



PROTOTYPE SMART TIME SCHEDULER LAMPU LALU LINTAS MENGUNAKAN ALGORITMA HAAR CASCADE

Neci Purwanda¹⁾, Helda Yenni²⁾, M. Khairul Anam^{3*)}, Lathifah⁴⁾

^{1,2}Jurusan Teknik Informatika, STMIK Amik Riau

³ Teknologi Informasi, STMIK Amik Riau

⁴ Teknik Informatika, Universitas Teknorat

^{1,2,3}Jalan Purwodadi Indah Km. 10 Panam, Pekanbaru

⁴ Jl. ZA. Pagar Alam No.9 -11, Labuhan Ratu, Kec. Kedaton, Kota Bandar Lampung

Email: ¹1610031802085@stmik-amik-riau.ac.id, ²heldayenni@sar.ac.id, ³khairulanam@sar.ac.id,
⁴lathifah@teknokrat.ac.id

Abstract

The traffic light functions as a traffic controller at crossroads and also as smoothness at intersections to avoid congestion. The current application of conventional traffic lights still has weaknesses in determining the duration of traffic lights which are not adjusted to the number of vehicle volumes which often change every time. Especially when there is a dense flow of vehicles at intersections, especially at certain times. Based on observations of traffic density at the intersection that connects Soekarno - Hatta road and Arifin Achmad road in Pekanbaru City. There is often a high density of vehicles from the direction of Jalan Soekarno-Hatta (Morning Market) to Jalan Jenderal Sudirman during working hours so that the duration of the traffic lights cannot reduce the amount of queue density in the light traffic. The implementation of the "Smart Time Scheduler" is a solution for adjusting the duration of traffic lights based on the level of traffic density, by building a tool to detect and calculate the number of vehicle queues at the Traffic Light and then enter the calculation results into 3 categories, namely Normal, Medium and Dense Density. The system uses Haar Cascade with the open cv library on the Raspberry pi, the results of the system testing that has been carried out can count the number of vehicles and instruct Arduino to set the duration of the traffic lights where in the category of normal vehicle queues, the green traffic light gets a waiting time of 10 seconds in countdown, Moderate category for 15 seconds, and Solid Category, for 20 seconds, while the calculation error value uses the Error Percentage equation formula, namely comparing the results of vehicle calculations by the system with Manual Calculation Results multiplied by 100% Then the System Calculation Error is 16% and the accuracy level of the tool is $100\% - 16\% = 84\%$, the size of the error value is influenced by the quality of the light intensity, and the distance of the camera's detection of the object when performing the detection.

Keyword: Traffic Light, Smart time, Raspberry pi, Haar Cascade, OpenCV

Abstrak

Lampu lalu berfungsi sebagai pengatur lalu lintas di persimpangan jalan dan juga sebagai kelancaran di persimpangan agar terhindar dari kemacetan. Penerapan lampu lalu lintas konvensional saat ini masih memiliki kelemahan pada penentuandurasi lampu lalu yang tidak disesuaikan dengan Jumlah volume kendaraan yang sering berubah setiap waktu. Terutama pada saat terjadi kepadatan arus kendaraan di persimpangan khususnya pada waktu tertentu. Berdasarkan observasi kepadatan lalu lintas di persimpangan yang mempertemukan jalan Soekarno – Hatta dan jalan Arifin Achmad di Kota Pekanbaru. Sering terjadi Kepadatan kendaraan dari arah jalan Soekarno – Hatta (Pasar Pagi) menuju jalan Jenderal Sudirman pada saat aktifitas jam kerja sehingga durasi nyala lampu lalu lintas tidak dapat mengurangi jumlah kepadatan antrian pada trafik light tersebut. Implementasi *Smart Time Scheduler* menjadi solusi untuk mengatur durasi lampu lalu lintas berdasarkan tingkat kepadatan lalu lintas, dengan membangun sebuah alat untuk mendeteksi dan menghitung jumlah antrian kendaraan pada *Traffic Light* kemudian memasukkan hasil perhitungan kedalam 3 kategori yaitu Kepadatan Normal, Sedang, dan Padat. Sistem menggunakan Haar Cascade dengan *library* open cv pada *Raspberry pi*, hasil Pengujian Sistem yang telah dilakukan dapat menghitung jumlah kendaraan dan menginstruksikan *Arduino* dalam mengatur durasi nyala lampu lintas dimana Pada kategori jumlah antrian kendaraan Normal, lampu hijau trafik light mendapatkan Waktu tunggu selama 10 detik dalam hitungan mundur, kategori Sedang selama 15 detik, dan Kategori Padat, selama 20 detik, adapun nilai error perhitungan dengan menggunakan Rumus persamaan Persentase Error yaitu membandingkan Hasil perhitungan kendaraan oleh sistem dengan Hasil Penghitungan secara Manual dikali 100 % Maka Error Perhitungan Sistem sebesar 16% dan tingkat akurasi alat adalah $100\% - 16\% = 84\%$, besarkecil nya nilai Error ini dipengaruhi oleh kualitas intensitas cahaya, dan jarak deteksi kamera dengan objek pada saat melakukan Deteksi.

Kata Kunci: Traffic Light, Smart time, Raspberry pi, Haar Cascade, OpenCV



1. PENDAHULUAN

Lampu lalu lintas atau bisa disebut juga *Traffic Light* adalah lampu yang terdiri dari tiga warna yaitu Merah, Kuning dan Hijau, yang berfungsi sebagai pengatur lalu lintas di persimpangan jalan [1]. Lampu lalu lintas berfungsi sebagai kelancaran di persimpangan agar terhindar dari kemacetan [2]. Namun jumlah kendaraan di Indonesia semakin lama kian bertambah [3]. Salah satunya di daerah Pekanbaru, menurut data dari Badan Pendapatan Daerah Provinsi Riau (2020) jumlah kendaraan pada tahun 2019 mencapai 770.836 kendaraan. Salah satu bukti lain kepadatan lalu lintas berdasarkan observasi adalah kepadatan lalu lintas di persimpangan yang mempertemukan jalan Soekarno – Hatta dan jalan Arifin Achmad di Kota Pekanbaru. Kepadatan sering terjadi dari arah jalan Soekarno – Hatta (Pasar Pagi) menuju jalan Jenderal Sudirman. Kepadatan ini disebabkan sempitnya ruas jalan dari arah jalan Soekarno – Hatta menuju jalan Arifin dan aktifitas jam kerja sehingga durasi nyala lampu lalu lintas menjadi tidak pas.

Penetapan kinerja lampu lalu lintas saat ini diatur oleh peraturan Menteri Perhubungan Republik Indonesia Nomor PM 49 Tahun 2014 dan metodologi diatur dalam Manual Kendaraan Jalan Indonesia (MKJI) 1997. Setelah diterapkan dilapangan ternyata tidak sesuai dengan durasi lampu lalu lintas dan volume kendaraan yang sering berubah setiap waktu [4]. Durasi lampu lalu lintas tidak disesuaikan saat terjadi kepadatan arus kendaraan di persimpangan jalan khususnya pada waktu tertentu [5].

Penelitian terdahulu terkait dengan lalu lintas telah banyak dibahas, diantaranya ialah melakukan pembelajaran rambu lalu lintas berbasis android [6]. Kemudian penelitian lain melakukan kendali lampu lalu lintas dengan menggunakan metode *fuzzy logic mamdani* [7], penelitian lainnya melakukan pengaturan lalu lintas dengan menggunakan sensor ultrasonic [8], pada penelitian lain juga melakukan pengendalian lalu lintas cerdas menggunakan algoritma *adaptive neuro fuzzy inference system* [9]. Dari beberapa penelitian tersebut penelitian ini menggunakan *Internet of Things* (IoT) dengan menerapkan *smart time scheduler* pada traffic light di kota Pekanbaru. IoT digunakan karena memiliki konsep di mana suatu benda atau objek ditanamkan teknologi-teknologi seperti sensor dan software dengan tujuan untuk berkomunikasi, mengendalikan, menghubungkan, dan bertukar data melalui perangkat lain selama masih terhubung ke internet [10]. Hal ini juga membantu pemerintah pekanbaru dalam menerapkan smart city [11], [12]. Penerapan *smart time scheduler* diharapkan menjadi solusi kemacetan di persimpangan kota pekanbaru.

Implementasi *Smart Time Scheduler* dibangun menggunakan sebuah alat yang dapat mendeteksi dan menghitung jumlah antrian kendaraan pada *traffic light* menggunakan kamera, *Arduino Uno R3* dan *Raspberry Pi* berbantuan library OpenCV [13]. Sistem kerja alat ini yaitu ketika *traffic light* yang terpasang kamera berada pada kondisi lampu merah maka sistem akan menangkap citra objek berupa kendaraan yang antri menunggu lampu hijau kemudian sistem akan mendeteksi dan menghitung jumlah kendaraan lalu memasukkan hasil perhitungan kedalam 3 kategori yaitu normal, sedang, dan padat. Kemudian *Raspberry Pi* akan mengirimkan *Signal HIGH* dan *LOW* ke *arduino* sebagai pengontrol *traffic light* berdasarkan waktu tunggu, dimana ketika *arduino* mendapatkan signal dari kedua input adalah *LOW – LOW* maka waktu tunggu untuk lampu hijau *traffic light* adalah normal yaitu 10 detik, ketika *arduino* mendapatkan signal dari kedua input adalah *HIGH – LOW* maka waktu tunggu untuk lampu hijau *traffic light* adalah sedang yaitu 15 detik, dan ketika *arduino* mendapatkan signal dari kedua input adalah *HIGH – HIGH* maka waktu tunggu untuk lampu hijau *traffic light* adalah 20 detik. Dalam hitungan mundur, library OpenCV dengan algoritma Haar Cascade digunakan sebagai pengolah data citra digital hasil dari kamera yang terinstal pada *Raspberry Pi*.

Dengan diterapkannya alat ini diharapkan bisa menjadi solusi untuk meminimalisir kemacetan bagian ruas jalan yang memiliki kepadatan di waktu tertentu. Kemudian Dengan adanya alat sensor bisa manajemen durasi nyala lampu lalu lintas sesuai dengan kepadatan lalu di waktu tertentu. Semakin panjang antrian kendaraan pada kendaraan maka durasi waktu lampu lalu lintas semakin lama dan semakin pendek antrian kendaraan maka durasi waktu lampu lintas semakin cepat.

2. METODE PENELITIAN

Dalam penelitian ini memerlukan beberapa informasi yang bersifat objektif yang digunakan sebagai acuan penelitian [14]. Dari informasi tersebut diharapkan penelitian ini dapat menghasilkan penelitian yang berkualitas. Proses dalam melakukan penelitian ini digambarkan dalam sebuah bagan alir metodologi penelitian pada Gambar 1 sebagai berikut:



Gambar 1. Bagan Alir Metodologi Penelitian

2.1 Identifikasi Masalah

Pada tahap ini dilakukan proses identifikasi masalah penelitian dan menentukan batasan masalah yang akan dibahas dalam penelitian. Dalam proses identifikasi masalah ini juga penulis menentukan objek penelitian yang digunakan yaitu persimpangan jalan.

2.2 Pengumpulan Data

Adapun metode pengumpulan data yang dilakukan dalam penelitian ini adalah sebagai berikut:

1. Observasi / Pengamatan secara langsung

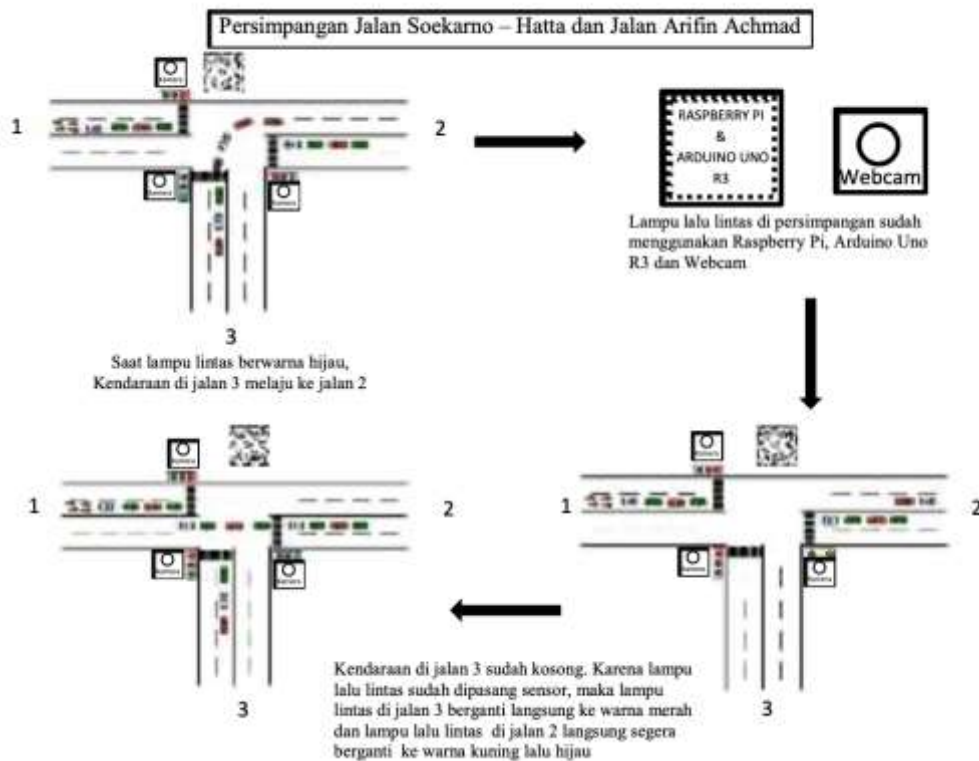
Penelitian yang dilakukan langsung melalui pengamatan pada objek atau sasaran penelitian (Persimpangan Lalu lintas jalan Arifin Achmad dan jalan Soekarno - Hatta), maka akan diperoleh data mengenai penelitian serta data lain yang termasuk dalam perancangan *Smart Time Scheduler* Lampu Lalu lintas Menggunakan Algoritma Haar Cascade yaitu Webcam, *Arduino Uno R3*, *Raspberry Pi 3 B* dan flowchart yang tepat sehingga proses penelitian dapat berlangsung sampai selesai. Untuk itu data yang dicari tersebut harus sesuai dengan masalah dan tujuan penelitian. Sampel yang digunakan merupakan sebuah kendaraan di persimpangan.

2. Studi literatur / Studi Pustaka

Studi literatur atau studi pustaka dilakukan untuk mencari bahan yang mendukung dalam pendefinisian yang terkait dengan penelitian penulis. Bahan-bahan di dapatkan langsung dari jurnal, skripsi, dan sumber terkait lainnya. Dalam proses mencari referensi penulis mempelajari topik penelitian dan permasalahan yang berhubungan dengan penelitian serta mencari landasan-landasan teori mengenai penggunaan Webcam, *Arduino Uno R3*, *Raspberry Pi 3 B* dan yang lainnya yang berhubungan dengan penelitian penulis.

2.2 Analisa dan Perancangan Sistem

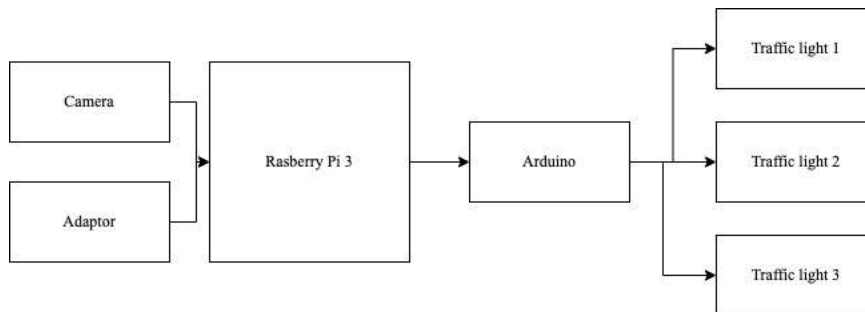
Analisa Sistem baru merupakan tindak lanjut setelah melihat dan mengamati Penetapan durasi lampu lalu lintas yang saat ini diatur oleh peraturan Menteri Perhubungan Republik Indonesia Nomor PM 49 Tahun 2014 atau sistem yang sedang berjalan maka penulis menuangkan sebuah ide berupa pengembangan dari sistem lama. Adapun gambaran dari analisa sistem baru atau sistem yang akan dibangun dapat dilihat dari gambar 2:



Gambar 2. Analisa Sistem yang diusulkan

Pada gambar 2 diatas dapat dijelaskan alur kerjanya sebagai berikut, lampu lalu lintas sudah menerapkan *smart time scheduler*. Lampu lalu lintas sudah dipasang alat berupa kamera dan *Raspberry Pi* sebagai pendeteksi dan penghitung jumlah kepadatan kendaraan sebagai penentu durasi waktu tunggu *traffic light* berdasarkan kategori normal, sedang, dan padat Sehingga meminimalisir waktu menunggu durasi lampu lalu lintas dan menghindari kemacetan di salah satu ruas jalan. Kemudian setelah diketahui konsepnya Langkah selanjutnya ialah merancang alat yang akan diterapkan.

Pada tahap perancangan ini adalah perancangan perangkat keras (hardware), yang meliputi perancangan bagian elektronik dan perancangan bagian mekanik. Adapun bentuk perancangan secara umum dapat dilihat pada blok diagram gambar 3.



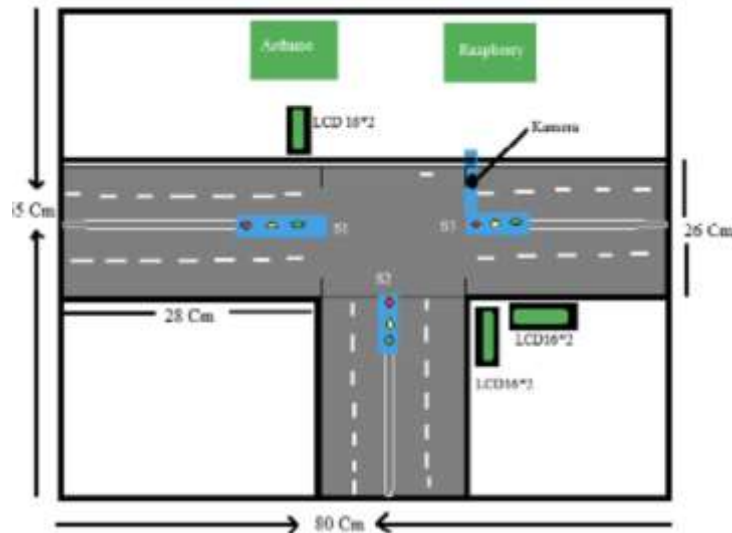
Gambar 3. Blok Diagram

Dari gambar 3 dapat dilihat Perancangan alat secara umum yang diilustrasikan menggunakan blok diagram, pada blok diagram tersebut terdapat 2 buah input yaitu Kamera (Webcam) untuk menangkap citra objek berupa video secara realtime dan Adaptor 5V sebagai sumber tegangan rangkaian, kemudian pada bagian blok Proses terdapat *Raspberry Pi* berfungsi untuk mengolah data input Kamera [15], mendeteksi objek pada citra video, dan menghitung jumlah kendaraan pada citra objek video, hasil output perhitungan jumlah kendaraan diumpankan ke *arduino* yang berfungsi sebagai



pengolah data *Smartime* Management pada *traffic light* dan mengendalikan *traffic light*. Dari perancangan alat, Langkah selanjutnya ialah melakukan perancangan prototype.

Rancangan ini menggambarkan bentuk dari prototype yang dibangun, memiliki ukuran alas jalan dengan Panjang = 80 Cm, Lebar = 65 Cm, kemudian lebar masing-masing ruas jalan dengan 2 jalur adalah 26 Cm, Panjang Jalan (P) = 28 Cm, bahan yang digunakan dalam pembuatan prototype adalah triplex ukuran 5mm, berikut bentuk rancangan prototype terlihat pada gambar 4.



Gambar 4. Rancangan Prototype

2.4 Implementasi Algoritma

pada implementasi ini akan dijelaskan Alur kerja Algoritma sistem dengan OpenCV dimulai dari tahapan Pengambilan citra objek oleh kamera, kemudian proses data citra menggunakan Harscascade untuk mendeteksi objek dilanjutkan dengan menghitung jumlah kendaraan yang terdeteksi, menampilkan Hasil proses data pada terminal python, mengirimkan signal *HIGH* dan *LOW* ke *arduino* sebagai penentu Kategori kepadatan antrian lalu lintas dan mengontrol *traffic light* sesuai dengan durasi tunggu. Berikut ini merupakan penjelasan implementasi sistem yang dibangun.

1. Dimulai dari Inisialisasi sistem dengan memasukkan library OpenCV kedalam program dan membuat variabel yang dibutuhkan dalam menampung data
2. Kamera menangkap objek citra video secara realtime dengan perintah `cv2.VideoCapture(0)` disimpan dalam variabel `Cap` sehingga menjadi `Cap=cv2.VideoCapture(0)`,
3. Membaca data file `detek.xml` dari directory *Raspberry* Pi berfungsi untuk mendeteksi objek dengan perintah `car_cascade=cv2.CascadeClassifier(cascade_src)`
4. *Raspberry* Pi memproses data kamera berupa citra objek video, jika sistem berhasil membaca data kamera disimpan dalam variabel "img" maka ubah warna gambar ke grey dengan perintah `grey=cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)`
5. Setelah warna citra objek diubah ke grey maka sistem akan memulai proses deteksi objek dengan perintah `cars=car_cascade.detectMultiScale(gray,1,1,2)`
6. Ketika objek terdeteksi maka sistem akan melakukan framing terhadap objek yang terdeteksi sebagai tanda atau indikator terdeteksi
7. Menghitung objek didalam *frame* dengan perintah `counter=counter+1` dan tampilkan hasil gambar serta hasil perhitungan kedalam terminal python
8. Membuat hasil perhitungan kedalam 3 kategori yaitu normal, sedang, padat. Kategori normal apabila jumlah kendaraan ≤ 3 , kategori sedang jumlah kendaraan > 3 dan ≤ 5 , dan kategori padat jumlah kendaraan > 5 .
9. Mengirimkan signal *HIGH-LOW* instruksi ke *output GPIO Raspberry* Pi yang terhubung ke *Arduino* sebagai input pemanggilan durasi waktu tunggu pada *Traffic Light*, berikut tabel perintah yang dibuat dalam menandai kategori normal, sedang, dan padat pada *Arduino*.



Tabel 1. Signal perintah kategori kepadatan antrian hasil perhitungan

| No | Logika perintah | | Keterangan |
|----|-------------------------------|-------------------------------|---|
| | <i>GPIO 20 (Output 1)</i> | <i>GPIO 21 (Output 2)</i> | |
| 1 | <i>LOW</i> | <i>LOW</i> | Sebagai input perintah pada <i>Arduino</i> untuk memanggil durasi waktu tunggu Normal |
| 2 | <i>HIGH</i> | <i>LOW</i> | Sebagai input perintah pada <i>Arduino</i> untuk memanggil durasi waktu tunggu Sedang |
| 3 | <i>HIGH</i> | <i>HIGH</i> | Sebagai input perintah pada <i>Arduino</i> untuk memanggil durasi waktu tunggu Padat |

10. *Arduino* menerima data perintah dari input yang terhubung ke GPIO output *Raspberry Pi* kemudian mengontrol LED traffic light pada simpang 1 berdasarkan durasi waktu tunggu normal, sedang, dan padat dari hasil perhitungan jumlah kendaraan yang antri pada saat lampu merah.

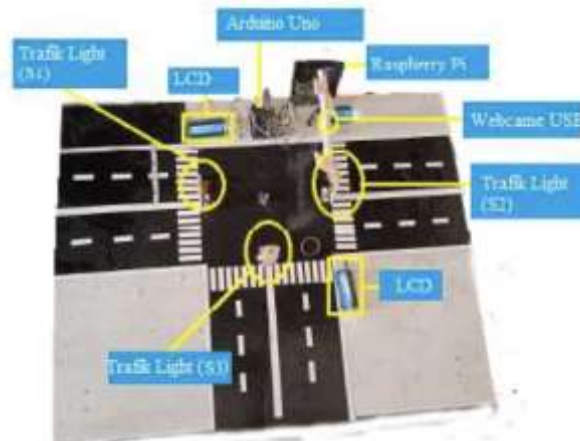
2.4 Implementasi Algoritma

Pengujian ini dilakukan setelah Alat selesai dirancang dan mengimplementasikan Algoritma kedalam pemograman python untuk mendeteksi dan menghitung jumlah kendaraan sebagai penentu kategori kepadatan antrian *traffic light*, kemudian Proses pengujian terhadap masing-masing rangkaian, komponen dan dilanjutkan dengan pengujian alat secara keseluruhan dan di akhiri dengan evaluasi hasil uji untuk mendapatkan kesimpulan keberhasilan alat yang telah dirancang.

3. HASIL DAN PEMBAHASAN

3.1 Hasil Implementasi Alat

Implementasi alat merupakan hasil pembuatan rancangan Alat secara keseluruhan yang telah dipogram menghasilkan sebuah alat *smart time* pada *traffic light* simpang 3, dengan 3 buah Lampu sebagai *traffic light*, 1 buah kamera untuk mendeteksi dan menghitung objek, dan 1 buah *Raspberry Pi* sebagai pengolah data objek berupa citra video, berikut hasil implementasi alat terlihat pada gambar 5.



Gambar 5. Hasil Implementasi Alat

3.2 Pengujian Alat

Dari hasil pengujian alat yang dilakukan dalam beberapa kali pengujian didapatkan data hasil pengujian. Berikut dibawah ini adalah hasil dari pengujian alat yang dilakukan:



3.2.1 Pengujian Alat

Setelah dilakukan proses instalasi *OS streetch* pada *memory card* lalu dipasang ke *Conector Memory Raspberry Pi* , maka selanjutnya dilakukan pengujian dengan menghubungkan *Raspberry Pi* ke sumber tegangan 5V dan Sumber jaringan internet agar dapat di Remote melalui software VNC kemudian mengamati tampilan OS *Raspberry Pi* pada terminal python melalui *VNC Viewer*, berikut hasil pengujian terlihat pada gambar 6.





Gambar 6. Hasil Pengujian *Raspberry Pi*





3.2.2 Pengujian LED Traffic Light

Pengujian ini dilakukan dengan mengupload program uji *_LED.hex* kedalam *board Arduino*, untuk menyalakan seluruh *LED* yang digunakan pada *traffic light* dengan tujuan untuk memastikan bahwa seluruh *LED* dapat berfungsi dengan baik, berikut tabel hasil pengujian *LED traffic light (S1)*, *LED traffic light (S2)* dan *LED traffic light (S3)*.

Tabel 2. Hasil Pengujian LED Traffic Light

| No | Pengujian | Kondisi | | Keterangan |
|----|-------------------------------|---|--|---|
| | | On | Off | |
| 1 | <i>LED traffic light (S1)</i> |  |  | <i>LED traffic light (S1)</i> dapat berfungsi dengan Baik |



| | | | | |
|---|-------------------------------|--|---|---|
| 2 | <i>LED traffic light (S2)</i> |  |  | <i>LED traffic light (S2) dapat berfungsi dengan Baik</i> |
| 3 | <i>LED traffic light (S3)</i> |  |  | <i>LED traffic light (S3) dapat berfungsi dengan Baik</i> |

3.2.3 Pengujian Deteksi Objek

Pengujian ini dilakukan dengan menjalankan perintah pada terminal *Python Raspberry Pi*, kemudian meletakkan objek kendaraan pada sisi simpang yang terpasang kamera, lalu mengamati hasil deteksi objek pada terminal *python*, pengujian ini bertujuan untuk memastikan kamera dapat menangkap dan mendeteksi objek citra video berupa kendaraan, berikut hasil pengujian yang telah dilakukan:



Gambar 6. Hasil Pengujian deteksi objek

3.2.4 Pengujian Smart Time pada Traffic Light

Pada pengujian ini dilakukan dengan 3 pengujian yaitu jumlah antrian kendaraan normal, jumlah antrian kendaraan sedang, dan jumlah antrian kendaraan padat, ketika sistem mendeteksi jumlah antrian kendaraan normal < 3 kendaraan maka waktu tunggu pada *traffic light* simpang 1 adalah 10s, kemudian ketika sistem mendeteksi jumlah



kendaraan >3 dan < 5 maka *traffic light* pada simpang 1 akan mendapatkan waktu tunggu selama 15s, dan ketika sistem mendeteksi jumlah kendaraan > 5 maka *traffic light* pada simpang 1 akan mendapatkan waktu tunggu selama 18s, adapun pengujian dan hasil pengujian dapat dilihat pada poin-poin.

1. Untuk pengujian waktu tunggu normal dilakukan skenario dengan meletakkan 3 buah kendaraan yang sedang antri lampu merah pada simpang 1, dan sistem akan mendeteksi objek dan menghitung jumlah kendaraan tersebut, berikut skenario pengujian yang dilakukan



Gambar 7. Sistem menghitung objek

Dari gambar 7 diatas terlihat hasil perhitungan jumlah kendaraan oleh sistem sebanyak 3 kendaraan yang sedang antri di *traffic light* simpang 1, kemudian sistem menginstruksikan *arduino* untuk mengaktifkan *traffic light* simpang 1 pada saat lampu hijau dengan waktu tunggu normal yaitu 10 detik dalam hitungan mundur (*Counter Down*).



Gambar 8. Hasil pengujian *smart time* dengan waktu tunggu normal

2. Untuk pengujian waktu tunggu sedang, dilakukan skenario dengan meletakkan 4 dan 5 buah kendaraan yang sedang antri lampu merah pada simpang 1, dan sistem akan mendeteksi dan menghitung jumlah kendaraan tersebut, berikut skenario pengujian yang dilakukan:





Gambar 9. Waktu tunggu sedang dengan jumlah 4 kendaraan



Gambar 10. tunggu sedang dengan jumlah 5 kendaraan

Dari gambar 9 dan gambar 10 dapat dilihat hasil deteksi dan hitung jumlah objek oleh sistem adalah 4 – 5 buah kendaraan yang mengantri pada waktu lampu merah *traffic light* simpang 1 menyala, dan sistem menginstruksikan ke *arduino* untuk mengaktifkan lampu hijau pada *traffic light* 1 dengan waktu tunggu 15 detik dalam hitungan mundur.



Gambar 11. Hasil pengujian *smart time* dengan kondisi sedang pada *traffic light*

Dari gambar 4.13 tersebut dapat dilihat *traffic light* simpang 1 mendapatkan waktu tunggu pada saat lampu hijau dengan kategori sedang selama 15 detik.

3. Pada pengujian waktu tunggu Padat, dilakukan dengan scenario meletakkan > 5 kendaraan pada *traffic light* simpang 1 pada saat lampu merah menyala, dan sistem akan mendeteksi dan menghitung jumlah kendaraan tersebut dengan hasil penghitungan dapat dilihat pada gambar 12.



Gambar 12. Hasil deteksi dan hitung jumlah kendaraan kondisi padat



Dari gambar 13 dapat dilihat bahwa kendaraan yang sedang antri pada *traffic light* simpang 1 dalam kondisi menunggu lampu hijau adalah sebanyak 8 namun dari hasil perhitungan jumlah kendaraan oleh sistem terdeteksi sebanyak 7 buah, selanjutnya sistem menginstruksikan *arduino* untuk mengaktifkan *traffic light* simpang 1 pada kondisi lampu hijau menyala mendapatkan waktu tunggu selama 20 detik.



Gambar 13. Hasil pengujian *smart time* dengan kondisi padat pada *traffic light*

Dari gambar 4.15 diatas dapat dilihat bahwa *traffic light* simpang 1 pada kondisi lampu hijau menyala mendapatkan waktu tunggu selama 20 detik dalam hitungan mundur (*Counter Down*).

3.3 Evaluasi Hasil Uji

Pada tahapan Evaluasi Hasil uji ini dilakukan untuk melakukan evaluasi terhadap pengujian alat *smart time* pada *traffic light* yang telah dilakukan untuk mendapatkan kepastian dan kesimpulan keberhasilan alat tersebut, berikut data hasil evaluasi hasil uji terlampir pad Tabel 3.

Tabel 3. Evaluasi Hasil Uji

| No | Pengujian | Harapan | Hasil | Keterangan | |
|----|--------------------------------|--|---|------------|-------|
| | | | | Berhasil | Gagal |
| 1 | Pengujian <i>RaspberryPi</i> | Dapat menampilkan OS yang telah diinstal dan menjalankan program python | Dari hasil pengujian, <i>Raspberry Pi</i> berhasil menampilkan OS dan menjalankan program python | v | |
| 2 | Pengujian pada <i>LCD16x2</i> | Dapat menampilkan informasi nama simpang dan penghitungan waktu secara counter down | Dari hasil pengujian, <i>LCD</i> dapat menampilkan informasi sesuai dengan yang diharapkan | v | |
| 3 | Pengujian <i>traffic light</i> | Dapat menyala secara bergantian dari Lampu Merah, Kuning, dan Lampu Hijau sesuai dengan waktu tunggu | Dari hasil pengujian <i>traffic light</i> dapat menyala secara bergantian | v | |
| 4 | Pengujian kamera | Kamera diharapkan dapat menangkap citra objek berupa video secara realtime | Berdasarkan hasil pengujian kamera dapat berfungsi dengan baik dalam menangkap citra objek berupa video secara realtime | v | |



| | | | | | |
|----|--|---|--|---|---|
| 5 | Deteksi objek | Diharapkan sistem dapat mendeteksi objek dari citra video ditangkap oleh kamera | Sistem dapat mendeteksi objek | v | |
| 6 | Pengujian deteksi objek berdasarkan jarak <50 cm | Diharapkan sistem dapat mendeteksi objek pada jarak <50 Cm | Sistem dapat mendeteksi objek pada jarak < 50 Cm | v | |
| 7 | Pengujian deteksi objek berdasarkan jarak >50 cm | Diharapkan sistem dapat mendeteksi objek pada jarak > 50 Cm | Sistem tidak dapat mendeteksi objek pada jarak > 50 Cm karna objek tertangkap semakin kecil | | v |
| 8 | Pengujian deteksi objek dengan intensitas cahaya <110 Lux | Diharapkan sistem dapat mendeteksi objek dengan intensitas cahaya < 110 Lux | Sistem tidak berhasil mendeteksi objek dengan cahaya < 110 lux | | v |
| 9 | Pengujian deteksi objek dengan intensitas cahaya >110 Lux | Diharapkan sistem dapat mendeteksi objek dengan intensitas cahaya > 110 Lux | Sistem berhasil mendeteksi objek dengan intensitas cahaya > 110 Lux | v | |
| 10 | Pengujian deteksi objek dengan sudut $\geq 0^\circ$ dan $<45^\circ$ | Diharapkan sistem dapat mendeteksi objek dari berbagai sudut dengan kisaran 0° sampai 45° | Sistem dapat mendeteksi objek pada sudut 0° sampai 45° | v | |
| 11 | Pengujian deteksi objek dengan sudut > 45° | Diharapkan sistem dapat mendeteksi objek dari berbagai sudut dengan kisaran > 45° | Sistem sulit mendeteksi objek dengan sudut > 45° | | v |
| 12 | Pengujian sistem menghitung jumlah kendaraan berdasarkan objek yang terdeteksi | Diharapkan sistem dapat menghitung jumlah kendaraan berdasarkan objek yang terdeteksi | Sistem berhasil menghitung jumlah kendaraan | | v |
| 13 | Pengujian sistem menghitung jumlah kendaraan secara akurat | Diharapkan sistem dapat menghitung jumlah kendaraan dengan akurat | Sistem dapat menghitung jumlah kendaraan dengan akurat tergantung intensitas cahaya, Jarak objek, dan sudut antara objek dengan Kamera | | v |



| | | | | | |
|----|--|--|--|---|--|
| 14 | Alat dapat menjalankan traffic light dengan waktu tunggu yang disesuaikan dengan Kategori jumlah antrian kendaraan | Diharapkan alat dapat menyesuaikan waktu tunggu pada traffic light sesuai dengan jumlah antrian kendaraan pada lampu merah | Alat dapat menyesuaikan waktu tunggu pada traffic light sesuai dengan jumlah antrian kendaraan yang terdeteksi oleh sistem pada saat lampu merah | v | |
|----|--|--|--|---|--|

Berdasarkan tabel 3, dapat diketahui bahwa terdapat point yang tidak berjalan, seperti point 7 dan 8. Hal ini dikarenakan pada jarak > 50 Cm, sistem tidak berhasil mendeteksi objek karena dimensi objek semakin kecil dan sehingga sulit untuk dideteksi. Berdasarkan hasil pengujian deteksi objek terhadap intensitas cahaya dapat diketahui bahwa semakin kecil nilai intensitas cahaya < 110 lux yang diterima oleh kamera maka sistem sulit melakukan proses deteksi, dan semakin besar nilai intensitas cahaya yang ditangkap oleh kamera maka proses deteksi akan semakin baik. Namun secara keseluruhan uji coba yang dilakukan berhasil, walaupun untuk diterapkan secara nyata harus dilakukan perbaikan pada *prototype* yang telah dibuat saat ini.

4. KESIMPULAN

Telah dirancang sebuah alat Prototype *Smart Time Scheduler* Lampu Lalu Lintas Menggunakan Algoritma Haar Cascade”, dimana kamera terpasang pada salah satu traffic light untuk menangkap objek berupa kendaraan berdasarkan video secara realtime yang sedang antri pada saat lampu merah kemudian sistem akan mendeteksi kendaraan dengan melakukan framing pada objek yang terdeteksi lalu menghitung jumlah kendaraan tersebut dan menginstruksikan *arduino* sebagai pengontrol traffic light untuk menyesuaikan waktu tunggu lampu hijau pada traffic light berdasarkan kategori normal, sedang, dan padat, dari hasil pengujian yang telah dilakukan maka dapat ditarik kesimpulan bahwa alat dapat mendeteksi, dan menghitung jumlah kendaraan yang antri pada saat lampu merah dan menyesuaikan waktu tunggu pada traffic light berdasarkan kategori normal, sedang dan padat. Kemudian sistem dapat menerapkan haar cascade dalam pengolahan citra digital untuk mendeteksi dan menghitung jumlah kendaraan menggunakan OpenCV. Pada kategori jumlah antrian kendaraan normal, lampu hijau *traffic light* mendapatkan waktu tunggu selama 10 detik dalam hitungan mundur, sedangkan kategori jumlah antrian kendaraan sedang, lampu hijau *traffic light* mendapatkan waktu tunggu selama 15 detik, dan kategori jumlah antrian kendaraan padat, lampu hijau mendapatkan waktu tunggu selama 20 detik. Selanjutnya jarak, intensitas cahaya, dan sudut objek mempengaruhi keberhasilan sistem dalam mendeteksi objek dan melakukan perhitungan jumlah kendaran. Kemudian panjang dan lebarnya objek yang ditangkap oleh kamera mempengaruhi jarak deteksi objek, semakin kecil objek yang akan dideteksi diperlukan jarak yang dekat antara objek dengan kamera. Berdasarkan hasil pengujian penghitungan jumlah kendaraan dan perbandingan data antara sistem dengan manual, didapat error pengukuran sebesar 16 % dengan tingkat akurasi 84 %.

DAFTAR PUSTAKA

- [1] A. Darma Sutisna, H. Sulastri, and E. W. Hidayat, “Sistem Pengefektifan Pemberian Waktu Lampu Lalu Lintas dengan Memanfaatkan CCTV ATCS (Auto Traffic Control System) dan Metode Background Subtraction Korespondensi,” *Scientific Articles of Informatics Students*, vol. 4, no. 1, pp. 15–24, 2021, [Online]. Available: <https://publikasi.unsil.ac.id/index.php/sais>
- [2] Pauzun, “Pengendalian Kemacetan Kendaraan Pada Traffic Light Menggunakan ATMEGA8535,” *JURNAL TEKNOLOGI DAN OPEN SOURCE*, vol. 2, no. 1, pp. 34–40, 2019.
- [3] Priyambodo, “Analisis Korelasi Jumlah Kendaraan dan Pengaruhnya Terhadap PDRB di Provinsi Jawa Timur,” *Warta Penelitian Perhubungan*, vol. 30, no. 1, pp. 59–65, Jul. 2018, doi: 10.25104/warlit.v30i1.634.
- [4] S. M. Hutabarat, F. Lubis, and A. Saleh, “Perencanaan Traffic Light Pada Persimpangan Jalan Garuda Sakti - Jalan Melati -Jalan Binawidya Kota Pekanbaru,” *Jurnal Teknik*, vol. 14, no. 2, pp. 193–202, 2020.
- [5] M. Maslim, B. Yudi Dwiandiyanta, and N. V. Susilo, “Implementasi Metode Logika Fuzzy dalam Pembangunan Sistem Optimalisasi Lampu Lalu Lintas,” *Jurnal Buana Informatika*, vol. 9, no. 1, pp. 11–20, 2018.



- [6] A. Tejawati, M. B. Firdaus, M. N. Ihwan, F. Alameka, and M. K. Anam, "Prototipe Media Pembelajaran Rambu Lalu Lintas Berbasis Android," *Metik Jurnal*, vol. 5, no. 1, pp. 12–18, 2021, doi: 10.47002/metik.v5i1.210.
- [7] P. Juniana and L. Hakim, "Kendali Lampu Lalu Lintas Dengan Menggunakan Metode Fuzzy Logic Mamdani," *JUTEI*, vol. 3, no. 1, pp. 1–10, 2019, doi: 10.21460/jutei.2018.31.126.
- [8] D. O. Deltania, D. Djuniadi, and E. Apriaskar, "Pengaturan Lampu Lalu Lintas (Traffic Light) dengan Sensor Ultrasonik," *Jetri : Jurnal Ilmiah Teknik Elektro*, pp. 77–95, Aug. 2021, doi: 10.25105/jetri.v19i1.8660.
- [9] B. Santoso, A. I. S. Azis, and A. Bode, "Pengendalian Lampu Lalu Lintas Cerdasdi Persimpangan Empat Ruas yang Kompleks Menggunakan Algoritma Adaptive Neuro Fuzzy Inference System," *JEPIN (Jurnal Edukasi dan Penelitian Informatika)*, vol. 6, no. 1, pp. 29–38, 2020.
- [10] M. Susanti, D. Daryanto, R. Wifra, A. Pranata, M. K. Anam, and E. T. Emerlada, "Penerapan Internet of Things Pada Kursi Pintar Menggunakan Arduino Uno Berbasis Android," *INFOTEKJAR: JURNAL NASIONAL INFORMATIKA DAN TEKNOLOGI JARINGAN*, vol. 6, no. 2, pp. 309–313, 2022, doi: 10.30743/infotekjar.v6i2.5091.
- [11] M. K. Anam, T. P. Lestari, Latifah, M. B. Firdaus, and S. Fadli, "Analisis Kesiapan Masyarakat Pada Penerapan Smart City di Sosial Media Menggunakan SNA," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 1, pp. 69–81, 2021, doi: <https://doi.org/10.29207/resti.v5i1.2742>.
- [12] M. K. Anam, B. Nanti, P. Gulo, M. B. Firdaus, and S. Erlinda, "Penerapan Naïve Bayes Classifier , K-Nearest Neighbor dan Decision Tree untuk Menganalisis Sentimen pada Interaksi Netizen dan Pemeritah Applications of Naïve Bayes Classifier , K-Nearest Neighbor and Decision Tree to Analyze Sentiment on Netizen and Gove," *Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, vol. 21, no. 1, pp. 139–150, 2021, doi: 10.30812/matrik.v21i1.1092.
- [13] Sutarti, S. Sasuni, and I. Asseghaf, "Sistem Keamanan Rumah melalui Pengenalan Wajah Menggunakan Webcam dan Library Opencv Berbasis Raspberry Pi," *Jurnal Dinamika Informatika*, vol. 8, no. 2, pp. 13–26, 2019.
- [14] A. Kartino, M. Khairul Anam, Rahmaddeni, and Junadhi, "Analisis Akun Twitter Berpengaruh terkait Covid-19 menggunakan Social Network Analysis," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 5, no. 4, pp. 697–704, 2021, doi: 10.29207/resti.v5i4.3160.
- [15] F. B. Setiawan, H. W. Kusuma, S. Riyadi, and leonardo H. Pratomo, "Penerapan PI Cam Menggunakan Program Berbasis Raspberry PI 4," *CYCLOTRON : Jurnal Teknik Elektro*, vol. 5, no. 2, pp. 51–56, 2022.