IF2211 Strategi Algoritma

Laporan Tugas Kecil 1

Fayyaz Akmal Lauda – 13524076

PROGRAM STUDI TEKNIK INFORMATIKA

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

JL. GANESA 10, BANDUNG 40132

2026

A. Deskripsi Singkat Masalah

Queens merupakan sebuah permainan teka-teki logika yang dimainkan pada papan berukuran N x N. Pada kondisi awal, papan ini dalam keadaan kosong dan terbagi menjadi beberapa area atau blok warna yang berbeda. Tujuan dari permainan ini adalah menempatkan bidak queen ke dalam kotak-kotak di papan sedemikian rupa sehingga memenuhi seluruh aturan keseimbangan.

Agar sebuah solusi dinyatakan valid, penempatan bidak queen harus mematuhi empat aturan (batasan) berikut:

1. Batasan Baris: Hanya boleh terdapat tepat satu queen pada setiap baris.
2. Batasan Kolom: Hanya boleh terdapat tepat satu queen pada setiap kolom.
3. Batasan Area Warna: Hanya boleh terdapat tepat satu queen pada setiap area warna yang sama.
4. Batasan Persinggungan (Adjacency): Dua buah queen tidak diperbolehkan diletakkan saling bersebelahan atau bersentuhan, baik secara vertikal, horizontal, maupun diagonal.

Program akan membaca masukan berupa berkas teks (.txt) yang mendeskripsikan konfigurasi papan awal. Papan direpresentasikan menggunakan karakter huruf alfabet (A - Z), di mana karakter yang sama menunjukkan area warna yang sama. Keluaran dari program adalah representasi visual papan pada terminal di mana huruf yang ditempati oleh bidak queen akan diubah menjadi huruf kecil, sementara huruf lain tetap kapital. Selain itu, program juga akan menampilkan informasi analitis berupa total konfigurasi (iterasi) yang ditinjau dan waktu eksekusi pencarian dalam satuan milisecond.

B. Desain Algoritma Brute Force

a. Representasi Ruang

Saya menggunakan array 1 dimensi bertipe integer bernama queens dengan panjang sebesar N untuk merepresentasikan posisi queens dan tidak menggunakan matriks 2 dimensi untuk mengurangi pemborosan memori dan memberlambat pencarian.

Cara membacanya:

- Indeks array mewakili nomor baris (dari 0 sampai N-1).
- Nilai array nya mewakili nomor kolom tempat queen diletakkan.

Misal pada papan 4 x 4, jika isi array queens = [0, 2, 1, 3], maka berarti queen berada di koordinat (0, 0), (1, 2) . (2, 1), dan (3, 3). Struktur data ini dipilih selain untuk menghemat pekerjaan juga karena secara otomatis menggugurkan kemungkinan ada lebih dari satu queen di baris yang sama.

b. Fase Generate

Di fase ini letak implementasi algoritma brute force terjadi, tepatnya di dalam metode generateTest(). Sesuai dengan yang diinginkan dalam spesifikasi tugas, program tidak menggunakan heuristik apapun untuk memangkas solusi yang salah. Program benar-benar menghasilkan kombinasi secara buta dan iteratif seperti counter bilangan berbasis N.

Cara kerjanya adalah sebagai berikut:

1. Program memulai pencarian dengan meletakkan semua queen di kolom paling kiri, sehingga array bernilai [0, 0, ..., 0].
2. Program akan mengevaluasi susunan tersebut, jika salah, queen pada baris paling bawah (indeks N-1) akan digeser satu langkah ke kanan (nilai kolomnya ditambah 1).
3. Jika queen di baris bawah sudah mentok di ujung kanan papan (mencapai nilai N-1), posisinya di-reset kembali ke 0, dan queen di baris tepat di atasnya akan ditambahkan nilainya.

4. Proses ini terus berulang menghasilkan deret kombinasi seperti [0, 0, 0, 0], [0, 0, 0, 1], [0, 0, 0, 2] dan seterusnya hingga mencapai kombinasi terakhir [N-1, N-1, ..., N-1].

Sifat brute force: Karena program ini menggunakan algoritma murni brute force, maka program tetap akan menghasilkan kombinasi yang jelas salah. Misalnya, kombinasi [0, 0, 0, 0] atau [0, 0, 1, 1] tetap dibangkitkan dan dikirim ke fase evaluasi, padahal sudah jelas melanggar aturan kolom. Karena setiap baris memiliki N kemungkinan letak kolom, maka ruang pencarian yang dihasilkan dan harus dilewati secara brute force adalah sebanyak N^N kemungkinan.

c. Fase Evaluasi

Setiap kali program selesai menghasilkan satu kombinasi posisi atau iterasi, kombinasi tersebut langsung dikirim ke metode isValid() untuk diuji. Karena aturan «satu baris satu queen» sudah dijamin dari bentuk struktur array 1 dimensi, maka hanya perlu memvalidasi 3 aturan lainnya:

1. Aturan kolom: Program menggunakan array boolean col. Saat membaca letak queen, program akan mengecek apakah nilai kolom tersebut sudah bernilai true (sudah ada yang menempati). Jika sudah, evaluasi langsung berhenti dan mengembalikan false (kombinasi ditolak).
2. Aturan area warna: Program membaca peta area pada matriks karakter warna di koordinat baris dan kolom queen saat ini. Dengan mengkonversi karakter A-Z menjadi 0-26, program menggunakan array boolean region untuk mengecek apakah area warna tersebut sudah pernah diisi oleh queen lain.
3. Aturan persinggungan: Program melakukan pengecekan secara vertikal, horizontal, dan diagonal. Dengan menggunakan nested loop, program membandingkan posisi queen di bari saat ini (r1) dengan posisi queen di baris-baris atasnya (r2 < r1). Jika selisih absolut antar baris ≤ 1 dan selisih absolut antar kolom juga ≤ 1, maka kedua queen saling bersentuhan yang mana kombinasi langusng ditolak.

Jika satu kombinasi lolos dari ketiga evaluasi ini tanpa mengembalikan false, maka variabel isSolved akan menjadi true, proses brute force dihentikan, dan solusi akhir dicetak ke layar dan bisa disimpan.

d. Kompleksitas Algoritma

Jumlah maksimal kandidat Solusi yang dibangkitkan oleh program adalah N^N. Pada setiap kandidat solusi, fungsi evaluasi melakukan pemeriksaan terhadap letak queen sebelumnya yang diimplementasikan menggunakan nested loop. Proses nested loop ini berjalan sekitar N(N - 1)/2 kali operasi, sehingga kompleksitas pengecekannya adalah O(N^2).

Dengan begitu, kompleksitas waktu dari algoritma brute force ini pada kasus terburuk (worst case) adalah O($N^2 \cdot N^N$). Sementara itu, untuk kompleksitas ruang, program setidaknya membutuhkan memori sebesar O(N^2) yang mayoritas digunakan untuk menyimpan peta awal area warna papan di dalam matriks 2 dimensi.

C. Source Code Program

```
import java.io.*;
import java.util.*;

public class SolverGameQueens {
    static int n;
    static char[][] warna;
    static int[] queens;
    static boolean isSolved = false;
    static long iterasi = 0;

    static long lastUpdateTime = 0;
    static final long UPDATE_INTERVAL = 300;

    public static void main(String[] args) {
        try (Scanner scanner = new Scanner(System.in)) {
            System.out.println("=== Program Solver Game Queens ===");
            System.out.print("Masukkan nama file input (misal:
test/test1.txt): ");
            String filename = scanner.nextLine();

            if (!loadValidasi(filename)) {
                System.out.println("File input tidak valid. Pastikan
formatnya benar.");
                return;
            }
            queens = new int[n];
            System.out.println("Memulai proses...\n");

            long StartTime = System.currentTimeMillis();
            lastUpdateTime = StartTime;
            generateTest();
            long EndTime = System.currentTimeMillis();

            if (isSolved) {
                System.out.println("=== Solusi ditemukan! ===\n");
                printBoard(queens);
            } else {
                System.out.println("=== Tidak ada solusi yang ditemukan.
===");
            }

            long executionTime = EndTime - StartTime;
            System.out.println("Waktu eksekusi: " + executionTime + " ms");
            System.out.println("Total iterasi: " + iterasi);

            if (isSolved) {
                System.out.print("Apakah Anda ingin menyimpan solusi ke
file? (y/n): ");
```

```
                String saveChoice =
scanner.nextLine().trim().toLowerCase();
                if (saveChoice.equals("y")) {
                    System.out.print("Masukkan nama file output:");
                    String outputFilename = scanner.nextLine();
                    save(outputFilename, executionTime);
                }
            }
        }
    }

    private static void generateTest() {
        boolean lanjut = true;

        while (lanjut && !isSolved) {
            iterasi++;
            long currentTime = System.currentTimeMillis();
            if (currentTime - lastUpdateTime >= UPDATE_INTERVAL) {
                System.out.println("Live update: Iterasi ke-" + iterasi +
": ");
                printBoard(queens);
                System.out.println("==============================");
                lastUpdateTime = currentTime;
            }

            if (isValid(queens)) {
                isSolved = true;
                break;
            }

            int i = n - 1;
            while (i >= 0 && queens[i] == n - 1) {
                queens[i] = 0;
                i--;
            }

            if (i >= 0) {
                queens[i]++;
            } else {
                lanjut = false;
            }
        }
    }

    private static boolean isValid(int[] state) {
        boolean[] col = new boolean[n];
        boolean[] region = new boolean[26];

        for (int r1 = 0; r1 < n; r1++) {
            int c1 = state[r1];
            char regionChar = warna[r1][c1];
```

```
            int regionIdx = regionChar - 'A';

            if (col[c1]) {
                return false;
            }
            col[c1] = true;
            if (region[regionIdx]) {
                return false;
            }
            region[regionIdx] = true;

            for (int r2 = 0; r2 < r1; r2++) {
                int c2 = state[r2];
                if (Math.abs(r1 - r2) <= 1 && Math.abs(c1 - c2) <= 1) {
                    return false;
                }
            }
        }
        return true;
    }

    // Utilitas
    private static boolean loadValidasi(String filename) {
        try (BufferedReader br = new BufferedReader(new
FileReader(filename))) {
            List<String> lines = new ArrayList<>();
            String line;
            while ((line = br.readLine()) != null) {
                line = line.trim();
                if (!line.isEmpty()) {
                    lines.add(line);
                }
            }

            if (lines.isEmpty()) {
                System.out.println("File input kosong.");
                return false;
            }
            n = lines.size();
            warna = new char[n][n];
            Set<Character> daerahValid = new HashSet<>();

            for (int i = 0; i < n; i++) {
                if (lines.get(i).length() != n) {
                    System.out.println("Panjang baris tidak sesuai dengan
ukuran papan.");
                    return false;
                }
                for (int j = 0; j < n; j++) {
                    char c = lines.get(i).charAt(j);
                    if (c < 'A' || c > 'Z') {
```

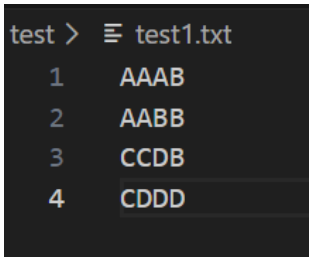
```
                        System.out.println("Error: Karakter harus antara A-
Z.");
                        return false;
                    }
                    warna[i][j] = c;
                    daerahValid.add(c);
                }
            }

            if (daerahValid.size() != n) {
                System.out.println("Jumlah daerah tidak sesuai dengan
ukuran papan.");
                return false;
            }
            return true;
        } catch (IOException e) {
            System.out.println("Gagal membaca file: " + e.getMessage());
            return false;
        }
    }

    private static void printBoard(int[] state) {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                if (state[i] == j) {
                    System.out.print(Character.toLowerCase(warna[i][j]) + "
");
                } else {
                    System.out.print(warna[i][j] + " ");
                }
            }
            System.out.println();
        }
    }

    private static void save(String filename, long executionTime) {
        try (PrintWriter pw = new PrintWriter(new FileWriter(filename))) {
            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    if (queens[i] == j) {
                        pw.print(Character.toLowerCase(warna[i][j]) + " ");
                    } else {
                        pw.print(warna[i][j] + " ");
                    }
                }
                pw.println();
            }

            pw.println();
            pw.println("Waktu eksekusi: " + executionTime + " ms");
            pw.println("Total iterasi: " + iterasi);
```

```
                System.out.println("Solusi berhasil disimpan ke " + filename);
            } catch (IOException e) {
                System.out.println("Gagal menyimpan file: " + e.getMessage());
            }
        }
}
```

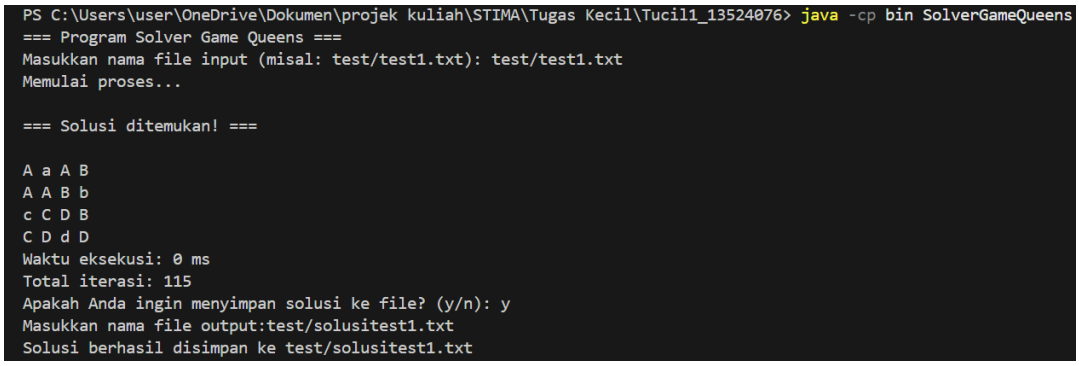
D. Hasil Uji Coba (Tangkapan Layar)

a. Test case 1

Input: test/test1.txt

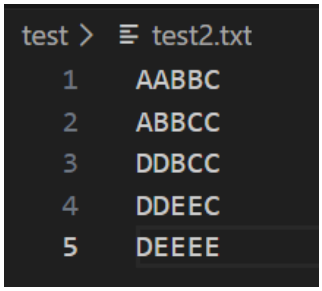


Output: test/solusitest1.txt



b. Test case 2

Input: test/test2.txt

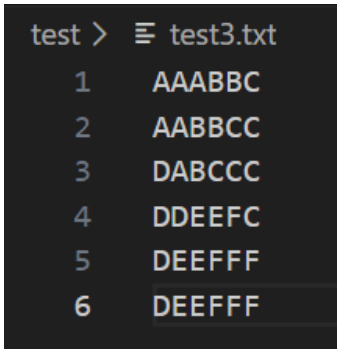


Output: test/solusitest2.txt

```
PS C:\Users\user\OneDrive\Dokumen\projek kuliah\STIMA\Tugas Kecil\Tucil1_13524076> java -cp bin SolverGameQueens
=== Program Solver Game Queens ===
Masukkan nama file input (misal: test/test1.txt): test/test2.txt
Memulai proses...

=== Solusi ditemukan! ===

a A B B C
A B b C C
D D B C c
D d E E C
D E E e E
Waktu eksekusi: 0 ms
Total iterasi: 359
Apakah Anda ingin menyimpan solusi ke file? (y/n): y
Masukkan nama file output:test/solusitest2.txt
Solusi berhasil disimpan ke test/solusitest2.txt
```

c. Test case 3

Input: test/test3.txt



Output: test/solusitest3.txt

```
PS C:\Users\user\OneDrive\Dokumen\projek kuliah\STIMA\Tugas Kecil\Tucil1_13524076> java -cp bin SolverGameQueens
=== Program Solver Game Queens ===
Masukkan nama file input (misal: test/test1.txt): test/test3.txt
Memulai proses...

=== Solusi ditemukan! ===

a A A B B C
A A B b C C
D A B C C c
D d E E F C
D E E F f F
D E e F F F
Waktu eksekusi: 2 ms
Total iterasi: 5031
Apakah Anda ingin menyimpan solusi ke file? (y/n): y
Masukkan nama file output:test/solusitest3.txt
Solusi berhasil disimpan ke test/solusitest3.txt
```

d. Test case 4

Input: test/test4.txt

test > test4.txt

```
AABBCCD
ABBCCDD
ABBCCDD
EEBFFDD
EEFFGGD
EEFFGGG
EEFFGGG
```

Output: test/solusitest4.txt

```
PS C:\Users\user\OneDrive\Dokumen\projek kuliah\STIMA\Tugas Kecil\Tucil1_13524076> java -cp bin SolverGameQueens
=== Program Solver Game Queens ===
Masukkan nama file input (misal: test/test1.txt): test/test4.txt
Memulai proses...

=== Solusi ditemukan! ===

a A B B C C D
A B b C C D D
A B B C c D D
E e B F F D D
E E F F G G d
E E F f G G G
E E F F G g G
Waktu eksekusi: 7 ms
Total iterasi: 43882
Apakah Anda ingin menyimpan solusi ke file? (y/n): y
Masukkan nama file output:test/solusitest4.txt
Solusi berhasil disimpan ke test/solusitest4.txt
```

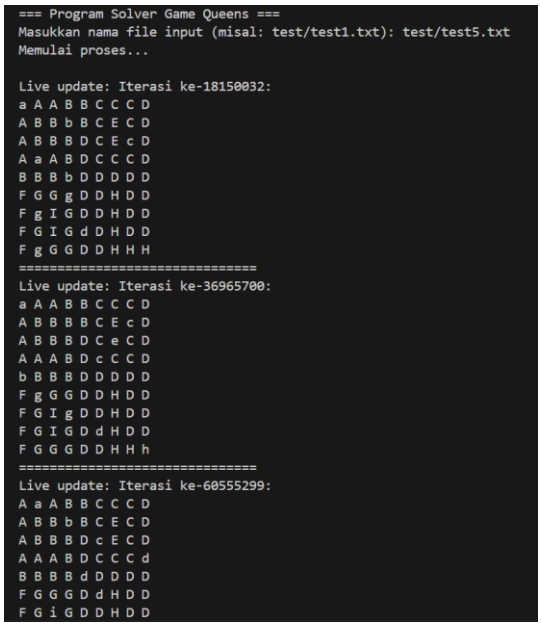
e. Test case 5

Input: test/test5.txt

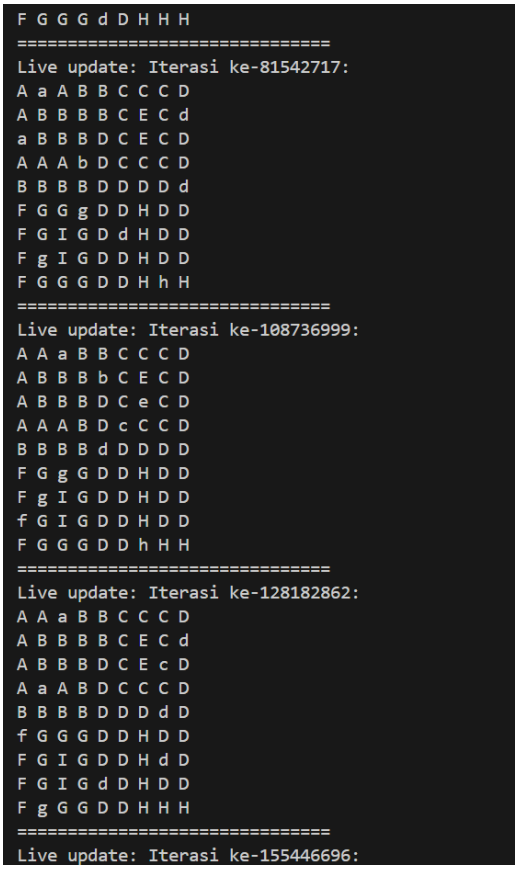
test > test5.txt

```
AAABBCCCD
ABBBBCECD
ABBBDCECD
AAABDCCCD
BBBBDDDDD
FGGGDDHDD
FGIGDDHDD
FGIGDDHDD
FGGGDDHHH
```

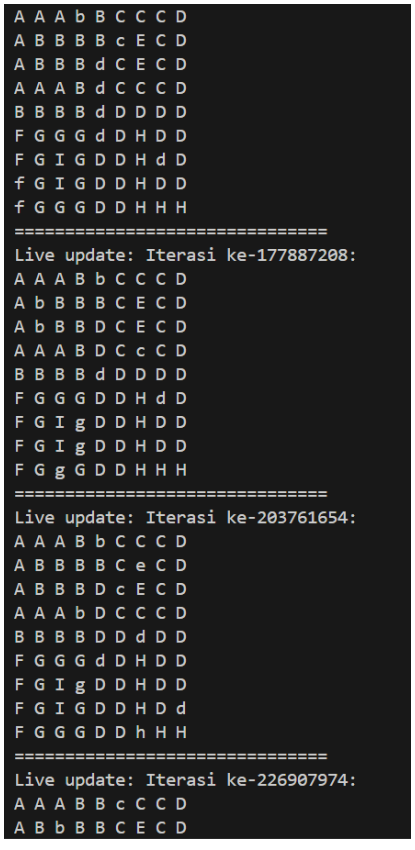
Output: test/solusitest5.txt



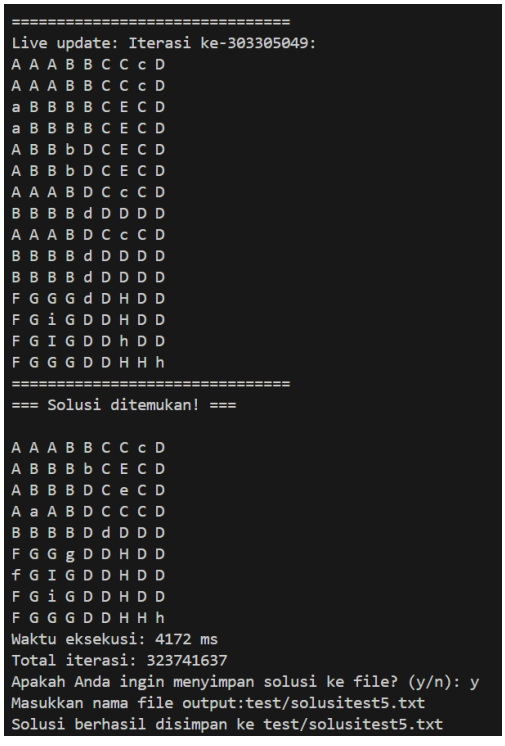
```
=== Program Solver Game Queens ===
Masukkan nama file input (misal: test/test1.txt): test/test5.txt
Memulai proses...

Live update: Iterasi ke-18150032:
a A A B B C C C D
A B B b B C E C D
A B B B D C E c D
A a A B D C C C D
B B B b D D D D D
F G G g D D H D D
F g I G D D H D D
F G I G d D H D D
F g G G D D H H H
==============================
Live update: Iterasi ke-36965700:
a A A B B C C C D
A B B B B C E c D
A B B B D C e C D
A A A B D c C C D
b B B B D D D D D
F g G G D D H D D
F G I g D D H D D
F G I G D d H D D
F G G G D D H H h
==============================
Live update: Iterasi ke-60555299:
A a A B B C C C D
A B B b B C E C D
A B B B D c E C D
A A A B D C C C d
B B B B d D D D D
F G G G D d H D D
F G i G D D H D D
```



```
F G G G d D H H H
==============================
Live update: Iterasi ke-81542717:
A a A B B C C C D
A B B B B C E C d
a B B B D C E C D
A A A b D C C C D
B B B B D D D D d
F G G g D D H D D
F G I G D d H D D
F g I G D D H D D
F G G G D D H h H
==============================
Live update: Iterasi ke-108736999:
A A a B B C C C D
A B B B b C E C D
A B B B D C e C D
A A A B D c C C D
B B B B d D D D D
F G g G D D H D D
F g I G D D H D D
f G I G D D H D D
F G G G D D h H H
==============================
Live update: Iterasi ke-128182862:
A A a B B C C C D
A B B B B C E C d
A B B B D C E c D
A a A B D C C C D
B B B B D D D d D
f G G G D D H D D
F G I G D D H d D
F G I G d D H D D
F g G G D D H H H
==============================
Live update: Iterasi ke-155446696:
```



```
A A A b B C C C D
A B B B B c E C D
A B B B d C E C D
A A A B d C C C D
B B B B d D D D D
F G G G d D H D D
F G I G D D H d D
f G I G D D H D D
f G G G D D H H H
==============================
Live update: Iterasi ke-177887208:
A A A B b C C C D
A b B B B C E C D
A b B B D C E C D
A A A B D C c C D
B B B B d D D D D
F G G G D D H d D
F G I g D D H D D
F G I g D D H D D
F G g G D D H H H
==============================
Live update: Iterasi ke-203761654:
A A A B b C C C D
A B B B B C e C D
A B B B D c E C D
A A A b D C C C D
B B B B D D d D D
F G G G d D H D D
F G I g D D H D D
F G I G D D H D d
F G G G D D h H H
==============================
Live update: Iterasi ke-226907974:
A A A B B c C C D
A B b B B C E C D
```



```
==============================
Live update: Iterasi ke-303305049:
A A A B B C C c D
A A A B B C C c D
a B B B B C E C D
a B B B B C E C D
A B B b D C E C D
A B B b D C E C D
A A A B D C c C D
B B B B d D D D D
A A A B D C c C D
B B B B d D D D D
B B B B d D D D D
F G G G d D H D D
F G i G D D H D D
F G I G D D h D D
F G G G D D H H h
==============================
=== Solusi ditemukan! ===

A A A B B C C c D
A B B B b C E C D
A B B B D C e C D
A a A B D C C C D
B B B B D d D D D
F G G g D D H D D
f G I G D D H D D
F G i G D D H D D
F G G G D D H H h
Waktu eksekusi: 4172 ms
Total iterasi: 323741637
Apakah Anda ingin menyimpan solusi ke file? (y/n): y
Masukkan nama file output:test/solusitest5.txt
Solusi berhasil disimpan ke test/solusitest5.txt
```

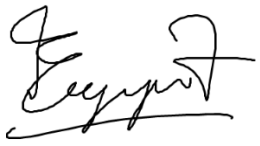
E. Tautan dan Lampiran

Link repository Github: https://github.com/pepayaz/Tucil1_13524076

No	Poin	Ya	Tidak
1	Program berhasil di kompilasi tanpa kesalahan	✓	
2	Program berhasil di jalankan	✓	
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		✓

F. Pernyataan

Tugas ini disusun sepenuhnya tanpa bantuan kecerdasan buatan (Generative AI), melainkan hasil pemikiran dan analisis mandiri.



Fayyaz Akmal Lauda