

LAPORAN TUGAS KECIL I

IF2211 - STRATEGI ALGORITMA



Disusun oleh :

Muhammad Raihan Nazhim Oktana – 13523021

Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

2025

Daftar Isi

Daftar Isi.....	2
BAB I DESKRIPSI PROGRAM.....	3
1.1. Deskripsi Permainan.....	3
1.2. Deskripsi Tugas.....	3
BAB II ALGORITMA BRUTE FORCE.....	5
2.1. Algoritma Brute Force.....	5
BAB III SOURCE CODE.....	11
3.1. Program Main.java.....	11
3.2. Program Puzzle.java.....	23
3.3. Program Solve2D.java.....	27
3.4. Program Solve3D.java.....	31
3.5. Program ReadFile.java.....	34
3.6. Program SaveFile.java.....	43
BAB IV TEST CASE.....	49
4.1. Test Case Tipe Default.....	49
4.2. Test Case Tipe Custom.....	51
4.3. Test Case Tipe Pyramid.....	54
4.4. Test Case Lainnya.....	55
LAMPIRAN.....	56

BAB I

DESKRIPSI PROGRAM

1.1. Deskripsi Permainan

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 2 hal sebagai berikut.

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. Piece (Blok) : 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.

1.2. Deskripsi Tugas

Program 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. Berikut adalah spesifikasi wajibnya :

- Buatlah program sederhana dalam bahasa Java yang mengimplementasikan algoritma Brute Force untuk mencari solusi dalam permainan IQ Puzzler Pro.
- Algoritma brute force yang diimplementasikan harus bersifat “murni”, tidak boleh memanfaatkan heuristik.
- Papan yang perlu diisi mulanya akan selalu kosong.
- Sebuah blok puzzle bisa saja dirotasi maupun dicerminkan sebelum diletakan pada papan.
- Input :

Program akan memberikan pengguna sebuah prompt untuk memilih file test case berekstensi .txt, kemudian program membaca file test case tersebut yang berisi :

1. Dimensi Papan terdiri atas dua buah variabel N dan M yang membentuk papan berdimensi N x M.
2. Banyak blok puzzle direpresentasikan oleh variabel integer P.
3. Jenis kasus sebuah variabel string S yang digunakan untuk mengidentifikasi kasus konfigurasi, hanya mungkin bernilai salah satu diantara DEFAULT / CUSTOM / PYRAMID.
4. Bentuk blok puzzle yang dilambangkan oleh konfigurasi Character berupa huruf. Akan ada P buah blok puzzle berbeda yang dibentuk oleh P buah huruf berbeda. Character yang digunakan adalah huruf A-Z dalam kapital.

```
N M P
S
puzzle_1_shape
puzzle_2_shape
...
puzzle_P_shape
```

- Output :

1. Tampilkan konfigurasi blok puzzle yang berhasil mengisi papan. Gunakan print berwarna untuk menunjukkan blok puzzle dengan jelas. Pastikan setiap blok puzzle berbeda memiliki warna berbeda. Beri tahu pengguna apabila puzzle tidak memiliki solusi.
2. Waktu eksekusi program dalam milisecond (tidak termasuk waktu membaca masukan dan menyimpan solusi, cukup waktu pencarian oleh algoritma).
3. Banyak kasus atau jumlah iterasi yang ditinjau oleh algoritma brute force.
4. Prompt untuk menyimpan solusi dalam sebuah berkas berekstensi .txt (Struktur untuk file output dibebaskan).

BAB II

ALGORITMA BRUTE FORCE

2.1. Algoritma Brute Force

IQ Puzzler Pro adalah permainan papan yang salah satu penyelesaian secara pemrograman adalah dengan algoritma brute force. Meskipun ada algoritma lain yang lebih baik, algoritma brute force tanpa heuristik menjadi pusat pembelajaran pada kesempatan kali ini.

Pada kesempatan kali ini, kita hanya akan fokus membahas pada algoritma solver dengan brute forcenya, kita akan mengecek dasar-dasarnya terlebih dahulu. Yang pertama adalah struktur data / kelas koordinat seperti pada gambar berikut:

```
class Koordinat:
    method Constructor(x , y):
        this.x ← x
        this.y ← y

    method equals(obj):
        if this == obj:
            return TRUE
        else if obj == NULL or this.getClass() != obj.getClass():
            return FALSE
        else:
            other ← (Koordinat) obj
            return this.x == other.x and this.y == other.y

    method hashCode():
        return 31 * this.x + this.y
end class
```

Struktur data / kelas koordinat ini membantu kita merepresentasikan setiap elemen piece dalam bentuk kumpulan dari titik-titik koordinat yang mewakilinya. Terdapat constructor untuk membuat sebuah koordinat dari 2 titik x dan y, pengecekan kesamaan, dan hashcode untuk membantu proses pemrograman.

Selanjutnya, struktur data / kelas yang kedua adalah PuzzlePiece. Bentuknya adalah kumpulan dari data-data yang merepresentasikan suatu piece, mulai dari namanya, ukurannya, hingga posisi detailnya seperti pada gambar berikut:

```
class PuzzlePiece:
```

```

variables:
  name: char
  position: List of Koordinat
  height, width: integers

method Constructor(name , position , height , width):
  this.name ← name
  this.position ← position
  this.height ← height
  this.width ← width

method rotate():
  npos ← new List of Koordinat
  maxHeight ← height - 1
  for each e in position:
    nx ← e.y
    ny ← maxHeight - e.x
    npos.add(new Koordinat(nx , ny))
  return normalize(name , npos)

method flip():
  npos ← new List of Koordinat
  maxCol ← width - 1
  for each e in position:
    nx ← e.x
    ny ← maxCol - e.y
    npos.add(new Koordinat(nx , ny))
  return normalize(name, npos)

private method normalize(id , opos):
  minRow ← MAX_INTEGER
  minCol ← MAX_INTEGER
  for each e in opos:
    minRow ← min(minRow, e.x)
    minCol ← min(minCol, e.y)
  npos ← new List of Koordinat
  for each e in opos:
    npos.add(new Koordinat(e.x - minRow, e.y - minCol))
  maxRow ← MIN_INTEGER
  maxCol ← MIN_INTEGER
  for each e in npos:
    maxRow ← max(maxRow, e.x)
    maxCol ← max(maxCol, e.y)
  newHeight ← maxRow + 1
  newWidth ← maxCol + 1
  return new PuzzlePiece(id , npos , newHeight , newWidth)

method generate():
  result ← new List of PuzzlePiece
  current ← this
  for i from 0 to 3:
    if i > 0:
      current ← current.rotate()
    if not check_list(result , current):

```

```

        result.add(current)
        flipped ← current.flip()
        if not check_list(result , flipped):
            result.add(flipped)
    return result

private method check_list(list, piece):
    for each item in list:
        if check_identical(item , piece):
            return TRUE
    return FALSE

private method check_identical(a , b):
    if a.position.size() != b.position.size():
        return FALSE
    else if a.height != b.height or a.width != b.width:
        return FALSE
    else:
        for each ea in a.position:
            check ← FALSE
            for each eb in b.position:
                if ea.x == eb.x and ea.y == eb.y:
                    check ← TRUE
                    break
            if not check:
                return FALSE
        return TRUE
end class

```

Struktur data / kelas PuzzlePiece ini membantu kita merepresentasikan setiap elemen piece dalam bentuk kumpulan dari titik-titik koordinat yang mewakilinya. Terdapat constructor untuk membuat sebuah piece, rotasi 90 derajat, pencerminan, peyesuaian ulang detail ukuran setelah diubah-ubah, pengecekan kesamaan, dan mesin pembuat semua kemungkinan untuk membantu proses pemrograman brute force mencoba seluruh kemungkinan susunan puzzle.

Selanjutnya, kita masuk pada fungsi bantuan yang membantu proses pemrograman. Namun, pada kesempatan ini, hanya akan dibahas versi 2D-nya saja, karena versi 3D-nya tidak sempat diselesaikan dan masih ada beberapa penyesuaian yang harus dilakukan. Yang membedakan hanya lebih banyak variasinya karena harus mengatasi koordinat x, y, dan z.

Fungsi bantuan yang pertama adalah inisialisasi papan untuk menyiapkan papan sebelum digunakan pengecekan. Secara manual iterasi diisi dengan tanda ‘.’ sebagai penanda boleh diisi. Dalam versi custom, pemasukan tipe disesuaikan dengan input board dari file yang telah diinputkan dengan tetap iterasi manual berulang. Pseudocodenya seperti pada gambar berikut:

```

function initializeBoard():
    for i from 0 to N-1:

```

```

        for j from 0 to M-1:
            board[i][j] ← '.'
        end for
    end for
end function

```

Fungsi bantuan yang kedua adalah mengecek apakah papan udah penuh saat semua piece sudah diisi untuk menentukan apakah solved atau tidak. Secara manual iterasi dicek jika ada '.', maka ada ruang kosong yang menandakan belum penuh. Pseudocodenya seperti pada gambar berikut:

```

function checkFull2D():
    for i from 0 to N-1:
        for j from 0 to M-1:
            if board[i][j] == '.':
                return FALSE
            end if
        end for
    end for
    return TRUE
end function

```

Fungsi bantuan yang ketiga adalah mengecek apakah di suatu posisi bisa diisi dengan sebuah piece. Secara manual iterasi dilakukan pengecekan apakah semua titik berada dalam papan dan apakah semua titik masih '.', atau sudah ada yang mengisi. Pseudocodenya seperti pada gambar berikut:

```

function checkSpot2D(piece , px , py):
    for each e in piece.position:
        rx ← px + e.x
        ry ← py + e.y
        if rx < 0 or rx >= N or ry < 0 or ry >= M:
            return FALSE
        else if board[rx][ry] != '.':
            return FALSE
        end if
    end for
    return TRUE
end function

```

Fungsi bantuan yang keempat adalah memasukkan piece secara manual ke dalam papan jika telah di cek bahwa itu memungkinkan. Secara manual iterasi dilakukan pemasukan input secara satu demi satu hingga semua bagian piece dimasukkan. Pseudocodenya seperti pada gambar berikut:

```

function fill2D(piece, px, py, id):
    for each e in piece.position:
        rx ← px + e.x

```



```

        ry ← py + e.y
        board[rx][ry] ← id
    end for
end function

```

Selanjutnya adalah fungsi utama dari solver2D yang bertujuan untuk melakukan brute force utama dengan sistem brute force : backtracking, dan sekaligus menghitung banyaknya langkah percobaan kasus yang dilakukan. Proses ini dilakukan dengan menguji peletakan setiap kemungkinan piece pada setiap petak yang ada, dan setiap kasus itu ditambah 1 perhitungan langkah. Jika indeks dari piece yang dicoba telah sampai pada indeks ke-P, maka akan dicek apakah papan sudah penuh atau belum, jika penuh maka solved, dan jika tidak maka failed. Proses ini sekilas seperti memasukkan 1 kemungkinan lalu menyuruh sistem mengecek, dan jika gagal maka akan dicabut lagi dan dibersihkan. Pseudocodenya seperti pada gambar berikut:

```

function solve2D(idx):
    if not solved:
        if idx == P:
            if checkFull2D():
                solved ← TRUE
            end if
        else:
            for each e in allSet[idx]:
                for x from 0 to N-1:
                    for y from 0 to M-1:
                        cnt ← cnt + 1
                        if checkSpot2D(e , x , y):
                            fill2D(e , x , y , e.name)
                            solve2D(idx + 1)
                            if not solved:
                                fill2D(e , x , y , '.')
                            end if
                        end if
                    end for
                end for
            end for
        end if
    end if
end function

```

Sekarang, masuk pada pembahasan 2 fungsi utama yang menjadi pusat dari program yang memanggil semuanya, yaitu fungsi default_solver dan custom_solver yang akan menggunakan seluruh fungsi bantuan yang telah dibuat. Fungsi default_solver akan menjalankan proses jika tipenya adalah Default, sedangkan Fungsi custom_solver akan menjalankan prosesnya jika tipenya adalah Custom. Sederhananya, mereka akan menginisialisasi papan dan menggenerate semua kemungkinan piece seperti yang saya sudah jelaskan sebelumnya, lalu

memanggil fungsi solve2D yang dimulai dari indeks 0 atau piece pertama. Jika hasil akhirnya solved, maka akan ditampilkan papan hasil akhir, dan jika tidak, maka akan ditampilkan pesan bahwa tidak ada solusi yang ditemukan. Pseudocodenya seperti pada gambar berikut:

```
function defaultSolver():
    print "-----"
    print "          PROSES PENCARIAN SOLUSI DEFAULT          "
    print "-----"
    print "          LOADING...          "
    print "-----"
    initialize board[n][m] to '.'
    clear allSet
    for each e in allPiece:
        allSet.add(e.generate())
    solve2D(0)
    print "          HASIL PENCARIAN SOLUSI DEFAULT          "
    print "-----"
    if solved:
        showBoard()
    else:
        print "Tidak ada solusi yang ditemukan!"
end function
```

```
function customSolver():
    print "-----"
    print "          PROSES PENCARIAN SOLUSI CUSTOM          "
    print "-----"
    print "          LOADING...          "
    print "-----"
    initialize board[n][m]
    for x from 0 to n-1:
        lineBoard ← customBoard[x]
        for y from 0 to m-1:
            ch ← lineBoard.charAt(y)
            if ch == 'X':
                board[x][y] ← '.'
            else:
                board[x][y] ← ' '
    clear allSet
    for each e in allPiece:
        allSet.add(e.generate())
    solve2D(0)
    print "          HASIL PENCARIAN SOLUSI CUSTOM          "
    print "-----"
    if solved:
        showBoard()
    else:
        print "Tidak ada solusi yang ditemukan!"
end function
```

Seluruh fungsi-fungsi tersebut bekerja sama dalam Main.java seperti pada umumnya, begitupun dengan fungsi `read_file`, `save_file`, dan lainnya. Namun fungsi-fungsi yang lain tidak saya jelaskan di sini karena fokus pembahasannya hanya pada algoritma brute forcenya.

BAB III

SOURCE CODE

3.1. Program Main.java

```
// PROGRAM K01-13523021-Tucil1-F01

// IDENTITAS
// NIM>Nama   : 13523021 - Muhammad Raihan Nazhim Oktana
// Instansi   : Sekolah Teknik Elektro dan Informatika (STEI-Komputasi) Institut
Teknologi Bandung (ITB)
// Jurusan   : Teknik Informatika - Kampus Ganesha (IF-G)
// Nama File  : Main.java
// Topik      : Tugas Kecil 1 Strategi Algoritma 2025 (IF2211-24)
// Tanggal    : Senin, 24 Februari 2025
// Deskripsi  : Subprogram F01 - Main

// KAMUS
// Main : class
// main , deafult_solver , custom_solver , pyramid_solver : function
// s : String
// n , m , p : integer
// solved : boolean
// start , end , cnt : long
// board : matrix of character
// custom_board : list of String
// all_piece : list of PuzzlePiece
// all_set : matrix of PuzzlePiece
// pyramid : list of matrix of character

// ALGORITMA
package src;
import java.util.ArrayList;
```

```

import java.util.List;
import java.util.Scanner;

public class Main {

    public static int n , m , p;
    public static String s;
    public static char[][] board;
    public static char[][][] pyramid;
    public static boolean solved = false;
    public static List<String> custom_board = new ArrayList<>();
    public static List<Puzzle.PuzzlePiece> all_piece = new ArrayList<>();
    public static List<List<Puzzle.PuzzlePiece>> all_set = new ArrayList<>();
    public static long start , end , cnt = 0;

    public static void main(String[] args) {
        System.out.println("                Selamat Datang di Game IQ Puzzle Pro
");

System.out.println("-----
");
        System.out.println("                MENU
");

System.out.println("-----
");
        System.out.println("1. Main Game
");
        System.out.println("2. Keluar
");

System.out.println("-----
");
        Scanner scanner = new Scanner(System.in);
        System.out.println("WARNING : Input 'EXIT' To Force Exit!!");
        System.out.print("Masukkan Pilihan Anda (1/2) : ");
        String string = scanner.nextLine().trim();
        while (!string.equals("EXIT") && !string.equals("1") &&

```

```

!string.equals("2")) {
    System.out.println("Error : Input Tidak Boleh Kosong / Tidak
Valid!");
    System.out.print("Masukkan Pilihan Anda (1/2) : ");
    string = scanner.nextLine().trim();
}
if (string.equals("1")) {
    while (true) {

System.out.println("-----
");
        System.out.println("                                GAME START
");

System.out.println("-----
");

        System.out.println("WARNING : Input 'EXIT' To Force Exit!!");
        System.out.print("Masukkan nama file input (pakai .txt) : ");
        String file_name = scanner.nextLine().trim();
        while (file_name.isEmpty()) {
            System.out.println("Error : Input Nama File Tidak Boleh
Kosong!");
            System.out.print("Masukkan nama file input (pakai .txt) :
");

            file_name = scanner.nextLine().trim();
        }
        if (file_name.equals("EXIT")) {
            break;
        } else {
            boolean valid = ReadFile.read_file(file_name);
            boolean force_stop = false;
            while (!valid) {
                System.out.print("Masukkan nama file input (pakai .txt)
: ");

                file_name = scanner.nextLine().trim();
                while (file_name.isEmpty()) {
                    System.out.println("Error : Input Nama File Tidak
Boleh Kosong!");

```

```

        System.out.print("Masukkan nama file input (pakai
.txt) : ");

        file_name = scanner.nextLine().trim();
    }
    if (file_name.equals("EXIT")) {
        force_stop = true;
        break;
    }
    valid = ReadFile.read_file(file_name);
}
if (valid) {
    String arg = "Tidak ada solusi yang ditemukan!";
    start = System.currentTimeMillis();
    if (s.equals("DEFAULT")) {
        default_solver();
    } else if (s.equals("CUSTOM")) {
        custom_solver();
    } else {
        pyramid_solver();
        if (!solved) {
            arg = "Maaf, PYRAMID Solver Tidak
Tersedia!\nTidak ada solusi yang ditemukan!";
        }
    }
    end = System.currentTimeMillis();
    long time = end - start;
    System.out.printf("Waktu pencarian : %d ms.\n" , time);
    System.out.printf("Banyak kasus yang ditinjau : %d
kasus.\n" , cnt);

    System.out.println("-----
");

    System.out.println("
SIMPAN HASIL KE
FILE TEKS?
");

    System.out.println("-----
");

    System.out.println("WARNING : Input 'EXIT' To Force

```

```

Exit!!");

        System.out.print("Apakah Hasil Mau Anda Save Ke File
Teks (y/n) : ");

        String respond = scanner.nextLine().trim();
        while (!respond.equals("EXIT") && !respond.equals("y")
&& !respond.equals("n")) {
            System.out.println("Error : Input Tidak Boleh
Kosong / Tidak Valid!");
            System.out.print("Apakah Hasil Mau Anda Save Ke
File Teks (y/n) : ");
            respond = scanner.nextLine().trim();
        }
        if (respond.equals("EXIT")) {
            break;
        } else if (respond.equals("y")) {

System.out.println("-----
");

                System.out.println("                SISTEM
PENYIMPANAN TEKS                ");

System.out.println("-----
");

                System.out.println("WARNING : Input 'EXIT' To Force
Exit!!");

                System.out.print("Masukkan nama file teks output
(pakai .txt) : ");

                file_name = scanner.nextLine().trim();
                while (file_name.isEmpty()) {
                    System.out.println("Error : Input Nama File
Tidak Boleh Kosong!");
                    System.out.print("Masukkan nama file teks
output (pakai .txt) : ");
                    file_name = scanner.nextLine().trim();
                }
                if (file_name.equals("EXIT")) {
                    break;
                } else {

```

```

        valid = SaveFile.save_text_2D(file_name , board
, time , cnt , arg);

        while (!valid) {
            System.out.print("Masukkan nama file output
(pakai .txt) : ");

            file_name = scanner.nextLine().trim();
            while (file_name.isEmpty()) {
                System.out.println("Error : Input Nama
File Tidak Boleh Kosong!");

                System.out.print("Masukkan nama file
output (pakai .txt) : ");

                file_name = scanner.nextLine().trim();
            }
            if (file_name.equals("EXIT")) {
                force_stop = true;
                break;
            }
            valid = SaveFile.save_text_2D(file_name ,
board , time , cnt , arg);
        }
        if (force_stop) {
            break;
        }
    }
    if (solved) {

System.out.println("-----
");

        System.out.println("SIMPAN HASIL KE
FILE GAMBAR? ");

System.out.println("-----
");

        System.out.println("WARNING : Input 'EXIT' To Force
Exit!!");

        System.out.print("Apakah Hasil Mau Anda Save Ke
File Gambar (y/n) : ");

```



```

        respond = scanner.nextLine().trim();
        while (!respond.equals("EXIT") &&
!respond.equals("y") && !respond.equals("n")) {
            System.out.println("Error : Input Tidak Boleh
Kosong / Tidak Valid!");
            System.out.print("Apakah Hasil Mau Anda Save Ke
File Gambar (y/n) : ");
            respond = scanner.nextLine().trim();
        }
        if (respond.equals("EXIT")) {
            break;
        } else if (respond.equals("y")) {

System.out.println("-----
");
            System.out.println("                SISTEM
PENYIMPANAN GAMBAR                ");

System.out.println("-----
");
            System.out.println("WARNING : Input 'EXIT' To
Force Exit!!");
            System.out.print("Masukkan nama file gambar
output (pakai .png) : ");
            file_name = scanner.nextLine().trim();
            while (file_name.isEmpty()) {
                System.out.println("Error : Input Nama File
Tidak Boleh Kosong!");
                System.out.print("Masukkan nama file gambar
output (pakai .png) : ");
                file_name = scanner.nextLine().trim();
            }
            if (file_name.equals("EXIT")) {
                break;
            } else {
                valid = SaveFile.save_image_2D(file_name ,
board);
                while (!valid) {

```

```

                                System.out.print("Masukkan nama file
gambar output (pakai .png) : ");

                                file_name = scanner.nextLine().trim();
                                while (file_name.isEmpty()) {
                                    System.out.println("Error : Input
Nama File Tidak Boleh Kosong!");

                                    System.out.print("Masukkan nama
file gambar output (pakai .png) : ");

                                    file_name =
scanner.nextLine().trim();

                                }
                                if (file_name.equals("EXIT")) {
                                    force_stop = true;
                                    break;
                                }
                                valid =
SaveFile.save_image_2D(file_name , board);
                                }
                                if (force_stop) {
                                    break;
                                }
                            }
                        }
                    }

System.out.println("-----
");

                                System.out.println("                                MENU
");

System.out.println("-----
");

                                System.out.println("1. Main Game
");

                                System.out.println("2. Keluar
");

System.out.println("-----

```

```

");
        System.out.println("WARNING : Input 'EXIT' To Force
Exit!!");

        System.out.print("Masukkan Pilihan Anda (1/2) : ");
        string = scanner.nextLine().trim();
        while (!string.equals("EXIT") && !string.equals("1") &&
!string.equals("2")) {
            System.out.println("Error : Input Tidak Boleh
Kosong / Tidak Valid!");
            System.out.print("Masukkan Pilihan Anda (1/2) : ");
            string = scanner.nextLine().trim();
        }
        if (string.equals("EXIT") || string.equals("2")) {
            break;
        }
        } else if (force_stop) {
            break;
        }
    }
}

System.out.println("-----
");

System.out.println("-----
");
        System.out.println("                ALHAMDULILLAH TUCIL 1 STIMA SELESAI
");

System.out.println("-----
");
        System.out.println("                BISMILLAH DAPAT A - 100
");

System.out.println("-----
");
        System.out.println("IDENTITAS PEMBUAT :

```

```

");
    System.out.println("Nama      : Muhammad Raihan Nazhim Oktana
");
    System.out.println("NIM      : 13523021 / IF-G / K-01
");
    System.out.println("Instansi : Teknik Informatika - Institut Teknologi
Bandung ");
    scanner.close();
}

public static void default_solver() {

System.out.println("-----
");
    System.out.println("                PROSES PENCARIAN SOLUSI DEFAULT
");

System.out.println("-----
");
    System.out.println("                LOADING...
");

System.out.println("-----
");
    board = new char[n][m];
    for (int x = 0 ; x < n ; x++) {
        for (int y = 0 ; y < m ; y++) {
            board[x][y] = '.';
        }
    }
    all_set.clear();
    for (Puzzle.PuzzlePiece e : all_piece) {
        all_set.add(e.generate());
    }
    Solve2D.solve_2D(0);
    System.out.println("                HASIL PENCARIAN SOLUSI DEFAULT
");

```

```

System.out.println("-----
");
    if (solved) {
        Solve2D.show_board();
    } else {
        System.out.println("Tidak ada solusi yang ditemukan!");
    }
}

public static void custom_solver() {

System.out.println("-----
");
    System.out.println("                PROSES PENCARIAN SOLUSI CUSTOM
");

System.out.println("-----
");
    System.out.println("                LOADING...
");

System.out.println("-----
");
    board = new char[n][m];
    for (int x = 0 ; x < n ; x++) {
        String line_board = custom_board.get(x);
        for (int y = 0 ; y < m ; y++) {
            char ch = line_board.charAt(y);
            if (ch == 'X') {
                board[x][y] = '.';
            } else {
                board[x][y] = ' ';
            }
        }
    }
    all_set.clear();
    for (Puzzle.PuzzlePiece e : all_piece) {
        all_set.add(e.generate());
    }
}

```

```

    }
    Solve2D.solve_2D(0);
    System.out.println("                HASIL PENCARIAN SOLUSI CUSTOM
");
");
System.out.println("-----
");
    if (solved) {
        Solve2D.show_board();
    } else {
        System.out.println("Tidak ada solusi yang ditemukan!");
    }
}

public static void pyramid_solver() {
System.out.println("-----
");
    System.out.println("                PROSES PENCARIAN SOLUSI PYRAMID
");
");
System.out.println("-----
");
    System.out.println("                LOADING...
");
");
System.out.println("-----
");
    pyramid = new char[n][m][m];
    for (int i = 0 ; i < n ; i++) {
        for (int j = 0 ; j < i + 1 ; j++) {
            for (int k = 0 ; k < i + 1 ; k++) {
                pyramid[i][j][k] = '.';
            }
        }
    }
    // Solve3D.solve_3D(0);
    System.out.println("                HASIL PENCARIAN SOLUSI PYRAMID

```

```
");

System.out.println("-----");
");

    if (solved) {
        Solve3D.show_pyramid();
    } else {
        System.out.println("Maaf, PYRAMID Solver Tidak Tersedia!");
        System.out.println("Tidak ada solusi yang ditemukan!");
    }
}
}
```

3.2. Program Puzzle.java

```
// PROGRAM K01-13523021-Tucil1-F02

// IDENTITAS
// NIM>Nama   : 13523021 - Muhammad Raihan Nazhim Oktana
// Instansi   : Sekolah Teknik Elektro dan Informatika (STEI-Komputasi) Institut
Teknologi Bandung (ITB)
// Jurusan   : Teknik Informatika - Kampus Ganesha (IF-G)
// Nama File  : Puzzle.java
// Topik      : Tugas Kecil 1 Strategi Algoritma 2025 (IF2211-24)
// Tanggal    : Senin, 24 Februari 2025
// Deskripsi  : Subprogram F02 - Puzzle

// KAMUS
// ...

// ALGORITMA
package src;
import java.util.ArrayList;
import java.util.List;

public class Puzzle {
```

```

public static class Koordinat {

    public int x , y;

    public Koordinat(int x , int y) {
        this.x = x;
        this.y = y;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) {
            return true;
        } else if (obj == null || getClass() != obj.getClass()) {
            return false;
        } else {
            Koordinat ot = (Koordinat) obj;
            return (x == ot.x && y == ot.y);
        }
    }

    @Override
    public int hashCode() {
        return 31 * x + y;
    }
}

public static class PuzzlePiece {
    public char name;
    public List<Koordinat> position;
    public int height;
    public int width;

    public PuzzlePiece(char name , List<Koordinat> position , int height ,
int width) {
        this.name = name;
        this.position = position;
        this.height = height;
    }
}

```



```

        this.width = width;
    }

    public PuzzlePiece rotate() {
        List<Koordinat> npos = new ArrayList<>();
        int mh = height - 1;
        for (Koordinat e : position) {
            int nx = e.y;
            int ny = mh - e.x;
            npos.add(new Koordinat(nx , ny));
        }
        return normalize(name , npos);
    }

    public PuzzlePiece flip() {
        List<Koordinat> npos = new ArrayList<>();
        int mc = width - 1;
        for (Koordinat e : position) {
            int nx = e.x;
            int ny = mc - e.y;
            npos.add(new Koordinat(nx , ny));
        }
        return normalize(name , npos);
    }

    private static PuzzlePiece normalize(char id , List<Koordinat> opos) {
        int mir = Integer.MAX_VALUE , mic = Integer.MAX_VALUE;
        for (Koordinat e : opos) {
            mir = Math.min(mir , e.x);
            mic = Math.min(mic , e.y);
        }
        List<Koordinat> npos = new ArrayList<>();
        for (Koordinat e : opos) {
            npos.add(new Koordinat(e.x - mir , e.y - mic));
        }
        int mar = Integer.MIN_VALUE , mac = Integer.MIN_VALUE;
        for (Koordinat e : npos) {
            mar = Math.max(mar , e.x);

```

```

        mac = Math.max(mac , e.y);
    }
    int nh = mar + 1;
    int nw = mac + 1;
    return new PuzzlePiece(id , npos , nh , nw);
}

public List<PuzzlePiece> generate() {
    List<PuzzlePiece> res = new ArrayList<>();
    PuzzlePiece cur = this;
    for (int i = 0 ; i < 4 ; i++) {
        if (i > 0) {
            cur = cur.rotate();
        }
        if (!check_list(res , cur)) {
            res.add(cur);
        }
        PuzzlePiece flp = cur.flip();
        if (!check_list(res , flp)) {
            res.add(flp);
        }
    }
    return res;
}

private boolean check_list(List<PuzzlePiece> list , PuzzlePiece comp) {
    for (PuzzlePiece e : list) {
        if (check_identical(e , comp)) {
            return true;
        }
    }
    return false;
}

private boolean check_identical(PuzzlePiece a , PuzzlePiece b) {
    if (a.position.size() != b.position.size()) {
        return false;
    } else if (a.height != b.height || a.width != b.width) {

```

```

        return false;
    } else {
        for (Koordinat ea : a.position) {
            boolean check = false;
            for (Koordinat eb : b.position) {
                if (ea.x == eb.x && ea.y == eb.y) {
                    check = true;
                    break;
                }
            }
            if (!check) {
                return false;
            }
        }
        return true;
    }
}
}
}
}
}

```

3.3. Program Solve2D.java

```

// PROGRAM K01-13523021-Tucil1-F03

// IDENTITAS
// NIM>Nama   : 13523021 - Muhammad Raihan Nazhim Oktana
// Instansi   : Sekolah Teknik Elektro dan Informatika (STEI-Komputasi) Institut
Teknologi Bandung (ITB)
// Jurusan   : Teknik Informatika - Kampus Ganesha (IF-G)
// Nama File  : Solve2D.java
// Topik      : Tugas Kecil 1 Strategi Algoritma 2025 (IF2211-24)
// Tanggal    : Senin, 24 Februari 2025
// Deskripsi  : Subprogram F03 - Solve2D

// KAMUS
// KAMUS
// Solve2D : class
// check_full_2D , solve_2D , check_spot_2D , fill_2D , show_board : function

```

```

// RESET : String
// LIST_WARNA : List of String

// ALGORITMA
package src;
import src.Puzzle.Koordinat;
import src.Puzzle.PuzzlePiece;

public class Solve2D {

    public static final String RESET = "\u001B[0m";
    public static final String[] LIST_WARNA = {
        "\u001B[31m",      // Merah
        "\u001B[32m",      // Hijau
        "\u001B[33m",      // Kuning
        "\u001B[34m",      // Biru
        "\u001B[35m",      // Magenta
        "\u001B[36m",      // Cyan
        "\u001B[91m",      // Merah Terang
        "\u001B[92m",      // Hijau Terang
        "\u001B[93m",      // Kuning Terang
        "\u001B[94m",      // Biru Terang
        "\u001B[95m",      // Magenta Terang
        "\u001B[96m",      // Cyan Terang
        "\u001B[90m",      // Abu-abu Gelap
        "\u001B[37m",      // Abu-abu Terang
        "\u001B[38;5;160m", // Merah Gelap
        "\u001B[38;5;28m",  // Hijau Gelap
        "\u001B[38;5;220m", // Kuning Emas
        "\u001B[38;5;21m",  // Biru Gelap
        "\u001B[38;5;129m", // Lavender
        "\u001B[38;5;87m",  // Biru Langit Terang
        "\u001B[38;5;123m", // Biru Cyan Terang
        "\u001B[38;5;227m", // Kuning Cerah
        "\u001B[38;5;208m", // Oranye
        "\u001B[38;5;196m", // Merah Cerah
        "\u001B[38;5;245m", // Abu-abu Tua
        "\u001B[38;5;243m"  // Abu-abu Sedang
    };
}

```

```

};

public static boolean check_full_2D() {
    for (int i = 0 ; i < Main.n ; i++) {
        for (int j = 0 ; j < Main.m ; j++) {
            if (Main.board[i][j] == '.') {
                return false;
            }
        }
    }
    return true;
}

public static void solve_2D(int idx) {
    if (!Main.solved) {
        if (idx == Main.p) {
            if (check_full_2D()) {
                Main.solved = true;
            }
        } else {
            for (PuzzlePiece e : Main.all_set.get(idx)) {
                for (int x = 0 ; x < Main.n ; x++) {
                    for (int y = 0 ; y < Main.m ; y++) {
                        Main.cnt++;
                        if (check_spot_2D(e , x , y)) {
                            fill_2D(e , x , y , e.name);
                            solve_2D(idx + 1);
                            if (!Main.solved) {
                                fill_2D(e , x , y , '.');
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

public static boolean check_spot_2D(PuzzlePiece piece , int px , int py) {
    for (Koordinat e : piece.position) {
        int rx = px + e.x;
        int ry = py + e.y;
        if (rx < 0 || rx >= Main.n || ry < 0 || ry >= Main.m) {
            return false;
        } else if (Main.board[rx][ry] != '.') {
            return false;
        }
    }
    return true;
}

public static void fill_2D(PuzzlePiece piece , int px , int py , char id) {
    for (Koordinat e : piece.position) {
        int rx = px + e.x;
        int ry = py + e.y;
        Main.board[rx][ry] = id;
    }
}

public static void show_board() {
    for (int x = 0 ; x < Main.n ; x++) {
        for (int y = 0 ; y < Main.m ; y++) {
            char c = Main.board[x][y];
            if (c >= 'A' && c <= 'Z') {
                int idx = c - 'A';
                if (idx >= 0 && idx < LIST_WARNALength) {
                    System.out.print(LIST_WARNALength[idx] + c + RESET);
                } else {
                    System.out.print(c);
                }
            } else {
                System.out.print(c);
            }
        }
        System.out.println();
    }
}

```

```
}
}
```

3.4. Program Solve3D.java

```
// PROGRAM K01-13523021-Tucil1-F04

// IDENTITAS
// NIM>Nama   : 13523021 - Muhammad Raihan Nazhim Oktana
// Instansi   : Sekolah Teknik Elektro dan Informatika (STEI-Komputasi) Institut
Teknologi Bandung (ITB)
// Jurusan    : Teknik Informatika - Kampus Ganesha (IF-G)
// Nama File   : Solve3D.java
// Topik       : Tugas Kecil 1 Strategi Algoritma 2025 (IF2211-24)
// Tanggal     : Senin, 24 Februari 2025
// Deskripsi   : Subprogram F04 - Solve3D

// KAMUS
// Solve3D : class
// check_full_3D , solve_3D , check_spot_3D , fill_3D , show_pyramid : function
// RESET : String
// LIST_WARNA : List of String

// ALGORITMA
package src;
import src.Puzzle.Koordinat;
import src.Puzzle.PuzzlePiece;

public class Solve3D {

    public static final String RESET = "\u001B[0m";
    public static final String[] LIST_WARNA = {
        "\u001B[31m",    // Merah
        "\u001B[32m",    // Hijau
        "\u001B[33m",    // Kuning
        "\u001B[34m",    // Biru
        "\u001B[35m",    // Magenta
        "\u001B[36m",    // Cyan
    }
```

```

        "\u001B[91m",      // Merah Terang
        "\u001B[92m",      // Hijau Terang
        "\u001B[93m",      // Kuning Terang
        "\u001B[94m",      // Biru Terang
        "\u001B[95m",      // Magenta Terang
        "\u001B[96m",      // Cyan Terang
        "\u001B[90m",      // Abu-abu Gelap
        "\u001B[37m",      // Abu-abu Terang
        "\u001B[38;5;160m", // Merah Gelap
        "\u001B[38;5;28m", // Hijau Gelap
        "\u001B[38;5;220m", // Kuning Emas
        "\u001B[38;5;21m",  // Biru Gelap
        "\u001B[38;5;129m", // Lavender
        "\u001B[38;5;87m",  // Biru Langit Terang
        "\u001B[38;5;123m", // Biru Cyan Terang
        "\u001B[38;5;227m", // Kuning Cerah
        "\u001B[38;5;208m", // Oranye
        "\u001B[38;5;196m", // Merah Cerah
        "\u001B[38;5;245m", // Abu-abu Tua
        "\u001B[38;5;243m"  // Abu-abu Sedang
    };

    public static boolean check_full_3D() {
        for (int i = 0 ; i < Main.n ; i++) {
            for (int j = 0 ; j < Main.m ; j++) {
                for (int k = 0 ; k < Main.m ; k++) {
                    if (Main.pyramid[i][j][k] == '.') {
                        return false;
                    }
                }
            }
        }
        return true;
    }

    public static void solve_3D(int idx) {
        if (!Main.solved) {
            if (idx == Main.p) {

```



```

        if (check_full_3D()) {
            Main.solved = true;
        }
    } else {
        for (PuzzlePiece e : Main.all_set.get(idx)) {
            for (int x = 0 ; x < Main.n ; x++) {
                for (int y = 0 ; y < Main.m ; y++) {
                    for (int z = 0 ; z < Main.m ; z++) {
                        Main.cnt++;
                        if (check_spot_3D(e , x , y , z)) {
                            fill_3D(e , x , y , z , e.name);
                            solve_3D(idx + 1);
                            if (!Main.solved) {
                                fill_3D(e , x , y , z , '.');
                            }
                        }
                    }
                }
            }
        }
    }
}

public static boolean check_spot_3D(PuzzlePiece piece , int px , int py ,
int pz) {
    for (Koordinat e : piece.position) {
        int rx = px + e.x;
        int ry = py + e.y;
        int rz = pz + e.x + e.y;
        if (rx < 0 || rx >= Main.n || ry < 0 || ry >= Main.m || rz < 0 ||
rz >= Main.m) {
            return false;
        } else if (Main.pyramid[rx][ry][rz] != '.') {
            return false;
        }
    }
    return true;
}

```

```

    }

    public static void fill_3D(PuzzlePiece piece , int px , int py , int pz ,
char id) {
        for (Koordinat e : piece.position) {
            int rx = px + e.x;
            int ry = py + e.y;
            int rz = pz + e.x + e.y;
            Main.pyramid[rx][ry][rz] = id;
        }
    }

    public static void show_pyramid() {
        for (int x = 0 ; x < Main.n ; x++) {
            for (int y = 0 ; y < x + 1 ; y++) {
                for (int z = 0 ; z < x + 1 ; z++) {
                    char c = Main.pyramid[x][y][z];
                    if (c >= 'A' && c <= 'Z') {
                        int idx = c - 'A';
                        if (idx >= 0 && idx < LIST_WARNALength) {
                            System.out.print(LIST_WARNALength[idx] + c + RESET);
                        } else {
                            System.out.print(c);
                        }
                    } else {
                        System.out.print(c);
                    }
                }
                System.out.println();
            }
        }
    }
}

```

3.5. Program ReadFile.java

```
// PROGRAM K01-13523021-Tucil1-F05
```

```
// IDENTITAS
// NIM>Nama   : 13523021 - Muhammad Raihan Nazhim Oktana
// Instansi   : Sekolah Teknik Elektro dan Informatika (STEI-Komputasi) Institut
Teknologi Bandung (ITB)
// Jurusan   : Teknik Informatika - Kampus Ganesha (IF-G)
// Nama File  : ReadFile.java
// Topik      : Tugas Kecil 1 Strategi Algoritma 2025 (IF2211-24)
// Tanggal    : Senin, 24 Februari 2025
// Deskripsi  : Subprogram F05 - Read File

// KAMUS
// ReadFile : class
// read_file , build_piece , connected : Function
// BASE_DIRECTORY_IN : String

// ALGORITMA
package src;
import java.io.File;
import java.io.FileNotFoundException;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.List;
import java.util.Scanner;
import java.util.Stack;
import src.Puzzle.PuzzlePiece;

public class ReadFile {

    public static final String BASE_DIRECTORY_IN = "../test/input/";

    public static boolean read_file(String file_name) {
        String[] list_illegal_char = {"/" , "\\" , ":" , "*" , "?" , "\"" , "<"
, ">" , "|"};
        String[] allowed_name = {"A" , "B" , "C" , "D" , "E" , "F" , "G" , "H"
, "I" , "J" , "K" , "L" , "M" , "N" , "O" , "P" , "Q" , "R" , "S" , "T" , "U" ,
"V" , "W" , "X" , "Y" , "Z"};
        StringBuilder found_illegal = new StringBuilder();
        for (String e : list_illegal_char) {
```

```

        if (file_name.contains(e)) {
            if (found_illegal.length() > 0) {
                found_illegal.append(" , ");
            }
            found_illegal.append(e);
        }
    }
    if (found_illegal.length() > 0) {
        System.out.println("Error : Gagal Membaca File : Nama file
mengandung karakter yang tidak diizinkan (" + found_illegal + ").");
        return false;
    } else {
        Main.custom_board.clear();
        Main.all_piece.clear();
        Main.all_set.clear();
        Main.solved = false;
        Main.cnt = 0;
        if (!file_name.endsWith(".txt")) {
            System.out.println("Error : Gagal Membaca File : Format file
harus (.txt).");
            return false;
        } else {
            String file_path = BASE_DIRECTORY_IN + file_name;
            File file = new File(file_path);
            if (!file.exists()) {
                System.out.println("Error : Gagal Membaca File : " +
file_path + " tidak ditemukan.");
                return false;
            } else {
                try {
                    Scanner scn = new Scanner(file);
                    if (!scn.hasNextInt()) {
                        System.out.println("Error : N bukan integer.");
                        scn.close();
                        return false;
                    } else {
                        Main.n = scn.nextInt();
                    }
                }
            }
        }
    }
}

```

```

        if (!scn.hasNextInt()) {
            System.out.println("Error : M bukan integer.");
            scn.close();
            return false;
        } else {
            Main.m = scn.nextInt();
        }
        if (!scn.hasNextInt()) {
            System.out.println("Error : P bukan integer.");
            scn.close();
            return false;
        } else {
            Main.p = scn.nextInt();
        }
        if (Main.n < 1 || Main.m < 1 || Main.p < 1) {
            System.out.println("Error : N , M , P harus berupa
bilangan bulat positif.");
            scn.close();
            return false;
        } else if (Main.p > 26) {
            System.out.println("Error : P harus berupa bilangan
bulat positif yang tidak lebih dari 26.");
            scn.close();
            return false;
        }
        if (!scn.hasNext()) {
            System.out.println("Error : Tidak ada tipe puzzle
(DEFAULT / CUSTOM / PYRAMID).");
            scn.close();
            return false;
        } else {
            Main.s = scn.next().trim();
            scn.nextLine();
            if (!Main.s.equals("DEFAULT") &&
!Main.s.equals("CUSTOM") && !Main.s.equals("PYRAMID")) {
                System.out.println("Error : Tipe puzzle hanya
ada DEFAULT / CUSTOM / PYRAMID.");
                scn.close();
            }
        }
    }
}

```

```

        return false;
    }
}
if (Main.s.equals("CUSTOM")) {
    for (int x = 0 ; x < Main.n ; x++) {
        if (!scn.hasNextLine()) {
            System.out.println("Error : Baris papan
permainan kurang dari " + Main.n + " baris.");
            scn.close();
            return false;
        } else {
            String board_line = scn.nextLine();
            if (board_line.length() != Main.m) {
                System.out.println("Error : Panjang
baris ke-" + (x + 1) + " bukan " + Main.m + ".");
                scn.close();
                return false;
            } else {
                Main.custom_board.add(board_line);
                for (int y = 0 ; y < Main.m ; y++) {
                    char c = board_line.charAt(y);
                    if (c != '.' && c != 'X') {
                        System.out.println("Error :
Board hanya boleh '.' atau 'X'.");
                        scn.close();
                        return false;
                    }
                }
            }
        }
    }
} else if (Main.s.equals("PYRAMID")) {
    if (Main.n != Main.m) {
        System.out.println("Error : Pada tipe puzzle
PYRAMID nilai N dan M harus sama.");
        scn.close();
        return false;
    }
}

```

```

    }
    List<Character> vname = new ArrayList<>();
    scn.useDelimiter("\r?\n");
    String line_comp = null;
    for (int i = 0 ; i < Main.p ; i++) {
        char target = (char) ('A' + i);
        List<String> piece_line = new ArrayList<>();
        boolean check_piece = false;
        if (line_comp != null) {
            if (line_comp.trim().isEmpty()) {
                break;
            }
            char f = line_comp.trim().charAt(0);
            if
((!Arrays.asList(allowed_name).contains(String.valueOf(f)))) {
                System.out.println("Error : Potongan puzzle
memiliki karakter yang tidak valid.");
                scn.close();
                return false;
            }
            target = f;
            for (int j = 0 ; j < line_comp.length() ; j++)
{
                if (line_comp.charAt(j) != ' ' &&
line_comp.charAt(j) != target) {
                    System.out.println("Error : Baris
potongan puzzle mengandung huruf campuran : " + line_comp + ".");
                    scn.close();
                    return false;
                } else if
((!Arrays.asList(allowed_name).contains(String.valueOf(line_comp.charAt(j))))
{
                    System.out.println("Error : Potongan
puzzle memiliki karakter yang tidak valid.");
                    scn.close();
                    return false;
                }
            }
        }
    }
}

```

```

        if (vname.contains(target)) {
            System.out.println("Error : Terdapat
duplikat potongan puzzle '" + target + "'.");
            scn.close();
            return false;
        } else {
            vname.add(target);
        }
        piece_line.add(line_comp);
        check_piece = true;
        line_comp = null;
    }
    boolean over = false;
    while (scn.hasNextLine()) {
        String line = scn.nextLine();
        if (line.trim().isEmpty()) {
            break;
        }
        char f = line.trim().charAt(0);
        if
((!Arrays.asList(allowed_name).contains(String.valueOf(f)))) {
            System.out.println("Error : Potongan puzzle
memiliki karakter yang tidak valid.");
            scn.close();
            return false;
        }
        if (i == 0 && !over) {
            target = f;
            over = true;
            vname.add(target);
        }
        if (f != target) {
            if (i == Main.p - 1) {
                System.out.println("Error : Terlalu
banyak potongan. Seharusnya hanya " + Main.p + " potongan.");
                scn.close();
                return false;
            } else {

```



```

        line_comp = line;
        break;
    }
}
for (int c = 0 ; c < line.length() ; c++) {
    char ch = line.charAt(c);
    if (ch != ' ' && ch != target) {
        System.out.println("Error : Baris
potongan puzzle mengandung huruf campuran : " + line + ".");
        scn.close();
        return false;
    } else if
((!Arrays.asList(allowed_name).contains(String.valueOf(ch)))) {
        System.out.println("Error : Potongan
puzzle memiliki karakter yang tidak valid.");
        scn.close();
        return false;
    }
}
piece_line.add(line);
check_piece = true;
}
if (!check_piece) {
    System.out.println("Error : Potongan puzzle
ke-" + (i + 1) + " tidak ada di data.");
    scn.close();
    return false;
}
PuzzlePiece piece = build_piece(piece_line ,
target);

if (!connected(piece.position)) {
    System.out.println("Error : Potongan puzzle
ke-" + (i + 1) + " tidak terhubung.");
    scn.close();
    return false;
} else {
    Main.all_piece.add(piece);
}
}

```

```

        }
        scn.close();
        return true;
    } catch (FileNotFoundException e) {
        System.out.println("Error : Gagal Membaca File : File "
+ file_name + " tidak ditemukan.");
        return false;
    } catch (Exception e) {
        System.out.println("Error : Gagal Membaca File : " +
e.getMessage());
        return false;
    }
}
}
}
}

private static PuzzlePiece build_piece(List<String> str_shape , char name)
{
    List<Puzzle.Koordinat> opos = new ArrayList<>();
    int mir = Integer.MAX_VALUE , mic = Integer.MAX_VALUE;
    int mar = Integer.MIN_VALUE , mac = Integer.MIN_VALUE;
    for (int x = 0 ; x < str_shape.size() ; x++) {
        String str = str_shape.get(x);
        for (int y = 0 ; y < str.length() ; y++) {
            char c = str.charAt(y);
            if (c == name) {
                opos.add(new Puzzle.Koordinat(x , y));
                mir = Math.min(mir , x);
                mic = Math.min(mic , y);
                mar = Math.max(mar , x);
                mac = Math.max(mac , y);
            }
        }
    }
    List<Puzzle.Koordinat> npos = new ArrayList<>();
    for (Puzzle.Koordinat e : opos) {
        npos.add(new Puzzle.Koordinat(e.x - mir , e.y - mic));
    }
}

```

```

    }
    int nh = mar - mir + 1;
    int nw = mac - mic + 1;
    return new PuzzlePiece(name , npos , nh , nw);
}

private static boolean connected(List<Puzzle.Koordinat> vpos) {
    List<Puzzle.Koordinat> vvis = new ArrayList<>();
    Stack<Puzzle.Koordinat> stack = new Stack<>();
    stack.push(vpos.get(0));
    while (!stack.isEmpty()) {
        Puzzle.Koordinat cur = stack.pop();
        if (!vvis.contains(cur)) {
            vvis.add(cur);
            int[] dx = {1 , 1 , 1 , 0 , 0 , -1 , -1 , -1};
            int[] dy = {1 , 0 , -1 , 1 , -1 , 1 , 0 , -1};
            for (int i = 0 ; i < 8 ; i++) {
                Puzzle.Koordinat npos = new Puzzle.Koordinat(cur.x + dx[i]
, cur.y + dy[i]);
                if (vpos.contains(npos) && !vvis.contains(npos)) {
                    stack.push(npos);
                }
            }
        }
    }
    return vvis.size() == vpos.size();
}
}

```

3.6. Program SaveFile.java

```

// PROGRAM K01-13523021-Tucil1-F06

// IDENTITAS
// NIM>Nama   : 13523021 - Muhammad Raihan Nazhim Oktana
// Instansi   : Sekolah Teknik Elektro dan Informatika (STEI-Komputasi) Institut
Teknologi Bandung (ITB)
// Jurusan   : Teknik Informatika - Kampus Ganesha (IF-G)

```

```

// Nama File : SaveFile.java
// Topik      : Tugas Kecil 1 Strategi Algoritma 2025 (IF2211-24)
// Tanggal    : Senin, 24 Februari 2025
// Deskripsi  : Subprogram F06 - Save File

// KAMUS
// SaveFile : class
// save_text_2D , save_image_2D : Function
// BASE_DIRECTORY_OUT : String
// LIST_COLOR : List of String

// ALGORITMA
package src;
import java.awt.Color;
import java.awt.Graphics2D;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.PrintWriter;

public class SaveFile {

    public static final String BASE_DIRECTORY_OUT = "../test/output/";
    public static final Color[] LIST_COLOR = {
        new Color(255 , 0 , 0),      // Merah
        new Color(0 , 255 , 0),      // Hijau
        new Color(255 , 255 , 0),    // Kuning
        new Color(0 , 0 , 255),      // Biru
        new Color(255 , 0 , 255),    // Magenta
        new Color(0 , 255 , 255),    // Cyan
        new Color(255 , 175 , 175),   // Pink Muda
        new Color(64 , 64 , 64),      // Abu-abu Gelap
        new Color(255 , 165 , 0),     // Oranye
        new Color(192 , 192 , 192),   // Abu-abu Terang
        new Color(128 , 128 , 128),   // Abu-abu
        new Color(255 , 105 , 180),    // Pink Cerah
        new Color(100 , 150 , 200),   // Biru Muda Custom
        new Color(150 , 100 , 200),   // Ungu Muda Custom
        new Color(210 , 105 , 30),    // Coklat
    }
}

```

```

        new Color(255 , 69 , 0),      // Merah Oranye
        new Color(154 , 205 , 50),    // Kuning Hijau
        new Color(255 , 215 , 0),      // Emas
        new Color(0 , 191 , 255),      // Biru Langit Cerah
        new Color(199 , 21 , 133),     // Merah Magenta
        new Color(72 , 209 , 204),     // Biru Hijau Medium
        new Color(218 , 112 , 214),    // Orchid
        new Color(188 , 143 , 143),    // Pink Abu-Abu
        new Color(255 , 140 , 0),      // Oranye Gelap
        new Color(173 , 216 , 230),    // Biru Muda
        new Color(244 , 164 , 96)      // Pasir
    };

    public static boolean save_text_2D(String file_name , char[][] board , long
time , long cnt , String arg) {
        String[] list_illegal_char = {"/" , "\"" , ":" , "*" , "?" , "\"" , "<"
, ">" , "|"};
        StringBuilder found_illegal = new StringBuilder();
        for (String e : list_illegal_char) {
            if (file_name.contains(e)) {
                if (found_illegal.length() > 0) {
                    found_illegal.append(" , ");
                }
                found_illegal.append(e);
            }
        }
        if (found_illegal.length() > 0) {
            System.out.println("Error : Gagal Menyimpan File Teks : Nama file
mengandung karakter yang tidak diizinkan (" + found_illegal + ").");
            return false;
        } else if (!file_name.endsWith(".txt")) {
            System.out.println("Error : Gagal Menyimpan File Teks : Format file
harus (.txt).");
            return false;
        } else {
            String file_path = BASE_DIRECTORY_OUT + "text/" + file_name;
            File file = new File(file_path);
            try (PrintWriter writer = new PrintWriter(file)) {

```

```

        if (Main.solved) {
            for (int i = 0 ; i < board.length ; i++) {
                for (int j = 0 ; j < board[0].length ; j++) {
                    writer.print(board[i][j]);
                }
                writer.println();
            }
        } else {
            writer.println(arg);
        }
        writer.println("Waktu pencarian : " + time + " ms.");
        writer.println("Banyak kasus yang ditinjau : " + cnt + "
kasus.");

        System.out.println("Success : File Teks Berhasil Disimpan Di :
" + file_path);
        return true;
    } catch (Exception ex) {
        System.out.println("Error : Gagal Menyimpan File Teks : " +
ex.getMessage());
        return false;
    }
}

}

}

public static boolean save_image_2D(String file_name , char[][] board_data)
{
    String[] list_illegal_char = {"/" , "\\" , ":" , "*" , "?" , "\"" , "<"
, ">" , "|"};
    StringBuilder found_illegal = new StringBuilder();
    for (String e : list_illegal_char) {
        if (file_name.contains(e)) {
            if (found_illegal.length() > 0) {
                found_illegal.append(" , ");
            }
            found_illegal.append(e);
        }
    }
    if (found_illegal.length() > 0) {

```

```

        System.out.println("Error : Gagal Menyimpan File Gambar : Nama file
mengandung karakter yang tidak diizinkan (" + found_illegal + ").");
        return false;
    } else if (!file_name.endsWith(".png")) {
        System.out.println("Error : Gagal Menyimpan File Gambar : Format
file harus (.png).");
        return false;
    } else {
        try {
            String file_path = BASE_DIRECTORY_OUT + "image/" + file_name;
            File file = new File(file_path);
            int r = board_data.length;
            int c = board_data[0].length;
            int size = 30;
            int w = c * size;
            int h = r * size;
            BufferedImage image = new BufferedImage(w , h ,
BufferedImage.TYPE_INT_RGB);
            Graphics2D g = image.createGraphics();
            g.setColor(Color.WHITE);
            g.fillRect(0 , 0 , w , h);
            for (int x = 0 ; x < r ; x++) {
                for (int y = 0 ; y < c ; y++) {
                    char ch = board_data[x][y];
                    Color color = Color.LIGHT_GRAY;
                    if (ch == '.') {
                        color = Color.WHITE;
                    }
                    else if ('A' <= ch && ch <= 'Z') {
                        int idx = ch - 'A';
                        if (idx >= 0 && idx < LIST_COLOR.length) {
                            color = LIST_COLOR[idx];
                        }
                    } else if (ch == ' ') {
                        color = Color.BLACK;
                    }
                    int xx = y*size;
                    int yy = x*size;

```

```
        g.setColor(color);
        g.fillRect(xx , yy , size , size);
        g.setColor(Color.BLACK);
        g.drawRect(xx , yy , size , size);
    }
}
g.dispose();
String format = "png";
javax.imageio.ImageIO.write(image , format , file);
System.out.println("Success : File Gambar Berhasil Disimpan Di
: " + file_path);
return true;
} catch (Exception ex) {
    System.out.println("Error : Gagal Menyimpan File Gambar : " +
ex.getMessage());
    return false;
}
}
}
```


BAB IV

TEST CASE

4.1. Test Case Tipe Default

1. Test Case Spesifikasi Default

```

itc1.txt x
test > input > itc1.txt
You, 6 days ago
1 5 5 7
2 DEFAULT
3 A
4 AA
5 B
6 BB
7 C
8 CC
9 D
10 DD
11 EE
12 EE
13 E
14 FF
15 FF
16 F
17 GGG
18 |

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : itc1.txt
-----
PROSES PENCARIAN SOLUSI DEFAULT
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI DEFAULT
-----
AGGGC
AABCC
EEBBF
EEDFF
EDDFF
Waktu pencarian : 15 ms.
Banyak kasus yang ditinjau : 1313850 kasus.
-----
SIMPAN HASIL KE FILE TEKS?
-----
WARNING : Input 'EXIT' To Force Exit!!
Apakah Hasil Mau Anda Save Ke File Teks (y/n) : 

```

2. Test Case Sukses Valid 1 Default

```

dtc1.txt x
test > input > dt
You, 1 hour
1 5 5 7
2 DEFAULT
3 MM
4 MM
5 M
6 AA
7 AA
8 Y
9 Y
10 Y
11 Y
12 ZZ
13 ZZ
14 LLL
15 L
16 B
17 SSS
18 |

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : dtc1.txt
-----
PROSES PENCARIAN SOLUSI DEFAULT
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI DEFAULT
-----
MMAAY
MMAAY
MLLLY
BLZZY
SSSZZ
Waktu pencarian : 13 ms.
Banyak kasus yang ditinjau : 2600 kasus.
-----

```

3. Test Case Gagal Valid Default

```

test > input > dt
You, 2 hour
1 6 6 7
2 DEFAULT
3 AAAA
4 AAA
5 BBB
6 BB
7 B
8 CCCC
9 CC
10 DDD
11 DD
12 E
13 EEE
14 FFF
15 GG
16 GG
17 |

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : dtc2.txt
-----
PROSES PENCARIAN SOLUSI DEFAULT
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI DEFAULT
-----
Tidak ada solusi yang ditemukan!
Waktu pencarian : 1009 ms.
Banyak kasus yang ditinjau : 142606368 kasus.
-----

```

4. Test Case Sukses Valid 2 Default

```

test > input > ≡ d
You, 2 hours
1 3 6 6
2 DEFAULT
3 A
4 A
5 BBB
6 B
7 CCC
8 D
9 DDD
10 EE
11 EE
12 F
13

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : dtc3.txt
-----
PROSES PENCARIAN SOLUSI DEFAULT
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI DEFAULT
-----
ABBBEE
ABFEED
CCCD DD
Waktu pencarian : 3 ms.
Banyak kasus yang ditinjau : 1314 kasus.
-----

```

5. Test Case Tidak Valid 1 Default

```

test > input > ≡ ed
You, 2 hours
1 3 6 7
2 DEFAULT
3 A
4 A
5 BBB
6 B
7 CCC
8 D
9 DDD
10 EE
11 EE
12 F
13

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : edtc1.txt
Error : Potongan puzzle ke-7 tidak ada di data.
Masukkan nama file input (pakai .txt) : edtc2.txt
Error : Terlalu banyak potongan. Seharusnya hanya 5 potongan.
Masukkan nama file input (pakai .txt) : edtc3.txt
Error : M bukan integer.
Masukkan nama file input (pakai .txt) : 

```

6. Test Case Tidak Valid 2 Default

```

test > input > ≡ ee
You, 3 minutes
1 3 6 5
2 DEFAULT
3 A
4 A
5 BBB
6 B
7 CCC
8 D
9 DDD
10 EE
11 EE
12 F
13

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : edtc1.txt
Error : Potongan puzzle ke-7 tidak ada di data.
Masukkan nama file input (pakai .txt) : edtc2.txt
Error : Terlalu banyak potongan. Seharusnya hanya 5 potongan.
Masukkan nama file input (pakai .txt) : edtc3.txt
Error : M bukan integer.
Masukkan nama file input (pakai .txt) : 

```

7. Test Case Tidak Valid 3 Default

```

test > input > ☰ e
    You, 2 hours
1  2 M 5
2  DEFAULT
3  A
4  A
5  BBB
6  B
7  CCC
8  D
9  DDD
10 EE
11 EE
12 F
13 |

-----
                    GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : edtc1.txt
Error : Potongan puzzle ke-7 tidak ada di data.
Masukkan nama file input (pakai .txt) : edtc2.txt
Error : Terlalu banyak potongan. Seharusnya hanya 5 potongan.
Masukkan nama file input (pakai .txt) : edtc3.txt
Error : M bukan integer.
Masukkan nama file input (pakai .txt) : 

```

4.2. Test Case Tipe Custom

1. Test Case Spesifikasi Custom

```

test > input > ☰ itc
    You, 6 hours
1  5 7 5
2  CUSTOM
3  ...X...
4  .XXXXX.
5  XXXXXXX
6  .XXXXX.
7  ...X...
8  A
9  AAA
10 BB
11 BBB
12 CCCC
13 | C
14 D
15 EEE
16 E
17 |

-----
                    GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : itc2.txt
-----
                PROSES PENCARIAN SOLUSI CUSTOM
-----
                    LOADING...
-----
                HASIL PENCARIAN SOLUSI CUSTOM
-----
                    A
                BBAAA
            BBBCCCC
        EEECD
            E
Waktu pencarian : 4 ms.
Banyak kasus yang ditinjau : 10290 kasus.
-----

```

2. Test Case Sukses Valid 1 Custom

```

test > input > You, 2 hour
1 4 4 4
2 CUSTOM
3 XXXX
4 X..X
5 X..X
6 XXXX
7 A
8 AA
9 B
10 BB
11 CCC
12 C
13 D
14 D
15

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : ctc1.txt
-----
PROSES PENCARIAN SOLUSI CUSTOM
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI CUSTOM
-----
CCCD
C D
A B
AABB
Waktu pencarian : 3 ms.
Banyak kasus yang ditinjau : 288 kasus.
-----

```

3. Test Case Gagal Valid Custom

```

test > input > You, 2 hour
1 6 6 7
2 CUSTOM
3 XXXXXX
4 XX..XX
5 X..XX
6 X..XX
7 XX..XX
8 XXXXXX
9 A
10 AAA
11 B
12 BBB
13 CCCC
14 C
15 C
16 D
17 D
18 D
19 E
20 E
21 E
22 FFF
23 F F
24 G
25 G
26

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : ctc2.txt
-----
PROSES PENCARIAN SOLUSI CUSTOM
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI CUSTOM
-----
Tidak ada solusi yang ditemukan!
Waktu pencarian : 7 ms.
Banyak kasus yang ditinjau : 299232 kasus.
-----

```

4. Test Case Sukses Valid 2 Custom

```

test > input > You, 2 hour
1 4 4 2
2 CUSTOM
3 XX..
4 .XX.
5 .XX.
6 ..XX
7 A
8 AAA
9 A
10 B
11 BB
12

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : ctc3.txt
-----
PROSES PENCARIAN SOLUSI CUSTOM
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI CUSTOM
-----
AA
AA
AB
BB
Waktu pencarian : 3 ms.
Banyak kasus yang ditinjau : 192 kasus.
-----

```

5. Test Case Tidak Valid 1 Custom

```

test > input > You, 2 hours
1 5 7 5
2 CUSTOM
3 ...X...
4 .XXXXX.
5 XXXXXMX
6 .XXXXX.
7 ...X...
8 A
9 AAA
10 BB
11 BBB
12 CCCC
13 C
14 D
15 EEE
16 E
17

-----
                        GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : ectc1.txt
Error : Board hanya boleh '.' atau 'X'.
Masukkan nama file input (pakai .txt) : ectc2.txt
Error : Panjang baris ke-2 bukan 7.
Masukkan nama file input (pakai .txt) : ectc3.txt
Error : Panjang baris ke-5 bukan 7.

```

6. Test Case Tidak Valid 2 Custom

```

test > input > You, 2 hours
1 5 7 5
2 CUSTOM
3 ...X...
4 .XXXXX.
5 XXXXXMX
6 .XXXXX.
7 ...X...
8 A
9 AAA
10 BB
11 BBB
12 CCCC
13 C
14 D
15 EEE
16 E
17

-----
                        GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : ectc1.txt
Error : Board hanya boleh '.' atau 'X'.
Masukkan nama file input (pakai .txt) : ectc2.txt
Error : Panjang baris ke-2 bukan 7.
Masukkan nama file input (pakai .txt) : ectc3.txt
Error : Panjang baris ke-5 bukan 7.

```

7. Test Case Tidak Valid 3 Custom

```

test > input > You, 3 minutes
1 5 7 5
2 CUSTOM
3 ...X...
4 .XXXXX.
5 XXXXXXXX
6 .XXXXX.
7 A
8 AAA
9 BB
10 BBB
11 CCCC
12 C
13 D
14 EEE
15 E
16

-----
                        GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : ectc1.txt
Error : Board hanya boleh '.' atau 'X'.
Masukkan nama file input (pakai .txt) : ectc2.txt
Error : Panjang baris ke-2 bukan 7.
Masukkan nama file input (pakai .txt) : ectc3.txt
Error : Panjang baris ke-5 bukan 7.

```

4.3. Test Case Tipe Pyramid

1. Test Case Spesifikasi Pyramid

```

test > input > 3 3 6
You, 6 days
1 3 3 6
2 PYRAMID
3 A
4 AA
5 BB
6 CC
7 DDD
8 E
9 EE
10 F
11

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : itc3.txt
-----
PROSES PENCARIAN SOLUSI PYRAMID
-----
LOADING...
-----
HASIL PENCARIAN SOLUSI PYRAMID
-----
Maaf, PYRAMID Solver Tidak Tersedia!
Tidak ada solusi yang ditemukan!
Waktu pencarian : 1 ms.
Banyak kasus yang ditinjau : 0 kasus.
-----

```

2. Test Case Tidak Valid 1 Pyramid

```

test > input > 3 4 6
You, 2 hour
1 3 4 6
2 PYRAMID
3 A
4 AA
5 BB
6 CC
7 DDD
8 E
9 EE
10 F
11

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : eptc1.txt
Error : Pada tipe puzzle PYRAMID nilai N dan M harus sama.
Masukkan nama file input (pakai .txt) : eptc2.txt
Error : N , M , P harus berupa bilangan bulat positif.
Masukkan nama file input (pakai .txt) : eptc3.txt
Error : P harus berupa bilangan bulat positif yang tidak lebih dari 26.
-----

```

3. Test Case Tidak Valid 2 Pyramid

```

test > input > -1 -1 6
You, 2 hour
1 -1 -1 6
2 PYRAMID
3 A
4 AA
5 BB
6 CC
7 DDD
8 E
9 EE
10 F
11

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : eptc1.txt
Error : Pada tipe puzzle PYRAMID nilai N dan M harus sama.
Masukkan nama file input (pakai .txt) : eptc2.txt
Error : N , M , P harus berupa bilangan bulat positif.
Masukkan nama file input (pakai .txt) : eptc3.txt
Error : P harus berupa bilangan bulat positif yang tidak lebih dari 26.
-----

```

4. Test Case Tidak Valid 3 Pyramid

```

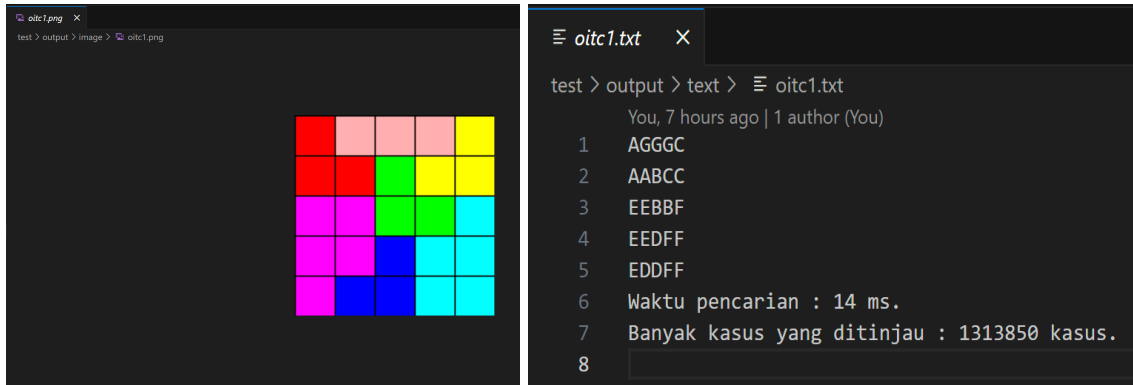
test > input > 3 3 36
You, 2 hour
1 3 3 36
2 PYRAMID
3 A
4 AA
5 BB
6 CC
7 DDD
8 E
9 EE
10 F
11

-----
GAME START
-----
WARNING : Input 'EXIT' To Force Exit!!
Masukkan nama file input (pakai .txt) : eptc1.txt
Error : Pada tipe puzzle PYRAMID nilai N dan M harus sama.
Masukkan nama file input (pakai .txt) : eptc2.txt
Error : N , M , P harus berupa bilangan bulat positif.
Masukkan nama file input (pakai .txt) : eptc3.txt
Error : P harus berupa bilangan bulat positif yang tidak lebih dari 26.
-----

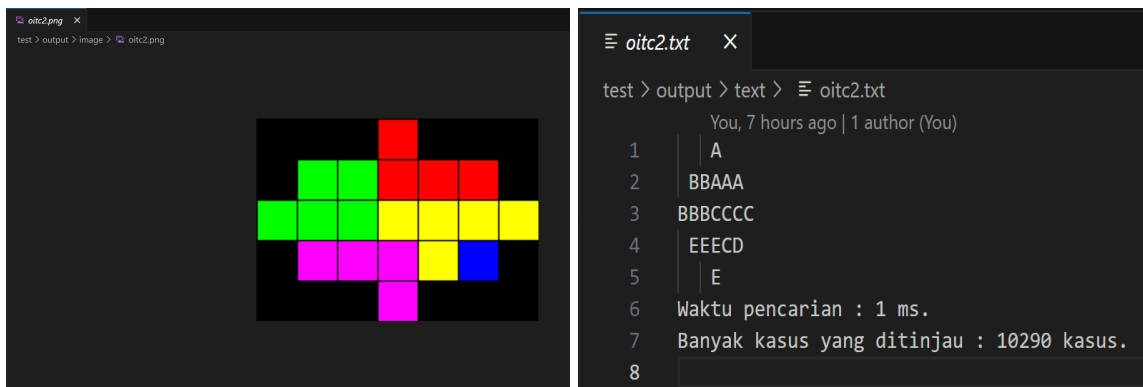
```

4.4. Test Case Lainnya

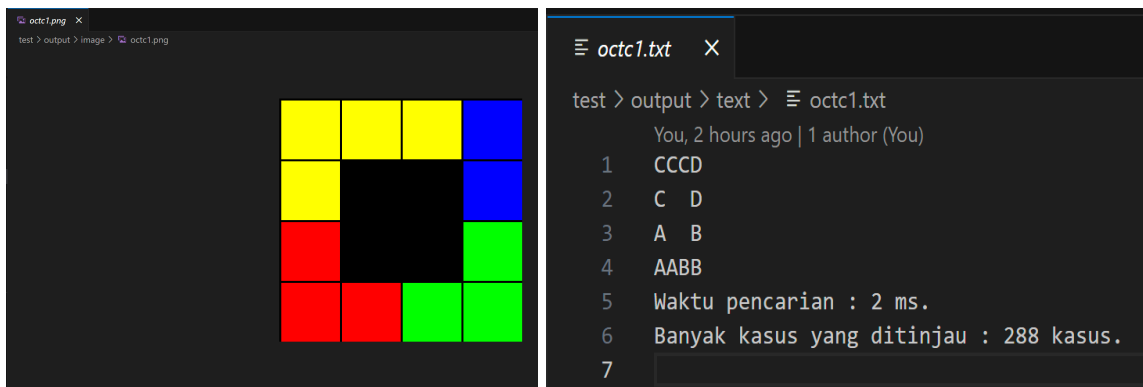
1. Test Case Output Teks & Gambar 1 (oitc1.txt & oitc1.png)



2. Test Case Output Teks & Gambar 2 (oitc2.txt & oitc2.png)



3. Test Case Output Teks & Gambar 3 (oetc1.txt & oetc1.png)



LAMPIRAN

1. Pranala Github : https://github.com/RNXFreeze/Tucil1_13523021
2. Tabel Checklist :

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 Graphical User Interface (GUI).		✓
6.	Program dapat menyimpan solusi dalam bentuk file gambar.	✓	
7.	Program dapat menyelesaikan kasus konfigurasi custom.	✓	
8.	Program dapat menyelesaikan kasus konfigurasi Piramida (3D).		✓
9.	Program dibuat oleh saya sendiri.	✓	