

**TUGAS KECIL 1**  
**STRATEGI ALGORITMA - IF2211**  
**IQ PUZZLER PRO SOLVER MENGGUNAKAN**  
**BRUTE FORCE ALGORITHM**



**Dibuat Oleh:**

David Bakti Lodianto - 13523083

## **DAFTAR ISI**

<b>DAFTAR ISI</b>	<b>2</b>
<b>BAB 1: Pendahuluan</b>	<b>3</b>
<b>BAB 2: Algoritma Penyelesaian</b>	<b>4</b>
<b>BAB 3: Implementasi</b>	<b>5</b>
<b>BAB 4: Testing</b>	<b>13</b>
<b>LAMPIRAN</b>	<b>21</b>

## BAB 1: Pendahuluan

IQ Puzzler Pro adalah suatu permainan puzzle dimana terdapat papan berbentuk persegi panjang dan potongan-potongan puzzle yang dapat memenuhi papan tersebut dengan tepat.



Permainan IQ Puzzler Pro  
(Sumber: <https://www.smartgamesusa.com>)

Komponen penting dari permainan IQ Puzzler Pro terdiri dari:

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

## **BAB 2: Algoritma Penyelesaian**

Pendekatan penyelesaian puzzle ini diambil dari fakta bahwa jika puzzle dapat diselesaikan, maka sepotong puzzle pasti akan selalu memiliki tempat, dan lainnya dapat menyesuaikan. Jika sepotong puzzle dari awal sudah tidak memiliki tempat, dapat disimpulkan bahwa puzzle tidak dapat diselesaikan.

Algoritma penyelesaian:

1. Ambil satu potongan puzzle paling atas yang masih belum terpasang.
2. Cocokkan pojok kiri atas potongan dengan pojok kiri atas papan.
3. Coba apakah potongan tersebut muat atau tidak.
  - a. Jika iya, letakkan potongan tersebut, dan ulangi langkah 1.
  - b. Jika tidak, geser ke kanan 1 petak, dan jika potongan berlebih ke kanan, pindah ke pojok kiri dan geser ke bawah 1 petak. Ulangi langkah 3.
    - i. Jika potongan sudah mencapai paling ujung kanan bawah dari papan, maka putar potongan 90 derajat, dan ulangi langkah 3.
    - ii. Jika potongan sudah mencapai paling ujung kanan bawah dari papan dan sudah diputar sebanyak 3 kali, maka putar sekali lagi dan cerminkan terhadap sumbu Y, dan ulangi langkah 3.
    - iii. Jika potongan sudah mencapai paling ujung kanan bawah dari papan dan sudah dicerminkan, maka putar lagi potongan sebanyak 90 derajat, dan ulangi langkah 3.
    - iv. Jika potongan sudah mencapai paling ujung kanan bawah dari papan dan sudah dicerminkan dan diputar lagi sebanyak 3 kali, maka lakukan langkah 3b pada potongan sebelumnya.
    - v. Jika potongan pertama sudah mencapai paling ujung kanan bawah dari papan, sudah dicerminkan dan diputar, maka permainan puzzle tidak dapat diselesaikan.

## BAB 3: Implementasi

Pemrograman aplikasi solver ini menggunakan bahasa Java, sehingga menggunakan paradigma *Object-Oriented Programming*, dibagi menjadi 6 kelas.

### 1. Kelas Piece

- **Attribute**

Nama	Tipe	Deskripsi
matrix	public boolean [][]	Array boolean 2D yang merepresentasikan piece.
num	public int	Angka yang merepresentasikan piece ke-berapa.
rotation	public int	Status rotasi dari piece
mirror	public boolean	Status apakah piece tercerminkan atau tidak.

- **Method**

Nama	Tipe	Parameter	Deskripsi
Piece	Public Constructor	ArrayList<String> pieceString	Membuat piece dari list string dari input teks.
Piece	Public Constructor	boolean[][] mat, int n, int r, boolean m	Membuat piece dari atribut-atribut pada parameter.
getRow	Public int		Mengambil jumlah baris dari matriks
getCol	Public int		Mengambil jumlah kolom dari matriks
lineCheck	Public static int	String s	Mengembalikan angka urutan piece ke berapa berdasarkan abjad pada piece.

rotate	Public boolean		Merrotasi piece sebanyak 90 derajat searah jarum jam.
mirror	Public boolean	Piece piece, int r, int c	Mencerminkan piece terhadap sumbu Y
reset	Public void	Piece piece, int r, int c	Mengembalikan rotasi dan pencerminan piece pada piece yang sudah terrotasi 270 derajat dan dicerminkan
printPiece	Private void		Mencetak piece.

## 2. Kelas Board

- **Attribute**

Nama	Tipe	Deskripsi
matrix	public int [][]	Array Integer 2D yang menyimpan nilai piece yang menempati posisinya.

- **Method**

Nama	Tipe	Parameter	Deskripsi
Board	Public Constructor	int row, int col	Membuat board kosong dengan dimensi row x col.
getRow	Public int		Mengambil jumlah baris dari matriks
getCol	Public int		Mengambil jumlah kolom dari matriks
isSolved	Public boolean		Mengecek apakah board terpenuhi.

isFit	Public boolean	Piece piece, int r, int c	Mengecek apakah piece dapat muat di posisi r, c pada board. (pojok kiri atas piece ditempatkan di (r, c))
addPiece	Public void	Piece piece, int r, int c	Memasukkan piece ke board, dengan pojok kiri atas piece ditempatkan di (r, c).
removePieceHelper	Private void	int r, int c	Membantu pelepasan piece
removePiece	Public void	Piece piece	Melepas piece dari board
printBoard	Public void		Mencetak isi board

### 3. Kelas Game

- **Attribute**

Nama	Tipe	Deskripsi
runtime	public long	Waktu pencarian solusi (dalam milisekon).
iterations	public long	Banyaknya iterasi kasus penyelesaian sebelum solusi ditemukan.
solved	public boolean	Status penanda solusi sudah ditemukan.
board	public Board	Board dari permainan
pieces	public Piece[]	Array pieces dari permainan
invalid	public boolean	Penanda config invalid
deadend	private boolean	Penanda piece sudah tidak dapat dilanjutkan (sudah dicerminkan dan terrotasi

		3 kali dan di pojok kanan bawah board).
--	--	-----------------------------------------

- **Method**

Nama	Tipe	Parameter	Deskripsi
Game	Public Constructor	Piece[] p, Board b	Membuat game dengan atribut Piece-piece p dan Board b.
Game	Public Constructor		Membuat game invalid.
solve	Public void		Mencari solusi, dan merekam banyak iterations dan runtime program.
solveRecurse	Private void	int idx, int r, int c	Method pembantu solve, untuk piece ke-idx dan posisi r, c secara rekursif.

#### 4. Kelas GUI

- **Attribute**

Tidak ada atribut untuk kelas ini.

- **Method**

Nama	Tipe	Parameter	Deskripsi
show	Public static void		Meluncurkan aplikasi
start	Public void	Stage primaryStage	Membuat stage dan scene dari aplikasi utama

#### 5. Kelas IO

- **Atribut**

Nama	Tipe	Deskripsi
------	------	-----------



escCodes	private final String[]	Array string berisikan ANSI escape code untuk mencetak teks berwarna pada terminal
resetCode	private final String	ANSI reset code untuk mengembalikan formatting print teks terminal.
hexCodes	private final int[]	Array integer berisikan nilai warna hex untuk mewarnai gambar pada board.

- **Method**

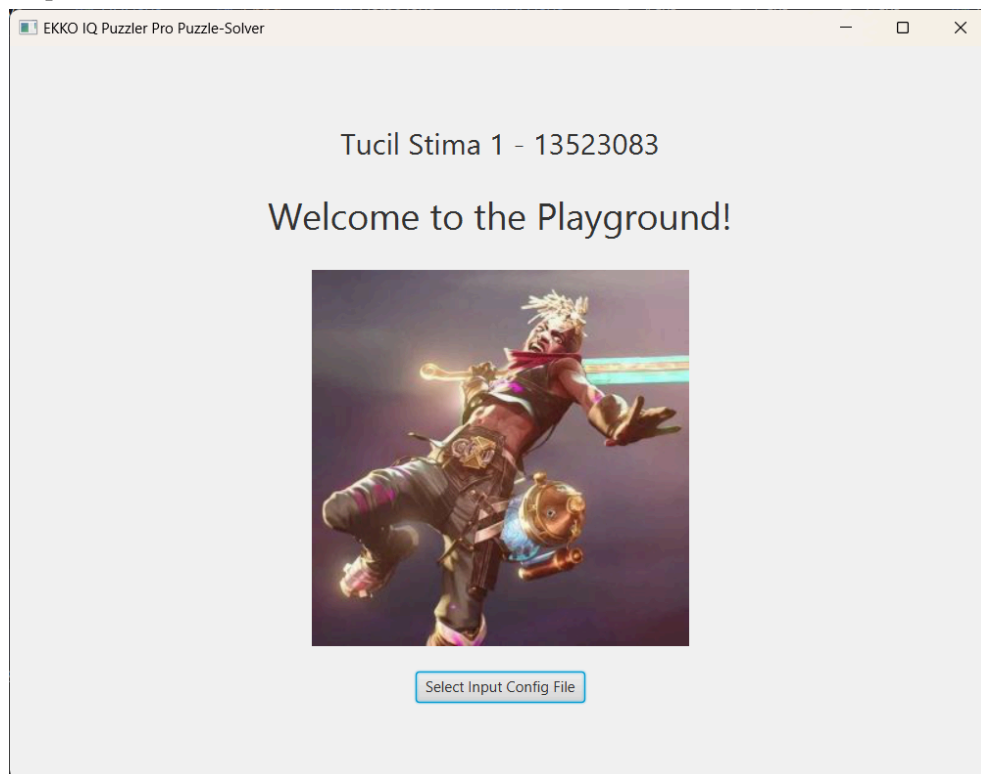
Nama	Tipe	Parameter	Deskripsi
readConfigFile	Public static Game	File file	Membaca file input .txt dan mengembalikan objek game
printColor	Public void	char a, int colorCode	Mencetak karakter a pada terminal dengan index warna colorCode
saveText	Public void	Board board, String fileName	Menuliskan hasil penyelesaian board pada file dengan nama fileName.
generateImage	Public BufferedImage	Board board	Menghasilkan gambar dari board, menggunakan warna hexCodes.
imageSave	Public void	BufferedImage img, String fileName	Menyimpan gambar img ke file dengan nama fileName

## 6. Kelas Main

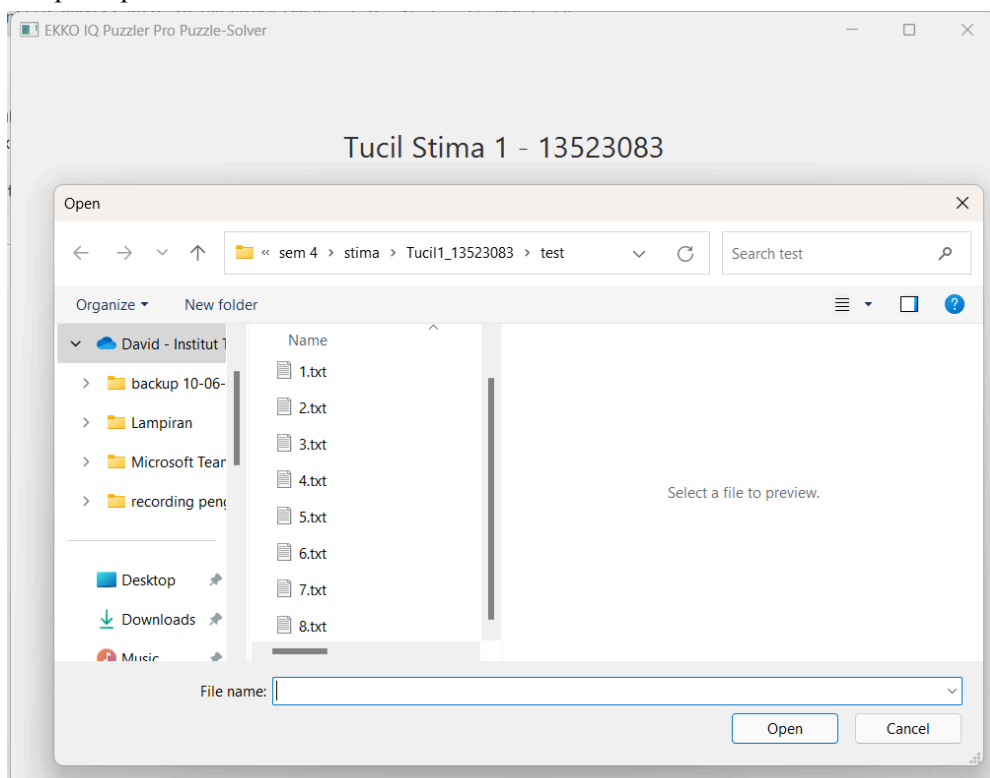
Kelas ini memiliki 1 method – main, untuk dijalankan sebagai kelas utama.

## User Interface

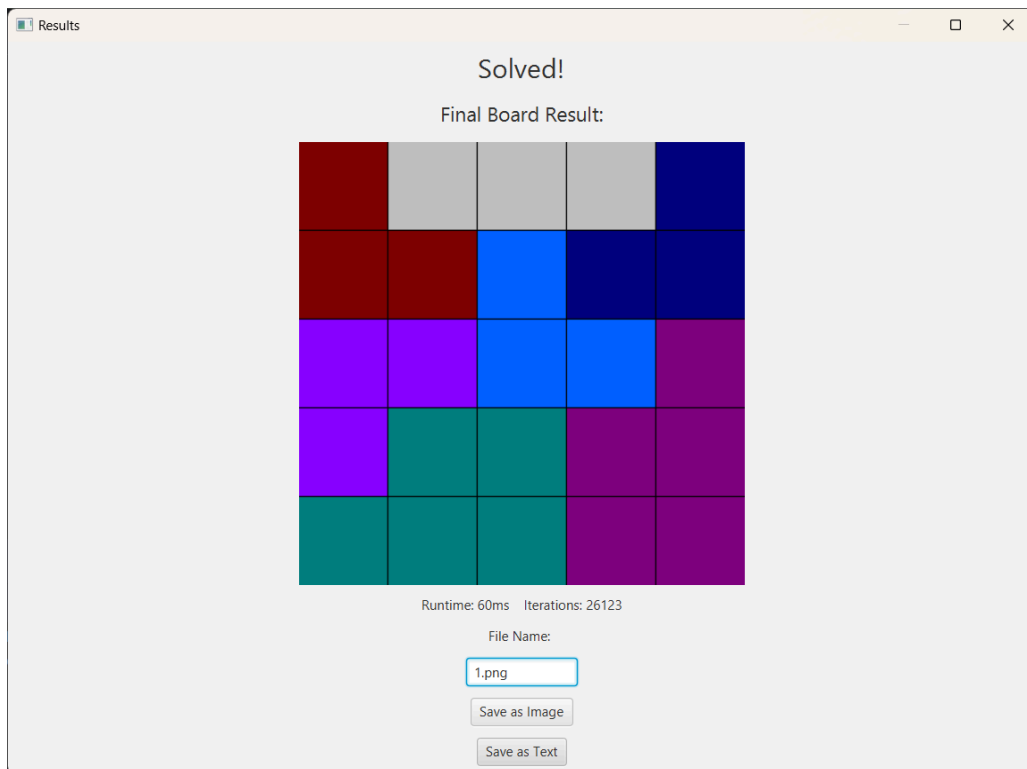
### 1. Tampilan Utama



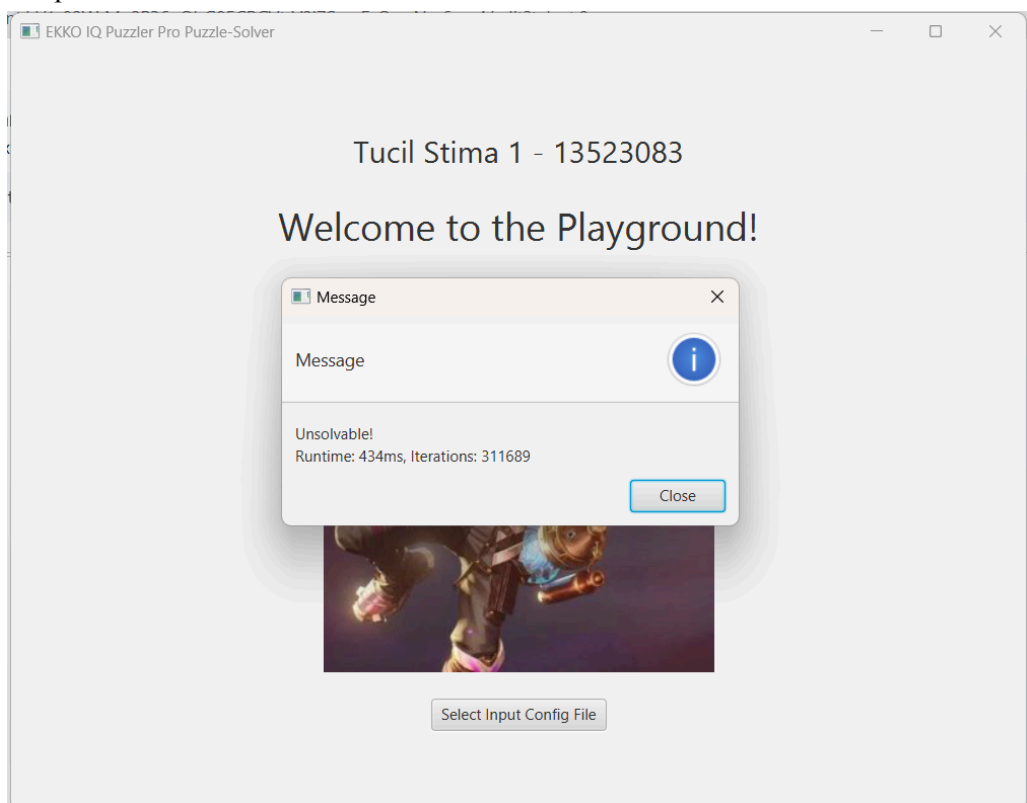
### 2. Tampilan pemilihan file



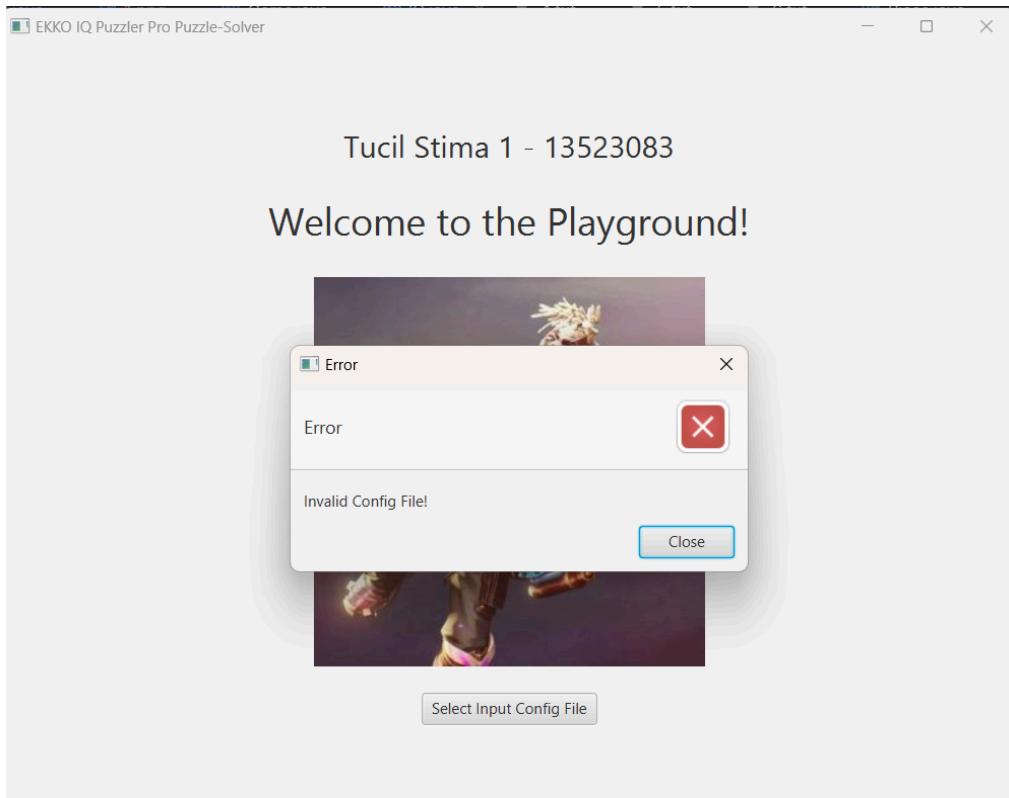
### 3. Tampilan solusi ditemukan



#### 4. Tampilan solusi tidak ditemukan



#### 5. Tampilan file konfigurasi .txt invalid

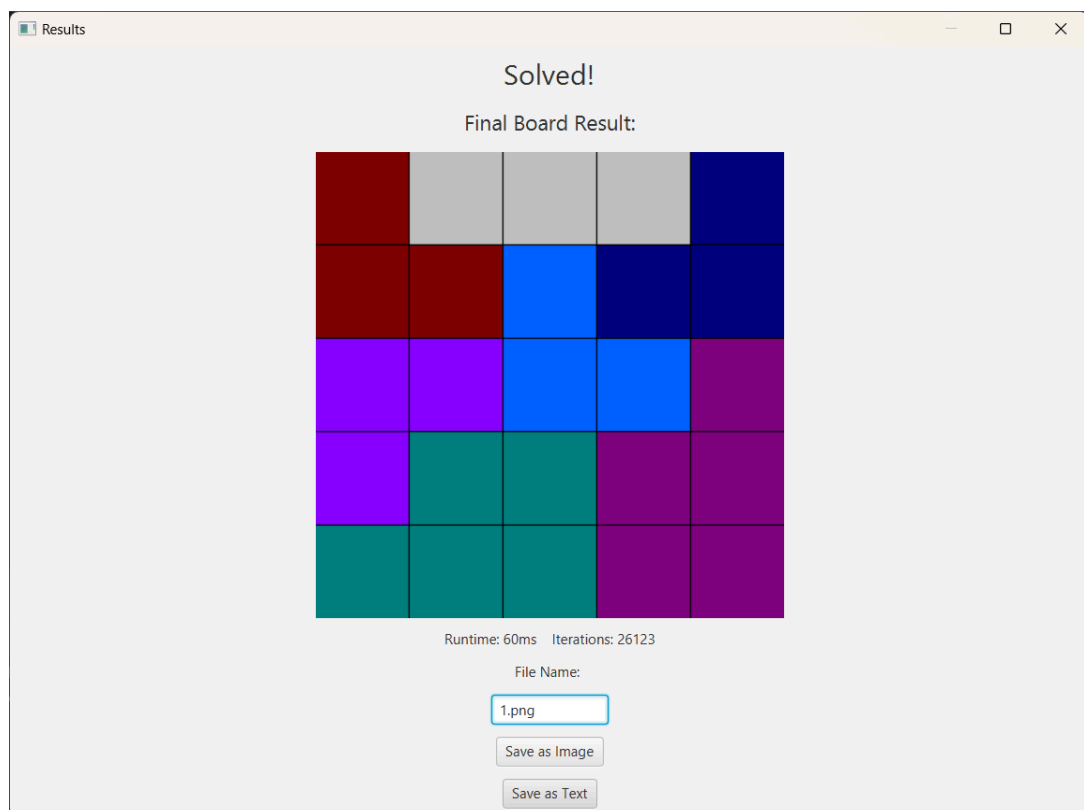


## BAB 4: Testing

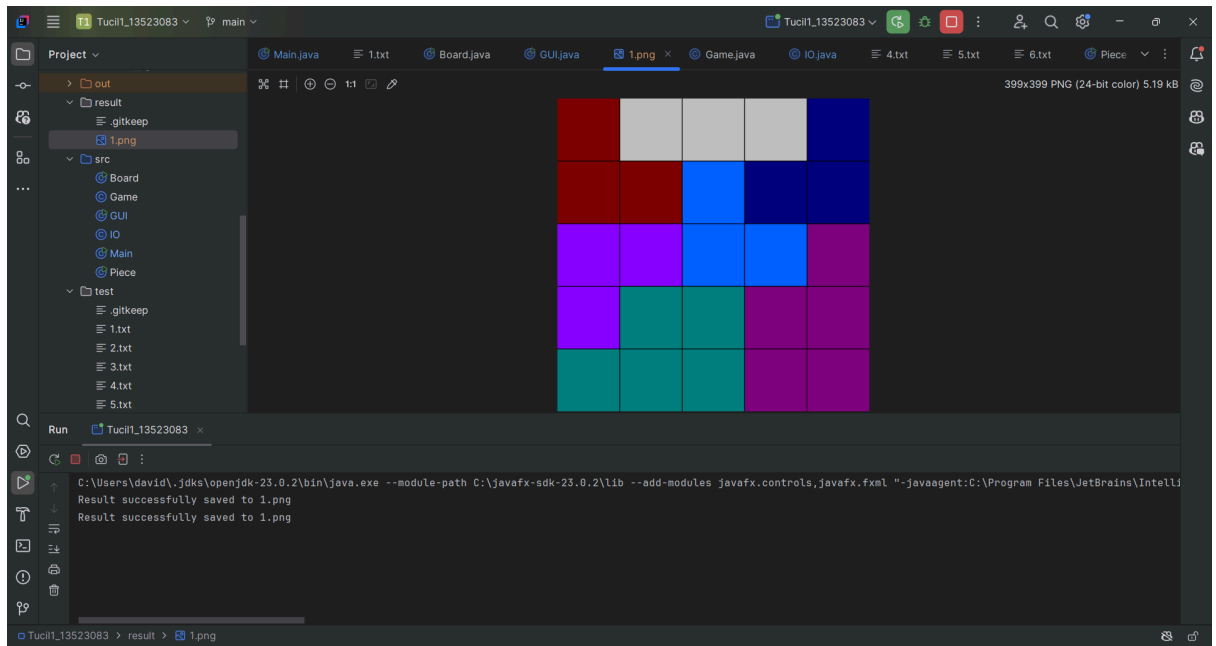
### 1. File 1.txt

```
5 5 7
DEFAULT
A
AA
B
BB
CC
C
D
DD
EE
EEE
FF
FFF
GGG
```

Hasil:



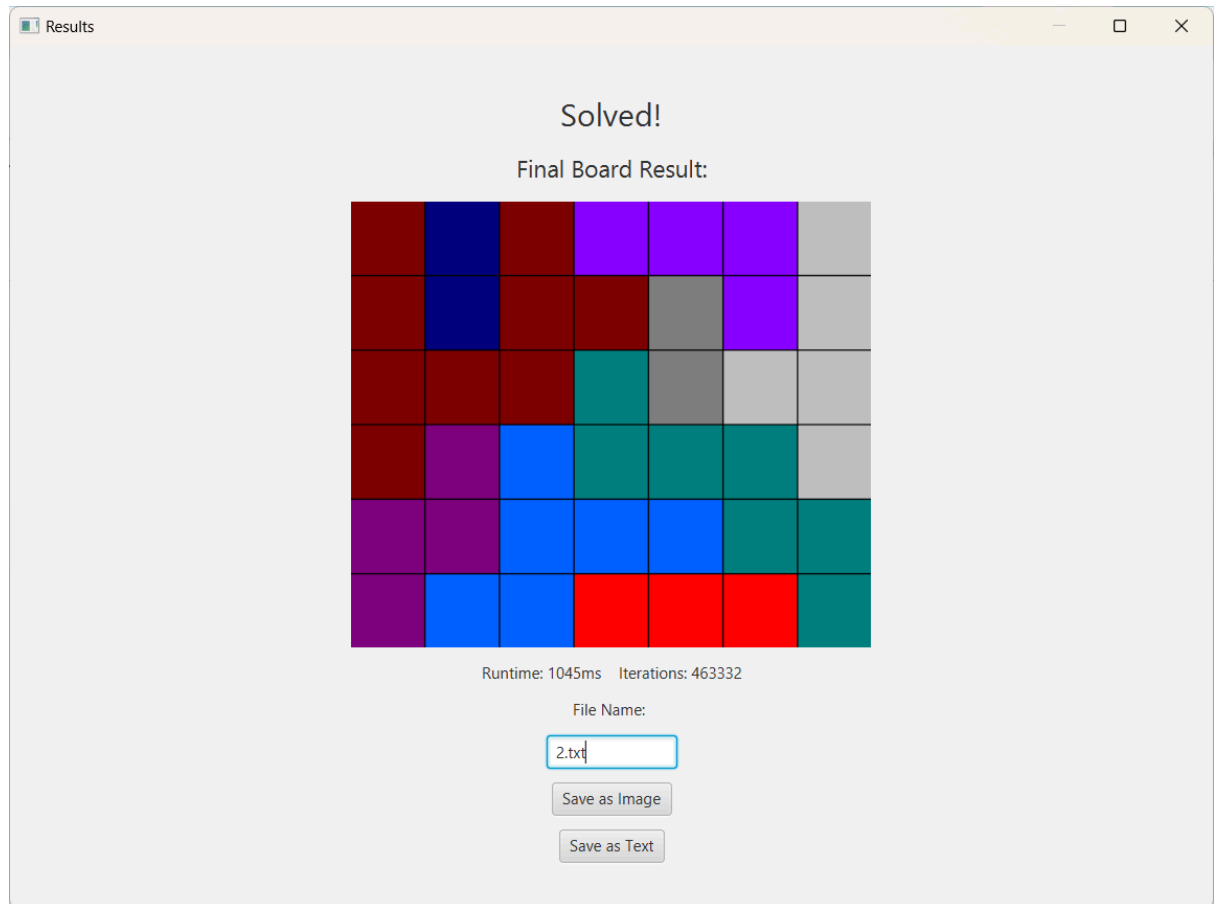
Hasil Penyimpanan Gambar:



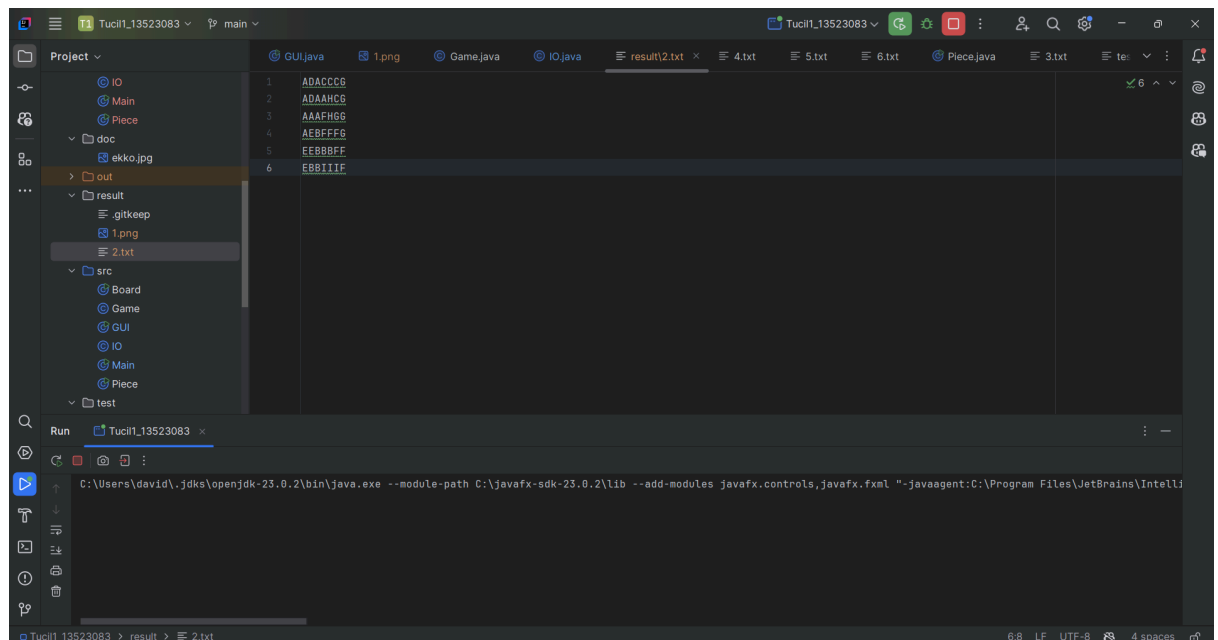
2. File 2.txt

```
6 7 9
DEFAULT
A A
A AA
AAA
A
B
BBB
BB
CC
C
C
DD
EE
EE
FF
FF
F
FF
G
GGGG
HH
III
```

Hasil:



Hasil penyimpanan teks:

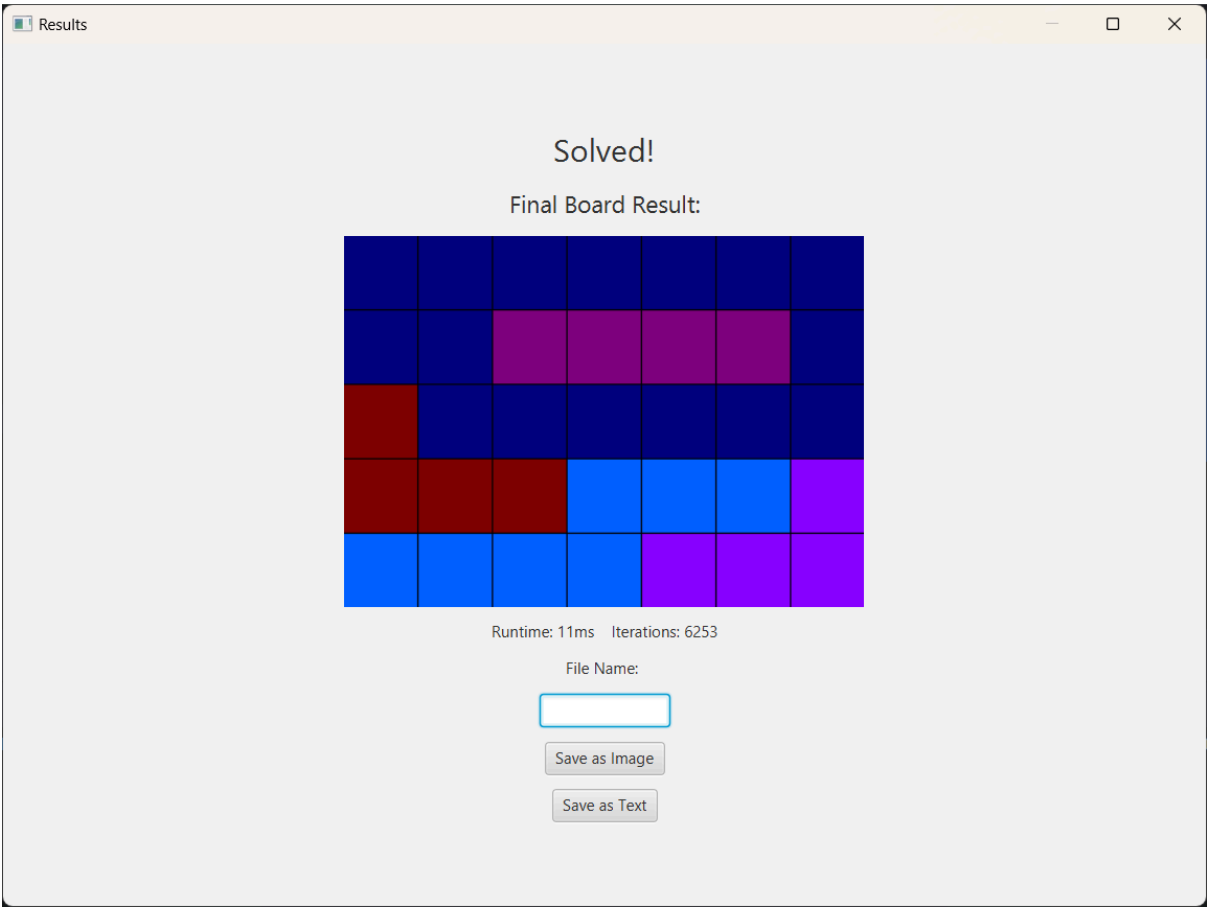


3. File 3.txt

```
5 7 5
DEFAULT
A
```

```
AAA
BBB
 BBB
C
CCC
DDDDDD
D   DD
DDDDDD
EEEE
```

Hasil:

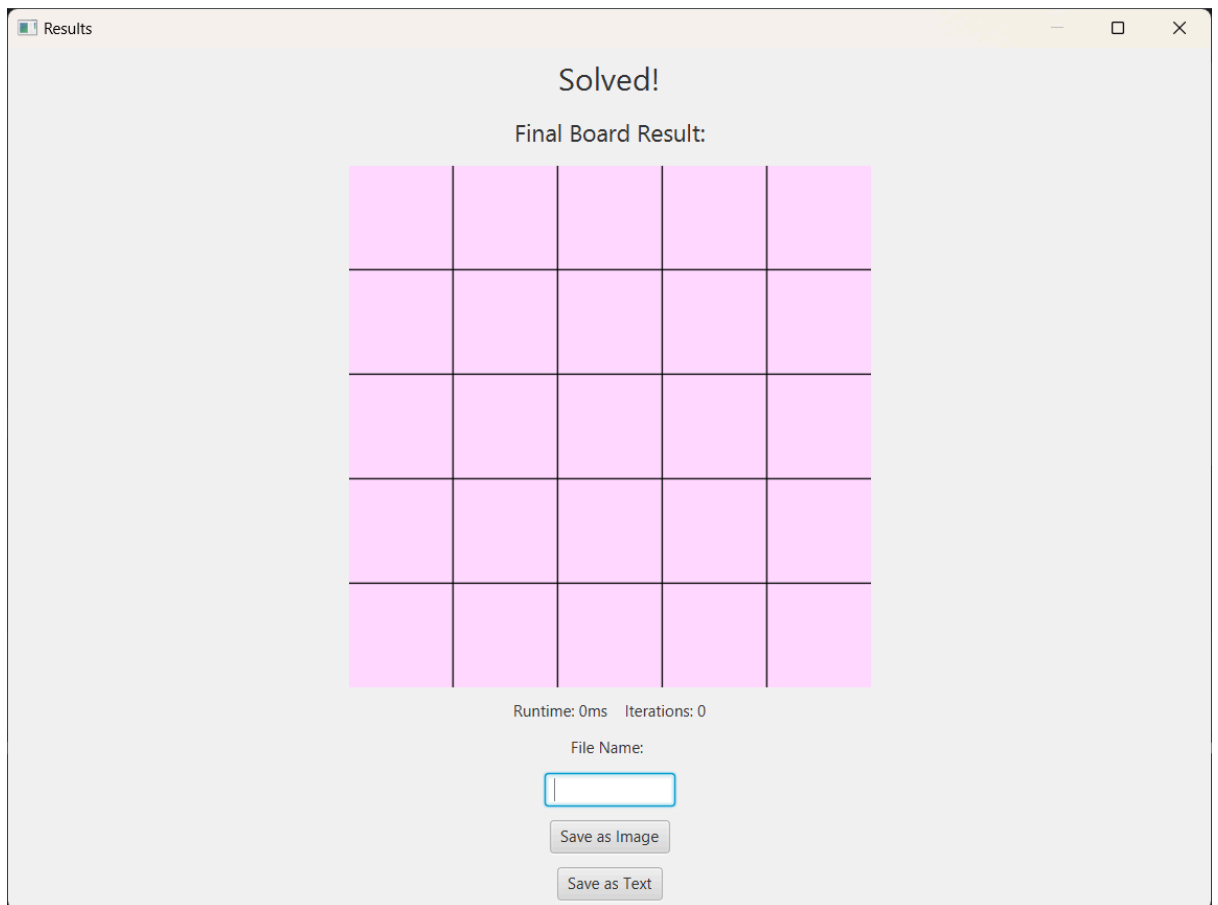


4. File 4.txt

```
5 5 1
DEFAULT
XXXXX
XXXXX
XXXXX
XXXXX
XXXXX
```

Hasil:



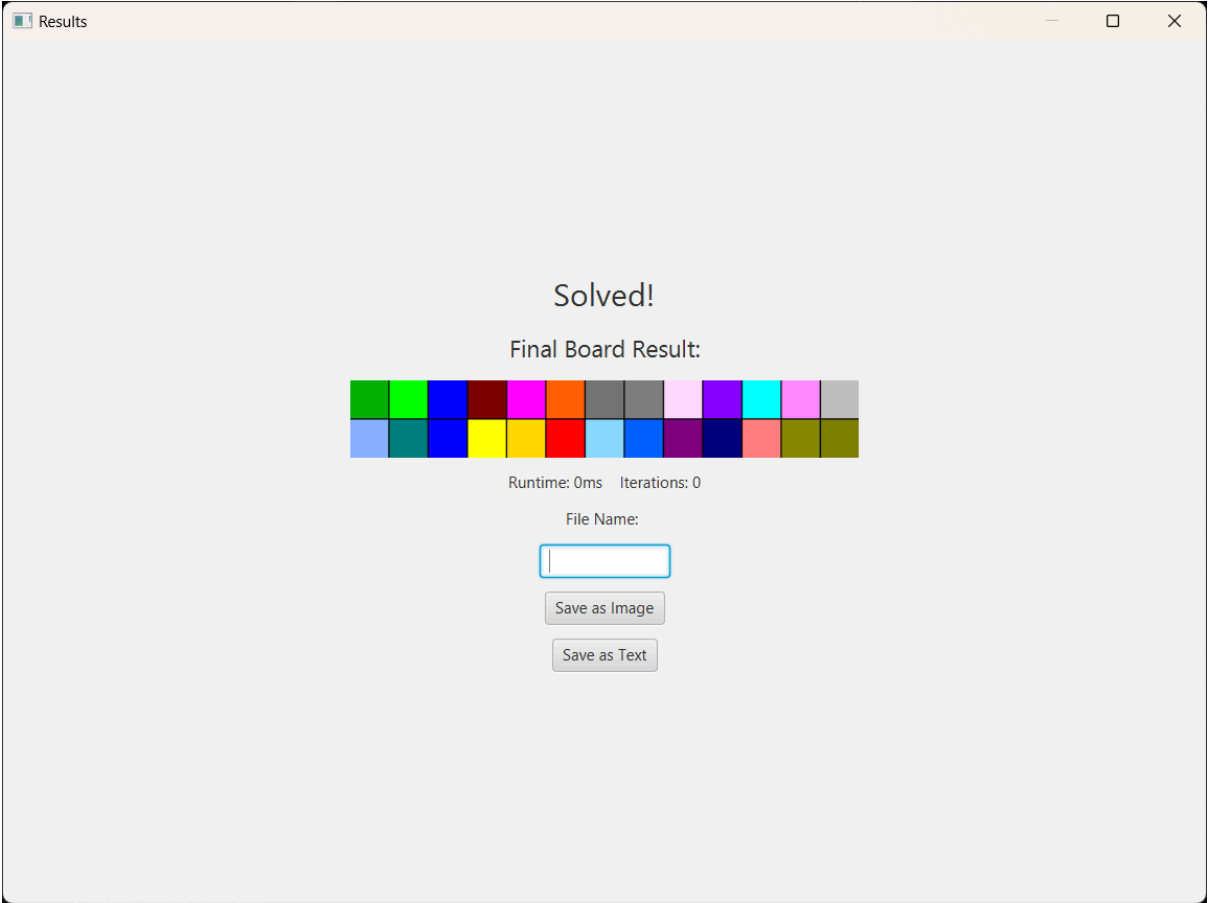


5. File 5.txt

```
2 13 26
DEFAULT
W
J
O
A
M
T
Z
H
X
C
N
Q
G
Y
F
L
K
S
I
U
```

B  
E  
D  
P  
V  
R

Hasil:

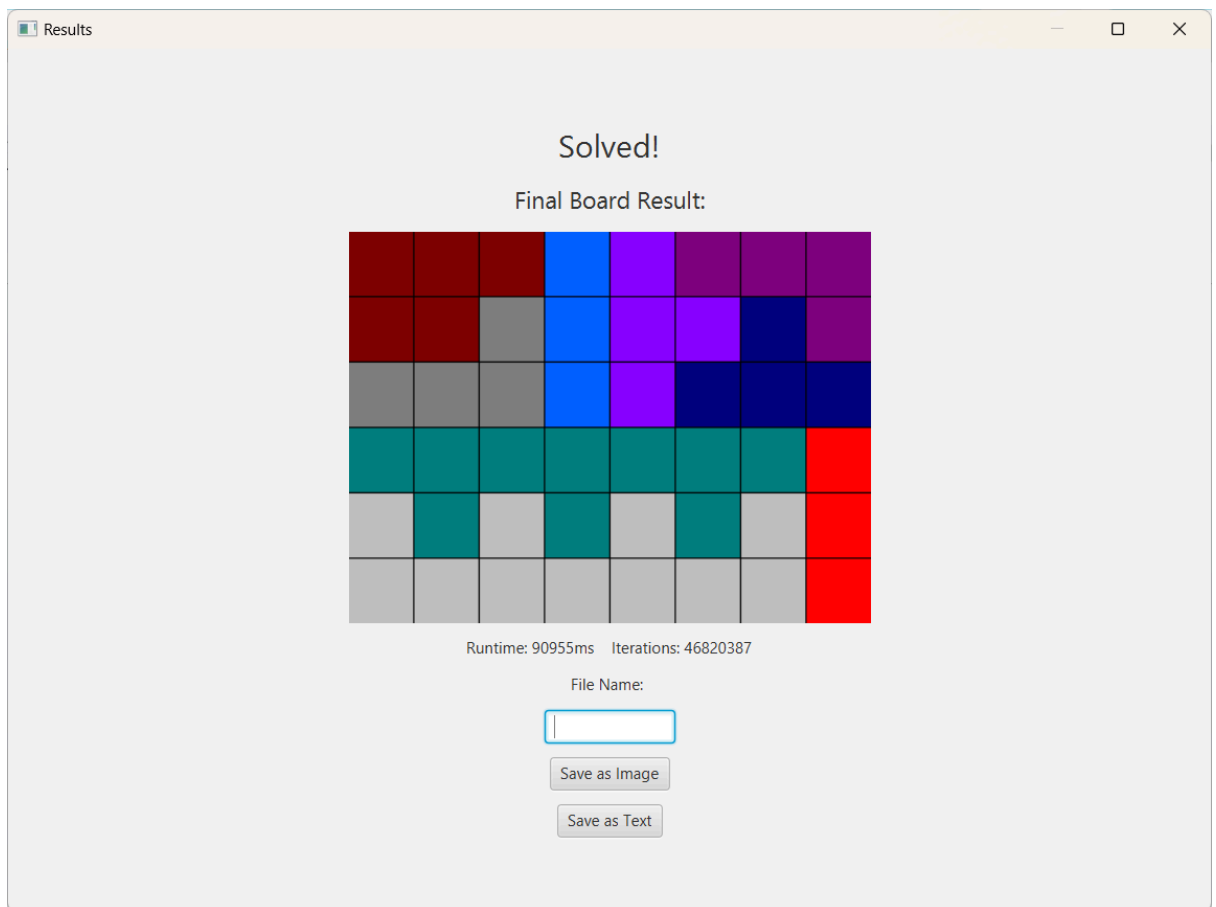


6. File 6.txt

6 8 9  
DEFAULT  
AAA  
AA  
BBB  
C  
CC  
C  
D  
DDD  
E  
EEE  
F  
FF

```
F
FF
F
FF
F
G G G G
GGGGGGG
HHH
H
III
```

Hasil:

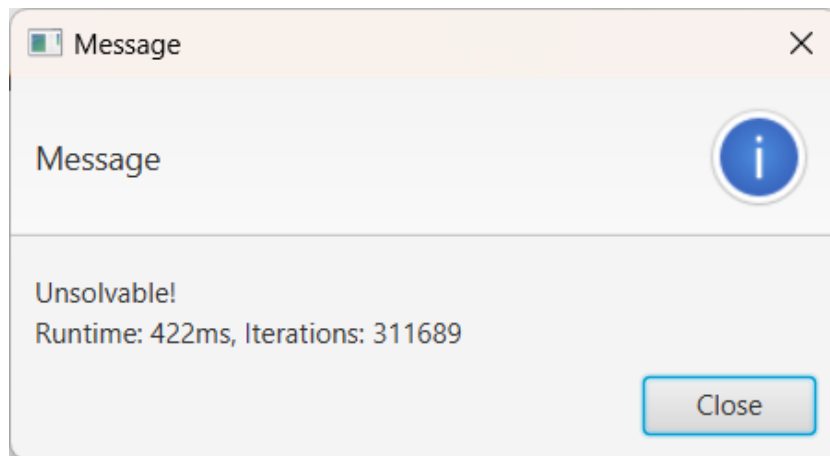


#### 7. File 7.txt

```
4 4 5
DEFAULT
A
AA
B
BB
CC
C
D
DD
```

```
EE
E
```

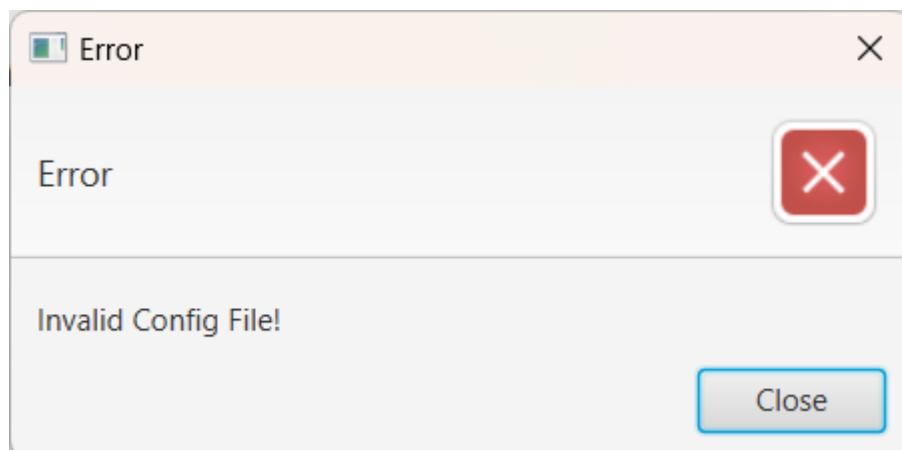
Hasil:



8. File 8.txt

```
4 5 6
DEFAULT
A
AA
B
BB
CC
C
D
DD
EE
EEE
FF
FFF
GGG
```

Hasil:



## LAMPIRAN

**Tabel Kelengkapan**

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

**Link Repositori Github:** [https://github.com/koinen/Tucil1\\_13523083](https://github.com/koinen/Tucil1_13523083)