

**LAPORAN TUGAS KECIL 1  
IF2211 STRATEGI ALGORITMA**

**PENYELESAIAN IQ PUZZLER PRO DENGAN  
ALGORITMA BRUTE FORCE**



**Disusun oleh:  
Kefas Kurnia Jonathan - 13523113**

**PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG**

**2025**

## Daftar Isi

Deskripsi Program.....	3
Algoritma Program.....	4
Source Code.....	5
Eksperimen.....	9
Lampiran.....	14

## BAB I

### Deskripsi Program

IQ Puzzler Pro adalah sebuah permainan papan yang meminta pemain untuk mengisi seluruh papan dengan blok yang disediakan. Permainan dimulai dengan papan kosong, dan pemain dapat meletakkan blok yang tersedia sedemikian rupa, sehingga seluruh papan terisi tanpa ada yang tertumpang tindih. Setiap blok puzzle dapat dirotasi maupun dicerminkan, baik secara horizontal maupun vertikal. Permainan dinyatakan selesai ketika papan terisi penuh dan seluruh blok puzzle diletakkan di papan.

Dalam mata kuliah IF2211 - Strategi Algoritma, mahasiswa diminta untuk membuat sebuah program yang dapat mencari salah satu solusi untuk menyelesaikan permainan ini menggunakan algoritma *brute force* murni. Program dibuat dalam bahasa Java dan mampu membaca file dengan ekstensi .txt yang berisikan konfigurasi permainan sebagai berikut:

1. Dimensi papan yang disimpan dalam variabel integer N dan M untuk papan berukuran NxM
2. Jumlah blok yang tersedia, disimpan dalam variabel integer P
3. Tipe konfigurasi yang disimpan dalam variabel string S
4. Bentuk blok puzzle yang dibentuk oleh P buah huruf yang berbeda.

Kemudian, program menampilkan tampilan akhir papan setelah permainan selesai menggunakan print berwarna, waktu eksekusi program dalam milisecond, banyak kasus atau jumlah iterasi yang ditinjau, serta pilihan untuk menyimpan hasil output ke dalam sebuah file berekstensi .txt kembali.



Gambar 1. Permainan IQ Puzzler Pro

Sumber: <https://www.smartgamesusa.com/one-player-games/iq-puzzler-pro>

## BAB II

### Algoritma Program

Untuk menyelesaikan persoalan ini menggunakan algoritma *brute force*, maka dapat dilakukan pengecekan seluruh kemungkinan yang ada, dan mengambil salah satu solusinya. Algoritma penyelesaian secara *brute force* sebagai berikut:

1. Pertama, program akan menerima input file.txt pada folder test dari user yang berisikan konfigurasi permainan. Program akan membaca file konfigurasi dan menyimpan data ukuran papan, banyaknya blok, jenis permainan, serta bentuk setiap blok.
2. Setiap bentuk blok akan disimpan dalam bentuk koordinat relatif dan program membuat papan permainan menggunakan matriks char. Setiap sel berisi '.' sebagai awalan.
3. Blok yang pertama akan dicoba dipasang di papan dalam bentuk asli blok terlebih dahulu. Jika blok berhasil dipasang, maka akan lanjut untuk mengecek blok lainnya. Program akan mengecek apakah blok dapat dipasang di posisi (x,y) aslinya atau tidak. Jika tidak, blok akan dirotasikan 90° kemudian dicek kembali. Blok dirotasi hingga 180° dan 270°. Jika masih tidak bisa, maka akan dicoba transformasi menggunakan pencerminan dan rotasi-rotasinya. Jika hingga akhir tidak bisa, maka lanjut ke posisi (x,y) selanjutnya pada matriks.
4. Jika blok bisa ditempatkan pada posisi aslinya maupun setelah melalui segala kemungkinan transformasi, blok ditempatkan pada papan, dan dilanjutkan ke blok selanjutnya.
5. Jika seluruh kemungkinan gagal setelah penempatan blok, program akan menghapus blok terakhir dari papan, kemudian mencoba kemungkinan lain (*backtracking*).
6. Ketika seluruh papan telah terisi, maka permainan dinyatakan selesai dan salah satu solusi ditemukan. Program akan menampilkan konfigurasi papan solusi, waktu pencarian solusi, banyaknya kemungkinan yang ditelusuri, serta pilihan untuk menyimpan hasil ke dalam suatu file.

Dengan algoritma ini, solusi untuk menyelesaikan IQ Puzzler Pro dapat ditentukan, meskipun akan memakan waktu yang lama untuk papan yang besar dan jumlah blok yang banyak.

Pranala Repository: [https://github.com/hgnhao/Tucil1\\_13523113](https://github.com/hgnhao/Tucil1_13523113)

## BAB III

### Source Code

```

Main.java

import java.util.Scanner;
import java.io.FileNotFoundException;

public class Main {
    public static void main(String[] args) {
        System.out.println("Selamat datang di IQ Puzzler Solver!\n");
        Scanner userInput = new Scanner(System.in);
        boolean fileReadSuccess = false;
        Solver solver = null;
        long startTime = 0, endTime = 0;
        boolean hasSolution = false;

        /* Proses membaca file */
        while (!fileReadSuccess) {
            try {
                System.out.print("Masukkan nama file (.txt): ");
                String filePath = userInput.nextLine();

                IO.readFile(filePath);
                fileReadSuccess = true;
            } catch (FileNotFoundException e) {
                System.out.println("Error: File tidak ditemukan!\nSilakan coba lagi.\n");
            } catch (Exception e) {
                System.out.println("Error: File tidak valid atau format salah!\nSilakan coba
                lagi.\n");
            }
        }

        if (!fileReadSuccess) {
            System.out.println("Program dihentikan karena file tidak valid.");
            userInput.close();
            return;
        }

        solver = new Solver(IO.N, IO.M, IO.pieces);
        startTime = System.currentTimeMillis();
        hasSolution = solver.solve();
        endTime = System.currentTimeMillis();

        /* Mencari solusi */
        if (solver != null) {
            if (hasSolution) {
                System.out.println("\nSolusi ditemukan!");
                System.out.println("\nFinal board:\n");
                IO.printColorBoard(solver.getBoard());
            } else {
                System.out.println("\nMaaf! Sepertinya tidak ada solusi.");
            }
        }

        /* Data waktu dan banyak kasus yang ditinjau */
        System.out.println("\nWaktu pencarian: " + (endTime - startTime) + " ms");
        System.out.println("Banyak kasus yang ditinjau: " + solver.getCountIteration());

        String saveChoice;
        boolean validChoice = false;
    }
}
```

```

        /* Prompt untuk menyimpan solusi */
        while (!validChoice) {
            System.out.print("\nApakah Anda ingin menyimpan solusi? (Y/N): ");
            saveChoice = userInput.nextLine().toUpperCase();

            if (saveChoice.equals("Y")) {
                System.out.print("Masukkan nama file (.txt): ");
                String savePath = userInput.nextLine();
                IO.saveToFile(savePath, solver.getBoard(), (endTime-startTime),
                    solver.getCountIteration(), hasSolution);
                ImageOutput.saveBoardAsImage(savePath, solver.getBoard());
                System.out.println("File berhasil disimpan!");
                validChoice = true;
            } else if (saveChoice.equals("N")) {
                validChoice = true;
            } else {
                System.out.println("Input tidak valid! Mohon masukkan Y atau N.");
            }
        }

        System.out.println("\nTerima kasih sudah menggunakan program ini!");
        userInput.close();
    }
}

```

```

Solve.java

import java.util.*;

public class Solver {

    public Board board;
    private List<Piece> pieces;
    public Board getBoard() { return board; }

    public Solver(int N, int M, List<Piece> pieces){
        this.board = new Board(N, M);
        this.pieces = pieces;
    }

    public boolean solve() {
        return solveRecursive(0); // mulai dari piece pertama
    }

    /* Algoritma rekursif untuk mencoba seluruh kemungkinan dan mencari solusi */
    private boolean solveRecursive(int pieceIndex) {
        /* Basis */
        if (pieceIndex >= pieces.size()) {
            return board.isFull();
        }

        if (pieceIndex == 0) {
            System.out.println("\nMencari solusi...");
        }

        Piece piece = pieces.get(pieceIndex);
    }
}

```

```

    /* Mencoba seluruh kemungkinan */
    for (int x = 0; x < board.getM(); x++) {
        for (int y = 0; y < board.getM(); y++) {
            if (tryPieceWithTransformations(piece, x, y, pieceIndex)) {
                return true;
            }
        }
    }

    /* Kasus tidak ada solusi */
    if (pieceIndex == 0) {
        System.out.println("\nMaaf! Tidak ada solusi yang ditemukan.");
    }
    return false;
}

private long countIteration = 0;
public long getCountIteration(){
    return countIteration;
}

/* Fungsi bantu untuk mencoba seluruh kemungkinan */
private boolean tryPieceWithTransformations(Piece piece, int x, int y, int pieceIndex) {
    countIteration++;

    // ORI BOS
    if (tryPlacePiece(piece, x, y, pieceIndex)) {
        return true;
    }

    // Mulai berotasi~
    Piece currentPiece = piece;
    for (int rot = 1; rot < 4; rot++) {
        currentPiece = rotatePiece(currentPiece);
        if (tryPlacePiece(currentPiece, x, y, pieceIndex)) {
            return true;
        }
    }

    // Bercermin dan rotasi~
    currentPiece = flipVertical(piece);
    if (tryPlacePiece(currentPiece, x, y, pieceIndex)) {
        return true;
    }

    for (int rot = 1; rot < 4; rot++) {
        currentPiece = rotatePiece(currentPiece);
        if (tryPlacePiece(currentPiece, x, y, pieceIndex)) {
            return true;
        }
    }

    return false;
}

private boolean tryPlacePiece(Piece piece, int x, int y, int pieceIndex) {
    if (board.canPlacePiece(piece, x, y)) {
        board.placePiece(piece, x, y);

        // Jika berhasil, dilanjutkan ke piece berikutnya
        if (solveRecursive(pieceIndex + 1)) {
            return true;
        }

        // Jika gagal, piece dihapus dari papan
        board.removePiece(piece, x, y);
    }
    return false;
}

```

```

/* FUNGSI BANTU TRANSFORMASI */

// Rotasi 90 derajat clockwise
private Piece rotatePiece(Piece piece) {
    List<int[]> newShape = new ArrayList<>();

    for (int[] coord : piece.shape) {
        int newX = -coord[1];
        int newY = coord[0];
        newShape.add(new int[]{newX, newY});
    }

    return Piece.fromCoordinates(piece.id, normalizeShape(newShape));
}

// Flip vertical
private Piece flipVertical(Piece piece) {
    List<int[]> newShape = new ArrayList<>();
    int maxY = 0;
    for (int[] coord : piece.shape) {
        maxY = Math.max(maxY, coord[1]);
    }
    for (int[] coord : piece.shape) {
        newShape.add(new int[]{coord[0], maxY - coord[1]});
    }
    return Piece.fromCoordinates(piece.id, normalizeShape(newShape));
}

/* Fungsi normalisasi */
private List<int[]> normalizeShape(List<int[]> shape) {
    int minX = Integer.MAX_VALUE, minY = Integer.MAX_VALUE;

    for (int[] coord : shape) {
        minX = Math.min(minX, coord[0]);
        minY = Math.min(minY, coord[1]);
    }

    List<int[]> normalizedShape = new ArrayList<>();
    for (int[] coord : shape) {
        normalizedShape.add(new int[]{coord[0] - minX, coord[1] - minY});
    }

    return normalizedShape;
}
}

```



## BAB IV

### Eksperimen

#### A. Masukkan file tidak ada

```
PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): tidakada.txt
Error: File tidak ditemukan!
Silakan coba lagi.

Masukkan nama file (.txt):
```

#### B. Format file.txt tidak valid

```
PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): 2.txt
Error: File tidak valid atau format salah!
Silakan coba lagi.

Masukkan nama file (.txt):
```

```
test > 2.txt
1 3 3 3
2 DEFAULT
3 AAA
4 A A
5 AAA
6 B
```

Keterangan: Jumlah P dan blok tidak sesuai.

### C. File tepat, tetapi user tidak ingin menyimpan solusi.

```
PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): 1.txt

Mencari solusi...

Solusi ditemukan!

Final board:

A B B C C
A A B D C
E E D D G
E E F F G
E F F F G

Waktu pencarian: 95 ms
Banyak kasus yang ditinjau: 60933

Apakah Anda ingin menyimpan solusi? (Y/N): N

Terima kasih sudah menggunakan program ini!
```

### D. File tepat, user ingin menyimpan solusi ke dalam file

```
PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): 1.txt

Mencari solusi...

Solusi ditemukan!

Final board:

A B B C C
A A B D C
E E D D G
E E F F G
E F F F G

Waktu pencarian: 97 ms
Banyak kasus yang ditinjau: 60933

Apakah Anda ingin menyimpan solusi? (Y/N): aaaa123123
Input tidak valid! Mohon masukkan Y atau N.

Apakah Anda ingin menyimpan solusi? (Y/N): Y
Masukkan nama file (.txt): 1.txt
File berhasil disimpan!

Terima kasih sudah menggunakan program ini!
```

Keterangan: terdapat pengecekan input juga pada prompt penyimpanan solusi.

```

test > ≡ output_1.txt
1   Solusi ditemukan!
2
3   Final board:
4   A B B C C
5   A A B D C
6   E E D D G
7   E E F F G
8   E F F F G
9   Waktu pencarian: 97 ms
10  Banyak kasus yang ditinjau: 60933
11

```

Keterangan: contoh hasil output ke dalam file.

### E. Tidak ada solusi

```

PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): 5.txt

Mencari solusi...

Maaf! Tidak ada solusi yang ditemukan.

Waktu pencarian: 3 ms
Banyak kasus yang ditinjau: 81

Apakah Anda ingin menyimpan solusi? (Y/N): n

Terima kasih sudah menggunakan program ini!

```

```

test > ≡ 5.txt
1   3 3 3
2   DEFAULT
3   A
4   | A
5   | A
6   B
7   | B
8   | B
9   C
10  | C
11  | C

```

## F. Kasus blok pertama pada file tidak rata kiri

test >	≡	4.txt
1		3 3 2
2		DEFAULT
3		AA
4		A A
5		AA
6		B
7		B
8		B

```
PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): 4.txt

Mencari solusi...

Solusi ditemukan!

Final board:

B A A
A B A
B A A

Waktu pencarian: 1 ms
Banyak kasus yang ditinjau: 2

Apakah Anda ingin menyimpan solusi? (Y/N): n

Terima kasih sudah menggunakan program ini!
```

### G. Contoh kasus lain dan disimpan ke dalam gambar

```
PS C:\Users\kefas\Documents\VSCode\ITB 196\SMT 4\STIMA\Tucil 1\Tucil1_13523113\src> java Main
Selamat datang di IQ Puzzler Solver!

Masukkan nama file (.txt): 8.txt

Mencari solusi...

Solusi ditemukan!

Final board:

A B B E C
A A B C C
A A F C D
G A F D D
G F F D D

Waktu pencarian: 7 ms
Banyak kasus yang ditinjau: 1478

Apakah Anda ingin menyimpan solusi? (Y/N): Y
Masukkan nama file (.txt): 8.txt
File berhasil disimpan!

Terima kasih sudah menggunakan program ini!
```

A	B	B	E	C
A	A	B	C	C
A	A	F	C	D
G	A	F	D	D
G	F	F	D	D

## Lampiran

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

Pranala Repository: [https://github.com/hgnhao/Tucil1\\_13523113](https://github.com/hgnhao/Tucil1_13523113)