

PROTOTYPE SMART TRAFFIC LIGHT BERBASIS PENGOLAHAN CITRA DIGITAL MENGGUNAKAN MIKROKONTROLLER

(PROTOTYPE SMART TRAFFIC LIGHT BASED ON DIGITAL IMAGE PROCESSING USING MICROCONTROLLER)

Rifky Abdul Khafid¹, Raditiana Patmasari S.T., M.T², R Yunendah Nur Fuadah S.T., M.T³

^{1,2,3} Prodi S1 Teknik Telekomunikasi, Fakultas Teknik Elektro, Universitas Telkom

¹rifkyabdulkhafid@student.telkomuniversity.ac.id, ²raditianapatmasari@telkomuniversity.co.id,

³yunendah@telkomuniversity.ac.id

Abstrak

Jalan raya merupakan sarana penghubung antara tempat satu ke tempat yang lain untuk mempermudah masyarakat dalam berkendara. Namun peningkatan jumlah kendaraan dari tahun ke tahun sangat mempengaruhi tingkat kepadatan di berbagai ruas jalan. Lampu lalu lintas merupakan salah satu solusi untuk mengurangi angka kemacetan di berbagai ruas jalan. Namun, sistem lampu lintas saat ini belum efektif karena menggunakan sistem fixed time traffic dengan waktu yang telah ditentukan.

Oleh karena itu, pada penelitian kali ini dilakukan sistem pengaturan lampu lintas yang mampu mendeteksi kepadatan jumlah kendaraan di persimpangan jalan dengan menggunakan pengolahan citra digital. Jika salah satu ruas jalan di persimpangan tersebut memiliki jumlah antrian terbanyak, maka lampu lalu lintas di ruas jalan tersebut akan menyala hijau terlebih dahulu. Sistem ini bekerja dengan merekam ruas jalan dan mengambil *frame* dari rekaman tersebut di waktu yang telah ditentukan untuk dijadikan data masukan sistem. Selanjutnya data masukan tersebut diproses menggunakan pengolahan citra digital, kemudian data *output* pada sistem ini diimplementasikan menggunakan lampu LED yang akan menyala pada bagian ruas jalan dengan jumlah kendaraan terbanyak.

Hasil yang diperoleh dari sistem ini yaitu mengetahui ruas mana yang memiliki jumlah kendaraan terbanyak, melalui proses *labelling* dan deteksi tepi untuk mengambil objek yang dibutuhkan. Setelah pengujian pada sistem ini dilakukan dapat ditarik kesimpulan bahwa intensitas cahaya sangat berpengaruh terhadap kinerja dan akurasi sistem. Pada pagi hari kondisi cerah menghasilkan akurasi sistem sebesar 92,50%. Sedangkan pada siang hari kondisi cerah akurasi sistem sebesar 80,00% dikarenakan terdapat intensitas cahaya yang tinggi. Dan pada sore hari kondisi berawan sistem menghasilkan akurasi paling optimal sebesar 95,00%. Dengan demikian rata – rata akurasi sistem yang diperoleh yaitu sebesar 89,16%.

Kata Kunci : Lampu lalu lintas, pengolahan citra digital, mikrokontroler, LED

Abstract

Roads are a way to connect between places to ease commuting. But the increase of vehicle from year to year causes an increase in vehicle density on the road. Traffic lights are one of the solutions to reduce congestion on roads. However, traffic lights system nowadays are mostly inefficient because they usually use fixed time traffic.

Because of that, in this study a cross-lamp regulation system was conducted which was able to detect the density of the number of vehicles at crossroads using digital image processing. If one of the roads at the intersection has the highest number of queues, then the traffic lights on the road will light green first. This system works by recording the road and taking frames from the recording at a specified time to be used as system input data. Furthermore, the input data is processed using digital image processing, then the output data in this system is implemented using LED lights that will light up on the section of the road with the highest number of vehicles.

The results obtained from this system are knowing which section has the highest number of vehicles, through the labeling and edge detection process to retrieve the required objects. After testing the system, it can be concluded that the intensity of the light is very influential on the performance and accuracy of the system. In the morning bright conditions produce system accuracy of 92.50%. Whereas in bright daylight the system's accuracy is 80.00% due to high light intensity. And in the afternoon the cloudy condition of the system produces the most optimal accuracy of 95.00%. Thus, the average system accuracy obtained is 89.16%.

Keywords : Traffic Light, Digital Image Processing, Microcontroller, LED

1. Pendahuluan

Fungsi utama dari jalan raya digunakan sebagai sarana memperlancar transportasi. Namun yang sering menjadi hambatan yaitu terjadi kemacetan yang bermula dari pengguna jalan yang begitu padatnya. Di kota – kota

besar seperti Jakarta, Bandung, Semarang, Surabaya dan lain - lain yang berpotensi mengalami kemacetan akibat dari jumlah pengguna jalan yang semakin memadati lalu lintas.

Salah satu sumber kemacetan yang terjadi berasal dari persimpangan jalan yang padat laju lalu lintasnya. Lampu lalu lintas dengan menggunakan sistem hitungan mundur dan polisi yang bertugas di jalan untuk mengatur lalu lintas, bertujuan untuk memperlancar laju lalu lintas. Namun cara ini tidaklah efektif, karena model lampu lalu lintas normal yang berada di persimpangan jalan tidak bisa mendeteksi padatnya kendaraan, dan polisi yang bertugas tidak setiap waktu bisa mengatur lalu lintas [1]. Oleh karena itu pada tugas akhir kali ini dilakukan sistem pengaturan lampu lalu lintas yang mampu mendeteksi jumlah kendaraan di persimpangan jalan dengan menggunakan pengolahan citra digital.

Hasil yang diperoleh dari pengolahan citra digital tersebut di ilustrasikan dengan skema yang tepat yaitu lampu lalu lintas akan menyala hijau terhadap ruas jalan yang memiliki tingkat kepadatan dan panjang antrian kendaraan yang melewati batas ketentuan. Dan hasil akhirnya di implementasikan menggunakan perangkat keras.

2. Konsep Dasar

2.1 Lampu Lalu Lintas

Lampu lalu lintas adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan. Lampu ini menandakan kapan kendaraan harus berjalan dan berhenti secara bergantian dari berbagai arah. Pengaturan lampu lalu lintas dimaksudkan untuk mengatur pergerakan kendaraan agar bisa berjalan secara bergantian sehingga tidak saling mengganggu arus jalan yang ada [2].

Lampu lalu lintas terdiri dari tiga warna lampu, merah menandakan bahwa kendaraan harus berhenti, lampu kuning untuk hati – hati, dan lampu hijau menandakan kendaraan dapat berjalan.

2.2 Deteksi Tepi

Deteksi tepi berfungsi untuk memperoleh tepi objek. Deteksi tepi memanfaatkan perubahan nilai intensitas yang drastis pada batas dua area [3]. Definisi tepi disini adalah “himpunan piksel yang terhubung yang terletak pada batas dua area” (Gonzales & Woods, 2002). Tepi sesungguhnya mengandung informasi sangat penting. Informasi yang diperoleh dapat berupa bentuk maupun ukuran objek. Umumnya, deteksi tepi menggunakan dua macam *detektor*, yaitu *detektor* baris dan *detektor* kolom. Beberapa contoh yang tergolong jenis ini adalah operator *Prewitt*, *Sobel*, dan *canny*.

2.2.1 Operator *canny*

Operator *canny* yang dikemukakan oleh Jhon Canny pada tahun 1986, terkenal sebagai operator deteksi tepi yang optimal, Abdul Kadir (2013:361). Salah satu algoritma deteksi tepi modern adalah deteksi tepi dengan menggunakan operator *canny* [4]. Terdapat lima langkah yang dilakukan untuk mengimplementasikan deteksi tepi operator *canny* [5], yaitu :

- Langkah pertama : *Smoothing*. Menghilangkan *noise* pada gambar asli, dilakukan dengan mengubah gambar *input* menjadi skala abu – abu dengan menyesuaikan kontras dan kecerahan, sehingga gambar menjadi buram untuk menghilangkan *noise*. Secara umum operator *canny* menggunakan filter *Gaussian* untuk menghilangkan derau atau *noise*.
- Langkah kedua : *Finding gradients*. Perubahan yang tajam pada tingkat nilai abu – abu di *pixel* tepi. *Gradient* adalah satu unit *vector* yang menunjuk ke arah perubahan intensitas maksimum. Komponen *gradient vertikal* dan *horizontal* dihitung terlebih dahulu, kemudian menghitung besar dan arah *gradient*. *Gradient* citra dapat dihitung dengan menggunakan rumus :

$$|G| = |G_x| + |G_y| \quad (1)$$

- Langkah ketiga : *Non maxima suppression*. Pada langkah ini berdasarkan besaran *gradient*, tepi yang tebal pada gambar akan dikonversi menjadi tepi yang tipis dan tajam. Dalam proses ini, gambar dipindai sepanjang arah tepi dan membuang nilai *pixel* yang tidak dianggap sebagai tepi yang akan menghasilkan garis tipis pada gambar *output*. Menghitung arah tepi rumus yang digunakan adalah :

$$\theta = \tan^{-1}(G_x/G_y) \quad (2)$$

- Langkah keempat : *Double Thresholding*. $T1 = High Threshold$, $T2 = Low Threshold$. Piksel yang memiliki nilai skala abu – abu lebih tinggi dari $T1$ artinya piksel tersebut memiliki tepi yang kuat, dan hasilnya menjadi wilayah tepi. Jika piksel yang memiliki nilai skala abu – abu kurang dari $T2$ artinya piksel tersebut memiliki tepi yang lemah, dan hasilnya adalah wilayah non tepi. Jika piksel memiliki nilai tingkat skala abu – abu diantara $T1$ dan $T2$, maka hasilnya tergantung pada piksel tetangga.

- e) Langkah kelima : *Edge tracking by hysteresis*. Proses *hysteresis* bertujuan untuk menghilangkan garis – garis yang terputus – putus.

2.4 Segmentasi Citra

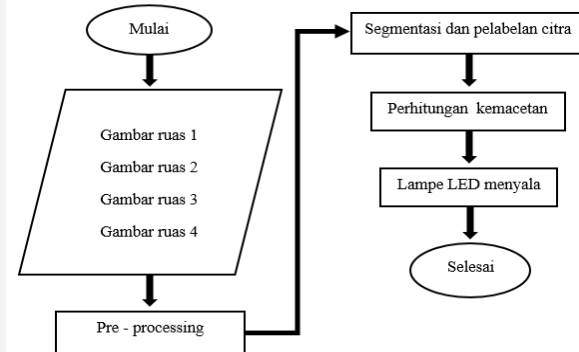
Segmentasi citra merupakan proses yang ditujukan untuk mendapatkan objek – objek yang terkandung di dalam citra atau membagi citra ke dalam beberapa daerah dengan setiap objek atau daerah memiliki kemiripan atribut [6]. Pada citra yang mengandung hanya satu objek, objek dibedakan dari latar belakangnya. Teknik segmentasi citra didasarkan pada aras keabuan, ketidaksinambungan dan kesamaan antar piksel. Pada bentuk pertama, pemisahan citra didasarkan pada perubahan mendadak pada aras keabuan, yang menggunakan pendekatan seperti ini adalah *detector* garis dan *detector* tepi pada citra. Cara kedua didasarkan pada kesamaan antar piksel dalam suatu area (Acharya dan Ray, 2005)

2.5 Arduino Uno

Arduino Uno adalah *board mikrokontroler* berdasarkan ATmega328p (*datasheet*). Arduino uno memiliki 14 pin *input output digital*, 6 diantaranya dapat digunakan sebagai *output PWM (pulse width modulation)*. Uno memiliki *clock speed* 16 MHz, koneksi USB, *power jack*, *header ICSP*, dan tombol *reset* [7]. Komponen tersebut diperlukan untuk mendukung *mikrokontroler*, cukup dengan menyambungkan computer dengan kabel USB untuk memulai.

3. Desain Sistem

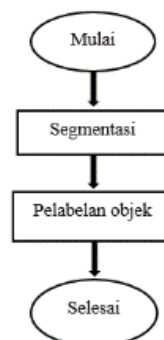
Pada penelitian kali ini akan dirancang sebuah sistem lampu lalu lintas yang mampu mendeteksi kepadatan antrian pada ruas jalan di persimpangan dengan sistem *smart traffic light*. Pada masing – masing ruas akan diambil datanya berupa video *streaming* yang kemudian pada waktu yang telah ditentukan *frame* dari video tersebut dijadikan data masukan yang selanjutnya akan diproses menggunakan pengolahan citra untuk mendapatkan nilai antrian terpadat pada masing – masing ruas jalan. Dan sistem tersebut di implementasikan menggunakan perangkat keras. Gambar 1 merupakan diagram alir yang akan menerangkan langkah – langkah dalam pengerjaan sistem :



Gambar 1. Diagram alir sistem

3.1 Segmentasi

Pada proses segmentasi ini dapat dilihat pada gambar 2, citra yang telah diproses melalui *pre-processing* dipilih kemudian dipisahkan dengan objek yang tidak diinginkan dari latar seluruh citra tersebut.



Gambar 2. Diagram alir proses segmentasi

Citra yang telah diproses melalui pre-processing dipilih kemudian dipisahkan dengan objek yang tidak diinginkan dari latar seluruh citra tersebut. Berikut beberapa tahapan pada bagian proses segmentasi :

1. Segmentasi

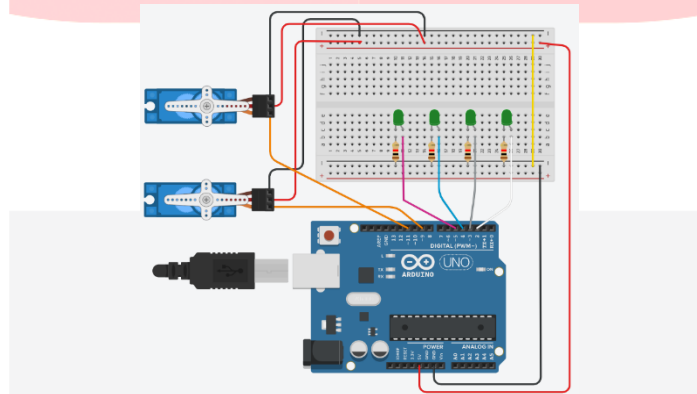
Segmentasi citra dilakukan untuk memisahkan objek dengan latar belakang. Proses pemisahan bertujuan untuk memudahkan proses perhitungan dan klasifikasi dengan tepat dan akurat.

2. Pelabelan objek

Dalam proses ini, informasi penting yang harus diambil adalah objek kendaraan. Objek – objek tersebut akan diberi *label* masing – masing, dengan menggunakan acuan nilai *threshold*, objek yang nilainya lebih kecil dari *threshold* akan dibuang. Proses *labelling* ini, digunakan untuk menghitung banyaknya nilai kendaraan yang telah diberi label. Jumlah label merupakan jumlah kendaraan yang ada di ruas jalan tersebut.

3.2 Perancangan Perangkat Keras

Pada tugas akhir ini untuk mengambil objek kendaraan yang ada di setiap ruas jalan, hanya menggunakan satu *webcam* yang digerakan menggunakan motor servo dengan arah sudut masing – masing ruas jalan sebesar 90 derajat. Citra objek yang berhasil diambil tersebut akan dijadikan citra *input* untuk di proses ke pengolahan citra digital. Setelah citra tersebut di proses, maka akan didapatkan hasil jumlah kendaraan yang paling padat dari ke empat ruas jalan tersebut. *Output* dari hasil tersebut adalah LED berwarna hijau akan menyala di ruas jalan yang terdeteksi mempunyai jumlah kendaraan paling padat. Adapun *design circuit component* tersebut dapat dilihat pada gambar 3.

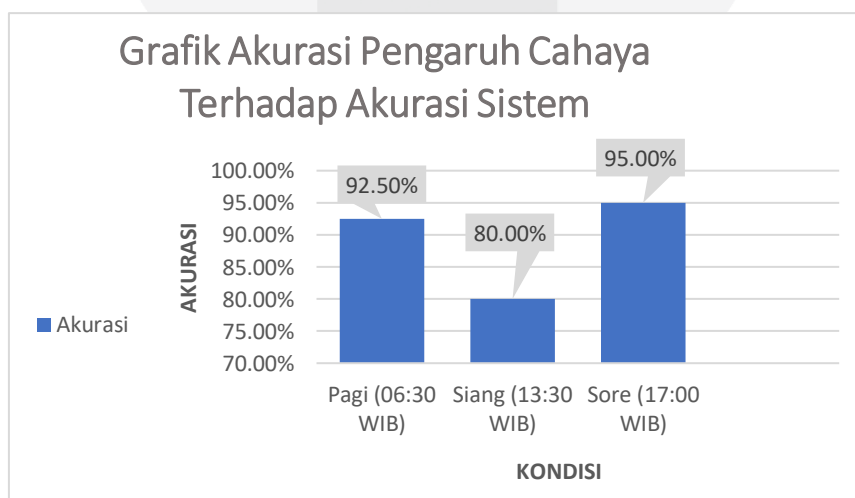


Gambar 3. Rangkaian komponen perangkat keras

4. Pengujian dan Analisis Sistem

4.1 Analisis Pengaruh Cahaya Terhadap Akurasi Sistem

Pada pengujian ini sistem akan dijalankan di tiga kondisi waktu yang berbeda. Dari pengujian pada masing – masing waktu tersebut menghasilkan tingkat akurasi yang berbeda – beda. Perbedaan tingkat akurasi tersebut ditunjukkan dengan grafik batang yang dapat dilihat pada gambar 4 di bawah ini.

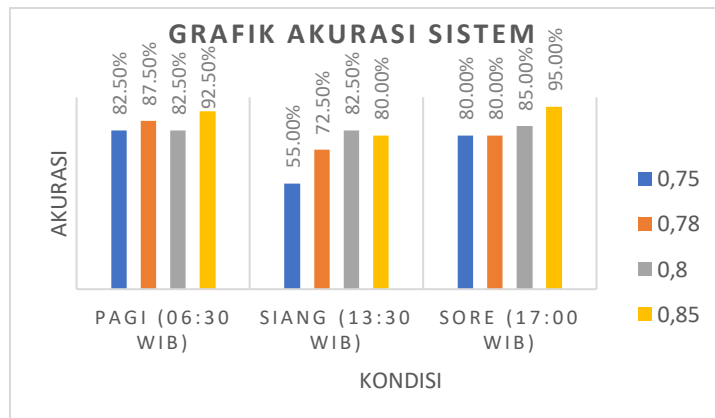


Gambar 4. Pengaruh cahaya terhadap akurasi sistem

Pada gambar 4. menyatakan bahwa tingkat akurasi sistem pada pagi hari kondisi cerah mencapai 92,50%, siang hari kondisi cerah mencapai 80,00%, dan sore hari kondisi berawan mencapai 95,00%. Data *frame* masukan yang diambil pada pagi hari kondisi cerah dan sore hari kondisi berawan memiliki tingkat akurasi lebih tinggi dari pada *frame* masukan yang diambil pada siang hari kondisi cerah. Hal ini menunjukkan bahwa intensitas cahaya sangat mempengaruhi hasil proses pengolahan citra dan tingkat akurasi sistem. Selain itu, pengaruh kondisi warna dan tekstur objek *background* juga memberikan pengaruh pada proses pengolahan citra karena pengaruh komposisi RGB yang mirip antara objek dengan *background*.

4.2 Analisis Sistem Dengan Nilai Konstanta Pengali Level Threshold yang Berbeda Pada Motode Otsu

Pada pengujian kali ini nilai konstanta pengali level threshold pada citra biner akan diubah – ubah untuk mengetahui kinerja sistem. Pada gambar 5 menunjukkan grafik hasil pengujian.

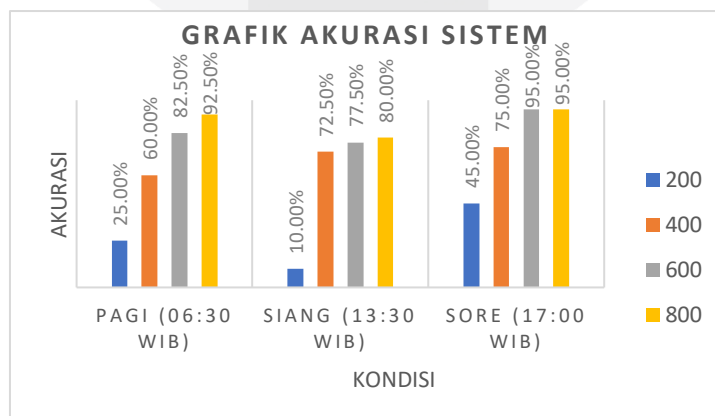


Gambar 5. Grafik pengaruh perbedaan nilai konstanta pengali level threshold pada citra biner

Pada pengujian ini diberikan empat nilai yang berbeda sebagai konstanta pengali *level threshold* untuk mengubah *grayscale* ke citra biner, nilai yang diberikan adalah 0.7, 0.75, 0.78, dan 0.85. Untuk nilai *default* yang diberikan pada sistem ini adalah 0.85. Pengujian ini dilakukan pada tiga kondisi yang berbeda, yaitu waktu pagi hari kondisi cerah, siang hari kondisi cerah, dan sore hari kondisi berawan. Dari gambar 5. dapat dilihat bahwa nilai 0.7 memiliki rata – rata akurasi paling rendah yaitu 61.66%, untuk nilai 0.75 rata rata akurasinya adalah 72,50%, nilai 0.78 rata – rata akurasinya 80,00%. Jika dibandingkan dengan nilai *default* yang diberikan pada sistem yaitu 0.85 sebagai nilai awal, maka tingkat akurasi sistem memiliki hasil yang paling optimal dengan menghasilkan rata – rata akurasi sebesar 89.16%.

4.3 Analisis Tingkat Akurasi Sistem dengan Nilai Threshold Opening yang Berbeda

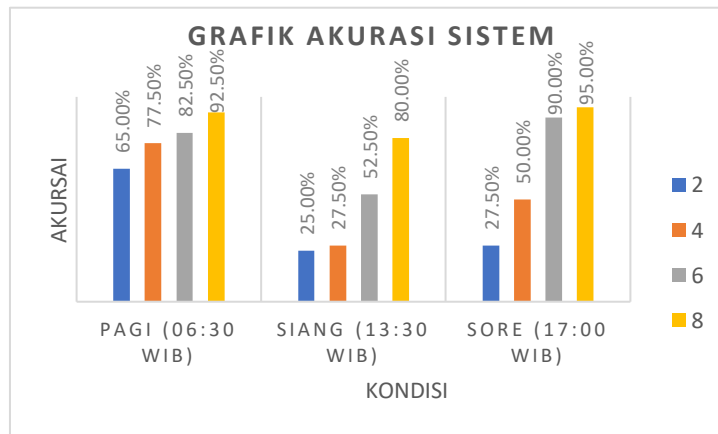
Opening merupakan proses yang bertujuan untuk memperluas batas objek, menghilangkan derau, dan memisahkan objek. Nilai *threshold opening* yang akan diujikan adalah 200, 400, 600, dan 800. Pengujian ini dilakukan pada tiga kondisi waktu yang berbeda, yaitu pagi hari kondisi cerah, siang hari kondisi cerah, dan sore hari kondisi berawan. Gambar 6. menunjukkan grafik hasil pengujian tingkat akurasi sistem dengan nilai *threshold opening* yang berbeda.



Gambar 6. Graifk pengaruh perbedaan nilai *threshold opening*

4.3 Analisis Tingkat Akurasi Sistem dengan Nilai Radius *Structuring Element* yang Berbeda Pada Operasi *Dilasi*

Pada pengujian ini ukuran nilai radius *structuring element* pada operasi *dilasi* akan diubah – ubah untuk mengetahui apakah ukuran nilai radius *structuring element* pada operasi *dilasi* berpengaruh terhadap akurasi sistem. Ukuran yang diujikan pada sistem ini adalah 2, 4, 6, dan 8. Pengujian dilakukan pada tiga kondisi waktu yang berbeda yaitu di pagi hari kondisi cerah, siang hari kondisi cerah, dan sore hari kondisi berawan. Gambar 7 berikut menunjukkan grafik tingkat akurasi sistem terhadap perubahan ukuran nilai radius *structural element*.

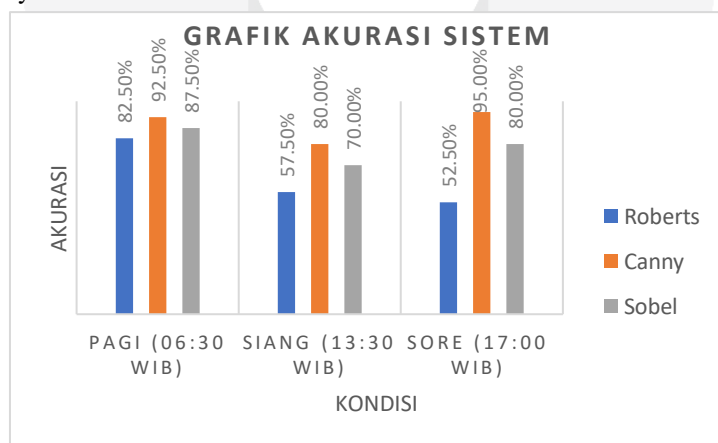


Gambar 7. Graifk pengaruh perbedaan ukuran nilai radius structuring element

Pada sistem ini jenis *structural element* yang digunakan adalah bentuk disk. Dari gambar 7 menunjukkan bahwa nilai 2 untuk ukuran radius pada *structuring element* menghasilkan rata – rata akurasi paling rendah yaitu 39,16%. Untuk ukuran nilai radius 4 pada *structuring element* menghasilkan nilai rata – rata akurasi sistem sebesar 50,00% dan untuk ukuran nilai radius 6 menghasilkan akurasi sistem sebesar 75,00%. Akan tetapi sistem lebih optimal jika diberikan ukuran nilai radius 8 sebagai nilai awal yang diberikan pada sistem dengan menghasilkan rata – rata akurasi sebesar 89,16%.

4.5 Analisis Tingkat Akurasi Sistem dengan Menggunakan Operator Deteksi Tepi yang Berbeda (*Roberts, Sobel, Canny*)

Pada pengujian kali ini sistem akan diberikan jenis operator deteksi tepi yang berbeda - beda untuk menandai objek kendaraan yang akan diambil. Jenis operator deteksi tepi yang akan diujikan adalah operator roberts, sobel, dan canny.



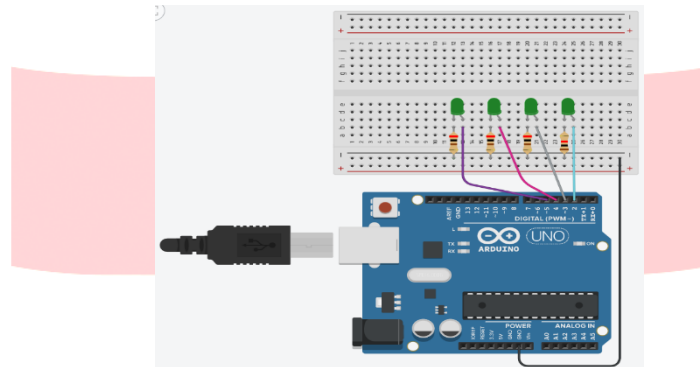
Gambar 8. Grafik pengaruh operator deteksi tepi terhadap akurasi sistem

Pada gambar 8 berikut menunjukkan grafik pengaruh operator deteksi tepi terhadap akurasi sistem. Grafik tersebut menunjukkan bahwa operator *canny* merupakan jenis operator deteksi tepi yang paling optimal untuk sistem ini karena menghasilkan rata – rata akurasi sistem sebesar 89,16%. Karena dari semua jenis operator deteksi tepi, canny merupakan jenis yang paling optimal dengan menggunakan *Gaussian Derivative Kernel* untuk menyaring *noise* dari citra awal supaya mendapatkan hasil deteksi tepi yang halus. Sedangkan untuk operator roberts dan sobel menghasilkan rata – rata akurasi dibawah operator canny yaitu sebesar 64,16% dan 79,16%. Hal

ini membuktikan bahwa operator canny lebih optimal untuk sistem ini dengan penghalusan untuk mengurangi dampak *noise* terhadap pendeteksian *edge*.

4.6 Pengujian Arduino Uno

Untuk pengujian selanjutnya, arduino uno akan diberikan program sederhana untuk menyala – matikan (*blinking*) LED untuk mengetahui waktu komputasi yang dibutuhkan perangkat arduino uno yang difungsikan sebagai mikrokontroler. Pengujian ini diberikan 4 LED yang terhubung ke pin 2,3,4,5 arduino uno, dan diberikan resistor sebesar 220 kΩ sebagai penahan tegangan dan arus yang masuk ke LED, sehingga LED menyala tetapi tidak merusak karena kelebihan arus. Proses *blinking* LED ini dinyala – matikan secara bergantian dengan delay 0,5 detik. Untuk rangkaian pada pengujian ini dapat dilihat pada gambar 9.



Gambar 9. Rangkaian LED yang dihubungkan ke Arduino

Dari hasil pengujian, waktu yang dibutuhkan hingga proses *blinking* LED berjalan satu persatu secara bergantian yaitu 6,15 detik. Pada sistem ini, LED diberikan pada setiap ruas jalan sebagai *output* bahwa ruas jalan yang terdeteksi mempunyai jumlah kendaraan paling padat maka LED pada ruas jalan tersebut akan menyala.

4.7 Pengujian Sistem Smart Traffic Light

Pada pengujian ini sistem akan dijalankan pada kondisi sore berawan untuk mengetahui ruas manakah yang mempunyai jumlah kendaraan terpadat. Dari hasil tersebut, salah satu LED berwarna hijau yang berada di tiap ruas jalan akan menyala sesuai dengan perintah sistem. Untuk hasil pengujian dapat dilihat pada tabel 1.

Tabel 1. Pengujian smart traffic light

Kondisi	Input Jumlah Kendaraan				Output Sistem				LED
	Ruas 1	Ruas 2	Ruas 3	Ruas 4	Ruas 1	Ruas 2	Ruas 3	Ruas 4	ON
1	5	3	6	4	4	6	2	1	Ruas 2
2	8	4	6	3	9	4	6	3	Ruas 1
3	4	6	2	1	4	6	2	1	Ruas 2
4	6	7	4	2	6	7	4	2	Ruas 2
5	6	8	4	3	6	8	4	3	Ruas 2
6	3	5	6	2	3	5	6	2	Ruas 3
7	6	3	7	5	6	3	9	5	Ruas 3
8	1	3	4	5	1	3	4	5	Ruas 4
9	3	2	4	3	3	2	4	3	Ruas 3
10	5	7	6	9	5	7	6	9	Ruas 4

Definisi “*smart*” pada sistem ini adalah ketika mampu mendeteksi jumlah kendaraan terbanyak pada empat ruas jalan dengan kondisi jumlah kendaraan yang berbeda – beda.

5. Kesimpulan

Berdasarkan hasil pengujian dan analisis dari simulasi sistem yang telah dilakukan, maka diperoleh beberapa kesimpulan sebagai berikut :

1. Sistem memiliki rata – rata akurasi sebesar 89,16%. Nilai rata – rata tersebut diperoleh dari hasil pengujian citra yang diambil pada tiga kondisi waktu yang berbeda, yaitu pagi hari kondisi cerah, siang hari kondisi cerah, dan sore kondisi berawan. Pada pagi hari kondisi cerah dan sore hari kondisi berawan merupakan kondisi yang paling optimal untuk sistem ini dalam pengambilan citra dengan akurasi sistem sebesar 92,50% dan 95,00%, karena intensitas cahaya yang tidak begitu tinggi yang dapat mempengaruhi kinerja sistem pada saat proses pengolahan citra. Sedangkan pada siang hari kondisi cerah diperoleh akurasi sistem sebesar 80,00% dikarenakan intensitas cahaya yang sangat tinggi.
2. Deteksi tepi pada citra memiliki beberapa operator seperti *roberts*, *sobel*, *prewitt*, *laplace*, dan *canny*. Operator deteksi tepi yang paling optimal adalah operator *canny*, karena merupakan deteksi modern dengan kelebihan mampu mengekstrak tepian dengan memilih parameter apapun yang akan digunakan sesuai kebutuhan.
3. Arduino uno yang difungsikan sebagai mikrokontroler pada sistem ini diujikan menggunakan program sederhana yaitu *blinking* LED. Waktu komputasi yang dihasilkan hingga program selesai adalah 6,15. Dapat ditarik kesimpulan bahwa arduino uno cukup optimal digunakan untuk difungsikan sebagai mikrokontroler pada sistem ini karena cukup baik dalam menerima rangsangan.

6. Daftar Pustaka

- [1] Maniswari, Soni Dian. Smart Traffic Light Menggunakan Image Processing dan Metode Fuzzy Logic. Bandung. Universitas Telkom, 2015
- [2] Sabdo Aryo, Panitis. Prototype Deteksi Pelanggaran Lampu Lalu Lintas Menggunakan Kamera Sebagai Pendukung Sistem Berbasis Vb6 dan Atmega16. D3 Thesis. Yogyakarta. Universitas Negeri Yogyakarta, 2012.
- [3] Kadir, Abdul, Adhi Susanto. Teori dan Aplikasi Pengolahan Citra. Yogyakarta : ANDI, 2013.
- [4] A. Zalukhu, "Implementasi Metode Canny Dan Sobel Untuk Mendeteksi Tepi Citra," *J. Ris. Komput.*, vol. 3, no. 6, pp. 25–29, 2016.
- [5] A. Bhatnagar, "Image Segmentation Using Canny Edge Detection Technique Kirti," vol. 04, no. 04, pp. 8–14, 2017.
- [6] R. D. Atmaja, F. Teknik, U. Telkom, D. Citra, M. Filter, and M. Operation, "Segmentasi Pembuluh Darah Pada Fundus Retina Menggunakan Matched Filter Dan Operasi Morfologi Segmentation of Blood Vessels in the Retina Fundus Using Matched Filter and Morphology Operation," vol. 4, no. 2, pp. 1661–1668, 2017.
- [7] Arduino, "ARDUINO UNO REV 3" [Online]. Available: <https://www.arduino.cc>. [Accessed on 24 April 2019, 23:00:00 WIB].