

Optimasi Penempatan Guru Menggunakan Algoritma Hibrida Genetika dan *Simulated Annealing*

S D P Aipassa¹, H Sriwindono^{*2},

^{1,2}Program Studi Informatika, Universitas Sanata Dharma, Yogyakarta, Indonesia

E-mail: stepanusdenny464@gmail.com¹, haris@usd.ac.id^{2*}

Abstrak. Pendidikan merupakan kebutuhan mutlak sepanjang hayat, dengan guru dan siswa sebagai elemen kunci dalam proses pembelajaran. Di Kabupaten Magelang, alokasi guru masih menghadapi tantangan dalam upaya optimalisasi. Hasil pengujian menggunakan algoritma HGSA menunjukkan bahwa konfigurasi parameter terbaik diperoleh pada populasi 60 kromosom, rasio mutasi: crossover 1:40, suhu awal 10°C, batas bawah suhu 0,1°C, laju penurunan suhu 0,9, dan 1.000 iterasi, menghasilkan rata-rata jarak terpendek sekitar 4,452 km. Analisis parameter menunjukkan bahwa peningkatan populasi, rasio mutasi, laju pendinginan dan suhu awal yang lebih rendah juga memberikan pada penurunan total jarak. Algoritma menunjukkan konvergensi pada iterasi maksimum dan batas suhu terendah. Dibandingkan dengan algoritma genetik murni, HGSA secara konsisten memberikan total jarak yang lebih kecil. Hal ini menunjukkan bahwa HGSA layak untuk dipertimbangkan sebagai metode yang efektif untuk mendukung perencanaan penempatan guru yang lebih optimal.

Kata kunci: Hibrida Genetika dan *Simulated annealing*; Penempatan Guru; Optimisasi

Abstract. Education is an essential lifelong necessity, with teachers and students serving as key elements in the learning process. In Magelang Regency, teacher allocation still faces challenges in achieving optimal placement. Testing using the HGSA algorithm indicates that the best parameter configuration consists of a population of 60 chromosomes, a mutation-to-crossover ratio of 1:40, an initial temperature of 10°C, a lower temperature limit of 0.1°C, a cooling rate of 0.9, and 1,000 iterations, producing an average shortest distance of approximately 4.452 km. Parameter analysis shows that increasing the population size, mutation ratio, cooling rate, and using a lower initial temperature contribute to reducing the total distance. The algorithm demonstrates convergence at the maximum iteration and minimum temperature limit. Compared to the pure genetic algorithm, HGSA consistently yields smaller total distances. These findings indicate that HGSA is a viable and effective method to support more optimal teacher placement planning.

Keywords: Hybrid Genetic and Simulated Annealing; Teacher Placement; Optimization

1. Pendahuluan

Pendidikan merupakan kebutuhan manusia yang mutlak dan harus dipenuhi sepanjang hayat[9] yang mengubah individu dari tidak bisa menjadi bisa, dan dari bisa menjadi ahli. Elemen kunci dalam proses ini adalah guru dan siswa. Namun, kualitas pendidikan di Indonesia masih menghadapi berbagai kendala, salah satu masalah yang ditemui adalah alokasi guru yang kurang optimal. Seperti diberitakan oleh ANTARA News[25], jarak antara rumah guru dan sekolah dapat sangat jauh, misalnya seorang guru perempuan yang harus menempuh sekitar 80 kilometer setiap hari karena lokasi domisili dan

sekolah berada di kabupaten yang berbeda, hal ini dapat berpengaruh pada kesehatan fisik dan psikis guru yang berdampak pada kinerja guru dan kualitas pendidikan.

Saat ini, Dinas Pendidikan Kabupaten Magelang menghadapi masalah dalam mengoptimalkan penempatan guru dengan mempertimbangkan jarak antara tempat tinggal guru dengan sekolah, khususnya di SDN Kabupaten Magelang. Hal ini penting karena, penempatan guru yang optimal dapat mendukung proses pembelajaran yang lebih efektif [10]. Penyelesaian masalah ini sangat sulit karena besarnya jumlah kombinasi yang mungkin terjadi antara guru dan sekolah serta, jumlah kombinasi ini akan meningkat, seiring dengan bertambahnya jumlah guru dan sekolah. Hal ini berakibat pada terciptanya ruang solusi yang sangat luas, karena masalah ini merupakan jenis masalah yang sulit dipecahkan dan membutuhkan waktu yang sangat lama untuk mencapai hasil yang diinginkan jika diselesaikan secara konvensional [3].

Oleh karena itu, diperlukan sebuah metode untuk menyelesaikan masalah ini. Dalam skripsi ini penulis menggunakan metode heuristik untuk mengatasi masalah ini, karena metode heuristik dianggap efektif untuk menyelesaikan masalah TSP (Travelling Salesman Problem) [8]. Algoritma ini mampu memberikan solusi yang mendekati optimal dan pastinya lebih cepat, dibandingkan dengan pencarian solusi secara menyeluruh [7].

Algoritma Hibrida Genetika Simulated Annealing (HGSA) menjadi pilihan yang menarik karena keduanya punya kelebihan yang bisa saling mendukung, seperti yang di tulis oleh Younis Elhaddad [28] dalam penlitianya berjudul “Combined Simulated Annealing And Genetic Algorithm To Solve Optimization Problems”, algoritma genetika memiliki kemampuan untuk menjelajahi banyak kemungkinan solusi. Namun, algoritma genetika cenderung mudah terjebak pada solusi yang belum optimal, yaitu solusi lokal (local minima). Di sisi lain, algoritma Simulated Annealing (SA) bekerja dengan memodifikasi satu solusi pada satu waktu, sehingga lebih fokus dalam memperbaiki solusi. Dengan demikian, GA bisa membantu pencarian menjadi lebih luas, sedangkan SA bisa mencegah GA berhenti di solusi yang kurang optimal. Selain itu menurut Smaili [4], SA juga dapat berperan penting dalam fase penyempurnaan (*refinement*), di mana SA bekerja untuk meningkatkan kualitas solusi dengan meningkatkan kromosom dengan fitness tertinggi dipopulasi, yang sebelumnya telah dibuat pada tahap GA. Oleh karena itu, metode Hibrida Genetika dan Simulated annealing dapat saling membantu untuk mendapatkan hasil yang lebih baik [11].

Penelitian dibidang ini yang cukup relevan dan menggunakan data yang sama pernah dilakukan oleh, H.Sriwindono [3] yang menggunakan metode K-Means Clustering dan algoritma Genetika dan menghasilkan jarak terbaik 9259,9 km, Laura Yohana Sinaga [12] yang menggabungkan algoritma Genetika dengan *insert local search* dan menghasilkan jarak terbaik 11119,7 km, Rosalina Winda Pratiwi [13], yang menggabungkan algoritma Genetika dengan beberapa metode local search dan mendapatkan jarak terbaik yaitu 11148,3 km dan Ni Komang Ayu Wirayanti [29] algoritma genetika yang digabungkan dengan metode local search yang menghasilkan jarak terpendek 10099 KM.

Berdasarkan hasil uraian tersebut, penulis tertarik untuk melakukan penelitian mengenai penerapan algoritma Hibrida Genetika dan Simulated Annealing (HGSA) dalam mengoptimalkan penempatan guru berdasarkan jarak total guru ke sekolah di SDN Kabupaten Magelang.

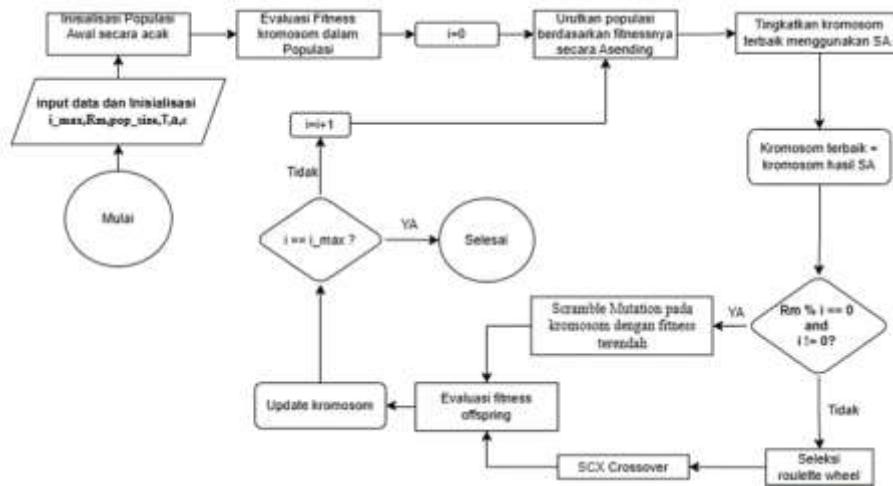
2. Metode

1.1. Deskripsi Data

Penelitian ini menggunakan data guru dan sekolah dasar di Kabupaten Magelang yang bersumber dari Dinas Pendidikan Pemuda dan Olahraga Kabupaten Magelang. Data guru memiliki atribut id guru, nama, jenis kelamin, serta lokasi rumah (*latitude* dan *longitude*), sedangkan data sekolah memiliki atribut id sekolah, nama, jumlah rombongan belajar, dan lokasi sekolah (*latitude* dan *longitude*). Terdapat 636 data guru dan 106 data sekolah sehingga diperoleh 67.416 data jarak, yang dihitung menggunakan rumus haversine untuk menentukan jarak rumah guru ke setiap sekolah.

1.2. Gambaran Umum Penerapan HGSA

Penelitian ini bertujuan menyelesaikan masalah penempatan guru di Kabupaten Magelang menggunakan algoritma hibrida genetika dan simulated annealing (HGSA) untuk meminimalkan total jarak penempatan. Diagram alir algoritma HGSA disusun berdasarkan penelitian Crossland [26], F. Smaili [4], dan Pakusadewa [27].



Gambar 1 Flowchart HGSA

Berikut langkah-langkah rancangan program penempatan guru menggunakan algoritma **HGSA**:

1. Inisialisasi Data dan Parameter

Memuat data jarak hasil algoritma Haversine dan menginisialisasi parameter HGSA seperti ukuran populasi, jumlah generasi, rasio mutasi, suhu awal, suhu akhir, dan laju penurunan suhu.

2. Pembentukan Populasi Awal

Membentuk populasi awal berupa kromosom yang merepresentasikan solusi penempatan guru ke sekolah secara acak, dengan ketentuan setiap sekolah memiliki 6 guru.

3. Evaluasi *Fitness* dan Pengurutan

Menghitung *fitness* setiap kromosom berdasarkan kebalikan dari total jarak penempatan, kemudian mengurutkan populasi berdasarkan nilai *fitness*.

4. Peningkatan dengan Simulated Annealing

Kromosom terbaik ditingkatkan menggunakan simulated annealing dengan operator swap hingga suhu mencapai batas minimum.

5. Seleksi Roulette Wheel

Memilih kromosom induk (*parent*) secara proporsional berdasarkan nilai *fitness* menggunakan metode roulette wheel.

6. Crossover

Parent terpilih mengalami proses crossover menggunakan metode Sequential Constructive Crossover (SCX) untuk menghasilkan *offspring*.

7. Mutasi

Jika kondisi mutasi terpenuhi, kromosom terburuk dimutasi menggunakan *scramble mutation*.

8. Penggantian Populasi

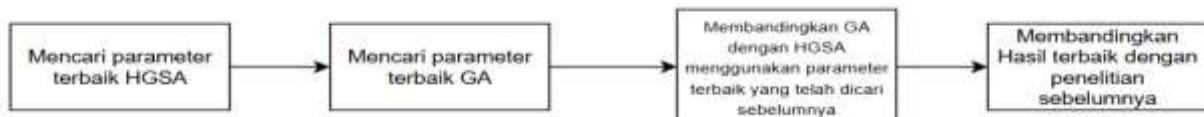
Offspring yang dihasilkan menggantikan kromosom dengan *fitness* terendah dalam populasi.

9. Penghentian Proses

Proses dihentikan ketika jumlah generasi maksimum tercapai.

1.3. Skenario Pengujian

Skenario pengujian ini bertujuan menguji pengaruh parameter algoritma genetika dan simulated annealing terhadap total jarak penempatan guru. Penyusunan skenario mengacu pada penelitian terdahulu dengan beberapa modifikasi oleh peneliti. Parameter dasar yang digunakan dalam pengujian ini adalah:



Gambar 2 Skenario Pengujian

3. Implementasi Algoritma Hibrida Genetika–Simulated Annealing (HGSA)

3.1 Perangkat yang Digunakan

Implementasi program dilakukan pada laptop dengan spesifikasi: prosesor Intel Core i5-10300H, RAM 16 GB, SSD 1 TB, sistem operasi Windows 11, dan bahasa pemrograman Python 3.11.9.

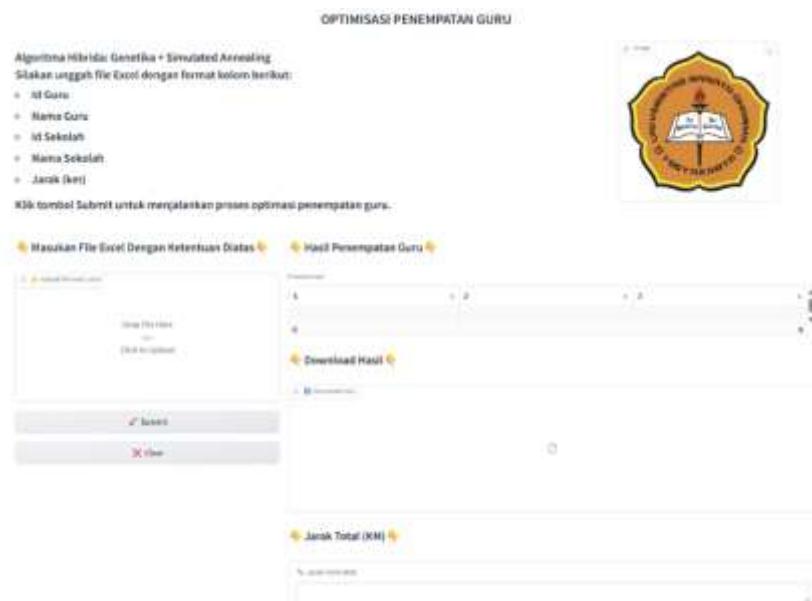
3.2 Implementasi Program

Tahap awal dilakukan preprocessing data menggunakan algoritma Haversine untuk menghitung jarak berdasarkan koordinat lintang dan bujur antara lokasi guru dan sekolah. Data koordinat diubah menjadi data jarak guru ke setiap sekolah.

```
import math
import pandas as pd
def haversine(lat1, lon1, lat2, lon2):
    # convertir coordenadas de grados
    lat1_rad = math.radians(lat1)
    lon1_rad = math.radians(lon1)
    lat2_rad = math.radians(lat2)
    lon2_rad = math.radians(lon2)
    # calcular distancia en radianes
    delta_lat = lat1_rad - lat2_rad
    delta_lon = lon2_rad - lon1_rad
    # fórmula de Haversine
    s = math.acos(delta_lat/2)**2 + math.cos(lat1_rad) * math.cos(lat2_rad) * math.sin(delta_lon/2)**2
    c = 2 * math.asin(math.sqrt(s))
    distancia = 6371 * c
    return distancia
guru_df = pd.read_excel('data_guru.xlsx')
sekolah_df = pd.read_excel('data_sekolah.xlsx')
results = []
for _, sekolah in sekolah_df.iterrows():
    for _, guru in guru_df.iterrows():
        jarak = haversine(guru['lat'], guru['long'], sekolah['lat'], sekolah['long'])
        results.append([
            'Id Guru', guru['Id Guru'],
            'Nama Guru', guru['Nama Guru'],
            'Id Sekolah', sekolah['Id'],
            'Nama Sekolah', sekolah['Nama'],
            'Jarak (km)', jarak
        ])
result_df = pd.DataFrame(results)
result_df.to_excel('output.xlsx', index=False)
```

Gambar 3 Code Algoritma Haversine

Implementasi algoritma Hibrida Genetika dan Simulated Annealing (HGSA) dimulai dengan memuat data jarak guru dan sekolah hasil *preprocessing*, kemudian menjalankan algoritma menggunakan bahasa pemrograman Python dengan bantuan library math, pandas, random, dan gradio. Seluruh parameter algoritma, seperti ukuran populasi, jumlah iterasi, rasio mutasi, serta parameter Simulated Annealing dikonfigurasikan dalam satu struktur untuk memudahkan pengujian. Data dibaca dari file Excel dan dipetakan ke dalam struktur yang memungkinkan perhitungan jarak secara efisien. Populasi awal dibentuk secara acak dengan kromosom unik yang merepresentasikan penempatan guru ke sekolah, kemudian dievaluasi menggunakan fungsi *fitness* berbasis total jarak. Kromosom terbaik ditingkatkan menggunakan Simulated Annealing, sementara proses seleksi dilakukan dengan metode roulette wheel, diikuti oleh crossover menggunakan Sequential Constructive Crossover (SCX) dan mutasi *scramble* untuk menjaga keberagaman solusi. Seluruh proses dikendalikan dalam satu alur utama hingga mencapai iterasi maksimum, dan solusi terbaik disimpan dalam bentuk file Excel. Untuk memudahkan penggunaan, sistem dilengkapi antarmuka berbasis Gradio yang memungkinkan pengguna memasukkan data, menjalankan algoritma, melihat hasil penempatan, serta mengunduh hasil secara langsung.



Gambar 4 Tampilan GUI

4. Pembahasan Hasil

4.1. Penentuan Parameter HGSA

4.1.1. Jumlah Generasi HGSA

Jumlah Iterasi	Tabel 1 Hasil pengujian Jumlah Generasi										
	100	200	300	400	500	600	700	800	900	1000	
Nomor pengujian	1	8172	7589	7380	6803	6882	6712	6356	6374	6362	6216
	2	8236	7412	7117	6877	6587	6657	6436	6398	6300	6271
	3	8332	7316	7193	6925	6616	6711	6403	6429	6357	6333
	4	8223	7438	7089	7003	6689	6627	6524	6290	6234	6228
	5	8220	7599	7175	6781	6770	6729	6368	6391	6231	6335
Mean		8256	7470	7190	6877	6708	6687	6417	6376	6296	6270

Pengujian ini mengevaluasi dampak jumlah generasi terhadap pencapaian Jarak total yang dihasilkan. Hasil dari pengujian jumlah generasi dapat diamati pada Tabel 1.



Gambar 5 Grafik Generasi

Dari Gambar 5, Generasi telah mencapai konvergen pada 1000 generasi, di mana peningkatan nilai *fitness* tidak signifikan, yang mengarah pada konvergensi di mana proses reproduksi menghasilkan *offspring* yang hampir sama dengan induknya (Mahmudy, 2013). Parameter yang dipakai adalah Jumlah populasi 50, Rasio Mutasi 1:10, Suhu Awal 100, suhu akhir 0,9, dan laju pendinginan 0,8.

4.1.2. Jumlah Populasi HGSA

Tabel 2 Analisa Jumlah Populasi

Jumlah Populasi	10	20	30	40	50	60	70	80	90	100	
Nomor pengujian	1	6525	6664	6383	6204	6158	6139	6202	6271	6147	6303
	2	6603	6516	6432	6240	6195	6244	6225	6134	6298	6273
	3	6534	6477	6495	6244	6217	6166	6150	6232	6248	6219
	4	6534	6330	6170	6324	6274	6235	6325	6283	6303	6242
	5	6435	6364	6188	6261	6335	6310	6203	6236	6237	6253
Mean		6526	6470	6333	6254	6235	6218	6221	6231	6246	6258

Pengujian ini bertujuan untuk mengetahui pengaruh jumlah populasi dalam mengoptimalkan jarak total. Hasil pengujian jumlah populasi dapat dilihat pada Tabel 2.



Gambar 6 Evaluasi Jumlah Populasi

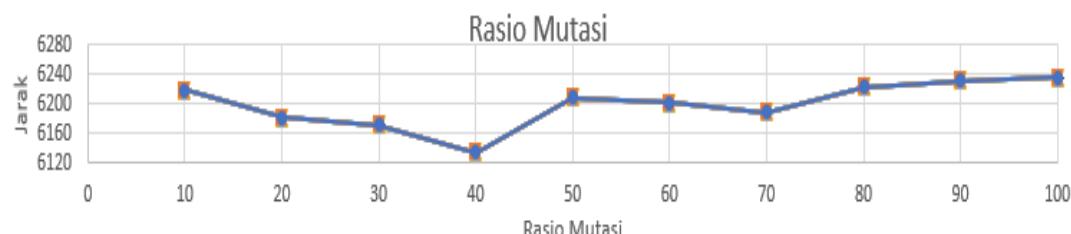
Berdasarkan Gambar 6, dapat dilihat bahwa populasi terbaik ada di 60, setelah populasi 60 terjadi sedikit peningkatan jarak total. Parameter yang dipakai adalah Jumlah Generasi 1000, Rasio Mutasi 1:10, Suhu Awal 100, suhu akhir 0,9, dan laju pendinginan 0,8

4.1.3. Rasio Mutasi HGSA

Tabel 3 Analisa Rasio Mutasi

Rasio Mutasi	1:10	1:20	1:30	1:40	1:50	1:60	1:70	1:80	1:90	1:100	
Nomor pengujian	1	6139	6060	5958	5952	6267	6321	6231	6303	6266	6276
	2	6244	6217	6310	6191	6233	6252	6317	6216	6231	6294
	3	6166	6264	6255	6159	6233	6064	6231	6107	6159	6185
	4	6235	6158	6139	6178	6160	6095	6105	6154	6271	6262
	5	6310	6207	6194	6191	6152	6276	6065	6332	6231	6167
Mean		6218	6181	6171	6134	6209	6201	6189	6222	6231	6236

Pengujian ini bertujuan untuk mengetahui pengaruh Rasio Mutasi dalam mengoptimalkan jarak total. Hasil pengujian jumlah populasi dapat dilihat pada **Tabel 7**.



Gambar 7 Analisa Rasio Mutasi

Pada **Gambar 5.3** terlihat bahwa total jarak mencapai nilai terendah pada rasio mutasi 1:40, dapat dilihat bahwa rata-rata nilai jarak total beragam pada nilai rasio mutasi yang berbeda. Parameter yang

di pakai Jumlah Generasi 1000, Jumlah populasi 60, Suhu Awal 100, suhu akhir 0,9, dan laju pendinginan 0,8.

4.1.4. Suhu Awal HGSA

Tabel 4 Analisa Suhu Awal

Suhu Awal	5	10	20	30	40	50	60	70	80	90	100
Nomor Pengujian	1	5269	5196	5366	5765	5882	6020	6133	6148	6272	6241
	2	5107	5212	5307	5660	5858	5771	6017	6067	6127	6189
	3	5224	5260	5288	5642	5965	6151	6083	6076	6127	6103
	4	5368	5242	5346	5550	5915	6047	5895	6162	6220	6277
	5	5186	5207	5329	5699	5882	5740	5903	6172	6100	6133
Mean	5230	5223	5327	5663	5900	5945	6006	6125	6169	6188	6217

Pengujian ini berfokus pada penentuan parameter *temperature* (t) yang optimal untuk mendapatkan jarak yang optimal. Hasil pengujian untuk parameter *temperature* (t) dapat dilihat pada Tabel 4.



Gambar 8 Grafik Suhu Awal

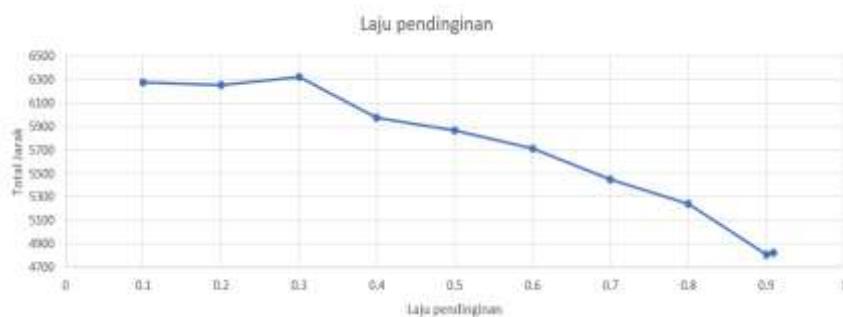
Pada **Gambar 5.4** terlihat bahwa total jarak terkecil tercapai pada suhu awal 10°C, kemudian semakin naik seiring peningkatan suhu awal selain itu, total jarak juga sedikit meningkat pada suhu awal 5°C. Parameter yang di pakai Jumlah Generasi 1000, Jumlah populasi 60, Rasio Mutasi 40, suhu akhir 0,9, dan laju pendinginan 0,8.

4.1.5. Pendinginan HGSA

Tabel 5 Hasil pengujian laju pendinginan

Laju Pendinginan	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	0.91
Nomor Pengujian	1	6222	6269	6362	5975	5911	5715	5379	5196	4766
	2	6416	6152	6327	5951	5830	5748	5464	5212	4886
	3	6358	6312	6254	6080	5796	5777	5482	5260	4818
	4	6258	6376	6251	5970	5927	5771	5514	5242	4816
	5	6113	6159	6306	5901	5863	5546	5416	5307	4754
Mean	6273	6253	6320	5975	5865	5711	5451	5243	4808	4827

Pengujian ini berfokus pada penentuan parameter laju pendinginan yang terbaik untuk mendapatkan jarak yang optimal. Hasil pengujian untuk parameter laju pendinginan dapat dilihat pada Tabel 5.



Gambar 9 Grafik Laju Pendinginan

Pada grafik di gambar 9 terlihat bahwa ketika laju pendinginan 0,1 sampai 0,3, total jarak relatif tinggi, sedangkan semakin mendekati 0,9 total jarak menurun.. Parameter yang di pakai Jumlah Generasi 1000, Jumlah populasi 60, Rasio Mutasi 40, suhu akhir 0,9, Suhu Awal 10, dan laju pendinginan 0,8.

4.1.6. Suhu Akhir HGSA

Tabel 6 Hasil Pengujian Suhu Akhir

Suhu Akhir	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	
Nomor Pengujian	1	4433	4512	4631	4644	4667	4646	4669	4761	4766
	2	4549	4509	4608	4614	4644	4754	4813	4705	4886
	3	4551	4451	4587	4583	4673	4690	4707	4821	4818
	4	4419	4541	4690	4610	4659	4755	4721	4751	4816
	5	4549	4535	4477	4599	4701	4797	4804	4791	4854
Mean	4500	4509	4598	4610	4668	4728	4742	4765	4828	

Pengujian ini berfokus pada menguji parameter suhu akhir yang terbaik untuk mendapatkan jarak yang optimal. Hasil pengujian untuk parameter laju suhu akhir dapat dilihat pada Tabel 6.



Gambar 10 Grafik suhu akhir

Berdasarkan grafik pada Gambar 10, total jarak terbaik dicapai ketika suhu akhir berada pada 0,1°C. Hal ini menunjukkan bahwa suhu akhir 0,1°C membuat algoritma menjadi lebih selektif dalam menerima solusi, sehingga hasil akhir yang diperoleh cenderung lebih optimal. Sebaliknya, apabila suhu akhir terlalu tinggi, peluang untuk menerima solusi dengan jarak yang lebih besar masih relatif tinggi. Parameter yang di pakai Jumlah Generasi 1000, Jumlah populasi 60, Rasio Mutasi 40, Suhu Awal 10°C, dan laju pendinginan 0,9.

Berdasarkan hasil penentuan parameter, diperoleh konfigurasi untuk algoritma HGSA sebagai berikut: Jumlah Generasi 1000, Jumlah populasi 60, Rasio Mutasi 40, Suhu Awal 10°C, dan laju pendinginan 0,9, Suhu Akhir 0,1.

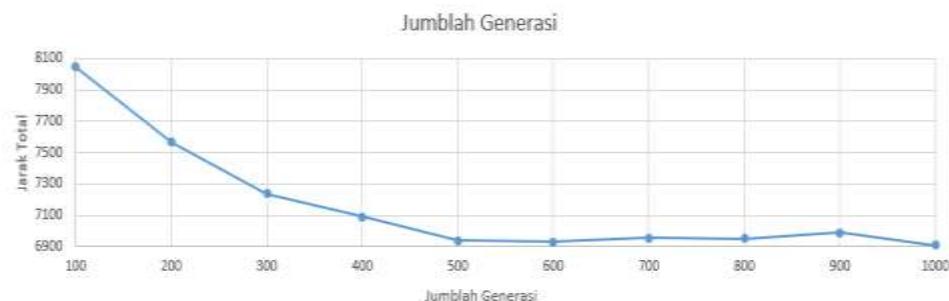
4.2. Penentuan Parameter Genetika Murni

4.2.1 Jumlah Generasi GA

Tabel 7 Hasil pengujian Jumlah Generasi genetika murni

Jumlah Iterasi	100	200	300	400	500	600	700	800	900	1000
Nomor Pengujian	1	8460	7551	7612	7064	7037	7017	7088	6982	6980
	2	8066	7660	7156	6788	6914	6914	6905	6828	7055
	3	8182	7553	6944	7108	7065	7065	7022	7030	6909
	4	8353	7432	7158	7149	6995	6985	7002	6996	6874
	5	8186	7644	7324	7355	6678	6678	6758	6925	7131
Mean		8049	7568	7238	7092	6937	6931	6955	6952	6907

Pengujian ini bertujuan untuk menentukan jumlah generasi yang optimal pada algoritma genetika murni untuk memperoleh jarak terbaik. Hasil pengujian terhadap parameter jumlah generasi pada algoritma genetika murni dapat dilihat pada Tabel 7.



Gambar 11 Grafik iterasi genetika murni

Berdasarkan Gambar 11, dapat dilihat bahwa pada jumlah generasi antara 100 hingga 500, terjadi perubahan total jarak yang cukup signifikan pada setiap percobaan. Sementara itu, mulai dari generasi ke-500 hingga 1000, nilai solusi mulai konvergen, ditandai dengan perubahan yang tidak signifikan. Jika jumlah generasi terlalu sedikit, ruang pencarian solusi menjadi sempit sehingga hasil yang diperoleh kurang optimal. Sebaliknya, jika jumlah generasi terlalu banyak, penurunan total jarak menjadi sangat kecil dan prosesnya cenderung mengalami konvergensi, yaitu ketika individu-individu baru yang dihasilkan hampir sama dengan induknya (Mahmudy, 2013). Pada percobaan ini, parameter yang digunakan adalah jumlah populasi sebanyak 50 dan rasio mutasi 1:10.

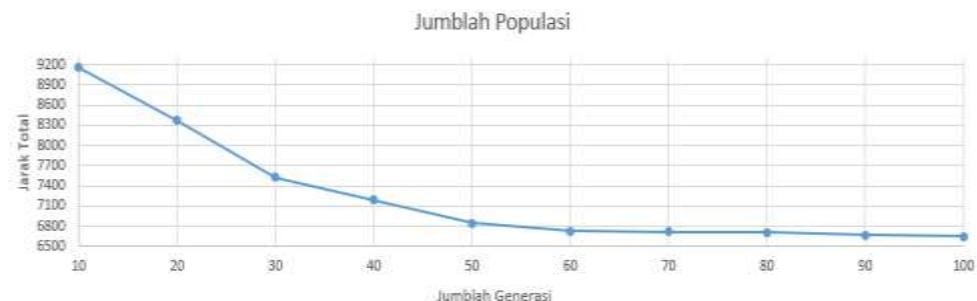
4.2.2 Jumlah Populasi GA

Tabel 8 Analisis Jumlah Populasi genetika murni

Jumlah Populasi	10	20	30	40	50	60	70	80	90	100
Nomor Pengujian	1	9275	8548	8047	7278	6902	6799	6778	6641	6686
	2	8993	8200	7544	7013	7018	6527	6624	6777	6578
	3	9142	8048	7376	7338	6814	6522	6736	6664	6585
	4	8792	8383	7516	7239	6966	6925	6781	6753	6774
	5	9497	8646	7141	7102	6545	6864	6691	6716	6698
Mean		9159	8365	7524	7194	6849	6727	6722	6710	6664
										6647

Pengujian ini dilakukan untuk mengetahui sejauh mana variasi jumlah populasi dapat memengaruhi hasil optimasi jarak total pada algoritma GA. Melalui pengujian ini, diharapkan dapat diketahui berapa

jumlah populasi yang optimal untuk mendapatkan hasil jarak total yang optimal. Selengkapnya, hasil pengujian terkait jumlah populasi dapat dilihat pada Tabel 8.



Gambar 12 Grafik jumlah populasi genetika murni

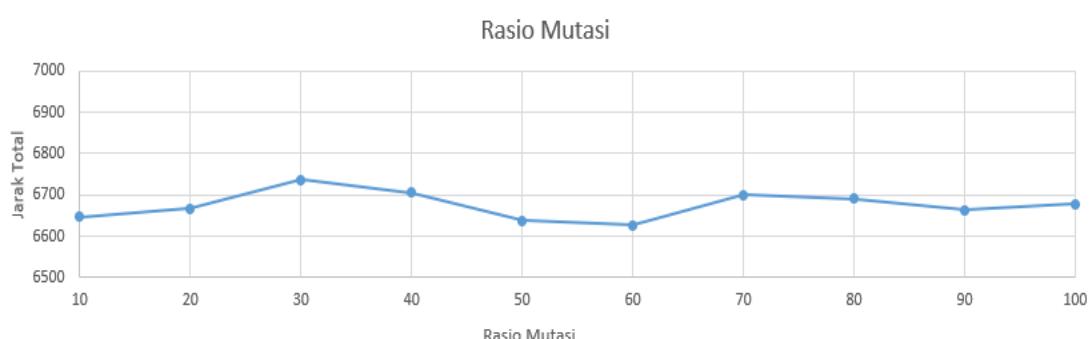
Berdasarkan Gambar 12, dapat dilihat bahwa jumlah populasi 10 hingga 60 memberikan pengaruh yang besar terhadap nilai *fitness* yang dihasilkan pada setiap percobaan. Hal ini terlihat dari rata-rata Jarak total yang terus turun pada variasi jumlah populasi di rentang tersebut. Namun, pada jumlah populasi 60 hingga 100, proses mulai mengalami konvergensi, yang ditandai dengan perubahan nilai *fitness* yang semakin kecil atau hampir tidak ada perubahan berarti. Temuan ini sesuai dengan rekomendasi dari [De Jong \(1975\)](#) yang menyarankan kisaran jumlah populasi ideal antara 50 hingga 100. Pada pengujian ini, parameter yang digunakan adalah jumlah generasi sebanyak 1000, rasio mutasi 1:10.

4.2.3 Rasio Mutasi GA

Tabel 9 Analisa Rasio Mutasi

Rasio Mutasi	1:10	1:20	1:30	1:40	1:50	1:60	1:70	1:80	1:90	1:100	
Nomor Pengujian	1	6694	6751	6748	6747	6617	6503	6694	6784	6410	6753
	2	6537	6640	6707	6672	6634	6608	6764	6554	6726	6503
	3	6784	6751	6848	6704	6609	6588	6881	6733	6692	6728
	4	6682	6749	6618	6720	6659	6737	6603	6733	6738	6630
	5	6540	6448	6759	6682	6674	6699	6559	6653	6754	6577
	Mean	6647	6667	6736	6705	6638	6627	6700	6691	6664	6678

Pengujian ini bertujuan untuk mengetahui pengaruh rasio mutasi terhadap jarak total. Hasil pengujian terkait variasi rasio mutasi dapat dilihat pada Tabel 9.



Gambar 13 Grafik Rasio Mutasi

Pada Gambar 13 terlihat bahwa total jarak mencapai nilai terendah pada rasio mutasi 1:40, dapat dilihat bahwa rata-rata nilai jarak total beragam pada nilai rasio mutasi yang berbeda. Hal ini

menunjukkan bahwa pencarian solusi paling optimal tercapai pada tingkat mutasi 1:40 yang cukup tinggi untuk mencegah algoritma terjebak di solusi lokal, tetapi tidak terlalu tinggi hingga merusak struktur solusi yang baik. Parameter yang di pakai Jumlah Generasi 1000, Jumlah populasi 100.

Berdasarkan hasil penentuan parameter, diperoleh konfigurasi untuk algoritma GA sebagai berikut: Jumlah Generasi 1000, Jumlah populasi 100, dan rasio mutasi 1:60.

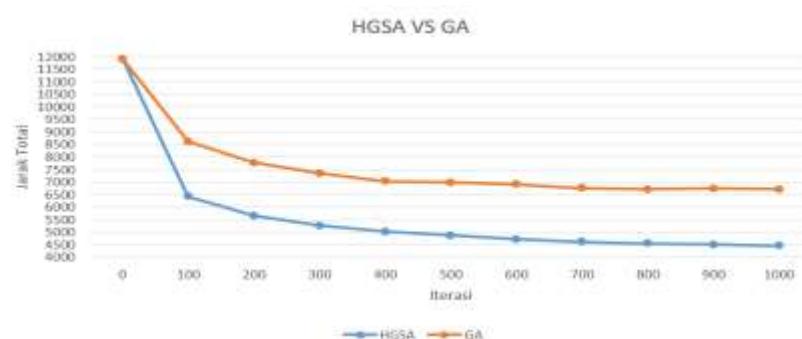
4.3 Perbandingan HGSA dan GA murni

Tabel 10 Perbandingan HGSA dan GA murni

Algoritma	Iterasi	Percobaan ke					Mean
		1	2	3	4	5	
HGSA	100	6471	6467	6398	6426	6372	6426
GA	100	8731	8612	8482	8442	8847	8622
HGSA	200	5680	5709	5556	5693	5619	5651
GA	200	7589	7831	7595	7981	7818	7762
HGSA	300	5298	5215	5261	5208	5278	5252
GA	300	7315	7287	7270	7484	7402	7351
HGSA	400	5054	5016	4942	5139	4914	5013
GA	400	7033	6934	7163	6949	7052	7026
HGSA	500	4837	4851	4849	4908	4902	4869
GA	500	6979	7011	6954	6939	6993	6975
HGSA	600	4737	4699	4660	4757	4737	4718
GA	600	7059	6671	6901	6975	6928	6906
HGSA	700	4634	4615	4558	4683	4595	4617
GA	700	6743	6819	6809	6735	6709	6763
HGSA	800	4608	4584	4561	4459	4523	4547
GA	800	6706	6765	6710	6627	6728	6707
HGSA	900	4500	4481	4544	4492	4520	4507
GA	900	6686	6715	6776	6764	6761	6740
HGSA	1000	4417	4466	4448	4472	4458	4452
GA	1000	6587	6763	6825	6715	6669	6711

Tabel 10 adalah hasil pengujian antara algoritma Hybrid Genetic Simulated Annealing (HGSA) dan algoritma *Genetic Algorithm* (GA) murni. Pengujian ini bertujuan untuk membandingkan performa *Genetic Algorithm* (GA) murni dengan *Genetic Algorithm* (GA) yang digabungkan dengan Simulated Annealing (SA), dalam hal total jarak yang dihasilkan oleh kedua algoritma tersebut dari iterasi 100 sampai 1000. Melalui pengujian ini, dapat diamati seberapa besar peningkatan kualitas solusi yang diperoleh ketika SA diintegrasikan ke dalam proses evolusi GA.

Untuk parameter yang digunakan yaitu, untuk GA parameter yang dipakai Jumlah populasi 100, dan rasio mutasi 60. Sedangkan untuk HGSA parameter yang di pakai adalah Jumlah populasi 60, Rasio Mutasi 40, Suhu Awal 10, dan laju pendinginan 0,1.



Gambar 14 Grafik HGSA vs GA

Berdasarkan grafik pada Gambar 14, terlihat bahwa HGSA memberikan hasil yang lebih optimal dibandingkan *Genetic Algorithm* murni dalam meminimasi total jarak. Pada iterasi awal (100 iterasi), HGSA telah menunjukkan perbaikan signifikan dengan rata-rata jarak sebesar 6.426 KM dibandingkan GA murni sebesar 8.622KM.

Selisih antara HGSA dan GA murni, menunjukkan bahwa HGSA dapat menghindari jebakan solusi lokal (local optima), yang sering menjadi kelemahan GA murni. Persentase peningkatan performa HGSA dibandingkan GA meningkat dari 25,5% di iterasi 100 hingga 33,7% di iterasi 1000. Hal ini menegaskan bahwa metode hybrid secara konsisten lebih baik dalam menemukan solusi optimal.

5. Kesimpulan

Berdasarkan hasil penelitian, parameter terbaik algoritma HGSA untuk menghasilkan rata-rata jarak terpendek (± 4452 km) adalah populasi 60 kromosom, rasio mutasi:crossover 1:40, suhu awal 10°C, suhu akhir 0,1°C, laju penurunan suhu 0,9, dan jumlah iterasi 1000. Hasil pengujian menunjukkan bahwa total jarak cenderung menurun seiring peningkatan jumlah populasi, rasio mutasi, dan laju pendinginan, serta penurunan suhu awal, dengan konvergensi terbaik terjadi pada iterasi 1000 dan suhu akhir 0,1°C. Selain itu, perbandingan kinerja menunjukkan bahwa algoritma HGSA secara konsisten menghasilkan total jarak yang lebih kecil dibandingkan algoritma genetika murni (GA), sehingga HGSA terbukti lebih efektif untuk menyelesaikan permasalahan penempatan guru di Kabupaten Magelang.

Referensi

- [1] A. Lambora, K. Gupta, and K. Chopra, "Genetika Algorithm—A Literature Review," in *2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS)*, Greater Noida, India, 2019, pp. 1–5. doi: 10.1109/comitcon.2019.8862255.
- [2] D. A. Prasetya, P. T. Nguyen, R. Faizullin, I. Iswanto, and E. F. Armay, "Resolving the shortest path problem using the haversine algorithm," *Journal of Critical Reviews*, vol. 7, no. 1, pp. 62–64, 2020. doi: 10.22159/jcr.07.01.11.
- [3] H. Sriwindono, P. P. Rosa, and K. Pinaryanto, "Teacher Placement using K-Means Clustering and Genetika Algorithm," *Conference Series*, vol. 4, pp. 43–51, 2022. doi: 10.34306/conferenceseries.v4i1.669.
- [4] F. Smaili, "A hybrid genetic-simulated annealing algorithm for multiple traveling salesman problems," *Decision Science Letters*, vol. 13, pp. 721–728, 2024.
- [5] S. Mirjalili, "Evolutionary Algorithms and Neural Networks: Theory and Applications," in *Studies in Computational Intelligence*, vol. 780, Springer, 2018.
- [6] S. Suyanto, *Artificial Intelligence: Searching, Reasoning, Planning, dan Learning*, Edisi 3. Bandung, Indonesia: Informatika, 2021.
- [7] I. A. Soenandi, Y. Juan, and M. Budi, "Optimization for routing vehicles of seafood product transportation," *IOP Conference Series: Materials Science and Engineering*, vol. 277, no. 1, p. 012048, 2017. doi: 10.1088/1757-899x/277/1/012048.
- [8] M. S. N. Afif, M. I. A. Tsauri, and S. Hadiwijaya, "Optimisasi Rute Pengiriman Produk Komponen Otomotif (Traveling Salesman Problem) Melalui Pendekatan Heuristik," *Jurnal Teknik Industri*, vol. 3, no. 1, pp. 38–46, 2022. doi: 10.37366/jutin0301.3846.
- [9] A. Yuristia, "Pendidikan sebagai transformasi kebudayaan," *Ijtima'iyah: Jurnal Ilmu Sosial dan Budaya*, vol. 2, no. 1, pp. 1–12, 2018.
- [10] D. N. Amalia, D. H. Lestari, S. Trihantoyo, and Nuphanudin, "Peran penting penempatan guru dalam mencapai tujuan pembelajaran di SMK Negeri 4 Surabaya," *Jurnal Penelitian Ilmu Pendidikan Indonesia*, vol. 3, no. 1, pp. 169–175, 2024. doi: 10.37366/jpi.v3i1.169.
- [11] Fatyanosa, T.N., Sihananto, A.N., Alfarisy, G.A.F., Burhan, M.S. and Mahmudy, W.F., 2017. Hibrida Genetika Algorithm and Simulated Annealing for Function Optimization. *Journal of Information Technology and Computer Science*, 1(2), p.82.
- [12] L. Y. Sinaga, "Penggunaan Algoritma Genetika Hibrida untuk Penempatan Guru", Skripsi, Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma, Yogyakarta, 2022.

- [13] R. W. Pratiwi, "Optimasi Penempatan Guru Menggunakan Algoritma Genetika Hibrida", Skripsi, Program Studi Informatika, Fakultas Sains dan Teknologi, Universitas Sanata Dharma, Yogyakarta, 2023.
- [14] J. H. Holland, "Genetika Algorithms," *Scientific American*, vol. 267, no. 1, pp. 66–72, Jul. 1992, doi: 10.1038/scientificamerican0792-66.
- [15] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of State Calculations by Fast Computing Machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087-1092, 1953.
- [16] D. Chen, C.-Y. Lee and C. H. Park, "Hibrida Genetika algorithm and simulated annealing (HGSA) in global function optimization," in 17th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'05), Hong Kong, China, 2005, pp. 5-133. doi: 10.1109/ICTAI.2005.72