

INTELLIGENT TRAFFIC LIGHT MENGUNAKAN FUZZY LOGIC ALGORITHM

Aulio Gigih Saputra¹, Ida Nurcahyani², Elvira Sukma Wahyuni³

Jurusan Teknik Elektro, Universitas Islam Indonesia
Jl Kaliurang KM 14.5 Yogyakarta, Indonesia

¹12524013@students.uui.ac.id

²ida.nurcahyani@uui.ac.id

³elvira.wahyuni@uui.ac.id

Abstrak— Pengaturan *traffic light* yang ada di Indonesia saat ini masih menggunakan sistem pengaturan yang statis, yaitu menggunakan sistem *fixed-time* signals tanpa memperhatikan banyaknya atau sedikitnya kendaraan pada arus lalu lintas. Kemacetan menjadi permasalahan di berbagai wilayah, terutama di kota-kota besar. Untuk itu dibutuhkan pengaturan *traffic light* yang dapat mengefisiensikannya, salah satunya dengan penerapan algoritme *fuzzy* dan *image processing* pada sistem transportasi yang diharapkan dapat mengefisiensikan alokasi waktu nyala lampu hijau pada *traffic light*. Dengan menggunakan metode *image matching* pada sistem *image processing* yang berfungsi untuk melakukan perhitungan kecocokan piksel citra referensi dan citra diambil dari kamera yang menghasilkan output berupa tingkat persentase *match*, variabel output dari *image matching* menjadi inputan bagi algoritme *fuzzy logic* untuk menentukan alokasi waktu nyala lampu hijau yang sesuai dengan padatnya kendaraan di jalan. Hasil dari penelitian ini menunjukkan bahwa dengan menambahkan algoritme *fuzzy logic* pada sistem *image processing* di pengaturan *traffic light* menunjukkan linearitas antara input yang diterapkan ke sistem dan output diambil darinya dengan keakuratan lebih dari 85% dalam menetapkan alokasi *green light time* untuk arus lalu lintas tetap lancar.

Kata kunci— algoritme *fuzzy*, *fuzzy logic*, *fuzzy rule*, *image matching*, *image processing*, dan *traffic light*

I. PENDAHULUAN

Kemacetan lalu lintas merupakan salah satu masalah yang sering terjadi di kota. Di Indonesia, banyak sekali masalah yang timbul di karenakan kemacetan lalu lintas seperti yang terjadi di kota-kota metropolitan Indonesia. Faktor penyebab dari kemacetan lalu lintas adalah kepadatan penduduk, jumlah kendaraan yang terlalu banyak, serata pengaturan *traffic light* yang tidak sesuai dengan kondisi pada persimpangan saat itu. Jika lampu lalu lintas dapat diatur warna lampu yang menyala dan lama waktu menyala sesuai dengan keadaan persimpangan saat itu, misalkan dengan mempertimbangkan jalur yang memiliki antrian kendaraan terpanjang, atau jalur yang memiliki kendaraan dengan waktu tunggu terlama, atau dengan mempertimbangkan kondisi di persimpangan yang bersebelahan, tentunya hal ini akan menurunkan tingkat kemacetan yang berdampak kepada efektifitas aktifitas masyarakat di kota tersebut.

Untuk itu, dibutuhkan suatu sistem yang dapat mengatur lampu lalu lintas dengan baik seperti mempertimbangkan kepadatan di setiap jalur dan mengatur durasi lampu sesuai dengan kepadatannya, agar dapat mengurangi tingkat kemacetan yang terjadi di satu atau lebih persimpangan. Dengan keadaan lalu lintas yang sekarang, di mana jumlah kendaraan dan kepadatan lalu lintas yang terus bertambah, hubungan antara sebuah persimpangan dengan persimpangan tetangganya semakin kuat. Keadaan yang terjadi di sebuah persimpangan dan keputusan yang diambil terhadap, lama waktu lampu menyala tentunya akan berpengaruh terhadap keadaan di persimpangan tetangganya. Sudah banyak peneliti yang mencoba untuk menyelesaikan masalah ini dengan menggunakan metode-metode seperti *fuzzy logic* [1], menggunakan sistem kontrol cerdas *Fuzzy Intelligent Traffic Signal* (FITS) [2] dan algoritme *fuzzy logic Shadowed Type-2* (ST2) [3], juga mengkombinasikan regresi logistik dengan *fuzzy logic* [4].

Pada dasarnya masalah utama dalam *traffic light control* adalah keharusan membuat keputusan secara sekuensial. Salah satu metode yang cocok dengan karakteristik permasalahan tersebut adalah *fuzzy logic*. Berdasarkan masalah tersebut dan hasil penelitian sebelumnya, penulis telah mengembangkan sebuah sistem yang mengatur mekanisme terkait *Intelligent Traffic Light Control* (ITSC) dengan algoritme *fuzzy logic* seperti dengan menggabungkan metode *image matching* dan *fuzzy logic*. Diharapkan sistem ini dapat membantu mengatur *traffic light* agar lebih efektif dalam menentukan alokasi waktu lampu hijau yang diperlukan sesuai dengan nilai persentase kecocokan (*match*) citra dari kamera di persimpangan dengan gambar yang menjadi referensi, sehingga tidak membuang-buang waktu secara sia-sia untuk pengendara.

II. TINJAUAN PUSTAKA

A. *Traffic light*

Traffic light adalah sebuah alat yang berfungsi untuk mengatur arus lalu lintas yang ada di jalanan baik itu pejalan kaki, mupun pengendara, dll. Penggunaan *Traffic light* awalnya diperkenalkan di Inggris tepatnya pada Desember tahun 1868 dan digunakan untuk mengatur lalu lintas untuk para pejalan kaki, atau pengendara sepeda, yang mana pada saat itu, sistemnya masih manual dengan polisi yang harus berjaga sepanjang hari untuk mengoprasikannya karena masih

menggunakan lentera, dengan warna merah dan hijau. Dimana merah artinya berhenti dan hijau hati hati. Pada masa jaman sekarang, sistem traffic light diluar negeri sudah banyak yang menggunakan *Automatic Control Light Sistem* (ACTS). Sistem ACTS ini secara otomatis dapat mengatur lampu lalu lintas dengan bantuan berupa kamera dan sensor yang berbasis mikrokontroler. Kamera ini biasanya terhubung dengan sistem yang ada pada sistem *traffic light* dimana berfungsi untuk mengamati banyaknya kendaraan di persimpangan jalan raya, yang kemudian hasil pengamatan kamera tersebut diolah oleh komputer atau monitor kemudian hal tersebut di rekam dan di kirimkan oleh mikrokontroler menggunakan *transfer flas* relay. Kemudian mikrokontroler bekerja menyalakan *traffic light* secara otomatis searah jarum jam, sehingga ketika komputer sudah terhubung dengan mikrokontroler, maka mikrokontroler secara otomatis mengirimkan informasi mengenai lampu mana yang sedang hijau/merah/kuning [7].

B. Sobel Edge Detection

Operator Sobel digunakan untuk pengukuran gradien spasial 2-D pada citra. Dengan menggunakan gradien, perkiraan nilai absolut besarnya di setiap titik dapat ditemukan [9]. Biasanya itu digunakan untuk menemukan perkiraan besarnya gradien absolut di masing-masing titik dalam citra dengan input skala abu-abu. Dibandingkan dengan operator tepi lainnya, Sobel memiliki dua keunggulan utama [10] yaitu sejak diperkenalkannya faktor rata-rata (*average*), sobel memiliki beberapa efek smoothing terhadap noise acak dari citra. Kedua karena sobel adalah diferensial dari dua baris atau dua kolom, sehingga unsur-unsur tepi di kedua sisi yang dimiliki telah ditingkatkan, sehingga ujungnya tampak tebal dan terang. Standar untuk operator sobel adalah kernel 3×3 , masing-masing estimasi gradien tengah adalah jumlah vektor dari sepasang vektor ortogonal. Setiap vektor ortogonal adalah vektor estimasi turunan dikalikan dengan vektor satuan yang menentukan derivatif. Jumlah vektor dari gradien sederhana ini memperkirakan jumlah ke jumlah vektor dari 8 vektor turunan [11].

C. Image Matching

Teknik pengenalan berdasarkan perbandingan mewakili setiap kelas dengan vektor pola prototipe. Pola yang tidak diketahui ditetapkan ke kelas yang paling dekat dalam hal metrik yang telah ditentukan. Pendekatan paling sederhana adalah pengklasifikasi jarak minimum, yang seperti namanya yaitu menghitung jarak (Euclidean) antara yang tidak diketahui dan masing-masing vektor prototipe dan memilih jarak terkecil untuk membuat keputusan. Ada pendekatan lain berdasarkan korelasi, yang bisa dirumuskan secara langsung dalam hal citra dan cukup intuitif. Peneliti telah menggunakan pendekatan yang sama sekali berbeda untuk perbandingan citra. Membandingkan citra referensi dengan piksel citra yang diambil dari kamera. Meskipun ada beberapa kelemahan terkait membandingkan berbasis piksel tetapi ini adalah salah satu teknik terbaik untuk digunakan dalam penelitian ini. Citra real disimpan dalam matrik dalam memori dan citra waktu real juga dikonversi dalam matrik yang diinginkan., agar citra dan nilai pikselnya dalam matriks harus sama. Ini adalah fakta paling sederhana yang digunakan dalam perbandingan piksel. Jika ada ketidaksesuaian dalam nilai piksel itu menambahkan ke penghitung yang digunakan untuk

menghitung jumlah ketidakcocokan piksel [8]. Akhirnya persentase perbandingan dinyatakan sebagai persamaan (5)

$$\%match = \frac{\text{No. of pixels matched successfully}}{\text{total no. of pixels}} \quad (1)$$

D. Fuzzy Rule

Setelah mengatur parameter input dan output, langkah selanjutnya adalah mencocokkannya dengan aturan IF-THEN dan agregasi level, sehingga kesimpulan dapat dibuat. Metode inferensi fuzzy mamdani digunakan di sini untuk diimplementasikan dalam MATLAB. Semua aturan yang digunakan untuk mengendalikan proses pemeriksaan disajikan dalam bentuk matriks. Konfigurasi matriks ini memungkinkan untuk melihat sejumlah besar aturan dan hasilnya secara sekilas. Semua aturan dievaluasi secara paralel, dan urutan aturan tidak penting dalam prosesnya. Di bawah ini adalah bagaimana aturan dievaluasi:

IF persentase *match* antara citra yang diambil kamera dan citra referensi hasil nilainya besar keadaan jalan ramai, maka itu berarti alokasi waktu lampu hijau tinggi. Ini dapat diwakili dengan mengikuti aturan.

$$IF(Ppr = B) \text{ and } (Sm = H) THEN(Tgl = H) \quad (2)$$

Ketika persentase *match* antara citra yang diambil kamera dan citra referensi hasil nilainya kecil dan keadaan jalan sepi kendaraan, maka alokasi waktu lampu hijau yang kecil harus diterapkan untuk mengefisienkan waktu pengendara. Ini dapat diwakili oleh aturan berikut.

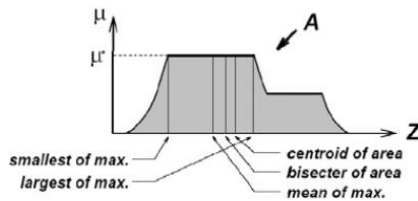
$$IF(Ppr = S) \text{ and } (Sm = L) THEN(Tgl = L) \quad (3)$$

Setelah persentase *match* antara citra yang diambil kamera dan citra referensi dan keadaan jalan memiliki ukuran yang sama, maka alokasi waktu lampu hijau normal saja. Ini dapat diwakili oleh aturan berikut.

$$IF(Ppr = N) \text{ and } (Sm = N) THEN(Tgl = N) \quad (4)$$

E. Defuzzifikasi

Defuzzifikasi menggunakan input yang diperoleh dari aturan-aturan *fuzzy (rule)*, sedangkan output yang dihasilkan merupakan suatu bilangan pada domain himpunan *fuzzy* tersebut, sehingga jika diberikan suatu himpunan *fuzzy* dalam range tertentu, maka harus dapat diambil suatu nilai crisp tertentu sebagai keluarannya [15]. Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, diantaranya yaitu metode *Center of Area* (COA), *Bisector*, *Mean of Maximum* (MOM), *Largest of Maximum* (LOM), dan *Smallest of Maximum* (SOM) untuk mengubah nilai *fuzzy* menjadi nilai bilangan [16] seperti pada gambar 2.3.



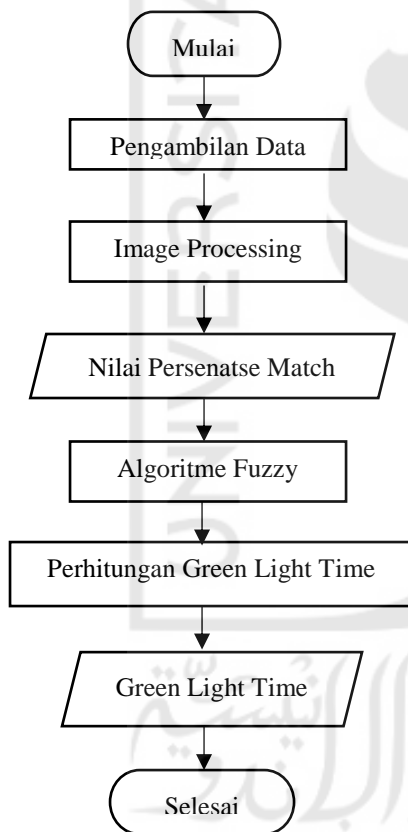
Gambar 1 Metode Bisector

Dalam penelitian ini peneliti memilih metode bisector untuk digunakan, dimana hasil nilai crisp diperoleh dengan cara mengambil nilai pada domain *fuzzy* yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy* dengan menggunakan persamaan (11).

$$Tgl = \int_{\alpha}^{TglBOA} \mu(Tgl)dTgl = \int_{TglBOA}^{\beta} \mu(Tgl)dgl \quad (5)$$

III. METODE PENELITIAN

Pada diagram alir di bawah, dijelaskan proses yang dimulai dari memasukkan citra referensi dan citra *real-time* sampai pada hasil perhitungan alokasi *green light time* oleh algoritme *fuzzy*.



Gambar 2 Diagram Alir Penelitian

A. Pengambilan Data

Proses dimulai dengan melakukan pengambilan data berupa citra dengan kamera untuk masing-masing kondisi tiap jalur pada persimpangan Kentungan, Sleman. Data citra yang diperoleh berupa kondisi jalan pada saat sepi, normal, dan padat.

B. Image Processing

Setelah proses pengambilan citra dengan kamera, data yang diperoleh kemudian melewati beberapa proses yaitu, ubah ukuran citra, tingkatkan kualitas citra dengan *image enhancement*, melakukan deteksi tepi, dan yang terakhir citra dibandingkan antara citra referensi dengan citra *real-time* dengan metode *image matching*, dimana metode ini menghasilkan nilai persentase *match*.

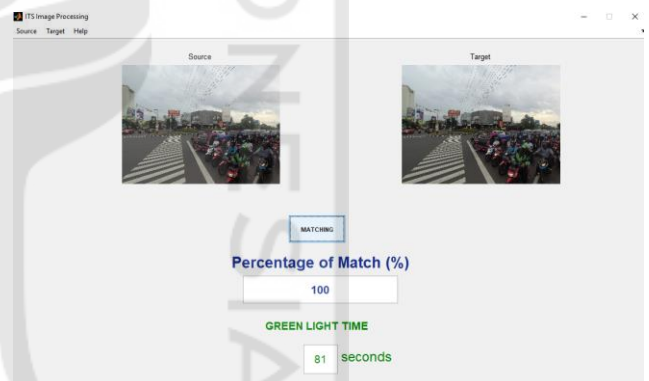
C. Algoritme Fuzzy

Setelah proses pengolahan citra dengan metode *image matching*, hasil dari proses yaitu nilai persentase *match* menjadi inputan algoritme *fuzzy* dan diolah menjadi variable-variabel *fuzzy*. Kemudian proses selanjutnya membuat aturan *fuzzy* (*fuzzy rule*) dengan aturan IF-THEN yang dihubungkan dengan logika operasi AND-OR, maka algoritme *fuzzy* dapat menentukan keadaan jalan *real-time* dengan rules menggunakan parameter yang telah ditentukan.

D. Perhitungan Green Light Time

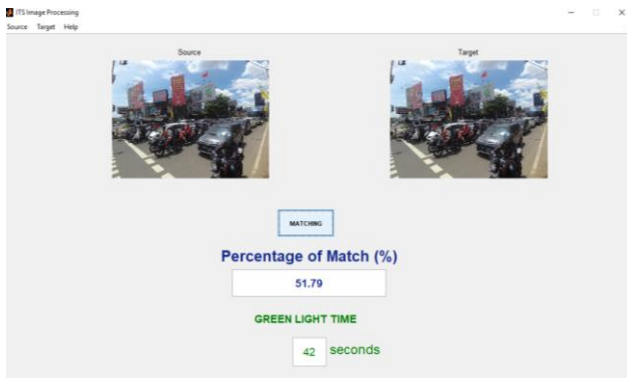
Pada proses ini algoritme *fuzzy* memulai perhitungan untuk alokasi *green light time* dengan rumus yang dibuat, dimana hasil dari defuzzikasi yaitu nilai himpunan tegas (*crisp*) yang diperoleh dengan cara mengambil nilai pada domain *fuzzy* [15 50 85] yang memiliki nilai keanggotaan setengah dari jumlah total nilai keanggotaan pada daerah *fuzzy* dijadikan salah satu parameter perhitungan.

IV. HASIL DAN ANALISIS



Gambar 3 Hasil Image Matching Padat

Pada percobaan gambar 3 keadaan pada citra yang diambil adalah pasangan yang cocok dengan citra referensi dimana berarti memiliki nilai piksel yang sama dan tidak ada error pada *match* pikselnya dengan demikian pada hasil *percentage of match*-nya menunjukkan nilai persentase sebesar 100% sehingga hasil percobaan ini termasuk dalam kategori keadaan jalan padat, dan untuk alokasi *green light time* yang ada pada kotak *green light time* hasil yang didapat dihitung algoritme *fuzzy* dengan menggunakan parameter-parameter yang telah ditentukan oleh *fuzzy rule* sehingga menampilkan hasil 81 detik seperti yang ditampilkan pada gambar 4.1.



Gambar 4 Hasil Image Matching Normal

Pada gambar 4 percobaan Hasil *Image Matching* keadaan normal yang dilakukan pada program ini citra kedua-duanya adalah berbeda sehingga memiliki nilai piksel yang berbeda dan ada nilai *error* yang dihasilkan dari *image matching*, dan karena nilai piksel citra yang diambil berbeda dengan citra referensi dengan demikian pada hasil persentase *of match*-nya menunjukkan nilai persentase 51,79% sehingga hasil percobaan ini termasuk dalam kategori keadaan jalan normal, dan pada kotak *green light time* yang menampilkan alokasi untuk waktu lampu hijau menyala dihitung menggunakan algoritme *fuzzy* dengan parameter-parameter yang telah di tentukan oleh *fuzzy rule* sehingga menampilkan hasil 42 detik seperti yang ditampilkan pada gambar 4.2.



Gambar 5 Hasil Image Matching Sepi

Pada gambar 5 percobaan Hasil *Image Matching* keadaan sepi yang dilakukan pada program ini kedua citranya adalah berbeda sehingga memiliki nilai piksel yang berbeda pula dan ada nilai *error* yang dihasilkan dari *image matching*-nya, dan karena nilai piksel citra yang diambil berbeda dengan citra referensi dengan demikian pada hasil persentase *of match*-nya mendapatkan nilai persentase 10,69% sehingga hasil percobaan ini termasuk dalam kategori keadaan jalan sepi, dan pada kotak *green light time* yang menampilkan alokasi waktu untuk lampu hijau menyala menampilkan hasil 11 detik seperti yang ditampilkan oleh gambar 4.2.

V. KESIMPULAN

Dari percobaan yang dilakukan, ketika inputan persentase *match* tidak sesuai dengan tinggi dari kepadatan lalu lintas, alokasi *green light time* cenderung menurun, sehingga memungkinkan penumpukkan jumlah kendaraan yang lebih tinggi untuk melewati persimpangan. Jadi dari hasil simulasi ini menunjukkan menunjukkan program berjalan dengan baik untuk menentukan alokasi green light pada berbagai macam keadaan di persimpangan jalan. Hasil ini memuaskan untuk mencapai tujuan yang ditentukan dalam penelitian ini.

DAFTAR PUSTAKA

- [1] M.A. Azim and M.N. Huda, "Fuzzy Traffic Control," Term Paper based on Case Study and Implementation of a Fuzzy Application", School of Computing Queen's University, no. 10036952, 2010.
- [2] J. Jin, X. Ma, and I. Kosonen, "An Intelligent Control System for Traffic Lights with Simulation-based Evaluation," *Control Engineering Practice*, vol. 58, no. May 2016, pp. 24–33, 2017.
- [3] K. Chatterjee, A. De, and F. T. S. Chan, "Real Time Traffic Delay Optimization using Shadowed Type-2 Fuzzy Rule Base," *Applied Soft Computing Journal*, vol. 74, pp. 226–241, 2019.
- [4] Anurag Singh, M. Singh, G. Sharma, and K. V. Arya, "Traffic Management using Logistic Regression with Fuzzy Logic," *Procedia Computer Science*, vol. 132, pp. 451–460, 2018.
- [5] S. D. Maniswari, A. Rusdinar, and B. Purnama, "Smart Traffic Light Menggunakan Image Processing dan Metode Fuzzy Logic Smart Traffic Light Using Image Processing and Fuzzy Logic," *e-Proceeding Eng.*, vol. 2, no. 2, pp. 2166–2170, 2015.
- [6] K. Nithiyanthan, "MATLAB Simulations on Object Counting and Density Calculations for an Image," *International Journal of Pure and Applied Mathematics*, Volume 118, No. 20, 1283-1290, 2018.
- [7] P. Khazron and L. Outerbridge, "Traffic Lights," 2009..
- [8] L. Antonio, "Traffic Control using Image Processing," National Institute of Technology Srinagar, pp. 1–44, 2014.
- [9] K. Parvathi and S. K. Mandal, "Development of Simple Edge Detection Technique," no. April, 2018.
- [10] R. Chandwadkar, S. Dhole, V. Gadewar, D. Raut, and S. Tiwaskar, "Comparison Of Edge Detection Techniques," *6th Annu. Conf. IRAJ*, no. 6, pp. 133–136, 2013.
- [11] S. Gupta and S. Ghosh Mazumdar, "Sobel edge detection algorithm," *Int. J. Comput. Sci. Manag. Res.*, vol. 2, no. 2, pp. 1578–1583, 2013.
- [12] A. Ullah, O. B. Kharisma, and I. Santoso, "Fuzzy Logic Implementation to Control Temperature and Humidity in a Bread Proofing Machine," vol. 1, no. 2, pp. 66–74, 2018.
- [13] L.A. Zadeh "Fuzzy Set Theory," *Information and Control* 8, pp. 338-353, 1965.
- [14] M. Maslim, "Implementasi Metode Logika Fuzzy

- dalam Pembangunan Sistem Optimalisasi Lampu Lalu Lintas,” *J. Buana Inform.*, vol. 9, no. 1, pp. 11–20, 2018.
- [15] S. Muzid and S. Kusumadewi, “Membangun Toolbox Algoritma Evolusi Fuzzy untuk Matlab,” *Semin. Nas. Apl. Teknol. Inf.*, vol. 2007, no. Snati, 2017.
- [16] E. Haerani, “Analisa Kendali Logika Fuzzy Dengan Metode Defuzzifikasi Center of Area (COA), Bisektor , Mean of Maximum (MOM), Largest of Maximum (LOM), DAN Smallest of Maximum (SOM),” *J. Sains dan Teknol. Ind.*, 2015.
- [17] Sutikno and I. Waspada, “Perbandingan Metode Defuzzifikasi Sistem Kendali Logika Fuzzy Model Mamdani Pada Motor DC,” vol. 2, no.3, pp. 27–38, 2011.
- [18] Candra Noor Santi, “Mengubah Citra Berwarna Menjadi Gray-Scale dan Citra biner,” *Teknologi Informasi DINAMIK*, vol. 16, no. 1, pp. 14–19, 2011.

