

Perhitungan Jumlah dan Jenis Kendaraan Menggunakan Metode Fuzzy C-means dan Segmentasi Deteksi Tepi Canny

Dimas Wahyu Wibowo, M. Aziz Muslim, M. Sarosa

Abstrak—Pada Penelitian ini memungkinkan kita untuk melakukan perhitungan kendaraan dengan cepat dan mudah dengan menggunakan metode deteksi tepi Canny untuk proses segmentasi dan Fuzzy C-Means untuk menghitung kendaraan. Berdasarkan hasil penelitian dapat disimpulkan bahwa sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-means dan Segmentasi menggunakan Deteksi Tepi Canny untuk penerangan cahaya yang cukup didapatkan persentase keberhasilan 87 persen tetapi untuk pencahayaan yang kurang, belum mendapat hasil yang memuaskan.

Kata kunci penelusuran katalog, Fuzzy C-means, Segmentasi, Deteksi Tepi Canny.

I. PENDAHULUAN

SEKARANG ini jumlah alat transportasi hampir setiap hari bertambah, tetapi tidak diiringi dengan perubahan volume (panjang dan lebar) jalan yang signifikan. Akibatnya dengan kondisi panjang maupun lebar jalan yang tetap dan semakin bertambahnya jumlah kendaraan, maka terjadinya penumpukan jumlah kendaraan pada ruas jalan. Kondisi ini terjadi karena lokasi tujuan yang sama dan jika tujuan berbeda, mereka tidak mengetahui rute alternatif untuk mencapai lokasi yang dituju. Oleh karena itu, seringkali di jalan tertentu timbul kemacetan yang disebabkan penumpukan kendaraan.

Untuk mencegah adanya penumpukan jumlah kendaraan dilokasi tertentu dibutuhkan suatu data statistik tentang pertumbuhan kendaraan di wilayah tersebut tiap tahunnya [1]. Pertumbuhan kendaraan ini dapat dilihat dari seberapa banyak kendaraan yang melewati jalan tertentu. Dengan mengetahui pertumbuhan kendaraan di suatu kota dapat digunakan sebagai tolak ukur perlu tidaknya adanya penambahan fasilitas jalan raya atau jalan alternatif.

Dari pemaparan di atas dapat diketahui salah satu penyebab kemacetan adalah kondisi panjang maupun

lebar jalan yang tetap dan semakin bertambahnya jumlah kendaraan. Hal ini berarti panjang dan lebar kendaraan berpengaruh pada luas dari jalan tersebut sehingga kedua faktor tersebut digunakan sebagai acuan pada penelitian ini.

Berdasarkan dari acuan tersebut dapat digunakan sebagai masukan Fuzzy C-Means untuk meng-Clusterkan jenis-jenis kendaraan berdasarkan pada panjang, lebar dan titik sudut yang terdapat pada kendaraan. Untuk mendapatkan ciri-ciri dari suatu kendaraan proses pertama yang harus dilakukan adalah proses ekstraksi ciri menggunakan deteksi tepi canny.

Deteksi tepi dengan menggunakan metode Canny merupakan deteksi tepi terbaik dikarenakan morfologi garis yang dihasilkan oleh deteksi tepi ini lebih halus dan tidak terputus-putus dibandingkan deteksi tepi sobel dan LOG (Laplasiian Of Gaussian). Karena alasan tersebut, maka sebagian besar proses ekstraksi ciri suatu citra menggunakan Metode Canny. Pada kasus segmentasi citra dan pengenalan obyek, deteksi tepi dengan metode canny terbukti memiliki performa yang lebih baik daripada metode sobel, prewitt, roberts, dan LoG. Algoritma canny *edge* memiliki kelebihan dapat memberikan hasil deteksi tepi yang optimal dan sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat ketebalan tepi sesuai yang diinginkan. [2] [3] [4].

Clustering merupakan proses pengelompokan objek atau data tidak berlabel kedalam suatu kelas atau *Cluster* dengan objek yang memiliki kesamaan. *Clustering* dengan menggunakan metode Fuzzy C-Means dapat memunculkan beberapa *Cluster* data yang dapat dianalisa lebih lanjut persamaan dan perbedaannya. Selain itu Fuzzy *Clustering* dapat digunakan dalam pengenalan karakter angka dan didapatkan kesimpulan bahwa dengan metode Fuzzy dapat mengenali dengan baik untuk data-data yang perbedaannya terlihat dengan jelas. Pada pengenalan karakter angka, tingkat pengenalan yang dicapai adalah 70% [5] [6].

Ada beberapa pendekatan yang berbeda untuk pengidentifikasian suatu kendaraan dengan menggunakan metode Hough Transform (HT) dan *Connected Component Labeling* (CCL). Metode HT dan CCL ini dapat digunakan untuk mendeteksi kendaraan pada jalan raya. Namun metode CCL mendeteksi dua kendaraan yang berdekatan menjadi satu objek sehingga

DimasWahyu Wibowo adalah mahasiswa Program Magister Jurusan Teknik Elektro Universitas Brawijaya, Malang, Indonesia. (Kontak phone 08179646264, email, dimaswahyuwibowoster@gmail.com)

M. Aziz Muslim adalah Dosen Jurusan Teknik Elektro Universitas Brawijaya, Malang, Indonesia, email : muh_aziz@ub.ac.id

M. Sarosa adalah Dosen Program Studi Jaringan Telekomunikasi Digital Diploma 4 Jurusan Teknik Elektro, Politeknik Negeri Malang, Malang, Indonesia, email : msarosa@yahoo.com

akurasi nilai kepadatan berkurang[7].

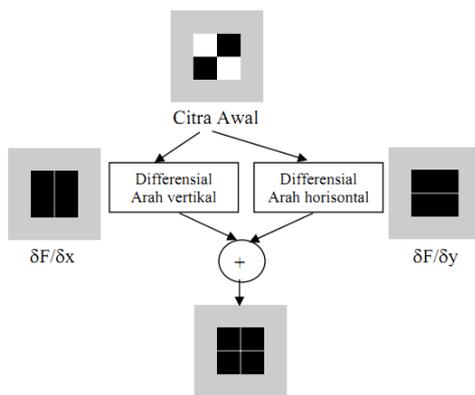
II. DASAR TEORI

A. Segmentasi Citra

Segmentasi citra bertujuan untuk membagi wilayah-wilayah yang homogen. Segmentasi adalah satu metode penting yang digunakan untuk mengubah citra input ke dalam citra output berdasarkan atribut yang diambil dari citra tersebut [8]. Segmentasi membagi citra ke dalam daerah intensitasnya masing-masing sehingga bisa membedakan antara objek dan background-nya. Pembagian ini tergantung pada masalah yang akan diselesaikan.

Pembagian citra berdasarkan kesamaan-kesamaan kriteria yang dimilikinya contohnya thresholding, region splitting, dan region merging.

B. Deteksi Tepi



Gambar 1 Proses Deteksi Tepi Citra

Deteksi Tepi adalah proses untuk menentukan perubahan intensitas yang berbeda nyata dalam sebuah bidang citra [8]. Sebuah operator deteksi tepi merupakan operasi bertetangga, yaitu sebuah operasi yang memodifikasi nilai keabuan sebuah titik berdasarkan nilai-nilai keabuan dari titik-titik yang ada di sekitarnya (tetangganya) yang masing-masing mempunyai bobot sendiri. Bobot-bobot tersebut nilainya tergantung pada operasi yang akan dilakukan, sedangkan banyaknya titik tetangga yang terlibat biasanya adalah 2x2, 3x3, 3x4, dan sebagainya. Gambar 1 menggambarkan bagaimana tepi suatu gambar diperoleh.

C. Algoritma Canny

Deteksi tepi canny ini ditemukan oleh Marr dan Hildreth yang meneliti pemodelan persepsi visual manusia. Ada beberapa kriteria pendeteksian tepian paling optimum yang dapat dipenuhi oleh algoritma Canny [9] :

- Mendeteksi dengan baik (criteria deteksi) Kemampuan untuk meletakkan dan menandai semua tepi yang ada sesuai dengan pemilihan parameter-parameter konvolusi yang dilakukan. Sekaligus juga memberikan fleksibilitas yang sangat tinggi dalam hal menentukan tingkat deteksi ketebalan tepi sesuai yang diinginkan.
- Melokalisasi dengan baik (criteria lokalisasi)

Dengan *canny* dimungkinkan dihasilkan jarak yang minimum antara tepi yang dideteksi dengan tepi yang asli.

- Respon yang jelas(kriteria respon)

D. Fuzzy C-Means

pengClusteran data yang mana keberadaan tiap-tiap data dalam suatu *Cluster* ditentukan oleh nilai keanggotaan [11]. Konsep dasar FCM, pertama kali 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 data memiliki derajat keanggotaan untuk tiap-tiap *Cluster*. Dengan cara memperbaiki pusat *Cluster* dan nilai keanggotaan tiap-tiap data secara berulang, maka akan dapat dilihat bahwa pusat *Cluster* akan bergerak menuju lokasi yang tepat.

Algoritma yang digunakan pada metode *Fuzzy C-means* adalah sebagai berikut:

- Menginputkan data yang akan di *Cluster* X Berupa matriks berukuran n x m :
n = jumlah sampel data
m = atribut setiap data.

X_{ij} = data sampel ke-i (i=1,2,...,n), atribut ke-j (j=1,2,...,m).

$$X = \begin{bmatrix} X_{11} & \dots & X_{1m} \\ \vdots & \ddots & \vdots \\ X_{n1} & \dots & X_{nm} \end{bmatrix} \quad (1)$$

- Menentukan:
Jumlah *Cluster* = c;
Pangkat = w;
Maksimum Iterasi = MaxIter;
Error Terkecil yang diharapkan = ξ ;
Fungsi Obyektif awal = 0;
Iterasi awal = t=1;

- Membangkitkan bilangan random μ_{ik}
i=1,2,...,n; k=1,2,...,c;
sebagai elemen-elemen matriks partisi awal U. Menghitung jumlah tiap kolom (atribut) :

$$\mu_{ik} = \frac{\mu_{ik}}{Q_j} \quad (2)$$

- Menghitung pusat *Cluster* ke-k: V_{kj} , dengan k=1,2,...,c; dan j=1,2,...,m

$$V_{kj} = \frac{\sum_{i=1}^n ((\mu_{ik})^w * X_{ij})}{\sum_{i=1}^n (\mu_{ik})^w} \quad (3)$$

- Menghitung fungsi Obyektif pada iterasi ke=t, Pt

$$P_t = \sum_{i=1}^n \sum_{k=1}^c \left[\sum_{j=1}^m (X_{ik} - V_{kj})^2 \right] (\mu_{ik})^w \quad (4)$$

- Menghitung perubahan matriks partisi :

$$\mu_{ik} = \frac{\left[\sum_{j=1}^m (X_{ik} - V_{kj})^2 \right]^{\frac{-1}{w-1}}}{\sum_{k=1}^c \left[\sum_{j=1}^m (X_{ik} - V_{kj})^2 \right]^{\frac{-1}{w-1}}} \quad (5)$$

dengan : $i=1,2,\dots,n$; dan $k=1,2,\dots,c$.

Cek kondisi berhenti :

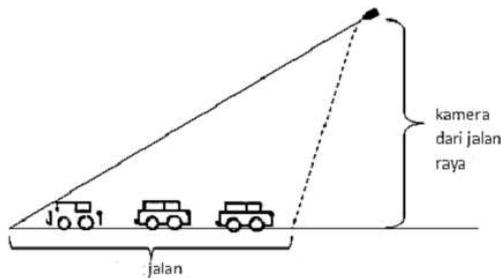
Jika $(|Pt - Pt-1| < \xi)$ atau $(t > \text{MaxIter})$ maka berhenti;

Jika tidak : $t=t+1$, ulangi langkah ke - 4.

III. METODE PENELITIAN

A. Data Primer

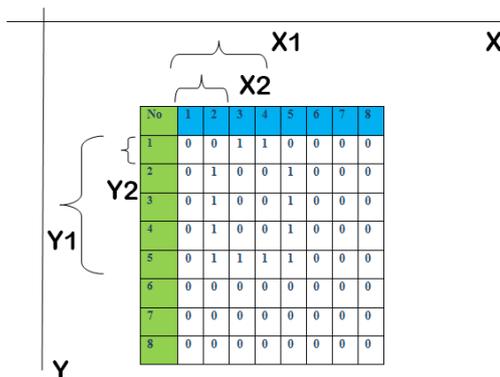
Data primer merupakan data yang didapatkan langsung dari subyek penelitian. Adapun mekanisme pengambilan gambar dilakukan pada pagi hari, siang hari, sore hari dan malam hari.



Gambar 2 Ilustrasi pengambilan gambar

Gambar 2 adalah ilustrasi pengambilan gambar yang dilakukan diatas sebuah jembatan penyeberangan dengan sudut kemiringan 45derajat. Untuk pengambilan gambar pada malam hari dilakukan dengan pencahayaan yang cukup.

B. Proses Ekstraksi Ciri

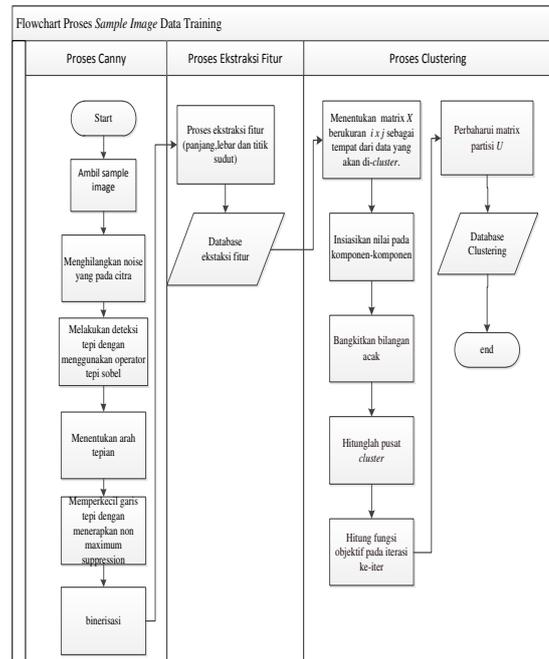


Gambar 3 proses ekstraksi

Pada gambar 3 untuk X_1 berada pada no 1 - 4 sehingga $X_1 = 4$, sedangkan untuk X_2 berada pada no 1 - 2 sehingga $X_2 = 2$. Dengan demikian lebar kendaraan dapat diketahui dengan hasil pengurangan X_1 dan X_2 . Untuk mendapatkan panjang kendaraan mengacu pada sumbu y, posisi awal angka 1 dan posisi akhir dari angka 1. Sehingga untuk panjang kendaraan dapat menggunakan rumus $Y = Y_1 - Y_2$ Untuk titik sudut

kendaraan mengacu pada sumbu x, sumbu y, posisi awal angka 1 dan posisi akhir angka 1 pada masing-masing sumbu.

C. Proses Pengambilan Sample Image untuk Data training



Gambar 3 Flowchart Proses Pengambilan Sample Image Data Training

Gambar 3 merupakan flowchart dari proses Pengambilan Sample Image Data Training. Pada flowchart tersebut terdapat tiga proses yaitu proses canny, proses ekstraksi ciri dan proses Clustering. Pada proses canny terdapat beberapa proses yaitu :

- pengambilan contoh gambar yang akan dijadikan sebagai acuan.
- menghilangkan noise pada citra
- melakukan deteksi tepi dengan menggunakan operator sobel
- menentukan arah tepian
- memperkecil garis tepi dengan menerapkan *non maximum suppression*
- binerisasi

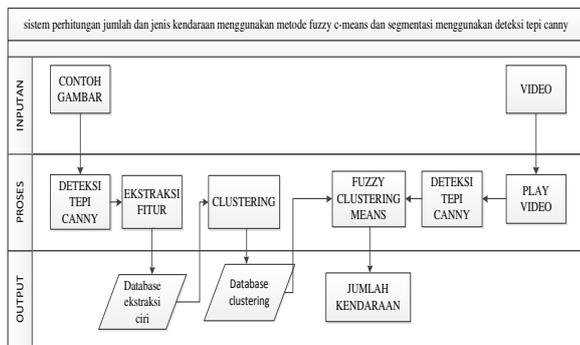
Proses selanjutnya adalah proses ekstraksi fitur dimana proses ini mengacu pada hasil akhir dari proses canny yaitu binerisasi sehingga didapatkan panjang, lebar dan titik sudut pada suatu kendaraan. Hasil dari ekstraksi ciri dimasukkan ke dalam database ekstraksi ciri untuk selanjutnya diproses dengan menggunakan Fuzzy Clustering means antara lain :

1. Menentukan matrix X berukuran $i \times j$ sebagai tempat dari data yang akan di-Cluster. Di mana i merupakan banyaknya data i dan j merupakan banyaknya atribut dari data tersebut. X_{ij} = data pada sampel ke- i dan atribut ke- j .
2. Inisiasikan nilai pada komponen-komponen berikut
 - Banyaknya Cluster yang diinginkan --> $c = 2$
 - Pangkat (pembobot) --> $w = 2$
 - Maksimum Iterasi --> $\text{maxIter} = 1$

- Error terkecil --> $e = 0.01$
 - Fungsi Objektif awal -> $P_0 = 0$
 - Iterasi awal --> iter = 1;
3. Bangkitkan bilangan acak U_{ik} , dimana banyaknya i sejumlah data dan k sejumlah *Cluster*
 4. Hitunglah pusat *Cluster* dengan rumus menggunakan rumus 3
 5. Hitung fungsi objektif pada iterasi ke-iter, menggunakan rumus 4
 6. Perbaharui matrix partisi U dengan menggunakan rumus 5

D. Desain Perancangan Sistem

Perancangan sistem dilakukan untuk mengumpulkan informasi yang berkenaan dengan aplikasi yang dibangun serta memudahkan pemahaman terhadap sistem. Dalam merancang sistem yang baik, maka harus melalui tahap-tahap perancangan sistem. Tahap-tahap perancangan sistem meliputi blok diagram sistem, konsep antarmuka dan konsep pembuatan database. Blok diagram dapat dilihat pada gambar 6 sebagai berikut ini.



Gambar 4 Blok Diagram Sistem

Gambar 4 merupakan gambar blok diagram sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny. Pada input blok diagram terdapat 2 inputan berupa contoh gambar dan video. Pada contoh gambar terdapat beberapa proses antara lain : proses deteksi tepi canny, proses ekstraksi ciri dan proses *Clustering*. Proses *Clustering* digunakan sebagai acuan proses Fuzzy C-Means pada saat video dimainkan. Sehingga didapatkan output berupa jumlah kendaraan

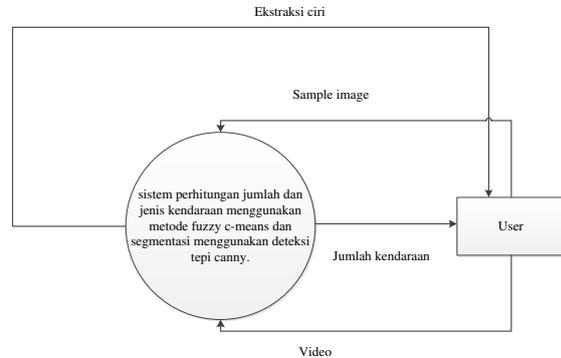
E. Data Flow Diagram (DFD)

Data Flow Diagram (DFD) menggambarkan aliran data yang terjadi di dalam sistem, sehingga dengan dibuatnya DFD ini akan terlihat arus data yang mengalir dalam sistem.

1) Context Diagram

Dalam context diagram dari sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny terdapat satu entity yaitu user. Masukan pada sistem tersebut serta hasil yang diperoleh user sebagai entity berbeda. User memberikan masukan berupa sample image, kemudian sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny akan

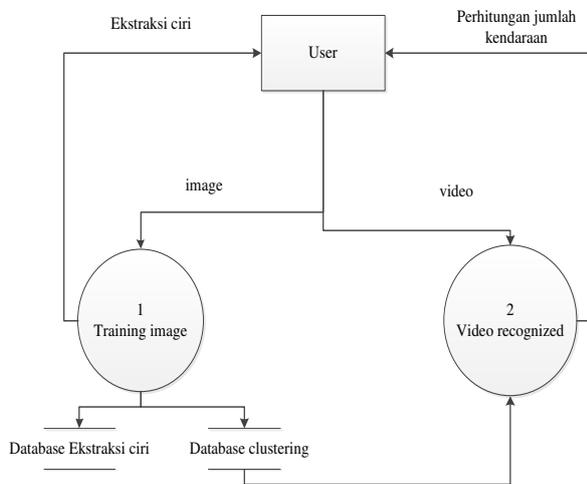
memberikan hasil dari ekstraksi ciri dari suatu image pada user. Jika user memberikan masukan video, kemudian sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny akan memberikan jumlah kendaraan. Untuk lebih detailnya dapat dilihat pada Gambar 5 berikut ini.



Gambar 5 Context Diagram sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny

2) DFD Level 0

DFD Level 0 dari sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny adalah decompose dari context diagram yang menjelaskan secara terperinci tentang proses yang ada di dalam sistem tersebut. Proses-proses tersebut yaitu proses training image dan video recognize. Untuk lebih jelasnya dapat dilihat pada Gambar 6 berikut ini.



Gambar 6 DFD Level 0 sistem perhitungan jumlah dan jenis kendaraan menggunakan metode Fuzzy C-Means dan segmentasi menggunakan deteksi tepi canny

3) DFD Level 1 Sub Proses Training Image dan Video Recognized

DFD Level 1 sub proses training image dan Video Recognized merupakan hasil decompose dari DFD Level 0 proses training image. Pada Gambar 7 proses training image terdapat beberapa sub proses antara lain :

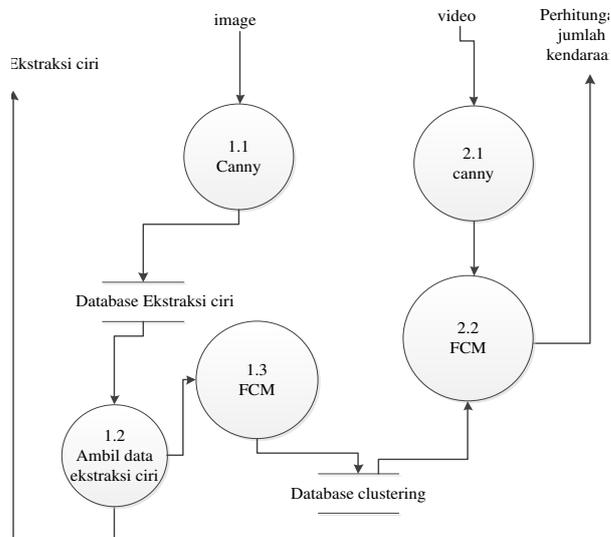
- Proses canny merupakan proses ekstraksi ciri dari sebuah *image* sehingga menghasilkan beberapa data yang disimpan kedalam database ekstraksi ciri. Hasil dari proses canny nantinya digunakan

ke dalam proses selanjutnya

- Proses ambil data ekstraksi ciri adalah proses menampilkan hasil dari ekstraksi ciri dari sebuah *image*
- Proses FCM (Fuzzy C-Means) adalah proses meng*Cluster* data-data yang didapat dari database ekstraksi ciri yang nantinya hasil dari ekstraksi ciri disimpan ke dalam database *Clustering*

Pada DFD Level 1 sub proses training image dan Video Recognized juga merupakan hasil decompose dari DFD Level 0 proses Video Recognized. Pada Gambar 9 proses Video Recognized. terdapat beberapa sub proses antara lain

- Proses canny merupakan proses ekstraksi ciri dari sebuah video sehingga menghasilkan beberapa data yang disimpan sementara kedalam *array*. Hasil dari proses canny nantinya digunakan ke dalam proses selanjutnya
- Proses FCM (Fuzzy C-Means) adalah proses meng*Cluster* data-data yang didapat dari ekstraksi ciri yang nantinya hasil dari ekstraksi ciri disimpan sementara ke dalam sebuah *Clustering* untuk dibandingkan hasilnya dengan database *Clustering* yang didapat pada proses training image. Sehingga didapatkan data berupa jumlah kendaraan dari suatu video yang dimainkan.



Gambar 7 DFD Level 1 Sub Proses Training Image dan Video Recognized

F. Metode Pengujian

Untuk mendapatkan hasil yang maksimal maka perlu adanya pengujian pada sisi aplikasi dengan menggunakan sampel video. Sebelum menggunakan sampel video untuk pengujian, diperlukan data training yang berupa jenis-jenis kendaraan seperti bus, sepeda motor, truk dll. Deteksi tepi canny akan mengekstraksi gambar untuk mendapatkan ciri-ciri dari sebuah kendaraan berdasarkan data training yang diinputkan. Setelah canny melakukan ekstraksi gambar, proses selanjutnya pengelompokkan kendaraan berdasarkan

jenisnya menggunakan Fuzzy C-Means berdasarkan ciri-ciri kendaraan yang didapatkan dari ekstraksi ciri. Setelah data training berhasil diinput proses selanjutnya dengan menggunakan video rekaman kendaraan untuk melihat performace dari aplikasi tersebut.

Pada saat video dimainkan secara bersamaan Fuzzy *Clustering* means berjalan. Pada saat ditemukan *Cluster* yang sama dengan salah satu jenis kendaraan yang tersimpan dalam database *Clustering* maka secara otomatis jumlah kendaraan akan bertambah sesuai dengan jenis kendaraan. Jika terdapat kesalahan pada aplikasi pada saat pengujian dengan video maka aplikasi belum dapat digunakan untuk membantu dalam proses analisis dan perlu adanya perbaikan. Adapun fungsi dilakukan perbaikan aplikasi adalah untuk mendapatkan hasil yang maksimal pada saat analisis.

IV. HASIL DAN PEMBAHASAN

A. Representasi Pengetahuan

Proses representasi pengetahuan dilakukan dengan cara mengumpulkan pengetahuan-pengetahuan pokok dan aturan-aturan yang digunakan dalam pemanfaatan aplikasi sebagai alat bantu untuk menghitung jumlah kendaraan yang disusun berdasarkan :

- *Segmentasi* menggunakan deteksi tepi canny berdasarkan *capture* gambar setiap 1 detik
- Ekstraksi fitur berdasarkan hasil *segmentasi* dan diperoleh berupa panjang, lebar dan titik sudut dari sebuah kendaraan
- Proses *Clustering* menggunakan FCM untuk memperoleh jumlah kendaraan berdasarkan jenisnya dari sebuah video

Hasil tersebut diatas merupakan blok-blok sub sistem yang mempunyai perhitungan yang berdiri sendiri. Akan tetapi merupakan suatu kesatuan utuh yang tidak dapat dipisahkan salah satunya untuk membangun sistem perhitungan jumlah dan jenis kendaraan, adapun penjelasannya sebagai berikut :

1. *Segmentasi* menggunakan deteksi tepi canny

Deteksi tepi canny dapat mendeteksi tepian yang sebenarnya dengan tingkat error minimum dengan kata lain, operator canny didesain untuk menghasilkan citra tepian yang optimal. Untuk melakukan *segmentasi* menggunakan deteksti tepi canny langkah-langkah yang harus dilakukan adalah sebagai berikut :

- Menghilangkan *noise* yang ada pada citra dengan mengimplementasikan Filter Gaussian. Hasil citra akan tampak lebih buram. Hal ini dimaksudkan untuk mendapatkan tepian citra yang sebenarnya. Bila tidak dilakukan maka garis-garis halus juga akan dideteksi sebagai tepian. Kernel dari gaussian dengan standar deviasi $\sigma = 1.4$ ditunjukkan dalam persamaan

$$B = \frac{1}{159} \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix} \quad (6)$$

- Melakukan deteksi tepi dengan operator deteksi

tepi sobel dengan melakukan pencarian secara horizontal (G_x) dan secara vertikal (G_y). Berikut ini deteksi tepi menggunakan operator deteksi tepi sobel

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (7)$$

Hasil dari kedua operator digabungkan untuk mendapatkan hasil gabungan tepi vertikal dan horizontal dengan menggunakan rumus

$$|G| = \sqrt{G_x^2 + G_y^2}$$

$$|G| = |G_x| + |G_y| \quad (8)$$

- Menentukan arah tepian yang ditemukan dengan menggunakan rumus

$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right) \quad (9)$$

Dan selanjutnya membagi ke dalam 4 warna sehingga garis dengan arah berbeda memiliki warna yang berbeda. Pembagiannya adalah :

- derajat 0 – 22,5 dan 157,5 – 180 berwarna kuning
- derajat 22,5 – 67,5 berwarna hijau dan
- derajat 67,5 – 157,5 berwarna merah

Memperkecil garis tepi yang muncul dengan menerapkan non maximum suppression sehingga menghasilkan garis tepian yang lebih ramping

- Melakukan binerisasi dengan menerapkan dua buah nilai ambang.

2. Ekstraksi fitur berdasarkan proses segmentasi

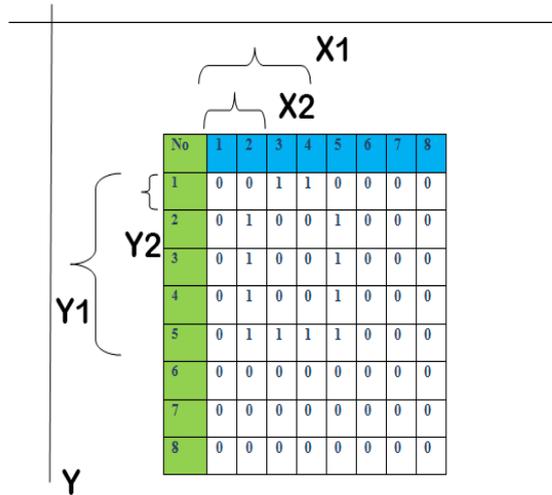
Berdasarkan pada Tabel 1 didapatkan hasil akhir dari sebuah citra adalah citra biner. Dimana dari citra biner tersebut akan digunakan untuk proses ekstraksi fitur untuk mendapatkan informasi yaitu panjang kendaraan, lebar kendaran, dan titik sudut kendaraan.

TABEL I
TABEL BINER SEDAN

		X							
		1	2	3	4	5	6	7	8
Y	1	0	0	1	1	0	0	0	0
	2	0	1	0	0	1	0	0	0
	3	0	1	0	0	1	0	0	0
	4	0	1	0	0	1	0	0	0
	5	0	1	1	1	1	0	0	0
	6	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0

Berdasarkan pada Tabel I dan Gambar 8 untuk mendapatkan lebar dari sebuah kendaraan digunakan sumbu x sebagai acuan dan posisi angka 1 dalam Tabel biner. Pada gambar 12 untuk X_1 berada pada no 1 – 4 sehingga $X_1=4$, sedangkan untuk X_2 berada pada no 1 – 2 sehingga $X_2=2$. Dengan demikian lebar kendaraan dapat diketahui dengan hasil pengurangan X_1 dan X_2 . Untuk mendapatkan panjang kendaraan mengacu pada

sumbu y, posisi awal angka 1 dan posisi akhir dari angka 1. Sehingga untuk panjang kendaraan dapat menggunakan rumus $Y=Y_1-Y_2$. Untuk titik sudut kendaraan mengacu pada sumbu x, sumbu y, posisi awal angka 1 dan posisi akhir angka 1 pada masing-masing sumbu.



Gambar 8 perhitungan jumlah dan jenis berdasarkan sumbu X dan sumbu Y



Gambar 9 hasil ekstraksi fitur

ID	Nama	Sudut	Panja...	Lebar
9	MPV	8	60	45
10	MPV	8	68	38
11	MPV	4	45	38
12	MPV	8	63	53
13	MPV	8	63	53
14	Bus	6	129	71
15	Bus	6	156	86
16	MPV	6	68	45
17	MPV	6	68	45
18	MPV	8	71	48
19	MPV	8	75	53
20	MPV	8	75	53
21	Bus	6	132	72
22	Bus	8	120	60
23	MPV	8	120	60

Gambar 10 Hasil dari ekstraksi ciri

Sehingga berdasarkan gambar 10 dapat diketahui panjang kendaraan MPV adalah 60, lebar 45 dan titik

sudut 8. Pada gambar 9 image yang digunakan sebagai training berukuran 200x300 dan hasil dari ekstraksi disimpan dalam database untuk dilakukan proses selanjutnya yaitu proses *Clustering*.

3. *Clustering* menggunakan FCM

Setelah proses ekstraksi dilakukan langkah selanjutnya adalah melakukan proses *Clustering*. Pada gambar 13 adalah hasil dari ekstraksi ciri yang nantinya digunakan pada proses selanjutnya pada proses *Clustering*. Proses *Clustering* dilakukan berdasarkan keluaran dari ekstraksi fitur yaitu panjang, lebar dan titik sudut kendaraan, sebagai contoh bus pada gambar 13 terdapat 4 bus dengan karakteristik yang berbeda. Hal ini disebabkan karena ada beberapa jenis bus antara lain bus dengan ukuran kecil, bus dengan ukuran sedang dan bus dengan ukuran besar dan memiliki informasi yang berbeda-beda. Untuk kasus ini dapat menjadi 1 *Cluster* yaitu *Cluster* bus. Adapun langkah-langkah *Clustering* :

- Menentukan matrix X berukuran $i \times j$ sebagai tempat dari data yang akan di-*Cluster*. Di mana i merupakan banyaknya data i dan j merupakan banyaknya atribut dari data tersebut. X_{ij} = data pada sampel ke- i dan atribut ke- j . Berdasarkan pada ukuran *image* yang digunakan berukuran 200x300 maka nilai dari $i=200$ dan $j=300$

TABEL II
Matrik X yang berukuran i=200 dan j=300

data ke -i	atribut		
	panjang	lebar	titik sudut
1	129	71	6
2	156	86	6
3	132	72	6
4	120	60	8

- Inisiasikan nilai pada komponen-komponen berikut
 - Banyaknya *Cluster* yang diinginkan $c=2$
 - Pangkat (pembobot) $w=2$
 - Maksimum Iterasi $\rightarrow \maxIter=1$
 - Error terkecil $\rightarrow e=0.01$
 - Fungsi Objektif awal $\rightarrow P_0=0$
 - Iterasi awal $\rightarrow iter=1$;

- Bangkitkan bilangan acak U_{ik} , dimana banyaknya i sejumlah data dan k sejumlah *Cluster*

TABEL III
BILANGAN ACAK U_{ik}

i	k1	k2
1	0,3	0,7
2	0,2	0,8
3	0,4	0,6
4	0,5	0,5

- Hitunglah pusat *Cluster* dengan rumus

menggunakan rumus 3

TABEL IV
PUSAT CLUSTER

vkj	1	2	3	4
1	134,37103	73,62966	6	0,1273825
2	161,456758	87,751775	7,342879	0,3123851

- Hitung fungsi objektif pada iterasi ke-iter, menggunakan rumus 4

TABEL V
FUNGSI OBJEKTIF

	kluster1		kluster2		
	$(X_{ij}-V_{kj})^2$	uik	$(X_{ij}-V_{kj})^2$	p	
35,804	0,09	3,222	1335,836	0,49	654,56001
620,701	0,04	24,828	34,642133	0,64	22,170965
8,2877	0,16	1,326	1117,5952	0,36	402,33426
396,287	0,25	99,0719	2489,2193	0,25	622,30483

- Perbaharui matrix partisi U dengan menggunakan rumus 5

TABEL VI
Matrix partisi U

i	k1	k2
1	0,3034552	9,06654478
2	0,9919311	3,25756152
3	0,8457174	5,13561931
4	0,2941002	4,40545336

Berdasarkan Tabel.VI bus kecil, sedang dan bus besar dapat menjadi 1 *Cluster* yaitu bus.

B. Perhitungan jumlah kendaraan

Pengujian ini adalah pengujian perhitungan yang dilakukan oleh aplikasi(apl) sistem cerdas yang dibandingkan dengan pengujian dengan perhitungan manual(mn)

TABEL VII
HASIL PERHITUNGAN MANUAL DAN APLIKASI

	Video mobil.mpeg		Video sore1.mpeg		Video malam2.mpeg	
	mn	apl	mn	apl	mn	apl
mpv	9	7	14	13	9	4
motor	40	30	34	23	6	0
bus	1	1	0	0	1	1
sedan	1	1	0	0	1	0
truck	1	1	3	3	1	1
total	52	40	49	37	18	6
presentase	90%		85%		49%	

TABEL VIII
HASIL PERHITUNGAN MANUAL DAN APLIKASI

	Video siang.mpeg		Video siang1.mpeg		Video siang2.mpeg	
	man	apl	man	apl	man	apl
	mpv	8	7	13	10	14
motor	45	35	45	36	55	32
bus	1	1	2	2	2	2
sedan	0	0	0	0	0	0
truck	1	1	4	4	2	2
total	55	44	64	52	73	50
presentase	91%		89%		89%	

Pada Tabel VII dan Tabel VIII tingkat keakuratan aplikasi didapatkan dengan membagi total presentase ketelitian software dengan jumlah tipe kendaraan yang diteliti dan tipe kendaraan yang ada pada video. Pada Tabel VII untuk kondisi di malam hari didapatkan presentase keakuratan aplikasi 49% hal ini disebabkan dikarenakan pencahayaan yang kurang dan kendaraan yang terekam berwarna gelap sehingga aplikasi tidak dapat mengenali kendaraan dengan baik. kesimpulan dan saran

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Dari hasil perancangan, analisis, desain, implementasi dan pengujian sistem maka dapat disimpulkan hal-hal sebagai berikut:

- Belum dapat mengukur kecepatan kendaraan yang dapat terdeteksi oleh Fuzzy Clustering Means
- Beberapa kendaraan memiliki nilai variabel yang sama, sehingga dapat menimbulkan kesalahan dalam deteksi kendaraan
- Pencahayan dari sinar matahari pada siang hari dapat menimbulkan bayangan pada kendaraan sehingga dapat mempengaruhi dalam menentukan panjang, lebar dan sudut kendaraan.
- Pengambilan gambar pada malam hari dengan pencahayan lampu kota, hanya kendaraan dengan warna cerah yang dapat terdeteksi sedangkan warna gelap tidak dapat terdeteksi. Hal ini disebabkan karena pada saat proses canny pada kendaraan berwarna gelap tidak dapat menghasilkan tepi yang baik sedangkan pada warna cerah tepi pada kendaraan masih dapat terlihat dengan baik. Sehingga mempengaruhi

proses perhitungan dengan menggunakan Fuzzy C-Means

- Belum dapat mengenali kendaraan secara spesifik berdasarkan pada merek hal ini disebabkan ada beberapa jenis mobil mempunyai variabel yang sama.

B. Saran

Berikut beberapa saran dalam penelitian ini sehingga dapat lebih ditingkatkan hasil gunanya yaitu :

- Penelitian ini dapat ditingkatkan manfaatnya dengan cara penerapan metode Tsukamoto untuk mengenali jenis kendaraan lebih detail atau untuk mengukur kecepatan kendaraan.
- Metode Tsukamoto dapat diteliti lebih lanjut untuk menghasilkan penemuan baru yang lebih bermanfaat

DAFTAR PUSTAKA

- [1] Abubakar, I., 2012. http://id.wikibooks.org/wiki/Moda_Transportasi/Moda_Transportasi_Jalan. [Online] [Accessed 6 Augustus 2012].
- [2] Indira, M., 2008. PERBANDINGAN METODE PENDETEKSI TEPI STUDI KASUS : CITRA USG JANIN. *Proceeding, Seminar Ilmiah Nasional Komputer dan Sistem Intelijen (KOMMIT 2008)*, pp.366 - 373.
- [3] Winarno, E., 2011. Aplikasi Deteksi Tepi pada Realtime Video menggunakan Algoritma Canny Detection. *Jurnal Teknologi Informasi DINAMIK*, 16, No.1, pp.44-49.
- [4] Y. Ramadevi, 2010. Segmentation And Object Recognition Using Edge Detection Techniques. *International Journal of Computer Science & Information Technology (IJCSIT)*, II, pp.153 - 161.
- [5] Luthfi, E.T., 2007. FUZZY C-MEANS UNTUK CLUSTERING DATA (STUDI KASUS : DATA PERFORMANCE MENGAJAR DOSEN). *Seminar Nasional Teknologi 2007 (SNT 2007)*, pp.1-7.
- [6] Thiang, S., 2011. Character Recognition Using Fuzzy Clustering. *Konferensi Nasional Sistem Informasi (KNSI)*, pp.1-4.
- [7] Hadi, S. & Samara, Y.R., 2012. DETEKSI OBJEK KENDARAAN PADA CITRA DIJITAL JALAN RAYA. *Jurnal Ilmiah Ilmu Komputer*, Vol. 8 No. 2.
- [8] Sutoyo, T.d., 2009. Teori Pengolahan Citra Digital. Yogyakarta : Andi.
- [9] CANNY, J., 1986. A Computational Approach to Edge Detection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, PAMI-8, NO. 6, pp.679 - 698.
- [10] Sadaaki Miyamoto, Hidetomo Ichihashi, Katsuhiro Honda: Algorithms for Fuzzy Clustering methods in C-means Clustering with applications. *Studies in Fuzziness and soft computing* 229, springer 2008, ISBN 978-3-540-78736-5, pp. 1-233.
- [11] BEZDEK, J.C., 1984. FCM: THE FUZZY c-MEANS CLUSTERING ALGORITHM. *Computers & Geosciences*, 10, No 2-3, pp.191-203.