**LAPORAN TUGAS KECIL 3**

**IF2211 STRATEGI ALGORITMA**

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

Logo

Description automatically generated

**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* (ĉ(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
8. ***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 *source code* program,

* 1. **main.py**

|  |
| --- |
|  |

* 1. **solver.py**

|  |
| --- |
|  |

* 1. **matrix.py**

|  |
| --- |
|  |

1. ***SCREENSHOT* *INPUT* DAN *OUTPUT***
   1. **testSolvable1.txt**
      1. **Input**

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

* + 1. **Output**

|  |
| --- |
| **­**  **Visualisasi Awal**    **Visualisasi Akhir** |

* 1. **testSolvable2.txt**
     1. **Input**

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

* + 1. **Output**

|  |
| --- |
| **Visualisasi Awal**    **Visualisasi Akhir** |

* 1. **testSolvable3.txt**
     1. **Input**

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

* + 1. **Output**

|  |
| --- |
| **Visualisasi Awal**    **Visualisasi Akhir** |

* 1. **testUnsolvable1.txt**
     1. **Input**

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

* + 1. **Output**

|  |
| --- |
| **Visualisasi Awal**    **Visualisasi Akhir** |

* 1. **testUnsolvable2.txt**
     1. **Input**

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

* + 1. **Output**

|  |
| --- |
| **Visualisasi Awal**    **Visualisasi Akhir** |

* 1. **Random Generated Input**
     1. **Input**

|  |
| --- |
|  |

* + 1. **Output**

|  |
| --- |
| **Visualisasi Awal**    **Visualisasi Akhir** |

1. **BERKAS TEKS CONTOH INSTANSIASI PERSOALAN 15-PUZZLE**
   1. **testSolvable1.txt**

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

* 1. **testSolvable2.txt**

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

* 1. **testSolvable3.txt**

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

* 1. **testUnsolvable1.txt**

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

* 1. **testUnsolvable2.txt**

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

1. **TAUTAN *REPOSITORY* GITHUB**

Tautan *repository* Github kode program ini adalah sebagai berikut

<https://github.com/eiffelaqila/Tucil3_13520074>

1. **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 | √ |  |