

# Laporan Tugas Kecil 1

IF2211 STRATEGI ALGORITMA  
PENYELESAIAN PERMAINAN QUEENS  
SEMESTER II TAHUN 2025/2026

Disusun oleh:

**Jason Edward Salim - 13524034**

LABORATORIUM ILMU DAN REKAYASA KOMPUTASI  
PROGRAM STUDI TEKNIK INFORMATIKA  
SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA  
INSTITUT TEKNOLOGI BANDUNG

## Daftar Isi

<b>1</b>	<b>Penerapan Algoritma Brute Force</b>	<b>2</b>
1.1	Pendahuluan . . . . .	2
1.2	Alur Berpikir . . . . .	2
1.3	Implementasi Penyelesaian Brute Force . . . . .	2
1.3.1	Exhaustive Search . . . . .	2
1.4	Implementasi GUI . . . . .	3
1.5	Implementasi Input dan Output Berkas .txt . . . . .	3
1.6	Kasus Uji . . . . .	3
	<b>Lampiran</b>	<b>3</b>

# 1 Penerapan Algoritma Brute Force

## 1.1 Pendahuluan

Algoritma *Brute force* merupakan metode penyelesaian permasalahan secara *straightforward*. Pada program penyelesaian permainan Queens ini, algoritma *Brute force* digunakan secara murni dengan *exhaustive search*.

## 1.2 Alur Berpikir

Queens memiliki area bujur sangkar berukuran dinamis  $N \times N$  dimana kita menempatkan ratu pada tiap warna dimana setiap ratu yang ditempatkan dilarang berada pada baris dan kolom yang sama, serta dilarang berada di 8 sel di sekitar ratu yang lain. Untuk mendapatkan solusi dari Queens, *Brute force* murni diterapkan dengan menggunakan *exhaustive search* pada tiap baris secara terus-menerus hingga ditemukan solusi yang tepat. Pada setiap baris, penempatan ratu akan dilakukan pada tiap kolom hingga baris terakhir dan dilakukan pengecekan apakah penempatan ratu sekarang sesuai dengan aturan yang ada atau tidak. Jika belum sesuai, maka penempatan berikutnya akan dilakukan hingga ditemukan solusi yang sesuai atau tidak ditemukan solusi setelah seluruh penempatan dicoba. Solusi ini akan memiliki kompleksitas waktu sekitar  $O(N^N)$  karena pada tiap  $N$  baris terdapat  $N$  kolom yang dapat dicoba penempatannya.

## 1.3 Implementasi Penyelesaian Brute Force

Pada penyelesaian ini, algoritma akan terbagi menjadi:

1. Fungsi utama yang akan melakukan *exhaustive search* untuk penempatan ratu pada tiap baris dan kolom.
2. Fungsi validasi penempatan yang akan mengecek apakah penempatan ratu pada suatu posisi sudah sesuai dengan aturan yang ada.
3. Fungsi yang akan menyimpan dan mengembalikan solusi yang didapatkan.

### 1.3.1 Exhaustive Search

Untuk setiap baris, kita akan menempatkan ratu pada setiap kolom dan melakukan validasi setelah tiap baris memiliki satu ratu. Langkah ini dapat diimplementasikan menggunakan pengulangan dan rekursi. Pada bahasa Go, algoritmanya sebagai berikut:

---

## 1.4 Implementasi GUI

## 1.5 Implementasi Input dan Output Berkas .txt

## 1.6 Kasus Uji

## Lampiran

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		