

Analisa Clustering Tingkat Rawan Kecelakaan Dengan Fuzzy C-Means

¹Hamzah, ²Arie Vatesia

¹Fakultas Hukum, Universitas Bengkulu

²Informatika, Fakultas Teknik, Universitas Bengkulu

¹hhatrik@unib.ac.id; ²arie.vatesia@unib.ac.id

Article Info

Article history:

Received, 14/12/2022

Revised, 20/01/2022

Accepted, 28/01/2022

Kata Kunci:

Logika Fuzzy
Kecelakaan
Ruas Jalan
Fuzzy C-Means
Bengkulu

Keywords:

Fuzzy Logic
Accidents
Roads
Fuzzy C-Means
Bengkulu

ABSTRAK

Kecelakaan lalu lintas adalah suatu peristiwa di jalan yang tidak diduga dan tidak disengaja melibatkan kendaraan dengan atau tanpa pengguna jalan lain yang mengakibatkan korban manusia dan kerugian harta benda, baik dari kecelakaan yang tergolong ringan sampai dengan kecelakaan yang tergolong berat. Saat ini kecelakaan telah menjadi penyebab kematian cukup banyak di Indonesia khususnya di daerah Kota Bengkulu. Maka dari itu tujuan penelitian ini adalah membangun sistem clustering tingkat rawan kecelakaan menggunakan algoritma Fuzzy C-Means clustering di Kota Bengkulu berbasis web dengan framework CodeIgneter. Clustering tingkat rawan kecelakaan berdasarkan ruas jalan di Kota Bengkulu menjadi 3 (tiga) kelompok yaitu: 1) aman, 2) cukup rawan, dan 3) rawan. Dari 85 data indikator dapat clustering menggunakan Algoritma Fuzzy C-Means clustering berdasarkan 7 kriteria yaitu: jumlah kasus kecelakaan, jumlah meninggal dunia, jumlah luka berat, jumlah luka ringan, jumlah kecelakaan berat, jumlah kecelakaan sedang dan jumlah kecelakaan ringan. Dan hasil perhitungan fungsi obyektif sampai pada kondisi berhenti yang tercapai pada iterasi ke-70 adalah sebagai berikut: tingkat rawan kecelakaan ditentukan dengan 3 (tiga) cluster yaitu (i) ruas jalan yang tergolong cluster rawan terdapat 11 ruas jalan, (ii) ruas jalan yang tergolong cluster cukup rawan terdapat 33 ruas jalan, (iii) ruas jalan yang tergolong cluster aman terdapat 42 ruas jalan berdasarkan 7 kriteria dengan perhitungan algoritma Fuzzy C-Means clustering.

ABSTRACT

Traffic accidents are an unexpected and accidental road event involving vehicles with or without other road users resulting in human casualties and property losses, both from accidents that are relatively mild to heavy accidents. Currently accidents have become a cause of death quite a lot in Indonesia, especially in Bengkulu. Therefore, the purpose of this research is to build a clustering system of accident-prone levels using the Fuzzy algorithm C-Means clustering in Bengkulu City based web with CodeIgneter framework. Clustering The accident-prone levels based on the road section in the city of Bengkulu into 3 (three) groups that are: 1) safe, 2) moderately vulnerable, and 3) prone. From 85 data indicators can clustering using the Fuzzy algorithm C-Means clustering based on 7 criteria: The number of cases of accidents, the number of dies, the number of wounds, the number of minor injuries, the number of heavy accidents, the number of moderate accidents and the number of minor accidents. And the result of the calculation of objective function until the stop condition reached in the iteration 70 is as follows: Accident prone levels are determined by 3 (three) clusters that are (i) road segments that belong to the vulnerable cluster there are 11 roads, (ii) road segments that belong to the cluster is prone to 33 road segments, (iii) road segments that belong to a secure cluster there are 42 segments based on 7 criteria with a calculation of Fuzzy algorithm C-Means clustering.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-nc-nd/4.0/) license.



Penulis Korespondensi:

Hamzah,
Program Studi Informatika,
Universitas Komputer Indonesia,
Email: hhatrik@unib.ac.id

1. PENDAHULUAN

Angka kecelakaan semakin meningkat setiap tahunnya [1], [2], baik dari kecelakaan yang tergolong ringan sampai dengan kecelakaan yang tergolong berat. Saat ini kecelakaan telah menjadi penyebab kematian cukup banyak di Indonesia khususnya di daerah Kota Bengkulu . Faktor utama yang menjadi penyebab semakin tingginya jumlah kecelakaan lalu lintas yaitu pertumbuhan kepemilikan kendaraan bermotor terutama jenis sepeda motor. Sepeda motor merupakan kendaraan yang paling banyak terlibat kecelakaan. Di Kota Bengkulu [3], [4] terhimpun data kejadian kecelakaan lalu lintas sepanjang tahun 2012 berjumlah 220 kasus, terdiri dari korban meninggal 30 jiwa; korban mengalami luka berat 240 jiwa; korban mengalami luka ringan 207 jiwa; serta jumlah kerugian material yaitu Rp 453.400.000,-. Kejadian kecelakaan lalu lintas di Kota Bengkulu yang begitu banyak hingga bisa melebihi angka 200 per tahun ini terjadi disebabkan faktor manusia, faktor kendaraan, faktor jalan dan faktor lingkungan. Hal ini sepatutnya menjadi tanggung jawab pihak berwajib khususnya Unit Laka Lantas Polres Bengkulu dan Dinas Perhubungan untuk penanganan kecelakaan di wilayah Kota Bengkulu. Namun dengan sistem pengolahan data yang masih bersifat statis, data kecelakaan banyak yang hilang dan terpisah. Petugas pun menjadi kewalahan dalam menangani daerah rawan kecelakaan dikarenakan data yang tidak lengkap dan informasi yang tidak aktual [5].

Kecelakaan lalu lintas banyak terjadi pada kendaraan, baik kendaraan roda dua, kendaraan roda empat maupun kendaraan jenis truk. Disamping faktor manusianya yang tidak menaati peraturan lalu lintas juga dipengaruhi oleh faktor geometrik jalan, kondisi kepadatan lalu lintas serta kurangnya rambu peringatan daerah rawan kecelakaan. Penelitian analisis daerah titik rawan kecelakaan diharapkan para petugas dapat mengetahui clustering tingkat rawan kecelakaan berdasarkan ruas jalan di Kota Bengkulu dalam kategori : Rawan, Cukup Rawan dan Aman. Salah satu dari algoritma clustering data adalah Fuzzy C-Means (FCM)[6]–[8]. Fuzzy C-Means (FCM) adalah salah satu teknik clustering data yang mana keberadaan tiap-tiap titik data dalam suatu cluster ditentukan oleh derajat keanggotaan. Teknik ini pertama kali diperkenalkan oleh Jim Bezdek pada tahun 1981. Konsep dasar FCM, selanjutnya adalah menentukan pusat cluster, yang akan menandai lokasi rata-rata untuk tiap-tiap cluster. Pada kondisi awal, pusat cluster ini masih belum akurat. Tiap-tiap titik data memiliki derajat keanggotaan untuk tiap-tiap cluster. Cara memperbaiki pusat cluster dan derajat keanggotaan tiap-tiap titik data secara berulang, maka akan dapat dilihat bahwa pusat cluster akan bergerak menuju lokasi yang tepat. Output dari FCM bukan merupakan fuzzy inference system, namun merupakan deretan pusat cluster dan beberapa derajat keanggotaan untuk tiap-tiap titik data. Informasi ini dapat digunakan untuk membangun suatu fuzzy inference system [6]. Dari hasil penelitian yang ada menggunakan metode Fuzzy C-Means untuk menentukan klasifikasi usaha kecil dan menengah (UKM) sektor industry guna mendapatkan clustering atau pengelompokan yang akurat[9]. Algoritma Fuzzy C-Means disini dihitung berdasarkan beberapa kriteria sebagai penunjang untuk mengklasifikasikan tingkat rawan kecelakaan diantaranya jumlah kejadian kecelakaan, jumlah meninggal dunia, jumlah luka berat, jumlah luka ringan, jumlah kecelakaan berat, jumlah kecelakaan sedang dan jumlah kecelakaan ringan. Jadi dari hasil clustering dengan menggunakan Fuzzy C-Means tersebut adalah pemetaan tingkat rawan kecelakaan berdasarkan ruas jalan di Kota Bengkulu.

2. METODE PENELITIAN

Pada penelitian ini akan dibangun sebuah sistem informasi *clustering* tingkat rawan kecelakaan dengan menggunakan metode *Fuzzy C-Means*[8], [10]. *Algoritma Fuzzy C-Means (FCM)* adalah sebagai berikut:

1. Input data yang akan di *cluster* X, berupa matriks berukuran $n \times m$ (n = jumlah sampel data, m = atribut setiap data). X_{ij} = data sampel ke- i ($i=1,2,\dots,n$), atribut ke- j ($j=1,2,\dots,m$).
2. Langkah selanjutnya ialah menentukan beberapa input yang dibutuhkan dalam perhitungan *fuzzy c-means*, yaitu:
 - a) Jumlah *cluster* (c) ialah banyaknya *cluster* yang akan dibentuk sesuai dengan kebutuhan.
 - b) Pangkat (w) ialah nilai eksponen.
 - c) Maksimum iterasi (MaxIter) merupakan batas pengulangan atau *looping*. *Looping* akan berhenti jika nilai maksimal iterasi sudah tercapai.
 - d) Error terkecil (ξ) berupa batasan nilai yang membuat perulangan akan berakhir setelah didapatkan nilai error yang diharapkan.
 - e) Fungsi objektif awal ($P_0 = 0$) ialah suatu fungsi yang akan dioptimumkan (maksimum atau minimum), nilai 0 berarti untuk mendapatkan nilai minimum.
 - f) Iterasi awal ($t = 1$), iterasi adalah adalah sifat tertentu dari algoritma atau program komputer di mana suatu urutan atau lebih dari langkah algoritmik dilakukan secara berulang. Iterasi awal ialah pada perulangan keberapakan program akan dimulai.
3. Membangkitkan bilangan random u_{ik} , $i=1,2,\dots,n$; $k=1,2,\dots,c$; sebagai elemen-elemen matrik partisi awal U. Hitung jumlah setiap kolom:

$$Q_i = \sum_{k=1}^c \mu_{ik} \tag{1}$$

Q_i adalah jumlah setiap kolom dari nilai random sebuah matriks, jumlah Q tergantung dari berapa jumlah kriteria penilaian.

4. Hitung pusat *cluster* ke- k : V_{kj} , dengan $k=1,2,\dots,c$; dan $j=1,2,\dots,m$

$$V_{kj} = \frac{\sum_{i=1}^n (\mu_{ik})^w \cdot x_{ij}}{\sum_{i=1}^n (\mu_{ik})^w} \tag{2}$$

Keterangan :

V_{kj}	=	Pusat Kluster data ke- i dan kriteria ke- j
x_{ij}	=	Data ke- i dan kriteria ke- j
μ_{ik}	=	Bilangan <i>random</i> data ke- i dan kriteria ke- j
W	=	Nilai pangkat

V_{kj} adalah titik pusat tiap *cluster*, jumlah V_{kj} tergantung dari berapa *cluster* yang akan dibentuk dan n ialah jumlah proposal.

5. Menghitung fungsi objektif pada iterasi ke- t , P_t

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \left(\left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right] (\mu_{ik})^w \right) \tag{3}$$

Keterangan :

P_t	=	Fungsi objektif pada iterasi ke- t
X_{ij}	=	Data ke- i dan kriteria ke- j
V_{kjj}	=	Pusat Kluster kluster ke- k dan kriteria ke- j
μ_{ik}	=	Bilangan <i>random</i> data ke- i dan kluster ke- k
W	=	Nilai pangkat

t merupakan *iterasi* yang dihitung, jika *iterasi* dimulai dari 1 maka pada awal perhitungan nilai t ialah 1. *Iterasi* akan berulang sesuai dengan ketentuan *iterasi* yang sedang berjalan. Hitung perubahan matrik partisi.

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[\sum_{j=1}^m (X_{ij} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \tag{4}$$

Keterangan :

μ_{ik}	=	Elemen matriks baru data ke- i dan kluster ke- k
X_{ij}	=	Data ke- i dan kriteria ke- j
V_{kj}	=	Pusat Kluster ke- k dan kriteria ke- j
W	=	Nilai pangkat

Iterasi akan tetap berulang jika nilai atau kondisi-kondisi tertentu belum tercapai, adapun kondisi tersebut ialah jika: ($|P_t - P_{t-1}| < \xi$) atau ($t > \text{MaxIter}$) maka berhenti yang mana P_t ialah pusat *cluster iterasi* ke t kurang dari nilai error yang diharapkan atau jika t (jumlah iterasi) sudah lebih besar daripada iterasi maksimum. Namun jika iterasi akan diulang lagi dengan $t + 1$ akan mengulang proses yang ke-4 atau menghitung pusat *cluster* lagi [7], [9].

Metode pengembangan untuk sistem Implementasi *Fuzzy C-Means* untuk *clustering* tingkat rawan kecelakaan ini adalah metode air terjun (*waterfall*) [11], [12]. Model air terjun menyediakan pendekatan alur hidup perangkat lunak secara sekuensial atau terurut dimulai dari analisis, desain, pengkodean, pengujian, dan tahap pendukung (*support*). Adapun tahapan tahapan dalam metode *waterfall* adalah :

1. Analisis Kebutuhan Perangkat Lunak

- a) Mengumpulkan data-data yang mendukung, yaitu data dari indikator-indikator yang digunakan untuk *clustering* tingkat rawan kecelakaan.
- b) Menganalisis perangkat lunak yang akan digunakan untuk pembuatan sistem *clustering* tingkat rawan kecelakaan.
- c) Membuat alur untuk pembuatan sistem *clustering* tingkat rawan kecelakaan.

2. Desain

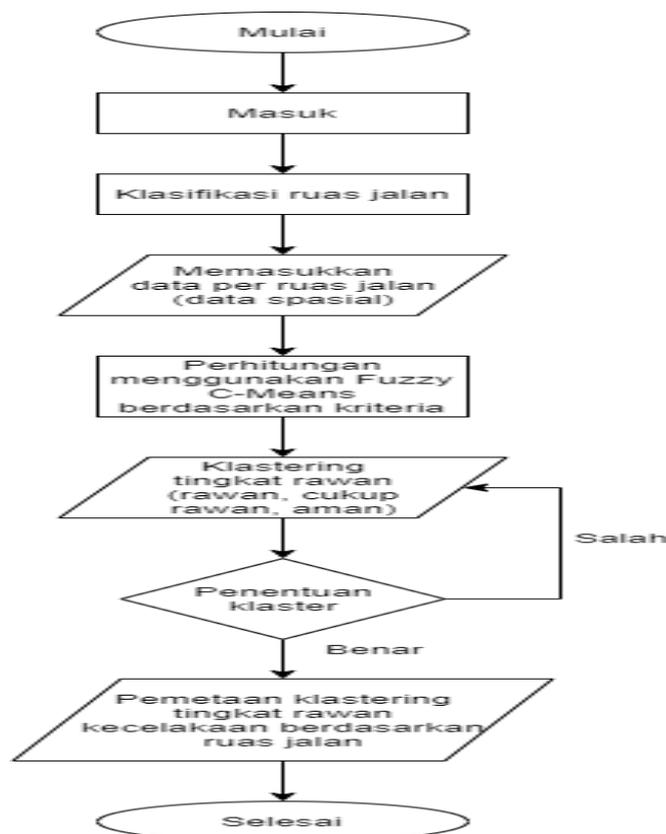
Desain perangkat lunak adalah proses multilangkah yang fokus pada desain pembuatan program perangkat lunak. Pada tahap ini, dilakukan perancangan sistem menggunakan 4 diagram *Unified Modeling Language*. Selain itu, juga dilakukan perancangan *database*, dan perancangan *interface* sebagai antarmuka yang akan menghubungkan pengguna dengan sistem.

3. Implementasi
Tahap ini adalah mentranslasikan desain yang telah dibuat kedalam kode-kode program. Dalam hal ini, bahasa pemrograman yang digunakan adalah bahasa pemrograman PHP dengan bantuan *framework codeigniter*.
4. Pengujian
Pengujian yang dilakukan pada penelitian ini dilakukan dengan melakukan pengujian fungsional dan teknis pada aplikasi yang dibangun, apakah sesuai dengan tujuan dari penelitian ini, yaitu aplikasi berjalan dengan baik dan benar sehingga dapat mengklasifikasi tingkat rawan kecelakaan seperti tujuan dari penelitian ini.
5. Penggunaan dan pemeliharaan
Tidak menutup kemungkinan sebuah perangkat lunak mengalami perubahan ketika sudah dikirimkan oleh pihak yang membutuhkan. Tahapan ini mengantisipasi jika ada ketidaksesuaian sistem setelah dilakukan pengujian. Langkah pada tahapan ini mengulangi tahapan-tahapan sebelumnya.

3. HASIL DAN ANALISIS

3.1 Identifikasi Masalah

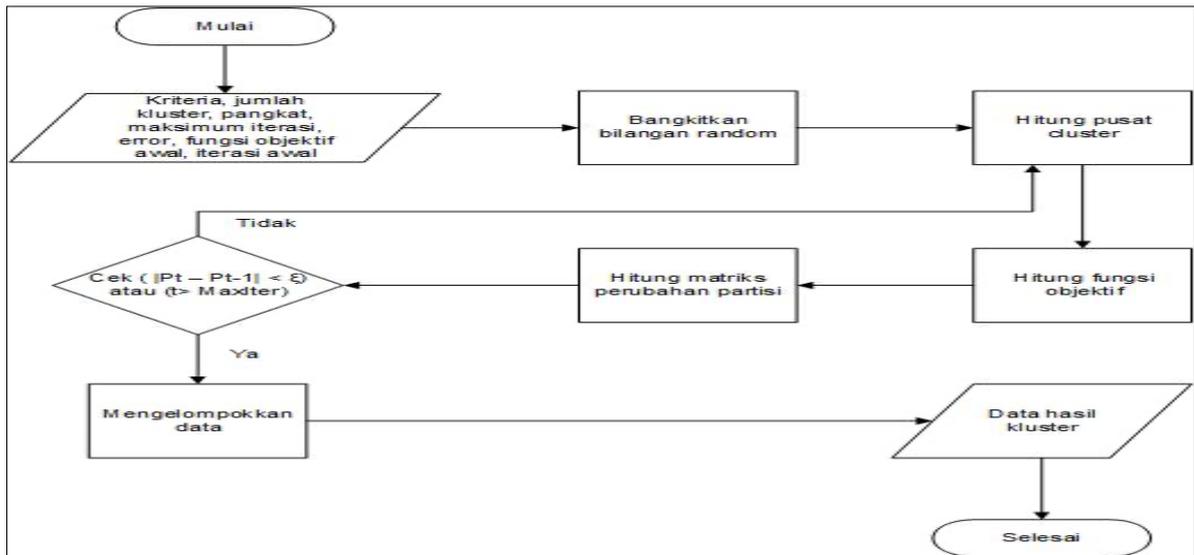
Penelitian analisis daerah titik rawan kecelakaan diharapkan para petugas dapat mengetahui *clustering* tingkat rawan kecelakaan berdasarkan ruas jalan di Kota Bengkulu dalam kategori: Rawan, Cukup Rawan dan Aman.



Gambar 1 Diagram Alur Sistem

Pada Gambar 1 diagram alir, Pada sistem ini dilakukan *clustering* tingkat rawan kecelakaan di Kota Bengkulu dengan metode *Fuzzy C-Means*. Pertama memasukkan data kecelakaan yang terjadi pada ruas jalan yang terdiri 7 kriteria yaitu jumlah kejadian, jumlah meninggal dunia, jumlah luka berat, jumlah luka ringan, jumlah kecelakaan berat, jumlah kecelakaan sedang dan jumlah kecelakaan ringan. Dan setelah data diinputkan selanjutnya dilakukan perhitungan *Fuzzy C-Means* sehingga didapatkan *clustering* tingkat rawan kecelakaan dengan kategori rawan, cukup rawan dan aman dengan penyajian berupa peta.

3.2 Diagram Alir Fuzzy C-Means



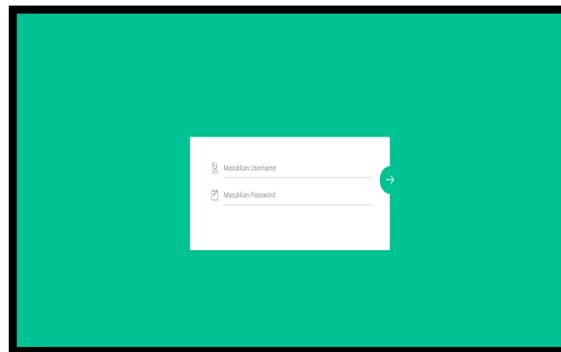
Gambar 2 Diagram Alir Fuzzy

Diagram alir metode *Fuzzy C Means Clustering* untuk *clustering* tingkat rawan kecelakaan menjadi tiga kelompok berdasarkan tujuh kriteria yaitu jumlah kejadian kecelakaan, jumlah meninggal dunia, jumlah luka berat, jumlah luka ringan, jumlah kecelakaan berat, jumlah kecelakaan sedang dan jumlah kecelakaan ringan.

3.3 Implementasi Antar Muka

(1) Halaman Login

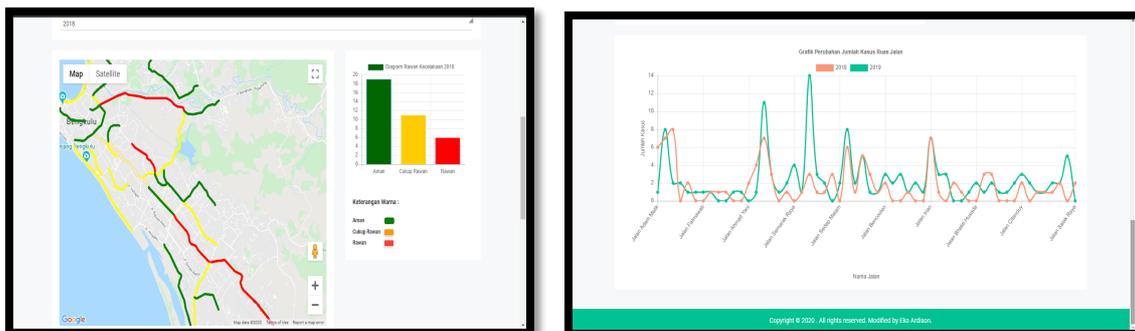
Halaman Login adalah tampilan awal dari sistem ketika pertama kali admin mengakses system clustering tingkat rawan kecelakaan. Pada Gambar 3 menunjukkan rancangan antarmuka halaman utama sistem.



Gambar 3 Halaman Login

(2) Halaman Utama

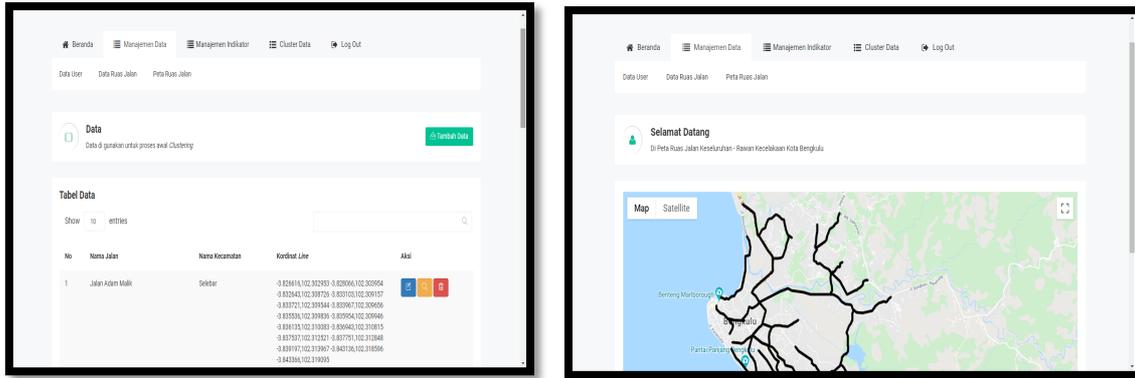
Halaman utama adalah tampilan awal dari sistem ketika pertama kali admin login kedalam web sistem clustering tingkat rawan kecelakaan Gambar 4.



Gambar 4 (a) Halaman Ruas Jalan; (b) Halaman Grafik Jumlah Kecelakaan

(3) Halaman Manajemen Data

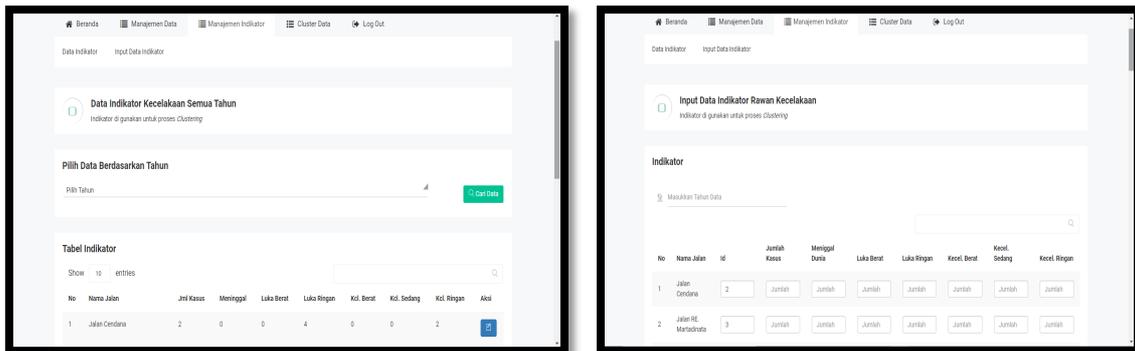
Halaman manajemen data adalah tampilan awal dari sistem ketika pertama kali admin mengakses menu manajemen data kedalam web sistem *clustering* tingkat rawan kecelakaan.



Gambar 5 (a) Submenu Data Ruas Jalan; (b) Submenu Peta Ruas Jalan

(4) Halaman Manajemen Indikator

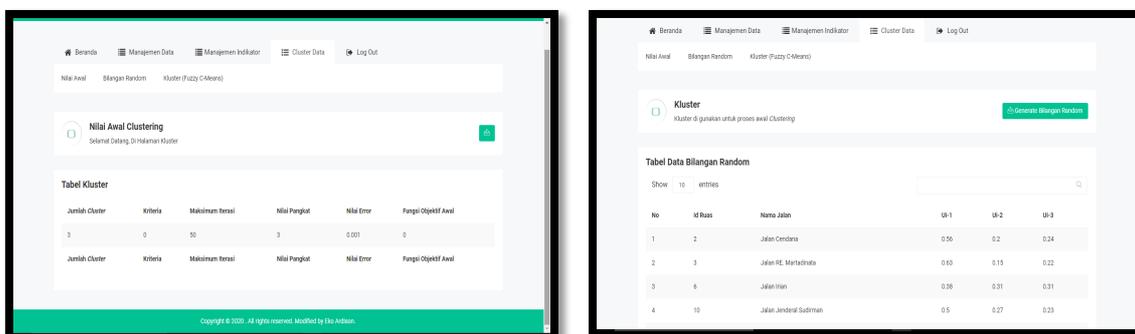
Halaman manajemen indikator adalah tampilan awal dari sistem ketika admin mengklik menu manajemen indikator dari sistem *clustering* tingkat rawan kecelakaan. Pada Gambar 6 dapat terlihat proses dari submenu yang ditampilkan pada manajemen indicator yang dapat berperan dalam pengelolaan data pada sistem.



Gambar 6 (a) Submenu Data Indikator; (b) Submenu Input Data Indikator

(5) Halaman Cluster

Halaman utama adalah tampilan awal dari sistem ketika pertama kali admin mengklik menu *cluster* di web sistem *clustering* tingkat rawan kecelakaan Gambar 7.



Gambar 7(a) Submenu Perhitungan Nilai Awal; (b) Submenu Perhitungan Nilai Random

Pada proses perhitungan algoritma, kriteria sangat mempengaruhi hasil akhir dari perhitungan. Pada Algoritma *Fuzzy C-Means* memiliki tujuh kriteria yaitu jumlah kejadian, jumlah meninggal dunia, jumlah luka berat, jumlah luka ringan, jumlah kecelakaan berat, jumlah kecelakaan sedang, dan jumlah kecelakaan ringan. Dari kriteria tersebut akan dilakukan perhitungan algoritma, Berikut adalah perhitungan yang dilakukan. Masukkan dsata terlebih dahulu seperti pada tabel dibawah ini:

Tabel 1 Data Kecelakaan Tahun 2019

No	Ruas jalan	Jumlah Kejadian	Jumlah Meninggal Dunia	Jumlah Luka Berat	Jumlah Luka Ringan	Jumlah Kec. Berat	Jumlah Kec. sedang	Jumlah Kec. Ringan
1.	Jl. Cendana	2	0	0	4	0	0	2
2.	Jl. RE martadinata	14	5	2	11	5	2	7
3.	Jl. Irian	7	0	2	6	0	2	5
4.	Jl. Jendral sudirman	2	0	0	2	0	0	2
5.	Jl. Sedap malam	2	0	0	2	0	0	2
6.	Jl. Pariwisata	8	1	3	11	1	1	6
7.	Jl. Bumi ayu	3	0	0	8	0	0	3
8.	Jl. Hibrida	1	0	0	2	0	0	1
9.	Jl. Merapi raya	5	1	2	5	1	1	3
10.	Jl. WR supratman	11s	3	3	10	3	3	5

Dalam proses perhitungan ini hanya digunakan 10 data saja dengan alasan efisiensi. Meskipun demikian, 10 data ini tetap diakumulasikan dari semua data yang ada. Semua rumus perhitungan mempertimbangkan keberadaan data-data lainnya, sehingga hasilnya tetap relevan dengan keluaran dari sistem.

1. Menentukan Nilai Awal

Setelah memasukkan data kriteria ke dalam tabel, selanjutnya adalah menentukan ketetapan nilai-nilai awal yang digunakan. Ketetapan nilai pada penelitian ini dengan menetapkan nilai Jumlah cluster (c)= 3, Jumlah kriteria (j) = 7, nilai pangkat (w)= 3, Maksimum iterasi ($MaxIter$)= 70, Error terkecil (ϵ) = 0.001, Fungsi objektif awal (P_0) = 0, Iterasi awal (t awal)=0.

2. Membangkitkan Bilangan Random

Langkah berikutnya adalah membangkitkan bilangan *random*. Bilangan ini digunakan sebagai nilai derajat keanggotaan awal dari setiap data. Dalam sistem ini, bilangan *random* dibangkitkan (*generate*) menggunakan sebuah fungsi khusus. Hal ini dikarenakan bilangan *random* yang dibangkitkan memiliki syarat khusus, yaitu jumlah keseluruhan kluster sama dengan 1. Ada tiga bilangan *random* yang akan dibangkitkan dalam setiap data yang diwakilkan dengan variabel U_{i1} (*cluster* pertama), U_{i2} (*cluster* kedua), dan U_{i3} (*cluster* ketiga). Dengan menggunakan data *sample* dengan persamaan pada Algoritma *Fuzzy C-Means clustering*.

3. Menghitung Pusat Cluster

Setelah mendapatkan bilangan *random*, selanjutnya adalah menghitung pusat *cluster* dari masing-masing kriteria yang ada pada masing-masing *cluster*. Pusat *cluster* dihitung dengan mengalikan perpangkatan bilangan *random* dengan setiap kriteria pada masing-masing *cluster*. Hasil dari setiap kriteria akan dijumlahkan, kemudian dibagi dengan jumlah perpangkatan bilangan *random* untuk masing-masing kluster.

4. Menghitung Matriks Perubahan Partisi

Langkah berikutnya adalah menghitung perubahan matriks partisi (derajat keanggotaan baru). Perhitungan nilai matriks partisi baru dilakukan dengan memangkatkan hasil pangkat 3 nilai selisih kriteria dengan pusat *cluster*. Seperti yang dilakukan ketika menghitung fungsi objektif. Perbedaannya pada perubahan matriks partisi, nilai pangkat tersebut dipangkatkan lagi dengan $-1/(w-1)$ untuk masing-masing *cluster*.

3.4 Hasil Clustering

Tabel 2. Hasil Cluster

NO	Nama Jalan	Ui-1 baru	Ui-2 baru	Ui-3 baru	Cluster
1.	Jalan Cendana	0.0388	0.958	0.0032	2
2.	Jalan RE martadinata	0.0743	0.1152	0.8105	3
3.	Jalan Irian	0.1015	0.2743	0.6242	3
4.	Jalan Jend Sudirman	0.4566	0.5361	0.0074	2
5.	Jalan Sedap Malam	0.4566	0.5361	0.0074	2
6.	Jalan Pariwisata	0.0162	0.0343	0.9495	3
7.	Jalan Bumi Ayu	0.1446	0.6384	0.217	2
8.	Jalan Hibrida	0.81	0.1863	0.0036	1
9.	Jalan Merapi raya	0.1362	0.6867	0.1771	2
10.	Jalan WR supratman	0.0226	0.0405	0.937	3

Keterangan (Iterasi ke-70):

- Merah = Kelompok 3 (*cluster* 3) ruas jalan tergolong rawan yang berjumlah 4 data.

- Kuning = Kelompok 2 (*cluster 2*) ruas jalan yang tergolong cukup rawan berjumlah 5 data
- Hijau = Kelompok 1 (*cluster 1*) ruas jalan tergolong aman yang berjumlah 1 data. (Ui11, Ui22, Ui33) yang mewakili *cluster 1, 2, dan 3* lebih tinggi dari yang lainnya. Contohnya, pada data pertama, Ui-3 baru dengan nilai 0.8105 lebih tinggi dari Ui-1 baru yaitu 0.0743 dan Ui-2 baru yaitu 0.1152 sehingga dapat disimpulkan bahwa data kedua masuk pada *cluster* ketiga.

4. KESIMPULAN

Penelitian ini telah menghasilkan sebuah sistem implementasi *Fuzzy C-Means* untuk *clustering* tingkat rawan kecelakaan di Kota Bengkulu dengan *framework codeigniter*. Sistem telah berhasil menampilkan *clustering* tingkat rawan kecelakaan berdasarkan ruas jalan di Kota Bengkulu dengan algoritma *Fuzzy C-Means clustering* berdasarkan pengujian *blackbox testing*. Dari 85 data indikator tahun 2018 dan 2019 dapat diclusterkan secara optimal pada 3 cluster. Klasterisasi tingkat rawan kecelakaan yang tergolong rawan (*Cluster ke-1*) terdapat 11 ruas jalan. Klasterisasi tingkat rawan kecelakaan yang tergolong cukup rawan (*Cluster ke-2*) terdapat 33 ruas jalan. Klasterisasi tingkat rawan kecelakaan yang tergolong aman (*Cluster ke-3*) 42 ruas jalan. Dari 56 ruas jalan terdapat perbandingan jumlah kejadian kecelakaan diantara tahun 2018 dan 2019 dimana ada ruas jalan yang mengalami peningkatan dan penurunan jumlah kecelakaan serta ada yang stabil.

DAFTAR PUSTAKA

- [1] H. Herawati, "Karakteristik Dan Penyebab Kecelakaan Lalu Lintas Di Indonesia Tahun 2012," *War. Penelit. Perhub.*, vol. 26, no. 3, 2019.
- [2] A. Zauardi and H. Suprayitno, "Analisa Karakteristik Kecelakaan Lalu Lintas di Jalan Ahmad Yani Surabaya melalui Pendekatan Knowledge Discovery in Database," *J. Manajemen Aset Infrastruktur Fasilitas*, vol. 2, no. 1, 2018.
- [3] E. T. Pujiastutie and E. Diana, "Karakteristik kecelakaan dan solusi penanganan untuk mengurangi angka kecelakaan di kota bengkulu," *Res. Rep.*, 2016.
- [4] L. NOVIYANTI, H. SUMARNO, and P. SIANTURI, "ANALISIS FAULT TREE DAN APLIKASINYA PADA MASALAH KECELAKAAN LALU LINTAS DI PROVINSI BENGKULU," *J. Math. Its Appl.*, vol. 13, no. 1, 2014.
- [5] A. D. Saputra, "Studi Tingkat Kecelakaan Lalu Lintas Jalan di Indonesia Berdasarkan Data KNKT (Komite Nasional Keselamatan Transportasi) dari Tahun 2007-2016," *War. Penelit. Perhub.*, vol. 29, no. 2, 2018.
- [6] K. Benmouiza and A. Cheknane, "Clustered ANFIS network using fuzzy c-means, subtractive clustering, and grid partitioning for hourly solar radiation forecasting," *Theor. Appl. Climatol.*, vol. 137, no. 1–2, 2019.
- [7] S. Qaiyum, I. Aziz, J. Jaafar, and A. K. L. Wong, "Ant colony optimization of interval Type-2 Fuzzy C-Means with subtractive clustering and multi-round sampling for large data," *Int. J. Adv. Comput. Sci. Appl.*, vol. 10, no. 1, 2019.
- [8] B. Abdeljabar, L. Rahmani, B. Mebrouk, and B. Kada, "Estimating daily diffuse solar radiation using neuro-fuzzy, fuzzy C-means and subtractive clustering: Case study: Adrar, Algeria," in *PervasiveHealth: Pervasive Computing Technologies for Healthcare*, 2020.
- [9] J. Lu, D. Bai, N. Zhang, T. Yu, and X. Zhang, "Fuzzy case-based reasoning system," *Appl. Sci.*, vol. 6, no. 7, 2016.
- [10] B. N. Haqiqi and R. Kurniawan, "ANALISIS PERBANDINGAN METODE FUZZY C-MEANS DAN SUBTRACTIVE FUZZY C-MEANS," *MEDIA Stat.*, vol. 8, no. 2, 2015.
- [11] S. Sinanović and D. H. Johnson, "Toward a theory of information processing," *Signal Processing*, vol. 87, no. 6, 2007.
- [12] S. Silva, L. Alçada-Almeida, and L. C. Dias, "Development of a web-based multi-criteria spatial decision support system for the assessment of environmental sustainability of dairy farms," *Comput. Electron. Agric.*, vol. 108, pp. 46–57, 2014.