

# Implementasi Klasterisasi K-Means pada *Master-Slave Genetic Algorithm* untuk Menurunkan Waktu Komputasi: Studi Kasus Optimasi Mesin CNC Batik Tulis

Hafidz Ridho<sup>1</sup>, Andi Sudiarso<sup>2</sup>, dan Ren J. Kuo<sup>3</sup>

<sup>1</sup>Departemen Teknik Sistem dan Industri, Institut Teknologi Sepuluh Nopember (ITS)

<sup>2</sup>Departemen Teknik Mesin dan Industri, Universitas Gadjah Mada (UGM)

<sup>3</sup>Department of Industrial Management, National Taiwan University of Science and Technology (NTUST)  
e-mail: hafidz.ridho@its.ac.id

**Abstrak**—Mayoritas penelitian mengenai otomatisasi proses pematikan menggunakan mesin Computer Numerical Control (CNC) fokus pada aspek kualitas. Tantangan selanjutnya adalah menentukan bagaimana cara menurunkan waktu permesinan. Penelitian sebelumnya telah berhasil menurunkan waktu permesinan dengan mengoptimalkan urutan pematikan menggunakan *Master-Slave Genetic Algorithm* sehingga dapat meminimalkan total jarak tempuh pematikan. Namun, penelitian tersebut masih memiliki kekurangan yaitu proses komputasi yang lama, yang dikarenakan banyaknya *instances* atau jumlah segmen batik yang harus dioptimasi. Penelitian ini bertujuan untuk menurunkan waktu komputasi dengan teknik klasterisasi K-means, sehingga jumlah segmen yang dioptimasi pada setiap klaster menjadi lebih sederhana dan proses komputasi menjadi lebih efisien. Hasil penelitian menunjukkan bahwa teknik klasterisasi K-means dapat menurunkan waktu komputasi secara signifikan dengan hasil yang lebih baik.

**Kata Kunci**—Genetic Algorithm, K-Means, Klasterisasi, Mesin CNC Batik, Optimasi.

## I. PENDAHULUAN

BATIK merupakan salah satu ikon budaya Indonesia yang sudah ditetapkan sebagai warisan kemanusiaan untuk budaya lisan dan budaya takbenda oleh United Nations Educational Scientific and Cultural Organization (UNESCO) sejak 2 Oktober 2009. Meski demikian, masih banyak masalah yang dihadapi oleh industri batik nasional, salah satunya adalah rendahnya regenerasi perajin batik tradisional terutama batik tulis [1]. Beberapa penelitian sudah dilakukan untuk mengatasi permasalahan tersebut, diantaranya adalah otomatisasi proses produksi menggunakan mesin Computer Numerical Control (CNC) untuk batik cap [2], [3], [4] dan batik tulis [1]. Berbagai penelitian juga telah dilakukan untuk mengoptimalkan proses produksi batik menggunakan mesin CNC, diantaranya adalah optimasi suhu dan *feed rate* [5], optimasi *feed rate*, jenis malam, dan ukuran *nozzle* [6], dan optimasi rute pematikan menggunakan metaheuristik [7].

Topik yang diangkat oleh mayoritas penelitian mengenai mesin batik adalah tentang kualitas, dimana mesin batik diharapkan mampu untuk memproduksi kain batik dengan kualitas yang sama dengan kain batik buatan perajin tradisional [1]-[6]. Tantangan selanjutnya adalah mempercepat proses pematikan. Berdasarkan studi literatur yang sudah dilakukan, belum banyak penelitian yang membahas mengenai peningkatan produktivitas mesin batik. Rujukan [7] telah berhasil memodelkan proses pematikan

motif Megamendung sebagai *open arc routing problem* dan telah berhasil mengoptimalkan rute pematikan menggunakan *master-slave genetic algorithm* (MS-GA) [8] sehingga dapat menurunkan waktu produksi. Namun, penelitian tersebut masih memiliki kelemahan, yaitu waktu komputasi yang lama. Terdapat 466 segmen pada motif batik yang dioptimasi, dimana setiap segmen memiliki dua ujung sebagai *entry* dan *exit point*, sehingga terdapat sejumlah  $(2 \times 466)!$  alternatif solusi yang harus dievaluasi. Hal ini meningkatkan kompleksitas dan menyebabkan waktu komputasi menjadi lama. Berdasarkan tantangan tersebut, maka penelitian ini dilakukan untuk menurunkan waktu komputasi dari [7].

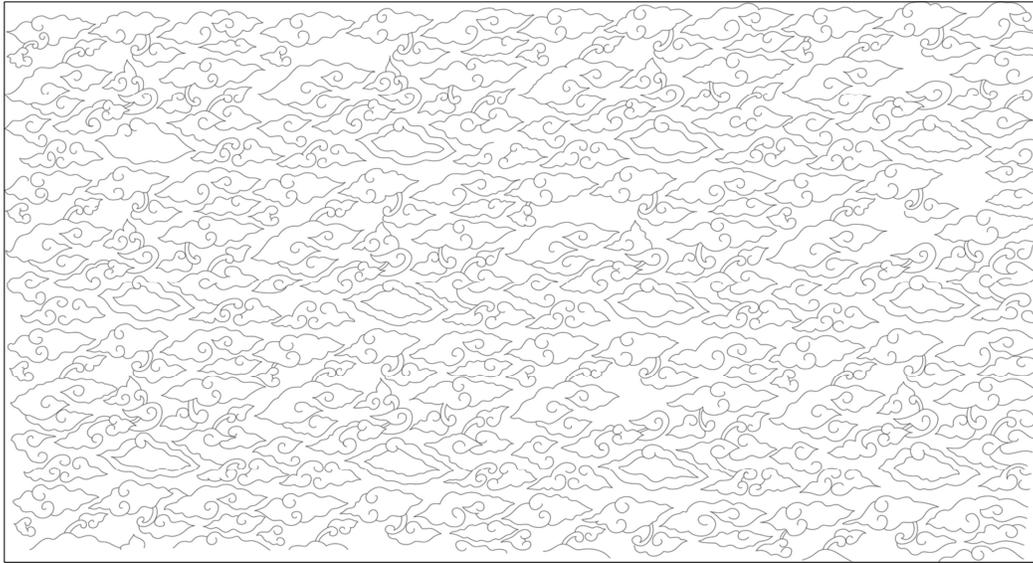
Hasil evaluasi [7] menunjukkan bahwa kontribusi terbesar penyebab lamanya waktu komputasi adalah pada perhitungan fungsi tujuan. MS-GA membutuhkan waktu komputasi yang lebih lama karena pada setiap iterasi terdapat proses *mating* yang menghasilkan generasi baru, sehingga jumlah individu yang harus dievaluasi menjadi semakin banyak. Sedangkan pada algoritma pembandingan, yaitu *Particle Swarm Optimization* (PSO) dan *Sine-Cosine Algorithm* (SCA), jumlah individu yang dievaluasi pada setiap iterasi jumlahnya tetap. Metode yang digunakan dalam penelitian ini untuk menyederhanakan dan mengurangi beban perhitungan fungsi tujuan adalah klasterisasi. Dengan membagi *dataset* menjadi beberapa klaster, maka jumlah segmen yang dioptimasi pada setiap klaster akan berkurang sehingga waktu komputasi menjadi lebih cepat.

Terdapat beberapa teknik klasterisasi yang dapat digunakan, diantaranya adalah K-means [9], DBSCAN [10], *Decision Tree* [11], dan *Self-Organizing Maps* (SOM) [12]. Dari metode klasterisasi tersebut, K-means dianggap paling sesuai karena masalah rute pematikan merupakan permasalahan dua dimensi, sehingga teknik *euclidean distance* bisa langsung diterapkan untuk mengevaluasi jarak antar klaster dan jarak intra klaster.

## II. METODE PENELITIAN

### A. Dataset

Penelitian ini menggunakan *dataset* yang sama dengan penelitian sebelumnya, yaitu motif batik Megamendung [7]. Ilustrasi dari motif batik yang akan dioptimasi ditunjukkan oleh Gambar 1. Koordinat dari setiap titik didapatkan dari G-Code yang dihasilkan oleh *software* Mach3.

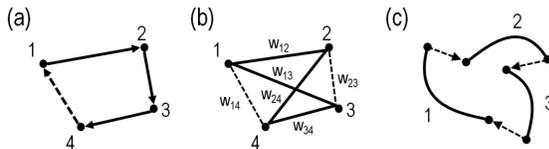


Gambar 1. Dataset motif batik Megamendung [7].

Proses komputasi dan optimasi dilakukan menggunakan bahasa pemrograman Python pada komputer dengan Intel® Core™ i7-11370H CPU @3,30GHz (8CPUs) dan 8.192 MB RAM.

**B. Model Matematis**

*Open arc routing problem* merupakan permasalahan optimasi rute dimana setiap rute segmen memiliki *entry point* dan *exit point* yang berbeda [8]. Gambar 2 menunjukkan perbedaan antara *travelling salesman problem* (TSP), *capacitated arc routing problem*, dan *open arc routing problem*.



Gambar 2. Perbandingan *routing problem* [7]. (a) TSP. (b) *Capacitated arc routing problem*. (c) *Open arc routing problem*

Model matematis pada penelitian ini menggunakan formulasi yang sama dengan [7], dimana fungsi tujuan dari permasalahan ini, seperti yang dijelaskan oleh (1), adalah meminimalkan total jarak tempuh (*TL*) yang terdiri dari jarak tempuh segmen utama (*TS*) dan jarak tempuh antar segmen (*TC*). *TS* merupakan pola batik yang akan ditulis dan tidak bisa diminimalkan, sehingga fungsi tujuan dapat disederhanakan seperti pada (2),

$$TL = TS + TC \tag{1}$$

$$\min TC = \sum_{i,j \in S} x_{ij} d_{x_{ik}jl} \tag{2}$$

dimana variabel keputusannya adalah menentukan nilai dari  $x_{ikjl}$ , yaitu keputusan pemilihan rute dari segmen  $i$  yang keluar dari titik  $k$  menuju segmen  $j$  yang masuk dari titik  $l$ . Jarak antar titik  $d_{x_{ikjl}}$  dihitung menggunakan prinsip *euclidean distance*, yang dijelaskan oleh (3),

$$d = \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2} \tag{3}$$

dimana  $x_i$  dan  $y_i$  adalah koordinat titik terakhir pada segmen asal, sedangkan  $x_{i+1}$  dan  $y_{i+1}$  adalah koordinat titik pertama pada segmen tujuan. Batasan dari model ini dijelaskan sebagai berikut:

$$\sum_{i=1}^n x_{ij} \leq 1 \quad j = 1, \dots, n \tag{4}$$

$$\sum_{j=1}^n x_{ij} \leq 1 \quad i = 1, \dots, n \tag{5}$$

$$\sum_{i=1}^n x_{im} + \sum_{j=1}^n x_{mj} = 1 \quad m = 1, \dots, n \tag{6}$$

$$u_i - u_j + nx_{ij} \leq n - 1 \quad i, j = 2, \dots, n \tag{7}$$

$$x_{ij} \in \{0,1\} \tag{8}$$

Konstrain (4) dan (5) menjelaskan bahwa setiap titik bisa dikunjungi maksimal satu kali. Konstrain (6) memaksa variabel keputusan untuk memilih titik masuk dan titik keluar yang berbeda pada setiap segmen, untuk urutan segmen berikutnya seperti yang dijelaskan pada (2). Untuk setiap segmen utama  $S$ , jika titik  $k$  sudah dipilih sebagai titik masuk, maka titik  $l$  otomatis menjadi titik keluar untuk urutan selanjutnya dan titik  $k$  tidak dapat dipilih kembali. Konstrain (7) menjelaskan bahwa solusi yang didapatkan hanya memiliki satu siklus yang mencakup semua segmen.

**C. Algoritma Penyelesaian**

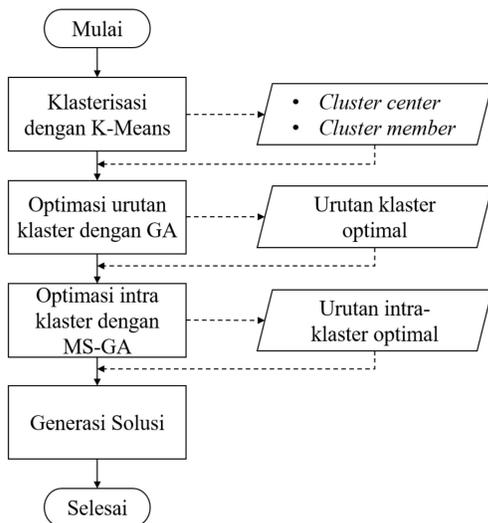
Secara umum, *model pipeline* dari algoritma penyelesaian yang digunakan pada penelitian ini dijelaskan oleh Gambar 3.

Teknik klusterisasi digunakan untuk membagi dataset menjadi beberapa klaster, sehingga beban perhitungan dapat berkurang karena titik yang diperhitungkan tidak terlalu banyak. Algoritma K-means dijelaskan sebagai berikut:

**Algoritma 1.** K-means

```

1: t = 0
2: Initialize cluster center
3: Set number of cluster (k)
4: While not termination do:
5:   For each k do:
6:     Calculate jarak dari data point ke cluster center k
7:     Generate matriks jarak
8:   end for
9:   For each data point do:
10:    Assign data point ke cluster k ← jarak minimum
11:  end for
12:  For each k do:
13:    Calculate new cluster center ← rata-rata anggota cluster
14:  t = t + 1
15: End while
16: Output cluster center dan anggota cluster
    
```



Gambar 3. Model pipeline algoritma penyelesaian.

Hasil dari K-means adalah pengelompokan segmen berdasarkan posisi yang terdekat. Setiap kluster memiliki titik berat yang merupakan *center* dari kluster tersebut. *Cluster center* tersebut selanjutnya dioptimasi menggunakan algoritma genetika (GA) untuk menentukan urutan kluster pembatikan yang optimal. Algoritma GA dijelaskan sebagai berikut:

**Algoritma 2.** Genetic algorithm

```

1: t = 0
2: Inisialisasi populasi (P)
3: Evaluasi populasi (P)
4: While not termination do:
5:   Generate offspring (P') ← kawin silang & mutasi
6:   Evaluasi offspring (P')
7:   Select best offspring (Pt+1)
8:   t = t + 1
9: End while
10: Output solusi terbaik
    
```

Setelah didapatkan urutan kluster yang optimal, anggota segmen dari setiap kluster dioptimasi menggunakan MS-GA untuk menentukan urutan segmen pembatikan yang optimal. Optimasi intra kluster dilakukan secara bertahap sesuai dengan urutan kluster yang optimal. Perbedaan antara MS-GA dengan GA konvensional terletak pada

kromosomnya, dimana pada MS-GA setiap individu memiliki dua set kromosom. Kromosom *master* mewakili identitas segmen utama, sedangkan kromosom *slave* mewakili *entry point* dari segmen tersebut [8]. Ilustrasi dari kromosom MS-GA dijelaskan oleh Gambar 4.

Master chromosome (main segment)	3	4	2	7	1	6	5	8
Slave chromosome (entry point)	0	1	1	0	0	1	0	1

Gambar 4. Kromosom pada MS-GA [8]

Perbedaan kromosom pada MS-GA tidak berpengaruh pada skema perkawinan silang GA konvensional. Skema perkawinan silang yang digunakan dalam penelitian ini adalah *single-point partially mapped crossover* [13], dimana titik potong yang membagi rangkaian kromosom menjadi dua bagian ditentukan menggunakan bilangan random. Ilustrasi skema perkawinan silang pada MS-GA dijelaskan oleh Gambar 5.

Parent1-master	3	4	2	5	1
Parent1-slave	0	1	1	0	0
Parent2-master	1	2	3	4	5
Parent2-slave	1	0	1	1	0

↓

Offspring1-master	3	4	2	1	5
Offspring1-slave	0	1	1	1	0
Offspring2-master	1	2	3	5	4
Offspring2-slave	1	0	1	0	0

Gambar 5. Single-point partially mapped crossover [13]

Sedangkan pada mutasi, karena setiap individu memiliki dua set kromosom, mutasi dilakukan melalui dua tahap [7]. Pada mutasi tahap pertama untuk setiap individu, dua gen *master* dan *slave* yang ditentukan secara random ditukar posisinya. Sedangkan pada mutasi tahap kedua, sebuah gen *slave* yang posisinya ditentukan secara random dirubah menjadi nilai yang berlawanan. Ilustrasi dari mutasi dua tahap tersebut dijelaskan oleh Gambar 6.

(a)

Parent1-master	3	4	2	1	5
Parent1-slave	0	1	1	1	0

↓

Offspring1.1-master	3	5	1	1	4
Offspring1.1-slave	0	0	1	1	1

(b)

Parent1-master	3	4	2	1	5
Parent1-slave	0	1	1	1	0

↓

Offspring1.1-master	3	5	1	1	4
Offspring1.1-slave	0	0	1	1	1

Gambar 6. Mutasi dua tahap [7]. (a) Pertukaran posisi gen dalam satu individu, (b) Mutasi pada gen *slave*.

Tahap terakhir dilakukan generasi solusi untuk mendapatkan urutan keseluruhan segmen yang optimal. Solusi optimal intra kluster pada setiap kluster dikombinasikan sesuai dengan urutan optimal dari solusi inter kluster. Dengan demikian, solusi yang diperoleh

sudah mencakupi semua segmen yang ada.

D. Parameter Proses

Terdapat tiga proses utama dalam penelitian ini, yaitu K-means, optimasi inter kluster, dan optimasi intra kluster. Beberapa parameter yang digunakan dalam algoritma usulan disamakan dengan parameter yang digunakan pada [7] untuk kepentingan perbandingan. Tabel 1 menunjukkan parameter yang digunakan dalam penelitian ini.

Tabel 1. Parameter proses

Parameter	Algoritma original [7]	Algoritma usulan		
		Klasterisasi	Optimasi inter kluster	Optimasi intra kluster
Metode	MS-GA	K-Means	GA	MS-GA
Mutation rate	0,8	-	0,8	0,8
Crossover rate	0,3	-	0,3	0,3
n-kromosom	50	-	50	50
Max. iterasi	10.000*	100	200	500
Evaluasi	-	Euclidean	-	-
n-cluster	-	20	-	-

\*iterasi maksimum dibatasi menjadi 10.000 karena hasil percobaan awal sudah menunjukkan waktu komputasi yang lebih lama dan hasil yang tidak lebih baik dari algoritma usulan, sehingga tidak perlu diteruskan sampai 100.000 iterasi

E. Metode Perbandingan

Uji statistik digunakan untuk membandingkan jarak tempuh total pada waktu komputasi tertentu antara algoritma original dengan algoritma usulan. Hipotesis yang akan diuji dalam penelitian ini adalah:

$$H0: \mu_1 = \mu_2 \tag{9}$$

$$H1: \mu_1 \neq \mu_2 \tag{10}$$

dimana  $\mu_1$  adalah solusi terbaik yang didapatkan oleh algoritma original pada waktu komputasi ke- $n$ , dan  $\mu_2$  adalah solusi terbaik yang didapatkan oleh algoritma usulan pada waktu komputasi ke- $n$ .

III. HASIL DAN PEMBAHASAN

Dilakukan sebanyak 30 replikasi untuk mendapatkan alternatif solusi terbaik. Hasil optimasi menggunakan algoritma usulan K-means MS-GA ditunjukkan oleh Tabel 2. Berdasarkan Tabel 2, diketahui rata-rata waktu komputasi untuk menyelesaikan proses optimasi adalah 372,84 detik.

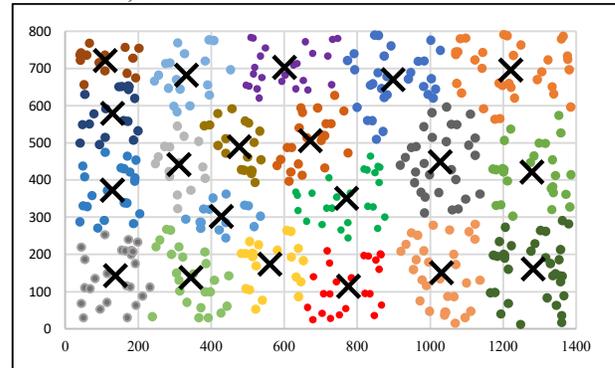
Tabel 2. Hasil K-Means MS-GA

	Total jarak (mm)	Waktu komputasi (s)			Total
		k-means	inter cluster	intra cluster	
Max	37.238,37	5,69	9,49	584,02	594,00
Min	30.308,45*	2,92	4,83	308,34	316,41
Average	32.558,24	3,42	5,97	363,45	372,84
Std. Dev.	1.448,23	0,81	1,39	89,18	91,11

\*Solusi terbaik didapatkan pada replikasi ke-19

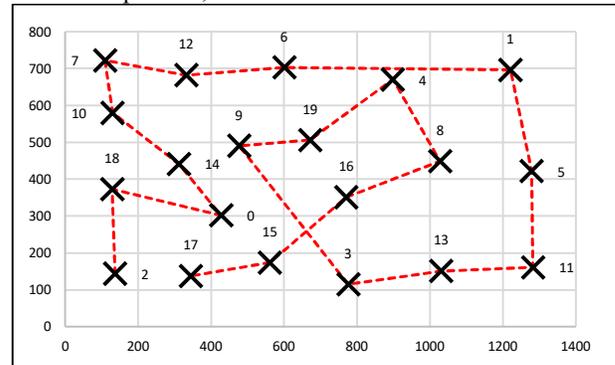
Pada algoritma usulan K-means MS-GA, berdasarkan hasil yang didapatkan, diketahui replikasi 19 memiliki total jarak tempuh minimal, yaitu 30.308,45 mm, atau turun sebesar 48% dibandingkan dengan solusi awal yaitu 58.389,98 mm. Berdasarkan [1], diketahui *feedrate* dari proses pematikan adalah 500 mm/menit, sehingga dapat

diketahui penurunan waktu *TC* sebesar 56,16 menit dari solusi awal. Hasil tersebut digunakan untuk menggambarkan algoritma usulan. Hasil dari klasterisasi menggunakan K-means diilustrasikan oleh Gambar 7, dimana tanda “X” menunjukkan titik berat dari setiap kluster. Waktu komputasi proses klasterisasi pada replikasi 19 adalah 5,24 detik.



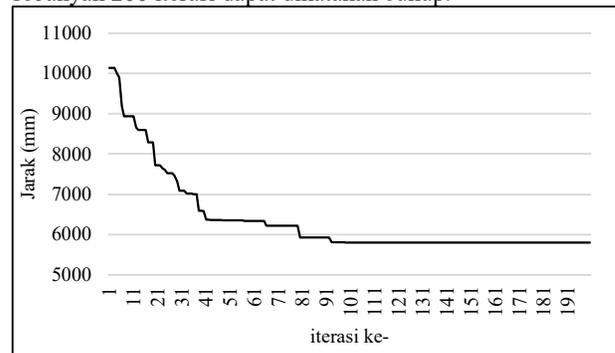
Gambar 7. Hasil klasterisasi menggunakan K-means.

Langkah selanjutnya adalah menentukan urutan kluster yang optimal. Optimasi inter kluster dilakukan menggunakan GA dan didapatkan hasil seperti yang dijelaskan oleh Gambar 8. Jarak tempuh inter kluster optimal pada replikasi 19 adalah 5.805,05 mm dengan waktu komputasi 9,07 detik.



Gambar 8. Hasil optimasi inter kluster.

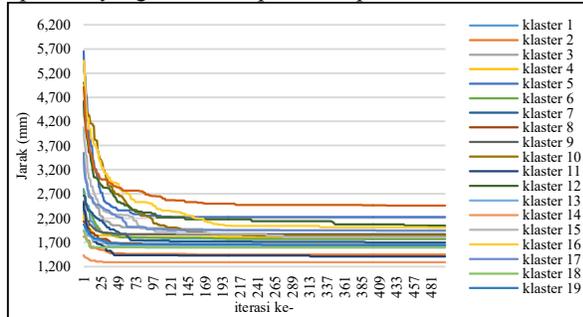
Gambar 9 menunjukkan konvergensi dari proses optimasi inter kluster. Berdasarkan Gambar 9, dapat diketahui bahwa GA mulai konvergen pada iterasi ke-100. Sehingga penentuan parameter jumlah iterasi maksimum sebanyak 200 iterasi dapat dikatakan cukup.



Gambar 9. Konvergensi optimasi inter kluster

Setelah dilakukan optimasi inter kluster, selanjutnya dilakukan optimasi intra kluster menggunakan MS-GA.

Optimasi dilakukan secara berurutan sesuai dengan urutan kluster yang optimal. Posisi terakhir dari solusi optimal kluster yang sudah dioptimasi kemudian digunakan sebagai titik awal pada kluster selanjutnya yang akan dioptimasi. Solusi terbaik didapatkan pada replikasi 19 dengan total jarak tempuh 30.308,45 mm dengan waktu komputasi pada proses optimasi intra kluster adalah 541,72 detik. Gambar 10 menunjukkan konvergensi dari proses optimasi yang dilakukan pada setiap kluster.



Gambar 10. Konvergensi optimasi intra kluster

Berdasarkan Gambar 10, dapat diketahui bahwa optimasi yang dilakukan pada setiap kluster cenderung konvergen pada iterasi ke 200. Sehingga, dapat disimpulkan bahwa penentuan jumlah iterasi maksimal pada optimasi intra kluster sebanyak 5000 sudah bisa dianggap cukup. Hal ini membuktikan bahwa dengan mengelompokkan segmen menjadi beberapa kluster menggunakan K-means, dapat mengurangi jumlah iterasi yang dibutuhkan secara signifikan. Bila dibandingkan dengan [7] yang membutuhkan 40.000 iterasi untuk mencapai konvergensi, algoritma usulan K-means MS-GA bisa dianggap lebih efisien untuk menyelesaikan *open arc routing problem*.

Total waktu komputasi proses klusterisasi, optimasi iter kluster, dan optimasi intra kluster pada replikasi 19 adalah 556,03 detik. Sedangkan berdasarkan rata-rata, jarak tempuh optimal yang bisa didapatkan dari algoritma K-means MS-GA adalah 32.558,24 mm, dengan waktu komputasi rata-rata yaitu 372,84 detik.

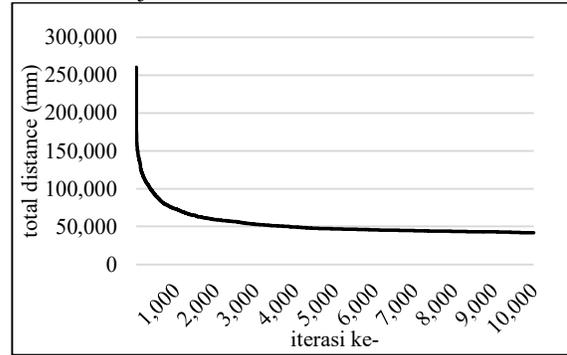
Untuk mengetahui efisiensi algoritma usulan, dilakukan perbandingan dengan algoritma original MS-GA [7]. Hasil yang didapatkan pada [7] menunjukkan bahwa total jarak tempuh optimal adalah 18.660,25 mm dengan 100.000 iterasi. Namun, hal ini tidak murni disebabkan oleh performansi algoritma, karena pada awal iterasi sudah diberikan alternatif solusi yang cukup baik dengan total jarak tempuh 26,905.06 mm yang didapatkan dengan cara meniru urutan pembatikan yang dilakukan oleh perajin batik [7]. Untuk kepentingan perbandingan, alternatif solusi awal tersebut dihilangkan, sehingga solusi yang didapatkan merupakan murni hasil dari performansi algoritma. Hasil optimasi menggunakan MS-GA ditunjukkan oleh Tabel 3.

Tabel 3. Hasil MS-GA dengan  $it_{max} = 10.000$

	Total Jarak (mm)	Waktu Komputasi (s)
Max	44.078,82	5.167,48
Min	41.900,50*	4.738,38
Average	42.779,65	4.923,34
Std. Dev.	826,63	164,17

\*solusi terbaik didapatkan pada replikasi ke-1

Berdasarkan Tabel 3, dapat diketahui bahwa iterasi ke-10.000 dapat diselesaikan dengan rata-rata waktu komputasi 4.923,36 detik dengan solusi terbaik 41.900,5 mm yang didapatkan pada replikasi ke-1. Konvergensi dari MS-GA ditunjukkan oleh Gambar 11.



Gambar 11. Grafik konvergensi MS-GA.

Berdasarkan Gambar 11, dapat diketahui bahwa pada iterasi ke-10.000 algoritma optimasi masih belum menunjukkan konvergensi dan masih ada kecenderungan untuk mendapatkan hasil yang lebih optimal. Namun, waktu komputasi pada iterasi ke-10.000 sudah melebihi waktu komputasi algoritma K-means MS-GA sebesar 9 kali lipat, dengan total jarak yang dihasilkan tidak menunjukkan hasil yang lebih baik. Oleh sebab itu, perbandingan tidak dilakukan pada hasil akhir iterasi, namun perlu ditentukan titik waktu tertentu untuk membandingkan hasil optimal yang didapatkan pada titik waktu tersebut. Waktu komputasi K-means MS-GA terlama adalah 594 detik. Waktu tersebut bisa dijadikan referensi untuk mengevaluasi hasil terbaik yang didapatkan oleh MS-GA. Tabel 4 menjelaskan perbandingan hasil optimasi antara MS-GA dan K-means MS-GA pada  $t = 594$  detik.

Tabel 4.

Perbandingan hasil MS-GA dengan K-means MS-GA pada  $t = 594$  detik

	MS-GA		K-Means MS-GA	
	Total jarak (mm)	Waktu Komputasi (s)	Total Jarak (mm)	Waktu komputasi (s)
Max	71.982,79	594	37.238,37	594
Min	63.038,81	-	30.308,45	316.41
Average	66.595,66	-	32.558,24	372.84
Std. Dev.	3.581,01	-	1.448,23	91.11

Berdasarkan Tabel 4, dapat diketahui bahwa pada  $t = 594$  detik, solusi terbaik yang dihasilkan oleh MS-GA adalah 63.038,81 mm, sedangkan pada K-Means MS-GA didapatkan solusi terbaik sebesar 30.308,45 mm. Uji statistik dilakukan untuk mengetahui apakah terdapat perbedaan signifikan antara MS-GA dan K-means MS-GA. Diasumsikan kedua populasi tersebut terdistribusi normal dan standar deviasi populasi tidak diketahui. *F-test* dilakukan untuk menguji apakah kedua sampel memiliki varians yang sama atau tidak. Berdasarkan *F-test*, dengan nilai  $\alpha = 0,05$  diketahui  $F_{stat} = 0,1635$  sedangkan  $F_{critical} = 0,3701$  sehingga dapat disimpulkan bahwa kedua sampel memiliki varian yang sama. Karena diketahui kedua sampel memiliki varian yang sama, maka *t-test assuming equal variances* dapat digunakan untuk membandingkan

solusi terbaik antara MS-GA dan K-means MS-GA. Berdasarkan uji  $t$ , dengan nilai  $\alpha = 0,05$  diketahui nilai  $t_{stat} = -38,23$  sedangkan nilai  $t_{critical\ two\ tail} = 2.03$ . Karena  $|t_{stat}| > t_{critical}$ , maka  $H_0$  (9) ditolak, sehingga dapat disimpulkan bahwa terdapat perbedaan signifikan antara solusi optimal MS-GA dan K-means MS-GA pada  $t = 594$  detik.

Berdasarkan hasil yang diperoleh, dapat diketahui bahwa algoritma K-means MS-GA bisa memberikan solusi yang lebih baik dengan waktu komputasi yang lebih singkat. Meskipun demikian, masih terdapat beberapa kelemahan pada algoritma tersebut. Salah satunya adalah penentuan jumlah kluster yang masih dilakukan secara manual. Jumlah kluster berpengaruh pada kompleksitas optimasi. Jumlah kluster yang banyak akan meningkatkan kompleksitas optimasi inter kluster. Sedangkan jumlah kluster yang sedikit akan meningkatkan kompleksitas intra kluster, sehingga perlu ditentukan titik keseimbangan antar keduanya. Selain itu, proses optimasi intra kluster yang dilakukan secara *sequential* tidak dapat menjamin bahwa titik terakhir pada kluster yang dioptimasi merupakan titik terdekat dengan kluster berikutnya. Sehingga terdapat kemungkinan adanya *excess motion* pada perpindahan antar kluster.

#### IV. KESIMPULAN

Penelitian ini bertujuan untuk menurunkan waktu komputasi dari permasalahan optimasi rute pematikan pada mesin CNC batik tulis menggunakan teknik klusterisasi. K-means digunakan untuk membagi segmen batik menjadi beberapa kluster, kemudian dilakukan optimasi inter kluster menggunakan GA sehingga didapatkan urutan kluster yang optimal. Selanjutnya dilakukan optimasi intra kluster menggunakan MS-GA untuk mendapatkan urutan segmen pematikan yang optimal. Solusi terbaik yang didapatkan MS-GA adalah 41.900,5 mm dengan 10.000 iterasi sedangkan K-Means MS-GA mendapatkan total jarak 30.308,45 mm dengan 500 iterasi per kluster. Hasil uji statistik yang dilakukan menunjukkan bahwa terdapat perbedaan signifikan antara solusi optimal MS-GA dan K-means MS-GA pada  $t = 594$  detik. Sehingga dapat disimpulkan bahwa algoritma usulan mampu memberikan waktu komputasi yang lebih singkat dan solusi optimal yang lebih baik.

Terdapat beberapa rekomendasi penelitian selanjutnya untuk mengatasi kelemahan dalam penelitian ini, diantaranya: (1) melakukan metode *hyper-heuristic* untuk memasukkan optimasi parameter dalam algoritma utama, dan (2) memberikan *constraint* tambahan untuk menjamin bahwa titik terakhir yang ditentukan pada optimasi intra kluster merupakan titik terdekat dengan kluster berikutnya.

#### DAFTAR PUSTAKA

- [1] R. Kusumawardani, Perancangan Motif dan Produksi Batik Tulis pada Mesin CNC Batik Tulis untuk Meminimalkan Waktu Pematikan, Yogyakarta: Thesis Pascasarjana Teknik Industri Universitas Gadjah Mada, 2018.
- [2] M. A. Wibisono, I. G. B. B. Dharma, A. Suwastono and M. A. Imani, "Integrasi Desain dan Manufaktur Batik Cap," *Jurnal Ilmu Pengetahuan dan Teknologi Tepat Guna Universitas Gadjah Mada*, vol. 1, no. 2, pp. 73-81, 2012.
- [3] T. B. Dwinugroho, D. Y. H. Kumarajati, Kurniawanti and Y. T. Hapsari, "Design and Implementation of CNC (Computer Numerical Control) Based Automatic Stamp Batik Machine Program with Automatic Gripper Using Mach 3," *Journal of Physics: Conference Series*, no. 1254, p. 012069, 2019.
- [4] M. Hanif, M. A. Wibisono and I. G. B. B. Dharma, "Perancangan Mesin Batik Cap Otomatis Tipe Modul Cap Bergerak," in *Seminar Nasional Teknik Industri UGM*, 2017, 2017.
- [5] R. Kusumawardani, F. Risqi and A. Sudiarso, "Penentuan Parameter Suhu dan Feed Rate pada Mesin CNC Batik Tulis," in *Seminar Nasional IENACO*, Surakarta, 2018.
- [6] M. B. Anugerah and A. Sudiarso, Optimasi Parameter Mesin CNC Batik Menggunakan Metode Taguchi dan Pendekatan Expert Judgement pada Pematikan dengan Motif Kontemporer, Yogyakarta: Skripsi Sarjana Teknik Industri UGM, 2021.
- [7] H. Ridho, A. Sudiarso and R. J. Kuo, "Tool Path Optimization of CNC Hand-Drawn Batik Machine using Master-Slave Genetic Algorithm," in *55TH International Conference on Engineering, Technology and Applied Science*, Taipei, 2020.
- [8] J. C. Chen and T. X. Zhong, "A Hybrid-Coded Genetic Algorithm Based Optimisation of Non-Productive Paths in CNC Machining," *International Journal of Advance Manufacturing Technology*, vol. 20, pp. 163-168, 2002.
- [9] S. Z. Selim and M. A. Ismail, "K-Means-Type Algorithms: A Generalized Convergence Theorem and Characterization of Local Optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vols. PAMI-6, no. 1, pp. 81-87, 1984.
- [10] M. Ester, K. H. P., S. Sander and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, California, 1996.
- [11] B. Liu, Y. Xia and P. Yu, "Clustering Via Decision Tree Construction," in *Foundations and Advances in Data Mining (Studies in Fuzziness and Soft Computing)*, Berlin, Springer, 2005.
- [12] S. Kaski, "Self-Organizing Maps," in *Encyclopedia of Machine Learning*, Boston, Springer, 2011.
- [13] B. Santosa and P. Willy, *Metoda Metaheuristik: Konsep dan Implementasi*, Surabaya: Prima Printing, 2011.