

Penerapan Pohon dengan Algoritma *Branch and Bound* dalam Menyelesaikan *N-Queen Problem*

Arie Tando (13510018)

Program Studi Teknik Informatika

Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia

13510018@std.stei.itb.ac.id

Abstract—Makalah ini membahas tentang teori dasar tentang pohon dan juga penerapannya dalam algoritma *branch and bound*. Algoritma *branch and bound* adalah algoritma yang digunakan untuk menyelesaikan solusi pencarian di dalam ruang solusi secara sistematis dengan langkah terpendek. Salah satu permasalahan yang dapat diselesaikan dengan algoritma *branch and bound* adalah *N-Queen Problem* yang merupakan permasalahan dalam menyusun *N* buah bidak Ratu dalam permainan papan catur dengan ukuran papan adalah $N \times N$ dengan syarat bidak Ratu tidak saling makan dengan cara terpendek.

Kata kunci—Algoritma *Branch and Bound*, *N-Queen Problem*, Pohon.

1. PENDAHULUAN

Permasalahan yang ada di dalam kehidupan sehari-hari pastilah membutuhkan solusi yang tepat serta efisien agar masalah dapat selesai dengan benar tanpa waktu yang cukup lama. Sama halnya di dalam dunia informatika, permasalahan di dunia informatika sangatlah banyak sehingga ada banyak cara untuk menyelesaikan suatu permasalahan. Contoh permasalahan yang berkaitan dengan kehidupan sehari-hari adalah pencarian jalur terpendek. Dalam mencari jalur terpendek bisa dengan beberapa algoritma untuk menyelesaikan masalah tersebut sebagai contoh bisa dengan *BFS (Breadth-First Search)*, *DFS (Depth-First Search)*, Algoritma Runut Balik, Algoritma *Brute Force*, Algoritma *Branch and Bound*, dan lain lain. Algoritma-algoritma untuk mencari jalan terpendek adalah algoritma pencari solusi dan masih banyak algoritma yang bisa digunakan untuk mencari jalur terpendek, tetapi tidak semua algoritma yang telah disebutkan dapat digunakan untuk menyelesaikan suatu permasalahan secara efisien. Dalam permasalahan tertentu setiap algoritma dapat menyelesaikan masalah dengan tingkat efisien yang berbeda-beda.

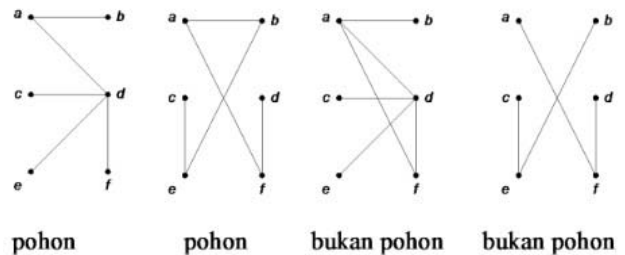
Dalam makalah ini, akan dibahas tentang *N-Queen Problem*. *N-Queen Problem* adalah permasalahan yang dapat diselesaikan dengan algoritma pencari solusi. Salah satu algoritma pencari solusi yang akan digunakan untuk *N-Queen Problem* adalah Algoritma *Branch and Bound* atau biasa disebut Algoritma *B&B*. Algoritma *Branch and Bound* adalah algoritma yang dapat di representasikan

dengan struktur pohon yang merupakan salah satu pokok bahasan didalam Struktur Diskrit.

2. DASAR TEORI

2.1 Pohon

Definisi pohon adalah graf tak-berarah terhubung yang tidak mengandung sirkuit.

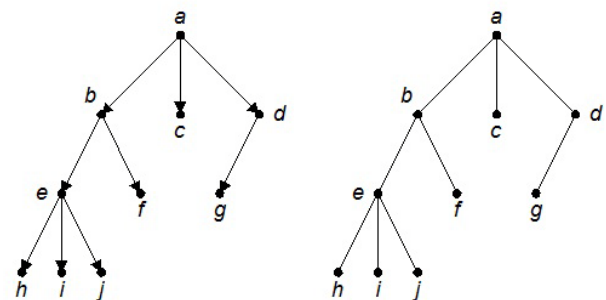


Gambar 1

Sifat-sifat pohon : misalkan $G = (V, E)$ adalah graf tak-berarah sederhana dan jumlah simpulnya n . Maka, semua pernyataan di bawah ini adalah ekuivalen :

1. G adalah pohon
2. Setiap pasang simpul di dalam G terhubung dengan lintasan tunggal
3. G terhubung dan memiliki $m = n - 1$ buah sisi
4. G tidak mengandung sirkuit dan memiliki $m = n - 1$ buah sisi
5. G tidak mengandung sirkuit dan penambahan satu sisi pada graf akan membuat hanya satu sirkuit
6. G terhubung dan semua sisinya adalah jembatan

Pohon berakar (*rooted tree*) adalah pohon yang satu buah simpulnya diperlakukan sebagai akar dan sisi-sisinya diberi arah sehingga menjadi graf berarah.

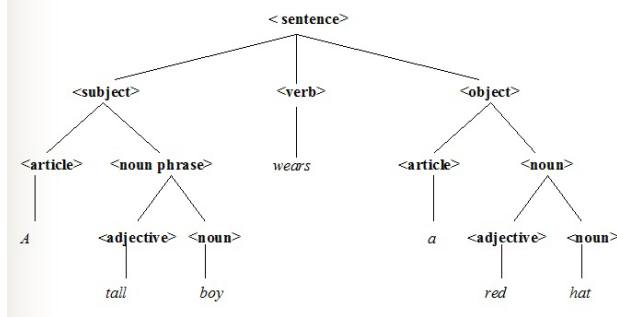


Gambar 2 : contoh pohon berakar

Terminologi pada pohon berakar (Gambar 2)

1. Anak (*child*) dan Orangtua (*parent*)
b,c, dan d adalah anak-anak simpul a, a adalah orangtua dari anak-anak itu
2. Lintasan (*path*)
Lintasan dari a ke j adalah a,b,e,j. Panjang lintasan dari a ke j adalah 3
3. Saudara kandung (*sibling*)
f adalah saudara kandung e, tetapi g bukan saudara kandung e, karena orangtua mereka berbeda
4. Derajat (*degree*)
Derajat sebuah simpul adalah jumlah anak pada simpul tersebut.
Derajat a adalah 3, derajat b adalah 2, derajat d adalah 1 dan derajat e adalah 0.
Derajat maksimum dari semua simpul merupakan derajat pohon itu sendiri.
5. Daun (*leaf*)
Simpul yang berderajat nol disebut daun
6. Simpul (*node*)
Simpul yang memiliki anak disebut simpul dalam.

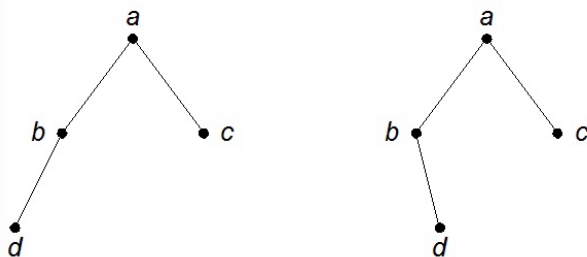
Pohon *n*-ary adalah Pohon berakar yang setiap simpul cabangnya mempunyai paling banyak n buah anak disebut pohon *n*-ary



Gambar 3 : contoh pohon 3-ary

Pohon *n*-ary dikatakan teratur atau penuh jika setiap simpul cabangnya mempunyai tepat n anak.

Pohon Biner adalah pohon *n*-ary dengan $n = 2$ dengan simpul dalam pohon biner mempunyai paling banyak 2 buah anak dibedakan antara anak kiri (*left child*) dan anak kanan (*right child*).

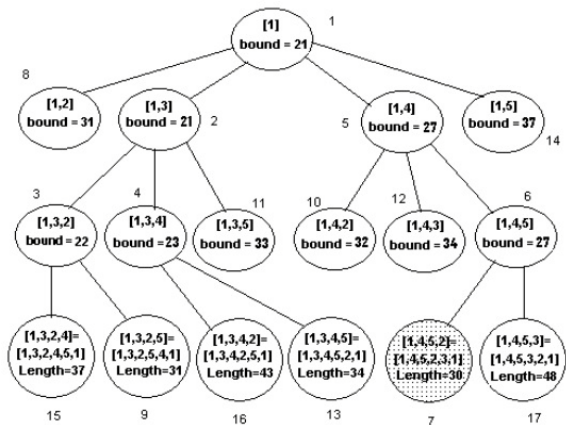


Gambar 4

Pada gambar 4 merupakan gambar pohon biner yang berbeda karena pada simpul b memiliki anak d dengan posisi berbeda. Pada gambar kiri d merupakan right dari b sedangkan pada gambar kanan d merupakan left dari b.

2.2 Algoritma Branch and Bound

Algoritma *Branch and Bound* (B&B) merupakan metode pencarian di dalam ruang solusi secara sistematis. Ruang solusi direpresentasikan kedalam pohon ruang status. Algoritma *Branch and Bound* mirip dengan algoritma Runut-Balik. Pada algoritma Runut-Balik menggunakan skema DFS, sedangkan pada Algoritma *Branch and Bound* menggunakan skema BFS. Walaupun algoritma *Branch and Bound* berdasarkan skema BFS Algoritma *Branch and Bound* berbeda dengan BFS. Untuk mempercepat pencarian solusi, maka pada algoritma *Branch and Bound* diberikan sebuah nilai ongkos pada setiap simpul sehingga dalam pencarian nilai tidak lagi berdasarkan urutan pembangkitnya tetapi berdasarkan nilai ongkos yang di berikan pada setiap simpul. Simpul dengan ongkos yang lebih kecil akan memberikan taksiran ongkos termurah yang merupakan jalur dari simpul menuju simpul solusi. Ongkos yang dimaksud pada algoritma ini adalah suatu fungsi pembatas yang berguna untuk membatasi pembangkit yang simpulnya tidak mengarah ke simpul solusi.



Gambar 5

Pada gambar 5 merupakan contoh algoritma *Branch and Bound*. Pada simpul 1 merupakan simpul awal dengan urutan simpul berikutnya adalah simpul 2,5,8,14 karena nilai bound pada simpul $2 < \text{simpul } 5 < \text{simpul } 8 < \text{simpul } 14$. Lalu simpul berpindah pada simpul 2 dan dilakukan lagi pengurutan simpul seperti diatas hingga menemukan solusi. Pada simpul 6 ditemukan solusi yaitu merupakan perluasan dari simpul 6 adalah simpul 7 dan simpul 17 dengan simpul 7 sebagai solusi. Setelah simpul solusi ditemukan maka simpul hidup yang lain tidak dilanjutkan lagi karena sudah menemukan ongkos dengan nilai terkecil.

3. N-QUEEN PROBLEM

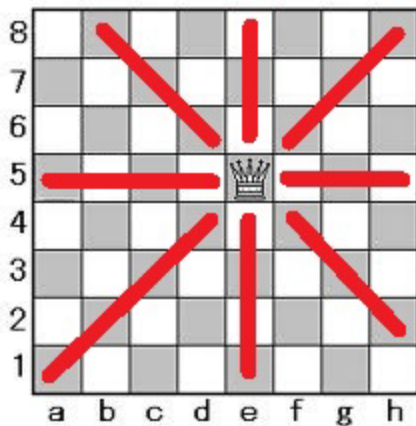
3.1 Definisi N-Queen Problem

Pertama kali permasalahan ini bernama 8 Queen Puzzle. 8 Queen Puzzle pertama kali diusulkan pada tahun 1848 oleh pemain catur Max Bezzel. Setelah beberapa tahun, banyak matematikawa, termasuk Gauss, telah bekerja

mencari solusi puzzle tersebut dan akhirnya diberi nama N-Queen Problem. Solusi pertama kali di berikan oleh Franz Nauck pada tahun 1850. Nauck juga memberikan solusi puzzle dengan N-Queen Problem.

Edsger Dijkstra pada tahun 1972 memakai masalah ini untuk memperkenalkan pemrograman terstruktur. Dia mempublikasikan dengan detail tentang pengembangan dari Depth First Backtracking Algorithm.

N-Queen Problem merupakan suatu permasalahan dimana ada terdapat N buah bidak Ratu dalam permainan catur dan dengan papan catur berukuran N x N. Bidak Ratu sebanyak N ditempatkan sedemikian rupa dalam papan catur dengan syarat Bidak Ratu tidak boleh saling memakan satu sama lain. Arah gerak makan bidak Ratu pada *N-Queen Problem* sama dengan arah gerak bidak Ratu pada permainan catur secara umum yaitu bidak Ratu dapat bergerak ke arah horizontal kiri, horizontal kanan, vertikal atas, vertikal bawah dan juga ke arah diagonal, sehingga tidak boleh ada bidak Ratu yang saling berada di dalam satu garis horizontal, vertikal, dan diagonal. Gambar arah gerak bidak Ratu seperti gambar di bawah ini.



Gambar 6

Gambar 6 merupakan gambar arah gerak bidak Ratu sehingga pada jalur dengan warna merah tidak boleh ada bidak Ratu yang lain.

3.2 Solusi N-Queen Problems

Solusi dalam *N-Queen Problems* mempunyai solusi yang berbeda-beda dan dapat mempunyai lebih dari 1 solusi. Untuk 8 Queen problem memiliki 92 solusi penempatan bidak Ratu. Jika sebuah solusi pada 8 Queen problem yang simetris (rotasi atau refleksi) hanya dianggap sebagai 1 solusi, maka akan terdapat 12 solusi unik dalam 8 Queen problem.

<i>n</i> :	1	2	3	4	5	6	7	8	9	10
unique:	1	0	0	1	2	1	6	12	46	92
distinct:	1	0	0	2	10	4	40	92	352	724

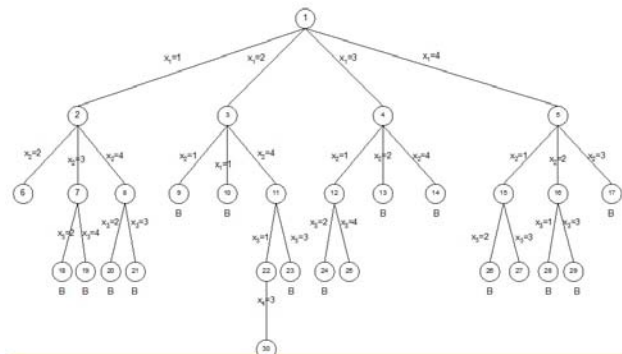
Gambar 7

Table jumlah solusi dari 1 sampai 10 dalam *N-Queen Problem*.

3.3 Prinsip BFS dalam pencarian solusi N-Queen Problem

Prinsip pencarian solusi pada algoritma *Branch and Bound* adalah dengan metode *BFS*. Pada metode ini, semua anak yang dibangkitkan dari simpul-E yaitu simpul saat sedang diacu, berada pada aras yang sama. Algoritma *BFS* menggunakan sistem antrian (*queue*) untuk menyimpan data-data yang sedang diproses. Simpul-simpul yang dibangkitkan akan disimpan ke dalam antrian (*queue*). Simpul-simpul yang akan di proses atau di bangkitkan akan diekspansi dari kepala antrian. Prinsip antrian pada metode *BFS* sama dengan skema *FIFO (First In First Out)*. Simpul hidup dimasukkan ke dalam antrian lalu simpul berikutnya yang akan menjadi simpul-E (simpul yang akan di proses) adalah simpul pertama pada antrian.

Persoalan *N-Queen Problem* dengan algoritma *Branch and Bound* yang akan di tinjau adalah persoalan 4-Ratu. Pada metode *BFS*, pencarian dilakukan secara dinamis. Setiap simpul diberi urutan secara melebar sesuai dengan urutan pembangkitannya.

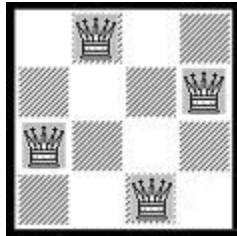


Gambar 7

Pada gambar 7 merupakan algoritma *BFS* yang direalisasikan sebagai bentuk pohon ruang status untuk persoalan 4-Ratu. Pada simpul 1 dibangkitkan 4 simpul dan disimpan dalam antrian. Pada kepala antrian diambil simpul 2 sebagai simpul-E dan di dalamnya dibangkitkan lagi 3 simpul dan disimpan dalam antrian. Begitu juga dengan simpul 3 sampai simpul 5. Setelah simpul 5 dijalankan, diambil kepala antrian yaitu simpul 6 dan diproses. Karena simpul 6 tidak dapat menemukan solusi maka nilai simpul 6 diberi nilai B sehingga tidak dapat diteruskan lagi. Pengecekan dilakukan pada simpul-simpul yang lain sesuai dengan urutan antrian, jika simpul tidak layak diteruskan maka tidak dimasukkan pada antrian sehingga simpul tersebut tidak akan diproses lagi.

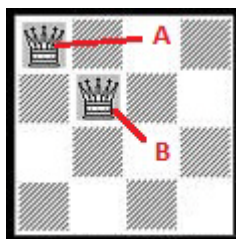
Jika dilihat dari gambar 7, maka simpul solusi ditemukan pada simpul 30 karena simpul 30 merupakan simpul solusi. Pada saat simpul solusi pertama ditemukan, masih terdapat simpul-simpul yang masih hidup di dalam antrian yaitu simpul 25 dan simpul 27. Metode *BFS* bekerja berdasarkan *FIFO (First In First Out)* sehingga mengakibatkan metode *BFS* kalah cepat dalam menemukan solusi di dalam persoalan *N-Queen Problem* karena metode ini melakukan pengecekan dari seluruh

simpul pohon pada tingkat yang sama secara melebar. Mengecekan pohon pada simpul-E dilakukan berdasarkan urutan pembangkit di dalam antrian sehingga seakan-akan “buta” karena tidak tahu dimana solusi berada dimana menyebabkan pencarian solusi dari simpul-E tidak bisa dipastikan menuju solusi yang tepat sehingga untuk mencari solusi maka perlu dilakukan pengecekan simpul secara keseluruhan. Skema FIFO yang berlaku memang mengharuskan semua simpul hidup di dalam antrian di proses dan diperluas sebelum mencapai simpul 30.



Gambar 8

Gambar 8 merupakan hasil dari pencarian solusi *N-Queen Problem* dengan metode *BFS*. Pada simpul 1, $X_1=1$ yang artinya penempatan posisi bidak Ratu pada baris-1 kolom ke-1. Lalu pada simpul 2 untuk nilai $X_2=2$. Karena bidak Ratu tidak dapat diposisikan pada baris ke-2 kolom ke-2 maka nilai dari simpul tersebut adalah B yang artinya tidak layak lagi dilakukan pencarian karena bidak Ratu saling makan. Berikut ini ilustrasi untuk persoalan diatas.



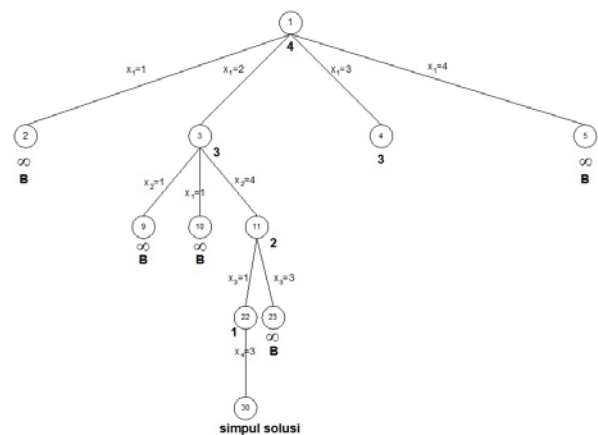
Gambar 9

Pada gambar 9 posisi A adalah posisi dari simpul 1 arah $X_1=1$, menempatkan bidak Ratu di posisi 1,1. Lalu dilanjutkan dengan simpul 2 dengan $X_2=2$, menempatkan bidak Ratu di posisi 2,2. Hal ini dilakukan secara rekursif hingga ditemukan simpul yang menjadi solusi sehingga ditemukan sesuai dengan gambar 8 yang merupakan realisasi dari pohon ruang status pada gambar 7.

3.4 Pencarian solusi N-Queen Problem dengan cara terpendek menggunakan algoritma *Branch and Bound*

Prinsip algoritma *Branch and Bound* sangat mirip dengan *BFS*, hanya saja pada algoritma *Branch and Bound* ditambahkan nilai *cost* atau ongkos dari setiap simpul yang bertujuan sebagai penanda prioritas di dalam antrian. Dengan menambah nilai ongkos dari masing – masing simpul maka nilai ongkos tersebut akan menjadi penentu prioritas yang ada di dalam antrian yang di pakai di dalam metode *BFS*, dengan prioritas terkecil akan berada pada antrian terdepan.

Nilai ongkos pada simpul-simpul bertujuan untuk menyatakan nilai batas (*bound*). Nilai batas dalam algoritma *Branch and Bound* dicari dengan suatu fungsi pembatas yaitu suatu fungsi yang bertujuan membangkitkan suatu simpul dengan melihat apakah simpul tersebut mengarah ke simpul solusi dengan memberikan suatu nilai sebagai nilai batas (*bound*). Cara kerja dari fungsi pembatas dapat berupa [HOR78] yaitu dengan cara mencari jumlah simpul dalam subpohon X yang perlu dibangkitkan sebelum simpul solusi ditemukan atau panjang lintasan dari simpul X ke simpul solusi terdekat. Simpul X adalah simpul dimana akan diberi nilai batas atau nilai ongkos. Gambar dibawah ini adalah realisasi algoritma *Branch and Bound* menggunakan teori pohon dalam persoalan 4-Ratu.



Gambar 10

Bila dilihat pada gambar 10, pada simpul-E pertama kali adalah simpul 1, pada bagian bawah simpul 1 diberi nilai ongkos 4. Simpul 1 dilanjutkan dengan membangkitkan 4 simpul lainnya yaitu simpul 2,3,4, dan 5. Dengan menggunakan fungsi pembatas maka didapat pada simpul 2 dan 5 yaitu $X_1=1$ dan $X_1=4$ diberi nilai ongkos sebesar ∞ yang artinya simpul tersebut akan berada di paling belakang antrian sehingga tidak akan diproses karena tidak mengarah pada simpul solusi sehingga tidak mungkin bidak Ratu berada pada posisi (1,1) atau (1,4). Langkah selanjutnya dengan fungsi pembatas, nilai simpul 3 dan 4 yaitu $X_1=2$ dan $X_1=3$ diberi nilai batas sebesar 3. Simpul-simpul yang telah diberi nilai batas dimasukkan ke dalam antrian dengan urutan simpul adalah 3,4,2, dan 5.

Selanjutnya simpul yang akan diproses berdasarkan antrian adalah simpul 3. Pada simpul 3 dilakukan dengan membangkitkan 3 simpul dengan nama simpul 9,10, dan 11. Simpul-simpul yang baru dibangkitkan diberi nilai batas dari fungsi pembatas sehingga hasil yang didapat berupa simpul 9 dan simpul 10 diberi nilai, simpul 11 diberi nilai 2. Dengan simpul-simpul yang telah diberikan nilai maka dimasukkan ke dalam antrian sehingga antrian berubah dengan urutan simpul menjadi 11, 4, 2, 5, 9,dan 10. Setelah itu simpul-E dilanjutkan dengan mengambil kepala antrian yaitu simpul 11. Pada simpul 11 dibangkitkan 2 simpul lainnya dengan nama simpul 22 dan 23. Simpul 22 diberi ongkos 1 dan simpul 23 diberi

ongkos ∞ . Nilai simpul yang telah diberikan dimasukkan ke dalam antrian dengan urutan simpul adalah 22, 4, 2, 5, 9, 10, dan 23.

Simpul dilanjutkan dengan mengambil nilai kepala antrian yaitu simpul 22. Pada simpul 22 diperluas dan di peroleh simpul 30 yang merupakan simpul solusi pertama sehingga hasil pencarian untuk solusi 4-Ratu. Jika dilanjutkan maka akan menemukan solusi yang berikutnya hingga semua antrian selesai diproses.

Dari algoritma *BFS* dan algoritma *Branch and Bound* dalam *N-Queen Problem* didapatkan bahwa algoritma *Branch and Bound* lebih singkat dari algoritma *BFS* dalam persoalan mencari cara terpendek dari *N-Queen Problem*. Dalam algoritma *BFS* semua simpul hasil perluasan dari simpul-E diproses dan diberi nilai pembatas, proses tersebut diulang hingga nilai simpul solusi ditemukan. Sedangkan algoritma *Branch and Bound* melakukan pencarian dengan cara memberi nilai pembatas pada semua simpul perluasan tetapi hanya diproses dari simpul yang memiliki prioritas simpul tersebut mengarah pada simpul solusi, sehingga jalan terpendek dalam mencari simpul solusi dapat dengan cepat ditemukan.

4. KESIMPULAN

Pohon adalah sebuah graf tak berarah yang memiliki banyak kegunaan di dalam aplikasinya. Konsep pohon digunakan untuk merealisasikan bentuk algoritma *BFS* dan algoritma *Branch and Bound* di dalam *N-Queen Problem*.

Algoritma *Branch and Bound* adalah algoritma dengan metode dasar dari algoritma *BFS*. Algoritma *Branch and Bound* berbeda dengan *BFS* adalah pada fungsi pembatas, pemasukan di dalam antrian serta proses dalam mencari simpul solusi. Dalam mencari jarak terpendek dalam menemukan solusi dari *N-Queen Problem* dapat digunakan algoritma *Branch and Bound* karena dengan algoritma ini proses dalam mencari solusi lebih cepat dibandingkan dengan algoritma *BFS*.

DAFTAR PUSTAKA

- [1] Munir, Rinaldi, Slide Perkuliahan IF2091, Pohon, 2011.
- [2] Munir, Rinaldi, Slide Perkuliahan, Algoritma *Branch and Bound*.
- [3] http://en.wikipedia.org/wiki/Eight_queens_puzzle diakses tanggal 9 Desember 2011, pukul 21.00

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 10 Desember 2010



Arie Tando / 13510018