

LAPORAN TUGAS KECIL 3
IF2211 STRATEGI ALGORITMA

Penyelesaian Persoalan 15-Puzzle dengan Algoritma
Branch and Bound



Disusun oleh:

Eiffel Aqila Amarendra 13520074

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung
2022

1. PENJELASAN ALGORITMA *BRANCH AND BOUND*

Algoritma *branch and bound* merupakan strategi algoritma pemecahan permasalahan optimasi dengan menggabungkan algoritma *Breadth First Search* dengan *least cost search* (pada kasus minimasi). Pada BFS, pembangkitan simpul selanjutnya yang akan diekspansi diurut berdasarkan urutan pembangkitannya (FIFO), sedangkan pada B&B, pembangkitan simpul berikutnya yang akan diekspansi diurut berdasarkan *cost* termurah (pada kasus minimasi). *Cost* ($\hat{c}(i)$) merupakan nilai taksiran lintasan termurah kepada *goal state* yang melalui *state node* ke-*i*. Di dalam B&B, dikenal pula istilah "Fungsi Pembatas", yakni fungsi pemangkas jalur yang dianggap tidak lagi mengarah ke solusi.

Algoritma *branch and bound* dapat digunakan untuk menyelesaikan berbagai macam persoalan, mulai dari persoalan N-Ratu (*The N-Queens Problem*), *Traveling Salesperson Problem* (TSP), *Assignment Problem*, *Integer Knapsack Problem*, hingga Permainan 15-Puzzle. Pada tugas kecil ini, algoritma *branch and bound* digunakan untuk menyelesaikan Persoalan 15-Puzzle.

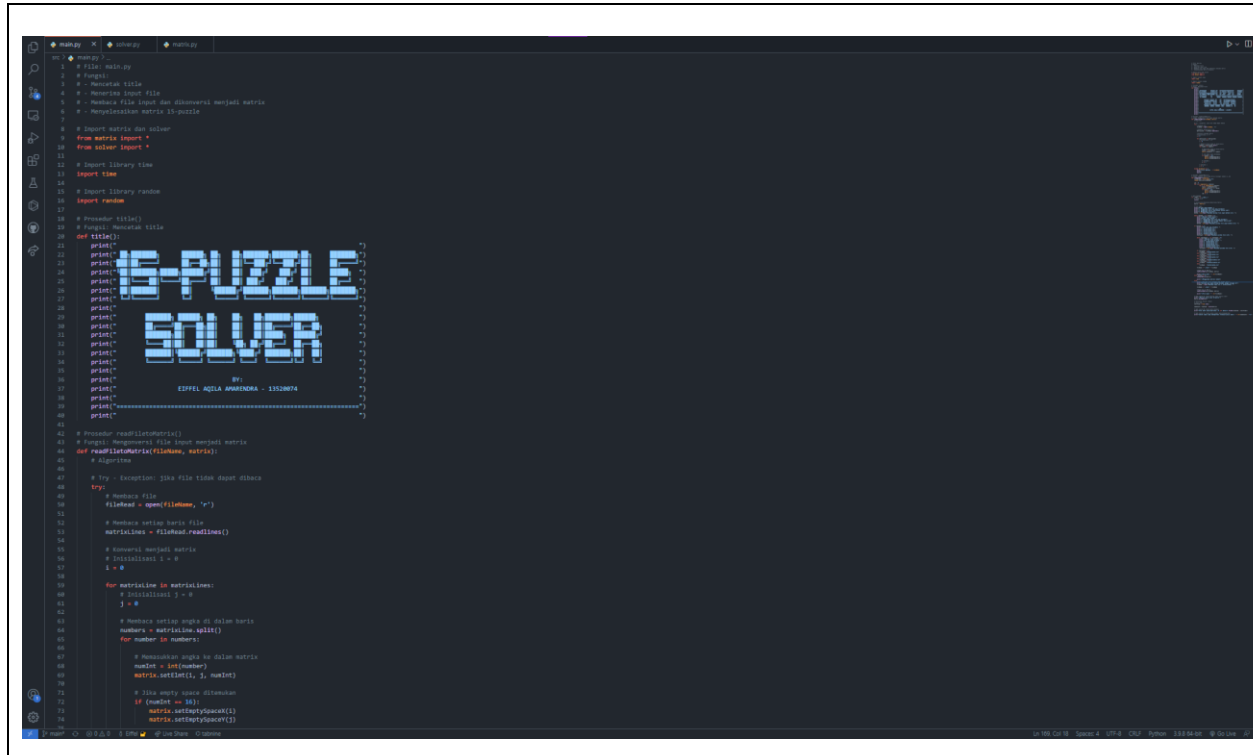
Secara umum, penjelasan algoritma *branch and bound* di dalam file 'solver.py' pada program ini adalah sebagai berikut,

1. Program menerima matrix sebagai simpul akar.
2. Program memasukkan simpul akar tersebut ke dalam *priority queue*. Jika simpul akar merupakan *goal state*, maka solusi telah ditemukan dan dicetak ke layar.
3. Jika *priority queue* kosong, pencarian dihentikan.
4. Jika belum, pilih dari antrian *priority queue* sebuah simpul dengan *cost* terendah (elemen pertama *priority queue*).
5. Jika simpul tersebut merupakan *goal state*, maka solusi telah ditemukan dan dicetak ke layar.
6. Jika bukan, bangkitkan seluruh anak-anaknya, hitung *cost* untuk setiap anak-anak tersebut, dan masukkan ke dalam *priority queue*. Jika simpul tidak memiliki anak, kembali ke langkah 2.
7. Kembali ke langkah 2

2. SOURCE CODE PROGRAM

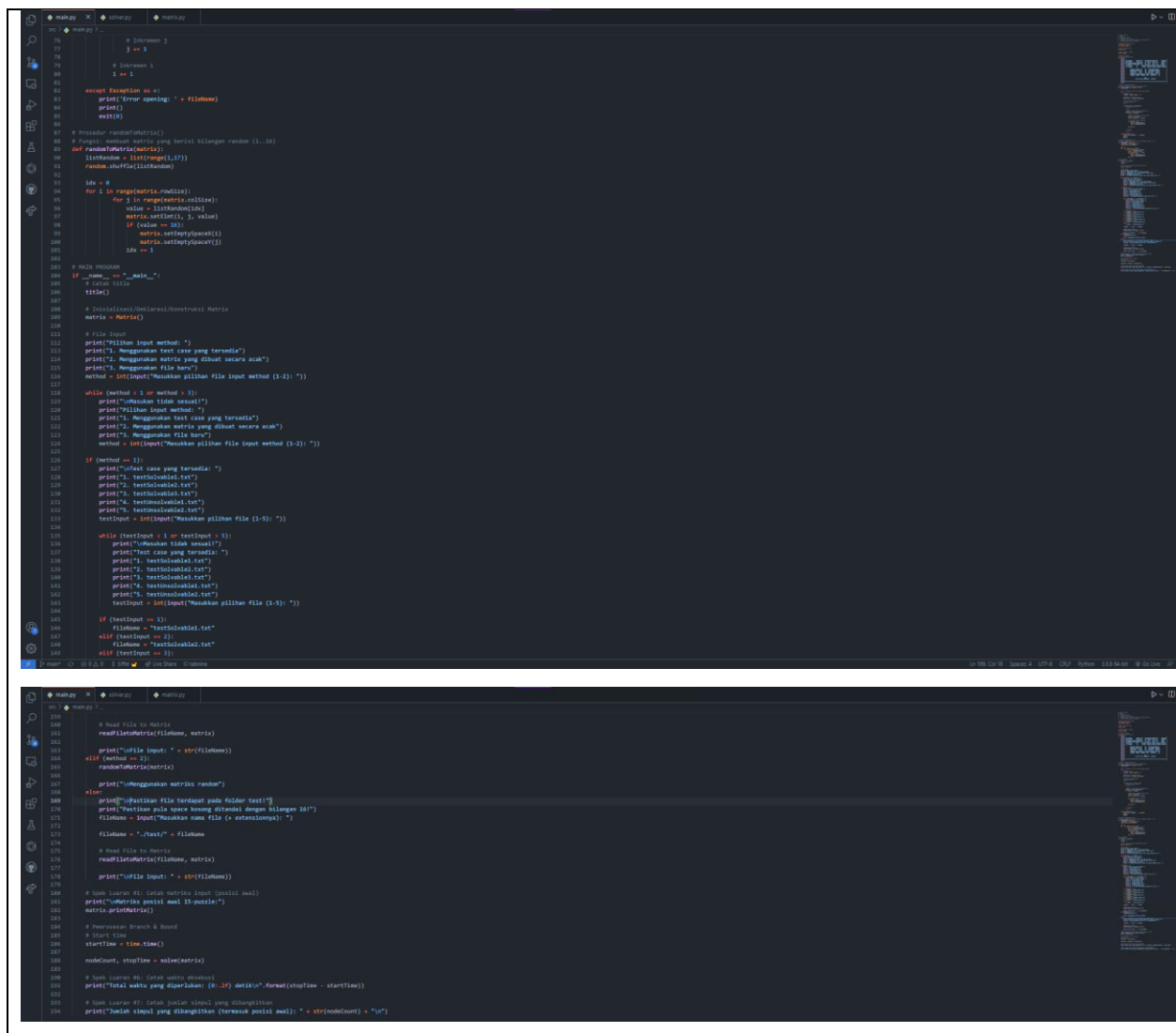
Program penyelesaian permasalahan 15-*Puzzle* dengan menggunakan algoritma *branch and bound* dalam tugas ini menggunakan bahasa pemrograman Python. Berikut hasil tangkapan layar program,

2.1. main.py



```
1 # main.py
2 # Fungsi
3 # Memuat data
4 # Memuat input file
5 # Memuat file input dan diinputkan menjadi matriks
6 # Mengkonversi matriks 15-puzzle
7
8 # Import matriks dan solver
9 from matriks import *
10 from solver import *
11
12 # Import library time
13 import time
14
15 # Import library random
16 import random
17
18 # Preset title()
19 # Fungsi Memuat data
20 def title():
21     print("\n")
22     print("15-PUZZLE")
23     print("SOLVER")
24     print("\n")
25     print("BY")
26     print("EFFEL AGILLA AMARTAMA - 15110074")
27     print("=====")
28     print("\n")
29
30 # Prosedur readInputMatrix()
31 # Fungsi Memuat input file menjadi matriks
32 def readInputMatrix(filename, matrix):
33     # Algoritma
34
35     # try - Exception: Jika file tidak dapat dibaca
36     try:
37         # Membaca file
38         fileRead = open(filename, "r")
39
40         # Membaca setiap baris file
41         matrixLines = fileRead.readlines()
42
43         # Memuat menjadi matriks
44         # Inisialisasi i = 0
45         i = 0
46
47         for matrixLine in matrixLines:
48             # Inisialisasi j = 0
49             j = 0
50
51             # Membaca setiap angka di dalam baris
52             numbers = matrixLine.split()
53             for number in numbers:
54                 # Memasukkan angka ke dalam matriks
55                 matrix[i][j] = int(number)
56                 matrix.setint(i, j, matrix[i][j])
57
58             # Jika empty space ditemukan
59             if (matrix[i][0] == 0):
60                 matrix.setemptySpace(i)
61                 matrix.setemptySpace(i)
62
63     except:
64         print("File tidak dapat dibaca")
65
66 # try - Exception: Jika file tidak dapat dibaca
67 try:
68     # Membaca file
69     fileRead = open(filename, "r")
70
71     # Membaca setiap baris file
72     matrixLines = fileRead.readlines()
73
74     # Memuat menjadi matriks
75     # Inisialisasi i = 0
76     i = 0
77
78     for matrixLine in matrixLines:
79         # Inisialisasi j = 0
80         j = 0
81
82         # Membaca setiap angka di dalam baris
83         numbers = matrixLine.split()
84         for number in numbers:
85             # Memasukkan angka ke dalam matriks
86             matrix[i][j] = int(number)
87             matrix.setint(i, j, matrix[i][j])
88
89         # Jika empty space ditemukan
90         if (matrix[i][0] == 0):
91             matrix.setemptySpace(i)
92             matrix.setemptySpace(i)
93
94 except:
95     print("File tidak dapat dibaca")
96
97 # try - Exception: Jika file tidak dapat dibaca
98 try:
99     # Membaca file
100    fileRead = open(filename, "r")
101
102    # Membaca setiap baris file
103    matrixLines = fileRead.readlines()
104
105    # Memuat menjadi matriks
106    # Inisialisasi i = 0
107    i = 0
108
109    for matrixLine in matrixLines:
110        # Inisialisasi j = 0
111        j = 0
112
113        # Membaca setiap angka di dalam baris
114        numbers = matrixLine.split()
115        for number in numbers:
116            # Memasukkan angka ke dalam matriks
117            matrix[i][j] = int(number)
118            matrix.setint(i, j, matrix[i][j])
119
120        # Jika empty space ditemukan
121        if (matrix[i][0] == 0):
122            matrix.setemptySpace(i)
123            matrix.setemptySpace(i)
124
125 except:
126     print("File tidak dapat dibaca")
127
128 # try - Exception: Jika file tidak dapat dibaca
129 try:
130     # Membaca file
131     fileRead = open(filename, "r")
132
133     # Membaca setiap baris file
134     matrixLines = fileRead.readlines()
135
136     # Memuat menjadi matriks
137     # Inisialisasi i = 0
138     i = 0
139
140     for matrixLine in matrixLines:
141         # Inisialisasi j = 0
142         j = 0
143
144         # Membaca setiap angka di dalam baris
145         numbers = matrixLine.split()
146         for number in numbers:
147             # Memasukkan angka ke dalam matriks
148             matrix[i][j] = int(number)
149             matrix.setint(i, j, matrix[i][j])
150
151         # Jika empty space ditemukan
152         if (matrix[i][0] == 0):
153             matrix.setemptySpace(i)
154             matrix.setemptySpace(i)
155
156 except:
157     print("File tidak dapat dibaca")
158
159 # try - Exception: Jika file tidak dapat dibaca
160 try:
161     # Membaca file
162     fileRead = open(filename, "r")
163
164     # Membaca setiap baris file
165     matrixLines = fileRead.readlines()
166
167     # Memuat menjadi matriks
168     # Inisialisasi i = 0
169     i = 0
170
171     for matrixLine in matrixLines:
172         # Inisialisasi j = 0
173         j = 0
174
175         # Membaca setiap angka di dalam baris
176         numbers = matrixLine.split()
177         for number in numbers:
178             # Memasukkan angka ke dalam matriks
179             matrix[i][j] = int(number)
180             matrix.setint(i, j, matrix[i][j])
181
182         # Jika empty space ditemukan
183         if (matrix[i][0] == 0):
184             matrix.setemptySpace(i)
185             matrix.setemptySpace(i)
186
187 except:
188     print("File tidak dapat dibaca")
189
190 # try - Exception: Jika file tidak dapat dibaca
191 try:
192     # Membaca file
193     fileRead = open(filename, "r")
194
195     # Membaca setiap baris file
196     matrixLines = fileRead.readlines()
197
198     # Memuat menjadi matriks
199     # Inisialisasi i = 0
200     i = 0
201
202     for matrixLine in matrixLines:
203         # Inisialisasi j = 0
204         j = 0
205
206         # Membaca setiap angka di dalam baris
207         numbers = matrixLine.split()
208         for number in numbers:
209             # Memasukkan angka ke dalam matriks
210             matrix[i][j] = int(number)
211             matrix.setint(i, j, matrix[i][j])
212
213         # Jika empty space ditemukan
214         if (matrix[i][0] == 0):
215             matrix.setemptySpace(i)
216             matrix.setemptySpace(i)
217
218 except:
219     print("File tidak dapat dibaca")
220
221 # try - Exception: Jika file tidak dapat dibaca
222 try:
223     # Membaca file
224     fileRead = open(filename, "r")
225
226     # Membaca setiap baris file
227     matrixLines = fileRead.readlines()
228
229     # Memuat menjadi matriks
230     # Inisialisasi i = 0
231     i = 0
232
233     for matrixLine in matrixLines:
234         # Inisialisasi j = 0
235         j = 0
236
237         # Membaca setiap angka di dalam baris
238         numbers = matrixLine.split()
239         for number in numbers:
240             # Memasukkan angka ke dalam matriks
241             matrix[i][j] = int(number)
242             matrix.setint(i, j, matrix[i][j])
243
244         # Jika empty space ditemukan
245         if (matrix[i][0] == 0):
246             matrix.setemptySpace(i)
247             matrix.setemptySpace(i)
248
249 except:
250     print("File tidak dapat dibaca")
251
252 # try - Exception: Jika file tidak dapat dibaca
253 try:
254     # Membaca file
255     fileRead = open(filename, "r")
256
257     # Membaca setiap baris file
258     matrixLines = fileRead.readlines()
259
260     # Memuat menjadi matriks
261     # Inisialisasi i = 0
262     i = 0
263
264     for matrixLine in matrixLines:
265         # Inisialisasi j = 0
266         j = 0
267
268         # Membaca setiap angka di dalam baris
269         numbers = matrixLine.split()
270         for number in numbers:
271             # Memasukkan angka ke dalam matriks
272             matrix[i][j] = int(number)
273             matrix.setint(i, j, matrix[i][j])
274
275         # Jika empty space ditemukan
276         if (matrix[i][0] == 0):
277             matrix.setemptySpace(i)
278             matrix.setemptySpace(i)
279
280 except:
281     print("File tidak dapat dibaca")
282
283 # try - Exception: Jika file tidak dapat dibaca
284 try:
285     # Membaca file
286     fileRead = open(filename, "r")
287
288     # Membaca setiap baris file
289     matrixLines = fileRead.readlines()
290
291     # Memuat menjadi matriks
292     # Inisialisasi i = 0
293     i = 0
294
295     for matrixLine in matrixLines:
296         # Inisialisasi j = 0
297         j = 0
298
299         # Membaca setiap angka di dalam baris
300         numbers = matrixLine.split()
301         for number in numbers:
302             # Memasukkan angka ke dalam matriks
303             matrix[i][j] = int(number)
304             matrix.setint(i, j, matrix[i][j])
305
306         # Jika empty space ditemukan
307         if (matrix[i][0] == 0):
308             matrix.setemptySpace(i)
309             matrix.setemptySpace(i)
310
311 except:
312     print("File tidak dapat dibaca")
313
314 # try - Exception: Jika file tidak dapat dibaca
315 try:
316     # Membaca file
317     fileRead = open(filename, "r")
318
319     # Membaca setiap baris file
320     matrixLines = fileRead.readlines()
321
322     # Memuat menjadi matriks
323     # Inisialisasi i = 0
324     i = 0
325
326     for matrixLine in matrixLines:
327         # Inisialisasi j = 0
328         j = 0
329
330         # Membaca setiap angka di dalam baris
331         numbers = matrixLine.split()
332         for number in numbers:
333             # Memasukkan angka ke dalam matriks
334             matrix[i][j] = int(number)
335             matrix.setint(i, j, matrix[i][j])
336
337         # Jika empty space ditemukan
338         if (matrix[i][0] == 0):
339             matrix.setemptySpace(i)
340             matrix.setemptySpace(i)
341
342 except:
343     print("File tidak dapat dibaca")
344
345 # try - Exception: Jika file tidak dapat dibaca
346 try:
347     # Membaca file
348     fileRead = open(filename, "r")
349
350     # Membaca setiap baris file
351     matrixLines = fileRead.readlines()
352
353     # Memuat menjadi matriks
354     # Inisialisasi i = 0
355     i = 0
356
357     for matrixLine in matrixLines:
358         # Inisialisasi j = 0
359         j = 0
360
361         # Membaca setiap angka di dalam baris
362         numbers = matrixLine.split()
363         for number in numbers:
364             # Memasukkan angka ke dalam matriks
365             matrix[i][j] = int(number)
366             matrix.setint(i, j, matrix[i][j])
367
368         # Jika empty space ditemukan
369         if (matrix[i][0] == 0):
370             matrix.setemptySpace(i)
371             matrix.setemptySpace(i)
372
373 except:
374     print("File tidak dapat dibaca")
375
376 # try - Exception: Jika file tidak dapat dibaca
377 try:
378     # Membaca file
379     fileRead = open(filename, "r")
380
381     # Membaca setiap baris file
382     matrixLines = fileRead.readlines()
383
384     # Memuat menjadi matriks
385     # Inisialisasi i = 0
386     i = 0
387
388     for matrixLine in matrixLines:
389         # Inisialisasi j = 0
390         j = 0
391
392         # Membaca setiap angka di dalam baris
393         numbers = matrixLine.split()
394         for number in numbers:
395             # Memasukkan angka ke dalam matriks
396             matrix[i][j] = int(number)
397             matrix.setint(i, j, matrix[i][j])
398
399         # Jika empty space ditemukan
400         if (matrix[i][0] == 0):
401             matrix.setemptySpace(i)
402             matrix.setemptySpace(i)
403
404 except:
405     print("File tidak dapat dibaca")
406
407 # try - Exception: Jika file tidak dapat dibaca
408 try:
409     # Membaca file
410     fileRead = open(filename, "r")
411
412     # Membaca setiap baris file
413     matrixLines = fileRead.readlines()
414
415     # Memuat menjadi matriks
416     # Inisialisasi i = 0
417     i = 0
418
419     for matrixLine in matrixLines:
420         # Inisialisasi j = 0
421         j = 0
422
423         # Membaca setiap angka di dalam baris
424         numbers = matrixLine.split()
425         for number in numbers:
426             # Memasukkan angka ke dalam matriks
427             matrix[i][j] = int(number)
428             matrix.setint(i, j, matrix[i][j])
429
430         # Jika empty space ditemukan
431         if (matrix[i][0] == 0):
432             matrix.setemptySpace(i)
433             matrix.setemptySpace(i)
434
435 except:
436     print("File tidak dapat dibaca")
437
438 # try - Exception: Jika file tidak dapat dibaca
439 try:
440     # Membaca file
441     fileRead = open(filename, "r")
442
443     # Membaca setiap baris file
444     matrixLines = fileRead.readlines()
445
446     # Memuat menjadi matriks
447     # Inisialisasi i = 0
448     i = 0
449
450     for matrixLine in matrixLines:
451         # Inisialisasi j = 0
452         j = 0
453
454         # Membaca setiap angka di dalam baris
455         numbers = matrixLine.split()
456         for number in numbers:
457             # Memasukkan angka ke dalam matriks
458             matrix[i][j] = int(number)
459             matrix.setint(i, j, matrix[i][j])
460
461         # Jika empty space ditemukan
462         if (matrix[i][0] == 0):
463             matrix.setemptySpace(i)
464             matrix.setemptySpace(i)
465
466 except:
467     print("File tidak dapat dibaca")
468
469 # try - Exception: Jika file tidak dapat dibaca
470 try:
471     # Membaca file
472     fileRead = open(filename, "r")
473
474     # Membaca setiap baris file
475     matrixLines = fileRead.readlines()
476
477     # Memuat menjadi matriks
478     # Inisialisasi i = 0
479     i = 0
480
481     for matrixLine in matrixLines:
482         # Inisialisasi j = 0
483         j = 0
484
485         # Membaca setiap angka di dalam baris
486         numbers = matrixLine.split()
487         for number in numbers:
488             # Memasukkan angka ke dalam matriks
489             matrix[i][j] = int(number)
490             matrix.setint(i, j, matrix[i][j])
491
492         # Jika empty space ditemukan
493         if (matrix[i][0] == 0):
494             matrix.setemptySpace(i)
495             matrix.setemptySpace(i)
496
497 except:
498     print("File tidak dapat dibaca")
499
500 # try - Exception: Jika file tidak dapat dibaca
501 try:
502     # Membaca file
503     fileRead = open(filename, "r")
504
505     # Membaca setiap baris file
506     matrixLines = fileRead.readlines()
507
508     # Memuat menjadi matriks
509     # Inisialisasi i = 0
510     i = 0
511
512     for matrixLine in matrixLines:
513         # Inisialisasi j = 0
514         j = 0
515
516         # Membaca setiap angka di dalam baris
517         numbers = matrixLine.split()
518         for number in numbers:
519             # Memasukkan angka ke dalam matriks
520             matrix[i][j] = int(number)
521             matrix.setint(i, j, matrix[i][j])
522
523         # Jika empty space ditemukan
524         if (matrix[i][0] == 0):
525             matrix.setemptySpace(i)
526             matrix.setemptySpace(i)
527
528 except:
529     print("File tidak dapat dibaca")
530
531 # try - Exception: Jika file tidak dapat dibaca
532 try:
533     # Membaca file
534     fileRead = open(filename, "r")
535
536     # Membaca setiap baris file
537     matrixLines = fileRead.readlines()
538
539     # Memuat menjadi matriks
540     # Inisialisasi i = 0
541     i = 0
542
543     for matrixLine in matrixLines:
544         # Inisialisasi j = 0
545         j = 0
546
547         # Membaca setiap angka di dalam baris
548         numbers = matrixLine.split()
549         for number in numbers:
550             # Memasukkan angka ke dalam matriks
551             matrix[i][j] = int(number)
552             matrix.setint(i, j, matrix[i][j])
553
554         # Jika empty space ditemukan
555         if (matrix[i][0] == 0):
556             matrix.setemptySpace(i)
557             matrix.setemptySpace(i)
558
559 except:
560     print("File tidak dapat dibaca")
561
562 # try - Exception: Jika file tidak dapat dibaca
563 try:
564     # Membaca file
565     fileRead = open(filename, "r")
566
567     # Membaca setiap baris file
568     matrixLines = fileRead.readlines()
569
570     # Memuat menjadi matriks
571     # Inisialisasi i = 0
572     i = 0
573
574     for matrixLine in matrixLines:
575         # Inisialisasi j = 0
576         j = 0
577
578         # Membaca setiap angka di dalam baris
579         numbers = matrixLine.split()
580         for number in numbers:
581             # Memasukkan angka ke dalam matriks
582             matrix[i][j] = int(number)
583             matrix.setint(i, j, matrix[i][j])
584
585         # Jika empty space ditemukan
586         if (matrix[i][0] == 0):
587             matrix.setemptySpace(i)
588             matrix.setemptySpace(i)
589
590 except:
591     print("File tidak dapat dibaca")
592
593 # try - Exception: Jika file tidak dapat dibaca
594 try:
595     # Membaca file
596     fileRead = open(filename, "r")
597
598     # Membaca setiap baris file
599     matrixLines = fileRead.readlines()
600
601     # Memuat menjadi matriks
602     # Inisialisasi i = 0
603     i = 0
604
605     for matrixLine in matrixLines:
606         # Inisialisasi j = 0
607         j = 0
608
609         # Membaca setiap angka di dalam baris
610         numbers = matrixLine.split()
611         for number in numbers:
612             # Memasukkan angka ke dalam matriks
613             matrix[i][j] = int(number)
614             matrix.setint(i, j, matrix[i][j])
615
616         # Jika empty space ditemukan
617         if (matrix[i][0] == 0):
618             matrix.setemptySpace(i)
619             matrix.setemptySpace(i)
620
621 except:
622     print("File tidak dapat dibaca")
623
624 # try - Exception: Jika file tidak dapat dibaca
625 try:
626     # Membaca file
627     fileRead = open(filename, "r")
628
629     # Membaca setiap baris file
630     matrixLines = fileRead.readlines()
631
632     # Memuat menjadi matriks
633     # Inisialisasi i = 0
634     i = 0
635
636     for matrixLine in matrixLines:
637         # Inisialisasi j = 0
638         j = 0
639
640         # Membaca setiap angka di dalam baris
641         numbers = matrixLine.split()
642         for number in numbers:
643             # Memasukkan angka ke dalam matriks
644             matrix[i][j] = int(number)
645             matrix.setint(i, j, matrix[i][j])
646
647         # Jika empty space ditemukan
648         if (matrix[i][0] == 0):
649             matrix.setemptySpace(i)
650             matrix.setemptySpace(i)
651
652 except:
653     print("File tidak dapat dibaca")
654
655 # try - Exception: Jika file tidak dapat dibaca
656 try:
657     # Membaca file
658     fileRead = open(filename, "r")
659
660     # Membaca setiap baris file
661     matrixLines = fileRead.readlines()
662
663     # Memuat menjadi matriks
664     # Inisialisasi i = 0
665     i = 0
666
667     for matrixLine in matrixLines:
668         # Inisialisasi j = 0
669         j = 0
670
671         # Membaca setiap angka di dalam baris
672         numbers = matrixLine.split()
673         for number in numbers:
674             # Memasukkan angka ke dalam matriks
675             matrix[i][j] = int(number)
676             matrix.setint(i, j, matrix[i][j])
677
678         # Jika empty space ditemukan
679         if (matrix[i][0] == 0):
680             matrix.setemptySpace(i)
681             matrix.setemptySpace(i)
682
683 except:
684     print("File tidak dapat dibaca")
685
686 # try - Exception: Jika file tidak dapat dibaca
687 try:
688     # Membaca file
689     fileRead = open(filename, "r")
690
691     # Membaca setiap baris file
692     matrixLines = fileRead.readlines()
693
694     # Memuat menjadi matriks
695     # Inisialisasi i = 0
696     i = 0
697
698     for matrixLine in matrixLines:
699         # Inisialisasi j = 0
700         j = 0
701
702         # Membaca setiap angka di dalam baris
703         numbers = matrixLine.split()
704         for number in numbers:
705             # Memasukkan angka ke dalam matriks
706             matrix[i][j] = int(number)
707             matrix.setint(i, j, matrix[i][j])
708
709         # Jika empty space ditemukan
710         if (matrix[i][0] == 0):
711             matrix.setemptySpace(i)
712             matrix.setemptySpace(i)
713
714 except:
715     print("File tidak dapat dibaca")
716
717 # try - Exception: Jika file tidak dapat dibaca
718 try:
719     # Membaca file
720     fileRead = open(filename, "r")
721
722     # Membaca setiap baris file
723     matrixLines = fileRead.readlines()
724
725     # Memuat menjadi matriks
726     # Inisialisasi i = 0
727     i = 0
728
729     for matrixLine in matrixLines:
730         # Inisialisasi j = 0
731         j = 0
732
733         # Membaca setiap angka di dalam baris
734         numbers = matrixLine.split()
735         for number in numbers:
736             # Memasukkan angka ke dalam matriks
737             matrix[i][j] = int(number)
738             matrix.setint(i, j, matrix[i][j])
739
740         # Jika empty space ditemukan
741         if (matrix[i][0] == 0):
742             matrix.setemptySpace(i)
743             matrix.setemptySpace(i)
744
745 except:
746     print("File tidak dapat dibaca")
747
748 # try - Exception: Jika file tidak dapat dibaca
749 try:
750     # Membaca file
751     fileRead = open(filename, "r")
752
753     # Membaca setiap baris file
754     matrixLines = fileRead.readlines()
755
756     # Memuat menjadi matriks
757     # Inisialisasi i = 0
758     i = 0
759
760     for matrixLine in matrixLines:
761         # Inisialisasi j = 0
762         j = 0
763
764         # Membaca setiap angka di dalam baris
765         numbers = matrixLine.split()
766         for number in numbers:
767             # Memasukkan angka ke dalam matriks
768             matrix[i][j] = int(number)
769             matrix.setint(i, j, matrix[i][j])
770
771         # Jika empty space ditemukan
772         if (matrix[i][0] == 0):
773             matrix.setemptySpace(i)
774             matrix.setemptySpace(i)
775
776 except:
777     print("File tidak dapat dibaca")
778
779 # try - Exception: Jika file tidak dapat dibaca
780 try:
781     # Membaca file
782     fileRead = open(filename, "r")
783
784     # Membaca setiap baris file
785     matrixLines = fileRead.readlines()
786
787     # Memuat menjadi matriks
788     # Inisialisasi i = 0
789     i = 0
790
791     for matrixLine in matrixLines:
792         # Inisialisasi j = 0
793         j = 0
794
795         # Membaca setiap angka di dalam baris
796         numbers = matrixLine.split()
797         for number in numbers:
798             # Memasukkan angka ke dalam matriks
799             matrix[i][j] = int(number)
800             matrix.setint(i, j, matrix[i][j])
801
802         # Jika empty space ditemukan
803         if (matrix[i][0] == 0):
804             matrix.setemptySpace(i)
805             matrix.setemptySpace(i)
806
807 except:
808     print("File tidak dapat dibaca")
809
810 # try - Exception: Jika file tidak dapat dibaca
811 try:
812     # Membaca file
813     fileRead = open(filename, "r")
814
815     # Membaca setiap baris file
816     matrixLines = fileRead.readlines()
817
818     # Memuat menjadi matriks
819     # Inisialisasi i = 0
820     i = 0
821
822     for matrixLine in matrixLines:
823         # Inisialisasi j = 0
824         j = 0
825
826         # Membaca setiap angka di dalam baris
827         numbers = matrixLine.split()
828         for number in numbers:
829             # Memasukkan angka ke dalam matriks
830             matrix[i][j] = int(number)
831             matrix.setint(i, j, matrix[i][j])
832
833         # Jika empty space ditemukan
834         if (matrix[i][0] == 0):
835             matrix.setemptySpace(i)
836             matrix.setemptySpace(i)
837
838 except:
839     print("File tidak dapat dibaca")
840
841 # try - Exception: Jika file tidak dapat dibaca
842 try:
843     # Membaca file
844     fileRead = open(filename, "r")
845
846     # Membaca setiap baris file
847     matrixLines = fileRead.readlines()
848
849     # Memuat menjadi matriks
850     # Inisialisasi i = 0
851     i = 0
852
853     for matrixLine in matrixLines:
854         # Inisialisasi j = 0
855         j = 0
856
857         # Membaca setiap angka di dalam baris
858         numbers = matrixLine.split()
859         for number in numbers:
860             # Memasukkan angka ke dalam matriks
861             matrix[i][j] = int(number)
862             matrix.setint(i, j, matrix[i][j])
863
864         # Jika empty space ditemukan
865         if (matrix[i][0] == 0):
866             matrix.setemptySpace(i)
867             matrix.setemptySpace(i)
868
869 except:
870     print("File tidak dapat dibaca")
871
872 # try - Exception: Jika file tidak dapat dibaca
873 try:
874     # Membaca file
875     fileRead = open(filename, "r")
876
877     # Membaca setiap baris file
878     matrixLines = fileRead.readlines()
879
880     # Memuat menjadi matriks
881     # Inisialisasi i = 0
882     i = 0
883
884     for matrixLine in matrixLines:
885         # Inisialisasi j = 0
886         j = 0
887
888         # Membaca setiap angka di dalam baris
889         numbers = matrixLine.split()
890         for number in numbers:
891             # Memasukkan angka ke dalam matriks
892             matrix[i][j] = int(number)
893             matrix.setint(i, j, matrix[i][j])
894
895         # Jika empty space ditemukan
896         if (matrix[i][0] == 0):
897             matrix.setemptySpace(i)
898             matrix.setemptySpace(i)
899
900 except:
901     print("File tidak dapat dibaca")
902
903 # try - Exception: Jika file tidak dapat dibaca
904 try:
905     # Membaca file
906     fileRead = open(filename, "r")
907
908     # Membaca setiap baris file
909     matrixLines = fileRead.readlines()
910
911     # Memuat menjadi matriks
912     # Inisialisasi i = 0
913     i = 0
914
915     for matrixLine in matrixLines:
916         # Inisialisasi j = 0
917         j = 0
918
919         # Membaca setiap angka di dalam baris
920         numbers = matrixLine.split()
921         for number in numbers:
922             # Memasukkan angka ke dalam matriks
923             matrix[i][j] = int(number)
924             matrix.setint(i, j, matrix[i][j])
925
926         # Jika empty space ditemukan
927         if (matrix[i][0] == 0):
928             matrix.setemptySpace(i)
929             matrix.setemptySpace(i)
930
931 except:
932     print("File tidak dapat dibaca")
933
934 # try - Exception: Jika file tidak dapat dibaca
935 try:
936     # Membaca file
937     fileRead = open(filename, "r")
938
939     # Membaca setiap baris file
940     matrixLines = fileRead.readlines()
941
942     # Memuat menjadi matriks
943     # Inisialisasi i = 0
944     i = 0
945
946     for matrixLine in matrixLines:
947         # Inisialisasi j = 0
948         j = 0
949
950         # Membaca setiap angka di dalam baris
951         numbers = matrixLine.split()
952         for number in numbers:
953             # Memasukkan angka ke dalam matriks
954             matrix[i][j] = int(number)
955             matrix.setint(i, j, matrix[i][j])
956
957         # Jika empty space ditemukan
958         if (matrix[i][0] == 0):
959             matrix.setemptySpace(i)
960             matrix.setemptySpace(i)
961
962 except:
963     print("File tidak dapat dibaca")
964
965 # try - Exception: Jika file tidak dapat dibaca
966 try:
967     # Membaca file
968     fileRead = open(filename, "r")
969
970     # Membaca setiap baris file
971     matrixLines = fileRead.readlines()
972
973     # Memuat menjadi matriks
974     # Inisialisasi i = 0
975     i = 0
976
977     for matrixLine in matrixLines:
978         # Inisialisasi j = 0
979         j = 0
980
981         # Membaca setiap angka di dalam baris
982         numbers = matrixLine.split()
983         for number in numbers:
984             # Memasukkan angka ke dalam matriks
985             matrix[i][j] = int(number)
986             matrix.setint(i, j, matrix[i][j])
987
988         # Jika empty space ditemukan
989         if (matrix[i][0] == 0):
990             matrix.setemptySpace(i)
991             matrix.setemptySpace(i)
992
993 except:
994     print("File tidak dapat dibaca")
995
996 # try - Exception: Jika file tidak dapat dibaca
997 try:
998     # Membaca file
999     fileRead = open(filename, "r")
1000
1001     # Membaca setiap baris file
1002     matrixLines = fileRead.readlines()
1003
1004     # Memuat menjadi matriks
1005     # Inisialisasi i = 0
1006     i = 0
1007
1008     for matrixLine in matrixLines:
1009         # Inisialisasi j = 0
1010         j = 0
1011
1012         # Membaca setiap angka di dalam baris
1013         numbers = matrixLine.split()
1014         for number in numbers:
1015             # Memasukkan angka ke dalam matriks
1016             matrix[i][j] = int(number)
1017             matrix.setint(i, j, matrix[i][j])
1018
1019         # Jika empty space ditemukan
1020         if (matrix[i][0] == 0):
1021             matrix.setemptySpace(i)
1022             matrix.setemptySpace(i)
1023
1024 except:
1025     print("File tidak dapat dibaca")
1026
1027 # try - Exception: Jika file tidak dapat dibaca
1028 try:
1029     # Membaca file
1030     fileRead = open(filename, "r")
1031
1032     # Membaca setiap baris file
1033     matrixLines = fileRead.readlines()
1034
1035     # Memuat menjadi matriks
1036     # Inisialisasi i = 0
1037     i = 0
1038
1039     for matrixLine in matrixLines:
1040         # Inisialisasi j = 0
1041         j = 0
1042
1043         # Membaca setiap angka di dalam baris
1044         numbers = matrixLine.split()
1045         for number in numbers:
1046             # Memasukkan angka ke dalam matriks
1047             matrix[i][j] = int(number)
1048             matrix.setint(i, j, matrix[i][j])
1049
1050         # Jika empty space ditemukan
1051         if (matrix[i][0] == 0):
1052             matrix.setemptySpace(i)
1053             matrix.setemptySpace(i)
1054
1055 except:
1056     print("File tidak dapat dibaca")
1057
1058 # try - Exception: Jika file tidak dapat dibaca
1059 try:
1060     # Membaca file
1061     fileRead = open(filename, "r")
1062
1063     # Membaca setiap baris file
1064     matrixLines = fileRead.readlines()
1065
1066     # Memuat menjadi matriks
1067     # Inisialisasi i = 0
1068     i = 0
1069
1070     for matrixLine in matrixLines:
1071         # Inisialisasi j = 0
1072         j = 0
1073
1074         # Membaca setiap angka di dalam baris
1075         numbers = matrixLine.split()
1076         for number in numbers:
1077             # Memasukkan angka ke dalam matriks
1078             matrix[i][j] = int(number)
1079             matrix.setint(i, j, matrix[i][j])
1080
1081         # Jika empty space ditemukan
1082         if (matrix[i][0] == 0):
1083             matrix.setemptySpace(i)
1084             matrix.setemptySpace(i)
1085
1086 except:
1087     print("File tidak dapat dibaca")
1088
1089 # try - Exception: Jika file tidak dapat dibaca
1090 try:
1091     # Membaca file
1092     fileRead = open(filename, "r")
1093
1094     # Membaca setiap baris file
1095     matrixLines = fileRead.readlines()
1096
1097     # Memuat menjadi matriks
1098     # Inisialisasi i = 0
1099     i = 0
1100
1101     for matrixLine in matrixLines:
1102         # Inisialisasi j = 0
1103         j = 0
1104
1105         # Membaca setiap angka di dalam baris
1106         numbers = matrixLine.split()
1107         for number in numbers:
1108             # Memasukkan angka ke dalam matriks
1109             matrix[i][j] = int(number)
1110             matrix.setint(i, j, matrix[i][j])
1111
1112         # Jika empty space ditemukan
1113         if (matrix[i][0] == 0):
1114             matrix.setemptySpace(i)
1115             matrix.setemptySpace(i)
1116
1117 except:
1118     print("File tidak dapat dibaca")
1119
1120 # try - Exception: Jika file tidak dapat dibaca
1121 try:
1122     # Membaca file
1123     fileRead = open(filename, "r")
1124
1125     # Membaca setiap baris file
1126     matrixLines = fileRead.readlines()
1127
1128     # Memuat menjadi matriks
1129     # Inisialisasi i = 0
1130     i = 0
1131
1132     for matrixLine in matrixLines:
1133         # Inisialisasi j = 0
1134         j = 0
1135
1136         # Membaca setiap angka di dalam baris
1137         numbers = matrixLine.split()
1138         for number in numbers:
1139             # Memasukkan angka ke dalam matriks
1140             matrix[i][j] = int(number)
1141             matrix.setint(i, j, matrix[i][j])
1142
1143         # Jika empty space ditemukan
1144         if (matrix[i][0] == 0):
1145             matrix.setemptySpace(i)
1146             matrix.setemptySpace(i)
1147
1148 except:
1149     print("File tidak dapat dibaca")
1150
1151 # try - Exception: Jika file tidak dapat dibaca
1152 try:
1153     # Membaca file
1154     fileRead = open(filename, "r")
1155
1156     # Membaca setiap baris file
1157     matrixLines = fileRead.readlines()
1158
1159     # Memuat menjadi matriks
1160     # Inisialisasi i = 0
1161     i = 0
1162
1163     for matrixLine in matrixLines:
1164         # Inisialisasi j = 0
1165         j = 0
1166
1167         # Membaca setiap angka di dalam baris
1168         numbers = matrixLine.split()
1169         for number in numbers:
1170             # Memasukkan angka ke dalam matriks
1171             matrix[i][j] = int(number)
1172             matrix.setint(i, j, matrix[i][j])
1173
1174         # Jika empty space ditemukan
1175         if (matrix[i][0] == 0):
1176             matrix.setemptySpace(i)
1177             matrix.setemptySpace(i)
1178
1179 except:
1180     print("File tidak dapat dibaca")
1181
1182 # try - Exception: Jika file tidak dapat dibaca
1183 try:
1184     # Membaca file
1185     fileRead = open(filename, "r")
1186
1187     # Membaca setiap baris file
1188     matrixLines = fileRead.readlines()
1189
1190     # Memuat menjadi matriks
1191     # Inisialisasi i = 0
1192     i = 0
1193
1194     for matrixLine in matrixLines:
1195         # Inisialisasi j = 0
1196         j = 0
1197
1198         # Membaca setiap angka di dalam baris
1199         numbers = matrixLine.split()

```



2.2. solver.py

[illegible]

```
150 matrixExpanditas = copy.deepcopy(matrix)
151 matrixExpanditas.aten()
152 matrixExpanditas.setPreviousMatrix(matrix)
153
154 # Perhitungan cost
155 costitas = cost(matrixExpanditas)
156 matrixExpanditas.setCost(costitas)
157
158 # Masukkan ke dalam antrian
159 heapq.heappush(priorityQueue, matrixExpanditas)
160 nodeCount += 1
161
162 # User empty space ke kanan
163 if ([emptyY != 3] and (matrix.getPreviousMove() != "L")):
164     # found matrix
165     matrixExpandkanan = copy.deepcopy(matrix)
166     matrixExpandkanan.kanan()
167     matrixExpandkanan.setPreviousMatrix(matrix)
168
169     # Perhitungan cost
170 costkanan = cost(matrixExpandkanan)
171 matrixExpandkanan.setCost(costkanan)
172
173 # Masukkan ke dalam antrian
174 heapq.heappush(priorityQueue, matrixExpandkanan)
175 nodeCount += 1
176
177 # User empty space ke bawah
178 if ([emptyY != 3] and (matrix.getPreviousMove() != "U")):
179     # found matrix
180     matrixExpandbawah = copy.deepcopy(matrix)
181     matrixExpandbawah.bawah()
182     matrixExpandbawah.setPreviousMatrix(matrix)
183
184     # Perhitungan cost
185 costbawah = cost(matrixExpandbawah)
186 matrixExpandbawah.setCost(costbawah)
187
188 # Masukkan ke dalam antrian
189 heapq.heappush(priorityQueue, matrixExpandbawah)
190 nodeCount += 1
191
192 # User empty space ke kiri
193 if ([emptyY != 0] and (matrix.getPreviousMove() != "R")):
194     # found matrix
195     matrixExpandkiri = copy.deepcopy(matrix)
196     matrixExpandkiri.kiri()
197     matrixExpandkiri.setPreviousMatrix(matrix)
198
199     # Perhitungan cost
200 costkiri = cost(matrixExpandkiri)
201 matrixExpandkiri.setCost(costkiri)
202
203 # Masukkan ke dalam antrian
204 heapq.heappush(priorityQueue, matrixExpandkiri)
205 nodeCount += 1
```

2.3. matrix.py

```

1 # -*- coding: utf-8 -*-
2 # File: matrix.py
3
4 # CLASS MATRIX
5 class Matrix:
6     # ***** INIT *****
7     # matrix: array of array of integers
8     # matrix_row: integer - index first empty space
9     # matrix_col: integer - index last empty space
10    # count_row: integer - number of rows (rows that have been processed)
11    # count_col: integer - number of columns (columns that have been processed)
12    # count_row = 0
13    # count_col = 0
14
15    # ***** INIT *****
16    def __init__(self):
17        self.matrix = []
18        self.row = 0
19        self.col = 0
20        self.matrix_row = 0
21        self.matrix_col = 0
22        self.count_row = 0
23        self.count_col = 0
24        self.matrix_row = None
25        self.matrix_col = None
26        self.count_row = 0
27        self.count_col = 0
28
29    def __str__(self, other):
30        return self.count + other.count
31
32    # ***** INIT *****
33    # set row
34    def get_row(self, i):
35        return self.matrix[i]
36
37    # set row
38    def set_row(self, i, value):
39        self.matrix[i] = value
40
41    # set row space
42    def get_row_space(self):
43        return self.empty_row
44
45    def set_row_space(self):
46        return self.empty_row
47
48    # set row space
49    def get_row_space(self):
50        return self.empty_row
51
52    def set_row_space(self):
53        return self.empty_row
54
55    # set row space
56    def get_row_space(self):
57        return self.empty_row
58
59    def set_row_space(self):
60        return self.empty_row
61
62    # set row space
63    def get_row_space(self):
64        return self.empty_row
65
66    def set_row_space(self):
67        return self.empty_row
68
69    # set row space
70    def get_row_space(self):
71        return self.empty_row
72
73    def set_row_space(self):
74        return self.empty_row
75
76    # set row space
77    def get_row_space(self):
78        return self.empty_row
79
80    def set_row_space(self):
81        return self.empty_row
82
83    # set row space
84    def get_row_space(self):
85        return self.empty_row
86
87    def set_row_space(self):
88        return self.empty_row
89
90    # set row space
91    def get_row_space(self):
92        return self.empty_row
93
94    def set_row_space(self):
95        return self.empty_row
96
97    # set row space
98    def get_row_space(self):
99        return self.empty_row
100    def set_row_space(self):
101        return self.empty_row
102
103    # set row space
104    def get_row_space(self):
105        return self.empty_row
106    def set_row_space(self):
107        return self.empty_row
108
109    # set row space
110    def get_row_space(self):
111        return self.empty_row
112    def set_row_space(self):
113        return self.empty_row
114
115    # set row space
116    def get_row_space(self):
117        return self.empty_row
118    def set_row_space(self):
119        return self.empty_row
120
121    # set row space
122    def get_row_space(self):
123        return self.empty_row
124    def set_row_space(self):
125        return self.empty_row
126
127    # set row space
128    def get_row_space(self):
129        return self.empty_row
130    def set_row_space(self):
131        return self.empty_row
132
133    # set row space
134    def get_row_space(self):
135        return self.empty_row
136    def set_row_space(self):
137        return self.empty_row
138
139    # set row space
140    def get_row_space(self):
141        return self.empty_row
142    def set_row_space(self):
143        return self.empty_row
144
145    # set row space
146    def get_row_space(self):
147        return self.empty_row
148    def set_row_space(self):
149        return self.empty_row
150
151    # set row space
152    def get_row_space(self):
153        return self.empty_row
154    def set_row_space(self):
155        return self.empty_row
156
157    # set row space
158    def get_row_space(self):
159        return self.empty_row
160    def set_row_space(self):
161        return self.empty_row
162
163    # set row space
164    def get_row_space(self):
165        return self.empty_row
166    def set_row_space(self):
167        return self.empty_row
168
169    # set row space
170    def get_row_space(self):
171        return self.empty_row
172    def set_row_space(self):
173        return self.empty_row
174
175    # set row space
176    def get_row_space(self):
177        return self.empty_row
178    def set_row_space(self):
179        return self.empty_row
180
181    # set row space
182    def get_row_space(self):
183        return self.empty_row
184    def set_row_space(self):
185        return self.empty_row
186
187    # set row space
188    def get_row_space(self):
189        return self.empty_row
190    def set_row_space(self):
191        return self.empty_row
192
193    # set row space
194    def get_row_space(self):
195        return self.empty_row
196    def set_row_space(self):
197        return self.empty_row
198
199    # set row space
200    def get_row_space(self):
201        return self.empty_row
202    def set_row_space(self):
203        return self.empty_row
204
205    # set row space
206    def get_row_space(self):
207        return self.empty_row
208    def set_row_space(self):
209        return self.empty_row
210
211    # set row space
212    def get_row_space(self):
213        return self.empty_row
214    def set_row_space(self):
215        return self.empty_row
216
217    # set row space
218    def get_row_space(self):
219        return self.empty_row
220    def set_row_space(self):
221        return self.empty_row
222
223    # set row space
224    def get_row_space(self):
225        return self.empty_row
226    def set_row_space(self):
227        return self.empty_row
228
229    # set row space
230    def get_row_space(self):
231        return self.empty_row
232    def set_row_space(self):
233        return self.empty_row
234
235    # set row space
236    def get_row_space(self):
237        return self.empty_row
238    def set_row_space(self):
239        return self.empty_row
240
241    # set row space
242    def get_row_space(self):
243        return self.empty_row
244    def set_row_space(self):
245        return self.empty_row
246
247    # set row space
248    def get_row_space(self):
249        return self.empty_row
250    def set_row_space(self):
251        return self.empty_row
252
253    # set row space
254    def get_row_space(self):
255        return self.empty_row
256    def set_row_space(self):
257        return self.empty_row
258
259    # set row space
260    def get_row_space(self):
261        return self.empty_row
262    def set_row_space(self):
263        return self.empty_row
264
265    # set row space
266    def get_row_space(self):
267        return self.empty_row
268    def set_row_space(self):
269        return self.empty_row
270
271    # set row space
272    def get_row_space(self):
273        return self.empty_row
274    def set_row_space(self):
275        return self.empty_row
276
277    # set row space
278    def get_row_space(self):
279        return self.empty_row
280    def set_row_space(self):
281        return self.empty_row
282
283    # set row space
284    def get_row_space(self):
285        return self.empty_row
286    def set_row_space(self):
287        return self.empty_row
288
289    # set row space
290    def get_row_space(self):
291        return self.empty_row
292    def set_row_space(self):
293        return self.empty_row
294
295    # set row space
296    def get_row_space(self):
297        return self.empty_row
298    def set_row_space(self):
299        return self.empty_row
300
301    # set row space
302    def get_row_space(self):
303        return self.empty_row
304    def set_row_space(self):
305        return self.empty_row
306
307    # set row space
308    def get_row_space(self):
309        return self.empty_row
310    def set_row_space(self):
311        return self.empty_row
312
313    # set row space
314    def get_row_space(self):
315        return self.empty_row
316    def set_row_space(self):
317        return self.empty_row
318
319    # set row space
320    def get_row_space(self):
321        return self.empty_row
322    def set_row_space(self):
323        return self.empty_row
324
325    # set row space
326    def get_row_space(self):
327        return self.empty_row
328    def set_row_space(self):
329        return self.empty_row
330
331    # set row space
332    def get_row_space(self):
333        return self.empty_row
334    def set_row_space(self):
335        return self.empty_row
336
337    # set row space
338    def get_row_space(self):
339        return self.empty_row
340    def set_row_space(self):
341        return self.empty_row
342
343    # set row space
344    def get_row_space(self):
345        return self.empty_row
346    def set_row_space(self):
347        return self.empty_row
348
349    # set row space
350    def get_row_space(self):
351        return self.empty_row
352    def set_row_space(self):
353        return self.empty_row
354
355    # set row space
356    def get_row_space(self):
357        return self.empty_row
358    def set_row_space(self):
359        return self.empty_row
360
361    # set row space
362    def get_row_space(self):
363        return self.empty_row
364    def set_row_space(self):
365        return self.empty_row
366
367    # set row space
368    def get_row_space(self):
369        return self.empty_row
370    def set_row_space(self):
371        return self.empty_row
372
373    # set row space
374    def get_row_space(self):
375        return self.empty_row
376    def set_row_space(self):
377        return self.empty_row
378
379    # set row space
380    def get_row_space(self):
381        return self.empty_row
382    def set_row_space(self):
383        return self.empty_row
384
385    # set row space
386    def get_row_space(self):
387        return self.empty_row
388    def set_row_space(self):
389        return self.empty_row
390
391    # set row space
392    def get_row_space(self):
393        return self.empty_row
394    def set_row_space(self):
395        return self.empty_row
396
397    # set row space
398    def get_row_space(self):
399        return self.empty_row
400    def set_row_space(self):
401        return self.empty_row
402
403    # set row space
404    def get_row_space(self):
405        return self.empty_row
406    def set_row_space(self):
407        return self.empty_row
408
409    # set row space
410    def get_row_space(self):
411        return self.empty_row
412    def set_row_space(self):
413        return self.empty_row
414
415    # set row space
416    def get_row_space(self):
417        return self.empty_row
418    def set_row_space(self):
419        return self.empty_row
420
421    # set row space
422    def get_row_space(self):
423        return self.empty_row
424    def set_row_space(self):
425        return self.empty_row
426
427    # set row space
428    def get_row_space(self):
429        return self.empty_row
430    def set_row_space(self):
431        return self.empty_row
432
433    # set row space
434    def get_row_space(self):
435        return self.empty_row
436    def set_row_space(self):
437        return self.empty_row
438
439    # set row space
440    def get_row_space(self):
441        return self.empty_row
442    def set_row_space(self):
443        return self.empty_row
444
445    # set row space
446    def get_row_space(self):
447        return self.empty_row
448    def set_row_space(self):
449        return self.empty_row
450
451    # set row space
452    def get_row_space(self):
453        return self.empty_row
454    def set_row_space(self):
455        return self.empty_row
456
457    # set row space
458    def get_row_space(self):
459        return self.empty_row
460    def set_row_space(self):
461        return self.empty_row
462
463    # set row space
464    def get_row_space(self):
465        return self.empty_row
466    def set_row_space(self):
467        return self.empty_row
468
469    # set row space
470    def get_row_space(self):
471        return self
```

```
140
141
142 # Given empty space to fill
143 def (empty) is 0:
144     # Printout count Move
145     self.setCountMove(self.getCountMove()+1)
146     # Set previousMove
147     self.setPreviousMove("L")
148
149     # Loop through
150     alist = self.getList(empty, empty)
151     element = self.getList(empty, empty[0])
152     self.setList(empty, empty[0], element)
153     self.setList(empty, empty[1], alist)
154     self.setPreviousMove(empty[0])
155
156 # PRINT OUT THE
157 def printMatrix(self):
158     print("Matrix: ")
159     for i in range(self.rowSize):
160         print(i, end=" ")
161         for j in range(self.colSize):
162             element = self.getList(i, j)
163             if element == 0:
164                 print(" ", end=" ")
165             else:
166                 print(element, end=" ")
167         print("\n")
168     print("\n")
169     print("Matrix: ")
```


3. SCREENSHOT INPUT DAN OUTPUT

3.1. testSolvable1.txt

3.1.1. Input

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

3.1.2. Output

```
15-PUZZLE
SOLVER

BY:
EIFFEL AQILA AMARENDRA - 13520074

=====

Pilihan input method:
1. Menggunakan test case yang tersedia
2. Menggunakan matrix yang dibuat secara acak
3. Menggunakan file baru
Masukkan pilihan file input method (1-2): 1

Test case yang tersedia:
1. testSolvable1.txt
2. testSolvable2.txt
3. testSolvable3.txt
4. testUnsolvabel1.txt
5. testUnsolvabel2.txt
Masukkan pilihan file (1-5): 1

File input: ./test/testSolvable1.txt

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

Nilai fungsi Murang(i):
=====
| ELMEN(i) | MURANG(i) |
=====
| 1 | 0 |
| 2 | 0 |
| 3 | 0 |
| 4 | 0 |
| 5 | 0 |
| 6 | 0 |
| 16 | 9 |
| 8 | 1 |
| 9 | 1 |
| 10 | 1 |
| 7 | 0 |
| 11 | 0 |
| 13 | 1 |
| 14 | 1 |
| 15 | 1 |
| 12 | 0 |
=====

Nilai dari TOTALKURANG(i) + X: 16

Memproses persoalan...
Persoalan yang kompleks terkadang membutuhkan waktu yang cukup lama...

Urutan Matrix dari Posisi Awal ke Goal State:
=====
| 1   2   3   4 |
| 5   6   7   8 |
| 9  10  11  12 |
| 13  14  15  12 |
=====

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

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

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

Total waktu yang diperlukan: 0.00 detik
Jumlah simpul yang dibangkitkan (termasuk posisi awal): 10
```

3.2. testSolvable2.txt

3.2.1. Input

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

3.2.2. Output

File input: ./test/testSolvable2.txt

Matriks posisi awal 15-puzzle:

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

Nilai fungsi Kurang(i):

ELHT(i)	KURANG(i)
12	11
13	11
14	11
18	14
11	10
17	12
12	15
11	14
16	7
5	10
6	10
10	11
9	10
13	10
14	10
15	10

Nilai dari TOTALKURANG(i) + X: 26

Memproses persoalan...

Persoalan yang kompleks terkadang membutuhkan waktu yang cukup lama...

Urutan Matrix dari Posisi Awal ke Goal State:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Total waktu yang diperlukan: 0.42 detik

Jumlah simpul yang dibangkitkan (termasuk posisi awal): 1659

3.3. testSolvable3.txt

3.3.1. Input

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

3.3.2. Output

```
File input: ./test/testSolvable3.txt

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

Nilai fungsi Kurang(i):
=====
|ELMT(i)| KURANG(i)|
=====
|1| 0|
|2| 0|
|7| 4|
|8| 4|
|5| 2|
|6| 2|
|15| 0|
|3| 0|
|9| 1|
|10| 1|
|4| 0|
|16| 4|
|13| 2|
|14| 2|
|12| 1|
|11| 0|
=====

Nilai dari TOTALKURANG(i) + X: 32

Memproses persoalan...
Persoalan yang kompleks terkadang membutuhkan waktu yang cukup lama...

Urutan Matrix dari Posisi Awal ke Goal State:

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Total waktu yang diperlukan: 0.66 detik
Jumlah simpul yang dibangkitkan (termasuk posisi awal): 2742
```

3.4. testUnsolvable1.txt

3.4.1. Input

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

3.4.2. Output

```
File input: ./test/testUnsolvable1.txt

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

Nilai fungsi Kurang(i):
=====|
|ELMT(i)| |KURANG(i)|
=====|
|1|      |0|
|3|      |1|
|4|      |1|
|15|     |11|
|2|      |0|
|16|     |10|
|5|      |0|
|12|     |6|
|7|      |1|
|6|      |0|
|11|     |3|
|14|     |4|
|8|      |0|
|9|      |0|
|10|     |0|
|13|     |0|
=====|

Nilai dari TOTALKURANG(i) + X: 37

Status tujuan permainan tidak dapat dicapai.

Total waktu yang diperlukan: 0.00 detik

Jumlah simpul yang dibangkitkan (termasuk posisi awal): 1
```

3.5. testUnsolvale2.txt

3.5.1. Input

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

3.5.2. Output

```
Pilihan input method:
1. Menggunakan test case yang tersedia
2. Menggunakan matrix yang dibuat secara acak
3. Menggunakan file baru
Masukkan pilihan file input method (1-2): 1

Test case yang tersedia:
1. testSolvable1.txt
2. testSolvable2.txt
3. testSolvable3.txt
4. testUnsolvale1.txt
5. testUnsolvale2.txt
Masukkan pilihan file (1-5): 5

File input: ./test/testUnsolvale2.txt

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

Nilai fungsi Kurang(i):
=====
|ELMT(i)| KURANG(i)|
=====
|1|0|
|2|0|
|7|4|
|8|4|
|5|2|
|15|9|
|6|2|
|3|0|
|9|1|
|10|1|
|4|0|
|16|4|
|13|2|
|14|2|
|12|1|
|11|0|
=====

Nilai dari TOTALKURANG(i) + X: 33

Status tujuan permainan tidak dapat dicapai.

Total waktu yang diperlukan: 0.00 detik

Jumlah simpul yang dibangkitkan (termasuk posisi awal): 1
```

4. BERKAS TEKS CONTOH INSTANSIASI PERSOALAN 15-PUZZLE

4.1. testSolvable1.txt

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

4.2. testSolvable2.txt

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

4.3. testSolvable3.txt

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

4.4. testUnsolvable1.txt

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

4.5. testUnsolvale2.txt

1 2 7 8

5 15 6 3

9 10 4 16

13 14 12 11

5. TAUTAN *REPOSITORY* GITHUB

Tautan *repository* Github kode program ini adalah sebagai berikut

https://github.com/eiffelaqila/Tucil3_13520074

6. CEKLIST

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		√