURANGI()
    print()
    time.sleep(0.8)
```

```python
    bound, can = board.BOUNDING()
    print(f"Nilai dari sum_KURANG(i) + X adalah {bound}")
    print()
    print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")

    if not(can):
        print("--->>    !...WARNING...!    <<---")
        print("This Puzzle-15 can not be solved..! Because, its BOUNDING VALUE is ODD...")
    else:
        # DEKLARASI
        # PQ = []                                                 # LIST IMPLEMENT
        PQ = PriorityQueue()                                      # Prio IMPLEMENT
        node_created = SimpleNamespace()
        node_created.count = 0

        # INISIALIASI Puzzle awal di dalam antrian
        # PQ.append( (0,board) )                                  # LIST IMPLEMENT
        PQ.put( (0,board) )                                       # Prio IMPLEMENT
        node_created.count += 1

        start = time.time()
        # while not( isGOAL_STATE(PQ[0][1].getMatrix()) ):        # LIST IMPLEMENT
        while not( isGOAL_STATE(PQ.queue[0][1].getMatrix()) ):  # Prio IMPLEMENT

            # Expand simpul pertama yang ada pada antrian
            # currBoard = PQ[0][1]                                # LIST IMPLEMENT
            # PQ.pop(0)                                           # LIST IMPLEMENT
            currBoard = PQ.get()[1]                               # Prio IMPLEMENT

            xn, yn = currBoard.getBlank()

            # 1. Raising child node based on move Up
            if (xn != 0) and (currBoard.getprev_path()[-1] != "d"):
                new1 = deepcopy(currBoard)
                new1.move("u")
                moveProgress(new1,currBoard.getprev_path(),"u",PQ,node_created)

            # 2. Raising child node based on move Right
            if (yn != 3) and (currBoard.getprev_path()[-1] != "l"):
                new2 = deepcopy(currBoard)
                new2.move("r")
                moveProgress(new2,currBoard.getprev_path(),"r",PQ,node_created)

            # 3. Raising child node based on move Down
            if (xn != 3) and (currBoard.getprev_path()[-1] != "u"):
                new3 = deepcopy(currBoard)
                new3.move("d")
                moveProgress(new3,currBoard.getprev_path(),"d",PQ,node_created)
```

```python
            # 4. Raising child node based on move Down
            if (yn != 0) and (currBoard.getprev_path()[-1] != "r"):
                new4 = deepcopy(currBoard)
                new4.move("l")
                moveProgress(new4,currBoard.getprev_path(),"l",PQ,node_created)

        stop = time.time()
        timeTaken = (stop-start)

        print("-->    !...SOLUTION FOUND...!    <--")
        print()
        time.sleep(1)
        print("--> INITIAL STATE of PUZZLE-15 <--")
        board.printMatrix()
        print()

        i = -1
        # for direction in PQ[0][1].getprev_path():             # LIST IMPLEMENT
        for direction in PQ.queue[0][1].getprev_path():         # Prio IMPLEMENT
            if i == -1:
                i += 1
                continue
            i += 1
            print("~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~")
            print("--> Taken MOVE : ",end="")
            if direction == "u":
                print("ATAS")
                board.move("u")
            elif direction == "r":
                print("KANAN")
                board.move("r")
            elif direction == "d":
                print("BAWAH")
                board.move("d")
            elif direction == "l":
                print("KIRI")
                board.move("l")
            print()
            board.printMatrix()
            print()
            time.sleep(1)

        print(f"Program Execution time : {timeTaken} seconds")
        print(f"                       : {timeTaken*1000} miliseconds")
        print(f"Count of Move Taken : {i}")
        print("Path that able to take : ",end="")
        PQ.queue[0][1].printPATH()
        print(f"Count of Raised Node (living-node ever) : {node_created.count}")
```

C. CONTOH EKSEKUSI PROGRAM

1. Dataset 1 (test/in1.txt)

- Input

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

- Output:

```
Now, please input your filename (without extension) : in1
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS YOUR INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
| 1   2       4|
| 5   6   3   8|
| 9  10   7  11|
|13  14  15  12|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS THE CHECKING TABLE of BOUND <--
Tabel Nilai KURANGI:
|-----------|
| ANGKA|  N |
|-----------|
|   1  |  0 |
|   2  |  0 |
|   3  |  0 |
|   4  |  1 |
|   5  |  1 |
|   6  |  1 |
|   7  |  0 |
|   8  |  1 |
|   9  |  1 |
|  10  |  1 |
|  11  |  0 |
|  12  |  0 |
|  13  |  1 |
|  14  |  1 |
|  15  |  1 |
|  16  | 13 |
|-----------|

Nilai dari sum_KURANG(i) + X adalah 22

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-->     !...SOLUTION FOUND...!     <--

--> INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
| 1   2       4|
| 5   6   3   8|
| 9  10   7  11|
|13  14  15  12|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : KANAN

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

Program Execution time : 0.0010035037994384766 seconds
                       : 1.0035037994384766 miliseconds
Count of Move Taken : 4
Path that able to take : d -> d -> r -> d -> GOAL
Count of Raised Node (living-node ever) : 12

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

2. Dataset 2 (test/in2.txt)

- Input

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

- Output:

```
Now, please input your filename (without extension) : in2
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS YOUR INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
| 1   2   3   4|
| 5  10   6   7|
| 9      11   8|
|13  14  15  12|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS THE CHECKING TABLE of BOUND <--
Tabel Nilai KURANGI:
|-----------|
| ANGKA|  N |
|-----------|
|   1  |  0 |
|   2  |  0 |
|   3  |  0 |
|   4  |  0 |
|   5  |  0 |
|   6  |  0 |
|   7  |  0 |
|   8  |  0 |
|   9  |  1 |
|  10  |  4 |
|  11  |  1 |
|  12  |  0 |
|  13  |  1 |
|  14  |  1 |
|  15  |  1 |
|  16  |  6 |
|-----------|

Nilai dari sum_KURANG(i) + X adalah 16

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-->     !...SOLUTION FOUND...!     <--

--> INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
| 1   2   3   4|
| 5  10   6   7|
| 9      11   8|
|13  14  15  12|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : ATAS

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : KANAN

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : KANAN

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

Program Execution time : 0.0019769668579101562 seconds
                       : 1.9769668579101562 miliseconds
Count of Move Taken : 5 Moves
Path that able to take : u -> r -> r -> d -> d -> GOAL
Count of Raised Node (living-node ever) : 15 Nodes
```

3. Dataset 3 (test/in3.txt)

- Input

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

- Output:

```
Now, please input your filename (without extension) : in3
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS YOUR INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
|     2   3   4|
| 1   6   7   8|
| 5   9  10  12|
|13  14  11  15|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS THE CHECKING TABLE of BOUND <--
Tabel Nilai KURANGI:
|----------|
| ANGKA|  N |
|----------|
|   1  |  0 |
|   2  |  1 |
|   3  |  1 |
|   4  |  1 |
|   5  |  0 |
|   6  |  1 |
|   7  |  1 |
|   8  |  1 |
|   9  |  0 |
|  10  |  0 |
|  11  |  0 |
|  12  |  1 |
|  13  |  1 |
|  14  |  1 |
|  15  |  0 |
|  16  | 15 |
|----------|

Nilai dari sum_KURANG(i) + X adalah 24

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
-->     !...SOLUTION FOUND...!     <--

--> INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
|     2   3   4|
| 1   6   7   8|
| 5   9  10  12|
|13  14  11  15|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : KANAN

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

```
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : KANAN

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : BAWAH

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

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> Taken MOVE : KANAN

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

Program Execution time : 0.002021312713623047 seconds
                       : 2.021312713623047 miliseconds
Count of Move Taken : 6 Moves
Path that able to take : d -> d -> r -> r -> d -> r -> GOAL
Count of Raised Node (living-node ever) : 15 Nodes

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

4. Dataset 4 (test/in4.txt)

- Input

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

- Output:

```
Now, please input your filename (without extension) : in4
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS YOUR INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
|14   7   9    |
|11   6  15   8|
| 3   1   4  13|
| 2   5  10  12|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS THE CHECKING TABLE of BOUND <--
Tabel Nilai KURANGI:
|----------|
| ANGKA|  N |
|----------|
|   1  |  0 |
|   2  |  0 |
|   3  |  2 |
|   4  |  1 |
|   5  |  0 |
|   6  |  5 |
|   7  |  6 |
|   8  |  5 |
|   9  |  7 |
|  10  |  0 |
|  11  |  8 |
|  12  |  0 |
|  13  |  4 |
|  14  | 13 |
|  15  |  9 |
|  16  | 12 |
|----------|

Nilai dari sum_KURANG(i) + X adalah 73

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--->>    !...WARNING...!    <<---
This Puzzle-15 can not be solved..! Because, its BOUNDING VALUE is ODD...

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

5. Dataset 5 (test/in5.txt)

- Input

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

- Output:

```
Now, please input your filename (without extension) : in5
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS YOUR INITIAL STATE of PUZZLE-15 <--
/--..--..--..--\
|     9  15   4|
|11  13   8   6|
| 2   1   5  14|
|10   7   3   1|
\--..--..--..--/

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--> HERE IS THE CHECKING TABLE of BOUND <--
Tabel Nilai KURANGI:
|-----------|
| ANGKA|  N |
|-----------|
|   1  |  0 |
|   2  |  2 |
|   3  |  1 |
|   4  |  4 |
|   5  |  2 |
|   6  |  5 |
|   7  |  2 |
|   8  |  7 |
|   9  |  9 |
|  10  |  3 |
|  11  |  9 |
|  12  |  0 |
|  13  |  9 |
|  14  |  4 |
|  15  | 13 |
|  16  | 15 |
|-----------|

Nilai dari sum_KURANG(i) + X adalah 85

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
--->>    !...WARNING...!    <<---
This Puzzle-15 can not be solved..! Because, its BOUNDING VALUE is ODD...

~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

D. LINK GITHUB

Source code dan laporan dapat diakses melalui:

https://github.com/Amike31/Tucil_Stima_3

E. Tabel Penilaian Umum

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