

PENENTUAN JARAK TERPENDEK PADA JALUR DISTRIBUSI BARANG DI PULAU JAWA DENGAN MENGGUNAKAN ALGORITMA GENETIKA

I Dewa Made Adi Baskara Joni¹, Vivine Nurcahyawati²

¹STMIK STIKOM Indonesia, ²STMIK STIKOM Surabaya

E-mail : ¹dewadi.414@gmail.com, ²vivine@stikom.edu

Abstraksi

Terdapat banyak kemungkinan kombinasi jalur distribusi barang melalui darat yang dapat digunakan untuk mengoptimalkan waktu dan biaya perjalanan. Namun demikian, tidak semua kombinasi jalur distribusi tersebut akan memberikan solusi terbaik. Agar solusi terbaik dapat dicapai, suatu penelitian untuk menentukan jalur distribusi barang melalui jalur darat dilakukan. Untuk mempermudah proses penentuan jalur distribusi barang tersebut didukung dengan pembangunan perangkat lunak.

Algoritma genetika memiliki kehandalan dalam menghasilkan output yang optimal dapat dimanfaatkan untuk memecahkan masalah tersebut. Penerapan metode algoritma genetika diaplikasikan dalam suatu perangkat lunak. Pada perangkat lunak yang dibangun terdapat beberapa input yang dibutuhkan, yaitu: kota-kota tujuan distribusi sebagai jumlah kromosom awal, jumlah generasi, probabilitas crossover dan probabilitas mutasi. Hasil pemrosesan merupakan kombinasi jalur distribusi barang yang akan diambil yang merepresentasikan penyelesaian masalah ini. Hanya kromosom terbaik yang akan diberikan sebagai hasil. Melalui perangkat lunak yang dibangun, penentuan jalur distribusi barang diharapkan dapat dilakukan dengan lebih baik dan dapat mengoptimalkan waktu dan biaya perjalanan.

Berdasarkan penelitian yang telah dilakukan, digunakan 5 kombinasi kota sebagai tujuan distribusi yang digunakan pada pengujian. Melalui pengujian yang telah dilakukan terhadap tiga skenario didapatkan bahwa penerapan skenario pertama dengan nilai PC dan PM yang rendah ($PC \ \& \ PM < 50$). Skenario tersebut menghasilkan rute distribusi dengan nilai *fitness* yang lebih rendah daripada nilai *fitness* pada tahap inisialiasi data awal. Nilai *fitness* pada tahap inisialiasi dari kromosom adalah 931 dan setelah melalui proses algoritma genetika menghasilkan nilai *fitness* 762.

Kata Kunci: Algoritma Genetika, optimasi waktu, biaya perjalanan, perangkat lunak

I. Pendahuluan

Distribusi adalah salah satu hal yang penting dalam suatu bidang usaha. Segala upaya diusahakan agar barang cepat sampai pada konsumen dan bisa diterima dengan baik. Dalam jurnal oleh Yanbin Liu (2010) dkk. [1] ditulis tentang adanya suatu software yang dibuat untuk mendukung keputusan top manajemen dalam hal transportasi laut. Dari enam modul yang ada, salah satunya adalah membahas tentang bagaimana mencari jarak terpendek pada jalur distribusinya. Mengacu pada jurnal tersebut maka akan dicoba penerapan algoritma genetika dalam jalur distribusi darat.

Algoritma genetika dapat dimanfaatkan pada masalah tersebut, karena memiliki kelebihan dalam menghasilkan *output* yang optimal. Dengan menggunakan konsep evolusi biologis maka akan dihasilkan suatu *output* berupa kombinasi jalur distribusi barang yang sebaiknya diambil untuk mengoptimalkan waktu dan biaya perjalanan. Untuk menerapkan algoritma genetika pada persoalan ini, dibuatlah perangkat lunak. Pada perangkat lunak yang dibuat, diinputkan beberapa kombinasi kota tujuan distribusi barang. Kemudian dilakukan pengujian perangkat lunak dengan parameter-parameter algoritma genetika yang bervariasi dan akan diketahui kromosom terbaik.

Tujuan yang ingin dicapai dalam penelitian ini adalah untuk menentukan jalur distribusi barang yang terbaik dengan menggunakan algoritma genetika. Dalam penentuan jalur tersebut difokuskan pada pendistribusian barang ke agen-agen yang melalui jalur darat. Jalur distribusi yang digunakan dalam penelitian ini hanya yang ada dipulau jawa dan kota awal merupakan kota tujuan akhir dari jalur distribusi. Dalam penelitian ini, terdapat 5 kota yang menjadi input dalam satu jalur distribusi.

II. Dasar Teori

2.1 Algoritma Genetika

Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan atas mekanisme evolusi biologis. Keberagaman pada evolusi biologis adalah variasi dari kromosom antar individu organisme. Algoritma ini didasari oleh konsep evolusi biologi, dan dapat memberikan solusi alternatif atas suatu masalah yang hendak diselesaikan. Algoritma genetika menawarkan suatu solusi pemecahan masalah yang terbaik, dengan memanfaatkan metode seleksi, *crossover*, dan mutasi [2].

Algoritma genetika adalah suatu bentuk teknik pencarian secara *stochastic*, berdasarkan mekanisme yang ada pada seleksi alam dan genetik secara natural. Algoritma genetika telah banyak diaplikasikan untuk penyelesaian masalah dan pemodelan dalam bidang teknologi seperti optimasi, pemrograman otomatis dan *machine learning*. Pada implementasi program algoritma genetika dapat digunakan untuk mencari jalan terpendek bebas hambatan. Berbeda dengan teknik pencarian konvensional, tahap awal pencarian dalam algoritma genetika dimulai dari himpunan penyelesaian acak (*random*) yang disebut populasi [3].

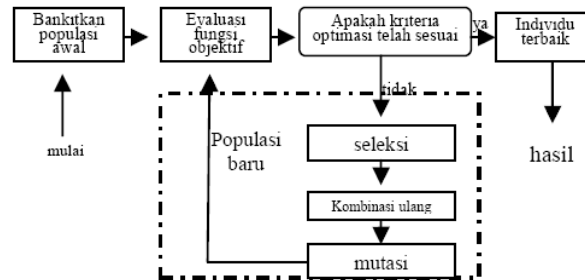
Populasi ini terdiri dari kromosom-kromosom. Setiap kromosom merupakan gambaran solusi atas pemecahan masalah. Populasi yang telah dipilih tersebut akan menghasilkan keturunan baru yang sifatnya diharapkan lebih baik dari populasi sebelumnya. Populasi yang baik sifatnya akan memiliki peluang untuk terus dikembangkan agar menghasilkan keturunan populasi yang lebih baik selanjutnya. Dengan demikian, solusi terbaik yang diinginkan dapat dicapai dengan terus mengulang proses pencarian keturunan.

Dalam proses tersebut, sebelum algoritma genetika dijalankan didefinisikan suatu fungsi *fitness* yang menyatakan tingkat keberhasilan sebuah populasi. Dengan melakukan perhitungan berdasarkan fungsi *fitness*, akan dapat ditentukan populasi yang akan dipertahankan untuk menghasilkan generasi selanjutnya. Proses ini biasa disebut sebagai proses seleksi. Proses ini merupakan salah satu tahap yang dirangkai dalam proses yang iteratif [4].

Proses lain yang termasuk pada rangkaian iterasi ini yaitu *crossover*. Pada proses ini, dilakukan persilangan atau perkawinan antar kromosom yang berada dalam satu generasi. Dengan demikian, kromosom yang terdapat pada populasi selanjutnya mewarisi sifat kedua induknya. Kromosom ini diharapkan bersifat lebih baik dibanding dengan generasi sebelumnya. Sedangkan proses mutasi merupakan proses diubahnya satu atau lebih nilai gen kromosom dalam satu populasi. Nilai gen tersebut akan digantikan dengan suatu nilai yang dipilih secara acak. Pada tahap ini, terdapat kemungkinan perubahan sifat di luar sifat induk pada keturunan yang dihasilkan.

2.2 Canonical Genetic Algorithm

Algoritma genetika adalah sebuah algoritma adaptif untuk menemukan solusi optimal menyeluruh untuk masalah optimasi. *Canonical Genetic Algorithm* (CGA) dikembangkan oleh *Holland* yang ditandai dengan representasi biner dari solusi individu, masalah sederhana-*crossover* bebas, operator mutasi dan aturan pemilihan proporsional [5]. Untuk lebih memahami CGA, dapat dilihat prosedur standar yang terdapat pada gambar 1 berikut.

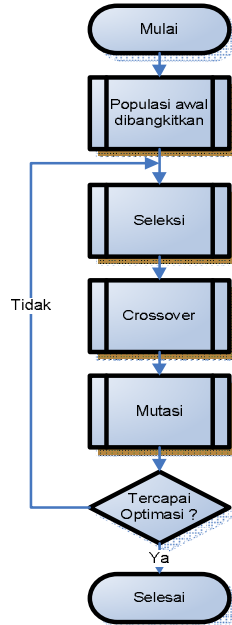


Gambar 1. Prosedur Algoritma Genetika

Anggota populasi adalah *string* atau kromosom, yang awalnya dipahami sebagai representasi biner dari vektor solusi. CGA memilih himpunan bagian yang menjadi solusi dari sebuah populasi yang dinamakan *parent* (orang tua), yang digabungkan untuk menghasilkan solusi baru yang dinamakan *children* (anak-anak) atau *offspring*. Aturan kombinasi untuk menghasilkan anak didasarkan pada gagasan genetik *crossover*, yang terdiri dari pertukaran tempat dari nilai solusi variabel tertentu. Pada suatu kondisi tertentu terjadi perubahan nilai acak yang disebut mutasi. Anak-anak yang dihasilkan oleh perkawinan orang tua dan yang lulus “tes bertahan hidup” kemudian akan tersedia untuk dipilih sebagai orang tua dari generasi berikutnya.

III. Implementasi Perangkat Lunak

Penggunaan piranti lunak dalam penelitian ini mengikuti tahapan-tahapan yang ada pada algoritma genetika secara umum. Yaitu seperti terlihat pada gambar 2 berikut.



Gambar 2. Flowchart Algoritma Genetika dalam Perangkat Lunak

3.1 Teknik Encoding

Proses *encoding* adalah salah satu proses yang sulit dalam algoritma genetika. Hal ini disebabkan karena proses *encoding* untuk setiap permasalahan berbeda-beda karena tidak semua teknik *encoding* cocok untuk setiap permasalahan. Proses *encoding* menghasilkan

string yang kemudian disebut kromosom. String terdiri dari sekumpulan bit. Bit ini dikenal sebagai gen. Jadi satu kromosom terdiri dari sejumlah gen.

Ada bermacam-macam teknik encoding yang dapat dilakukan dalam algoritma genetika. Beberapa teknik-teknik encoding itu antara lain adalah *binary encoding*, *permutation encoding*, *value encoding* serta *tree encoding*. Teknik encoding yang digunakan pada penelitian ini adalah *permutation encoding* dan *value encoding*.

Pada *permutation encoding*, kromosom-kromosom adalah kumpulan angka yang mewakili posisi dalam sebuah rangkaian. Pada *pencairian jalur distribusi ini*, kromosom mewakili urutan kota sebagai jalur distribusi. Jadi apabila satu kromosom berbentuk sebagai berikut $P1 = (X1, X2, X3, \dots, Xn)$ berarti jalur distribusi bergerak dari kota bernomor $X1$ ke $X2$ dst hingga ke kota ke Xn .

Sedangkan Pada *value encoding*, setiap kota tujuan distribusi diwakili oleh sebuah angka, dan angka tersebut merupakan sebuah gen yang ada pada sebuah kromosom. Setiap kromosom yang ada pada satu generasi merepresentasikan kombinasi jalur distribusi barang yang digunakan untuk mendapatkan hasil yang optimal.

Berikut ini adalah contoh kromosom pada persoalan ini : 1 2 3 4 5, 3 4 2 1 5, 5 4 3 2 1, 3 2 4 5 1. Itu berarti bahwa angka 1,2,3,4,5 adalah angka yang mewakili nama kota pada jalur distribusi dan urutan penulisan angka menunjukkan urutan kota pada jalur distribusi barangnya.

3.2 Proses Seleksi

Proses seleksi adalah proses yang memegang peranan penting dalam algoritma genetika. Proses seleksi ini digunakan agar hanya kromosom-kromosom yang berkualitas yang dapat melanjutkan perannya dalam proses algoritma genetika berikutnya. Ada bermacam-macam teknik untuk melakukan proses seleksi pada suatu permasalahan. Teknik seleksi yang akan digunakan tergantung pada permasalahan yang akan diselesaikan. Ada beberapa metode seleksi dari induk, diantaranya adalah *Rank-based Fitness Assignment*, *Roulette Wheel Selection*, *Stochastic Universal Sampling*, *Local Selection*, *Truncation Selection*, *Tournament Selection* [2].

Proses penyeleksian yang digunakan disini adalah *Roulette Wheel Selection*. Pada proses penyeleksian digunakan suatu parameter yang disebut kesesuaian atau *fitness*. *Fitness* digunakan untuk menentukan seberapa baik kromosom akan bertahan hidup. Penentuan berapa besar nilai *fitness* suatu kromosom berdasarkan fungsi *fitness* yang didefinisikan tersendiri. Untuk makalah ini maka fungsi *fitness* didefinisikan sebagai:

$$\text{Fitness} = \text{TotalJarak} \dots\dots\dots(1)$$

Sebelumnya telah didapatkan jarak antar kota di Jawa sebagai berikut pada Gambar 3.

DAFTAR JARAK ANTAR KOTA DI PULAU JAWA

	MALINI	MAGBANG	MALANG	PEKALONGAN	PURWOREJO	PROBOLINGGO	REMBANG	SEMARANG	SEWANG	SURABUHI	SUPABAYA	SOLO	TASIKMALAYA	TEGAL	CILICAP	WONOSOBO
Bandung	581	403	764	266	362	774	476	367	258	96	675	467	106	202	259	339
Banyuwangi	412	633	281	698	637	190	490	597	1171	1060	289	506	893	762	768	678
Jakarta	699	522	882	384	504	892	594	485	89	115	793	585	275	320	428	481
Bondowoso	315	517	184	601	540	93	393	500	1074	965	192	409	796	665	671	560
Bojor	707	530	890	392	488	900	602	493	143	61	801	593	232	328	385	465
Cirebon	451	274	634	136	256	644	346	237	337	226	545	337	121	72	203	233
Jember	320	522	185	607	545	99	399	506	1080	968	198	414	801	671	695	565
Jogjakarta	179	43	363	181	66	361	212	118	654	524	327	65	322	245	197	107
Kediri	81	283	103	376	306	177	206	295	869	758	124	175	562	460	437	326
Malind		221	184	315	245	222	145	214	788	677	169	114	501	379	375	265
Magelang	221		406	138	44	443	184	75	611	500	370	108	297	202	174	64
Malang	184	406		499	429	91	275	397	671	860	89	298	682	562	560	449
Pekalongan	315	138	499		170	508	210	101	473	362	409	201	257	64	195	134
Purworejo	245	44	429	170		446	228	119	593	458	393	131	256	223	131	54
Probolinggo	222	443	31	508	445		300	407	981	870	99	316	702	572	598	487
Rembang	145	184	275	210	228	300		109	683	572	201	147	467	274	355	228
Semarang	214	75	397	101	119	407	109		574	483	308	100	358	165	246	119
Serang	788	611	971	473	593	981	683	574		203	882	674	364	409	517	570
Sukabumi	677	500	860	362	458	870	572	463	203		771	563	202	298	355	435
Surabaya	169	370	89	409	393	99	201	308	882	771		262	649	473	523	434
Solo	114	108	298	201	131	316	147	100	674	563	262		387	265	262	151
Tasikmalaya	501	297	682	257	256	702	467	358	384	202	649	387		193	153	233
Tegal	379	202	562	64	223	572	274	165	409	298	473	265	193		131	197
Cilacap	375	174	560	195	131	598	355	246	517	355	523	262	153	131		127
Wonosobo	265	64	449	134	54	487	228	119	570	435	434	151	233	197	127	

Gambar 3. Jarak antar Kota di Jawa

Dimisalkan jalur distribusi yang akan ditempuh adalah Kediri – Malang – Semarang – Rembang – Solo. Kemudian dilakukan teknik encoding dengan menginisialisasi kota-kota tersebut dengan angka yaitu Kediri(1) – Malang(2) – Semarang(3) – Rembang(4) – Solo(5). Didapatkan gennya adalah Malang(2) – Semarang(3) – Rembang(4) – Solo(5), Kediri(1) tidak dimasukkan kedalam gen karena asumsinya adalah bahwa kota asal adalah juga merupakan kota akhir tujuan distribusi barang.

Kemudian misalkan ada enam populasi dalam satu generasi yaitu :

Kromosom[1] : [Malang – Rembang – Solo – Semarang] atau [B D E C]
 Kromosom[2] : [Rembang – Malang – Solo - Semarang] atau [D B E C]
 Kromosom[3] : [Semarang – Malang – Rembang - Solo] atau [C B D E]
 Kromosom[4] : [Solo – Malang – Semarang - Rembang] atau [E B C D]
 Kromosom[5] : [Solo – Semarang – Malang - Rembang] atau [E C B D]
 Kromosom[6] : [Semarang – Rembang – Solo - Malang] atau [C D E B]

Kemudian dihitung nilai fitnessnya :

Fitness[1] = AB+BD+DE+EC+CA= 103+275+147+100+295 = 920
 Fitness[2] = AD+DB+BE+EC+CA= 206+275+298+100+295 = 1174
 Fitness[3] = AC+CB+BD+DE+EA= 295+397+275+147+175 = 1289
 Fitness[4] = AE+EB+BC+CD+DA= 175+298+397+109+206 = 1185
 Fitness[5] = AE+EC+CB+BD+DA= 175+100+397+275+206 = 1153
 Fitness[6] = AC+CD+DE+EB+BA= 295+109+147+298+103 = 952

Dari hasil diatas dicari kromosom dengan *fitness yang lebih kecil dan kromosom tersebut akan* mempunyai probabilitas lebih besar untuk terpilih kembali maka digunakan inverse. Dengan rumus :

$$Q[i] = \frac{1}{\text{Fitness [i]}} \dots\dots\dots(2)$$

Q[1] = 1/920 = 0,001087
 Q[2] = 1/1174 = 0,000852
 Q[3] = 1/1289 = 0,000776
 Q[4] = 1/1185 = 0,000844
 Q[5] = 1/1153 = 0,000867
 Q[6] = 1/952 = 0,00105
 Total Q[] = 0,001087 + 0,000852 + 0,000776 + 0,000844 + 0,000867 + 0,00105
 Total Q[] = 0,005476

Kemudian setelah menemukan nilai Q-nya, dilanjutkan dengan mencari probabilitasnya, untuk tahu kromosom mana yang terpilih sebagai generasi baru. Rumus probabilitasnya :

$$P[i] = \frac{Q[i]}{\text{Total Q}} \dots\dots\dots(3)$$

P[1] = 0,001087 / 0,005476 = 0,199
 P[2] = 0,000852 / 0,005476 = 0,156
 P[3] = 0,000776 / 0,005476 = 0,142
 P[4] = 0,000844 / 0,005476 = 0,154
 P[5] = 0,000867 / 0,005476 = 0,158
 P[6] = 0,00105 / 0,005476 = 0,192

Terlihat bahwa kromosom 1 mempunyai fitness terkecil sehingga probabilitas untuk terpilih pada generasi selanjutnya lebih besar dari pada kromosom yang lain. Proses seleksi dengan menggunakan *Roulette Wheel Selection* mempunyai tahap-tahap sebagai berikut:

1. [Sum] : Jumlahkan semua nilai probabilitasnya → C
2. [Select] : Men-generate bilangan random dengan interval (0-C) → R
3. [Loop] : Membandingkan nilai R dan C, Jika R[k]<C[k] maka kromosom ke-k sebagai induk, selain itu pilih kromosom ke-k sebagai induk dengan syarat C[k-1] < R[k] < C[k]

Looping atau proses memutar *Roulette Wheel* nya dilakukan sebanyak jumlah kromosom (enam) untuk mendapatkan nilai randomnya.

Mencari nilai kumulatif dari probabilitasnya (C).

$$\begin{aligned} C[1] &= 0,199 \\ C[2] &= 0,199 + 0,156 = 0,355 \\ C[3] &= 0,355 + 0,142 = 0,497 \\ C[4] &= 0,497 + 0,154 = 0,651 \\ C[5] &= 0,651 + 0,158 = 0,809 \\ C[6] &= 0,809 + 0,192 = 1,001 \end{aligned}$$

Kemudian nilai random (R) nya :

$$\begin{aligned} R[1] &= 0,314 \\ R[2] &= 0,111 \\ R[3] &= 0,342 \\ R[4] &= 0,743 \\ R[5] &= 0,521 \\ R[6] &= 0,411 \end{aligned}$$

Setelah perbandingan dilakukan, di dapatkan populasi yang terbaru :

Kromosom[1] : [2] : [Rembang – Malang – Solo - Semarang] atau [D B E C]
 Kromosom[2] : [1] : [Malang – Rembang – Solo – Semarang] atau [B D E C]
 Kromosom[3] : [3] : [Semarang – Malang – Rembang - Solo] atau [C B D E]
 Kromosom[4] : [5] : [Solo – Semarang – Malang - Rembang] atau [E C B D]
 Kromosom[5] : [4] : [Solo – Malang – Semarang - Rembang] atau [E B C D]
 Kromosom[6] : [6] : [Semarang – Rembang – Solo - Malang] atau [C D E B]

3.3 Proses Rekombinasi atau Crossover

Proses rekombinasi atau yang lebih dikenal dengan nama proses *crossover* adalah menyilangkan dua kromosom sehingga membentuk kromosom baru yang harapannya lebih baik dari pada induknya. Tidak semua kromosom pada suatu populasi akan mengalami proses rekombinasi. Kemungkinan suatu kromosom mengalami proses rekombinasi didasarkan pada probabilitas *crossover* (PC) yang telah ditentukan terlebih dahulu. Probabilitas *crossover* menyatakan peluang suatu kromosom akan mengalami *crossover*. Ada beberapa teknik rekombinasi yang dapat digunakan untuk menyelesaikan masalah seperti ini, antara lain adalah *partially mapped crossover* (PMX), *order crossover* dan *cycle crossover* [6].

Teknik rekombinasi yang digunakan adalah teknik *order crossover*. *Order crossover* (OX) diperkenalkan oleh Davis. Teknik ini diawali dengan membangkitkan dua bilangan acak. Kemudian gen yang berada diantara kedua bilangan acak akan disalin ke *offspring* dengan posisi yang sama. Langkah berikutnya untuk mendapatkan *offspring* pertama adalah mengurutkan gen yang berada pada *parent* kedua dengan urutan gen yang berada pada posisi setelah bilangan acak kedua diikuti dengan gen yang berada pada posisi sebelum bilangan acak pertama dan diakhiri dengan gen yang berada pada posisi diantara kedua bilangan acak. Kemudian gen yang telah diurutkan tersebut dibandingkan dengan *offspring* pertama. Apabila gen tersebut ada pada *offspring* kedua maka abaikan gen tersebut dari urutan itu. Kemudian masukkan urutan yang baru saja didapat pada *offspring* dengan cara memasukkan urutan gen pada posisi setelah bilangan acak kedua terlebih dahulu dan sisanya dimasukkan pada posisi sebelum bilangan acak pertama. Begitu juga untuk menghasilkan *offspring* kedua.

Misal kita tentukan PC = 50%, maka diharapkan dalam 1 generasi ada 50% * 6 kromosom = 3 kromosom dari populasi mengalami *crossover*.

Sebelumnya, bangkitkan bilangan acak R sebanyak jumlah populasi yaitu 6 kali.

$$\begin{aligned} R[1] &= 0,541 \\ R[2] &= 0,211 \\ R[3] &= 0,302 \end{aligned}$$

R[4] = 0,877
 R[5] = 0,771
 R[6] = 0,131

Kemudian dicari dengan kategori Kromosom ke-k yang dipilih sebagai induk jika $R[k] < pc$. Maka yang akan dijadikan induk adalah kromosom[2], kromosom[3], dan kromosom[6]. Proses selanjutnya adalah menentukan posisi crossover. Dilakukan dengan membangkitkan bilangan acak antara 1 sampai dengan panjang kromosom-1. $(4-1 = 3)$

Dimisalkan bilangan acak untuk 3 kromosom induk yang akan di-crossover :

C[2] = 2
 C[3] = 1
 C[6] = 2

Misal diperoleh bilangan acaknya 2, maka gen yang ke-2 pada kromosom induk pertama diambil kemudian ditukar dengan gen pada kromosom induk kedua yang belum ada pada induk pertama dengan tetap memperhatikan urutannya.

Proses *crossover* :

Kromosom[2] = Kromosom[2] >< Kromosom[3] = [B D E C] >< [C B D E] = [B D C E]
 Kromosom[3] = Kromosom[3] >< Kromosom[6] = [C B D E] >< [C D E B] = [C D E B]
 Kromosom[6] = Kromosom[6] >< Kromosom[2] = [C D E B] >< [B D E C] = [C D B E]

Sehingga didapatkan populasi setelah di-crossover :

Kromosom[1] = [D B E C]
 Kromosom[2] = [B D C E]
 Kromosom[3] = [C D E B]
 Kromosom[4] = [E C B D]
 Kromosom[5] = [E B C D]
 Kromosom[6] = [C D B E]

3.4 Proses Mutasi

Proses mutasi ini dilakukan setelah proses rekombinasi dengan cara memilih kromosom yang akan dimutasi secara acak, dan kemudian menentukan titik mutasi pada kromosom tersebut secara acak pula. Banyaknya kromosom yang akan mengalami mutasi dihitung berdasarkan probabilitas mutasi yang telah ditentukan terlebih dahulu.

Apabila probabilitas mutasi adalah 100% maka semua kromosom yang ada pada populasi tersebut akan mengalami mutasi. Sebaliknya, jika probabilitas mutasi yang digunakan adalah 0% maka tidak ada kromosom yang mengalami mutasi pada populasi tersebut.

Ada bermacam-macam teknik mutasi yang dapat digunakan untuk menyelesaikan suatu masalah dengan algoritma genetika. Seperti pada teknik rekombinasi, teknik mutasi juga dirancang untuk digunakan pada suatu masalah yang spesifik sehingga tidak setiap teknik mutasi dapat diterapkan pada suatu masalah yang akan diselesaikan. Selain itu, teknik mutasi yang digunakan juga harus sesuai dengan teknik *encoding* yang digunakan untuk menyelesaikan permasalahan tersebut. Beberapa teknik mutasi yang dapat digunakan dalam penyelesaian masalah ini adalah *inversion mutation*, *insertion mutation*, *swapping mutation*, dan *reciprocal mutation*.

Pada kasus ini skema mutasi yang digunakan adalah *swapping mutation*. Jumlah kromosom yang mengalami mutasi dalam satu populasi ditentukan oleh parameter *mutation rate* (PM). Proses mutasi dilakukan dengan cara menukar gen yang dipilih secara acak dengan gen sesudahnya. Jika gen tersebut berada di akhir kromosom, maka ditukar dengan gen yang pertama.

Pertama dihitung dahulu panjang total gen yang ada pada satu populasi:

$$\text{Panjang Total Gen} = \text{jumlah gen dalam 1 kromosom} * \text{jumlah Kromosom} \dots\dots\dots(4)$$

$$\text{Panjang total gen} = 4 * 6$$

$$\text{Panjang total gen} = 24$$

Untuk memilih posisi gen yang mengalami mutasi dilakukan dengan membangkitkan bilangan acak antara 1 – Panjang total gen yaitu 1- 24. Misal, Ditetapkan PM = 20 %. Maka jumlah gen yang akan dimutasi adalah $= 0,2 * 24 = 4,8 = 5$. Kemudian 5 buah posisi gen yang akan dimutasi, setelah diacak adalah posisi 3, 7, 10, 20, 24.

Kromosom hasil crossover:

Kromosom[1] = [D B E C]
Kromosom[2] = [B D C E]
Kromosom[3] = [C D E B]
Kromosom[4] = [E C B D]
Kromosom[5] = [E B C D]
Kromosom[6] = [C D B E]

Proses mutasi :

Kromosom[1] = [D B C E]
Kromosom[2] = [B D E C]
Kromosom[3] = [C E D B]
Kromosom[4] = [E C B D]
Kromosom[5] = [D B C E]
Kromosom[6] = [E D B C]

Setelah proses ini selesai maka satu generasi telah selesai pula di-*generate* dengan algoritma genetika. Sebelumnya ditentukan dulu kapan proses algoritma genetika ini akan berhenti. Ada beberapa kondisi untuk mengeceknya yaitu jika diperoleh nilai *fitness* yang terendah yang tidak berubah atau kita tentukan akan di *generate* hingga generasi ke – N.

Kemudian dicari lagi nilai Fitnessnya. Nilai *fitness* setelah 1 generasi adalah:

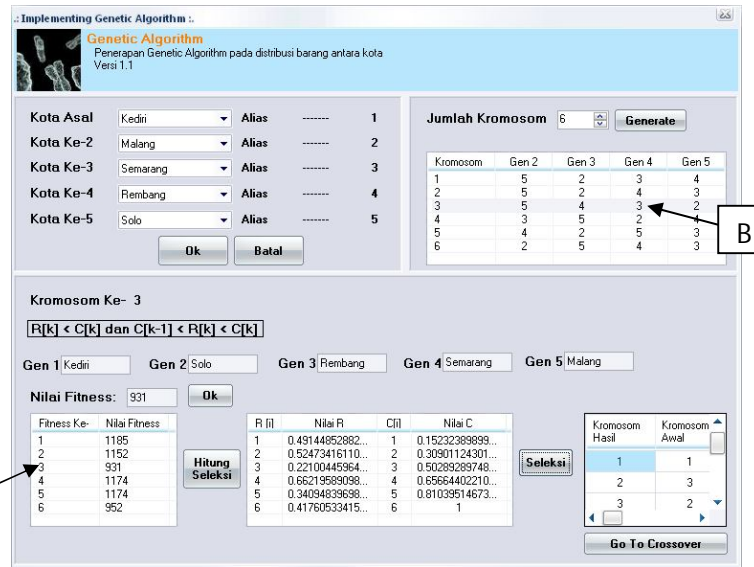
Fitness[1] = AD+DB+BC+CE+EA = 206 + 275 + 397 + 100 + 175 = 1153
Fitness[2] = AB+BD+DE+EC+CA = 103 + 275 + 147 + 100 + 295 = 920
Fitness[3] = AC+CE+ED+DB+BA = 295 + 100 + 147 + 275 + 103 = 920
Fitness[4] = AE+EC+CB+BD+DA = 175 + 100 + 397 + 275 + 206 = 1153
Fitness[5] = AD+DB+BC+CE+EA = 206 + 275 + 397 + 100 + 175 = 1153
Fitness[6] = AE+ED+DB+BC+CA = 175 + 147 + 275 + 397 + 295 = 1289

IV. Skenario Uji Coba Aplikasi

Skenario uji coba aplikasi dilakukan sebanyak tiga kali, dengan pola distribusi awal yang sama (Kediri-Malang-Semarang-Rembang-Solo). Skenario yang pertama yaitu dengan menggunakan PC (*Crossover Probability*) dan PM (*Mutation Rate*) yang rendah, skenario yang kedua dengan menggunakan PC (*Crossover Probability*) dan PM (*Mutation Rate*) yang sedang dan skenario yang ketiga dengan menggunakan PC (*Crossover Probability*) dan PM (*Mutation Rate*) yang tinggi.

4.1 Skenario Pertama

Pada Gambar 4 berikut ini adalah inisialisasi data awal yang menghasilkan rute distribusi dengan nilai *fitness* terendah 931 (A) dengan rute distribusi: Kediri-Solo-Rembang-Semarang-Malang (B).



Gambar4. Inisialisasi Data Awal Skenario Pertama

Proses seleksi tersebut menghasilkan perubahan urutan kromosom yang dapat dilihat pada Gambar 5 dan Gambar 6 berikut.

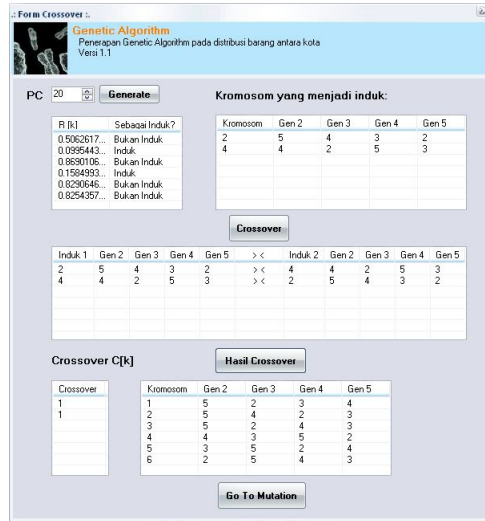
Kromosom Hasil	Kromosom Awal
1	1
2	3
3	2

Gambar 5. Perubahan Urutan Kromosom 1-3

Kromosom Hasil	Kromosom Awal
4	5
5	4
6	6

Gambar 6. Perubahan Urutan Kromosom 4-6

Inputan untuk PC pada skenario pertama ini adalah 20, yang menghasilkan 2 induk *crossover* yang dapat dilihat pada Gambar 7 berikut.



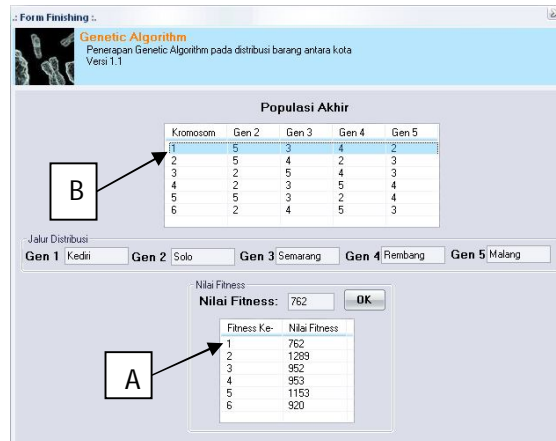
Gambar 7. Crossover Skenario Pertama

Inputan untuk PM pada skenario pertama ini adalah 25, yang menghasilkan posisi mutasi sebanyak 5 yang dapat dilihat pada Gambar 8 berikut.



Gambar 8. Mutasi Skenario Pertama

Hasil akhir setelah dilakukan proses seleksi, crossover dan mutasi dapat dilihat pada Gambar 9 berikut.

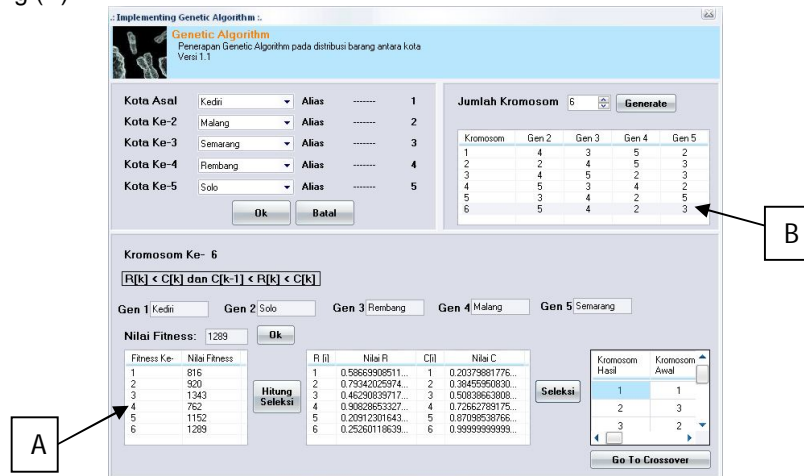


Gambar 9. Hasil Akhir Skenario Pertama

Dari hasil akhir ini dapat dilihat bahwa kromosom yang pertama dengan nilai *fitness* terendah yaitu 762 (A), dengan rute distribusi: Kediri-Solo-Semarang-Rembang-Malang (B). Dari proses pada skenario pertama ini dapat disimpulkan bahwa dengan nilai PC dan PM yang rendah (<50) menghasilkan rute distribusi dengan nilai *fitness* yang lebih rendah daripada nilai *fitness* pada tahap inialisasi data awal.

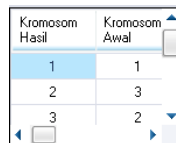
4.2 Skenario Kedua

Pada Gambar 10 berikut ini adalah inialisasi data awal yang menghasilkan rute distribusi dengan nilai *fitness* terendah 762 (A) dengan rute distribusi: Kediri-Solo-Semarang-Rembang-Malang (B).

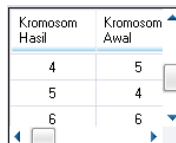


Gambar 10. Inisialisasi Data Awal Skenario Kedua

Proses seleksi tersebut menghasilkan perubahan urutan kromosom yang dapat dilihat pada Gambar 11 dan Gambar 12 berikut.



Gambar 11. Perubahan Urutan Kromosom 1-3



Gambar 12. Perubahan Urutan Kromosom 4-6

Inputan untuk PC pada skenario kedua ini adalah 50, yang menghasilkan 2 induk *crossover* yang dapat dilihat pada Gambar 13 berikut.



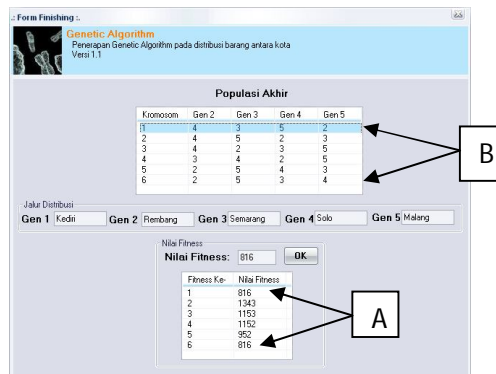
Gambar 13. Crossover Skenario Kedua

Inputan untuk PM pada skenario kedua ini adalah 50, yang menghasilkan posisi mutasi sebanyak 12 yang dapat dilihat pada Gambar 14 berikut.



Gambar 14. Mutasi Skenario Kedua

Hasil akhir setelah dilakukan proses seleksi, crossover dan mutasi dapat dilihat pada Gambar 15 berikut.



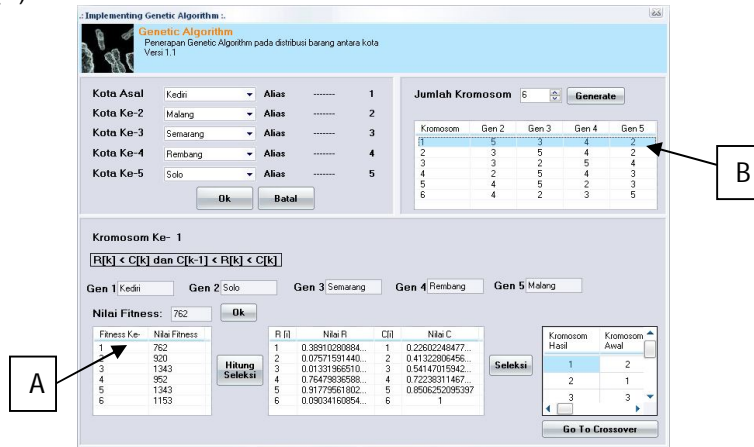
Gambar 15. Hasil Akhir Skenario Kedua

Dari hasil akhir ini dapat dilihat bahwa kromosom yang pertama dan keenam dengan nilai *fitness* terendah yaitu 816 (A), dengan rute distribusi: (Kromosom1) Kediri-Rembang-Semarang-Solo-Malang dan (Kromosom6) Kediri-Malang-Solo-Semarang-Rembang (B). Dari proses pada skenario pertama ini dapat disimpulkan bahwa dengan nilai PC dan PM yang sedang (=50) menghasilkan rute distribusi dengan nilai *fitness* yang lebih tinggi daripada nilai

fitness pada tahap inialisasi data awal dan ada kemungkinan dua kromosom dengan rute yang berbeda memiliki nilai *fitness* yang sama.

4.3 Skenario Ketiga

Pada Gambar 16 berikut ini adalah inialisasi data awal yang menghasilkan rute distribusi dengan nilai *fitness* terendah 762 (A) dengan rute distribusi: Kediri-Solo-Semarang-Rembang-Malang (B).



Gambar 16. Inialisasi Data Awal Skenario Ketiga

Proses seleksi tersebut menghasilkan perubahan urutan kromosom yang dapat dilihat pada Gambar 17 dan Gambar 18 berikut.

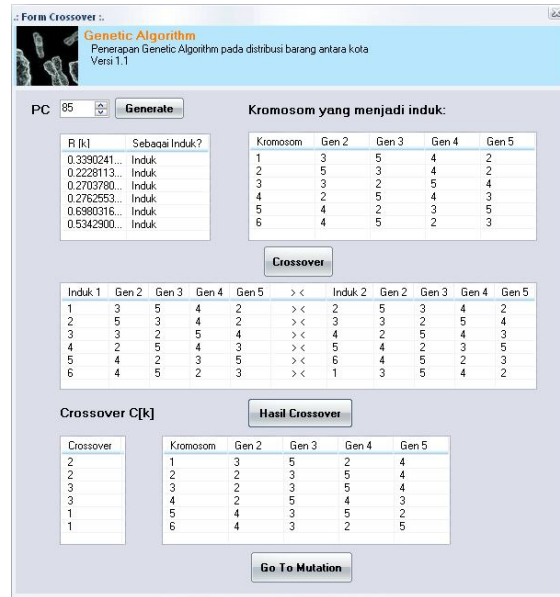
Kromosom Hasil	Kromosom Awal
1	2
2	1
3	3

Gambar 17. Perubahan Urutan Kromosom 1-3

Kromosom Hasil	Kromosom Awal
4	4
5	6
6	5

Gambar 18. Perubahan Urutan Kromosom 4-6

Inputan untuk PC pada skenario ketiga ini adalah 85, yang menghasilkan 6 induk (semua kromosom merupakan induk *crossover*) *crossover* yang dapat dilihat pada Gambar 19 berikut.



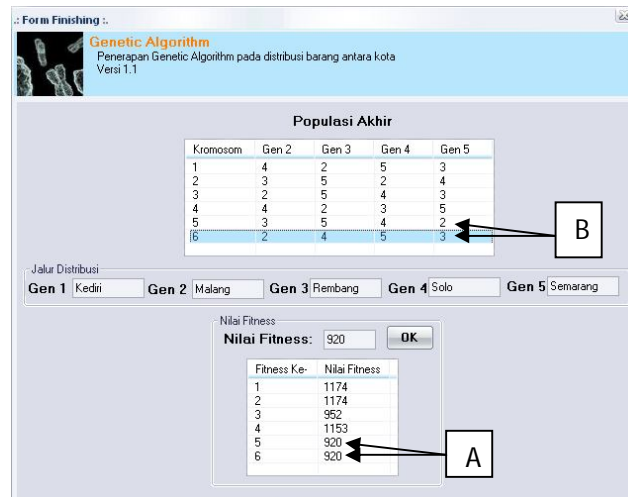
Gambar 19. Crossover Skenario Ketiga

Inputan untuk PM pada skenario ketiga ini adalah 85, yang menghasilkan posisi mutasi sebanyak 20 yang dapat dilihat pada Gambar 20 berikut.



Gambar 20. Mutasi Skenario Ketiga

Hasil akhir setelah dilakukan proses seleksi, crossover dan mutasi dapat dilihat pada Gambar 21 berikut.



Gambar 21. Hasil Akhir Skenario Ketiga

Dari hasil akhir ini dapat dilihat bahwa kromosom yang kelima dan keenam dengan nilai *fitness* terendah yaitu 920 (A), dengan rute distribusi: (Kromosom 5) Kediri-Semarang-Solo-Rembang-Malang dan (Kromosom 6) Kediri-Malang-Rembang-Solo-Semarang (B). Dari proses pada skenario pertama ini dapat disimpulkan bahwa dengan nilai PC dan PM yang tinggi (>50) menghasilkan rute distribusi dengan nilai *fitness* yang lebih tinggi daripada nilai *fitness* pada tahap inisialisasi data awal dan ada kemungkinan dua kromosom dengan rute yang berbeda memiliki nilai *fitness* yang sama.

V. Kesimpulan

Beberapa kesimpulan yang dapat diambil setelah mempelajari mengenai algoritma genetika dan mencoba mengaplikasikannya ke perangkat lunak adalah sebagai berikut: Algoritma genetika merupakan suatu algoritma yang menggunakan parameter dan bilangan random (acak), sehingga pada setiap perhitungan menghasilkan hasil perhitungan yang berbeda. Algoritma genetika bagus untuk menyelesaikan persoalan optimasi suatu rute namun harus ditentukan kriteria optimasi yang diinginkan dalam bentuk nilai PC dan PM yang sesuai. Nilai PC dan PM yang rendah bagus untuk menghasilkan nilai *fitness* yang terendah (rute terbaik), namun sulit untuk menentukan nilai terendah tersebut dikarenakan proses *crossover* dan mutasi bergantung pada bilangan *random* (acak) yang dibandingkan dengan inputan PC dan PM tersebut. Dapat juga disimpulkan bahwa dengan menerapkan algoritma genetika untuk kasus penentuan jalur terpendek adalah efektif. Terbukti pada skenario pertama dengan penentuan PC dan PM < 50 didapatkan nilai *fitness* yang lebih rendah yang berarti bahwa rute tersebut adalah jalur terpendek. Hasil pemrosesan tergantung pada hasil generasi kromosom pada saat awal dijalankan. Jika pada saat generasi awal kromosom tersebut sudah didapatkan nilai *fitness* yang lebih rendah daripada rute yang diinputkan, maka sebaiknya ditentukan bahwa hal itu adalah kriteria untuk pemrosesan algoritma genetika ini berhenti. Jika pemrosesan tetap dijalankan dengan seleksi, *crossover* maupun mutasi maka kemungkinan besar pemrosesan algoritma genetika untuk menemukan jalur terpendek akan gagal. Hal tersebut terbukti pada skenario kedua dan ketiga, yang menghasilkan kromosom dengan nilai *fitness* yang lebih tinggi dibandingkan dengan kromosom pada saat inisialisasi data awal.

Daftar Pustaka

- [1] Liu, Y., Zhou, C., Guo, D., Wang., K., Pang, W., Zhai, Y., 2010, A Decision Support System Using Soft Computing for Modern International Container Transportation Services, *Applied Soft Computing*, Volume 10, Issue 4, September 2010, pp. 1087–1095
- [2] Kusumadewi, Sri, 2003, *Artificial Intelligence (Teknik dan Aplikasinya)*, Yogyakarta: Graha Ilmu.
- [3] Lumbantobing, H., Hidayatno, A., Darjat, 2011, Penerapan Algoritma Genetika pada Perencanaan Lintas Kendaraan, Undergraduate Thesis, Universitas Diponegoro
- [4] Fitrah, A., Zaky, A., Fitrasani, 2006, Penerapan Algoritma Genetika pada Persoalan Pedagang Keliling (TSP), Sekolah Tinggi Elektro Dan Informatika ITB.
- [5] Cao, Y. J., Wu, Q. H., 1999, Teaching Genetic Algorithm Using Matlab, *International Journal Electrical Engineering Education*, Vol. 36, pp. 139–153.
- [6] Lukas, S., Anwar, T., Yuliani, W., 2005, Penerapan Algoritma Genetika untuk *Traveling Salesmen Problem* dengan Menggunakan Metode *Order Crossover* dan *Insertion Mutation*, Seminar Nasional Aplikasi Teknologi Informasi.