

# Tugas Besar IF3170 Inteligensi Buatan

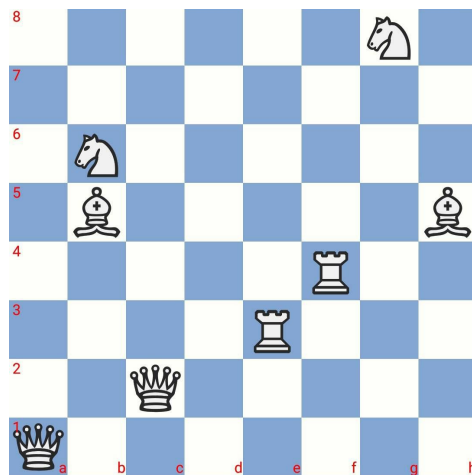
## “N-ything Problem”

### Spesifikasi

N-ything problem merupakan modifikasi N-queen problem. Perbedaannya, buah catur yang menjadi pertimbangan bukan hanya ratu (*queen*), namun juga meliputi kuda (*knight*), gajah (*bishop*), dan benteng (*rook*). Seperti N-queen problem, permasalahan dari N-ything problem adalah mencari susunan buah-buah catur pada papan catur berukuran 8x8 dengan jumlah buah catur yang menyerang buah catur lain minimum.

Secara lebih formal, cari susunan buah-buah catur sehingga jumlah pasangan terurut  $(p, q)$  di mana  $p$  menyerang  $q$  minimum. Perhatikan bahwa bila  $p$  menyerang  $q$ , belum tentu  $q$  juga menyerang  $p$ . Perhatikan juga bahwa  $(p, q)$  dan  $(q, p)$  dianggap sebagai dua pasangan yang berbeda.

Adapun sifat penyerangan ini mengikuti sifat penyerangan pada permainan catur pada umumnya. Misalnya, sebuah benteng dapat menyerang buah catur lain yang berada pada jalur vertikal/horizontal apabila buah catur tersebut tidak terhalang oleh buah catur lainnya, dan seterusnya.



**Gambar 1. Contoh solusi N-ything problem**

Untuk menyelesaikan N-ything problem ini, Anda diminta menggunakan ketiga algoritma *local search* berikut:

1. Hill climbing
2. Simulated annealing
3. Genetic algorithm

### Bonus:

Buat program Anda dapat menangani 2 warna buah catur, yaitu hitam dan putih. Cari susunan buah catur sehingga jumlah pasangan terurut  $(p, q)$  di mana  $p$  menyerang  $q$ , minimum bila  $p$  dan  $q$  berwarna sama, maksimum bila  $p$  dan  $q$  berbeda warna.

### Input:

Jumlah setiap buah catur (*knight*, *bishop*, *rook*, *queen*) berupa bilangan bulat tak negatif. Jumlah seluruh buah catur pasti kurang dari 64.

Format *input*: <warna> <jenis buah catur> <jumlah> (*input* dalam bentuk *file*)

```
WHITE KNIGHT 2
WHITE BISHOP 2
WHITE ROOK 2
WHITE QUEEN 2
BLACK KNIGHT 0
BLACK BISHOP 0
BLACK ROOK 0
BLACK QUEEN 0
```

*Catatan: bila tidak mengerjakan bonus, asumsikan seluruh buah catur berwarna hitam berjumlah 0.*

### Output:

Susunan buah-buah catur sesuai deskripsi di atas, dengan contoh format seperti di bawah ini. Pada baris terakhir, tertulis 2 buah angka  $a$  dan  $b$ , di mana  $a$  adalah jumlah bidak catur yang menyerang bidak sewarna, dan  $b$  adalah jumlah bidak catur yang menyerang bidak berbeda warna (bila tidak mengerjakan bonus, tentu  $b$  akan selalu bernilai 0).

```
. . . . . K .
.K . . . . .
. . . . .
.B . . . . . B
. . . . . R . .
. . . . . R . .
. . Q . . . .
Q . . . . .
0 0
```

*Catatan: bagi yang mengerjakan bonus, representasi untuk buah catur berwarna hitam menggunakan huruf kecil.*

## **Penilaian dan Indikator Keberhasilan**

Program dikatakan berhasil jika dapat mencari formasi buah catur di papan catur sesuai dengan spesifikasi yang telah dijelaskan sebelumnya.

Program harus menampilkan:

1. Susunan buah catur yang telah dibentuk.
2. Jumlah pasangan terurut buah catur (p, q) di mana p dapat menyerang q. Jika mengerjakan bonus, pisahkan menjadi dua: jumlah pasangan untuk p dan q yang memiliki warna yang sama dan jumlah pasangan untuk p dan q yang memiliki warna berbeda. Keterangan: tidak dilakukan penghitungan penyerangan antara 2 buah bidak catur yang terhalang oleh bidak catur lain.

Poin kelompok akan dikurangi jika terdapat hal-hal berikut:

1. Menggunakan program/library/kode buatan orang lain untuk algoritma *local search*.
2. Jumlah buah catur yang bentrok tidak minimum.
3. Jumlah setiap jenis (dan warna jika mengerjakan bonus) buah catur yang diletakkan di papan catur tidak sesuai input.
4. Kode sulit dibaca dan dimengerti, tidak memiliki struktur yang baik (P.S. usahakan kode algoritma *local search* yang Anda implementasikan *decoupled* dari kode-kode lainnya).

## Aturan Tambahan

1. Bahasa yang boleh dipergunakan adalah Python.
2. 1 kelompok terdiri dari 4-5 orang anggota.
3. Anggota kelompok boleh lintas kelas namun anggota kelompok tidak boleh sama dengan tugas kecil yang lalu.
4. Isi nama kelompok dan setiap anggotanya di sheet berikut:  
[https://docs.google.com/spreadsheets/d/1ILYFk60fmx7xAWBvn35HO7oTpkqjEdwnevXUxvt\\_1Pc/](https://docs.google.com/spreadsheets/d/1ILYFk60fmx7xAWBvn35HO7oTpkqjEdwnevXUxvt_1Pc/)
5. Pengisian nama kelompok sampai **Senin, 17 September 2018**.

## Pengumpulan dan Demo Program

Batas akhir pengumpulan file hasil pekerjaan adalah 3 minggu terhitung setelah tugas besar ini dirilis, yaitu **Rabu, 3 Oktober 2018**. Tugas dikumpulkan melalui situs kuliah dalam suatu *archive* (.zip) yang terdiri dari:

1. *Source code*.
2. Laporan. Jelaskan algoritma yang digunakan dan bagaimana implementasinya untuk persoalan ini. Cantumkan juga contoh input dan output dari program yang telah dibuat.

Penamaan file: **Tubes1\_[NIM terkecil anggota kelompok].zip** (contoh: Tubes1\_13516000.zip)

Demo akan dilakukan pada **Senin, 15 Oktober 2018** pada saat jam kuliah (pukul 07.00 sampai 08.40).