

ANALISIS PENERAPAN ALGORITMA RUNUT BALIK PADA PERMAINAN NUMBERFILL

Muhammad Iqbal - 13505060

Program Studi Teknik Informatika,
Sekolah Teknik Elektro dan Informatika,
Institut Teknologi Bandung
Labtek V, Jln. Ganesha 10, Bandung
Email: kirimkeiqbal@students.itb.ac.id

ABSTRAK

Permainan di dunia ada berbagai macam jenis. Banyak di antaranya merupakan permainan *puzzle*. *Puzzle* atau teka-teki adalah jenis permainan yang menuntut kita berpikir dalam memecahkan permasalahan yang ditawarkan oleh teka-teki itu sendiri. Salah satu teka-teki jenis ini adalah *Numberfill*.

Setiap teka-teki dapat dipecahkan dengan mudah jika kita tahu bagaimana cara untuk memecahkannya. Cara-cara ini disebut algoritma, dan salah satu cara untuk memecahkan teka-teki *Numberfill* adalah dengan menggunakan algoritma *backtracking* atau runut Balik. Algoritma ini memungkinkan kita atau komputer menyelesaikan teka-teki ini dengan mudah.

Kata kunci: Backtracking, Puzzle, Numberfill.

1. Pendahuluan

Banyak permainan yang populer di dunia, salah satunya adalah jenis permainan teka-teki. Permainan ini digemari banyak orang karena dianggap lebih menantang dengan mengharuskan pemain berpikir lebih untuk memenangkan atau menyelesaikan permainan ini.

Teka-teki di dunia memiliki berbagai macam jenis dan bentuk. Salah satu jenis teka-teki yang terkenal adalah *Crossword* atau Teka-Teki Silang. Jenis teka-teki ini sering kita jumpai pada media cetak koran. Teka-teki ini mengharuskan kita mengisi setiap kotak kosong yang tersedia dengan kata-kata. Seiring berkembangnya waktu dan jaman, teka-teki ini pun memiliki variannya yaitu *Numberfill*.

Numberfill adalah teka-teki perkembangan dari Teka-Teki Silang yang mengganti kata-kata pada Teka-Teki Silang dengan angka-angka acak dan unik. Setiap angka pada teka-teki ini memiliki panjang mulai dari tiga angka hingga delapan sampai sepuluh angka. Tiap-tiap bilangan harus dimasukkan ke dalam kotak-kotak yang tersedia sesuai dengan panjang kotak tersebut. Biasanya ada dua atau tiga bilangan yang telah diisi terlebih dahulu sebagai bantuan.

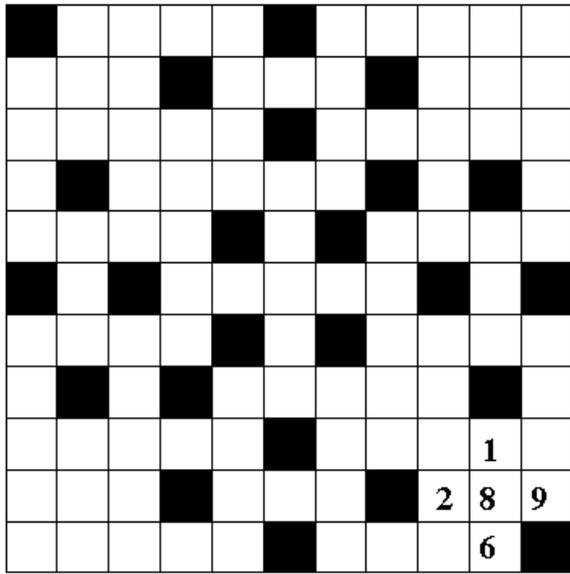
Salah satu cara untuk memenangkan atau menyelesaikan permainan ini adalah dengan menggunakan algoritma *backtracking* atau runut balik. Algoritma ini memungkinkan pemain menyelesaikan permainan ini dengan sempurna. Algoritma ini bekerja dengan cara mencoba mengisi salah satu bilangan ke dalam kotak sesuai dengan panjang kotak tersebut. Kemudian akan melanjutkan pengisian pada kotak yang tersambung dengan kotak yang baru saja diisi dengan bilangan yang sesuai. Jika tidak sesuai, maka akan dilakukan runut balik dengan cara mengganti bilangan yang tidak sesuai dengan bilangan lainnya yang sesuai.

2. Pencarian Solusi *Numberfill*

2.1. Teka-Teki *Numberfill*

Seperti yang telah dijelaskan di atas, Teka-Teki *Numberfill* ini memiliki bentuk seperti Teka-Teki Silang. Perbedaan drastisnya adalah pada benda yang diisi dan pada penomoran kotak. Teka-teki ini lebih menantang daripada Teka-Teki Silang biasanya karena penggunaan angka yang dapat digunakan tanpa urutan dan tetap memiliki arti. Angka-angka yang dapat diisikan telah ditentukan sebelumnya, sehingga kita tidak dapat mengisinya dengan bilangan acak.

Contoh tempat permainan Teka-Teki *Numberfill*:



Gambar 1: Teka-Teki *Numberfill*

Contoh daftar angka yang telah ditentukan sebelumnya pada Teka-Teki *Numberfill*:

3 Digits	9127
106	9473
254	9756
261	
269	5 Digits
209	12336
328	15142
353	21214
384	21421
423	27010
465	28174
925	33335
968	34025
	34117
4 Digits	38464
1356	43823
2239	59187
2536	73033
3320	77311
3697	77334
4213	82348
6609	88466
7248	96125
8473	

Gambar 2: Daftar Angka

2.2. Deskripsi Umum Algoritma Runut Balik

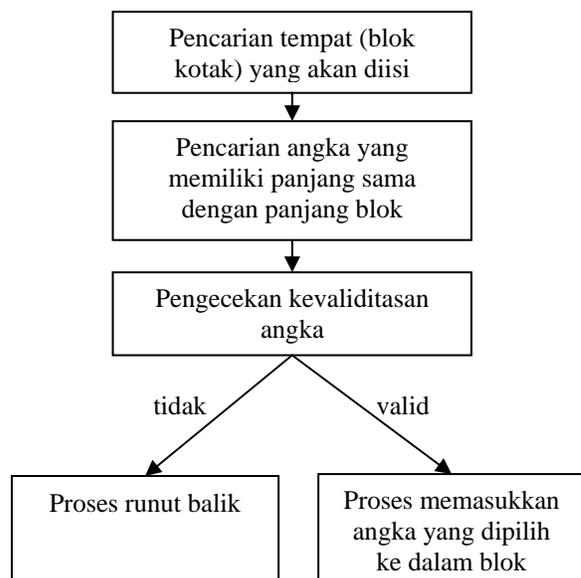
Algoritma *backtracking* atau runut balik pertama kali diperkenalkan oleh D.H. Lehmer pada tahun 1950. Dalam perkembangannya beberapa ahli seperti RJ Walker, Golomb, dan Baumert menyajikan uraian umum tentang runut balik dan penerapannya dalam berbagai persoalan dan aplikasi. Algoritma runut balik merupakan salah satu metode pemecahan masalah yang termasuk dalam strategi yang berbasis pencarian pada ruang status. Algoritma runut balik bekerja secara rekursif dan melakukan pencarian solusi persoalan secara sistematis pada semua kemungkinan solusi yang ada. Oleh karena algoritma ini berbasis pada algoritma *Depth-First Search (DFS)*, maka pencarian solusi dilakukan dengan menelusuri suatu struktur berbentuk pohon berakar secara preorder. Proses ini dicirikan dengan ekspansi simpul terdalam lebih dahulu sampai tidak ditemukan lagi suksesor dari suatu simpul.

2.3. Algoritma Runut Balik Sebagai Pemecahan Solusi

Penggunaan algoritma runut balik ini dapat terlihat pada proses pengisian yang dilakukan pada papan permainan ini di mana terdapat beberapa kemungkinan angka yang sesuai. Angka-angka yang aka dimasukkan dicari dan diambil yang sesuai dari daftar angka yang telah disediakan sebelumnya.

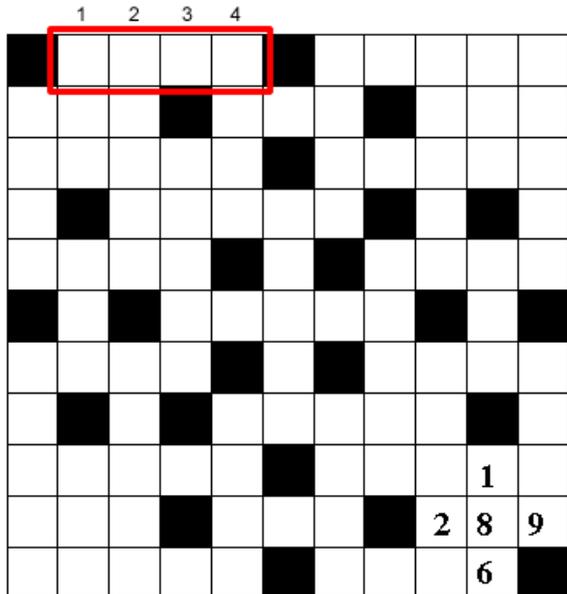
Metode pengisian papan bermain ini dapat dilakukan secara iteratif. Begitu pula halnya dengan pencarian dan pemilihan angka pada daftar angka.

Proses pencarian solusi ini berlangsung dalam beberapa tahap, yaitu:



Gambar 3: Tahap pencarian solusi

Secara spesifik, proses ini dimulai dengan mencari tempat (blok kotak) yang akan diisi dengan angka. Kemudian dilanjutkan dengan menghitung panjang dari blok tersebut.



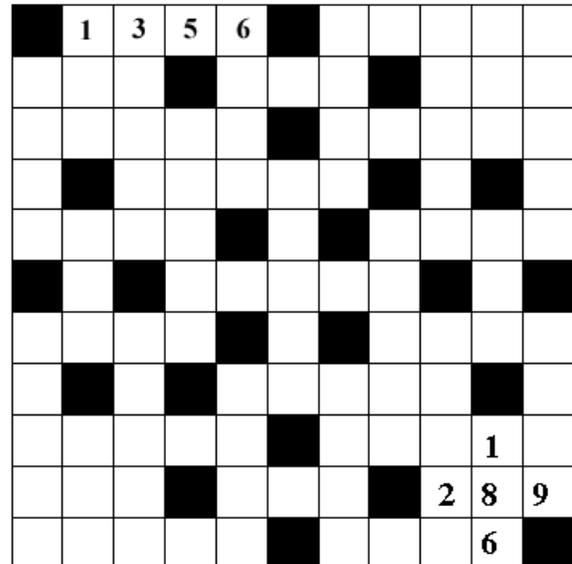
Gambar 4: Proses pemilihan dan perhitungan blok kotak

Setelah didapat blok kotak dan panjangnya, proses akan dilanjutkan dengan mencari angka yang panjangnya sesuai dengan panjang blok kotak (dalam hal ini panjangnya empat *digit*).

- 4 Digits**
1356
 2239
 2536
 3320
 3697
 4213
 6609
 7248
 8473
 9127
 9473
 9756

Gambar 5: Proses pencarian angka

Setelah itu akan dilakukan pengecekan kevaliditasan angka yang telah dipilih. Jika angkanya valid, maka proses akan dilanjutkan ke memasukkan angka yang telah dipilih dan angka tersebut akan ditandai dari daftar angka.

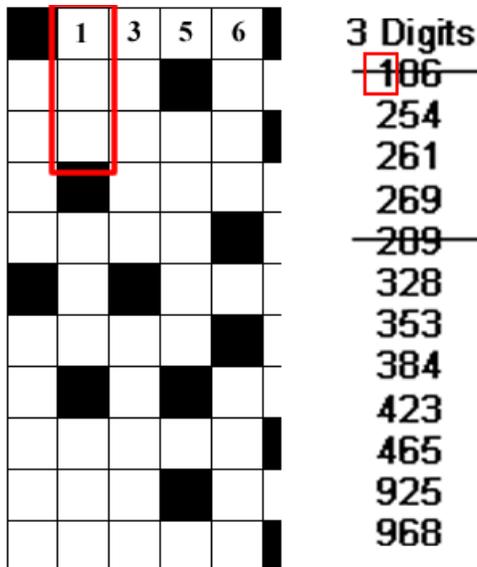


Gambar 6: Proses memasukkan angka

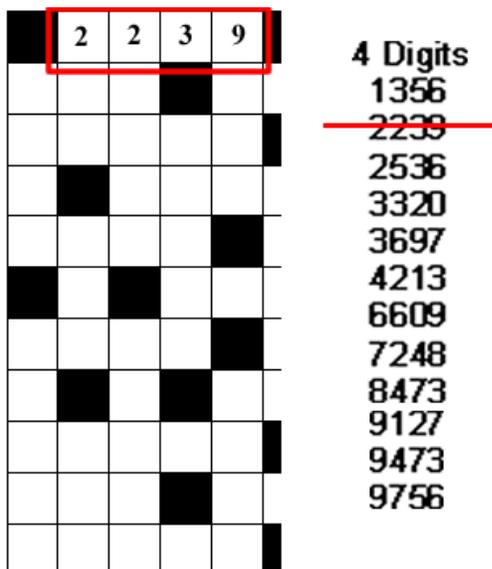
- 4 Digits**
~~1356~~
 2239
 2536
 3320
 3697
 4213
 6609
 7248
 8473
 9127
 9473
 9756

Gambar 7: Proses menandai angka yang telah digunakan

Jika tidak valid, maka akan dilakukan proses runut balik. Yaitu dengan mencari lagi angka yang masih belum ditandai dan memiliki panjang sesuai dengan panjang blok kotak. Jika sudah mencari secara iteratif sampai habis tidak ditemukan pula, maka ulangilah proses pencarian angka blok kotak sebelum blok kotak ini dan begitu seterusnya.



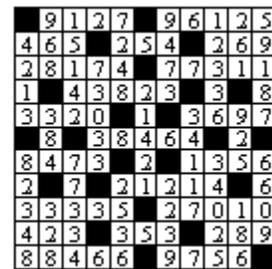
Gambar 8: Proses runut balik - 1



Gambar 8: Proses runut balik – 2

Kemudian setelah terisi oleh suatu angka, maka proses akan kembali ke pencarian blok kotak yang akan diisi dengan angka lainnya yang didapat dari daftar angka yang telah ditentukan sebelumnya. Namun untuk proses selanjutnya jumlah angka-angka yang dapat dimasukkan (valid) akan semakin berkurang.

Proses ini akan terus diulang hingga menemukan solusi dari Teka-Teki *Numberfill* ini atau hingga dinyatakan tidak memiliki solusi (jika terjadi runut balik pada blok kotak pertama yang diisikan).



Gambar 7: Solusi dari Teka-Teki *Numberfill*

3. Kesimpulan

Algoritma runut balik merupakan algoritma yang cukup mangkus untuk menyelesaikan berbagai persoalan. Hal ini disebabkan karena pada prinsipnya, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Pencarian hanya mengarah pada solusi yang dipertimbangkan saja. Oleh karena algoritma ini cukup mangkus, maka runut balik banyak diterapkan dalam berbagai program permainan dan persoalan yang berkaitan dengan bidang kecerdasan buatan (*artificial intelligence*). Hasil analisis kemampuan algoritma runut balik dalam menyelesaikan persoalan pengisian Teka-Teki *Numberfill* menunjukkan bahwa algoritma ini cukup mangkus untuk mendapatkan solusi persoalan tersebut. Sistem kerja algoritma runut balik yang sistematis dan ciri khasnya yang hanya memeriksa kemungkinan solusi yang memang dapat dipertimbangkan untuk menjadi solusi akhir, diperkirakan dapat menjadi solusi yang efektif dan mangkus untuk persoalan ini. Sebenarnya, sebagai sesuatu yang cukup baru, penyelesaian persoalan pengisian Teka-Teki *Numberfill* ini belum dapat dianalisis untuk berbagai algoritma. Namun melalui studi literatur dan analisis yang telah penulis lakukan, penulis berpendapat bahwa algoritma runut balik cukup efektif untuk menyelesaikan persoalan ini.

REFERENSI

- [1] Munir, Rinaldi. 2005. *Diktat Kuliah IF2251 Strategi Algoritmik*. Program Studi Teknik Informatika ITB: Bandung.
- [2] Number And Word Puzzles. <http://users.tpg.com.au/users/puzzles/page12.html>