

ANALISIS TEKNIK PENGISIAN KATA DALAM PERMAINAN TEKA-TEKI SILANG

Alsasian Atmopawiro (13505074)

Laboratorium Ilmu dan Rekayasa Komputasi
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika, Institut Teknologi Bandung
Jl. Ganesha 10, Bandung
e-mail: hokage6th@yahoo.com

ABSTRAK

Pada zaman ini, permainan-permainan yang menggunakan karakter semakin banyak. Tidak semua menganggap permainan tersebut mudah. Oleh karena itu, beberapa orang mencoba menyelesaikan permainan tersebut menggunakan teknik komputasi. Adakalanya suatu penyelesaian membutuhkan lebih dari satu algoritma untuk menyelesaikannya. Teka-teki silang adalah teka-teki kata yang membentuk suatu kotak/matriks yang terdiri dari blok hitam dan putih. Tujuan dari permainan ini adalah mengisi kotak putih dengan huruf sehingga membentuk suatu kata baik mendatar atau menurun dengan bantuan yang ada. Dalam metoda analisis yang akan kita pakai, kita membutuhkan sebuah algoritma yang akan mengisi kata-kata dalam basisdata kedalam suatu blok kata dalam kotak/matriks. Algoritma biasa digunakan dalam pencarian solusi adalah *brute force*—algoritma yang paling kuat, dan algoritma *backtracking*. Dengan optimasi yang juga diterapkan dalam algoritma, kita dapat membentuk metoda-metoida penyelesaian/pencarian solusi yang akan kita analisis. Analisis dilakukan bergantung kepada suatu kriteria yang terdapat pada permainan teka-teki. Seperti jumlah huruf dalam kotak kata atau keterkaitan antar kata. Diharapkan teknik ini menjadi dasar penelitian lebih lanjut.

Kata kunci: Algoritma *Brute Force*, Algoritma *Backtracking*, Teka-teki silang.

1. PENDAHULUAN

Pada zaman ini, permainan-permainan yang menggunakan karakter semakin banyak. Contoh permainan tersebut adalah *scramble*, teka-teki silang, dan anagram.

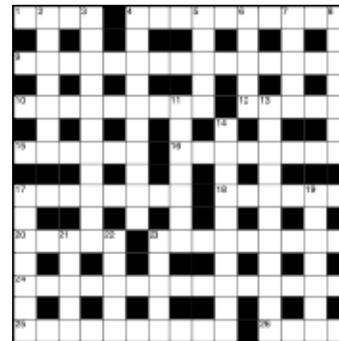
Tidak semua menganggap permainan tersebut mudah. Oleh karena itu, beberapa orang mencoba menyelesaikan permainan tersebut menggunakan teknik komputasi.

Namun timbul permasalahan pada teknik penyelesaiannya. Dengan permainan yang berbeda, teknik dalam penyelesaiannya juga berbeda. Banyak teknik yang dapat kita gunakan untuk menyelesaikan permainan kata itu. Yang akan kita tuju adalah pencarian teknik penyelesaian yang memiliki tingkat efektifitas paling baik.

Adakalanya suatu penyelesaian membutuhkan lebih dari satu algoritma untuk menyelesaikannya. Contohnya algoritma runut-balik digabung dengan pencarian kata dalam pola.

2. ANALISIS TEKA-TEKI SILANG

2.1. Teka-Teki Silang



Gambar 1. Gambar teka-teki silang gaya Amerika. [3]

Teka-teki silang adalah teka-teki kata yang membentuk suatu kotak/matriks yang terdiri dari blok hitam dan putih. Tujuan dari permainan ini adalah mengisi kotak putih dengan huruf sehingga membentuk suatu kata baik mendatar atau menurun dengan bantuan yang ada. Dalam bahasa yang tertulis dari kiri ke kanan, jawaban kata di simpan dalam blok dari kiri kekanan (mendatar) dan dari atas ke bawah (menurun). Fungsi dari kotak hitam itu sendiri adalah untuk membatasi kata.

Di setiap awal kotak kata, biasanya akan diberi nomor yang menunjukkan urutan soal pada permainan teka-teki

silang ini. Angka ini merujuk kepada bantuan yang diberikan untuk diselesaikan, contohnya 3 mendatar atau 12 menurun.

2.2. Batasan Analisis

Dalam analisis penyelesaian teka-teki menggunakan komputasi ini terdapat batasan-batasan yang harus dipenuhi sebelumnya. Yang pertama, kotak/matriks yang akan kita isi dapat kosong ataupun sudah terisi.

Kemudian yang kedua, karena kita menyelesaikan menggunakan komputer, kita membutuhkan basisdata yang menampung kata-kata yang akan diisikan kedalam kotak/matriks. Dalam analisis ini, kita akan menggunakan bahasa Indonesia sebagai kata kunci dan basisdata yang kita gunakan sangat ideal—semua kata terdapat di dalamnya.

Hasil yang akan kita dapatkan adalah solusi yang mungkin untuk menyelesaikan teka-teki silang tersebut. Jika terdapat n kata yang harus di isi maka solusinya akan membentuk n -tuple solusi, X_n dan kumpulan solusi dilambangkan dengan himpunan H_m :

$$x_n = \{k_1, k_2, \dots, k_n\} \quad (1)$$

dan

$$H_n = \{x_1, x_2, \dots, x_m\} \quad (2)$$

n adalah jumlah kata yang harus diisi, yaitu a kata mendatar ditambah b kata menurun ($n=a+b$). k_n adalah kata yang dimasukkan ke dalam kata pada himpunan solusi ke- n . x_n adalah solusi pencarian. H_n adalah himpunan dari solusi yang mungkin untuk digunakan dalam teka-teki.

2.3. Dasar Teori

Dalam metoda analisis yang akan kita pakai, kita membutuhkan sebuah algoritma yang akan mengisi kata-kata dalam basisdata kedalam suatu blok kata dalam kotak/matriks. Algoritma yang dapat kita gunakan adalah algoritma *brute-force* dan algoritma runut-balik.

Brute force adalah algoritma yang lempang untuk memecahkan suatu masalah, biasanya didasarkan langsung pada pernyataan masalah (*problem statement*) dan definisi konsep yang dilibatkan [1].

Algoritma *brute-force* mencari semua kemungkinan solusi yang dapat dibentuk. Kemudian setelah terbentuk semua solusi, algoritma ini akan mengecek kelayakan dari setiap solusi yang dibentuk dan terbentuklah suatu ruang solusi.

Runut-balik (*backtracking*) adalah algoritma yang berbasis pada *DFS* untuk mencari solusi persoalan secara lebih mangkus [2]. Algoritma ini dapat dikatakan perbaikan dari algoritma *brute force*. Algoritma ini

mencari secara sistematis kemungkinan solusi dengan melakukan pencarian yang mengarah ke solusi. Sebagai akibatnya, waktu pencarian dapat jauh berkurang dibanding *brute force*.

Ada beberapa property yang dimiliki oleh algoritma runut-balik yaitu:

1. Solusi persoalan, dinyatakan sebagai n -tuple vector.
2. Fungsi pembangkit isi solusi tadi.
3. Fungsi pembatas. Fungsi ini mengecek apakah solusi yang diberikan valid atau tidak.

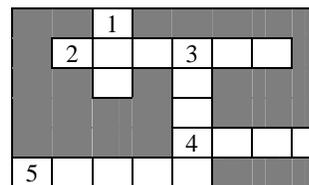
Untuk memfasilitasi pencarian tadi, maka ruang solusi diorganisasikan ke dalam struktur pohon.

2.4. Metoda Analisis

Dalam analisis teka-teki silang ini, metoda analisis untuk penyelesaiannya dibagi menjadi dua bagian, cara lempang dan cara runut balik, sesuai judulnya metoda-metoda ini menggunakan algoritma *brute force* atau *backtracking*. Dari metoda-metoda ini, kita akan menjabarkan teknik yang akan digunakan untuk mengisi teka-teki silang. Kemudian kita menganalisis bagaimana tersebut bekerja.

2.5. Metoda : Brute Force

Dengan contoh teka-teki silang dengan spesifikasi seperti pada gambar 2 dan basisdata pada tabel 1, kita akan menganalisis teknik-teknik pengisian teka-teki silang.



Gambar 2. Teka-teki Silang

Dengan metoda ini maka kita akan menggunakan algoritma *brute force*. Pertama-tama kita bentuk solusi dari kata-kata yang terdapat didalam basisdata. Solusi akan membentuk 5-tuple solusi. Solusi tersebut berupa himpunan, jadi tidak mungkin ada dua kata yang sama di dalamnya. Jika kita hitung total solusinya maka ada $18 \times 17 \times 16 \times 15 \times 14 = 10281160$ total solusi. Sungguh total yang sangat besar jika kita hitung sendiri. Namun ada suatu yang perlu diperhatikan bahwa, solusi tersebut didapat tanpa memperhatikan jumlah huruf dengan tempat yang mungkin.

Jika kita membuat fungsi yang membatasi solusi yaitu dengan mengambil kata yang mempunyai jumlah huruf yang sesuai maka total solusinya akan menjadi $4 \times 4 \times 7 \times$

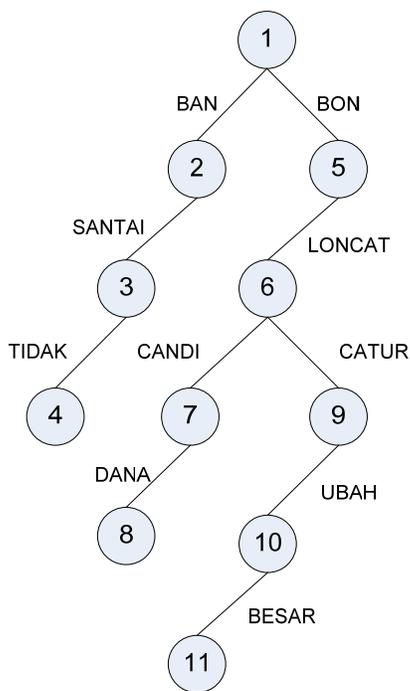
$6 \times 3 = 2016$ solusi. Total yang kita dapat memang jauh lebih sedikit namun cukup memakan waktu dan memori.

Tabel 1. Basisdata kata-kata

3 huruf	4 huruf	5 huruf	6 huruf
Ban	Dana	Besar	Dengan
Bon	Kita	Candi	Loncat
Tas	Kini	Catur	Santai
Dan	Ubah	Damar	
		Dodol	
		Getir	
		Tidak	

2.5. Metoda : Backtraking

Metoda ini akan memangkas solusi-solusi yang tidak mungkin lebih awal. Masalah yang kita hadapi sekarang adalah pemilihan akar pohon yang mungkin. Pemilihan akar akan dibagi-bagi menurut kemungkinan cepatnya penyelesaian teka-teki silang. Untuk pemilihan kata dari basisdata, akan dipilih yang mempunyai jumlah huruf yang sama dan dimulai secara *alphabetic*.



Gambar 3. Gambar pohon pertama

Pemilihan Pertama adalah mengisi dari nomor satu dan lanjut ke kata yang terhubung dengan kata pertama tadi. Pemilihan ini didasarkan kepada kebiasaan beberapa orang mengisi kata pada teka-teki silang yang sudah

terdapat huruf minimal satu buah. Dengan dasar ini kita akan membuat komputasinya.

Kita akan mengambil contoh teka-teki tadi. Maka akan terbentuk pohon dinamis seperti pada gambar 3. Langkah yang ditempuh hanya sampai 10 langkah untuk mencapai solusi pertama. Sangat efektif dibandingkan dengan algoritma *brute force*.

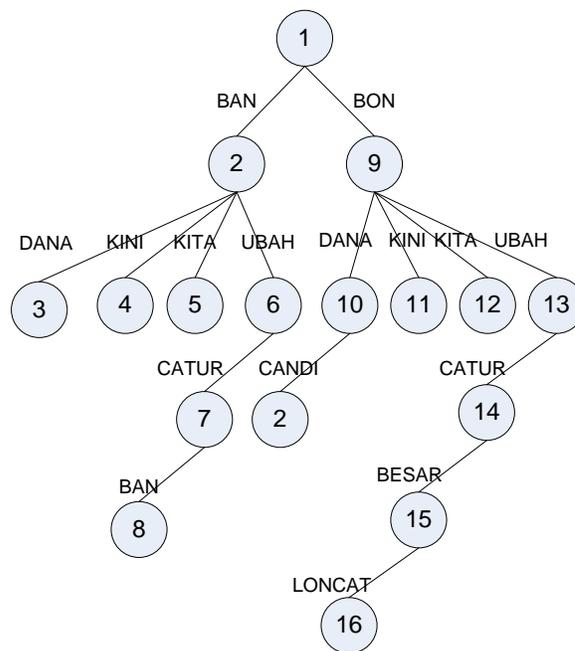
Pemilihan Kedua adalah memilih urutan dari kata yang paling sedikit hurufnya ke kata yang paling banyak hurufnya. Jika terdapat kata dengan jumlah huruf yang sama maka diambil kata yang mempunyai no soal paling kecil. Pemilihan ini hanya didasarkan kepada jumlah huruf yang terdapat pada kotak kata.

Data-data mengenai langkah-langkah harus disimpan kedalam sebuah larik agar kita tahu kata mana yang harus dipilih terlebih dahulu seperti pada tabel 2.

Tabel 2. Larik urutan kata Pemilihan Kedua

1	4	3	5	2
---	---	---	---	---

Larik tersebut berarti kita akan memulai pencarian kata dari kotak nomor 1 kemudian 4 dan seterusnya berdasarkan jumlah huruf pada kotak tersebut. Dari pembentukan pohon dinamisnya, didapat perhitungan yang membutuhkan 18 langkah sebelum akhirnya menemukan solusi pertama.



Gambar 4. Gambar pohon kedua

Pemilihan Ketiga ini mirip dengan pemilihan kedua, hanya saja pemilihan ketiga ini memilih urutan kata dari

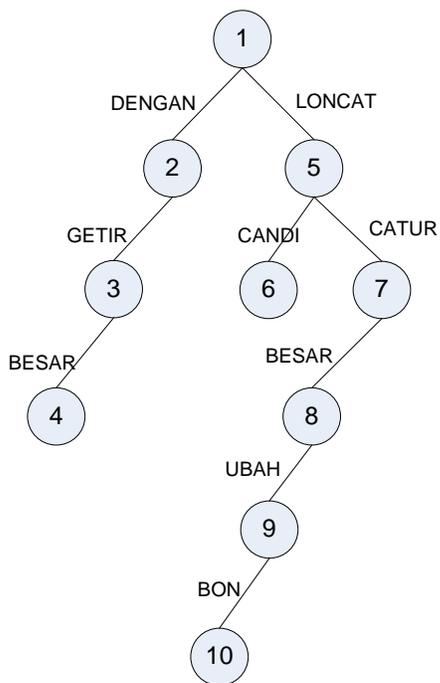
yang jumlah hurufnya paling banyak. Jika terdapat huruf yang sama akan diambil huruf yang mempunyai no soal paling kecil. Larik yang akan dibentuk hanya perlu kita urut ulang seperti pada tabel 3.

Tabel 3. Larik urutan kata Pemilihan Ketiga

2	3	5	4	1
---	---	---	---	---

Pohon yang dibentuk akan seperti gambar 5. Terlihat perbedaannya dengan pemilihan kedua bahwa pohon yang dibentuk lebih kecil dengan langkah yang ditempuh sebelum mencapai solusi adalah 9 langkah.

Dengan pemilihan ketiga ini akan kita lihat perbedaan pohannya dengan pemilihan kedua. Dapat kita lihat bahwa langkah yang ditempuh antara kedua cara sebelumnya berbeda jauh. Namun jika kita perhatikan lebih jauh, hal ini (langkah sedikit) disebabkan pada cara kedua (Pemilihan ketiga), pada langkah awal, kotak kata yang dicari setelah kata "LONCAT" memiliki satu huruf yang berasal dari pencarian sebelumnya ('C'), sehingga pemilihan katanya akan lebih mudah dalam basisdata. Sedangkan pada Pemilihan Kedua, pencarian kata selanjutnya (kotak nomor 4) memiliki kotak kosong.



Gambar 5. Gambar pohon ketiga

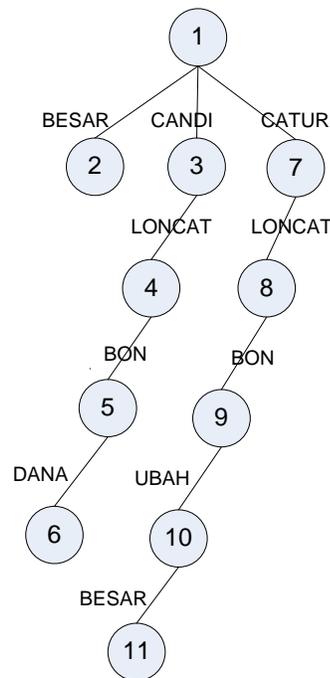
Dari metoda analisis tadi, kita akan mencoba membuat suatu pemilihan yang akan mengisi kotak yang terhubung dengan beberapa kotak lain terlebih dahulu, dengan tujuan pencarian solusi yang lebih cepat.

Pemilihan Keempat adalah memilih kotak yang memiliki huruf yang terhubung dengan kotak lain paling banyak. Dengan contoh di atas, kita dapat membentuk larik pengecekan *backtracking* yang dapat dilihat pada tabel 4. Kata nomor tiga mempunyai 3 huruf yang berhubungan dengan kotak lain yaitu kotak 2, 4, dan 5. Jika terdapat kotak dengan jumlah terhubung yang sama maka akan diurutkan berdasarkan nomor soal.

Tabel 4. Larik urutan kata Pemilihan Keempat

3	2	1	4	5
---	---	---	---	---

Dari pemilihan tersebut kita dapat pohon dinamis seperti yang ditunjukkan oleh gambar 6. Langkah yang harus kita tempuh adalah sebanyak 10 langkah. Jumlah yang cukup sedikit kita dapatkan dari pemilihan ini.



Gambar 6. Gambar pohon keempat

Pemilihan Kelima ini adalah kebalikan dari pemilihan keempat—memilih kotak yang memiliki huruf terhubung paling kecil. Pemilihan ini bias disebut ekstrem karena dapat kita bayangkan apa yang akan terjadi—*tracking* yang sangat banyak. Hal ini dapat terjadi karena kejadian terhubungnya suatu kotak (pencocokan) akan terjadi pada aras pohon bagian bawah (kasus terburuk dapat mencapai daun kemudian kembali lagi keakar terus-menerus). sehingga pencarian solusi juga akan bertambah lama.

Larik yang akan dibentuk bermula dari kata yang mempunyai huruf terhubung paling sedikit, jika ada yang

sama diurut berdasarkan nomor soal seperti pada tabel 5. Langkah yang ditempuh lebih dari 70 langkah. Gambar dari pohon tidak disertakan karena mengambil lahan yang cukup luas.

Tabel 5. Larik urutan kata Pemilihan Kelima

1	4	5	2	3
---	---	---	---	---

2.5. Analisis : *Backtraking*

Jika kita lihat secara kasar maka metoda penyelesaian yang mempunyai langkah yang lebih sedikit adalah cara dari kotak kata yang mempunyai huruf paling banyak ke kotak kata yang mempunyai huruf yang paling sedikit.

Untuk pemilihan pertama, bisa dikatakan metoda ini termasuk metoda yang bodoh untuk dilakukan secara komputasi, karena hanya melakukan runut balik biasa dengan mengecek dari nomor urut soal dan tidak menggunakan optimasi tertentu.

Pemilihan selanjutnya menggunakan optimasi yang dilakukan berdasarkan kriteria-kriteria tertentu. Kriteria tersebut adalah berdasarkan jumlah huruf pada kotak yang akan ditebak, banyaknya huruf pada kotak yang terhubung dengan kotak lain. Setiap kriteria tersebut juga dilakukan secara terbalik sehingga kita bisa melihat keadaan teka-teki silang ketika suatu kriteria dilakukan secara ekstrem.

Pemilihan optimasi tersebut ternyata sedikit melenceng dari perkiraan karena ada pohon yang ternyata lebih buruk dari metoda pemilihan pertama. Sehingga pada akhirnya kita tidak dapat melihat metoda yang terbaik karena kondisi papan menjadi suatu kendala yang terkadang baik untuk suatu metoda namun tidak untuk metoda yang lain.

3. KESIMPULAN

Kesimpulan yang dapat kita dapat adalah bahwa sangat jelas bahwa algoritma *brute force* sangat buruk dalam bidang waktu ataupun ruang. Untuk menanganinya digunakan algoritma lain seperti algoritma runut balik yang memotong kemungkinan solusi yang tidak valid.

Selain algoritma, kita juga membutuhkan suatu optimasi agar suatu penyelesaian dapat menemukan solusi lebih cepat. Namun, metoda yang dilakukan diatas kurang dapat dijadikan sebagai kesimpulan karena kurangnya contoh dan analisis yang kurang mendalam akibat waktu yang kurang pula.

Dengan mengembangkan metoda tadi, kita dapat menggali tekniknya lebih dalam. Alangkah baiknya jika dicari metoda lain seperti memperhitungkan faktor basisdata yang terdapat pada permainan atau penggunaan *artificial intelligence* yang diterapkan oleh beberapa kelompok penganalisa permainan ini.

REFERENSI

- [1] Levitin, Anany, *Introduction to The Design and Analysis of Algorithms*, Addison-Wesley, 2003.
- [2] Munir, Rinaldi, *Strategi Algoritmik*, 2007
- [3] Wikipedia. *Crossword – Wkipedia, the Free Encyclopedia*. <http://en.wikipedia.org/wiki/Crossword>. diakses pada tanggal 19 Mei 2007.
- [4] Widya Wardani, Dian Intania Savitri H, Allentine Tanujaya, *Analisis Penerapan Algoritma Backtracking Dalam Pencarian Solusi Game “Crossword Puzzle”*, Laboratorium Ilmu dan Rekayasa Komputasi. 2005.