

# Estimasi Panjang Antrean Kendaraan pada Persimpangan Jalan Raya dengan Sensor Kamera Menggunakan Metode *Queue Length Estimation*

Riandini<sup>1</sup>, Dwi Kuncoro,<sup>2</sup>

<sup>1</sup>Dept. Teknik Elektro, Politeknik Negeri Jakarta, Depok, Indonesia

<sup>2</sup>Dept. Teknik Komputer Institut Teknologi Sepuluh Nopember, Surabaya 60111, Indonesia

Email: riandini@elektro.pnj.ac.id, irfananda2212@gmail.com

## Abstrak

Kemacetan sering terjadi di kota besar, terutama di persimpangan yang dekat dengan pusat keramaian. Ruas jalan yang satu memiliki kepadatan yang berbeda dengan ruas lain, sehingga dibutuhkan adanya sistem lampu lalu-lintas cerdas yang bisa beradaptasi terhadap perbedaan kepadatan di tiap ruas pada suatu persimpangan. Pada penelitian ini, dibangun sebuah sistem untuk melakukan estimasi panjang antrean di persimpangan, kemudian data tersebut dapat digunakan untuk membangun sistem lalu-lintas cerdas yang adaptif terhadap kondisi kepadatan di tiap ruas jalan pada persimpangan tersebut.

**Kata kunci** : kemacetan, lampu lalu-lintas adaptif, panjang antrean

Diterima Redaksi: 20-11-2022, Selesai Revisi: 09-12-2022, Diterbitkan Online: 27-02-2023

DOI: <https://doi.org/10.59378/jcenim.v1i1.4>

## I. PENDAHULUAN

Kemacetan lalu-lintas merupakan salah satu masalah yang banyak terjadi di kota-kota besar, khususnya di persimpangan yang dekat dengan pusat-pusat keramaian seperti kawasan industri, perkantoran, sekolah, dan sebagainya. Adanya pusat-pusat keramaian tersebut menyebabkan sebagian ruas jalan terisi oleh kendaraan dalam jumlah yang sangat banyak, sedangkan ruas jalan yang lain hanya terisi oleh kendaraan dalam jumlah yang sedikit. Kepadatan yang berbeda tersebut memerlukan penanganan khusus agar lalu-lintas di semua ruas dapat berjalan dengan lancar dan seimbang. Sementara itu, sistem lampu lalu-lintas yang ada di Indonesia masih menggunakan sistem pewaktuan yang tidak adaptif terhadap tingkat kepadatan lalu-lintas di tiap ruas jalan. Akibatnya, terjadi ketidakseimbangan antrean kendaraan pada persimpangan tersebut. Lalu-lintas di sebagian ruas berjalan lancar, sedangkan di ruas lainnya terjadi penumpukan dan antrean yang panjang karena lamanya lampu hijau menyala tidak sebanding dengan banyaknya kendaraan pada ruas tersebut.

Permasalahan tersebut dapat dikurangi, misalnya dengan menerapkan sistem Smart Traffic Light. Smart Traffic Light adalah sistem lampu lalu-lintas cerdas yang dapat mengatur pewaktuan pada lampu lalu-lintas secara adaptif berdasarkan tingkat kepadatan lalu-lintas pada masing-masing ruas jalan. Dengan sistem ini, ruas yang memiliki tingkat kepadatan lebih tinggi akan diberi durasi yang lebih lama untuk menyalakan lampu hijau, dibandingkan ruas lain yang tingkat kepadatannya lebih rendah. Dengan demikian, akan terjadi keseimbangan dan kelancaran lalu-lintas pada semua ruas jalan di persimpangan tersebut.

## II. TINJAUAN PUSTAKA

Untuk mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini akan menjadi lebih terarah dan tepat sasaran.



### C. OpenCV

OpenCV (Open-source Computer Vision) merupakan merupakan suatu pustaka (library) yang bersifat open source, dan banyak digunakan di bidang computer vision karena dapat digunakan untuk melakukan pengolahan citra, video, dan sebagainya[4].

Pustaka ini ditulis dalam bahasa C dan C++, dan berjalan di sistem operasi Linux, Windows, Mac OS, iOS, dan Android. Antarmukanya tersedia untuk Python, Java, Ruby, Matlab, dan bahasa-bahasa pemrograman lainnya.

OpenCV didesain untuk efisiensi komputasional dengan fokus kuat pada aplikasi real-time: optimisasi dibuat pada semua level, dari algoritma hingga instruksi-instruksi multicore dan CPU.

Salah satu tujuan OpenCV adalah untuk menyediakan infrastruktur computer vision yang sederhana untuk digunakan, yang membantu orang membangun aplikasi computer vision dengan cepat. Pustaka OpenCV memuat lebih dari 500 fungsi yang menjangkau banyak area, seperti inspeksi produk pabrik, medical imaging, keamanan, user interface, kalibrasi kamera, stereo vision, dan robotika[5].

## III. PERANCANGAN SISTEM

Penelitian ini bertujuan untuk membuat sistem yang dapat memproses masukan berupa video lalu-lintas di persimpangan jalan raya, melakukan image processing, mengidentifikasi objek berupa kendaraan, dan melakukan estimasi panjang antrean kendaraan di suatu ruas jalan di persimpangan tersebut. Hasil dari estimasi tersebut dapat digunakan untuk menentukan durasi yang akan diberikan kepada lampu lalu-lintas di tiap ruas sehingga menghasilkan sistem lampu lalu-lintas yang adaptif terhadap kondisi kepadatan jalan. Desain sistem secara keseluruhan dan alur kerja sistem dijelaskan pada gambar berikut.

### A. *Background Reconstruction*

Background reconstruction atau rekonstruksi citra latar merupakan proses untuk mendapatkan citra latar (background image) dari sebuah video. Citra latar tersebut biasanya digunakan pada proses lain, yaitu Background Subtraction. Salah satu cara untuk melakukan proses rekonstruksi citra latar adalah dengan menggunakan bantuan fungsi pada pustaka OpenCV yang bernama `accumulateWeighted`. Fungsi tersebut berguna untuk menghitung jumlah bobot piksel pada citra masukan kemudian dilakukan rata-rata berjalan (running average) pada citra tersebut. Sebelum dilakukan proses tersebut, citra masukan dikonversi terlebih dahulu ke dalam bentuk citra abu-abu (grayscale image).

### B. *Background Subtraction*

Background subtraction merupakan proses yang bertujuan untuk mendapatkan foreground image dari suatu frame atau citra. Proses ini dilakukan dengan cara menghitung nilai perbedaan absolut antara citra masukan dan citra latar yang didapatkan dari proses sebelumnya, yaitu background reconstruction. Fungsi pada pustaka OpenCV yang digunakan pada proses ini adalah fungsi `absdiff`. Setelah didapatkan foreground image, nantinya citra tersebut akan diproses pada tahap selanjutnya yaitu segmentasi citra (image segmentation).

### C. Segmentasi Citra

Segmentasi citra (image segmentation) meliputi 3 proses yaitu binary treshold, erosi, dan dilasi. Binary treshold adalah proses pengolahan citra yang bertujuan untuk mendapatkan citra biner dari sebuah citra abu-abu dengan menggunakan suatu nilai treshold tertentu. Apabila nilai intensitas pada sebuah piksel lebih kecil daripada nilai treshold yang ditentukan maka akan diubah menjadi 0 (nol) atau warna hitam. Sebaliknya, apabila nilai intensitasnya lebih besar daripada nilai treshold, maka akan diubah menjadi 255 atau warna putih.

Erosi merupakan suatu operasi morfologis (morphological operation) pada citra digital untuk menghilangkan atau mengurangi adanya noise pada citra, yang berupa titik-titik putih dalam jumlah kecil. Dengan dihilangkannya noise tersebut maka proses deteksi objek pada citra akan lebih akurat. Proses ini dilakukan dengan menggunakan suatu matriks kernel (disebut juga structuring element) berukuran ganjil, misalnya 1x1, 3x3, dan sebagainya sesuai kebutuhan.

Dilasi merupakan proses yang bertujuan untuk mempertebal titik-titik putih pada objek yang akan dideteksi sehingga objek tersebut lebih mudah untuk dideteksi oleh komputer. Seperti halnya erosi, proses ini juga menggunakan suatu structuring element dengan ukuran tertentu sesuai kebutuhan. Kombinasi

Tabel 1: Spesifikasi Komputer

Komponen	Spesifikasi
Sistem Operasi	Manjaro Linux 64 bit
Prosesor	Intel Core i5 CPU @ 2,50 GHz
Memori	4 GB DDR3 RAM
Grafis	Intel HD
<i>Library</i>	OpenCV 3.3.1
<i>Compiler</i>	GCC 7.2.1

antara proses erosi yang diikuti dengan dilasi disebut juga opening. Dengan proses tersebut maka noise akan berkurang dan objek deteksi menjadi semakin jelas.

#### D. Estimasi Panjang Antrean

Proses ini meliputi 3 hal, yaitu menentukan ROI (Region of Interest) berupa garis, memeriksa nilai intensitas di setiap titik di sepanjang ROI, dan melakukan penghitungan panjang antrean berdasarkan perbandingan antara panjang ROI pada citra dengan jarak sebenarnya di lapangan.

### IV. IMPLEMENTASI DAN PENGUJIAN

Perancangan dan pengujian program dilakukan menggunakan komputer dengan sistem operasi Manjaro Linux 64 bit, library OpenCV 3.3.1, bahasa pemrograman C++ , dan compiler GCC 7.2.1. Spesifikasi lengkap untuk perangkat keras dan perangkat lunak yang digunakan dalam penelitian ini dapat dilihat pada tabel berikut

Pengujian pada penelitian ini meliputi pengujian background reconstruction, pengujian segmentasi citra, pengujian letak ROI, pengujian nilai intensitas di sepanjang ROI, dan estimasi panjang antrean kendaraan.

#### A. Pengujian background reconstruction

Pengujian ini mengukur seberapa jelas citra latar yang dihasilkan apabila background reconstruction dilakukan menggunakan 100 frame, 200 frame, dan 300 frame. Pada penggunaan 100 frame, citra latar yang dihasilkan masih buram dan gelap. Pada penggunaan 200 frame, citra latar yang dihasilkan lebih jelas dan halus. Sedangkan pada penggunaan 300 frame, citra latar yang dihasilkan mirip dengan hasil pengujian kedua, tetapi lebih jelas.

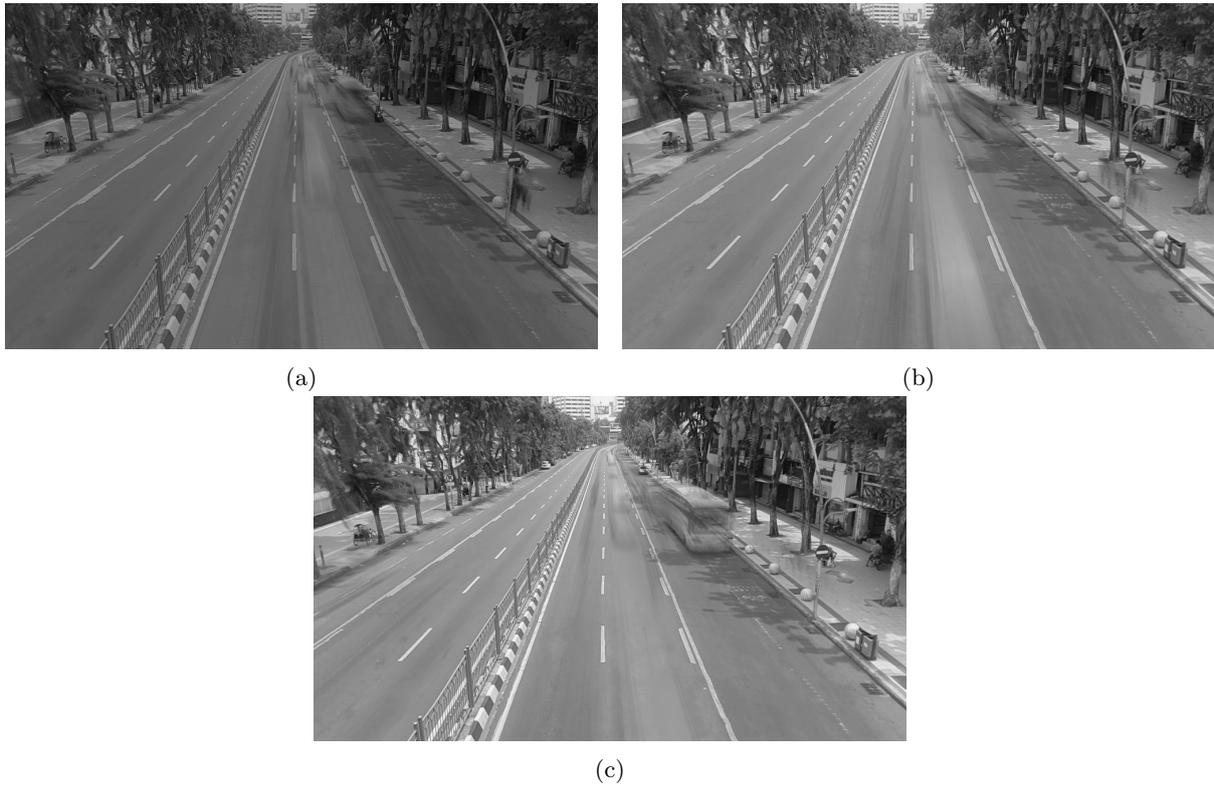
#### B. Pengujian segmentasi citra

Pengujian ini dilakukan untuk mengetahui seberapa baik citra latar (background image) yang didapatkan pada tahap sebelumnya, untuk digunakan sebagai background sehingga menghasilkan foreground image yang paling jelas. Dengan demikian, deteksi objeknya juga menjadi lebih akurat.

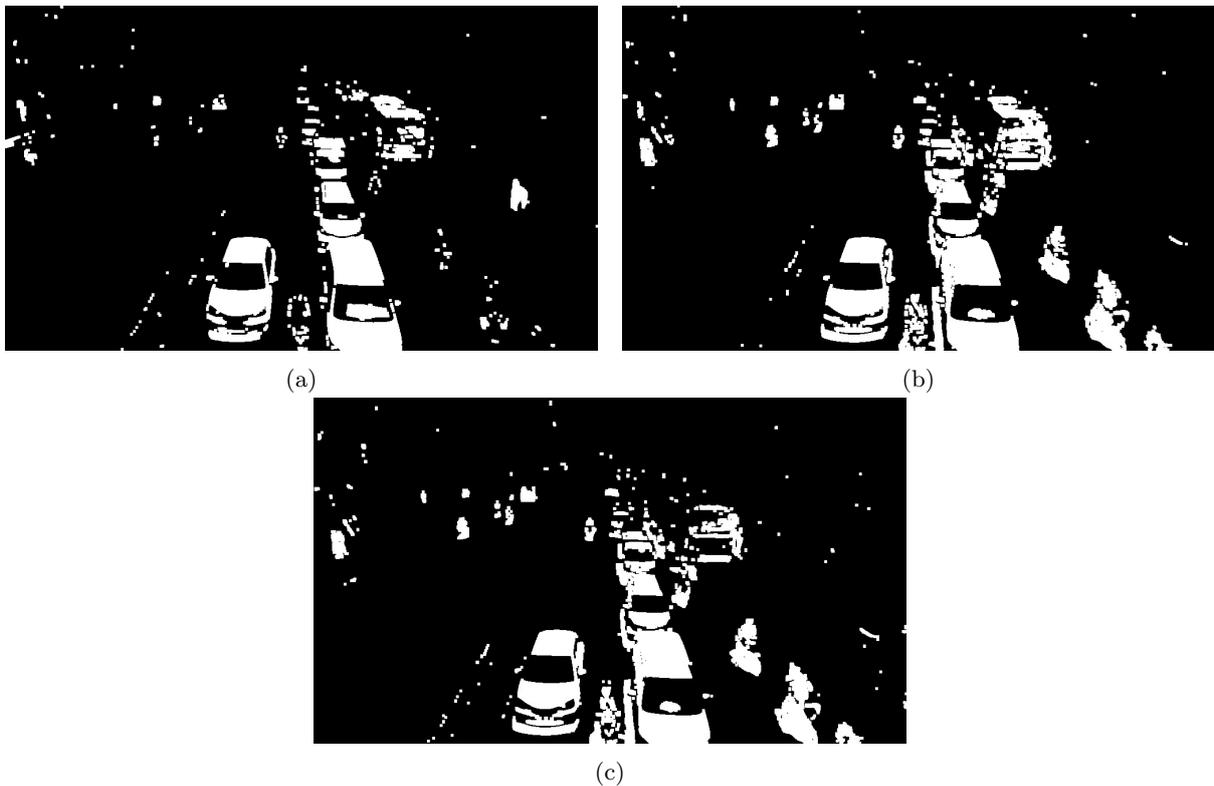
Berdasarkan hasil pengujian pada Gambar 6, Gambar 7, dan Gambar 8, dapat diketahui bahwa citra akhir yang paling baik deteksinya adalah yang menggunakan 300 frame.

#### C. Pengujian letak ROI

Pengujian ini dilakukan untuk mengetahui posisi terbaik untuk menempatkan ROI (Region of Interest) yang berupa garis, agar proses deteksi berada tepat pada bagian tengah antrean kendaraan, tidak terlalu ke kiri dan tidak terlalu ke kanan. Pada pengujian terhadap video lalu-lintas beresolusi 1024x600 piksel, didapatkan posisi sebagai berikut:

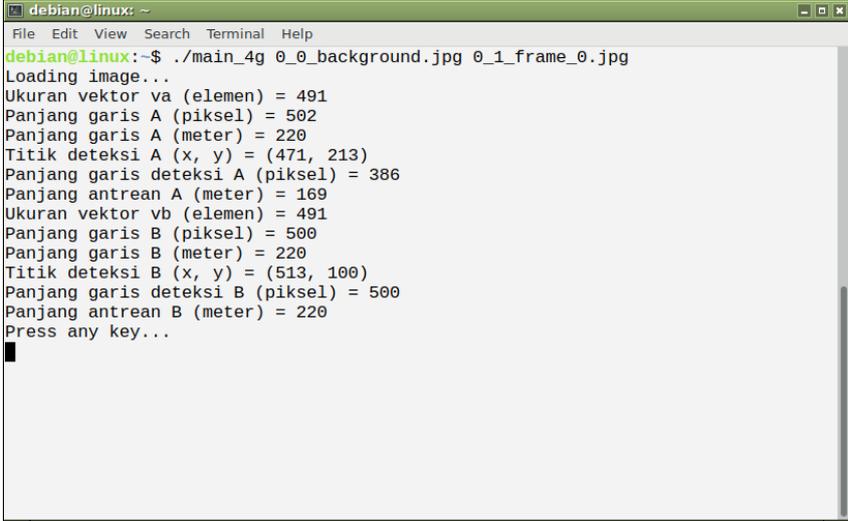


Gambar 3: Background reconstruction menggunakan (a) 100 frame, (b) 200 frame dan (c) 300 frame.



Gambar 4





```

debian@linux: ~
File Edit View Search Terminal Help
debian@linux:~$ ./main_4g 0_0_background.jpg 0_1_frame_0.jpg
Loading image...
Ukuran vektor va (elemen) = 491
Panjang garis A (piksel) = 502
Panjang garis A (meter) = 220
Titik deteksi A (x, y) = (471, 213)
Panjang garis deteksi A (piksel) = 386
Panjang antrean A (meter) = 169
Ukuran vektor vb (elemen) = 491
Panjang garis B (piksel) = 500
Panjang garis B (meter) = 220
Titik deteksi B (x, y) = (513, 100)
Panjang garis deteksi B (piksel) = 500
Panjang antrean B (meter) = 220
Press any key...

```

Gambar 7

pada bagian belakang garis A tidak terdapat objek kendaraan sehingga titik-titiknya berwarna hitam, yaitu titiktitik dengan nilai intensitas 0 (nol).

Dari perhitungan tersebut, didapatkan estimasi panjang antrean pada lajur kiri (dari kamera) adalah 169 meter, sedangkan panjang antrean pada lajur tengah adalah 220 meter.

## E. Estimasi panjang antrean

Berdasarkan pengecekan nilai intensitas titik-titik di sepanjang ROI sebagaimana tampak pada Gambar 10 yang pengecekannya dimulai dari titik di ujung belakang, kemudian program memeriksa alamat piksel dari titik putih yang terdeteksi, serta menghitung panjang garis mulai dari titik awal antrean (bawah) hingga titik putih tersebut, kemudian nilainya dikalikan dengan 220 meter yang merupakan jarak sebenarnya. Hasil perhitungan untuk garis A (merah) dan garis B (hijau) dapat dilihat pada gambar berikut:

## V. Kesimpulan

Berdasarkan pada hasil perancangan dan pengujian dari keseluruhan sistem dalam penelitian ini, maka diperoleh kesimpulan bahwa estimasi panjang antrean dengan hanya melakukan perhitungan panjang garis, memberikan hasil yang kurang akurat jika diterapkan pada citra yang bersifat perspektif.

## Daftar Pustaka

- [1] Andi Muhammad Ali Mahdi Akbar. Traffic IP Camera untuk Menghitung Kendaraan Roda Empat Menggunakan Metode Luasan Piksel. Intitut Teknologi Sepuluh Nopember. 2016.
- [2] R. K. Satzoda, S. Suchitra, T. Srikanthan, dan J. Y. Chia. Vision-based Vehicle Queue Detection at Traffic Junctions. Nanyang Technological University.
- [3] M. Wiering, J. Van Veenen, J. Vreeken, dan A. Koopman. Intelligent Traffic Light Control. Institute of Information and Computing Sciences. Utrecht University. 2004.
- [4] Adrian Kaehler, dan Gary Bradsky. Learning OpenCV, 2nd Edition. O'Reilly Media, Inc. 2014.
- [5] Helmiriawan. Rancang Bangun dan Analisis Sistem Pemantau Lalu Lintas Menggunakan OpenCV dengan Algoritma Canny dan Blob Detection. Universitas Indonesia. 2012.