**LAPORAN TUGAS KECIL III**

**PENYELESAIAN PERSOALAN 15-PUZZLE DENGAN ALGORITMA BRANCH AND BOUND**

Laporan dibuat untuk memenuhi salah satu tugas mata kuliah

IF2211 Strategi Algoritma



Disusun oleh:

**Kristo Abdi Wiguna 13520058**

**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2022**

## DAFTAR ISI

[DAFTAR ISI 1](#_Toc99681200)

[Algoritma Branch And Bound 2](#_Toc99681201)

[Source Program 4](#_Toc99681202)

[Berkas File Uji 11](#_Toc99681203)

[Screenshot Input dan Output 12](#_Toc99681204)

[Link Kode Program 22](#_Toc99681205)

[Checklist 22](#_Toc99681206)

## Algoritma Branch And Bound

Algoritma BnB pada penyelesaian 15 puzzle akan meliputi class Puzzle yang dipanggil pertama kali setelah program sudah menerima matriks yang akan dipecahkan. Puzzle memiliki konstruktor yang terdiri dar 5 member yang berjenis dictionary yaitu tracker, linkedList, costDict, parentDict, dan depthDict. Tracker berguna untuk memasukkan tiap node masing – masing state matriks yang pernah dikunjungi untuk pruning langkah gerak tile kosong. Linked list berguna untuk menyimpan matriks simpul yang dieksplorasi sesuai gerak tile kosong. Untuk cost function c(i) = f(n) + h(n), costDict berguna untuk menyimpan taksiran jarak dari simpul ke goal state dengan menggunaka jumlah cost yang dimiliki tiap simpul yaitu jumlah letak tile yang tidak sesuai dengan goal state yaitu h(n). DepthDict berguna untuk menampung kedalaman atau jarak ke simpul dari akar yaitu f(n).

Pertama – tama, matriks akan ditentukan apakah dapat berjalan dengan Kurang(i) + X. Kurang(i) adalah jumlah dari tile yang memiliki posisi setelah nomor i yang kurang nilainya dari i. Paritas X adalah nilai biner tile kosong berada di state tile yang diarsir dibawah ini. Jika sesuai maka X bernilai satu dan sebaliknya nol. Jika nilai genap, maka matriks bisa dipecahkan dan sebaliknya jika ganjil maka tidak dapat dipecahkan.

A picture containing crossword puzzle

Description automatically generated

Ketika matrix dapat dipecahkan, maka lanjut ke langkah dimana matrix akan dijadikan dalam bentuk 1D array terlebih dahulu demi optimasi sistem. Matriks akan dihash sehingga untuk keperluan pruning tidak mengunjungi state matriks yang sama menjadi lebih cepat. Lalu, akan disimpan cost, depth, indeks root, dan state matriks di costDict, depthDict, parentDict dan linkedList. Algoritma branch and bound ingin memilih simpul dengan cost terkecil sehingga akan membuat child dari simpul tersebut, untuk itu digunakan priority queue yang menyimpan cost matriks pertama dan key simpul pertama.

Setelah itu, selama tidak ditemukan goal state dan priority queue masih berisi, loop akan dijalankan untuk mengenerate child dari simpul yang telah diambil dari priority queue dengan cost terminimum. Child yang digenerate adalah matriks dari simpul yang tile kosong digerakkan atas, bawah, kiri, dan kanan dan dimasukkan ke dalam directionMatrix jika pergerakan tile kosong valid dengan memanggil fungsi move. Ketika simpul dengan matriks pernah dikunjungi, maka pruning dilakukan dengan melewati simpul tersebut untuk tidak dikunjungi. Setelah sebuah simpul dikunjungi, maka cost dan depth akan dihitung dan state matriks simpul disimpan dalam tracker. Jika simpul yang sedang dieksplorasi adalah goal state, maka loop akan berhenti, jika bukan maka akan memasukkan simpul ke dalam priority queue. Hal ini dilakukan terus menerus hingga loop berhenti dan mendapat jumlah simpul yang dibangkitkan dan key dari goal node.

## Source Program

Menggunakan bahasa pemrograman Python. Library yang digunakan adalah time, heapq, \_pickle, dan random.

File main.py

Text

Description automatically generated

A picture containing rectangle

Description automatically generated

File node,py

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

File util.py

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated Text

Description automatically generated

File inout.py

Text

Description automatically generated

## Berkas File Uji

File tc.txt

Text

Description automatically generated

File tc2.txt

Text

Description automatically generated with medium confidence

File tc3.txt

Text

Description automatically generated

File tc4.txt

Text

Description automatically generated

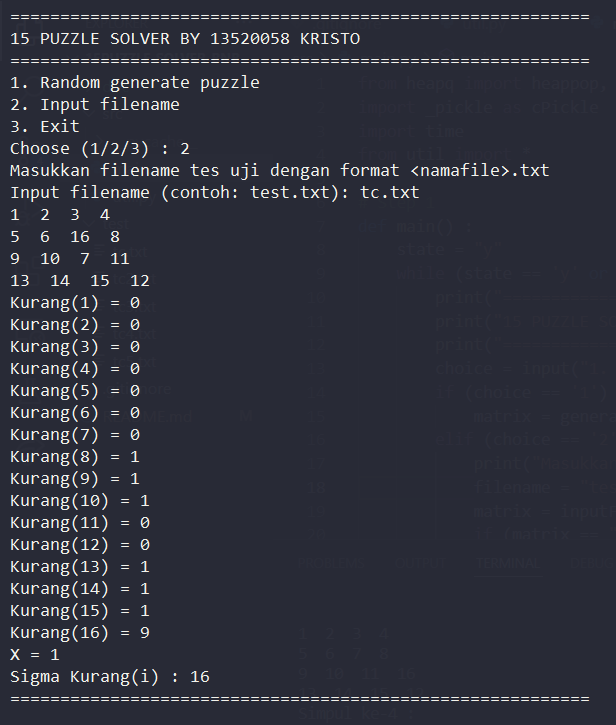
File tc5.txt

Text

Description automatically generated with medium confidence

## Screenshot Input dan Output

File tc.txt



Text

Description automatically generated

File tc2.txt

Text

Description automatically generated

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

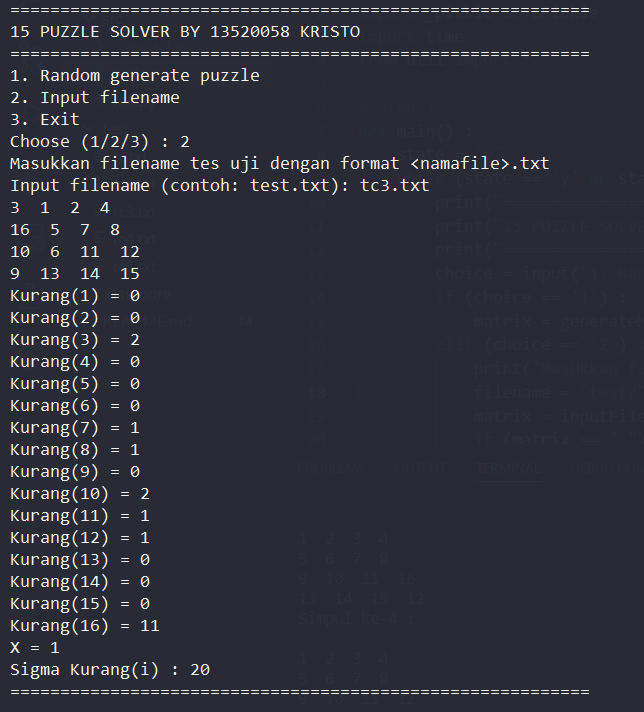
Description automatically generated with medium confidenceA screenshot of a computer

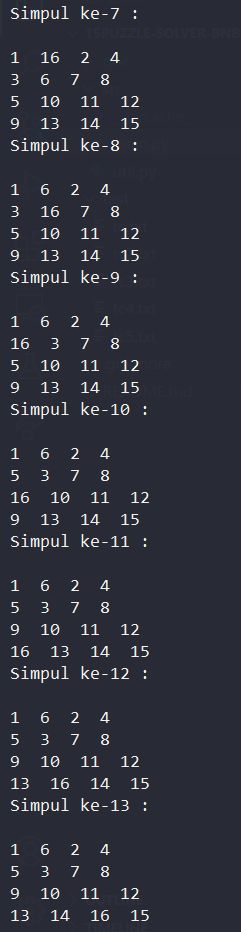
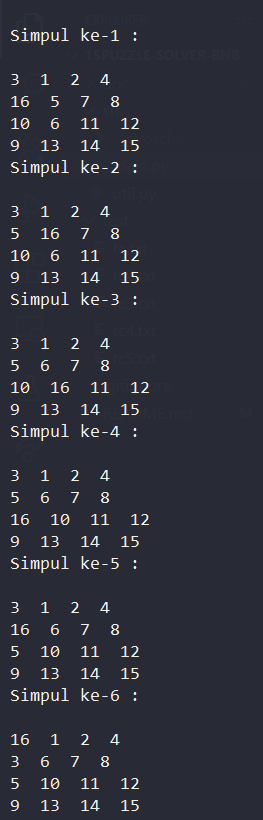
Description automatically generated with medium confidence

Text

Description automatically generated

File tc3.txt



A screenshot of a computer

Description automatically generated with medium confidence

Text

Description automatically generated

File tc4.txt

Text

Description automatically generated

File tc5.txt

Text

Description automatically generated

## Link Kode Program

[*https://github.com/kristabdi/Tucil3\_13520058*](https://github.com/kristabdi/Tucil3_13520058)

## Checklist

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1. Program berhasil dikompilasi | V |  |
| 2. Program berhasil running | V |  |
| 3. Program dapat menerima input dan menuliskan output | V |  |
| 4. Luaran sudah benar untuk semua data uji | V |  |
| 5. Bonus dibuat |  | V |