IMPLEMENTASI DAN ANALISIS ALGORITMA GRAMMATICAL EVOLUTION UNTUK MENYELESAIKAN KASUS SANTA FE TRAIL

Ergandara Purba Setra[1], Agung Toto Wibowo ST., MT.[2], Untary Novia Wisesty ST., MT.[3]

Fakultas Informatika, Universitas Telkom
Jl. Telekomunikasi no 1, Bandung
[1]egansetra@gmail.com

Abstraksi

Pada pengujian ini *Santa fe Trail* akan diselesaikan dengan algoritma *Grammatical Evolution* (GE). *Grammar* yang digunakan pada GE ini adalah *grammar* yang didefinisikan Koza serta *grammar* yang memodifikasi *grammar* Koza. Metode seleksi yang digunakan adalah *roulette wheel* dan *Tournament selection*. Seleksi menggunakan dua metode untuk dibandingkan perfomasinya.

Hasil pengujian didapatkan langkah terbaik pada *grammar1* adalah 462 sedangkan pada *grammar2* langkah terbaiknya adalah 405. Dengan menggunakan peta2, peta3 dan peta4 solusi tidak dapat ditemukan. Metode seleksi dengan menggunakan *roulette wheel* menghasilkan solusi yang lebih banyak dibandingkan dengan *grammar2*.

Kata kunci: Algoritma evolusi, *grammatical evolution,* Santa fe Trail

Abstract

In this experiments santa fe trail will be solved by using Grammatical Evolution (GE). Grammar that will be used with GE is grammar defined by Koza and another grammar that is modification of Koza's Grammar. Roulette Wheel and Tournament selection will be used as parent selection method. Two selection methods is used to see its performance.

The best step for ant to get all of food is 462 with grammar1 and 405 with grammar2. Solution cannot be achieved when layout2, layout3 and layout4 is used. Roulette wheel achieved more solution than tournament selection.

Keyword: ***Evolution algorithm, grammatical evolution, santa fe trail***

1. PENDAHULUAN

Santa fe trail merupakan kasus yang biasa digunakan untuk *benchmarking* algoritma.Tujuan dari kasus ini adalah mencarikan perintah – perintah untuk menggerakan *ant* agar *ant* ini dapat mengambil seluruh *food* yang disebarkan. *Food* yang disebarkan membentuk jalur, namun jalur tersebut tidak sepenuhnya tersambung. Terdapat beberapa rintangan yang harus bisa dilewati oleh *ant.*

Pengujian dengan menggunakan peta umum *santa fe* trail solusi bisa ditemukan. Sebagai contoh, Zusana, Ivan dan Roman dalam tulisannya yang berjudul *Santa fe trail for artificial Ant by means of analytic programming and evolutionary computation* melakukan pengujian dengan algoritma Differential Evolution (DE), Self-Organizing Migrating Algorithm (SOMA), Simulated Annealing (SA) untuk menyelesaikan kasus santa fe trail. Hasil pengujian mereka, didapat langkah minimum 367 untuk DE, 396 ntuk SOMA dan 406 untuk SA.[14]

Namun bagaimana bila peta tersebut dimodifikasi? Akan dicoba beberapa peta yang memiliki penempatan *food* berbeda. Jalur yang dibentuk sama dengan peta umum. Yang membedakannya adalah pada peta2 akan dibuat dengan pola terpotong satu langkah antara *food* satu dengan lainnya, peta3 *food* disebarkan dengan jarak dua langkah antara *food* yang satu dengan lainnya dan pada peta4 polanya mengikuti peta umum namun pada beberapa tempat sebagian *food* akan dihilangkan.

Pengujian akan dilakukan dengan peta umum terlebih dahulu dengan berbagai *grammar*, metode seleksi dan parameter lainnya. Variasi parameter yang

terbaik kemudian akan digunakan untuk menguji kasus dengan beberapa peta lain.

2. GRAMMATICAL EVOLUTION (GE)

Algoritma ini merupakan salah satu dari algoritma evolusi. Untuk mencari solusinya GE ini menggunakan *context free grammar*. *Backus Naus Form* dan *Wijngaarden Form* merupakan contoh *context free grammar*. Berikut komponen – komponen yang terdapat pada algoritma:

2.1. Backus Naus Form

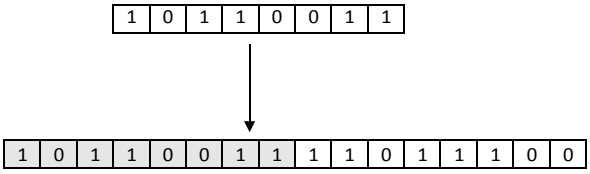
Backus Naus Form (BNF) merupakan notasi untuk mengekspresikan *grammar* suatu bahasa dalam bentuk *production rules*. Komponen – komponen BNF terdiri dari {N, T, P, S}. N merupakan non terminal, N nilai bisa berubah tergantung *production rules* yang didefinisikan. T merupakan terminal, T ini merupakan bentuk akhir dari *grammar* yang tidak dapat berubah lagi nilainya. P merupakan *production rules*, P ini berisikan aturan – aturan yang dapat merubah bentuk dari non terminal. S merupakan definisi awal dari *grammar*.

2.2. Translasi

Translasi merupakan proses perubahan individu menjadi solusi dengan menggunakan *grammar* yang telah didefinisikan. Proses dimulai dengan melakukan modulo gen pertama terhadap jumlah *production rules* yang berlaku. Hasil yang didapat bisa berupa non terminal atau terminal. Proses tersebut diulang terus hingga semua gen digunakan atau tidak ada lagi non terminal yang tersisa.

2.3. Duplikasi

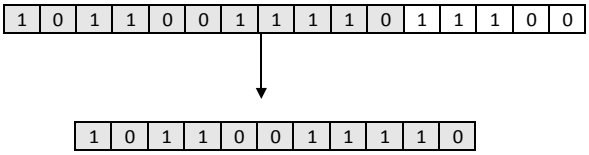
Duplikasi merupakan proses pemanjangan kromosom. Kromosom dengan panjang N akan diperpanjang menjadi 2N. Nilai – nilai pada gen baru merupakan nilai – nilai yang ada pada gen sebelumnya namun urutan gen tersebut diacak.



Gambar2.1 Proses Duplikasi

2.4. Prune

Pada proses ini gen – gen pada kromosom yang tidak terpakai pada proses translasi akan dibuang. Gen – gen tersebut dibuang karena bila tetap disimpan akan mempengaruhi hasil *crossocer* apa bila individu tersebut terpilih menjadi orang tua.[11]



*Gambar **2.2** Proses prune*

2.5. Fitness

Nilai *fitness* menunjukan seberapa baik solusi yang diberikan oleh individu[11][12]. Nilai *fitness* suatu individu mempengaruhi kemungkinan individu tersebut terpilih menjadi orang tua untuk menghasilkan individu baru yang lebih baik. Semakin besar nilai *fitness* semakin baik kualitas dari individu tersebut.

Misalkan nilai *fitness* dirumuskan sebagai = $\frac{a}{b+1}$. Semakin besar nilai a dan semakin kecil nilai b maka suatu individu kualitasnya semakin baik. Bila terdapat individu setelah diproses didapatkan nilai a = 100 dan b = 0. Maka nilai *fitness* individu tersebut adalah$f = \frac{100}{0+1} = 100$.

2.6. Metode Seleksi

Seteleh semua individu dalam popluasi selesai diproses selanjutnya akan dipilih beberapa individu yang akan menjadi orang tua. Para orang tua digunakan untuk menghasilkan individu baru yang akan digunakan di generasi selanjutnya. Pada GE ini metode yang digunakan adalah *Roulette Wheel* dan *Tournament Selection.*

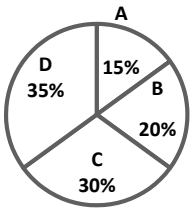
2.6.1. Roulette Wheel

Metode ini memberikan probabilitas yang lebih tinggi terhadap individu yang berkualitas. Semakin tinggi *fitness* (ukuran kualitas) individu maka probabilitasnya semakin tinggi. Misalkan terdapat individu berikut:

Tabel 21 Contoh Individu

Individu	Fitness
Individu A	0.15
Individu B	0.2
Individu C	0.3
Individu D	0.35

Terdapat empat individu dengan *fitness* yang berbeda – beda. Maka setiap individu tersebut memiliki probabilitas terpilih yang berbeda pula.



Gambar2.1 Probabilitas terpilih

2.6.2. Tournament Selection

Dengan menggunakan metode ini semua individu memiliki kesempatan yang sama untuk menjadi calon orang tua. Mula – mula akan dipilih beberapa individu sejumlah K secara acak. Lalu tiap – tiap individu yang terpilih ditandingkan. Individu terbaik akan terpilih menjadi orang tua.[8]

Gambar 2.2 Proses Tournament

2.7. Crossover

Setelah seleksi selesai dilakukan maka individu – individu yang terpilih sebagai orang tua akan digabungkan sehingga tercipta individu baru. Bit pada orang tua akan dipotong pada titik tertentu, kemudian potongan – potongannya digabungkan dengan orang tua lainnya. Terdapat dua metode *crossover*, yaitu *crossover* satu titik dan *crossover* banyak titik.[11][12]

2.8. Elitisme

Proses ini menyimpan satu individu terbaik dalam satu generasi. Proses ini dilakukan agar individu yang baik tidak hilang ketiga populasi baru dibentuk.

2.9. Mutasi

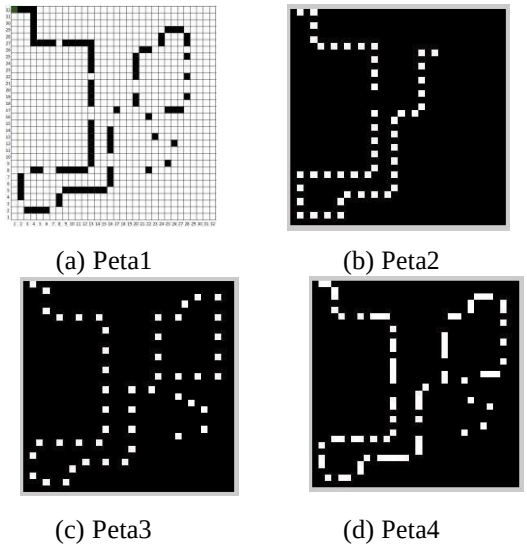
Mutasi mengubah nilai – nilai pada individu. Bit yang bernilai 0 akan berubah menjadi 1 sedangkan bit bernilai 1 berubah menjadi 0. Mutasi ada tiga macam. Mutasi tingkat bit, mutasi tingkat gen dam mutasi tingkat kromosom.[11][12]

3. DESAIN PERMASALAHAN

3.1. Santa fe trail

Permasalahan dari kasus ini adalah mencari perintah – perintah untuk menggerakan *ant*. *Ant* disini hanya dapat melakukan beberapa perintah, seperti: Bergerak maju ke depan, berbalik arah ke samping kanan, berbalik arah ke samping kiri dan mengecek keberadaan makanan di depannya. Umumnya terdapat 89 *food* yang disebarkan. Aturan – aturan yang akan digunakan adalah sebagai berikut:
Titik awal *ant* berada di pojok kiri atas.

- ketika bergerak *ant* tidak bisa menembus batasan, maksudnya bila didepannya merupakan batas peta. *Ant* tidak bisa maju dan muncul dari sisi sebaliknya. Melainkan *ant* tidak bisa menjalankan perintah.
- Peta yang akan digunakan dalam pengujian jumlahnya empat. Peta1 merupakan peta *santa fe trail* umum. peta2 mengikuti jalur yang sama dengan peta1 namun pola penyebarannya berbeda, yaitu terpotong sebanyak satu langkah antara satu *food* dengan *food* lainnya. Peta3 mirip dengan peta2, perbedaannya peta3 ini langkah yang terpotong sebanyak dua langkah. Peta4 mirip dengan peta1, hanya saja pada beberapa titik ada *food* yang dihilangkan dari tempat semula. Jumlah *food* yang disebarkan dari masing – masing peta adalah peta1 sebanyak 89, peta2 sebanyak 40, peta3 sebanyak 40 dan peta4 sebanyak 75.



(a) Peta1 (b) Peta2

(c) Peta3 (d) Peta4

Gambar 3.1 Peta Pengujian

- Jumlah langkah maksimum adalah 650 langkah.
- Kondisi berhenti pencarian adalah apabila *ant* berhasil mendapatkan seluruh *food* atau jumlah langkah telah mencapai batas maksimum.

3.2. Grammar BNF

Grammar yang akan digunakan pada pengujian terdapat dua *grammar*. *Grammar1* merupakan *grammar* yang didefinisikan Koza, sedanglan *grammar2* merupakan modifikasi dari *grammar1*. *Grammar1* setelah mengecek keberadaan *food*, walaupun didapat nilai *true* belum tentu akan

memberikan perintah maju karenanya pada *grammar*2 hal tersebut diubah sehingga dipastikan *ant* bergerak maju bila kondisi pengecekan *food* bernilai *true*. Pada *grammar2* juga ditambahkan pengecekan arah samping kiri dan kanan *ant.*

Tabel 3.1 Grammar1

T = {move, turn_left, turn_right, food_ahead}	
S = <expr>	
<expr> :=	<line> | <expr> <line>
<line> :=	IFELSE food_ahead <expr> <expr> | <op>
<op> :=	move | turn_left | turn_right

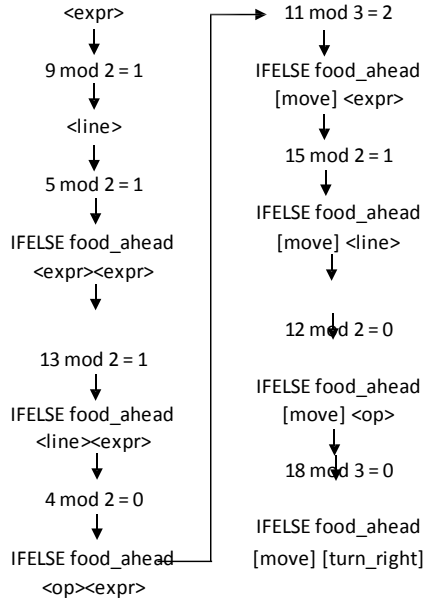
Tabel 3.2 Grammar2

T = { move_forward, turn_left, turn_right, food_ahead, food_left, food_right }	
S = { <expr> }	
<expr> :=	IF food_ahead move_forward IFELSE food_right turn_right IFELSE food_left turn_left ELSE <op> <expr>
	IF food_ahead move_forward IFELSE food_right turn_right IFELSE food_left turn_left ELSE <expr>
	IF food_ahead move_forward IFELSE food_right turn_right IFELSE food_left turn_left ELSE <op>
	<op>
<op> :=	move forward | turn_left | turn_right

3.3. Alur Proses

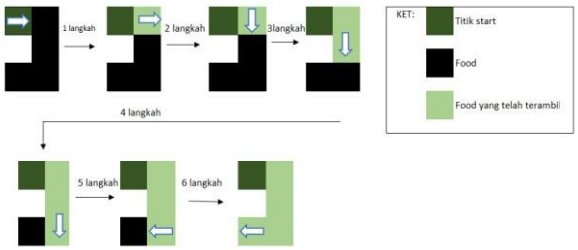
Algoritma akan diimplementasikan dengan alur proses sebagai berikut:

1) Populasi dibangkitkan dengan masing – masing individu diresentasikan dengan bilangan biner.
2) Ambil satu individu dari populasi, lalu konversi kromosomnya ke representasi integer.
3) Translasikan individu dengan *grammar.* Translasi dilakukan seperti contoh berikut:



Gambar 3.2 Proses Translasi

4) Lakukan proses Duplikasi atau *prune* bila diperlukan.
5) Eksekusi perintah yang dihasilkan terhadap peta. *Ant* akan berjalan terus menerus hingga kondisi berhenti tercapai.



Gambar 3.2 Proses Eksekusi

6) Hitung nilai *fitnessnya*. Nilai *fitness* dihitung dengan rumus $f = \frac{M\text{[illegible]}}{\text{[illegible]}+1} * 1000$. Rumus tersebut digunakan karena semakin banyak *food*

yang terambil dan semakin dikit langkah yang diperlukan maka semakin baik *fitness* individu.
7) Ulangi langkah (2) hingga seluruh individu diproses.
8) Simpan Individu dengan *fitness* terbaik. Individu ini akan digunakan lagi di generasi selanjutnya.
9) Pilih beberapa individu untuk dijadikan orang tua.
10) Lakukan *crossover* terhadap individu yang terpilih menjadi orang tua.
11) Lakukan Mutasi. (Dengan probabilitas tertentu).
12) *Update* populasi.
13) Ulangi hingga maksimum genereasi tercapai.

4. Pengujian

4.1. Skenario Pengujian

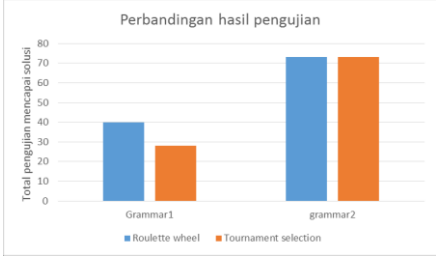
Setiap pengujian akan dilakukan dengan menggunakan dua populasi. Setiap populasi yang dibangkitkan akan dilahirkan individu – individu yang sama. Yang menyebabkan beragamnya individu tersebut adalah apabila *crossover* dilakukan karena dua populasi ini menggunakan metode seleksi yang berbeda untuk menyeleksi orang tuanya. Pengujian akan dilakukan dengan dua skenario.

Skenario1, Skenario ini menggunakan peta1; *grammar1 & grammar2*; metode seleksi *roulette wheel & tournament selection.* Hasil yang dilihat dari pengujian ini adalah performasi dari masing – masing *grammar* beserta pengaruh metode seleksi yang digunakan.

Skenario2, Skenario ini menggunakan peta2, peta3 dan peta4; *grammar1 & grammar2;* metode seleksi *roulette wheel & tournament selection.* Hal yang dilihat dari pegujian ini adalah apakah *ant* dapat mengambil seluruh *food* yang disebarkan pada peta – peta tersebut.

4.2. Hasil Pengujian 4.3. Skenario1

Hasil pengujian dengan menggunakan *grammar1* terdapat 68 dari 240 pengujian yang mencapai solusi. Sedangkan dengan menggunakan *grammar2* terdapat 146 dari 240 pengujian yang mencapai solusi. Hasil pengujian dengan *grammar2* lebih konsisten dalam mencapai solusi. *Grammar2* menambahkan pengecekan keaarah depan dan samping *ant*, hal ini menyebabkan performasi dengan *grammar2* meningkat.



Gambar4.1 Hasil Pengujian skenario1

Dari pengujian dengan *grammar1* dapat dilihat bahwa seleksi dengan *roulette wheel* memiliki performasi yang lebih baik dibandingkan dengan *tournament selection.* Namun pada pengujian dengan *grammar2* kedua metode memiliki performasi yang sama. Hal ini bisa terjadi karena *grammar2* memiliki performasi yang lebih baik dibanding *grammar1* sehingga jumlah individu yang bisa menjadi solusi lebih banyak. Pada *grammar1* kemungkinan individu yang merupakan solusi tidak sebanyak *grammar2* sehingga metode seleksi lebih terlihat perbedaan performasinya. *Roulette Wheel* lebih cenderung individu yang memiliki *fitness* tinggi dalam seleksinya, dengan menggunakan individu yang berkualitas dalam *crossover* individu baru yang dilahirkan akan menjadi lebih baik.

4.4. Skenario2

Tabel 4.2 Hasil pengujian skenario2

Peta	Food terambil	Total Food
Peta2	43	48
Peta3	28	48
peta4	50	75

Pada skenario ini digunakan peta dengan berbagai macam pola penyebaran *food*. Semua pengujian dengan peta – peta tersebut tidak mecapai solusi. Dari hasil pengujian dapat dilihat bahwa *peta2* yang penyebarannya *foodnya* lebih rapat memiliki performasi yang lebih baik, 43 dari 48 *food* terambil, sedangkan pada peta3 hanya 28 dari 48 *food* yang terambil.

Pada peta4 *food* yang terambil sebanyak 50 dari 75. Walaupun peta ini memiliki pola yang sama dengan peta1 namun dengan menghilangkan beberapa *food* dibeberapa titik menyebabkan solusi tidak tercapai. Hal ini bisa terjadi karena pada peta1 penempatan *foodnya* sangat strategis sehingga banyak individu yang mungkin bisa menjadi solusi, namun bila pada beberapa titik ada *food* yang hilang pola perintah yang bisa menjadi solusi pun bisa menghilang.

5. PENUTUP

5.1. Kesimpulan

Kesimpulan yang dapat diambil dari hasil pengujian & analisis adalah sebagai berikut:

1) Pada skenario1 solusi dapat ditemukan. Jumlah langkah terbaik dengan *grammar1* adalah 461 sedangkan pada *grammar2* 405. Walaupun solusi

dapat tercapai namun jumlah langkah yang dicapai tidak sesuai harapan.

2) Metode seleksi memberi pengaruh terhadap solusi yang dihasilkan. Namun kinerja penyeleksian juga dipengaruhi oleh parameter lain. Seperti pada pengujian dengan *grammar1* perbedaannya cukup mencolok dimana *roulette wheel* menghasilkan solusi yang lebih banyak. Namun pada pengujian lainnya kinerja kedua metode sama baiknya.
3) Hasil pengujian pada skenario2 tidak ada yang mencapai solusi. Hal ini bisa terjadi karena dengan peta – peta tersebut kemungkinan individu yang merupakan solusi jumlahnya sedikit sehingga dengan maksimum generasi tersebut sulit untuk ditemukan atau bisa juga tidak ada individu (dengan paramter yang diujikan) yang merupakan solusi.
4) Solusi yang didapatkan berupa perintah untuk *ant* yang dilakukan berulang – ulang. Perintah tersebut memiliki batas dengan panjang tertentu.
Oleh karena itu pola penempatan *food* mempengaruhi apakah solusi (seluruh *food* terambil) dapat dicapai. Sebab tidak semua pola dapat terselesaikan dengan pengulangan perintah yang sama dengan panjang yang terbatas.

4.2. Saran

Hal – hal yang dapat dikembangkan untuk pembahasan berikutnya:

1) Menggunakan algoritma lainnya untuk perbandingan hasil.
2) Menggunakan representasi *grammar* lainnya selain BNF, seperti: Extended-Backus Naus Form (EBNF).

DAFTAR PUSTAKA

[1] C Ferreira.2001. Gene Expression Programming: A New Adaptive Algorithm for Solving Problems. Complex Systems, Vol. 13, issue 2: 87129.

[2] Eiben Agoston E., Smith J E.2007. Introduction to evolutionary Computing (Natural Computing Series). Springer.

[3] Futuyma, Douglas J. (2005). Evolution. Sunderland, Massachusetts: Sinauer Associates, Inc. ISBN 0-87893-187-2.

[4] J.H. Holland.1998. Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to biology, control and artificial intelligence. MIT Press, ISBN 0-26258111- 6.

[5] Koza, J.1992. Genetic Programming. MITPress.

[6] Lande R, Arnold SJ (1983). "The measurement of selection on correlated characters". Evolution 37: 1210–26}.

[7] M. Mitchell.1996.An Introduction to Genetic Algorithms.MITPress.

[8] Noraini Mohd Razali, John Geraghty.2011.

Genetic Algorithm Performance with Different Selection Strategies in Solving TSP. WCE 2011, London UK. [9] Ryan C., Collins J.J. and O'Neill M. Grammatical Evolution: Evolutionary Automatic programming for an arbitary Language.

[10] Ryan C. and O'Neil M.2003. Grammatical

Evolution: A steady state approach.

[11] Suyanto.2008.Evolutionary Computation: Komputasi berbasis evolusi dan genetika. Bandung: Informatika.

[12] Suyanto.2011.Artificial Intelligence: Searching, Reasoning, Planning dan Learning (Cetakan kedua), Bandung: Informatika.

[13] Urbano Paulo, Georgiou Loukas. 2013. Improving Santa fe Trail using novelty search. http://dx.doi.org/10.7551/978-0-262-31709-

2ch137 [Online]

[14] Zusana Oplatkova, Ivan Zelinka, Roman

Senkerik.2008. Santa fe trail for artificial ant by means of analytic programming and evolutionary computation. International Journal of simulation System, Science and Technology. Volume 9, Number 3, September 2008, page 20.

[15] Grune, Dick (1999). Parsing Techniques: A Practical Guide. US: Springer.