

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Studi literatur ini berisi survei yang dilakukan terhadap penelitian lain sebelumnya untuk mengatasi kemacetan yang terjadi pada perempatan jalan menggunakan metode tertentu pada beberapa jurnal. Salah satunya penelitian yang dilakukan oleh M.A. Azim dan M.N Huda [1] membuat sistem kontrol lalu lintas dengan menggunakan algoritme *fuzzy*, yang didasarkan pada teori logika *fuzzy*. Durasi waktu lampu hijau pada penelitian ini dihitung dengan menggunakan informasi tentang jumlah kendaraan yang menunggu pada fase lampu merah menyala dan jumlah frekuensi kendaraan selama fase lampu hijau menyala. Sistem tersebut menggunakan prosedur inferensi *Min-Max* dan strategi Defuzzifikasi Centroid untuk mendapatkan nilai yang paling akurat, untuk durasi waktu lampu hijau sebagai outputannya. Hasil dari penelitian, sistem *fuzzy* dapat secara dinamis mengubah durasi waktu menyala lampu hijau berdasarkan sembilan aturan yang dibuat untuk mempertahankan aliran arus lalu lintas yang sangat baik di persimpangan multi-jalur.

Penelitian memperkenalkan sistem kontrol cerdas untuk aplikasi sinyal lampu lalu lintas, yang disebut kontrol *Fuzzy Intelligent Traffic Signal (FITS)* [2]. Sistem kontrol pada penelitian ini dapat diterapkan dengan mudah dan ekonomis untuk meningkatkan infrastruktur lampu lalu lintas. Sistem kontrol ini diprogram pada peranti perangkat keras menengah yang mampu menerima pesan dari peranti pengontrol sinyal serta mengutamakan indikasi lampu lalu lintas selama operasi real-time. *Toolbox* pengendali dan pengoptimal sinyal diintegrasikan ke dalam perangkat lunak yang tertanam di perangkat perangkat keras FITS. Sistem kontrol yang berbasis logika *fuzzy* telah diterapkan di FITS untuk mengevaluasi kinerja dari sistem FITS, peneliti mengembangkan kerangka kerja komputasi untuk mengevaluasi sistem FITS dengan menggunakan simulasi lalu lintas mikroskopis. Peneliti membandingkan berbagai strategi kontrol sinyal yang biasa digunakan dengan pendekatan sistem kontrol FITS. Hasil simulasi menunjukkan bahwa sistem kontrol memiliki potensi untuk meningkatkan mobilitas arus lalu lintas, dibandingkan dengan semua metode kontrol sinyal yang sudah diuji, yaitu karena kemampuannya dalam menghasilkan durasi waktu dengan struktur fase yang fleksibel dan membuat keputusan yang cerdas. Selain itu, hasil percobaan ini juga menunjukkan bahwa FITS menunjukkan kinerja yang unggul daripada beberapa sistem pengontrol lain ketika beberapa detektor rusak karena FITS memiliki fitur diagnosa sendiri dalam sistemnya.

Penelitian [3] menggunakan algoritme *fuzzy logic Shadowed Type-2* (ST2) karena berfungsi dengan baik dalam domain yang memiliki data yang sangat canggung dan tidak pasti. ST2 pada penelitian ini dapat meningkatkan variabel dari bentuk *fuzzy* dengan membaginya ke dalam wilayah yang berbeda. Dengan demikian, simulator waktu tunggu *fuzzy* ST2 dibuat yang berdasarkan pada basis aturan *fuzzy* ST2, yang outputnya dapat diimplementasikan dalam model *Time Optimized Shortest Path* (TOSP) berbasis lalu lintas secara *real-time*. Ini dapat membantu dalam penataan jalur dengan waktu yang dioptimalkan dari satu lokasi ke lokasi lain. Ini dapat dilakukan dengan mengambil data lalu lintas yang *real-time* dari persimpangan setelahnya, lalu menghitung waktu tunggu menggunakan basis aturan *fuzzy* ST2, dan akhirnya mengarahkan kendaraan untuk mengambil jalur yang sudah dioptimalkan, sehingga menghasilkan pengurangan waktu tunggu keseluruhan dari setiap persimpangan. Oleh karena itu, pendekatan dengan menggunakan model TOSP terbukti sangat efektif dalam menyeimbangkan waktu tunggu kendaraan di persimpangan keseluruhan di wilayah metropolitan.

Peneliti A. Singh, M. Singh, G. Sharma, dan K. V. Arya [4] mengkombinasikan regresi logistik dengan logika *fuzzy* yang akan membuat sistem manajemen lalu lintas adaptif yang lebih baik dan juga untuk menyediakan kerangka kerja sistem transportasi cerdas yang akan dapat mengatur lalu lintas secara efisien sehingga keputusan tepat untuk memilih jalur mana yang cepat dapat diambil. Pada penelitian ini sistem gabungan digunakan untuk menghitung berbagai probabilitas pada setiap jalur yang mungkin dengan mempertimbangkan informasi lalu lintas secara *real-time*, jarak dan kondisi jalan kemudian parameter-parameter tersebut digunakan untuk mengambil keputusan dalam skenario yang tidak pasti. Metode yang dibuat mempertimbangkan sejumlah parameter seperti jarak, kondisi cuaca, lokasi jalan, hari dalam seminggu dan waktu di mana seseorang bepergian. Hasil penelitian tersebut menunjukkan secara umum, pengguna mengikuti jalur jarak terpendek tetapi terkadang menjadi jalur lalu lintas terburuk.

Penelitian [5] menggunakan sistem pengolahan citra (*image processing*) untuk mengolah data kepadatan kendaraan yang dideteksi oleh sensor kamera dan metode *fuzzy logic* untuk meminimalkan waktu tunggu. Dalam penelitian ini uji cobanya menggunakan webcam untuk menangkap citra pada purwarupa lalu lintas yang telah dibuat, sistem dapat mengeksekusi untuk digunakan dalam perhitungan *fuzzy logic* dan menyimpan hasilnya berupa foto dalam waktu 2,90 detik. Hasilnya metode *image processing* memiliki biaya yang relatif murah, setup program yang mudah, akurasi dan kecepatan dalam mengolah data dengan baik. Untuk metode *fuzzy logic* yang digunakan mampu melakukan perhitungan waktu menyala *traffic light* dengan perbandingan pengambilan data secara manual dan secara otomatis yang dijalankan oleh program dengan tingkat akurasi sebesar 100%.

K. Nithiyananthan [6] pada penelitian ini membuat sebuah sistem dengan metode yang sederhana untuk secara otomatis menentukan jumlah objek dalam suatu gambar. Setelah jumlah objek ditentukan, maka objek per satuan luas atau kerapatan juga dapat diperkirakan. Sistem yang dibuat melibatkan konversi gambar input ke dalam format yang diinginkan sehingga jumlah objek dapat dihitung berdasarkan komponen yang terhubung ada dalam gambar yang disempurnakan. Dalam makalah ini, ada tiga aplikasi yang diuji cobakan yaitu *cell counting*, *people counting based on faces*, dan *vehicle counting* dengan menggunakan *toolbox image processing* yang ada pada Matlab. Untuk hasil penelitian dalam *cell counting*, *cell* secara otomatis dapat menentukan nilai kepadatan yang akurat. Pada *people counting based on faces*, algoritme dapat mendeteksi jumlah orang berdasarkan wajah dengan akurat. terakhir dalam *vehicle counting*, algoritme yang digunakan dapat mendeteksi jumlah kendaraan secara akurat termasuk kendaraan roda dua tanpa menggunakan komponen perangkat keras apa pun.

Setelah melakukan review literatur pada penelitian sebelumnya maka penulis mencoba menggabungkan dua metode yaitu algoritme *fuzzy logic* dan *image processing* dengan metode *image matching*, untuk menentukan alokasi waktu lampu hijau yang berdasarkan keadaan kepadatan jalan dengan menggunakan software MATLAB R2013b menghasilkan keluaran berupa persentase *match* citra dan alokasi waktu lampu hijau.

2.2 Tinjauan Teori

2.2.1 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. *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 System (ACTS)*. Sistem ACTS ini secara otomatis dapat mengatur lampu lalu lintas dengan bantuan 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 akan 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 akan secara otomatis mengirimkan informasi mengenai lampu mana yang sedang hijau/merah/kuning [7].

2.2.2 Image Processing

1. Citra

Umumnya citra adalah fungsi dua dimensi $f(x, y)$ di sini x dan y adalah bidang koordinat. Amplitudo citra pada suatu titik (f) disebut dengan intensitas citra. Ini juga dapat disebut tingkat citra abu-abu. Kita perlu mengkonversi nilai x dan y ini menjadi nilai diskrit terbatas untuk membentuk citra digital. Citra input adalah basis data dan basis data drive. Citra diambil untuk diproses dan diperiksa kondisi citra tersebut. Setiap citra terdiri dari elemen, setiap elemen itu disebut piksel [8].

2. Bentuk Citra

Citra memiliki beberapa kondisi untuk membentuk suatu citra dengan fungsi $f(x, y)$ karena nilai citra sebanding dengan energi yang dipancarkan oleh sumber fisik. Jadi fungsi $f(x, y)$ harus bukan nol dan terbatas yaitu $0 < f(x, y) < \infty$ [8].

3. Penskalaan Citra

Penskalaan citra (*Resize Citra*) terjadi di semua citra digital pada tahap tertentu apakah ini dalam *demosaicing Bayer* atau dalam pembesaran citra. Ini terjadi kapan saja saat mengubah ukuran citra dari satu kisi piksel menjadi yang lain. Mengubah ukuran citra diperlukan ketika kita perlu menambah atau mengurangi jumlah total piksel. Bahkan jika ukuran citra yang sama dilakukan, hasilnya dapat sangat bervariasi tergantung pada algoritme. Citra diubah ukurannya karena sejumlah alasan tetapi salah satunya sangat penting untuk penelitian ini. Setiap kamera memiliki resolusi, jadi ketika suatu sistem dirancang untuk beberapa kamera spesifikasi itu tidak akan berjalan dengan benar untuk kamera lain tergantung pada kesamaan spesifikasinya. Jadi perlu untuk membuat resolusi konstan untuk aplikasi dan karenanya perlu melakukan perubahan ukuran citra [8].

4. Konversi Citra berwarna ke Skala Abu-abu

Manusia merasakan warna menggunakan sel-sel sensor peka pada gelombang yang disebut kerucut. Ada tiga macam kerucut yang berbeda, masing-masing memiliki sensitivitas yang berbeda terhadap radiasi elektromagnetik (cahaya) dengan panjang gelombang berbeda. Satu kerucut terutama sensitif terhadap cahaya hijau, satu untuk cahaya merah, dan satu untuk cahaya biru. Dengan memancarkan kombinasi terbatas dari ketiga warna ini (merah, hijau dan biru), dan karenanya itu dapat merangsang tiga jenis kerucut, kita dapat menghasilkan hampir semua jenis

kerucut warna yang terdeteksi. Inilah alasan di balik mengapa citra berwarna sering disimpan sebagai tiga matriks citra yang terpisah; satu menyimpan jumlah merah (R) di setiap piksel, satu jumlah hijau (G) dan satu jumlah biru (B).

Kita menyebutnya citra berwarna yang disimpan dalam format RGB. Namun, dalam citra skala abu-abu, kita tidak membedakan berapa banyak yang kita hasilkan berbeda warna, kita memancarkan jumlah yang sama di setiap saluran. Kita akan dapat membedakan total jumlah cahaya yang dipancarkan untuk setiap piksel; sedikit cahaya memberi piksel gelap dan banyak cahaya yang dirasakan sebagai piksel cerah. Saat mengkonversi citra RGB ke skala abu-abu, kita harus mempertimbangkan nilai RGB untuk setiap piksel dan menjadikannya sebagai output nilai tunggal yang mencerminkan kecerahan piksel itu. Salah satu pendekatannya adalah mengambil rata-rata kontribusi dari setiap saluran dengan nilai $\frac{R+G+B}{3}$. Namun, karena kecerahan yang dirasakan sering didominasi oleh komponen hijau, sebuah perbedaan, lebih "berorientasi manusia", metode ini mempertimbangkan rata-rata, misal: $0,3R + 0,59G + 0,11B$ [8].

5. Peningkatan Kualitas Citra

Peningkatan kualitas citra (*Image Enhancement*) adalah proses menyesuaikan citra digital sehingga hasilnya lebih banyak sesuai untuk tampilan atau analisis lebih lanjut. Misalnya, kita dapat menghilangkan *noise*, yang akan membuatnya lebih mudah untuk mengidentifikasi karakteristik utama. Dalam citra kontras yang buruk, karakter yang berdekatan bergabung selama binarisasi. Kita harus mengurangi penyebaran karakter sebelum menerapkan ambang batas pada citra. Power - Law Transformation yang meningkatkan kontras karakter dan membantu dalam segmentasi yang lebih baik. Bentuk dasar dari Power - Law Transformation adalah seperti [8]:

$$S = cr^\gamma \quad (1)$$

Dimana r dan S adalah intensitas input dan output, masing-masing; c dan γ adalah konstanta positif. Berbagai perangkat yang digunakan untuk pengambilan citra, pencetakan, dan tampilan merespons sesuai dengan Power - Law. Dengan konvensi, eksponen dalam persamaan Power - Law disebut sebagai gamma. Karenanya, proses yang digunakan untuk memperbaiki fenomena respons Power - Law ini disebut koreksi gamma. Koreksi gamma itu penting, jika menampilkan citra secara akurat di layar komputer adalah dari perhatian. Dalam eksperimen ini, γ bervariasi di kisaran 1 hingga 5. Jika c tidak sama dengan '1', maka rentang dinamis dari nilai piksel akan terpengaruh secara signifikan oleh penskalaan. Jadi, harus dihindari tahap lain dari penyelamatan setelah transformasi Power - Law, kami memperbaiki nilai $c = 1$. Dengan $\gamma = 1$, jika citra Power - Law yang diubah dengan binarisasi, tidak akan ada perubahan hasilnya dibandingkan dengan binarisasi sederhana. Ketika $\gamma < 1$, akan ada perubahan dalam histogram plot, karena ada

peningkatan sampel di *thresholding* ke nilai abu-abu nol. Koreksi gamma penting untuk menampilkan citra secara akurat di layar komputer [8].

6. Deteksi Tepi

Deteksi tepi (*Edge Detection*) adalah metode yang bertujuan mengidentifikasi titik-titik di citra digital di mana kecerahan citra berubah tajam atau, lebih teknis, memiliki diskontinuitas atau *noise*. Titik-titik di mana kecerahan citra berubah dengan tajam biasanya disusun menjadi satu set segmen garis melengkung yang disebut tepi. Masalah yang sama dalam mendeteksi diskontinuitas dalam sinyal 1D dikenal sebagai langkah untuk deteksi dan masalah untuk menemukan diskontinuitas sinyal dari waktu ke waktu dikenal sebagai deteksi perubahan.

Deteksi tepi adalah alat dasar dalam pemrosesan citra, khususnya di bidang pembentukan fitur dan ekstraksi fitur. Warna yang berbeda memiliki nilai kecerahan warna yang berbeda. Citra hijau memiliki cahaya cerah lebih banyak daripada citra merah dan biru atau citra biru adalah citra kabur dan citra merah adalah gambar dengan *noise* yang tinggi. Berikut ini adalah daftar berbagai metode deteksi tepi:

1. Teknik deteksi tepi Sobel
2. Teknik deteksi tepi Perwitt
3. Teknik deteksi tepi Roberts
4. Teknik deteksi tepi Zero-Crossing
5. Teknik deteksi tepi Canny

Dalam penelitian ini, peneliti menggunakan teknik deteksi tepi dengan filter sobel karena berbagai keunggulan dibandingkan teknik deteksi tepi lainnya [8].

7. Teknik Deteksi Tepi Sobel

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]:

1. Sejak diperkenalkannya faktor rata-rata (*average*), sobel memiliki beberapa efek smoothing terhadap *noise* acak dari citra.
2. 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], demikian titik pada cartesian dan nilai kepadatan kernel ditunjukkan seperti gambar 2.1:

a	b	c
d	e	f
g	h	i

Gambar 2.1 Gradien sederhana.

Dalam vektor estimasi turunan ‘G’ didefinisikan seperti perbedaan kepadatan / jarak antar kernel. Vektor ini ditentukan sedemikian rupa sehingga arah ‘G’ akan diberikan oleh vektor satuan ke kernel yang diperkirakan. Perhatikan bahwa, kelompok kernel menjadi pasangan antipodal: (a, i), (b, h), (c, g), (f, d). Jumlah vektor untuk menghitung gradien ini [11]:

$$G = \frac{(c - g)}{R} * \frac{[1,1]}{R} + \frac{(a - i)}{R} * \frac{[1,1]}{R} + (b - h) * [0,1] + (f - d) * [1,0] \quad (2)$$

Dimana $R = \sqrt{2}$, Vektor ini diperoleh dari:

$$G = \frac{(c - g - a + i)}{2} + f - d, \frac{(c - g - a + i)}{2} + b - h \quad (3)$$

Di sini, vektor dikalikan dengan 2 karena untuk mengganti yang dibagi dengan 2. Formula yang dihasilkan diberikan sebagai berikut:

$$G' = 2G = [(c - g - a + i) + 2(f - d), (c - g - a + i) + 2(b - h)] \quad (4)$$

-1	0	+1
-2	0	+2
-1	0	+1

Gx

+1	+2	+1
-2	0	+2
-1	0	+1

Gy

Gambar 2.2 Orientasi operator.

Untuk menghasilkan pengukuran terpisah dari komponen gradien di setiap orientasi operator (sebut Gx dan Gy) terdiri dari sepasang masker konvolusi 3×3 sebagai ditunjukkan di atas. Satu *mask* hanyalah yang lainnya jika diputar sebesar 90° . *Mask* ini dirancang untuk merespon garis tepi secara maksimal saat berjalan secara vertikal dan horizontal terhadap kisi piksel, dan satu *mask* untuk masing-masing dari dua orientasi tegak lurus [11].

8. *Image Matching*

Teknik pengenalan berdasarkan pembandingan 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 prototype 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 pembandingan 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 kemudian disimpan ke 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 membandingkan piksel. Jika ada ketidaksesuaian dalam nilai piksel itu maka akan ditambahkan ke dalam penghitung yang digunakan untuk menghitung jumlah ketidakcocokan piksel. Akhirnya persentase perbandingan dinyatakan sebagai [8]:

$$\%match = \frac{\text{No. of pixels matched sucessfully}}{\text{total no.of pixels}} \quad (5)$$

2.2.3 *Algoritme Fuzzy Logic*

Sistem *fuzzy* memiliki tahapan tiga langkah. Langkah pertama, data garing dikonversi menjadi data *fuzzy* atau fungsi keanggotaan (*fuzzifikasi*). Pada langkah selanjutnya keanggotaan fungsi digabungkan dengan aturan kontrol untuk memperoleh keluaran *fuzzy* (*inferensi fuzzy*). Langkah terakhir disebut *defuzzifikasi* di mana setiap output yang terkait dihitung [12].

1. *Pemilihan parameter untuk sistem fuzzy*

Ada beberapa parameter yang dapat mempengaruhi alokasi waktu lampu hijau, salah satunya persentase *match* citra yang diambil dari kamera dengan citra yang menjadi referensi, keadaan jalan (*sepi, normal, dan ramai*), dan luas permukaan jalan (*m*). Dampak dari parameter-parameter ini pada proses pembuktiannya telah dipelajari oleh berbagai peneliti sebelumnya. Namun, faktor pengendali utama yang bisa berubah selama proses percobaan adalah persentase kecocokan. Di sini luas jalan ditetapkan sebagai konstanta. Oleh sebab itu, persentase kecocokan dan keadaan jalan yang dipilih sebagai variabel kontrol. Karena itu nilai yang diperoleh dari membagi persentase relatif yang diukur dari sampel citra dengan persentase relatif dari citra referensi pada

titik waktu yang sama digunakan sebagai satu variabel input (Ppr), input kedua adalah keadaan jalan (Sm), dan outputnya alokasi waktu lampu hijau yang akan disesuaikan dengan inputan (Tgl).

2. Penetapan input dan output *fuzzy*

Gagasan tentang fungsi keanggotaan pada algoritme *fuzzy* yang dikemukakan oleh [13] digunakan pada penelitian ini untuk mengevaluasi parameter input dan output sebagai variabel *fuzzy*. Fungsi keanggotaan berada dalam bentuk segitiga dan fungsi trapesium. Bentuk segitiga adalah fungsi vektor x , dan bergantung pada tiga parameter yaitu a , b , dan c , seperti yang diberikan oleh persamaan (6) dan fungsi trapesium adalah fungsi vector y , yang bergantung pada empat parameter yaitu a , b , c , dan d , seperti diberikan oleh persamaan [14].

$$f(x; a, b, c) = \begin{cases} 0, & x \leq a \\ \frac{c-x}{c-b}, & b \leq x \leq c \\ 0, & x \geq c \end{cases} \quad (6)$$

$$f(y; a, b, c, d) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & b \leq x \leq c \\ 1, & a \leq x \leq b \\ \frac{d-x}{d-c}, & c \leq x \leq d \\ 0, & d \leq x \end{cases} \quad (7)$$

Dalam persamaan (6) parameter a dan c berada pada kaki segitiga dan parameter b ada di ujung puncak segitiga, sedangkan dalam persamaan (7) parameter a dan d berada pada kaki trapesium dan parameter b dan c ada di bahu trapesium. Keluaran sistem (alokasi waktu lampu hijau) bertanggung jawab untuk mengendalikan kecepatan konvergensi sistem, tetapi karena sensitivitas deteksi tepi terhadap citra yang bercahaya tinggi, maka perhatian khusus harus diperhitungkan ketika memilih fungsi keanggotaan dan rentang yang sesuai. Oleh karena itu, untuk mendefinisikan fungsi keanggotaan, beberapa percobaan dilakukan [15]. Untuk input pertama, berdasarkan pada rasio persentase kecocokan relatif (Ppr), tiga keanggotaan fungsi didefinisikan: kecil (S), normal (N) dan besar (B). Untuk variabel input kedua yaitu keadaan jalan (Sm), tiga istilah linguistik dipilih dan diekspresikan oleh set *fuzzy* sebagai: sepi (L), normal (N), ramai (H).

Untuk variabel output alokasi waktu lampu hijau (Tgl) tiga fungsi keanggotaan didefinisikan dan dinyatakan sebagai: rendah (L), normal (N), tinggi (H).

3. Fuzzy rule base

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 akan tinggi. Ini dapat diwakili dengan mengikuti aturan.

$$IF(Ppr = B)and(Sm = H)THEN(Tgl = H) \quad (8)$$

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.

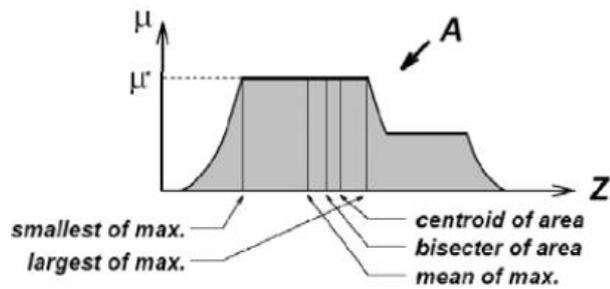
$$IF(Ppr = S)and(Sm = L)THEN(Tgl = L) \quad (9)$$

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

$$IF(Ppr = N)and(Sm = N)THEN(Tgl = N) \quad (10)$$

4. 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 [16]. Ada beberapa metode defuzzifikasi pada komposisi aturan Mamdani, diantaranya yaitu metode *Center of Area* (COA), *Bisektor*, *Mean of Maximum* (MOM), *Largest of Maximum* (LOM), dan *Smallest of Maximum* (SOM) untuk mengubah nilai *fuzzy* menjadi nilai bilangan [14] seperti pada gambar 2.3.



Gambar 2.3 Metode defuzzifikasi pada aturan Mamdani [17].

Dalam penelitian ini penulis 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 [14]:

$$Tgl = \int_{\alpha}^{TglBOA} \mu(Tgl) dTgl = \int_{TglBOA}^{\beta} \mu(Tgl) dgl \quad (11)$$

Dimana:

Tgl = nilai output dari *fuzzy*

$\mu(Tgl)$ = mewakili nilai keanggotaannya

$\alpha = \min \{Tgl | Tsp \in TGL\}$

$\beta = \max \{Tgl | Tsp \in TGL\}$