

MENGATASI KEMACETAN DI LAMPU MERAH DENGAN PENDEKATAN *IMAGE PROCESSING*

Ismail Setiawan

Program Studi Sistem Dan Teknologi Informasi, Universitas 'Aisyiyah Surakarta
Jl. Kapulaga No. 2 Laweyan, surakarta
e-mail: ismail@aiska-university.ac.id

Abstract

The phenomenon of congestion in a big city is generally caused by the large number of vehicles while roads do not develop every year. The growth in the number of vehicles on the highway contributes to a large number of causes of congestion. Congestion often occurs at red light intersections. Various solutions have been implemented such as limiting the number of vehicles based on odd-even numbers by date, limiting motorized vehicles such as motorbikes that are not allowed to pass through the main road, increasing the number of public vehicles and arranging the departure schedule for school children and office employees. The solution has been carried out but there are still long queues of vehicles. One of the reasons for this is because the number of vehicle queues is long while the timer that shows the time to walk or the green light is short. Intelligent systems technology-based approach can be used to overcome these problems. Because humans have limitations to do so, the task can be done by computers. The result of this activity is that the right camera position is next to the traffic timer and the best time to read the image is during the day

Keyword: Camera, Congestion, Intelligent System

PENDAHULUAN

Kemacetan adalah situasi atau keadaan tersendatnya atau bahkan terhentinya lalu lintas yang disebabkan oleh banyaknya jumlah kendaraan melebihi kapasitas jalan (Hayashi et al., 2021). Kemacetan banyak terjadi di kota-kota besar, terutamanya yang tidak mempunyai transportasi publik atau sistem' lalu lintas yang tidak baik atau memadai ataupun juga tidak seimbangnnya kebutuhan jalan dengan kepadatan penduduk, misalnya Jakarta. Kemacetan lalu lintas menjadi permasalahan sehari-hari ditemukan di Pasar, Sekolah, Terminal bus, stasiun (seperti kejadian angkot ngetem sembarangan, kebakaran di pemukiman, dan lain-lain), Lampu merah dan Persimpangan jalan raya maupun rel kereta api di atau perlintasan rel kereta api yang hampir setiap jalan 4 sampai 5 kali yang memakan waktu cukup lama yang akhirnya terjadi penumpukan kendaraan di perlintasan penyeberangan jalan rel kereta api seperti di Jakarta, Surabaya, Bandung, Medan, Semarang, Makassar, Palembang, Denpasar, Yogyakarta, Kota Surakarta, dan kota-kota besar lainnya di Indonesia. Kemacetan lalu lintas dapat disebabkan adanya kecelakaan, banjir, tanah longsor, kebakaran yang menghancurkan mobil dan kebakaran di pemukiman.

Kemacetan dapat terjadi karena beberapa alasan: Arus yang melewati jalan telah melampaui kapasitas jalan, Terjadi kecelakaan terjadi gangguan kelancaran karena masyarakat yang menonton kejadian kecelakaan atau karena kendaran yang terlibat kecelakaan belum disingkirkan dari jalur lalu lintas, Terjadi banjir sehingga kendaraan memperlambat kendaraan, Ada perbaikan jalan, Bagian jalan tertentu yang longsor, Adanya rumah-rumah kumuh/bangunan liar, Kemacetan lalu lintas di Perlintasan sebidang karena adanya kereta api yang lewat, Adanya kendaraan keluar-masuk dan Adanya pembangunan infrastruktur.

Kemacetan lalu lintas memberikan dampak negatif yang besar yang antara lain disebabkan: Kerugian waktu karena kecepatan perjalanan yang rendah, Pemborosan energi karena pada

kecepatan rendah konsumsi bahan bakar lebih tinggi, Keausan kendaraan lebih tinggi karena waktu yang lebih lama untuk jarak yang pendek, Lampu lalu lintas (menurut UU no. 22/2009 tentang Lalu lintas dan Angkutan Jalan: alat pemberi isyarat lalu lintas atau APILL) (Waluyo, 2020) adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyeberangan pejalan kaki (zebra cross), dan tempat arus lalu lintas lainnya. Lampu ini yang menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lalu lintas di persimpangan jalan dimaksudkan untuk mengatur pergerakan kendaraan pada masing-masing kelompok pergerakan kendaraan agar dapat bergerak secara bergantian sehingga tidak saling mengganggu antar-arus yang ada.

Lampu lalu lintas telah diadopsi di hampir semua kota di dunia ini (Wei et al., 2018). Lampu ini menggunakan warna yang diakui secara universal; untuk menandakan berhenti adalah warna merah, hati-hati yang ditandai dengan warna kuning, dan hijau yang berarti dapat berjalan. radiator tidak berfungsi dengan baik dan penggunaan rem yang lebih tinggi, Meningkatkan polusi udara karena pada kecepatan rendah konsumsi energi lebih tinggi, dan mesin tidak beroperasi pada kondisi yang optimal, Meningkatkan stres pengguna jalan, Mengganggu kelancaran kendaraan darurat seperti ambulans, pemadam kebakaran dalam menjalankan tugasnya.

Li & Zhang, (2022) mencoba memberikan solusi untuk mengatasi pengiriman barang menggunakan neuron. Transportasi dikendalikan untuk menghindari kemacetan lalu lintas material dengan meminimalkan fungsi tujuan yang telah ditentukan sebelumnya. Subjek optimasi ke satu set kendala persamaan diferensial parsial (PDE) yang menggambarkan proses transportasi material berdasarkan model transportasi makroskopik molekul-motor-dibantu partikel intraseluler. Model optimasi terbatas PDE yang diusulkan diselesaikan dalam struktur pohon kompleks dengan menggunakan analisis isogeometrik (IGA). Parameter simulasi yang berbeda digunakan untuk memperkenalkan kemacetan lalu lintas dan mempelajari bagaimana neuron menangani masalah transportasi. Hayashi et al., (2021) mengevaluasi metode yang diusulkan menggunakan data aktual sistem penentuan posisi global (GPS) dan menemukan bahwa indeks yang diusulkan dapat mencakup 100% kemacetan lalu lintas dadakan per jalur sementara mengecualikan 82,2% kandidat kemacetan lalu lintas. Indeks yang diusulkan dengan cepat mengkuantifikasi perbedaan antara jumlah agregat mobil saat ini dan nilai keadaan biasa untuk memberi label area yang berisi lebih banyak mobil daripada biasanya sebagai kandidat prioritas tinggi untuk pemrosesan gambar.

Kumar et al., (2022) memberikan usulan dalam mengatur Arus lalu lintas dapat ditangani secara efektif dengan melakukan pra-perkiraan jumlah kendaraan yang melintasi persimpangan yang sibuk pada interval waktu tertentu. Asumsi ini menyajikan sebuah karya yang mampu menghitung kendaraan secara berkala dan menghasilkan alarm ketika kendaraan besar tiba di persimpangan. Selain itu, detail yang dipantau dapat dikirim ke pusat kendali area terpencil yang terletak di mana saja di kota melalui penggunaan internet, pengenalan plat nomor untuk identifikasi kendaraan pencuri menggunakan ALPR dan jika ambulans lewat, itu bisa mengubah sinyal menjadi Hijau menggunakan LORA Trans -modul penerima.

Penelitian ini dilakukan dengan mencoba yang dapat dilihat pada gambar berikut



Gambar 1. Usulan metode penelitian

METODE PENELITIAN

Penelitian ini akan menghitung jumlah kendaraan yang bergerak berdasarkan jenisnya dengan berbasis computer vision dengan tahapan (Pare et al., 2020) (Abdel-Basset et al., 2021): pemilihan ROI (Shah et al., 2020), segmentasi citra dengan metode Gaussian Mixture Model (Meruane et al., 2019), proses filtering, deteksi dan tracking blob, dan klasifikasi kendaraan dengan *thresholding*. Aplikasi yang digunakan adalah matlab versi 2014a (Setiawan et al., 2019). untuk perangkat keras yang digunakan adalah Raspberry pi dan Arduino. Pengambilan data dilakukan di kota Surakarta pada jam 7 pagi hingga 6 sore selama satu minggu di Jalan Slamet Riyadi yang merupakan jalur utama.

HASIL DAN PEMBAHASAN

Jumlah kendaraan yang berhasil direkam selama 10 hari pengambilan data dari masing – masing ruas jalan di persimpangan UMS adalah sebagai berikut.

Tabel 1. Jumlah kendaraan

Ruas Jalan	Motor	Mobil Pribadi	Bis	Truk
Dari Arah Barat	1840	1550	710	1010
Dari Arah Timur	1560	1510	840	1150
Dari Arah Utara	1200	1660	300	490
Total	4600	4720	1850	2650

Tabel 2. Jumlah timer lampu hijau

Ruas Jalan	Timer lampu hijau (detik)
Dari Arah Barat	50
Dari Arah Timur	50
Dari Arah Utara	15
Total	115

Proses awal yang dilakukan pada *image processing* adalah mengubah citra berwarna menjadi citra *gray-scale*. Hal ini dilakukan untuk menyederhanakan model citra. Di dalam suatu gambar *true color* (RGB) terdapat tiga layer matriks, yaitu R-layer, G-layer, dan B-layer. Pada *image processing* dilakukan proses-proses terhadap ketiga layer tersebut, berarti dilakukan perhitungan yang sama pada setiap layer. Dengan demikian konsep *grey-scale* adalah mengubah tiga layer tersebut menjadi satu layer matriks *grey-scale*, yang menghasilkan satu citra *grey-scale*. Di dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan.

Secara umum untuk menghasilkan citra *greyscale*, konversi dilakukan dengan mengambil rata-rata dari nilai r, g, dan b, sehingga menghasilkan nilai s sebagai nilai *grey-scale* nya. Secara matematis dituliskan sebagai berikut (Netto et al., 2018):

$$S = \frac{r+g+b}{3} \quad (1)$$

Sumber : (Netto et al., 2018)

Dimana :

s: nilai gray scale

r: red

g: green

b: blue

Pada penjelasan di atas, perubahan citra berwarna menjadi citra *grey-scale* adalah dengan mencari nilai rata-rata *grey-scale* dari setiap layer r, g, dan b. Beberapa *image* belum optimal jika diberikan cara di atas, untuk keoptimalan citra *greyscale* diperlukan perubahan komposisi pada layer r, g, dan b. Model di bangun dengan *interface* yang sederhana menggunakan MATLAB (Xiang et al., 2021). Beberapa fitur yang ditampilkan adalah tombol *input* gambar, deteksi, konfigurasi kamera, thumbnail, jumlah kendaraan dan waktu menyala lampu merah.

A. Perancangan

Model di bangun dengan merepresentasikan kondisi lalu lintas yang memiliki tiga persimpangan.



Gambar 2. Desain rancangan *interface* aplikasi
Sumber : Dokumentasi Pribadi, 2022

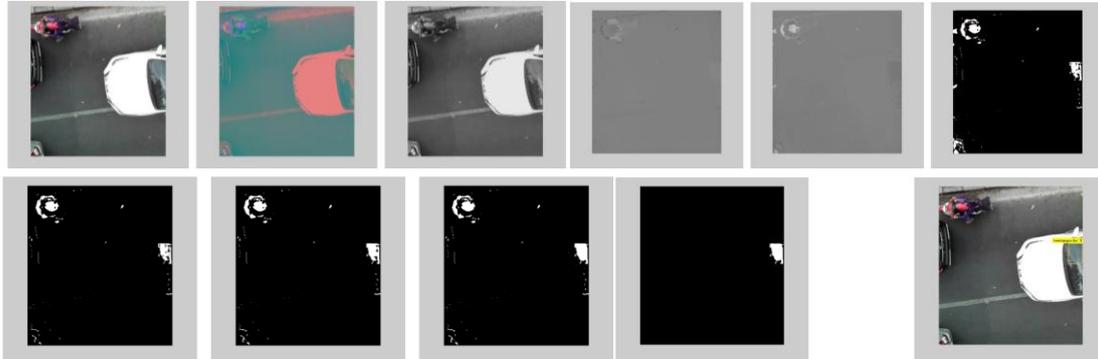
Desain *inteface* dibuat dengan tampilan yang sederhana dan tidak terlalu rumit sehingga memudahkan dalam penggunaannya. Komponen yang terdapat pada model memiliki beberapa item diantaranya tombol konfigurasi kamera, tombol untuk menangkap atau mengambil gambar, tombol *load gambar* dari pengimanan lokal, tombol deteksi untuk memulai kinerja model, informasi jumlah kendaraan dan informasi saran lampu merah menyala dalam satuan detik.

B. Implementasi

Citra yang telah ditangkap oleh kamera kemudian di inputkan dalam aplikasi. Proses yang terjadi sesaat setelah tombol deteksi di klik adalah sebagai berikut :

1. Mengubah citra asli dalam format RGB.
2. Dari format RGB kemudian dilakukan transformasi kdelam ruang warna Yellow, kemudian Chrome Blue lalu Chrome RED.
3. Kemudian gambar ditransformasikan kembali dengan metode *thresholding*.
4. Gambar akan berubah menjadi hitam putih.
5. Ruang citra dengan derajat putih lebih luas dengan lainnya akan dipertahankan dan lainnya dihapus.
6. Ruang citra putih yang masih memiliki kekosongan (bolong-bolong) akan di isi dengan warna hitam untuk mempertegas citra objek yang akan mendapatkan label.
7. Tepian warna citra putih akan dihaluskan dengan operasi morfologi berupa *filling holes* dan *area opening*.

8. Kemudian aplikasi akan menghitung jumlah ruang citra putih.
9. Jumlah ruang citra putih yang masih bertahan akan diberikan label dengan keterangan “kendaraan-n”.



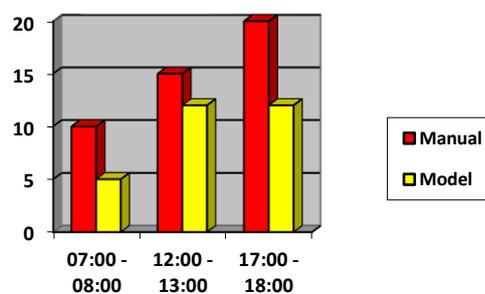
Gambar 3. Proses *Tresholding*
Sumber : Dokumentasi Pribadi, 2022

10. Jumlah label yang bertahan akan digunakan untuk melakukan perhitungan, jika jumlah kendaraan dibawah atau sama dengan 2 maka timer yang muncul menunjukkan angka 25 detik.
11. Jika jumlah kendaraan dibawah atau sama dengan 4 maka timer yang muncul menunjukkan angka 20 detik.
12. Jika jumlah kendaraan dibawah atau sama dengan 6 maka timer yang muncul menunjukkan angka 15 detik .
13. Jika jumlah kendaraan diatas 6 maka timer yang muncul menunjukkan angka dibawah 15 detik.

Angka timer adalah angka yang akan ditunjukkan pada layar LED di lampu merah.

a) Kamera berada disamping posisi timer

Posisi kamera berada di depan antrian mobil dengan sudut antara 45° sampai 70° memiliki kelebihan dan kekurangan masing-masing. Kelebihannya adalah dapat membaca dengan luas jarak yang lebih jauh. Sedangkan kelemahannya adalah jika posisi kamera menghadap matahari pada pukul 7 pagi menyebabkan citra yang ditangkap akan banyak menghasilkan warna putih. Tentu dengan banyaknya warna putih akan menyebabkan model menjadi tidak maksimal dalam mengolah citra. Berikut hasil data dilapangan mengenai kinerja model dalam membaca citra.

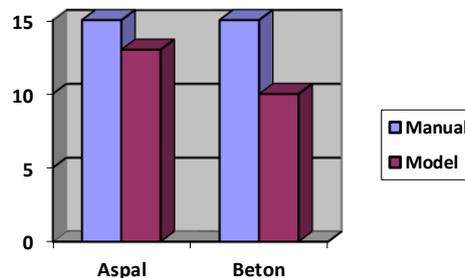


Gambar 4. Kinerja model dengan posisi kamera berada disamping posisi *timer*

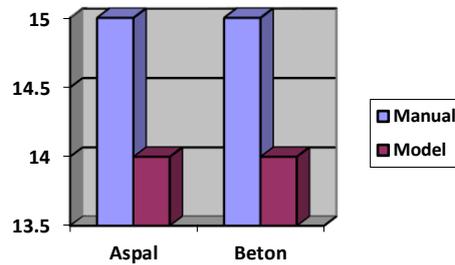
Cahaya matahari pada pukul 7 pagi memiliki dominan warna putih karena kondisi udara pada waktu ini masih memiliki kadar air yang tinggi sehingga cahaya matahari di sebarakan oleh butiran air tersebut. Sedangkan kondisi lainnya adalah pada jalanan yang memiliki konstruksi beton dan aspal ternyata juga memiliki perbedaan seperti pada gambar 6. Rumus penghitungan dapat dilihat sebagai berikut :

```

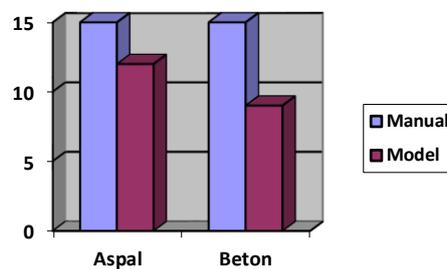
try
    Img3 = handles.Img3;
    out = uint8(zeros(size(Img3,1), size(Img3,2), size(Img3,3)));
    %R,G,B components of the input image
    R = Img3(:,:,1);
    G = Img3(:,:,2);
    B = Img3(:,:,3);
    %Inverse of the Avg values of the R,G,B
    mR = 1/(mean(mean(R)));
    mG = 1/(mean(mean(G)));
    mB = 1/(mean(mean(B)));
    %Smallest Avg Value (MAX because we are dealing with the inverses)
    maxRGB = max(max(mR, mG), mB);
    %Calculate the scaling factors
    mR = mR/maxRGB;
    mG = mG/maxRGB;
    mB = mB/maxRGB;
    %Scale the values
    out(:,:,1) = R*mR;
    out(:,:,2) = G*mG;
    out(:,:,3) = B*mB;
    %Convert the image from RGB to YCbCr
    img_ycbcr = rgb2ycbcr(out);
    Cr = img_ycbcr(:,:,3);
    %Detect Object
    [r,c,~] = find(Cr>120 & Cr<140);
    numind = size(r,1);
    bin = ones(size(Img3,1), size(Img3,2));
    %Mark Object Pixels
    for i=1:numind
        bin(r(i),c(i)) = 0;
    end
    bin = imclearborder(bin);
    str = strel('square',5);
    bin = imclose(bin,str);
    bin = imfill(bin,'holes');
    bin = bwareaopen(bin,3000);
    s = regionprops(bin,'BoundingBox','Eccentricity');
    bbox = cat(1,s.BoundingBox);
    jumlah_kendaraan = size(bbox,1);
    for n=1:jumlah_kendaraan
        label_str{n} = ['kendaraan ke: ' num2str(n)];
    end
    RGB = insertObjectAnnotation(Img3,'rectangle',bbox,label_str,'TextOpacity',0.9,'FontSize',18);
    axes(handles.axes6)
    imshow(RGB)
    title('Citra Hasil Deteksi')
    if jumlah_kendaraan<=2
        durasi = '25 detik';
    elseif jumlah_kendaraan<=4
        durasi = '20 detk';
    elseif jumlah_kendaraan<=6
        durasi = '15 detik';
    end
    set(handles.edit5,'String',jumlah_kendaraan)
    set(handles.edit6,'String',durasi)
catch
    set(handles.edit5,'String','0')
    set(handles.edit6,'String','25 detik')
end
    
```



Gambar 5. Perbandingan Kinerja perhitungan model dengan manual pada waktu antara 07:00 – 08:00



Gambar 6. Perbandingan Kinerja perhitungan model dengan manual pada waktu antara 12:00 – 13:00

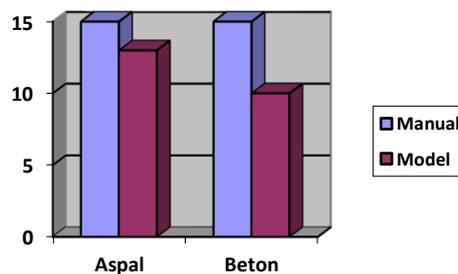


Gambar 7. Perbandingan Kinerja perhitungan model dengan manual pada waktu antara 17:00 – 18:00

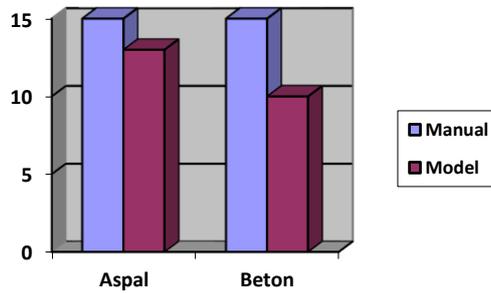
Kinerja terbaik pada keadaan ini adalah saat waktu disiang hari (12:00 – 13:00) karena posisi kamera dalam mengambil citra tidak dipengaruhi oleh cahaya matahari. Objek atau mobil yang memiliki warna kontras dengan jalanan sangat mudah terbaca oleh model.

b) Kamera posisi tegak lurus

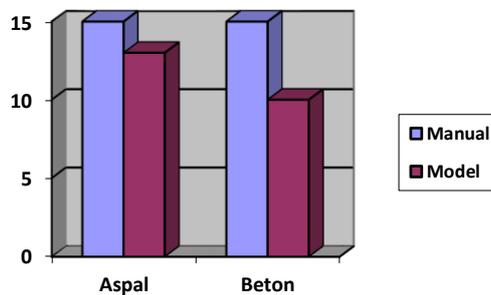
Sama halnya dengan kamera yang terpasang miring, kamera dengan posisi tegak lurus juga memiliki kelebihan dan kekurangan. Kelebihan lebih baik dalam membaca kendaraan pada waktu pagi dan sore. Kelemahan sulit membedakan objek kendaraan dengan jalan saat matahari berada pada posisi tegak lurus atau pada waktu siang hari. Untuk waktu pagi dan sore memiliki kinerja yang lebih baik namun tetap memiliki kendala objek yang memiliki warna berdekatan dengan jalan.



Gambar 8. Perbandingan Kinerja perhitungan model dengan manual pada waktu antara 07:00 – 08:00



Gambar 9.Perbandingan Kinerja perhitungan model dengan manual pada waktu antara 12:00 – 13:00

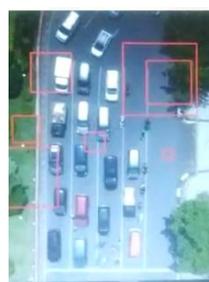


Gambar 10. Perbandingan Kinerja perhitungan model dengan manual pada waktu antara 17:00 – 18:00

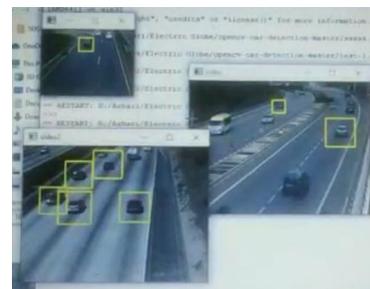
Berdasarkan kegiatan ujicoba model dengan kamera yang terpasang pada drone, lalu didapatkan hasil model mampu membaca objek baik mobil maupun motor. Penelitian ini dikembangkan dengan membuat prototype pengatur timer lampu hijau sesuai dengan kepadatan kendaraan dari masing masing ruas. Timer di rekayasa menggunakan data kepadatan sebelumnya. Sehingga hasil kalkulasi disajikan pada lampu hijau selanjutnya. Program perlu merekam dan menghitung jumlah kendaraan yang berhenti. Pada awalnya kendaraan yang terbaca hanya yang memiliki warna terang dibandingkan dengan warna jalan. Untuk memaksimalkan diberikan tahapan thresholding untuk memisahkan objek yang seharusnya terbaca sebagai sebuah kendaraan.



Gambar 11. Pembacaan objek kendaraan simpang 5 Semarang



Gambar 12. Posisi kamera tegak lurus terhadap jalan simpang 5 Semarang



Gambar 13. Pengujian terhadap objek bergerak di jalan tol

Sumber : Dokumentasi Pribadi, 2022

Setelah metode *thresholding* diterapkan kemudian diujikan kembali menggunakan kamera dengan posisi tegak lurus terhadap ruas jalan. hasilnya pembacaan meningkat dibanding dengan keadaan awal seperti gambar 2. Kemudian program dicoba dengan membaca video yang berisi rekaman kendaraan. Seperti gambar 3 aplikasi berhasil mendeteksi objek bergerak.

Kamera dengan posisi tegak lurus atau 90^0 menunjukkan kinerja optimal saat matahari berada pada posisi pagi dan sore hari. Hal ini disebabkan karena cahaya yang diserap oleh objek tidak dipantulkan secara sempurna ke kamera. Namun kendala yang dihadapi oleh kamera posisi 90^0 pada saat matahari berada tepat diatas kendaraan adalah objek yang memiliki warna hampir sama dengan jalanan akan menyerap dan tidak memantulkan cahaya ke kamera, sehingga model menganggap objek tersebut adalah bagian dari jalanan.

KESIMPULAN

Beberapa kesimpulan yang dapat diambil dari penelitian ini adalah:

- a. Model bekerja lebih baik jika posisi kamera disamping posisi timer adalah pada waktu siang hari
- b. Model bekerja lebih baik jika posisi kamera tegak lurus adalah pada waktu pagi dan sore hari.
- c. Penelitian ini belum diujicobakan pada keadaan hujan dan malam hari.
- d. Sehingga model ini akan lebih maksimal jika penerapannya menggunakan 2 posisi kamera yang difungsikan pada waktu yang memiliki kinerja optimal.

Untuk melengkapi dan memperbaiki penelitian ini selanjutnya dapat menggunakan metode pengolahan citra seperti deteksi tepi. Selain itu metode – metode yang mampu menangkap objek walaupun warnanya sama dengan kondisi jalan sama, model dipastikan dapat bekerja dengan baik. Perlu pengujian lebih lanjut dengan metode tersebut untuk kondisi cuaca mendung atau hujan.

DAFTAR PUSTAKA

- Abdel-Basset, M., Chang, V., & Mohamed, R. (2021). A novel equilibrium optimization algorithm for multi-thresholding image segmentation problems. *Neural Computing and Applications*, 33(17), 10685–10718.
- Goel, V., Singhal, S., Jain, T., & Kole, S. (2017). Specific color detection in images using RGB modelling in MATLAB. *International Journal of Computer Applications*, 161(8), 38–42.
- Hayashi, A., Yokohata, Y., Hata, T., Mori, K., & Obana, K. (2021). Prioritization of Lane-based Traffic Jam Detection for Automotive Navigation System utilizing Suddenness Index Calculation Method for Aggregated Values. *2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan, SICE 2021*, 874–880.
- Kumar, N. S., Surendar, A. A., Prasad, S. H., Vishal, V., & Tribhuvanam, S. (2022). Real Time Intelligent Traffic Light and Density Controller–A Literature Review. *International Journal of Modern Developments in Engineering and Science*, 1(6), 64–66.
- Li, A., & Zhang, Y. J. (2022). Modeling material transport regulation and traffic jam in neurons using PDE-constrained optimization. *Scientific Reports*, 12(1), 1–13.
- Meruane, V., Fernandez, I., Ruiz, R. O., Petrone, G., & Lopez-Droguett, E. (2019). Gapped Gaussian smoothing technique for debonding assessment with automatic thresholding. *Structural Control and Health Monitoring*, 26(8), e2371.
- Netto, A. F. A., Martins, R. N., de Souza, G. S. A., de Moura Araújo, G., de Almeida, S. L. H.,

- & Capelini, V. A. (2018). Segmentation of RGB images using different vegetation indices and thresholding methods. *Nativa*, 6(4), 389–394.
- Pare, S., Kumar, A., Singh, G. K., & Bajaj, V. (2020). Image segmentation using multilevel thresholding: a research review. *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, 44(1), 1–29.
- Setiawan, I., Dewanta, W., Nugroho, H. A., & Supriyono, H. (2019). Pengolah Citra Dengan Metode Thresholding Dengan Matlab R2014A. *JURNAL MEDIA INFOTAMA*, 15(2).
- Shah, S. K., Mishra, R., Mishra, B. S. P., & Pandey, O. (2020). Prediction of abnormal hepatic region using ROI thresholding based segmentation and deep learning based classification. *International Journal of Computer Applications in Technology*, 64(4), 382–392.
- Waluyo, J. (2020). *SOSIALISASI “BERKENDARA YANG BAIK” UU LLAJ NO. 22/2009*.
- Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). Intellilight: A reinforcement learning approach for intelligent traffic light control. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496–2505.
- Xiang, J., Dong, Y., & Yang, Y. (2021). FISTA-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging. *IEEE Transactions on Medical Imaging*, 40(5), 1329–1339.