Algoritma Penyelesaian IQ Puzzle Secara Brute Force

INSTITUT TEKNOLOGI BANDUNG

JL. GANESA 10, BANDUNG 40132

2024

1. Pendahuluan

(Sumber: https://www.smartgamesusa.com)

IQ Puzzler Pro adalah permainan papan yang diproduksi oleh perusahaan Smart Games. Tujuan dari permainan ini adalah pemain harus dapat mengisi seluruh papan dengan pece (blok puzzle) yang telah tersedia. Komponen penting dari permainan IQ Puzzler Pro terdiri dari:

A. Board (Papan) – Board merupakan komponen utama yang menjadi tujuan permainan dimana pemain harus mampu mengisi seluruh area papan menggunakan blok-blok yang telah disediakan.
B. Blok/Piece – Blok adalah komponen yang digunakan pemain untuk mengisi papan kosong hingga terisi penuh. Setiap blok memiliki bentuk yang unik dan semua blok harus digunakan untuk menyelesaikan puzzle

2. Strategi Algoritma

Algoritma yang digunakan adalah brute force, dimulai dengan membaca input yang ada dari text, setiap block dibaca sebagai list dari koordinat, koordinat yang digunakan dimodifikasi sesuai dengan indeks pada matrix untuk mempermudah penghitungan.

Solver yang digunakan dibuat dengan menggunakan fungsi rekursif untuk melakukan backtrack, setiap awal fungsi akan dicek jika board sudah solved, setiap solve dimulai dari (0, 0) dan terus bergeser ke kanan, jika sudah melebihi board akan digeser ke bawah paling kiri, jika sudah di level terbawah akan melakukan backtrack. saat sampai ke koordinat yang kosong, akan mengiterasi tiap shape dari shape yang tersedia (yang belum digunakan) dan mengiterasi lagi untuk setiap kemungkinan rotasi, dan masuk ke pengecekan apakah set itu bisa dimasukkan.

Proses pengecekan dilakukan dengan mengiterasi tiap koordinat dan mengecek relatif dengan titik central (titik yang sedang di iterasi oleh solver di awal), disini dilakukan pergeseran jika ada titik yang melewati batas board, dilakukan dengan memanipulasi nilai central point dengan shift. Jika set bisa dimasukkan akan lanjut ke solver selanjutnya, jika tidak bisa akan backtrack

3. Source code
   A. Solver

```
public class Solver {
    public PuzzleData puzzleData; // Store puzzle input
    public Solution solution; // Store solver output
    public Time timer;

    public Solver(File file) {
        this.puzzleData = readPuzzleFile(file);
        String msgres = new String();
        Time Timer = new Time();
        this.timer = Timer;
        this.solution = new Solution(msgres, puzzleData.Board);
    }
```

   B. Puzzle Data

```
public PuzzleData(int N, int M, int P, String S, List<Shape> arrayOfShapes) {
    this.Wide = M;
    this.Length = N;
    this.Puzzle = P;
    this.State = S;
    this.arrayOfShapes = arrayOfShapes;
    this.availShape = arrayOfShapes;
    this.Board = new char[M][N];
    this.Attempt = 0;
    for (int i = 0; i < M; i++) {
        for (int j = 0; j < N; j++) {
            Board[i][j] = ' ';
        }
    }
    char letter = 'A';
    for (Color color : colors) {
        colorMap.put(letter, color);
        letter++;
    }
}
```

   C. Solution

```java
public class Solution {
    public String result;
    public char[][] board;

    public Solution(String msgres, char[][] board) {
        this.result = msgres;
        this.board = board;
    }
}
```

D. Shape

```java
public Shape(int length, int wide, char symbol, List<int[]> dots) {
    this.length = length;
    this.wide = wide;
    this.symbol = symbol;
    this.dots = dots;
}
```

E. SetShape

```java
public SetShape(Shape shape) {
    this.availSet = new ArrayList<>();
    availSet.add(shape);
    availSet.add(shape.rotate());
    availSet.add(shape.rotate().rotate());
    availSet.add(shape.rotate().rotate().rotate());
    availSet.add(shape.flip());
    availSet.add(shape.flip().rotate());
    availSet.add(shape.flip().rotate().rotate());
    availSet.add(shape.flip().rotate().rotate().rotate());
}
```

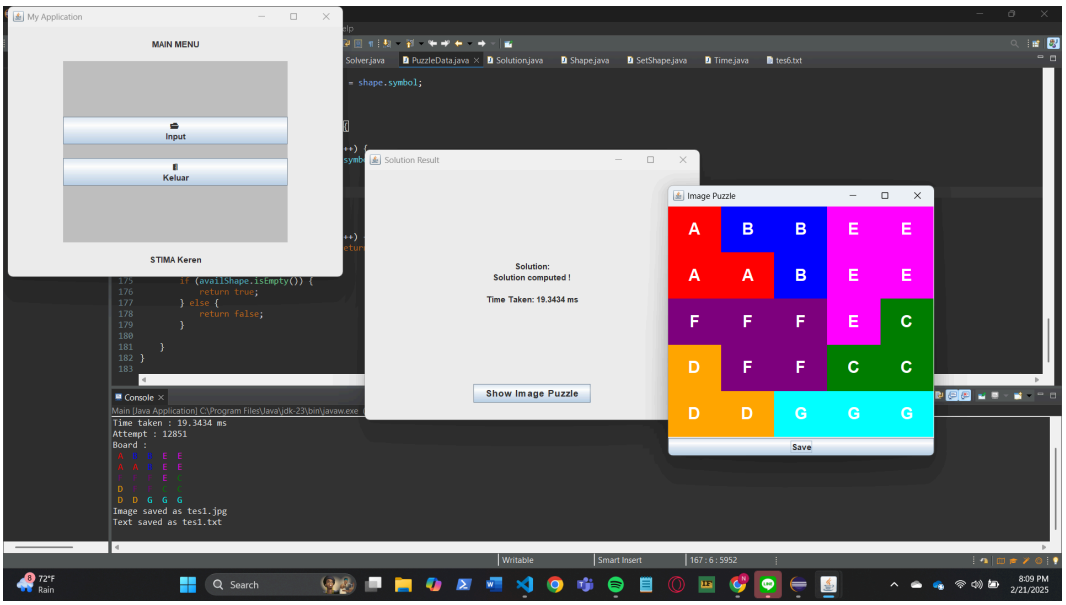
F. Time

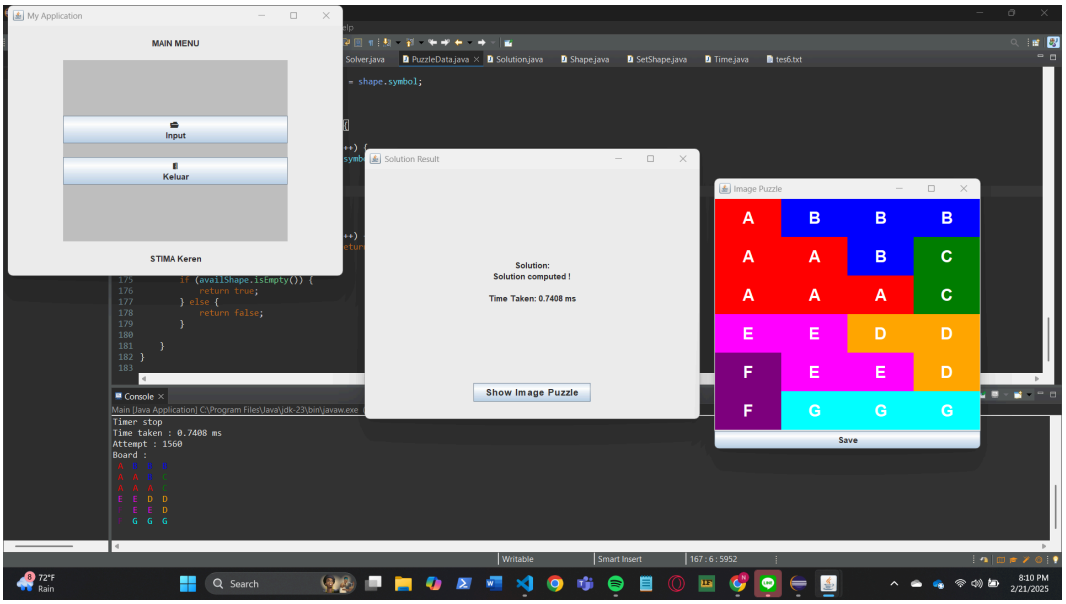
```java
public class Time {
    public boolean run;
    public long startTime;
    public long endTime;

    public void startTimer() {
        startTime = System.nanoTime();
        run = true;
        System.out.println(x:"Timer start");
    }

    public void stopTimer() {
        if (run) {
            endTime = System.nanoTime();
            run = false;
            System.out.println(x:"Timer stop");
        } else {
            System.out.println(x:"Timer is not running.");
        }
    }
```

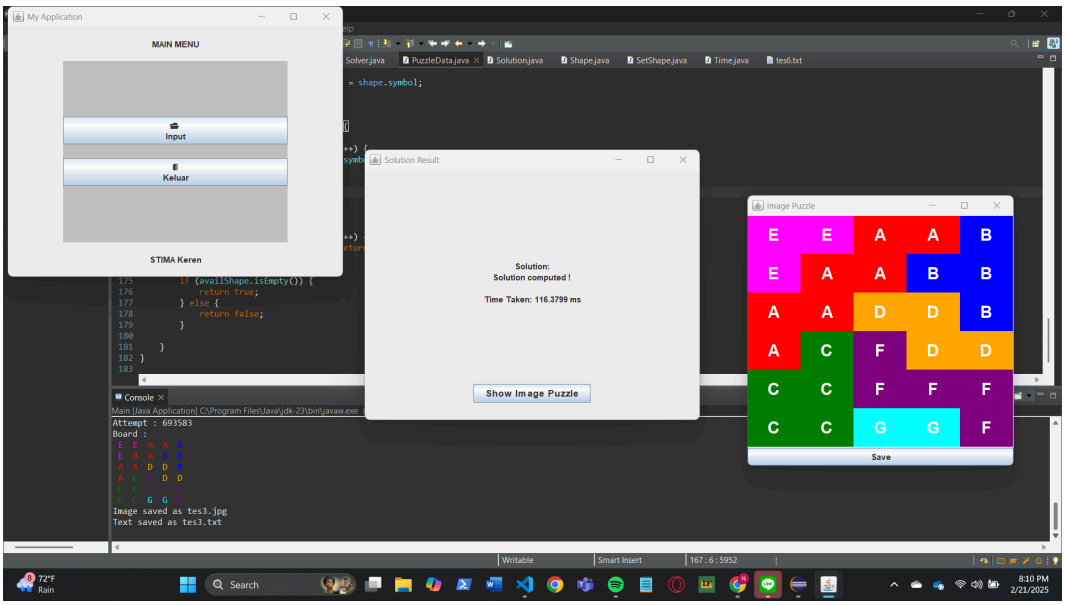
G. MainMenu

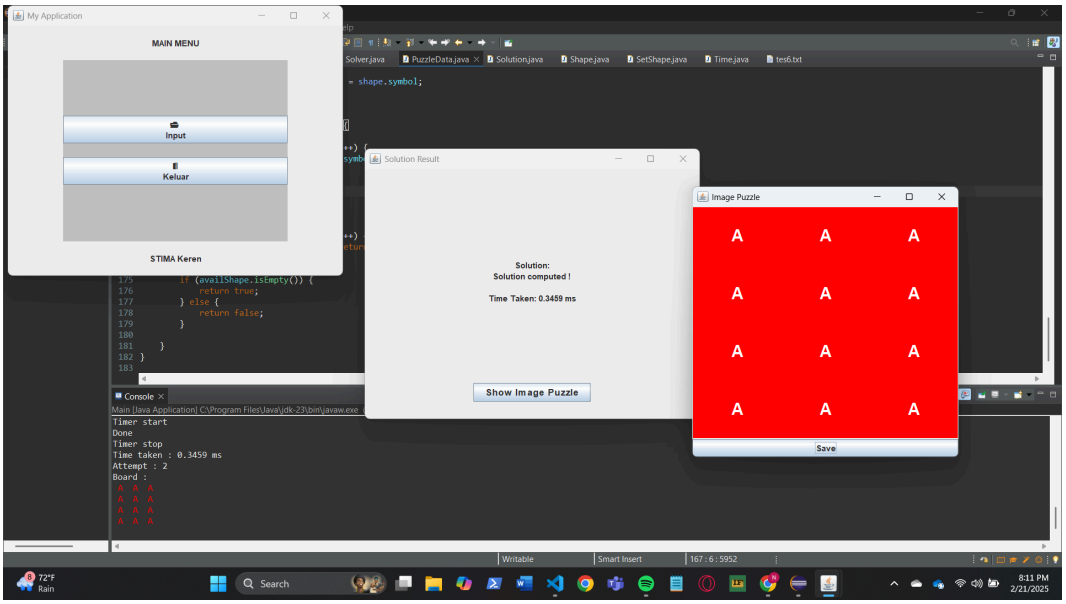
```java
public class MainMenu {
    public MainMenu() {
        MyFrame frame = new MyFrame();
        placeButton(frame.body, frame);
    }
```

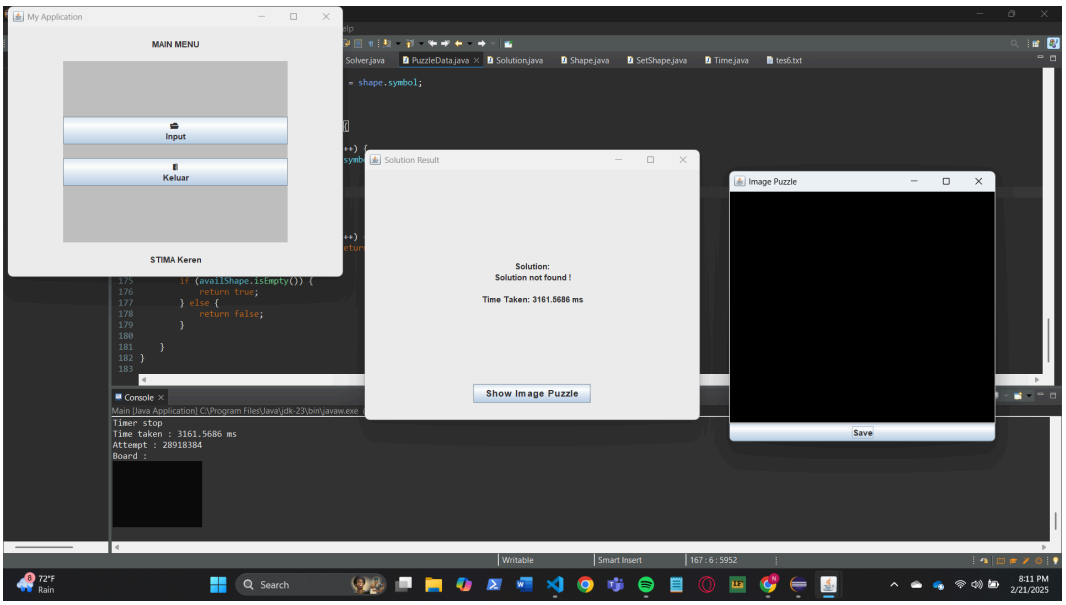
4. Hasil

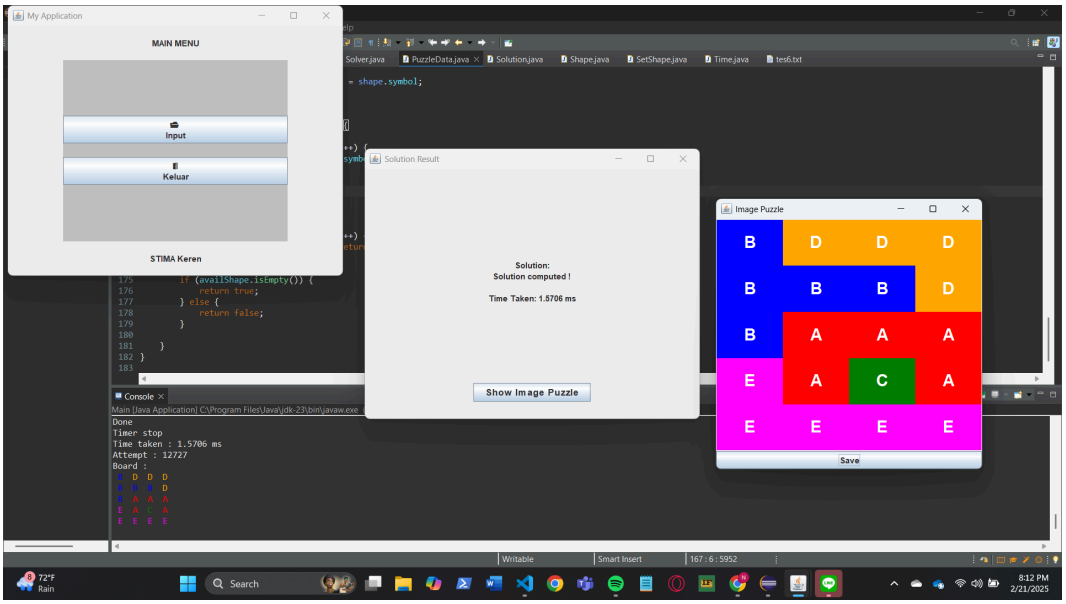

My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution computed !
Time Taken: 19.3434 ms
Show Image Puzzle
Image Puzzle
A B B E E
A A B E E
F F F E C
D F F C C
D D G G G
Save
Time taken : 19.3434 ms
Attempt : 12851
Image saved as tes1.jpg
Text saved as tes1.txt

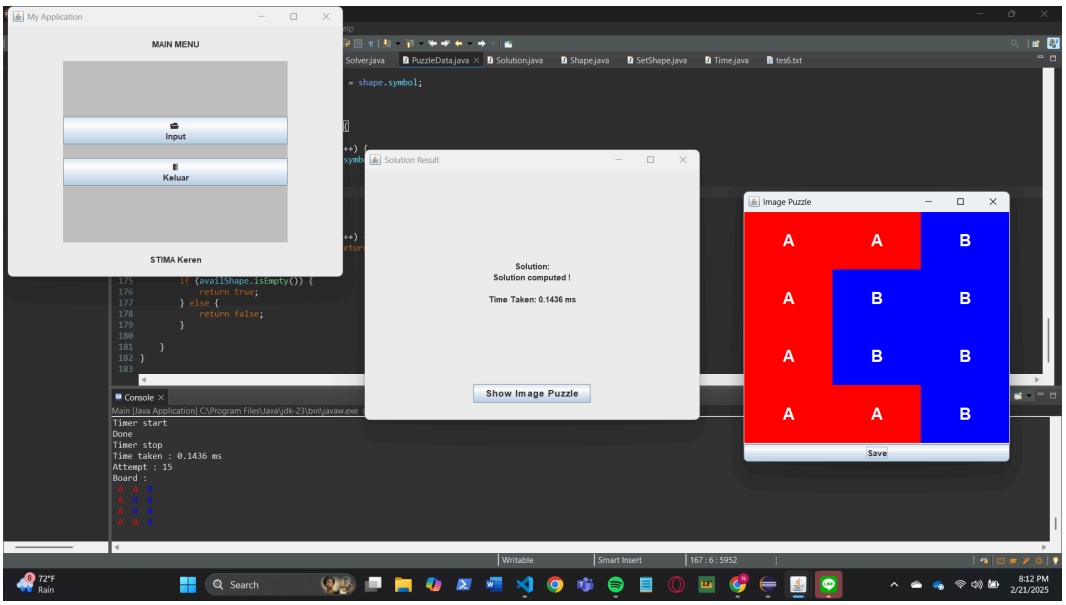

My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution computed !
Time Taken: 0.7408 ms
Show Image Puzzle
Image Puzzle
A B B B
A A B C
A A A C
E E D D
F E E D
F G G G
Save
Timer stop
Time taken : 0.7408 ms
Attempt : 1560


My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution computed !
Time Taken: 116.3799 ms
Show Image Puzzle
Image Puzzle
Save
Console
Attempt : 693583
Board :
Image saved as tes3.jpg
Text saved as tes3.txt


My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution computed !
Time Taken: 0.3459 ms
Show Image Puzzle
Image Puzzle
Save
Console
Timer start
Done
Timer stop
Time taken : 0.3459 ms
Attempt : 2
Board :


My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution not found !
Time Taken: 3161.5686 ms
Show Image Puzzle
Image Puzzle
Save
Console
Timer stop
Time taken : 3161.5686 ms
Attempt : 28918384
Board :


My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution computed !
Time Taken: 1.5706 ms
Show Image Puzzle
Image Puzzle
B D D D
B B B D
B A A A
E A C A
E E E E
Save
Console
Done
Timer stop
Time taken : 1.5706 ms
Attempt : 12727
Board :


My Application
MAIN MENU
Input
Keluar
STIMA Keren
Solution Result
Solution:
Solution computed !
Time Taken: 0.1436 ms
Show Image Puzzle
Image Puzzle
A A B
A B B
A B B
A A B
Save
Solver.java
PuzzleData.java
Solution.java
Shape.java
SetShape.java
Time.java
tes6.txt
= shape.symbol;
Console
Timer start
Done
Timer stop
Time taken : 0.1436 ms
Attempt : 15
Board :
A A B
A B B
A B B
A A B
Writable
Smart Insert
167 : 6 : 5952
72°F
Rain
Search
8:12 PM
2/21/2025

LAMPIRAN

Github : https://github.com/Narrr21/Tucil1_13523076

No.	Poin	Ya	Tidak
1.	Program berhasil dikompilasi tanpa kesalahan	X	
2.	Program berhasil dijalankan	X	
3.	Solusi yang diberikan program benar dan mematuhi aturan permainan	X	
4.	Program dapat membaca masukan berkas .txt serta menyimpan solusi dalam berkas .txt	X	
5.	Program memiliki Graphical User Interface (GUI)	X	
6.	Program dapat menyimpan solusi dalam bentuk file gambar	X	
7.	Program dapat menyelesaikan kasus konfigurasi custom		X
8.	Program dapat menyelesaikan kasus konfigurasi Piramida (3D)		X
9.	Program dibuat oleh saya sendiri	X	