

ANALISA OPTIMALISASI RUTE DISTRIBUSI UNTUK MENGEFISIENSIKAN LOGISTIK MENGGUNAKAN ALGORITMA GENETIKA

Oleh:

Nur Qomaruddin Saputra,

Tedjo Sukmono

Progam Studi Teknik Industri

Universitas Muhammadiyah Sidoarjo

Juli, 2024



Pendahuluan

- Munculnya e-commerce di Indonesia ini sangat mendorong berkembangnya ekonomi dan trend masyarakat dengan berbelanja online sangat meningkat dari tahun ketahun. Keberadaan penyedia layanan angkutan barang memainkan peran strategi untuk meningkatkan arus barang dari dan ke suatu daerah. Industri jasa ekspedisi ternama merupakan perusahaan pengiriman express yang berbasis teknologi. Salah satu permasalahan yang di hadapi perusahaan adalah optimalisasi pengiriman. Banyak nya rute saat ini mengakibatkan biaya pendistribusian cukup besar karena terdapat rute yang tidak teratur dengan baik, sehingga kalah bersaing dengan perusahaan kompetitor, target biaya pengiriman adalah sebesar 2.8 % sedangkan saat ini biaya pengiriman sebesar 3.3 % perhari. Proses pengiriman atau distribusi yang di laksanakan jika tidak memperhatikan rute dan jadwal pengiriman yang telah di tentukan dan dilakukan-nya berulang ulang maka menyebabkan biaya pengiriman menjadi besar
- Dalam mencoba mengatasi permasalahan pendistribusian tersebut metode algoritma genetika diusulkan menjadi solusi pada penelitian ini dengan tujuan untuk menyederhanakan atau mengoptimalkan rute agar mendapatkan jarak rute terpendek dan biaya yang rendah. Penelitian sebelum-nya algoritma genetika dapat direkomendasikan untuk penyelesaian distribusi dengan metode multi traveling salesman problem memodifikasi operator cycle crossover dimana hasil pengujian yang dilakukan dapat disimpulkan dapat membuat rute yang optimal dengan jarak terpendek [2]. Penelitian terkait selanjutnya, agar mendapatkan jarak terpendek dan tercepat dalam proses distribusi, disimpulkan bahwa metode Vehicle Routing Problem (VRP) yang di selesaikan algoritma genetika dapat menghasilkan rute terpendek dan waktu tercepat sehingga dapat mengoptimalkan pendistribusian

Pendahuluan

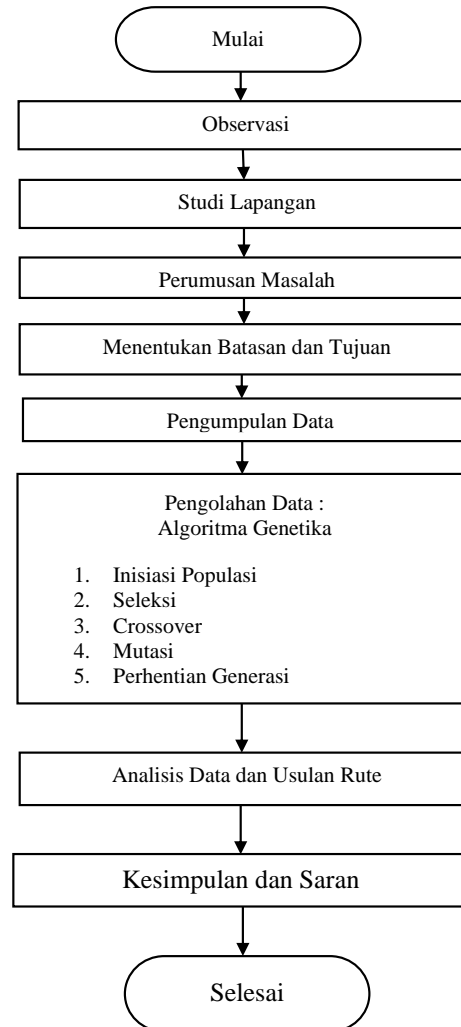
- Kemudian pada penelitian terkait selanjutnya, dimana menggunakan metode Capacity yang optimal meskipun tidak mengalami error dalam uji cobanya, algoritma genetika dapat menyelesaikan meskipun dengan jumlah titik yang banyak [4]. Sementara penelitian yang dilakukan Sihombing menyatakan bahwa penggunaan metode Algoritma Genetika dapat dan mampu menghasilkan rute yang efektif dan terpendek dan dapat menyelesaikan masalah-masalah yang tidak dapat diselesaikan dengan menggunakan perhitungan matematika biasa
- Kemudian penelitian yang dilakukan oleh Amdani menjelaskan bahwa tingkat ketepatan dan akurasi dengan menggunakan metode algoritma genetika mendapat nilai terbaik dan paling tinggi melalui tingkat akurasi sebesar 95.55 % [6]. Dengan demikian implementasi metode algoritma genetika diharapkan perusahaan dapat menemukan solusi terkait pendistribusian sehingga kedepan-nya lebih baik lagi dan dapat mengefisiensi biaya bahan bakar pada pengiriman

Pertanyaan Penelitian (Rumusan Masalah)

1. Membuat model rute pendistribusian baru dengan jarak terpendek sebagai evaluasi dalam ketepatan waktu pengiriman.
2. Mengevaluasi biaya penggunaan bahan bakar pada unit yang digunakan dalam pendistribusian barang.

Metode

- Alur Penelitian



Metode

- Algoritma Genetika

Penelitian ini menggunakan algoritma genetika. Algoritma genetika dapat menyelesaikan masalah distribusi dengan cara mencari rute terpendek dalam suatu rute distribusi. Algoritma Genetika merupakan sebuah metode optimasi yang didapatkan dari buah pikiran terhadap proses seleksi alam. Algoritma genetika dimulai dengan merpresentasikan solusi dari suatu masalah yang akan dipecahkan ke dalam urutan kromosom yang kemudian menghitung nilai fitness dan memilih individu mana yang memiliki nilai fitness terbaik. Algoritma genetika adalah algoritma pencarian heuristik yang didasarkan pada mekanisme evaluasi biologis, keanekaragaman dalam evaluasi biologis adalah keanekaragaman kromosom di antara individu organisme. Individu yang lebih kuat (lebih bugar) memiliki tingkat kelangsungan hidup dan reproduksi yang lebih tinggi dibandingkan dengan individu yang kurang bugar

Hasil dan Pembahasan

- Import Library

Gambar berikut Merupakan model pada software google colabotatory dilakukan import library pada python sesuai yang dibutuhkan dalam proses pembuatan model

```
import numpy as np
import random
from array import *
```

- Membuat Daftar Kota

Data yang di kumpulkan sebelumnya pada exel kemudian di masukkan titik koordinat daftar kota pada kolom daftar kota tempat tujuan yang akan di operasikan sesuai dengan tampilan pada gambar dibawah ini

```
[ ]
daftarKota = [[-7.22351894399141,112.7314441441122], [-7.31463778287319,112.784173901965], [-7.26255683865382,112.666996403086]]
for i in daftarKota:
    for j in i:
        print(j,end = " ")
    print()

jumlahKota = len(daftarKota)
print(jumlahKota)
```

Hasil dan Pembahasan

- Inisialisasi Populasi

Pada tahap inisiasi populasi ini operasi inisialisasi ini dilakukan secara acak dengan memanfaatkan random permutation dengan ukuran jumlah kota yang akan dituju. Perintah yang digunakan tersaji pada gambar sebagai berikut.

```
[ ]
def initPopulasi(ukuranPopulasi, jumlahKota):
    P = np.empty((ukuranPopulasi, jumlahKota))
    P = P.astype('int')
    for i in range(ukuranPopulasi):
        I = np.random.permutation(jumlahKota)
        for j in range(jumlahKota):
            P[i][j]=I[j]
    return P

ukuranPopulasi = 4
P = initPopulasi (ukuranPopulasi, jumlahKota)
print(P)
```


Hasil dan Pembahasan

- Membuat Fungsi Objektif

Pada tahap ini membuat fungsi objektif dengan inputan jumlah populasi dan jumlah kota yang akan dituju, kemudian pada tahap ini juga mengukur jarak dengan melakukan pengulangan dari individu setiap rute dimana pengukuran tersebut di hitung dengan jarak euclidian. Perintah yang digunakan tersaji pada gambar sebagai berikut

```
[ ]
def fungsiObjektif(populasi,daftarKota):
    sz = populasi.shape
    ukuranPopulasi = sz[0]
    jumlahKota = sz[1]

    matrikJarak = np.empty((ukuranPopulasi))
    matrikJarak = matrikJarak.astype('float')
    n = len(daftarKota)

    for i in range(ukuranPopulasi):
        jarak = 0
        for j in range(n-1):
            jrkX = daftarKota[populasi[i][j]][0]-daftarKota[populasi[i][j+1]][0]
            jrkY = daftarKota[populasi[i][j]][1]-daftarKota[populasi[i][j+1]][1]
            d = np.sqrt(jrkX**2 + jrkY**2 )
            jarak = jarak + d
```

```
dn = np.sqrt((daftarKota[populasi[i][n-1]][0]-daftarKota[populasi[i][0]][0])**2+ (daftarKota[populasi[i][n-1]][1]-daftarKota[populasi[i][0]][1])**2 )
jarak = jarak + dn
matrikJarak[i] = jarak

return matrikJarak

fitP = fungsiObjektif(P,daftarKota)
print(fitP)
#print(fitP.dtype)
```

Hasil dan Pembahasan

- Seleksi

Pada tahap ini adalah tahap seleksi dimana tahap tersebut digunakan untuk memilih 2 individu terbaik yang akan digunakan sebagai induk dengan metode turnamen, dengan memilih individu secara acak pada populasi. Dimana perintah tersebut tersajikan pada gambar dibawah ini sebagai berikut.

```
[ ]

def seleksi (P,fsObj):
    jmlInduk = 2
    sz = P.shape
    ukuranPopulasi = sz[0]
    jumlahKota = sz[1]

    Induk = np.empty((jmlInduk,jumlahKota))
    Induk = Induk.astype('int')

    for i in range (2):
        # memilih dua individu secara acak
        r1 = random.randint(0,ukuranPopulasi-1)
        r2 = random.randint(0,ukuranPopulasi-1)

        if fsObj[r1] < fsObj [r2]:
            for j in range(jumlahKota):
                Induk [i][j] = P[r1][j]
        else:
            for j in range(jumlahKota):
                Induk [i][j] = P[r2][j]
    return Induk

Induk = seleksi(P,fitP)
print(Induk)
```

Hasil dan Pembahasan

- Crossover

Pada tahap ini merupakan tahap reproduksi dimana tahapan proses reproduksi pada algoritma genetika bertujuan untuk memperoleh kromosom atau individu baru berdasarkan individu yang sudah ada dengan tetap menjaga sifat-sifat baik dari induknya. Proses reproduksi melibatkan dua proses yaitu crossover dan mutasi (mutasi). Dari proses sebelumnya terdapat hasil 2 induk terbaik yang akan digunakan pada proses crossover ini, kemudian induk 1 dan induk 2 akan dibangkitkan secara random agar mendapatkan anak 1 dan anak 2 dan setiap pembangkitan dicek apakah hasilnya tidak dapat dimasukkan pada hasil anak pada operasi tersebut. Perintah yang digunakan dapat terlihat pada gambar berikut

```
[ ]
# crossover

def OrderXO(Induk):
    sz = Induk.shape
    brs = sz[0]
    kol = sz[1]
    #print(sz, brs, kol)

    Anak = np.empty((2, kol))
    Anak = Anak.astype('int')

    r1 = random.randint(1, kol-2)

    for i0 in range(r1):

        Anak [0][i0] = Induk[0][i0]
        Anak [1][i0] = Induk[1][i0]

    batasCek1 = r1
    batasCek2 = r1

    for i in range(kol):
        m = (i+r1)%kol

        n1 = Induk[0][i]
        n2 = Induk[1][i]

    #mengecek apakah kota tertentu sudah ada di anak
    idxAda1 = 0
    idxAda2 = 0
```

```
for j1 in range(batasCek1):

    if batasCek1 > (kol-1):
        idxAda1 = 1

    if Anak [0][j1] == n2:
        idxAda1 = 1

for j2 in range(batasCek2):

    if batasCek2 > (kol-1):
        idxAda2 = 1

    if Anak [1][j2] == n1:
        idxAda2 = 1

if idxAda1 == 0:
    Anak[0][batasCek1] = n2
    batasCek1 = batasCek1 + 1

if idxAda2 == 0:
    Anak[1][batasCek2] = n1
    batasCek2 = batasCek2 + 1

return Anak

AnakCX = OrderXO(Induk)
print("Anak :", AnakCX)
```

Hasil dan Pembahasan

- Mutasi

Pada tahap mutasi ini adalah proses lanjutan dalam urutan operasi reproduksi dimana proses mutasi ini menggunakan swab dimana inputannya adalah induknya kemudian dilakukan inisiasi dan memilih 2 titik secara random dan dilakukan pembangkitan. Perintah yang digunakan dapat terlihat pada gambar berikut

```
[ ]
def MutasiSwap(Induk):
    sz = Induk.shape
    brs = sz[0]
    kol = sz[1]

    Anak = np.empty((2,kol))
    Anak = Anak.astype('int')

    Anak = Induk

    r1 = random.randint(0,kol-1)
    r2 = random.randint(0,kol-1)

    while r1 == r2:
        r2 = random.randint(0,kol-1)

    I1rs1 = Induk [0][r1]
    I1rs2 = Induk [0][r2]
    I2rs1 = Induk [1][r1]
    I2rs2 = Induk [1][r2]

    Anak [0][r1] = I1rs2
    Anak [0][r2] = I1rs1
    Anak [1][r1] = I2rs2
    Anak [1][r2] = I2rs1

    return Anak

print("AnakCX :",AnakCX)
AnakM = MutasiSwap(AnakCX)
print("AnakM :",AnakM)
```

Hasil dan Pembahasan

- Elitism

Pada tahap ini adalah proses pengecekan individu mana yang memiliki kualitas jelek dengan mendapat nilai yang paling besar dengan rute yang paling panjang. Perintah yang digunakan dapat terlihat pada gambar berikut

```
def Elitism(P, Anak, fitP, fitAnak):  
    szP = P.shape  
    brsP = szP[0]  
    kolP = szP[1]  
    szA = Anak.shape  
    brsA = szA[0]  
    kolA = szA[1]  
  
    #print (fitP)  
    for i in range(brsA):  
        iJelek = fitP.max()  
        idxJelek = fitP.argmax()  
  
        if fitAnak[i]<iJelek:  
            fitP[idxJelek] = fitAnak[i]  
            for j in range (kolP):  
                P[idxJelek][j] = Anak[i][j]  
  
    return (P,fitP)  
  
fitAnak= fungsiObjectif(AnakM,daftarKota)  
print(P)  
print(AnakM)  
print(fitP)  
print(fitAnak)  
P = Elitism (P,AnakM,fitP,fitAnak)  
print(P)  
print(fitP)
```

Hasil dan Pembahasan

- Prosedur Algoritma genetika

Pada tahapan ini adalah tahapan dimana proses dari masing masing operator diatas dijalankan secara bersama dan secara keseluruhan Perintah yang digunakan dapat terlihat pada gambar berikut.

```
[ ]
# Inisialisasi paramater
PCX = 0.95
Pm = 0.01
ukuranPopulasi = 20
max_generasi = 60

# Inisialisasi populasi

P = initPopulasi (ukuranPopulasi,jumlahKota)
print("P0 :",P)

#melakukan evaluasi P
fitP = fungsiObjectif(P,daftarKota)
print("fitness populasi awal:",fitP)

print("fitness terbaik awal:",fitP.min())
print("rute awal :",P[fitP.argmin()])

for i in range(max_generasi):
    print("generasi :",i)
    Induk = seleksi(P,fitP)
    #print("Induk :",Induk)

    Anak = Induk
```

```
Anak = Induk

rxo = random.random()

if rxo < PCX:
    Anak = OrderXO(Induk)
    #print("AnakXO :",Anak)

    rm = random.random()
    if rm < Pm:
        Anak = MutasiSwap(Anak)
        #print("AnakR :",Anak)

    fitAnak = fungsiObjectif(Anak,daftarKota)
    #print("fitAnak :",fitAnak)

    print("fitness:",fitAnak.min())

    [P, fitP]= Elitism (P,Anak,fitP,fitAnak)

    print("fitness populasi akhir:",fitP)
    print("fitness akhir:",fitP.min())
    print("rute akhir :",P[fitP.argmin()])
```

Manfaat Penelitian

- Penggunaan algoritma genetika dapat memberikan solusi pendistribusian yang optimal
- Penggunaan algoritma genetika mampu menghasilkan efisiensi biaya pengiriman pada Perusahaan logistik

Kesimpulan

- Hasil penelitian dan analisis menunjukkan bahwa penyelesaian penyusunan rute terdekat dengan menggunakan metode algoritma genetika, perusahaan dapat memperpendek jarak tempuh rute pendistribusian barang dari gudang ke tempat tujuan dengan jarak tempuh total sebesar 1.521 km, dengan konsumsi bahan bakar mesin (solar) sebanyak 152.10 liter menggunakan unit CDD L dengan rasio 10 yang sudah ditetapkan oleh perusahaan. Biaya yang di keluarkan oleh perusahaan dalam setiap rute sebesar Rp. 10.342.800, dimana dalam satu hari berjalan sebanyak 6 kali ritase tanpa hari libur. Dengan menggunakan metode algoritma genetika untuk penyusunan rute tercepat perusahaan dapat menghemat waktu tempuh pengiriman dan efisiensi penggunaan BBM pada unit pendistribusian logostik, terjadi peningkatan efisiensi biaya operasional sebesar Rp. 16.748.400 dalam 1 bulan

Referensi

- [1] R. Wijayanti, "Optimalisasi Rute Distribusi Untuk Meminimalkan Biaya Pengiriman dengan Metode Saving Matrix dan Algoritma Nearest Neighbor di PT. XYZ," *Scientific Journal of Industrial Engineering*, vol. 3, no. 2, pp. 60–66, Sep. 2022.
- [2] Aswandi, S. Cokrowibowo, and A. Irianti, "Model Penentuan Rute Terpendek Penjemputan Sampah Menggunakan Metode MTSP dan Algoritma Genetika," *Journal of Applied Computer Science and Technology*, vol. 2, no. 1, pp. 43–48, Jun. 2021, doi: 10.52158/jacost.v2i1.168.
- [3] S. R. Andani, "Optimasi Rute Menggunakan Vehicle Routing Problem (VRP) Dengan Algoritma Genetika," *Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)*, vol. 4, no. 1, pp. 148–156, Jan. 2022.
- [4] A. N. Waridah and M. S. Madja, "Algoritma hybrid genetika pada capacitated vehicle routing problem (CVRP) dan implementasinya," *Jurnal MIPA Dan Pembelajarannya (JMIPAP)*, vol. 2, no. 8, pp. 1–10, May 2023.
- [5] D. E. Sihombing and F. Ahyaningsih, "Optimalisasi Rute Distribusi Air Minum Dalam Kemasan Menggunakan Algoritma Genetika Pada PT. Mual Natio Maju Bersama," *Jurnal Riset Rumpun Ilmu Pendidikan (JURRIPEN)*, vol. 2, no. 1, 2023.
- [6] S. Amdani and D. Permana, "Penerapan Algoritma Genetika Untuk Penyeleksian Variabel Pada Analisis Regresi Logistik Biner," *Jurnal Pendidikan Tambusai*, vol. 7, no. 2, pp. 3488–3853, 2023.
- [7] N. Ariyanti and N. L. Azizah, *Buku Ajar Mata Kuliah Teknik Optimasi*, 1st ed., vol. 1. Sidoarjo: UMSIDA Press, 2019.
- [8] F. A. S. Rifa and E. Yuliawati, "Optimalisasi Pengiriman Semen Curah Melalui Jalur Laut Menggunakan Algoritma Transportasi dan Penugasan," *Jurnal Teknologi dan Manajemen*, vol. 2, no. 1, pp. 7–12, Mar. 2021, doi: 10.31284/j.jtm.2021.v2i1.1524.

Referensi

- 9] H. Patmawati and Y. A. Nugroho, "Optimalisasi Rute Distribusi Matras Pada Penyelesaian Capacitated Vehicle Routing Problem Dengan Metode Algoritma Genetika," *Jurnal Cakrawala Ilmiah*, vol. 1, no. 11, pp. 2745–2756, Jun. 2022.
- [10] indah Fatikawati, Syaripuddin, and Moh. N. Huda, "Implementasi Algoritma Genetika dalam Menentukan Rute Terpendek Pendistribusian Barang PT. J&T Samarinda," *Jurnal Ilmiah Matematika*, vol. 2, no. 2, pp. 12–21, Mar. 2023.
- [11] R. B. Prastyo, B. Santoso, and Y. G. Wibowo, "Analisis Distribusi Selling-in UD. Bumi Pepaya untuk Meningkatkan Kinerja Pemasaran pada Perkebunan Pepaya di Desa Wonosobo Kecamatan Srono Kabupaten Banyuwangi," *Journal of Economics, Assets, and Evaluation*, vol. 1, no. 1, pp. 1–12, Aug. 2023.
- [12] E. B. Saputra, "Perancangan Aplikasi Pengaturan Kapal Untuk Transportasi Semen Curah di PT Semen Padang Dengan Menggunakan Algoritma Genetika," *Jurnal Siber Transportasi dan Logistik (JSTL)*, vol. 1, no. 1, pp. 1–9, Apr. 2023.
- [13] A. F. Sutanto, A. A. Soebroto, and I. Cholissodin, "Optimasi Rute Wisata Religi di Jawa Timur menggunakan Algoritme Genetika," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 7, no. 4, pp. 1586–1594, Apr. 2023.
- [14] A. S. Fauziah, I. Cholissodin, and B. Rahayudi, "Optimasi Pendistribusian Air Mineral menggunakan Algoritma Genetika," *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 2, pp. 966–978, Feb. 2022.

Referensi

- [15] G. C. Ramadhan, P. Bagus W, and Y. Diah Rosita, "Penentuan Rute Optimal Untuk Jasa Pengiriman Barang Menggunakan Algoritma Genetika," JTIM : Jurnal Teknologi Informasi dan Multimedia, vol. 5, no. 1, pp. 48–55, May 2023, doi: 10.35746/jtim.v5i1.322.
- [16] D. E. Sihombing and F. Ahyaningsih, "Optimalisasi Rute Distribusi Air Minum Dalam Kemasan Menggunakan Algoritma Genetika Pada PT. Mual Natio Maju Bersama," Jurnal Riset Rumpun Ilmu Pendidikan (JURRIPEN), vol. 2, no. 1, 2023.
- [17] S. Rohman, L. Zakaria, A. Asmiati, and A. Nuryaman, "Optimisasi Travelling Salesman Problem dengan Algoritma Genetika pada Kasus Pendistribusian Barang PT. Pos Indonesia di Kota Bandar Lampung," Jurnal Matematika Integratif, vol. 16, no. 1, p. 61, Apr. 2020, doi: 10.24198/jmi.v16.n1.27804.61-73.

