

INTEGRATED SMART TRAFFIC CONTROL SYSTEM MENUJU PEKANBARU SEBAGAI SMART CITY

Reny Medikawati Taufiq^{1*}, Sunanto¹, Yoze Rizki¹

¹Teknik Informatika, Universitas Muhammadiyah Riau

email: *renymedikawati@umri.ac.id

Abstract: Pekanbaru still using conventional traffic light control system. Pekanbaru as the capital of Riau Province is predicted undergo the increased of urban population by 54.5% in 2025. It is important for Pekanbaru to immediately implement smart and efficient traffic management system, so that traffic congestion can be resolved quickly. This research paper provides a design solution for smart traffic light management (Smart Traffic Control System), based on object detection technology that uses deep learning to detect the number and type of vehicles. The number of vehicle is the basis for determining the green light timer automatically. The Smart Traffic Control System (STCS) is integrated with a web based geographic information system (smart map) that can display the current condition (picture, the number of vehicle, congestion level) of each STCS location. This integrated system has been tested on a traffic light prototype, using a mini computer and a miniature vehicle. This integrated system is able to detect 9 out of 12 vehicles, and able to send data regularly to the smart map.

Keywords: deep learning; smart mobility; smart traffic control system

Abstrak: Pengaturan lampu lalu lintas di Kota Pekanbaru masih dilakukan secara konvensional. Pekanbaru sebagai ibukota Provinsi Riau diprediksikan akan mengalami peningkatan jumlah penduduk perkotaan sebesar 54,5% pada tahun 2025. Dengan melihat prediksi ini, penting bagi kota Pekanbaru untuk segera memiliki tata kelola lalu lintas yang cerdas dan efisien agar kemacetan dapat ditanggulangi dengan cepat. Penelitian ini memberikan rancangan solusi untuk tata kelola lampu lalu lintas cerdas (*Smart Traffic Control System*), berbasis teknologi *object detection* yang menggunakan *deep learning* untuk mendeteksi jumlah dan jenis kendaraan. Jumlah kendaraan menjadi dasar penentuan timer lampu hijau secara otomatis. *Smart Traffic Control System* (STCS) terintegrasi dengan sistem informasi geografis berbasis web (smart map) yang secara kontinu menerima informasi kepadatan (gambar terkini, jumlah kendaraan, level kepadatan), kemudian menampilkannya diatas peta Kota Pekanbaru. Solusi sistem terintegrasi ini telah diujikan pada sebuah prototipe lampu lalu lintas, menggunakan komputer mini dan miniatur kendaraan. Sistem terintegrasi ini mampu mendeteksi 9 dari 12 kendaraan, dan mampu mengirimkan data secara berkala kepada smart map.

Kata kunci: deep learning; smart mobility; smart traffic control system

PENDAHULUAN

Lebih dari separuh populasi dunia saat ini tinggal di kawasan perkotaan [1]. Tren urbanisasi diproyeksikan akan terus berlangsung. Jumlah penduduk perkotaan di Provinsi Riau, diproyeksikan mening-

kat menjadi 54,5% pada tahun 2025 [2]. Urbanisasi menyebabkan perkembangan kota, namun juga menimbulkan dampak negatif, seperti berkurangnya lahan, polusi, bencana alam, pencemaran sosial ekonomi dan merusak tata kota, juga berdampak terhadap kemacetan lalu

lintas [3]. Kota Pekanbaru sebagai ibukota Provinsi Riau juga mengalami dampak urbanisasi, terlihat dari kondisi Kota Pekanbaru yang mulai sering mengalami kemacetan lalu lintas pada waktu-waktu tertentu seperti di pagi hari (waktu berangkat sekolah dan berangkat kerja), siang hari (waktu istirahat siang) dan sore hari (waktu pulang kerja). Tata kelola sistem pengaturan kepadatan lalu lintas (*traffic control system*) di Pekanbaru saat ini masih menggunakan lampu lalu lintas dengan pengaturan waktu tetap (*fix timer*), tanpa mempertimbangkan kondisi kepadatan kendaraan. Kondisi ini sering menjadi penyebab kemacetan di persimpangan.

Konsep *smart city* (kota cerdas) dianggap dapat menjadi solusi dari permasalahan kemacetan dipersimpangan diatas. *Smart city* dideskripsikan sebagai penggunaan teknologi komputasi cerdas untuk mengintegrasikan komponen-komponen penting dari infrastruktur dan layanan kota, seperti administrasi kota, pendidikan, kesehatan, keselamatan publik, *real estate*, transportasi, dan keperluan kota lainnya, dimana penggunaan keseluruhannya harus dilakukan secara cerdas, saling berhubungan, dan efisien [4].

Beberapa penelitian terdahulu yang relevan adalah deteksi kendaraan dengan menggunakan metode *blob detection* [5], *realtime background subtraction* [6], *edge detection* [7]. Beberapa solusi pengaturan lampu lalu lintas cerdas menggunakan sensor sebagai alat deteksi kendaraan, seperti sensor *infrared* dan sensor ultrasonik [8][9]. Namun dengan menggunakan sensor, data yang didapat terbatas dan sulit untuk dilakukan perawatan alat [10]. Teknologi pengontrolan lampu lalu lintas menggunakan *image processing* merupakan solusi yang menghemat biaya penggunaan *hardware*

[10]. Untuk memperbaiki sistem kontrol lalu lintas, penggunaan teknologi *object detection* dapat membuat lebih fleksibel dan lebih praktis [11].

Menuju Pekanbaru Kota Pintar, *Integrated Smart Traffic Control System (ISTCS)* dapat digunakan sebagai solusi tata kelola lalu lintas kendaraan berbasis teknologi. *ISTCS* merupakan solusi sistem deteksi dan kontrol otomatis yang terintegrasi dengan sistem informasi geografis. Terdapat dua bagian pada *ISTCS* yang saling terintegrasi, yaitu *Smart Traffic Control System (STCS)* dan *Smart Map*. *STCS* adalah *processing unit* yang terkoneksi dengan *IP camera*, diprogram menggunakan bahasa Python untuk mengatur *timer* lampu lalu lintas berdasarkan data jumlah kendaraan yang terdeteksi. Jumlah dan jenis kendaraan didapat dari proses deteksi objek menggunakan *deep learning* dengan inputan video dari *IP camera*. Jika antrian kendaraan berstatus padat, maka *STCS* akan mengatur lama menyala lampu hijau lebih panjang. *STCS* juga mengirimkan informasi kepadatan lalu lintas tersebut ke *smart map*. *Smart map* adalah sistem informasi geografis yang menampilkan peta dan lokasi lampu lalu lintas yang ada di Kota Pekanbaru. Secara kontinu *smart map* menerima informasi kepadatan lalu lintas dari masing-masing *STCS* dan menampilkan informasi tersebut pada peta.

METODE

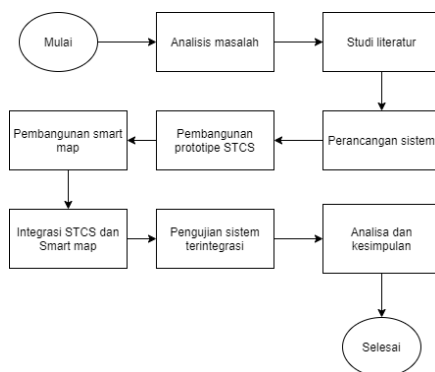
Analisis Masalah

Pengaturan durasi *timer* lampu lalu lintas secara tetap tanpa mempertimbangkan kondisi kepadatan kendaraan (tidak adaptif) pada titik lampu lalu lintas dapat memicu kemacetan. Permasalahan ini, dapat diatasi dengan menerapkan Sis-

tem Cerdas Pengatur Kepadatan Lalu Lintas (*Smart Traffic Control System (STCS)*), yang dapat mengatur *timer* lampu hijau secara otomatis, sesuai dengan keadaan kepadatan antrian kendaraan.

Studi Literatur

Dari hasil tinjauan beberapa referensi, pada penelitian ini akan menggunakan *object detection* yang menggunakan metode *deep learning*. Langkah-langkah lengkap yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 1.



Gambar 1. Langkah Penelitian

HASIL DAN PEMBAHASAN

Perancangan Sistem

Pada *Integrated Smart Traffic Control System (ISTCS)* terdapat dua bagian sistem yaitu *Smart Traffic Control System (STCS)* dan *smart map*. STCS berfungsi mendeteksi kepadatan kendaraan yang sedang mengantri di lampu lalu lintas. Pada setiap STCS terdapat sebuah *IP camera*, modul kontrol (*object detection* dan logika pengaturan lampu) dan rangkaian prototipe lampu lalu lintas. Kamera akan melakukan streaming video keadaan lalu lintas kepada modul kontrol. Oleh modul kontrol, video akan diproses menggunakan metode *object detection*,

hasil pemrosesan berupa jumlah dan jenis kendaraan. Jumlah kendaraan akan diterjemahkan menjadi informasi level kepadatan (lengang, sedang, padat). Informasi jumlah dan jenis kendaraan, serta level kepadatan kemudian akan dikirimkan ke *smart map* secara kontinu. Modul kontrol menggunakan Raspberry Pi 4b+, program kontrol dan *object detection* disusun dengan bahasa pemrograman Python. Library OpenCV digunakan untuk pemrosesan image. Framework *deep learning* menggunakan *Tensorflow Lite*. Rancangan umum sistem prototipe ISTCS dapat dilihat pada Gambar 2.



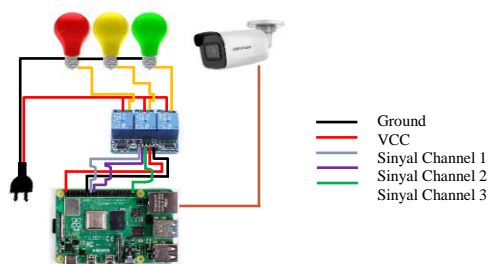
Gambar 2. Rancangan Sistem ISTCS

Pembangunan Prototipe STCS

STCS adalah bagian yang melakukan kontrol lampu lalu lintas. Perangkat keras yang ada pada STCS adalah sebuah Raspberry Pi yang terhubung dengan modul relay 3 channel dan lampu bohlam berwarna. Pada ujicoba digunakan Raspberry PI 4B+, 64-bit SoC dengan RAM 4GB dan memori 16 GB. IP camera yang digunakan adalah Hikvision DS-2CD2021G1-I, terhubung dengan Raspberry PI melalui ethernet. STCS dan Smart Map terhubung melalui jaringan *wi-fi*. Untuk implementasi riil, STCS dipasang pada setiap titik lampu lalu lintas. Namun pada simulasi ujicoba yang dilakukan pada pembahasan penelitian ini, STCS mengontrol nyala lampu bohlam berwarna merah, kuning, hijau yang merupakan prototipe lampu lalu lintas. Skema lengkap rangkaian

STCS dapat dilihat pada Gambar 3.

Untuk mengirim sinyal pengaturan lampu bohlam digunakan 3 pin GPIO yang terhubung ke modul relay dengan 3 *channel*. Masing-masing pin GPIO mengirimkan sinyal untuk pengaturan lampu merah (terhubung ke *channel 1*), kuning (terhubung ke *channel 2*) dan hijau (terhubung ke *channel 3*). Modul relay menjadi perantara GPIO untuk mengontrol lampu dengan membuka atau menutup arus listrik, jika GPIO mengirim sinyal HIGH maka relay akan membuka sehingga lampu menyala, sebaliknya ketika GPIO mengirim sinyal LOW maka relay akan menutup arus sehingga lampu padam. Pin GPIO dan pin relay dihubungkan dengan kabel *jumper female-female*. Pin GPIO yang digunakan pada skema rangkaian dapat dilihat pada Tabel 1.



Gambar 3. Skema Rangkaian STCS

Tabel 1. Penggunaan Pin Raspberry Pi

| Pin | Koneksi | Fungsi |
|-------------|---------------|-------------------------|
| 1 (3V) | Relay pin VCC | Power modul relay |
| 14 (GROUND) | Relay GND | Ground modul relay |
| 16 (GPIO23) | Relay CH1 | Data relay lampu merah |
| 18 (GPIO24) | Relay CH2 | Data relay lampu kuning |
| 40 (GPIO21) | Relay CH3 | Data relay lampu hijau |

Program pengontrol yang dijalankan pada Raspberry Pi dibangun menggunakan bahasa Python. Setiap detik program ini mengambil frame dari streaming video IP camera. Protokol yang digunakan untuk transfer video adalah *Real Time Streaming Protokol (RTSP)*. Terdapat beberapa protokol *streaming*, protokol standar yang sering digunakan adalah HTTP dan RTSP. Rombe [12] melakukan perbandingan QoS untuk dua protokol ini dan menyarankan penggunaan protokol RTSP. HTTP tidak direkomendasikan sebagai protokol *streaming* melalui internet karena berdasarkan analisis pada semua parameter yang diujikan memiliki Delay, Packet Loss yang sangat tinggi dan Throughput-nya yang kecil [12].

Video akan diproses menggunakan metode *object detection*, sehingga menghasilkan data jumlah dan jenis kendaraan, lalu diterjemahkan menjadi informasi level kepadatan (lengang, sedang, padat). Proses *deep learning* pendeteksian objek harus mempertimbangkan keterbatasan lingkungan implementasi STCS yaitu komputer mini Raspberry Pi. Maka perlu ditentukan penggunaan interpreter dan model terlatih yang dapat bekerja pada lingkungan terbatas. Pada penelitian ini digunakan interpreter Tensorflow Lite dengan model terlatih *MobileNet-Single Shot Multibox Detector* yang sudah dioptimasi dengan metode kuantisasi (Quantized MobileNet-SSD). Terjadi penurunan ukuran model yang sudah dikuantisasi, sehingga membutuhkan lebih sedikit tempat penyimpanan dan penggunaan RAM, hal ini dapat berpengaruh pada performa dan stabilitas proses deteksi [13].

Setiap detik program pengontrol melakukan deteksi objek dan mendapatkan status level kepadatan, lalu me-

manggil fungsi *LightController* dengan mengirimkan parameter jumlah kendaraan yang terdeteksi. Fungsi *LightController* bertugas mengatur nyala lampu merah, kuning dan hijau secara bergantian. Timer lampu hijau akan disesuaikan dengan level kepadatan. Jika level kepadatan= “padat” maka lampu hijau akan menyala selama 20 detik. Jika level kepadatan=”sedang” maka lampu hijau akan menyala selama 10 detik. Jika level kepadatan=”lengang” maka timer akan mengikuti *default* yaitu 5 detik. Penyesuaian *timer* lampu hijau hanya dilakukan 3 detik sebelum lampu merah padam (tidak setiap detik), namun deteksi kepadatan dan pengiriman data ke *Smart Map* dilakukan setiap detik.

Pembangunan Smart Map

Smart Map adalah Sistem Informasi Geografis (SIG) berbasis web yang menampilkan peta lokasi lampu lalu lintas yang ada di Kota Pekanbaru dan informasi kepadatan pada titik lampu lalu lintas. *Smart Map* dibangun menggunakan bahasa pemrograman PHP, Apache webserver sebagai penyedia layanan web dan menggunakan DBMS MySQL untuk menyimpan data lokasi dan status *STCS*. *Smart Map* menggunakan *Google Maps API* sehingga dapat mengambil peta yang disediakan oleh Google Maps untuk ditampilkan sebagai peta dasar pada *Smart Map* dengan parameter koordinat Kota Pekanbaru.

Simbol lampu lalu lintas digunakan untuk menampilkan lokasi *STCS* pada peta. Untuk memberikan peringatan (*warning*) kondisi kepadatan, digunakan simbol dengan tiga warna berbeda, yaitu warna simbol merah, kuning dan hijau yang menandakan kondisi padat, sedang dan lengang. Terdapat informasi berupa teks pada posisi sebelah kanan peta yang menampilkan seluruh

daftar *STCS* beserta status kepadatannya. *STCS* dengan status padat akan tampil pada urutan teratas. Tampilan peta dan simbol pada *Smart Map* diperbaharui setiap detik agar dapat menampilkan kondisi terkini. Pada *Smart Map* terdapat *listener* yang menangkap data yang dikirimkan oleh program kontrol *STCS* melalui protokol HTTP method POST.

Pengujian Sistem Terintegrasi

Implementasi dan uji coba penelitiannya ini dilakukan pada sebuah prototipe sistem. Prototipe *STCS* dapat dilihat pada Gambar 4.



Gambar 4. Prototipe *STCS*

Terdapat beberapa ujicoba yang dilakukan pada *STCS* yaitu pengukuran *frame rate per second (fps)*, akurasi deteksi jumlah kendaraan pada kondisi terang dan kondisi redup.

Tabel 2 menampilkan hasil pengukuran nilai *fps* (S=Detik, FR=Frame Rate). Pengukuran *frame rate* dilakukan dengan memanfaatkan fungsi *cv2.getTickCount* dan *cv2.getTickFrequency*. Fungsi *cv2.getTickCount* mengembalikan nilai jumlah *clock-cycle* pada suatu waktu. *cv2.getTickFrequency* mengembalikan nilai jumlah *clock-cycle* per detik. *Fps* didapat dengan menggunakan persamaan (1)

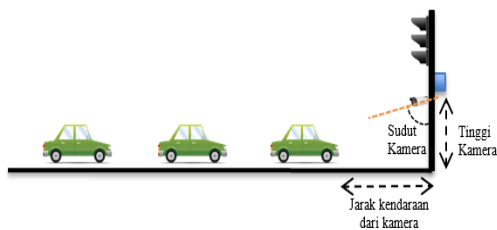
$$Frame\ rate = \frac{Jumlah\ clock\ cycle\ detik}{t2 - t1}$$

t1 = jumlah *clock cycle* sebelum *object detection* dilakukan
t2 = jumlah *clock cycle* sesudah *object detection* dilakukan

(1)

Rata-rata jumlah *fps* yang mampu diolah oleh program pengontrol STCS adalah 4,56. Nilai ini sudah mendekati *fps* minimum yang direkomendasikan oleh *American Public Transportation Association* [14] yaitu 5 *fps* untuk kategori area *low-traffic*. Antrian kendaraan pada lampu merah dapat dikatan sebagai *low-traffic* area karena kendaraan dalam kondisi berhenti/hampir berhenti pada saat lampu merah menyala.

Pengujian STCS dilakukan menggunakan miniatur kendaraan dalam kondisi terang (seolah pagi/siang hari) dan redup (seolah kondisi malam hari). Gambar 5 menunjukkan skema ujicoba deteksi kendaraan.



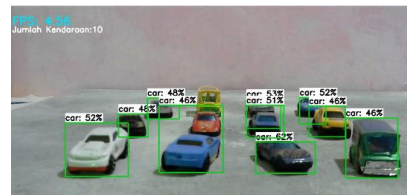
Gambar 5. Skema Uji Coba Deteksi Kendaraan

Dilakukan 2 uji coba dengan dengan sudut dan tinggi kamera yang berbeda, dan 2 uji coba dengan 2 jenis nilai jarak antar mobil. Tabel 3 merupakan hasil uji coba deteksi kendaraan dengan pengaturan ketinggian kamera = 10 cm, jarak kamera dari kendaraan terdepan= 36 cm, sudut kamera = 65 °, jarak antar kendaraan (ke samping) = 4 cm, jarak antar kendaraan (depan/belakang)= 7 cm, jumlah miniatur kendaraan =12 unit, jumlah terdeteksi=9 unit. Hasil deteksi dalam keadaan terang dan redup dapat dilihat pada Gambar 6 dan 7. Dari hasil deteksi ini akurasiya adalah sebesar 75%. Miniatur kendaraan terdeteksi sebagai mobil, miniatur yang berukuran lebih besar kadang terdeteksi sebagai truk atau bus.

Ketika dilakukan uji coba deteksi kendaraan, beberapa kali terjadi deteksi ganda, sehingga muncul *multiple bounding box* pada satu gambar kendaraan seperti yang terlihat pada Gambar 8.

Tabel 2. Pengukuran FPS

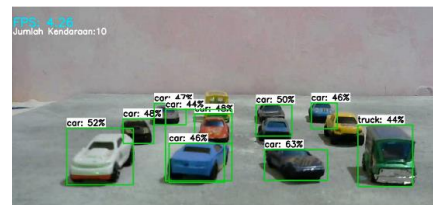
| S | FR | S | FR | S | FR |
|----|-----|----|------|----|-----|
| 1 | 4,1 | 11 | 4,11 | 21 | 5,2 |
| 2 | 4,7 | 12 | 5,06 | 22 | 3,7 |
| 3 | 4,8 | 13 | 5,16 | 23 | 5,0 |
| 4 | 3,9 | 14 | 3,92 | 24 | 5,2 |
| 5 | 4,1 | 15 | 4,70 | 25 | 4,1 |
| 6 | 5,0 | 16 | 3,62 | 26 | 3,9 |
| 7 | 4,0 | 17 | 5,05 | 27 | 5,1 |
| 8 | 4,1 | 18 | 4,16 | 28 | 5,1 |
| 9 | 5,1 | 19 | 5,22 | 29 | 4,4 |
| 10 | 4,0 | 20 | 5,21 | 30 | 4,4 |



Gambar 6. Pengujian Kondisi Terang



Gambar 7. Pengujian Kondisi Redup



Gambar 8. Multiple Bounding Box

Uji coba dengan tinggi peletakan kamera 30 cm dan sudut kamera ± 50° menunjukkan tidak ada kendaraan atau hanya maksimal 1 kendaraan yang

terdeteksi. Hasil deteksi uji coba ini dapat dilihat pada Gambar 9. Ketika uji coba dilakukan dengan jarak antar kendaraan diperkecil menjadi 1 cm, STCS belum mampu mendeteksi dengan baik, terlihat dari beberapa nilai *confidence* nya yang kecil yaitu 30% hingga 16% dan terdapat banyak deteksi ganda seperti yang terlihat pada Gambar 10.



Gambar 10. Jarak Kendaraan 1 Cm

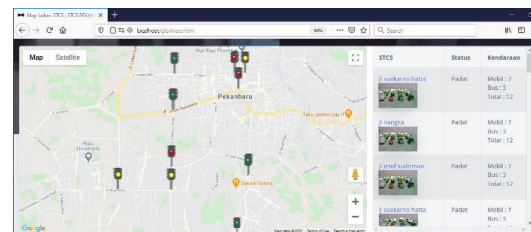
Tabel 3. Hasi Deteksi Sudut Kamera 65°

| Detik | Deteksi Kondisi Terang | Deteksi Kondisi Redup |
|------------------|------------------------|-----------------------|
| 1 | 10 | 10 |
| 2 | 10 | 10 |
| 3 | 7 | 9 |
| 4 | 8 | 10 |
| 5 | 9 | 9 |
| 6 | 9 | 10 |
| 7 | 9 | 9 |
| 8 | 10 | 9 |
| 9 | 9 | 9 |
| 10 | 9 | 8 |
| Rata-rata | 9.0 | 9.3 |

Smart Map menampilkan peta 30 lokasi lampu lalu lintas yang ada di Kota Pekanbaru. *Smart map* telah dapat menangkap data yang dikirimkan oleh STCS melalui HTTP POST, juga telah dapat menampilkan kondisi terkini keadaan lalu lintas pada STCS dan jumlah kendaraan yang terdeteksi. Fitur pengolahan data (input, update, delete, cari) juga disediakan pada *Smart Map*. Tampilan halaman utama *Smart Map* dapat dilihat pada Gambar 11.



Gambar 9. Pengujian Sudut Kamera 50°



Gambar 11. Halaman Utama Smart Map

SIMPULAN

Implementasi *object detection* menggunakan *deep learning* pada Raspberry Pi 4B+ harus mempertimbangkan pemilihan algoritma yang ringan dan optimal. Algoritma Quantized MobileNet-SSD memberikan performa cukup baik yaitu 4,56 fps hampir mendekati rekomendasi minimal APTA yaitu 5 fps untuk area *low-traffic*. Untuk mengatasi keterbatasan unit pemroses Raspberry Pi dapat menggunakan *hardware accelerator* dengan unit pemroses Visual Processing Unit/Graphic Processing Unit/Tensor Processing Unit, sehingga *fps* dapat ditingkatkan. Akurasi deteksi dipengaruhi oleh sudut peletakkan kamera terhadap objek, jarak antar objek. Menurut hasil pengujian, sudut (*tilt*) optimal untuk mendeteksi objek adalah 65°. Pada beberapa hasil deteksi terdapat *multiple bounding box*, yang dapat mempengaruhi akurasi, untuk memperbaikinya dapat digunakan metode *non maximum superession*. Sistem ini tidak dapat mendeteksi dengan baik objek yang jaraknya terlalu dekat terbukti dengan nilai *confidence* yang rendah dan *bounding box* yang tidak akurat (dalam satu

bounding box terdapat lebih dari satu objek). STCS terbukti dapat diintegrasikan dengan sistem informasi geografis (*smart map*) berbasis web. Informasi kepadatan lalu lintas ditampilkan oleh *smart map* dapat digunakan oleh pihak terkait untuk memantau dan mengambil keputusan atas kondisi kepadatan tersebut.

DAFTAR PUSTAKA

- [1] S. Dirks, M. Keeling, and J. Dencik, "How Smart is Your City? Helping Cities Measure Progress," *IBM Glob. Bus. Serv.*, 2009.
- [2] BAPPENAS, "Proyeksi Penduduk Indonesia 2020-2025 (Publikasi bersama oleh BPS, BAPPENAS, dan UNFPA Indonesia)," p. 398, 2008.
- [3] F. R. Harahap, "Dampak Urbanisasi Bagi Perkembangan Kota Di Indonesia," *Society*, vol. 1, no. 1, pp. 35–45, 2013.
- [4] D. Washburn and U. Sindhu, "Helping CIOs Understand 'Smart City' Initiatives," 2010.
- [5] Q. Hidayati, "Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 6, no. 2, 2017.
- [6] S. Saravanan, "Implementation of efficient automatic traffic surveillance using digital image processing," *2014 IEEE Int. Conf. Comput. Intell. Comput. Res. IEEE ICCIC 2014*, pp. 3–6, 2014.
- [7] P. Maheshwari, D. Suneja, P. Singh, and Y. Mutneja, "Smart traffic optimization using image processing," *Proc. 2015 IEEE 3rd Int. Conf. MOOCs, Innov. Technol. Educ. MITE 2015*, pp. 1–4, 2015.
- [8] B. Ghazal, K. Elkhatab, K. Chahine, and M. Kherfan, "Smart traffic light control system," *2016 3rd Int. Conf. Electr. Electron. Comput. Eng. their Appl. EECEA 2016*, no. April, pp. 140–145, 2016.
- [9] M. B. Natafqi, M. Osman, A. S. Haidar, and L. Hamandi, "Smart Traffic Light System Using Machine Learning," *2018 IEEE Int. Multidiscip. Conf. Eng. Technol. IMCET 2018*, pp. 1–6, 2018.
- [10] Y. Nie, "Intelligent traffic lights based on MATLAB," *AIP Conf. Proc.*, vol. 1955, no. April 2018, 2018.
- [11] S. C. Ng and C. P. Kwok, "An intelligent traffic light system using object detection and evolutionary algorithm for alleviating traffic congestion in hong kong," *Int. J. Comput. Intell. Syst.*, vol. 13, no. 1, pp. 802–809, 2020.
- [12] A. N. Rombe, L. F. Aksara, and L. Surimi, "Analisis Perbandingan Real Time Streaming Protocol (Rtsp) Dan Hypertext Transfer Protocol (Http) Pada Layanan Live Video Streaming," *semantIK*, vol. 5, no. 1, pp. 149–156, 2019.
- [13] "Model optimization," *TensorFlow Lite guide*. [Online]. Available:https://www.tensorflow.org/lite/performance/model_optimization. [Accessed: 27-Oct-2020].
- [14] APTA, "Selection of Cameras, Digital Recording Systems, Digital High-Speed Networks and Trainlines for Use in Transit-Related CCTV Systems," *CCTV Stand. Work. Gr.*, no. June, pp. 1–48, 2011.