

USULAN RUTE DISTRIBUSI *FROZEN FOOD* MENGGUNAKAN ALGORITMA GENETIKA PADA PERUSAHAAN DISTRIBUTOR DI BANDUNG

MUHAMAD HASANUDIN¹, RISPIANDA²

^{1,2} Program Studi Teknik Industri Institut Teknologi Nasional
Email: hasanudinmuhamad42@mhs.itenas.ac.id

Received 13 09 2021 | *Revised* 23 09 2021 | *Accepted* 23 09 2021

ABSTRAK

Perusahaan bergerak dalam bidang general trading dan mempunyai kendala dalam cara pendistribusian frozen food secara efektif dan efisien dengan pengeluaran cost serendah mungkin dan dapat dipercaya konsumen dalam ketepatan waktu pengiriman akibat meningkatnya permintaan dari konsumen. Selain itu perusahaan melakukan penentuan rute hanya menurut perkiraan saja. Permasalahan ini termasuk kedalam Travelling Salesman Problem (TSP) yaitu penentuan rute tercepat dalam pendistribusian. Permasalahan TSP yang dihadapi perusahaan dapat diselesaikan dengan menggunakan metode Algoritma Genetika yang meniru proses evolusi manusia, dimana terdapat beberapa langkah yang perlu dilakukan yaitu encoding, evaluasi nilai fitness, seleksi, crossover, mutasi, regenerasi, elitisme dan terminasi. Data yang diperlukan dalam penelitian adalah data jarak antar lokasi. Percobaan dilakukan menggunakan software MATLAB sebanyak 10 kali dengan 50 kali pegulangan pada setiap percobaannya. Hasil percobaan terbaik didapatkan pada percobaan ke-7 yang menghasilkan jarak rute 171,47 km dan mampu melakukan penghematan cost bagi perusahaan serta membantu pendistribusian frozen food agar lebih efektif dan efisien.

Kata Kunci: *distribusi, travelling salesman problem (tsp), algoritma genetika.*

ABSTRACT

The company is engaged in general trading and has problems with how to distribute frozen food effectively and efficiently with the lowest possible cost and can be trusted by consumers in on time delivery due to increasing demand from consumers. In addition to that, the company conducted distributions only according to estimates. This problem is included in the Traveling Salesman Problem (TSP), namely determining the fastest route in distribution. The TSP issues that the company faced can be solved by using a genetic algorithm that mimics the process of human evolution, where there are some steps that need to be done: encoding, fitness value evaluations, selections, crossover, mutations, regeneration, elitism and termination. The data needed in the study is the distance data between locations. The experiment was carried out using MATLAB software 10 times with 50 repetitions in each experiment. The best probationary results were obtained at an experiment 7 with a distance of 171.47 km and were able to provide cost savings to companies and help distribution of frozen food to be more effective and

efficient.

Keywords: *distribution, travelling salesman problem (tsp), genetic algorithm.*

1. PENDAHULUAN

Perusahaan bergerak pada bidang manufaktur pembuatan komponen otomotif. Akan tetapi dengan adanya kondisi pandemi seperti sekarang ini pesanan konsumen menurun sehingga perusahaan mencoba bergerak dalam bidang baru yaitu general trading dalam penjualan produk-produk makanan ringan dan frozen food. Perubahan pergerakan ini didasarkan pada value yang akan didapatkan perusahaan serta mempermudah masyarakat agar tidak terlalu banyak berkegiatan diluar, seperti bolak-balik berbelanja makanan dengan solusi membeli frozen food yang praktis dan dapat diawetkan dengan menyimpannya di dalam freezer.

Pengiriman frozen food dilakukan setiap satu minggu sekali dengan menggunakan satu unit armada sewa jenis pick up freezer box dengan kapasitas angkut maksimal 200 karton. Kendala yang dihadapi oleh perusahaan adalah meningkatnya permintaan frozen food dari konsumen sehingga diperlukan cara pendistribusian yang efektif dan efisien dengan pengeluaran cost serendah mungkin dan dapat dipercaya konsumen dalam ketepatan waktu pengiriman. Selain itu permasalahan pendistribusian frozen food yang dilakukan saat ini hanya mengandalkan perkiraan saja untuk penentuan tujuan lokasi yang akan dicapainya.

Sesuai dengan permasalahan perusahaan dimana perusahaan hanya menggunakan 1 buah armada dan permintaan konsumen masih dalam batas angkut armada serta dilakukan dalam satu hari kerja, maka dapat dikategorikan kedalam permasalahan Travelling Salesman Problem (TSP) yaitu suatu pencarian rute perjalanan n kota dengan biaya termurah dengan mengunjungi semua kota dan kembali lagi ke tempat asalnya tepat satu kali. Biaya yang dimaksud dapat diasumsikan sebagai jarak, waktu, bahan bakar, harga sewa kendaraan, dan sebagainya (Greco, 2008). Salah satu teknik yang dapat digunakan dalam pemecahan masalah TSP yaitu menggunakan algoritma genetika dan algoritma ant colony dimana pemecahan masalah dilakukan secara logis dengan langkah-langkah yang disusun secara sistematis. Algoritma yang akan digunakan pada penelitian ini adalah algoritma genetika karena memiliki performansi lebih baik dalam pencarian rute terpendek dengan jumlah titik lokasi yang banyak dan memiliki banyak percabangan dibandingkan algoritma ant colony (Fallo, 2015). Algoritma genetika merupakan teknik pencarian (searching) dan optimasi yang dapat digunakan dalam pencarian rute terpendek dimana prinsip kerja dari algoritma genetika adalah meniru proses evolusi manusia pada struktur genetik dan seleksi alam. Penyelesaian dengan algoritma genetika terbagi kedalam beberapa tahap yaitu inialisasi (encoding), evaluasi fitness, seleksi, pindah silang (crossover), mutasi, regenerasi, elitisme dan terminasi.

2. METODOLOGI PENELITIAN

Permasalahan yang terjadi pada perusahaan yaitu cara pendistribusian frozen food yang efektif dan efisien dengan pengeluaran cost serendah mungkin dan dapat dipercaya konsumen dalam ketepatan waktu pengiriman akibat meningkatnya permintaan, serta akibat penentuan rute yang dilakukan perusahaan saat ini hanya berdasarkan perkiraan saja. Setelah permasalahan diketahui, selanjutnya mengumpulkan literatur yang dapat mendukung memecahkan masalah yang dihadapi perusahaan yaitu teori Travelling Salesman Problem (TSP) dan teori algoritma genetika.

Setelah teori pendukung didapatkan, langkah selanjutnya melakukan pengumpulan data seperti data lokasi depot, data lokasi pasar dan toko, data jarak antar lokasi, data rute dan jarak tempuh aktual perusahaan, data armada yang digunakan, serta data permintaan frozen

food. Setelah data yang diperlukan terkumpul langkah selanjutnya adalah melakukan pengolahan data menggunakan algoritma genetika yang terbagi kedalam beberapa tahap yaitu inialisasi (encoding), evaluasi fitness, seleksi, pindah silang (crossover), mutasi, regenerasi, elitisme dan terminasi.

2.1 Inialisasi Rute (Encoding)

Parameter atau jalur-jalur distribusi yang akan dituju dikodekan terlebih dahulu menggunakan permutation encoding (inialisasi permutasi) sesuai dengan ketentuan Travelling Salesman Problem (TSP). Tujuan pengkodean ini merepresentasikan suatu permutasi dari nomor urut kota 1, 2, 3, ..., n (Gen & Cheng, 1997).

- **Evaluasi Nilai Fitness**

Evaluasi nilai fitness dilakukan dengan cara menghitung nilai fitness (kebugaran) setiap jalur dari populasi random yang dihasilkan oleh proses encoding untuk memudahkan proses seleksi dalam mencari individu terbaik.

2.3 Seleksi

Proses seleksi merupakan proses memilih kromosom-kromosom yang akan dipakai pada generasi selanjutnya sebagai parent yang didasarkan pada urutan nilai fitness, karena dengan nilai fitness yang baik maka kromosom tersebut akan dipertahankan. Metode seleksi yang akan digunakan adalah metode tournament selection yaitu metode yang akan memilih dua individu dengan nilai fitness terbaik dari sebuah populasi untuk dijadikan parent (orangtua) yang akan digunakan untuk proses crossover dan mutasi (Golberg, 1989). Proses seleksi akan menghasilkan dua parent dengan nilai fitness individu terkecil (jarak terpendek) dari populasi yang dihasilkan, karena sesuai dengan fungsi tujuan yang ingin dicapai yaitu mencari jarak rute terpendek.

2.4 Crossover (Pindah Silang)

Crossover atau pindah silang merupakan teknik membangkitkan dua bilangan acak dengan melibatkan dua induk (parent) untuk membentuk kromosom baru. Pindah silang dilakukan dengan harapan kromosom-kromosom baru yang terbentuk lebih baik daripada kromosom sebelumnya. Proses crossover akan berjalan apabila perintah nilai random antara 0-1 yang dihasilkan lebih kecil daripada nilai crossrate terpenuhi, apabila perintah tersebut tidak terpenuhi maka proses crossover akan dilewat dan masuk ke proses selanjutnya. Crossover (pindah silang) dilakukan atas 2 kromosom untuk menghasilkan kromosom anak (offspring). Kromosom anak yang terbentuk akan mewarisi sebagian sifat kromosom induknya. Teknik crossover yang akan digunakan yaitu order crossover dengan penentuan titik pindah silang dilakukan secara random permutation dari nilai jumlah kota yang akan dilewati.

2.5 Mutasi

Mutasi akan menciptakan individu baru dengan mengubah satu atau lebih gen yang terdapat dalam suatu kromosom. Mutasi berperan untuk menggantikan gen yang hilang dari populasi akibat proses seleksi dan memungkinkan munculnya gen yang tidak ada dalam populasi awal. Proses mutasi akan berjalan sama seperti proses crossover apabila perintah nilai random antara 0-1 yang dihasilkan lebih kecil daripada nilai mutrate terpenuhi, apabila perintah tersebut tidak terpenuhi maka proses mutasi akan dilewat dan masuk ke proses selanjutnya. Teknik mutasi yang akan digunakan yaitu swapping mutation, dimana akan dilakukan mutasi terhadap dua individu dengan menukar posisinya.

2.6 Regenerasi

Regenerasi dilakukan untuk proses membangkitkan populasi baru dengan tujuan membentuk populasi baru yang berbeda dengan populasi pendahulunya. Pembentukan populasi baru didapatkan dari anak (child) hasil proses crossover atau mutasi bila proses tersebut berjalan, akan tetapi apabila proses crossover atau mutasi tidak berjalan maka akan diambil individu baru dari hasil poses seleksi yaitu nilai parent.

2.7 Elitisme

Elitisme sebagai proses mempertahankan individu yang memiliki nilai fitness terbaik dalam suatu iterasi agar tetap ada selama proses evolusi berlangsung. Selain itu tujuan dari elitisme adalah menjaga agar hasil fitness terbaik yang telah didapat tidak rusak (nilai fitness turun) akibat proses pindah silang atau mutasi berikutnya. Oleh sebab itu dengan adanya elitisme individu terbaik akan tetap muncul di dalam populasi pada iterasi berikutnya.

2.8 Terminasi

Terminasi merupakan acuan berhentinya iterasi. Terminasi yang digunakan yaitu akan berhenti apabila nilai maksimal iterasi terpenuhi, dengan nilai maksimal iterasi yang akan digunakan adalah 100.000 kali.

Setelah dilakukan pengolahan data, tahap selanjutnya adalah melakukan analisis. Tahap analisis dilakukan untuk mengetahui apakah hasil pengolahan data menggunakan metode algoritma genetika telah memenuhi tujuan yang ingin dicapai atau tidak. Setelah didapatkan hasil analisis, langkah selanjutnya membuat kesimpulan yang berisikan rangkuman penyelesaian masalah menggunakan metode algoritma genetika serta memberikan saran terhadap perusahaan dan penelitian selanjutnya.

3. HASIL PENELITIAN

3.1 Matriks Jarak Antar Lokasi

Matriks jarak antar lokasi menunjukkan jarak antar setiap lokasi yang dibuat kedalam tabel matriks dengan satuan km yang dapat dilihat pada Tabel 1.

Tabel 1. Matriks Jarak Antar Lokasi

| Matriks Jarak | | | | | | | | | | | |
|-------------------------|------|------|------|------|------|------|------|------|------|-----|------|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | ... | 48 |
| 1 Depot | 0 | 21.8 | 15 | 15 | 21.9 | 14.7 | 22.6 | 13.8 | 38.5 | ... | 14.6 |
| 2 ARDI frozen foods | 21.8 | 0 | 20.9 | 20.8 | 0.13 | 27.3 | 0.65 | 15.9 | 38.3 | ... | 20.4 |
| 3 A&A Frozen Food | 15 | 20.9 | 0 | 0.55 | 20.7 | 8.1 | 21.3 | 5.7 | 50.7 | ... | 0.45 |
| 4 ATAYA Frozen Food | 15 | 20.8 | 0.55 | 0 | 20.7 | 8 | 21.3 | 5.6 | 50.7 | ... | 0.6 |
| 5 Azva Frozen | 21.9 | 0.13 | 20.7 | 20.7 | 0 | 27.2 | 0.55 | 15.8 | 38.2 | ... | 20.3 |
| 6 Biant Frozen Food | 14.7 | 27.3 | 8.1 | 8 | 27.2 | 0 | 27.1 | 49 | 1.1 | ... | 7 |
| 7 Cipta Frozen Majalaya | 22.6 | 0.65 | 21.3 | 21.3 | 0.55 | 27.1 | 0 | 16.3 | 37.7 | ... | 20.8 |
| 8 Fadlan Frozen Food | 13.8 | 15.9 | 5.7 | 5.6 | 15.8 | 49 | 16.3 | 0 | 49.5 | ... | 5.2 |
| 9 Fortuna Frozen Food | 38.5 | 38.3 | 50.7 | 50.7 | 38.2 | 1.1 | 37.7 | 49.5 | 0 | ... | 49.6 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 48 SAKANA Frozen Food | 14.6 | 20.4 | 0.45 | 0.6 | 20.3 | 7 | 20.8 | 5.2 | 49.6 | ... | 0 |

Jarak rute aktual perusahaan saat ini adalah sejauh 212,59 km, hasil tersebut didapat karena penentuan jarak yang dilakukan hanya berdasarkan perkiraan.

3.2 Algoritma Genetika

Langkah pertama adalah inisialisasi rute untuk menciptakan rute atau individu (kromosom) dengan susunan gen tertentu sebanyak 48 gen. Listing proses inisialisasi rute dapat dilihat pada Tabel 2.

Tabel 2. Listing Inisialisasi Rute

```
%MEMBANGKITKAN POPULASI
for i = 1:size_populasi
    populasi(:,i) = randperm(kota);
    pop_random = populasi;
```

Populasi sebagai variabel inisialisasi rute mengambil data secara random permutation (permutasi acak) dari variabel kota yang terdiri dari 48 kota. Lalu data variabel populasi tersebut disimpan pada variabel `pop_random` untuk ditampung. Langkah selanjutnya adalah evaluasi nilai fitness yaitu menghitung nilai jarak antar lokasi untuk setiap individu yang dihasilkan proses inisialisasi untuk memudahkan proses seleksi. Listing proses evaluasi nilai fitness dapat dilihat pada Tabel 3.

Tabel 3. Listing Evaluasi Nilai Fitness

```
JarakTempuh(i,:) = jarak(pop(1,i),pop(2,i))+jarak(pop(2,i),pop(3,i))+jarak(pop(3,i),pop(4,i))+jarak(pop(4,i)...
, pop(5,i))+jarak(pop(5,i),pop(6,i))+jarak(pop(6,i),pop(7,i))+jarak(pop(7,i),pop(8,i))+jarak(pop(8,i),pop(9,i))...
+jarak(pop(9,i),pop(10,i))+jarak(pop(10,i),pop(11,i))+jarak(pop(11,i),pop(12,i))+jarak(pop(12,i),pop(13,i))...
+jarak(pop(13,i),pop(14,i))+jarak(pop(14,i),pop(15,i))+jarak(pop(15,i),pop(16,i))+jarak(pop(16,i),pop(17,i))...
+jarak(pop(17,i),pop(18,i))+jarak(pop(18,i),pop(19,i))+jarak(pop(19,i),pop(20,i))+jarak(pop(20,i),pop(21,i))...
+jarak(pop(21,i),pop(22,i))+jarak(pop(22,i),pop(23,i))+jarak(pop(23,i),pop(24,i))+jarak(pop(24,i),pop(25,i))...
+jarak(pop(25,i),pop(26,i))+jarak(pop(26,i),pop(27,i))+jarak(pop(27,i),pop(28,i))+jarak(pop(28,i),pop(29,i))...
+jarak(pop(29,i),pop(30,i))+jarak(pop(30,i),pop(31,i))+jarak(pop(31,i),pop(32,i))+jarak(pop(32,i),pop(33,i))...
+jarak(pop(33,i),pop(34,i))+jarak(pop(34,i),pop(35,i))+jarak(pop(35,i),pop(36,i))+jarak(pop(36,i),pop(37,i))...
+jarak(pop(37,i),pop(38,i))+jarak(pop(38,i),pop(39,i))+jarak(pop(39,i),pop(40,i))+jarak(pop(40,i),pop(41,i))...
+jarak(pop(41,i),pop(42,i))+jarak(pop(42,i),pop(43,i))+jarak(pop(43,i),pop(44,i))+jarak(pop(44,i),pop(45,i))...
+jarak(pop(45,i),pop(46,i))+jarak(pop(46,i),pop(47,i))+jarak(pop(47,i),pop(48,i))+jarak(pop(1,i),pop(2,i));
```

JarakTempuh sebagai variabel penampung jarak inisialisasi rute akan menghitung jarak setiap satu lokasi ke lokasi selanjutnya berdasarkan posisi matriks pada variabel populasi dan nilai matriks jarak antar lokasi dengan rumus mengikuti ketentuan syarat TSP yaitu dimulai dari depot (titik awal) dan berakhir di depot. Langkah selanjutnya adalah proses seleksi untuk mendapatkan individu yang akan dijadikan parent (orangtua) pada proses crossover atau mutasi. Listing proses seleksi individu terbaik dapat dilihat pada Tabel 4.

Tabel 4. Listing Proses Seleksi Individu Terbaik

```
%SELEKSI
%[~,index] = min(JarakTempuh) adalah rumus untuk mencari individu dengan
%fitness paling minimal [nilai,individu ke-]
 [~,index] = min(JarakTempuh); %mencari individu terbaik pertama
parent1 = pop(:,index); %parent1 adalah individu ke-index dari populasi
jarakparent1 = JarakTempuh(index,:);

pop(:,index) = []; %rumus menghapus parent1 dari populasi
JarakTempuh(index) = [];

 [~,index] = min(JarakTempuh); %mencari individu terbaik kedua
parent2 = pop(:,index);
jarakparent2 = JarakTempuh(index,:);

pop(:,populasi) = parent1; %rumus untuk mengemablikan parent1 ke dalam populasi
JarakTempuh(populasi,:) = jarakparent1;
```

Proses seleksi yang digunakan adalah tournament selection yaitu mengambil dua individu dengan nilai fitness terkecil dari populasi untuk dijadikan parent (orangtua). Langkah selanjutnya adalah proses *crossover* atau pindah silang. Nilai *crossover* (P_c) yang digunakan adalah 0,78 dan menggunakan model *order crossover*. Listing proses *crossover* dapat dilihat

pada Tabel 5.

Tabel 5. Listing Proses Crossover

```
%CROSSOVER
if rand< crossrate
child1 = zeros(kota,1);
child2 = zeros(kota,1);
CP = round(kota*crossrate);
CO = kota-CP-1;
child1(1:CO,1) = parent1(1:CO,1); %bit 1-CO pada child1=bit 1-CO parent1
child2(1:CO,1) = parent1(1:CO,1);
for i = CO+1:kota
    if parent2(i,1) ~= child1(:,1) %jika nilai bit parent2 belum ada semua
        %di bit anak 1 maka bit sisanya akan dimasukan
        child1(i,1) = parent2(i,1);
    end
end
for i = CO+1:kota
    if child1(i,1) == 0
        for j = 1:kota
            if parent2(j,1) ~= child1(:,1)
                child1(i,1) = parent2(j,1);
            end
        end
    end
end
for i = CO+1:kota
    if parent1(i,1) ~= child2(:,1)
        child2(i,1) = parent1(i,1);
    end
end
for i = CO+1:kota
    if child2(i,1) == 0
        for j = 1:kota
            if parent1(j,1) ~= child2(:,1)
                child2(i,1) = parent1(j,1);
            end
        end
    end
end
end
```

Proses crossover akan berjalan apabila nilai random (rand) [0, 1] yang dibangkitkan bernilai kurang dari nilai crossrate, tetapi apabila perintah tidak terpenuhi maka proses crossover akan dilewat dan berlanjut ke proses berikutnya. Langkah selanjutnya adalah proses mutasi yang dilakukan untuk mengganti atau menukar gen kromosom yang ada pada child (anak). Listing proses mutasi dapat dilihat pada Tabel 6.

Tabel 6. Listing Proses Mutasi

```
%MUTASI
if rand < mutrate
    kromosom1 = round (1+rand*(kota-1)); %menentukan bit yang akan dimutasi
    kromosom2 = round (1+rand*(kota-1)); %menentukan bit yang akan dimutasi
    child1([kromosom1 kromosom2]) = child1([kromosom2 kromosom1]); %menukar kromosom
    child2([kromosom1 kromosom2]) = child2([kromosom2 kromosom1]); %menukar kromosom
    backup_child1(:,it) = child1;
    backup_child2(:,it) = child2;
end
```

Proses mutasi ini menggunakan metode swapping mutation yaitu menukar posisi 2 kromosom. Sama seperti proses crossover, proses mutasi akan berjalan apabila nilai random (rand) [0, 1] yang dibangkitkan lebih kecil dari nilai mutrate, tetapi apabila perintah tidak terpenuhi maka proses mutasi akan dilewat dan berlanjut ke proses berikutnya. Langkah selanjutnya adalah proses regenerasi untuk mengubah populasi awal menjadi populasi baru dengan mengubah

atau menghilangkan individu dari populasi lama dan ditambahkan individu baru dari hasil crossover atau mutasi. Listing proses regenerasi dapat dilihat pada Tabel 7.

Tabel 7. Listing Proses Regenerasi

```
%REGENERASI (POPULASI BARU)
jarakchild1 = jarak(child1(1),child1(2))+jarak(child1(2),child1(3))+jarak(child1(3),child1(4))...
+jarak(child1(4),child1(5))+jarak(child1(5),child1(6))+jarak(child1(6),child1(7))...
+jarak(child1(7),child1(8))+jarak(child1(8),child1(9))+jarak(child1(9),child1(10))...
+jarak(child1(10),child1(11))+jarak(child1(11),child1(12))+jarak(child1(12),child1(13))...
+jarak(child1(13),child1(14))+jarak(child1(14),child1(15))+jarak(child1(15),child1(16))...
+jarak(child1(16),child1(17))+jarak(child1(17),child1(18))+jarak(child1(18),child1(19))...
+jarak(child1(19),child1(20))+jarak(child1(20),child1(21))+jarak(child1(21),child1(22))...
+jarak(child1(22),child1(23))+jarak(child1(23),child1(24))+jarak(child1(24),child1(25))...
+jarak(child1(25),child1(26))+jarak(child1(26),child1(27))+jarak(child1(27),child1(28))...
+jarak(child1(28),child1(29))+jarak(child1(29),child1(30))+jarak(child1(30),child1(31))...
+jarak(child1(31),child1(32))+jarak(child1(32),child1(33))+jarak(child1(33),child1(34))...
+jarak(child1(34),child1(35))+jarak(child1(35),child1(36))+jarak(child1(36),child1(37))...
+jarak(child1(37),child1(38))+jarak(child1(38),child1(39))+jarak(child1(39),child1(40))...
+jarak(child1(40),child1(41))+jarak(child1(41),child1(42))+jarak(child1(42),child1(43))...
+jarak(child1(43),child1(44))+jarak(child1(44),child1(45))+jarak(child1(45),child1(46))...
+jarak(child1(46),child1(47))+jarak(child1(47),child1(48))+jarak(child1(48),child1(1)):

jarakchild2 = jarak(child2(1),child2(2))+jarak(child2(2),child2(3))+jarak(child2(3),child2(4))...
+jarak(child2(4),child2(5))+jarak(child2(5),child2(6))+jarak(child2(6),child2(7))...
+jarak(child2(7),child2(8))+jarak(child2(8),child2(9))+jarak(child2(9),child2(10))...
+jarak(child2(10),child2(11))+jarak(child2(11),child2(12))+jarak(child2(12),child2(13))...
+jarak(child2(13),child2(14))+jarak(child2(14),child2(15))+jarak(child2(15),child2(16))...
+jarak(child2(16),child2(17))+jarak(child2(17),child2(18))+jarak(child2(18),child2(19))...
+jarak(child2(19),child2(20))+jarak(child2(20),child2(21))+jarak(child2(21),child2(22))...
+jarak(child2(22),child2(23))+jarak(child2(23),child2(24))+jarak(child2(24),child2(25))...
+jarak(child2(25),child2(26))+jarak(child2(26),child2(27))+jarak(child2(27),child2(28))...
+jarak(child2(28),child2(29))+jarak(child2(29),child2(30))+jarak(child2(30),child2(31))...
+jarak(child2(31),child2(32))+jarak(child2(32),child2(33))+jarak(child2(33),child2(34))...
+jarak(child2(34),child2(35))+jarak(child2(35),child2(36))+jarak(child2(36),child2(37))...
+jarak(child2(37),child2(38))+jarak(child2(38),child2(39))+jarak(child2(39),child2(40))...
+jarak(child2(40),child2(41))+jarak(child2(41),child2(42))+jarak(child2(42),child2(43))...
+jarak(child2(43),child2(44))+jarak(child2(44),child2(45))+jarak(child2(45),child2(46))...
+jarak(child2(46),child2(47))+jarak(child2(47),child2(48))+jarak(child2(48),child2(1)):

[~,index] = max(JarakTempuh);
populasi(:,index) = child1;
JarakTempuh(index,:) =jarakchild1;

[~,index] = max(JarakTempuh);
populasi(:,index) = child2;
JarakTempuh(index,:) =jarakchild2;
```

Langkah-langkah dari listing regenerasi yaitu melakukan perhitungan nilai jarak dari child1 dan child2. Setelah didapatkan hasil perhitungan jaraknya, perintah selanjutnya adalah memanggil nilai jarak tempuh paling tinggi pada populasi awal untuk digantikan dengan nilai matriks baru dari child1 dan child2. Proses selanjutnya adalah elitisme untuk menyimpan individu dengan nilai jarak paling minimum. Listing proses elitisme dapat dilihat pada Tabel 8.

Tabel 8. Listing Proses Elitisme

```
%ELITISME
[x,index] = min(JarakTempuh) %[nilai,individu ke-]
elite(:,it) = populasi(:,index);
totaljarak(it,:) = JarakTempuh(index,:);
```

Proses elitisme dilakukan dengan cara mencari nilai paling minimum pada individu untuk setiap iterasi yang dilakukan dan hasilnya akan ditampung pada variabel elite. Selanjutnya dilakukan penyimpanan jarak tempuh dari setiap individu yang memiliki nilai jarak terkecil pada setiap iterasinya menggunakan variabel totaljarak sebagai penampungnya. Percobaan dilakukan sebanyak 10 kali dengan pengulangan sebanyak 50 kali dalam setiap percobaannya. Hasil terbaik dari percobaan 1-10 dapat dilihat pada Tabel 9.

Hasil percobaan yang memiliki nilai jarak rute paling kecil adalah percobaan ke-7 dengan jarak rute sejauh 171,47 km dan hasil rutenya dapat dilihat pada Tabel 10.

Usulan Rute Distribusi Frozen Food Menggunakan Algoritma Genetika Pada Perusahaan Distributor Di Bandung

Tabel 9. Hasil Terbaik Percobaan 1-10

| Percobaan Ke- | Parameter Mutasi | Hasil (km) |
|---------------|------------------|------------|
| 1 | 15% | 214,78 |
| 2 | 20% | 206,17 |
| 3 | 30% | 201,63 |
| 4 | 40% | 206,17 |
| 5 | 50% | 191,12 |
| 6 | 60% | 194,32 |
| 7 | 70% | 171,47 |
| 8 | 80% | 211,02 |
| 9 | 90% | 208,40 |
| 10 | 100% | 204,02 |

Tabel 10. Hasil Pencarian Rute Terbaik Percobaan Ke-7

| No. | No. Kode Lokasi | Nama Lokasi | Pembuktian Jarak (Km) |
|-----|-----------------|-------------------------------------|-----------------------|
| 1 | 1 | Depot (Perusahaan) | - |
| 2 | 18 | Pasar Cibiuk | 10,2 |
| 3 | 31 | Pasar Rancamanyar | 5,77 |
| 4 | 4 | ATAYA Frozen food | 1,3 |
| 5 | 3 | A&A Frozen food | 0,5 |
| 6 | 48 | SAKANA Frozen food | 0,45 |
| 7 | 33 | Pasar Dayeuhkolot | 4,2 |
| 8 | 47 | Mabeku Frozen food | 1 |
| 9 | 16 | Pasar Baleendah | 0,5 |
| 10 | 32 | Pasar Tonggeng | 2 |
| 11 | 8 | Fadlan Frozen food | 2 |
| 12 | 20 | Pasar Ciparay | 10,4 |
| 13 | 44 | Raffasha frozen food s Ciparay | 0,21 |
| 14 | 22 | Pasar Wangisagara | 6,8 |
| 15 | 23 | Pasar Niagara | 0,18 |
| 16 | 5 | Azva Frozen | 4,3 |
| 17 | 2 | ARDI frozen foods | 0,13 |
| 18 | 7 | Cipta Frozen Majalaya | 0,65 |
| 19 | 21 | Pasar Majalaya | 0,19 |
| 20 | 37 | Pasar Tanjunglaya | 8,6 |
| 21 | 30 | Pasar Swamandiri | 0,85 |
| 22 | 40 | Toko Frozenfood Solusi | 2,3 |
| 23 | 17 | Pasar Baru Cicalengka | 1,8 |
| 24 | 39 | Toko CBS (Cicalengka Bakso Sosis) | 0,05 |
| 25 | 29 | Pasar Lama Cicalengka | 0,21 |
| 26 | 41 | Toko Ladzidz Frozen food | 3,7 |
| 27 | 42 | Pondok Frozen food | 5,2 |
| 28 | 10 | Herry Frozen food | 0,9 |
| 29 | 43 | Val Frozen food Cedeas Suki Bandung | 4 |
| 30 | 28 | Pasar Wahana Karya Rancaekek | 2,5 |
| 31 | 24 | Pasar Dangdeur | 2 |
| 32 | 34 | Pasar Tanjungsari | 9 |
| 33 | 14 | Nsb Frozen food | 1,1 |
| 34 | 38 | Toko Rumah Beku | 17,5 |
| 35 | 6 | Biant Frozen food | 0,85 |
| 36 | 9 | Fortuna Frozen food | 1,1 |
| 37 | 35 | Pasar Sumedang Kota | 0,5 |
| 38 | 27 | Pasar Resik Jatinangor | 25 |

| | | | |
|--------------------|----|---------------------------------|---------------|
| 39 | 12 | M.J Cat | 1 |
| 40 | 25 | Pasar Minggu Manglayang | 4,2 |
| 41 | 26 | Pasar Manglayang | 3,53 |
| 42 | 11 | Khanza Frozen food | 1,5 |
| 43 | 13 | Master Frozen | 2,4 |
| 44 | 45 | Permata Frozen food | 0,9 |
| 45 | 36 | Ujung Berung Traditional Market | 5,1 |
| 46 | 46 | Toko Mamina Frozen food | 1,7 |
| 47 | 19 | Pasar Cicaheum | 4,7 |
| 48 | 15 | Pasar Antapani | 1,5 |
| 49 | 1 | Depot (Perusahaan) | 7,2 |
| Total Jarak | | | 171,47 |

Setelah proses elitisme, langkah selanjutnya adalah proses terminasi, yaitu proses yang dilakukan untuk pengulangan iterasi, sehingga iterasi akan berulang terus sampai kondisi berhenti tercapai. Listing inisialisasi rute dapat dilihat pada Tabel 11.

Tabel 11. Listing Proses Terminasi

```

%TERMINASI
if it ==maxiterasi
    GATSP = false;
else
    it = it+1;
    GATSP = true;
end
    
```

Program akan berhenti apabila it sama dengan nilai maxiterasi yang diatur sebanyak 100.000 kali. Akan tetapi apabila it belum sama dengan maxiterasi maka iterasi akan berlanjut dan nilai iterasi akan bertambah sesuai dengan rumus $it = it+1$.

3.3 Perhitungan penghematan Total Jarak Tempuh

Perhitungan penghematan total jarak tempuh dapat dilihat pada Tabel 12.

Tabel 12. Perhitungan Penghematan Total Jarak Tempuh

| Rute Aktual Perusahaan (Km) | Rute Rancangan (Km) | Penghematan Jarak Total (Km) |
|-----------------------------|---------------------|------------------------------|
| 212,59 | 171,47 | 41,12 |

$$\begin{aligned}
 \text{Penghematan Jarak} &= \frac{(\text{Jarak Rute Aktual} - \text{Jarak Rute Rancangan})}{\text{Jarak Rute Rancangan}} \times 100\% \quad (1) \\
 &= \frac{212,59 - 171,47}{171,47} \times 100\% \\
 &= 23,98\%
 \end{aligned}$$

Penghematan total jarak dari hasil perhitungan adalah 41,12 km atau sebanyak 23,98% untuk setiap satu kali pengiriman.

3.4 Perhitungan Penghematan Total Waktu

Perhitungan total waktu tempuh rute aktual perusahaan:

$$\begin{aligned}
 \text{Waktu Tempuh} &= \frac{\text{Jarak Tempuh (km)}}{\text{Kecepatan } \left(\frac{\text{km}}{\text{jam}}\right)} \quad (2) \\
 &= \frac{212,59 \text{ km}}{40 \frac{\text{km}}{\text{jam}}} \\
 &= 5,31 \text{ jam}
 \end{aligned}$$

Perhitungan total waktu tempuh rute rancangan metode algoritma genetika:

$$\text{Waktu Tempuh} = \frac{\text{Jarak Tempuh (km)}}{\text{Kecepatan } \left(\frac{\text{km}}{\text{jam}}\right)} \quad (3)$$

Usulan Rute Distribusi Frozen Food Menggunakan Algoritma Genetika Pada Perusahaan Distributor Di Bandung

$$= \frac{171,47 \text{ km}}{40 \frac{\text{km}}{\text{jam}}}$$

$$= 4,29 \text{ jam}$$

Perhitungan total waktu unloading:

$$\begin{aligned} \text{Total Waktu Unloading} &= \text{Waktu Unloading} \times \text{Total Pemberhentian} & (4) \\ &= 5 \text{ menit} \times 47 \\ &= 3,92 \text{ jam} \end{aligned}$$

Perhitungan total waktu perjalanan untuk rute aktual:

$$\begin{aligned} \text{Total Waktu Aktual} &= \text{Waktu Aktual} + \text{Total Waktu Unloading} + \text{Waktu Loading} & (5) \\ &= 5,31 + 3,92 + 0,33 \\ &= 9,56 \text{ jam} \end{aligned}$$

Perhitungan total waktu perjalanan untuk rute rancangan metode algoritma genetika:

$$\begin{aligned} \text{Total Waktu Rancangan} &= \text{Waktu Rancangan} + \text{Total Waktu Unloading} + \text{Waktu Loading} & (6) \\ &= 4,29 + 3,92 + 0,33 \\ &= 8,54 \text{ jam} \end{aligned}$$

Penghematan total waktu yang dihasilkan dengan penggunaan metode algoritma genetika:

$$\begin{aligned} \text{Waktu penghematan} &= \text{waktu aktual (jam)} - \text{waktu rancangan (jam)} & (7) \\ &= 9,56 \text{ jam} - 8,54 \text{ jam} \\ &= 1,02 \text{ jam} \end{aligned}$$

3.5 Perhitungan Penghematan Total Biaya Distribusi

Dasar perhitungan biaya bahan bakar dilihat dari beberapa ketentuan, seperti kendaraan yang digunakan berjenis suzuki carry box freezer dengan bahan bakar pertalite, harga bahan bakar pertalite Rp7.850/liter, dan 1 liter pertalite dapat menempumpuh jarak ± 8 km.

Perhitungan biaya bahan bakar pada kondisi rute aktual perusahaan:

$$\begin{aligned} \text{Biaya Bahan Bakar} &= \frac{\text{Total Jarak Tempuh (km)}}{8 \text{ km}} \times \text{Harga BBM per Liter} & (8) \\ &= \frac{212,59 \text{ km}}{8 \text{ km}} \times \text{Rp}7.850 \\ &= \text{Rp}208.603,93 \end{aligned}$$

Perhitungan biaya bahan bakar pada kondisi rute rancangan metode algoritma genetika:

$$\begin{aligned} \text{Biaya Bahan Bakar} &= \frac{\text{Total Jarak Tempuh (km)}}{8 \text{ km}} \times \text{Harga BBM per Liter} & (9) \\ &= \frac{171,47 \text{ km}}{8 \text{ km}} \times \text{Rp}7.850 \\ &= \text{Rp}168.254,93 \end{aligned}$$

Hasil penghematan biaya bahan bakar, yaitu:

$$\begin{aligned} \text{Penghematan Biaya} &= \text{Biaya Aktual} - \text{Biaya Rancangan} & (10) \\ &= \text{Rp}208.603,93 - \text{Rp}168.254,93 \\ &= \text{Rp}40.349/\text{sekali distribusi} \end{aligned}$$

Perhitungan persentase hasil penghematan biaya bahan bakar, yaitu:

$$\begin{aligned} \text{Persentase Penghematan} &= \frac{\text{Hasil Penghematan}}{\text{Biaya Aktual}} \times 100\% & (11) \\ &= \frac{\text{Rp}40.349}{\text{Rp}208.603,93} \times 100\% \\ &= 19,34\% \end{aligned}$$

3.6 Analisis

Penggunaan TSP dan algoritma genetika dapat menghasilkan jarak rute yang lebih pendek jika dibandingkan jarak rute aktual perusahaan. Hasil dari pengolahan data menunjukkan pencarian rute terpendek dengan metode algoritma genetika dapat menghemat jarak sejauh 41,12 km atau sebesar 23,98% dari jarak rute aktual perusahaan, menghemat waktu pengiriman sebanyak 1,02 jam dan menghemat pengeluaran sebesar Rp40.349 atau sebesar 19,34%, sehingga dapat membuat pendistribusian menjadi lebih efektif dan efisien, meminimasi pengeluaran perusahaan serta dipercaya konsumen dalam ketepatan waktu pengiriman karena penghematan jarak yang dihasilkan.

Hasil rute distribusi terpilih didapatkan dari percobaan ke-7 dengan perubahan parameter mutasi menjadi 70%. Hasil dari percobaan yang telah dilakukan dapat dilihat bahwa kenaikan parameter mutasi menyebabkan jarak rute yang dihasilkan menjadi lebih kecil, akan tetapi pada percobaan ke-8, percobaan ke-9, dan percobaan ke-10 hasil jarak yang didapatkan menjadi kurang baik lagi yaitu berada di angka lebih dari 200 km. Hal tersebut bisa terjadi karena terlalu banyak proses mutasi yang dilakukan sehingga menghasilkan anak yang kurang optimal sebab proses mutasi dibangkitkan secara random dari 0-1.

4. KESIMPULAN

Metode algoritma genetika merupakan metode yang digunakan dalam teknik pencarian (searching) dan optimasi yang dapat digunakan dalam pencarian rute terpendek dimana prinsip kerja dari algoritma genetika adalah meniru proses evolusi manusia pada struktur genetik dan seleksi alam. Setelah dilakukan pengolahan data, algoritma genetika dapat menghasilkan jarak rute yang lebih pendek jika dibandingkan jarak rute aktual perusahaan. Hasil dari percobaan 1 sampai dengan percobaan 10 masing-masing didapatkan nilai jarak rute paling kecil sebesar 214,78 km, 206,17 km, 201,63 km, 206,17 km, 191,12 km, 194,32 km, 171,47 km, 211,02 km, 208,40 km, dan 204,02 km.

Hasil dari pengolahan data menunjukkan pencarian rute terpendek dengan metode algoritma genetika dapat menghemat jarak sejauh 41,12 km atau sebesar 23,98% dari jarak rute aktual perusahaan pada percobaan ke-7. Dapat menghemat waktu tempuh sebanyak 1,02 jam yang bisa membuat perusahaan lebih tepat waktu dalam melakukan pengiriman sehingga dipercaya oleh konsumen, selain itu juga dapat menghemat pengeluaran sebesar Rp40.349/minggu (sekali distribusi) atau sebesar 19,34%. Parameter yang dirubah pada saat melakukan percobaan 1-10 hanya parameter mutasi yang masing-masing bernilai 15%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, dan 100%, hal tersebut dilakukan karena parameter mutasi sangat berpengaruh terhadap anak yang dihasilkan.

DAFTAR PUSTAKA

- Fallo, D. Y. (2015). Perbandingan Algoritma Ant Colony dan Algoritma Genetika untuk Pencarian Jarak Terpendek dalam Pengangkutan Hasil Tambang.
- Gen, M., & Cheng, R. (1997). Genetic Algorithms & Engineering Design. Hoboken: John Wiley & Sons, Inc.
- Golberg, D. E. (1989). Genetic Algorithms in Search Optimization & Machine Learning . Alabama: Addison-Wesley.
- Greco, F. (2008). Travelling Salesman Problem. Vienna: In-Teh.