JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis

Penerapan Algoritma Backtracking Pada *N-Queen* Problem Permainan Catur

Santi Rahmawati[1], Rifda Triani Mutmainah[2], Raden Marissa Lestari[3], Asti Herliana[4]
[1,2,3,4] Fakultas Teknologi Informasi, Teknik Informatika, ARS University, Bandung, Indonesia
Email: [1]santirahmaaaaaa@gmail.com, [2]rifdatrianimutmainah21@gmail.com, [3]rdmarissalestari@gmail.com, [4]asti@ars.ac.id

Abstrak– Pada penulisan penelitian ini bertujuan untuk menerapkan Algoritma Backtracking dalam menyelesaikan masalah *N-Queen* pada permainan catur. *N-Queen* problem merupakan permasalahan yang mencari solusi penempatan bidak sebanyak N pada papan berukuran N x N dengan ilustrasi setiap bidak yang ditempatkan tidak boleh berada pada satu kolom, satu baris, dan satu diagonal yang sama sehingga tidak ada dua ratu yang saling menyerang dalam satu langkah Gerakan. Salah satu cara yang dapat menyelesaikan permasalahan *N-Queen* pada permainan catur ini yaitu dengan menggunakan Algoritma Backtracking. Dalam penerapan algoritma backtracking, solusi dicari dengan mencoba setiap kemungkinan langkah satu per satu dan kembali ke langkah sebelumnya jika langkah tersebut tidak menghasilkan solusi yang benar dan mencoba Langkah lain hingga solusi ditemukan atau semua kemungkinan telah dicoba. Penelitian ini menggunakan metode kepustakaan dimana metode ini melibatkan pencarian, peninjauan dari sumber-sumber tertulis yang berhubungan dengan topik yang sedang diteliti. Dengan membentuk pohon ruang status (State Space Tree) pada Algoritma Backtracking menghasilkan solusi pada N = 4 dengan jalur [2,4,1,3].
Kata Kunci: Algoritma Backtracking; Catur; *N-Queen*; State Space Tree

Abstract–In writing this study aims to apply the Backtracking Algorithm in solving the *N-Queen* problem in chess games. The *N-Queen* problem is a problem that seeks a solution to the placement of N pieces on a board measuring N x N with an illustration that each piece placed cannot be in the same column, row, and diagonal so that no two queens attack each other in one Movement Step. One way that can solve the *N-Queen* problem in this chess game is by using the Backtracking Algorithm. In the application of the backtracking algorithm, the solution is sought by trying each possible step one by one and returning to the previous step if the step does not produce the correct solution and trying another Step until the solution is found or all possibilities have been tried. This research uses the literature method where this method involves searching, reviewing written sources and other documentation related to the topic being studied. By forming a state space tree in Backtracking Algorithm, the solution at N = 4 is the solution path [2,4,1,3].
Keywords: Backtracking Algorithm; Chess; *N-Queen*; Pohon Ruang Status

1. PENDAHULUAN

Catur adalah permainan papan strategis untuk dua pemain. Papan catur berbentuk persegi dan dibagi menjadi 64 kotak terang dan gelap bergantian yang disusun dalam kisi berukuran 8x8. Setiap pemain memiliki 16 buah dengan nama yang berbeda dan pergerakan yang unik. Buah catur tersebut terdiri dari Raja, Menteri/Ratu, Gajah, Benteng dan Bidak/Pion. Bidak yang paling penting adalah raja, karena tujuan setiap pemain adalah untuk menskakmat raja lawan untuk memastikan bahwa ia tidak dapat menghindari penangkapan. Jika dihitung secara matematis, untuk satu langkah saja (putih dan hitam menjalankan buah caturnya satu kali) akan menunjukkan banyak kemungkinan yang berbeda.

Permainan catur telah menjadi bagian tak terpisahkan dari sejarah dan budaya manusia selama berabad-abad. Keterampilan yang dibutuhkan untuk bermain catur bukan hanya kemampuan untuk menggerakkan buah catur di sekitar papan, tetapi juga kemampuan untuk merencanakan strategi, berpikir kritis, dan membuat keputusan yang tepat. Salah satu tantangan paling menarik dalam catur adalah menempatkan ratu di papan dengan cara yang tidak saling mengancam, yang disebut dengan *N-Queen* problem.

Salah satu upaya memenangkan permainan ini yaitu dengan cara Minimalkan kekalahan dengan menerima Informasi tentang posisi/kedudukan bidak catur yang aman dan tidak berbahaya. Satu permasalahan yang ada dan menarik untuk diteliti yaitu *N-Queen* problem. Masalah *N-Queen* terdiri dari menempatkan N- Queen pada papan catur N × N sehingga tidak ada ratu yang menyerang satu sama lain, yaitu tidak ada dua ratu yang berada di baris yang sama, di kolom yang sama, dan di diagonal yang sama.

Solusi untuk masalah ini adalah menemukan konfigurasi penempatan ratu yang memenuhi semua batasan ini. Banyak algoritma telah dikembangkan untuk memecahkan masalah *N-Queen*. Salah satu algoritma yang efektif memecahkan masalah ini adalah algoritma backtracing. Algoritma backtracking ini secara rekursif memilih langkah yang mengarah ke solusi. Jika langkah ini tidak mengarah pada solusi yang diinginkan, sistem kembali mencoba langkah alternatif. Dengan menggunakan algoritma backtracking, sistem dapat mencapai solusi optimal dalam menyelesaikan masalah *N-Queen* dengan lebih efisien dibandingkan dengan metode pencarian lainnya.Hal ini memungkinkan pemain catur untuk mempelajari strategi dan gerakan yang tepat dalam situasi permainan akhir tertentu.

Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**

JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis

Beberapa jurnal yang mengatakan bahwa permasalahan *N-Queen* dapat diselesaikan oleh algoritma backtracing seperti penelitian yang dilakukan oleh Halida Astatin "Pemanfaatan pohon Realisasi Algoritma backtracing untuk memecahkan *N-Queen* problem"[1]. Kemudian Fritno Purba "Penerapan Algoritma Ranut Balik (Backtracing) dalam *N-Queen* problem permainan catur[2].

Tujuan dari penelitian ini adalah menerapkan algoritma backtracing untuk menyelesaikan masalah *N-Queen* dalam permainan catur. Dalam penelitian ini, peneliti menjelaskan langkah-langkah implementasi algoritma backtracing dan menganalisis kinerja algoritma dalam menyelesaikan masalah *N-Queen* untuk ukuran papan tertentu.

2. TINJAUAN PUSTAKA

2.1 Catur

Catur merupakan salah satu permainan papan yang paling populer di dunia pada saat ini. Permainan catur merupakan sebuah permainan berbasis strategi abstrak yang mucul pada abad ke-6 Masehi. Permainan ini berasal dari India utara pada abad ke-6 Masehi sebagai Chaturaga dan semakin popular diseluruh India pada abad ke-7. Hingga pada tahun 1924 catur diakui sebagai permainan dunia oleh *Federation International des Echecs (FIDE)* atau Federasi Catur Internasional merupakan organisasi internasional yang bertanggung jawab atas mengatur dan mengembangkan catur di tingkat global dan telah memberikan legitiasi global bagi pemain catur sebagai olahraga intelektual dan kompetitif yang diakui secara Internasional. Catur adalah permainan papan strategis untuk dua pemain. Papan catur berbentuk persegi dan dibagi menjadi 64 kotak terang dan gelap bergantian yang disusun dalam kisi berukuran 8x8. Setiap pemain memiliki 16 buah dengan nama yang berbeda dan pergerakan yang unik. Buah catur tersebut terdiri dari Raja, Menteri/Ratu, Gajah, Benteng dan Bidak/Pion. Bidak yang paling penting adalah raja, karena tujuan setiap pemain adalah untuk menskakmat raja lawan untuk memastikan bahwa ia tidak dapat menghindari penangkapan [3]

2.2 Algoritma Backtracking (Runut-Balik)

Algoritma Backtracking adalah algoritma yang berbasis pada DFS untuk mencari solusi persoalan secara lebih efisien, secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada dengan metode runut-balik, tidak perlu memeriksa semua kemungkinan solusi. Hanya langkah yang mengarah pada solusi saja yang perlu dipertimbangkan. Akibatnya, waktu pencarian dapat dihemat. Ciri khas dalam Algoritma Backtracking yaitu adanya fungsi pemangkasan (pruning). Pencarian solusi pada masalah direpresentasikan dalam bentuk pohon solusi, proses pemangkasan akan dilakukan terhadap simpul-simpul yang tidak mengarah kepada solusi. Jika suatu simpul telah dipangkas, simpul-simpul yang menjadi anak dari simpul tersebut tidak akan diproses, karena memangkas sebuah simpul sama halnya membuang seluruh lintasan yang berada di bawah simpul tersebut. Algoritma backtracking banyak digunakan dalam membuat permainan komputer, salah satunya adalah catur [1].

2.3 *N-Queen*

N-Queen merupakan generalisasi dari masalah 4-Queen dan 8-Queen, ditempatkan pada papan catur N x N dengan masing-masing ratu tidak berada pada baris, kolom dan diagonal yang sama. Dalam permasalahan *N-Queen*, tujuan utamanya adalah menempatkan N ratu pada papan catur sehinggan tidak ada dua ratu yang saling menyerang Ketika digerakkan. Ini berarti setiap baris, kolom maupun pada diagonal hanya boleh diisi oleh satu ratu. Jika solusi ditemukan, berarti semua ratu telah ditempatkan denga aman dan tidak saling menyerang satu sama lain [4]

2.4 State Space Tree (Pohon Ruang Status)

State space tree adalah salah satu teknik dalam pemecahan masalah yang melibatkan pemodelan masalah dalam bentuk struktur pohon. Pada setiap simpul di pohon, terdapat sebuah keadaan atau state yang merepresentasikan suatu solusi dari masalah yang sedang dipecahkan, yaitu penempatan beberapa ratu pada papan catur. Setiap cabang dari simpul tersebut merepresentasikan sebuah langkah atau operasi yang dapat dilakukan untuk menghasilkan solusi baru, yaitu penempatan satu ratu pada baris berikutnya. Penerapan algoritma backtracking pada *N-Queen* problem dapat dilakukan dengan membangun state space tree yang merepresentasikan semua kemungkinan penempatan ratu pada papan catur [5].

Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**

JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis

3. METODE

3.1 Pendekatan Penelitian

Pendekatan penelitian yang digunakan dalam penelitian ini menggunakan metode kepustakaan (*library research*), penelitian ini dilakukan dengan cara mengumpulkan data dari berbagai sumber seperti jurnal, skripsi, dan dokumen lainnya yang berhubungan dengan topik yang sedang diteliti. Selain itu, peneliti juga menggunakan strategi Depth-First-Search (DFS) pada backtracking dalam proses pencarian solusi dan menerapkan pohon solusi (State Space Tree) dalam merepresentasikan langkah-langkah yang dapat diambil dalam pencarian solusi

3.2 Tahapan penelitian

Adapun tahapan dalam penelitian ini meliputi perumusan masalah, inisialiasi, membuat pohon ruang status, penerapan DFS dan penyelesaian solusi menggunakan Algoritma Backtracking. Tahapan tersebut dapat dijelaskan sebagai berikut:

a. Merumuskan masalah

Sebelum melakukan penelitian, yakni menentukan masalah yang akan diselesaikan. Pada penelitian ini masalah yang akan diselesaikan yaitu penyelesaian *N-Queen* pada catur dengan menggunakan Algoritma Backtracking.

b. Pohon ruang status (State Space Tree)

Pohon ruang status (State Space Tree) cara untuk menggambarkan status-status persoalan kedalam simpul pohon tersebut yang merupakan representasi visual dari semua kondisi, yang mungkin dalam permasalahan. Setiap simpul mewakili kondisi dan cabang-cabang pohon mewakili langkah-langkah yang mungkin untuk mencapai kondisi berikutnya dengan demikian melihat pohon ruang status dapat mengidentifikasi dan mengumpulkan semua solusi yang valid [5]

c. Penerapan Depth-First-Search (DFS)

Setelah membuat pohon ruang status, selanjutnya menerapkan Depth-First-Search (DFS) yang secara sistematis digunakan untuk pencarian solusi dengan menjelajahi pohon ruang status, melibatkan penjelajahan mendalam pada pohon ruang status dengan mengunjungi dari setiap simpul-simpul sebelum kembali ke simpul sebelumnya [6]

d. Algoritma Backtracking

Algoritma Backtracking menggunakan strategi DFS dalam proses pencarian solusi. Algoritma Backtraking menerapkan DFS untuk menjelajahi pohon ruang status, Algoritma Backtracking juga merupakan algoritma pencarian yang digunakan untuk menemukan kemungkinan solusi. Kemunginan solusi yang ada perlu diuji secara bertahap. Algoritma ini bekerja dengan mencoba setiap langkah secara bertahap dan mundur (*Backtrack*) ketika langkah tersebut tidak menghasilkan solusi yang valid [2]

Dengan kata lain, jika lintasan pada suatu simpul yang dibuat hasilnya melanggar aturan batasan, maka simpul tersebut akan dihentikan dan melakukan *Backtrack* untuk kembali pada simpul sebelumnya, lalu diteruskan dengan mencoba simpul anak lainnya [5]

4. HASIL DAN PEMBAHASAN

4.1 Pengujian Algoritma

Algoritma Backtracking memiliki property umum, salah satunya *Bounding Function* (Fungsi Pembatas). *Bounding Function* adalah kondisi yang bernilai benar. Kondisi dapat dikatakan benar jika ratu (*Queen*) tidak pada baris, kolom dan diagonal yang sama. Fungsi pembatas ini dinyatakan sebagai B(x1,x2,x3,...xn) akan bernilai true jika (x1,x2,x3,…..xn) mengarah pada solusi dan tidak melanggar batasan.

Dalam pencarian solusi, perlu dilakukan pembangkitan simpul-simpul status pada pohon sehingga menghasilkan lintasan dari akar ke daun. Dalam tahap ini, menggunakan strategi *Depth-First-Search* (DFS) untuk membangkitkan simpul. Simpul-simpul yang dibangkitkan disebut simpul hidup. Jika simpul hidup yang diperluas maka lintasan akan semakin Panjang dan jika lintasan tidak mengarah pada solusi dengan artian melanggar fungsi pembatas, maka simpul tersebut akan dimatikan yang disebut simpul mati. Dengan kondisi seperti ini maka akan dijalankan Algoritma Backtracking dengan syarat jika pembentukan lintasan berhenti karena aadnya simpul mati maka, akan dilakukan pemunduran (*Backtrack*) untuk Kembali pada simpul sebelumnya (orangtua), lalu diteruskan dengan membangkitkan simpul anak lainnya. Pencarian dihentikan jika sudah mencapai simpul tujuan.

Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**

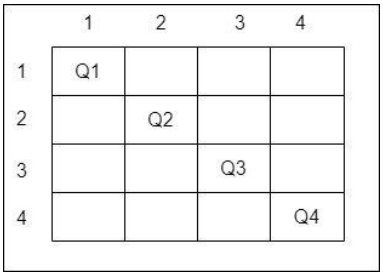
JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis

Algoritma Backtracking ini akan direpresentasikan dalam bentuk pohon ruang status yang mengandung semua kemungkinan solusi dari permasalahan 4-Queen dengan papan catur 4x4. Menempatkan 4 ratu (*Queen*) ditempat yang berbeda dengan tidak berada pada baris, kolom dan diagonal yang sama sehingga tidak ada ratu yang saling menyerang. Rincinya pada pohon ruang status (State Space Tree) tiap simpul menyatakan status dari persoalan, sedangkan sisi-sisi dilabeli nilai vektor solusi (x1,x2,….xn).

4.2 Implementasi Algoritma

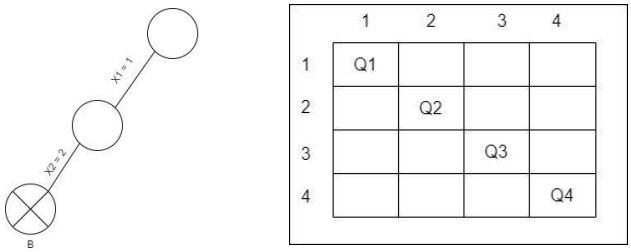
Penerapan tahapan-tahapan dalam penyelesaian permasalahan *N-Queen* akan dilakukan dimana peneliti akan menggunakan papan catur berukuran 4x4 dan N= 4 yang berarti menempatkan 4 buah ratu pada tempat yang berbeda dimulai dengan membuat pohon ruang status dan gambaran papan dari hasil kemungkinan pada pohon ruang status tersbut sesuai fungsi pembatas.

Langkah pertama dalam membuat pohon status, peneliti akan terlebih dahulu menempatkan Q1,Q2,Q3,Q4 = Queen dikolom 1,2,3,4 sebagai inisialisasi awal dengan gambaran sebagai berikut



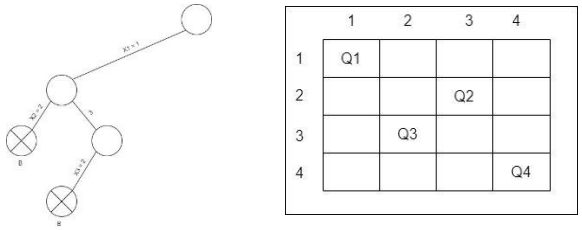
Gambar 1. Inisialisasi awal penempatan ratu

Gambar diatas menunjukkan bahwa Q1 menempati baris dan kolom ke 1, Q2 menempati baris dan kolom 2, Q3 menempati baris dan kolom 3, dan Q4 menempati baris dan kolom 4. Selanjutnya, mulai percobaan memindahkan ratu satu ke kolom yang lain dengan pohon ruang status dan gambaran pada papan catur sebagai berikut



Gambar 2. Percobaan pertama pada pohon ruang status dan hasil pada papan catur 4 x 4

Gambar 2 menunjukkan bahwa tidak masalah menempatkan x1 = 1 (Q1 pada kolom dan baris 1), tetapi bermasalah jika menempatkan Q2 pada kolom dan baris kedua, karena ia menjadi sejajar diagonal dengan Q1,Q3,Q4, maka hal ini melanggar fungsi batasan bahwa antara ratu tidak boleh berada pada diagonal yang sama sehingga simpul x2 = 2 akan dihentikan. Selanjutnya mencoba untuk membuat langkah berikutnya



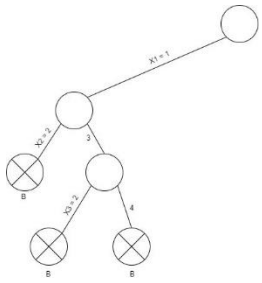
Gambar 3. Percobaan kedua pada pohon ruang status dan hasil pada papan catur 4 x 4

Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**

JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis

Gambar 3 menunjukkan perpindahan Q2 ke kolom 3 baris 2 maka dibuatlah simpul baru x2 = 3 dan Q3 berpindah ke kolom 2 baris 3. Maka, lintasan akan dibuat di sisi kanan karena simpul kiri sudah dihentikan. Namun, Q3 melanggar fungsi batas karena sejajar diagonal dengan Q2 dengan itu simpul x3=2 dihentikan dan mundur (*Backtrack*) pada simpul sebelumnya. Disinilah algoritma Backtracking dan DFS berjalan.

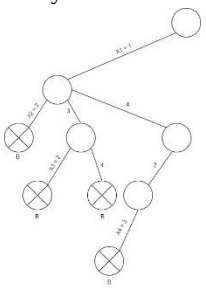
Percobaan masih tetap berlanjut, dengan memindahkan Q3 dan ditempatkan pada kolom 4 baris 3 dengan gambaran sebagai berikut



	1	2	3	4
1	Q1			
2			Q2	
3				Q3
4				Q4

Gambar 4. Percobaan ketiga pada pohon ruang status dan hasil pada papan catur 4 x 4

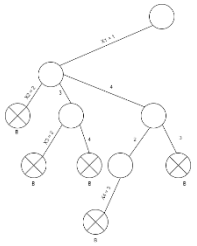
Gambar 4 menunjukkan perpindahan Q3 di kolom 4 baris 3, namun percobaan ini melanggar fungsi Batasan karena menjadi sejajar diagonal dengan Q2 dan berada sejajar dengan kolom Q4. Maka simpul ini dihentikan dan kembali pada simpul sebelumnya.



	1	2	3	4
1	Q1			
2				Q2
3		Q3		
4			Q4	

Gambar 5. Percobaan keempat pada pohon ruang status dan hasil pada papan catur 4 x 4

Gambar 5. Menunjukkan perpindahan Q2 ke dalam kolom 4 baris 2, Q3 ke dalam kolom 2 baris 3 dan Q4 ke dalam kolom 3 baris 4. Terlihat pada gambar papan catur bahwa Q4 melanggar fungsi batas dimana Q4 menjadi sejajar diagonal dengan Q3, maka simpul dihentikan dan kembali pada simpul sebelumnya yaitu simpul x3 = 2. DFS disini berperan menjelajahi sejauh mungkin sebelum kembali ke simpul sebelumnya. Hal ini dapat menghasilkan pencarian yang lebih dalam pada stuktur pohon ruang status.



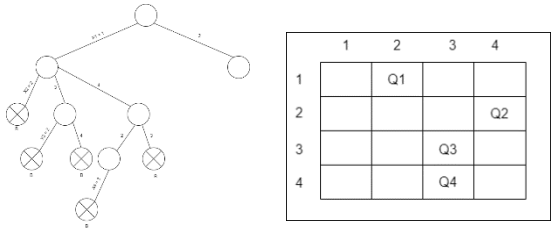
	1	2	3	4
1	Q1			
2				Q2
3			Q3	
4			Q4	

Gambar 6. Percobaan kelima pada pohon ruang status dan hasil pada papan catur 4 x 4

Pada percobaan kelima memindahkan Q3 ke dalam kolom dan baris 3, hasil percobaan tersebut melanggar fungsi batasan yaitu Q3 sejajar dengan kolom Q4 sehingga simpul x3=3 dihentikan. Jika terus menerus hanya memindahkan Q2,Q3 dan Q4 maka solusi tidak akan ditemukan, dengan itu peneliti akan mencoba memindahkan Q1 dengan gambaran sebagai berikut

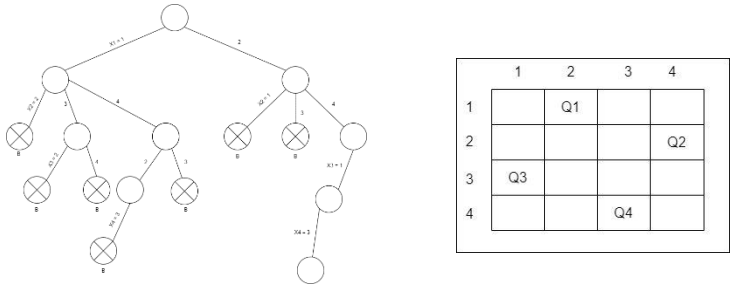
Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**

JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis



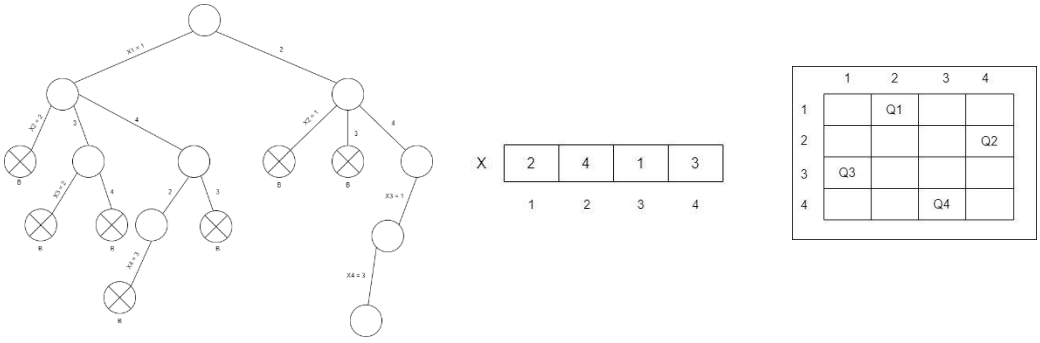
Gambar 7. Percobaan keenam pada pohon ruang status dan hasil pada papan catur 4 x 4

Pada percobaan keenam menujukkan bahwa Q3 sejajar diagonal dengan Q2 dan berada pada kolom yang sama dengan Q4. Jika Q3 dipindahkan kedalam kolom 4 akan sejajar dengan Q2, maka Q3 dipindahkan dan menempati kolom 1.



Gambar 8. Percobaan ketujuh pada pohon ruang status dan hasil pada papan catur 4 x 4

Pada percobaan ketujuh terlihat bahwa Q3 tidak berada pada diagonal yang sama dengan Q1 dan Q4, serta tidak berada pada baris, kolom yang sama dengan Q1,Q2 dan Q4. Selanjutnya, jika melakukan percobaan pemindahan pada Q4 ditempatkan pada kolom 1 maka akan berada pada kolom yang sama dengan Q3, jika dipindahkan kedalam kolom 2 maka akan berada pada kolom yang sama dengan Q1 dan jika dipindahkan kedalam kolom 4, maka akan berada sejajar pada kolom Q2, demikian Q4 akan ditempatkan pada kolom 3 dimana Q4 tidak berada pada baris, kolom, diagonal yang sama dengan Q1,Q2 dan Q3.



Gambar 9. Hasil akir percobaan ketujuh pada pohon ruang status dan hasil pada papan 4 x 4

Hasil dari representasi pohon ruang status, penggunaan DFS dan Algoritma Backtracking pada permasalahan *N-Queen* pada papan catur berukuran 4 x 4 menghasilkan solusi dengan tujuh percobaan pada N = 4 berupa jalur solusi [2,4,1,3] dimana Q1 ditempatkan pada kolom 2, Q2 ditempatkan pada kolom 4, Q3 ditempatkan pada kolom 1 dan Q4 ditempatkan pada kolom 3. Dengan ini tiap ratu tidak akan saling menyerang dalam satu langkah gerakan dikarenakan tiap ratu tidak berada pada baris, kolom maupun diagonal yang sama dan jika langkah-langkah baru dilakukan maka akan menghasilkan solusi baru dengan jalur solusi [3,1,4,2].

Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**

JURNAL INFORMATION SYSTEM
Volume 3, Nomor 1, Mei 2023, Page 010-016
Email : jis@unkris.ac.id
ISSN 2807-7849 (media online)
Available Online at https://journal.teknikunkris.ac.id/index.php/jis

4.2 Implementasi Aplikasi

Pada penelitian ini, untuk menyelesaikan permasalahan *N-Queen*. Selain melalui langkah-langkah diatas, peneliti juga menggunakan program C++ dalam penyelesaiannya dengan kode utama dan output sebagai berikut

```cpp
int main() {
  int n = 4; // mengambil papan ukuran 4*4 dan 4 ratu
  Solution obj;
  vector < vector < string >> ans = obj.solveNQueens(n);
  for (int i = 0; i < ans.size(); i++) {
    cout << "Arrangement " << i + 1 << "\n";
    for (int j = 0; j < ans[0].size(); j++) {
      cout << ans[i][j];
      cout << endl;
    }
    cout << endl;
  }
  return 0;
}
```

```
Arrangement 1
..Q.
Q...
...Q
.Q..

Arrangement 2
.Q..
...Q
Q...
..Q.
```

Gambar 9. Implementasi dan hasil program C++ dalam penyelesaian *N-Queen*

5. KESIMPULAN

Penggunaan Algoritma Backtracking pada permasalahan ini mampu menyelesaikan permasalahan pada *N-Queen* dengan menempatkan 4 ratu pada papan catur yang relatif kecil, karena untuk ukuran catur dan jumlah ratu yang lebih banyak memerlukan waktu yang cukup lama untuk mencapai pada solusi dengan kata lain implementasi saat ini belum sepenuhnya optimal untuk menangani kasus dengan nilai *n* yang lebih besar. Oleh karena itu, penelitian selanjutnya dapat mengatasi keterbatasan ini dengan mengusulkan strategi baru untuk meningkatkan efisiensi algoritma, serta mempertimbangkan pendekatan yang mampu menemukan semua solusi yang valid. Selain itu, penelitian mendatang dapat melibatkan analisis lebih mendalam tentang keterkaitan antar jumlah ratu dan ukuran papan catur untuk memberikan wawasan yang lebih komprehensif. Dengan memperbaiki keterbatasan-keterbatasan ini, penelitian di masa depan diharapkan dapat memberikan kontribusi yang lebih dalam pemecahan masalah *N-Queen* menggunakan *Algoritma Backtracking*.

REFERENCES

[1] H. Astatin, “Pemanfaatan Pohon dalam Realisasi Algoritma Backtracking untuk Memecahkan N-Queens Problem.”
[2] F. Purba, “PENERAPAN ALGORITMA RANUT BALIK (BACKTRACKING) DALAM N-QUEEN PROBLEM PERMAINAN CATUR,” *Jurnal Pelita Informatika*, vol. 6, no. 1, 2017, [Online]. Available: http://www.kyma26.co.cc/2009/01/sejarahcatur
[3] R. E. Jeremiah, W. Sukmo Wardhono, and H. Muslimah Az-Zahra, “Analisis Pengalaman Interaksi Pengguna Terhadap Permainan Catur Sebagai Obyek Augmented Reality Menggunakan Game Experience Questionaire,” 2019. [Online]. Available: http://j-ptiik.ub.ac.id
[4] A. Adrifina, S. Wati, and U. Gunadarma, “Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008) Auditorium Universitas Gunadarma”, [Online]. Available: http://www.stttelkom.ac.id/staf/ZKA/
[5] M. Sheva and A. Sofjan, “Pemodelan Algoritme Backtracking dalam Menyelesaikan Permainan 2048 Deterministik.” [Online]. Available: http://informatika.stei.itb.ac.id/~rinaldi.munir/Matdis/2020-
[6] M. Khoirussolih *et al.*, “Penyelesaian Masalah 8-Queen Dengan Depth First Search Menggunakan Algoritma Backtracking,” vol. 4, no. 1, 2015, [Online]. Available: http://socs.binus.ac.id/2013/04/23/uninformed-

Submitted: **14/05/2023**; Accepted: **26/05/2023**; Published: **31/05/2023**