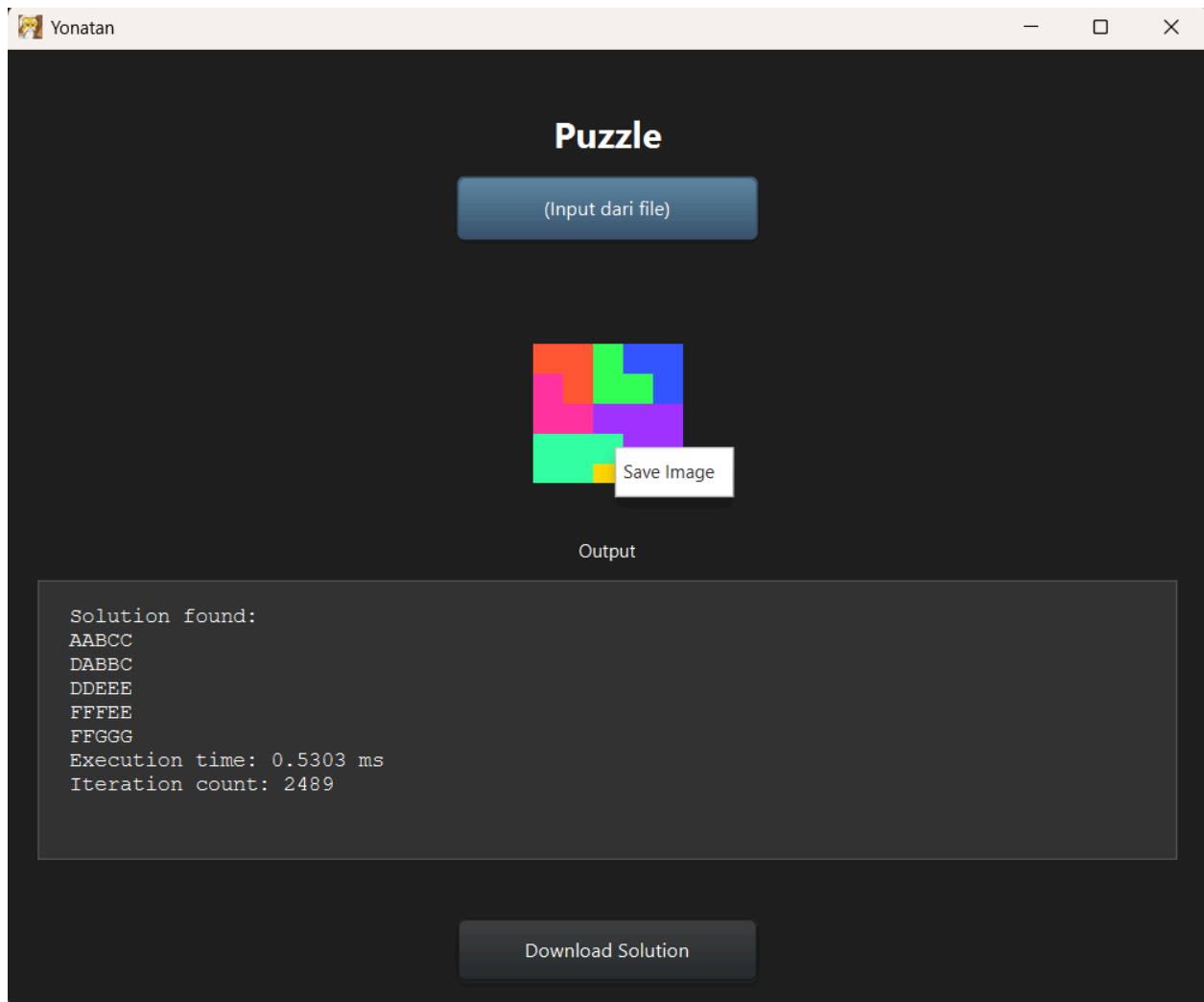


IQ Puzzler Pro Solver



Oleh:

13523036

Yonatan Edward Njoto

Daftar Isi

Daftar Isi.....	2
BAB 1. Deskripsi Masalah.....	3
BAB 2. Implementasi.....	4
I. Spek Bonus Dikerjakan.....	4
IV. Algoritma Brute Force.....	4
BAB 3. Pengujian.....	6
Lampiran.....	10

BAB 1. Deskripsi Masalah

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

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. **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.

Permainan dimulai dengan **papan yang kosong**. Pemain dapat meletakkan blok puzzle sedemikian sehingga **tidak ada blok yang bertumpang tindih** (kecuali dalam kasus 3D). Setiap blok puzzle dapat **dirotasikan** maupun **dicerminkan**. Puzzle dinyatakan **selesai** jika dan hanya jika papan **terisi penuh** dan **seluruh blok puzzle berhasil diletakkan**.

BAB 2. Implementasi

I. Spek Bonus Dikerjakan

- II. Output berupa Gambar (2 poin)
Pengguna dapat menyimpan solusi suatu puzzle dalam bentuk file gambar. Pastikan setiap blok puzzle yang ditampilkan memiliki warna berbeda.
- III. Graphical User Interface (8 poin)
Interface ini harus memvisualisasikan papan yang sudah terisi oleh blok puzzle berwarna. Kakas untuk implementasi GUI yang digunakan adalah JavaFX.

IV. Algoritma Brute Force

Brute Force diimplementasikan secara garis besar:

1. Ambil semua variasi (perputaran dan pencerminan dari piece)

```
PuzzlePiece[] transformations = {
    this, rotate90(), rotate180(), rotate270(),
    flipHorizontal(), flipHorizontal().rotate90(), flipHorizontal().rotate180(),
    flipHorizontal().rotate270()
};
```

2. Looping semua posisi, Cari posisi bisa ditaruh
 - a. Jika bisa, coba taruh dan ambil hasil return rekursi placement piece berikutnya
 - b. Jika tidak bisa, coba variasi berikutnya dari piece itu

```
private boolean solvePuzzle(int pieceIndex, List<PuzzlePiece> pieces) {
    if (pieceIndex == pieces.size()) {
        return true;
    }
    if (pieceIndex < 0 || pieceIndex >= pieces.size()) {
        System.err.println("Invalid piece index: " + pieceIndex);
        return false;
    }

    PuzzlePiece piece = pieces.get(pieceIndex);
    List<PuzzlePiece> variations = piece.generateAllVariations();

    for (int r = 0; r < rows; r++) {
        for (int c = 0; c < columns; c++) {
            for (PuzzlePiece variation : variations) {
                iterationCount++;
                if (canPlacePiece(r, c, variation, board)) {
                    placePiece(r, c, variation, board);

                    if (solvePuzzle(pieceIndex + 1, pieces)) {
                        return true;
                    }

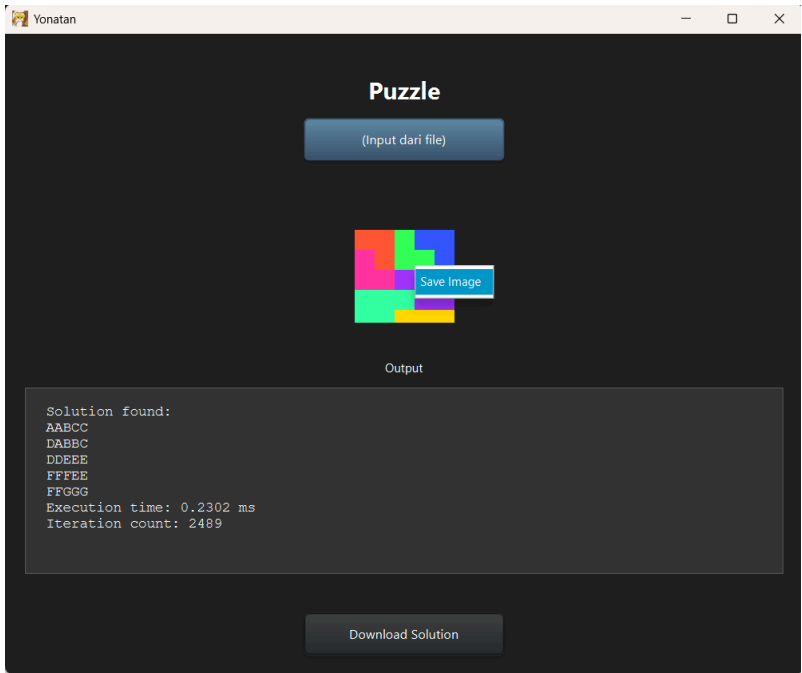
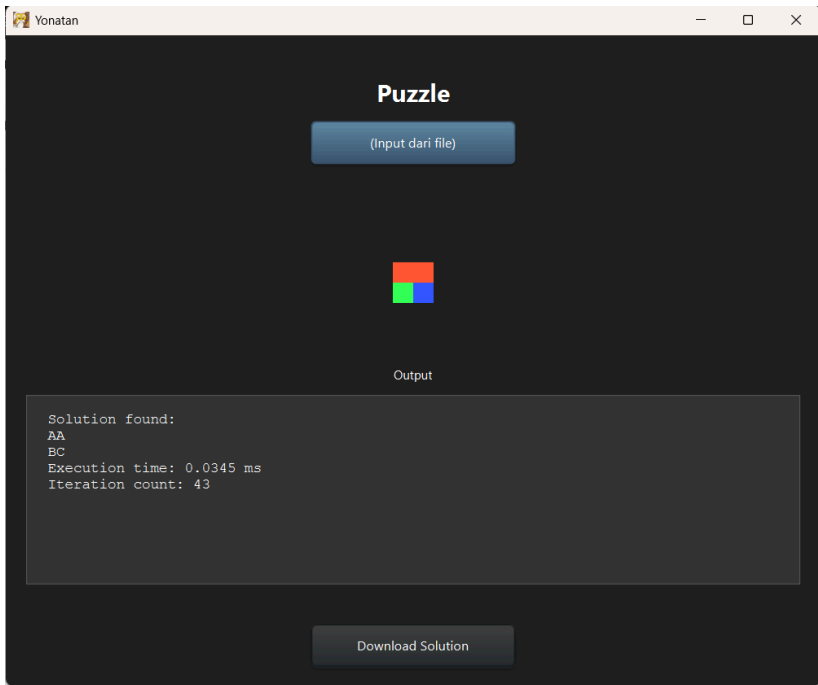
                    removePiece(r, c, variation, board);
                }
            }
        }
    }

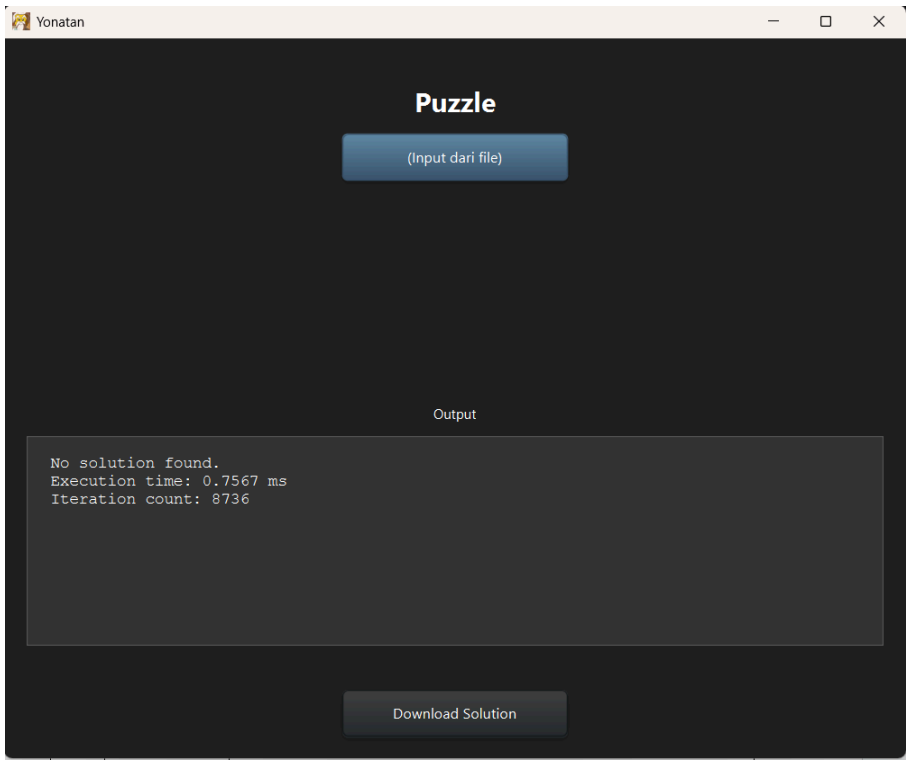
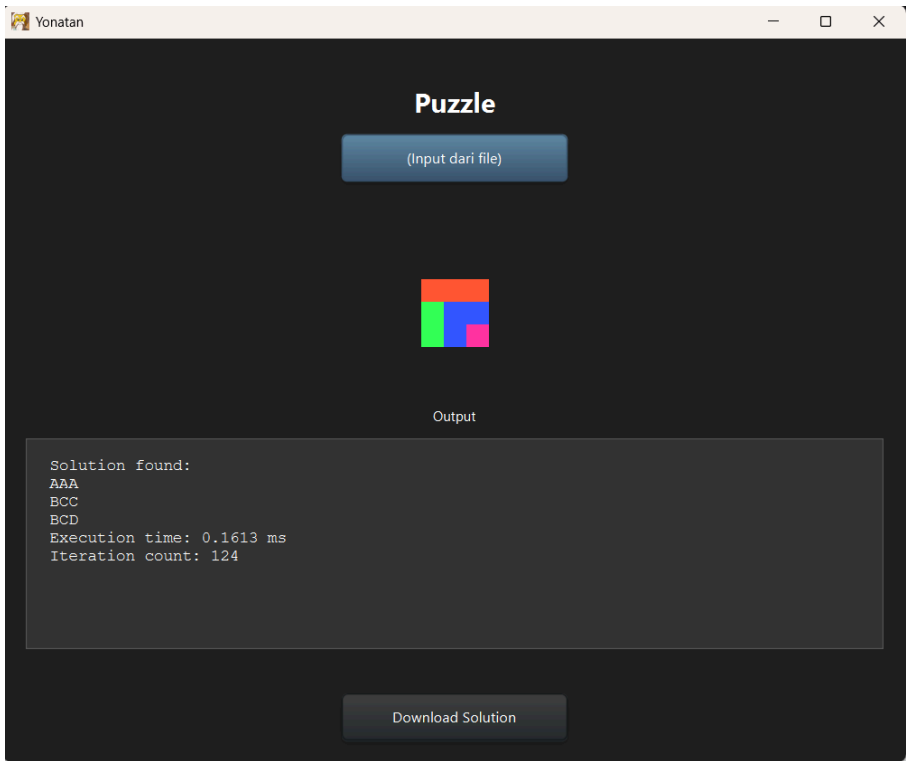
    // If no valid placement is found, return false to backtrack
    return false;
}
```

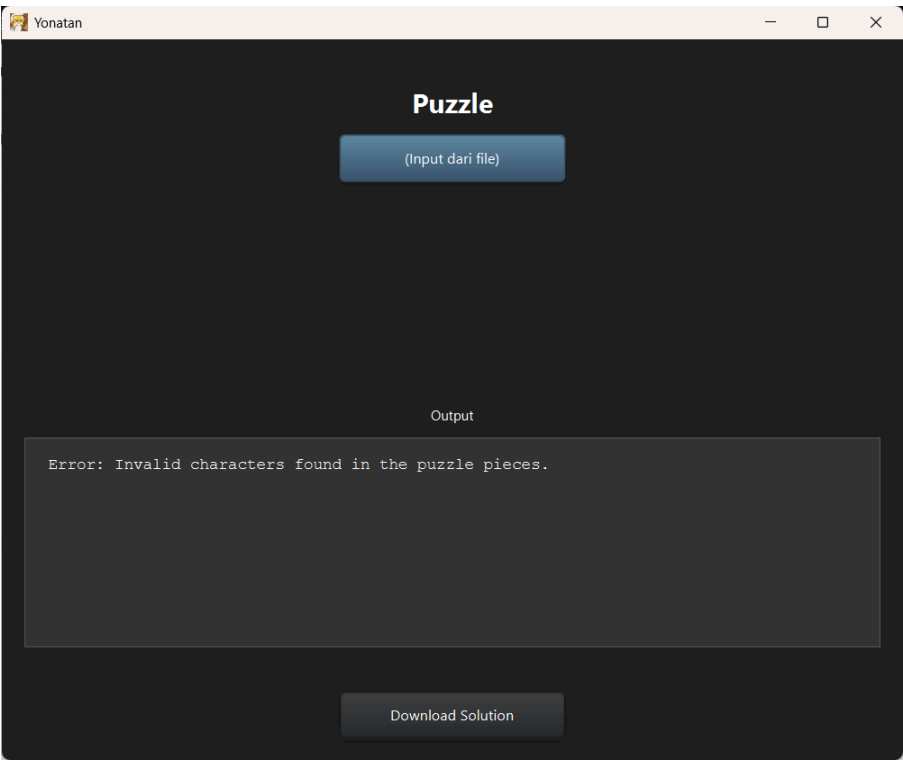
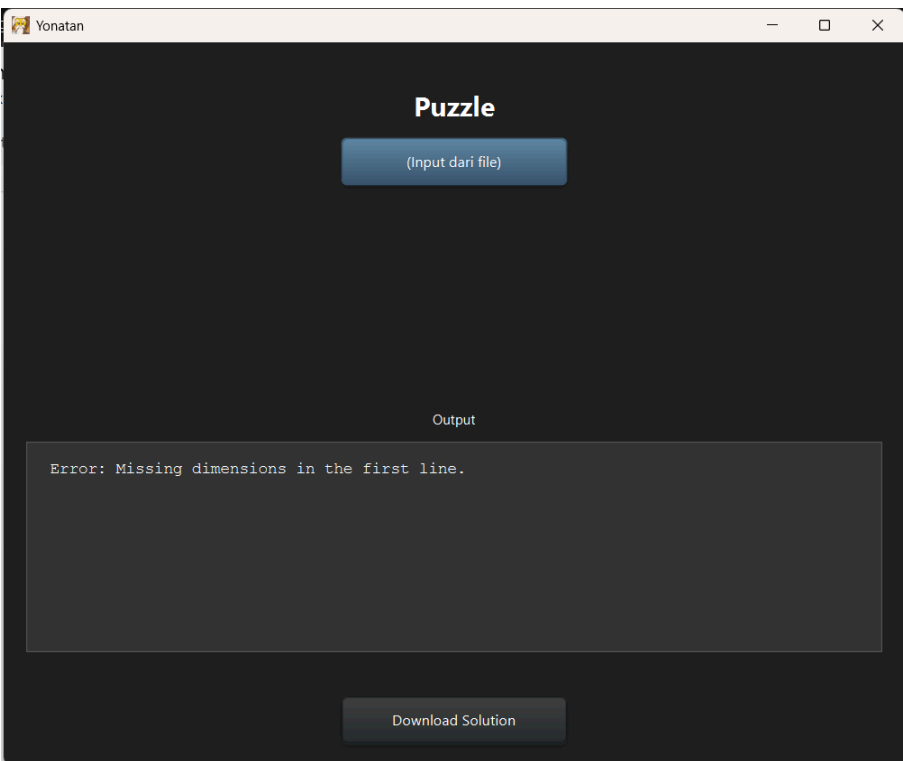
3. Jika semuanya bisa ditaruh, coba cek lagi apa satu papan sudah full dengan mengecek apabila masih ada yang '.'
 - a. Jika terisi semua, berhasil
 - b. Jika tidak terisi semua, tidak ada solusi

```
if (placedAll) {  
    // check if the whole board is filled (not '.' character)  
    solved = true;  
    for (char[] row : board) {  
        for (char cell : row) {  
            if (cell == '.') {  
                solved = false;  
                break;  
            }  
        }  
    }  
}
```

BAB 3. Pengujian

1	5 5 7 DEFAULT AA BA BB C CC D DD EE EE E FF FF F GGG	 <p>(semuanya di bawah bisa save image)</p>
2	2 2 3 DEFAULT AA BC	

3	2 2 3 DEFAULT AA BCC	
4	3 3 3 DEFAULT AAA BCC BCD	

5	3 3 3 DEFAULT AAA BCC BC.	 <p>The screenshot shows a web browser window with the title 'Yonatan'. The page has a dark background and is titled 'Puzzle'. There is a blue button labeled '(Input dari file)'. Below it, the word 'Output' is centered. A large dark gray box contains the error message: 'Error: Invalid characters found in the puzzle pieces.' At the bottom, there is a button labeled 'Download Solution'.</p>
6		 <p>The screenshot shows a web browser window with the title 'Yonatan'. The page has a dark background and is titled 'Puzzle'. There is a blue button labeled '(Input dari file)'. Below it, the word 'Output' is centered. A large dark gray box contains the error message: 'Error: Missing dimensions in the first line.' At the bottom, there is a button labeled 'Download Solution'.</p>

7	heheheha	<div><div>Yonatan</div><div><div>Puzzle</div><div>(Input dari file)</div><div>Output</div><div>Error: The first line must contain exactly 3 numbers (rows, columns, and pieces count).</div><div>Download Solution</div></div></div>
---	----------	--

Lampiran

Link repository - https://github.com/yonatan-nyo/Tucil1_13523036