

GENERATOR TEKA TEKI SILANG MENGGUNAKAN ALGORITMA GENETIKA DAN MULTITHREADING UNTUK MENGHITUNG FITNESSNYA

Donny Kurniawan Widodo

Program Studi Teknik Informatika, Unika Soegijapranata Semarang

dny265@gmail.com

Abstract

Crossword is a word game where the player must form a word by answering a clue. There are some difficulties in making crossword, such as forming word that intersects with other words. This project intends to develop a crossword generator using steady-state Genetic Algorithm. The crossword is generated based on a set of words and clues from user input. Multithreading is implemented in the fitness value calculation to speed up the calculation process. The best result can be found in the first generation but can also be found after hundreds of generation because of the randomness in the initial population.

Keywords: *crossword, genetika, multithreading*

Pendahuluan

Terdapat beberapa kesulitan di dalam pembuatan teka teki silang di antaranya adalah membuat setiap kata harus saling berpotongan dan kata yang di pasang tidak boleh ada yang berhimpitan. Untuk mempermudah pembuatan teka teki silang di perlukan program yang dapat secara otomatis menyusun kata kata dalam teka teki silang. Program akan menggunakan algoritma genetika untuk dapat memasang seluruh kata yang telah di input.

Landasan Teori

Teka teki silang adalah sebuah permainan kata. Permainan ini berbentuk sebuah papan berbentuk persegi atau persegi panjang yang berisi kotak hitam dan putih. Kotak hitam putih ini adalah hasil dari susunan kata kata. Kata kata yang di pasang di papan akan di susun mendatar dan menurun hingga saling berpotongan satu sama lain. Setiap huruf dari susunan yang terbentuk akan digambarkan sebagai kotak putih. Cara memainkan permainan teka teki ini adalah dengan menjawab petunjuk dan mengisi kotak putih di papan dengan jawaban dari petunjuk itu. Teka teki dianggap terpecahkan jika seluruh kotak putih telah terisi dengan huruf yang tersusun.

Algoritma genetika adalah algoritma yang mengadaptasi konsep seleksi alam dan evolusi. Konsep ini menjelaskan di dalam sebuah populasi hanya yang

tidak dapat bertahan akan tereliminasi. Sedangkan yang dapat bertahan akan dapat melakukan proses perkawinan hingga dapat menghasilkan individu baru menggantikan yang tidak dapat bertahan.

Proses eliminasi dalam algoritma genetika didasari oleh nilai fitness masing-masing individu. Contoh implementasi algoritma genetika dapat di temukan dalam generator teka teki silang. Kondisi yang ingin dicapai di dalam masalah generator teka teki silang adalah dapat memasang seluruh kata yang di input user. Untuk menangani permasalahan tersebut, individu yang terbentuk akan terdiri oleh susunan kata kata sebagai gen nya. Setiap kata dapat disusun mendatar atau menurun, oleh karena itu setiap gen akan memiliki arah pemasangan kata. Sehingga gen dalam satu individu akan berisi satu kata dan arahnya.

Di permasalahan ini setiap individu tidak boleh memiliki 2 kata yang sama di dalamnya oleh karena itu jumlah gen di individu akan sama dengan jumlah kata yang ingin dipasang. Penilaian setiap individu akan berdasarkan jumlah kata yang dapat dipasang di papan. Pemasangan kata ke papan akan dilakukan secara berurutan. Ketika menemukan kata yang tidak dapat dipasang kata tersebut tidak akan dihitung, nilai fitnessnya adalah jumlah kata yang telah terpasang dan proses perhitungan tidak akan dilanjutkan ke gen selanjutnya.

Setelah menghitung fitness, proses selanjutnya adalah memilih individu-individu untuk dijadikan parent dan disilangkan. Hasil persilangan nantinya akan menggantikan individu yang memiliki nilai fitness lebih rendah. Proses ini akan di ulang hingga ditemukan individu yang balik baik. Individu paling baik yang dimaksud adalah memiliki nilai fitness yang sama dengan jumlah kata yang diinput. Jika hasil terbaik tidak dapat ditemukan program harus memiliki proses terminasi lain. Proses terminasi bisa dilakukan dengan membatasi jumlah generasi atau membatasi dengan waktu berjalannya program.

Di dalam sebuah proses terdapat sekumpulan instruksi. Instruksi-instruksi ini disebut thread. Ketika sebuah program multi-thread di jalankan di prosesor tunggal, thread akan berjalan seolah bersamaan karena pergantian eksekusi thread berjalan sangat cepat. Tetapi ketika dijalankan di multi-prosesor thread akan dijalankan bersamaan karena setiap prosesor akan menjalankan satu thread.

Metodologi Penelitian

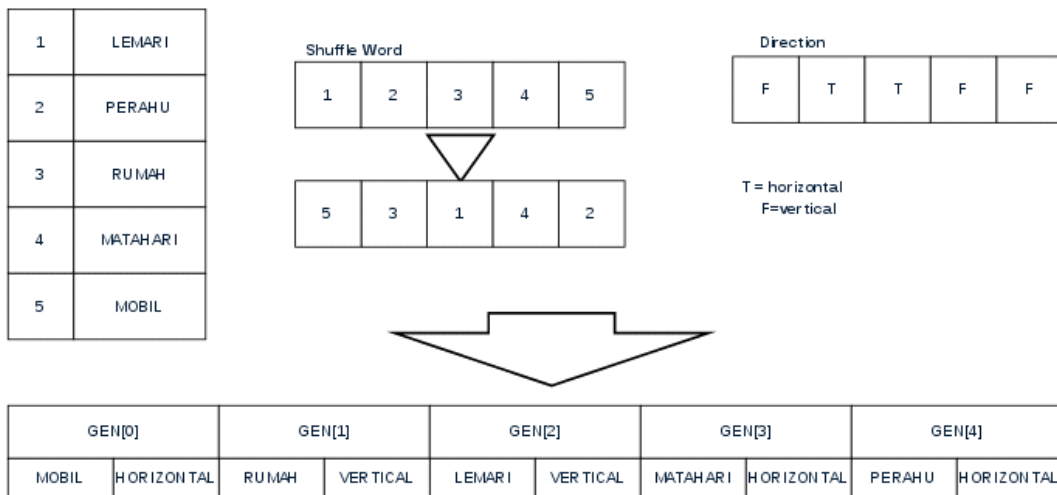
1. Menganalisa masalah dan mencari algoritma yang dapat memecahkan masalah tersebut. Masalah utama dalam program adalah sulitnya menyusun seluruh kata di papan sehingga saling bersilangan.

2. Melakukan tinjauan pustaka seputar permasalahan, algoritma dan data struktur. Algoritma genetika yang di gunakan menggunakan metode steady-state dan menggunakan array of object untuk menyimpan individunya.
3. Mendesain program untuk mengimplementasi algoritma dan data struktur. Desain utama algoritma genetika akan mengikuti metode steady-state.
4. Melakukan testing program. Program akan di testing dengan berbagai input untuk mengecek adanya bug atau error. Selain itu testing dapat menunjukkan hasil-hasil uji coba jalannya algoritma dan program.

Hasil dan Pembahasan

Algoritma genetika yang digunakan akan menggunakan metode steady-state. Di metode ini akan tetap ada proses proses utama di genetika seperti inisialisasi, seleksi, crossover, dan mutasi. Perbedaan steady-state dengan generasional genetika adalah di generasional, individu di dalam populasi akan digantikan seluruhnya dengan hasil persilangan parent-parent yang terpilih sebelumnya tetapi di steady-state akan ada kemungkinan individu populasi sebelumnya dapat bertahan ke generasi setelahnya. Hasil hasil persilangan di steady-state akan menggantikan individu di populasi yang bernilai fitness lebih rendah. Penggunaan metode steady-state ini bertujuan untuk mempertahankan variasi individu didalam populasi.

Di proses inisialisasi akan dilakukan pembentukan individu individu sebagai populasi awal dan penghitungan nilai fitnessnya. Di dalam satu individu akan terdapat kata-kata dan arah masing-masing kata. Kata kata ini akan tersusun secara acak dan tidak boleh terjadi pengulangan kata di dalam satu individu. Setiap kata akan memiliki nomor. Nomor ini akan disimpan di dalam array dan akan dilakukan penacakan. Dan untuk menentukan arah membuat array sebesar jumlah kata yang ada. Dan secara acak memunculkan nilai true dan false. Nilai true berarti kata akan di pasang secara mendatar dan nilai false berarti menurun. Setelah itu kedua array akan digabungkan dan akan menjadi sebuah individu. Lihat gambar 1.



Gambar 1. Pembentukan individu

Setelah pembentukan populasi awal setiap individu akan di hitung nilai fitness nya. Thread akan dibuat setiap menghitung nilai fitness sebuah individu. Thread ini dimaksudkan mempercepat proses penghitungan ketika terdapat jumlah individu yang besar di dalam populasi.

Nilai fitness di dalam individu akan berdasarkan jumlah kata yang terpasang dalam papan sementara. Papan yang dimaksud adalah array 2 dimensi bertipe char sebesar 50x50. Pemasangan kata ke papan akan dilakukan secara berurutan. Sebagai contoh jika di dalam individu gen pertama memiliki kata "MOBIL" dipasang mendatar. Program akan memasang 1 persatu huruf di papan. Huruf pertama di kata pertama akan diletakkan di tengah papan(lihat gambar 2). Setelah memasang kata di papan, kata yang berhasil dipasang akan disimpan dalam array list beserta koordinat setiap huruf nya. Akan terdapat 2 array list yaitu list yang menyimpan kata kata yang disusun mendatar dan yang disusun menurun.

	...	24	25	26	27	28	29	30	31	32	33	34	...	(j)
...														
21														
22														
23														
24														
25			M	O	B	I	L							
26														
27														
28														
29														
...														
(i)														

Gambar 2. memasang kata di papan

	...	24	25	26	27	28	29	30	31	32	33	34	...	(j)
...														
21														
22														
23					R									
24					U									
25					M	O	B	I	L					
26					A									
27					H									
28														
29														
...														
(i)														

Gambar 3. Pemasangan kata ke dua

Untuk pemasangan kata selanjut nya program akan mengecek posisi kata terlebih dahulu. Contoh untuk memasang gen selanjutnya yang memiliki kata "RUMAH" yang dipasang menurun, program akan mengecek list mendatar, jika kata "RUMAH" dipasang mendatar program akan mengecek list menurun sebelumnya. Pengecekan ini bermaksud untuk mencari kata yang dapat berpotongan. Setelah menemukan kata yang dapat di potongkan, program akan mencoba memasang kata itu. Kata yang dipasang harus memotong minimal satu kata yang telah dipasang sebelumnya. Kata yang akan dipasang juga tidak boleh saling berhimpitan dengan kata lain. Nilai fitness sementara untuk contoh di gambar 3 adalah 2 karena kata yang berhasil dipasang adalah 2.

Setelah menemukan seluruh nilai fitness individu di populasi, program akan melakukan seleksi. Seleksi akan dilakukan dengan dua tahap, pertama menentukan individu mana saja yang akan dipersilangkan lalu tahap kedua menyilangkan individu-individu yang telah terpilih sebelumnya. Di tahap pertama menentukan individu mana saja yang akan disilangkan adalah dengan cara generate nilai random antara 0 hingga 1 dan membandingkan nya dengan kemungkinan silang. Kemungkinan silang akan di set sebesar 80% atau berarti 0.8. Jika nilai random lebih kecil atau sama dengan nilai kemungkinan silang maka individu tersebut akan disilangkan nantinya. Contoh di tabel 1, Individu nomor 1 memiliki nilai random 0.339, nilai tersebut lebih kecil dari 0.8 maka individu tersebut akan disimpan dalam populasi sementara. Populasi sementara ini nantinya akan saling disilangkan.

Table 1. seleksi parent pertama

Individu	fitness	Nilai random	Kemungkinan silang	status
1	2	0.339	0.8	disilangkan
2	1	0.652	0.8	disilangkan
3	2	0.834	0.8	diabaikan
4	3	0.522	0.8	disilangkan
5	1	0.690	0.8	disilangkan
6	2	0.234	0.8	disilangkan
7	1	0.903	0.8	diabaikan

Setelah melakukan seleksi tahap pertama, tahap selanjutnya adalah saling menyilangkan individu-individu yang berada di populasi sementara. Setiap individu di populasi sementara ini akan disilangkan. Cara menentukan pasangan setiap individu adalah dengan melakukan seleksi roulette wheel. Proses seleksi ini

adalah dengan memberikan setiap individu range nilai. Range nilai akan berdasarkan nilai fitness masing-masing individu. Semakin tinggi nilai fitness semakin besar range nilai nya. Contoh di tabel 2 telah terdapat 5 individu yang terpilih dari seleksi sebelumnya. Pertama tama program akan menjumlahkan seluruh nilai fitness individu dan didapatkan nilai 9. Di tabel terdapat nilai F_i . F_i adalah nilai jumlah fitness dari individu paling atas hingga individu yang dimaksud. Contoh nilai F_i untuk individu 4 adalah jumlah dari individu 1, 2 dan 4 yaitu $2+1+3 = 6$. setelah memberikan nilai F_i ke masing masing individu, setiap individu akan men-generate nilai random dari 0 hingga nilai total fitness yaitu 9. Nilai random ini akan dibandingkan dengan nilai F_i . Perbandingan dilakukan secara berurutan dari individu pertama. Jika nilai F_i sebuah individu \geq nilai random maka individu itu akan menjadi pasangan pemiliki nilai random. Contoh di tabel 2 individu nomor 2 memiliki nilai random 5. nilai itu dibandingkan pertama kali dengan F_i individu 1. karena $2 \not\geq 5$ maka lanjut ke F_i selanjutnya. Karena $3 \not\geq 5$ lanjut ke individu berikutnya. Karena nilai F_i individu 4 bernilai 6 dan $6 \geq 5$ maka individu 4 akan dijadikan pasangan untuk individu 2. proses ini akan di lanjutkan hingga seluruh individu memiliki pasangan nya masing-masing.

Table 2. Seleksi parent ke dua

individu	fitness	F_i	Random(0,9)	Pasangan
1	2	2	1	1
2	1	3	5	3
4	3	6	8	4
5	1	7	4	4
6	2	9	7	5

Setelah mendapatkan pasangan masing masing program akan langsung menyilangkan kedua individu. Persilangan akan dilakukan dengan metode persilangan satu titik. Persilangan satu titik adalah dengan cara menetapkan sebuah titik di salah satu gen parent dan memotong gen parent di titik itu. Kedua parent yang telah terpotong akan disusun ulang menjadi 2 individu baru. Proses ini akan dapat menyebabkan dalam satu individu dapat memiliki 2 kata yang sama. Maka saat menyusun gen dari titik potong hingga gen terakhir dilakukan pemilihan. Sebagai contoh lihat gambar 4. Titik potong persilangan di contoh adalah 1. Child1 yang dihasilkan dari persilangan dibawah akan memiliki gen 0 dan gen 1 dari parent1. Dan sisa gen yang belum dimiliki oleh child1 akan diambil

dari parent2. Dengan proses ini sebuah individu tidak akan memiliki 2 kata yang sama.

PARENT1

GEN[0]		GEN[1]		GEN[2]		GEN[3]		GEN[4]	
MOBIL	TRUE	RUMAH	FALSE	LEMARI	FALSE	MATAHARI	TRUE	PERAHU	TRUE

X

PARENT2

GEN[0]		GEN[1]		GEN[2]		GEN[3]		GEN[4]	
MATAHARI	FALSE	LEMARI	TRUE	RUMAH	TRUE	MOBIL	TRUE	PERAHU	FALSE

CHILD 1

GEN[0]		GEN[1]		GEN[2]		GEN[3]		GEN[4]	
MOBIL	TRUE	RUMAH	FALSE	MATAHARI	FALSE	LEMARI	TRUE	PERAHU	FALSE

AND

CHILD 2

GEN[0]		GEN[1]		GEN[2]		GEN[3]		GEN[4]	
MATAHARI	FALSE	LEMARI	TRUE	MOBIL	TRUE	RUMAH	FALSE	PERAHU	TRUE

Gambar 4. Persilangan

Setelah melakukan proses persilangan akan dilakukan proses mutasi. Kemungkinan mutasi terjadi dapat bermacam macam, tetapi secara default hanya 3% kemungkinannya. Proses mutasi pertama dengan men-generate nilai secara acak antara 0 hingga 1. jika nilai yang didapat lebih kecil atau sama dengan nilai kemungkinan 0.03(3%) maka individu itu akan di mutasi. Mutasi terhadap individu dilakukan dengan cara mengacak ulang kata kata yang tersusun di individu tersebut. Contoh jika susunan id kata sebelum di mutasi adalah 5,2,3,1,4 ketika terjadi mutasi akan diacak ulang dan bisa menjadi 2,1,4,3,5.

Setelah terbentuknya individu baru dari hasil persilangan. Individu-individu baru akan di hitung fitness nya. Setelah mendapatkan nilai fitness individu tersebut akan di dibandingkan individu-individu di populasi utama. Jika ada individu lama yang memiliki nilai lebih rendah dibandingkan dengan individu baru maka individu lama tersebut akan di gantikan oleh individu baru. Untuk mempertahankan variasi sebelum melakukan penggantian akan di cek apakah sudah ada individu serupa di dalam populasi tersebut. Jika ada individu serupa maka individu baru tersebut tidak akan menggantikan individu lama. Setelah semua individu baru diseleksi maka populasi yang terbentuk akan di cek apakah nilai terbaik telah tercapai oleh salah satu individu. Ketika sudah ada program akan berhenti dan menampilkan hasil nya. Jika belum ditemukan maka program

akan masuk ke generasi selanjutnya dengan mengulang tahap seleksi kembali menggunakan populasi terakhir.

Kesimpulan

Algoritma genetika dapat digunakan untuk generator teka teki silang. Program terkadang dapat menemukan hasil terbaik di generasi pertama tetapi terkadang juga dapat menemukan hasil terbaik setelah ratusan generasi. Hal ini disebabkan karena populasi awal yang acak.

Daftar Pustaka

- [1] Sun Microsystems, Inc., "Multithreaded Programming Guide", [online], (<https://docs.oracle.com/cd/E19120-01/open.solaris/816-5137/book-info/index.html> , diakses tanggal 4 Oktober 2016.)
- [2] Suyanto, "Artificial Intelligence : Searching, Reasoning, Planning dan Learning.", Informatika Bandung, 2014.
- [3] Denise, "American vs British crosswords", [online], (<http://alwayspuzzling.blogspot.co.id/2013/01/american-vs-british-crosswords.html> , diakses tanggal 4 Oktober 2016.)
- [4] Sinaga, Edison., "Skripsi: Implementasi Algoritma Genetika Dalam Penyusunan Teka Teki Silang", STMIK Mikroskil, Medan, 2009.