

Laporan Tugas Kecil 1

IF2211 Strategi Algoritma

Semester II Tahun 2025/2026



Disusun oleh:

Mikhael Andrian Yonatan / 13524051

Laboratorium Ilmu dan Rekayasa Komputasi
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

1. Desain dan Implementasi

a. Struktur Folder

```
TUCIL1_13524051/
|
|__ bin/
|   |__ opencv_world4120.dll
|   |__ QueensGame.exe
|   |__ QueensGame.pdb
|
|__ doc/
|
|__ src/
|   |__ fileproc.cpp
|   |__ fileproc.h
|   |__ imgproc.cpp
|   |__ imgproc.h
|   |__ main.cpp
|   |__ queen.cpp
|   |__ queen.h
|   |__ queen.png
|
|__ test/
|
|__ CMakeLists.txt
|__ README.md
```

Secara garis besar,

- Folder `src` berisi source code program dalam bahasa C++.
- Folder `bin` berisi *executable file* hasil kompilasi beserta *library* pendukung untuk menjalankan program.
- Folder `test` berisi file data *input* serta data *output* (solusi) yang dihasilkan oleh program.
- Folder `doc` berisi laporan tugas kecil dalam bentuk dokumen PDF.
- File `CMakeLists.txt` berisi konfigurasi sistem *build* untuk mengompilasi kode sumber C++.

b. Algoritma Brute Force Yang Diterapkan

Pada *Queens Game*, apabila terdapat papan permainan berukuran $N \times N$, maka akan memiliki solusi yang mengandung N buah *Queens* juga. Algoritma *brute force* yang dirancang akan terus dilakukan selama $foundQueens < N$, dengan algoritmanya sebagai berikut,

1. Periksa sel pertama (pojok kiri atas). Jika *valid*, maka tempatkan *Queen* lalu ingat posisi dan warna selnya. Lakukan juga penambahan pada *foundQueens*.

2. Lanjut ke sel berikutnya dan periksa. Jika *valid*, maka lakukan hal yang sama seperti sebelumnya. Jika tidak *valid*, maka lanjut ke sel berikutnya dan periksa lagi sampai sel pojok kanan bawah.
3. Jika tidak ada sel yang valid sampai sel paling akhir (pojok kanan bawah), maka kembali ke posisi *Queen* yang terakhir kali ditempatkan untuk dicari sel *valid* yang baru, lakukan pengurangan pada *foundQueens*.
4. Ulangi langkah 2 dan 3.

Algoritma yang diterapkan sangatlah “*brute force* garis keras” tanpa heuristik karena akan terus mencoba semua kemungkinan penempatan *Queen*. Contoh yang paling ekstrem adalah ketika memang tidak ada solusi, program seharusnya dapat berhenti lebih cepat ketika suatu baris tidak dapat ditempati *Queen*. Namun, program justru akan terus mencoba sampai sel terakhir pada papan permainan, walaupun sia-sia.

c. Implementasi Kode Program

1. fileproc.cpp

```
File ini mengandung fungsi dan/atau prosedur yang berkaitan dengan pemrosesan file berekstensi txt.
```

```
Membaca file .txt menjadi vector<string>
vector<string> loadBoardTxt(string filename) {
    INPUT : Nama file input (.txt)
    OUTPUT: Papan permainan dalam vector<string>
    CARA KERJA:
    1. Membuka file dengan nama yang sudah diinput
    2. Jika file ada, selanjutnya baca setiap baris sebagai
       string dan masukkan ke dalam vector
    3. Tentukan ukuran baris dan kolom dari vector
    4. Apabila tidak persegi atau ada char selain capital
       alphabet, maka input tidak valid
}
```

```
Membaca vector<string> dan menuliskannya menjadi file .txt
void saveBoardTxt(vector<string>& board, string filename) {
    INPUT : Papan permainan dan nama file output
    OUTPUT: File (.txt)
    CARA KERJA:
    1. Membuat file baru (.txt) sesuai dengan nama yang
       diinginkan.
    2. Membaca setiap baris dan kolom pada vector<string> (papan
       permainan) dan menuliskannya ke dalam file.
}
```

2. imgproc.cpp

File ini mengandung fungsi dan/atau prosedur yang berkaitan dengan pemrosesan file berekstensi png atau jpg.

```
struct Color {  
    int r, g, b;  
};
```

Membaca file .png/.jpg menjadi vector<string>
vector<string> processImage(string imagePath, int N){

INPUT : Nama file input (.png/.jpg) dan ukuran papan
permainan

OUTPUT: Papan permainan dalam vector<string>

CARA KERJA:

1. Membuka file dengan nama yang sudah diinput
2. Jika file ada, dilakukan perhitungan untuk mengestimasi
ukuran tiap sel (piksel) pada papan permainan
3. Pada tiap sel di papan permainan, baca titik (piksel)
tengahnya dan ambil informasi warnanya
4. Lakukan perhitungan pada informasi warna yang baru
diperoleh dengan warna yang sudah pernah ditemui
sebelumnya. Apabila lebih kecil dari threshold yang
ditentukan, maka dianggap sebagai warna yang sama.
5. Masukkan warna sebagai char ke dalam vector<string> (papan
permainan).

Membaca vector<string> dan membuatnya menjadi file .png

```
void saveBoardPng(vector<string>& board, vector<pair<int, char>>&  
ans, string filename){
```

INPUT : Nama file output (.png), papan permainan, informasi
posisi dan warna dari sel yang ditempati Queen

OUTPUT: File (.png)

CARA KERJA:

1. Membuat kanvas putih berukuran N x piksel/sel (ditentukan)
2. Pada tiap sel di papan permainan. Jika bukan Queen (#),
ambil charnya dan lakukan perhitungan untuk mendapatkan
(r, g, b), lalu beri warna pada sel dengan acuan titik
pojok kiri atas dan pojok kanan bawah dari suatu sel.
3. Jika Queen (#), ambil infromasi warnanya dari ans yang
mengandung informasi posisi dan warna (char sebelum diisi
Queen). Warnai sel nya, lalu timpa dengan icon Queen.

```
}
```

3. queen.cpp

File ini mengandung fungsi dan/atau prosedur yang berkaitan dengan logika utama penyelesaian masalah.

Menentukan apakah suatu sel valid untuk ditempati Queen

```
bool isValid(int x, int y, char color, int N, vector<pair<int, char>>& ans) {
```

 INPUT : Koordinat (x, y) dan warna sel, jumlah Queen,
 informasi posisi dan warna dari sel yang ditempati
 Queen

 OUTPUT: Valid atau tidak valid

 CARA KERJA:

1. Apabila di baris tersebut sudah ada Queen, maka tidak valid
2. Iterasi seluruh Queen dan ambil informasi koordinat serta warnanya. Apabila kolom atau warnanya sama atau diagonal bersebelahan, maka tidak valid

```
}
```

Mengisi vector<string> dengan Queen, dilambangkan dengan char '#'

```
void updateBoard(vector<string>& board, vector<pair<int, char>>& ans) {
```

 INPUT : Papan permainan, informasi posisi dan warna dari sel
 yang ditempati Queen

 OUTPUT: Memperbarui vector<string>

 CARA KERJA:

1. Menelusuri seluruh Queen yang akan ditempatkan dan
 mengganti posisi yang sesuai menjadi char '#'

```
}
```

Menampilkan vector<string> ke terminal

```
void printBoard(vector<string>& board) {
```

 INPUT : Papan permainan

 OUTPUT: Menampilkan vector<string> ke terminal

 CARA KERJA:

1. Menelusuri seluruh elemen (sel) pada papan permainan dan
 menampilkannya ke terminal

```
}
```

```

void probSolver(vector<string>& board, vector<pair<int, char>>& ans) {
    INPUT : Papan permainan, informasi posisi dan warna dari sel yang ditempati Queen
    OUTPUT: Memberikan solusi, jika ada
    CARA KERJA:
    1. Menerapkan algoritma brute force dengan alur seperti yang telah dijelaskan sebelumnya (poin b)
}

```

d. Pengujian

* Contoh *live update* dapat dilihat pada Tes 9 (tidak didokumentasikan di seluruh tes karena terlalu panjang)

1. Tes 1

- Input: File tidak ada di ‘test’

```
Masukkan nama file (contoh: input.txt ATAU input.png ATAU input.jpg): a.txt
```

- Hasil yang Diharapkan: Error

- Hasil Pengujian:

```
ERROR: File tidak ditemukan di: test/a.txt
Pastikan file sudah ditaruh di dalam folder 'test'!
```

2. Tes 2

- Input:

```
AA
BB
```

- Hasil yang Diharapkan: Tidak ada solusi

- Hasil Pengujian:

```
Mendeteksi file teks: test/file1.txt
Berhasil membaca papan berukuran 2x2
A A
B B

PAPANINI MUSTAHIL DISELESAIKAN, HIKS :(
```

3. Tes 3

- Input:

```
AAABBCCCD
ABBBBCECD
ABBBDCED
AAABDCCCD
BBBBDDDDD
FGGGDDHDD
FGIGDDHDD
FGIGDDHDD
```

- Hasil yang Diharapkan: Error, papan tidak persegi

- Hasil Pengujian:

```
Mendeteksi file teks: test/file2.txt
Error: Ukuran papan tidak persegi!(Baris: 8, Kolom: 9)
```

4. Tes 4

- Input:

```
AAABBCCCD
ABBBBCECD
ABBBDCEDC
AAABDCCCD
BBBBBDDDD
FGGGDDHDD
FGIGDDHDD
FGGGDDHHH
```

- Hasil yang Diharapkan: Error, ada baris/kolom yang tidak valid
- Hasil Pengujian:

```
Mendeteksi file teks: test/file3.txt
Ukuran papan tidak valid di baris-8
```

5. Tes 5

- Input:

```
AAABBCCCD
ABBB CECD
A#BBBCECD
AAABDCCCD
BBBBBD@DD
FGGGDDHDD
FGIGDDHDD
FGIGDDHDD
FGGGDDHHH
```

- Hasil yang Diharapkan: Error, ada char yang tidak valid
- Hasil Pengujian:

```
Mendeteksi file teks: test/file4.txt
Error: Karakter tidak valid ditemukan di baris 2, kolom 5
Papan hanya boleh berisi Huruf Kapital (A-Z) tanpa spasi.
```

6. Tes 6

- Input:

```
AABBC
ABBC
ABBC
DDCCC
DDCCC
```

- Hasil yang Diharapkan: Tidak ada solusi
- Hasil Pengujian:

```
Mendeteksi file teks: test/file5.txt
Berhasil membaca papan berukuran 5x5
A A B B C
A B B C C
A B B C C
D D C C C
D D D C C

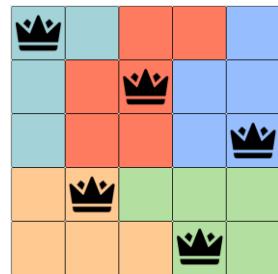
PAPAN INI MUSTAHIL DISELESAIKAN, HIKS :(
```

7. Tes 7

- Input:

```
AABBC
ABBCC
ABBCC
DDCCC
DDDCC
```

- Hasil yang Diharapkan:



- Hasil Pengujian:

```
INI JAWABANNYA!
# A B B C
A B # C C
A B B C #
D # E E E
D D D # E

Jumlah iterasi yang dilakukan: 28
Waktu Eksekusi: 0 ms
```

```
#ABBC
AB#CC
ABBC#
D#EEE
DDD#E
```



8. Tes 8

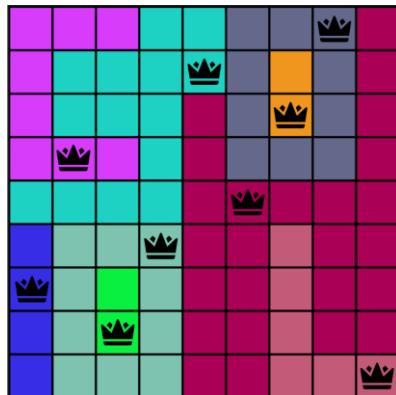
- Input:

```

AAABBCCD
ABBBBCECD
ABBBDCED
AAABDCCCD
BBBBDDDDD
FGGGDDHDD
FGIGDDHDD
FGGGDDHHH

```

- Hasil yang Diharapkan:



- Hasil Pengujian:

```

INI JAWABANNYA!
A A A B B C C # D
A B B B # C E C D
A B B B D C # C D
A # A B D C C C D
B B B B D # D D D
F G G # D D H D D
# G I G D D H D D
F G # G D D H D D
F G G G D D H H #

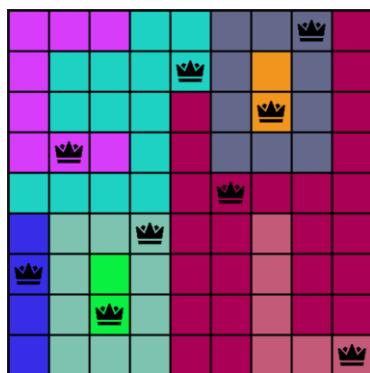
Jumlah iterasi yang dilakukan: 6106378
Waktu Eksekusi: 238 ms

```

```

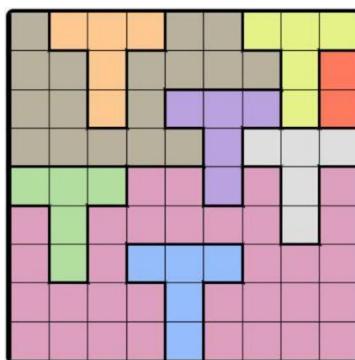
AAABBCC#D
ABBB#CECD
ABBBDC#CD
A#ABDCCCD
BBBBD#DDD
FGG#DDHDD
#GIGDDHDD
FG#GDDHDD
FGGGDDHH#

```

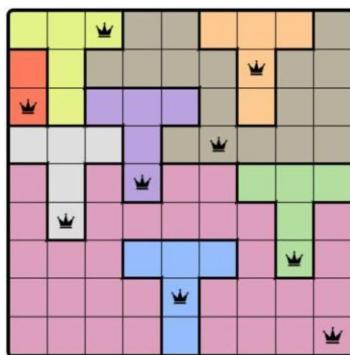


9. Tes 9

- Input:



- Hasil yang Diharapkan:



- Hasil Pengujian:

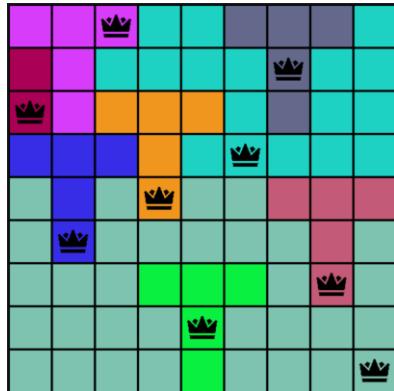
```
==== LIVE UPDATE ====
Iterasi ke: 1807838
Queens ditempatkan: 5/9
Sedang mengecek: (7, 6)
A # A B B C C C B
D A B B B B C B B
D A E E E B # B B
F F F E B B B B #
G F # E G G H H H
G F G G G G G H G
G G G I I I G # G
G G G G I G G G G
G G G G I G G G G

==== LIVE UPDATE ====
Iterasi ke: 2373220
Queens ditempatkan: 4/9
Sedang mengecek: (6, 6)
A A # B B C C C B
D A B B B # C B B
D A E E E B C B B
F F F E B B B B B
G F G E # G H H H
G F G G G G G # G
G G G I I I G H G
G G G G I G G G G
G G G G I G G G G

INI JAWABANNYA!
A A # B B C C C B
D A B B B # B B
# A E E E B C B B
F F F E B # B B B
G F G # G G H H H
G # G G G G G H G
G G G I I I G # G
G G G G # G G G G
G G G G I G G G #

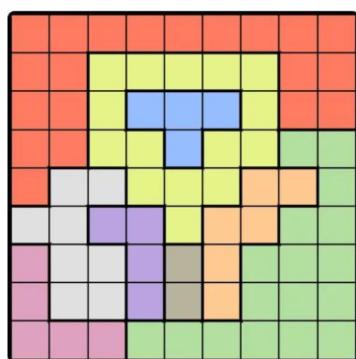
Jumlah iterasi yang dilakukan: 2398849
Waktu Eksekusi: 106 ms
```

```
AA#BBCCCB  
DABBBB#BB  
#AEEEBCBB  
FFFEB#BBBB  
GFG#GGHHH  
G#GGGGGHG  
GGGIIG#G  
GGGG#GGGG  
GGGGIGGG#
```

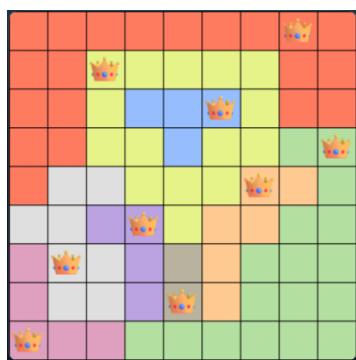


10. Tes 10

- Input:



- Hasil yang Diharapkan:



- Hasil Pengujian:

```

INI JAWABANNYA!
A A A A A A A # A
A A # B B B B A A
A A B C C # B A A
A A B B C B B D #
A E E B B B # F D
E E G # B F F D D
H # E G I F D D D
H E E G # F D D D
# H H D D D D D D

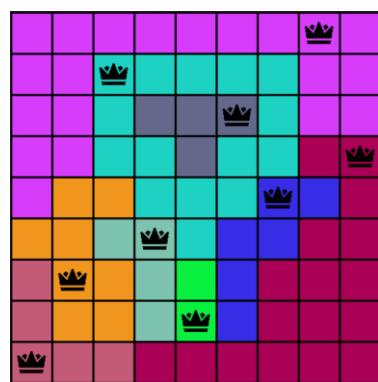
Jumlah iterasi yang dilakukan: 5040131
Waktu Eksekusi: 187 ms

```

```

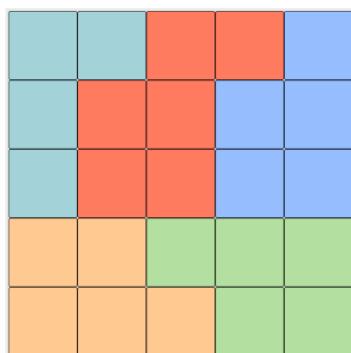
AAAAAAA#A
AA#BBBBAA
AABCC#BA
AABBCCBD#
AEEBBB#FD
EEG#BFFDD
H#EGIFDDD
HEEG#FDDD
#HHDDDDDD

```

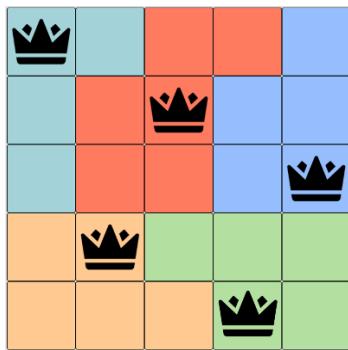


11. Tes 11

- Input:



- Hasil yang Diharapkan:

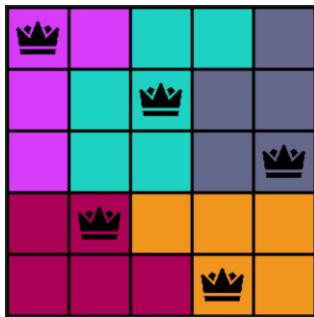


- Hasil Pengujian:

```
INI JAWABANNYA!
# A B B C
A B # C C
A B B C #
D # E E E
D D D # E

Jumlah iterasi yang dilakukan: 28
Waktu Eksekusi: 0 ms
```

```
#ABBC
AB#CC
ABBC#
D#EEE
DDD#E
```



e. LAMPIRAN

1. Kode program secara lengkap dapat dilihat pada link github berikut:

https://github.com/MikhaelYonatan/Tucil1_13524051.git

- 2.

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	✓	

f. REFERENSI

- [1] GeeksforGeeks. (n.d.). File Handling with C++ Classes. 12 Februari 2026, dari <https://www.geeksforgeeks.org/cpp/file-handling-c-classes/>
- [2] Munir, Rinaldi. (2026). Algoritma Brute Force (Bagian 1). 13 Februari 2026, dari [https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/02-Algoritma-Brute-Force-\(2026\)-Bag1.pdf](https://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/2025-2026/02-Algoritma-Brute-Force-(2026)-Bag1.pdf)
- [3] GeeksforGeeks. (n.d.). Reading and Displaying an Image in OpenCV using C++. 14 Februari 2026, dari <https://www.geeksforgeeks.org/cpp/reading-and-displaying-an-image-in-opencv-using-c/>
- [4] YouTube (Nicolai Nielsen). (n.d.). Step-by-Step Guide: Installing OpenCV C++ and Setting It Up in Visual Studio Code with CMake. 14 Februari 2026, dari https://youtu.be/m9HBM1m_EMU?si=H76lEgedfxpf14m
- [5] GeeksforGeeks. (n.d.). Draw a Rectangle using OpenCV in C++. 15 Februari 2026, dari <https://www.geeksforgeeks.org/cpp/draw-an-rectangle-using-opencv-in-cpp/>
- [6] Queens Game. (n.d.). Queens Game - Map 1. 16 Februari 2026, dari <https://www.queens-game.com/?map=map1>

g. PERNYATAAN

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



Mikhael Andrian Yonatan