

PENYUSUNAN PAPAN JAWABAN TEKA-TEKI SILANG MENGUNAKAN BACKTRACKING

Novan Wijaya

Program Studi Manajemen Informatika
AMIK Multi Data Palembang
Email: novan.wijaya@mdp.ac.id

ABSTRAK

Teka-teki silang merupakan permainan klasik sederhana yang sudah umum dimainkan masyarakat awam. Teka-teki silang dapat dimainkan menggunakan media cetak bahkan media online. Untuk membuatnya secara manual tentu diperlukan ketelitian, dan waktu yang lama karena *designer* harus memikirkan tiap huruf yang terkait antara data yang satu dengan data yang lainnya. Kecerdasan Buatan (*Artificial Intelligence*) diperlukan untuk menangani permasalahan ini. Algoritma *backtracking* dipilih untuk memecahkan permasalahan ini karena algoritma ini menghasilkan satu solusi akhir dalam proses pencarian solusi, *designer* hanya perlu mencari *file* data pertanyaan dan jawaban yang terlebih dahulu dimasukkan kedalam file “xls” dan sistem akan meng-*generate* data yang ada sehingga terbentuk papan teka-teki silang. Hasil pengujian menunjukkan bahwa algoritma *backtracking* dapat digunakan untuk membentuk papan teka-teki silang setelah dilakukan pengujian 20 data dengan 19 data berhasil dibentuk dan 1 data tidak berhasil dibentuk.

Kata kunci: teka-teki silang, algoritma *backtracking*, kecerdasan buatan.

ABSTRACT

Crossword puzzle is a classic game that is commonly played every people. Crossword puzzle can be played using print media and even online. To manual it certainly required accuracy, and a long time. Its because the designer have to think about every single letter that relate to one and another. Artificial intelligence is required to solve the problem. Backtracking algorithm selected to find the problems, its because the algorithm produces one final solution in the process of finding solution. Designer only need to find the file perior answer and question included in the “xls” file and the system will generate the data that will be formed crossword puzzle board. The test results show that the backtracking algorithm can be used to form crossword puzzle board after testing 20 data with 19 data has been established and 1 data has not established.

Keywords: crossword puzzle, backtracking algorithm, artificial intelligence.

1. PENDAHULUAN

Permainan teka-teki silang merupakan suatu permainan yang tidak asing lagi bagi masyarakat awam, olah kata untuk melatih daya ingat dan daya pikir seseorang menjadi tujuan dari permainan ini. Beberapa majalah dan media cetak memanfaatkan teka-teki silang sebagai sarana hiburan dengan tujuan agar pembaca tidak bosan karena hanya monoton membaca, tetapi pembaca diajak juga untuk ikut berpikir dan mengingat mengenai bahan bacaan tersebut. Teka-teki silang merupakan suatu *game* yang memungkinkan pemain memasukkan kata yang bersesuaian dengan panjang kotak yang tersedia secara berkesinambungan sampai seluruh kotak terisi penuh (1). Aturan pengisian kata-kata tersebut berhubungan dengan penyamaan jumlah kotak dengan jumlah karakter pada kata dan pengisian kata-kata kedalam kotak pada *crossword puzzle* secara berkesinambungan. Dalam perkembangannya teka-teki silang tidak hanya digunakan dalam mengisi beberapa media cetak atau majalah tetapi sudah mulai merambah di dunia pendidikan, contohnya buku-buku pelajaran siswa yang disusun sedemikian rupa dengan memasukkan teka-teki silang guna melatih wawasan dan minat siswa terhadap bahan pelajaran yang di pelajarnya dan permainan teka-teki silang sebagai media dalam pembelajaran bahasa arab (2). Pembuatan papan teka-teki silang bukanlah hal yang mudah untuk dikerjakan, butuh kesabaran, ketelitian dan waktu yang cukup lama untuk menyusun suatu papan teka-teki silang berdasarkan kata-kata yang terlebih dahulu dipersiapkan. Kesabaran, ketelitian dan waktu yang lama diperlukan karena dalam penyusunannya perlu memperhatikan huruf yang sama berhubungan antara satu kata dengan kata lainnya. Terlebih lagi jika teka-teki silang yang akan dibuat memiliki banyak pertanyaan yang akan disusun.

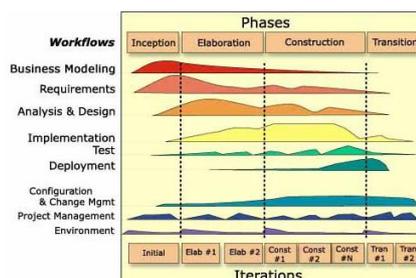
Penerapan AI (*Artificial Intelligence*) telah diupayakan untuk menjadi solusi atas permasalahan yang berkaitan dengan teknik pencarian. AI secara garis besar dapat dikelompokkan menjadi dua yaitu menitikberatkan pada pendekatan “*human-Centered*” dan pendekatan “*Rationality*” (3). Dimana “*human-centered*” harus memiliki keilmuan yang empiris, menyertakan hipotesis dan telah dilakukan percobaan (*Thinking Humanly* dan *Acting Humanly*). Pendekatan “*Rationality*” menyertakan kombinasi antara kemampuan matematika dan rekayasa (*Thinking Rationality* dan *Acting Rationality*). Algoritma *backtracking* dan *brute force* merupakan algoritma pencarian yang umum digunakan dalam masalah pencarian. Setelah melakukan penelitian, dapat disimpulkan bahwa *backtracking* merupakan algoritma yang lebih efisien dalam mencari solusi dibandingkan algoritma *brute force* (4). Algoritma *brute force* tersebut memecahkan masalah dengan sangat sederhana, langsung, dan dengan cara yang jelas. Namun algoritma *brute force* membutuhkan jumlah langkah yang besar dalam penyelesaiannya, sehingga menyebabkan kompleksitas waktu maupun memori yang besar pula. Algoritma *backtracking* merupakan salah satu metode pemecahan masalah yang termasuk dalam strategi yang berbasis pencarian pada ruang status (kutipan). Secara khusus algoritma ini merupakan salah satu pengembangan dari pencarian DFS (*Deep First Search*) yang melakukan pencarian secara mendalam pada suatu permasalahan (5). Penerapan Algoritma *backtracking* dapat dilihat pada permainan *Math Maze* (6). Papan Permainan *Math Maze* dengan menerapkan algoritma *backtracking* dapat menghasilkan 1 solusi untuk setiap *problem* yang dibangkitkan dan *maze* yang dihasilkan dengan algoritma *backtracking* akan menghasilkan *maze* yang tidak memiliki *loop* dan ruang terbuang. Untuk teka-teki silang sudah menggunakan *backtracking* sebagai algoritma untuk mencari kata-kata yang terhubung berdasarkan *template* papan teka-teki silang yang terlebih dahulu dibentuk. Proses pembuatan teka-teki silang berdasarkan *template* yang telah dibuat terlebih dahulu memerlukan tahap yang lebih banyak dan tentunya akan memerlukan waktu yang lebih lama karena hal ini dipengaruhi oleh kepriawaian desainer dalam membuat *template* papan teka-teki silang (7). Berdasarkan latar belakang yang telah dipaparkan maka yang menjadi masalah utama dalam penelitian ini adalah menerapkan *Artificial Intelligence* dalam menyusun suatu papan teka-teki silang secara akurat tanpa menrancang terlebih dahulu papan teka-teki silang dengan mengimplementasikan algoritma *backtracking*.

2. METODOLOGI PENELITIAN

Penyusunan Papan Jawaban Teka-Teki Silang (*Crossword Puzzle*) merupakan suatu game dengan *template* berbentuk segi empat yang terdiri dari kumpulan kotak-kotak berwarna hitam putih serta dilengkapi dua lajur, yaitu mendatar (kumpulan kotak yang membentuk satu baris dan beberapa kolom) dan menurun (kumpulan kotak yang membentuk satu kolom dan beberapa baris). Secara spesifik teka-teki silang merupakan suatu *game* yang memungkinkan pemain memasukkan kata yang bersesuaian dengan panjang kotak yang tersedia secara berkesinambungan sampai seluruh kotak terisi penuh. Aturan pengisian kata-kata tersebut berhubungan dengan penyamaan jumlah kotak dengan jumlah karakter pada kata dan pengisian kata-kata kedalam kotak pada *crossword puzzle* secara berkesinambungan.

Penyusunan papan jawaban teka-teki silang secara otomatis yang ditujukan untuk mempermudah para pembuat teka-teki silang. Pengguna hanya diminta untuk memasukkan deret soal dan pertanyaan teka-teki silang yang akan disusun. Pengguna dapat melihat hasil penyusunan dalam bentuk kotak teka-teki silang yang sudah tersusun beserta pertanyaan yang sudah dikelompokkan menjadi dua bagian yaitu kelompok mendatar dan kelompok menurun.

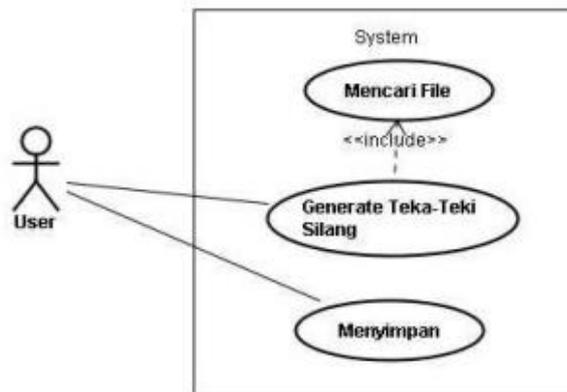
Penyusunan papan jawaban teka-teki silang menggunakan *backtracking* dikembangkan menggunakan model *Rational Unified Process* (RUP). RUP merupakan suatu metode rekayasa perangkat lunak yang dikembangkan dengan mengumpulkan berbagai *best practises* yang terdapat dalam industri pengembangan perangkat lunak (8). RUP menggunakan konsep *object oriented*, dengan aktifitas yang berfokus pada pengembangan model dengan menggunakan *Unified Model Language* (UML) dengan melalui beberapa fase yaitu *inception*, *elaboration*, *construction* dan *transition* (9).



Gambar 1. Arsitektur *Rational Unified Process* (8)

Fase *inception* dilakukan perkiraan terhadap visi, analisis kebutuhan dan ruang lingkup perancangan. Proses bisnis yang diharapkan mampu menggambarkan semua urutan aktifitas yang terjadi pada sistem. Fase ini telah memasuki tahapan dari kebutuhan, baik itu kebutuhan fungsional dan non-fungsional. Fase *elaboration* akan dilakukan desain dan implementasi inti arsitektur secara iteratif, menganalisis berbagai resiko yang akan terjadi pada perancangan yang akan dibangun, mengidentifikasi kebutuhan yang telah ada. Analisis dan desain yang akan dilakukan pada sistem ini dimulai dari menspesifikasikan fitur-fitur perangkat lunak. Fase *construction* mengimplementasikan rancangan perangkat lunak ke dalam bentuk *code* yang tergantung dari hasil tahapan sebelumnya (10) dan fase terakhir yaitu *transition* melakukan *performance testing* terhadap sistem untuk melakukan penyusunan jawaban teka-teki silang.

2.1 Use Case Diagram



Gambar 2. Use Case Diagram

Use case merupakan suatu urutan yang menggambarkan apa-apa saja yang dilakukan aktor terhadap sistem untuk mencapai tujuan tertentu. Gambar 2 menjelaskan user melakukan mencari *file* yang bertujuan untuk mencari data dalam direktori yang berisi data pertanyaan dan jawaban. Setelah itu *user* melakukan *generate* teka-teki silang yang bertujuan untuk membentuk papan teka-teki silang berdasarkan data jawaban yang telah tersedia. Dan *use case* yang terakhir, akan menyimpan *file* hasil *generate* yang sebelumnya telah dilakukan.

2.2 Skenario Use Case

Skenario *use case* merupakan perancangan secara detail yang didapatkan dari *use case diagram* sebelumnya. Pada perancangan skenario *use case* akan dilihat *feed back* yang akan diberikan kembali kepada aktor yang menggunakan sistem berupa tampilan atau informasi. Pada tabel 1, tabel 2, dan tabel 3 merupakan skenario *use case* yang didapatkan dari *use case* sebelumnya. Tiap-tiap tabel menggambarkan aksi yang dilakukan oleh aktor dan reaksi yang ditimbulkan oleh sistem.

2.2.1 Skenario Use Case “Mencari File”

Tabel 1. Skenario use case “mencari file”

Identifikasi	
Nama	Mencari <i>File</i>
Tujuan	Mencari data dan memasukkannya kedalam sistem
Deskripsi	Proses ini dilakukan untuk mencari data dalam direktori yang berisi data pertanyaan dan jawaban
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Data belum dipilih dan belum terisi didalam sistem
Aksi Aktor	Reaksi Sistem
1. Memilih menu ‘ <i>Search File</i> ’	2. Menampilkan lokasi (<i>trace root</i>) data yang akan dipilih
3. Memilih data yang akan ditampilkan pada table	4. Menampilkan data pertanyaan dan jawaban pada tabel
Skenario Alternatif	
Kondisi Akhir	4.1 Menampilkan pesan ekstensi tidak sesuai Sistem menampilkan data dalam tabel pertanyaan dan jawaban

2.2.2 Skenario Use Case “Generate”

Tabel 2. Skenario use case “generate”

Identifikasi	
Nama	<i>Generate</i> Teka-Teki Silang
Tujuan	Mengembangkan papan teka-teki silang
Deskripsi	Proses ini dilakukan untuk membentuk papan teka-teki silang berdasarkan data jawaban yang telah tersedia
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Data sudah dipilih dan ditampilkan dalam tabel
Aksi Aktor	Reaksi Sistem
1. Memilih menu ‘ <i>Generate</i> ’	2. Menampilan papan teka-teki silang yang sudah tersusun dan pertanyaan yang telah terurut mendarat dan menurun
Skenario Alternatif	
Kondisi Akhir	2.1 Menampilkan pesan teka-teki silang tidak dapat tersusun dan petunjuk kepada user Sistem menampilkan hasil <i>generate</i>

2.2.3 Skenario Use Case “Menyimpan”

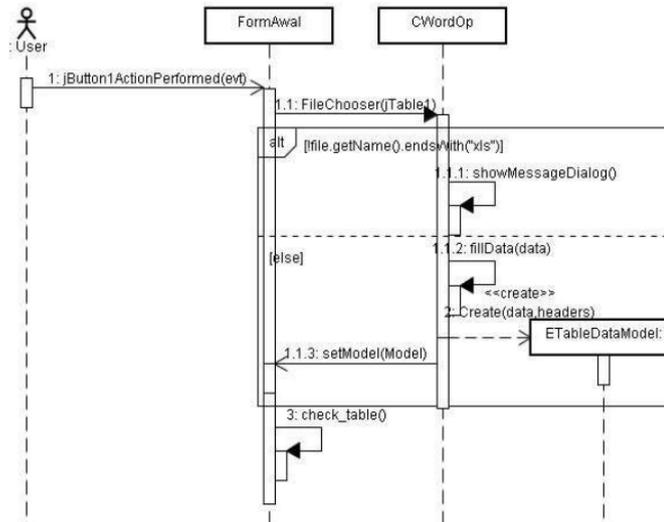
Tabel 3. Skenario Use Case “Menyimpan”

Identifikasi	
Nama	Menyimpan
Tujuan	Menyimpan <i>file</i> hasil <i>generate</i>
Deskripsi	Proses ini dilakukan untuk menyimpan file hasil generate yang sebelumnya telah dilakukan
Aktor	<i>User</i>
Skenario Utama	
Kondisi Awal	Teka-teki silang telah terbentuk dan ditampilkan kepada user
Aksi Aktor	Reaksi Sistem
1. Memilih menu ‘ <i>Save File</i> ’	2. Menampilkan lokasi (<i>trace root</i>) penyimpanan data
3. Memasukkan nama <i>file</i> yang akan disimpan	4. Menampilkan pesan data telah berhasil disimpan
Skenario Alternatif	
Kondisi Akhir	4.1 Menampilkan pesan gagal menyimpan Sistem kembali ke <i>form awal</i>

2.3 Sequence Diagram

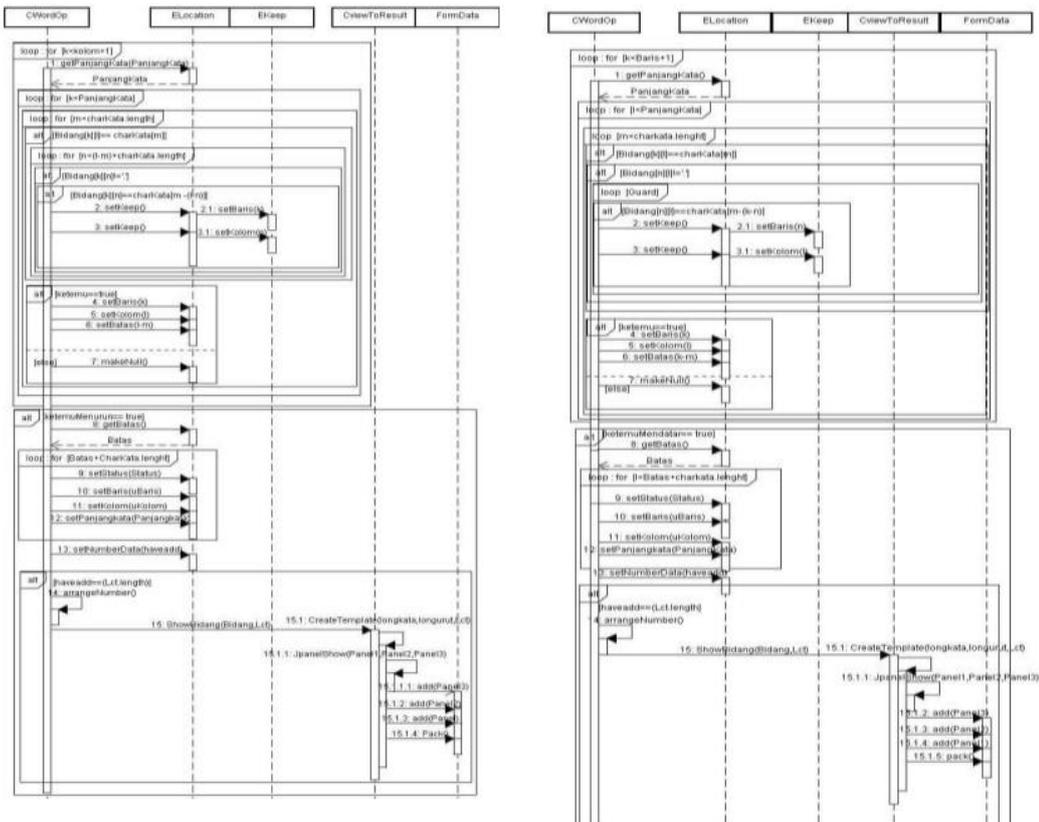
Sequence diagram menggambarkan interaksi antar objek di dalam dan disekitar sistem berupa message yang digambarkan terhadap waktu (9). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah message untuk menghasilkan output tertentu. Sequence diagram terdiri dari dimensi horizontal (objek) dan dimensi vertikal (waktu). Gambar 3, 4, dan 5 menggambarkan interaksi yang terjadi pada sistem secara detail dan bersifat dinamis.

2.3.1 Sequence Diagram “Mencari File”



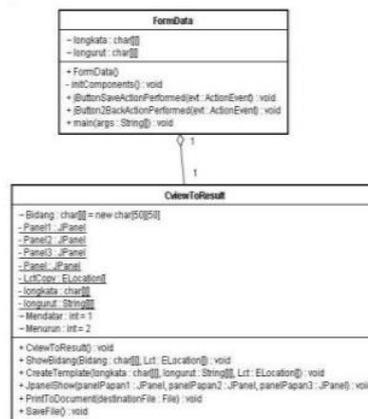
Gambar 3. Sequence Diagram “Mencari File”

2.3.2 Sequence Diagram “Pencarian Mendatar Dan Menurun”



Gambar 4. Sequence Diagram “Pencarian Mendatar dan Menurun”

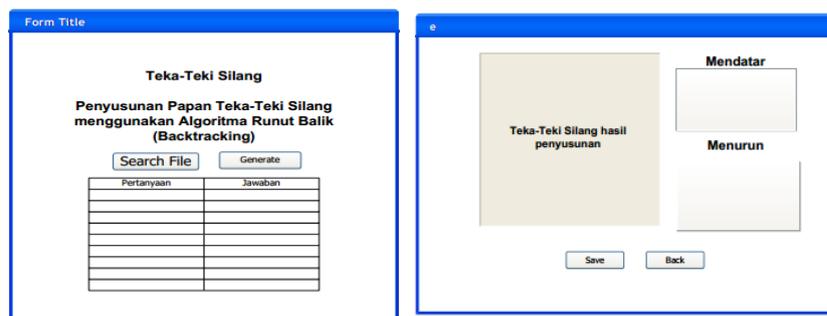
2.4.3 Class Diagram “Menyimpan”



Gambar 8. Class Diagram “Menyimpan”

2.5 Perancangan Antar Muka

Rancangan antar muka dari perangkat lunak seperti yang telah dianalisa, ada dua kelas antarmuka (*interface*) yaitu *FormAwal* dan *FormData*. Antarmuka berfungsi sebagai penghubung antara *user* dengan sistem. Gambar 9 merupakan rancangan antarmuka sistem yang terdiri dari tampilan awal sistem dan tampilan hasil.



Gambar 9. Rancangan Antarmuka Sistem (Tampilan Awal Dan Tampilan Hasil)

3. HASIL PENELITIAN DAN PEMBAHASAN

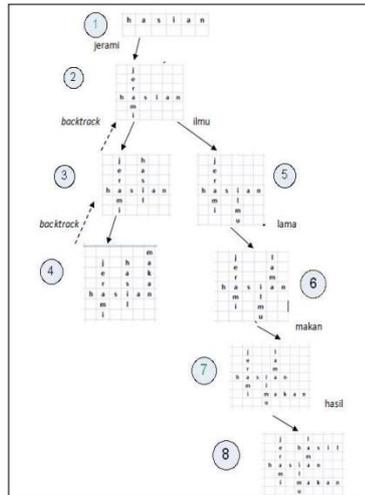
Pengujian perangkat lunak akan dilakukan dengan melakukan pengujian tiap *use case* menggunakan jenis pengujian *blackbox*. Pengujian menggunakan *blackbox* dilakukan untuk melihat kesalahan-kesalahan yang mungkin saja terjadi pada sistem yang telah dibuat, baik itu kesalahan fungsional, kesalahan *interface*, ataupun kesalahan sistem yang terjadi akibat masukan dari *user* yang tidak sesuai dengan permintaan perangkat lunak. Dalam pengujian ini perangkat lunak akan diuji apakah keluaran yang akan dihasilkan sesuai dengan keinginan sehingga didapat kesimpulan apakah pengujian tersebut dapat diterima atau tidak. Sementara untuk pengujian data teka-teki silang dilakukan dengan dua kondisi yaitu pengujian dengan data yang valid dan pengujian dengan data yang tidak valid.

Tabel 4. Contoh data pengujian

Pertanyaan	Jawaban
Panggilan sayang dalam bahasa batak	Hasian
Sesuatu yang didapat dari suatu pekerjaan	Hasil
Tidak sebentar	Lama
Yang dicari dari proses belajar	Ilmu
Batang dan daun padi yang telang mengering	Jerami
Hal yang dilakukan saat lapar	Makan

Pada tabel 4 merupakan contoh data pengujian dengan pertanyaan dan jawaban yang telah ditentukan. Data jawaban akan di urutkan berdasarkan jumlah huruf yang paling banyak, yaitu: “hasian”, “jerami”, “hasil”, “makan”, “lama”, “ilmu”. Sedangkan proses pohon pembentukan dengan algoritma

backtracking dapat dilihat pada gambar 10. Dengan menggunakan data yang valid atau jawaban yang benar, maka data tersebut akan dimasukkan dalam tabel seperti gambar 11. Kemudian akan dibentuk dari susunan teka-teki silang dari data valid tersebut seperti gambar 12.



Gambar 10. Pohon Pembentukan Dengan Algoritma Backtracking

TEKA-TEKI SILANG
 PENYUSUNAN PAPAN TEKA-TEKI SILANG
 MENGGUNAKAN
 ALGORITMA RUNJUT BALIK (BACKTRACKING)

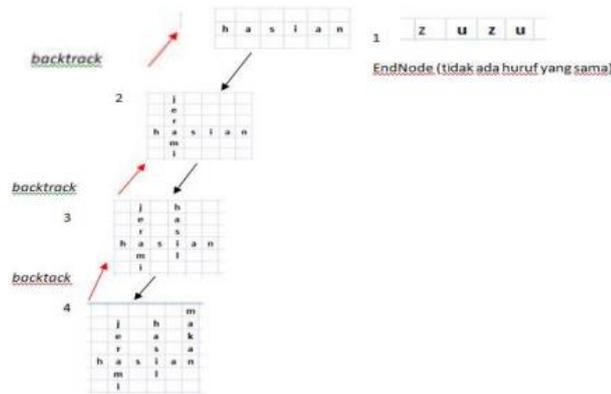
Pertanyaan	Jawaban
Panggilan sayang dalam bahasa batak	hasian
Sesuatu yang didapat dari suatu perke...	hasil
Tidak sebentar	lama
Yang dicari dari proses belajar	ilmu
Batang dan daun padi yang telah men...	jerami
hal yang dilakukan saat lapar	makan

Gambar 11. Data Dimasukkan Dalam Tabel

	Mendatar :
	3. Sesuatu yang didapat dari suatu pekerjaan
	4. Panggilan sayang dalam bahasa batak
	6. hal yang dilakukan saat lapar
	Menurun :
	1. Batang dan daun padi yang telah mengering
	2. Tidak sebentar
	5. Yang dicari dari proses belajar
	<input type="button" value="Save"/>
	<input type="button" value="Back"/>

Gambar 12. Data Berhasil Dibentuk Menjadi Teka-Teki Silang

Percobaan berikutnya dilakukan dengan memasukkan salah satu data yang tidak valid dengan cara mengganti jawaban “ilmu” dengan jawaban “zuzu”. Sehingga didapat susunan pohon seperti gambar 13. Dengan mengganti jawaban yang benar dengan jawaban yang salah seperti gambar 14, akan berakibat tidak terbentuknya pohon pembentukan dan teka-teki silang tidak terbentuk, dikarekan *backtracking* tidak menemukan susunan teka-teki silang tersebut.



Gambar 13. Pohon Pembentukan Dengan Data Tidak Valid



Gambar 14. Teka-Teki Silang Tidak Terbentuk

Untuk melihat hasil banding pembentukan dari aplikasi agar dilakukan pengujian lebih lanjut dengan membandingkan teka-teki silang yang sudah ada dengan hasil pembentukan perangkat lunak dengan pertanyaan dan jawaban yang sama.

4. KESIMPULAN

Percobaan implementasi algoritma *backtracking* dalam otomatisasi pembentukan papan teka-teki silang tanpa terlebih dahulu merancang kota- teka-teki silang. Tetapi papan tidak akan terbentuk jika terdapat data pada kumpulan jawaban yang tidak memiliki kesamaan huruf terhadap data jawaban lainnya. Teka-teki silang secara akurat dapat terbentuk karena pencariannya dilakukan pada setiap huruf pada *list* jawaban, sehingga semua huruf pada data jawaban ditelusuri. Penggunaan algoritma *backtracking* lebih terfokus pada pencarian solusi terhadap kemungkinan solusi yang ada sehingga teka-teki silang yang terbentuk hanya satu dan memiliki jumlah kotak hitam yang lebih dominan jika dibandingkan dengan kotak putih.

DAFTAR PUSTAKA

- [1] Rakhmadhani N, Yamtinah S, Budi U. PENGARUH PENGGUNAAN METODE TEAMS GAMES TOURNAMENTS BERBANTUAN MEDIA TEKA - TEKI SILANG DAN ULAR TANGGA DENGAN MOTIVASI BELAJAR TERHADAP PRESTASI SISWA PADA MATERI KOLOID KELAS XI SMA NEGERI 1 SIMO TAHUN PELAJARAN 2011/2012. J Pendidik Kim. 2013;2(4):190–7.
- [2] Khalilullah M. PERMAINAN TEKA-TEKI SILANG SEBAGAI MEDIA DALAM PEMBELAJARAN BAHASA ARAB (MUFRADAT). J Pemikir Islam. 2012;37(1):15–25.
- [3] Igulu KT, Ziah Z. An Investigation into the Conceptual Controversies between Artificial Intelligence and Computational Intelligence. African J Comput ICT. 2015;8(2):2015.
- [4] Rahayu DS, Suryapratama A, Amongsaufa AZ, Koloay BIK. Evaluasi Algoritma Runtut Balik dan Simulated Annealing pada Permainan Sudoku. JuTISI. 2017;3(1):169–78.

- [5] Khoirussolih M, Wicaksono G, Prayogi M, Nurrohman R. Penyelesaian Masalah 8-Queen Dengan Depth First Search Menggunakan Algoritma Backtracking. *SETRUM*. 2015;4(1):37–9.
- [6] Cui-hua T, Wei-ping X, Yu-ming C. Application of Depth-first Traversal, Randomly Distributed Point Algorithm and Backtracking Method to Maze Game. *J Hebei North Univ (Natural Sci Ed)*. 2013;3(7).
- [7] Syakur MS, Wijanarto. Auto Generated Crossword Puzzle Using Backtracking Algorithm and Brute Force. *J Appl Intell Syst*. 2015;1(1):25–34.
- [8] Larman C. *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development*, Third Edition. United States of America: Addison Wesley Professional; 2004. 736 p.
- [9] Wijaya N, Irsyad H, Taqwiym A. DESIGN VERIFICATION USING PALMPRINT. *TEKNOMATIKA*. 2017;7(2):36–46.
- [10] Pandawana IDGA, Ardiana DPY. Aplikasi Game Cerita Rakyat Bali Sebagai Sarana Pendidikan Karakter Anak Berbasis Mobile. *Lontar Komput*. 2017;8(3):208–18.