



## Implementasi algoritma pendeteksi tingkat kepadatan lalu lintas menggunakan metode *background subtraction*

Muhammad Yusuf Fadhlan<sup>1\*</sup>, Usman B. Hanafi<sup>2</sup>, Mohamad Rifki Aulia<sup>3</sup>

<sup>1,2,3</sup>Jurusan Teknik Elektro, Politeknik Negeri Bandung

Jl. Gegerkalong Hilir, Ciwaruga, Kec. Parongpong, Kabupaten Bandung Barat, Indonesia

<sup>1\*</sup>muhammad.yusuf.ttel410@polban.ac.id, <sup>2</sup>usmanb@polban.ac.id, <sup>3</sup>rifkiiuliaa@gmail.com@email.ac.id

### ABSTRAK

Kondisi kepadatan lalu lintas merupakan salah satu faktor yang berpengaruh terhadap kapasitas jalan. Pemanfaatan teknologi dapat dilakukan untuk mempermudah proses pemantauan kepadatan lalu lintas. Beberapa teknologi telah dibuat dan dikembangkan, namun masih memiliki fitur yang terbatas. Dalam penelitian ini, dirancang sebuah purwarupa sistem informasi tingkat kepadatan lalu lintas berdasarkan persentase kapasitas jalan yang dilalui oleh kendaraan. Penelitian ini menggunakan *image processing* dengan metode *background subtraction* dari *library* OpenCV yang dibuat di *software* Pycharm untuk mengolah rekaman video. Metode tersebut digunakan untuk menghasilkan persentase kapasitas jalan berdasarkan perbandingan luas area kendaraan terhadap luas area jalan. Dari hasil pengujian yang dilakukan dapat diketahui bahwa tingkat kepadatan lalu lintas yang tinggi dan jumlah kendaraan yang banyak berpengaruh terhadap kinerja sistem dalam mendeteksi kendaraan yang melintas.

**Kata kunci:** *background subtraction*, kapasitas jalan, OpenCV, Python

### ABSTRACT

*Traffic density conditions are one of the factors that affect road capacity. The use of technology can be made to simplify the traffic density monitoring process. Several technologies have been created and developed, but still have limited features. In this study, a prototype of the traffic density information system was designed based on the percentage of road capacity traversed by vehicles. This study uses image processing with the background subtraction method from the OpenCV library made in Pycharm software to process video recordings. This method is used to produce the percentage of road capacity based on the ratio of the vehicle area to the road area. From the results of the tests carried out, it can be seen that the high level of traffic density and the number of vehicles has a large effect on the performance of the system in detecting passing vehicles.*

**Keywords:** *background subtraction*, road capacity, OpenCV, Python

## 1. PENDAHULUAN

Lalu lintas jalan merupakan sebuah prasarana yang digunakan untuk gerak pindah pengguna jalan untuk kendaraan roda dua atau empat. Kelancaran gerak pindah kendaraan menjadi salah satu tujuan dari pemerintah untuk mewujudkan lalu lintas yang teratur. Lalu lintas sering kali mengalami hambatan, salah satunya adalah kemacetan. Kepadatan lalu lintas merupakan salah satu permasalahan yang sering menyebabkan kemacetan bagi pengguna jalan. Kemacetan dapat dihindari dengan memberlakukan beberapa peraturan, melakukan perluasan jalan, dan melakukan pemantauan. Pemantauan kepadatan lalu lintas diperlukan untuk mengetahui situasi dan kondisi kendaraan yang melintas agar dapat mengurangi kemacetan. Pada umumnya pemantauan dilakukan secara manual. Namun, dengan pengembangan teknologi, pemantauan lalu lintas dapat dilakukan dari jarak jauh.

Telah banyak solusi yang diciptakan untuk mengatasi permasalahan yang sering dihadapi oleh pengguna jalan untuk mengurangi kepadatan lalu lintas di jalan. Solusinya yaitu dengan meningkatkan kapasitas, memperluas lebar jalan atau menambah lajur lalu lintas, pembatasan kendaraan pribadi, baik menggunakan sistem ganjil genap maupun sistem 3 in 1, dan mengembangkan *intelligent transportation system* (ITS) atau sistem transportasi pintar yang saat ini sedang populer dikembangkan oleh beberapa peneliti. Metode deteksi jenis kendaraan di jalan menggunakan OpenCV telah dikembangkan pada penelitian [1]. Program ini dapat menghitung jumlah dan mengidentifikasi jenis

kendaraan yang melintas berdasarkan jenis kendaraan yang terdeteksi melalui video yang dimasukkan ke dalam program. Video hasil rekaman lalu lintas dapat dimasukkan oleh pengguna ke dalam program yang telah dibuat. Perancangan aplikasi sistem pemantauan kondisi lalu lintas berbasis Android juga telah dikembangkan [2]. Aplikasi ini dapat digunakan untuk melakukan pemantauan dan memberikan informasi berupa informasi tingkat kepadatan lalu lintas kendaraan dan cuaca secara langsung berbasis Android. Aplikasi penghitung kendaraan dengan deteksi kendaraan menggunakan berbasis Android Cvblob juga telah dibuat sebelumnya [3]. Cara kerja dari alat ini yaitu dengan cara menghitung seluruh kendaraan yang melintas di jalan raya yang sudah dipasang kamera yang terintegrasi dengan Android yang disinkronisasikan dengan Google Map. Raspberry pi juga telah dikembangkan untuk membuat sistem informasi kepadatan lalu lintas [4]-[6]. Penelitian [4] menggunakan *library computer vision* dengan metode *background subtraction*. Informasi kepadatan lalu lintas kendaraan didapatkan dari video yang telah diolah dengan menggunakan metode *contour feature*. Pada penelitian [5] menggunakan pengolahan citra digital dengan fitur pengenalan jenis, kecepatan, dan jumlah kendaraan yang melintas. Dalam tahapan pembuatan program, sistem ini dibuat dengan menggunakan metode *background subtraction*. Penelitian [6] menggunakan metode *haar cascade classifier*. Setelah itu, kendaraan diklasifikasikan menjadi tiga jenis kendaraan dengan menggunakan metode *multilayer perceptron*. Kemudian, penelitian terkait ITS berhasil dikembangkan untuk mengatasi solusi lalu lintas [7]. Penelitian ini membuat purwarupa dari penerapan teknologi ITS dalam pencacatan jumlah, pengklasifikasian, dan penghitungan rata-rata kecepatan setiap jenis kendaraan. *Library Cvblob* juga telah digunakan dalam perancangan penghitung jumlah kendaraan [8]. Penelitian ini digunakan untuk memantau dan menghitung jumlah kendaraan yang melintas di dua jalur berbeda dengan menggunakan dua video sebagai *input* dengan kepadatan lalu lintas dari jalur yang berbeda. Sensor ultrasonik berbasis Arduino Uno juga telah dimanfaatkan untuk sistem ITS [9]. Pada penelitian ini, sensor ultrasonik SRF05 digunakan untuk menghitung kecepatan dan jumlah kendaraan yang melintas. Terakhir, purwarupa sistem klasifikasi dan penghitung jumlah kendaraan bermotor menggunakan kamera *webcam* berbasis citra digital telah dikembangkan oleh [10]. Metode yang digunakan yaitu deteksi tepi dan kontur, *background subtractor*, *line intersect counting*, dan *size thresholding*.

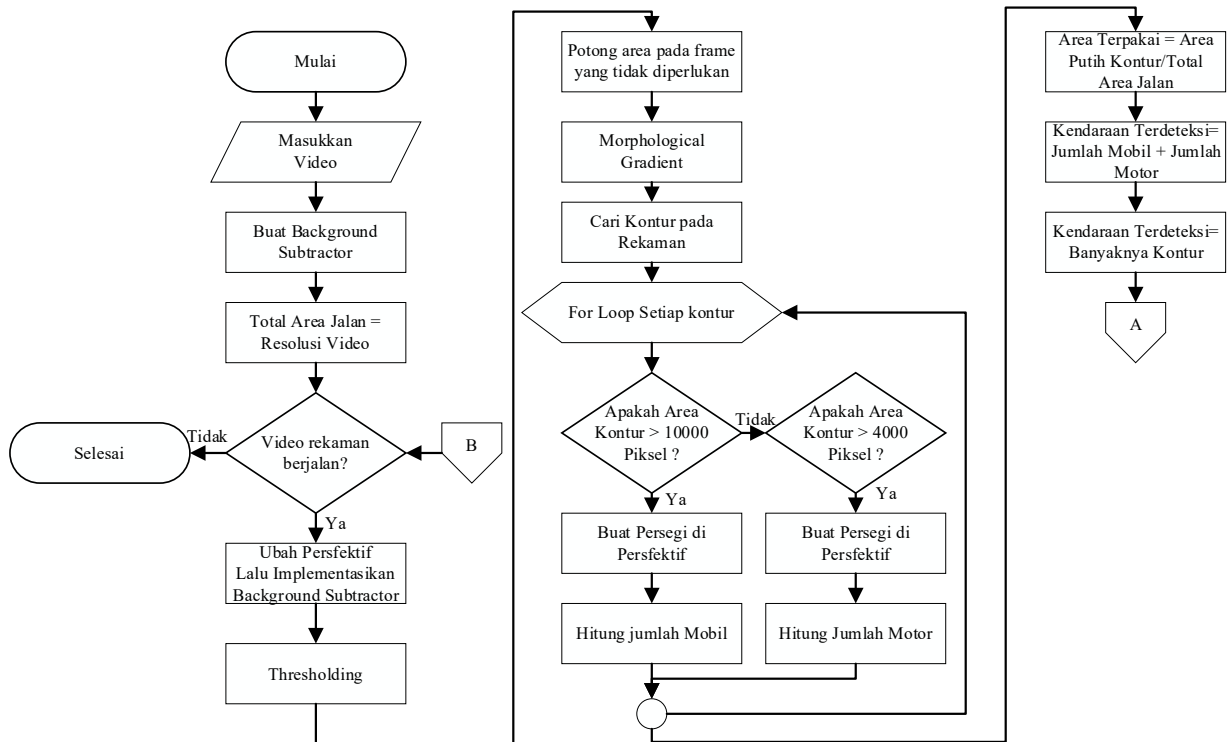
Tujuan penelitian ini adalah untuk merancang purwarupa sistem informasi pendeteksian kepadatan jalan raya menggunakan metode *image processing*. Pengembangan dari metode-metode sebelumnya yang dilakukan pada penelitian ini antara lain adalah dengan membuat klasifikasi dari kepadatan kendaraan yang melintas. Hasil klasifikasi tersebut ditampilkan secara *real-time* melalui *website*. Metode *image processing* yang digunakan yaitu *background subtraction*. Metode tersebut digunakan untuk menghasilkan persentase kapasitas jalan berdasarkan perbandingan luas area kendaraan terhadap luas area jalan. Dengan demikian dapat diketahui tingkat kepadatan lalu lintas pada berbagai kondisi jalan.

## 2. METODE PENELITIAN

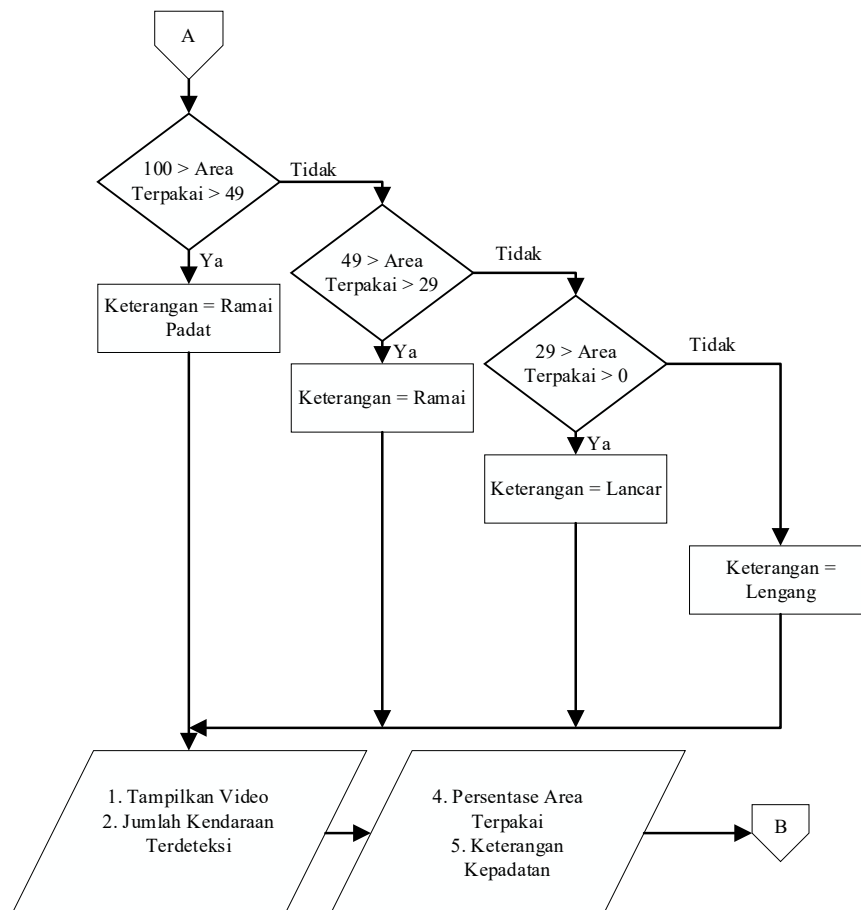
### 2.1 Perancangan Algoritma *Background Substraction*

Pada penelitian ini algoritma perhitungan kendaraan akan dirancang menggunakan *image processing* dengan metode *background subtraction*. Metode ini digunakan untuk memisahkan *foreground* yang bergerak dari latar belakang yang statis agar dapat mendeteksi kepadatan kendaraan di jalan raya. Hasil dari deteksi tersebut akan dikonversikan ke dalam persentase dari luas area yang dipakai kendaraan terhadap luas dari jalan raya. Persentase tersebut kemudian akan diklasifikasikan menjadi beberapa kelompok jenis kepadatan jalan raya seperti ramai padat, ramai, lancer, dan lengang. Kemudian hasil tersebut akan ditampilkan lewat internet secara *real time*.

Algoritma yang digunakan merupakan algoritma yang telah dibuat dan terbentuk menjadi sebuah *library* bernama OpenCV yang dapat digunakan secara umum. Pengguna dapat menggunakan *library* tersebut untuk mengembangkan program yang diinginkan. Adapun diagram alir pendeteksian kepadatan lalu lintas pada penelitian ini digambarkan pada Gambar 1, sedangkan diagram alir yang menampilkan hasil keluaran digambarkan pada Gambar 2.



Gambar 1. Diagram alir pendeteksian



Gambar 2. Diagram alir menampilkan hasil keluaran

Terdapat beberapa proses yang dilakukan untuk mendapatkan parameter-parameter yang diinginkan. Proses tersebut merupakan bagian dari tahapan yang digunakan untuk menunjang pendeteksian objek suatu citra. Proses tersebut dapat dilihat pada Gambar 1. Gambar tersebut menunjukkan diagram alir yang digunakan untuk pendeteksian objek. Citra didapatkan dari video sebagai masukan, lalu objek yang bergerak dipisahkan dari latar belakang yang statis menggunakan *background subtraction*. Selama rekaman diputar, perspektif video diubah dan *background subtraction* diimplementasikan. Warna citra diubah menjadi hitam dan putih menggunakan metode *thresholding*, dan *noise* yang tidak diinginkan berupa titik-titik kecil dikurangi dengan menggunakan metode morfologi agar objek yang terdeteksi hanya kendaraan yang melintas di jalan. Untuk mengidentifikasi objek yang terdeteksi dibutuhkan kontur. Apabila area kontur yang terdeteksi lebih dari 10.000 piksel, maka diidentifikasi sebagai mobil dengan bentuk persegi. Apabila kontur yang terdeteksi lebih dari 4.000 piksel, maka diidentifikasi sebagai motor.

Parameter-parameter yang telah didapatkan kemudian dirumuskan. Kondisi kepadatan kendaraan dipengaruhi oleh besar nilai dari area yang terpakai. Semua parameter meliputi video yang telah diolah, total area jalan yang didapatkan dari resolusi video, jumlah kendaraan terdeteksi yang didapatkan dari kontur, persentase kapasitas jalan yang terpakai, serta keterangan kepadatan kendaraan yang terdeteksi lalu ditampilkan pada *user interface* (UI) yang akan digunakan. UI yang digunakan dapat dilihat melalui *website* maupun dari *integrated development environmental* (IDE) dari Pycharm sendiri.

## 2.2 Implementasi Perangkat Lunak

Realisasi dilakukan berdasarkan diagram alir yang telah dibuat. Untuk merealisasikan sistem pada penelitian ini dibutuhkan alat yang mendukung *software* untuk mengolah citra dan *library* penunjang. Program dibuat dengan menggunakan bahasa pemrograman Python 3 yang diakses melalui *software* IDE Pycharm untuk sistem operasi Windows. Inisialisasi masing-masing variabel dibutuhkan untuk mempersingkat kode program. Variabel tersebut digunakan sebagai parameter-parameter yang dibutuhkan untuk membentuk suatu objek didalam *library* OpenCV dalam struktur bahasa pemrograman Python 3. Perspektif video diubah berdasarkan titik-titik yang telah ditentukan. Lalu resolusi dari video perspektif diubah sesuai dengan resolusi video asli.

Transformasi citra dilakukan dengan melalui beberapa tahapan metode. Proses pertama adalah menentukan variabel *fgmask* untuk memisahkan objek dari latar belakang agar menjadi hitam putih. Proses selanjutnya yaitu proses mengurangi *noise* dengan menggunakan morfologi. Selanjutnya kontur dicari berdasarkan variabel *fgmask*. Tahapan berikutnya adalah pendeteksian objek. Objek yang bergerak diidentifikasi dengan memberikan *shape* dengan bentuk persegi pada kontur yang terdeteksi. Persegi tersebut dibutuhkan untuk mencari area dari kendaraan. Selanjutnya adalah proses mendeteksi jenis kendaraan. Jenis kendaraan dibagi menjadi dua bagian. Area yang lebih dari 10.000 piksel diidentifikasi sebagai mobil, sedangkan area yang kurang dari 10.000 dan lebih dari 4000 piksel akan diidentifikasi sebagai motor. Kendaraan yang terdeteksi oleh kamera ditandai dengan persegi berwarna hijau saat melintas di jalan. Bentuk persegi yang ditampilkan akan menyesuaikan dengan luas objek yang terdeteksi. Adapun *pseudocode* untuk sistem ini ditampilkan pada Gambar 3.

```

if      100 > areaterpakai > 49:
    keterangan = "ramai padat"
    warna text = kuning

else if 49 > areaterpakai > 29:
    keterangan = "ramai"
    warna text = biru

else if 29 > areaterpakai > 1:
    keterangan = "lancar"
    warna text = biru muda

else   keterangan = "lengang"
      warna text = putih

```

Gambar 3. *Pseudocode* program kondisi kepadatan

Kepadatan kendaraan kemudian diklasifikasikan menjadi beberapa bagian. Pengklasifikasian kepadatan dipengaruhi oleh total area yang dilalui oleh kendaraan terhadap jalan. Keterangan kepadatan kendaraan akan menampilkan warna yang berbeda sesuai dengan kondisi. Program akan menampilkan parameter-parameter yang telah didapatkan. Informasi yang ditampilkan kemudian ditempatkan pada koordinat *frame* video tertentu. Keterangan beserta warna dari informasi kepadatan kendaraan akan disesuaikan dengan kondisi kepadatan yang sedang terjadi.

Program yang telah dibuat dengan menggunakan bahasa pemrograman Python dihubungkan ke *website* dengan menggunakan *library* Flask. Untuk menampilkan video hasil olahan di *website* tersebut, dibutuhkan *file* .html agar dapat dilihat di *browser*. Berkas program Python, .html, .css, dan video hasil rekaman kemudian di *upload* ke *website* penyedia *hosting* yang mendukung pemrograman Flask. Nama berkas yang akan ditampilkan serta *application root* dari program yang telah dibuat diaplikasikan ke dalam setelan *website* yang akan di *hosting*. Setelah setelan disimpan maka *virtual environment* akan dibuat secara otomatis oleh sistem.

Untuk menjalankan program yang akan ditampilkan pada *website* diperlukan akses *secure shell* (SSH). *Virtual environment* yang telah dibuat kemudian diaktifkan untuk menjalankan *library* yang telah dipasang di *server*. Nama berkas program Python yang akan ditampilkan kemudian dijalankan. *Website* akan menampilkan program yang telah disimpan pada *server* sesaat setelah *website* dengan IP *address default server* diakses melalui *browser*.

### 3. HASIL DAN PEMBAHASAN

#### 3.1 Skenario Pengujian

Setelah merealisasikan program, proses selanjutnya adalah melakukan pengujian. Pengujian ini dilakukan untuk mengetahui hasil dan keandalan sistem penelitian ini. Pengujian dilakukan dengan beberapa parameter sebagai tolak ukur keandalan sistem dalam mendeteksi kapasitas jalan. Terdapat dua parameter yang diuji dalam penelitian ini. Parameter tersebut meliputi jumlah kendaraan yang melintas berdasarkan jenis dan informasi kepadatan kendaraan berdasarkan perbandingan antara persentase area kendaraan terhadap area jalan.

Pengujian dilakukan di jembatan penyebrangan orang yang terdapat di jalan Dr. Djujungan Bandung pada siang hari. Lokasi pengujian dipilih karena memenuhi kriteria agar dapat memudahkan proses pengerjaan proyek penelitian. Lokasi pelaksanaan tidak terlalu jauh dengan pusat kota dan destinasi wisata Kota Bandung. Pengujian dilakukan dengan merekam lalu lintas kendaraan di jembatan penyebrangan orang setinggi 5 meter. Video diambil menggunakan telepon seluler dengan resolusi 1080 x 720 piksel dan sudut pengambilan gambar yang direkam adalah 60° dari sisi horizontal. Pada tahap ini, video disimpan dalam format MPEG-4 (\*.mp4).

Pelaksanaan pengujian dilakukan dengan menggunakan video rekaman dari tempat pelaksanaan. Ukuran dari video rekaman kemudian diubah ulang ke resolusi 360 piksel agar proses pengolahan yang dilakukan oleh *server* berjalan lancar. Video kemudian diolah dengan menggunakan bahasa pemrograman Python dengan memanfaatkan *library* OpenCV. Pada tahapan pengolahan video, hasil rekaman lalu lintas dimasukkan ke dalam program. Program kemudian disimpan kedalam bentuk format Python (\*.py). Berkas video dan program tersebut kemudian diunggah ke dalam *server* penyedia *website*. Program akan menampilkan hasil video yang telah diolah setelah pengguna mengakses *website* yang telah di-*hosting*. Pada tahap ini pengguna dapat mengetahui hasil uji dari fungsi utama sistem yaitu melakukan pemantauan kepadatan lalu lintas berdasarkan hasil rekaman.

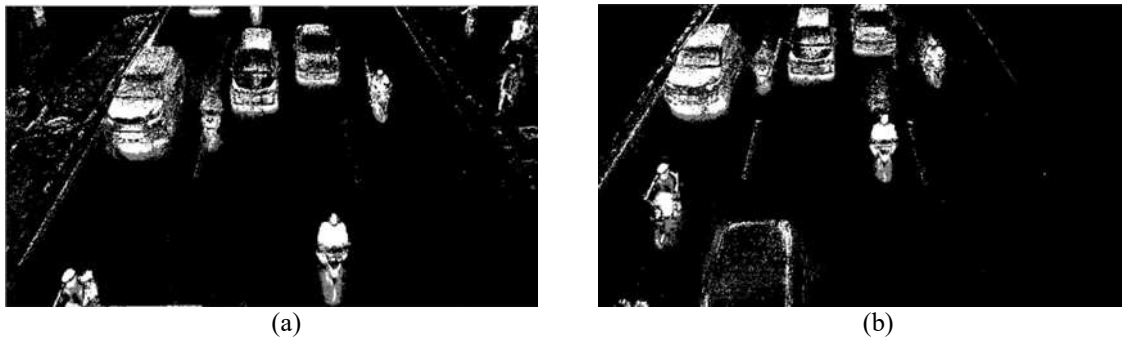
#### 3.2 Hasil Pengujian Pendeteksian Kendaraan

Gambar 4 merupakan tampilan dari hasil pengujian program yang telah dijalankan. Terdapat dua jenis kendaraan dengan total tujuh kendaraan yang terdeteksi oleh program. Gambar pengujian tersebut merupakan gambar yang diambil pada siang hari dari atas jembatan penyebrangan orang di Jl. Dr. Djujungan, Bandung. Gambar tersebut adalah tampilan dari hasil deteksi kendaraan dengan citra yang diambil dari kamera telepon seluler. Kamera menghasilkan gambar yang jelas sehingga citra dapat diproses oleh program. Berdasarkan pengamatan pada gambar, dapat dilihat bahwa pendeteksian jumlah kendaraan yang terbaca oleh program menghasilkan keluaran yang bagus. Program dapat mendeteksi kendaraan yang melintas. Hasil ini menunjukkan sistem 100% berhasil dalam mendeteksi kendaraan yang ditandai dengan kotak hijau.



Gambar 4. Tampilan pendeteksian kendaraan

Untuk mendapatkan hasil deteksi kendaraan, diperlukan metode untuk memisahkan objek yang bergerak dengan *background* yang statis. Metode tersebut dapat dilihat seperti pada Gambar 5. Gambar tersebut adalah hasil dari metode *background subtractor* dengan mengubah warna asli ke gambar hitam putih. Metode tersebut dilakukan untuk memisahkan citra antara *foreground* berwarna putih yang dianggap sebagai objek bergerak dengan *background* yang berwarna hitam. Untuk mengurangi pendeteksian objek dari area yang tidak diinginkan, gambar harus diseleksi seperti pada gambar sehingga pendeteksian objek yang bergerak terfokus pada area tertentu.



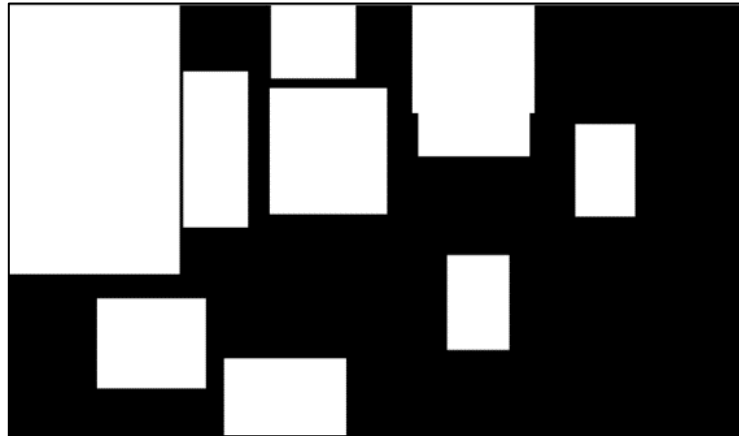
Gambar 5. Pendeteksian objek yang bergerak: (a) area sebelum *crop*, (b) area setelah *crop*

Deteksi kontur diperlukan untuk mendapatkan jumlah objek yang terdeteksi. Kontur tersebut digunakan untuk membentuk bidang persegi berwarna hijau yang menandakan ada objek yang terdeteksi. Bidang persegi tersebut juga dapat digunakan untuk mendapatkan area dari kendaraan yang terdeteksi. Area kendaraan didapatkan setelah perspektif video diubah seperti pada Gambar 6.



Gambar 6. Tampilan perubahan perspektif

Gambar 6 merupakan tampilan gambar dengan perspektif yang telah diubah. Perubahan tersebut bertujuan agar dapat mengetahui total area dari jalan. Resolusi video yang digunakan berukuran 640 x 360 piksel, sehingga hasil perkalian dari resolusi video tersebut adalah 230.400 piksel. Apabila tampilan video mencakup seluruh area jalan, maka pada dasarnya luas area jalan tersebut berukuran sama dengan resolusi dari video asli. Setelah perspektif video diubah, luas area objek didapatkan dengan mengubah bentuk area kontur yang terdeteksi kedalam bentuk persegi. Hal ini dilakukan agar dimensi yang terbentuk lebih mudah untuk dimasukkan ke dalam perhitungan.

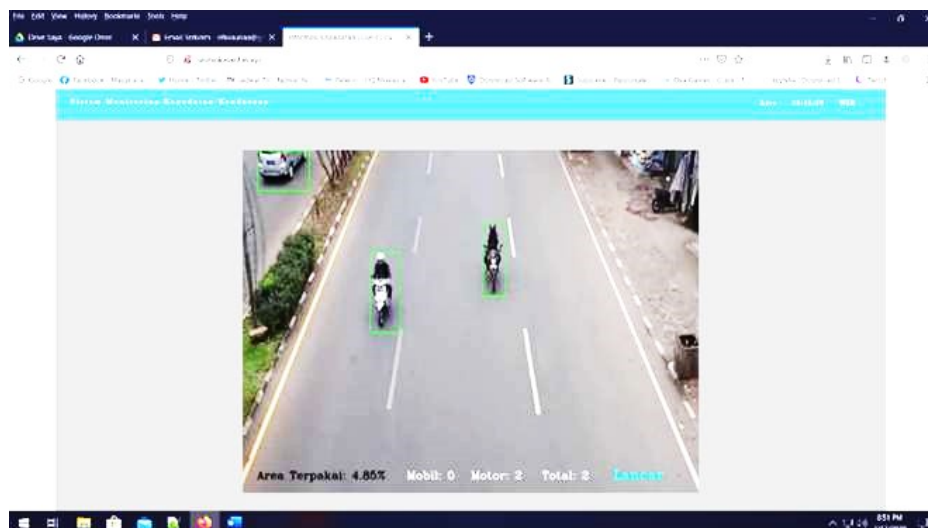


Gambar. 7 Area kontur yang terdeteksi

Kontur yang terdeteksi membentuk sebuah dimensi yang berbentuk persegi berwarna putih dengan latar belakang hitam seperti pada Gambar 7. Luas area dari masing-masing kontur yang terdeteksi dapat terbaca, sehingga parameter luas area dari kendaraan yang melintas dapat dibandingkan dengan luas area jalan. Perbandingan luas area tersebut dapat digunakan sebagai tolak ukur kepadatan lalu lintas kendaraan di jalan. Selain itu, jenis kendaraan dapat dibedakan menjadi kendaraan roda dua maupun roda empat dengan membedakan luas area kontur berdasarkan piksel. Piksel kendaraan roda empat lebih besar daripada piksel kendaraan roda dua, maka dari itu jenis kendaraan dapat diklasifikasikan.

### 3.3 Pengujian Sistem Antarmuka

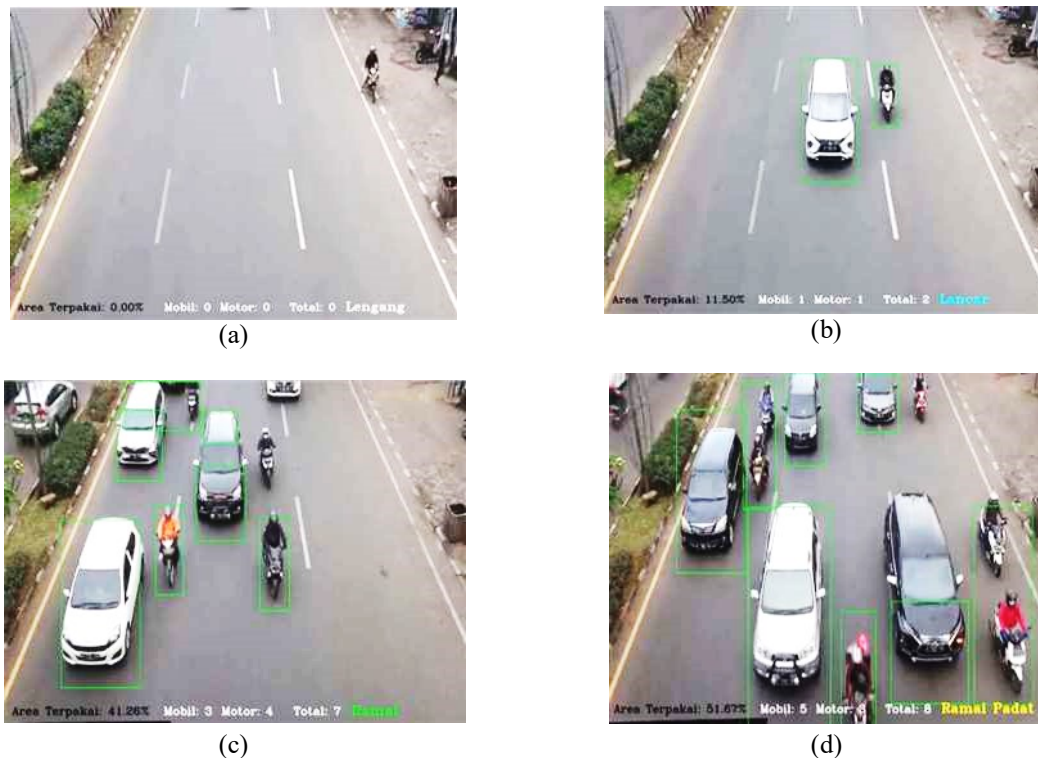
Program yang telah dibuat selanjutnya di-*hosting* ke dalam *server* penyedia *website* yang mendukung program Flask. Pada bagian ini *server* diuji dengan cara mengakses *website* yang telah di-*hosting*. Hasil pengujian berupa tampilan *website* ditunjukkan pada Gambar 8.



Gambar 8. Tampilan *Website* di Komputer



Pengujian pertama *website* diakses melalui internet dengan menggunakan *browser* yang terdapat pada komputer personal. Dari Gambar 8 dapat ditunjukkan bahwa video yang telah diolah dapat ditampilkan. Pendeteksi jumlah kendaraan dibagi kedalam dua jenis, yaitu motor dan mobil. Kedua jenis kendaraan lalu dijumlahkan untuk mendapatkan parameter total kendaraan. Deteksi jumlah dan area kendaraan dilihat dari keterangan informasi yang ditampilkan pada Gambar 9.



**Gambar 9. Kondisi Kepadatan Kendaraan: (a) kondisi lengang, (b) kondisi lancar, (c) kondisi ramai, (d) kondisi padat**

Pada Gambar 9 diperlihatkan kumpulan hasil deteksi jumlah kendaraan beserta keterangan kepadatan lalu lintas. Gambar 9 (a) memperlihatkan beberapa kondisi kepadatan lalu lintas kondisi lengang dengan hasil deteksi area terpakai 0%, mobil 0, dan motor 0. Gambar 9 (b) memperlihatkan beberapa kondisi kepadatan lalu lintas kondisi lancar dengan hasil deteksi area terpakai 11,5%, mobil 1, dan motor 1. Gambar 9 (c) memperlihatkan beberapa kondisi kepadatan lalu lintas kondisi ramai dengan hasil deteksi area terpakai 41,26%, mobil 3, dan motor 4. Gambar 9 (d) memperlihatkan beberapa kondisi kepadatan lalu lintas kondisi padat dengan hasil deteksi area terpakai 51,67%, mobil 5, dan motor 3. Keterangan kepadatan lalu lintas berubah sesuai dengan persentase area yang terpakai oleh kendaraan. Dari hasil pengujian yang telah dilakukan dapat diketahui bahwa semakin banyak dan semakin rapat objek yang terdeteksi, maka kinerja sistem dalam mendeteksi jumlah kendaraan yang melintas semakin rendah. Hal ini karena semakin banyak objek yang dideteksi maka diperlukan kemampuan komputasi yang lebih handal.

#### 4. KESIMPULAN

Algoritma pendeteksi kendaraan menggunakan *background subtraction* telah berhasil diimplementasikan pada penelitian ini. Berdasarkan hasil pengujian sistem yang telah dilakukan, dapat disimpulkan bahwa sistem dapat mendeteksi kendaraan yang melintas pada jalan satu arah dengan menggunakan *library* OpenCV. Sistem yang telah dibuat dapat menampilkan jumlah kendaraan berdasarkan jenis kendaraan yang terdeteksi dengan mengidentifikasi luas area piksel kendaraan yang terbaca. Persentase kapasitas jalan dapat dibuat berdasarkan perbandingan luas area kendaraan dengan luas area jalan. Sistem dapat menampilkan keterangan kepadatan kendaraan berdasarkan kapasitas jalan yang terpakai oleh kendaraan. Penelitian selanjutnya dapat dikembangkan algoritma *image processing* lain untuk meningkatkan akurasi pendeteksian.



## UCAPAN TERIMA KASIH

Ucapan terima kasih disampaikan kepada Pusat Penelitian dan Pengabdian kepada Masyarakat Politeknik Negeri Bandung (P3M) Politeknik Negeri Bandung.

## REFERENSI

- [1] A. Lazaro, J. Buliali, and B. Amaliah, "Deteksi Jenis Kendaraan di jalan Menggunakan OPENCV," *Jurnal Teknik ITS*, vol. 6, no. 2, pp. 293-299, 2017.
- [2] A. Anggara and Y. Handoko, "Perancangan Aplikasi Sistem Monitoring Kondisi Lalu Lintas Berbasis Android Pada Sikomolintas," *Tugas Akhir, Dept. Sistem Komputer, Universitas Komputer Indonesia, Bandung*, 2017.
- [3] D. Ramadhani, "Aplikasi Penghitung Kendaraan Dengan Deteksi Kendaraan Menggunakan Cvblob Dan Penampil Google Map Berbasis Android," *Tugas Akhir, Dept. Teknik Elektro, Universitas Telkom, Bandung*, 2014.
- [4] F. Rahmadina and Z. Zaini, "Sistem Informasi kepadatan Lalu Lintas berbasis Raspberry Pi PC Board", *Jurnal Nasional Teknik Elektro*, vol. 5, no. 1, 2016.
- [5] G. E. Setyawan, B. Adiwijaya, and H. Fitriah, "Sistem Deteksi Jumlah, Jenis Dan Kecepatan Kendaraan Menggunakan Analisa Blob Berbasis Raspberry Pi," *Jurnal Teknologi Informasi dan Ilmu Komputer*, vol. 6, no. 2, pp. 211-218, 2019.
- [6] M. Irfan, B. A. A. Sumbodo, and I. Candradewi, "Sistem Klasifikasi Kendaraan Berbasis Pengolahan Citra Digital dengan Metode Multilayer Perceptron," *Indonesian Journal of Electronics and Instrumentation System (IJEIS)*, vol. 7, no. 2, pp. 139-148, 2017.
- [7] M. N. F. Mastika, "Sistem Pendeteksi Kemacetan Berdasarkan Jenis Dan Jumlah Kendaraan Berbasis Intelligent Transportation System," *Skripsi, Dept. Ilmu Komputer, Institut Pertanian Bogor, Bogor*, 2018.
- [8] M. Hafizulahudin, "Perancangan Program Penghitung Jumlah Kendaraan Di Lintasan Jalan Dua Arah Menggunakan Library Cvblob," *Skripsi, Dept. Teknik Elektro, Universitas Lampung, Lampung*, 2018.
- [9] R. Resilawati and W. Rizky, "Penghitung Jumlah Kendaraan Dan Pengukur Kemacetan Menggunakan Sensor Ultrasonik Berbasis Arduino Uno," *Skripsi, Dept. Teknik Elektro, Universitas Gadjah Mada, Yogyakarta*, 2015.
- [10] W. R. Kurniawan, "Purwarupa Sistem Klasifikasi Dan Penghitung Jumlah Kendaraan Bermotor Menggunakan Kamera Webcam Berbasis Citra Digital," *Skripsi, Dept. Teknik Elektro, Universitas Gadjah Mada, Yogyakarta*, 2015.

