Aplikasi Algoritma *Backtracking* pada Permainan Angka Ajaib

Arga Dinata – 13505066

Program Studi Inforamtika
Sekolah Tinggi Elektro dan Informatika
Institut Teknologi Bandung

if15066@students.if.itb.ac.id

ABSTRAK

Pada makalah ini akan dibahas mengenai penerapan algoritma *backtracking* (runut-balik) dalam menyelesaikan permainan angka ajaib. Algoritma runut-balik adalah algoritma yang berbasis pada *DFS (Depth First Search)* untuk mencari solusi persoalan secara lebih efisien. Algoritma runut-balik sebenarnya merupakan perbaikan dari algoritma *brute-force*, yang secara sistematis mencari solusi persoalan di antara semua kemungkinan solusi yang ada. Dengan menggunakan metode ini, kita tidak perlu memeriksa semua kemungkinan solusi yang ada, namun hanya langkah yang cenderung mengarah ke himpunan solusi saja yang selalu dipertimbangkan. Karena itulah waktu pencarian dapat dihemat. Algoritma runut-balik lebih alami dinyatakan dalam algoritma rekursif.

Permainan angka ajaib adalah sebuah permainan klasik dalam menyusun angka-angka. Permainan ini menggunakan 9 buah kotak pada satu bujur sangkar dengan sisi yang masing-masing memiliki 3 buah kotak. Angka 1 sampai 9 dimasukkan secara merata pada tiap kotak sedemikian rupa sehingga jumlah angka pada tiap baris maupun kolom memiliki jumlah yang sama, yaitu 15. Makalah ini akan mengupas bagaimana algoritma runut-balik menyelesaikan permainan ini.

Kata kunci: *backtracking*, *brute-force*, *DFS,* rekursif, angka ajaib *(magic number)*.

1. PENDAHULUAN

Dalam kehidupan sehari-hari, kita sering melakukan permainan kecil antara 2 orang untuk mengisi waktu yang sedang luang. Biasanya permainan yang dilakukan adalah permainan memanipulasi angka-angka yang membutuhkan sedikit kinerja otak untuk menyelesaikannya. Salah satu contoh permainan tersebut adalah permainan angka ajaib, atau yang lebih familiar disebut dengan *magic number game.*

Permainan ini merupakan permainan olah angka yang sederhana, namun dibutuhkan sedikit usaha untuk meyelesaikannya. Permainan ini menggunakan 9 buah kotak pada satu bujur sangkar dengan sisi yang masing – masing memiliki 3 buah kotak. Angka 1 sampai 9 didistribusikan secara merata pada tiap kotak sedemikian rupa sehingga jumlah angka pada tiap baris maupun kolom memiliki nilai yang sama, yaitu 15.

Sebenarnya algoritma *brute-force* dapat digunakan untuk menyelesaikan permainan ini dengan mencoba semua kemungkinan solusi. Untuk mendapatkan pencarian solusi yang optimal maka digunakanlah algoritma runut-balik ini. Algoritma runut-balik merupakan perbaikan dari algoritma *brute-force*, yang digunakan untuk mengeliminasi kandidat solusi yang tidak mengarah ke himpunan solusi yang sebenarnya. Berikut ini merupakan salah satu contoh penyelesaian permainan angka ajaib yang menggunakan algoritma runut-balik.

1	5	9
6	7	2
8	3	4

Gambar 1. Contoh solusi permainan angka ajaib

2. METODE

Pada bab ini akan dijelaskan mengenai beberapa metode yang dapat digunakan untuk menyelesaikan permainan angka ajaib ini. Setiap metode yang digunakan akan dilihat perbandingan kompleksitasnya dengan algoritma runut-balik yang akan dibahas pada makalah ini.

2.1 *Brute-Force*

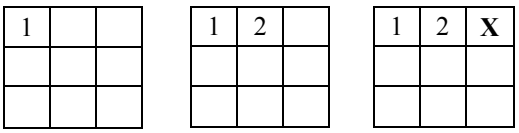
Dengan metode *brute-force* ini, kita mencoba semua kemungkinan solusi penempatan sembilan buah angka pada tiap kotak papan permainan. Sembilan angka tersebut kita letakkan pada tiap kotak secara sembarang, kemudian kita periksa apakah penempatan angka-angka tersebut memenuhi syarat solusi, yaitu jumlah angka pada tiap kolom dan baris adalah 15. Cara ini merupakan salah satu metode yang dapat menemukan solusi pada permainan angka ajaib, namun metode ini tidak efisien, karena kita perlu memeriksa sebanyak

$$9! = 362.880$$

buah kemungkinan solusi.

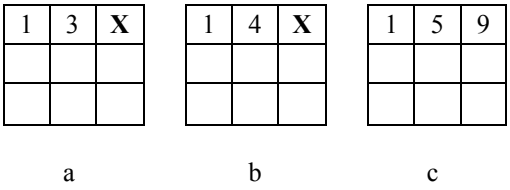
2.2 Runut-balik 1

Metode yang kedua adalah menggunakan algoritma runut-balik. Dengan menggunakan algoritma ini, jika salah satu langkah tidak mengarah ke solusi, maka kita akan kembali ke langkah sebelumnya. Dengan begitu, langkah yang tidak memiliki solusi tidak perlu dilanjutkan. Berikut merupakan gambar langkah-langkah metode ini.



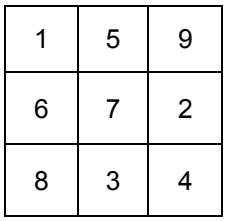
Gambar 2. Contoh langkah pada metode runut-balik 1

Angka 1 diletakkan pertama kali pada kotak pertama, kemudian angka 2 (Gambar 2b) dan seterusnya. Semua angka harus diletakkan pada setiap kotak pada papan permainan. Namun pada kotak ketiga, angka 3 dimasukkan dan bukan merupakan solusi (karena jumlah pada baris tersebut tidak sama dengan 15). Dikarenakan 3 bukan angka yang tepat, angka 4 dimasukkan, namun angka ini juga bukan merupakan angka yang tepat, angka berikutnya dimasukkan. Langkah ini terus berulang sampai angka 9. Jika sampai angka 9 tidak kunjung ditemukannya solusi, maka langkah harus mundur ke kotak sebelumnya, dan mengganti angka pada kotak tersebut dengan angka selanjutnya, pada kasus ini angka yang dimaksud adalah 3 (Gambar 3a).



Gambar 3. Langkah pada algoritma runut-balik 1

Metode ini lebih efektif daripada algoritma *brute-force*, karena metode ini tidak perlu mencoba semua kemungkinan. Metode ini hanya melanjutkan langkah-langkah yang mengarah ke solusi. Algoritma ini akan diperbaiki pada metode berikutnya unutk mendapatkan langkah-langkah yang lebih efektif. Hasil akhir pencarian solusi dengan algoritma ini adalah sebagai berikut.



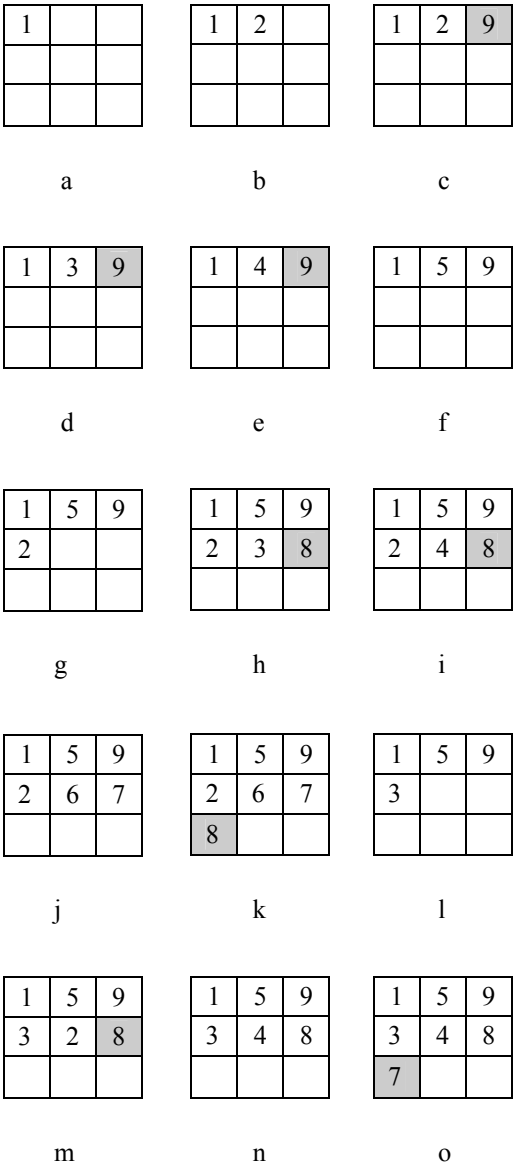
Gambar 4. Solusi runut-balik 1

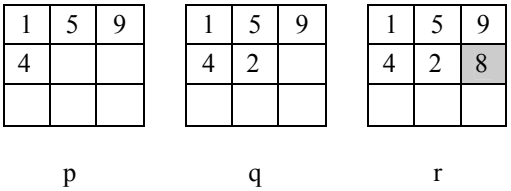
2.3 Runut-balik 2

Metode runut-balik 2 ini merupakan perbaikan dari metode runut-balik 1. Pada metode ini, kolom atau baris terakhir akan diisi dengan angka yang terbesar terlebih dahulu. Hal ini dilakukan mengingat jika angka terbesar saja tidak mencapai nilai 15, maka angka yang di bawahnya pun tidak akan dapat mencapai angka tersebut. Dengan begitu dapat dipastikan bahwa kolom sebelumnya

adalah angka yang salah, sehingga kita harus melakukan runut balik ke kolom kedua.

Selain itu, perbedaan juga terletak pada sistem runut-baliknya. Jika pada kolom pertama tidak ditemukan solusi, maka langkah akan mundur ke baris sebelumnya pada kolom pertama, bukan pada kolom terakhir. Hal ini dilakukan karena yang salah bukan angka pada kotak sebelumnya, namun pada kolom pertama baris sebelumnya yang menyebabkan kolom pertama tersebut tidak berjumlah 15. Berikut ini adalah contoh gambar untuk memperjelas penjelasan di atas.





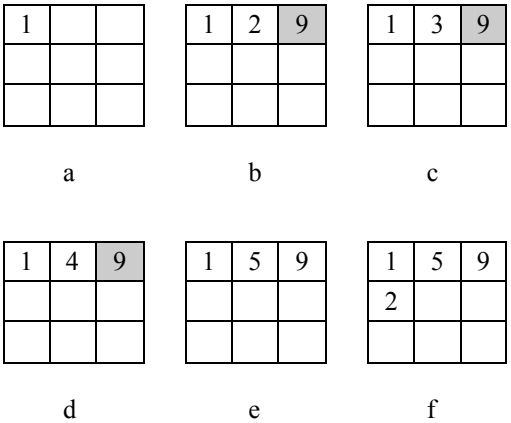
Gambar 5. Contoh langkah pada metode runut-balik 2

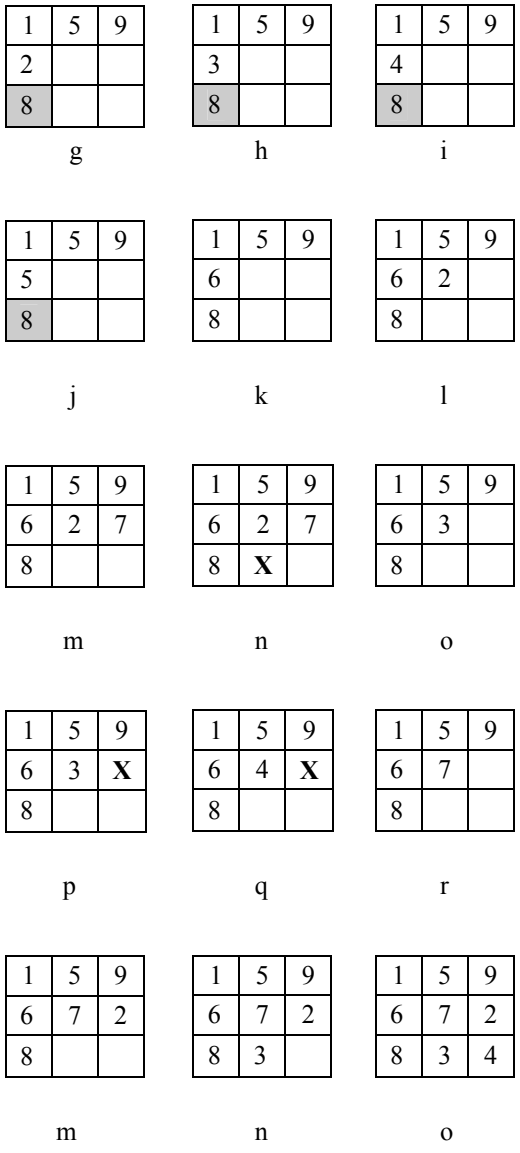
Metode ini juga belum seutuhnya efektif, hal ini akan diperbaiki dengan algoritma runut-balik 3 yang akan dijelaskan berikut.

2.4 Runut-balik 3

Runut-balik 3 adalah metode yang menurut saya terbaik dari runut-balik yang lainnya. Metode ini merupakan perbaikan dari runut-balik 1 dan 2.

Pada runut-balik 3 ini menggunakan sistem yang digunakan pada runut-balik 2. Namun, algoritma ini menggunakan pola penempatan angka pada baris dan kolom secara bergantian. Maksudnya, pertama kita mengisi baris kesatu terlebih dahulu, kemudian kolom kesatu, baris kedua, kolom kedua, dan seterusnya. Pola seperti ini akan menghemat penempatan angka yang dianggap tidak memiliki jumlah 15. Jadi, jumlah langkah untuk meyelesaikan permainan ini lebih sedikit dari metode-metode lain yang telah dijabarkan di atas. Berikut merupakan gambar langkah-langkah solusi metode runut-balik 3.





Gambar 6. Langkah-langkah runut-balik 3

4. KESIMPULAN

Permainan angka ajaib dapat diselesaikan dengan berbagai macam metode, diantaranya *brute-force* ataupun runut-balik. Dengan algoritma *brute-force*, pencarian solusi permainan ini menjadi tidak cukup efektif, karena kita harus mencoba seluruh kemungkinan solusi.

Dikarenakan algoritma *brute-force* ini tidak cekup efektif, maka digunakanlah metode runut-balik yang pada dasarnya sama dengan *brute-force*, namun langkah yang tidak mengarah ke himpunan solusi kita eliminasi dan mundur satu langkah ke belakang.

Pada algoritma runut-balik ini juga masih ada langkah yang sia-sia. Kita dapat melakukan eliminasi pada langkah-langkah tersebut dengan tujuan mendapatkan metode yang benar-benar optimal.

REFERENSI

[1] Rinaldi Munir, Diktat Kuliah IF2251, Program Studi Teknik Informatika, Sekolah Teknik dan Informatika ITB, 2007.