

Menentukan Jumlah Kendaraan pada Persimpangan Jalan Melalui *Image Processing* Menggunakan Metode Jaringan Fungsi Basis Radial (JFBR) pada *Traffic Light* Cerdas

Yoyok Heru Prasetyo Isnomo*, Aji Prasetya Wibawa*

Universitas Negeri Malang, Jl. Semarang No. 5 Malang, Jawa Timur, Indonesia

*Penulis korespondensi, Surel: urehkoyoy@gmail.com

Paper received: Paper received: 01-9-2021; revised: 11-9-2021; accepted: 17-9-2021

Abstract

The number of large, light, and heavy vehicles continues to grow, while the addition of road expansion has not been adequate, resulting in traffic jams, especially in big cities. Conventional, automatic, and adaptive control systems that existed at that time were still unable to overcome the problem of traffic jams. This article proposes to calculate the number of vehicles on each signaled road section, which is applied at each phase of the red, yellow, and green light periods, using a combination of a Radial Basis Function Network (RBFN) and K-Means methods. The results of the calculation of the vehicles are used to set the duration of the red, yellow, and green lights. The experimental results show that the system can calculate and recognize large, small, super large and non-engine vehicles with an error rate of 0.05397.

Keywords: *jfbr; traffic light; k-means; image processing*

Abstrak

Jumlah kendaraan tipe besar, ringan, dan berat terus bertambah, sedangkan penambahan perluasan jalan belum berimbang, sehingga memberi dampak kemacetan lalu lintas, terutama pada kota-kota besar. Sistem pengontrolan konvensional, otomatis dan adaptif yang ada pada saat itu masih belum bisa mengatasi permasalahan kemacetan arus kendaraan. Artikel ini mengusulkan penghitungan jumlah kendaraan pada masing-masing ruas jalan bersinyal, yang dilakukan pada setiap fase periode lampu nyala merah, kuning dan hijau, dengan menggunakan gabungan metode Jaringan Fungsi Basis Radial (JFBR) dan K-Means. Hasil perhitungan kendaraan digunakan untuk mensetting lama nyala lampu merah, kuning, dan hijau. Hasil eksperimen menunjukkan bahwa sistem mampu menghitung dan mengenali jenis kendaraan besar, ringan, berat, dan tanpa mesin dengan tingkat kesalahan rata-rata sebesar 0,05397.

Kata kunci: *jfbr; traffic light; k-means; image processing*

1. Pendahuluan

Berdasarkan data kementerian perhubungan jumlah kendaraan bermotor di Indonesia mencapai 141,99 juta unit pada tahun 2021, jumlah tersebut meningkat 4,30% dari tahun sebelumnya sebanyak 136,14 juta unit (<https://dataindonesia.id>, 2021). Pertambahan jumlah kendaraan tidak diimbangi oleh penambahan jumlah jalan atau pelebaran jalan akan berdampak pada kemacetan arus lalu lintas. Salah satu solusi untuk mengurangi tingkat kemacetan adalah pengaturan lampu lalu lintas yang efisien (Marta L. S., 2018). Kemacetan arus lalu lintas merupakan masalah yang serius, terutama kota-kota besar seperti Jakarta. Analisis transportasi pada tahun 2016 menyebutkan bahwa kemacetan menyebabkan rata-rata rakyat Indonesia membuang waktu di jalan selama 47 jam dalam setahun (Billie P., 2018).

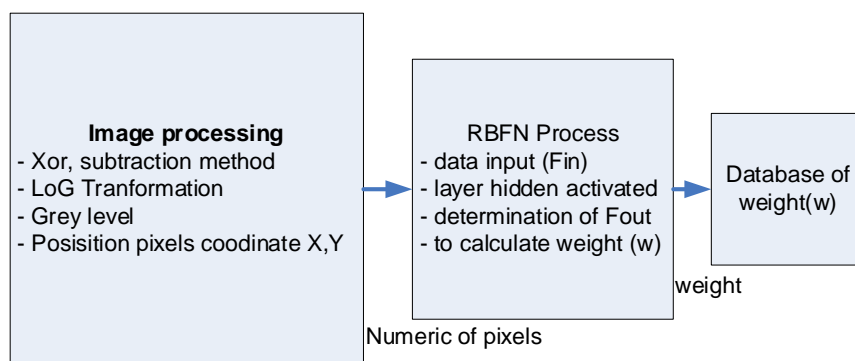
Sistem pengontrolan lampu lalu lintas konvensional tidak mempunyai kecerdasan dalam mengatur lama nyala lampu merah, kuning, dan hijau, sehingga cenderung menimbulkan kemacetan. Sistem *traffic light* berbasis kepadatan cerdas dengan menggunakan sensor infra-red akan dapat mengatasi kemacetan lalu lintas (Anam F., 2020). Bagaimanapun,

sensor infra-red sebagai pendeteksi tingkat kepadatan arus lalu lintas, dalam operasinya mempunyai beberapa kelemahan antara lain: tidak bisa membedakan jenis kendaraan (besar, sedang, kecil, kendaraan roda empat atau kendaraan beroda dua), tidak bisa memprediksi jumlah kendaraan, sehingga sistem ini kurang akurat dalam memberikan informasi tingkat kepadatan arus lalu lintas. Sistem kontrol traffic light didasarkan pada artificial *intelligence* (AI) untuk performa efektif, menggunakan video untuk merekam sinyal kendaraan pada empat jalur jalan di Dehradun, hasil rekaman digunakan untuk menghitung semua jenis kendaraan (Anuj S., 2021). Kelemahan dari sistem ini adalah tidak melakukan perhitungan jumlah kendaraan setiap periode putaran pergantian lampu merah dan hijau, dengan hasil pencapaian akurasi penghitungan kendaraan pada jarak pandang tertinggi rata-rata 96,15% untuk pagi hari, sedangkan akurasi kabut/jarak pandang rendah pada malam hari yaitu 66,66%.

Sistem yang kami usulkan menggunakan sensor CCTV IP untuk men-*capture* kondisi kepadatan kendaraan secara real time sebagai bahan dasar pemrosesan image, sedangkan penghitungan jumlah kendaraan menggunakan gabungan metode K-Means dan JFBR. Peng-*capture*-an dan penghitungan jumlah kendaraan dilakukan setiap periode putaran lampu nyala merah dan hijau. Jumlah kendaraan masing-masing ruas jalan digunakan sebagai dasar menentukan lama menyala lampu merah, kuning dan hijau pada traffic light. Sistem kontrol mampu memprediksi kecepatan kendaraan, dengan menggunakan persamaan kecepatan dan sistem dapat menentukan kehilangan waktu secara otomatis, dalam menentukan lama menyala lampu kuning dengan tepat (YHP. Isnomo, 2014).

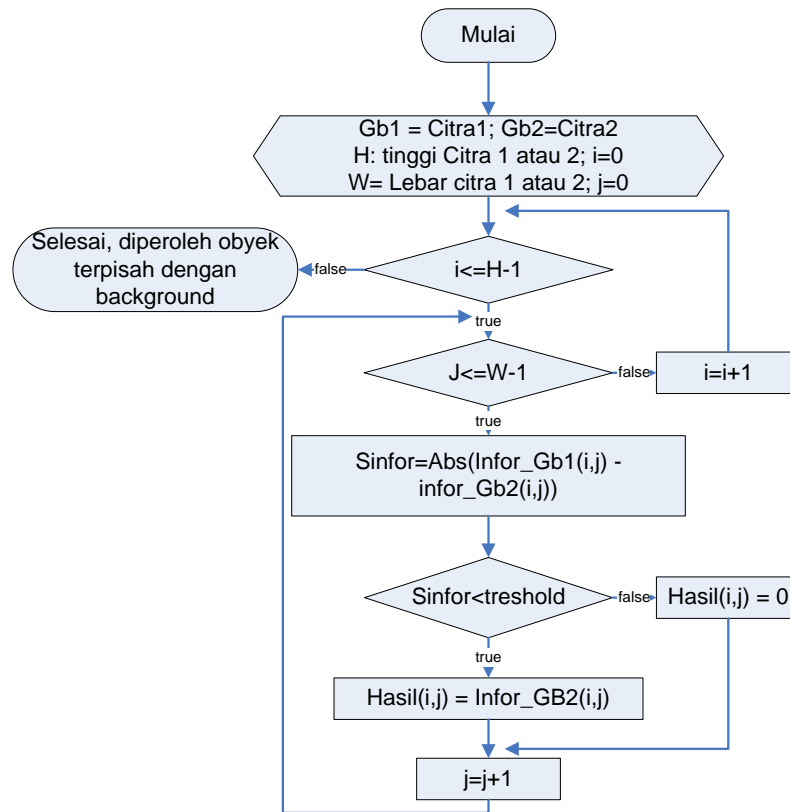
2. Metode

Tempat pemasangan prototipe alat serta pengambilan data dilakukan di jalan persimpangan tiga traffic light jalan borobudur kecamatan Blimbing dekat dengan Masjid Sabilillah pada tahun 2014. Metode yang diusulkan untuk menghitung jumlah kendaraan melalui *image processing* menggunakan gabungan JFBR dan K-Means dengan 3 tahap utama yaitu: (a) image processing (capture citra dengan CCTV, pemisahan obyek dengan *background*, deteksi tepi), (b) Pembelajaran model JFBR, (c) pembobotan, blok diagram utama di tunjukkan pada Gambar 1,



Gambar 1. Blok Diagram Utama Perhitungan Jumlah Kendaraan

Capture arus kendaraan masing-masing ruas jalan dilakukan pada tiap fase periode, dimana CCTV akan meng-*capture* secara otomatis berdasarkan program kontrol yang di install pada memori primer alat kontrol. Selanjutnya adalah proses memisahkan antara obyek dengan *background* dari citra yang sudah di capture oleh CCTV, dengan menggunakan metode XOR. Alur logika pemisahan obyek dengan background di tunjukkan oleh Gambar 2,



Gambar 2. Flowchart Pemisahan Obyek dan Background

Variabel Gb1 dan Gb2 pada gambar 2. merupakan variabel pengganti citra *background* dan citra obyek, yaitu Gb1 pengganti citra *background*, dan Gb2 adalah citra obyek termasuk *background*, jika citra ini di capture dengan peralatan yang sama maka resolusi dan *aspect ratio* antara Gb1 dan Gb2 adalah sama, sehingga tinggi dan lebar citra 1 dan 2 adalah sama. *Threshold* adalah batas ketidak miripan informasi citra 1 dengan informasi citra 2, yang nilainya ditentukan dari hasil analisis keadaan secara *real time*.

Deteksi tepi bertujuan untuk mendapatkan batas yang jelas antara wilayah satu dengan wilayah yang lain, dimana antar wilayah memiliki perbedaan informasi piksel yang cukup besar. Hasil pengujian tekstur citra menggunakan Metode *Laplacian of Gaussian (LoG)*, menunjukkan hasil yang maksimal dan memuaskan (Rocky H., 2020). Dengan merujuk hasil penelitian ini maka metode deteksi tepi yang digunakan adalah *LoG*. Transformasi Laplace pada citra berfungsi sebagai filter derivatif untuk mencari daerah perubahan secara cepat (tepi) citra, dengan persamaan berikut,

$$L(x, y) = \frac{\delta^2 I}{\delta x^2} + \frac{\delta^2 I}{\delta y^2} \quad (1)$$

Karena informasi piksel citra dalam bentuk sekumpulan data diskrit, maka persamaan (1) dikonversikan dalam bentuk kernel diskrit seperti dibawah,

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Karena metode Laplace ini sangat sensitif terhadap noise, maka di gabungkan dengan filter *Gaussian* dimensi 2, dinyatakan oleh persamaan (2) berikut,

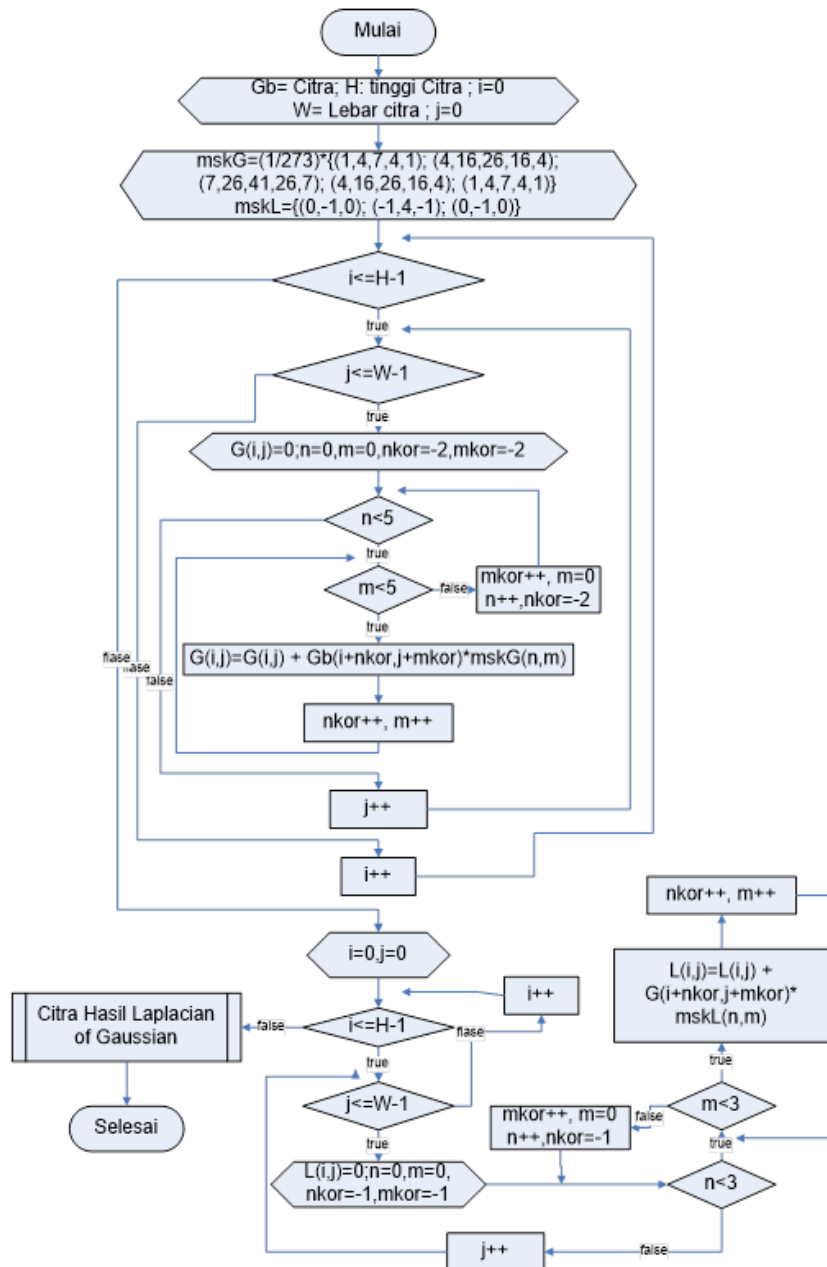
$$g(x, y, t) = \frac{1}{2\pi t^2} e^{-(x^2+y^2)/(2t)} \quad (2)$$

dengan pendekatan kernel konvolusi *Gaussian* yang sesuai dengan persamaan (2) adalah,

Tabel 1. Kernel *Gaussian*

	1	4	7	4	1
1/273	4	16	26	16	4
	7	26	41	26	7
	4	16	26	16	4
	1	4	7	4	1

Menggabungkan kedua metode laplace dengan *Gaussian* dikenal dengan istilah *Laplacian of Gaussian (LOG)* maka dapat disusun alur logika seperti Gambar 3,

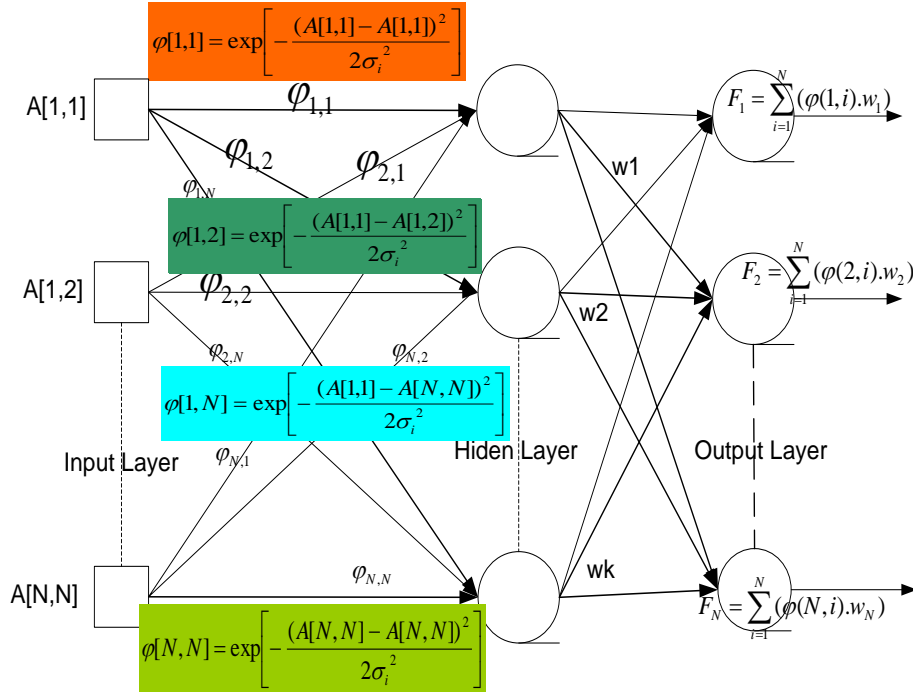


Gambar 3. Alur Logika Deteksi Tepi dengan LoG

- dengan,
- Gb : Gambar atau citra
 - maskG : masking *Gaussian*
 - maskL : masking *Laplacian*
 - G : nilai fungsi *Gaussian* lapisan *hidden*

Proses Konversi Nilai Pixel Kedalam Nilai Fungsi Gauss Pada Lapisan Hidden Dalam Proses Pembelajaran, model ini menggunakan matriks nilai hasil deteksi tepi sebagai data input, yang kemudian ditransformasikan terhadap nilai matriks itu sendiri sebagai pusat data input kemudian dimasukkan ke dalam fungsi Gauss untuk mengaktifkan lapisan hidden. Langkah yang dilakukan dalam merancang sistem jaringan sistem tiruan (JST) adalah sebagai berikut.

1. Desain struktur JST JFBR ditunjukkan pada Gambar (4). Pada struktur ini tampak bahwa lapisan *hidden* diaktifkan oleh fungsi Gauss, dengan ketentuan jika data piksel mempunyai informasi maka piksel sebagai data input jika tidak maka data diabaikan. Maka struktur JST yang dirancang adalah sebagai berikut;

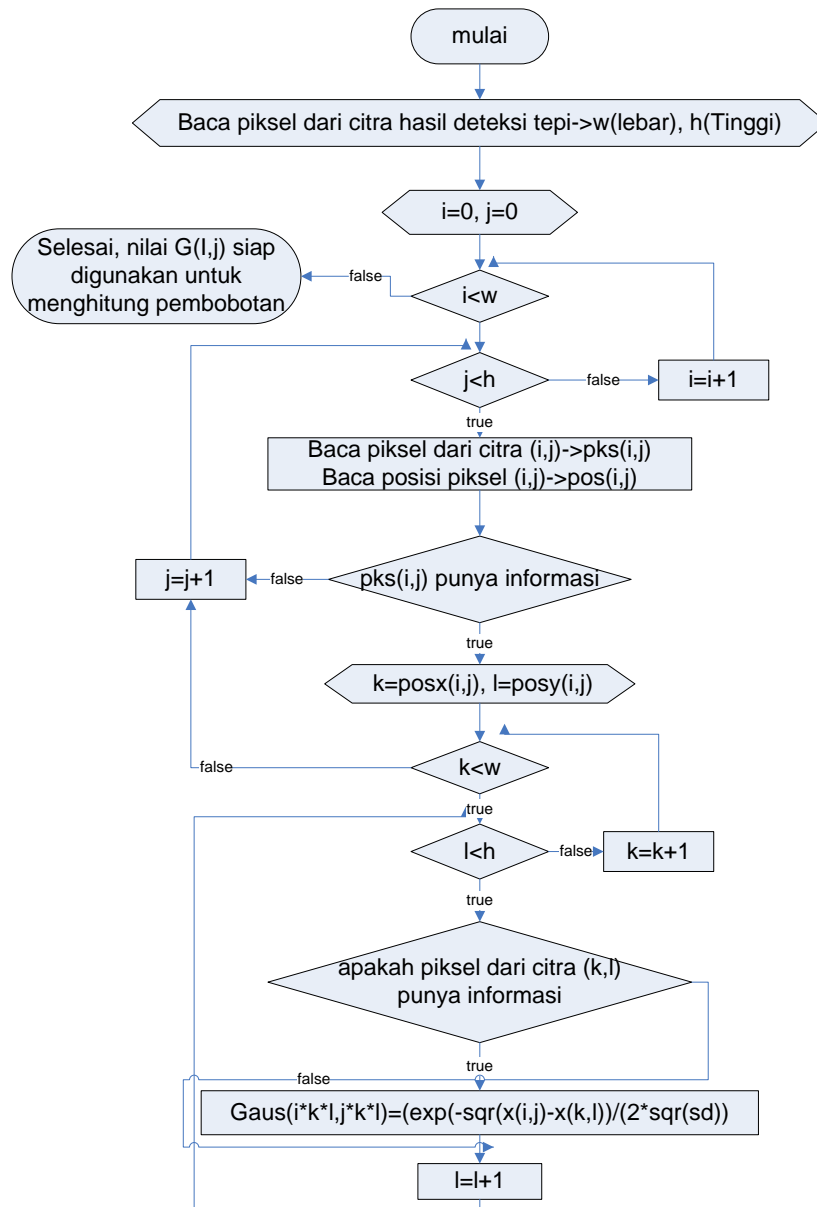


Gambar 4. Struktur JST model JFBR dengan N output,

2. Matriks fungsi Gauss $\phi_{k,k}$ pada Gambar 5. dengan k bernilai N, dan N adalah nilai yang berubah-ubah besarnya, hal ini disebabkan oleh faktor ada atau tidaknya informasi dalam sebuah piksel citra. Nilai fungsi Gauss dapat peroleh dengan menggunakan alur logika yang ditunjukkan oleh Gambar 6.

$$\phi_{k,k} = \begin{pmatrix} \exp\left[-\frac{(A_{1,1} - A_{1,1})^2}{2\sigma_i^2}\right] & \exp\left[-\frac{(A_{1,1} - A_{1,2})^2}{2\sigma_i^2}\right] & \dots & \exp\left[-\frac{(A_{1,1} - A_{N,N})^2}{2\sigma_i^2}\right] \\ \exp\left[-\frac{(A_{2,1} - A_{1,1})^2}{2\sigma_i^2}\right] & \exp\left[-\frac{(A_{2,1} - A_{1,2})^2}{2\sigma_i^2}\right] & \dots & \exp\left[-\frac{(A_{2,1} - A_{N,N})^2}{2\sigma_i^2}\right] \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \exp\left[-\frac{(A_{N,N} - A_{1,1})^2}{2\sigma_i^2}\right] & \exp\left[-\frac{(A_{N,N} - A_{1,2})^2}{2\sigma_i^2}\right] & \dots & \exp\left[-\frac{(A_{N,N} - A_{N,N})^2}{2\sigma_i^2}\right] \end{pmatrix}$$

Gambar 5. Matrik Fungsi Gauss



Gambar 6. Diagram Alir Matrik Fungsi Gauss pada Lapisan *Hidden*

- Menghitung nilai bobot dengan menggunakan fungsi JFBR yang diimplementasikan pada transformasi matrik invers, ditunjukkan oleh gambar 7.

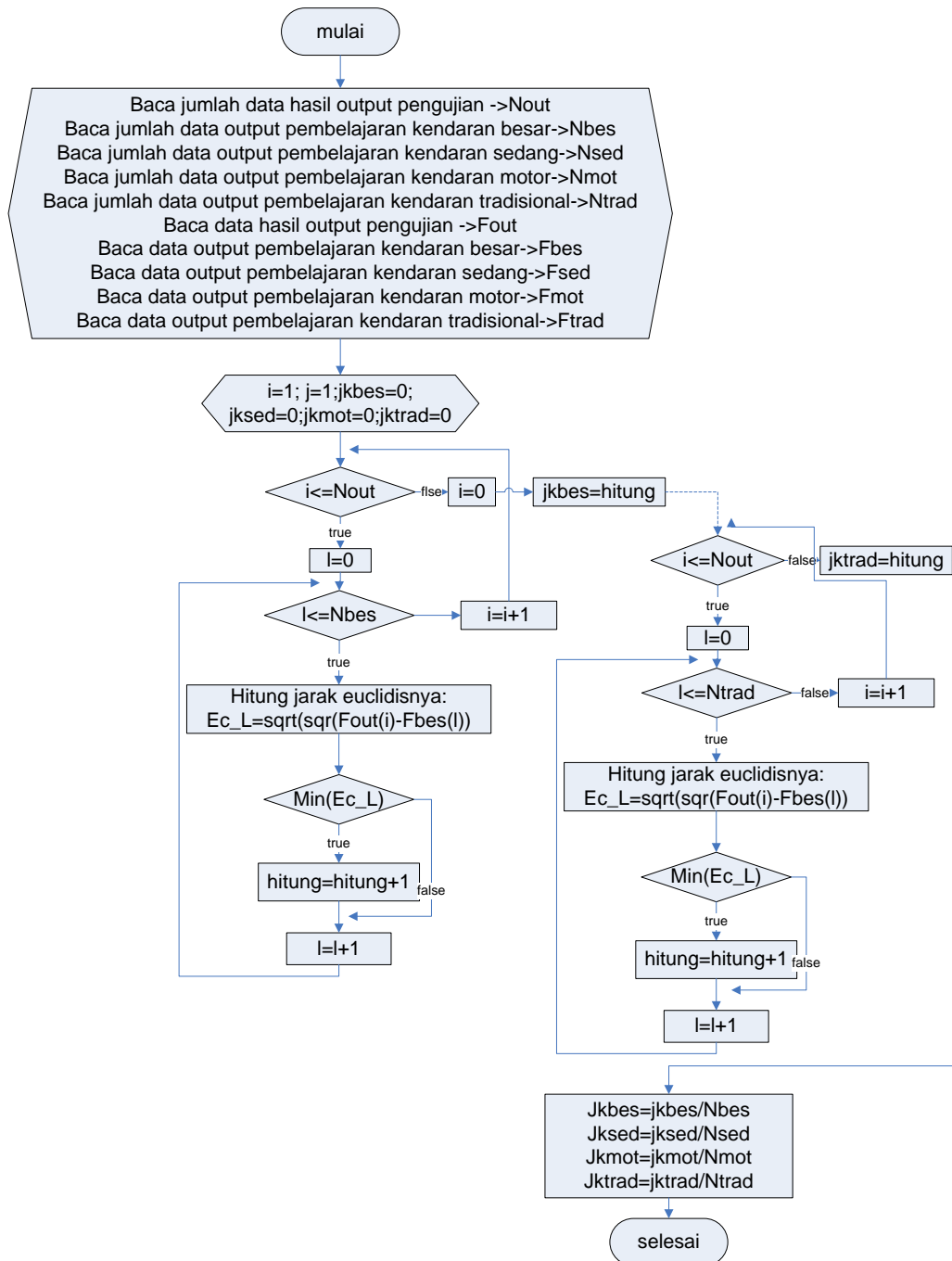
$$\begin{matrix} \mathbf{W}_1 \\ \mathbf{W}_2 \\ \vdots \\ \vdots \\ \mathbf{W}_N \end{matrix} = \begin{matrix} \varphi_{1,1} & \varphi_{1,2} & \dots & \varphi_{1,N} \\ \varphi_{2,1} & \varphi_{2,2} & \dots & \varphi_{2,N} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \varphi_{N,1} & \varphi_{N,2} & \dots & \varphi_{N,N} \end{matrix}^{-1} \begin{matrix} \mathbf{F}_1 \\ \mathbf{F}_2 \\ \vdots \\ \vdots \\ \mathbf{F}_N \end{matrix}$$

Gambar 7. Matrik Nilai Bobot

Metode Jaringan Fungsi Basis Radial (JFBR) dengan Cluster K-Means untuk Prediksi Jumlah Kendaraan, Model ini menggunakan matriks nilai integer sebagai data input, yang diperoleh dari hasil deteksi tepi citra arus kendaraan secara real time. Prinsip kerja pengujian sama dengan pembelajaran, perbedaannya terletak pada input data, vektor data pusat, bobot dan output. Langkah-langkah proses pengujian dapat disusun sebagai berikut.

- Siapkan data input, bobot, dan output hasil pembelajaran.
- Siapkan data hasil deteksi tepi, baca informasi piksel dalam nilai integer.
- Apakah piksel data mempunyai informasi, jika ya maka data piksel diproses sebagai data input, jika tidak diabaikan
- Data yang mempunyai informasi baca nilai posisinya dalam x dan y,
- Gunakan nilai x dan piksel sebagai input data JFBR
- Aktifkan lapisan hidden dengan persamaan (2.4) dengan menggunakan vektor data masing-masing input pembelajaran sebagai vektor data pusat
- Dengan menggunakan pembobotan hasil pembelajaran hitung nilai output,
- Gunakan *cluster K-Means* untuk mengelompokkan data output, berdasarkan output hasil pembelajaran.
- Hitung jumlah cluster, jumlah cluster menunjukkan jumlah kendaraan dengan jenis yang sama
- Ulangi langkah no. 6 s.d. 9 untuk data input pembelajaran atau vektor data pusat untuk jenis kendaraan yang lain, dan bobot kendaraan jenis lain.

Matrik output hasil pengujian untuk jenis kendaraan yang sama di kelompokkan dengan metode *cluster K-Means* dengan alur logika ditunjukkan gambar 8. sebagai berikut,



Gambar 8. Proses Cluster K-Means, dan Perhitungan Jumlah Kendaraan

3. Hasil dan Pembahasan

Data yang digunakan pada proses pengujian adalah data yang diperoleh dari *capture*-an CCTV secara real time dari ke tiga lajur persimpangan jalan bersinyal di Jl. Borobudur Kecamatan Blimbing Kota Malang. Proses pengujian hampir sama dengan proses pembelajaran yaitu melalui *pra*-proses data, yaitu tahap pemisahan obyek dari background, seleksi obyek sesuai dengan wilayah yang sudah ditentukan, kemudian dilanjutkan proses deteksi tepi. Data hasil pendeteksian tepi digunakan sebagai data input proses pengujian JFBR yang digabungkan dengan *cluster K-Means*. Hasil dari pengujian adalah menentukan jumlah

kendaraan besar, kendaraan ringan, kendaraan berat, kendaraan roda dua, dan kendaraan tanpa mesin.

Tabel 2 adalah salah satu contoh urutan proses pengujian data, dimulai dari pemisahan obyek sampai dengan penentuan jenis kendaraan beserta jumlah kendaraan;

Tabel 2. Contoh Hasil Urutan Proses Pengujian Data

No	Urutan Proses
1	Pemisahan obyek dari <i>background</i>



Hasil Capture Bakground Jl. Borobudur



Wilayah pengambilan obyek

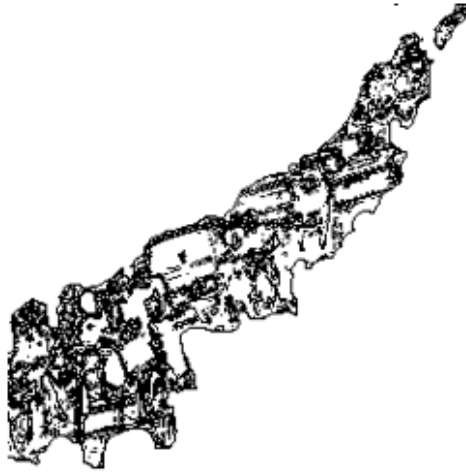


Hasil capture obyek dengan bakground Jl. Borobudur

No Urutan Proses



Segementasi obyek arus kendaraan



Hasil deteksi tepi

2 Nilai numerik informasi piksel yang digunakan sebagai data input JST

No.	Piksel R	Piksel G	Piksel B	Posisi Hor. (X)	Posisi Vert. (Y)
71	255	255	255	243	8
72	131	123	120	244	8
73	104	97	96	245	8
74	50	50	52	247	8
75	46	47	51	248	8
76	67	70	74	249	8
77	96	105	104	250	8
78	152	162	158	251	8
79	197	204	201	252	8
80	214	223	213	255	8
81	255	255	255	242	9
82	142	131	126	243	9
83	113	103	98	244	9


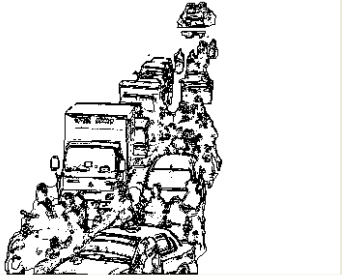
3 Hasil aktivasi lapisan hidden oleh fungsi *Gauss*

No	Urutan Proses	No.	Data Pksel	Data Posisi	Nilai Gauss
		1	0.3333333	216	1.0000000000
		2	0.3328446	216	0.6065307000
		3	0.3333333	216	0.0000000000
		4	0.3473492	216	0.0000000000
		5	0.3407290	216	0.0000000000
		6	0.3398230	216	0.0000000000
		7	0.3333333	216	0.0000000000
		8	0.3290735	216	0.0000000000
		9	0.3451777	216	0.0000000000
		10	0.3680782	216	0.0000000000
		11	0.3617021	216	0.0000000000
		12	0.3536755	216	0.0000000000
		13	0.3328446	217	1.0000000000

4 Hasil Data output (Fout) dan Bobot (W)

No.	Finput kromatik	Finput Pos. X	F Output	Nilai Bobot(W)
105	0.37611	259	1	13.95020485
106	0.37611	260	1	23.00000000
107	0.33333	250	2	210.00000000
108	0.35023	250	2	-4.33072901
109	0.35357	250	2	-1.44416952
110	0.34466	250	2	-0.05769957
111	0.33333	250	2	-0.00033540
112	0.35589	250	2	-0.00000091
113	0.48515	250	2	0.00000000
114	0.44112	250	2	0.00000000
115	0.35023	252	2	-32.00000000
116	0.35357	252	2	-78.84898376
117	0.34466	252	2	-23.27766800

5 Hasil Pengujian jenis dan jumlah kendaraan

Jumlah Kendaraan Besar	<input type="text" value="1"/>
Jumlah Kendaraan Sedang	<input type="text" value="11"/>
Jumlah Sepeda Motor	<input type="text" value="31"/>
Jumlah Kendaraan Tradisional	<input type="text" value="1"/>

Contoh hasil proses perhitungan jumlah kendaraan secara menyeluruh yang ditunjukkan oleh Tabel 2, selanjutnya disajikan hasil total pemrosesan pada Tabel 3.

Tabel 3. Output hasil JFBR dan clustering K-Means dalam memprediksi jenis dan jumlah kendaraan

No	P ke	T (det)	Phase I - Barat (Jumlah Kendaraan)				Phase II - US (Jumlah Kendaraan)								Phase III - T (Jumlah Kendaraan)			
			Barat				Utara				Selatan				Timur			
			KR	KB	KRD	KTM	KR	KB	KRD	KTM	KR	KB	KRD	KTM	KR	KB	KRD	KTM
1	0	19	3	1	14	0	0	0	0	0	0	0	0	0	0	0	0	0
2	1	44					3	1	21	1	4	0	13	0				
3		60													6	1	21	1
4		54	5	1	31	1												
5	2	52					5	2	21	0	5	1	17	1				
6		60													7	1	21	0
7		54	6	0	35	0												
8	3	52					6	0	23	0	4	1	35	0				
9		60													5	0	27	1
10		54	4	2	41	1												
11	4	52					5	1	20	2	6	1	24	1				
12		60													4	2	34	2
13		54	7	0	37	0												
14	5	52					5	2	29	0	6	0	14	0				
15		60																
16		54	6	1	34	2												
17	16	52					3	1	16	0	3	0	17	0				
18		60													5	2	32	0
19		54	4	2	21	2												
20	17	52					4	2	19	1	6	1	9	1				
21		60													3	0	17	2
22		54	6	1	15	2												
23	18	52					3	1	17	0	2	1	8	1				
24		60													3	1	33	0
25		54	2	2	38	0												
26	19	52					2	1	19	1	4	1	21	1				
27		60													2	0	24	0
28		54	6	1	22	1												
29	20	52					3	0	14	1	2	0	11	0				
30		60													6	0	21	1
31		54	5	2	40	1												
32		60																
33		54	6	1	34	2												
34	21	52					3	1	16	0	3	0	17	0				
35		60													5	2	32	0
36		54	4	2	21	2												
37	17	52					4	2	19	1	6	1	9	1				
38		60													3	0	17	2
39		54	6	1	15	2												
40	18	52					3	1	17	0	2	1	8	1				
41		60													3	1	33	0
42		54	2	2	38	0												
43	19	52					2	1	19	1	4	1	21	1				
44		60													2	0	24	0
45		54	6	1	22	1												
46	20	52					3	0	14	1	2	0	11	0				
47		60													6	0	21	1
48		54	5	2	40	1												
49		60																
50		54	6	1	34	2												
51	21	52					3	1	16	0	3	0	17	0				
52		60													5	2	32	0
53		54	4	2	21	2												
54	17	52					4	2	19	1	6	1	9	1				
55		60													3	0	17	2
56		54	6	1	15	2												
57	18	52					3	1	17	0	2	1	8	1				
58		60													3	1	33	0
59		54	2	2	38	0												
60	19	52					2	1	19	1	4	1	21	1				
61		60													2	0	24	0
62		54	6	1	22	1												
63	20	52					3	0	14	1	2	0	11	0				
64		60													6	0	21	1
65		54	5	2	40	1												
66		60																
67		54	6	1	34	2												
68	21	52					3	1	16	0	3	0	17	0				
69		60													5	2	32	0
70		54	4	2	21	2												
71	17	52					4	2	19	1	6	1	9	1				
72		60													3	0	17	2
73		54	6	1	15	2												
74	18	52					3	1	17	0	2	1	8	1				
75		60													3	1	33	0
76		54	2	2	38	0												
77	19	52					2	1	19	1	4	1	21	1				
78		60													2	0	24	0
79		54	6	1	22	1												
80	20	52					3	0	14	1	2	0	11	0				
81		60													6	0	21	1
82		54	5	2	40	1												
83		60																
84		54	6	1	34	2												
85	21	52					3	1	16	0	3	0	17	0				
86		60													5	2	32	0
87		54	4	2	21	2												
88	17	52					4	2	19	1	6	1	9	1				
89		60													3	0	17	2
90		54	6	1	15	2												
91	18	52					3	1	17	0	2	1	8	1				
92		60													3	1	33	0
93		54	2	2	38	0												
94	19	52					2	1	19	1	4	1	21	1				
95		60													2	0	24	0
96		54	6	1	22	1												
97	20	52					3	0	14	1	2	0	11	0				
98		60													6	0	21	1
99		54	5	2	40	1												
100		60																

dengan :

- KR : Kendaraan Ringan
- KB : Kendaraan Berat
- KRD : Kendaraan Roda Dua
- KTM : Kendaraan Tanpa Mesin
- P ke : Periode atau siklus ke
- T : Lama waktu tunggu (antrian) saat capture kondisi arus lalulintas

Untuk data no 1 s.d. 31. adalah hasil prediksi jumlah kendaraan pada saat arus kendaraan padat, sedangkan data no. 32 s.d. 61 adalah hasil prediksi jumlah kendaraan pada saat kondisi arus kendaraan sedang. Data ini diperoleh dari hasil simulasi software, dengan input adalah hasil capture arus kendaraan yang diperoleh secara real time dengan lama

periode konstan : 95 det + 15 det = 110det, 95 det adalah lama waktu nyala hijau yaitu phase I : 27 detik, phase II : 35 detik dan phase III : 33 detik, dan nyala lampu kuning total pada masing-masing phase adalah 5 detik.

Selanjutnya data tersebut dikomparasikan dengan perhitungan secara manual untuk mengetahui tingkat akurasi dari perhitungan otomatis dari metode JFBR dengan *cluster K-Means*. Tingkat kesalahan masing-masing jenis kendaraan dihitung berdasarkan selisih antara perhitungan manual dengan perhitungan otomatis dibagi dengan perhitungan manual, atau dapat ditulis

$$\frac{Abs(\sum kendaraan (hitungan manual) - \sum kendaraan (hitungan otomatis))}{\sum kendaraan (hitungan manual)}$$

dan kesalahan rata-rata dihitung dari total kesalahan dibagi dengan jumlah uji data, hasilnya ditunjukkan oleh tabel 4 berikut;

Tabel 4. Tingkat error hasil perhitungan jumlah kendaraan

No	Jenis Kendaraan	Perhitungan Manual	Perhitungan Otomatis	error
1	Kendaraan Besar	2	2	0
2	Kendaraan Sedang	2	2	0
3	Sepeda Motor	44	43	0,0227273
4	Kendaraan tradisional	2	2	0
5	Kendaraan Besar	1	1	0
6	Kendaraan Sedang	4	4	0
7	Sepeda Motor	31	31	0
8	Kendaraan tradisional	0	0	0
9	Kendaraan Besar	1	1	0
10	Kendaraan Sedang	7	7	0
11	Sepeda Motor	36	36	0
12	Kendaraan tradisional	1	1	0
13	Kendaraan Besar	1	1	0
14	Kendaraan Sedang	11	11	0
15	Sepeda Motor	32	31	0,03125
16	Kendaraan tradisional	1	1	0
17	Kendaraan Besar	0	0	0
18	Kendaraan Sedang	4	4	0
19	Sepeda Motor	29	29	0
20	Kendaraan tradisional	1	1	0
21	Kendaraan Besar	1	1	0
22	Kendaraan Sedang	7	7	0
23	Sepeda Motor	32	32	0
24	Kendaraan tradisional	1	1	0
25	Kendaraan Besar	0	0	0
26	Kendaraan Sedang	4	4	0
27	Sepeda Motor	14	14	0
28	Kendaraan tradisional	0	0	0
29	Kendaraan Besar	0	0	0
30	Kendaraan Sedang	6	6	0
31	Sepeda Motor	11	11	0
32	Kendaraan tradisional	1	1	0

Total error = 0,053977

4. Simpulan

Hasil implementasi metode gabungan Jaringan Fungsi Basis Radial (JFBR) dan *Cluster K-Means* ke dalam desain software untuk mengidentifikasi dan menentukan jenis kendaraan dengan menggunakan sensor CCTV (IP), menunjukkan bahwa sistem mampu mengenali jenis kendaraan dan memprediksi jumlah kendaraan sesuai dengan jenisnya. Dari hasil pengujian diperoleh *error* rata-rata sebesar 0,053977 untuk 32 jenis kendaraan, hasil ini ditunjukkan oleh tabel (3) dan tabel (4).

Daftar Rujukan

- <https://dataindonesia.id/sektor-riil/detail/jumlah-kendaraan-di-indonesia-capai-14199-juta-unit-pada-2021>, diakses 29 Januari 2023
- Marta L. S., Lina E. A., Paolo A. C., Luis C. M., Bairon V., Guillermo G., Ramiro A. A., Mauricio C., Fernando G. M., Alejandro V., Cesar G., Martin J., & Sara H. (2018), Dynamic allocation of traffic light plans as a traffic reduction strategy, IET, <https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Dynamic%20allocation%20of%20traffic%20light%20plans%20as%20a%20traffic%20reduction%20strategy>
- Billie P., Josua C., Muhammad T. H., & Abdul M., (2018), Adaptive Traffic Lights through Traffic Density Calculation on Road Pattern, ICAST, <https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Adaptive%20Traffic%20Lights%20through%20Traffic%20Density%20Calculation%20on%20Road%20Pattern>
- Anam F., Indu, Vandana N., (2020), Smart Density Based Traffic Light System, ICRITO, <https://ieeexplore.ieee.org/search/searchresult.jsp?newsearch=true&queryText=Smart%20Density%20Based%20Traffic%20Light%20System>
- Anuj S., Sachin S., Kamlesh C. P., K. C. Nithin K. (2021), Artificial Intelligence based Framework for Effective Performance of Traffic Light Control System, ICSES, <https://ieeexplore.ieee.org/document/9633913>
- Y.H.P. Isnomo, (2014), Predict of Lost Time at Traffic Lights Intersection Road Using Image Processing, IC-ITECHS, <http://jurnal.stiki.ac.id/IC-ITECHS/article/view/353>
- Rocky H., (2020), Penerapan Metode Laplacian of Gaussian Dalam Mendeteksi Tepi Citra Pada Penyakit Meningitis, KLIK, <https://djournals.com/klik/article/download/21/5/65>