

Laporan Tugas Kecil 1
IF2211 Strategi Algoritma
Penyelesaian IQ Puzzler Pro dengan Algoritma Brute Force



Disusun oleh:
Barru Adi Utomo
13523101

PROGRAM STUDI TEKNIK INFORMATIKA
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA
INSTITUT TEKNOLOGI BANDUNG
2024/2025

Daftar Isi

Daftar Isi.....	2
Bab 1 Identifikasi Masalah.....	3
Bab 2 Algoritma Penyelesaian IQ Puzzler Pro.....	4
Bab 3 Program.....	5
A. InputOutput.java.....	5
B. Block.java.....	7
C. Board.java.....	9
D. Algorithm.java.....	11
E. Main.java.....	13
F. Utils.java.....	14
Bab 4 Uji Coba.....	16
A. Uji Coba 1.....	16
B. Uji Coba 2.....	16
C. Uji Coba 3.....	17
D. Uji Coba 4.....	18
E. Uji Coba 5.....	18
F. Uji Coba 6.....	19
G. Uji Coba 7.....	19
H. Uji Coba 8.....	20
I. Uji Coba 9.....	21
J. Uji Coba 10.....	21

Bab 1

Identifikasi 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 (block puzzle) yang telah tersedia.



Gambar 1 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. 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.

Tugas yang dikerjakan adalah menemukan cukup satu solusi dari permainan IQ Puzzler Pro dengan menggunakan algoritma Brute Force, atau menampilkan bahwa solusi tidak ditemukan jika tidak ada solusi yang mungkin dari puzzle.

Pemain berusaha untuk mengisi bagian papan yang kosong dengan menggunakan blok yang tersedia. Permainan dinyatakan selesai jika pemain mampu mengisi seluruh papan dengan blok (dalam tugas kecil ini ada kemungkinan pemain tidak dapat mengisi seluruh papan)

Bab 2

Algoritma Penyelesaian IQ Puzzler Pro

IQ Puzzler Pro diselesaikan dengan algoritma Brute Force. Algoritma brute force dilakukan dengan mencoba setiap kemungkinan hingga terdapat satu solusi yang valid dengan melakukan iterasi semua kemungkinan piece yang ada, seperti diputar dan dicerminkan.

Setiap object Block (piece dari puzzle) mempunyai properti berupa ArrayList yang memuat segala kemungkinan perputaran piece tersebut dengan memanggil fungsi dalam object Block tersebut, sehingga setiap ArrayList mempunyai 8 kondisi berbeda.

Algoritma brute force dilakukan dengan rekursif dengan looping yang pertama untuk mencari posisi x dan y. Setiap posisi akan dicari puzzle piece dengan rotasi yang valid di lokasi tersebut. Jika lokasi tersebut valid, maka Block akan ditaruh dalam Board tersebut. Kemudian fungsi memanggil dirinya sendiri dengan index + 1 yang menyatakan Block selanjutnya. Jika piece tersebut tidak dapat dipasang, maka akan dilakukan *backtrack* dengan mengambil kembali piece sebelumnya.

Bab 3

Program

Pranala Link Repository: https://github.com/barruadi/tucil1_13523101

A. InputOutput.java

Input file

Mengambil informasi dari filename dengan format yang sudah ditentukan, jika tidak valid maka program akan berhenti dan memberikan errorinfo berdasarkan error yang didapatkan.

Jika program berhasil, maka readFile akan mengembalikan Objek Board yang sudah ditentukan ukuran dan piece yang tersedia.

```
public Board readFile(String filename) {
    try {
        File myObj = new File("test/input/" + filename + ".txt");
        Scanner reader = new Scanner(myObj);

        Integer n = null;
        if (reader.hasNextInt()) {
            n = reader.nextInt();
        } else {
            System.out.println("input tidak valid\nErrorInfo: N M P tidak sesuai format");
            reader.close();
            return new Board(0, 0, null, new char[0][0]);
        }

        Integer m = null;
        if (reader.hasNextInt()) {
            m = reader.nextInt();
        } else {
            System.out.println("input tidak valid\nErrorInfo: N M P tidak sesuai format");
            reader.close();
            return new Board(0, 0, null, new char[0][0]);
        }

        Integer p = null;
        if (reader.hasNextInt()) {
            p = reader.nextInt();
        } else {
            System.out.println("input tidak valid\nErrorInfo: N M P tidak sesuai format");
            reader.close();
            return new Board(0, 0, null, new char[0][0]);
        }

        reader.nextLine();

        String s = reader.nextLine();
        char[][] layout = new char[n][m];
        if (s.equals("DEFAULT")) {
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < m; j++) {
                    layout[i][j] = '.';
                }
            }
        } else if (s.equals("CUSTOM")) {
            for (int i = 0; i < n; i++) {
```

```

        String layoutRow = reader.nextLine();
        for (int j = 0; j < m; j++) {
            if (layoutRow.charAt(j) == '.') {
                layout[i][j] = '#';
            } else {
                layout[i][j] = '.';
            }
        }
    }

    Integer blockHeight = 1;
    Integer totalBlock = 0;
    char blockLetter = '-';
    ArrayList<Block> blocks = new ArrayList<Block>();
    Block newBlock = new Block(n, m, blockLetter);

    for (int i = 0; i < p; i++) {
        if (totalBlock >= p) {
            System.out.println("input tidak valid\nErrorInfo: total piece melebihi");
            reader.close();
            return new Board(0, 0, null, new char[0][0]);
        }
        if (reader.hasNextLine() == false) {
            newBlock.shrinkBlock();
            blocks.add(newBlock);
            totalBlock++;
            if (totalBlock < p) {
                System.out.println("input tidak valid \nErrorInfo: total piece kurang");
                reader.close();
                return new Board(0, 0, null, new char[0][0]);
            }
            break;
        }

        // read line
        String row = reader.nextLine();

        // if its new block
        if (findChar(row) != blockLetter) {
            // add block to list
            if (blockLetter != '-') {
                newBlock.shrinkBlock();
                blocks.add(newBlock);
                i--;
                totalBlock++;
            }
            // create new block
            blockLetter = findChar(row);
            newBlock = new Block(n, m, blockLetter);
            blockHeight = 1;
            if (blockHeight - 1 >= m) {
                System.out.println("input tidak valid \nErrorInfo: tinggi melebihi piece " +
blockLetter);
                reader.close();
                return new Board(0, 0, null, new char[0][0]);
            }
            newBlock.addRow(row, blockHeight - 1);
            blockHeight++;
        } else {
            if (blockHeight - 1 >= m) {
                System.out.println("input tidak valid \nErrorInfo: tinggi melebihi piece " +
blockLetter);
                reader.close();
                return new Board(0, 0, null, new char[0][0]);
            }
            newBlock.addRow(row, blockHeight - 1);

```

```

        blockHeight++;
        i--;
    }
}
reader.close();
return new Board(n, m, blocks, layout);

} catch (FileNotFoundException e) {
    System.out.println("error..");
    e.printStackTrace();
}

return new Board(0, 0, null, new char[0][0]);
}

```

Output File

Fungsi writeFile membuat file pada test/output/ dengan format .txt dengan parameter Board, filename sebagai nama file, dan time sebagai lama waktu.

```

public void writeFile(String filename, Board board, int iteration, long time) {
    try {
        File file = new File("test/output/" + filename + ".txt");
        FileWriter writer = new FileWriter(file);
        for (int i = 0; i < board.getLength(); i++) {
            for (int j = 0; j < board.getWidth(); j++) {
                writer.write(board.getBoardLocation(i, j));
            }
            writer.write("\n");
        }
        writer.write(iteration + "\n");
        writer.write(time + "\n");
        writer.close();
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

B. Block.java

Constructor

constructor membuat objek Block dengan properti: length, width, dan blockLetter.

```

public Block(int lengthBlock, int widthBlock, char blockLetter) {
    this.lengthBlock = lengthBlock;
    this.widthBlock = widthBlock;
    this.block = new char[lengthBlock][widthBlock];
}

```

```
    this.letter = blockLetter;
}
```

Getter/setter

Mengambil dan mengedit properti pada Object Block

```
public int getLengthBlock() {
    return this.lengthBlock;
}

public int getWidthBlock() {
    return this.widthBlock;
}

public char[][] getBlock() {
    return this.block;
}

public char getLetter() {
    return this.letter;
}
```

Functions

rotateBlock mengembalikan char[][] dengan Object Block yang sudah dirotasikan sebesar 90 derajat.

mirrorBlock mengubah properti block dengan mencerminkan dengan sumbu X.

getAllVariantBlock mengembalikan semua kemungkinan rotasi dan cermin dari Block.

```
public char[][] rotateBlock(char[][] block) {
    int newwidthBlock = block[0].length;
    int newlengthBlock = block.length;
    char[][] rotatedBlock = new char[newwidthBlock][newlengthBlock];
    for (int i = 0; i < newlengthBlock; i++) {
        for (int j = 0; j < newwidthBlock; j++) {
            rotatedBlock[j][newlengthBlock - 1 - i] = block[i][j];
        }
    }
    return rotatedBlock;
}

public void mirrorBlock() {
```



```

char[][] mirroredBlock = new char[lengthBlock][widthBlock];
for (int i = 0; i < lengthBlock; i++) {
    for (int j = 0; j < widthBlock; j++) {
        mirroredBlock[i][widthBlock - 1 - j] = block[i][j];
    }
}
this.block = mirroredBlock;
}

public ArrayList<char[][]> getAllVariantBlock() {
    ArrayList<char[][]> temp = new ArrayList<char[][]>();
    for (int i = 0; i < 2; i++) {
        for (int j = 0; j < 4; j++) {
            temp.add(block);
            block = rotateBlock(block);
        }
        mirrorBlock();
    }
    return temp;
}

```

C. Board.java

Constructor

constructor membuat objek Board dengan properti: length, width, ArrayList Block, dan layout.

```

public Board(int length, int width, ArrayList<Block> blocks, char[][] layout) {
    this.length = length;
    this.width = width;
    this.blocks = blocks;
    this.board = layout;
    this.blockAmount = (blocks != null) ? blocks.size() : 0;
}

```

getter/setter

Mengambil dan mengedit properti pada Object Board

```

public int getLength() {
    return this.length;
}

public int getWidth() {
    return this.width;
}

```

```

}

public int getBlockAmount() {
    return this.blockAmount;
}

public char getBoardLocation(int x, int y) {
    return this.board[x][y];
}

```

Functions

validLocation mengembalikan boolean dan memastikan apakah posisi x, y valid jika dimasukkan char[][] block dengan iterasi setiap elemen.

addBlockUsingMatrix mengubah properti layout setelah memasukkan piece yang valid.

removeBlockUsingMatrix mengubah properti layout setelah mengeluarkan piece dari Board.

```

public boolean validLocation(char[][] block, int x, int y) {
    int xOffset = Utils.findIndex(block[0]);

    for (int i = 0; i < block.length; i++) {
        for (int j = 0; j < block[0].length; j++) {
            int xloc = (x + i);
            int yloc = (y + j - xOffset);
            if (block[i][j] != '.') {
                if (xloc >= length || yloc >= width || xloc < 0 || yloc < 0) {
                    return false;
                } else if (board[xloc][yloc] != '.') {
                    return false;
                }
            }
        }
    }
    return true;
}

public void addBlockUsingMatrix(char[][] block, int x, int y) {
    int xOffset = Utils.findIndex(block[0]);

    for (int i = 0; i < block.length; i++) {
        for (int j = 0; j < block[0].length; j++) {
            if (block[i][j] != '.') {

```

```

        board[x + i][y + j - xOffset] = block[i][j];
    }
}
}

public void removeBlockUsingMatrix(char[][] block, int x, int y, char letter) {
    int xOffset = Utils.findIndex(block[0]);

    for (int i = 0; i < block.length; i++) {
        for (int j = 0; j < block[0].length; j++) {
            if (block[i][j] == letter) {
                board[x + i][y + j - xOffset] = '.';
            }
        }
    }
}
}

```

D. Algorithm.java

Constructor

constructor membuat objek Board dengan properti: length, width, ArrayList Block, dan layout.

```

private Board board;
private ArrayList<Block> blocks;
private int iteration = 0;

public Algorithm(Board board, ArrayList<Block> blocks) {
    this.board = board;
    this.blocks = blocks;
}

```

Getter/setter

Mengambil dan mengedit properti pada Object Algorithm

```

public void setIteration(int iteration) {
    this.iteration = iteration;
}

public int getIteration() {
    return this.iteration;
}

```

Algoritma

Algoritma brute force dilakukan dengan rekursif dengan looping yang pertama untuk mencari posisi x dan y. Setiap posisi akan dicari puzzle piece dengan rotasi yang valid di lokasi tersebut. Jika lokasi tersebut valid, maka Block akan ditaruh dalam Board tersebut. Kemudian fungsi memanggil dirinya sendiri dengan index + 1 yang menyatakan Block selanjutnya. Jika piece tersebut tidak dapat dipasang, maka akan dilakukan backtrack dengan mengambil kembali piece sebelumnya.

```
public boolean algorithmV2(int index, int iteration) {
    if (index == board.getBlockAmount()) {
        // make a func in board
        for (int i = 0; i < board.getLength(); i++) {
            for (int j = 0; j < board.getWidth(); j++) {
                if (board.getBoardLocation(i, j) == '.') {
                    return false;
                }
            }
        }
        setIteration(iteration);
        return true;
    }
    if (index > board.getBlockAmount()) {
        return false;
    }

    Block currentBlock = blocks.get(index);
    ArrayList<char[][]> allVariantBlock = currentBlock.getAllVariantBlock();

    for (int i = 0; i < board.getLength(); i++) {
        for (int j = 0; j < board.getWidth(); j++) {
            for (char[][] blockVariant: allVariantBlock) {
                iteration++;
                if (board.validLocation(blockVariant, i, j)) {
                    board.addBlockUsingMatrix(blockVariant, i, j);
                    if (algorithmV2(index + 1, iteration)) {
                        return true;
                    };
                    // backtrack
                    board.removeBlockUsingMatrix(blockVariant, i, j, currentBlock.getLetter());
                }
            }
        }
    }
    return false;
}
```

E. Main.java

Main Program

Menjalankan semua program algoritma brute force IQ Puzzler Pro dengan import package dari semua file

```
public static void main(String[] args) {
    // WELCOME!
    Utils.clearScreen();
    Utils.printWelcome();

    // INPUT
    while (true) {
        result = Utils.readInput(keyboardInput, "filename: ");
        if (fileInput.fileExist(result)) break;
        System.out.println("File not found.");
    }
    System.out.println();

    Board board = fileInput.readFile(result);
    if (board.getBlockAmount() == 0) {
        // error
        // System.out.println("Input tidak valid.");
        return;
    }
    ArrayList<Block> blocks = board.blocks;

    // START TIMER
    long startTime = System.currentTimeMillis();

    // ALGORITHM
    Algorithm algorithm = new Algorithm(board, blocks);
    if (algorithm.algorithmV2(0, 0)) {
        // solved
        board.printBoard();
        status = true;
    } else {
        System.out.println("tidak ada solusi!");
    }

    // END TIMER
    long endTime = System.currentTimeMillis();

    // OUTPUT
    long executionTime = endTime - startTime;
    if (status) {
        System.out.println("\nWaktu Pencarian: " + executionTime + " ms\n");
        System.out.println("Banyak kasus ditinjau: " + algorithm.getIteration() + "\n");
    }
}
```

```

    }

    result = Utils.readInput(keyboardInput, "Apakah anda ingin menyimpan solusi? (ya/tidak): ");
    if (result.equals("ya")) {
        result = Utils.readInput(keyboardInput, "filename: ");
        fileInput.writeFile(result, board, algorithm.getIteration(), executionTime);
        System.out.println("Solusi berhasil disimpan di" + result + ".txt");
    } else {
        System.out.println("hasil tidak disimpan!");
    }
    System.out.println("Terima kasih!");
}

```

F. Utils.java

```

package utils;

import java.util.Scanner;

public class Utils {
    String colors[] = {
        "\u001B[30m", // Black
        "\u001B[31m", // Red
        "\u001B[32m", // Green
        "\u001B[33m", // Yellow
        "\u001B[34m", // Blue
        "\u001B[35m", // Magenta
        "\u001B[36m", // Cyan
        "\u001B[37m", // White
        "\u001B[90m", // Bright Black (Gray)
        "\u001B[91m", // Bright Red
        "\u001B[92m", // Bright Green
        "\u001B[93m", // Bright Yellow
        "\u001B[94m", // Bright Blue
        "\u001B[95m", // Bright Magenta
        "\u001B[96m", // Bright Cyan
        "\u001B[97m", // Bright White
        "\u001B[38;5;202m", // Orange
        "\u001B[38;5;208m", // Deep Orange
        "\u001B[38;5;214m", // Light Orange
        "\u001B[38;5;172m", // Gold
        "\u001B[38;5;120m", // Lime Green
        "\u001B[38;5;57m", // Purple
        "\u001B[38;5;198m", // Pink
        "\u001B[38;5;118m", // Light Green
        "\u001B[38;5;27m", // Deep Blue
        "\u001B[38;5;135m", // Lavender
    }
}

```

```

};

String reset = "\u001B[0m";

public void printColor(char alphabet) {
    if (alphabet == '#' || alphabet == '.') {
        System.out.print(' ');
        return;
    }
    System.out.print(colors[alphabet - 'A'] + alphabet + reset);
};

public static void printWelcome() {
    System.out.println("+-----+");
    System.out.println("|  Welcome to IQ Puzzler Pro Solver!  |");
    System.out.println("+-----+");
};

public static String readInput(Scanner input, String line) {
    System.out.print(line);
    String result = input.nextLine();
    return result;
}

public static void clearScreen() {
    System.out.print("\033[H\033[2J");
    System.out.flush();
}

public static int findIndex(char[] row) {
    for (int i = 0; i < row.length; i++) {
        if (row[i] != '.') {
            return i;
        }
    }
    return -1;
}
}

```

Bab 4

Uji Coba

A. Uji Coba 1

Testcase:

```
5 5 5
DEFAULT
A
AAAAA
BBB
BB
CCC
CC
DDDD
EEEE
```

```
+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test4

ADDDD
AAAAA
BBBCC
BBCCC
EEEE

Waktu Pencarian: 17 ms

Banyak kasus ditinjau: 359

Apakah anda ingin menyimpan solusi? (ya/tidak): ya
filename: hasiltest4
Solusi berhasil disimpan dihasiltest4.txt
Terima kasih!
```

B. Uji Coba 2

Testcase:

```
6 5 7
DEFAULT
AAAA
A A
A A
A A
BB
```



```
CC
DDD
D
EEE
E
FFF
  FF
GG
G
```

```
+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test2

AAAAB
ACDAB
ACDAE
ADDAE
GFFEE
GGFFF

Waktu Pencarian: 28 ms

Banyak kasus ditinjau: 600

Apakah anda ingin menyimpan solusi? (ya/tidak): tidak
hasil tidak disimpan!
Terima kasih!
```

C. Uji Coba 3

Testcase:

```
5 7 5
CUSTOM
...X...
.XXXXX.
XXXXXXXX
.XXXXX.
...X...
A
AAA
BB
BBB
CCCC
C
D
EEE
E
```

```

+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test3

  A
BBAAA
BBBCCC
EEECD
E

Waktu Pencarian: 31 ms

Banyak kasus ditinjau: 621

Apakah anda ingin menyimpan solusi? (ya/tidak): tidak
hasil tidak disimpan!
Terima kasih!

```

D. Uji Coba 4

Testcase:

```

5 5 3
DEFAULT
AAA
BBB
B BBB
CCCCC

```

```

+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: tidak
File not found.
filename: test5

tidak ada solusi!
Terima kasih!

```

E. Uji Coba 5

Testcase:

```

5 5 5
CUSTOM
..X..
XXXXX
XXXXX
XXXX.
...X.
A
AAA

```

A
BB
BB
CCC
C
D
EE

```
+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test6

  A
AAABBB
ADCBBB
CCCE
  E

Waktu Pencarian: 18 ms

Banyak kasus ditinjau: 416

Apakah anda ingin menyimpan solusi? (ya/tidak): ya
filename: hasiltest6
Solusi berhasil disimpan dihasiltest6.txt
Terima kasih!
```

F. Uji Coba 6

Testcase:

3 3 2
DEFAULT
AAA
BB
CCC
C

```
+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test7

input tidak valid
ErrorInfo: total piece melebihi
```

G. Uji Coba 7

Testcase:

5 6 5

CUSTOM

..####

#####.
####..
AAA
 BB
BBBB
CCCC
 C
DDD
D D
 E
 EE
EE

```
+-----+  
| Welcome to IQ Puzzler Pro Solver! |  
+-----+  
filename: test8  
  
  BBBB  
  BBAAA  
  CCCEE  
  DCDEE  
  DDDE  
  
Waktu Pencarian: 61 ms  
  
Banyak kasus ditinjau: 467  
  
Apakah anda ingin menyimpan solusi? (ya/tidak): tidak  
hasil tidak disimpan!  
Terima kasih!
```

H. Uji Coba 8

Testcase:

2 2
DEFAULT
AA
A
B

```

+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test9

input tidak valid
ErrorInfo: N M P tidak sesuai format

```

I. Uji Coba 9

Testcase:

```

3 3 4
AA
AA
BBB
CC

```

```

+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test9

input tidak valid
ErrorInfo: total piece kurang

```

J. Uji Coba 10

Testcase:

```

4 3 3
CUSTOM
.XX
XXX
..X
XXX
AA
AA
BB
CCC

```

```

+-----+
| Welcome to IQ Puzzler Pro Solver! |
+-----+
filename: test9

tidak ada solusi!
Terima kasih!

```

LAMPIRAN

•

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