**Laporan Tugas Kecil III**

IF2211 – Strategi Algoritma

15-Puzzle Solver

Disusun oleh

Stephen Thajeb – 13518150 – K3



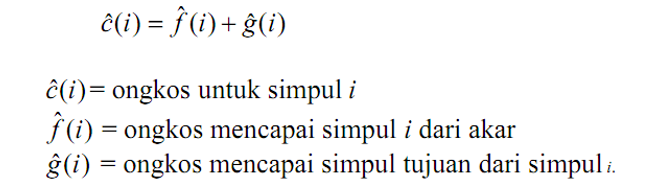
**PROGRAM STUDI TEKNIK INFORMATIKA**

**SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA**

**INSTITUT TEKNOLOGI BANDUNG**

**2020**

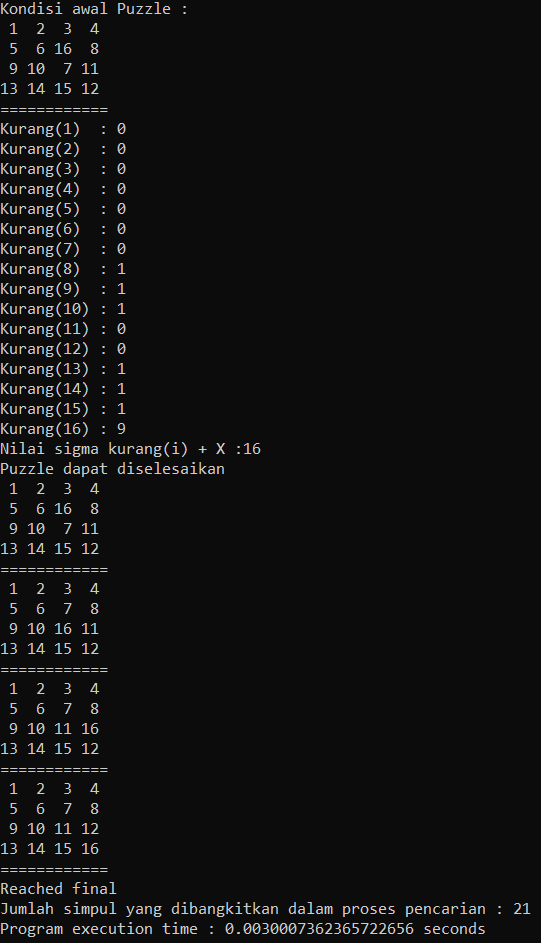
1. **Cara Kerja program dengan Algoritma Branch and Bound**
2. Program akan menerima matriks 4x4 dari file eksternal dan matriks yang merepresentasikan puzzle tersebut akan dievaluasi apakah puzzle tersebut dapat diselesaikan. Apabila dapat diselesaikan maka matriks tersebut dikonstruksi sebagai elemen simpul akar dari tree branch and bound. Untuk setiap simpul yang dibangkitkan secara dinamis akan disimpan informasi nilai cost
3. Membangkitkan anak dari simpul saat ini berdasarkan arah pergeseran ubin kosong yang ditandai dengan nilai 16. Pergeseran yang diizinkan sebesar 1 langkah ke arah up,right,down,left. Untuk setiap simpul baru yang dibangkitkan akan dihitungkan cost node. Nilai cost dapat diperhitung dengan rumus



Gambar 2. Pendekatan fungsi cost untuk persoalan

(sumber: <http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2017-2018/Algoritma-Branch-&-Bound-(2018).pdf>)

1. Semua simpul yang dibangkitkan pada langkah ke 3 akan ditambahkan ke dalam priorityqueue dengan priority nilai cost yang lebih kecil diprioritaskan. Penambahan elemen sesuai dengan aturan First In First Out namun dengan parameter tambahan cost sebagai prioritas
2. Melakukan aksi pop elemen , apabila matriks simpul yang dipop dari PrioQueue identik dengan target maka puzzle sudah berhasil di solve. Apabila berbeda, maka ulangi langkah ke-2.
3. Untuk setiap simpul yang di pop akan disimpan dalam suatu list untuk menandakan bahwa state matriks tersebut sudah pernah dilewati, sehingga untuk pembangkitan simpul ke depannya dapat dihindari pembangkitan simpul dengan matriks duplikat
4. **Screenshot Luaran Program**
   1. **Solvable Testcase**



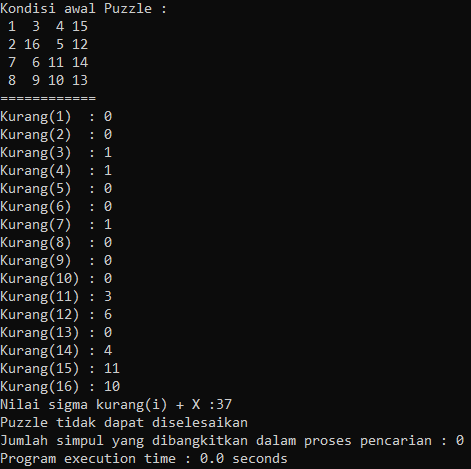
Gambar 2. TestCase 1

|  |  |
| --- | --- |
|  |  |
| Gambar 3. Testcase 2 | Gambar 4. Lanjutan Testcase 2 |

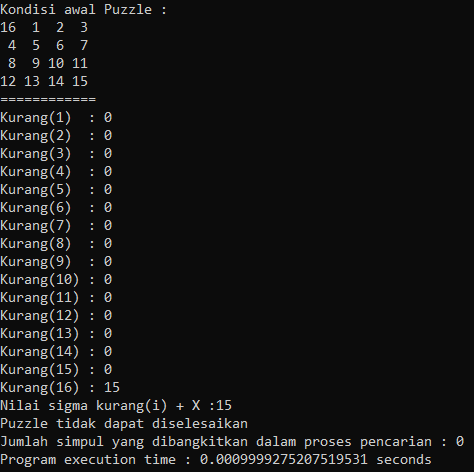
|  |  |
| --- | --- |
|  |  |
| Gambar 5. Testcase 3 | Gambar 6. Lanjutan Testcase 3 |

|  |
| --- |
|  |
| Gambar 7. Lanjutan Testcase 3 |

* 1. Unsolvable Testcase



Gambar 8. Testcase 4



Gambar 9. Testcase 5

|  |  |  |
| --- | --- | --- |
| Poin | Ya | Tidak |
| 1.Program berhasil dikompilasi | √ |  |
| 2.Program berhasil running | √ |  |
| 3.Program dapat menerima input dan output | √ |  |
| 4.Luaran sudah benar untuk semua data uji | √ |  |
| **Poin ke-4 dicentangkan dengan catatan untuk data uji tertentu program akan eksekusi dengan waktu yang lebih lama (+- 30 menit)** | | |