

OPTIMASI RUTE ANGKUTAN KOTA MALANG DENGAN PENERAPAN ALGORITMA GENETIKA

Khoiron Nisaa¹ ; Muh. Arif Rahman² ; Achmad. Ridok³

Program Studi Informatika / Ilmu Komputer, Program Teknologi Informasi dan Ilmu Komputer
Universitas Brawijaya Malang

Jl. Veteran No.8 Malang, 65145, Indonesia

Email : nisaelmuzammil@gmail.com¹; m_arif@ub.ac.id²; acridokg@gmail.com³

ABSTRAK

Angkutan kota sebagai angkutan umum yang murah dan menjangkau sebagian besar masyarakat kelas menengah kebawah masih banyak beroperasi dan diburu oleh masyarakat. Banyaknya angkutan kota yang beroperasi menimbulkan masalah tersendiri dalam hal pengaturan rute trayek angkutan kota dan pengoptimalan penggunaan bahan bakar. Pengoptimalan rute angkutan kota dapat dilakukan dengan optimasi rute angkutan kota, studi penentuan rute trayek angkutan kota Malang dengan berdasarkan pada jarak tempuh dan penumpukan (*intersection*) jalur antar trayek dengan mempertahankan bahwa setiap trayek harus memenuhi titik awal dan titik tujuan yang telah ditetapkan. Salah satu cara untuk menyelesaikan permasalahan optimasi tersebut yaitu dengan penerapan algoritma genetika. Pada algoritma genetika terdapat 4 parameter yang mempengaruhi hasil *fitness* yang diperoleh yaitu jumlah populasi, probabilitas *crossover*, probabilitas mutasi dan jumlah generasi. Pada penelitian ini dilakukan analisa pengaruh parameter genetika terhadap nilai *fitness* rata – rata yang diperoleh. Hasil penelitian menunjukkan bahwa perubahan parameter genetika probabilitas *crossover* (*pc*) dan jumlah generasi mempunyai pengaruh yang cukup signifikan terhadap kenaikan nilai *fitness* yang diperoleh, yaitu semakin besar nilai probabilitas *crossover*, nilai *fitness* mengalami kenaikan (nilai *fitness* tinggi) , begitu juga untuk jumlah generasi. Dan dari nilai *fitness* terbaik yang diperoleh pada pengujian dengan *pc* : 90%, *pm* : 30%, jumlah populasi : 30 dan jumlah generasi : 100 diperoleh hasil bahwa sebanyak 19 dari 25 trayek, memenuhi jarak tempuh yang telah ditentukan serta terdapat 95 total tumpukan jalur dari 176 tumpukan jalur pada keadaan yang sebenarnya.

Kata Kunci : Optimasi Rute, Angkutan Kota Malang, Algoritma Genetika

1. Pendahuluan

Kota Malang sebagai kota terbesar kedua di Jawa Timur memiliki tingkat pertumbuhan penduduk 3,9% per tahun. cDalam basis data dinas kependudukan menyebutkan bahwa penduduk kota malang per tanggal 25 Maret 2011 berjumlah 894.653 jiwa [1]. Laju peningkatan penduduk ini menjadi faktor yang berpengaruh terhadap peningkatan kendaraan bermotor dan tidak menutup kemungkinan meningkatnya pula jumlah transportasi umum. Meningkatnya jumlah kendaraan mengakibatkan semakin tingginya tingkat mobilitas sehingga semakin tinggi pula persaingan dalam hal penggunaan jalan. Peningkatan kegiatan transportasi memberikan banyak dampak negatif terhadap kota, antara lain : Kemacetan dan eksploitasi sumber energi

khususnya BBM [2]. Hal ini dapat dikurangi dengan pengoptimalan sistem transportasi, salah satunya adalah angkutan kota.

Angkutan kota sebagai angkutan umum yang murah dan menjangkau sebagian besar masyarakat kelas menengah kebawah masih banyak beroperasi dan diburu oleh masyarakat. Banyaknya angkutan kota yang beroperasi menimbulkan masalah dalam hal pengaturan rute trayek angkutan kota dan pengoptimalan penggunaan bahan bakar.

Pengoptimalan rute angkutan kota dapat dilakukan dengan optimasi rute angkutan kota, studi penentuan rute trayek angkutan kota Malang dengan berdasarkan pada jarak tempuh dan penumpukan (*intersection*) jalur antar trayek dengan tidak melupakan bahwa setiap trayek harus memenuhi titik awal dan titik tujuan yang

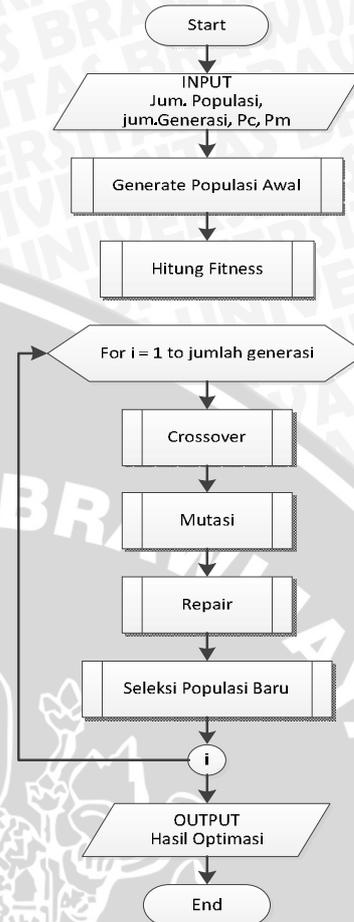
telah ditetapkan. Permasalahan optimasi mengacu pada sebuah metode yang dapat melaksanakan tugas secara optimal, dapat diselesaikan secara cepat dengan memilih solusi yang terbaik [3]. Salah satu cara untuk menyelesaikan permasalahan optimasi yaitu dengan algoritma genetika. Oleh karena itu dalam penelitian ini diterapkan algoritma genetika untuk penyelesaian permasalahan optimasi rute angkutan kota Malang.

Algoritma genetika merupakan salah satu algoritma yang memiliki kemampuan intelegen, dalam hal ini adalah algoritma stokastik dengan memanfaatkan fenomena alam [4]. Selain itu algoritma genetika sebagai cabang dari algoritma evolusi merupakan metode *adaptive* yang biasa digunakan untuk memecahkan suatu pencarian nilai dalam sebuah permasalahan optimasi [5].

Penerapan algoritma genetika sebelumnya juga telah diterapkan dalam penelitian untuk menentukan jalur kereta api yang optimal [6]. Namun dalam penelitian ini penerapan algoritma genetika akan dilakukan untuk menentukan rute angkutan kota Malang yang optimal dengan tujuan meminimalkan jarak tempuh dan meminimalkan penumpukan (*intersection*) jalur antar trayek.

2. Metode Penelitian

Dalam penelitian ini sistem menerima masukan berupa parameter genetika yang terdiri atas jumlah populasi, jumlah generasi, probabilitas *crossover* (*pc*) dan probabilitas mutasi (*pm*). Selanjutnya dilakukan tahapan – tahapan algoritma genetika yang terdiri atas *generate* populasi awal, hitung nilai *fitness*, *crossover*, mutasi, *repair* dan seleksi populasi baru. Secara visual proses algoritma genetika yang digunakan dalam penelitian ini digambarkan pada gambar 1.



Gambar 1. Alur Proses Algoritma Genetika

2.1. Pengkodean

Metode pengkodean yang digunakan dalam penelitian ini adalah metode pengkodean permutasi. *Gen* dalam individu merupakan representasi sebuah *node* dalam graf yang telah ditentukan. Setiap *gen* direpresentasikan dalam *type* data *integer*. Panjang dari trayek adalah dinamis, dengan berdasarkan pada jalur atau rute yang telah terpilih dari hasil *random*.

Pada penelitian ini, 1 individu merupakan satu solusi, dimana suatu solusi merupakan representasi dari keseluruhan trayek angkutan kota Malang, yang terdiri atas 25 trayek. Dalam individu mengandung *gen* – *gen* yang merepresentasikan rute setiap trayek angkutan kota.

Setiap individu akan direpresentasikan menggunakan *array* yang didalamnya memuat *arraylist*. Dimana dalam *arraylist* tersebut memuat *gen* - *gen* dengan panjang yang berbeda (dinamis), disesuaikan dengan rute yang telah terpilih dari hasil *random*. Ilustrasi representasi individu digambarkan pada gambar 2

K1	S1	N ₁	N ₂	...	E1		
K2	S2	N ₃	...		E2		
K3	S3	N ₄				E3	
K4	S4	...					E4
K...	S...	...					E..
K...	S...						E..
K25	S25						E25

Gambar 2. Ilustrasi Representasi Individu

Keterangan Gambar:

- S = *Start point* (Lokasi Awal)
- E = *End point* (Lokasi Tujuan)
- N = *Node* (Persimpangan Jalan)
- K = Trayek

2.2. Generate Populasi Awal

Proses *generate* populasi awal merupakan proses pembentukan individu sebanyak jumlah populasi awal yang telah ditentukan. Dalam penelitian ini 1 solusi adalah 1 individu, dimana 1 individu terdiri atas *gen - gen* yang merepresentasikan rute trayek angkutan kota. Jumlah populasi individu akan ditentukan oleh *user* melalui *input* yang diinginkan.

Didalam proses *generate* populasi awal akan diikuti pula pembentukan individu. Individu yang terbentuk dalam proses *generate* populasi dilakukan melalui mekanisme *random* dengan terlebih dahulu ditetapkan beberapa *gen transit*. *Random* dilakukan pada setiap *gen transit* hingga membentuk suatu rute trayek yaitu telah memenuhi *node* tujuan.

2.3. Nilai Fitness

Fungsi *fitness* dalam penelitian ini adalah fungsi *fitness* untuk menentukan rute trayek angkutan kota dengan meminimalkan jarak tempuh dan meminimalkan jumlah tumpukan jalur. Dalam permasalahan ini suatu nilai *fitness* yang baik dan dianggap sebagai solusi adalah nilai *fitness* yang besar.

Perhitungan nilai *fitness* dilakukan dengan memberikan nilai penalti untuk setiap aturan yang digunakan dalam optimasi rute angkutan kota Malang. Aturan yang semakin wajib untuk dilaksanakan, maka semakin besar nilai penalti yang diberikan, dan sebaliknya. Oleh karena itu, apabila suatu individu memiliki nilai penalti yang besar maka individu tersebut tidak dapat ditetapkan sebagai solusi sebab individu tersebut memiliki nilai *fitness* yang kecil. Adapun fungsi *fitness* yang digunakan adalah :

$$F = \frac{1}{(\sum_{i=1}^3 (p_i \times f)) + 1} \quad (1)$$

Keterangan :

- F = nilai *fitness*
- p_i = penalti pelanggaran yang diberikan pada aturan ke-i
- f = frekuensi pelanggaran pada aturan ke-i

Adapun aturan dan nilai penalti yang diterapkan dalam penelitian ini adalah :

Tabel 1 Aturan dan Nilai Penalti

No.	Aturan	Penalti
	Jarak tempuh trayek AG/AH > 14 km	
	Jarak tempuh trayek AL > 17 km	
	Jarak tempuh trayek AT > 18 km	
	Jarak tempuh trayek ABB > 16 km	
	Jarak tempuh trayek ABG/ABH > 26 km	
	Jarak tempuh trayek ADL > 15 km	
	Jarak tempuh trayek AJG/AJH > 18 km	
	Jarak tempuh trayek AMG/AMH > 16 km	
	Jarak tempuh trayek ASD > 10 km	
	Jarak tempuh trayek CKL > 22 km	
1	Jarak tempuh trayek GA/HA > 14 km	10
	Jarak tempuh trayek GL/HL > 16 km	
	Jarak tempuh trayek GM/HM > 8 km	
	Jarak tempuh trayek GML/HML > 18 km	
	Jarak tempuh trayek JDM > 13 km	
	Jarak tempuh trayek JPK > 10 km	
	Jarak tempuh trayek LG/LH > 17 km	
	Jarak tempuh trayek LDG/LDH > 15 km	
	Jarak tempuh trayek MK > 11 km	
	Jarak tempuh trayek MM > 15 km	
	Jarak tempuh trayek MT > 9	

	km	
	Jarak tempuh trayek MKS > 7 km	
	Jarak tempuh trayek TGT > 6 km	
	Jarak tempuh trayek TSG > 10 km	
	Jarak tempuh trayek TST > 16 km	
2	Suatu jalur (<i>node</i> terhubung) dilewati lebih dari 3 kali	10
3	Jarak tempuh tiap trayek angkutan < 5 km	5

Proses perhitungan nilai *fitness* dimulai dengan melakukan perhitungan nilai penalti. Dimana perhitungan nilai penalti dilakukan sesuai jumlah aturan yaitu 3. Untuk perhitungan penalti aturan 1 dan 3 terlebih dahulu akan dilakukan perhitungan jarak tempuh pada setiap trayek.

2.4. Crossover

Proses *crossover* merupakan proses penyilangan 2 atau lebih individu (*parent*) untuk mendapatkan suatu keturunan yang disebut *child* atau *offspring*. Sebelum dilakukan proses *crossover* terlebih dahulu ditentukan kandidat *parent*. Penentuan kandidat *parent* dilakukan dengan 2 cara, yaitu elitism dan berdasarkan probabilitas *crossover*. Dari hasil tersebut setiap kandidat *parent* akan dikombinasikan untuk kemudian dilakukan proses *crossover*.

Metode *crossover* yang digunakan adalah modifikasi dari metode *crossover* yang digunakan dalam penelitian oleh Karas dan Atila, 2011 [7] dan digunakan pula oleh Yan Mahdi Saranta, 2011 [8]. Adapun langkah – langkah proses *crossover* yang digunakan adalah :

1. Melakukan pencarian *gen – gen* yang sama dalam trayek, mulai dari trayek ke-1.
2. Menentukan titik *crossover* dengan cara mengambil *gen* yang sama dan pertama kali ditemukan dalam trayek pada *parent* 1 dan 2 kecuali *gen* pertama dan terakhir.
3. Apabila tidak terdapat *gen* yang sama dalam trayek ke-1 pada *parent* 1 dan 2, lanjutkan pada trayek berikutnya dan ulangi mulai langkah 1.
4. Menukarkan *gen - gen* antara 2 trayek mulai dari titik *crossover*.
5. Mengulangi langkah 1 – 4 hingga memenuhi trayek ke-25.

2.5. Mutasi

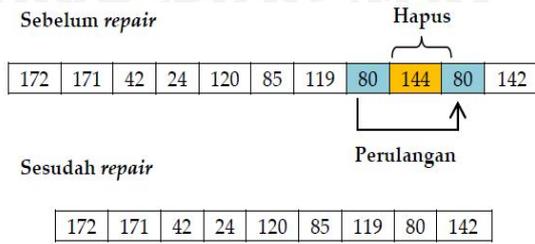
Proses mutasi dilakukan dengan tujuan untuk meningkatkan variasi individu dalam populasi atau mencegah konvergensi dini. Proses mutasi didasarkan pada nilai probabilitas mutasi (*pm*) sebagai nilai peluang besarnya individu yang akan mengalami mutasi. Metode mutasi yang digunakan adalah mutasi *random*, dengan memodifikasi dari penelitian yang telah dilakukan oleh Lamhot Pranata, 2011 [9]. Adapun langkah – langkah proses mutasi yang digunakan adalah sebagai berikut :

1. *Generate* bilangan *random* [0..1] sebanyak jumlah individu populasi awal ditambah *child* atau *offspring* untuk menentukan individu yang akan dimutasi.
2. Apabila hasil *generate* bilangan *random* < probabilitas mutasi (*pm*), maka lakukan langkah 4.
3. Apabila hasil *generate* bilangan *random* > probabilitas mutasi (*pm*), maka tidak terjadi mutasi dan keturunan sama dengan induk.
4. *Generate* bilangan *random* [0..1] sebanyak jumlah total trayek dari individu terpilih untuk menentukan trayek yang akan dimutasi.
5. Menentukan titik mutasi secara *random* *gen*, pada trayek terpilih kecuali pada *gen* pertama dan terakhir.
6. Menentukan *gen – gen* yang terhubung dengan *gen* sebelum dan sesudah titik mutasi.
7. Memilih *gen – gen* yang terhubung.
8. Mengganti nilai *gen* pada titik mutasi dengan *gen – gen* baru yang terhubung.

2.6. Repair

Proses *repair* merupakan proses untuk memperbaiki individu sehingga mendapatkan individu yang valid dan siap untuk diseleksi guna mendapatkan generasi baru. Individu yang dianggap valid adalah apabila telah memenuhi kriteria bahwa tidak ada *gen* yang berulang dalam 1 trayek.

Untuk dapat memenuhi kriteria valid diatas maka dilakukan proses *repair*. Proses *repair* dilakukan dengan cara menghapus *gen – gen* yang berada pada perulangan. Berikut adalah ilustrasi gambar dari proses *repair* :



Gambar 3. Ilustrasi Proses Repair

Adapun langkah – langkah dari proses repair adalah sebagai berikut :

1. Mengambil 1 gen ke k dan menyimpannya sementara
2. Melakukan pengecekan terhadap trayek apakah mengandung gen k.
3. Apabila masih mengandung gen k maka hanya 1 gen yang diambil dan menghapus gen – gen dalam perulangan dan menyisipkan kembali gen k.
4. Apabila tidak mengandung gen k maka gen k langsung disisipkan kembali.

2.7. Seleksi Populasi Baru

Proses terakhir dari algoritma genetika adalah penentuan populasi baru. Dalam penelitian ini penentuan populasi baru dilakukan dengan cara seleksi terhadap seluruh individu, baik dari individu dalam populasi awal hingga *child* atau *offspring* dari hasil *crossover* dan mutasi. Metode seleksi yang digunakan adalah *rank base selection* (seleksi rangking). Adapun langkah – langkah dari seleksi rangking adalah :

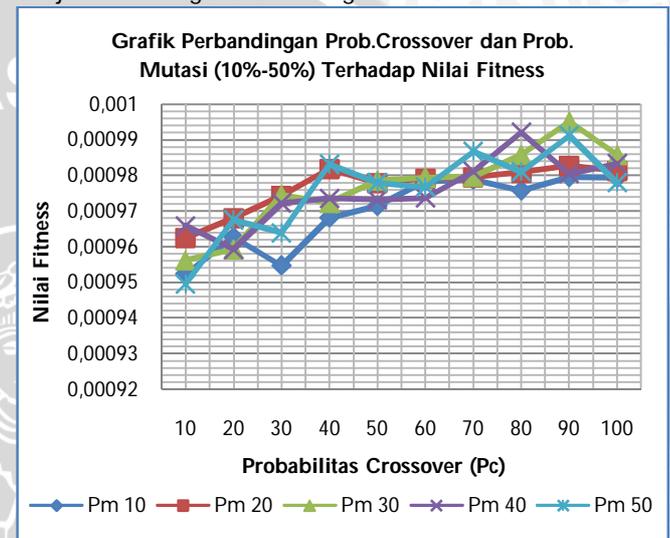
1. Menambahkan hasil *child* dari proses *crossover* dan mutasi kedalam populasi awal.
2. Mengurutkan populasi awal yang telah ditambahkan berdasarkan nilai *fitness*-nya,
3. Memilih individu – individu baru yang memiliki nilai *fitness* terbaik sebanyak jumlah populasi awal.

3. Hasil dan Analisa Pembahasan

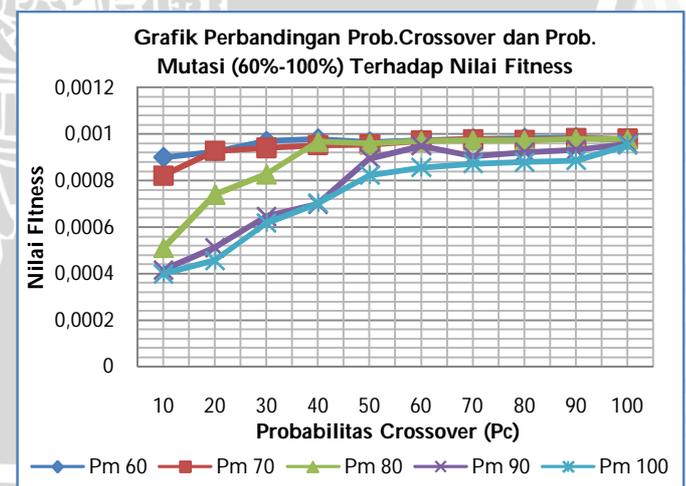
Pengujian dilakukan untuk mengetahui hasil optimasi rute angkutan kota dengan penerapan algoritma genetika. Proses pengujian dilakukan dengan mengombinasikan parameter – parameter genetika (probabilitas *crossover*, probabilitas mutasi, jumlah populasi dan jumlah generasi). Tujuan dari uji coba adalah untuk mengetahui pengaruh parameter genetika terhadap hasil optimasi rute angkutan kota, dalam hal ini adalah nilai *fitness*-nya. Terdapat 3 macam pengujian yang dilakukan yaitu uji

probabilitas *crossover* dan probabilitas mutasi, uji ukuran populasi dan uji jumlah generasi.

Pengujian terhadap pengaruh probabilitas *crossover* dan probabilitas mutasi dilakukan pada rentang nilai 10% - 100% untuk masing probabilitas *crossover* dan probabilitas mutasi. Setiap kombinasi nilai probabilitas *crossover* dan probabilitas mutasi dilakukan ujicoba sebanyak 5 kali dan diambil nilai rata – rata *fitness*. Tabel hasil uji coba probabilitas *crossover* dan probabilitas mutasi yang disajikan dalam lampiran 1. Sedangkan grafik hasil uji coba disajikan dalam gambar 4 dan gambar 5.



Gambar 4. Grafik Perbandingan Prob.Crossover dan Prob. Mutasi (10%-50%) Terhadap Nilai Fitness



Gambar 5. Grafik Perbandingan Prob.Crossover dan Prob. Mutasi (60%-100%) Terhadap Nilai Fitness

Berdasarkan tabel hasil pengujian pada lampiran dapat dilihat bahwa nilai *fitness* terbesar adalah 0.000995. Nilai ini didapat pada saat probabilitas *crossover* 90% dan probabilitas mutasi 30%.

Berdasarkan kedua grafik perbandingan probabilitas *crossover* dan probabilitas mutasi pada gambar 4 dan 5, secara keseluruhan dapat dilihat bahwa seiring bertambahnya nilai probabilitas *crossover* grafik yang dihasilkan cenderung naik. Hal ini berarti bahwa seiring bertambahnya nilai probabilitas *crossover* maka nilai *fitness* cenderung semakin besar. Perubahan nilai *fitness* berdasar probabilitas *crossover* juga dipengaruhi oleh probabilitas mutasi sehingga tidak semua nilai *fitness* naik untuk setiap kenaikan probabilitas *crossover*. Berdasarkan grafik pada gambar 4, menunjukkan hampir semua grafik menunjukkan kondisi yang *fluktuatif*. Akan tetapi kondisi *fluktuatif* jelas terlihat ketika probabilitas mutasi 50%, kenaikan nilai probabilitas *crossover* mengakibatkan nilai *fitness* yang dihasilkan mengalami penurunan dan kenaikan. Penurunan terjadi ketika nilai probabilitas *crossover* 30%, 60% dan 80%.

Perubahan nilai *fitness* berdasar kenaikan probabilitas *crossover* dapat disebabkan karena semakin seringnya suatu individu mengalami proses persilangan sehingga variasi individu baru akan semakin banyak. Telah diketahui bahwa semakin besar nilai probabilitas *crossover* maka peluang individu yang mengalami proses *crossover* akan semakin besar, dan tentunya akan semakin banyak individu – individu baru yang dihasilkan. Dengan demikian nilai *fitness* juga semakin bervariasi, sehingga peluang untuk mendapatkan individu dengan nilai *fitness* yang besar juga akan semakin besar.

Berbeda halnya dengan probabilitas *crossover*, dari grafik diatas terlihat bahwa perubahan nilai probabilitas mutasi menghasilkan grafik tak tentu, dalam artian mengalami kenaikan dan penurunan dalam kondisi probabilitas *crossover* sama. Grafik *fluktuatif* terlihat jelas pada gambar 4, pada grafik ini seiring bertambahnya nilai probabilitas mutasi tidak diiringi bertambahnya nilai *fitness*. Nilai *fitness* yang dihasilkan adalah tak tentu, dalam artian mengalami kenaikan dan penurunan. Hal ini terlihat jelas ketika probabilitas *crossover* 30%, pada keadaan ini nilai *fitness* mengalami kenaikan ketika probabilitas mutasi 10% hingga 20%. Kemudian mengalami penurunan ketika 30% hingga 50%. Penurunan yang cukup signifikan terjadi ketika probabilitas mutasi 50%. Penurunan nilai *fitness* terus terjadi hingga nilai probabilitas mutasi mencapai 100%. Pada grafik gambar 4.8 terlihat bahwa seiring bertambahnya nilai probabilitas

mutasi menghasilkan nilai *fitness* yang cenderung turun. Jelas terlihat bahwa ketika dalam kondisi probabilitas *crossover* sama, nilai *fitness* cenderung mengalami penurunan. Sehingga dapat disimpulkan semakin besar nilai probabilitas mutasi maka nilai *fitness* cenderung mengalami penurunan. Hal ini dapat disebabkan karena metode mutasi yang *random* dan dengan menyisipkan gen – gen baru, sehingga memungkinkan individu memiliki jarak tempuh yang semakin jauh dan memungkinkan pula naiknya jumlah tumpukan jalur. Tentunya hal ini berakibat semakin besarnya penalti pelanggaran, sehingga nilai *fitness* semakin kecil.

Pengujian berikutnya adalah uji pengaruh ukuran populasi terhadap nilai *fitness* yang dihasilkan. Pengujian dilakukan dengan menggunakan 6 nilai uji yaitu 10, 20, 30, 40, 50 dan 60. Untuk setiap nilai uji dilakukan 5 kali uji coba untuk kemudian diambil rata – rata. Hasil rata – rata nilai *fitness* dari uji coba ukuran populasi disajikan dalam tabel 2.

Tabel 2. Hasil Uji Coba Ukuran Populasi

No.	Generasi : 30 ; Pc : 90% ; Pm : 40%	
	Populasi	Rata – Rata <i>Fitness</i>
1	10	0.00092524
2	20	0.0009543
3	30	0.00095292
4	40	0.00098702
5	50	0.00100042
6	60	0.000997

Pada tabel diatas terlihat bahwa nilai *fitness* terbesar dihasilkan oleh pengujian dengan ukuran populasi 50 yaitu 0.00100042. Sedangkan nilai *fitness* terkecil diperoleh pada saat ukuran populasi 10. Adapun grafik pengaruh ukuran populasi terhadap nilai *fitness* disajikan pada gambar 6.



Gambar 6. Grafik Uji Ukuran Populasi Terhadap *Fitness*

Grafik perbandingan ukuran populasi dengan nilai *fitness* pada gambar 6 cenderung naik. Hal ini menunjukkan bahwa seiring bertambahnya ukuran populasi nilai *fitness* juga semakin besar. Dalam grafik diatas menunjukkan bahwa nilai *fitness* mengalami penurunan ketika jumlah populasi 30 dan naik kembali hingga jumlah populasi 50. Akan tetapi mengalami penurunan kembali ketika jumlah populasi 60.

Secara teori ukuran populasi yang besar memungkinkan terdapat variasi individu yang beragam dan ukuran populasi yang kecil memungkinkan variasi individu sedikit (tidak beragam). Akan tetapi dari grafik diatas tidak menjamin bahwa ukuran populasi yang besar menghasilkan nilai *fitness* yang besar dan ukuran populasi yang kecil menghasilkan nilai *fitness* yang kecil. Semakin besar ukuran populasi yang digunakan maka kemungkinan untuk menemukan solusi cenderung menjadi lambat. Sedangkan ukuran populasi yang terlalu kecil memungkinkan semakin kecilnya ruang pencarian yang dapat dieksplorasi ketika proses rekombinasi (*crossover* dan mutasi).

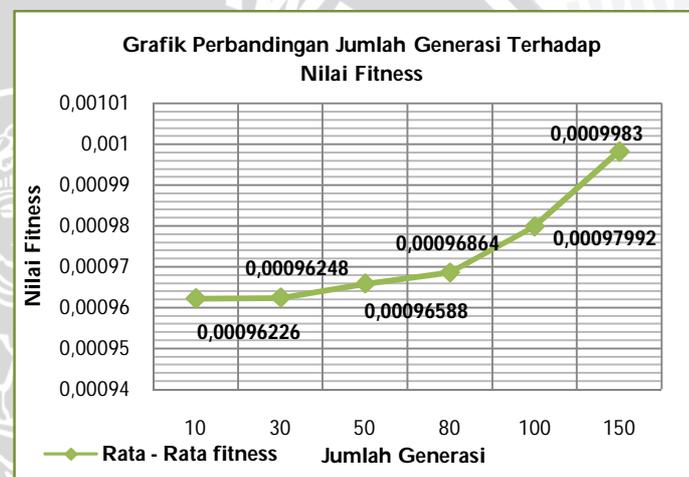
Hal ini dapat disebabkan pada saat pemilihan kandidat *parent* untuk proses *crossover*. Oleh karena pemilihan dilakukan secara *random* dengan berdasarkan probabilitas *crossover*, maka tidak dapat dipastikan bahwa individu yang terpilih adalah individu dengan nilai *fitness* besar dari proporsi jumlah individu yang besar. Sehingga apabila kandidat *parent* yang terpilih adalah individu dengan nilai *fitness* kecil, anak yang dihasilkan dari proses rekombinasi juga memiliki kemungkinan bernilai *fitness* kecil. Hal yang sama pada ukuran populasi yang kecil dimana variasi individu-individu di dalamnya sedikit. Individu yang akan terpilih sebagai kandidat *parent* akan memiliki variasi terbatas, dan anak yang dihasilkan bisa jadi memiliki sifat yang mirip dengan nilai *fitness* yang hampir dekat (tidak mengalami improvisasi).

Pengujian terakhir yaitu uji pangaruh jumlah generasi terhadap nilai *fitness* yang dihasilkan. Pengujian dilakukan dengan menggunakan 6 nilai uji yaitu 10, 30, 50, 80, 100 dan 150. Untuk setiap nilai uji dilakukan 5 kali uji coba untuk kemudian diambil rata – rata. Hasil rata – rata nilai *fitness* dari uji coba jumlah generasi disajikan dalam tabel 3.

Tabel 3. Hasil Uji Coba Jumlah Generasi

No.	Populasi: 30 ; Pc : 90% ; Pm : 40%	
	Generasi	Rata – Rata <i>Fitness</i>
1	10	0.00096226
2	30	0.00096248
3	50	0.00096588
4	80	0.00096864
5	100	0.00097992
6	150	0.0009983

Pada tabel diatas terlihat bahwa nilai *fitness* terbesar dihasilkan oleh pengujian dengan jumlah generasi 150 yaitu 0.0009983. Sedangkan nilai *fitness* terkecil diperoleh pada saat jumlah generasi 10. Adapun grafik pengaruh jumlah generasi terhadap nilai *fitness* disajikan pada gambar 7.



Gambar 7. Grafik Uji Jumlah Generasi Terhadap *Fitness*

Grafik pada gambar 7 dapat dilihat bahwa nilai *fitness* tidak mengalami perubahan yang signifikan ketika jumlah generasi 10 hingga 80. Selanjutnya mengalami kenaikan ketika jumlah generasi 100. Kenaikan mulai terlihat signifikan ketika jumlah generasi 150. Sehingga secara keseluruhan dapat dikatakan bahwa nilai *fitness* cenderung naik seiring bertambahnya jumlah generasi.

Jumlah generasi yang tinggi akan mengakibatkan proses evolusi semakin sering dilakukan. Pada setiap satu generasi, akan dilakukan proses rekombinasi yang terdiri atas *crossover* dan mutasi. Sehingga semakin banyak jumlah generasi maka proses rekombinasi akan semakin sering dilakukan. Tentunya hal ini akan berpengaruh pula terhadap individu – individu baru yang dihasilkan. Semakin banyak melakukan proses *crossover* dan mutasi maka individu – individu baru yang dihasilkan akan semakin bervariasi dan memungkinkan pula

bervariasinya nilai *fitness* yang dihasilkan. Dengan begitu akan memberikan peluang yang besar untuk mendapatkan nilai *fitness* yang baik.

4. Kesimpulan

Berdasarkan hasil pengujian dan analisa pembahasan diatas maka dapat disimpulkan bahwa :

1. Algoritma genetika dapat diterapkan dalam permasalahan optimasi rute angkutan kota, studi penentuan rute trayek angkutan kota Malang dengan berdasarkan pada jarak tempuh dan penumpukan (*intersection*) jalur antar trayek. Dengan metode yang digunakan adalah pengkodean permutasi, *crossover* modifikasi, mutasi *random* dan seleksi *rank base fitness*.
2. Penyelesaian permasalahan optimasi rute angkutan kota Malang dengan penerapan algoritma genetika dipengaruhi oleh beberapa parameter algoritma genetika, yaitu :
 - Pengaruh parameter probabilitas *crossover* menunjukkan bahwa semakin besar nilai probabilitas *crossover* maka nilai *fitness* cenderung semakin tinggi (baik)
 - Pengaruh parameter jumlah generasi menunjukkan bahwa semakin besar jumlah generasi yang digunakan, maka nilai *fitness* juga cenderung semakin tinggi (baik).
 - Pengaruh parameter probabilitas mutasi menunjukkan perubahan nilai probabilitas mutasi yang semakin tinggi mengakibatkan penurunan nilai *fitness*.
 - Sedangkan pengaruh jumlah populasi menunjukkan bahwa jumlah populasi antara 10 hingga 50 menghasilkan nilai *fitness* yang cenderung naik. Akan tetapi jumlah populasi diatas 50 mengakibatkan penurunan nilai *fitness*.
3. Pada permasalahan optimasi rute angkutan kota dengan menggunakan 548 data jalan terhubung yang terdiri atas 367 *node* percabangan, algoritma genetika mampu menyelesaikan permasalahan optimasi rute angkutan kota Malang dengan hasil pengujian probabilitas *crossover* dan probabilitas mutasi terbaik yang diperoleh yaitu jarak tempuh trayek yang memenuhi aturan sebanyak 19 dari 25 trayek. Sehingga terdapat 6 trayek yang tidak memenuhi aturan. Dan terdapat 95 total tumpukan jalur dari 176 tumpukan jalur pada keadaan yang sebenarnya.

DAFTAR PUSTAKA

- [1] Dinas Kependudukan dan Pencatatan Sipil. 2012. www.pemkot-malang.go.id. Diakses tanggal 1 Maret 2012.
- [2] Handajani, Mudjiastuti. 2009. *Analisis Gradien Kepadatan Penduduk dan Konsumsi BBM*. Jurnal Teknik Sipil & Perencanaan, Nomor 2 Volume 11 – Juli 2009, hal : 141 – 148
- [3] Setiawan, Kuswara. 2003. *Paradigma Sistem Cerdas*. Bayumedia. Surabaya.
- [4] Michalewicz, Z. 1999. *Genetic Algorithms + Data Structure = Evolution Programs*. Springer Verlag Berlin Heidelberg. New York.
- [5] Anonim. 2012. *Bab 7 Algoritma Genetika*. <http://lecturer.eepis-its.edu/~entin/Kecerdasan%20Buatan/Buku/Bab%207%20Algoritma%20Genetika.pdf> . Diakses tanggal 29 Februari 2012.
- [6] Budyanto. 2007. *Menentukan Jalur Kereta Api yang Optimal Menggunakan Algoritma Genetika*. SKRIPSI. Fakultas MIPA. Universitas Brawijaya.
- [7] Karas, Ismail Rakip and Atila Umit. 2011. *A Genetic Algorithm Approach For Finding The Shortest Driving Time On Mobile Devices*. Scientific Research and Essays Vol 6(2), pp. 394-405.
- [8] Saranta, Yan Mahdi. 2011. *Pencarian Rute Tercepat Kendaraan Bermotor Roda Empat Menggunakan Algoritma Genetika*. SKSIPSI. Fakultas MIPA. Universitas Brawijaya.
- [9] Pranata, Lamhot. 2011. *Penerapan Algoritma Genetika Pada Penentuan Komposisi Pakan Ikan*. SKRIPSI. Fakultas MIPA. Universitas Brawijaya.

Lampiran 1

Tabel 1. Hasil Uji Coba Probabilitas Crossover dan Probabilitas Mutasi

Pop : 30 ; Generasi : 100										
Pm	Pc (%)									
(%)	10	20	30	40	50	60	70	80	90	100
10	0.000952	0.000963	0.000955	0.000968	0.000972	0.000978	0.000979	0.000976	0.00098	0.000979
20	0.000963	0.000968	0.000974	0.000982	0.000978	0.000979	0.00098	0.000981	0.000983	0.000981
30	0.000956	0.000959	0.000975	0.000972	0.000979	0.000979	0.00098	0.000986	0.000995	0.000986
40	0.000966	0.000959	0.000972	0.000974	0.000973	0.000974	0.000981	0.000992	0.000981	0.000983
50	0.00095	0.000968	0.000964	0.000983	0.000978	0.000977	0.000987	0.000981	0.000991	0.000978
60	0.0009	0.000925	0.000971	0.000981	0.000966	0.000975	0.00098	0.000983	0.000985	0.000977
70	0.000822	0.000927	0.00094	0.000953	0.000955	0.00097	0.000978	0.000974	0.000982	0.000978
80	0.00051	0.00074	0.000826	0.000968	0.000957	0.00097	0.000971	0.000973	0.000976	0.000978
90	0.000415	0.000511	0.000645	0.000698	0.000894	0.000949	0.000903	0.000921	0.000931	0.000957
100	0.000399	0.000456	0.000617	0.000702	0.000823	0.000856	0.000872	0.000879	0.000887	0.000952

