

# Laporan Tucil 1

Nama : Wildan Abdurrahman Ghazali

NIM : 13524054

## A. Algoritma Brute Force

Dalam penyelesaian Queens LinkedIn Problem, penulis menggunakan algoritma brute force dengan langkah - langkah berikut :

- a. Buat N himpunan koordinat (x,y) untuk setiap warna pada papan permainan, masing-masing himpunan memiliki jumlah koordinat berbeda / sama (tidak dibatasi).
- b. Kemudian, buat himpunan indeks yang berisi N elemen. Himpunan ini awalnya berisi 0 sebanyak N elemen.
- c. Permutasi : Untuk setiap himpunan koordinat i, ambil satu koordinat untuk menempatkan Queen. Aturan pengambilannya adalah untuk himpunan koordinat ke-i, ambil koordinat ke-i pada himpunan indeks . Contoh :
  - i. Himpunan koordinat 1 =  $\{(0,1),(0,2)\}$
  - ii. Himpunan koordinat 2 =  $\{(2,3),(2,4)\}$
  - iii. Himpunan indeks = {2,1}Maka, permutasi koordinat nya adalah  $\{(0,2),(2,3)\}$  (Koordinat ke-2 untuk himpunan 1 dan koordinat ke-1 untuk himpunan 2).
- d. Cek apakah permutasi koordinat tersebut valid atau tidak. Permutasi dianggap valid jika tidak ada absis yang sama (berada di satu kolom yang sama), ordinat yang sama (berada di satu baris yang sama) , atau jarak antar absis dan ordinat sama dengan satu (bersebelahan secara diagonal)
- e. Jika permutasi tidak valid, increment elemen terakhir himpunan indeks.
- f. Ulangi langkah c, d, dan e hingga permutasi valid.

## B. Source Program:

- a. Algoritma Brute Force. Disini, koordinat tiap warna disimpan ke dalam struktur data map.

```

73 ▼ void MainWindow::solve(){
74     map<QChar, vector<pair<int,int>>> colours;
75     vector<QChar> huruf;
76
77 ▼     for(int i = 0; i < lines.size(); i++){
78         for(int j = 0; j < lines[i].length(); j++){
79             colours[lines[i][j]].push_back({i,j});
80         }
81     }
82     for(const auto &c : colours)
83         huruf.push_back(c.first);
84     int n = colours.size();
85     vector<pair<int,int>> queens(n);
86     vector<int> idx(n, 0);
87     bool flag = false;
88     long long iter = 0;
89
90     auto start = chrono::high_resolution_clock::now();
91     while (true) {
92         for (int i = 0; i < n; i++) {
93             queens[i] = colours[huruf[i]][idx[i]];
94         }
95         if (isValid(queens)) {
96             flag = true;
97             break;
98         }
99         if (iter > 0 && iter % 100000 == 0){
100             printUpdate(queens, iter);
101         }
102         int pos = n - 1;
103         while (pos >= 0 && ++idx[pos] == colours[huruf[pos]].size()) {
104             idx[pos] = 0;
105             pos--;
106         }
107
108         if (pos < 0) break;
109         iter++;
110     }
111     auto end = chrono::high_resolution_clock::now();
112
113     printHasil(flag, queens, iter, start, end);

```

b. Cek Validasi (Langkah d):

```

25 ▼ bool isValid(vector<pair<int,int>> queens){
26     if(queens.empty()) return 0;
27     for(int i = 0; i < queens.size(); i++){
28         for(int j = i + 1; j < queens.size(); j++){
29             if(queens[i].first == queens[j].first || queens[i].second == queens[j].second) return 0;
30             if(queens[i].first == queens[j].first-1 &&
31                 queens[i].second == queens[j].second -1) return 0;
32             if(queens[i].first == queens[j].first-1 &&
33                 queens[i].second == queens[j].second +1) return 0;
34             if(queens[i].first == queens[j].first+1 &&
35                 queens[i].second == queens[j].second -1) return 0;
36             if(queens[i].first == queens[j].first+1 &&
37                 queens[i].second == queens[j].second +1) return 0;
38         }
39     }
40     return 1;
41 }

```

C. Uji Kasus

- a. Kasus 1:  
AAABBCCCD  
ABBBBCECD  
ABBDCECD  
AAABDCCCD  
BBBBDDDDD  
FGGGDDHDD  
FGIGDDHDD  
FGGGDDHHH

- b. Hasil 1:

```
Ditemukan solusi :  
  
AAABBCC#D  
ABBB#CECD  
ABBD#CD  
A#ABDCCCD  
BBBB#DDD  
FGG#DDHDD  
#GIGDDHDD  
FG#GDDHDD  
FGGGDDHH#  
  
Banyak kasus yang ditinjau: 21415355  
Waktu pencarian : 289979628ms
```

- c. Kasus 2:

A A B B  
A A B B  
C C D D  
C C D D

- d. Hasil 2:

```
Ditemukan solusi :  
  
A#BB  
AAB#  
#CDD  
CC#D  
  
Banyak kasus yang ditinjau: 114  
Waktu pencarian : 65200ms
```

e. Kasus 3 :

A A B B C  
A D D B C  
E D D C C  
E E F F C  
E F F F C

f. Hasil 3:

Tidak ditemukan solusi

g. Kasus 4:

A A B B C C D  
A A B B C C D  
A A B B C C D  
E E F F G G D  
E E F F G G D  
A A B B C C D  
E E F F G G D  
A A B B C C D

h. Hasil 4:

Ditemukan solusi :

#ABBCCD

AA#BCCD

EEFF#GD

E#FFGGD

AABC#D

EEF#GGD

AABBCC#

Banyak kasus yang ditinjau: 33186

Waktu pencarian : 30642500ms

i. Kasus 5:

A A B B C C D D

A A B B C C D D  
E E F F G G H H  
E E F F G G H H  
A A B B C C D D  
E E F F G G H H  
A A B B C C D D  
E E F F G G H H

j. Hasil 5 :

```
Ditemukan solusi :  
  
#ABBCCDD  
  
AA#BCCDD  
  
EEFFG#HH  
  
EEFFGGH#  
  
AABB#CDD  
  
E#FFGGHH  
  
AABBCC#D  
  
EEF#GGHH  
  
Banyak kasus yang ditinjau: 682955  
Waktu pencarian : 598271000ms
```

D. Github

[wzlyy/Tucill\\_13524054: Queens Linkedin Problem](https://wzlyy/Tucill_13524054: Queens Linkedin Problem)

E. Pernyataan

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

A handwritten signature consisting of several fluid, dark, ink-like strokes forming a stylized, abstract shape.

Wildan Abdurrahman Ghazali