

Laporan Tugas Kecil 1 IF2211 Strategi Algoritma
Semester II tahun 2021/2022

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



Dipersiapkan oleh:

Nama: Jeremy Rionaldo Pasaribu

NIM: 13520082

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

Daftar Isi

Daftar Isi	1
BAB 1: Cara Kerja Program <i>Branch and Bound</i>	2
BAB 2: Kode Program dalam Bahasa Python	3
BAB 3: Hasil <i>Screenshot</i> Pengujian.....	12
BAB 4: Berkas Teks Studi Kasus	19
Lampiran	20

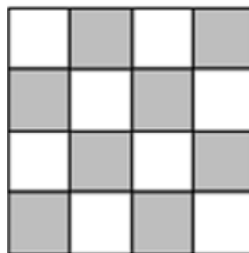
BAB 1: Cara Kerja Program *Branch and Bound*

1.1. Pemisalan yang Digunakan

- Antrian *prioq_bnb* merupakan antrian berjenis PriorityQueue dengan prioritas *cost* terkecil pada simpul. *Cost* yang dimaksud adalah:

$c(i) = f(i) + g(i)$ <p>$c(i)$ = ongkos untuk simpul i $f(i)$ = ongkos mencapai simpul i dari akar $g(i)$ = ongkos mencapai simpul <i>goal</i> dari simpul i</p>
--

- Matriks awal 15-puzzle merupakan matriks yang dibaca dari file teks.
- Simpul solusi merupakan simpul yang memiliki *cost* dari simpul itu tersendiri sampai ke *goal* ($g(i)$) bernilai 0
- X bernilai 1 jika ubin kosong berada dalam daerah ubin arsiran. Jika ubin kosong tidak berada pada daerah ubin arsiran, X akan bernilai 0. Gambar ubin arsiran sebagai berikut:



- Fungsi Kurang(i) yang merupakan banyaknya ubin bernomor j sedemikian sehingga $j < i$ dan $POSISI(j) > POSISI(i)$. $POSISI(i)$ = posisi ubin bernomor i pada susunan yang diperiksa.

1.2. Cara Kerja Program *Branch and Bound*

- Program akan mengecek terlebih dahulu apakah matriks awal 15-puzzle dapat diselesaikan dengan perhitungan $\sum_{i=0}^{16} \text{Kurang}(i) + X$. Jika perhitungan $\sum_{i=0}^{16} \text{Kurang}(i) + X$ bernilai ganjil, 15-puzzle tidak dapat diselesaikan dan program akan berhenti.
- Program akan membangkitkan semua anak-anaknya dari simpul matriks awal 15-puzzle lalu memasukkan anak-anaknya ke dalam antrian *prioq_bnb*. Jika simpul akar merupakan simpul solusi, maka program akan berhenti.
- Program akan mengambil elemen simpul depan antrian *prioq_bnb* dan dimisalkan sebagai *current_node*
- Jika simpul *current_node* merupakan simpul solusi, maka program akan berhenti.
- Jika simpul *current_node* bukan merupakan simpul solusi, program akan membangkitkan semua simpul anak-anaknya. Jika state simpul *current_node* sudah pernah dibangkitkan sebelumnya, kembali ke langkah 2.
- Program menghitung *cost* setiap anak simpul yang telah dibangkitkan lalu menyimpannya ke dalam antrian *prioq_bnb*

BAB 2: Kode Program dalam Bahasa Python

2.1. PuzzleSolver.py

```
from heapq import heappop, heappush
import time

class PuzzleNode:
    """
    Kelas PuzzleNode untuk menyimpan matriks puzzle,
    depth, g_cost, dan parent dari PuzzleNode
    """

    # instansiasi variabel g_cost
    g_cost = 999

    def __init__(self, puzzle_board, depth, parent):
        self.puzzle_board = puzzle_board.copy()
        self.depth = depth
        self.parent = parent

def ReadFileGUI(file_name):
    """
    Membaca file matriks puzzle dan memasukkannya ke dalam list
    puzzle_board (untuk GUI)
    """
    puzzle_board = []
    for raw_lines in open(file_name, 'r'):
        lines = raw_lines.replace("\n", "").split()
        for line in lines:
            puzzle_board.append(int(line))
    return puzzle_board

def ReadFile():
    """
    Membaca file matriks puzzle dan memasukkannya ke dalam list
    puzzle_board
    """
    while True:
        try:
            puzzle_board = []
            filename = input("Masukkan nama file (tanpa .txt): ")
            for raw_lines in open("../test/" + filename + ".txt", 'r'):
```

```
        lines = raw_lines.replace("\n", "").split()
        for line in lines:
            puzzle_board.append(int(line))
    return puzzle_board
except:
    print("Nama file tidak ditemukan, ulangi")

def PrintPuzzle(puzzle_board):
    """
    Mencetak list puzzle_board ke layar dalam bentuk puzzle matriks
    """
    for i in range(16):
        print("|" + str(puzzle_board[i]).ljust(2), end="")
        if i % 4 == 3:
            print("|")

def GetX(idx):
    """
    Mendapatkan nilai X untuk penjumlahan Kurang(i) + X
    """
    list_x = [1, 3, 4, 6, 9, 11, 12, 14]
    if idx in list_x:
        return 1
    else:
        return 0

def GetListKurangAndSum(puzzle_board):
    """
    Mendapatkan semua nilai Kurang(i) dan total nilai Kurang(i) + X
    """
    list_kurang = []
    kurang_sum = 0
    for i in range(1, 17):
        count = 0
        i_idx = puzzle_board.index(i)
        for j in range(i_idx+1, 16):
            if(i > puzzle_board[j]):
                count += 1
        list_kurang.append(count)
        kurang_sum += count
    kurang_sum += GetX(i_idx)
    return kurang_sum, list_kurang
```

```
def Get_g_cost(puzzle_board):
    """
    Menghitung jumlah ubin tidak kosong dalam list puzzle_board yang
    tidak berada pada tempat sesuai susunan akhir (goal state)
    """
    cost = 0
    for i in range(15):
        if(i+1 != puzzle_board[i]):
            cost += 1
    return cost

def Swap(puzzle_board, idx1, idx2):
    """
    Menukar nilai idx1 dan idx2 pada list puzzle_board
    """
    puzzle_board[idx1], puzzle_board[idx2] = puzzle_board[idx2],
    puzzle_board[idx1]

def GetPossibleNodes(node_count, prioq_bnb, puzzle_node):
    """
    Mengembalikan semua simpul yang dapat dibangkitkan dari puzzle_node
    serta mengembalikan jumlah simpul yang dibangkitkan
    """
    empty_idx = puzzle_node.puzzle_board.index(16)
    puzzle_board = puzzle_node.puzzle_board
    new_depth = puzzle_node.depth + 1

    # Menambahkan simpul state ketika ubin kosong digeser ke arah kanan
    if ((empty_idx + 1) % 4 != 0):
        node_count += 1
        new_node = PuzzleNode(puzzle_board, new_depth, puzzle_board)
        Swap(new_node.puzzle_board, empty_idx, empty_idx + 1)
        new_node.g_cost = Get_g_cost(new_node.puzzle_board)
        heappush(prioq_bnb, (new_depth + new_node.g_cost, node_count,
new_node))

    # Menambahkan simpul state ketika ubin kosong digeser ke arah kiri
    if(empty_idx % 4 != 0):
        node_count += 1
        new_node = PuzzleNode(puzzle_board, new_depth, puzzle_board)
        Swap(new_node.puzzle_board, empty_idx, empty_idx - 1)
        new_node.g_cost = Get_g_cost(new_node.puzzle_board)
        heappush(prioq_bnb, (new_depth + new_node.g_cost, node_count,
```

```
new_node))

# Menambahkan simpul state ketika ubin kosong digeser ke arah atas
if(empty_idx - 3 > 0):
    node_count += 1
    new_node = PuzzleNode(puzzle_board, new_depth, puzzle_board)
    Swap(new_node.puzzle_board, empty_idx, empty_idx - 4)
    new_node.g_cost = Get_g_cost(new_node.puzzle_board)
    heappush(prioq_bnb, (new_depth + new_node.g_cost, node_count,
new_node))

# Menambahkan simpul state ketika ubin kosong digeser ke arah bawah
if(empty_idx + 3 < 15):
    node_count += 1
    new_node = PuzzleNode(puzzle_board, new_depth, puzzle_board)
    Swap(new_node.puzzle_board, empty_idx, empty_idx + 4)
    new_node.g_cost = Get_g_cost(new_node.puzzle_board)
    heappush(prioq_bnb, (new_depth + new_node.g_cost, node_count,
new_node))

    return node_count

def BranchAndBoundSolve(puzzle_board):
    """
    Algoritma Branch And Bound untuk mencari solusi dari 15 Puzzle
    """
    prioq_bnb = []
    node_path = {}
    node_count = 0
    current_node = PuzzleNode(puzzle_board, 0, "ROOT")
    g_cost = Get_g_cost(puzzle_board)
    if g_cost != 0:
        node_count = GetPossibleNodes(node_count, prioq_bnb,
current_node)
        current_node = heappop(prioq_bnb)[2]

    # Dilakukan iterasi sampai semua ubin yang tidak kosong dalam list
puzzle_board
    # sesuai pada tempatnya (goal node)
    while g_cost != 0:
        if(str(current_node.puzzle_board) not in node_path):
            node_path[str(current_node.puzzle_board)
] = current_node.parent
```

```
        node_count = GetPossibleNodes(node_count, prioq_bnb,
current_node)
        current_node = heappop(prioq_bnb)[2]
        g_cost = current_node.g_cost

    if (str(current_node.puzzle_board) != str(puzzle_board)):
        node_path[str(current_node.puzzle_board)] = current_node.parent
    return current_node.puzzle_board, node_path, node_count
```

2.2. CLI.py

```
from PuzzleSolver import *

# Driver utama CLI program
if __name__ == "__main__":

    # Membaca file
    puzzle_board = ReadFile()
    print("Program dalam progress...")

    # Mendapatkan perhitungan waktu awal program
    runtime_start = time.time()

    # Mengecek apakah puzzle_board dapat diselesaikan
    kurang_sum, list_kurang = GetListKurangAndSum(puzzle_board)
    if kurang_sum % 2 != 0:
        message = "Puzzle tidak bisa diselesaikan!"
    else:
        message = "Puzzle bisa diselesaikan, solusi sebagai berikut:"
        puzzle, node_path, node_count =
BranchAndBoundSolve(puzzle_board)

    # Mendapatkan perhitungan waktu akhir program
    runtime_end = time.time()

    # Output program
    print("\nMatriks posisi awal: ")
    PrintPuzzle(puzzle_board)
    print("\nNilai Kurang(i): ")
    for i in range(16):
        print(f"Kurang({i+1}) = {list_kurang[i]}")
    print(f"Total Kurang(i) + X = {kurang_sum}\n")
```



```
print(message)

if(kurang_sum % 2 == 0):
    node_solution = [puzzle]
    while str(puzzle) != str(puzzle_board):
        node_solution.insert(0, node_path[str(puzzle)])
        puzzle = node_path[str(puzzle)]
    node_solution.pop(0)
    print("Matriks posisi awal: ")
    PrintPuzzle(puzzle_board)
    for i in range(len(node_solution)):
        print(f"\nMatriks langkah ke-{i+1}: ")
        PrintPuzzle(node_solution[i])
    print("\nWaktu eksekusi: {:.4f} detik".format(
        runtime_end-runtime_start))
    print(f"Jumlah simpul yang dibangkitkan: {node_count}")
```

2.3. GUI.py

```
import PySimpleGUI as sg
from PuzzleSolver import *
import time

# Pembuatan Layout GUI
sg.theme('DarkAmber')
kurang_layout_1 = [
    [sg.Text(key="-KURANG(1)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(2)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(3)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(4)-", font=("Casadia Code", 11))]
]

kurang_layout_2 = [
    [sg.Text(key="-KURANG(5)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(6)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(7)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(8)-", font=("Casadia Code", 11))]
]

kurang_layout_3 = [
    [sg.Text(key="-KURANG(9)-", font=("Casadia Code", 11))],
    [sg.Text(key="-KURANG(10)-", font=("Casadia Code", 11))],

```

```
[sg.Text(key="-KURANG(11)-", font=("Cascadia Code", 11))],
[sg.Text(key="-KURANG(12)-", font=("Cascadia Code", 11))]
]

kurang_layout_4 = [
    [sg.Text(key="-KURANG(13)-", font=("Cascadia Code", 11))],
    [sg.Text(key="-KURANG(14)-", font=("Cascadia Code", 11))],
    [sg.Text(key="-KURANG(15)-", font=("Cascadia Code", 11))],
    [sg.Text(key="-KURANG(16)-", font=("Cascadia Code", 11))]
]

input_output_layout = [
    [sg.Text(text="15-Puzzle Solver", justification="center",
        font=("Cascadia Code", 20))],
    [sg.Text(text="Masukkan matriks puzzle:", font=("Cascadia Code",
12))],
    [
        sg.FileBrowse(button_text="Pilih File",
            file_types=(("Text Files", "*.txt"),),
font=("Cascadia Code", 10)),
        sg.In(size=(20, 1), enable_events=True, font=(
            "Cascadia Code", 10), key="-FILE INPUT-")
    ],
    [
        sg.Button(button_text="Solve", font=(
            "Cascadia Code", 10), key="-SOLVE-")
    ],
    [sg.Column(kurang_layout_1),
    sg.Column(kurang_layout_2),
    sg.Column(kurang_layout_3),
    sg.Column(kurang_layout_4)],
    [sg.Text(key="-KURANG(i)+X-", font=("Cascadia Code", 11))],
    [sg.Text(key="-WAKTU EKSEKUSI-", font=("Cascadia Code", 11))],
    [sg.Text(key="-NODE BANGKIT-", font=("Cascadia Code", 11))]
]

puzzle_layout_1 = [
    [sg.Image(key="-SLOT1-", filename="./assets/block1.png")],
    [sg.Image(key="-SLOT5-", filename="./assets/block5.png")],
    [sg.Image(key="-SLOT9-", filename="./assets/block9.png")],
    [sg.Image(key="-SLOT13-", filename="./assets/block13.png")]
]
```

```
puzzle_layout_2 = [
    [sg.Image(key="-SLOT2-", filename="./assets/block2.png")],
    [sg.Image(key="-SLOT6-", filename="./assets/block6.png")],
    [sg.Image(key="-SLOT10-", filename="./assets/block10.png")],
    [sg.Image(key="-SLOT14-", filename="./assets/block14.png")]
]

puzzle_layout_3 = [
    [sg.Image(key="-SLOT3-", filename="./assets/block3.png")],
    [sg.Image(key="-SLOT7-", filename="./assets/block7.png")],
    [sg.Image(key="-SLOT11-", filename="./assets/block11.png")],
    [sg.Image(key="-SLOT15-", filename="./assets/block15.png")]
]

puzzle_layout_4 = [
    [sg.Image(key="-SLOT4-", filename="./assets/block4.png")],
    [sg.Image(key="-SLOT8-", filename="./assets/block8.png")],
    [sg.Image(key="-SLOT12-", filename="./assets/block12.png")],
    [sg.Image(key="-SLOT16-", filename="./assets/block16.png")]
]

layout = [
    [
        sg.Column(puzzle_layout_1),
        sg.Column(puzzle_layout_2),
        sg.Column(puzzle_layout_3),
        sg.Column(puzzle_layout_4),
        sg.Column(""),
        sg.Column(""),
        sg.Column(""),
        sg.Column(input_output_layout),
    ]
]

# Driver Utama GUI
window = sg.Window("15-Puzzle Solver", layout)
while True:
    event, values = window.read()
    if event == "-FILE INPUT-":
        try:
            puzzle_board = ReadFileGUI(values["-FILE INPUT-"])
            for i in range(16):
                window[f"-SLOT{i+1}-"].update(
```

```
        filename="./assets/block" + str(puzzle_board[i]) +
        ".png")
    except:
        sg.popup("File tidak ditemukan")

    if event == "-SOLVE-":
        # Mendapatkan perhitungan waktu awal program
        runtime_start = time.time()

        # Mengecek apakah puzzle_board dapat diselesaikan
        kurang_sum, list_kurang = GetListKurangAndSum(puzzle_board)
        if kurang_sum % 2 != 0:
            sg.popup("Puzzle ini tidak dapat diselesaikan")
        else:
            puzzle, node_path = BranchAndBoundSolve(puzzle_board)
            # Mendapatkan perhitungan waktu akhir program
            runtime_end = time.time()

            # Output program dan pergerakan Puzzle
            node_solution = [puzzle]
            while str(puzzle) != str(puzzle_board):
                node_solution.insert(0, node_path[str(puzzle)])
                puzzle = node_path[str(puzzle)]
            for i in range(16):
                window[f"-KURANG({i+1})-"].update(f"Kurang({i+1}) =
{list_kurang[i]}")
            window["-KURANG(i)+X-"].update(
                f"Total Kurang(i) + X = {kurang_sum}")
            window["-WAKTU EKSEKUSI-"].update("Waktu eksekusi: {:.4f}
detik".format(
                runtime_end-runtime_start))
            window["-NODE BANGKIT-"].update(
                f"Jumlah simpul yang dibangkitkan: {len(node_path)}")
            for i in range(len(node_solution)):
                for j in range(len(node_solution[i])):
                    window[f"-SLOT{j+1}-"].update(
                        filename="assets/block" +
                        str(node_solution[i][j]) + ".png")
                    window.read(timeout=550)
            if event == "Exit" or event == sg.WIN_CLOSED:
                break
    window.close()
```

BAB 3: Hasil Screenshot Pengujian

3.1. Studi Kasus 1 Solvable (TC1.txt)

a. Input/Output CLI (Command Line Interface)

```
Windows PowerShell
PS D:\Github\StimaTucil3\src> py CLI.py
Masukkan nama file (tanpa .txt): TC1
Program dalam progress...

Matriks posisi awal:
|1 |2 |3 |4 |
|5 |6 |16|8 |
|9 |10|7 |11|
|13|14|15|12|

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

Puzzle bisa diselesaikan, solusi sebagai berikut:
Matriks posisi awal:
|1 |2 |3 |4 |
|5 |6 |16|8 |
|9 |10|7 |11|
|13|14|15|12|

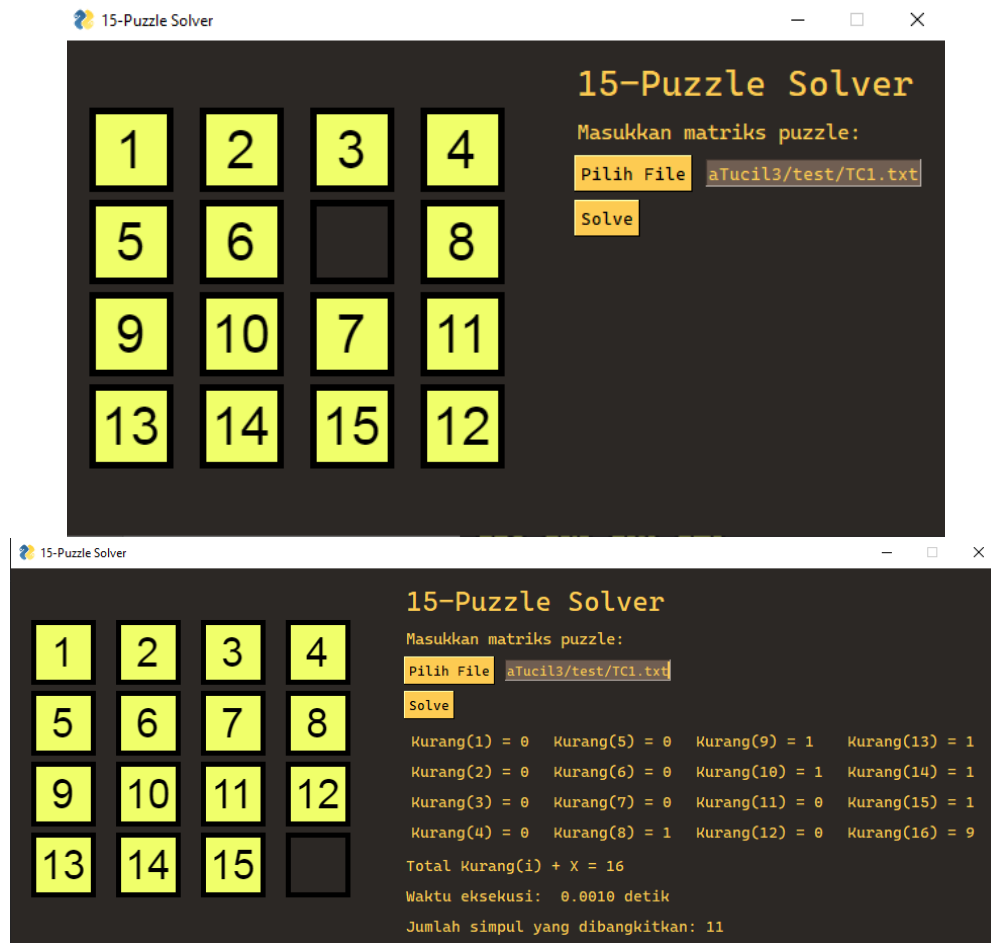
Matriks langkah ke-1:
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|16|11|
|13|14|15|12|

Matriks langkah ke-2:
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|16|
|13|14|15|12|

Matriks langkah ke-3:
|1 |2 |3 |4 |
|5 |6 |7 |8 |
|9 |10|11|12|
|13|14|15|16|

Waktu eksekusi: 0.0000 detik
Jumlah simpul yang dibangkitkan: 11
PS D:\Github\StimaTucil3\src> |
```

b. Input/Output GUI (Graphical User Interface)



3.2. Studi Kasus 2 Solvable (TC2.txt)

a. Input/Output CLI (Command Line Interface)

```

Windows PowerShell
PS D:\Github\StimaTucil3\src> py CLI.py
Masukkan nama file (tanpa .txt): TC2
Program dalam progress ...

Matriks posisi awal:
|2 |16|3 |4 |
|1 |5 |10|7 |
|9 |14|6 |8 |
|13|11|12|15|

Nilai Kurang(i):
Kurang(1) = 0
Kurang(2) = 1
Kurang(3) = 1
Kurang(4) = 1
Kurang(5) = 0
Kurang(6) = 0
Kurang(7) = 1
Kurang(8) = 0
Kurang(9) = 2
Kurang(10) = 4
Kurang(11) = 0
Kurang(12) = 0
Kurang(13) = 2
Kurang(14) = 5
Kurang(15) = 0
Kurang(16) = 14
Total Kurang(i) + X = 32

Puzzle bisa diselesaikan, solusi sebagai berikut:
Matriks posisi awal:
|2 |16|3 |4 |
|1 |5 |10|7 |
|9 |14|6 |8 |
|13|11|12|15|
    
```

```

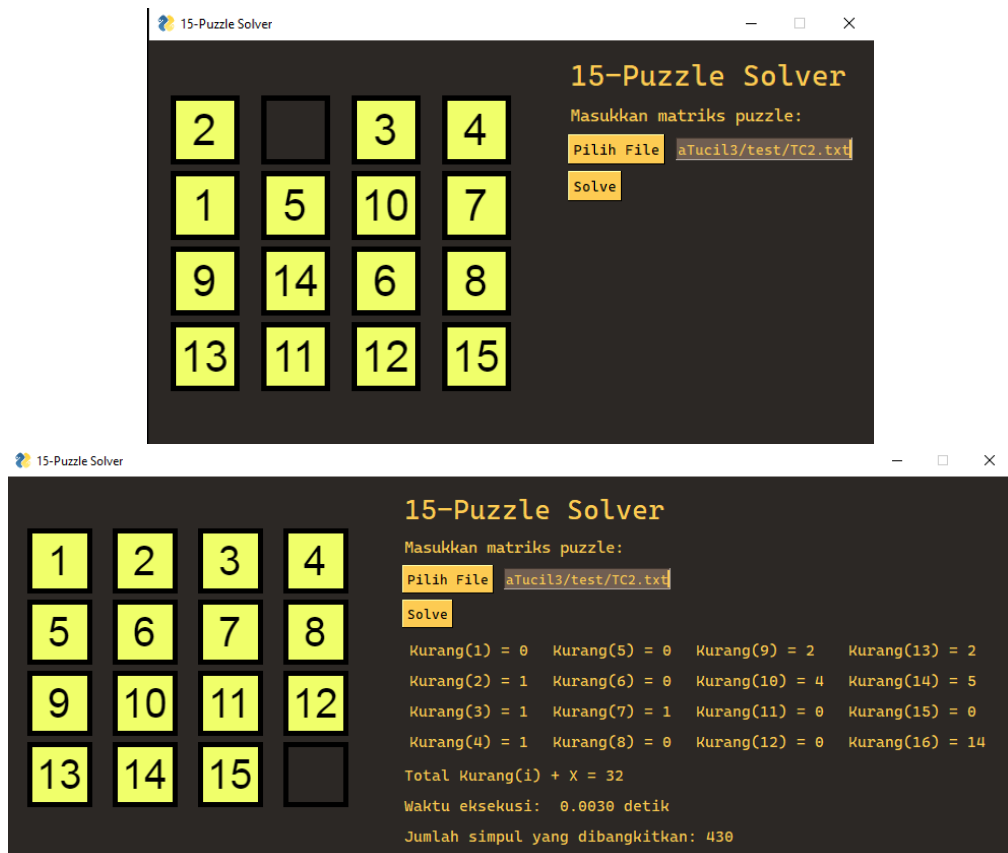
Matriks langkah ke-14:
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 |10|11|12|
|13|14|16|15|

Matriks langkah ke-15:
| 1 | 2 | 3 | 4 |
| 5 | 6 | 7 | 8 |
| 9 |10|11|12|
|13|14|15|16|

Waktu eksekusi: 0.0040 detik
Jumlah simpul yang dibangkitkan: 430
PS D:\Github\StimaTucil3\src>

```

b. Input/Output GUI (Graphical User Interface)



3.3. Studi Kasus 3 Solvable (TC3.txt)

a. Input/Output CLI (Command Line Interface)

```
Windows PowerShell
PS D:\Github\StimaTucil3\src> py CLI.py
Masukkan nama file (tanpa .txt): TC3
Program dalam progress ...

Matriks posisi awal:
|5|3|16|2|
|9|1|6|4|
|13|10|7|15|
|14|12|8|11|

Nilai Kurang(i):
Kurang(1) = 0
Kurang(2) = 1
Kurang(3) = 2
Kurang(4) = 0
Kurang(5) = 4
Kurang(6) = 1
Kurang(7) = 0
Kurang(8) = 0
Kurang(9) = 5
Kurang(10) = 2
Kurang(11) = 0
Kurang(12) = 2
Kurang(13) = 5
Kurang(14) = 3
Kurang(15) = 4
Kurang(16) = 13
Total Kurang(i) + X = 42

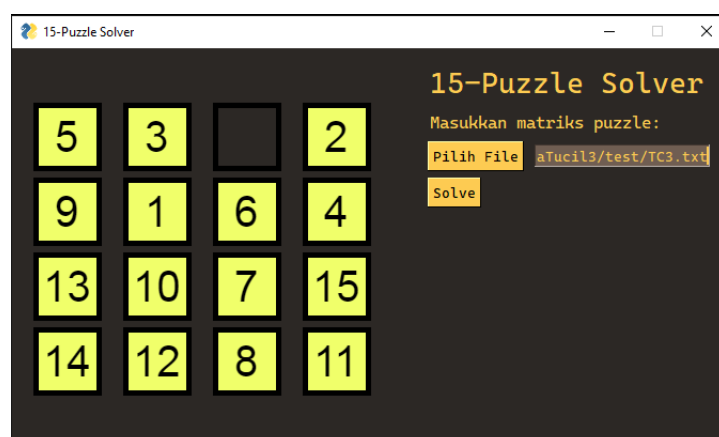
Puzzle bisa diselesaikan, solusi sebagai berikut:
Matriks posisi awal:
|5|3|16|2|
|9|1|6|4|
|13|10|7|15|
|14|12|8|11|

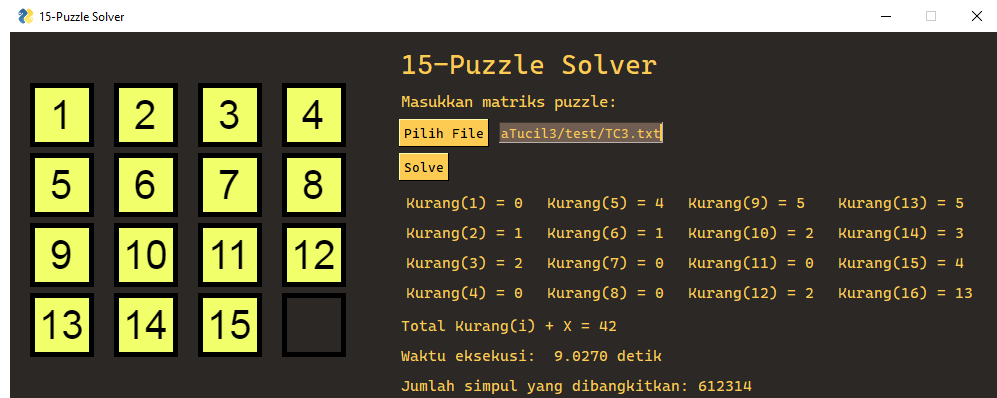
Matriks langkah ke-27:
|1|2|3|4|
|5|6|7|8|
|9|10|11|16|
|13|14|15|12|

Matriks langkah ke-28:
|1|2|3|4|
|5|6|7|8|
|9|10|11|12|
|13|14|15|16|

Waktu eksekusi: 8.1420 detik
Jumlah simpul yang dibangkitkan: 612314
PS D:\Github\StimaTucil3\src> |
```

b. Input/Output GUI (Graphical User Interface)





3.4. Studi Kasus 4 *Unsolvable* (TC4.txt)

a. Input/Output CLI (Command Line Interface)

```

PS D:\Github\StimaTucil3\src> py CLI.py
Masukkan nama file (tanpa .txt): TC4
Program dalam progress ...

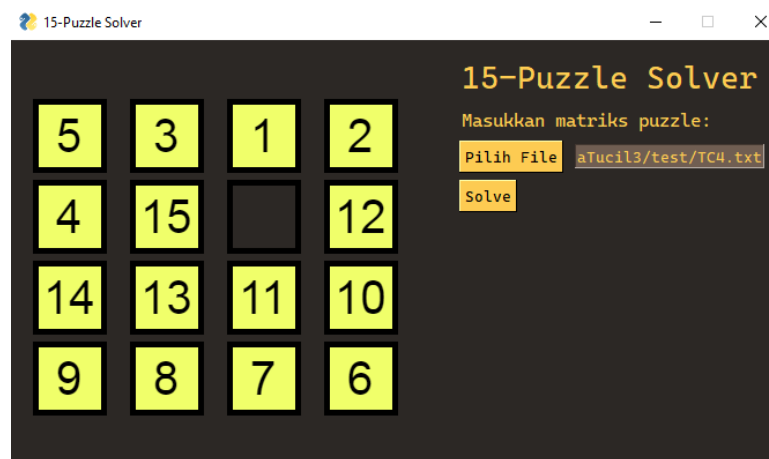
Matriks posisi awal:
| 5 | 3 | 1 | 2 |
| 4 | 15 | 16 | 12 |
| 14 | 13 | 11 | 10 |
| 9 | 8 | 7 | 6 |

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

Puzzle tidak bisa diselesaikan!
PS D:\Github\StimaTucil3\src>

```

b. Input/Output GUI (Graphical User Interface)





3.5. Studi Kasus 5 *Unsolvable* (TC5.txt)

a. Input/Output CLI (Command Line Interface)

```
Windows PowerShell
PS D:\Github\StimaTucil3\src> py CLI.py
Masukkan nama file (tanpa .txt): TC5
Program dalam progress ...

Matriks posisi awal:
| 5 | 8 | 7 | 2 |
| 1 | 3 | 4 | 10 |
| 6 | 9 | 11 | 15 |
| 12 | 13 | 14 | 16 |

Nilai Kurang(i):
Kurang(1) = 0
Kurang(2) = 1
Kurang(3) = 0
Kurang(4) = 0
Kurang(5) = 4
Kurang(6) = 0
Kurang(7) = 5
Kurang(8) = 6
Kurang(9) = 0
Kurang(10) = 2
Kurang(11) = 0
Kurang(12) = 0
Kurang(13) = 0
Kurang(14) = 0
Kurang(15) = 3
Kurang(16) = 0
Total Kurang(i) + X = 21

Puzzle tidak bisa diselesaikan!
PS D:\Github\StimaTucil3\src> |
```

b. Input/Output GUI (Graphical User Interface)



BAB 4: Berkas Teks Studi Kasus

4.1. Studi Kasus 1 *Solvable* (TC1.txt)

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

4.2. Studi Kasus 2 *Solvable* (TC2.txt)

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

4.3. Studi Kasus 3 *Solvable* (TC3.txt)

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

4.4. Studi Kasus 4 *Unsolvable* (TC4.txt)

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

4.5. Studi Kasus 5 *Unsolvable* (TC5.txt)

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

Lampiran

1. Alamat Kode Program (*Github*)

<https://github.com/JeremyRio/StimaTucil3>

2. Tabel Pengujian

Poin	Ya	Tidak
1. Program berhasil dikompilasi	√	
2. Program berhasil running	√	
3. Program dapat menerima input dan menuliskan output.	√	
4. Program dapat menerima input dan menuliskan output.	√	
5. Bonus dibuat	√	