Semantika
Seminar Nasional Teknologi Informatika

Penerapan Algoritma Backtracking Dalam Game Pencaria Kata Menggunakan Aksara Batak Toba

Melva Lumban Gaol
Magister Teknik Informatika, Universitas Sumatera Utara
naipospos melva@yahoo.co.id

Kristin L Sitompul
Magister Teknik Informatika, Universitas Sumatera Utara
sriwijaya11121987@gmail.com

Muhammad Zarlis
Magister Teknik Informatika, Universitas Sumatera Utara
m.zarlis@usu.ac.id

Abstrak

Game **merupakan bagian tak terpisahkan dari keseharian manusia.** ***Game*** **adalah aktifitas yang dapat membawa kesenangan atau hiburan bagi penggunanya.** ***Game*** **sering dipakai dalam kecerdasan buatan. Kecerdasan buatan adalah bagian dari komputer sains yang mempelajari atau merancang sistem komputer yang berintelegensi, yaitu sistem yang memiliki karakteristik berpikir seperti manusia. Algoritma yang digunakan untuk mengimplementasikan kecerdasan buatan begitu banyak. Salah satunya adalajh algoritma Runut Balik (*Backtracking*). Algoritma Runut Balik (*Backtracking*) sering digunakan beberapa game populer semisal *Sudoku, Labyrinth, Catur, Chess* dan lain sebagainya. Adapun game yang menggunakan algoritma ini adalah game pencarian kata dengan aksara batak toba. Algoritma Runut Balik (*Backtracking*) dalam program ini untuk mengecek kotak jawaban yang telah disediakan. Program akan mengecek apakah terdapat kata layak dalam data base untuk dimasukkan kedalam kotak-katik permainan.**

Kata kunci ***:*** **Kecerdasan buatan*****,*** ***Backtracking,*** **Aksara Batak Toba.**

Semantika
Seminar Nasional Teknologi Informatika

I. PENDAHULUAN

A. Latar belakang

Aksara Batak Toba yang merupakan rumpun dari aksara batak yang perlu mendapat perhatian khusus karena terancam punah akibat keterbatasan data dan informasi. Penyebab keterbatasan tersebut yaitu dulu sastra diturunkan hanya secara lisan.

Seiring dengan kemajuan teknologi tersebut maka kebudayaan tersebut juga harus beradaptasi dengan sifat dan manusia dalam melestarikan kebudayaan daerah tersebut salah satunya adalah dengan mengaplikasikan Aksara batak toba dalam sebuah permainan. Oleh sebab itu penulis mendapat sebuah ide untuk menggabungkan kebudayaan Batak Toba (aksara batak toba) dengan teknologi yang akan menghasilkan sebuah game pembelajaran yang menarik yaitu permainan pencari kata. Untuk menemukan kata-kata yang diacak menggunakan algoritma runut balik (backtracking). Penggunaan algoritma runut balik (backtracking) untuk membangkitkan setiap suku kata dalam game sehingga menjadi satu kata yang utuh. Algoritma runut balik (backtracking) itu sendiri adalah algoritma yang digunakan untuk mencari solusi persoalan secara lebih mangkus daripada algoritma brute force. Algoritma ini akan mencari solusi berdasarkan ruang solusi akan diperiksa, hanya pencarian yang mengarah kepada solusi yang akan diproses.

II. LANDASAN TEORI

A. Aksara

Aksara (*surat*) Batak termasuk keluarga tulisan India. Aksara India yang tertua adalah aksara Brahmi yang menurunkan dua kelompok tulisan, yakni kelompok India Utara (Aksara Nagari) dan kelompok India Selatan (Aksara Palawa). Aksara Palawa paling berpengaruh di Indonesia dan tulisan asli Indonesia berinduk pada aksara tersebut. Kerabat aksara Batak yang paling dekat adalah aksara-aksara Nusantara (aksara-aksara turunan India yang terdapat di kepulauan Asia Tenggara), khususnya yang di Sumatera. Urutan yang diketahui selama ini dan sering dipakai di sekolah adalah A-HA-NA-RA-TA-BA-WA-I-MA-NGA-LA-PA-SA-DA-GA-JA. Urutan ini adalah ciptaan baru dan tidak memiliki landasan tradisional. Urutan tersebut digunakan agar mudah untuk diingat dalam bentuk kalimat “aha na rata bawa imangalapa sada gaja” yang artinya “apa yang hijau lelaki itu memotong seekor gajah”.

a. Ina Ni Surat

Ada 19 *ina ni surat*, *ina ni surat* tersebut adalah: **A, Ha, Ma, Na, Ra, Ta, Sa, Da, Ga, Ja, La, Pa, Ba, NGa, NYa, Wa, Ya, I dan U.** *Ina ni surat* (*ina* = ibu) terdiri dari huruf-huruf silabik dasar yang diakhiri bunyi /a/ (kecuali untuk huruf i dan u) seperti yang ditunjukkan tabel .

Tabel 1. Huruf-huruf *Ina Ni Surat* dan variannya

a	varian	ta			
ha/ka		ba			
na		wa	varian		
ra		ma	varian		
nga		da		nya	
la		ga		i	
pa	varian	ja		u	
sa		ya			

(Sumber : Uli Kozok, 2009)

b. Anak Ni Surat

Semua ina ni surat berakhir dengan bunyi /a/. Bunyi ini dapat diubah dengan menambah nilai fonetisnya. Pengubah ini disebut diakritik. Diakritik di dalam anak ni surat sebagai berikut (**Sitinjak, 2012**):

1. Bunyi /e/ (pepet/keras) disebut ‘hatadingan’, dengan menambah garis kecil disebelah kiri atas *ina ni surat*, contoh :

   /pa/ ppe pe

   /ba/ bbe be

   /ga/ gge ge

2. Bunyi /ng/ disebut ‘paninggil’, dengan menambah garis kecil disebelah kanan atas *ina ni surat*, contoh :

   /pa/ ppang p^

   /ba/ bbang b^

   /ga/ ggang g^

3. Bunyi /u/ disebut ‘haborotan’ disebelah bawah *ina ni surat*, contoh :

   /pa/ ppu P

   /ba/ bbu B

   /ga/ ggu G

4. Bunyi /i/ disebut ‘hauluan’ bentuk lingkaran kecil setelah *ina ni surat*, contoh:

/pa/ ppi pi

/ba/ bbi bi

/ga/ ggi gi

5. Bunyi /o/ disebut 'sihora' atau 'siala' berupa tanda kali setelah *ina ni surat*, contoh:

/pa/ ppo po

/ba/ bbo bo

/ga/ ggo go

B. Algoritma Runut Balik

Algoritma runut balik pertama kali diperkenalkan oleh D.H Lehmer pada tahun 1950. Algoritma ini cukup mangkus untuk digunakan dalam beberapa penyelesaian masalah dan juga untuk memberikan kecerdasan buatan dalam *game*. Beberapa *game* populer semisal Sudoku, Labirin, Catur juga bisa diimplementasikan dengan menggunakan algoritma runut balik. Algoritma runut balik (*backtracking*) merupakan algoritma yang digunakan untuk mencari solusi persoalan secara lebih mangkus daripada menggunakan algoritma *brute force.*

Algoritma ini akan mencari solusi berdasarkan ruang solusi yang ada secara sistematis namun tidak semua ruang solusi akan diperiksa, hanya pencarian yang mengarah kepada solusi yang akan diproses.

Algoritma runut balik berbasis pada DFS (*Depth First Search*) sehingga aturan pencariannya akan mengikut kepada aturan pencarian DFS yaitu dengan mencari solusi dari akar ke daun (dalam pohon ruang solusi) dengan pencarian mendalam. Simpul-simpul yang sudah dilahirkan (diperiksa) dinamakan simpul hidup (*live node*). Simpul hidup yang sedang diperluas dinamakan simpul-E atau *Expand Node.*

Algoritma *backtracking* mempunyai prinsip dasar yang sama seperti *brute-force* yaitu mencoba segala kemungkinan solusi. Perbedaan utamanya adalah pada ide dasarnya, semua solusi dibuat dalam bentuk pohon solusi (pohon ini tentunya berbentuk abstrak) dan algoritma akan menelusuri pohon tersebut secara *DFS* (*depth field search)* sampai ditemukan solusi yang layak.

1. Properti Umum Metode Runut balik (*Backtracking*)

   Untuk menerapkan metode runut-balik, properti berikut didefinisikan (**Amin, 2007**) :

   a. Solusi persoalan.

      Solusi dinyatakan sebagai vektor n-tuple:

      X=(x1, x2, ..., xn), xi anggota himpunan berhingga Si .

      Mungkin saja S1 = S2 = ... = Sn.

      Contoh: Si = {0,1}

      Si = 0 atau 1

   b. Fungsi pembangkit nilai xk

      Dinyatakan sebagai:

      T(k)

      T(k) membangkitkan nilai untuk xk, yang merupakan komponen vektor solusi

   c. Fungsi Pembatas (fungsi kriteria)

      Dinyatakan sebagai:

      B(x1, x2, ..., xk)

      Fungsi pembatas menentukan apakah (x1, x2, ..., xk) mengarah ke solusi. Jika ya, maka pembangkitan nilai untuk xk+1 dilanjutkan, tetapi jika tidak, maka (x1, x2, ..., xk) dibuang dan tidak dipertimbangkan lagi dalam pencarian solusi.

2. Pengorganisasian Solusi

Semua kemungkinan solusi dari persoalan disebut ruang solusi (*solution space*).

Jika $x_i \in S_i$, maka $S_1 \times S_2 \times \ldots \times S_n$ disebut ruang solusi. Jumlah anggota di dalam ruang solusi adalah $|S_1| \cdot |S_2| \cdot \ldots \cdot |S_n|$. Tinjau persoalan *Knapsack* 0/1 untuk $n = 3$. Solusi persoalan dinyatakan sebagai vektor (x_1, x_2, x_3) dengan $x_i \in \{0,1\}$.

Ruang solusinya adalah

$\{0,1\} \times \{0,1\} \times \{0,1\}$ = {(0, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), (1, 1, 1)}.

Pada persoalan *Knapsack* 0/1 dengan $n = 3$ terdapat $2^n = 2^3 = 8$ kemungkinan solusi, yaitu: (0, 0, 0), (0, 1, 0), (0, 0, 1), (1, 0, 0), (1, 1, 0), (1, 0, 1), (0, 1, 1), dan (1, 1, 1).

Penyelesaian secara *exhaustive search* adalah dengan menguji setiap kemungkinan solusi. Ruang solusi diorganisasikan ke dalam struktur pohon. Tiap simpul pohon menyatakan status (*state*) persoalan, sedangkan sisi (cabang) dilabeli dengan nilai-nilai x_i. Lintasan dari akar ke daun menyatakan solusi yang mungkin. Seluruh lintasan dari akar ke daun membentuk ruang solusi. Pengorganisasian pohon ruang solusi diacu sebagai pohon ruang status (*state space tree*).

III.MODEL DAN ANALISIS PERANCANGAN

1. Tahap Perancangan

Semantika
Seminar Nasional Teknologi Informatika

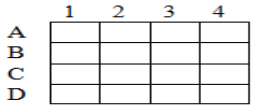
Berikut ini akan dijelaskan bagaimana gambaran dari komputer dalam menentukan langkah terbaik yang dapat meminimalkan langkah pengguna pada kasus. Agar mudah dipahami , maka ditempatkan pada sebuah papan catur yang berukuran NxN. Langkah pada sebuah papan catur dapat bergerak dalam petak horizontal, vertical dan diagonal yang dijelaskan pada tabel 2 :

		x			x		
x		x		x			
	x	x	x				
x	x	S	x	x	x	x	X
	x	x	x				
x		x		x			
		x			x		
		x				x	

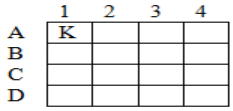
Petak yang bisa dilalui oleh S diberi tanda “x”. Diberikan sebuah papan yamh berisikan huruf-huruf yang berukuran 4x4 dengan menggunakan *backtracking*. . Huruf pertama bisa diletakkan diawal, tengah maupun diakhir. Misalkan huruf pertama diletakkan pada posisi awal yakni posisi bebas (misalkan pada kotak 1, maka huruf kedua harus dicari melewati semua kotak yang ada di sekitar kotak 1 dan harus kembali ke posisi awal yakni posisi kotak 1 dan begitu juga huruf yang selanjutnya yang akan dicari. Persoalan dalam menemukan kata yang tersembunyi ini merupakan permainan kecerdasan logika dimana pemain harus memikirkan strategi dalam setiap langkah.

Langkah – langkah pemecahan masalah akan diperjelaskan sebagai berikut :

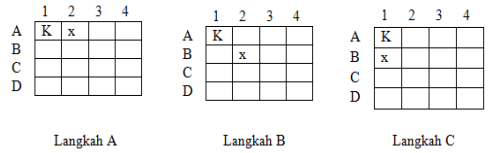
a. Terdapat papan berukuran 4x4



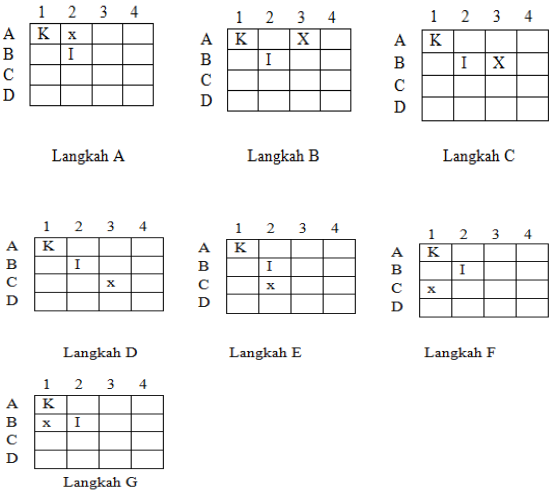
b. Soal yang dicari,misalkan: KITA
c. Huruf 1 terletak pada A1



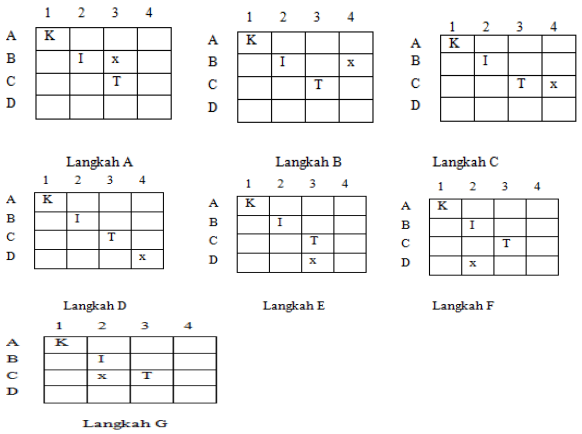
d. Satu huruf sudah ditemukan, selanjutnya mencari huruf kedua. Langkah yang memungkinkan adalah :



e. Huruf kedua ternyata ditempatkan di B2, maka dari itu untuk langkah yang memungkinkan adalah langkah B. Lanjutkan dengan mencari huruf ke 3, langkah yang memungkinkan adalah :



f. Huruf ketiga ternyata ditempatkan di C3, maka dari itu untuk langkah yang memungkinkan adalah langkah D. Lanjutkan dengan mencari huruf ke 4, langkah yang memungkinkan adalah :



g. Huruf keempat ternyata ditempatkan di D4, maka dari itu untuk langkah yang memungkinkan adalah langkah D. Lanjutkan dengan mencari huruf ke 4.
h. Jadi, solusi untuk pemecahan masalah adalah A1, B2, C3, D4.

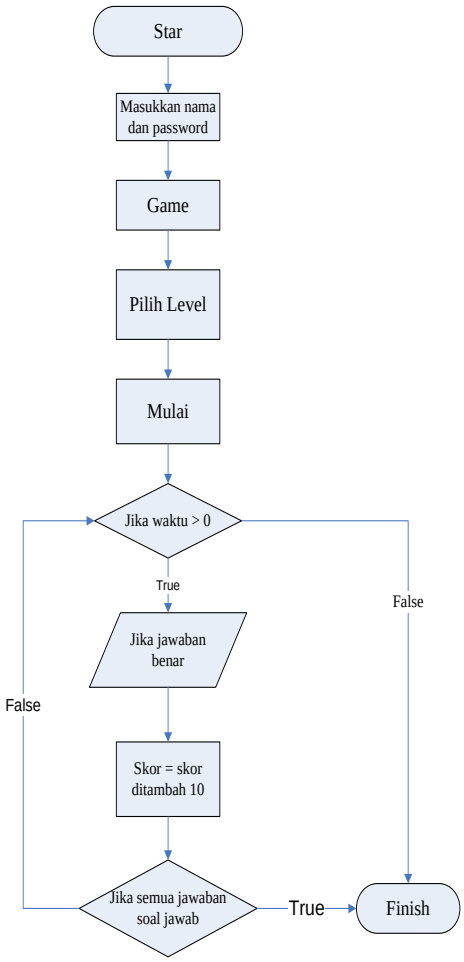
Semantika
Seminar Nasional Teknologi Informatika

	1	2	3	4
A	K			
B		I		
C			T	
D				A

Jadi, solusi untuk pemecahan masalah adalah A1, B2, C3, D4.

A. Flowchart Game Pencarian Kata dengan Aksara Toba

Flowchart menolong analis dan programer untuk memecahkan masalah kedalam segmen-segmen yang lebih kecil dan menolong menganalisis alternatif-alternatif lain dalam pengoperasian. Dalam pembuatan game pencarian kata dengan Aksara Toba menggunakan algoritma runut balik (*backtracking)*, maka dilakukan beberapa tahapan seperti yang terlihat pada *Flowchart* Gambar 2.



Gambar 1. Flowchart Game Pencarian Kata dengan Aksara Toba

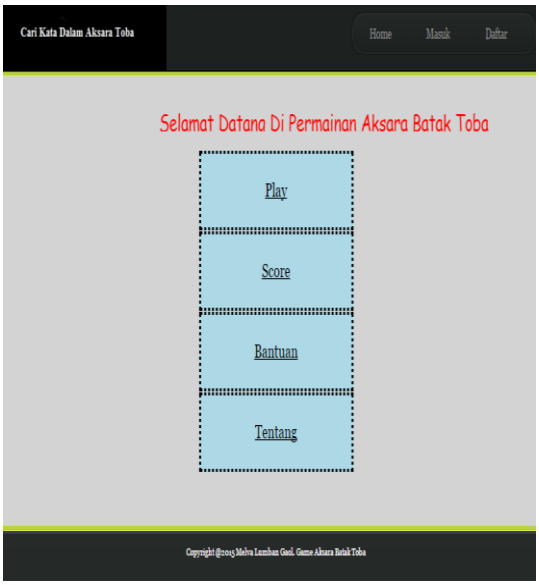
Dari gambar 1 diatas maka dapat dijelaskan bagaimana player atau user akan memasuki dalam permainan tersebut. Jika player belum memiliki akun atau password sendiri, maka player atau registrasi atau daftar diri terlebih dahulu. Jika player sudah memiliki akun atau password, maka player akan langsung login kedalam permainan. Setelah memasuki permainan, maka player memilih salah satu level. Bagi player yang masih pemula lebih baik memilih level mudah. Jika player memilih level mudah , maka player akan diberi 2 soal dan hanya terdiri dari 5 level. Bila player sudah menyelesaikan level mudah, maka skor yang akan ditotalkan dan akan tersimpan didalam menu skor.

IV. IMPLEMENTASI SISTEM

Implementasi antarmuka untuk aplikasi yang dibangun adalah sebagai berikut :

A. Antarmuka Halaman Utama

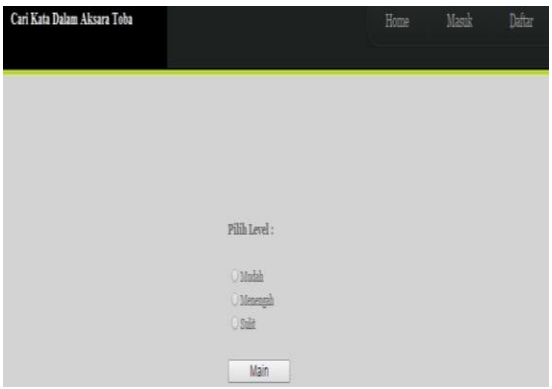
Tampilan halaman utama merupakan tampilan awal dari aplikasi. Pada halaman utama terdapat pengertian tentang permainan yang dibangun. Antarmuka halaman utama dapat dilihat pada gambar 2.



Gambar 2. Antarmuka Halaman Utama

B. Antarmuka Level

Di dalam halaman ini terdapat level yang diberikan kepada pengguna atau pemain. Level terdiri dari mudah, menengah dan sulit. Tampilan antarmuka level dapat dilihat seperti pada gambar 3.



Gambar 3. Antarmuka Level

C. Antarmuka Permainan

Didalam halaman permainan ini terdapat waktu/durasi yang diberikan kepada pengguna untuk berpikir. Selain itu terdapat juga jumlah objek yang di gunakan dan jumlah mill yang telah terbentuk. Objek bewarna merah yaitu pemain/pengguna dan objek berwarna hijau yaitu komputer. Tampilan antarmuka permainan *cari kata* dapat dilihat seperti pada gambar 4.

Gambar 4. Antarmuka Permainan *Cari Kata*

V. KESIMPULAN DAN SARAN

1. Kesimpulan

Dalam permainan, algoritma runut balik sudah bisa memberikan jawaban yang pasti sehingga algoritma runut balik ini bisa diimplementasikan. Selain itu algoritma runut balik juga merupakan algoritma yang sederhana namun cukup mangkus. Hal ini disebabkan karena pada prinsipnya, kita tidak perlu memeriksa semua kemungkinan solusi yang ada. Pencarian hanya mengarah pada solusi yang dipertimbangkan saja dan juga masih bisa dikembangkan lagi dalam permainan ini.

2. Saran

Berdasarkan hasil penelitian, terdapat beberapa saran untuk pengembangan lebih lanjut antara lain:

a. Game ini terdiri dari 3 level dan dapat dimainkan 1 pemain. Namun dapat dikembangkan menjadi lebih dari 3 level dengan menggunakan tema dan juga bisa dilakukan lebih dari 1 pemain.
b. Agar permainan lebih menarik, maka dapat ditambahkan beberapa aturan dan level yang lebih bervariasi.
c. Permainan ini dapat dikembangkan dengan cara kita menambahkan kata yang kita mau ke dalam permainan tersebut.

DAFTAR PUSTAKA

[1] Amin, Imaduddin. 2007. " Penerapan Algoritma Backtracking dalam Permainan Teka-Teki Silang ". Institut Teknologi Bandung.Bandung.

http://informatika.stei.itb.ac.id/~rinaldi.munir/Stmik/20062007/Makalah_2007/MakalahSTMIK2007-098.pdf (19 Maret 2015)

[2] A.S, Rosa. 2011. "Modul Pembelajaran Rekayasa Perangkat Lunak" . Modula. Bandung.

[3] Fahrudin, B. 2016. "Penerapan Algoritma Backtracking Pada Permainan Capsa Banting". STMIK BUDIDARMA. Medan. http://www.stmikbudidarma.ac.id/ejurnal/index.php/jurikom/article/download/168/150

[4] Kozok, Uli . 2009."Warisan LeluhurSastraLama dan Aksara Batak". Kepustakaan Populer Gramedia. Jakarta.

[5] Sitinjak, Suriski. 2012."Pengenalan TulisanTangan Aksara Batak Toba Menggunakan BackPropagation". Universitas Atma Jaya. Yogyakarta.

http://e-journal.uajy.ac.id/389/(23April 2015).

[6] Teneng, Joko Purwadi, dkk. 2010. "Penerapan Algoritma Backtracking pada Permainan Math Maze".KristenDutaWacana.Yohyakarta. http://download.portalgaruda.org/article.php?article=59727&val=4485/(20 April 2015)

[7] Widjaja,VS, Sudirma. 2013. "Implementasi Algoritma Backtracking Dengan Optimasi Menggunakan Teknik Hidden Single Pada Penyelesaian Permainan Sudoku". Universitas Multimedia Nusantara. Tanggerang. http://download.portalgaruda.org/article.php?article=95067&val=57