

CLONING-BASED ALGORITHM DAN APLIKASINYA DALAM TRAVELLING SALESPERSON PROBLEM

Pranoto Budi Sasongko – NIM: 13506098

Jurusan Teknik Informatika Institut Teknologi Bandung

Jl. Ganesha 10, Bandung

pranoto_budi_s@students.itb.ac.id

Abstrak -- *Clonal Selection* adalah mekanisme yang digunakan oleh sistem kekebalan tubuh (*immune system*) untuk menyeleksi sel yang akan diperbanyak (*di-cloning*) berdasarkan kemampuan untuk mengenali antigen pada patogen (benda asing dari luar tubuh). Pada *clonal selection*, terjadi proses seleksi positif dan seleksi negatif, yaitu upaya memilih sel mana yang akan *di-cloning* atau diperbanyak dan sel mana yang akan dibiarkan mati dalam upaya untuk meningkatkan kemampuan untuk mengenali dan mengikat antigen. Makalah ini akan menyajikan algoritma yang terinspirasi oleh *clonal selection* yang terjadi pada sistem kekebalan tubuh manusia (*immune system*) untuk menemukan solusi optimum dalam persoalan optimasi (*optimization problem*). Makalah ini juga akan menunjukkan bahwa algoritma ini memberikan hasil yang lebih baik jika dibandingkan dengan pendekatan lain yang dilakukan oleh Algoritma Semut (*Ant-based Algorithm*).

Kata Kunci: *clonal selection, Sistem Kekebalan Tubuh, Traveling Salesman Problem, TSP, Positive/negative selection, Optimasi.*

1. PENDAHULUAN

Saat sekarang ini sudah mulai ditemukan algoritma-algoritma baru yang disebut sebagai *bio-inspired algorithm* yaitu algoritma komputasi yang diinspirasi oleh mekanisme alamiah makhluk hidup. Sebagai contoh adalah *neural network* yang didasar oleh mekanisme jaringan syaraf, algoritma genetika yang didasari system evolusi, algoritma semut yang didasari sistem koloni semut, dan komunikasi antar protokol yang didasari oleh proses penyebaran

penyakit dan masih banyak lagi. Salah satu metode dalam *bio-inspired algorithm* yang baru berkembang adalah *immune system-based algorithm* yaitu algoritma yang diinspirasi dari cara kerja sistem kekebalan tubuh pada manusia.

Pada sistem kekebalan tubuh manusia terjadi proses yang dinamakan *clonal selection*, algoritma optimasi yang akan dibahas dalam makalah ini terinspirasi oleh proses *clonal selection* tersebut. Algoritma ini diajukan oleh M. Bakhouya and J. Gaber dari *Université de Technologie de Belfort-Montbéliard* pada tahun 2006.

Mekanisme *Clonal selection* digunakan oleh sistem kekebalan tubuh manusia untuk menentukan respons terhadap stimulus dari antigen. Untuk mengilustrasikan algoritma ini, digunakan permasalahan *Traveling Salesman Problem (TSP)* yang merupakan contoh masalah dalam optimasi kombinatorial. Persoalan TSP secara mudah dapat dinyatakan sebagai berikut: “jika seseorang harus mengunjungi banyak kota, dan harus mengunjungi setiap kota tepat satu kali dan kembali ke kota asalnya. Jika diberikan jarak antar kota yang satu dengan kota yang lain, tentukan jarak minimum yang diperlukan untuk melalui semua kota tadi.”

Algoritma yang disajikan dalam makalah ini yang didasari oleh *clonal selection* dengan mekanisme seleksi positif dan seleksi negatif, terbukti berhasil menggantikan algoritma pendekatan optimasi lain yang didasarkan pada sifat koloni semut, yaitu algoritma semut. Secara umum, dengan aksi kloning, jumlah agen pencari ini tidak konstan namun berubah-ubah selama operasi pencarian.

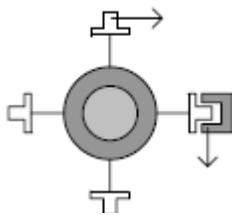
Dalam proses pencariannya, ketika menemukan 2 jalur yang berbeda, sebuah agen akan mengkloning dirinya sendiri dan hasil kloningnya ini bergerak ke node tetangganya secara acak.

2. SISTEM KEKEBALAN TUBUH MANUSIA

Tubuh manusia sistem kekebalan berfungsi untuk mendeteksi masuknya benda asing atau *patogen* ke dalam tubuh seperti virus, bakteri yang dapat mengganggu kestabilan tubuh. Sistem kekebalan tubuh tidak dapat mendeteksi secara langsung adanya *patogen* yang masuk ke dalam tubuh, tetapi dapat dideteksi melalui bagian dari *patogen* yang disebut dengan *antigen*.

Jika terdeteksi adanya *patogen*, sistem kekebalan tubuh bertugas untuk mengeleminasinya dari tubuh. Agar proses eliminasi *patogen* berlangsung dengan baik dan benar, sistem kekebalan tubuh harus mampu untuk membedakan antara *antigen* pada *patogen* yang selanjutnya disebut dengan *nonself-antigen* dengan *antigen* yang dimiliki sel-sel tubuh yang disebut dengan *self-antigen*.

Sel-sel kekebalan tubuh yang fungsinya khusus mendeteksi adanya patogen adalah *lymphocytes*. Ada dua tipe *lymphocytes* yaitu *B-cells* dan *T-cells*. Kedua tipe *lymphocytes* mempunyai molekul pada permukaan sel-nya, yang disebut sebagai *receptor*, yang berfungsi untuk mengikat molekul *antigen* dari *patogen*. *Receptor* dari *T-cells* disebut dengan TCR sedangkan *receptor B-cell* disebut dengan BCR atau biasa dikenal dengan antibodi. Proses deteksi *patogen* oleh *B-cells* dan *T-cells* menggunakan prinsip komplemen struktur bentuk (lihat Gambar. 1).



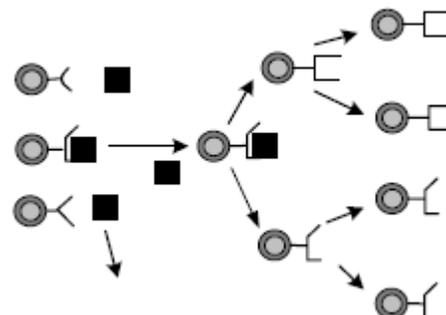
Gambar 1. Proses deteksi antigen oleh receptor

Agar *receptor* mampu mengikat antigen maka *receptor* harus mempunyai struktur bentuk yang berkomplemen terhadap struktur *antigen*. Untuk menyatakan kekuatan ikatan antara *antigen* dengan *receptor* digunakan istilah *affinity* (afinitas). Semakin besar nilai *affinity* menunjukkan semakin

kuatnya ikatan antara *antigen* dan *receptor*, demikian pula sebaliknya jika nilai *affinity* kecil menunjukkan ikatan *receptor* dengan *antigen* yang lemah. Melalui proses yang dinamakan *affinity maturation*, *receptor* melakukan adaptasi terhadap bentuk struktur antigen dengan cara mengikat struktur *antigen-antigen* yang telah dikenali tubuh.

Dari banyak sel-sel *lymphocytes* hanya *lymphocytes* yang *receptor*-nya mampu mengikat *antigen* saja yang dipertahankan tetap hidup sedangkan yang tidak mampu mengikat antigen akan mati. Jika ada sel *lymphocytes* yang dapat mengikat *antigen* maka sel ini akan diperbanyak dengan cara *mitosis*. Karena adanya mutasi sel-sel hasil perbanyakan tidak semuanya sama persis dengan sel induk tetapi akan ada sedikit perbedaan dalam struktur bentuk *receptor*.

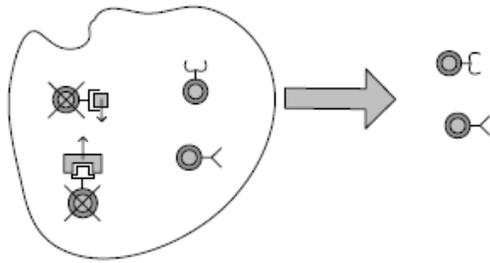
Keseluruhan proses *affinity maturation* dan perbanyakan sel ini disebut dengan *clonal selection* (lihat Gambar 2).



Gambar 2. Proses Clonal Selection

Selain bertugas untuk mendeteksi patogen, sel *lymphocytes* juga harus bisa membedakan *self-antigen* (antigen yang berasal dari sel-sel tubuh) dengan *nonself-antigen* (antigen yang berasal dari *patogen*). Metode yang digunakan berlawanan dengan *clonal selection* yaitu dengan membuat *lymphocytes* beradaptasi untuk mengikat *self-antigen*. Jika struktur *lymphocytes* dapat mengikat *self-antigen* maka *lymphocytes* tersebut akan dimatikan sedangkan *lymphocytes* yang tidak dapat mengikat *self-antigen* akan tetap dipertahankan hidup.

Dengan kata lain jika *affinity* dibawah suatu nilai ambang ikatan maka *lymphocytes* tetap dipertahankan hidup sedangkan jika nilai *affinity* diatas batas ikatan maka *lymphocytes* tersebut akan dimatikan. Proses ini disebut dengan seleksi negatif dan nama organ tempat terjadinya proses seleksi negatif adalah di dalam *thymus* (lihat Gambar 3).



Gambar 3. Proses *Negative Selection* dalam Thymus

Lymphocytes yang tetap hidup selanjutnya disebut sebagai *detector* akan masuk dalam aliran darah dan bersirkulasi ke seluruh tubuh untuk mendeteksi adanya nonself-antigen yang masuk tubuh.

Gambar 3. Proses *negative selection* dalam *thymus*

3. ANALISIS *CLONING-BASED ALGORITHM* UNTUK MENCARI NILAI OPTIMAL

Pada bagian ini, akan disajikan bagaimana *clonal selection* dengan mekanisme seleksi positif dan seleksi negatif dapat diterapkan pada lingkungan simulasi yang dihuni oleh agen-agen buatan untuk menyelesaikan masalah optimasi. Akan disajikan bagaimana pendekatan ini bisa digunakan untuk menyelesaikan traveling salesman problem (TSP).

3.1. Deskripsi Masalah

Analogian antara sistem kekebalan tubuh dan masalah optimasi adalah sebagai berikut. Response dari sistem imune merepresentasikan solusi dan *antigen* merepresentasikan masalah yang harus diselesaikan. Lebih tepatnya, Sel B adalah sebagai agen-agen buatan yang menjelajahi dan mengeksplorasi lingkungan buatan. Patogen adalah sebagai masalah optimasi, dalam kasus ini, masalah optimasi digambarkan oleh *antigen* pada *patogen*. Mekanisme seleksi positif dan seleksi negatif digunakan untuk mengontrol perbanyakan agen dengan mengeliminasi solusi yang buruk atau tidak berguna. Jadi, aturan seleksi positif dan seleksi negatif dapat dipertimbangkan sebagai mekanisme yang tidak hanya memilih solusi yang tepat, tetapi juga mengatur jumlah populasi agen yang tumbuh pada proses kloning .

Sistem Tubuh	Kekebalan	Masalah Optimasi
Patogen		Permasalahan (dalam hal ini graf kota dimana tiap node(kota) digambarkan sebagai antigen)
Respon Tubuh		Solusi (jarak terdekat)
Sel-B		Agen pencari
<i>Clonal Selection</i>		Menciptakan agen pencari baru untuk menjelajahi kota
Seleksi Positif dan Seleksi Negatif		Penyeleksian agen yang buruk/tidak berguna untuk membunuh dirinya sendiri (apoptosis)

Tabel 1. Analogi Sistem Kekebalan Tubuh pada Masalah Optimasi

Ingat bahwa pada sistem imune jumlah sel yang melawan *antigen* meningkat (melalui proses kloning/proliferasi) ketika antigen muncul di dalam tubuh, dan berkurang ketika antigen ini sudah dimatikan. Selama proses ini, sebuah sel merubah sifatnya tubuhnya seperti bergantinya waktunya hidupnya. Jdi, proses proliferasi meningkatkan jumlah agen yang dapat melawan antigen agar dapat menghambat dan menghancurkannya. Dengan kata lain, proliferasi berkorespondensi dengan pembuatan agen baru. Agen baru yang dibuat mempunyai struktur dan sifat yang mirip dengan induknya tetapi tidak sama persis untuk memungkinkan proses adaptasi pada sistem. Apoptosis berkorespondensi dengan program untuk mematikan sel. Mekanisme ini terjadi ketika sel tidak dapat beradaptasi untuk proses pengeliminasian antigen. Jadi sel yang tidak berguna dihancurkan.

Menggunakan sifat dasar dari sistem kekebalan tubuh yaitu *clonal selection* melalui mekanisme seleksi positif dan seleksi negatif, jumlah agen di dalam sistem diatur secara dinamik untuk mencari solusi optimal pada masalah yang diberikan. Faktanya, agen yang diprediksi tidak cocok dapat dimatikan sebelum dilakukan kloning pada agen tersebut. Keputusan itu dibuat pada level agen secara lokal, tidak dibutuhkan kontrol secara global.

3.2 Aplikasi pada TSP

Pada *Traveling SalesPerson Problem* lingkungan buaatannya adalah graf kota dimana node merepresentasikan tiap kota, yang dianalogikan sebagai *antigen* pada patogen. Sel B adalah agen pencari yang bergerak dari satu kota ke kota tetangganya dan dapat mengkloning dirinya atau menghancurkan dirinya sendiri berdasarkan kriteria seleksi positif dan seleksi negatif.

Algoritma ini dimulai dengan meletakkan sebuah agen pencari pada kota asal. Dalam setiap siklus algoritma, agen dapat mengkloning dirinya dan hasil kloning-nya ini bergerak ke kota tetangganya. Ketika suatu agen mencapai kota yang sudah pernah dikunjungi, maka akan memicu *seleksi positif* dan agen hasil kloning tadi dimatikan (solusi tidak berguna).

Sebaliknya, Jika tidak maka agen tadi akan mengkloning dirinya dan hasil kloning ini akan mendapat salinan dari kota-kota yang sudah dikunjungi oleh pendahulunya. Ketika semua agen yang bertahan hidup sudah menyelesaikan perjalanannya (kembali ke kota asal), akan memicu *seleksi negatif* dan diantara agen-agen yang berhasil menyelesaikan tugasnya, akan dipilih yang paling cepat melakukan perjalanan. (agen lain yang tidak berguna dihancurkan).

Untuk menghentikan proses kloning, jalan terbaik adalah untuk mencoba membuat batas bawah dari solusi optimal. Batas bawah ini akan menjadi tanda berhenti untuk agen lain yang masih berusaha mencari solusi lain, karena solusi tercepat sudah ditemukan. Hal ini secara signifikan mengurangi perjalanan yang sia-sia dan mengurangi jumlah populasi agen. Kondisi ini mengabaikan sejumlah jalur pada graf TSP segera ketika jalur ini menjadi tidak mungkin untuk mendapatkan solusi terbaik.

Secara formal, misalkan $C=\{a, \dots, z\}$ adalah himpunan kota-kota, $A=\{(x,y):x,y \in C\}$ adalah ujung-ujung himpunan tersebut. Dan $\delta(x,y)$ adalah jarak antara $(x,y) \in C$. pertimbangkan bahwa vektor-vektor yang terdiri dari elemen di dalam himpunan $C=\{a,\dots,z\}$ merepresentasikan jalur yang mungkin. Masing-masing komponen di dalam vektor merepresentasikan kota. Panjang total dari tiap tur/perjalanan memberikan ukuran afinitas dari vektor yang berkorespondensi.

TSP adalah masalah menemukan nilai afinitas minimum perjalanan yang mengunjungi tiap kota hanya satu kali dan kembali ke kota asal. Menggunakan pendekatan yang diajukan sebelumnya, sifat agen dideskripsikan sebagai berikut

```
Initialisasi
Buat Agen A
A.citiesList = Kota //himpunan kota
A.sourceCity = KotaAsal(dipilih random) // Agen diposisikan di kota asal
A.visitedList={} //himpunan dari kota yang sudah dikunjungi
A.mAffinity= jarak //afinitas maksimal yang dibuat secara random
A.LastCity=A.currentCity //Kota terakhir yang dikunjungi=Kota awal
A.cAffinity=0 //Afinitas awal, dan agen akan berhenti
//jika semua kota sudah dikunjungi

While (A. CitiesList ≠ Null )do
A.cAffinity= A.cAffinity+ + δ(LastCity, currentCity)
If (A.currentCity∉A.visitedList and A.cAffinity < A. mAffinity)
A.visitedList.Add(currentCity)
```

```

A.citiesList.Removed(currentCity)

//Agen mengkloning dirinya sendiri dan bergerak dengan kloningnya
A.LastCity=currentCity
B=A.clone() //jika ada setidaknya dua kota tetangga
//n1 dipilih secara acak dari kota-kota tersebut dengan syarat n1∉visitedList
//n2 dipilih secara random dengan syarat n2∉visitedList-{n1}
B.currentcity= n2; A.move()
Endfor
Else
//positive selection, agen tidak meneruskan perjalanan dan membunuh dirinya
sendiri
A.die() //solusi yang tidak berguna
Endif
Done
//negative selection, agen mati sendiri jika ada agen lain yang memiliki rute
yang lebih baik
A.die() //solusi yang buruk

```

Tidak ada gunanya jika *seleksi positif* diterapkan jika agen tidak membuat rute atau afinitasnya menjadi lebih besar daripada afinitas dari rute perjalanan lain.

Seleksi positif *Seleksi positif* diterapkan jika semua agen yang berhasil membuat rute (solusi-solusi yang mungkin). Dalam kasus ini, hanya agen-agen yang memiliki afinitas terkecil yang akan bertahan dan menjadi solusi yang paling tepat.

4. HASIL KOMPUTANSI

Hasil yang disajikan di tabel 2 menunjukkan waktu yang dibutuhkan dan jarak yang ditempuh yang ditemukan oleh *cloning-based algorithm*

Number of cities	cpu Time	Tour distance
10	0	2,978
20	15	3,296
30	27	3,296
40	32	5,376
50	94	6,208
100	593	7,132
150	1602	8,534
200	4076	9,634
500	12298	11,498
1000	274355	15,234

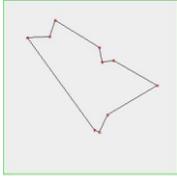
Tabel 2. Hasil Komputansi *Clonal-based Algorithm*

Hasil yang disajikan di tabel 3 menunjukkan waktu CPU dan jarak perjalanan yang ditemukan oleh algoritma semut (*ant-based algorithm*) yang diajukan (Marco Dorigo pada tahun 1996) ketika algoritma ini coba diimplementasikan.

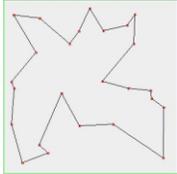
Number of cities	cpu Time	Tour distance
10	15	3,219
20	78	3,385
30	94	3,336
40	1375	5,903
50	3406	6,458
100	9743	7,585
150	94446	9,505
200	606745	10,556
500	2359753	12,357
1000	4350567	15,654

Tabel 3. Hasil Komputansi *Clonal-based Algorithm*

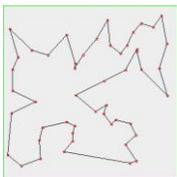
Jika dibandingkan dengan *clonal-based algorithm* atau disebut juga *immune-based algorithm* dapat ditarik kesimpulan bahwa algoritma semut memiliki waktu komputansi yang lebih lama.



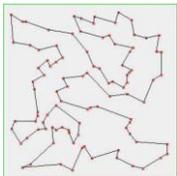
Gambar 1: Rute optimal untuk 10 kota



Gambar 2: Rute optimal untuk 25 kota



Gambar 3: Rute optimal untuk 50 kota



Gambar 4: Rute optimal untuk 100 kota

5. KESIMPULAN

Cloning-based Algorithm adalah algoritma yang terinspirasi dari *cloning-selection* yang terjadi pada sistem kekebalan tubuh manusia. Dan merupakan bagian dari *bio-inspired algorithm* yaitu metode-metode yang terinspirasi dari alam.

Algoritma-algoritma ini dapat dimanfaatkan salah satunya dalam persoalan optimasi. *Cloning-based Algorithm* merupakan terobosan baru untuk menyelesaikan masalah optimasi. Percobaan yang dilakukan dalam kasus ini digunakan untuk menyelesaikan masalah TSP (Travelling Sales Problem) telah menunjukkan bahwa algoritma ini memiliki efisiensi yang lebih baik dari algoritma yang diajukan sebelumnya yaitu algoritma semut.

DAFTAR REFERENSI

- [1] Bakhouya M., J. Gaber, "An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salesman Problem" *AMO - Advanced Modeling and Optimization*, Volume 9, Number 1, 2007, hal 105-116
- [2] Denny Hermawanto, "Komputansi Sistem Kekebalan Tubuh"
- [3] Munir, Rinaldi. (2006). *Diktat Kuliah IF2153 Matematika Diskrit. Program Studi Teknik Informatika, Institut Teknologi Bandung*. Hal 44-50
- [4] Wikipedia (2007). "bio-inspired computing – Wikipedia." http://id.wikipedia.org/wiki/bio-inspired_computing
Tanggal akses: 28 Desember 2007 pukul 16:00.