

Laporan Tugas Kecil 3 IF2211 Strategi Algoritma

Semester 2 2021/2022

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



Hafidz Nur Rahman Ghozali

13520117

Sekolah Teknik Elektro dan Informatika

Teknik Informatika

2022

1. Algoritma *Branch and Bound*

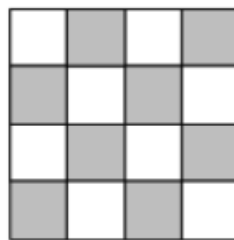
Algoritma *Branch and Bound* merupakan algoritma untuk persoalan maksimasi atau minimasi. Prinsip pembangkitan simpul pada algoritma ini mirip dengan algoritma *Breadth First Search*, namun terdapat tambahan pengecekan berupa ongkos atau *cost/bound* dari setiap simpul yang dibangkitkan. Sebagai perbandingan, dalam algoritma BFS, simpul yang akan diekspansi berikutnya akan terurut berdasarkan urutan pembangkitannya. Sedangkan pada algoritma *Branch and Bound*, simpul yang akan diekspansi berikutnya merupakan simpul yang masih hidup dan memiliki ongkos paling minimum atau maksimum, tidak bergantung pada urutan pembangkitan simpulnya. Pada algoritma ini terdapat pemangkasan pada jalur yang dianggap tidak lagi mengarah pada simpul solusi. Hal ini dilakukan dengan “mematikan” simpul yang masih hidup yang memiliki ongkos tidak lebih baik dari ongkos terbaik sejauh ini.

Persoalan 15-Puzzle merupakan persoalan minimasi yang bertujuan untuk menentukan langkah paling minimal dalam membentuk susunan matriks sehingga memiliki susunan yang diharapkan. Sebelum menyelesaikan persoalan 15-Puzzle, terlebih dahulu dilakukan pengecekan apakah puzzle tersebut dapat diselesaikan atau tidak dengan menghitung nilai

$$\sum_{i=1}^{16} Kurang(i) + x \dots (1)$$

dengan

- $Kurang(i)$ merupakan banyaknya ubin bernomor j sedemikian sehingga $j < i$ dan $posisi(j) > posisi(i)$. $posisi(i)$ merupakan posisi ubin bernomor i pada susunan yang diperiksa.
- x merupakan nilai berdasarkan letak ubin kosong pada susunan. x bernilai 1 jika ubin kosong terletak pada sel yang diarsir, selain itu x bernilai 0.



Apabila perhitungan pada (1) bernilai genap, maka susunan puzzle dapat diselesaikan. Apabila ganjil, susunan tidak dapat diselesaikan.

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

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

$$\sum_{i=1}^{16} Kurang(i) + X = 15 + 1 = 16$$

i	Kurang (i)
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	1
9	1
10	1
11	0
12	0
13	1
14	1
15	1
16	9

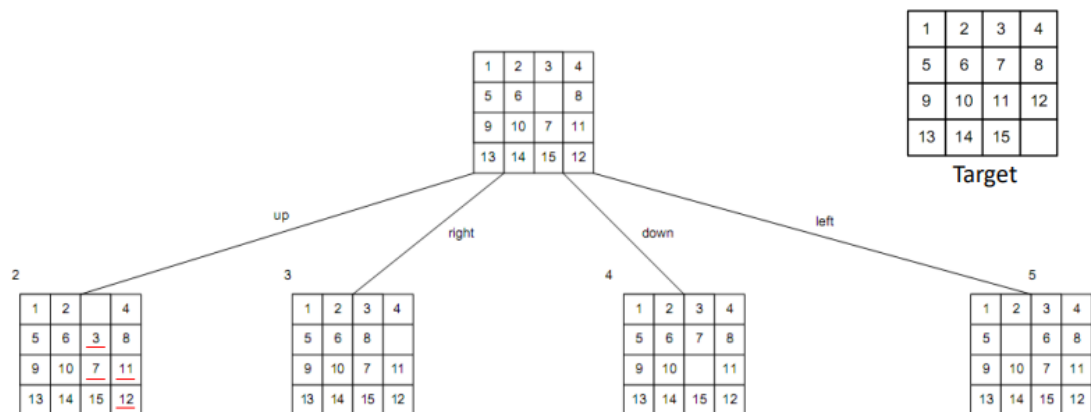
Pada penyelesaian 15-Puzzle ini, setiap simpul baru yang dibangkitkan akan dilakukan perhitungan ongkos dari susunan matriks yang diperoleh. Ongkos dari setiap simpul merupakan suatu taksiran yang dihitung berdasarkan rumus

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

dengan

- $\hat{c}(i)$ merupakan taksiran ongkos untuk simpul i
- $f(i)$ merupakan panjang lintasan dari simpul akar ke simpul i
- $\hat{g}(i)$ merupakan taksiran panjang terpendek dari simpul p ke simpul solusi pada upapohon yang akarnya p . Dalam program ini, nilai taksiran merupakan jumlah ubin tidak kosong yang tidak terdapat pada susunan akhir (*misplaced tiles*).

Contoh perhitungan ongkos pada kasus di bawah ini



- Pada simpul 2, $f(2)$ bernilai 1 dan $\hat{g}(2)$ bernilai 4 karena ubin nomor 3,7,11, dan 12 tidak terletak pada susunan akhir. Sehingga $\hat{c}(2)$ bernilai 5.
- Pada simpul 3, $f(3)$ bernilai 1 dan $\hat{g}(3)$ bernilai 4 karena ubin nomor 7,8,11, dan 12 tidak terletak pada susunan akhir. Sehingga $\hat{c}(3)$ bernilai 5.
- Pada simpul 4, $f(4)$ bernilai 1 dan $\hat{g}(4)$ bernilai 2 karena ubin nomor 11 dan 12 tidak terletak pada susunan akhir. Sehingga $\hat{c}(4)$ bernilai 3.
- Pada simpul 5, $f(5)$ bernilai 1 dan $\hat{g}(5)$ bernilai 4 karena ubin nomor 6,7,11, dan 12 tidak terletak pada susunan akhir. Sehingga $\hat{c}(5)$ bernilai 5.

Karena persoalan 15-Puzzle merupakan persoalan minimasi, simpul-simpul yang dibangkitkan tersebut akan dimasukkan ke dalam *queue* yang urutannya menaik berdasarkan ongkos setiap simpul. Sehingga simpul berikutnya yang akan diekspansi memiliki ongkos paling minimum.

2. *Input* dan *Output* Program

Di dalam program ini, sel kosong disimbolkan dengan bilangan 16

2.1. Puzzle bisa diselesaikan dengan masukan dari sebuah file

Input:

```
22Moves.txt
1 10 2 4
3 13 6 7
16 5 9 8
14 15 11 12
```

Output:

```
Pilih metode masukan data:
1. Matriks Random
2. Input File
Masukkan pilihan: 2
Masukkan nama file: test/22Moves.txt

Konfigurasi game:
+-----+-----+-----+-----+
| 1 | 10 | 2 | 4 |
+-----+-----+-----+-----+
| 3 | 13 | 6 | 7 |
+-----+-----+-----+-----+
|   | 5 | 9 | 8 |
+-----+-----+-----+-----+
| 14 | 15 | 11 | 12 |
+-----+-----+-----+-----+

=====
Nilai fungsi kurang(i):
 1 : 0      9 : 1
 2 : 0     10 : 8
 3 : 0     11 : 0
 4 : 1     12 : 0
 5 : 0     13 : 7
 6 : 1     14 : 2
 7 : 1     15 : 2
 8 : 0     16 : 7
           X : 0
Total Kurang + X : 30
=====
Puzzle dapat diselesaikan
=====
```

Langkah ke- 1 : RIGHT

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

Langkah ke- 6 : RIGHT

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

Langkah ke- 2 : UP

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

Langkah ke- 7 : UP

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

Langkah ke- 3 : LEFT

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

Langkah ke- 8 : LEFT

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

Langkah ke- 4 : DOWN

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

Langkah ke- 9 : UP

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

Langkah ke- 5 : RIGHT

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

Langkah ke- 10 : RIGHT

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

Langkah ke- 11 : DOWN

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

Langkah ke- 16 : UP

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

Langkah ke- 12 : DOWN

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

Langkah ke- 17 : RIGHT

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

Langkah ke- 13 : DOWN

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

Langkah ke- 18 : UP

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

Langkah ke- 14 : LEFT

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

Langkah ke- 19 : RIGHT

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

Langkah ke- 15 : LEFT

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

Langkah ke- 20 : RIGHT

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

Langkah ke- 21 : DOWN

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

Langkah ke- 22 : DOWN

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

Jumlah simpul yang dibangkitkan: 13646

Total Pergeseran: 22

Waktu eksekusi: 234.375 ms

2.2. Puzzle tidak bisa diselesaikan dengan masukan random oleh program

```
Pilih metode masukan data:
1. Matriks Random
2. Input File
Masukkan pilihan: 1

Konfigurasi game:

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

=====
Nilai fungsi kurang(i):
      1 : 0      9 : 0
      2 : 1     10 : 5
      3 : 2     11 : 4
      4 : 2     12 : 6
      5 : 0     13 : 5
      6 : 0     14 : 4
      7 : 0     15 : 11
      8 : 4     16 : 5
              X : 0
Total Kurang + X : 49
=====
Puzzle tidak dapat diselesaikan
=====
```


3. Checklist Program

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

4. Kode Program

4.1. File node.py

File ini berisi kelas Node yang digunakan untuk melambangkan simpul yang menyimpan matriks, rute, node *parent*, kedalaman, jumlah *misplaced tiles*, *cost*, dan posisi sel kosong.

```
from enum import Enum

class Direction(Enum):
    RIGHT = 1
    DOWN = 2
    LEFT = 3
    UP = 4

class Node:
    # Constructor
    def __init__(self, matriks, misplaced, emptyRow, emptyCol, route = [],
parent = None, depth = 0):
        self.matriks = matriks
        self.route = route
        self.parent = parent
        self.depth = depth
        self.misplaced = misplaced
        self.cost = misplaced + depth
        self.emptyRow = emptyRow
        self.emptyCol = emptyCol

    # Method print untuk mencetak info node, untuk pengecekan
    def print(self):
        print("Route: ", self.route)
        print("Depth: " + str(self.depth))
        print("Misplaced: " + str(self.misplaced))
        print("Cost: " + str(self.cost))
        self.printMatriks()

    # Method untuk mengecek apakah merupakan goal node
```

```

def isGoal(self):
    return self.misplaced == 0

# Method untuk membangkitkan anak-anak dari suatu node
def getChildren(self):
    children = []

    # Iterasi pada arah yang dapat dilalui
    for direction in Direction:
        # Menyalin informasi dari parent
        childMat = [[num for num in row] for row in self.matriks]
        childRoute = self.route + [direction]
        newEmptyRow = self.emptyRow
        newEmptyCol = self.emptyCol

        # Membandingkan arah yang dipilih
        match direction:
            case Direction.UP:
                if (self.emptyRow > 0 and (self.route == [] or
self.route[-1] != Direction.DOWN)):
                    newEmptyRow -= 1
            case Direction.DOWN:
                if (self.emptyRow < 3 and (self.route == [] or
self.route[-1] != Direction.UP)):
                    newEmptyRow += 1
            case Direction.LEFT:
                if (self.emptyCol > 0 and (self.route == [] or
self.route[-1] != Direction.RIGHT)):
                    newEmptyCol -= 1
            case Direction.RIGHT:
                if (self.emptyCol < 3 and (self.route == [] or
self.route[-1] != Direction.LEFT)):
                    newEmptyCol += 1

        # Jika arah yang dipilih valid, tambahkan ke children
        if (self.emptyRow != newEmptyRow or self.emptyCol != newEmptyCol):
            # Menghitung misplaced pada susunan matriks baru
            misplaced = 0
            if (childMat[newEmptyRow][newEmptyCol] == newEmptyRow*4 +
newEmptyCol + 1): misplaced += 1
            if (childMat[newEmptyRow][newEmptyCol] == self.emptyRow*4 +
self.emptyCol + 1): misplaced -= 1
            childMat[self.emptyRow][self.emptyCol],
childMat[newEmptyRow][newEmptyCol] = childMat[newEmptyRow][newEmptyCol],
childMat[self.emptyRow][self.emptyCol]
            # Menambahkan node baru ke children
            children.append(Node(childMat, self.misplaced + misplaced,
newEmptyRow, newEmptyCol, childRoute, self, self.depth + 1))

```

```

        return children

# Method untuk mencetak matriks
def printMatriks(self):
    print("\t+-----+-----+-----+-----+")
    for i in range(4):
        print("\t|", end="")
        for j in range(4):
            if (self.matriks[i][j] == 16):
                print("    |", end = "")
            elif (self.matriks[i][j] > 9):
                print("  " + str(self.matriks[i][j]) + " |", end="")
            else:
                print("   " + str(self.matriks[i][j]) + " |", end="")
        print("\n\t+-----+-----+-----+-----+")

# Method untuk mencetak langkah terakhir suatu node
def getLastRoute(self):
    match self.route[-1]:
        case Direction.UP:
            return "UP"
        case Direction.DOWN:
            return "DOWN"
        case Direction.LEFT:
            return "LEFT"
        case Direction.RIGHT:
            return "RIGHT"
    return None

# Method overloading '<' untuk membantu heapq
def __lt__(self, other):
    if (self.cost == other.cost):
        return self.depth > other.depth
    return self.cost < other.cost

```

4.2. File queue_node.py

File ini berisi kelas QueueNode untuk menyimpan *queue of Node*. Penulis menggunakan *library* heapq untuk mempercepat jalannya program. *Library* heapq cocok diterapkan pada kasus ini karena terdapat banyak sekali operasi *push* atau penambahan elemen ke dalam *queue*, sehingga memerlukan tipe data dengan kompleksitas *insertion*-nya paling optimal.

```

import heapq as hq

class QueueNode:
    # Constructor
    def __init__(self):

```

```

        self.queue = []
        # ubah queue menjadi min-heap
        heapq.heapify(self.queue)

    # Method untuk menambahkan node ke dalam queue
    def push(self, node):
        heapq.heappush(self.queue, node)

    # Method untuk mengambil node dengan cost paling minimum
    def pop(self):
        return heapq.heappop(self.queue)

    # Method untuk menghitung panjang dari queue
    def length(self):
        return len(self.queue)

    # Method untuk mengecek apakah queue kosong
    def isEmpty(self):
        return self.length() == 0

    # Method untuk menghapus node yang memiliki cost yang lebih besar dengan
    # cost node yang diberikan
    def kill(self, cost):
        i = 0
        while (i < self.length()):
            if (self.queue[i].cost > cost):
                self.queue.pop(i)
            else:
                i += 1
        heapq.heapify(self.queue)

    # Method untuk mencetak queue
    def print(self):
        for node in self.queue:
            node.print()

```

4.3. File main.py

File ini berisi kelas Puzzle sekaligus main program pada permainan 15-Puzzle ini. Terdapat beberapa method seperti getInput untuk mendapatkan input dari pengguna sehingga pengguna bisa memilih susunan matriks secara random atau berasal dari sebuah file.

```

from os.path import exists
import numpy as np
from queue_node import QueueNode
from node import Node
import time

class Puzzle:

```

```

# Constructor
def __init__(self):
    self.matriks = self.getInput()
    self.EmptyRow = 0
    self.EmptyCol = 0
    self.cost = 0
    self.kurang = [0 for i in range(16)]
    self.totalKurang = 0
    self.solution = None
    self.route = []
    self.visited = set()
    self.banyakSimpul = 1
    self.kalkulasi()

# Method untuk mendapatkan input matriks
def getInput(self):
    print("Pilih metode masukan data:")
    print("1. Matriks Random")
    print("2. Input File")

    # Pilihan input
    choice = int(input("Masukkan pilihan: "))
    while (choice < 1 or choice > 2):
        print("Pilihan tidak valid")
        choice = int(input("Masukkan pilihan: "))
    matriks = []

    if choice == 1:
        # Input matriks secara random
        temp = np.random.permutation(np.arange(1, 17))
        temp = np.ndarray.tolist(temp)

        # Menyalin list ke dalam matriks
        matriks.append(temp[0:4])
        matriks.append(temp[4:8])
        matriks.append(temp[8:12])
        matriks.append(temp[12:16])
    else:
        # Input matriks dari file
        filename = input("Masukkan nama file: ")
        while not exists(filename):
            print("File tidak ditemukan")
            filename = input("Masukkan nama file: ")

        f = open(filename, "r")
        # Menyalin isi file ke dalam matriks
        matriks = [[int(num) for num in line.strip("\n").split(" ")] for
line in f]

```

```

        return matriks

# Method untuk melakukan kalkulasi sum kurang dan cost matriks awal
def kalkulasi(self):
    # Perulangan untuk mencari sel kosong dan menghitung cost
    for i in range(4):
        for j in range(4):
            if self.matriks[i][j] == 16:
                self.EmptyRow = i
                self.EmptyCol = j
            else:
                self.cost += (self.matriks[i][j] != i*4 + j + 1)

    # Perulangan untuk mencari tabel kurang
    for i in range(16):
        temp = 0
        for j in range(i+1,16):
            if (self.matriks[j//4][j%4] < self.matriks[i//4][i%4]):
                temp += 1
        self.kurang[self.matriks[i//4][i%4] - 1] = temp

# Method untuk menyelesaikan puzzle
def solve(self):
    # Inisialisasi root node
    start = Node(self.matriks, self.cost, self.EmptyRow, self.EmptyCol)
    queue = QueueNode()
    queue.push(start)

    # Perulangan untuk mencari solusi
    while (not queue.isEmpty()):
        current = queue.pop()
        # print("Current: ", current.depth, " misplaced:",
current.misplaced, " dsan cost: ", current.cost)

        # Mengecek apakah sebuah matriks sudah pernah dicek atau belum
        if (np.array(current.matriks).tobytes() in self.visited):
            continue
        else:
            self.visited.add(np.array(current.matriks).tobytes())

        if (current.isGoal()):
            # Cek apakah node ini merupakan goal node
            self.solution = current
            # print("Solusi ditemukan dengan depth: ", current.depth, "
misplaced:", current.misplaced, " dan cost: ", current.cost)
            queue.kill(current.cost)
        else:

```

```

        # Jika bukan goal node, bangkitkan anak-anak berdasarkan node yang
dicek
        child = current.getChildren()
        self.banyakSimpul += len(child)
        for node in child:
            if (self.solution == None or node.cost <
self.solution.cost):
                queue.push(node)

# Method untuk mencari rute dari sebuah solusi yang ditemukan
def findRoute(self):
    result = self.solution
    while (result.parent != None):
        self.route.append(result)
        result = result.parent
    self.route.reverse()

# Method untuk mencetak langkah-langkah penyelesaian
def printResult(self):
    for i in range(len(self.route)):
        print()
        print("Langkah ke-", i+1, ": ", self.route[i].getLastRoute())
        self.route[i].printMatriks()

    print("\nJumlah simpul yang dibangkitkan: ", self.banyakSimpul)

# Method untuk mencetak matriks
def printMatriks(self):
    print("\t+-----+-----+-----+-----+")
    for i in range(4):
        print("\t|", end="")
        for j in range(4):
            if (self.matriks[i][j] == 16):
                print("    |", end = "")
            elif (self.matriks[i][j] > 9):
                print("  " + str(self.matriks[i][j]) + " |", end="")
            else:
                print("  " + str(self.matriks[i][j]) + "  |", end="")
        print("\n\t+-----+-----+-----+-----+")

# ===== MAIN PROGRAM ===== #
if __name__ == "__main__":
    game = Puzzle()
    print("\nKonfigurasi game:")
    game.printMatriks()

    # Mencetak tabel kurang
    print("\n=====")

```

```

print("Nilai fungsi kurang(i):")
for i in range(8):
    print("\t", i+1, ": ", game.kurang[i], "\t", i+9, ": ",
game.kurang[i+8])
    print("\t\t X :", (game.EmptyRow+game.EmptyCol)%2)

# check Kurang(i) + X
print("Total Kurang + X : " + str(sum(game.kurang) +
(game.EmptyRow+game.EmptyCol)%2))
print("=====")

# Jika Kurang(i) + X ganjil maka tidak ada solusi
if (game.totalKurang%2 == 1):
    print("\tPuzzle tidak dapat diselesaikan")
    print("=====\n")
else:
    print("\tPuzzle dapat diselesaikan")
    print("=====\n")
    # Memulai timer
    start = time.process_time_ns()

    # Solve puzzle
    game.solve()

    # Mencari route dari solusi yang ditemukan
    game.findRoute()

    # Mengakhiri timer
    end = time.process_time_ns()
    duration = end - start

    # Menampilkan hasil
    game.printResult()
    print("Total Pergeseran: ", len(game.route))
    print("Waktu eksekusi: ", duration/1000000, " ms")

```

5. Berkas Teks Instansiasi 5 buah persoalan 15-Puzzle
Catatan: Sel kosong disimbolkan dengan bilangan 16

5.1. 10Moves.txt

```

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

```


Hasil jalannya program

```
Pilih metode masukan data:  
1. Matriks Random  
2. Input File  
Masukkan pilihan: 2  
Masukkan nama file: test/10Moves.txt
```

Konfigurasi game:

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

=====

Nilai fungsi kurang(i):

```
1 : 0      9 : 1  
2 : 0     10 : 0  
3 : 0     11 : 2  
4 : 1     12 : 2  
5 : 1     13 : 1  
6 : 4     14 : 0  
7 : 0     15 : 0  
8 : 0     16 : 10
```

X : 0

Total Kurang + X : 22

=====

Puzzle dapat diselesaikan

=====

Langkah ke- 3 : DOWN

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

Langkah ke- 4 : RIGHT

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

Langkah ke- 5 : DOWN

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

Langkah ke- 1 : UP

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

Langkah ke- 6 : LEFT

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

Langkah ke- 2 : RIGHT

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

Langkah ke- 7 : LEFT

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

```

Langkah ke- 8 : DOWN
+-----+-----+-----+-----+
| 1 | 2 | 3 | 4 |
+-----+-----+-----+-----+
| 5 | 6 | 7 | 8 |
+-----+-----+-----+-----+
| 9 | 10 | 11 | 12 |
+-----+-----+-----+-----+
| 13 |  | 14 | 15 |
+-----+-----+-----+-----+

Langkah ke- 9 : RIGHT
+-----+-----+-----+-----+
| 1 | 2 | 3 | 4 |
+-----+-----+-----+-----+
| 5 | 6 | 7 | 8 |
+-----+-----+-----+-----+
| 9 | 10 | 11 | 12 |
+-----+-----+-----+-----+
| 13 | 14 |  | 15 |
+-----+-----+-----+-----+

Langkah ke- 10 : RIGHT
+-----+-----+-----+-----+
| 1 | 2 | 3 | 4 |
+-----+-----+-----+-----+
| 5 | 6 | 7 | 8 |
+-----+-----+-----+-----+
| 9 | 10 | 11 | 12 |
+-----+-----+-----+-----+
| 13 | 14 | 15 |  |
+-----+-----+-----+-----+

Jumlah simpul yang dibangkitkan: 26
Total Pergeseran: 10
Waktu eksekusi: 15.625 ms

```

5.2. 15Moves.txt

```

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

```

Hasil jalannya program

```
Pilih metode masukan data:
1. Matriks Random
2. Input File
Masukkan pilihan: 2
Masukkan nama file: test/15Moves.txt

Konfigurasi game:
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 |  | 8 |
+-----+-----+-----+-----+
| 13 | 10 | 7 | 11 |
+-----+-----+-----+-----+
| 14 | 5 | 15 | 12 |
+-----+-----+-----+-----+

=====
Nilai fungsi kurang(i):
      1 : 0      9 : 3
      2 : 1     10 : 2
      3 : 1     11 : 1
      4 : 1     12 : 0
      5 : 0     13 : 5
      6 : 4     14 : 2
      7 : 1     15 : 1
      8 : 2     16 : 9
              X : 1
Total Kurang + X : 34
=====
      Puzzle dapat diselesaikan
=====

Langkah ke- 1 : DOWN
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 | 7 | 8 |
+-----+-----+-----+-----+
| 13 | 10 |  | 11 |
+-----+-----+-----+-----+
| 14 | 5 | 15 | 12 |
+-----+-----+-----+-----+

Langkah ke- 2 : LEFT
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 | 7 | 8 |
+-----+-----+-----+-----+
| 13 |  | 10 | 11 |
+-----+-----+-----+-----+
| 14 | 5 | 15 | 12 |
+-----+-----+-----+-----+

Langkah ke- 3 : DOWN
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 | 7 | 8 |
+-----+-----+-----+-----+
| 13 | 5 | 10 | 11 |
+-----+-----+-----+-----+
| 14 |  | 15 | 12 |
+-----+-----+-----+-----+

Langkah ke- 4 : LEFT
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 | 7 | 8 |
+-----+-----+-----+-----+
| 13 | 5 | 10 | 11 |
+-----+-----+-----+-----+
|  | 14 | 15 | 12 |
+-----+-----+-----+-----+

Langkah ke- 5 : UP
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 | 7 | 8 |
+-----+-----+-----+-----+
|  | 5 | 10 | 11 |
+-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
+-----+-----+-----+-----+

Langkah ke- 6 : RIGHT
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 | 9 | 7 | 8 |
+-----+-----+-----+-----+
| 5 |  | 10 | 11 |
+-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
+-----+-----+-----+-----+

Langkah ke- 7 : UP
+-----+-----+-----+-----+
| 2 | 6 | 3 | 4 |
+-----+-----+-----+-----+
| 1 |  | 7 | 8 |
+-----+-----+-----+-----+
| 5 | 9 | 10 | 11 |
+-----+-----+-----+-----+
| 13 | 14 | 15 | 12 |
+-----+-----+-----+-----+
```

```

Langkah ke- 8 :  UP
+-----+-----+-----+-----+
|  2  |      |  3  |  4  |
+-----+-----+-----+-----+
|  1  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  5  |  9  | 10  | 11  |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 9 :  LEFT
+-----+-----+-----+-----+
|      |  2  |  3  |  4  |
+-----+-----+-----+-----+
|  1  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  5  |  9  | 10  | 11  |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 10 : DOWN
+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |
+-----+-----+-----+-----+
|      |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  5  |  9  | 10  | 11  |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 11 : DOWN
+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |
+-----+-----+-----+-----+
|  5  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|      |  9  | 10  | 11  |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 12 : RIGHT
+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |
+-----+-----+-----+-----+
|  5  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  9  |      | 10  | 11  |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 13 : RIGHT
+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |
+-----+-----+-----+-----+
|  5  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  9  | 10  |      | 11  |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 14 : RIGHT
+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |
+-----+-----+-----+-----+
|  5  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  9  | 10  | 11  |      |
+-----+-----+-----+-----+
| 13  | 14  | 15  | 12  |
+-----+-----+-----+-----+

Langkah ke- 15 : DOWN
+-----+-----+-----+-----+
|  1  |  2  |  3  |  4  |
+-----+-----+-----+-----+
|  5  |  6  |  7  |  8  |
+-----+-----+-----+-----+
|  9  | 10  | 11  | 12  |
+-----+-----+-----+-----+
| 13  | 14  | 15  |      |
+-----+-----+-----+-----+

Jumlah simpul yang dibangkitkan: 296
Total Pergeseran: 15
Waktu eksekusi: 0.0 ms

```

5.3. 26Moves.txt

```

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

```

Hasil jalannya program

```
Pilih metode masukan data:
1. Matriks Random
2. Input File
Masukkan pilihan: 2
Masukkan nama file: test/26Moves.txt

Konfigurasi game:
+-----+-----+-----+-----+
| 2 | 1 | 6 | 3 |
+-----+-----+-----+-----+
| 9 | 5 | 8 | 4 |
+-----+-----+-----+-----+
| 13 | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 10 |   | 14 | 11 |
+-----+-----+-----+-----+

=====
Nilai fungsi kurang(i):
      1 : 0      9 : 4
      2 : 1     10 : 0
      3 : 0     11 : 0
      4 : 0     12 : 3
      5 : 1     13 : 4
      6 : 3     14 : 1
      7 : 0     15 : 3
      8 : 2     16 : 2
              X : 0
Total Kurang + X : 24
=====
Puzzle dapat diselesaikan
=====

Langkah ke- 1 : LEFT
+-----+-----+-----+-----+
| 2 | 1 | 6 | 3 |
+-----+-----+-----+-----+
| 9 | 5 | 8 | 4 |
+-----+-----+-----+-----+
| 13 | 12 | 7 | 15 |
+-----+-----+-----+-----+
|   | 10 | 14 | 11 |
+-----+-----+-----+-----+

Langkah ke- 2 : UP
+-----+-----+-----+-----+
| 2 | 1 | 6 | 3 |
+-----+-----+-----+-----+
| 9 | 5 | 8 | 4 |
+-----+-----+-----+-----+
|   | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 13 | 10 | 14 | 11 |
+-----+-----+-----+-----+

Langkah ke- 3 : UP
+-----+-----+-----+-----+
| 2 | 1 | 6 | 3 |
+-----+-----+-----+-----+
|   | 5 | 8 | 4 |
+-----+-----+-----+-----+
| 9 | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 13 | 10 | 14 | 11 |
+-----+-----+-----+-----+

Langkah ke- 4 : RIGHT
+-----+-----+-----+-----+
| 2 | 1 | 6 | 3 |
+-----+-----+-----+-----+
| 5 |   | 8 | 4 |
+-----+-----+-----+-----+
| 9 | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 13 | 10 | 14 | 11 |
+-----+-----+-----+-----+

Langkah ke- 5 : UP
+-----+-----+-----+-----+
| 2 |   | 6 | 3 |
+-----+-----+-----+-----+
| 5 | 1 | 8 | 4 |
+-----+-----+-----+-----+
| 9 | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 13 | 10 | 14 | 11 |
+-----+-----+-----+-----+

Langkah ke- 6 : RIGHT
+-----+-----+-----+-----+
| 2 | 6 |   | 3 |
+-----+-----+-----+-----+
| 5 | 1 | 8 | 4 |
+-----+-----+-----+-----+
| 9 | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 13 | 10 | 14 | 11 |
+-----+-----+-----+-----+

Langkah ke- 7 : RIGHT
+-----+-----+-----+-----+
| 2 | 6 | 3 |   |
+-----+-----+-----+-----+
| 5 | 1 | 8 | 4 |
+-----+-----+-----+-----+
| 9 | 12 | 7 | 15 |
+-----+-----+-----+-----+
| 13 | 10 | 14 | 11 |
+-----+-----+-----+-----+
```

Langkah ke- 8 : DOWN

	2		6	
	5		1	
	9		12	
	13		10	

Langkah ke- 13 : UP

	2		6	
			1	
	5		9	
	13		10	

Langkah ke- 9 : LEFT

	2		6	
	5		1	
	9		12	
	13		10	

Langkah ke- 14 : RIGHT

	2		6	
	1			
	5		9	
	13		10	

Langkah ke- 10 : DOWN

	2		6	
	5		1	
	9		12	
	13		10	

Langkah ke- 15 : UP

	2			
	1		6	
	5		9	
	13		10	

Langkah ke- 11 : LEFT

	2		6	
	5		1	
	9			
	13		10	

Langkah ke- 16 : LEFT

			2	
	1		6	
	5		9	
	13		10	

Langkah ke- 12 : LEFT

	2		6	
	5		1	
			9	
	13		10	

Langkah ke- 17 : DOWN

	1		2	
			6	
	5		9	
	13		10	

Langkah ke- 18 : DOWN

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

Langkah ke- 19 : RIGHT

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

Langkah ke- 20 : DOWN

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

Langkah ke- 21 : RIGHT

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

Langkah ke- 22 : RIGHT

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

Langkah ke- 23 : UP

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

Langkah ke- 24 : LEFT

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

Langkah ke- 25 : DOWN

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

Langkah ke- 26 : RIGHT

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

Jumlah simpul yang dibangkitkan: 108496

Total Pergeseran: 26

Waktu eksekusi: 1593.75 ms

5.4. Unsolveable1.txt

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

Hasil jalannya program

```
Pilih metode masukan data:
1. Matriks Random
2. Input File
Masukkan pilihan: 2
Masukkan nama file: test/Unsolveable1.txt

Konfigurasi game:
+---+---+---+---+
| 5 | 4 | 9 | 14 |
+---+---+---+---+
| 1 | 7 | 8 | 11 |
+---+---+---+---+
| 13 | 10 | 15 |   |
+---+---+---+---+
| 6 | 3 | 2 | 12 |
+---+---+---+---+

=====
Nilai fungsi kurang(i):
      1 : 0          9 : 6
      2 : 0         10 : 3
      3 : 1         11 : 4
      4 : 3         12 : 0
      5 : 4         13 : 5
      6 : 2         14 : 10
      7 : 3         15 : 4
      8 : 3         16 : 4
                  X : 1
Total Kurang + X : 53
=====
Puzzle tidak dapat diselesaikan
=====
```


5.5. Unsolveable2.txt

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

Hasil jalannya program

```
Pilih metode masukan data:
1. Matriks Random
2. Input File
Masukkan pilihan: 2
Masukkan nama file: test/Unsolveable2.txt

Konfigurasi game:

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

=====
Nilai fungsi kurang(i):
      1 : 0      9 : 7
      2 : 0     10 : 7
      3 : 2     11 : 0
      4 : 1     12 : 0
      5 : 2     13 : 2
      6 : 0     14 : 0
      7 : 5     15 : 14
      8 : 5     16 : 11
                X : 1
Total Kurang + X : 57
=====
      Puzzle tidak dapat diselesaikan
=====
```

6. Alamat kode program:

<https://github.com/hafidznrg/Stima-15Puzzle>

Referensi:

Rinaldi M., Nur Ulfa M., Masayu L. K. 2021. *Algoritma Brand and Bound (Bagian 1)*