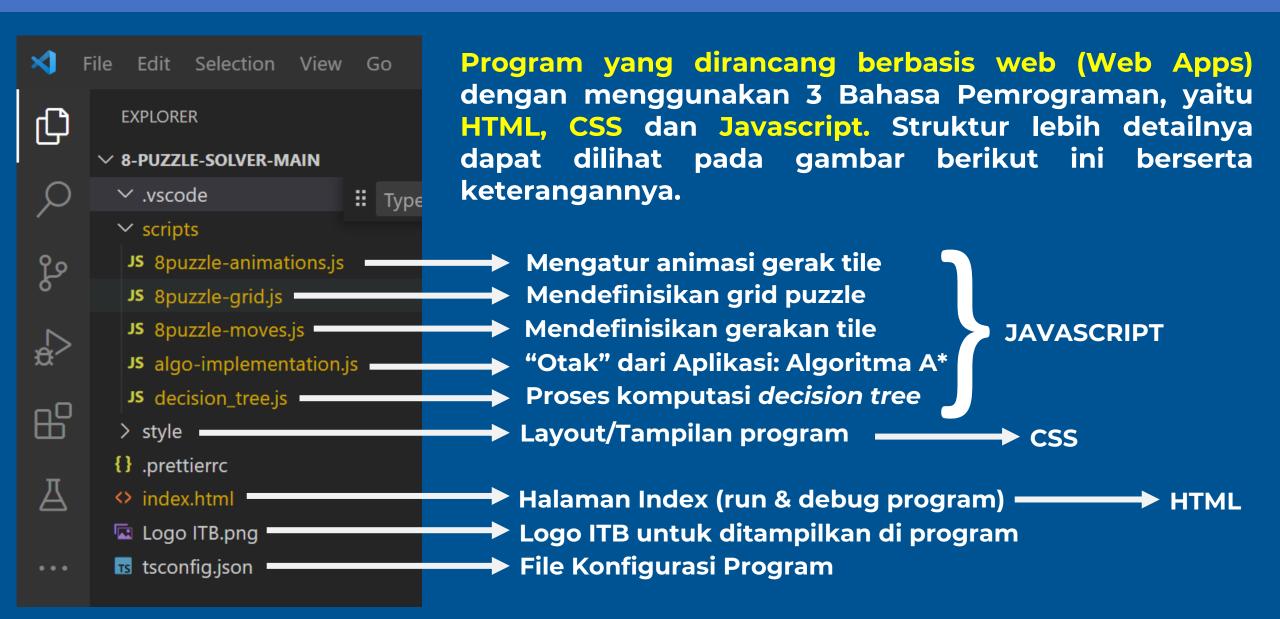


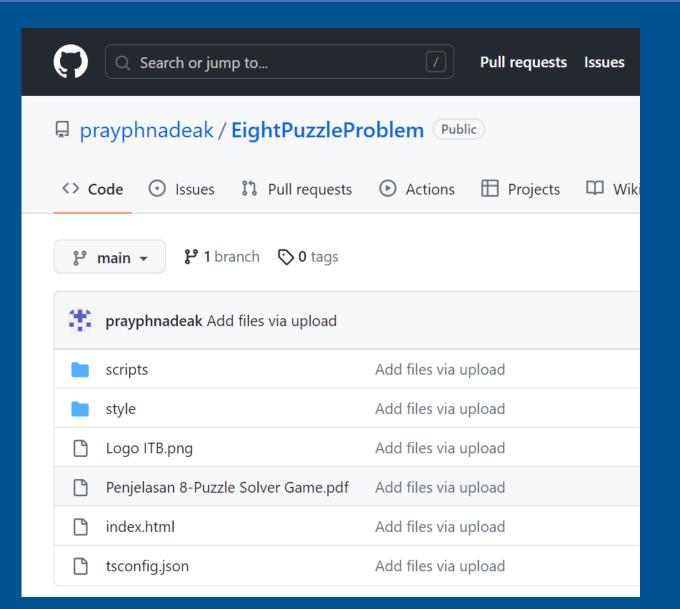
# 8-PUZZLE SOLVER GAME Using A\* Algorithm

Pray Putra Hasianro Nadeak (NIM. 23222048)

### Struktur Program



# Repository Program



Seluruh file terkait repository program dapat diakses pada tautan GitHub Berikut ini:

https://github.com/prayphnadeak/ /EightPuzzleProblem

# Tampilan *Initial State* Program



UTS Sistem Intelijen 2023 | Nama: Pray Putra Hasianro Nadeak | NIM: 23222048 | Angkatan: 2022 | Prodi: Magister Teknik Elektro | Opsi: Layanan Teknologi Informasi(LTI)

#### 8-Puzzle Solver



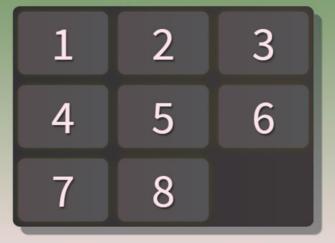
RESET

# Tampilan *Goal State* Program



Intelijen 2023 | Nama: Pray Putra Hasianro Nadeak | NIM: 23222048 | Angkatan: 2022 | Prodi: Magister Teknik Elektro | Opsi: Layanan Teknologi Informasi(LTI)

#### Puzzle Selesai!

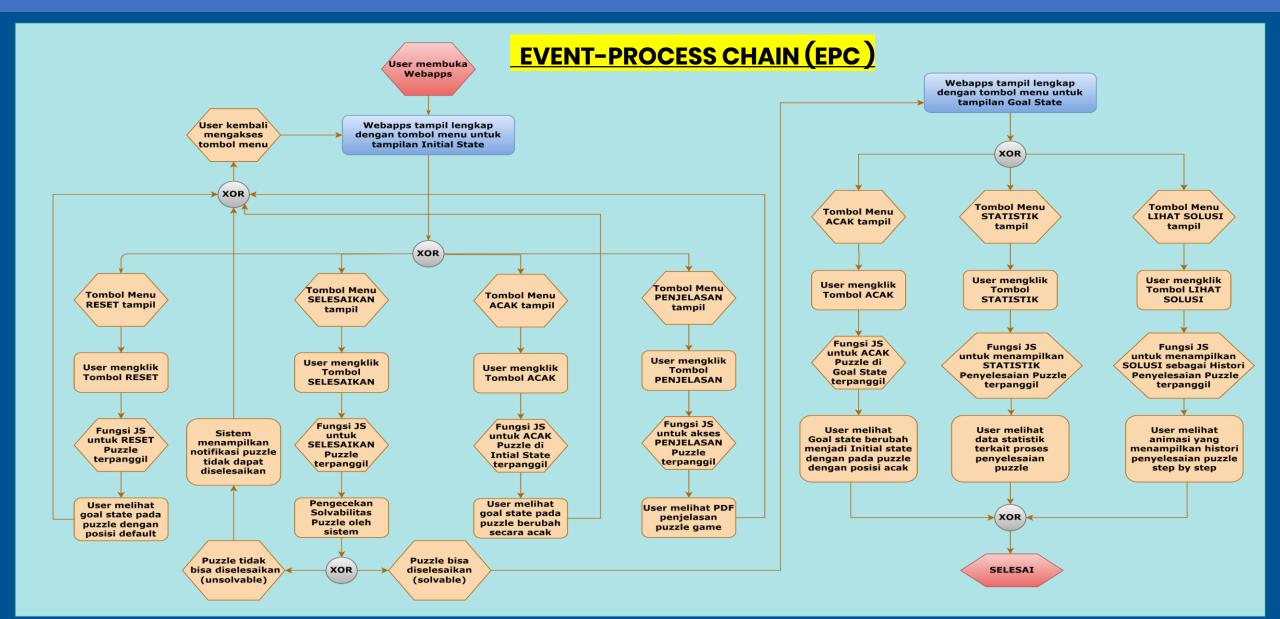


ACAK

STATISTIK

**LIHAT SOLUSI** 

# Proses Bisnis Program



#### Ketentuan Dasar Program

1. Default Intial State dan Goal State didefinisikan sebagai:

5	4 1	
	2	8
3	6	7

Initial State (Customized)

1	2	3
4	5	6
7	8	

Goal State (Fixed)

- 2. Tile yang kosong dapat dipindahkan ke berbagai tile terdekat (empty tile following)
- 3. Tile yang terdekat adalah tile yang secara horizontal atau vertikal berdekatan dengan tile kosong
- 4. Tile yang kosong tidak dapat bergerak diagonal

# Kemungkinan Algoritma Program

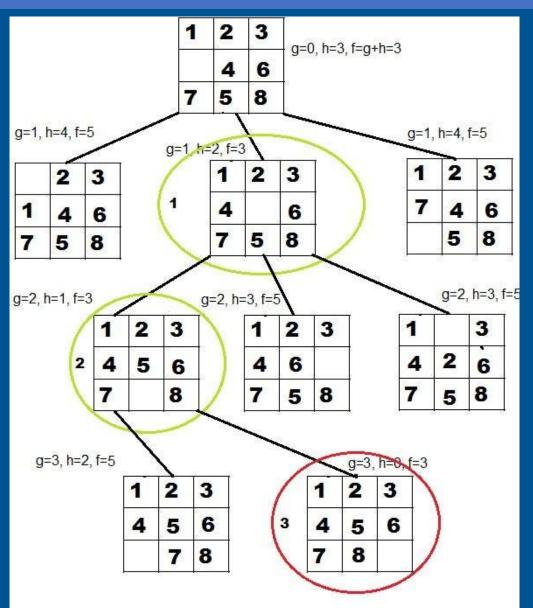
- 1. Breadth-First Search (BFS)→ First In First Out (FIFO)
- 2. Depth-First Search (DFS) → Last In First Out (LIFO)
- 3. A\* Algorithm  $\rightarrow$  f\*(n)=d\*(n)+p\*(n)

TABLE I. OPERATIONAL DATA TABLE OF THESE ALGORITHMS					
algorithm	Generated total node	The optimal path node number	Search target need layer	elapsed time (ms)	
Breadth-first Search	35	20	4	322	
Depth-first Search	258	143	4	84263	
A* algorithm	10	4	4	99	

**Best Algorithm** 

**Source:** Hua Shi, "Searching algorithms implementation and comparison of Eight-puzzle problem," *Proceedings of 2011 International Conference on Computer Science and Network Technology*, Harbin, China, 2011, pp. 1203-1206,

# Algoritma Terpilih: Algoritma A\*



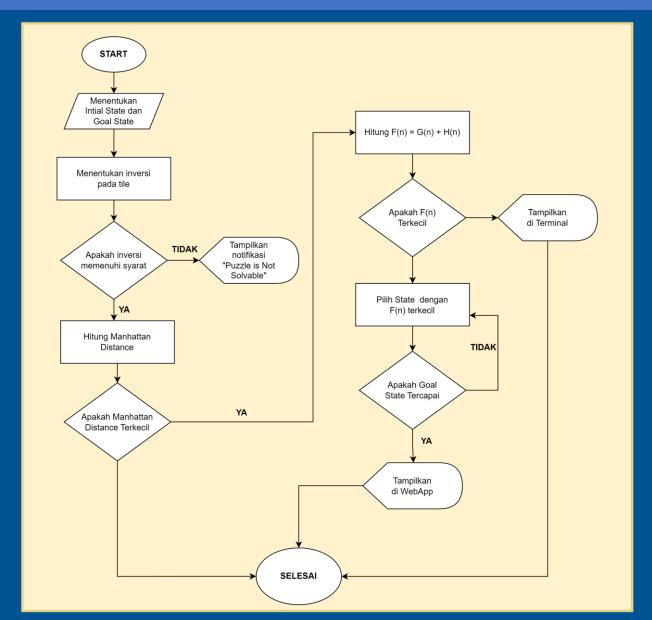
$$f*(n)=d*(n)+p*(n)$$

f\*(n) = Fungsi Heuristik Tile

g\*(n)= Jumlah *Tile* yang tidak sesuai posisinya dengan posisi tile di *Goal State* 

h\*(n)= Jumlah gerakan *Empty Tile* sejak terhitung sejak dari Initial State hingga ke *state*-n

# Algoritma Terpilih: Algoritma A\*



Algoritma A\* terpilih karena telah terbukti secara efektif dan efisien untuk menyelesaikan masalah 8-Puzzle Problem dengan kekuatan dari Fungsi Heuristik yang dibangun dalam algoritmanya.

Secara umum, langkah penyelesaian 8-Puzzle Problem dengan algoritma A\* dapat dilihat pada *flow chart* di samping

#### Jarak Tile: Manhatttan Distance

Manhattan distance adalah metrik jarak yang digunakan dalam geometri dan matematika untuk mengukur jarak antara dua titik dalam ruang berdimensi banyak. Dalam hal ini, jarak diukur dengan menjumlahkan perbedaan absolut antara setiap koordinat dari dua titik.

Dalam koordinat kartesian dua dimensi, Manhattan distance antara dua titik (x1, y1) dan (x2, y2) dapat dihitung dengan rumus berikut:

Manhattan distance = |x1 - x2| + |y1 - y2|

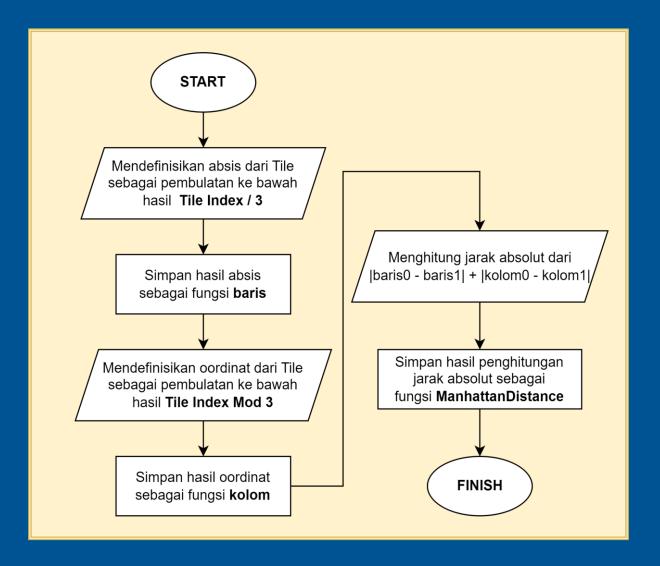
Istilah "Manhattan" digunakan karena pengukuran jarak ini menyerupai cara navigasi di Manhattan, New York, di mana jarak antara dua titik diukur dengan menjumlahkan perbedaan antara koordinat horizontal dan vertikal.

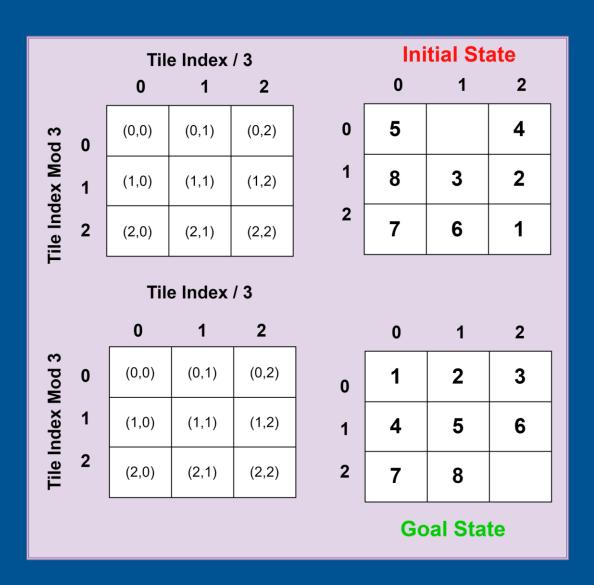
#### Kriteria Solusi Problem Puzzle

Metode Count Inversions digunakan untuk menentukan apakah sebuah initial state dari 8-puzzle game bersifat solvable atau unsolvable. Ketentuan dari Metode ini adalah sebagai berikut:

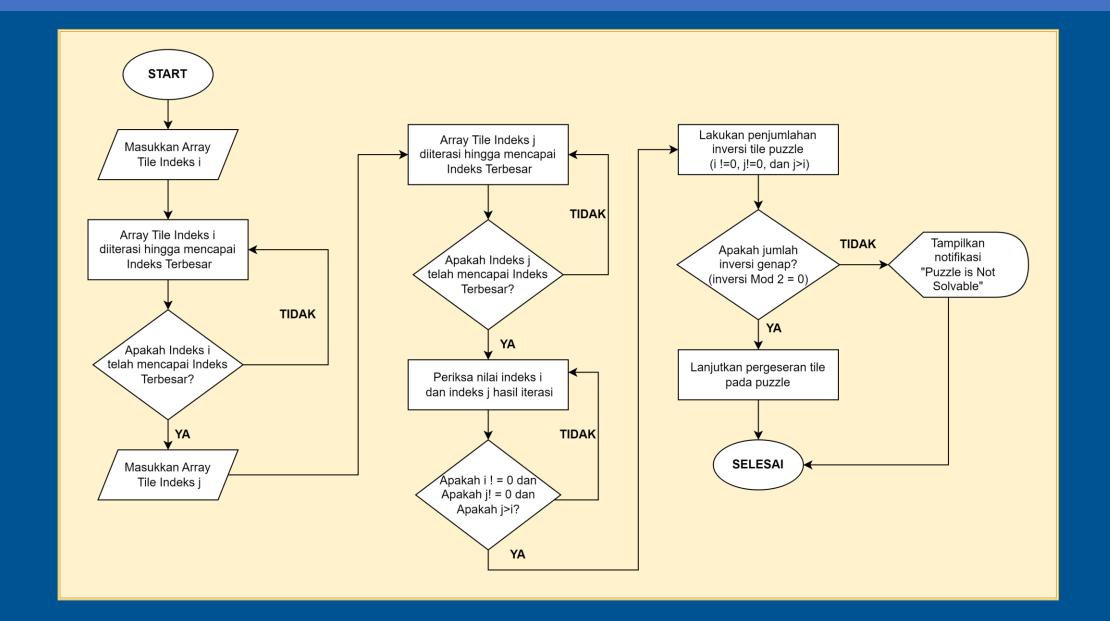
- 1. Metode ini melibatkan penghitungan jumlah pasangan angka yang tidak terurut (inversi) dalam initial state. Sebuah pasangan angka (i,j) dianggap sebagai inversi jika i berada di sebelah kiri j, tetapi i>j. Misalnya, dalam initial state [1, 3, 2, 4, 5, 6, 7, 8, 0], terdapat 1 inversi: (3,2).
- 2. Untuk menghitung jumlah inversi dalam *initial state,* **kita dapat melihat setiap pasangan angka dan memeriksa apakah ia merupakan inversi atau tidak**. Jumlah total inversi dalam *initial state* akan menjadi jumlah dari semua pasangan yang merupakan inversi.
- 3. Jika salah satu jumlah total baris yang mengandung inversi dalam *initial state* adalah bilangan genap, maka puzzle tersebut *solvable*, karena kondisi ini menunjukkan bahwa kita dapat menukar posisi pasangan angka untuk mencapai *goal state*. Sebaliknya, jika salah satu totalnya ganjil, maka puzzle *unsolvable*.

#### Kalkulasi Manhattan Distance





#### Inversi Deteksi Solvable Puzzle



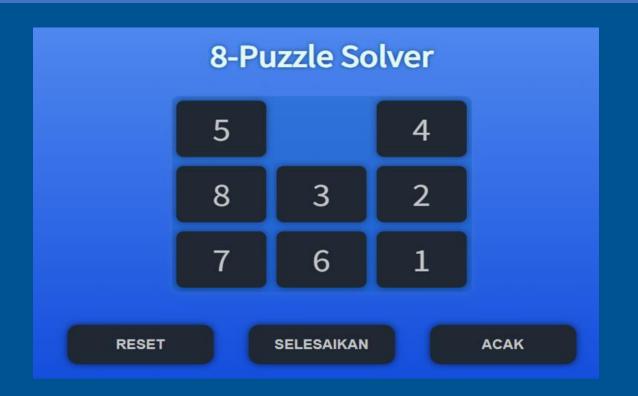
#### Unsolvable Puzzle



```
<div class="grid ease" id="grid">
    <!--Mendefinisikan Nilai dari Setiap Tile pada Puzzle-->
    <!--Mendefinisikan Default Initial State-->
        <div class="grid-item no-select ease" id="1">1</div>
        <div class="grid-item no-select ease" id="4">4</div>
        <div class="grid-item no-select ease" id="5">5</div>
        <div class="grid-item hidden" id="0">0</div>
        <div class="grid-item no-select ease" id="2">2</div>
        <div class="grid-item no-select ease" id="8">8</div>
        <div class="grid-item no-select ease" id="3">3</div>
        <div class="grid-item no-select ease" id="6">6</div>
11
        <div class="grid-item no-select ease" id="7">7</div>
    </div>
13
```

Susunan angka pada puzzle di atas dari kiri ke kanan untuk setiap barisnya posisinya terurut secara ascending sehingga tidak ditemukan inversi sebagaimana langkah awal untuk menentukan apakah susunan puzzle solvable atau unsolvable. Jadi, dapat disimpulkan bahwa susunan angka pada puzzle seperti ini adalah susunan yang unsolvable.

#### Solvable Puzzle



Susunan angka pada puzzle di atas terdapat dua jenis inversi, yakni [3,2] dan [6,1]. Jumlah nilai pada baris inversi [8,3] adalah 8+3+2=13 (ganjl). Kemudian, jumlah nilai untuk inversi [6,1] adalah 7+6+1=14 (genap). Karena ada salah satu jumlah nilai pada baris inversi bernilai genap, maka susunan angka pada puzzle di atas adalah susunan yang solvable

