

## DETEKSI, PELACAKAN DAN JUMLAH KENDARAAN DENGAN *MOVING AVERAGE SUBTRACTION* DAN *HUNGARIAN*

Amran Swandy Simamora<sup>1)</sup>, Imelda Imelda<sup>2)</sup>

<sup>1,2</sup>Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

<sup>1,2</sup>Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

Telp. (021) 5853753, Fax. (021) 5853752

E-mail : amransimamora540@gmail.com<sup>1)</sup>, imelda@budiluhur.ac.id<sup>2)</sup>

### ABSTRAK

Selama ini Dinas Perhubungan melakukan perhitungan kendaraan dengan cara manual. Caranya dengan menugaskan petugas mengamati dan menghitung kendaraan yang melintas. Namun hasilnya masih banyak *human error*. Selain itu Dinas Perhubungan juga menggunakan alat *detector* untuk perhitungan kendaraan. Alat tersebut sudah membantu Dinas Perhubungan untuk *me-monitoring* keadaan jalur lalu lintas. Namun harga alat *detector* itu mahal sehingga Dinas Perhubungan masih menggunakan cara manual. Oleh karena itu dibutuhkan deteksi, pelacakan dan perhitungan kendaraan secara otomatis. Proses deteksi kendaraan menggunakan dua tahap, yaitu tahap pelatihan (*training*) dan pengujian. Tahap pelatihan (*training*) terdiri dari pra pemrosesan, segmentasi dengan *moving average*, dan ekstraksi ciri. Tahap pengujian terdiri dari pra pemrosesan, segmentasi, ekstraksi ciri, *feature matching*. Setelah itu lanjut ke tahap pelacakan (*tracking*) dengan algoritma *hungarian* dan perhitungan kendaraan. Keistimewaan penelitian ini adalah deteksi kendaraannya menggunakan *moving average subtraction*, *morphological operation*, *shadow removal*, dan pada pelacakan dan perhitungan kendaraan menggunakan algoritma *Hungarian*. Akurasi tertinggi diperoleh pada kendaraan panjang yaitu 77%.

**Kata kunci :** *CCTV*, *Background Subtraction*, *Moving Average Subtraction*, *Morphological Operation*, *Shadow Removal*, *Algoritme Hungarian*.

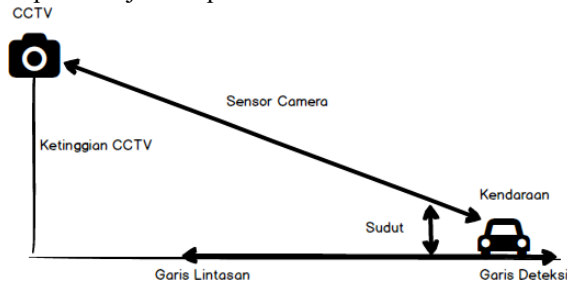
### 1. PENDAHULUAN

Selama ini Dinas Perhubungan melakukan perhitungan kendaraan dengan cara manual. Cara manual tersebut dilakukan dengan cara menugaskan petugas mengamati kendaraan yang melintas. Dengan cara tersebut mengakibatkan terjadinya *human error* sehingga tidak menghasilkan perhitungan yang tepat. Selain menghitung manual, Dinas Perhubungan juga menggunakan alat *detector* untuk menghitung kendaraan yang melintas di jalan raya. Penggunaan alat ini sudah memudahkan proses perhitungan dibandingkan dengan menghitung secara manual yang membutuhkan tenaga kerja yang banyak. Namun menggunakan alat *detector* tersebut membutuhkan biaya yang sangat besar. Sehingga alat *detector* tersebut jarang digunakan. Proses perhitungan dilakukan untuk analisa pemberian atau pembagian waktu lampu lalu lintas pada setiap jalur lalu lintas DKI Jakarta. Penelitian [1] telah membahas Implementasi *image subtraction* dan metode *regionprops* untuk mendeteksi jumlah objek berwarna RGB pada file video. Hasil yang didapat dari pengimplementasian cara *image subtraction* dan pemanfaatan fungsi *regionprops* sampai hasil yang

dicapai menunjukkan bahwa objek dengan kandungan warna pokok dapat terdeteksi jumlahnya. Penelitian [8] telah membahas *background subtraction* yang digunakan untuk mendeteksi gerakan objek. *Background subtraction* dapat menghitung kendaraan lalu lintas berdasarkan jenis kendaraan yang melintas di ruas jalan. Akurasi perhitungan kendaraan sebesar 90%. Kekurangannya adalah siang hari menimbulkan pantulan cahaya yang tinggi sehingga sistem mendeteksi tidak optimal. Penelitian [5] *Shadow removal* digunakan untuk menghilangkan bayangan. Keteguhan warna di eksplorasi menghilangkan warna sebenarnya. Kekurangannya gambar tidak bisa di anggap sebagai gambar abu-abu. Dari beberapa penelitian terdahulu belum ada penelitian yang membahas tentang deteksi, pelacakan dan jumlah kendaraan dengan metode *Moving Average Subtraction* dan algoritme *Hungarian* yang dapat bermanfaat untuk *analytic* lalu lintas. Oleh karena itu keistimewahan dari penelitian ini adalah dapat melakukan deteksi, pelacakan dan perhitungan kendaraan untuk melakukan *analytic* lalu lintas menggunakan metode *moving average subtraction* dan algoritme *Hungarian*.

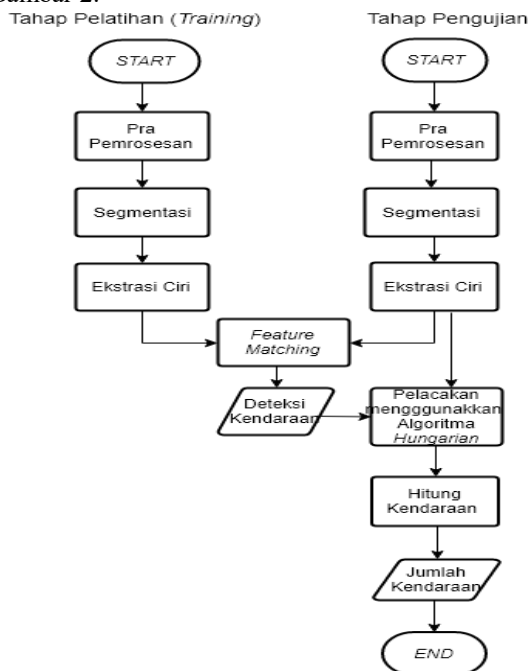
## 2. DETEKSI KENDARAAN DENGAN MOVING AVERAGE SUBTRACTION, PELACAKAN DAN PERHITUNGAN KENDARAAN DENGAN ALGORITME HUNGARIAN

Deteksi dan perhitungan kendaraan secara otomatis menjadi salah satu solusi yang penting untuk membantu para petugas Dinas Perhubungan dalam melakukan perhitungan kendaraan di lintasan lalu lintas. Oleh karena itu dilakukan penelitian mengenai pendeteksian dan perhitungan kendaraan dari sumber video. Data tersebut akan diperoleh dari kamera CCTV yang terpasang di persimpangan setiap jalan yang ada di DKI Jakarta. Gambar ilustrasi data input dapat ditunjukkan pada Gambar 1.



Gambar 1 Ilustrasi Data Input

Adapun aplikasi deteksi dan penghitung kendaraan secara otomatis pada program yang penulis akan buat memiliki proses-proses secara garis besar sebagai berikut. Gambar Alur proses dapat ditunjukkan pada Gambar 2.



Gambar 2 Alur Proses

### 2.1. Tahap Pelatihan (Training)

Pada tahap pelatihan ini berguna untuk mendapatkan hasil gambar dari suatu video. Tahap pelatihan ini terdiri dari pra pemrosesan, segmentasi, dan ekstraksi ciri.

#### 2.1.1. Pra Pemrosesan

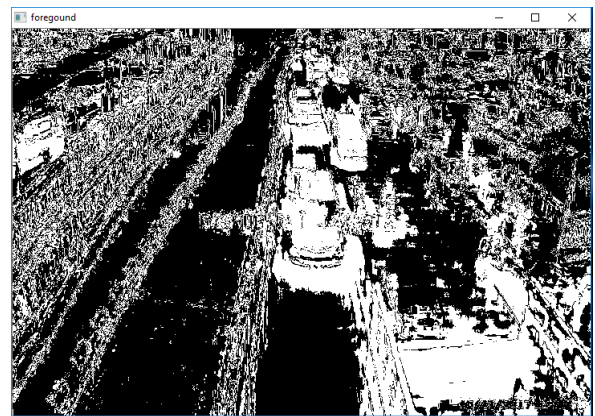
Pra pemrosesan ini berguna untuk normalisasi frame rate, frame pada setiap video akan disamakan menjadi 30 fps. Normalisasi video menggunakan *software handbrike*.

#### 2.1.2. Segmentasi

Segmentasi ini berguna untuk melakukan partisipasi citra menjadi wilayah-wilayah objek [9]. Tahap segmentasi merupakan proses yang cukup penting dalam suatu sistem pengenalan untuk mengenali karakter yang terdapat pada suatu citra di dalam video *digital*. Tahap Segmentasi terdiri dari *moving average subtraction*, *morphological operation*, *shadow removal*

##### 2.1.2.1. Moving Average Subtraction

*Moving Average Subtraction* adalah metode pengurangan *template*, dimana *template* terpisah dibuat untuk setiap kejadian artefak. *Moving average* memiliki waktu pemutakhiran latar belakang yang tepat dan panjang pergerakan yang tepat dapat dipilih dengan eksperimen sedemikian rupa sehingga benda yang bergerak dapat di deteksi dalam performa yang baik. Latar belakang akan memperbaharui benda dengan tepat dan segera mendeteksi benda bergerak dengan akurasi yang baik [2]. Gambar *Moving Average Subtraction* dapat ditunjukkan pada Gambar 3



Gambar 3 *Moving Average Subtraction*

##### 2.1.2.2. Morphological Operation

Dalam pemrosesan citra hasil yang diharapkan diperoleh berdasarkan bentuk atau struktur citra asal. Lebih lanjut dikatakan bahwa morfologi senantiasa

berkaitan dengan proses ketetanggaan, yang terbentuk dari blok nilai biner, satu dan nol. Lebih lanjut lagi, proses morfologi suatu citra merupakan kumpulan operasi nonlinier yang berkaitan dengan bentuk atau morfologi dalam suatu citra [3]. Hasil operasi morfologi dimanfaatkan untuk pengambilan keputusan dengan analisa lebih lanjut. Ada dua operator morfologi operasi dasar yaitu dilasi dan erosi.

2.1.2.2.1. Erosi

Proses erosi sama seperti pelebaran, namun pikselnya diubah menjadi putih bukan hitam. Itu dua input utama untuk operator erosi adalah gambar yang akan ditipiskan dan satu set titik koordinat yang dikenal sebagai elemen penataan yang mendefinisikan kernel [4]. Persamaan erosi dapat ditunjukkan pada Persamaan 1.

$$A \ominus B = \{z | (B)_z \cap A^c \neq \emptyset\} \quad (1)$$

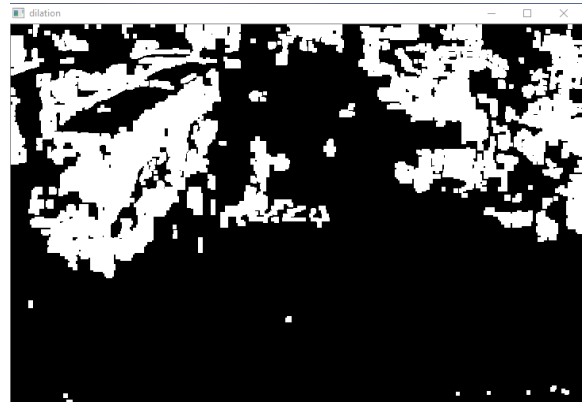
Variabel A adalah gambar A, variabel B adalah elemen penataan B,  $A \ominus B$  menyatakan bahwa A adalah erosi dari B dan  $A^c$  menyatakan bahwa komplement dari A.

2.1.2.2.2. Dilasi

Proses elemen penataan B pada gambar A dan memindahkannya melintasi dengan cara seperti konvolusi (dilibatkan dalam operasi ketetanggaan piksel) didefinisikan sebagai operasi pelebaran. Dua input utama untuk operator pelebaran adalah gambar yang akan dilebarkan dan satu set titik koordinat yang dikenal sebagai elemen penataan yang juga di defenisikan sebagai kernel [4]. Persamaan dilasi dapat ditunjukkan pada Persamaan 2.

$$A \oplus B = \{z | (\bar{B})_z \cap A \neq \emptyset\} \quad (2)$$

Variabel A adalah gambar A, variabel B adalah elemen penataan B,  $A \oplus B$  menyatakan bahwa refleksi dari penataan B. Gambar *morphological operation* dapat ditunjukkan pada Gambar 4.



Gambar 4 *Morphological Operation*

2.1.2.2.3. Shadow removal

*Shadow removal* adalah mengubah kondisi iluminasi tidak standar menjadi standar dan menentukan nilai abu-abu dalam standar. Jadi kuncinya adalah memperkirakan kondisi sumber cahaya standar.

$$e = k \left( \frac{\iint (f(x,y))^p dx dy}{\iint dx dy} \right)^{\frac{1}{p}} = k \left( \frac{\sum_{x=1}^M \sum_{y=1}^N (f(x,y))^p}{MN} \right)^{\frac{1}{p}} \quad (3)$$

Dimana e adalah nilai sumber cahaya dari satu *scene* saat ini, f adalah nilai abu-abu dari masing-masing band gambar, dan k adalah faktor skala. Dan p adalah parameter eksponensial dari norma, itu bisa menjadi bilangan bulat. P menentukan berat setiap nilai abu-abu pada sumber cahaya yang diperkirakan. Semakin banyak efek dari piksel kecerahaan tinggi. Nilai p yang lebih kecil, bobot piksel kecerahaan berbeda lebih besar [5]. Gambar *shadow removal* dapat ditunjukkan pada Gambar 5.



Gambar 5. *Shadow removal*

2.1.3. Ekstraksi Ciri

Ekstraksi ciri berguna untuk mengukur besaran kuantitatif ciri di setiap piksel dari sebuah citra [2]. Pada penelitian ini menggunakan ekstraksi ciri tekstur dan ekstraksi ciri bentuk. Ekstraksi ciri tekstur merupakan informasi berupa susunan struktur permukaan suatu gambar. Citra warna dirubah

menjadi citra grayscale dan masing-masing nilai dari RGB citra dirubah menjadi abu-abu dengan menggunakan rumus dari persamaan 3:

$$\text{Keabuan} = 0.2989 * R + 0.5870 * G + 0.1140 * B \quad (3)$$

Ekstraksi ciri bentuk adalah menemukan bagian pada citra yang mengalami perubahan intensitas secara drastis [11]. Ekstraksi ciri bentuk dikategorikan bergantung pada teknik yang digunakan. Katergori tersebut adalah berdasarkan batas (*boundary*). Berdasarkan batas (*boundary*) menggambarkan bentuk daerah dengan menggunakan karakteristik ekxternal, contohnya adalah piksel sepanjang batas objek [12]. Tahap ekstraksi ciri adalah tahap untuk mengurangi dimensi hiperspektral untuk menjaga variabel-variabel yang digunakan untuk proses perhitungan kendaraan.

Ciri-ciri objek meliputi: *width\_frame* (*WF*), *heigth\_frame* (*HF*), *width\_Contour* (*WC*), *hight\_Contour* (*HC*), *xContour* (*XC*), *yContour*(*YC*), *sensorwidth* (*SW*), *areaBoundary*, *HeightInFullFrame*(*HIF*), *HeigthSurplus* (*HS*), *y1Vehicle* (*Y1*), *y2Vehicle* (*Y2*), *AspectRatioHeight* (*ARH*), *LengthVehicle*, *maxLengthLV*. *Sensorwidth* (*SW*) sama dengan istilah *boundary*. *AreaBoundary* adalah area batas dapat ditunjukkan pada persamaan (4). *HeightInFullFrame*(*HIF*) berguna untuk mengetahui tinggi dalam frame penuh. *HIF* dapat di tunjukkan pada persamaan (5). *HeigthSurplus* (*HS*) dapat ditunjukkan pada persamaan (6), *y1Vehicle* (*Y1*) dimana ukuran veritikal *y1* pada objek yang terdeteksi dapat ditunjukkan pada persamaan (7), *y2Vehicle* (*Y2*) dimana ukuran veritikal *y2* pada objek yang terdeteksi dapat ditunjukkan pada persamaan (8). *AspectRatioHeight* (*ARH*) adalah aspek rasio dari lebar dan tinggi objek deteksi. *ARH* dapat ditunjukkan pada persamaan (9). *LengthVehicle* adalah panjang kendaraan. *MaxLengthLV* adalah maksimal panjang kendaraan. *VehilceDetection* merupakan variabel yang menampung jumlah kendaraan.

$$\text{areaBoundary} = WC * HC \quad (4)$$

$$\text{HeightInFullFrame} = (WF * 2.0) / 3 \quad (5)$$

$$\text{HeigthSurplus} = (HIF - HF) / 2 \quad (6)$$

$$y1Vehicle = (HF + HS) - (YC+HC) \quad (7)$$

$$y2Vehicle = (HF+HS) - YC \quad (8)$$

$$\text{AspectRatioHeigth} = (SW / WF) * HIF \quad (9)$$

Algoritme 1 digunakan untuk menentukan kendaraan panjang dan sedang. Jika *LengthVehicle* lebih kecil sama dengan *maxLengthLV* maka *VehicleDetection* sama dengan *LV* (Kendaraan Sedang). Jika *LengthVehicle* lebih besar sama dengan *maxLengthLV* maka *VehicleDetection* sama dengan *HV* (Kendaraan panjang).

Algoritme 1 Deteksi Kendaraan

if *LengthVehicle* <= *maxLengthLV*:

*VehicleDetection* = "LV"

*color* = *colorLV*

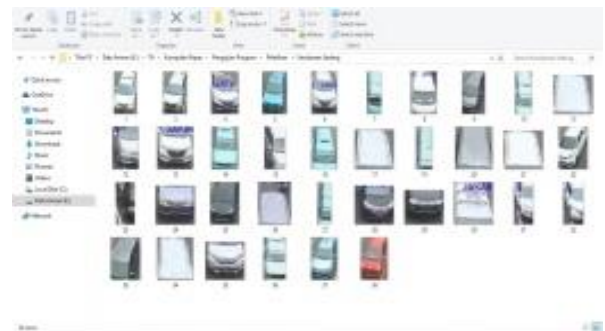
else:

*VehicleDetection* = "HV"

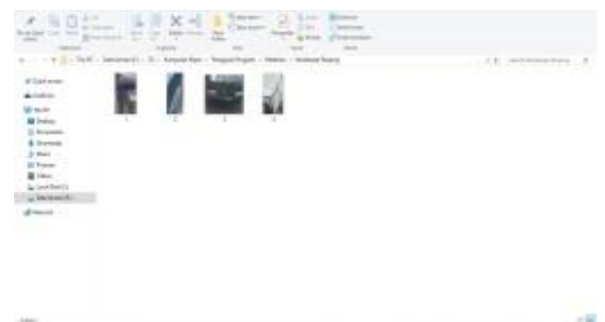
*color* = *colorHV*

2.2. Tahap Pengujian

Tahap pengujian dilakukan untuk menguji hasil gambar dari tahap pelatihan. Tahap pengujian terdiri dari pra pemrosesan, segmentasi dan ekstraksi ciri. Pada tahap pengujian digunakan 36 kendaraan sedang dan 4 untuk kendaraan panjang. Gambar kendaraan sedang dan kendaraan panjang dapat ditunjukkan pada Gambar 6 dan 7.



Gambar 6 Kendaraan Sedang



Gambar 7 Kendaraan Panjang

### 2.2.1. Pra Pemrosesan

Proses yang dilakukan pada tahap ini sama dengan tahap pra pemrosesan yang ada pada tahap pelatihan

### 2.2.2. Segmetasi

Pada tahap segmentasi ini proses yang dilakukan pada tahap ini sama dengan tahap segmentasi yang ada pada tahap Pelatihan [9]. Tahap segmentasi ini terdiri dari *moving average subtraction*, *morphological operation*, *shadow removal*.

#### 2.2.2.1. *Moving average subtraction*

*Moving average subtraction* adalah metode pengurangan *template*, dimana *template* terpisah dibuat untuk setiap kejadian artefak. *Moving average* memiliki waktu pemutakhiran latar belakang yang tepat dan panjang pergerakan yang tepat dapat dipilih dengan eksperimen sedemikian rupa sehingga benda yang bergerak dapat di deteksi dalam performa yang baik. Latar belakang akan memperbaharui benda dengan tepat dan segera mendeteksi benda bergerak dengan akurasi yang baik [2]. Gambar *Moving Average Subtraction* dapat di tunjukkan pada Gambar 3.

#### 2.2.2.2. *Morphological operation*

Dalam pemrosesan citra hasil yang diharapkan diperoleh berdasarkan bentuk atau struktur citra asal. Lebih lanjut dikatakan bahwa morfologi senantiasa berkaitan dengan proses ketetanggaan, yang terbentuk dari blok nilai biner, satu dan nol. Lebih lanjut lagi, proses morfologi suatu citra merupakan kumpulan operasi nonlinier yang berkaitan dengan bentuk atau morfologi dalam suatu citra [3]. Hasil operasi morfologi dimanfaatkan untuk pengambilan keputusan dengan analisa lebih lanjut. Ada dua operator morfologi operasi dasar yaitu dilasi dan erosi.

##### 2.2.2.2.1. Erosi

Proses erosi sama seperti pelebaran, namun pikselnya diubah menjadi putih bukan hitam. Itu dua input utama untuk operator erosi adalah gambar yang akan ditipiskan dan satu set titik koordinat yang dikenal sebagai elemen penataan yang mendefinisikan kernel [4]. Persamaan Erosi dapat ditunjukkan pada Persamaan 1.

##### 2.2.2.2.2. Dilasi

Proses elemen penataan B pada gambar A dan memindahkannya melintasi dengan cara seperti konvolusi (dilibatkan dalam operasi ketetanggaan piksel) didefinisikan sebagai operasi pelebaran. Dua input utama untuk operator pelebaran adalah gambar yang akan dilebarkan dan satu set titik koordinat yang dikenal sebagai elemen penataan yang juga di defenisikan sebagai kernel [4]. Persamaan Dilasi dapat ditunjukkan pada Persamaan 2. Gambar *Morphological Operation* dapat ditunjukkan pada Gambar 4.

##### 2.2.2.2.3. *Shadow removal*

*Shadow removal* adalah mengubah kondisi iluminasi tidak standar menjadi standar dan menentukan nilai abu-abu dalam standar. Jadi kuncinya adalah memperkirakan kondisi sumber cahaya standar[5]. Persamaan *Shadow removal* dapat ditunjukkan pada Persamaan 3 dan gambar dapat ditunjukkan pada Gambar 5.

### 2.2.3. Ekstrasi Ciri

Pada tahap ekstraksi ciri sama dengan tahap ekstraksi ciri pada tahap pelatihan.

#### 2.3. *Feature Matching*

Pada tahap *Feature Matching* berguna untuk mencocokkan hasil gambar dari tahap pelatihan dengan tahap pengujian.

#### 2.4. Pelacakan (*Tracking*)

Pada tahap Pelacakan (*Tracking*) kendaraan ditentukan di daerah yang di tentukan. Pelacakan menggunakan algoritme *hungarian*.

##### 2.4.1. *Algoritme hungarian*

*Algoritme Hungarian* merupakan algoritma untuk memecahkan masalah penugasan yang didasarkan pada karya D. Konig dan J. Egervary [6].

##### 2.4.2. Hitung Kendaraan

Pada tahap hitung kendaraan dilakukan setelah tahap pelatihan (*Training*), tahap pengujian, *feature matching* dan pelacakan. Hitung kendaraan dilakukan dengan cara deteksi kendaraan dari ROI (*Region of Interest*). Dari garis awal di deteksi dan di garis akhir di lakukan perhitungan kendaraan.

## 3. HASIL DAN PEMBAHASAN

### 3.1. Implementasi

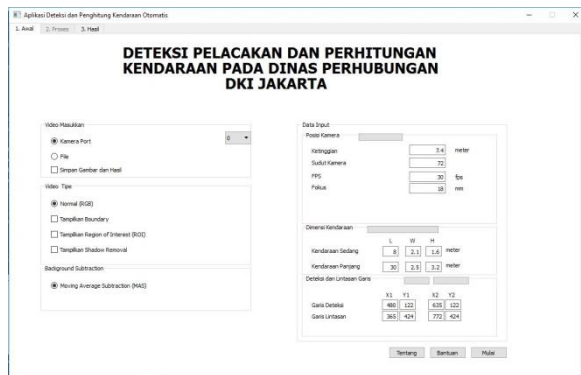
Agar deteksi, pelacakan dan perhitungan kendaraan ini dapat berjalan dengan baik dan sesuai yang diharapkan maka spesifikasi perangkat keras (*hardware*) dan perangkat lunak (*software*) yang di pakai untuk implementasi ini juga harus mendukung.



**3.2. Tampilan Layar**

**3.2.1. Halaman Awal Aplikasi**

Pada groupbox Input video terdapat *chosed file* untuk video masukkan, mode video terdapat *checkbox* tampilan *boundary*, tampilan ROI dan tampilan *shadow removal*. Pada *background subtraction* terdapat *moving average subtraction*. Data input kamera terdapat ketinggian kamera, sudut kamera, FPS, dan fokus yang diinput manual. Pada dimensi kendaraan di input berdasarkan Undang-Undang Republik Indonesia nomor 22 tahun 2009 [10] untuk kendaraan sedang dimensi kendaraan adalah tinggi 1,3 m, lebar 2,1 m, panjang 5,8 m dan untuk kendaraan panjang dimensinya adalah tinggi 4,1 m, lebar 2,6 m, panjang 12,1 m. Dan ROI diinput oleh user berdasarkan video yang akan di jalankan. Gambar halaman awal dapat ditunjukkan pada Gambar 8.



Gambar 8 Halaman Awal

**3.2.2. Halaman Proses**

Pada halaman proses dimana proses deteksi dan perhitungan kendaraan dilakukan. Gambar halaman proses dapat ditunjukkan pada Gambar 9.



Gambar 9 Halaman Awal

**3.3. Tabel Pengujian**

Pengujian ini berguna untuk mendapatkan hasil perhitungan kendaraan. Pengujian ini menggunakan

*Confusion Matrik* [7]. Tabel hasil pengujian dapat ditunjukkan pada Tabel 1.

$$Akurasi = \frac{(TN + TP)}{(TN + TP + FN + FP)} \times 100\% \quad (10)$$

- 1) *True Positive* (TP) = sebuah kendaraan terdeteksi di kondisi nyata dan dinyatakan sebagai kendaraan di aplikasi.
- 2) *False Positive* (FP) = tidak ada kendaraan terdeteksi dalam kondisi nyata namun nyatakan kendaraan di aplikasi.
- 3) *True Negative* (TN) = tidak ada kendaraan terdeteksi dalam kondisi nyata dan bukan pada aplikasi.
- 4) *False Negative* (FN) = kendaraan terdeteksi di kondisi nyata namun tidak dinyatakan kendaraan pada aplikasi.

Tabel 1 Hasil Akurasi

Nama Video		Layar	Kebopala	Pejaten
Kendaraan Sedang	TN	0	0	0
	TP	23	41	4
	FN	13	18	21
	FP	0	0	0
Akurasi		63%	69%	19%
Kendaraan Panjang	TN	0	0	0
	TP	6	7	4
	FN	3	2	3
	FP	0	0	0
Akurasi		66%	77%	57%

Pada pengujian ini menggunakan contoh video layar.avi

Akurasi Kendaraan Sedang pada Layar

$$Akurasi = \frac{(TN + TP)}{(TN + TP + FN + FP)} \times 100\%$$

$$Akurasi = \frac{(0 + 23)}{(0 + 23 + 13 + 0)} \times 100\%$$

$$Akurasi = \frac{23}{36} \times 100\% = 63\%$$

Akurasi Kendaraan Panjang pada Layar

$$Akurasi = \frac{(TN + TP)}{(TN + TP + FN + FP)} \times 100\%$$

$$Akurasi = \frac{(0 + 6)}{(0 + 6 + 3 + 0)} \times 100\%$$

$$Akurasi = \frac{6}{9} \times 100\% = 66\%$$

**4. KESIMPULAN**

Dari hasil penelitian yang dilakukan pada deteksi kendaraan menggunakan *Moving Average Subtraction*, pelacakan dan perhitungan kendaraan menggunakan algoritme *Hungarian* dapat diperoleh kesimpulan sebagai berikut:

- a. Dalam pendeteksian objek menggunakan metode *Moving Average Subtraction* sehingga dapat mendeteksi kendaraan secara cepat. Dan menggunakan morphological operation untuk mengumpulkan operasi nonlinier yang berkaitan dengan bentuk atau morfologi suatu citra. Dan menggunakan *shadow removal* untuk penghilangan bayangan.
- b. Untuk pelacakan dan perhitungan kendaraan menggunakan algoritme *Hungarian*. Algoritme *Hungarian* merupakan algoritme untuk memecahkan masalah penugasan.
- c. Dalam penelitian ini diperoleh tingkat akurasi yaitu sebesar 77%.

#### DAFTAR PUSTAKA

- [1] Ardhiyanto, E. and Hadikurniawati, W. (2013) 'Implementasi Metode Image Subtracting dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video', *Jurnal Teknologi Informasi DINAMIK*, 18(2), pp. 91–100.
- [2] Sun, L., Ahlfors, S. P. and Hinrichs, H. (2016) 'Removing Cardiac Artefacts in Magnetoencephalography with Resampled Moving Average Subtraction', *Brain Topography*. Springer US, 29(6), pp. 783–790. doi: 10.1007/s10548-016-0513-3.
- [3] Hidayati, Q. (2017) 'Kendali Lampu Lalu Lintas dengan Deteksi Kendaraan Menggunakan Metode Blob Detection', 6(2).
- [4] Raid, A. M. *et al.* (2014) 'Image Restoration Based on Morphological Operations', *International Journal of Computer Science, Engineering and Information Technology (IJCEIT)*, 4(3), pp. 9–21. doi: 10.5121/ijceit.2014.4302.
- [5] Ye, Q., Xie, H. and Xu, Q. (2012) 'Removing Shadows From High-resolution Urban Aerial Images Based On Color Constancy', *ISPRS Congress*, XXXIX-B3(September), pp. 525–530.
- [6] Yulistiana, M., Chaerani, D. and Lesmana, E. (2015) 'Penerapan Metode Hungarian dalam Penentuan Penjadwalan Matakuliah Optimal ( Studi Kasus: Departemen Matematika Universitas Padjadjaran Semester Ganjil 2013-2014 )', 11(1), pp. 45–64.
- [7] Sundoro, H. S. and Harjoko, A. (2016) 'Vehicle counting and vehicle speed measurement based on video processing', *Journal of Theoretical and Applied Information Technology*, 84(2), pp. 233–241.
- [8] Dewantoro, A. K. *et al.* (2015) 'Simulasi Dan Analisis Sistem Penghitung Kepadatan Lalu Lintas Dan Klasifikasi Kendaraan Berbasis Webcam Dengan Metode Background Subtraction', 2(2), pp. 2833–2840.
- [9] Ikhsanuddin, R. M. (2014) 'Identifikasi Citra Pada Plat Nomor Kendaraan Mobil Pribadi Menggunakan Metode K-Nearest Neighbor', pp. 1–7.
- [10] Undang-undang Republik Indonesia nomor 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Jalan.
- [11] Rai, I. G. *et al.* (2017) 'Ekstraksi Fitur Warna , Tekstur dan Bentuk untuk Clustered-Based Retrieval of Images ( CLUE )', 16(April).
- [12] Agung, I. G. R. (2016) 'Ekstraksi Warna, Tekstur, Dan Bentuk Untuk Image Retrieval', *Seminar Nasional Teknologi Informasi dan Multimedia 2016*, (2302–3805), p. 4.3-1-4.3-6.