

# **IMPLEMENTASI DYNAMIC CLUSTER MENGUNAKAN METODE K-MEANS CLUSTER DAN PARTICLE SWARM OPTIMIZATION UNTUK IDENTIFKASI KARAKTERISTIK LALU LINTAS BERDASARKAN JUMLAH DAN JENIS KENDARAAN**

**Nama Mahasiswa** : TSABBIT AQDAMI MUKHTAR  
**NRP** : 5110 100 227  
**Jurusan** : Teknik Informatika FTIF-ITS  
**Dosen Pembimbing 1** : Prof.Dr.Ir.Joko Lianto Buliali,M.Sc.  
**Dosen Pembimbing 2** : Diana Purwitasari, S.Kom,M.Sc.

## **Abstrak**

*Jalan raya merupakan fasilitas umum yang paling sering digunakan oleh masyarakat dalam kehidupan sehari-hari. Hal tersebut disebabkan karena jalan raya berfungsi untuk menghubungkan antar satu tempat dengan tempat lainnya. Setiap jalan raya memiliki lebar dan pembagian jalur yang berbeda-beda disesuaikan dengan jumlah pengguna yang melintas.*

*Meski terdapat algoritma pengelompokan K-Means Cluster, namun K-means Cluster saja tidak cukup untuk melakukan pengelompokan jalan raya karena K-means membutuhkan jumlah cluster yang pasti. maka dari itu, dikembangkan suatu metode Dynamic Cluster dengan menggunakan K-Means dan Particle Swarm Optimization (PSO) untuk mencari jumlah Cluster optimal. Algoritma PSO digunakan untuk mencari Jumlah Cluster yang optimal sementara K-Means digunakan untuk menghitung hasil Cluster terbaik.*

*Uji coba dilakukan dengan menggunakan uji parameter dan uji skenario berdasarkan jenis kendaraan. Uji parameter dimaksudkan untuk mencari nilai optimal ambang batas dan konstanta dari dari algoritma PSO. Uji skenario berdasarkan*

*jenis kendaraan dibagi menjadi dua yakni motor dan mobil di dalam rentang waktu tertentu. Hasil dari uji coba yang dilakukan adalah setiap Cluster yang terbentuk di dalam subskenario memiliki anggota yang selalu berubah-ubah berdasarkan kedekatan selisih jumlah kendaraan.*

***Kata kunci: K-Means Cluster, Dynamic Cluster, Particle Swarm Optimization.***

# **IMPLEMENTATION OF DYNAMIC CLUSTER USING K-MEANS CLUSTER AND PARTICLE SWARM OPTIMIZATION FOR TRAFFIC CHARACTERISTICS IDENTIFICATION CHARACTERISTICS BASED ON THE NUMBER AND THE KIND OF VEHICLES**

**Student's Name** : Tsabbit Aqdami Mulhtar  
**Student's ID** : 5110 100 227  
**Department** : Informatics Engineering, FTIF-ITS  
**First Advisor** : Prof.Dr.Ir. Joko Lianto Buliali, M.Sc.  
**Second Advisor** : Diana Purwitasari , S.Kom, M.Sc.

## **Abstract**

*The Roadways are the most of public facilities is used by people in their activities. It is because the roadway's function that connect one place to another places. Every roadway has its own capacity and the differences of compartmentation based on its vehicles.*

*Although K-Means Cluster can be used for grouping method, but it doesn't enough for this problem because it needs a spesific number of Clusters. This final pfoject implements new method in Dynamic Cluster using K-Means an d Particle Swarm Optimization for optimal solution in Cluster problem. Particle Swarm Optimization's algorithm is used to find the number of Clusters with which the K-Means is used to find the best Cluster result.*

*Experimental evaluation is done by testing the parameter values in Particle Swarm Optimization and some scenarios problem about the condition of roadways. Experimental is divided based on the two kind of vehicles, motorcyle and car in some ranges of the time. The result of Experimental that have been*

*done is in every Cluster in the subscenario has the different member of roadways that depends of the time.*

***Keywords: K-Means Cluster, Dynamic Cluster, Particle Swarm Optimization.***

## DAFTAR PERSAMAAN

Persamaan 2.1 .....	8
Persamaan 2.2 .....	8
Persamaan 2.3 .....	9
Persamaan 2.4 .....	10
Persamaan 2.5 .....	11
Persamaan 2.6 .....	14
Persamaan 2.7 .....	18

*[Halaman ini sengaja dikosongkan]*

## DAFTAR KODE SUMBER

Kode Sumber 4.1. Implementasi Kode Sumber Untuk Pembangkitan Bilangan Acak Berdistribusi <i>Exponential</i> .....	45
Kode Sumber 4.2. Implementasi Kode Sumber Untuk Pembangkitan Bilangan Acak Berdistribusi <i>Uniform</i> .....	45
Kode Sumber 4.3. Kode Sumber Implementasi Nilai Awal Kecepatan .....	46
Kode Sumber 4.4. Kode Sumber Implementasi Inisiasi Nilai Kecepatan .....	47
Kode Sumber 4.5. Kode Sumber Implementasi Inisiasi Nilai Awal Posisi .....	47
Kode Sumber 4.6. Kode Sumber Implementasi Posisi .....	47
Kode Sumber 4.7. Kode Sumber Implementasi Pengecekan Terhadap Nilai Ambang Batas .....	47
Kode Sumber 4.8. Kode Sumber untuk Menyimpan Partikel Sebagai <i>Pbest</i> .....	48
Kode Sumber 4.9. Kode Sumber Pengecekan <i>Pbest</i> untuk Menjadi <i>Gbest</i> .....	48
Kode Sumber 4.10. Kode Sumber Inisiasi Nilai Masukan Minimum dan Maksimum <i>Cluster</i> .....	48
Kode Sumber 4.11. Kode Sumber Pembentukan <i>Cluster</i> Dengan Distribusi Normal .....	49
Kode Sumber 4.12. Kode Sumber Inisiasi <i>Centroid</i> Setiap <i>Cluster</i> .....	49
Kode Sumber 4.13. Kode Sumber Pembentukan <i>Cluster</i> .....	50

*[Halaman ini sengaja dikosongkan]*

## **BAB II**

### **DASAR TEORI**

Bab ini berisi penjelasan teori-teori yang berkaitan dalam pengerjaan tugas akhir. Adapun teori yang dijelaskan meliputi berbagai algoritma yang digunakan sebagai algoritma *Dynamic Cluster* yakni *Particle Swarm Optimization*, *K-Means Cluster*, dan evaluasi kualitas hasil *Cluster* menggunakan *Davies-Bouldin Index*.

#### **2.1 Pengumpulan Data Jalan Raya**

Di dalam pengerjaan Tugas Akhir ini data berasal dari dua sumber, yakni sumber data berdasarkan pengamatan di lapangan dan pembangkitan menggunakan bilangan acak. Data berdasarkan pengamatan adalah data yang diamati secara langsung dengan kondisi di lapangan. Dari data yang diperoleh dilakukan pembangkitan dengan pola penyebaran distribusi yang sesuai dengan data tersebut. Pencarian pola penyebaran distribusi data dilakukan dengan permodelan nilai masukan ( *input modelling* ) [4]. Pembangkitan data dilakukan terhadap setiap jalan raya dengan rentang waktu satu menit. Perlu diketahui pula bahwa untuk permodelan nilai masukan ini menggunakan dua jenis distribusi, yakni distribusi *exponential* dan *uniform* sesuai dengan pola penyebaran data yang ada.

Berikut adalah contoh data yang telah diambil dalam 5 menit pertama. Adapun data setiap waktu kedatangan kendaraan yang dicatat yakni pada detik ke-1, 5, 7, 8, 9, 13, 15, 16, 28, 29, 34, 40, 43, 50, 52, 55, 56, 57, 63, 67, 68, 69, 70, 73, 78, 85, 89, 90, 93, 98, 103, 113, 116, 118, 119, 122, 126, 131, 133, 136, 138, 141, 145, 146, 148, 150, 155, 160, 161, 164, 166, 167, 172, 175, 181, 186, 192, 198, 202, 203, 215, 215, 216, 223, 225, 231, 233, 236, 238, 242, 243, 244, 245, 247, 248, 253, 261, 262, 263, 266, 269, 274, 276, 281, 285, 293, 295. Data-data tersebut adalah data yang tercatat pada kendaraan yang melintasi jalan raya dalam waktu 5 menit dan terdapat 87 kendaraan yang melintasi.

Uji permodelan terhadap data tersebut adalah untuk membuktikan bahwa data tersebut terdistribusi secara *exponential*. Untuk menguji hipotesis tersebut, perlu dihitung nilai daripada  $D^-$  dan  $D^+$ , dimana  $D^-$  dapat dihitung dengan Persamaan 2.1. dan  $D^+$  dihitung dengan menggunakan Persamaan 2.2.

$$D^- = \sum_{i=1}^n \frac{i}{n} - x_i \quad (2.1)$$

$$D^+ = \sum_{i=1}^n x_i - \frac{i-1}{n} \quad (2.2)$$

Di mana:

$n$  = jumlah data, dalam hal ini data berjumlah 87.

$i$  = indeks data yang bernilai 1 hingga 87.

$x_i$  = jumlah waktu dalam satuan detik, dimana dalam hal ini data hasil pengamatan selama 5 menit atau setara dengan 300 detik.

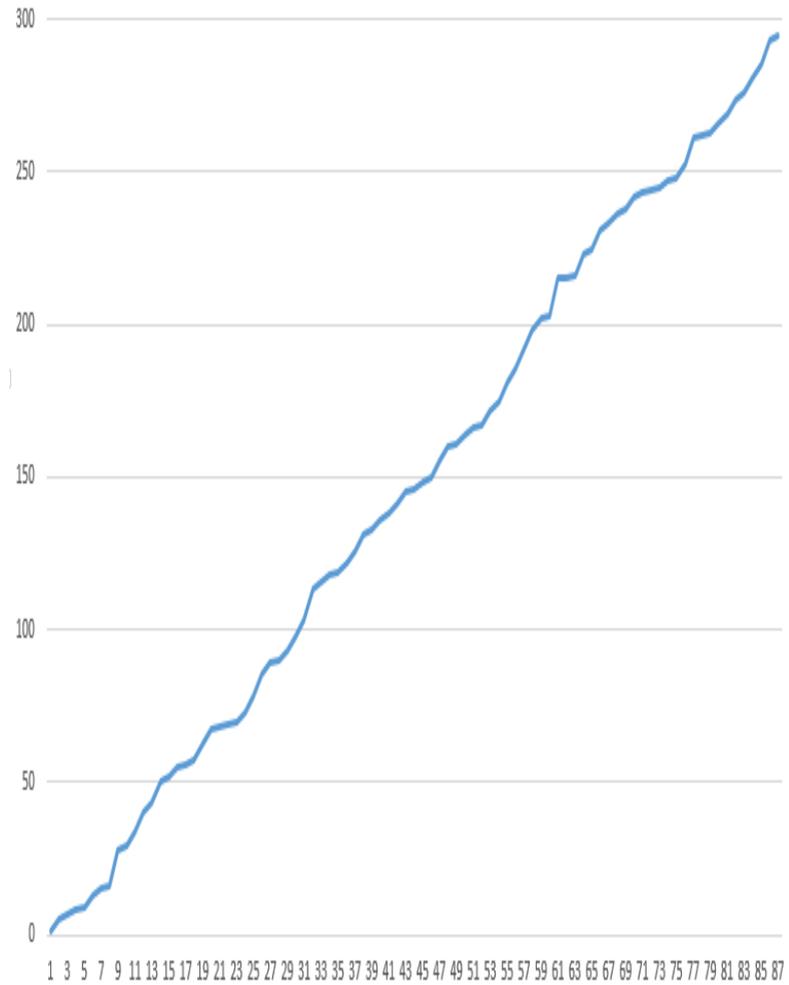
Nilai dari perhitungan nilai  $D^-$  terhadap data di atas adalah bernilai 0,1438621 dan  $D^+$  bernilai 0,18713. Untuk nilai maksimum dari keduanya adalah nilai yang ditunjukkan oleh  $D^-$  yakni 0,1438621.

Nilai tersebut akan dibandingkan dengan nilai dari tabel *Kolmogorov-Smirnov* dengan Persamaan 2.3. Dengan nilai yang ada, maka hasilnya adalah 0,145807. Nilai ini lebih kecil daripada nilai yang di dapatkan dari nilai  $D^-$  maka nilai tersebut diterima.

$$X = \frac{1,36}{\sqrt{n}} \quad (2.3)$$

Di mana:

$n$  = jumlah data, dalam hal ini berjumlah 87.



Gambar 2.1. Grafik Penyebaran Data Kedatangan Kendaraan Dalam 5 Menit

Cara pembangkitan bilangan dengan distribusi *exponential* dapat dilakukan dengan Persamaan 2.4.

$$x_i = -\log(1 - r) \times \mu \quad (2.4)$$

Di mana:

$x_i$  = nilai dari distribusi *exponential*.

$r$  = nilai acak dari 0 hingga 1.

$\mu$  = nilai rata-rata dari data.

Selanjutnya untuk pengujian hipotesis terhadap data yang berdistribusi *uniform* dapat dilihat pada Tabel 2.1.

Tabel 1.1. Tabel Pengujian Data Dengan Distribusi *Uniform*

$x_i$	Frekuensi	Frekuensi Relatif	$Fa(x_i)$	$Fe(x_i)$	$ Fa(x_i) - Fe(x_i) $
11	5	0,25	0,25	0,52	0,27
12	2	0,10	0,35	0,57	0,22
13	1	0,05	0,40	0,62	0,22
14	2	0,1	0,50	0,67	0,17
15	2	0,1	0,60	0,71	0,11
16	2	0,1	0,70	0,76	0,06
17	1	0,05	0,75	0,81	0,06
20	2	0,10	0,85	0,95	0,1
21	3	0,15	1	1	0

Pada Tabel 2.1 terdapat  $Fa(x_i)$  di mana nilai tersebut merupakan nilai frekuensi relatif kumulatif yang merupakan jumlah kendaraan sebelum waktu ke  $i$  melintasi jalan tersebut dan  $Fe(x_i)$  yang merupakan frekuensi teoritis dimana nilai dari frekuensi tersebut didapat dengan cara membagi nilai ke  $x_i$  dengan nilai terbesar yang ada pada kolom  $x_i$ . Untuk membuktikan bahwa nilai-nilai tersebut berdistribusi *Uniform* maka nilai terbesar dari kolom yang menghitung nilai  $|Fa(x_i) - Fe(x_i)|$  dibandingkan

dengan nilai di dalam tabel *Kolmogorov-Smirnov* pada Tabel 2.2 dengan jumlah data 20 dan nilai  $\alpha$  sama dengan 0,05. Di dalam pembuktian kasus ini nilai terbesar dari kolom terakhir adalah 0,24 dan nilai tabel dari tabel *Kolmogorov-Smirnov* adalah 0,294 sehingga hipotesis bahwa data ini berdistribusi *unifotm* dapat diterima. Untuk persamaan membangkitkan nilai berdistribusi *uniform* ditunjukkan oleh Persamaan 2.5.

$$x_i = a + (b - a) \times r \quad (2.5)$$

Di mana:

- $x_i$  = Nilai dengan distribusi *uniform*.
- $a$  = Nilai minimum dari data.
- $b$  = Nilai maksimum dari data.
- $r$  = Nilai acak dengan rentang nilai 0 hingga 1.

Tabel 1.2. Tabel Kolmogorov-Smirnov

$N$	$\alpha = 0,2$	$\alpha = 0,1$	$\alpha = 0,05$	$\alpha = 0,02$	$\alpha = 0,01$
1	0,900	0,950	0,975	0,990	0,995
5	0,447	0,509	0,563	0,627	0,669
10	0,323	0,369	0,409	0,457	0,486
15	0,266	0,304	0,338	0,377	0,404
20	0,232	0,265	<b>0,294</b>	0,329	0,352
25	0,208	0,238	0,264	0,295	0,317
30	0,190	0,202	0,242	0,270	0,290

### 2.1.1 Pembangkitan Data Dengan Bilangan Acak Berdistribusi *Exponential*

Di dalam pembangkitan bilangan acak berdistribusi *exponential*, dibutuhkan satu parameter yakni rata-rata kemunculan data. Sebagai contoh, terdapat suatu pembangkitan data dengan rata-rata kemunculan data 5 pada Tabel 2.3. Nilai

untuk rentang bilangan acak adalah 0 hingga 1. Nilai ini di dalam tabel direpresentasikan oleh kolom  $rI$  dan untuk hasil pembangkitannya direpresentasikan pada kolom  $V$ .

Tabel 1.3. Tabel Pembangkitan Bilangan Acak Berdistribusi *Exponential* dengan Rerata Kemunculan Data Bernilai 5

$rI$	$V$
0,890711	0,251317
0,026729	7,865052
0,551725	1,291388
0,182052	3,699018
0,617176	1,047955
0,117911	4,642227
0,832193	0,398881

### 2.1.2 Pembangkitan Data Dengan Bilangan Acak Berdistribusi Uniform

Pembangkitan data bilangan acak dengan distribusi *uniform* adalah pembangkitan bilangan acak yang berada pada rentang tertentu. Pada Tabel 2.4. di tunjukkan bagaimana distribusi ini dihasilkan antara 5 hingga 10. Pada kolom pertama terdapat nilai  $rI$  yang merupakan representasi nilai bilangan acak antara 0 hingga 1. Sementara pada kolom kedua yakni  $V$  merupakan kolom yang berisikan hasil pembangkitan bilangan acak.

Tabel 1.4. Tabel Nilai Pembangkitan Bilangan Acak Berdistribusi *Uniform* dengan Rentang 5 hingga 10

$rI$	$V$
0,887352	9,43676
0,636404	8,182021
0,387234	6,936168
0,847379	9,236895
0,86286	9,314299
0,668519	8,342596

$rI$	$V$
0,701255	8,506277

## 2.2 Algoritma *Dynamic Cluster*

Salah satu algoritma yang digunakan untuk menyelesaikan permasalahan pengelompokan data adalah algoritma *Dynamic Cluster* yang merupakan modifikasi dari dua algoritma yang lain, yakni *Particle Swarm Optimization* dan *K-Means Cluster* [5]. Di dalam algoritma *dynamic Cluster* tidak diberikan jumlah *Cluster* ataupun *centroid* (titik tengah) untuk setiap *Cluster*. Setiap *centroid* dan jumlah *Cluster* yang terbentuk ditetapkan secara random ataupun dengan algoritma lainnya.

### 2.2.1 Penentuan Jumlah *Centroid Cluster* Dengan Menggunakan *Particle Swarm Optimization*

*Particle Swarm Optimization* (PSO) merupakan teknik optimasi berdasarkan populasi yang terinspirasi dari perilaku sosial dan kawan-kawan burung ataupun sekelompok ikan yang sedang mencari makan [6]. PSO pertama kali diperkenalkan oleh J.Kennedy dan R.Eberhart pada tahun 1995. PSO sebagai alat optimasi menyediakan prosedur pencarian berbasis populasi dimana masing-masing individu yang disebut sebagai partikel mengubah posisinya terhadap waktu. Masing-masing partikel bergerak mengitari ruang solusi dan menyesuaikan posisinya berdasarkan pengalaman sebelumnya secara bersama-sama sebagai kelompok kawan-kawan (*swarm*).

Setiap partikel pada algoritma PSO memiliki dua hal penting, yakni posisi  $(x_1, x_2, \dots, x_n)$  dan kecepatan  $(v_1, v_2, \dots, v_n)$  dimana  $n$  merupakan jumlah partikel yang ada di dalam *swarm*. Di dalam algoritma PSO kecepatan dan posisi partikel diinisiasi dengan nilai random tertentu. Selanjutnya nilai kecepatan dan posisi akan dihitung melalui fungsi objektif. Adapun persamaan untuk menghitung kecepatan ditunjukkan oleh Persamaan 2.6.

Di dalam Tugas Akhir ini, pencarian ruang solusi menggunakan PSO adalah pencarian untuk mengoptimasi jumlah *Cluster*. Di dalam kegunaannya memecahkan solusi optimasi jumlah *Cluster*, PSO menjadikan semua ruang solusinya ke dalam bentuk *swarm*. *Swarm* merupakan sejumlah data yang kemudian tersusun atas sejumlah kemungkinan ruang solusi jumlah *Cluster*. Setiap kandidat yang menjadi ruang solusi di dalam *swarm* inilah yang disebut sebagai partikel. Antara partikel satu dengan yang lainnya memiliki jumlah *Cluster* yang berbeda-beda.

$$V_i^{t+1} = w^{t+1} * V_i^t + c1 * rand() * (-1 - Y_i^t) + c2 * rand() * (1 - Y_i^t) \quad (2.6)$$

Di mana:

- $V_i^{t+1}$  adalah kecepatan partikel  $i$  pada saat iterasi ke  $t+1$ .
- $w^{t+1}$  adalah konstanta pada saat iterasi ke  $t+1$ .
- $c1$  merupakan konstanta nilai kenaikan kecepatan.
- $c2$  merupakan konstanta nilai kenaikan posisi.
- Fungsi  $rand()$  merupakan nilai random berdistribusi *uniform* yang bernilai -1 hingga 1.
- Dan  $Y_i^t$  merupakan posisi partikel  $i$  pada iterasi ke  $t$ .

Adapun posisi partikel pada ruang solusi ditunjukkan oleh Persamaan 2.7.

$$Y_i^{t+1} = Y_i^t + V_i^{t+1} \quad (2.7)$$

Di mana:

- $Y_i^{t+1}$  merupakan posisi partikel  $i$  pada iterasi ke  $t+1$ .
- $Y_i^t$  merupakan posisi partikel  $i$  pada iterasi ke  $t$ .

- $V_i^{t+1}$  merupakan kecepatan partikel  $i$  pada iterasi ke  $t$ .

Sebagaimana pencarian ruang solusi pada umumnya dengan menggunakan PSO, pencarian ruang solusi ini nantinya akan menghasilkan suatu nilai partikel yang dianggap memiliki nilai paling optimal diantara nilai lainnya. Partikel terbaik ini dipilih dari sejumlah kandidat partikel yang ada di dalam ruang solusi. Partikel-partikel ini disebut sebagai *Local Best Solution* (*Pbest*) dan partikel terbaik dari *seluruh* kandidat di dalam pencarian ruang solusi disebut sebagai *Global Best Solution* (*Gbest*).

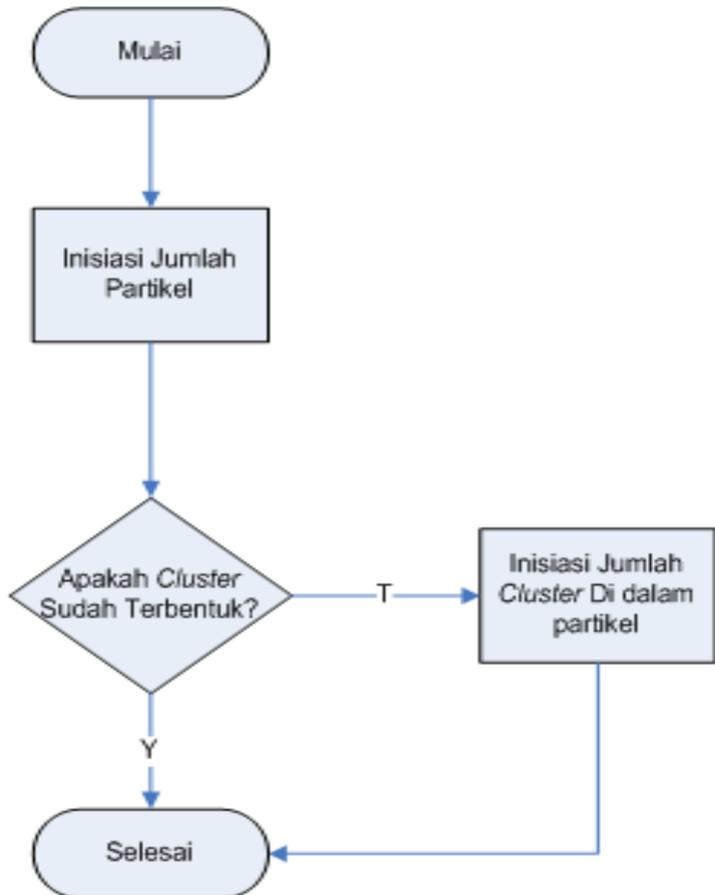
Pemilihan dari *Gbest* ini di dalam metode PSO untuk Tugas Akhir ini dilakukan dengan suatu fungsi evaluasi. Seperti yang sudah disebutkan sebelumnya bahwa setiap partikel terdiri atas sejumlah *Cluster*, maka dengan demikian evaluasi *Cluster* untuk setiap partikel inilah yang disebut sebagai fungsi untuk menentukan kandidat *Gbest* di dalam suatu *swarm*.

Pertama untuk setiap partikel yang ada di dalam *swarm* diinisiasi kecepatan dan posisinya dengan sebuah variabel random. Baru kemudian setiap partikel menginisiasi jumlah *Cluster* sendiri. Partikel yang telah memiliki jumlah *Cluster* kemudian dievaluasi. Hasil evaluasi inilah yang kemudian disimpan sebagai nilai penentu dari setiap kandidat *Pbest* untuk kemudian setelah seluruh partikel dievaluasi, nilai ini digunakan untuk memilih *Gbest* seperti yang dijelaskan sebelumnya. Hal ini terus berlanjut hingga jumlah iterasi maksimal. Adapun untuk pembentukan algoritma ini.

Di dalam Tugas Akhir ini, pembentukan partikel dilakukan dengan menginisiasi nilai acak *Cluster* untuk sejumlah nama jalan raya. Hal ini dilakukan agar setiap partikel memiliki identitas yang unik dengan partikel lainnya. Adapun untuk proses ini ditunjukkan oleh diagram alir pada Gambar 2.2.

Di dalam diagram alir pada Gambar 2.2 mula-mula sejumlah partikel di dalam *swarm* diinisiasi jumlahnya.

Selanjutnya, untuk setiap partikel yang sudah terbentuk diinisiasi secara acak jumlah *Cluster*nya dengan rentang nilai tertentu di dalam setiap partikel. Dalam Tugas Akhir ini, nilai rentang untuk bilangan acak pembentukan *Cluster* adalah 5 hingga 10.



Gambar 2.2. Diagram Alir Untuk Menginisiasi Partikel

### 2.2.2 Pembentukan *Cluster* Dengan Menggunakan *K-Means Cluster*

*K-Means Cluster* merupakan salah satu metode yang digunakan untuk menganalisis data ataupun proses permodelan untuk *Unsupervised Data*[7]. Metode *K-Means Cluster* digunakan untuk mengelompokkan data dengan cara memartisi sejumlah data yang besar menjadi beberapa kelompok yang memiliki karakteristik yang sama. Adapun yang dimaksud dengan karakteristik yang sama yakni dari sisi kuantitas. Metode ini berusaha untuk meminimalkan variasi antar data yang ada di dalam *Cluster* dengan data yang ada di dalam *Cluster*.

Di dalam pengerjaan Tugas Akhir ini, algoritma pengelompokkan ini berfungsi untuk menjadikan beberapa jalan yang memiliki selisih jumlah kendaraan yang tidak jauh berbeda menjadi satu. Seperti yang dijelaskan pada subbab 2.2.1 bahwasannya jumlah *Cluster* telah dihitung menggunakan metode optimasi PSO. Jadi seperti algoritma *K-Means Cluster* pada umumnya, pengukuran algoritma ini terletak pada kuantitas kendaraan yang melintas.

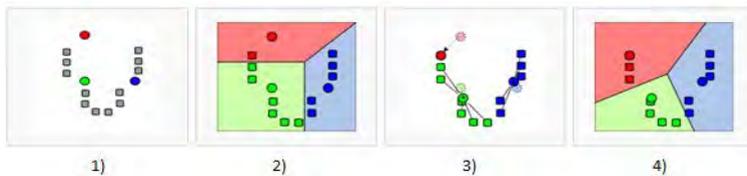
Perhitungan dalam menentukan kelompok *Cluster* ini menggunakan *Euclidian Distance*. Persamaan *Euclidean Distance* menghitung jarak antara suatu titik dengan titik yang lainnya dalam satu daerah atau himpunan. Adapun titik-titik yang dimaksud adalah data di dalam *dataset* dengan beberapa titik tengah (*centroid*) di dalam data. Jadi di dalam Tugas Akhir ini titik-titik ini adalah jumlah kendaraan yang melintas pada jalan-jalan raya tersebut. Adapun persamaan *Euclidean Distance* ditunjukkan oleh Persamaan 2.8.

$$d_{i,j} = \sqrt{\sum_{k=1}^p \{X_{ik} - X_{jk}\}^2} \quad (2.8)$$

Di mana:

- $d_{i,j}$  merupakan Jarak antara objek  $i$  dan  $j$ .
- $P$  merupakan jumlah data di dalam *dataset*.
- $X_{ik}$  merupakan objek data ke  $i$  pada data  $k$ .
- $X_{jk}$  merupakan objek data yang ke  $j$  pada data ke  $k$  atau yang sering disebut sebagai *centroid* (nilai titik tengah) *Cluster* untuk jumlah data  $P$ .

Hasil keluaran yang dari *Cluster* ini diilustrasikan di dalam Gambar 2. Terlihat ada pembagian menjadi beberapa kelompok pada langkah keempat. Setiap kelompok direpresentasikan dengan warna yang berbeda antara satu dengan yang lainnya.



Gambar 2.2. Ilustrasi langkah *K-Means Cluster*

Pada Gambar 2.2. terlihat bagaimana suatu *Cluster* terbentuk. Langkah awal adalah menentukan beberapa *centroid* sesuai dengan jumlah *Cluster* yang diinginkan dari *dataset* yang ada. Kemudian dengan Persamaan 2.3 akan didapat nilai masing-masing jarak antara *centroid* yang telah ditentukan sebelumnya dengan seluruh data yang ada di dalam data uji coba seperti yang ditunjukkan pada langkah 2. setelah didapat seluruh jarak data dengan *centroid* yang ada, maka data dikelompokkan dengan cara mendekatkan seluruh data dengan *centroid* yang memiliki selisih paling kecil.

Di dalam Tugas Akhir ini, pembentukan *Cluster* dilakukan terhadap partikel yang telah menjadi *Pbest* dan untuk memilih diantaranya menjadi *Gbest* seperti yang dijelaskan pada subbab 2.2.1. Untuk nilai *centroid* dan jumlah *Cluster* yang terbentuk, seluruhnya telah diinisiasi secara acak. Oleh karena itu, penggunaan algoritma *K-Means Cluster* adalah untuk membentuk

*Cluster* yang sesungguhnya dari inisiasi jumlah *Cluster* yang sudah ada.

### 2.2.3 Evaluasi Kualitas Hasil *Cluster* Dengan Menggunakan Davies-Bouldin Index

*Davies-Bouldin Index (DB Index)* merupakan salah satu metode yang digunakan untuk mengevaluasi hasil dari setiap *Cluster* [8]. Metode ini pertama kali diusulkan oleh David L. Davies dan Donald W. Bouldin pada tahun 1979. Evaluasi menggunakan *DB Index* ini memiliki skema evaluasi internal *Cluster*, dimana baik atau tidaknya hasil *Cluster* dilihat dari kuantitas dan kedekatan antar data hasil *Cluster*.

Di dalam peranannya sebagai metode untuk mengevaluasi hasil luaran *Cluster*, metode *DB Index* ini sekaligus menjadi nilai untuk menentukan apakah partikel yang dievaluasi menjadi *Gbest* atau bukan.

Adapun persamaan untuk menghitung *DB Index* ditunjukkan oleh Persamaan 2.6.

$$DB = \frac{1}{k} \sum_{i=1}^k \max_{j \neq i} \{D_{i,j}\} \quad (2.7)$$

Di mana:

- *DB* merupakan nilai dari hasil perhitungan *DB Index*.
- *k* merupakan jumlah *Cluster* yang akan dievaluasi.
- $D_{i,j}$  merupakan nilai terkecil hasil perbandingan antar *Cluster* *i* dan *j*. Nilai ini dihasilkan oleh Persamaan 2.5.

Selanjutnya *DB Index* dengan nilai terkecil akan dipilih sebagai nilai evaluasi *Cluster* terbaik. *DB Index* sendiri merupakan hasil yang sebelumnya dihasilkan dari Persamaan 2.7.

$$D_{i,j} = \frac{(d_i + d_j)}{d_{i,j}} \quad (2.7)$$

Di mana:

$D_{i,j}$  = Jarak antar *Cluster*  $i$  dan  $j$ .

$k$  = jumlah *Cluster*.

$\bar{d}_i$  = Rerata nilai setiap point dengan *centroid* di dalam *Cluster*  $i$ .

$\bar{d}_j$  = Rerata nilai setiap point dengan *centroid* di dalam *Cluster*  $j$ .

Di dalam Tugas Akhir ini, perhitungan untuk fungsi evaluasi nilai ini dilakukan setelah *Cluster* terbentuk oleh *K-Means Cluster*.

## **BAB III**

### **PERANCANGAN PERANGKAT LUNAK**

Di dalam bab ini akan dijelaskan mengenai perancangan-perancangan secara teknis aplikasi yang dibuat. Bab ini berawal dari deskripsi secara umum aplikasi hingga perancangan proses dan alur implementasinya.

#### **3.1. Deskripsi Umum Sistem**

Di dalam tugas akhir ini akan dibangun sebuah aplikasi *Cluster Analyst* terhadap kondisi jalan raya berbasis desktop. Aplikasi ini memiliki kegunaan untuk melakukan pengelompokkan sejumlah jalan raya ke dalam kelompok yang memiliki karakteristik yang sama. Keluaran yang dihasilkan dari aplikasi ini berupa diagram yang menunjukkan jumlah kelompok terbaik jalan raya dengan karakteristik yang sama, grafik yang merupakan nilai evaluasi hasil *Cluster* dan nama-nama jalan di setiap *Cluster*.

Sebelumnya dilakukan proses pengambilan data jalan raya dengan mencatat kendaraan yang lewat selama 30 menit setiap jalan raya. Data didapat dari pengamatan langsung di lapangan dan penelitian lainnya. Selain itu juga data untuk beberapa jalan lainnya didapat dengan mencari kesamaan jalur yang ada.

Data yang diperoleh tadi dicatat dalam Tabel yang berisikan nama jalan dan jumlah kendaraan. Karena implementasi algoritma *Dynamic Cluster* yang digunakan adalah PSO, maka data dimodelkan sebagai *swarm* dan partikel. Pada Gambar 3.1 ditunjukkan bagaimana partikel yang berasal dari data uji coba yang dimodelkan sedemikian rupa. Terdapat sejumlah nama jalan dan jumlah kendaraan yang melintasi nama-nama jalan tersebut. Setiap *Cluster* di dalam setiap partikel dibedakan oleh warna. Sub dari Tabel pada Gambar inilah yang disebut sebagai *bit* data di dalam partikel. Partikel  $P = \{p_1, p_2, \dots, p_n\}$  dimana  $n$  merupakan jumlah duplikasi *bit* data sekaligus jumlah partikel di dalam satu *swarm*. Misalkan dari gambar tersebut pada partikel 1, *Cluster*

yang terbentuk ada tiga buah macam *Cluster* yakni warna merah dengan anggotanya Gubeng, Basuki Rahmat dan Mayjen Sungkono. *Cluster* berikutnya yakni Dharmahasada yang berwarna hijau dan yang terakhir adalah Kenjeran dan Ahmadyani.

Setiap partikel di dalam *swarm* memiliki bit data yang sama, namun jumlah *Cluster* yang berbeda antara satu dengan yang lain. Jumlah *Cluster* yang ada di dalam setiap partikel ditentukan oleh suatu fungsi *random* tertentu. Hasil *swarm* ini berisikan  $n$  partikel yang ditentukan sebelumnya. Setiap partikel berisikan  $k$  *Cluster* yang nilai rentang minimum dan maksimumnya telah ditentukan besarnya.

Gubeng	Kenjeran	Ahmadyani	Basuki Rahmat	Mayjen Sungkono	Dharmahasada
23	44	37	20	22	33

→ Partikel 1

Gubeng	Kenjeran	Ahmadyani	Basuki Rahmat	Mayjen Sungkono	Dharmahasada
23	44	37	20	22	33

→ Partikel 2

Gubeng	Kenjeran	Ahmadyani	Basuki Rahmat	Mayjen Sungkono	Dharmahasada
23	44	37	20	22	33

→ Partikel 3

Gambar 3.1. Ilustrasi 3 Buah Partikel Di Dalam Suatu *Swarm*.

Hasil dari proses pengambilan data berisikan satu set data yang terdiri atas sejumlah nama jalan raya dan jumlah kendaraan. Nama jalan dan jumlah kendaraan dipisahkan oleh tanda ‘,’ contoh: {basukirahmat,130}. Setiap set data merepresentasikan nama jalan dan jumlah kendaraan.

Algoritma *K-Means Cluster* yang dibangun berfungsi menghitung *Cluster* yang ada di setiap partikel di dalam *swarm*. Algoritma *K-Means Cluster* yang diimplementasikan di dalam tugas akhir ini sedikit berbeda dengan *K-Means Cluster* pada umumnya. Perbedaan tersebut terletak pada fungsi evaluasi yang ada pada *K-Means Cluster*. Umumnya, evaluasi hasil *Cluster*

menggunakan metode *Between Cluster Variation (BCV)* dan *Within Cluster Variation (WCV)*. Nilai yang dihasilkan dari metode evaluasi ini adalah dengan cara *convergen*, artinya tidak ada batasan mengenai tahapan iterasi perhitungan. Perhitungan dengan menggunakan metode evaluasi ini akan berhenti ketika nilai evaluasi pada iterasi ke  $t+1$  sama dengan saat ke  $t$ . Sementara pada tugas akhir ini diajukan suatu fungsi evaluasi lain yakni *DB Index* seperti yang dijelaskan di dalam subbab 2.2.3.

### 3.2. Arsitektur Umum Sistem

Sistem yang dirancang di dalam pengerjaan Tugas Akhir ini berbasis *desktop*. Seluruh proses dari mulai *input data*, *processing* dan keluaran *output* dilakukan di dalam satu aplikasi ini. Berdasarkan perancangan arsitektur umum sistem pada Gambar 3.2, data uji coba untuk uji coba disiapkan ke dalam file dengan *format .txt* yang berisikan nama jalan dan jumlah kendaraan yang melintasi. Kemudian setelah data disiapkan, data tersebut diproses oleh aplikasi di dalam komputer *client* menggunakan aturan-aturan yang telah dibuat berdasarkan data yang diterima.

Hasil yang ditampilkan pada akhir proses ini adalah berupa grafik yang menunjukkan hasil dari perhitungan evaluasi *Cluster* di setiap iterasi. Selain itu pula akan ditampilkan jumlah *Cluster* yang terbentuk dan nama jalan yang berada di dalam masing-masing *Cluster*.



Gambar 3.2. Perancangan Arsitektur Umum Sistem

### 3.3. Perancangan Data

Perancangan data adalah hal yang penting dalam sistem karena diperlukan data yang tepat agar sistem dapat beroperasi dengan benar. Sistem yang dibuat membutuhkan data masukan dan akan memberikan data keluaran. Data masukan berupa himpunan data yang akan digunakan sebagai data uji coba. Data keluaran dari sistem ini adalah hasil *Cluster* terbaik dan nilai evaluasinya.

Data jalan raya yang digunakan adalah data jalan raya yang didapat dari hasil pengamatan langsung dan pembangkitan bilangan acak di kota Surabaya. Pembangkitan bilangan acak ini dilakukan karena terbatasnya waktu dan tenaga untuk mengamati sejumlah jalan raya tersebut. Terdapat 6 jalan raya yang diamati secara langsung yakni Kertajaya, Jalan Raya ITS, Arief Rahman Hakiem, Mulyosari, dan Mulyorejo. Untuk 34 jalan raya lainnya, dilakukan pembangkitan bilangan acak dengan pola distribusi data untuk rentang waktu seperti yang ditunjukkan pada Tabel 3.1.

Tabel 3.3.1. Distribusi Rentang Waktu

<b>Kode Rentang</b>	<b>Rentang Waktu</b>	<b>Jenis Distribusi</b>
1	06.00-09.00	<i>Exponential</i>
2	09.00-12.00	<i>Uniform</i>
3	12.00-13.00	<i>Exponential</i>
4	13.00-16.00	<i>Uniform</i>
5	16.00-19.00	<i>Exponential</i>
6	19.00-22.00	<i>Exponential</i>
7	22.00-06.00	<i>Uniform</i>

Setiap jalan raya memiliki fungsi distribusi yang sama untuk setiap waktunya. Namun perbedaannya adalah sebaran parameter data yang digunakan untuk setiap distribusi, yakni tergantung apakah jalan tersebut termasuk kategori padat apa tidak sehingga jumlah minimum dan maksimum maupun rata-rata kedatangan kendaraan setiap jalan dapat diperkirakan.

Setiap jalan raya memiliki rentang kedatangan minimum dan maksimum yang berbeda-beda meskipun jenis distribusinya sama. hal ini disebabkan perbedaan kondisi dari setiap jalan raya yang diamati.

### **3.3.1. Data Uji Coba dan Pembentukan Model Dynamic Cluster**

- **Data Masukan**

Himpunan data yang digunakan sebagai data masukan adalah data yang dikumpulkan dari beberapa jalan raya baik dengan pengamatan langsung, data penelitian lainnya yang signifikan terhadap penelitian tugas akhir ini, maupun data yang berasal dari pembangkitan dengan melihat kesamaan karakteristik jalan raya. Di dalam pengerjaan Tugas Akhir ini, adapun jalan raya tersebut menggunakan jalan raya yang berada di kota Surabaya, diantaranya: Ahmad Yani, Basuki Rahmat, Dharmawangsa, Dharmahasada, Kertajaya, Prof.Dr.Mestopo Pemuda, Pangeran Diponegoro, Mayjen Sungkono, Bubutan, Kenjeran dan lain sebagainya. Setiap jalan raya memiliki distribusi yang sama setiap rentang waktunya namun berbeda di dalam jumlah kendaraan minimum, maksimum ataupun rata-ratanya sesuai yang ditunjukkan oleh Tabel 3.1.

Himpunan data adalah *swarm* yang terdiri atas partikel-partikel. Partikel-partikel di dalam *swarm* ini memiliki atribut nama jalan raya, jumlah kendaraan yang melintas dan jumlah *Cluster* yang ada di dalamnya. Permodelan data ini diilustrasikan seperti yang ada pada Gambar 3.1.

- **Data Keluaran**

Data keluaran dari proses pembentukan model di dalam algoritma PSO ini adalah berupa data partikel yang memiliki  $n$  jumlah *Cluster*. Partikel-partikel inilah yang kemudian digunakan untuk melakukan pencarian *Cluster* terbaik. Partikel-partikel ini disimpan dalam *list* dari sebuah kelas.

### 3.3.2. Data Sistem Pencarian *Cluster* Terbaik

- **Data Masukan**

Data masukan dari aplikasi pencarian *Cluster* terbaik adalah data berupa nama jalan dan jumlah kendaraan yang telah membentuk kelompok-kelompok beserta *centroid* atau titik tengah dari kelompok-kelompok tersebut.

- **Data Keluaran**

Data keluaran dari aplikasi ini adalah bentuk *Cluster* yang dinilai paling efektif dan nilai evaluasi dari *Cluster* tersebut. Bentuk *Cluster* yang ditampilkan adalah keluaran berupa berkas file teks dan juga *Plotting* data *Cluster*.

## 3.4. Perancangan Algoritma

Perancangan proses merupakan tahap untuk membentuk alur dalam penerapan algoritma *Dynamic Cluster* yang akan digunakan sebagai pencarian *Cluster* terbaik.

### 3.4.1. Desain Secara Umum

Sebelum proses pelatihan dan pembentukan data untuk PSO dimulai, seluruh jalan raya yang akan digunakan sebagai data uji coba dicatat jumlah kendaraan yang melintasi baik sepeda motor maupun mobil. Kemudian seluruh jalan raya tersebut dicari pola dan karakteristiknya, kemudian dilakukan pembangkitan

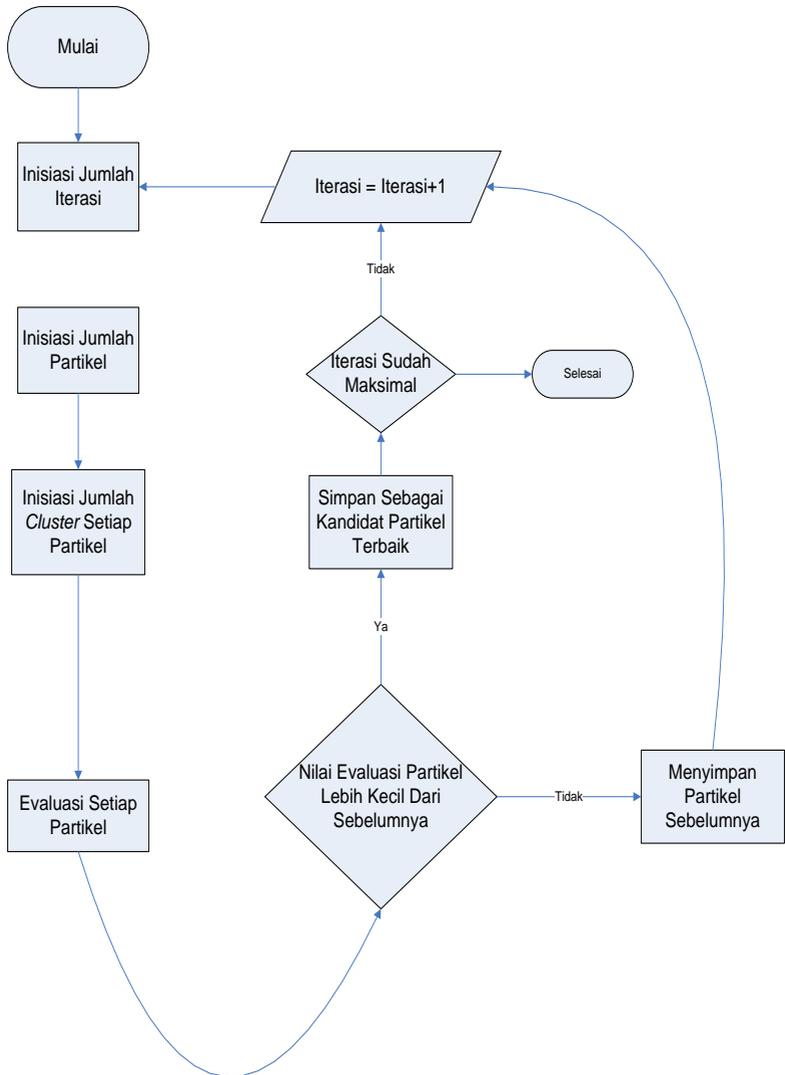
bilangan acak. Pembangkitan bilangan acak ini bertujuan untuk mendapatkan secara keseluruhan data di jalan raya tersebut selama satu tahun. Diasumsikan 1 tahun tersebut terdiri atas 12 bulan dimana setiap bulan terdiri dari 30 hari dengan hari efektif bekerja selama 24 hari dan hari libur 6 hari.

Setelah semua data untuk setiap jalan raya telah terkumpul, maka dipilihlah rentang waktu yang ingin diketahui jumlah kendaraan yang melintasi setiap jalan raya. Dari permodelan ini kemudian dicatat ke dalam file berekstensi *.txt* dengan nama jalan beserta jumlah kendaraan dipisahkan dengan koma.

Model dari algoritma *Dynamic Cluster* pada Tugas Akhir kali ini cukup berbeda dengan model *Dynamic Cluster* pada umumnya. Jika biasanya jumlah *Cluster* dalam *Dynamic Cluster* jumlah *Cluster* diacak diantara nilai minimum dan maksimum secara acak. Selain itu juga, evaluasi yang dilakukan jika umumnya menggunakan *Between Cluster Variation* (BCV) dan *Within Cluster Variation* (WCV) , di dalam Tugas Akhir ini menggunakan *DB Index*.

Perbedaan selanjutnya yang ada dengan *Dynamic Cluster* biasa dengan yang ada di dalam pengerjaan Tugas Akhir ini adalah cara untuk mendapatkan jumlah *Cluster*. Jika pada *Dynamic Cluster* biasanya menggunakan bilangan acak, sementara pada *Dynamic Cluster* yang dikerjakan menggunakan algoritma optimasi PSO dengan mencari jumlah *Cluster* pada ruang pencarian optimal.

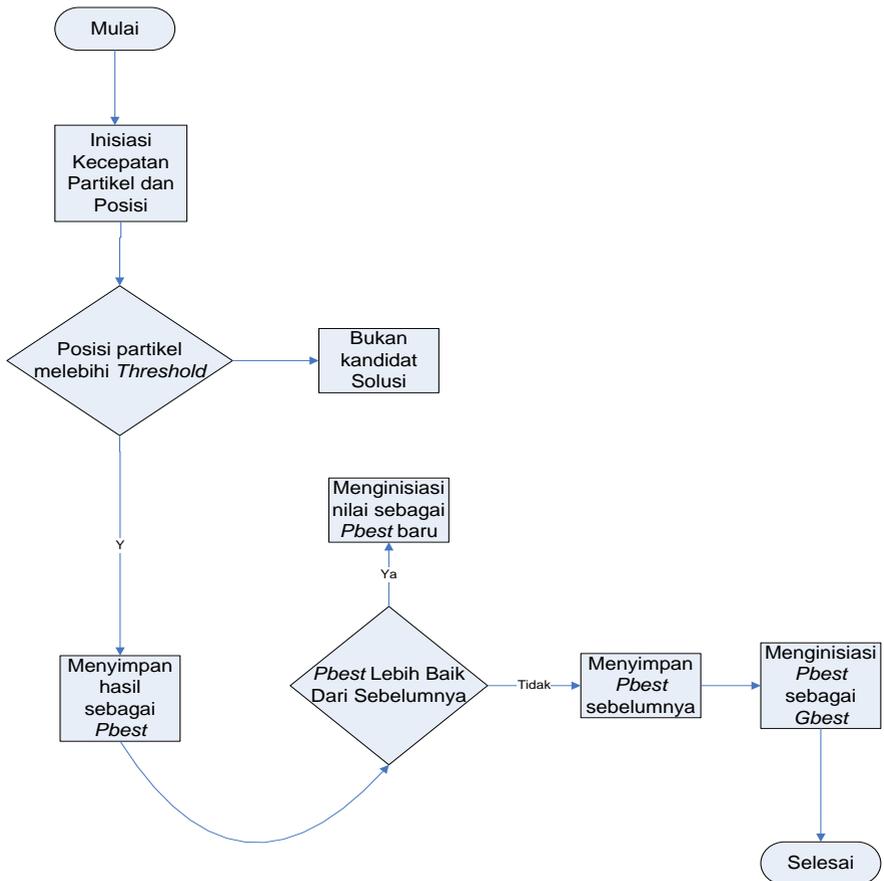
Berdasarkan aturan-aturan yang telah dibuat dari model *Dynamic Cluster*, proses pengelompokan data dilakukan dengan cara mengevaluasi setiap partikel yang ada dan memilih partikel dengan nilai evaluasi terkecil. Secara umum, perancangan algoritma untuk pengerjaan Tugas Akhir ini ditunjukkan oleh diagram alir pada Gambar 3.3.



Gambar 3.3. Diagram Alir Sistem Secara Umum

### 3.4.2. Perancangan Algoritma PSO Untuk Menentukan Jumlah Cluster

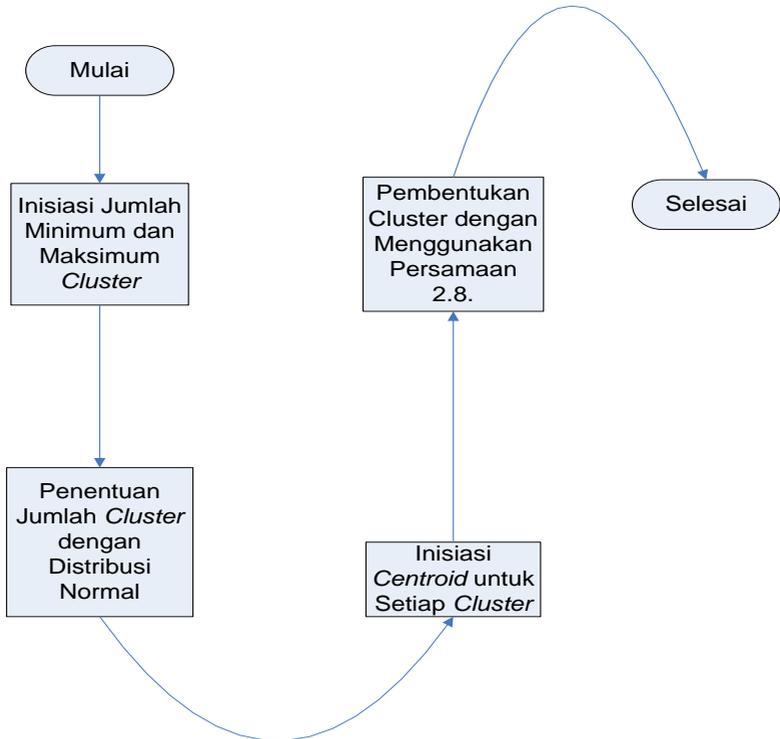
Di dalam subbab ini akan dijelaskan bagaimana algoritma PSO digunakan untuk menentukan Jumlah *Cluster*. Seperti yang dijelaskan pada subbab 2.2.1 bahwa pencarian ruang solusi adalah dengan memilih hasil *Cluster* dari partikel terbaik yang disebut sebagai *Pbest*. Adapun diagram alir untuk implementasi algoritma ini ditunjukkan oleh Gambar 3.4.



Gambar 3.4. Diagram Alir Algoritma PSO

### 3.4.3. Perancangan Algoritma K-Means Cluster Untuk Menentukan Hasil Cluster Terbaik

Di dalam subbab ini akan dijelaskan bagaimana membangun algoritma *K-Means Cluster* seperti yang telah dijelaskan pada subbab 2.2.2. Pembentukan *Cluster* ini dilakukan di dalam setiap partikel yang telah diinisiasi jumlahnya. Jumlah *Cluster* di dalam pengerjaan Tugas Akhir ini ditentukan oleh pengguna secara langsung. Nilai yang ditentukan adalah nilai rentang dari minimum hingga nilai maksimum *Cluster* yang ingin dibentuk. Adapun desain algoritma ini ditunjukkan oleh Gambar 3.5.



Gambar 3.5. Diagram Alir Untuk Algoritma *K-Means Cluster*

#### 3.4.4. Tahap Uji Coba

Tahap ini merupakan proses pembentukan model *Dynamic Cluster* dari data yang ada. Algoritma *Dynamic Cluster* yang dipakai di dalam tugas akhir ini adalah kombinasi dari algoritma PSO dan *K-Means Cluster* dengan fungsi evaluasi *Cluster* menggunakan *DB Index*. Keluaran dari tahap ini adalah jalan raya-jalan raya di dalam kelompok *Cluster* yang sama beserta nilai evaluasi *Cluster* yang terbentuk.

Mula-mula dilakukan pengambilan data jalan raya pada rentang waktu yang dipilih. Setelah data-data yang berasal dari berbagai jalan raya dipilih, data tersebut kemudian dituliskan ke dalam file yang akan diproses menjadi satu *swarm* yang berisikan beberapa partikel. Jumlah partikel yang akan diproses di dalam satu *swarm* merupakan nilai masukan. Setiap partikel akan diinisiasi jumlah *Cluster* di awal oleh nilai masukan. Nilai masukan untuk jumlah *Cluster* terdiri atas nilai rentang minimum dan maksimum.

Setiap partikel memiliki kecepatan awal dan posisi awal dimana di dalam setiap perpindahannya akan menghasilkan solusi partikel terbaik (*pbest*) dan akan sebagai kandidat solusi partikel terbaik global (*gbest*). Perhitungan untuk nilai kecepatan dan posisi partikel ditunjukkan dengan persamaan 2.3 dan 2.4. Hasil dari perhitungan ini menghasilkan sejumlah partikel di mana masing-masing partikel memiliki sejumlah *Cluster*. Nilai partikel yang hasil evaluasi *Cluster* menunjukkan nilai terkecil yang akan dipilih sebagai *gbest*. Pencarian nilai ini dilakukan setiap kali iterasi dilakukan.

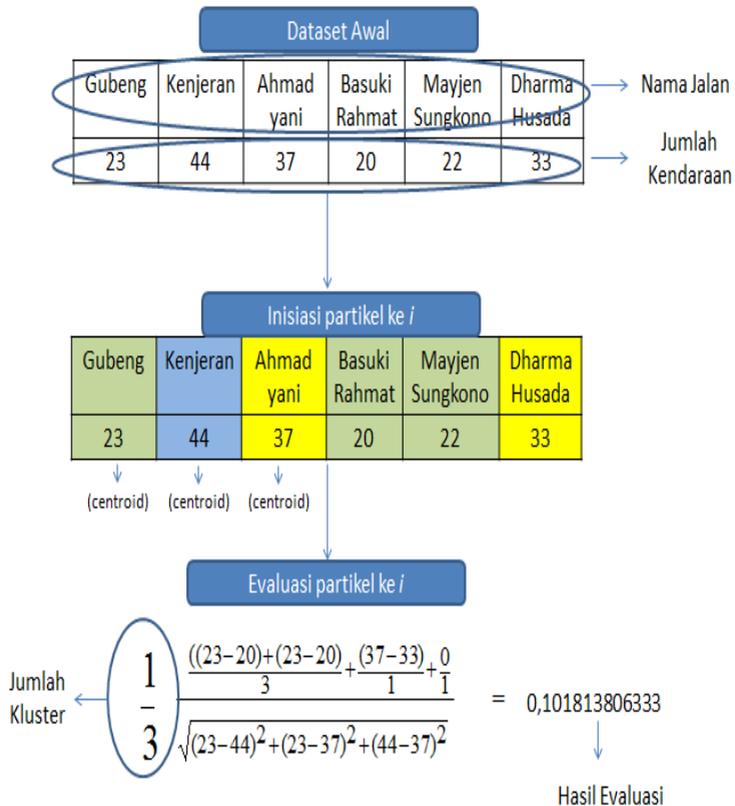
Di dalam pembentukan *Cluster* di dalam partikel terdapat proses pemilihan *centroid* yang akan menjadi pusat di dalam pengambilan keputusan. Jumlah *centroid* yang terbentuk sesuai dengan jumlah *Cluster* yang ingin dibentuk. Nilai-nilai *centroid* inilah yang kemudian menjadi patokan pencarian nilai selisih diantara dirinya dengan data lainnya. Perhitungan nilai ini dilakukan dengan metode *K-Means Cluster* menggunakan

*Euclidian Distance* seperti yang ditunjukkan oleh Persamaan 2.5. Hasil dari nilai perhitungan antara selisih yang di dapatkan kemudian disimpan untuk kemudian dilihat berdasarkan dari yang paling kecil hingga besar hasilnya.

Nilai dengan selisih terkecil untuk setiap data dengan *centroid* yang ada di dalam partikel kemudian dijadikan penanda ke mana data yang ada dikelompokkan ke dalam satu *Cluster*. Setelah terbentuk beberapa *Cluster* hasil perhitungan antara data yang ada dengan nilai titik tengahnya, maka selanjutnya dilakukan evaluasi terhadap *Cluster* yang ada di setiap partikel. Evaluasi terhadap partikel menggunakan *DB Index* dengan persamaan yang ditunjukkan oleh Persamaan 2.6. Mula-mula di dalam setiap *Cluster* dilakukan perhitungan rata-rata antara selisih nilai titik tengah *Cluster* dengan data yang ada di dalam *Cluster*. Hal ini dilakukan untuk setiap *Cluster* yang ada di dalam setiap partikel. Nilai ini kemudian disimpan sebagai pembilang. Selanjutnya dilakukan untuk perhitungan diantara titik tengah setiap *Cluster* di dalam setiap partikel. Nilai perhitungan tersebut kemudian disimpan sebagai penyebut. Hasil bagi pembilang dan penyebut inilah yang kemudian disimpan sebagai nilai evaluasi. Adapun ilustrasi terhadap perhitungan ini ditunjukkan oleh Gambar 3.7.

Pada Gambar 3.7. terdapat suatu partikel yang terdiri atas beberapa *bit* data. *Bit* data tersebut berisikan nama jalan dan jumlah kendaraan yang melintasi jalan-jalan tersebut. jalan-jalan ini kemudian bergerak dengan kecepatan seperti yang dijelaskan di dalam subbab 2.2.1. Selanjutnya dengan kecepatan tersebut posisi partikel tersebut berubah dan menghasilkan inisiasi *Cluster* sejumlah 3. Kemudian nilai jumlah *Cluster* ini dipakai untuk melakukan inisiasi *centroid* untuk setiap *Cluster* yang ada. Dengan menggunakan persamaan pada subbab 2.2.2 dihasilkan 3 buah *Cluster* pada partikel ke *i* sesuai pada Gambar 3.7. Pada ilustrasi di dalam Gambar tersebut, Gubeng, Basuki Rahmat, Mayjen Sungkono terletak pada satu *Cluster* yang sama dengan warna *Cluster* hijau. Pada *Cluster* kedua terdapat dua jalan

lainnya yakni ahmadyani dan dharmahusada dengan warna *Cluster* kuning. *Cluster* terakhir yang ada yakni kenjeran yang hanya memiliki satu anggota saja.



Gambar 3.7. Ilustrasi evaluasi salah satu partikel.

Hal terakhir dari pembentukan *Cluster* di dalam setiap partikel adalah mengevaluasi kualitas hasil *Cluster* dengan menggunakan *DB Index* seperti yang ada di dalam Persamaan 2.7. Setiap evaluasi dilakukan untuk setiap partikel di setiap

iterasinya. Hal ini dilakukan terus menerus sehingga iterasi maksimal terpenuhi.

### 3.5. Use Case Sistem

*Use case* sistem merupakan diagram kebutuhan yang menggambarkan fungsionalitas sistem dan aktor-aktornya. Berdasarkan Gambar 3.5. Di dalam pengaplikasiannya hanya ada satu aktor terlibat yaitu pengguna. Pengguna berinteraksi dengan aplikasi untuk menjalankan segala fungsionalitas sistem. *Use case* pada sistem secara umum dijelaskan pada Gambar 3.5.

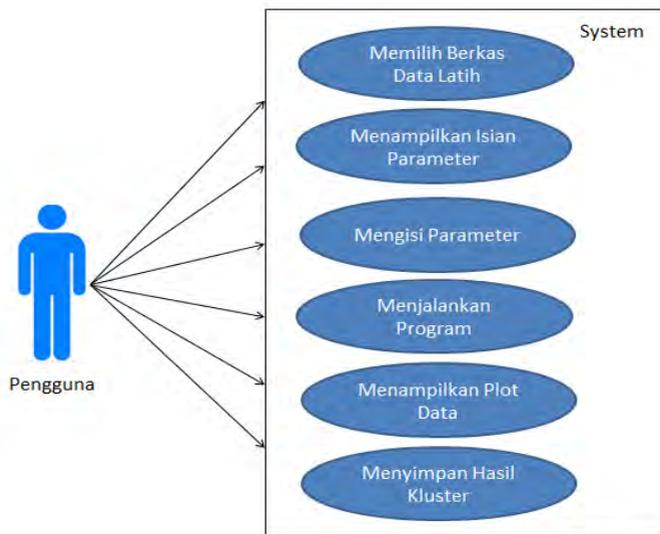
### 3.6. Perancangan Antarmuka Sistem

Di dalam Tugas Akhir ini antarmuka yang tersedia hanyalah untuk satu sisi saja yakni pengguna. Ketika pertama kali program dijalankan, pengguna dihadapkan pada satu tampilan antarmuka berbasis desktop. Adapun untuk penggunaannya ditunjukkan oleh Tabel *Use Case* di dalam Tabel 3.2.

Tabel 3.6.1. Tabel *Use Case*

No.	Kode	Nama	Keterangan
1.	UC-01	Memilih data uji coba	Pengguna dapat memilih data uji coba yang telah dipersiapkan sebelumnya.
2.	UC-02	Menampilkan Pengisian parameter	Pengguna dapat mengetahui isian parameter yang ada.
3.	UC-03	Mengisi parameter	Pengguna dapat mengisi parameter yang ada. Adapun parameter yang ada yakni jumlah minimum <i>Cluster</i> , jumlah maksimum <i>Cluster</i> , jumlah partikel dan jumlah iterasi

No.	Kode	Nama	Keterangan
4.	UC-04	Menjalankan Program	Pengguna menjalankan program setelah parameter dimasukkan.
5.	UC-05	Menampilkan plot data	Pengguna menampilkan data yang telah diproses <i>Cluster</i>
6.	UC-06	Menyimpan Hasil <i>Cluster</i>	Pengguna menyimpan hasil <i>Cluster</i>

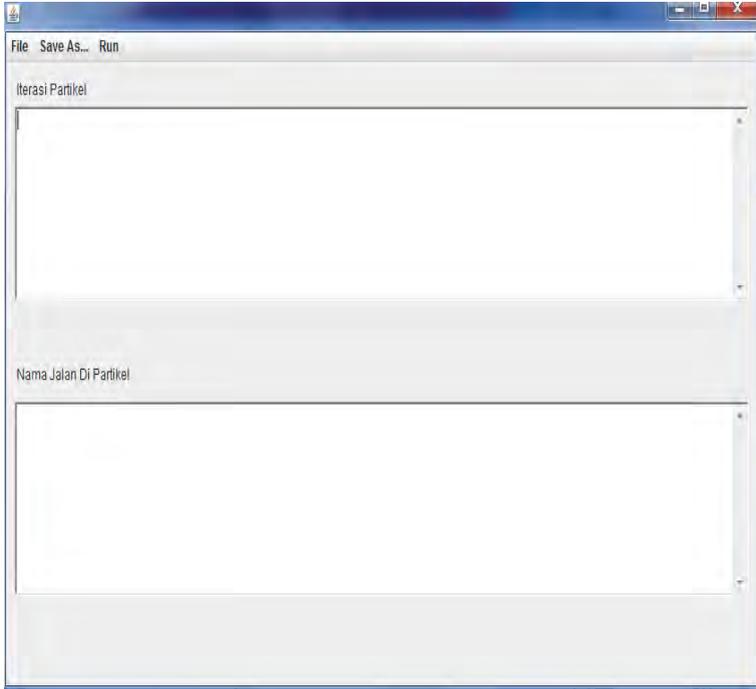


Gambar 3.5. Diagram *Use Case*

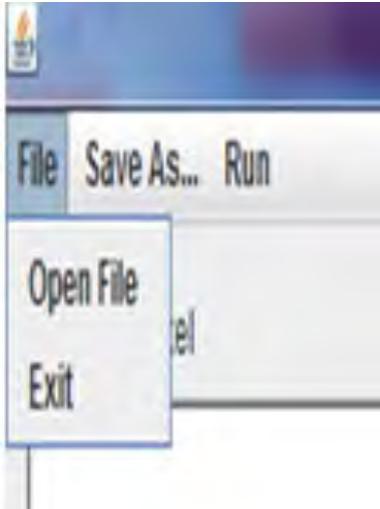
### 3.6.1. Perancangan Antarmuka Laman Utama

Perancangan antarmuka laman utama ditunjukkan pada Gambar 3.5. Terdapat sebuah tampilan aplikasi dengan beberapa *menu bar*. *Menu bar* ini lah yang memiliki beberapa fungsi utama di dalam aplikasi ini.

Antarmuka laman utama ini merupakan laman yang menyediakan berbagai fungsi dan menu di dalam aplikasi. Dimulai dari pemilihan data uji coba hingga menjalankan program dan menyimpan berkas hasil *Cluster*.



Gambar 3.8. Perancangan Antarmuka Laman Utama



Gambar 3.9. Menu Bar (2)



Gambar 3.10. Menu Bar (1)

Menu-menu yang ada di laman utama terdiri atas *file*, *Save As* dan *Run*. *File* berisikan *sub-menu* *Open File* dan *Exit*. *Open File* berfungsi untuk memilih file data uji coba yang akan dipilih. Adapun fungsi ini ditunjukkan oleh Gambar 3.10. Menu lainnya yang ada di dalam *menu bar* ini adalah *Exit*. Sesuai dengan namanya, fungsi dari menu ini adalah keluar dari program.

Pada tampilan laman utama lainnya terdapat *menu bar* *Save As*. *Menu bar* ini berfungsi untuk menyimpan data hasil *Cluster* ke dalam berkas lainnya. Adapun yang disimpan adalah hasil *Cluster* berupa nilai evaluasi serta nama-nama jalan raya yang berada di dalam satu *Cluster*.

*Menu bar* yang terakhir ada di dalam laman utama ini yakni *Running*. *Menu bar* ini berisi dua menu yakni *Running Program* dan *Setting Parameter*. *Setting Parameter* berisikan menu isian untuk mengisi parameter yang dibutuhkan untuk menjalankan program. Sementara untuk *running Program* yakni menjalankan program setelah berbagai *setting parameter* telah diisikan.

### 3.6.2. Perancangan Antarmuka Menampilkan Pengisian Parameter

Setelah menu utama ditampilkan, maka perlu untuk mengisi parameter-parameter yang dibutuhkan. Adapun parameter ini ditunjukkan oleh Gambar 3.8.

Gambar 3.11. Laman Pengisian Parameter

Komponen-komponen yang ada sebagai berikut :

- a. *TextField1* digunakan untuk mengisikan jumlah minimal *Cluster* yang diisikan.
- b. *TextField2* digunakan untuk mengisikan jumlah maksimal *Cluster* yang diisikan.
- c. *TextField3* digunakan untuk mengisikan jumlah iterasi yang akan dilakukan.
- d. *TextField4* digunakan untuk menginisiasi jumlah partikel.
- e. *TextField5* digunakan untuk mengisikan nilai *c1*.
- f. *TextField6* digunakan untuk mengisikan nilai *c2*.
- g. *TextField7* digunakan untuk mengisikan nilai *threshold*.
- h. *Button1* digunakan untuk menyimpan parameter yang telah diisikan.
- i. *Button2* digunakan untuk membatalkan isian parameter.

### **3.6.3. Perancangan Antarmuka Menampilkan Plot Data Hasil *Cluster***

Hasil evaluasi kualitas nilai *Cluster* terbaik selanjutnya akan dijadikan satu kelompok. Setiap hasil program ini dijalankan, maka *Plotting* nilai-nilai ini akan ditampilkan di dalam satu grafik.

*[Halaman ini sengaja dikosongkan]*

## **BAB IV**

### **RANCANGAN DAN IMPLEMENTASI**

Proses implementasi sistem ini dilakukan setelah melewati proses analisis dan perancangan perangkat lunak. Dalam Bab ini akan dibahas mengenai algoritma, *pseudocode* serta implementasi dari perancangan antarmuka yang terdapat dalam perangkat lunak sebagaimana telah dibahas pada Bab III.

#### **4.1 Lingkungan Implementasi**

Dalam merancang dan mengimplementasikan perangkat lunak ini, digunakan beberapa perangkat pendukung sebagai berikut

##### **4.1.1 Lingkungan Implementasi Perangkat Keras**

Spesifikasi perangkat keras yang digunakan pada lingkungan pengembangan perangkat lunak ini adalah sebagai berikut:

- Laptop Sony E Series VPCEG38FG
  - Windows 7 Home Premium 64-bit,
  - Prosesor Intel® Core(TM) i5-2450M CPU @ 2.50GHz, dengan Turbo Boost hingga 3.20 GHz.
  - RAM 4,00 GB.
  - Hard Disk Drive 500 GB.

##### **4.1.2 Lingkungan Implementasi Perangkat Lunak**

Spesifikasi perangkat lunak yang digunakan dalam pengembangan sistem adalah sebagai berikut:

- Microsoft Windows 7 Home Premium sebagai sistem operasi,
- Netbeans 7.3 sebagai *Integrated Development Enterprise* (IDE)
- Microsoft Visio 2003 untuk merancang diagram sistem.

## 4.2. Implementasi Pengambilan Data

Sebelum dilakukan proses pengelompokan data jalan raya, perlu disiapkan data-data jalan raya yang digunakan sebagai data uji coba. Beberapa data jalan raya dilakukan dengan pengamatan langsung, penelitian lainnya hingga pembangkitan secara acak dengan memperhatikan hubungan dan jalur jalan yang bersangkutan. Pengambilan data secara langsung maupun dari penelitian lainnya dilakukan untuk setiap rentang 30 menit. Selanjutnya kita tetap melakukan pengambilan data secara acak dengan distribusi yang sesuai dengan periode waktunya selama satu tahun.

Data yang dibangkitkan melalui *data generator* ini dibuat dengan rentang waktu setiap menit kedatangan kendaraan selama setahun. Data tersebut kemudian di simpan ke dalama format teks. Gambar 4.1 menunjukkan pengambilan data dengan pengamatan langsung dan Gambar 4.2 menunjukkan hasil dari pembangkitan data.



Gambar 4.1. Pengambilan data secara langsung

```

ahmadyani.txt
1 ahmadyani
2 Menit ke 0 Hasilnya: 77
3 Menit ke 1 Hasilnya: 77
4 Menit ke 2 Hasilnya: 77
5 Menit ke 3 Hasilnya: 77
6 Menit ke 4 Hasilnya: 77
7 Menit ke 5 Hasilnya: 77
8 Menit ke 6 Hasilnya: 77
9 Menit ke 7 Hasilnya: 77
10 Menit ke 8 Hasilnya: 77
11 Menit ke 9 Hasilnya: 77
12 Menit ke 10 Hasilnya: 77
13 Menit ke 11 Hasilnya: 77
14 Menit ke 12 Hasilnya: 77
15 Menit ke 13 Hasilnya: 77
16 Menit ke 14 Hasilnya: 76
17 Menit ke 15 Hasilnya: 77
18 Menit ke 16 Hasilnya: 77
19 Menit ke 17 Hasilnya: 77
20 Menit ke 18 Hasilnya: 77

```

Gambar 4.2. Hasil Dari Data Generator.

Pada Gambar 4.2. Diambil sebuah contoh jalan raya hasil pembangkitan menggunakan *data generator*. Di dalam gambar tersebut adalah pembangkitan untuk jalan raya ahmadyani untuk setiap menitnya. Menit ke-0 menunjukkan bahwa data ini merupakan data yang dibangkitkan pada jam 06.00 dan seterusnya.

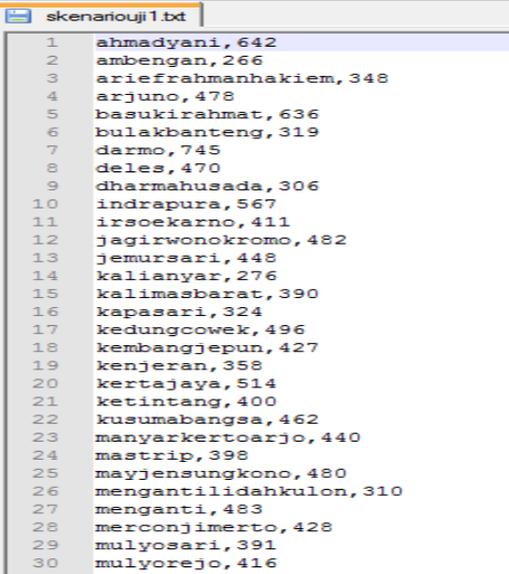
Pembangkitan data ini dilakukan untuk setiap jalan raya yang berjumlah 34. Proses ini diimplementasikan di dalam kode yang dapat dilihat pada Gambar 4.3. Setelah semua data dibangkitkan, setiap data jalan raya dimasukkan ke dalam berkas teks untuk diproses selanjutnya.

Input: -	
1	Dapatkan data jalan raya setiap menit selama satu tahun
2	Namafilename merupakan variabel penampung seluruh data jalan raya tiap menit
3	Selama $i <$ jumlah menit dalam setahun, lakukan
4	J index menit ← baris ke-i
5	Dapatkan menit dari i
6	Jika nilai i berada pada rentang waktu untuk satu distribusi, maka jumlah kendaraan bertambah sesuai parameter.
7	Jika tidak, maka iterasi berhenti

8	<p>dapatkan menit dari data Tulis data menit dan jumlah kendaraan pada berkas <math>i \leftarrow i+1</math></p> <p>Output: berkas file kendaraan per menit selama satu tahun.</p>
---	---

Gambar 4.3. *Pseudocode* Pembangkitan Data Jalan Raya

Setelah jumlah data jalan raya telah dibangkitkan seluruhnya, dilakukan pemilihan data uji coba untuk setiap jalan raya. Data uji coba merupakan data jumlah kendaraan untuk pada rentang waktu tertentu. Data uji coba inilah yang kemudian akan digunakan untuk diproses di dalam aplikasi. Adapun data uji coba ini memiliki format teks dimana antara nama jalan raya dan jumlah kendaraan dipisahkan oleh “,” koma, contohnya (ahmadyani,642).



```

skenariouji1.txt
1  ahmadyani, 642
2  ambengan, 266
3  ariefrahmanhakiem, 348
4  arjuno, 478
5  basukirahmat, 636
6  bulakbanteng, 319
7  darmo, 745
8  deles, 470
9  dharmahusada, 306
10 indrapura, 567
11 irsoekarno, 411
12 jagirwonokromo, 482
13 jemursari, 448
14 kalianyar, 276
15 kalimasbarat, 390
16 kapasari, 324
17 kedungcowek, 496
18 kembangjepun, 427
19 kenjeran, 358
20 kertajaya, 514
21 ketintang, 400
22 kusumabangsa, 462
23 manyarkertoarjo, 440
24 mastrip, 398
25 mayjensungkono, 480
26 mengantilidahkulon, 310
27 menganti, 483
28 merconjimerto, 428
29 mulyosari, 391
30 mulyorejo, 416

```

Gambar 4.4. Data Uji Coba Dalam Berkas berekstensi *.txt*

### 4.3. Implementasi Kode Sumber

Pada subbab ini akan dijelaskan mengenai implementasi algoritma *Dynamic Cluster* dengan menggunakan metode *Particle Swarm Optimization* dan *K-Means Cluster*.

#### 4.3.1. Implementasi Pembangkitan Bilangan Acak

Pada pengerjaan Tugas Akhir ini dilakukan pembangkitan bilangan acak. Hal ini dilakukan dalam rangka pengambilan data untuk keperluan lingkungan uji coba sistem. Seperti yang telah dijelaskan pada subbab 2.1.1 dan 2.1.2 tentang pembangkitan bilangan acak menggunakan dua jenis distribusi yakni *Exponential* dan *Uniform*.

Dalam penggunaan pembangkitan bilangan acak, terdapat suatu aturan terkait pembagian rentang waktu mengenai pembangkitan bilangan acak. Aturan-aturan ini ada di dalam Tabel 3.1. Selanjutnya untuk implementasi pembangkitan bilangan acak ditunjukkan dalam Kode Sumber 4.1.

```
jumkelstring = jTextField15.getText();
jumlahkendaraan = Integer.parseInt(jumkelstring);

int hasil = (int) (jumlahkendaraan+(-0.5)*(-
Math.log(Math.random())));
write.write("Menit ke "+j+" Hasilnya: "+hasil+"\n");
```

Kode Sumber 4.1. Implementasi Kode Sumber Untuk Pembangkitan Bilangan Acak Berdistribusi *Exponential*

```
jumminkelstring = jTextField2.getText();
jummakskelstring = jTextField9.getText();
jumlahminimum =Integer.parseInt(jumminkelstring);
jumlahmaksimum =Integer.parseInt(jummakskelstring);
int hasil = (int) (jumlahminimum+(jumlahmaksimum-
jumlahminimum)*Math.random());
write.write("Menit ke "+j+" Hasilnya: "+hasil+"\n");
```

Kode Sumber 4.2. Implementasi Kode Sumber Untuk Pembangkitan Bilangan Acak Berdistribusi *Uniform*

### 4.3.2. Implementasi Particle Swarm Optimization Dalam Menentukan Hasil Cluster Terbaik

Pada bagian ini akan dijelaskan implementasi algoritma *Particle Swarm Optimization*. Implementasi algoritma *Particle Swarm Optimization* ini meliputi pembentukan *swarm* dan partikel, inisiasi jumlah *Cluster* di dalam partikel, perhitungan nilai kecepatan partikel, perhitungan nilai posisi partikel dan pencarian kandidat partikel terbaik.

Adapun untuk setiap fungsi yang digunakan dalam menghitung di dalam Implementasi PSO ditunjukkan oleh Beberapa kode sumber, yakni kode sumber yang menjelaskan bagaimana inisiasi kecepatan partikel dilakukan, pengecekan terhadap nilai ambang batas, pemilihan *Pbest* dan *Gbest*.

Implementasi pada Kode Sumber 4.3 dan Kode Sumber 4.4 terlihat berbeda. Pada Kode Sumber 4.3 inisiasi dilakukan pada awal iterasi. Sementara pada Kode Sumber 4.4 inisiasi dilakukan ketika partikle sudah memasuki iterasi kedua hingga iterasi maksimum. Selain itu pada Kode Sumber 4.4 nilai kecepatan dilakukan berdasarkan jumlah partikel *i* dan saat iterasi ke *j*.

Inisiasi semacam ini juga berlaku pada posisi partikel di mana posisi awal partikel juga dilakukan dengan cara melakukan pembangkitan nilai acak berdistribusi *uniform* pada persamaan awal dan Persamaan 2.7.

```
public double randomawalvelocity()
{
    return (-1+(1-(-1))*Math.random());
}
```

Kode Sumber 4.3. Kode Sumber Implementasi Nilai Awal Kecepatan

```
velo[i][j]=
w*randomawalvelocity()+input.c1*randomuniformvelocity
()*updateY()+input.c2+randomawalvelocity()*updateY();
```

Kode Sumber 4.4. Kode Sumber Implementasi Inisiasi Nilai Kecepatan

```
public double randomawalposisi()
{
    return (-1+(1-(-1))*Math.random());
}
```

Kode Sumber 4.5. Kode Sumber Implementasi Inisiasi Nilai Awal Posisi

```
posisiawal=velo[i][j]+randomposisiawal();
```

Kode Sumber 4.6. Kode Sumber Implementasi Posisi

```
if(posisiawal>input.threshold)
    post[i][j]=1;
    int x = randomuniformcluster();
    indexcentroid=0;
    temp= new ArrayList<centroidclass>();
    for(int k =0; k<x;k++)
    {
        int dummy =(int)
        Math.abs(Math.random()*(jumdata/x));
        if(dummy==0){dummy++;}
        indexcentroid+=dummy;
        indexcentroid=indexcentroid%jumdata;
        centroid = new centroidclass();
        centroid.setNama("Jalan
        "+particle.jl.get(indexcentroid).getNamajalan(
        ));
        centroid.setJumlah(particle.jl.get(indexcentroid).jum
        lahkendaraan);
    }
    cent.set(j,temp);
}
```

Kode Sumber 4.7. Kode Sumber Implementasi Pengecekan Terhadap Nilai Ambang Batas

Di dalam Kode Sumber 4.7. dilakukan pengecekan terhadap posisi partikel. Pengecekan bertujuan untuk mengetahui apakah partikel merupakan kandidat partikel terbaik. Jika iya, maka kandidat partikel tersebut kemudian membentuk beberapa *Cluster* di dalamnya dengan nilai sesuai rentang masukan nilai minimum dan maksimum.

```
tempGbest.add(cc.distanceCentroid(cent,particle,form,
inputan,i));
indexlistIteration.add(cc.getIndexList());
```

Kode Sumber 4.8. Kode Sumber untuk Menyimpan Partikel Sebagai *Pbest*

```
if (min>g.getDBIndex())
{
    min=g.getDBIndex();
    idx=i;
}
```

Kode Sumber 4.9. Kode Sumber Pengecekan *Pbest* untuk Menjadi *Gbest*

### 4.3.3. Implementasi *K-Means Cluster* Dalam Pembentukan Cluster

Pada bagian ini akan dijelaskan bagaimana implementasi algoritma *K-Means Cluster* di dalam pengerjaan Tugas Akhir ini. Adapun implementasi yang telah dilakukan adalah tentang inisiasi jumlah minimum dan maksimum *Cluster*, Penentuan Jumlah *Cluster*, Inisiasi *Centroid* untuk setiap *Cluster* serta yang terakhir adalah pembentukan *Cluster*.

```
input.mincluster =
Integer.parseInt(textField1.getText());
input.maxcluster=
Integer.parseInt(textField2.getText());
```

Kode Sumber 4.10. Kode Sumber Inisiasi Nilai Masukan Minimum dan Maksimum *Cluster*

```

public int randomuniformcluster()
{
    return (int)
        (input.mincluster+(input.maxcluster-
        input.mincluster)*Math.random());
}

```

Kode Sumber 4.11. Kode Sumber Pembentukan *Cluster* Dengan Distribusi Normal

```

int dummy =(int) Math.abs(Math.random()*(jumdata/x));
if(dummy==0){dummy++;}
indexcentroid+=dummy;
indexcentroid=indexcentroid%jumdata;
centroid = new centroidclass();

```

Kode Sumber 4.12. Kode Sumber Inisiasi *Centroid* Setiap *Cluster*

Setelah *centroid* untuk setiap *Cluster* terbentuk maka selanjutnya adalah menghitung jarak masing-masing data terhadap *centroid* yang telah terbentuk. Setiap nilai selisih yang dihasilkan kemudian disimpan untuk diurutkan berdasarkan kedekatan antara *centroid* dan data yang sudah ada.

```

for(int i=0;i<cent.size();i++)
{
    if (cent.get(i).size()==0) continue;
    for(int j=0;j<particle.jl.size();j++)
    {
        d.clear();
        for(int k =0;k<cent.get(i).size();k++)
        {
            double jarak = Math.abs(
            cent.get(i).get(k).jumlah-
            particle.jl.get(j).jumlahkendaraan);
            Distance dd = new Distance();
            dd.setIndexCluster(k);
            dd.setJarakCluster(jarak);
            d.add(dd);
        }
        Collections.sort(d, new Distance.DistanceComparator());
        int jumlahBiteSementara =
        tempClass.get(i).get(d.get(0).getIndexCluster()).getJumlahBite();
        double
        jumlahKeseluruhanSementara=tempClass.get(i).get(d.get(0)
        .getIndexCluster()).getJumlahSeluruh();
        tempClass.get(i).get(d.get(0).getIndexCluster()).setJumlahBite(jumlahBiteSementara+1);
        tempClass.get(i).get(d.get(0).getIndexCluster()).setJumlahSeluruh(jumlahKeseluruhanSementara+d.get(0).getJarakCluster());
        indexlist.get(i).get(d.get(0).getIndexCluster()).add(j);
    }
}
}

```

Kode Sumber 4.13. Kode Sumber Pembentukan *Cluster*

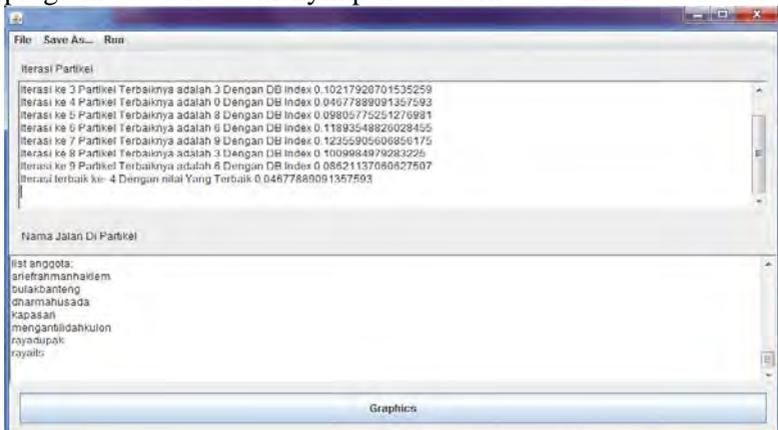
## 4.4. Implementasi Antarmuka

Pada subbab ini dijelaskan implementasi tampilan antarmuka yang telah dibahas pada subbab 3.6. Adapun antarmuka yang ada di dalam aplikasi ini terdiri atas antarmuka laman utama, antarmukan pengisian parameter, dan antarmuka menampilkan hasil *Cluster*.

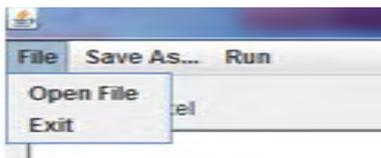
### 4.4.1. Antarmuka Laman Utama

Antarmuka laman utama ditunjukkan pada Gambar 4.8. Terdapat sebuah tampilan aplikasi dengan beberapa *menu bar*. *Menu bar* ini lah yang memiliki beberapa fungsi utama di dalam aplikasi ini.

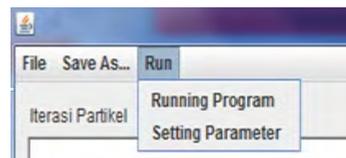
Antarmuka laman utama ini merupakan laman yang menyediakan berbagai fungsi dan menu di dalam aplikasi. Dimulai dari pemilihan data uji coba hingga menjalankan program dan menyimpan berkas hasil *Cluster*.



Gambar 4.3. Antarmuka Laman Utama



Gambar 4.5. Menu Bar (2)



Gambar 4.4. Menu Bar (1)

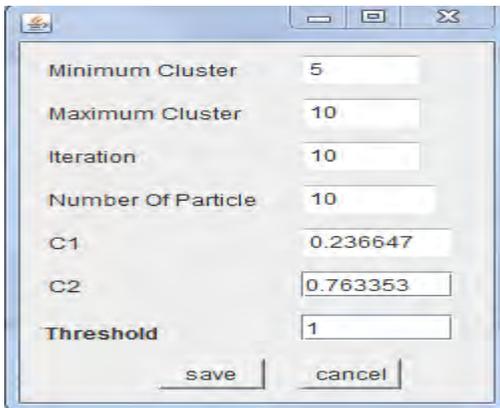
Menu-menu yang ada di laman utama terdiri atas *file, Save As dan Run*. *File* berisikan *sub-menu Open File dan Exit*. *Open File* berfungsi untuk memilih file data uji coba yang akan dipilih. Adapun fungsi ini ditunjukkan oleh Gambar 4.10. Menu lainnya yang ada di dalam *menu bar* ini adalah *Exit*. Sesuai dengan namanya, fungsi dari menu ini adalah keluar dari program.

Pada tampilan laman utama lainnya terdapat *menu bar Save As*. *Menu bar* ini berfungsi untuk menyimpan data hasil *Cluster* ke dalam berkas lainnya. Adapun yang disimpan adalah hasil *Cluster* berupa nilai evaluasi serta nama-nama jalan raya yang berada di dalam satu *Cluster*.

*Menu bar* yang terakhir ada di dalam laman utama ini yakni *Running*. *Menu bar* ini berisi dua menu yakni *Running Program dan Setting Parameter*. *Setting Parameter* berisikan menu isian untuk mengisi parameter yang dibutuhkan untuk menjalankan program. Sementara untuk *running Program* yakni menjalankan program setelah berbagai *setting parameter* telah diisikan.

#### 4.4.2. Antarmuka Menampilkan Pengisian Parameter

Setelah menu utama ditampilkan, maka perlu untuk mengisi parameter-parameter yang dibutuhkan. Adapun parameter ini ditunjukkan oleh Gambar 4.13.



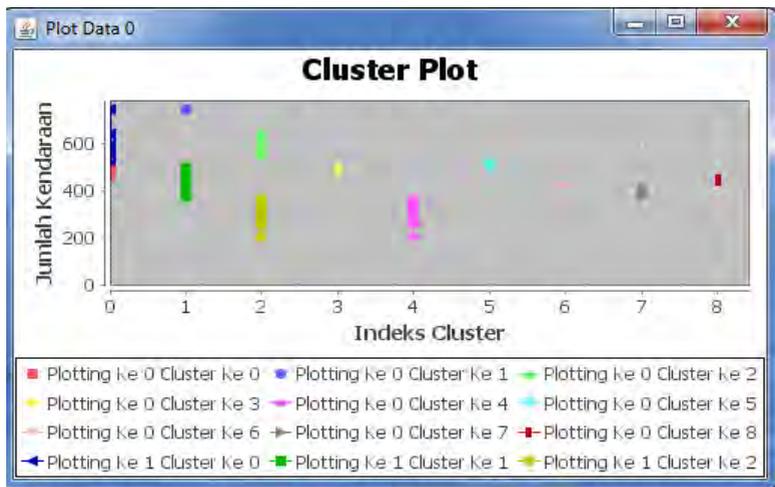
Minimum Cluster	5
Maximum Cluster	10
Iteration	10
Number Of Particle	10
C1	0.236647
C2	0.763353
Threshold	1
<input type="button" value="save"/> <input type="button" value="cancel"/>	

Gambar 4.11. Laman Pengisian Parameter

Parameter yang ada di Gambar 4.11 berisikan beberapa isian, yakni *Minimum Cluster* untuk isian jumlah minimum *Cluster* yang mungkin terbentuk, untuk *Maximum Cluster* digunakan untuk isian jumlah maksimum *Cluster*. *Iteration* merupakan jumlah iterasi yang akan dilakukan dan *Number Of Particle* merupakan jumlah partikel yang akan dibentuk sebagai jumlah partikel dalam *swarm*.

#### 4.4.3. Antarmuka Menampilkan Hasil *Cluster*

Nilai keluaran dari aplikasi ini adalah hasil *Cluster* dengan evaluasi *Cluster* terbaik. Adapun antarmuka menampilkan hasil *Cluster* ini ditunjukkan pada Gambar 4.12.



Gambar 4.6. Antarmuka Menampilkan Hasil *Cluster*

Di dalam antarmuka pada Gambar 4.12 ini adalah menampilkan hasil *Plotting* data sesuai dengan indeks *Cluster* berdasarkan jumlah kendarannya masing-masing. Selain informasi mengenai indeks *Cluster* saat *Plotting* data, legenda dari grafik tersebut menunjukkan indeks program yang telah dijalankan. Di dalam legenda grafik *Plotting* menunjukkan indeks

*running* aplikasi. Misalkan aplikasi telah dijalankan sebanyak 2 kali maka *plotting* akan menghasilkan 2 indeks *plotting* yakni 0 dan 1. Sementara Pada setiap *Cluster* yang ada, antara satu *Cluster* dengan *Cluster* lainnya memiliki perbedaan warna. Pada sumbu Y grafik menunjukkan indeks jumlah kendaraan dan sementara sumbu X menunjukkan indeks *Cluster*.

## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini dijelaskan mengenai rangkaian uji coba dan evaluasi yang dilakukan terhadap sistem yang dibuat. Pembahasan yang dipaparkan meliputi lingkungan uji coba, data uji coba, skenario uji coba, hasil uji coba, dan evaluasi.

#### **5.1. Lingkungan Uji Coba**

Proses uji coba terhadap sistem ini dilakukan menggunakan *notebook* Sony Vaio E Series VPCEG38FG dengan spesifikasi seperti berikut:

- Sistem Operasi : Windows 7 (Home Premium 64 bit),
- Processor : 2.5 GHz Scorpion
- Hard Disk Drive : 500 GB,
- RAM : 4 GB.

Selain itu, dalam uji coba ini juga digunakan perangkat lunak pembantu yaitu *microsoft excel*, *notepad* dan *notepad++*.

Ujicoba dilakukan dengan menggunakan 40 nama jalan raya yang ada di kota Surabaya. Jumlah dari data jalan raya tersebut didapatkan dari pengamatan langsung, penelitian lainnya dan juga pembangkitan bilangan acak. Adapun nama jalan raya yang digunakan di dalam uji coba ini ialah ahmadyani, ambengan, ariefrahmanhakiem, arjuno, basukirahmat, bulakbanteng, darmo, deles, dharmahusada, indrapura, irsoekarno, jagirwonokromo, jemursari, kaliyanar, kalimasbarat, kapasari, kedungcowek, kembangjepun, kenjeran, kertajaya, ketintang, kusumabangsa, manyarkertoarjo, mastrip, mayjensungkono, mengantilihakulon, menganti, merconjimerto, mulyosari, mulyorejo, ngagel, nginden, prof.dr.mestopo, rajawali, randu, rayadupak, rayaits, rayamenur, rungkutindustri dan tambaksari.

## 5.2. Data Uji Coba

Data uji coba yang digunakan adalah data hasil pengamatan langsung, penelitian terkait kondisi jalan raya dan prediksi kondisi jalan raya dengan memperhatikan hubungan antar jalan raya yang terhubung.

## 5.3. Skenario & Hasil Uji Coba

Pada subbab ini dijelaskan skenario uji coba yang dilakukan dan hasil yang didapatkan. Uji coba terdiri tiga uji coba, yakni:

- Pengujian penentuan nilai konstanta dan parameter
- Pengujian nilai ambang batas posisi partikel
- Pengujian kualitas parameter *Cluster*

### 5.3.1. Pengujian Penentuan Parameter Nilai Kenaikan Kecepatan Dan Posisi

Pada bagian ini dilakukan uji coba terhadap parameter yang ada di dalam fungsi objektif untuk menghitung nilai kenaikan kecepatan, dalam hal ini direpresentasikan sebagai  $c1$  dan posisi partikel dalam hal ini direpresentasikan sebagai  $c2$ . Nilai untuk penjumlahan keduanya bernilai 1. Pengujian ini menggunakan skenario pembangkitan bilangan acak dengan jenis kendaraan mobil pada pagi hari ketika hari efektif kerja. Adapun data yang digunakan ditunjukkan oleh Tabel 5.1. Jumlah kendaraan yang melintasi jalan raya ini didapat dengan pembangkitan bilangan acak dan pemilihan rentang waktu yang juga secara acak.

Perlu diketahui bahwasannya perhitungan nilai antara  $c1$  dan  $c2$  ini dilakukan sebanyak 50 kali. Maka dari itu, pengecekan nilai keduanya dilakukan di antara nilai acak 0 dan 1. Nilai yang dipilih nantinya merupakan nilai yang memiliki nilai evaluasi paling minimum.

Tabel 5.3.1. Data Uji Coba Skenario Mobil Pada Pagi Hari  
Dengan Rentang Waktu 10.10-10.20.

<b>Nomer</b>	<b>Nama Jalan</b>	<b>Jumlah Kendaraan</b>
1	Ahmadyani	642
2	Ambengan	266
3	Arief Rahman Hakim	348
4	Arjuno	478
5	Basuki Rahmat	636
6	Bulak Banteng	319
7	Darmo	745
8	Deles	470
9	Dharmahusada	306
10	Dharmawangsa	298
11	Indrapura	567
12	Ir.Soekarno	411
13	Jagir Wonokromo	482
14	Kalianyar	276
15	Kalimas Barat	390
16	Kapasari	324
17	Kedung Cowek	496
18	Kembang Jepun	427
19	Kenjeran	358
20	Kertajaya	514
21	Ketintang	400
22	Kusuma Bangsa	462
23	Manyar Kertoarjo	440
24	Mastrip	398
25	Mayjen Sungkono	480
26	Menganti Lidah Kulon	480
27	Menganti	483
28	Mercon Jimerto	428

Nomer	Nama Jalan	Jumlah Kendaraan
29	Mulyosari	391
30	Mulyorejo	416
31	Ngagel	433
32	Nginden	555
33	Prof.Dr.Mestopo	514
34	Rajawali	488
35	Randu	277
36	Raya Dupak	332
37	Raya ITS	342
38	Raya Menur	408
39	Rungkut Industri	383
40	Tambaksari	270

Adapun hasil evaluasi dari pengujian nilai konstanta untuk kenaikan nilai posisi partikel ( $c1$ ) dan kenaikan nilai posisi ( $c2$ ) ditunjukkan oleh Tabel 5.2.

Tabel 5.3.2. Tabel Hasil Evaluasi Konstanta Kenaikan Kecepatan dan Posisi

Nomer	Nilai Cluster Miniumm	Nilai Cluster Maksimum	Nilai		Hasil Evaluasi
			$c1$	$c2$	
1	5	10	0,327649	0,672351	0,1245
2	5	10	0,251747	0,748253	0,1315
3	5	10	0,17559	0,82441	0,1244
4	5	10	0,623258	0,376742	0,1240
5	<b>5</b>	<b>10</b>	<b>0,236647</b>	<b>0,763353</b>	<b>0,1048</b>
6	5	10	0,418479	0,581521	0,1264
7	5	10	0,561753	0,438247	0,1365
8	5	10	0,329709	0,670291	0,1228
9	5	10	0,083125	0,916875	0,1161

Nomer	Nilai <i>Cluster</i> Miniumm	Nilai <i>Cluster</i> Maksimum	Nilai		Hasil Evaluasi
			<i>c1</i>	<i>c2</i>	
10	5	10	0,238416	0,761584	0,1125

Seperti yang ditampilkan oleh Tabel 5.2 nilai evaluasi parameter terkecil ada pada nilai ketika *c1* bernilai 0,236647 dan *c2* bernilai 0,763353.

### 5.3.2. Pengujian Nilai Ambang Batas Posisi Partikel

Pada bagian ini dijelaskan bagaimana pengujian mengenai nilai ambang batas posisi partikel. Nilai ini memiliki peranan penting dalam menentukan hasil optimal dari nilai evaluasi *Cluster* seperti yang dijelaskan pada subbab 4.3.2. Pada pengujian ini dilakukan 50 kali percobaan yang melibatkan nilai diantara rentang nilai 0 hingga 1. Nilai rentang yang dipakai adalah 5 untuk minimum *Cluster* dan 10 untuk maksimum *Cluster*.

Pemilihan nilai terbaik di dalam pengujian ini sama persis dengan evaluasi nilai pada uji skenario pertama seperti yang dijelaskan di dalam subbab 5.3.1. Data uji coba yang digunakan juga sama seperti yang ditunjukkan pada Tabel 5.1. Hasil uji coba pengujian nilai ini ditunjukkan oleh Tabel 5.3.

Tabel 5.3.3. Tabel Pengujian Nilai Ambang Batas

Nomer	Nilai Minimum <i>Cluster</i>	Nilai Maksimum <i>Cluster</i>	Nilai Pengujian Ambang Batas	Nilai Evaluasi
1	5	10	0,23793	0.0694
2	5	10	1	0.0792
3	5	10	0,908567	0.0840

Nomer	Nilai Minimum Cluster	Nilai Maksimum Cluster	Nilai Pengujian Ambang Batas	Nilai Evaluasi
4	5	10	0,120745	0.0762
5	5	10	0,529116	0.0613
6	5	10	0,649316	0.0722
7	<b>5</b>	<b>10</b>	<b>0,73172</b>	<b>0.0652</b>
8	5	10	0,739758	0.0702
9	5	10	0,424013	0.0622
10	5	10	0,228878	0.0757

### 5.3.3. Pengujian Kualitas Fungsi Evaluasi Parameter Cluster

Pada bagian ini dilakukan uji coba terhadap parameter yang digunakan untuk setiap berkas skenario uji coba yang telah disiapkan dan terhadap nilai evaluasi dari setiap Cluster yang dihasilkan.

Ujicoba dilakukan di setiap jalan raya dengan membaginya ke dalam empat daerah waktu yakni pagi, siang, sore dan malam hari. Seperti yang dijelaskan pada bab 2.1, jalan raya yang diamati adalah jalan raya dengan pengamatan langsung menggunakan sensor, penelitian lain terkait kondisi jalan raya di kota Surabaya serta pembangkitan bilangan acak dari data yang sudah ada sebelumnya ataupun untuk jalan-jalan yang saling terhubung antara satu dengan yang lain.

Di dalam uji coba pada parameter yang pertama, akan dilakukan dengan menggunakan rentang waktu setiap 10 menit. Adapun data uji coba yang akan digunakan seperti yang telah dijelaskan pada subbab 5.3.1. Jenis kendaraan yang akan diuji ada dua macam, yakni sepeda motor dan mobil. Adapun Tabel mengenai skenario uji coba ini ditunjukkan oleh Tabel 5.4. Sementara untuk skenario uji coba evaluasi kedua ditunjukkan oleh Tabel 5.5.

Tabel 5.3.4. Skenario Uji Coba Evaluasi *Cluster* Pertama

Nama Subskenario	Jenis Kendaraan	Rentang Waktu
Skenario Pagi	Mobil	06.00-07.00
Skenario Siang	Mobil	12.00-13.00
Skenario Sore	Mobil	18.00-19.00
Skenario Malam	Mobil	22.30-23.30
Skenario Pagi	Motor	06.00-07.00
Skenario Siang	Motor	12.00-13.00
Skenario Sore	Motor	18.00-19.00
Skenario Malam	Motor	22.30-23.30

Tabel 5.3.5.Skenario Uji Coba Evaluasi *Cluster* Kedua

Nama Subskenario	Jenis Kendaraan	Rentang Waktu
Skenario Pagi	Mobil	10.10-10.20
Skenario Siang	Mobil	13.10-13.20
Skenario Sore	Mobil	16.20-16.30
Skenario Malam	Mobil	20.10-20.20
Skenario Pagi	Motor	10.10-10.20
Skenario Siang	Motor	13.10-13.20
Skenario Sore	Motor	16.20-16.30
Skenario Malam	Motor	20.10-20.20

Pada Tabel 5.4 dan 5.5 diberikan data Tabel yang memiliki jumlah baris yang sama namun rentang waktu yang berbeda. Jika pada Tabel 5.5 rentang waktu yang diberikan adalah 10 menit, maka pada Tabel 5.4 rentang waktu yang akan diuji sebesar 60 menit. Perbedaan yang lain adalah pada Tabel 5.4 data yang diambil ketika hari libur kerja, sementara untuk Tabel 5.5 data yang diambil ketika hari efektif kerja.

Adapun hasil dari ujicoba ini ditunjukkan oleh Tabel 5.6. Setiap Subskenario dilakukan proses uji hasil *Cluster* dengan menggunakan parameter dan nilai ambang batas yang diperoleh dari dua percobaan sebelumnya. Hasil percobaan pada skenario

ini dilampirkan pada bagian lampiran pada tabel TL1 hingga TL 16 dan GL 1 hingga GL16 di bagian akhir buku ini.

Tabel 5.3.6. Tabel Skenario Uji Coba

<b>Nomer</b>	<b>Nama Skenario</b>	<b>Jumlah Cluster Terbentuk</b>	<b>Nilai Evaluasi Hasil Cluster</b>
1	Subskenario 1 di dalam skenario 1	9	0.1029
2	Subskenario 2 di dalam skenario 1	7	0.1333
3	Subskenario 3 di dalam skenario 1	9	0.08127
4	Subskenario 4 di dalam skenario 1	7	0.0833
5	Subskenario 5 di dalam skenario 1	5	0.1129
6	Subskenario 6 di dalam skenario 1	5	0.0906
7	Subskenario 7 di dalam skenario 1	9	0.1233
8	Subskenario 8 di dalam skenario 1	8	0.1027
9	Subskenario 1 di dalam skenario 2	8	0.0631
10	Subskenario 2 di dalam skenario 2	5	0.0733
11	Subskenario 3 di dalam skenario 2	8	0.0864
12	Subskenario 4 di dalam skenario 2	6	0.0841
13	Subskenario 5 di dalam skenario 2	6	0.0444
14	Subskenario 6 di dalam skenario 2	9	0.0545

<b>Nomer</b>	<b>Nama Skenario</b>	<b>Jumlah Cluster Terbentuk</b>	<b>Nilai Evaluasi Hasil Cluster</b>
15	Subskenario 7 di dalam skenario 2	6	0.0858
16	Subskenario 8 di dalam skenario 2	6	0.1078

#### **5.4. Evaluasi**

Dari percobaan yang telah dilakukan diketahui bahwa Pada uji skenario pertama, didapatkan bahwa nilai konstanta terbaik untuk kenaikan kecepatan dan posisi untuk jumlah data 40, Cluster minimum bernilai 5 dan Cluster maksimum bernilai 10, dihasilkan nilai terbaiknya yakni 0,236647 dan 0,763353 dengan nilai evaluasi terbaiknya 0,1048. Selanjutnya pada uji skenario kedua, didapatkan bahwa nilai untuk ambang batas posisi partikel terbaik untuk jumlah data 40, jumlah minimum Cluster 5 dan maksimum Cluster 10, dihasilkan nilai ambang batas sebesar 0,73172 dan nilai evaluasi partikel terbaiknya 0,0652. Dan pada uji skenario ketiga, dengan melibatkan 40 jalan raya dan pengujian Cluster berdasarkan rentang waktu yang berbeda-beda, didapatkan informasi bahwa setiap jalan raya tidak selalu berada pada Cluster yang sama untuk setiap waktu.

Selain itu untuk *plotting* data hasil Cluster terbaik, grafik yang dihasilkan masih saling menimpali. Harusnya antara satu skenario dengan skenario lainnya berdiri sendiri dan tidak saling menimpali.

*[Halaman ini sengaja dikosongkan]*

## **BAB VI**

### **KESIMPULAN DAN SARAN**

Dalam bab terakhir ini dijelaskan kesimpulan yang didapat dari pengerjaan Tugas Akhir beserta saran-saran yang dapat dipertimbangkan untuk pengembangan atau penelitian lebih lanjut.

#### **6.1. Kesimpulan**

Implementasi *Dynamic Cluster* dalam studi kasus pengelompokan jalan raya dapat dilakukan dengan baik. Hal ini terlihat di dalam uji coba yang telah dilakukan melalui tiga skenario. Adapun kesimpulan yang dihasilkan dari pengerjaan Tugas Akhir ini adalah sebagai berikut:

1. Pada uji skenario pertama, didapatkan bahwa nilai konstanta terbaik untuk kenaikan kecepatan dan posisi hasilnya telah cukup signifikan untuk menghasilkan *Cluster* terbaik.
2. Pada uji skenario kedua, didapatkan bahwa nilai untuk ambang batas posisi partikel terbaik telah cukup signifikan untuk menghasilkan *Cluster* terbaik.
3. Pada uji skenario ketiga, dengan melibatkan 40 jalan raya dan pengujian *Cluster* berdasarkan rentang waktu yang berbeda-beda, didapatkan informasi bahwa setiap jalan raya tidak selalu berada pada *Cluster* yang sama untuk setiap waktu.

## 6.2. Saran

Adapun Saran yang dapat diberikan dalam rangka penelitian di masa mendatang adalah sebagai berikut:

1. Data uji coba di mana dalam hal ini data jalan raya sebaiknya ditambahkan .
2. Dilakukan pengamatan dengan rentang waktu yang lebih lama dan jalan yang lebih banyak untuk memperoleh hasil yang lebih nyata di lapangan.
3. Memberikan Informasi tambahan terkait pengelompokkan jalan raya.

## LAMPIRAN

Bagian ini merupakan lampiran sebagai dokumen pelengkap dari Tugas Akhir. Adapun lampiran ini berisikan tabel dan gambar yang dihasilkan pada uji skenario tiga.

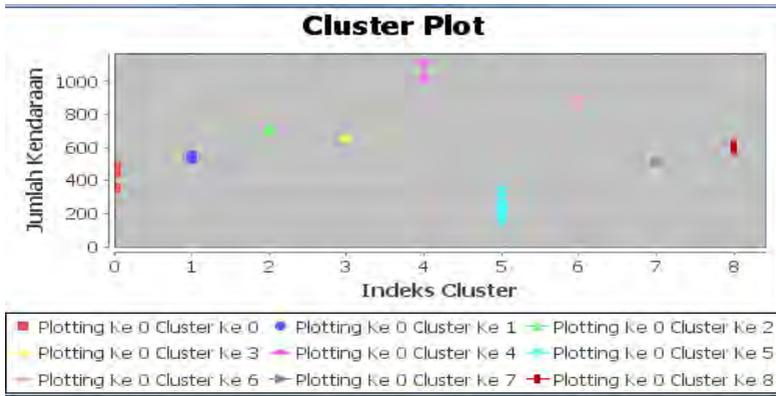
### A. Tabel Data Uji Coba Pada Skenario Tiga

Di dalam skenario uji parameter pertama, terdapat delapan subskenario uji coba. Untuk subskenario uji pertama, hasilnya ditunjukkan oleh Tabel TL.1 dan Gambar GL.1.

Tabel TL.1. Tabel Hasil *Cluster* Jalan Raya pada Pagi Hari Efektif untuk Jenis Kendaraan Mobil

Nilai <i>DB Index</i> terbaik	Indeks <i>Cluster</i>	Nama Jalan
0.1029	0	Ambengan, Arif rahman hakim, Arjuno, Kusuma Bangsa, Mastrip, Ngagel, Raya Dupak, Rungkut Industri, Tambaksari
	1	Basuki Rahmat, Indrapura, Kembang Jepun, Manyar Kertoarjo
	2	Deles, Jemursari
	3	Ir Soekarno, Prof. Dr. Mestopo
	4	Ahmadyani, Dharmahusada, Jagir, Wonokromo
	5	Bulak Banteng, Kalianyar, Kalimas Barat, Ketintang, Menganti Lidah Kulon, Menganti, Merconjimerto, Mulyosari, Mulyorejo, Nginden, Rajawali, Randu, Raya ITS, Raya Menur
	6	Kedungcowek, Mayjen Sungkono
	7	Kenjeran
	8	Darmo, Kapasari, Kertajaya

Terdapat delapan *Cluster* yang terbentuk dengan rentang *Cluster* diantara 5-10 *Cluster*. Jumlah pemilihan rentang *Cluster* ini didasarkan pada jumlah jalan yang ada yakni 40 jalan raya dengan harapan seluruhnya terbagi atas delapan *Cluster* dimana masing masing *Cluster* berisi lima jalan raya. Pun ini juga berlaku untuk setiap subskenario yang ada dan yang akan dijalankan.



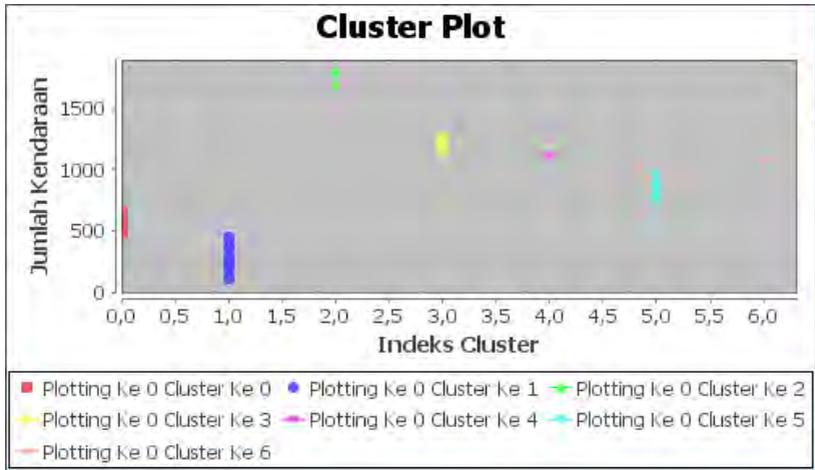
Gambar GL.1. *Plotting* Hasil *Cluster* Jalan Raya pada Pagi Hari Efektif untuk Jenis Kendaraan Mobil

Selanjutnya akan ditunjukkan Tabel dan Gambar untuk seluruh subskenario yang ada di dalam skenario pertama secara berturut-turut yakni Tabel TL.2, Tabel TL.3, Tabel TL.4, Tabel TL.5, Tabel TL.6, Tabel TL.7, dan Tabel TL.8 serta Gambar GL.2, Gambar GL.3, Gambar GL.4, Gambar GL.5, Gambar GL.6, Gambar GL.7 dan Gambar GL.8.

Perbedaan antara masing-masing Tabel dan Gambar yakni untuk Tabel TL.1, Tabel TL.2, Tabel TL.3 dan Tabel TL.4 serta Gambar GL.1, Gambar GL.2, Gambar GL.3 dan Gambar GL.4 jenis kendaraan mobil sementara untuk Tabel TL.5, Tabel TL.6, Tabel TL.7, dan Tabel TL.8 serta Gambar GL.5, Gambar GL.6, Gambar GL.7, dan Gambar GL.8 untuk jenis kendaraan motor.

Tabel TL.1. Tabel Hasil *Cluster* Jalan Raya pada Siang Hari Efektif untuk Jenis Kendaraan Mobil

Nilai <i>DB Index</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.1333	0	Arief Rahman Hakim, Arjuno, Basuki Rahmat, Kembang Jepun, Kertajaya Prof.dr.mestopo, Raya Dupak, Rungkut Industri
	1	Bulakbanteng, Kalimasbarat, Kenjeran, Ketintang, Kusuma Bangsa, Manyar Kertoarjo, Mastrip, Menganti Lidah Kulon, Menganti, Mercon Jimerto, Mulyosari, Mulyorejo, Ngagel, Nginden Rajawali, Randu, Raya ITS, Raya Menur, Tambaksari
	2	Ahmadyani, Darmo
	3	Ambengan, Dharmahusada, Indrapura, Jemursari
	4	Irsoekarno
	5	Deles, Kalianyar, Kapasari, Mayjen Sungkono
	6	Jagir Wonokromo, Kedungcowek

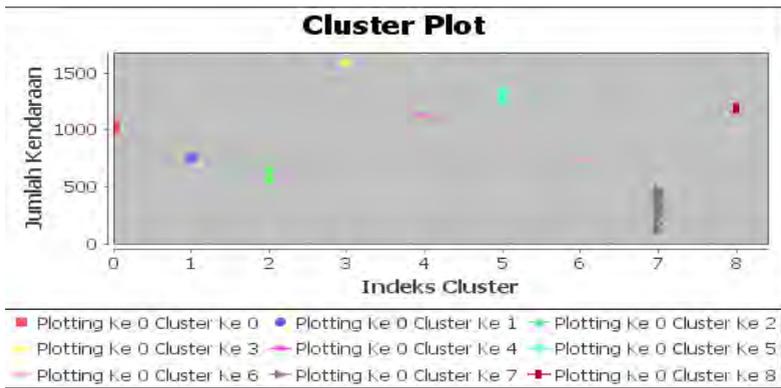


Gambar GL.2. *Plotting Hasil Cluster Jalan Raya pada Siang Hari Efektif untuk Jenis Kendaraan Mobil*

Tabel GL.2. *Tabel Hasil Cluster Jalan Raya pada Sore Hari Efektif Untuk Jenis Kendaraan Mobil*

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.08127	0	Ambengan, Bulak Banteng
	1	Arief Rahman Hakim
	2	Arjuno, Prof.Dr.Mestopo
	3	Darmo, Kedungcowek
	4	Dharmahasada, Indrapura, Jemursari
	5	Basuki Rahmat, Deles, Ir.Soekarno, Mayjen Sungkono
	6	Kalianyar
7	Kalimas Barat, Kembang Jepun, Kenjeran, Kertajaya, Ketintang, Kusuma Bangsa, Manyar Kertoarjo, Mastrip, Menganti Lidah Kulon, Menganti, Mercon Jimerto	

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
		Mulyosari, Mulyorejo, Ngagel Nginden, Rajawali, Randu, Raya Dupak, Raya ITS, Raya Menur, Rungkut Industri, Tambaksari
	8	Ahmadyani, Jagir Wonokromo Kapasari,

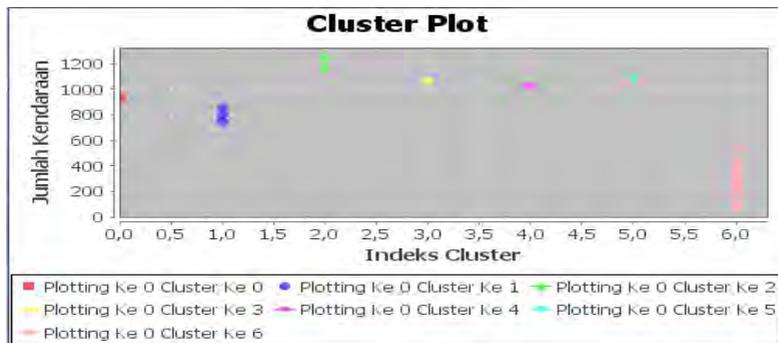


Gambar GL.3. *Plotting* Hasil *Cluster* Jalan Raya pada Sore Hari Efektif untuk Jenis Kendaraan Mobil

Tabel TL.3. Tabel Hasil *Cluster* Jalan Raya Pada Malam Hari Efektif untuk Jenis Kendaraan Mobil

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0833	0	Basuki Rahmat
	1	Ahmadyani, Ambengan, Dharmahasada, Ir. Soekarno, Kapasari
	2	Darmo, Indrapura

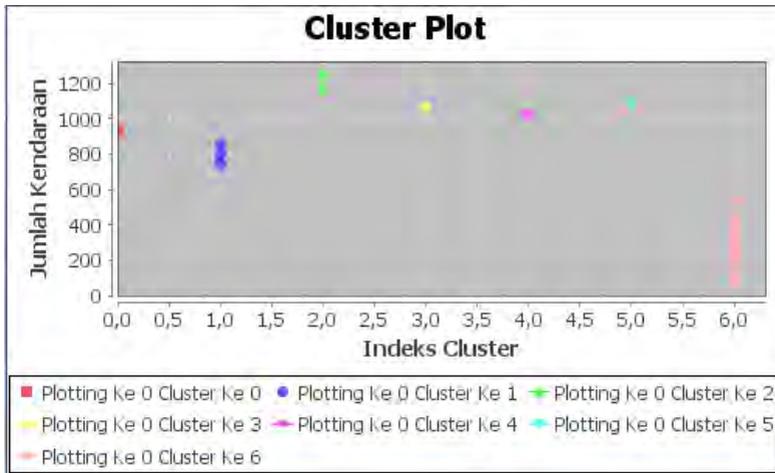
Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
	3	Jagir Wonokromo
	4	Deles, Jemursari
	5	Kedungcowek
	6	Arief Rahman Hakim, Arjuno, Bulak Banteng, Kalianyar, Kalimas Barat, Kembang Jepun, Kenjeran, Kertajaya, Ketintang, Kusuma Bangsa, Manyar Kertoarjo, Mastrip, Mayjen Sungkono, Menganti Lidah Kulon, Menganti, Mercon Jimerto, Mulyosari, Mulyorejo, Ngagel, Nginden, Prof.Dr.Mestopo, Rajawali, Raya ITS, Raya Menur, Rungkut Industri, Tambaksari



Gambar GL.4. *Plotting Hasil Cluster Jalan Raya Pada Malam Hari Efektif Untuk Jenis Kendaraan Mobil*

Tabel TL.4. Tabel Hasil *Cluster* Jalan Raya Pada Pagi Hari Efektif Untuk Jenis Kendaraan Motor

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.1129	0	Arjuno, Bulak Banteng, Kalianyar, Kembang Jepun, Ketintang, Menganti Lidah Kulon, Menganti, Mercon Jimerto,Randu
	1	Deles, Kalimas Barat, Kertajaya, Kusuma Bangsa, Mulyosari
	2	Ahmadyani, Ambengan, Arief Rahman Hakim, Basuki Rahmat, Dharmahusada, Ir.Soekarno, Jagir Wonokromo, Kedungcowek, Kenjeran, Manyar Kertoarjo, Mastrip, Mayjen Sungkono, Nginden, Raya Dupak, Raya ITS, Rungkut Industri
	3	Jemursari, Ngagel, Rajawali
	4	Darmo, Indrapura Kapasari, Mulyorejo, Prof.Dr.Mestopo, Raya Menur Tambaksari

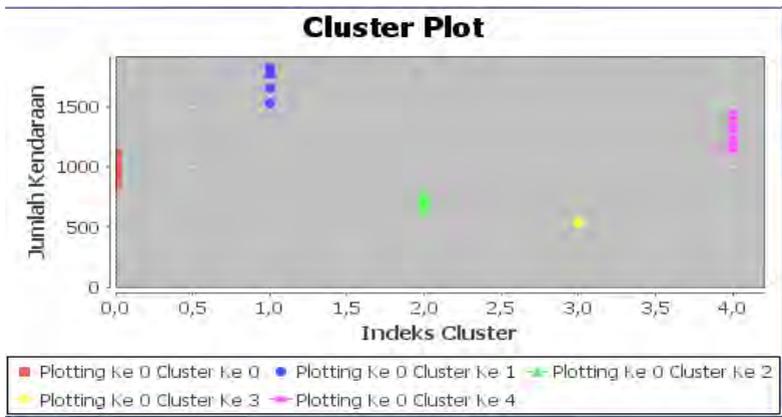


Gambar GL.5. *Plotting Hasil Cluster Jalan Raya pada Pagi Hari Efektif untuk Jenis Kendaraan Motor*

Tabel TL.5. *Tabel Hasil Cluster Jalan Raya pada Siang Hari Efektif untuk Jenis Kendaraan Motor*

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0906	0	Ahmadyani, Arief Rahman Hakim, Arjuno, Bulak Banteng, Deles, Jemursari, Ketintang, Manyar Kertoarjo, Mulyorejo, Tambaksari
	1	Darmo, Dharmahusada, Indrapura, Jagir Wonokromo, Raya Dupak
	2	Ambengan, Kalianyar, Kalimas Barat, Menganti Lidah Kulon, Menganti, Mercon Jimerto, Mulyosari, Ngagel
	3	Kapasari, Kembang Jepun

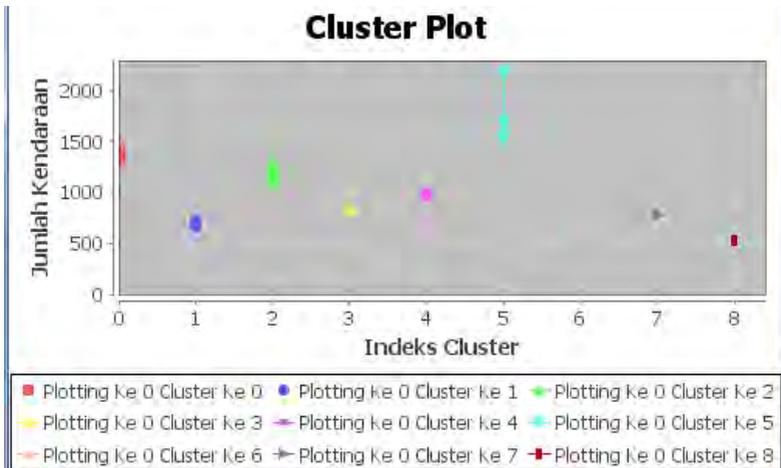
Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
	4	Basuki Rahmat, Ir.Soekarno, Kedungcowek, Kenjeran, Kertajaya, Kusuma Bangsa, Mastrip, Mayjen Sungkono, Nginden, Prof.Dr.Mestopo, Rajawali, Randu, Raya ITS, Raya Menur, Rungkut Industri



Gambar GL.6. *Plotting Hasil Cluster Jalan Raya pada Siang Hari Efektif untuk Jenis Kendaraan Motor*

Tabel TL.6. Tabel Hasil *Cluster* Jalan Raya pada Sore Hari Efektif untuk Jenis Kendaraan Motor

Nilai DB Indek <i>Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.1233	0	Ambengan, Ir.Soekarno, Kedungcowek, Raya Dupak
	1	Arjuno, Kalianyar, Mastrip
	2	Arif Rahman Hakim, Darmo, Dharmahusada, Kertajaya, Ketintang, Kusuma Bangsa, Manyar Kertoarjo, Menganti Lidah Kulon, Menganti, Mulyosari, Mulyorejo, Nginden, Prof.Dr.Mestopo, Rajawali, Raya Menur, Rungkut Industri
	3	Deles, Ngagel
	4	Indrapura, Jemursari, Randu
	5	Ahmadyani, Basuki Rahmat, Bulak Banteng, Jagir Wonokromo, Kenjeran, Mayjen Sungkono, Raya ITS
	6	Kalimas Barat
	7	Kapasari, Mercon Jimerto
	8	Kembang Jepun

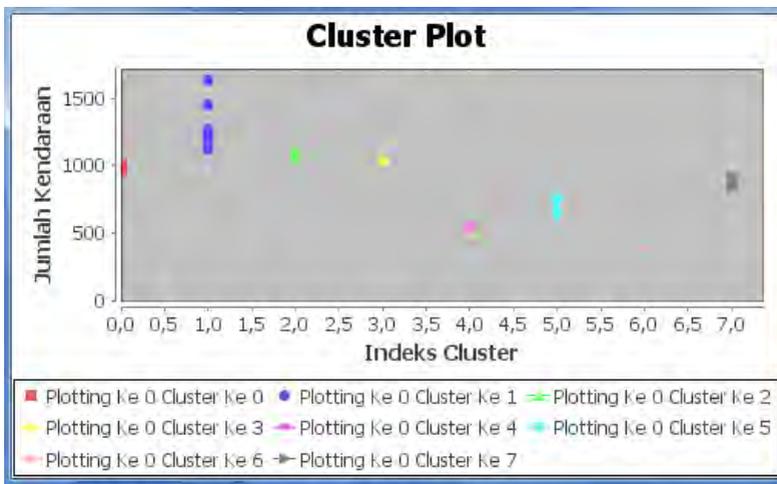


Gambar GL.7. *Plotting Hasil Cluster Jalan Raya pada Sore Hari Efektif untuk Jenis Kendaraan Motor*

Tabel TL.7. *Tabel Hasil Cluster Jalan Raya pada Malam Hari Efektif untuk Jenis Kendaraan Motor*

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.1027	0	Ambengan, Bulak Banteng, Jagir Wonokromo, Ketintang, Raya Menur
	1	Ahmadyani, Arief Rahman Hakim, Basuki Rahmat, Kenjeran Mayjen Sungkono, Menganti Lidah Kulon, Menganti, Raya Dupak, Raya ITS, Rungkut Industri, Tambaksari
	2	Darmo, Rajawali
	3	Dharmahusada, Ir.Soekarno, Prof.Dr.Mestopo
	4	Arjuno, Kalimas Barat
	5	Deles, Indrapura, Jemursari, Kalianyar,

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
		Kapasari, Mastrip, Mercon Jimerto, Mulyorejo, Ngagel, Randu
	6	Kembang Jepun
	7	Kedungcowek, Kertajaya, Kusuma Bangsa, Manyar Kertoarjo, Mulyosari, Nginden



Gambar GL.8. *Plotting Hasil Cluster Jalan Raya pada Malam Hari Efektif untuk Jenis Kendaraan Motor*

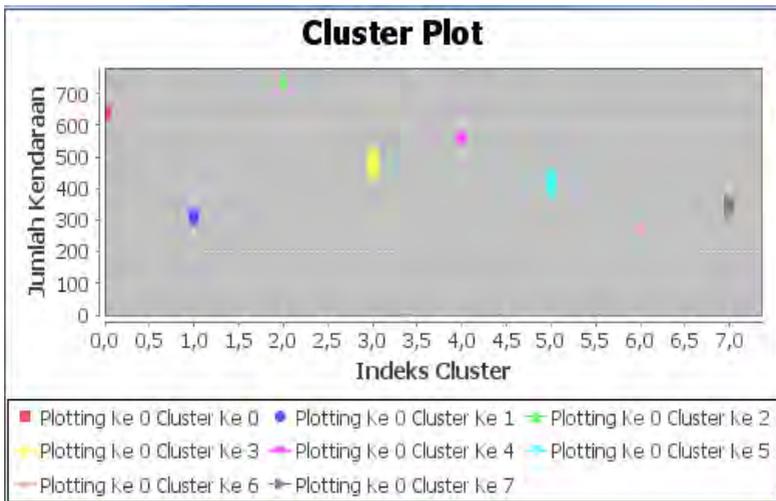
Selanjutnya uji coba parameter di dalam skenario kedua di mana dalam skenario ini dipilih hari efektif kerja dengan rentang waktu selama sepuluh menit. Subskenario yang ada di dalam skenario dua ini sama dengan yang ada di dalam skenario yang pertama yakni terbagi atas dua macam jenis kendaraan, yakni motor dan mobil. Adapun hasil uji coba skenario ini ditunjukkan

oleh Tabel TL.9, Tabel TL.10, Tabel TL.11, Tabel TL.12, Tabel TL.13, Tabel TL.14, Tabel TL.15 dan Tabel TL.16 serta Gambar GL.9, Gambar GL.10, Gambar GL.11, Gambar GL.12, Gambar GL.13, Gambar GL.14, Gambar GL.15 dan Gambar GL.16. Untuk pembagiannya, Tabel TL.9, Tabel TL.10, Tabel TL.11 dan Tabel TL.12 dan Gambar GL.9, Gambar GL.10, Gambar GL.11, dan Gambar GL.12 menunjukkan Tabel dengan subskenario kendaraan mobil, sementara Tabel TL.13, Tabel TL.14, Tabel TL.15 dan Tabel TL.16, serta Gambar GL.13, Gambar GL.14, Gambar GL.15 dan Gambar GL.16 untuk subskenario kendaraan motor.

Tabel TL.8. Tabel Hasil *Cluster* Jalan Raya pada Pagi Hari Libur untuk Jenis Kendaraan Mobil

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0631	0	Ahmadyani, Basuki Rahmat
	1	Bulak Banteng, Dharmahusada, Menganti Lidah Kulon
	2	Darmo
	3	Arjuno, Deles, Jagir Wonokromo, Jemursari, Kedungcowek, Kertajaya, Kusuma Bangsa, Mayjen Sungkono, Menganti, Prof.Dr.Mestopo, Rajawali
	4	Indrapura, Nginden
	5	Ir.Soekarno, Kalimas Barat, Kembang Jepun, Ketintang, Manyar Kertoarjo, Mastrip, Mercon Jimerto, Mulyosari, Mulyorejo, Ngagel, Raya Menur, Rungkut Industri
	6	Ambengan, Kalianyar, Randu, Tambaksari

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
	7	Arief Rahman Hakim, Kapasari Kenjeran, Raya Dupak, Raya ITS



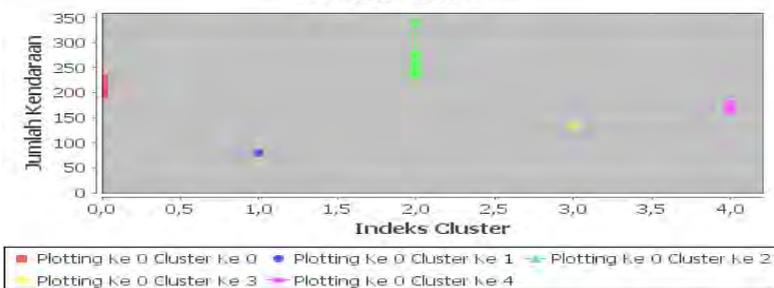
Gambar GL.9. *Plotting* Hasil *Cluster* Jalan Raya pada Pagi Hari Libur untuk Jenis Kendaraan Mobil

Tabel TL.9. Tabel Hasil *Cluster* Jalan Raya pada Siang Hari Libur untuk Jenis Kendaraan Mobil

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0733	0	Arief Rahman Hakim, Deles, Ir.Soekarno, Kalimas Barat, Ketintang, Mastrip, Menganti Lidah, Kulon, Mercon Jimerto, Mulyorejo, Rajawali, Raya Dupak, Rungkut,

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
		Industri, Tambaksari
	1	Arjuno
	2	Ahmadyani, Ambengan, Basuki Rahmat, Darmo, Jagir Wonokromo, Jemursari, Kalianyar, Kenjeran, Kertajaya, Kusuma Bangsa, Manyar Kertoarjo, Mayjen Sungkono, Prof.Dr.Mestopo, Raya Menur
	3	Bulak Banteng
	4	Dharmahusada, Indrapura, Kapasari, Kedungcowek, Kembang Jepun, Menganti, Mulyosari, Ngagel, Nginden, Randu, Raya ITS

**Cluster Plot**

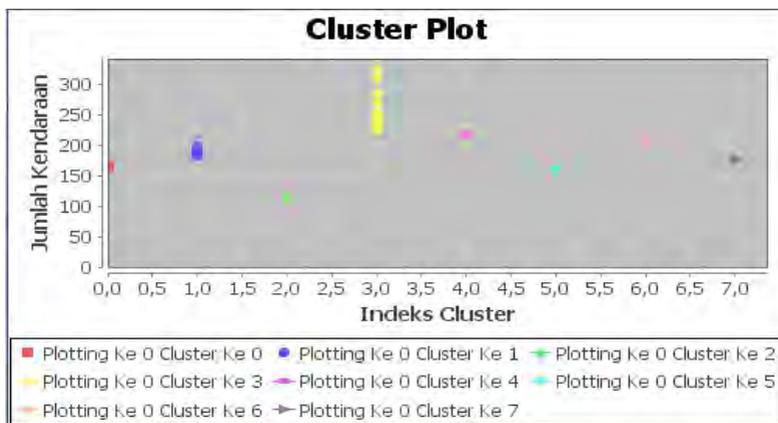


Gambar GL.10. *Plotting* Hasil *Cluster* Jalan Raya pada Siang Hari Libur untuk Jenis Kendaraan Mobil

Tabel TL.10. Tabel Hasil *Cluster* Jalan Raya pada Sore Hari Libur untuk Jenis Kendaraan Mobil

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0864	0	Ambengan, Kembang Jepun, Nginden
	1	Arief Rahman Hakiem, Indrapura

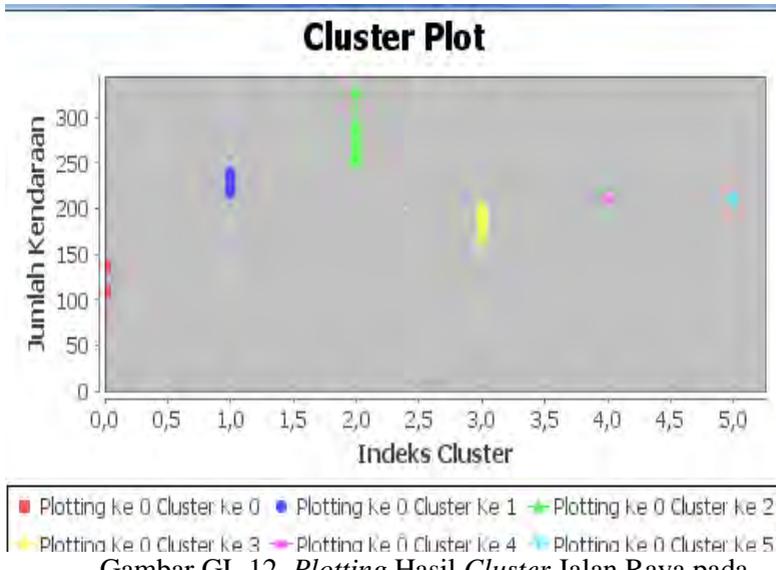
Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
		Ir.Soekarno, Kapasari, Mercon Jimerto, Raya Dupak
	2	Arjuno
	3	Ahmadyani, Basuki Rahmat, Darmo, Jagir Wonokromo, Kalianyar, Kenjeran, Kertajaya, Kusuma Bangsa, Mastrip, Menganti Lidah Kulon, Prof.Dr.Mestopo, Rajawali, Rungkut Industri, Tambaksari
	4	Deles, Ketintang, Manyar Kertoarjo
	5	Bulak Banteng, Dharmahasada, Raya ITS
	6	Jemursari, Kalimasbarat, Mayjen Sungkono, Mulyorejo, Randu, Raya Menur
	7	Kedungcowek, Menganti, Mulyosari, Ngagel



Gambar GL.11. *Plotting Hasil Cluster Jalan Raya pada Sore Hari Libur untuk Jenis Kendaraan Mobil*

Tabel TL.11. Tabel Hasil *Cluster* Jalan Raya pada Malam Hari Libur untuk Jenis Kendaraan Mobil

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0841	0	Arief Rahman Hakim, Raya ITS
	1	Deles, Kapasari, Kedungcowek, Kusuma Bangsa, Randu
	2	Ahmadyani, Basuki Rahmat, Darmo, Indrapura, Ir.Soekarno, Kalianyar, Kenjeran, Ketintang, Mastrip, Mayjen Sungkono, Mercon Jimerto, Mulyosari, Ngagel, Prof.Dr.Mestopo Rajawali
	3	Ambengan, Arjuno, Bulak Banteng, Dharmahusada, Jagir Wonokromo, Kalimas Barat, Kembang Jepun, Menganti Lidah Kulon, Menganti, Mulyorejo, Rungkut Industri
	4	Kertajaya, Nginden
	5	Jemursari, Manyar Kertoarjo, Raya Dupak, Raya Menur, Tambaksari

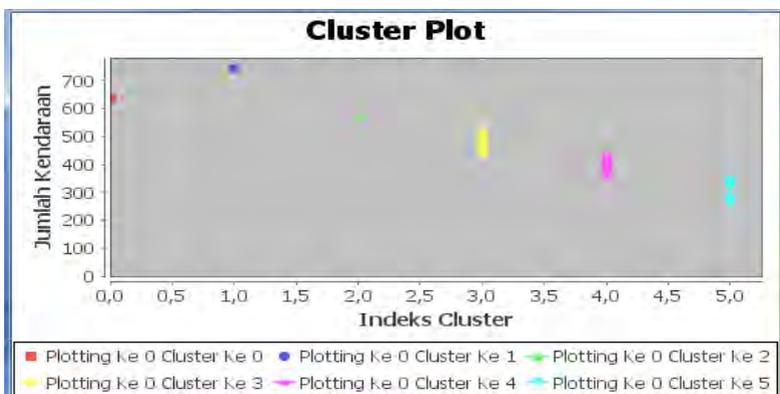


**Gambar GL.12. Plotting Hasil Cluster Jalan Raya pada Malam Hari Libur untuk Jenis Kendaraan Mobil**

**Tabel TL.12. Tabel Hasil Cluster Jalan Raya pada Pagi Hari Libur untuk Jenis Kendaraan Motor**

Nilai <i>DB Index Cluster</i> Terbaik	Indeks Cluster	Nama Jalan Raya
	0	Basuki Rahmat
	1	Darmo
	2	Indrapura
	3	Arjuno, Deles, Jagir Wonokromo, Jemursari, Kedungcowek, Kertajaya, Kusuma Bangsa, Manyar Kertoarjo, Mastrip,

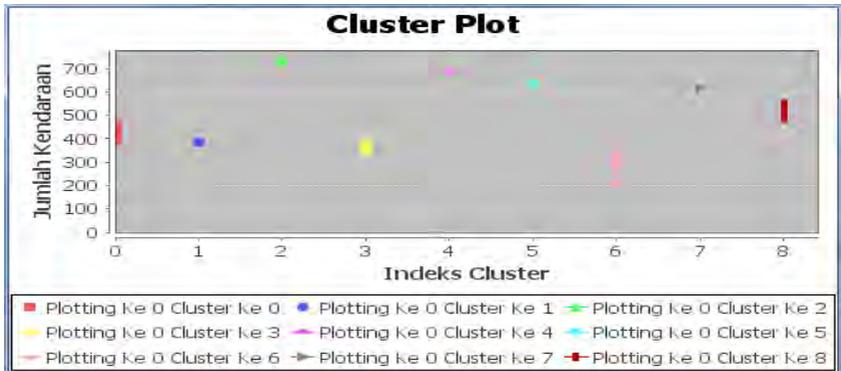
Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0444		Menganti, Nginden, Rajawali
	4	Ahmadyani, Arief Rahman Hakim, Ir.Soekarno, Kalimas Barat, Kembang Jepun, Kenjeran, Ketintang, Mayjen Sungkono, Menganti Lidah Kulon, Mercon Jimerto, Mulyosari, Mulyorejo, Ngagel, Raya Menur, Rungkut Industri
	5	Ambengan, Bulak Banteng, Dharmahusada, Kalianyar, Kapasari, Prof.Dr.Mestopo, Randu Raya Dupak, Raya ITS, Tambaksari



Gambar GL.13. *Plotting Hasil Cluster Jalan Raya pada Pagi Hari Libur untuk Jenis Kendaraan Motor*

Tabel TL.13. Tabel Hasil *Cluster* Jalan Raya pada Siang Hari  
Libur untuk Jenis Kendaraan Motor

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0545	0	Ambengan, Deles, Kedungcowek, Kusuma Bangsa, Mastrip, Mercon Jimerto, Mulyosari, Raya Dupak
	1	Arjuno, Ketintang
	2	Ahmadyani, Basuki Rahmat
	3	Bulak Banteng, Kalianyar, Kenjeran, Ngagel, Prof.Dr.Mestopo, Rajawali
	4	Darmo
	5	Indrapura
	6	Jagir Wonokromo, Kalimas Barat, Menganti Lidah Kulon, Menganti, Mulyorejo, Nginden, Raya ITS, Raya Menur, Rungkut Industri, Tambaksari
	7	Jemursari
	8	Arief Rahman Hakim, Dharmahusada, Ir.Soekarno, Kapasari, Kembang Jepun, Kertajaya, Manyar Kertoarjo, Mayjen Sungkono

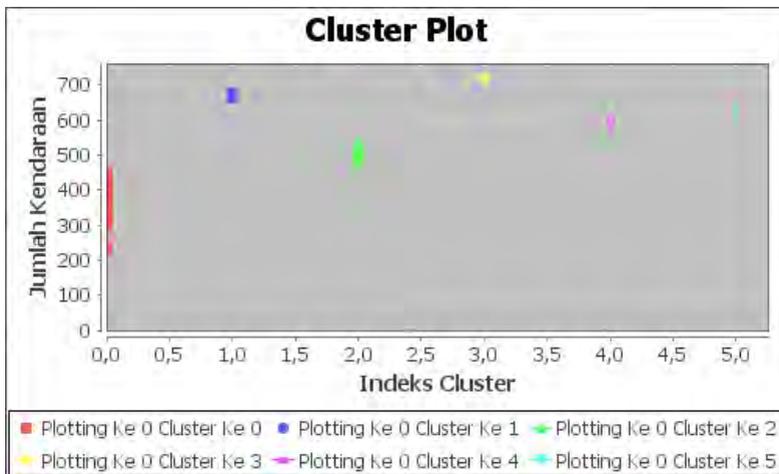


Gambar GL.14. *Plotting Hasil Cluster Jalan Raya pada Siang Hari Libur untuk Jenis Kendaraan Motor*

Tabel TL.14. *Tabel Hasil Cluster Jalan Raya pada Sore Hari Libur untuk Jenis Kendaraan Motor*

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.0858	0	Ambengan, Arjuno, Bulak Banteng, Deles, Jagir Wonokromo, Kalianyar, Kalimas Barat, Kenjeran Ketintang, Kusuma Bangsa, Mastrip, Menganti Lidah Kulon, Menganti, Mercon Jimerto, Mulyosari, Mulyorejo, Ngagel, Nginden, Rajawali, Randu, Raya Dupak, Raya ITS, Raya Menur, Rungkut Industri, Tambaksari
	1	Basuki Rahmat, Darmo
	2	Arief Rahman Hakim, Dharmahasada, Kapasari,

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
		Kedungcowek, Kembang Jepun, Kertajaya Manyar Kertoarjo, Prof.Dr.Mestopo
	3	Ahmadyani, Indrapura
	4	Ir.Soekarno, Mayjen Sungkono
	5	Jemursari

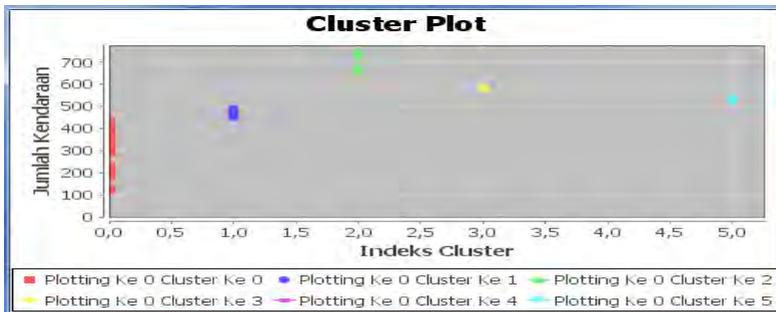


Gambar GL.15. *Plotting* Hasil *Cluster* Jalan Raya pada Sore Hari Libur untuk Jenis Kendaraan Motor

Tabel 6.2.15. Tabel Hasil *Cluster* Jalan Raya pada Malam Hari Libur untuk Jenis Kendaraan Motor

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
	0	Ambengan, Arjuno, Bulak Banteng,

Nilai <i>DB Index Cluster</i> Terbaik	Indeks <i>Cluster</i>	Nama Jalan Raya
0.1078		Kalianyar, Kalimas, Barat, Kapasari, Kedungcowek, Kembang Jepun, Kenjeran, Kertajaya, Menganti Lidah Kulon, Menganti, Mercon Jimerto, Mulyosari, Mulyorejo, Ngagel, Nginden, Rajawali, Randu, Raya ITS, Raya Menur, Rungkut Industri, Tambaksari
	1	Arief Rahman Hakim Dharmahasada Indrapura Ketintang Manyar Kertoarjo Prof.Dr.Mestopo
	2	Ahmadyani Darmo Mayjen Sungkono
	3	Deles Ir.Soekarno Jagir Wonokromo Mastrip
	4	Basuki Rahmat
	5	Jemursari Kusuma Bangsa



Gambar GL.16. *Plotting Hasil Cluster Jalan Raya pada Malam Hari Libur untuk Jenis Kendaraan Motor*

Data hasil pengamatan ini dibagi atas dua jenis pencatatan yang berbeda. Pada Tabel 7.17 diberikan tabel yang mencatat jarak kedatangan antar kendaraan setiap detik dalam satu menit. Pengamatan ini dilakukan pada tanggal 4 Juni 2014 dengan pengamatan diantara jam 10.00 hingga jam 11.00. Sementara Pada Tabel 7.18 jumlah kedatangan kendaraan setiap menitnya dalam 20 menit. Pengamatan pada tabel 7.18 dilakukan pada tanggal 5 Juni 2014 dengan rentang waktu pengamatan antara pukul 08.00 hingga 09.00.

Tabel 6.2.16. Contoh Tabel Kedatangan Kendaraan Di Dalam Periode Detik Untuk Jenis Kendaraan Mobil pada Siang hari

Nomer	Detik Kedatangan
1	1
2	5
3	7
4	8
5	9
6	13
7	15
8	16

Nomer	Detik Kedatangan
9	28
10	29
11	34
12	40
13	43
14	50
15	52
16	55
17	56
18	57
19	59
20	60

Tabel 6.2.17. Contoh Jumlah Kendaraan yang Melintas Setiap Menit di Jalan Raya ITS untuk Jenis Kendaraan Motor pada Siang hari

Menit Ke-	Jumlah Kedatangan Kendaraan
1	17
2	13
3	8
4	13
5	12
6	18
7	9
8	11
9	17
10	13
11	14
12	9

13	12
14	13
15	18
16	13
17	12
18	21
19	10
20	10

## BIODATA PENULIS



**Tsabbit Aqdami Mukhtar**, dilahirkan di kabupaten Pamekasan, Jawa Timur pada tanggal 25 Juli 1991. Penulis adalah anak pertama dari dua bersaudara. Penulis telah menempuh pendidikan formal yaitu di TK Islam Al-Munawwarah (1995-1998), SDI Al-Munawwarah (1998-2004), SMPN 2 Pamekasan (2004-2007) dan SMAN 1 Pamekasan (2007-2010). Setelah lulus dari SMAN 1 Pamekasan pada tahun 2010, Penulis mengikuti Tes

Penerimaan Unggul Mandiri ITS dan diterima di Jurusan Teknik Informatika ITS pada tahun 2010 dan terdaftar dengan NRP 5110100227. Selama menjadi mahasiswa, aktif sebagai anggota HMTC angkatan C1A. Penulis pernah menjadi staff departemen dalam negeri HMTC 2011-2012 dan Staff kementerian dalam negeri BEM FTIf dan menjadi Ketua HMTC periode 2012-2013. Penulis pernah menjadi asisten dosen Mata Kuliah Basis Data (2012-2013) PIKTI ITS. Di Jurusan Teknik Informatika ITS ini, Penulis mengambil Bidang Minat Komputasi Cerdas dan Visualisasi (KCV). Penulis dapat dihubungi melalui alamat *e-mail* di [tsabbitaqdami@gmail.com](mailto:tsabbitaqdami@gmail.com).