

APLIKASI PERMAINAN SUDOKU HURUF HIJAIYAH MENGGUNAKAN ALGORITMA *BACKTRACKING* DAN *MULTIPLICATIVE CRNG* SEBAGAI PEMBANGKIT DAN PENYELESAI PERMAINAN

Misbakhul Mustofin, Hani Nurhayati, MT
Jurusan Teknik Informatika, FSaintek, UIN Maulana Malik Ibrahim Malang
e-mail: mr.tofin@yahoo.com

ABSTRAK

Algoritma Backtracking (Runut-Balik) merupakan algoritma yang sangat baik dalam menyelesaikan permasalahan yang memiliki banyak kemungkinan karena algoritma ini tidak memeriksa semua kemungkinan yang ada. Algoritma ini hanya mempertimbangkan kemungkinan yang mengarah kepada solusi, sehingga proses pencarian menjadi jauh lebih cepat. Algoritma Multiplicative CRNG adalah algoritma pembangkit bilangan acak yang baik karena tidak membangkitkan bilangan yang sama secara berturut-turut. Pembangkit dan penyelesai permainan Sudoku Hijaiyah merupakan permasalahan yang dapat diselesaikan dengan baik menggunakan perpaduan antara algoritma Backtracking dan algoritma Multiplicative CRNG.

Kata kunci: *Backtracking, Multiplicative CRNG, Sudoku*

PENDAHULUAN

Al-Qur'an merupakan kitab suci Agama Islam yang diturunkan menggunakan bahasa Arab. Karena itu dalam mempelajarinya diperlukan pengetahuan mengenai bahasa Arab. Dalam mempelajari bahasa Arab diharuskan mengetahui huruf-huruf bahasa Arab terlebih dahulu. Huruf bahasa Arab biasa dikenal sebagai huruf hijaiyah. Terdapat berbagai macam cara pembelajaran, salah satunya yaitu melalui permainan.

Permainan saat ini sudah mulai beralih ke arah *mobile smartphone*, yang memungkinkan pemain untuk memainkan permainan dimana saja dan kapan saja. Salah satu sistem operasi *mobile smartphone* yang sedang berkembang saat ini adalah sistem operasi Android. Sistem operasi Android bersifat terbuka sehingga memberi kesempatan bagi para pengembang untuk menciptakan aplikasi mereka sendiri, terutama

aplikasi permainan. Aplikasi permainan berkembang pesat di sistem operasi Android. Salah satu permainan yang dapat dikembangkan di sistem operasi Android adalah Sudoku.

Sudoku merupakan salah satu permainan teka-teki angka. Secara umum permainan ini berbentuk **Tabel** berukuran 9x9 yang di dalamnya terdapat sembilan blok/kelompok berukuran 3x3. Tujuan dari permainan ini adalah mengisi setiap kotak (sel **Tabel**) yang kosong dengan angka yang terdiri dari 1-9 sedemikian hingga tidak terdapat pengulangan angka dalam satu blok, satu baris maupun satu kolom.

Dalam menyelesaikan permainan sudoku terdapat beberapa algoritma, salah satunya adalah algoritma *backtracking* (runut-balik). Algoritma *backtracking* berbasis pada *DFS* (*Depth First Search*). Pada algoritma *backtracking*, pencarian solusi lebih difokuskan pada pencarian yang mengarah ke solusi saja sehingga

mempersingkat proses pencarian. Selain algoritma *backtracking* terdapat algoritma *Multiplicative CRNG* yang berfungsi sebagai pembangkit angka secara acak. Algoritma pembangkit angka secara acak ini cukup mudah dipahami dan diaplikasikan karena memiliki batasan yang bisa disesuaikan dengan kebutuhan.

Penelitian tentang *sudoku* menggunakan bahasa Arab telah dilakukan oleh Riyadli Abrar (Abrar,2012). Pada penelitian tersebut, pembangkitan dan penyelesaian permainan menggunakan algoritma *Harmony Search*. Penggunaan *Harmony Search* pada penelitian tersebut dapat menyelesaikan *puzzle* Sudoku dengan benar dalam waktu tercepat antara 1 – 15 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* antara 35 – 40, 2 – 12 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* antara 40 – 45, dan 5 – 35 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* antara 45 – 50. Solusi tidak ditemukan pada parameter tertentu yaitu dengan parameter HMS = 10, HMCR = 0.9, dan PAR = 0.1, 0.5, setelah melakukan 1000 improvisasi dalam waktu 49 – 55 detik.

Pada penelitian lain yang membahas penyelesaian permainan *sudoku* menggunakan algoritma *Backtracking* dan *Elimination* oleh Agustinus Tri Gunawan (Gunawan, 2010) menghasilkan waktu penyelesaian yang lebih cepat. Pada penelitian tersebut dapat menyelesaikan permainan *sudoku* dalam waktu 0.1 detik dengan jumlah kotak kosong yang terdapat dalam *puzzle* berjumlah 10, 29 dan 58.

Fungsi acak yang digunakan pada penelitian Agustinus Tri Gunawan adalah fungsi acak bawaan dari bahasa pemrograman. Penulis mencoba membangkitkan bilangan bulat acak sebanyak 10 kali menggunakan fungsi

acak bawaan dari bahasa pemrograman yang akan digunakan yaitu java dengan *source code* sebagai berikut:

```
int acak = (int)(Math.random()*10);
```

menghasilkan *output* : 9, 9, 8, 1, 5, 0, 0, 5, 0 dan 6. Penulis juga melakukan uji coba pada algoritma pengacakan *Multiplicative CRNG* untuk membangkitkan bilangan bulat acak sebanyak 10 kali dengan parameter a=2, m=11, Z0=7 menghasilkan *output*: 3, 6, 1, 2, 4, 8, 5, 10, 9 dan 7.

Dari percobaan tersebut terlihat bahwa bilangan acak yang dibangkitkan menggunakan fungsi bawaan bahasa pemrograman masih kurang maksimal karena beberapa kali mengulang bilangan yang sama secara berturut-turut. Pada percobaan menggunakan algoritma *Multiplicative CRNG* tidak menghasilkan bilangan yang sama secara berturut-turut. Untuk itu pada penelitian ini tidak menggunakan fungsi bilangan acak bawaan bahasa pemrograman melainkan menggunakan algoritma *Multiplicative CRNG* sebagai pembangkit bilangan acak.

Pada penelitian ini menggunakan perpaduan antara algoritma *Backtracking* dan *Multiplicative CRNG* sebagai *Generator* dan *Solver* dari permainan sudoku yang akan dibangun dengan harapan dapat menyelesaikan permainan sudoku dengan waktu lebih cepat dan 100% solusi ditemukan. *Generator* digunakan untuk membangkitkan soal pada **Tabel** yang sudah disediakan berdasarkan level yang dipilih, sedangkan *Solver* digunakan untuk menyelesaikan permainan jika pemain menyerah dalam menyelesaikan permainan secara manual. Penelitian ini akan diimplementasikan pada Android OS *mobile* dengan harapan dapat diaplikasikan dimana saja dan kapan saja.

1. METODE, ALAT, DAN BAHAN

Metode:

1. *Backtracking*
2. *Multiplicative CRNG*

Alat dan Bahan:

1. Laptop Compaq Presario V3000
2. Smartphone Android Sony Ericsson Experia X8
3. Eclipse IDE
4. Android SDK
5. Android Emulator

3. HASIL DAN DISKUSI

Sudah dijelaskan sebelumnya bahwa pada penelitian yang sudah dilakukan oleh Riyadli Abrar dalam menyelesaikan teka-teki *Sudoku* menggunakan algoritma *Harmony Search* menemukan solusi dalam waktu lebih atau sama dengan 1 detik. Sedangkan pada penelitian Agustinus Tri Gunawan yang menggunakan algoritma *Backtracking* dan *Elimination* dalam menyelesaikan teka-teki *Sudoku* mampu menyelesaikannya dalam waktu kurang dari 1 detik dengan berbagai kondisi. Namun karena menggunakan fungsi random bawaan bahasa pemrograman maka hasil acak masih kurang maksimal.

Pada penelitian ini permasalahan *Sudoku* diselesaikan perpaduan antara algoritma *Backtracking* dan *Multiplicative CRNG* agar dapat membangkitkan permainan *Sudoku* secara unik (tidak ada pengulangan teka-teki secara berturut-turut) dan menyelesaikannya dengan waktu yang cepat.

Algoritma *generator*:

1. Mengisi seluruh kotak *puzzle* menggunakan algoritma *Backtracking* dan disimpan sebagai solusi cadangan.
2. Menghilangkan beberapa kotak *puzzle* sesuai dengan level

permainan secara acak menggunakan algoritma *Multiplicative CRNG*.

3. Menampilkan soal hasil *generate*.

Algoritma *solver*:

1. Mengisi seluruh kotak *puzzle* yang kosong menggunakan algoritma *Backtracking* secara acak menggunakan algoritma *Multiplicative CRNG*.
2. Jika ditemukan solusi dari inputan user, maka menampilkan hasil penyelesaian pada papan permainan.
3. Jika tidak ditemukan solusi dari masukan *user*, maka menampilkan penyelesaian cadangan yang telah disimpan sebelumnya pada saat pembuatan soal.

				2	7	8		5
					4			
5	2	4				9	6	
4	8			1	5	6		
		3	7		2			
1	9		8	4	6			
	3	5						8
2					1		9	6
	6			3		7	5	

Gambar 1. Permainan *Sudoku*.

4.KESIMPULAN

Algoritma *Backtracking* dan *Multiplicative CRNG* merupakan algoritma yang baik untuk digunakan sebagai pembangkit dan penyelesai teka-teki *Sudoku* karena menghasilkan teka-teki yang unik setiap dibangkitkan dan dapat diselesaikan dengan waktu yang cepat.

5.DAFTAR PUSTAKA

- Abrar, Riyadli. 2012 . Aplikasi Permainan Arabic Sudoku Menggunakan Metode Harmony Search sebagai Pembangkit dan Penyelesaian Permainan.Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim. Malang.
- Gunawan, Agustinus Tri. 2010. Penerapan Algoritma Backtracking dan Elimination untuk Membangun Generator dan Solver dalam Menyelesaikan Permainan Sudoku. Sekolah Tinggi Manajemen Informatika dan Komputer AMIKOM. Yogyakarta.
- Izzuddin, Muhammad. 2009. Memperbaiki bacaan Al-Qur'an (Metode Tartil 12 Jam). Assalam Publishing. Solo.
- L, Riani. 2010.Pembangkit Bilangan Acak. Mata Kuliah Pemodelan & Simulasi. Jurusan Teknik Informatika Universitas Komputer Indonesia. Bandung.
- Morenvino, M. Ray, I. dan Anton, S. 2006. Penerapan Algoritma Runut-Balik Untuk Penyelesaian Teka-Teki Sudoku.Laboratorium Ilmu dan Rekayasa Komputasi Departemen Teknik Informatika Institut Teknologi Bandung. Bandung., *in press*
- Safaat H, Nazruddin. 2011. Android, Pemrograman Aplikasi Mobile Smartphone dan Tablet PC Berbasis Android. Informatika. Bandung.
- Sari, Rina Dewi Indah. 2007. Analisis Penyelesaian Puzzle Sudoku dengan Menerapkan Algoritma Backtracking Memanfaatkan Bahasa Pemrograman Visual Basic 6.0 dan Database Microsoft Access 2003. Jurusan Teknik Informatika Sekolah Tinggi Manajemen Informatika dan Komputer (STMIK) "ASIA". Malang.
- Zeeb, Charles N. dan Patrick J. Burns. 1984.Random Number Generator Recommendation. Department of Mechanical Engineering Colorado State University. Colorado.
- Al-Qur'an dan Terjemahannya. Al-Jumanatul 'Ali. Bandung.
- http://en.wikipedia.org/wiki/Sudoku_algorithms, diakses pada tanggal 31 Mei 2012 pukul 14.04