

**TUGAS INTELIGENSI BUATAN**  
**PROBLEM SOLVING AGENT**  
**CLASICAL SEARCH & LOCAL SEARCH**

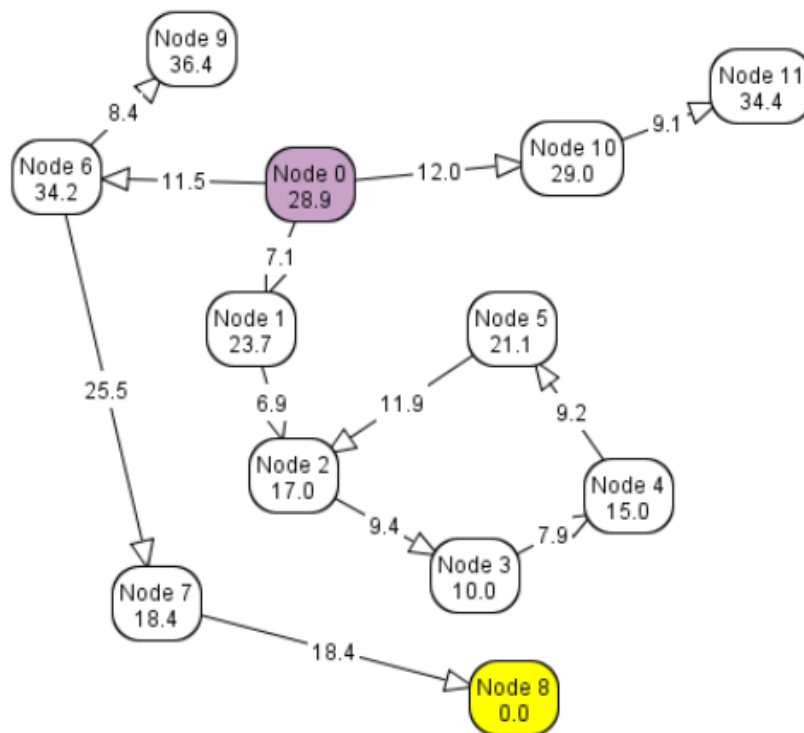


DISUSUN OLEH:  
GENTA ARI ANGGORO (14117078)

**PROGRAM STUDI : TEKNIK INFORMATIKA**  
**KELAS : INTELIGENSI BUATAN - RA**

**INSTITUT TEKNOLOGI SUMATERA**  
**2019**

1. mencari jalur dimulai dari Node\_0 untuk menuju Node\_8, dengan menggunakan teknik A\*.



## Tahapan proses

### Notasi Algoritme

$$f(n) = g(n) + h(n)$$

Keterangan:

1.  $f(n)$  adalah jumlah dari  $g(n)$  dan  $h(n)$ . ini adalah perkiraan jalur terpendek sementara. maka  $f(n)$  adalah jalur terpendek yang sebenarnya yang tidak ditelusuri sampai Algoritme A-Star (A\*) diselesaikan.
2.  $g(n)$ /**Geographical Cost** adalah total jarak yang didapat dari node awal ke node selanjutnya.
3.  $h(n)$ /**Heuristic Cost** adalah perkiraan jarak dari node sekarang (yang sedang dikunjungi) ke node tujuan. sebuah fungsi *heuristic* digunakan untuk membuat perkiraan seberapa jauh lintasan yang akan diambil ke vertek tujuan.

Open List		
Node Asal	Node Tujuan	Cost
		$f(n) = g(n) + h(n)$
Node_0	Node_1	30,8
Node_0	Node_6	45,7
Node_0	Node_10	41
Node_0	Node_2	31
Node_0	Node_3	33,4
Node_0	Node_4	46,3
Node_0	Node_5	61,6
Node_0	Node_10	55,5
Node_0	Node_9	56,3
Node_0	Node_7	55,4
Node_0	Node_8	55,4

Close List		
Node Asal	Node Tujuan	Cost
		$f(n) = g(n) + h(n)$
Node_0	Node_6	45,7
Node_0	Node_7	55,4
Node_0	Node_8	55,4

Jadi, jalur yang diambil untuk node\_0 sampai node\_8 adalah node\_0 → node\_6 → node\_7 → node\_8 dengan cost 55,4

Admissability dari heuristic pada graf terdapat pada heuristic pada node\_1, node\_2, node\_3, node\_4, node\_10, karena seharusnya jalur itu mempunyai cost yang besar, karena tidak mengarah pada solusi

## 2. Magic Square Puzzle

Aturan permainan tersebut adalah Puzzle harus diisi oleh angka antara 1 hingga 9 dan tidak diperbolehkan terdapat dua angka atau lebih yang sama dalam Solusi Puzzle. Jumlah dari angka-angka yang terdapat pada satu baris, kolom dan diagonal yang sama adalah 15

setiap penjumlahan baris, kolom, dan diagonal memiliki konstanta yang sama

$$const = \frac{n(n^2+1)}{2}$$

### Aturan magic square

- I. Posisi awal untuk angka 1 (satu)  $\rightarrow (n/2, n-1)$
- II. Tentukan posisi  $(i, j) \rightarrow (i-1, j+1)$
- III. Jika  $i = -1$  maka  $i = n-1$
- IV. Jika  $j = n$  maka  $j = 0$
- V. Jika posisi sudah di tempati maka  $i = i+1, j = j-2$
- VI. Jika  $i = -1$  dan  $j = n$  maka  $i = 0$  dan  $j = n-2$

Ket.

i : baris

j : kolom

n : matrik  $n \times n$  (yang digunakan disini adalah 3)

	0	1	2
0			
1			
2			

Setiap cell akan diisi dengan elemen 1,2,3,4,5,6,7,8,9

### Langkah 1

Posisi awal elemen 1

$(n/2, n-1) \rightarrow (3/2, 3-1)$  jadi  $(1, 2)$  satu karena interjer, maka angka 1 berada pada baris 1 kolom 2

	0	1	2
0			
1			1
2			

### Langkah 2

Posisi elemen 2,  $(1, 2) \rightarrow (1-1, 2+1)$  menjadi  $(0, 3)$ , karena  $j = n$  maka  $j$  menjadi 0 jadi posisi cell menjadi  $(0, 0)$

	0	1	2
0	2		
1			1
2			

### Langkah 3

Posisi elemen 3,  $(0, 0) \rightarrow (0-1, 0+1)$  menjadi  $(-1, 1)$ , karena  $i = -1$  maka  $i = n-1$  menjadi 2 jadi posisi cell  $(2, 1)$

	0	1	2
0	2		
1			1
2		3	

### Langkah 4

Posisi elemen 4,  $(2, 1) \rightarrow (2-1, 1+1)$  menjadi  $(1, 2)$ , karena posisi sudah terisi maka  $i = i + 1$  dan  $j = j - 2$  menjadi  $(1+1, 2-2)$  jadi posisi cell  $(2, 0)$

	0	1	2
0	2		
1			1
2	4	3	

### Langkah 5

Posisi elemen 5,  $(2, 0) \rightarrow (2-1, 0+1)$  menjadi  $(1, 1)$  jadi posisi cell  $(1, 1)$

	0	1	2
0	2		
1		5	1
2	4	3	

### Langkah 6

Posisi elemen 6,  $(1, 1) \rightarrow (1-1, 1+1)$  menjadi  $(0, 2)$  jadi posisi cell  $(0, 2)$

	0	1	2
0	2		6
1		5	1
2	4	3	

### Langkah 7

Posisi elemen 7,  $(0, 2) \rightarrow (0-1, 2+1)$  menjadi  $(-1, 3)$ , ini adalah spesial case karena  $i$  dan  $j$  kedua – duanya berada di luar cell yang ada, maka kita menggunakan aturan no 4 yaitu Jika  $i = -1$  dan  $j = n$  maka  $i = 0$  dan  $j = n-2$  jadi posisi cell nya menjadi  $(0, 1)$

	0	1	2
0	2	7	6
1		5	1
2	4	3	

Perbedaan antara aturan II, IV dan V adalah di aturan II dan IV hanya  $i$  dan  $j$  hanya salah satunya yang berada di luar cell sedangkan aturan V  $i$  dan  $j$  kedua – duanya berada di luar cell

### Langkah 8

Posisi elemen 8,  $(0, 1) \rightarrow (0-1, 1+1)$  menjadi  $(-1, 1)$  karena  $i = -1$  maka  $i = n-j$  menjadi 2 jadi, posisi cell  $(2, 2)$

	0	1	2
0	2	7	6
1		5	1
2	4	3	8

### Langkah 9

Posisi elemen 9,  $(2, 2) \rightarrow (2-1, 2+1)$  menjadi  $(1, 3)$  karena  $j = n$  maka  $j$  menjadi 0 jadi posisi cell  $(1, 0)$

	0	1	2
0	2	7	6
1	9	5	1
2	4	3	8

## Source code

```
#include <bits/stdc++.h>
using namespace std;

void magicSquare(int n)
{
    int Square[n][n];

    memset(Square, 0, sizeof(Square));

    int i = n/2;
    int j = n-1;

    for (int num = 1; num <= n*n; )
    {
        if (i == -1 && j == n)
        {
            j = n-2;
            i = 0;
        }
        else
        {
            if (j == n)
                j = 0;

            if (i < 0)
                i = n - 1;

            if (Square[i][j])
            {
                j -= 2;
                i++;
                continue;
            }
            else
                Square[i][j] = num++;

            j++; i--;
        }
    }

    cout<<"Magic Square untuk n="<<n<<"\njumlah semua
kolom dan baris = "<<n*(n*n+1)/2<<"\n\n";
    for (i = 0; i < n; i++)
```

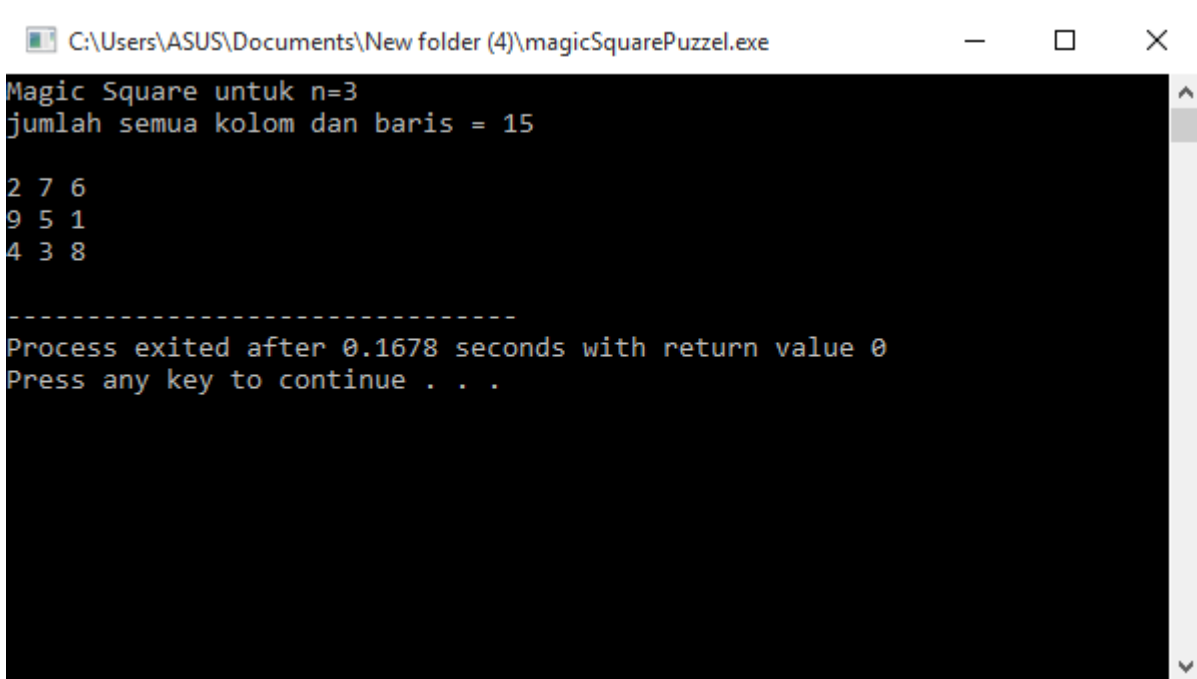


```

    {
        for (j = 0; j < n; j++)
            cout<<Square[i][j]<<" ";
        cout<<endl;
    }
}

int main()
{
    int n = 3; // Works only when n is odd
    magicSquare (n);
    return 0;
}

```



The screenshot shows a Windows command prompt window titled "C:\Users\ASUS\Documents\New folder (4)\magicSquarePuzzel.exe". The output of the program is as follows:

```

Magic Square untuk n=3
jumlah semua kolom dan baris = 15

2 7 6
9 5 1
4 3 8

-----
Process exited after 0.1678 seconds with return value 0
Press any key to continue . . .

```

## Daftar Pustaka

- Barnwal, A. (2019). *Magic Square*. Diambil kembali dari GeeksforGeeks:  
<https://www.geeksforgeeks.org/magic-square/>
- Khayde, V. (2018, January 20). *Magic Square Construction Algorithm N x N*.  
 Diambil kembali dari Youtube:  
[https://www.youtube.com/watch?v=GYBr8n-\\_Rq4](https://www.youtube.com/watch?v=GYBr8n-_Rq4)