

## Penyelesaian Masalah 8-Queen Dengan *Depth First Search* Menggunakan Algoritma *Backtracking*

Muhammad Khoirussolih<sup>1</sup>, Galang Wicaksono<sup>2</sup>, Muhammad Prayogi<sup>3</sup>, Rezky Nurrohman<sup>4</sup>

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Sultan Ageng Tirtayasa.

<sup>1</sup>khoirus99@gmail.com, <sup>2</sup>legendaarydark999@gmail.com, <sup>3</sup>mprayogi91@gmail.com,

<sup>4</sup>rezkynurrohmanx4@gmail.com

**Abstrak** – *Artificial Intelligence (A.I.)* atau kecerdasan buatan dapat membantu suatu program untuk bekerja dengan pemikiran program itu sendiri, sehingga meminimalisir campur tangan pengguna. Kecerdasan buatan saat ini umum digunakan dalam permainan-permainan komputer yang memungkinkan mode player atau versus computer, seperti permainan catur yang dapat berjalan dengan sendirinya. 8-Queen merupakan suatu bentuk masalah kombinatorial klasik yang menempatkan 8 bidak pada papan catur berukuran 8x8. Penyelesaian 8-Queen dengan menggunakan algoritma Brute Force dilakukan dengan mencoba keseluruhan kemungkinan penempatan 8 bidak pada papan catur yang berukuran  $8 \times 8$  pada kolom, baris, dan diagonal yang berbeda. Namun, kompleksitas waktu yang dihasilkan sangatlah besar (lebih besar dari eksponensial). Sehingga dibutuhkan algoritma lain yang mampu menyelesaikan permasalahan 8-Queen dengan kompleksitas waktu yang lebih kecil dibandingkan dengan algoritma Brute Force.

**Kata kunci** : Kecerdasan Buatan, Algoritma, Backtracking, Brute Force, Catur, Kompleksitas, 8-Ratu

**Abstract** – *Artificial intelligence* may help a program to work with thought the program itself, thereby minimizing user intervention. Artificial intelligence is now commonly used in computer games that allow the player mode or versus the computer, like a game of chess that can run by itself. 8-Queen is a form of classical combinatorial problems that put 8 pawns on 8x8 chess board. 8-Queen completion using Brute Force algorithm is done by trying all possible placements 8 pawns on a chess board measuring  $8 \times 8$  column, row, and diagonal different. However, the complexity of the resulting time is very large (greater than exponential). And so we need another algorithm that is able to solve the problems of the 8-Queen with smaller time complexity compared with Brute Force algorithm.

**Keywords** : Artificial Intelligent, Algorithm, Backtracking, Brute Force, Chess, Complexity, 8-Queen

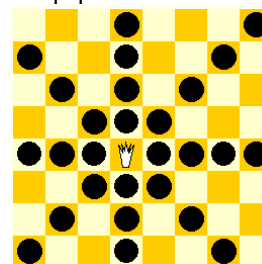
### I. PENDAHULUAN

Jurnal SETRUM adalah jurnal ilmiah untuk publikasi penelitian dan pengembangan di bidang : Sistem Kendali, Sistem Tenaga Listrik, Elektronika dan Instrumentasi, Telekomunikasi, Komputer.

Kecerdasan Buatan atau Intelegensi Artifiisial didefinisikan sebagai kecerdasan entitas ilmiah. Sistem seperti ini umumnya dianggap komputer. Kecerdasan diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Secara garis besar, AI terbagi ke dalam dua paham pemikiran yaitu AI Konvensional dan Kecerdasan Komputasional (CI, *Computational Intelligence*). AI konvensional kebanyakan melibatkan metode-metode yang sekarang di klasifikasikan sebagai pembelajaran mesin, yang ditandai dengan formalisme dan analisis statistik. Dikenal juga sebagai AI simbolis, AI logis, AI murni dan AI cara lama (GOFAI, *Good Old Fashioned Artificial Intelligence*). Metode-metodenya meliputi sistem pertimbangan berdasar kasus, jaringan Bayesian, dan AI berdasarkan tingkah laku. Sedangkan kecerdasan komputasional melibatkan pengembangan

atau pembelajaran iteratif (misalnya penalaan parameter seperti dalam sistem koneksionis. Pembelajaran ini berdasarkan pada data empiris dan diasosiasikan dengan AI non-simbolis, AI yang tak teratur dan perhitungan lunak. Metode-metode pokoknya meliputi jaringan syaraf, sistem *fuzzy*, dan komputasi evolusioner.

Dalam permainan catur, ratu adalah buah catur yang dapat bergerak dengan bebas ke segala arah ortogonal atau diagonal. Tapi tidak biasa melewati tempat yang dihalangi oleh buah sendiri. Gerakan ortogonal melewati deretan ruang yang ter-hubung di dekatnya berturut-turut dengan sisi yang sama. Gerakan diagonal melewati sudut-sudut ruang, yang menghubungkan ruang dengan warna yang sama di papan sesuai kotak-kotak.



Gambar 1. Diagram pergerakan ratu.

8-Queen merupakan salah satu bentuk permainan puzzle dengan persoalan menempatkan 8 ratu pada papan catur berukuran 8×8 sehingga tidak ada ratu yang mengancam satu sama lain. Dengan demikian, solusi meng-haruskan bahwa tidak boleh ada ratu yang berbagi baris, kolom, ataupun diagonal yang sama. 8-Queen puzzle adalah contoh yang lebih umum dari N-Queen dengan persoalan menempatkan N ratu pada papan catur berukuran N×N, di mana solusi yang ada untuk semua bilangan n dengan pengecualian dari n = 2 dan n = 3.

Permainan puzzle ini pertama kali dibentuk pada tahun 1848 oleh seorang pemain catur Max Bezzel. Dari tahun ke tahun, banyak matematikawan termasuk Gauss dan George Cantor telah bekerja keras untuk dapat menyelesaikan masalah N-Queen ini. Solusi pertama kali dibentuk oleh Franz Nauck pada tahun 1850. Nauck juga memperluas puzzle ke bentuk N-Queen. Pada tahun 1874, S. Gunter mengusulkan suatu metode dengan menggunakan metode determinan dan J.W.L. Glaisher menyaring pendekatan tersebut.

II. METODE

A. Problem Solving

Menyelesaikan masalah, dapat dilakukan secara langsung dari problem statement dan definisi konsep yang melibatkan atau dikenal dengan algoritma Brute Force. Algoritma Brute Force tersebut memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas. Namun algoritma Brute Force membutuhkan jumlah langkah yang besar dalam penyelesaiannya, sehingga menyebabkan kompleksitas waktu maupun memori yang besar pula. Hal ini berbeda dengan algoritma *Backtracking*, di mana pada algoritma *Backtracking* merupakan algoritma yang berbasis pada algoritma DFS (*Depth First Search*) yang juga merupakan perbaikan dari algoritma Brute Force yang secara sistematis mencari solusi persoalan di antara semua kemungkinan yang ada. Karena menggunakan DFS, maka proses pencarian tidak memerlukan banyak waktu. Secara umum, algoritma ini digunakan antara lain untuk menyelesaikan masalah-masalah yang berkembang secara dinamis (*dynamic problem solving*).

Ruang solusi *Backtracking* merupakan semua permutasi dari banyaknya bidak yang ingin ditempatkan pada papan catur sehingga menghasilkan kompleksitas waktu yang jauh lebih kecil. Algoritma ini memiliki kemampuan untuk menghasilkan jawaban yang sama dengan metode yang digunakan pada *Brute Force* yang menggunakan metode percobaan. Ide dasarnya adalah membangun satu komponen dengan vektor yang sama dan menggunakan kriteria modifikasi fungsi  $P(x_1, \dots, x_n)$ , kadang disebut *bounding function*, untuk menguji apakah vektor yang dibentuk mempunyai harapan untuk berhasil.

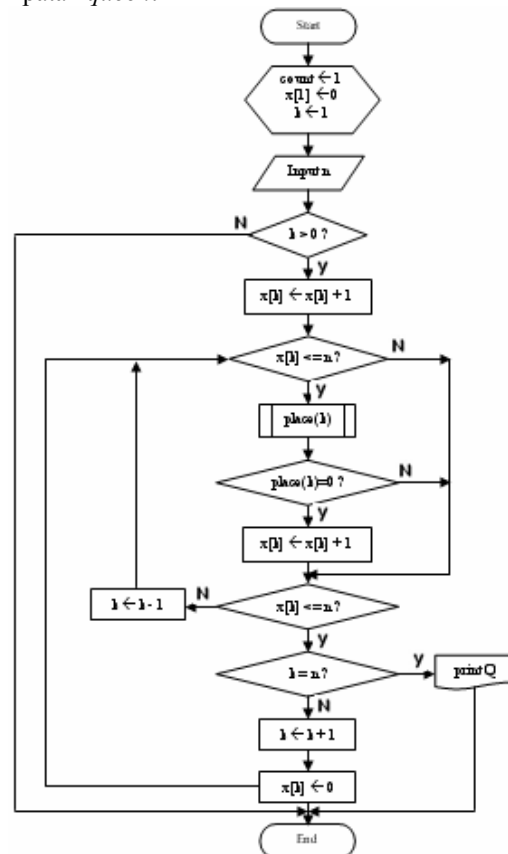
B. Program

Pada makalah ini untuk menyelesaikan permasalahan 8-Queen akan digunakan algoritma sebagai berikut :

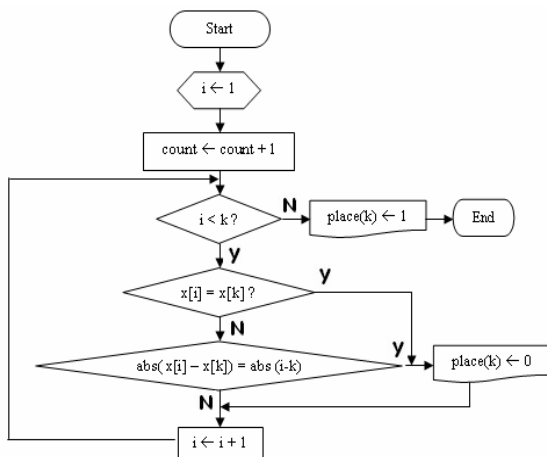
1. Inisialisasi nilai awal untuk kolom nol dan baris pertama.

2. Input bilangan integer untuk menentukan ukuran papan catur.
3. Periksa apakah posisi baris lebih besar dari nol:
  - a. Jika posisi baris lebih besar dari nol, maka pindah ke kolom berikutnya dan lanjutkan langkah 4.
  - b. Jika posisi baris lebih kecil dari nol, maka kondisi tidak terpenuhi dan akan keluar dari perulangan sehingga algoritma selesai.
4. Periksa apakah posisi kolom lebih kecil atau sama dengan ukuran papan dan periksa juga apakah posisi kolom dan diagonal yang sama dari *queen* sudah terisi oleh *queen* yang lain:
  - a. Jika salah satu kondisi tidak terpenuhi maka kembali ke langkah 5.
  - b. Jika kedua kondisi terpenuhi, maka pindah ke kolom selanjutnya dan lakukan kembali ke langkah 4.
5. Periksa apakah posisi kolom kurang dari atau sama dengan ukuran papan:
  - a. Jika memenuhi, cek apakah *queen* telah ditempatkan pada seluruh baris. Jika ya, maka cetak *queen* dan algoritma selesai. Jika tidak, maka pindah ke baris selanjutnya pada kolom awal.
  - b. Jika tidak memenuhi kondisi, maka kembali ke baris sebelumnya.
6. Ulangi langkah 3.

Algoritma di atas dapat dituangkan ke dalam bentuk *flowchart*. *Flowchart* pada program ini terdiri dari dua bagian, yaitu algoritma utama dan algoritma pemeriksaan penempatan *queen*.



Gambar 2. Flowchart Algoritma Utama.



Gambar 3. Flowchart Algoritma Pemeriksaan Penempatan Queen

Berikut program yang di tuangkan pada *pseudocode* untuk menyelesaikan masalah *8-queens* menggunakan *backtracking*:

```

SolveQueens (Integer boardSize, Queen
queen[boardSize]);
i <- 0 //Begin by placing
the queen number 0
while i < boardSize
    queen[i].row <- queen[i].row + 1
    //Place queen[i] to next row
    /* If queen[i] exceeds the row count,
    reset the queen and
    re-place queen[i-1]
    /*
    if(queen[i].row >= boardSize)
        queen[i] <- -1;
        i <- i - 1;
    else
        //While the queen[i] is under
        attack move it down the row
        while(isUnderAttack(queen[i])
            queen[i].row <- queen[i]
+ 1;
        //if queen[i] exceeds the row
        count, reset it, re-place queen[i-1]
        if(queen[i].row >= boardSize)
            queen[i].row <- -1
            i <- i - 1;
        else
            i++;
    end while
    
```

III. HASIL & PEMBAHASAN

Permasalahan pada *8-Queen* akan membutuhkan perhitungan yang kompleks jika menggunakan *Brute Force* karena ada 4.426.165.368 susunan yang mungkin dari delapan ratu pada papan catur berukuran 8x8. Namun dari banyaknya susunan tersebut hanya ada 92 solusi. Tabel berikut memberikan jumlah solusi untuk menempatkan N ratu pada papan catur berukuran NxN, baik secara fundamental maupun keseluruhan, untuk N = 1-8.

Tabel 1. Solusi *N-Queen* untuk n = 1 - 8.

n:	1	2	3	4	5	6	7	8
fundamental:	1	0	0	1	2	1	6	12
all:	1	0	0	2	10	4	40	92

Dengan menggunakan teknik *backtracking* dimungkinkan untuk melakukannya jauh lebih baik. Salah satu algoritma menghasilkan permutasi dari angka 1 sampai 8 (dimana ada 8! = 40320), menggunakan unsur-unsur setiap permutasi sebagai indeks untuk menempatkan ratu pada setiap baris, menjamin tidak ada yang saling serang secara vertikal dan horizontal. Lalu menolak posisi papan yang mendapat serangan diagonal. Program *backtracking* dengan DFS berikut, mengalami sedikit perbaikan pada metode permutasi, mengkonstruksi pohon pencarian dengan mempertimbangkan satu baris dari papan pada suatu waktu, menghilangkan sebagian posisi papan non-solusi pada tahap yang sangat awal dalam konstruksi mereka. Karena menolak serangan diagonal bahkan pada papan yang tidak lengkap, yang diperiksanya hanya 15.720 kemungkinan penempatan *queen*.

IV. KESIMPULAN

Program *8-Queens Problem* termasuk ke dalam *Artificial Intelligence* (A.I.) yang memiliki kemampuan untuk menentukan jalan keluar dari suatu permasalahan. Dalam program ini, didapat hasil bahwa penyelesaian masalah *8-Queen* dengan penggunaan algoritma *Backtracking* sangat tepat daripada menggunakan algoritma *Brute-force*.

V. DAFTAR PUSTAKA

- [1] D. Suhartono, "SEARCHING: UNINFORMED & INFORMED," 23 April 2013. [Online]. Tersedia di: <http://socs.binus.ac.id/2013/04/23/uninformed-search-dan-informed-search/>. [Diakses pada 18 Mei 2015].
- [2] F. Duniho dan H. Bodlaender, "Piecyclopedia: Queen," 11 September 1998. [Online]. Tersedia di: <http://www.chessvariants.org/piecyclopedia.dir/queen.html>. [Diakses pada 13 Mei 2015].
- [3] Hoffman. et all, "Construction for the Solutions of the m Queens Problem," Mathematics Magazine Vol. XX, 1969.
- [4] M. Richards, *Backtracking Algorithms in MCPL using Bit Patterns and Recursion*, Cambridge University Computer Laboratory, 2009.
- [5] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice Hall, 2009.
- [6] W. W. R. Ball, *The Eight Queens Problem, in Mathematical Recreations and Essays*, New York: Macmillan, 1960.