Laporan Tugas Kecil 1

IF2211 Strategi Algoritma
Penyelesaian Permainan Queens LinkedIn
Semester 2 Tahun Ajaran 2025/2026

Neswa Eka Anggara

13524136

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

1. Algoritma Brute Force

```
while (true) {

        casesChecked++;

        if (casesChecked % 100000 == 0) {
          …..
        }

        if (isValidCombination(index)) {
            applyCombination(index);
            endTime = System.nanoTime();

            if (casesChecked < 100000) {
              ….
            }
        }

        int pos = R - 1;
        while (pos >= 0) {
            index[pos]++;
            if (index[pos] < limits[pos]) {
                break;
            } else {
                index[pos] = 0;
                pos--;
            }
        }

        if (pos < 0) {
            break;
        }
    }
```

Program melakukan iterasi secara terus menerus sambil mengecek apakah sudah valid atau belum dengan bantuan isValidCombination

```
private boolean isValidCombination(int[] index) {

      boolean[] usedRow = new boolean[n];
      boolean[] usedCol = new boolean[n];

      List<int[]> queens = new ArrayList<>();
```

```
    for (int i = 0; i < index.length; i++) {

        List<int[]> cells = regionCells.get(regionList.get(i));
        int[] cell = cells.get(index[i]);

        int row = cell[0];
        int col = cell[1];

        if (usedRow[row] || usedCol[col]) {
            return false;
        }

        usedRow[row] = true;
        usedCol[col] = true;

        queens.add(cell);
    }

    for (int i = 0; i < queens.size(); i++) {
        for (int j = i + 1; j < queens.size(); j++) {

            int r1 = queens.get(i)[0];
            int c1 = queens.get(i)[1];

            int r2 = queens.get(j)[0];
            int c2 = queens.get(j)[1];

            if (Math.abs(r1 - r2) == 1 &&
                Math.abs(c1 - c2) == 1) {
                return false;
            }
        }
    }

    return true;
}
```

Program mengecek satu persatu di setiap iterasi untuk memastikan constraint terpenuhi:

1. Satu row hanya ada satu queen
2. Satu column hanya ada satu queen
3. Satu region hanya ada satu queen
4. Tidak ada queen yang berdempetan (juga secara diagonal)

2. Source Code

LoadBoard.java

```
package main.java.stima.modules;

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

public class LoadBoard {

    private char[][] board;
    private int size;
    private Set<Character> regions;

    public LoadBoard(String filePath) throws IOException {
        loadTXT(filePath);
        validateStructure();
    }

    public char[][] getBoard() {
        return board;
    }

    private void loadTXT(String filePath) throws IOException {

        List<String> lines = new ArrayList<>();
        BufferedReader br = new BufferedReader(new FileReader(filePath));
        String line;

        while ((line = br.readLine()) != null) {
            line = line.trim();
            if (!line.isEmpty()) {
                lines.add(line);
            }
        }
        br.close();

        if (lines.isEmpty()) {
            throw new IllegalArgumentException("File kosong.");
        }

        size = lines.size();

        if (size > 26) {
            throw new IllegalArgumentException("Ukuran maksimum adalah 26x26.");
        }

        board = new char[size][size];
        regions = new HashSet<>();

        for (int i = 0; i < size; i++) {

            if (lines.get(i).length() != size) {
                throw new IllegalArgumentException("Board harus berbentuk matriks n x
n.");
            }

            for (int j = 0; j < size; j++) {
                char c = lines.get(i).charAt(j);

                if (!Character.isUpperCase(c)) {
                    throw new IllegalArgumentException("Board hanya boleh berisi huruf A-Z
(kapital).");
                }

                board[i][j] = c;
                regions.add(c);
            }
        }

        if (regions.size() > size){
            throw new IllegalArgumentException("Jumlah region melebihi ukuran.");
        }
    }

    private void validateStructure() {
        if (size <= 3 || size > 26) {
            throw new IllegalArgumentException("Ukuran board harus 4 <= n <= 26.");
        }
    }

    public void printBoard() {
        for (int i = 0; i < size; i++) {
            for (int j = 0; j < size; j++) {
                System.out.print(board[i][j]);
            }
            System.out.println();
        }
    }
}
```

QueensSolver.java

```
package main.java.stima.modules;

import java.util.*;

public class QueensSolver {
```

```
private char[][] board;
private char[][] region;
private int n;

private List<Character> regionList;
private Map<Character, List<int[]>> regionCells;

// Statistik
private long casesChecked = 0;
private long startTime;
private long endTime;

public QueensSolver(char[][] inputBoard) {

    n = inputBoard.length;

    board = new char[n][n];
    region = new char[n][n];

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {
            board[i][j] = inputBoard[i][j];
            region[i][j] = inputBoard[i][j];
        }
    }

    buildRegionData();
}

private void buildRegionData() {

    regionCells = new HashMap<>();

    for (int i = 0; i < n; i++) {
        for (int j = 0; j < n; j++) {

            char r = region[i][j];

            regionCells.putIfAbsent(r, new ArrayList<>());
            regionCells.get(r).add(new int[]{i, j});
        }
    }

    regionList = new ArrayList<>(regionCells.keySet());
    Collections.sort(regionList);
}

public boolean solve() {

    startTime = System.nanoTime();
```

```
int R = regionList.size();

int[] index = new int[R];
int[] limits = new int[R];

for (int i = 0; i < R; i++) {
    limits[i] = regionCells.get(regionList.get(i)).size();
}

while (true) {

    casesChecked++;

    if (casesChecked % 100000 == 0) {
        System.out.println((""));
        System.out.println("Live Update:");
        System.out.println("Iterasi ke: " + casesChecked);
        applyCombination(index);
        printBoard();
    }

    if (isValidCombination(index)) {
        applyCombination(index);
        endTime = System.nanoTime();

        if (casesChecked < 100000) {
            System.out.println("");
            System.out.println("Live Update (Terakhir)");
            System.out.println("Iterasi terakhir: " + casesChecked);
            printBoard();
        }

        System.out.println("");
        System.out.println("Hasil akhir:");
        System.out.println("");
        printBoard();
        System.out.println();
        printStats(true);
        return true;
    }

    int pos = R - 1;
    while (pos >= 0) {
        index[pos]++;
        if (index[pos] < limits[pos]) {
            break;
        } else {
            index[pos] = 0;
            pos--;
            }
        }

        if (pos < 0) {
            break;
        }
    }

    endTime = System.nanoTime();
    printStats(false);
    return false;
}

private boolean isValidCombination(int[] index) {

    boolean[] usedRow = new boolean[n];
    boolean[] usedCol = new boolean[n];

    List<int[]> queens = new ArrayList<>();

    for (int i = 0; i < index.length; i++) {

        List<int[]> cells = regionCells.get(regionList.get(i));
        int[] cell = cells.get(index[i]);

        int row = cell[0];
        int col = cell[1];

        if (usedRow[row] || usedCol[col]) {
            return false;
        }

        usedRow[row] = true;
        usedCol[col] = true;

        queens.add(cell);
    }

    for (int i = 0; i < queens.size(); i++) {
        for (int j = i + 1; j < queens.size(); j++) {

            int r1 = queens.get(i)[0];
            int c1 = queens.get(i)[1];

            int r2 = queens.get(j)[0];
            int c2 = queens.get(j)[1];

            if (Math.abs(r1 - r2) == 1 &&
                Math.abs(c1 - c2) == 1) {
                return false;
                }
            }
        }

        return true;
    }

    private void applyCombination(int[] index) {

        for (int i = 0; i < n; i++)
            for (int j = 0; j < n; j++)
                board[i][j] = region[i][j];

        for (int i = 0; i < index.length; i++) {

            List<int[]> cells = regionCells.get(regionList.get(i));
            int[] cell = cells.get(index[i]);

            board[cell[0]][cell[1]] = '#';
        }
    }

    private void printStats(boolean found) {

        double durationMs = (endTime - startTime) / 1_000_000.0;

        System.out.print("Status : ");
        if (found){
            System.out.println("Solusi ditemukan");
        } else{
            System.out.println("Solusi tidak ditemukan");
        }

        System.out.printf("Waktu pencarian  : %.3f ms\n", durationMs);
        System.out.println("Kasus ditinjau   : " + casesChecked + " kasus");
    }

    public void printBoard() {
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n; j++) {
                System.out.print(board[i][j]);
            }
            System.out.println();
        }
    }
}
```

SaveBoard.java

```
package main.java.stima.modules;
```

```
import java.io.BufferedWriter;
import java.io.FileWriter;
import java.io.IOException;

public class SaveBoard {

    public static void saveToFile(char[][] board, String filePath) {

        try (BufferedWriter writer = new BufferedWriter(new FileWriter(filePath))) {

            int n = board.length;

            for (int i = 0; i < n; i++) {
                for (int j = 0; j < n; j++) {
                    writer.write(board[i][j]);
                }
                writer.newLine();
            }

            System.out.println("Board berhasil disimpan ke: " + filePath);

        } catch (IOException e) {
            System.out.println("Terjadi kesalahan saat menyimpan file.");
            e.printStackTrace();
        }
    }
}
```

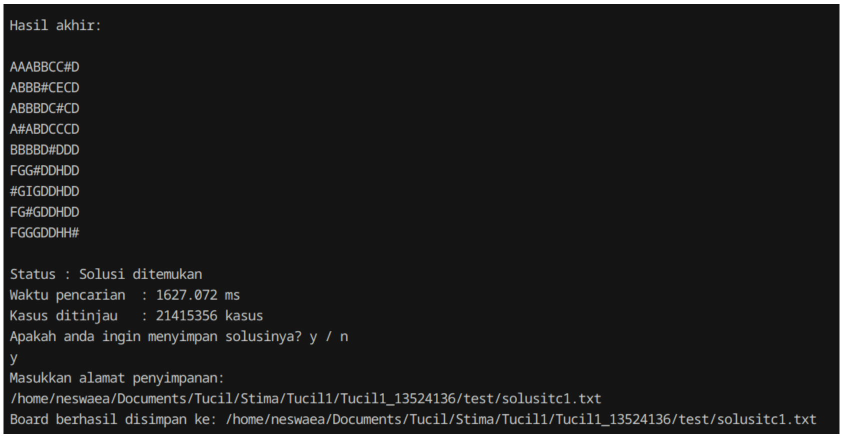
App.java

```
package stima;

import main.java.stima.modules.*;
import java.util.Scanner;

public class App
{
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.println("SELAMAT DATANG DI QUEENSSOLVER!");
        System.out.println("Masukkan alamat file txt: ");
        String fileLocation = scan.nextLine();
        System.out.println("");
        try {
            LoadBoard load = new LoadBoard(fileLocation);

            System.out.println("Board berhasil dimuat.");
            System.out.println("");
            load.printBoard();
            System.out.println("");

            System.out.println("Mencari solusi...");
            System.out.println("");

            char[][] board = load.getBoard();

            QueensSolver solver = new QueensSolver(board);

            if (solver.solve()){
                System.out.println("Apakah anda ingin menyimpan solusinya? y / n");
                String ans = scan.nextLine();
                if (ans.equals("y")){
                    System.out.println("Masukkan alamat penyimpanan: ");
                    String saveLoc = scan.nextLine();
                    SaveBoard.saveToFile(board, saveLoc);
                }
            }

        } catch (Exception e) {
            System.out.println("Error: " + e.getMessage());
        }
    }
}
```

3. Test Case

3.1 Test Case 1

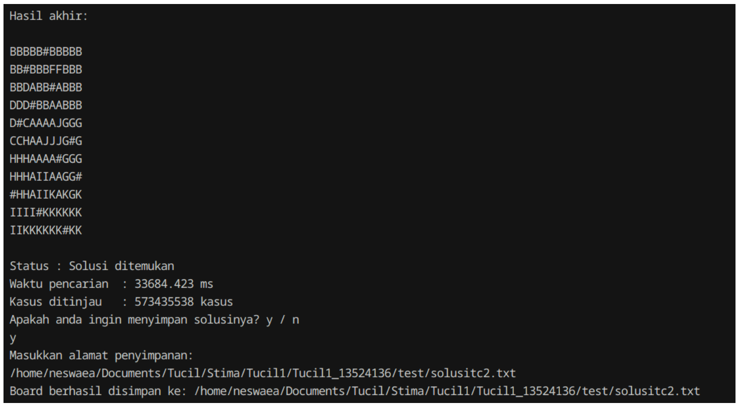
```
AAABBCCCD
ABBBBCECD
ABBBDCECD
AAABDCCCD
BBBBDDDDD
FGGGDDHDD
FGIGDDHDD
FGIGDDHDD
FGGGDDHHH
```



```
Hasil akhir:

AAABBCC#D
ABBB#CECD
ABBBDC#CD
A#ABDCCCD
BBBBD#DDD
FGG#DDHDD
#GIGDDHDD
FG#GDDHDD
FGGGDDHH#

Status : Solusi ditemukan
Waktu pencarian  : 1627.072 ms
Kasus ditinjau   : 21415356 kasus
Apakah anda ingin menyimpan solusinya? y / n
y
Masukkan alamat penyimpanan:
/home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc1.txt
Board berhasil disimpan ke: /home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc1.txt
```

3.2 Test Case 2

```
BBBBBBBBBBB
BBDBBBFFBBB
BBDABBFABBB
DDDABBAABBB
DCCAAAAJGGG
CCHAAJJJGEG
HHHAAAAJGGG
HHHAIIAAGGG
HHHAIIKAKGK
IIIIIKKKKKK
IIKKKKKKKKK
```

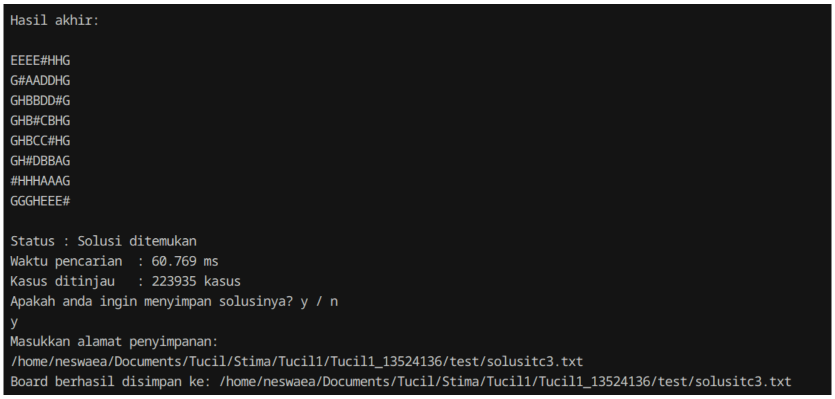


```
Hasil akhir:

BBBBB#BBBBB
BB#BBBFFBBB
BBDABB#ABBB
DDD#BBAABBB
D#CAAAAJGGG
CCHAAJJJG#G
HHHAAAA#GGG
HHHAIIAAGG#
#HHAIIKAKGK
IIII#KKKKKK
IIKKKKKK#KK

Status : Solusi ditemukan
Waktu pencarian  : 33684.423 ms
Kasus ditinjau   : 573435538 kasus
Apakah anda ingin menyimpan solusinya? y / n
y
Masukkan alamat penyimpanan:
/home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc2.txt
Board berhasil disimpan ke: /home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc2.txt
```

3.2 Test Case 3

```
EEEEFHHG
```

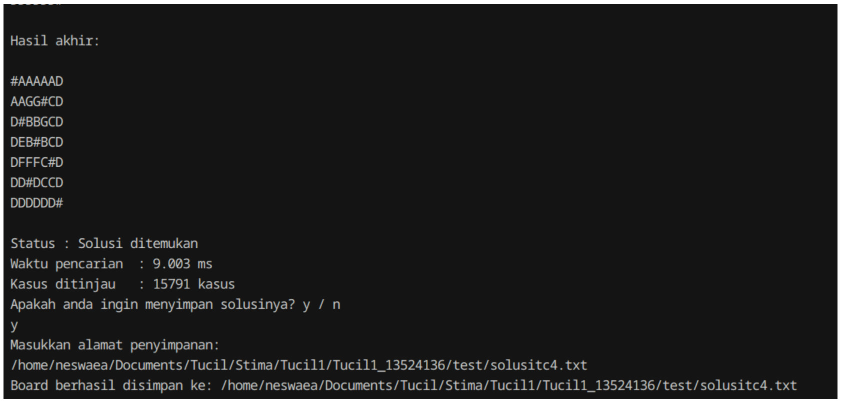
```
GAAADDHG
GHBBDDHG
GHBCCBHG
GHBCCBHG
GHDDBBAG
GHHHAAAG
GGGHEEEE
```



```
Hasil akhir:

EEEE#HHG
G#AADDHG
GHBBDD#G
GHB#CBHG
GHBCC#HG
GH#DBBAG
#HHHAAAG
GGGHEEE#

Status : Solusi ditemukan
Waktu pencarian  : 60.769 ms
Kasus ditinjau   : 223935 kasus
Apakah anda ingin menyimpan solusinya? y / n
y
Masukkan alamat penyimpanan:
/home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc3.txt
Board berhasil disimpan ke: /home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc3.txt
```

3.2 Test Case 4

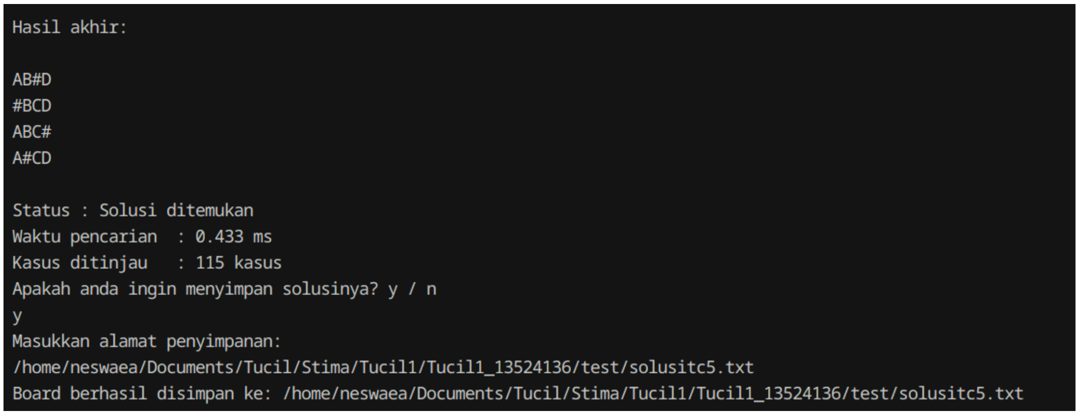
```
AAAAAAD
AAGGGCD
DEBBGCD
DEBBBCD
DFFFCCD
DDFDCCD
DDDDDDD
```



```
Hasil akhir:

#AAAAAD
AAGG#CD
D#BBGCD
DEB#BCD
DFFFC#D
DD#DCCD
DDDDDD#

Status : Solusi ditemukan
Waktu pencarian  : 9.003 ms
Kasus ditinjau   : 15791 kasus
Apakah anda ingin menyimpan solusinya? y / n
y
Masukkan alamat penyimpanan:
/home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc4.txt
Board berhasil disimpan ke: /home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc4.txt
```

3.2 Test Case 5

```
ABCD
ABCD
ABCD
ABCD
```



```
Hasil akhir:

AB#D
#BCD
ABC#
A#CD

Status : Solusi ditemukan
Waktu pencarian  : 0.433 ms
Kasus ditinjau   : 115 kasus
Apakah anda ingin menyimpan solusinya? y / n
y
Masukkan alamat penyimpanan:
/home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc5.txt
Board berhasil disimpan ke: /home/neswaea/Documents/Tucil/Stima/Tucil1/Tucil1_13524136/test/solusitc5.txt
```

4. Repository GitHub

https://github.com/NeswaEA/Tucil1_13524136

No	Poin	Ya	Tidak
1	Program berhasil di kompilasi tanpa kesalahan	✅	
2	Program berhasi 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 GUI		✅
6	Program dapat menyimpan solusi dalam bentuk file gambar		✅

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

13524136