LAPORAN TUGAS KECIL III

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

Laporan Ini Dibuat untuk Memenuhi Tugas Perkuliahan

Mata Kuliah Strategi Algoritma (IF2211)

**KELAS 01**

**Dosen : Dr. Masayu Leylia Khodra, S.T., M.T.**

**Logo

Description automatically generated**

Rio Alexander Audino / 13520088

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**SEMESTER II TAHUN 2021/2022**

1. Cara Kerja Algoritima Branch and Bound

Program yang kali ini saya buat adalah implementasi pemecahan 15-Puzzle menggunakan algoritma Branch and Bound. Pencarian solusi 15-Puzzle dilakukan dengan algoritma Branch and Bound. Secara umum, terdapat beberapa langkah dalam implementasi algoritma ke dalam pemecahan masalah, yaitu:

1. Memvalidasi puzzle sebelum mulai diselesaikan
2. Jika puzzle dapat diselesaikan, puzzle dijadikan root node
3. Dimulai dari root node, menambahkan child node pada tiap gerakan ubin kosong memungkinkan.
4. Mengecek kondisi node apakah pernah ada/dibentuk sebelumnya
5. Melakukan kalkulasi *cost* pada tiap node yang terbentuk
6. Memilih node dengan dengan *cost* terkecil
7. Melakukan langkah c-e sampai menemukan node dengan ubin yang tersusun rapi
8. Ubin telah tersusun rapi dan 15-Puzzle berhasil dipecahkan

Secara detil, proses validasi dilakukan menggunakan fungsi KURANG. Fungsi KURANG akan memvalidasi puzzle melalui perbandingan posisi ubin sekarang dengan posisi ubin *goal.* Selain itu, proses perhitungan nilai *bound* atau *cost* didasarkan pada dua hal, yakni kedalaman node dan taksiran *cost* node sekarang sampai ke *goal*. Taksiran *cost* yang digunakan adalah jumlah ubin yang tak sesuai.

Selain proses di atas, terdapat juga beberapa hal yang membantu saya dalam implementasi algoritma BnB. Pada program ini, saya juga membuat dua kelas baru, Puzzle dan Node. Kelas Puzzle menangani segala proses yang terjadi pada ubin, seperti perhitungan taksiran cost, pergeseran ubin kosong, dll. Kelas Node menangani segala proses yang dibutuhkan oleh sebuah node, seperti penyimpanan informasi, proses output, serta pengecekan node.

1. Input-Output Program

Contoh Input :

Memasukan nama file serta delay penggambaran proses di akhir

A screenshot of a computer

Description automatically generated with medium confidence

Contoh Output:

Proses penggambaran langkah-langkah penyelesaian dan hasil akhir

Text

Description automatically generated Text

Description automatically generated

1. Checklist

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi | ✓ |  |
| 1. Program berhasil dijalankan | ✓ |  |
| 1. Program dapat menerima input dan menuliskan output | ✓ |  |
| 1. Luaran sudah benar untuk semua data uji | ✓ |  |
| 1. Bonus dibuat |  | ✓ |

1. Kode Program

File Main.py:

Text

Description automatically generated

Text

Description automatically generated

file Puzzle.py:

Text

Description automatically generated

A screenshot of a computer screen

Description automatically generated with medium confidence Text

Description automatically generated

File Node.py:

Text

Description automatically generated Text

Description automatically generated

File PuzzleCheck.py :

Text

Description automatically generated Text

Description automatically generated

File BnBAlgorithm.py:

Text

Description automatically generated Text

Description automatically generated

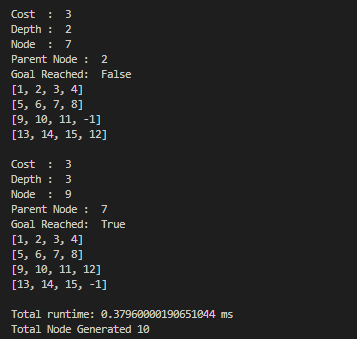
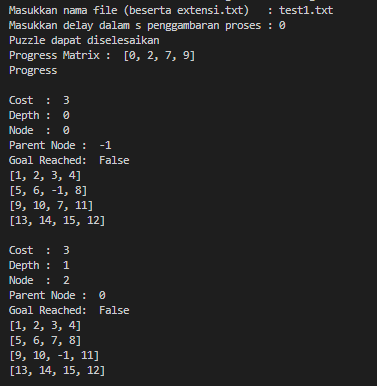
File PostProc.py :

A screenshot of a computer

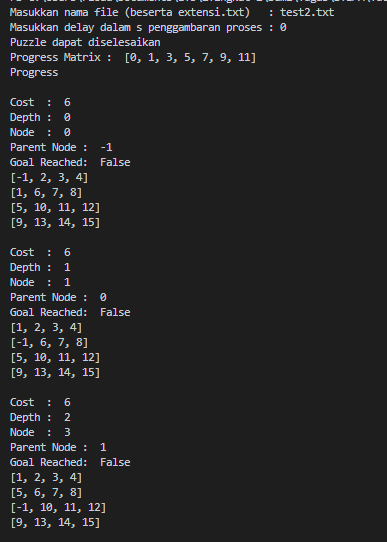
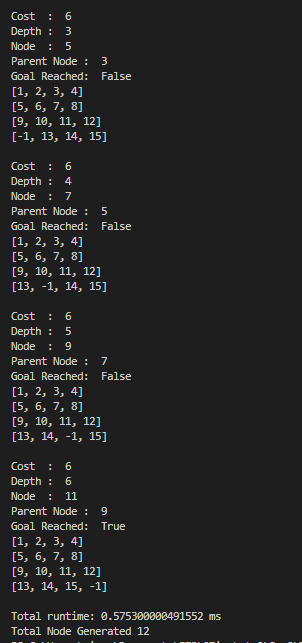
Description automatically generated with medium confidence

1. Testing

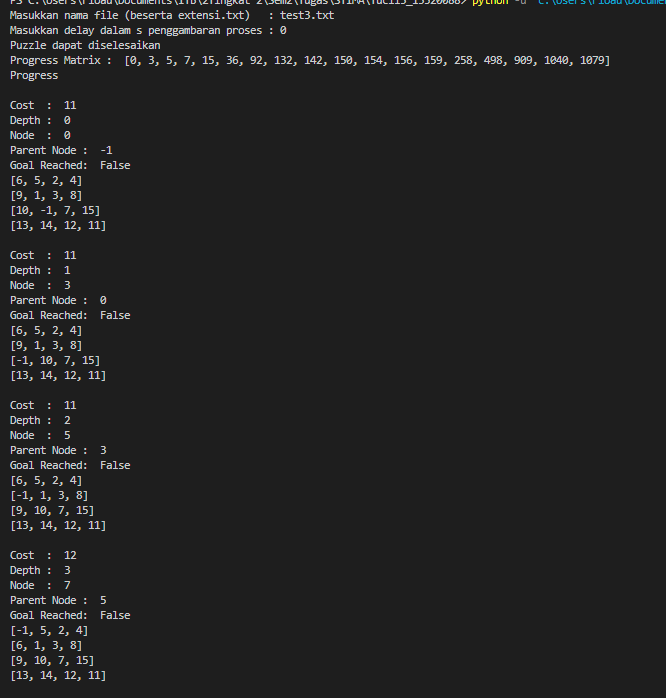
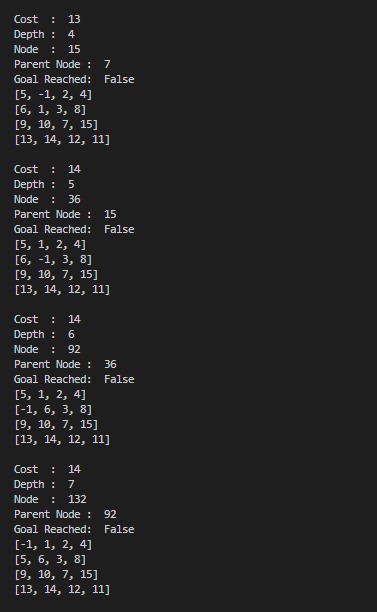
test1.txt :

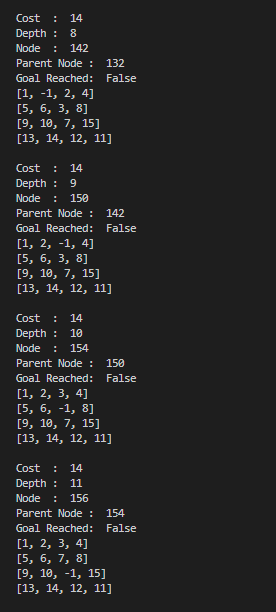
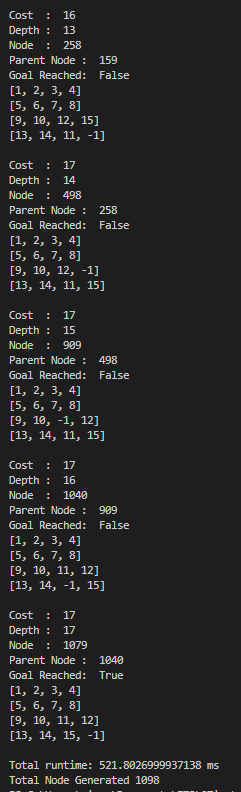


test2.txt :

test3.txt :

test4.txt :

A screenshot of a computer

Description automatically generated with medium confidence

test5.txt :

A screenshot of a computer

Description automatically generated with medium confidence

1. Sumber Kode Program

Link Github : <https://github.com/Audino723/Tucil3_13520088>