

## IMPLEMENTASI PERHITUNGAN KENDARAAN MOBIL DI JALAN RAYA MENGUNAKAN METODE BACKGROUND SUBTRACTION DAN TEKNIK MORFOLOGI CITRA

Solihun<sup>1</sup>, Ikhwan Ruslianto<sup>2</sup>, Uray Ristian<sup>3</sup>

<sup>1,2,3</sup>Jurusan Rekayasa Sistem Komputer, Fakultas MIPA Universitas Tanjungpura

Jalan Prof Dr. H. Hadari Nawawi Pontianak

Telp./Fax. : (0561) 577963

e-mail: <sup>1</sup>solihun@student.untan.ac.id, <sup>2</sup>ikhwanruslianto@siskom.untan.ac.id,

<sup>3</sup>eristian@siskom.untan.ac.id

### ABSTRAK

Jalan raya merupakan sarana umum masyarakat yang dibangun untuk memobilisasi masyarakat dibidang ekonomi, sosial, dan industri. Banyaknya kendaraan mobil yang melintas jalan raya mengakibatkan kepadatan lalu lintas yang terjadi di area tertentu. Dalam mengetahui kepadatan lalu lintas di area tertentu dibutuhkan survei kendaraan. Survei kendaraan yang ada khususnya di Kota Pontianak, masih menggunakan cara manual yaitu petugas survei turun langsung ke jalan raya untuk mengambil data survei kendaraan. Hal ini membutuhkan waktu yang lama dalam pengambilan data. Dalam penelitian ini, dibangun sebuah sistem perhitungan kendaraan mobil berbasis citra digital. Penelitian yang dilakukan menggunakan metode *Background Subtraction* dan Operasi Morfologi Citra. Ada 7 tahap yang dilakukan dalam menghitung jumlah kendaraan mobil yaitu tahap Akuisisi Citra, *Pre-processing*, *Background Subtraction*, Operasi Morfologi Citra, Proses Deteksi Kendaraan Mobil, Perhitungan Kendaraan Mobil, dan Integrasi Ke Halaman *Website*. Pengujian dilakukan menggunakan 30 data uji, data yang diuji menggunakan 2 durasi waktu yaitu 2 menit dan 3 menit. Masing masing data uji memiliki kondisi video normal, gelap, dan terang. Berdasarkan hasil pengujian, diperoleh nilai rata-rata tingkat akurasi pada kondisi video normal sebesar 86.20%, pada kondisi gelap sebesar 66.54%, dan pada kondisi terang sebesar 77.63%.

**Kata Kunci:** Deteksi, Perhitungan Kendaraan Mobil, *Background Subtraction*, Morfologi Citra, *Website*.

### 1. PENDAHULUAN

Jalan raya merupakan sarana transportasi darat yang dibangun sebagai jalur penghubung antar wilayah. Keberadaan jalan raya dapat mempermudah mobilitas dan aksesibilitas masyarakat dalam bidang sosial, ekonomi, industri dan perdagangan di Kota Pontianak. Seiring meningkatnya mobilitas masyarakat membuat volume kendaraan setiap tahun semakin tinggi. Berdasarkan data yang dihimpun dari Badan Statistik Kota Pontianak jumlah kendaraan tercatat pada tahun 2018 berjumlah 1.023.464 unit [1].

Dari data yang tercatat membuktikan bahwa banyaknya jumlah kendaraan yang meningkat setiap tahun akan berakibat pada kemacetan dan kepadatan di jalan raya. Hal ini dikarenakan

fasilitas jalan sudah tidak sebanding lagi dengan jumlah kendaraan yang melintas di jalan raya. Tingginya kepadatan lalu lintas di jalan raya akan berakibat pada kemacetan dan juga polusi udara. Untuk mengetahui rata-rata kepadatan kendaraan di jalan raya dibutuhkan survei dalam menghitung jumlah kendaraan [2]

Pengambilan data survei kendaraan biasanya dilakukan oleh para peneliti lapangan yang turun langsung ke jalan raya. Pengambilan data yang dilakukan masih menggunakan cara manual yaitu dengan menghitung satu persatu kendaraan yang melintas di jalan raya. Hal tersebut di rasa masih kurang efektif karena akan membutuhkan waktu yang sangat lama dalam pengambilan data kendaraan. Dari permasalahan yang ada maka dibutuhkan

komputasi yang dapat menghitung kendaraan secara otomatis. Dengan adanya sistem perhitungan kendaraan secara otomatis dapat mempermudah dan mempercepat perhitungan kendaraan. Adapun sistem ini berbasis citra digital dan menggunakan metode *Background Subtraction* serta operasi morfologi citra. Sistem ini nantinya dikembangkan menggunakan citra video dari rekaman CCTV, yang kemudian dari rekaman tersebut akan diproses menggunakan pengolahan citra digital.

Sistem ini dapat mendeteksi dan menghitung berapa jumlah kendaraan mobil yang melintas di jalan raya secara otomatis. Metode *Background Subtraction* diterapkan pada citra video untuk mendeteksi objek yang bergerak dan operasi morfologi digunakan untuk memperbaiki hasil *Background Subtraction*. Dengan adanya sistem ini nantinya dapat mempermudah para peneliti dalam pengambilan data kendaraan dengan menggunakan data video sebagai data pengujian.

Penelitian yang terkait dengan sistem deteksi melalui video pernah dilakukan sebelumnya dengan judul “Aplikasi Pendeteksi Objek Bergerak pada *Image Sequence* Dengan Metode *Background Subtraction*” [3]. Pada penelitian ini metode *Background Subtraction* diterapkan pada objek kendaraan yang bergerak dari *image sequence* yang di tandai dengan *boundingBox* serta pembuatan aplikasi menggunakan *Matlab*.

Penelitian selanjutnya dengan judul “Perhitungan Kendaraan Otomatis menggunakan Metode *Background Subtraction* pada Gambar *Grayscale* dan Operasi Morfologi” [4]. Pada penelitian ini metode *Background Subtraction* untuk melacak, mendeteksi, dan menghitung jumlah kendaraan. Akurasi perhitungan kendaraan tertinggi didapatkan pada kondisi pagi hari di 86,364% dan terendah pada malam hari di 21,429% .

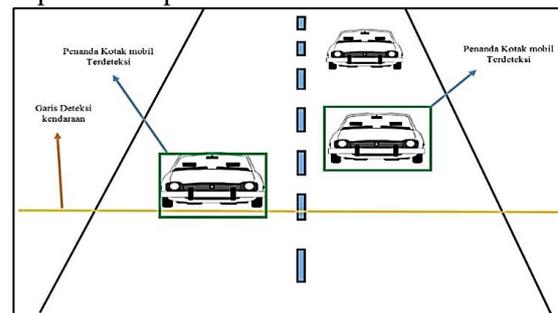
Penelitian selanjutnya dengan judul “Aplikasi Deteksi Objek Bergerak Berbasis Citra dengan Metode *Background Subtraction* dan *Blob Detection*” [5]. Pada penelitian ini *Background Subtraction* dan *Blob Detection* diterapkan untuk mendeteksi objek manusia dan hewan dalam keadaan gelap dan terang. Akurasi yang didapat dalam kondisi terang sebesar 70% dan gelap 75%.

Berdasarkan pemaparan yang telah di jelaskan sebelumnya, maka dirancang sebuah sistem perhitungan kendaraan mobil di jalan raya. Adapun penelitian yang dilakukan berjudul “Implementasi Perhitungan Kendaraan Mobil di Jalan Raya Menggunakan Metode *Background Subtraction* dan Teknik Morfologi Citra”.

## 2. LANDASAN TEORI

### 2.1 Deteksi Objek Bergerak

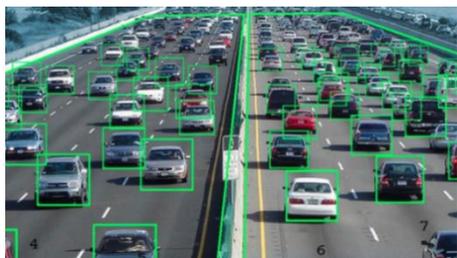
Deteksi objek bergerak merupakan proses dimana objek diperiksa dengan teknik dan cara tertentu sehingga, dapat menyelesaikan berbagai permasalahan contohnya deteksi objek bergerak pada mobil dengan cara objek mobil diidentifikasi menggunakan garis virtual [6]. Adapun gambar dari deteksi objek bergerak dapat dilihat pada Gambar 1.



Gambar 1. Deteksi Objek Bergerak

### 2.2 Traffic Counting

*Traffic Counting* merupakan survei yang dilakukan untuk menghitung jumlah volume kendaraan pada ruas jalan pada periode waktu tertentu. Volume lalu lintas adalah banyaknya kendaraan yang melewati suatu titik atau garis tertentu pada suatu penampang melintang jalan [7]. Tujuan dari *Traffic Counting* yaitu dapat mengumpulkan data lalu lintas, mengetahui karakteristik lalu lintas, mengetahui komposisi kendaraan dan untuk mengukur kinerja dari lalu lintas pada suatu ruas jalan. Dengan mengetahui volume kendaraan pada suatu jalan maka bisa didapatkan suatu informasi yang bisa kita manfaatkan sebagai data untuk mengukur kinerja jalan, membuat desain jalan, dan memberlakukan kebijakan di bidang manajemen dan rekayasa lalu lintas. Representasi dari *traffic counting* dapat dilihat pada Gambar 2 [7].

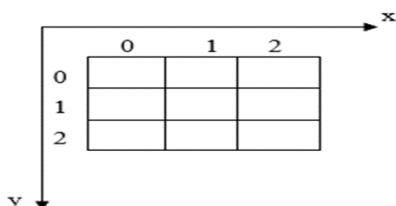


Gambar 2. Traffic Counting

### 2.3 Pengolahan Citra

Citra adalah gambar pada bidang dua dimensi yang dihasilkan dari gambar analog dua dimensi dan kontinu menjadi gambar diskrit, melalui proses sampling gambar analog dibagi menjadi M baris dan N kolom sehingga menjadi gambar diskrit [9]. Pengolahan citra digital adalah teknik mengolah citra agar mudah diinterpretasi oleh manusia atau mesin komputer yang dapat berupa foto maupun gambar bergerak [10]. Citra digital merupakan representasi dari fungsi intensitas cahaya dalam bentuk diskrit pada bidang dua dimensi. Citra tersusun dari beberapa pixel (*picture element*) yang memiliki koordinat (x,y) dan amplitudo f(x,y). Koordinat (x,y) menunjukkan letak/posisi pixel dalam suatu citra.

Pada penelitian ini menerapkan bidang pengolahan citra sebagai dasar untuk membuat sebuah aplikasi perhitungan kendaraan. Inputan video berasal dari kamera cctv atau kamera statis. Pengolahan citra merupakan suatu sistem dimana proses dilakukan dengan masukan berupa citra dan keluaran berupa citra [11]. Berdasarkan pembuatannya citra dapat dibagi menjadi dua bagian yaitu citra kontinu dan citra diskrit. Citra kontinu merupakan citra yang dihasilkan dari sistem optik yang menerima sinyal analog. Sedangkan citra diskrit yaitu hasil dari citra kontinu yang sebelumnya telah melalui proses digitalisasi, jadi bisa dikatakan bahwa citra diskrit merupakan citra digital [12]. Representasi citra digital dapat dilihat pada Gambar 3.



Gambar 3. Representasi Citra Digital

#### 2.3.1 Citra RGB

Citra RGB merupakan citra yang memiliki 3 komponen warna pada setiap pixelnya yaitu R (*Red/Merah*), G (*Green/Hijau*), dan B (*Blue/Biru*). Pada setiap komponen warnanya menggunakan 8 bit (bernilai dari 0 sampai dengan 255) [13]. Hal ini dapat memberikan kemungkinan total warna hingga 255 x 255 x 255 atau 16.581.375 warna [14]. Representasi citra RGB dapat dilihat pada Gambar 4 [14].

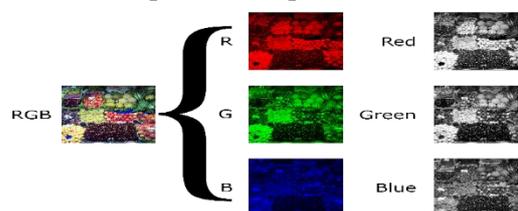
<b>Yellow</b> R = 255 G = 255 B = 0	<b>Orange</b> R = 255 G = 102 B = 0	<b>Green</b> R = 0 G = 255 B = 0
<b>Cyan</b> R = 0 G = 255 B = 255	<b>Violet</b> R = 204 G = 102 B = 204	<b>White</b> R = 255 G = 255 B = 255
<b>Black</b> R = 0 G = 0 B = 0	<b>Turquoise</b> R = 102 G = 255 B = 204	<b>Brown</b> R = 153 G = 102 B = 51

Gambar 4 Representasi RGB

#### 2.3.2 Citra Grayscale

Citra *grayscale* atau citra keabuan merupakan citra yang setiap pixelnya mewakili derajat keabuan dengan kisaran nilai 0 (hitam) sampai 255 (putih) [13]. Nilai pada kisaran 0 sampai 255 memiliki arti setiap piksel berukuran 8 bit atau 1 byte. Citra yang ditampilkan terdiri atas warna abu-abu, dimana warna putih intensitas warnanya lebih kuat dibandingkan dengan warna hitam. Citra *grayscale* bisa didapat dengan cara mengkonversikan citra RGB menjadi citra *grayscale* [15].

Adapun citra hasil konversi RGB menjadi *grayscale* dapat dilihat pada Gambar 5 [15].



Gambar 5 Hasil Konversi RGB ke Grayscale

Persamaan yang digunakan untuk mengubah nilai RGB menjadi Citra *grayscale* dapat dilihat pada Persamaan 1.

$$Grayscale = 0.299R + 0.5870G + 0.1140B \quad (1)$$

Keterangan:

R = nilai piksel merah

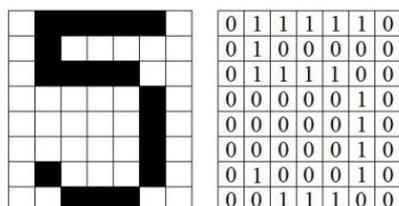
G = nilai piksel hijau

B = nilai piksel biru

#### 2.3.3 Citra Biner

Citra biner adalah citra dimana setiap *pixel*

hanya dinyatakan dengan sebuah nilai dari dua buah kemungkinan yaitu nilai 0 (hitam) dan 1 (putih)[sitasi]. Citra biner bisa didapatkan melalui konversi citra *grayscale* dengan operasi *thresholding*. *Thresholding* merupakan operasi pengelompokkan nilai derajat keabuan pada setiap *pixel* kedalam dua kelas yaitu hitam dan putih [13]. Representasi citra biner dapat dilihat pada Gambar 6 [13].



Gambar 6 Citra Biner

*Thresholding* biasanya digunakan pada segmentasi citra. Proses tersebut dilakukan untuk memisahkan antara *foreground* (objek yang diinginkan) dengan *background* (objek yang tidak diinginkan). *Foreground* direpresentasikan oleh warna putih (1) jika hasilnya melebihi nilai *threshold* yang telah ditentukan dan *background* direpresentasikan oleh warna hitam (0) jika hasilnya kurang dari nilai *threshold* yang sudah ditentukan [sitasi]. Berdasarkan nilai *threshold* (T) dapat dilihat pada Persamaan 2.

$$g(x,y) = \begin{cases} 1 & \text{jika } f(x,y) \geq T \\ 0 & \text{jika } f(x,y) < T \end{cases} \quad (2)$$

Keterangan :

$g(x,y)$  = Citra Biner

$f(x,y)$  = Citra *Grayscale*

T = Nilai *Threshold*

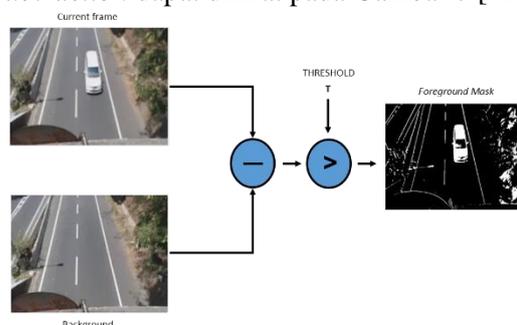
## 2.4 Background Subtraction

*Background Subtraction* adalah proses untuk menemukan objek pada gambar dengan cara membandingkan gambar yang ada dengan sebuah model latar belakang [17]. Proses untuk mendapatkan citra *foreground* yaitu dengan membandingkan *current frame* dengan *background model* kemudian citra tersebut akan di *thresholding*.

*Background Subtraction* biasanya digunakan dalam pendeteksian objek seperti manusia, kendaraan, tulisan atau bagian dari manusia itu sendiri. Secara umum cara kerja metode *background subtraction* adalah memisahkan objek yang akan dideteksi (*foreground*) dengan latar belakang (*background*).

Teknik *background subtraction* yaitu

metode yang digunakan untuk mendeteksi suatu objek bergerak dari sebuah video yang memiliki *background statis*. *Background* merupakan sejumlah *pixel* dari sebuah citra statis tak bergerak, sedangkan *foreground* yaitu semua objek yang ada selain *background*. *Foreground* bisa didapat ketika *background* telah ditentukan. Proses dari *Background Subtraction* dapat dilihat pada Gambar 7 [12].



Gambar 7 Proses *Background Subtraction*

Proses untuk mendapatkan citra *foreground* yaitu dengan membandingkan *current frame* dengan *background model* kemudian citra tersebut akan di *thresholding* [17]. Ada 3 tahapan dalam penerapan metode *background subtraction* yaitu:

### 1. Pre-processing

Pada tahap ini citra yang telah di akuisisi sebelumnya kemudian di proses secara sederhana seperti konversi format, perubahan ukuran gambar dan lain-lain.

### 2. Background Modelling

Pada tahap ini bertujuan untuk menentukan citra *background* yang statis yang kemudian nantinya akan di bandingkan dengan citra *foreground*.

### 3. Foreground Detection

Pada tahap ini bertujuan untuk menentukan citra *foreground* yang akan deteksi dengan cara mengekstraksi citra *foreground* dari *background*. Secara sederhana hal ini dapat dilakukan dengan menggunakan Persamaan 3:

$$f(x,y) = \begin{cases} 1, & \text{jika } |I(x,y,t) - I(x,y,t-1)| > T \\ 0, & \text{jika } |I(x,y,t) - I(x,y,t-1)| \leq T \end{cases} \quad (3)$$

Keterangan:

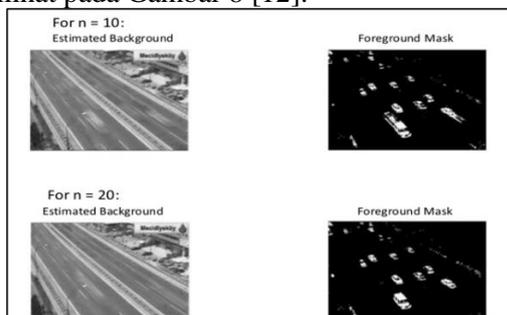
$I(x,y,t)$  = Frame *background model*

$I(x,y,t-1)$  = Frame objek bergerak

T = Nilai *Threshold*

Nilai *threshold* digunakan untuk proses pemisahan objek bergerak dari *background*, dimana jika hasil pengurangan antara matriks *background* dengan matriks objek bergerak

memenuhi syarat melebihi nilai ambang yang telah ditentukan maka akan di konversi menjadi nilai 1 (putih) dan dianggap sebagai foreground (objek bergerak). Sedangkan jika hasil pengurangan kurang atau sama dengan nilai ambang maka akan di konversi menjadi nilai 0 (hitam) dan dianggap sebagai *background* [4]. Hasil dari proses *background subtraction* dapat dilihat pada Gambar 8 [12].



Gambar 8 Hasil *Foreground* Deteksi

## 2.5 Operasi Morfologi Citra

Operasi Morfologi Citra merupakan bentuk operasi yang umum dilakukan pada bentuk citra biner (hitam-putih) untuk mengubah struktur bentuk objek yang terkandung dalam citra [19]. Tujuan penggunaan operasi morfologi citra yaitu dapat memperbaiki bentuk objek citra agar mendapatkan fitur-fitur yang lebih akurat ketika analisis dilakukan. Nilai setiap *pixel* didapat dari citra digital dengan cara proses perbandingan antara *pixel* yang bersesuaian pada citra digital masukkan dengan *pixel* sekitarnya [13].

Pengurangan dan penambahan *pixel* tergantung bentuk dari elemen penstruktur yang digunakan dalam mengolah citra. Elemen penstruktur merupakan himpunan kecil yang dapat digunakan untuk meneliti citra dalam pembelajaran propertinya. Elemen penstruktur memiliki ukuran yang lebih kecil dibandingkan dengan citra yang akan di proses [9].

Ada beberapa operasi dalam operasi morfologi, yaitu diantaranya sebagai berikut:

### 1. Operasi Dilasi

Operasi dilasi adalah operasi penambahan atau penebalan pada objek suatu gambar, sehingga citra tersebut akan mengalami pelebaran, hal ini dikarenakan citra yang di tambah *pixel*nya mengikuti bentuk dari elemen pestruktur yang digunakan. Persamaan dari

operasi dilasi dapat dilihat pada Persamaan 4.

$$D(A, B) = A \oplus B \quad (4)$$

Keterangan:

A = Citra masukan

B = *Structuring Element*

$A \oplus B$  = A didilasikan ke B

### 2. Operasi Erosi

Operasi erosi merupakan proses pengurangan *pixel* pada batas antar objek dalam suatu citra digital. Jumlah *pixel* dapat ditambahkan atau dikurangkan dari batas objek pada citra digital masukan tergantung dari bentuk dan ukuran penstruktur elemen yang dipakai. Persamaan operasi erosi dapat dilihat pada Persamaan 5.

$$E(A, B) = A \ominus B \quad (5)$$

Keterangan:

A = Citra masukan

B = *Structuring Element*

$A \ominus B$  = A dierosikan ke B

### 3. Operasi *Opening*

*Opening* yaitu kombinasi antara operasi erosi dan dilasi dimana fungsi erosi pada operasi *opening*, menghilangkan objek-objek kecil sedangkan dilasi berfungsi untuk mengatasi penurunan ukuran objek yang di akibatkan oleh operasi erosi sebelumnya. Persamaan operasi *opening* dapat dilihat pada Persamaan 6.

$$A \circ B = (A \ominus B) \oplus B \quad (6)$$

Keterangan:

$A \circ B$  = Proses *Opening*

$(A \ominus B) \oplus B$  = Hasil Erosi Dilasi ke B

### 4. Operasi *Closing*

*Closing* merupakan operasi *morfologi* penggabungan antara operasi dilasi dan erosi. Dimana fungsi dari operasi *closing* yaitu memperbesar batas luar dari objek *foreground* dan juga menutup lubang kecil yang terletak di tengah objek. Persamaan operasi *closing* dapat dilihat pada Persamaan 7.

$$A \bullet B = (A \oplus B) \ominus B \quad (7)$$

Keterangan:

$A \bullet B$  = Proses *Closing*

$(A \oplus B)$  = Dilasi A terhadap B

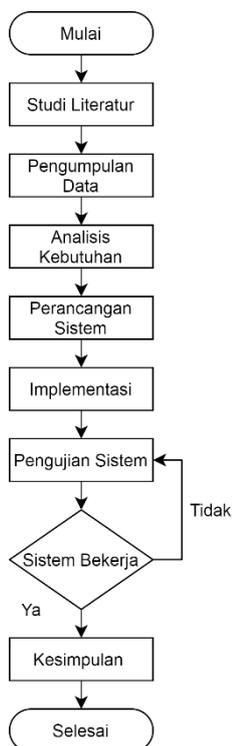
$(A \oplus B) \ominus B$  = Hasil dilasi dierosikan ke B

Adapun penggunaan operasi morfologi pada penelitian ini menggunakan 3 operasi morfologi antara lain operasi dilasi, operasi

*opening* dan operasi *closing*.

### 3. METODE PENELITIAN

Metode penelitian yang dilakukan dalam penelitian ini dapat dilihat pada Gambar 9.



Gambar 9 Diagram Alir Metode Penelitian.

#### 3.1 Studi Literatur

Studi literatur yang dilakukan bertujuan untuk mendapatkan informasi yang sesuai dengan topik permasalahan penelitian yang dilakukan. Pada penelitian ini, studi literatur yang dilakukan untuk memperoleh teori-teori pendukung yang berhubungan dengan pembuatan aplikasi perhitungan kendaraan mobil. Teori-teori pendukung yang diterapkan pada penelitian ini, berasal dari beberapa sumber literatur antara lain, jurnal penelitian, buku, tutorial-tutorial yang berkaitan dengan sistem aplikasi perhitungan kendaraan mobil, seperti pengolahan citra, dan pemrograman *python* dan *opencv*.

#### 3.2 Metode Pengumpulan Data

Tahap pengumpulan data dalam pembuatan sistem ini yaitu dengan mengumpulkan beberapa video objek bergerak yang berasal dari kamera cctv atau kamera statis, yang nantinya data tersebut dapat dijadikan sebagai objek observasi. Video yang

di akuisisi merupakan video yang berasal dari jalan raya Ahmad Yani Pontianak. Video tersebut dikumpulkan kemudian dijadikan dasar pengetahuan dalam penelitian ini. Adapun data video yang dikumpulkan pada penelitian ini sebanyak 30 data video, dimana dari data 30 video tersebut terbagi menjadi dua format video yaitu 2 menit dan 3 menit serta memiliki *frame per second* 30fps. Masing-masing data video tersebut memiliki tiga kategori yaitu normal, gelap, dan terang. Penggunaan kondisi cahaya normal, gelap, dan terang di pengaruhi oleh *exposure* pada kamera yang digunakan. Rentang nilai *exposure* pada kamera di mulai dari -1 hingga +1. Pada kondisi cahaya normal nilai *exposure* yang digunakan bernilai 0, pada kondisi gelap bernilai -1, dan pada kondisi terang bernilai +1.

#### 3.3 Analisa Kebutuhan

Pada tahap ini dilakukan analisis kebutuhan untuk membuat aplikasi perhitungan kendaraan mobil. Analisis kebutuhan perangkat keras yaitu kamera statis 16MP, Resolusi Video 640x360p, *Tripod*, *Processor Core i3* 3.40Ghz, Penyimpanan 5 GB. Analisis kebutuhan perangkat lunak yaitu video 360p, *Python* 3.8.0, *OpenCv* 4.4.0, *Balsamiq Mock-Up*, *Google Chrome*, *Xampp* 3.2.4, dan *Sublime Text* 3.0.

#### 3.4 Perancangan Sistem

Pada tahap ini sistem akan dibangun menggunakan diagram alir atau *flowchart* pada rancangan perangkat lunak. Pada perancangan *database* akan menggunakan *Data Flow Diagram (DFD)*, *Entity Relationship Diagram (ERD)*, dan perancangan tabel *database*. Pada perancangan antarmuka akan dibuat dalam bentuk *mockup* aplikasi. Sedangkan untuk perancangan pengujian akan dilakukan menggunakan pengujian *Black Box*.

#### 3.5 Implementasi

Pada tahap ini sistem yang dirancang akan dibangun berdasarkan perancangan yang ada. Adapun tahapan implementasi antara lain pembuatan aplikasi *backend* menggunakan *python*, implementasi metode, pembuatan *database*, dan pembuatan antarmuka website. Setelah selesai tahap implementasi, maka tahapan berikutnya, yaitu tahap pengujian.

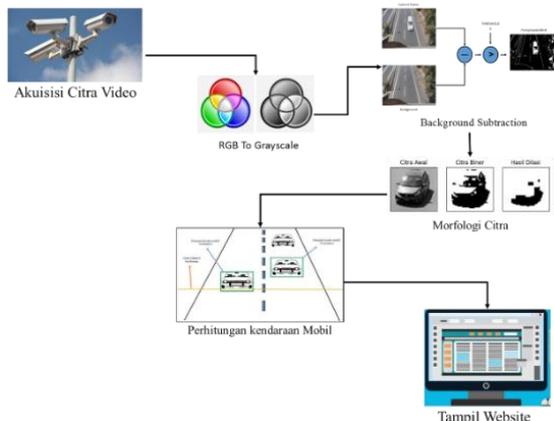
### 3.6 Pengujian

Pada tahap pengujian, sistem akan diuji menggunakan pengujian *blackbox* dan pengujian algoritma *Background Subtraction*. Pengujian *blackbox* dilakukan berdasarkan kegunaan fungsionalitas dari aplikasi, dengan cara menguji antara inputan dan keluaran apakah sesuai dengan yang diharapkan, baik tampilan aplikasi, dan fungsi-fungsi yang ada pada aplikasi. Sedangkan pengujian algoritma *Background Subtraction* dilakukan dengan cara membandingkan perhitungan yang dilakukan oleh sistem dengan data pengamatan manual.

## 4. HASIL DAN PEMBAHASAN

### 4.1 Arsitektur Sistem

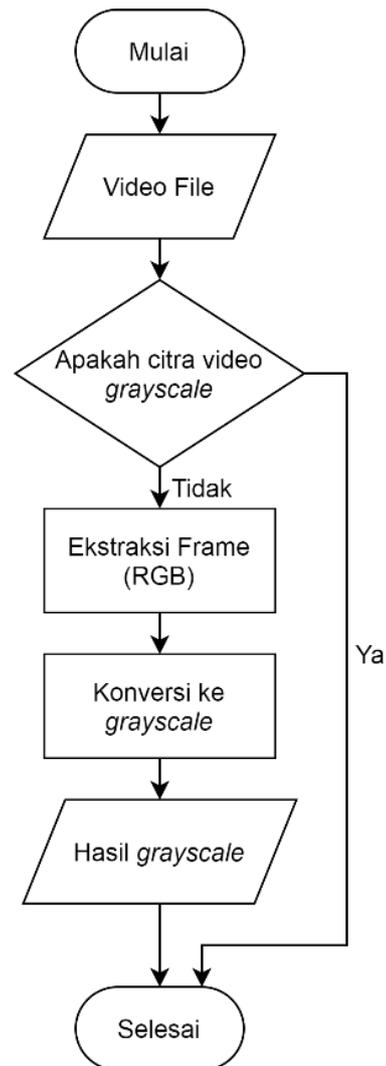
Arsitektur sistem yang digunakan pada penelitian ini, terdiri dari beberapa tahapan antara lain tahap *pre-processing*, *background subtraction*, operasi morfologi citra, perhitungan kendaraan, implementasi pada halaman *website*. Adapun gambaran umum dari arsitektur sistem pada perhitungan kendaraan mobil dapat dilihat pada Gambar 10.



Gambar 10 Gambaran umum arsitektur sistem

#### 4.1.1 Flowchart Pre-processing

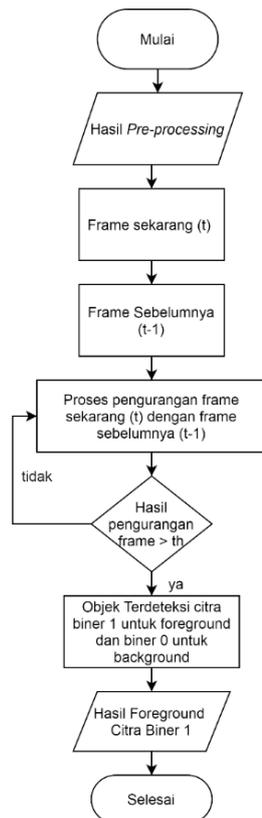
Pada Gambar 11 merupakan bentuk *flowchart* dari proses *Pre-processing*, dimana pada proses ini mengubah yang tadinya citra RGB menjadi citra *Grayscale*.



Gambar 11 Proses *Pre-processing*

#### 4.1.2 Flowchart Background Subtraction

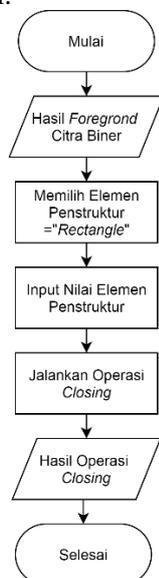
Pada Gambar 12 merupakan bentuk *flowchart* dari proses *background subtraction*. Dimana pada proses ini menggunakan pengurangan antar *frame* dalam mendapatkan citra hasil *background subtraction*. Hasil pengurangan *frame* tersebut berupa citra biner 1 sebagai *foreground* dan 0 sebagai *citra background*.



Gambar 12 Proses *Background Subtraction*

#### 4.1.3 Flowchart Operasi Morfologi Citra

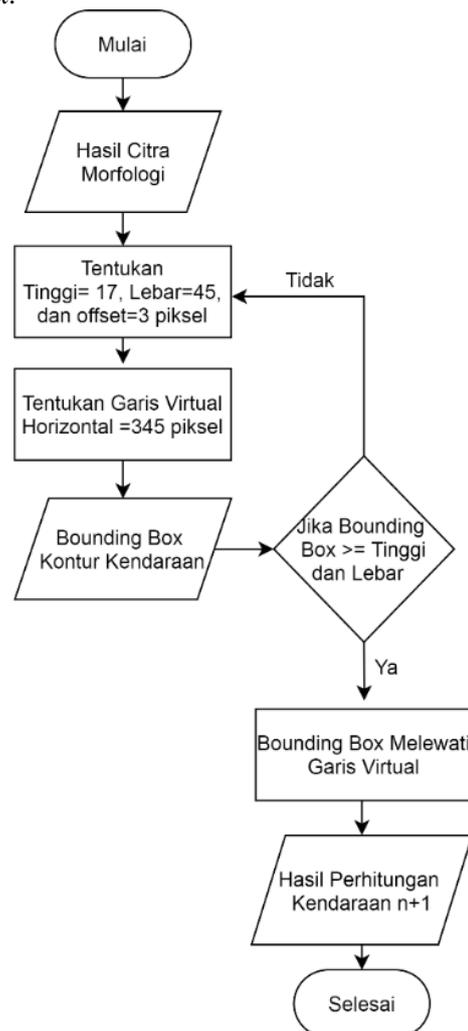
Pada Gambar 13 merupakan bentuk *flowchart* dari operasi morfologi citra yang memiliki 3 tahapan antara lain *closing*, *opening*, dan dilasi. Dimana 3 tahapan dilakukan agar *noise* yang ada dapat dikurangi. Dengan kurangnya *noise* diharapkan dapat memberikan hasil yang lebih optimal dalam perhitungan kendaraan mobil.



Gambar 13 Proses Morfologi Citra *Closing*

#### 4.1.4 Flowchart Perhitungan Kendaraan

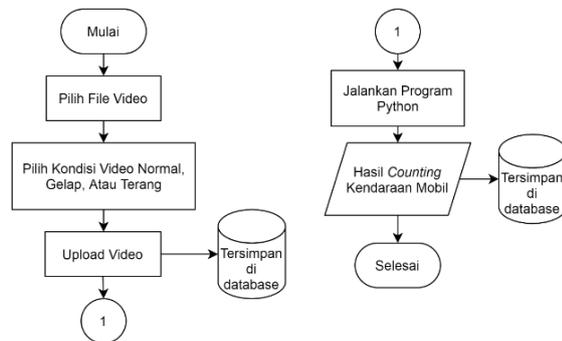
Pada Gambar 14 merupakan bentuk *flowchart* dari proses perhitungan kendaraan mobil, dimana pada proses ini adalah menentukan tinggi, lebar, dan offset piksel sebesar 17, 45, dan 3 piksel. Setelah menentukan tinggi, lebar, dan nilai *offset* selanjutnya menentukan tinggi garis *virtual*. Fungsi dari garis *virtual* ini digunakan sebagai parameter bahwa mobil dihitung sebagai kendaraan. Setelah garis *virtual* kendaraan dibuat langkah selanjutnya adalah menghitung kendaraan mobil dengan cara, mobil melewati garis *virtual* yang ditandai dengan *bounding box*.



Gambar 14 Proses Perhitungan Kendaraan

#### 4.1.5 Flowchart Halaman Website

Pada Gambar 15 merupakan bentuk *flowchart* dari implementasi pada halaman *website*, dimana proses yang dilakukan yaitu penyimpanan ke *database*.



Gambar 15 Proses Aplikasi pada Halaman Website

#### 4.2 Hasil Pre-processing

Pada proses ini dilakukan proses perhitungan konversi citra *grayscale*. Proses perhitungan *pre-processing* menggunakan persamaan 1. Adapun hasil perhitungan citra grayscale dapat dilihat pada Tabel 1 dan Tabel 2.

Tabel 1 Citra *Grayscale* Frame Sekarang

Grayscale									
175	156	151	152	154	153	155	153	153	153
158	151	153	154	152	152	151	153	154	154
154	151	151	156	153	154	156	152	153	154
156	154	154	177	181	176	181	179	181	181
153	154	153	178	179	178	179	180	181	179
154	155	155	179	179	179	180	179	180	179
155	156	156	179	180	179	180	181	180	179
155	156	155	179	179	178	179	181	180	180
154	155	154	178	178	177	178	179	179	180
153	154	153	178	179	178	179	180	180	182

Proses perhitungan *pre-processing frame* sekarang dapat dilihat pada persamaan berikut:

$$\begin{aligned} \text{Grayscale} &= 0.299(105) + 0.587(195) + 0.114(255) \\ &= 31.395 + 114.465 + 29.07 \\ &= 175 \end{aligned}$$

Tabel 2 Citra *Grayscale* Frame Sebelumnya

Grayscale									
179	180	181	181	182	183	184	185	184	184
180	180	181	181	182	183	184	185	185	185
180	180	182	182	182	183	184	185	185	185
180	181	182	182	182	183	184	185	185	185
181	181	182	182	182	183	184	185	185	185
181	182	182	182	182	183	184	185	185	186
182	182	182	182	182	183	185	184	185	185
182	182	182	182	182	183	185	185	185	186
182	182	182	182	182	184	185	186	186	187
182	182	182	182	182	184	185	185	186	186

Proses perhitungan *pre-processing frame* sebelumnya dapat dilihat pada persamaan berikut:

$$\begin{aligned} \text{Grayscale} &= 0.299(150) + 0.587(180) + 0.114(253) \\ &= 44.85 + 105.66 + 28.842 \\ &= 179 \end{aligned}$$

#### 4.3 Hasil Background Subtraction

Pada tahap perhitungan *Background Subtraction*, nilai *Grayscale* yang telah didapatkan sebelumnya akan dikurangi setiap pikselnya dengan menggunakan Persamaan 2.

Sehingga:

$$\begin{aligned} f(x, y) &= I(x, y, t) - I(x, y, t - 1) > T \\ &= 175 - 179 \\ &= -4 \text{ nilai yang diambil absolute} \end{aligned}$$

$I(x, y, t)$  = Frame Sekarang

$I(x, y, t - 1)$  = Frame Sebelumnya

$T$  = Nilai *Threshold*

Perhitungan tersebut merupakan proses untuk mendapatkan *frame foreground*. Pada proses *Background Subtraction*, frame sekarang atau bisa disebut sebagai frame *background* akan dikurangi dengan frame sebelumnya. Setelah kedua frame tersebut dikurangi, maka langkah selanjutnya membandingkan hasil pengurangan frame tersebut dengan nilai *threshold* yang telah ditentukan.

Adapun hasil perhitungan pengurangan antar frame dapat dilihat pada Tabel 3 dan Tabel 4.

Tabel 3 Pengurangan Antar *Frame*

Pengurangan Frame Sekarang - Frame Sebelumnya									
4	24	30	29	28	30	29	32	31	31
22	29	28	27	30	31	33	32	31	31
26	29	31	26	29	29	28	33	32	31
24	27	28	5	1	7	3	6	4	4
28	27	29	4	3	5	5	5	4	6
27	27	27	3	3	4	4	6	5	7
27	26	26	3	2	4	5	3	5	6
27	26	27	3	3	5	6	4	5	6
28	27	28	4	4	7	7	7	7	7
29	28	29	4	3	6	6	5	6	4

Tabel 4 Hasil Proses *Threshold*

Jika Nilai Th >25 Maka Bernilai 1									
0	0	1	1	1	1	1	1	1	1
0	1	1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1	1
0	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0

#### 4.4 Hasil Morfologi Citra

Pada tahap operasi morfologi citra, hasil dari nilai *threshold* akan dijadikan dasar sebagai perhitungan operasi morfologi citra. Adapun operasi morfologi citra yang digunakan pada penelitian ini antara lain, operasi closing, operasi opening, dan operasi dilasi. Hasil operasi morfologi dapat dilihat pada Tabel 5.

Tabel 5 Citra Asli

Citra Asli										
x/y	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	1	1	1	1	1	1
2	0	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	0	1	1	0	0	0	0	0	0	0
5	1	1	1	0	0	0	0	0	0	0
6	1	1	1	0	0	0	0	0	0	0
7	1	1	1	0	0	0	0	0	0	0
8	1	1	1	0	0	0	0	0	0	0
9	1	1	1	0	0	0	0	0	0	0
10	1	1	1	0	0	0	0	0	0	0

Operasi morfologi pertama yaitu operasi closing, dimana citra asli terlebih dahulu di dilasikan dengan *structuring element* 3x3, setelah itu hasil dilasi tersebut kemudian di erosikan kembali dengan *structuring element* 3x3. Sehingga terjadi penambahan piksel pada Tabel 6 dan pengurangan piksel pada Tabel 7.

Tabel 6 Hasil Proses Closing (Dilasi)

Closing (Dilasi)										
x/y	1	2	3	4	5	6	7	8	9	10
1	0	1	1	1	1	1	1	1	1	1
2	1	1	1	1	1	1	1	1	1	1
3	1	1	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	0	0	0	0	0	0
6	1	1	1	1	0	0	0	0	0	0
7	1	1	1	1	0	0	0	0	0	0
8	1	1	1	1	0	0	0	0	0	0
9	1	1	1	1	0	0	0	0	0	0
10	1	1	1	1	0	0	0	0	0	0

Tabel 7 Hasil Proses Closing (Erosi)

Closing (Erosi)										
x/y	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	1	1	1	1	1	1	1	1	0
3	0	1	1	1	1	1	1	1	1	0
4	0	1	1	1	0	0	0	0	0	0
5	0	1	1	0	0	0	0	0	0	0
6	0	1	1	0	0	0	0	0	0	0
7	0	1	1	0	0	0	0	0	0	0
8	0	1	1	0	0	0	0	0	0	0
9	0	1	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Setelah dilakukan proses closing, tahap selanjutnya yaitu proses opening, dimana pada proses opening ini hasil dari proses closing tadi akan di erosikan terlebih dahulu dan kemudian di dilasikan kembali dengan *structuring element* 3x3. Adapun hasil dari erosi dari proses opening dapat dilihat pada Tabel 8, dan hasil dilasi dari proses opening dapat dilihat pada Tabel 9.

Tabel 8 Hasil Proses Opening (Erosi)

Opening (Erosi)										
x/y	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0
3	0	0	1	1	0	0	0	0	0	0
4	0	0	1	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Tabel 9 Hasil Proses Opening (Dilasi)

Opening (Dilasi)										
x/y	1	2	3	4	5	6	7	8	9	10
1	0	0	0	0	0	0	0	0	0	0
2	0	0	1	1	0	0	0	0	0	0
3	0	1	1	1	1	0	0	0	0	0
4	0	1	1	1	0	0	0	0	0	0
5	0	0	1	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

Operasi morfologi yang terakhir pada penelitian ini yaitu operasi dilasi. Pada proses ini hasil dari operasi opening akan di dilasikan kembali dengan *structuring element* 3x3. Proses tersebut dilakukan agar dapat mempertebal piksel dari citra mobil yang akan dideteksi. Adapun hasil dari proses dilasi ini dapat dilihat pada Tabel 10.

Tabel 10 Hasil Proses Dilasi

Dilasi										
x/y	1	2	3	4	5	6	7	8	9	10
1	0	0	1	1	0	0	0	0	0	0
2	0	1	1	1	1	0	0	0	0	0
3	1	1	1	1	1	1	0	0	0	0
4	1	1	1	1	1	0	0	0	0	0
5	0	1	1	1	0	0	0	0	0	0
6	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0

#### 4.5 Hasil Perhitungan Kendaraan Mobil

Pengujian sistem dilakukan secara langsung menggunakan data video. Data video yang digunakan sebanyak 30 data video, dimana pada pengujian video menggunakan 3 kondisi cahaya yaitu normal, gelap, dan terang. Masing masing kondisi video tersebut terdapat data video yang memiliki 2 durasi waktu yaitu 2 menit dan 3 menit. Adapun hasil pengujian tersebut dapat dilihat pada Tabel 11, Tabel 12, dan Tabel 13.

Tabel 11 Hasil pengujian Video Normal 30fps

No	Nama Video	Waktu (Menit)	Manual	Sistem	Error (%)	Akurasi (%)
1	Video1.mp4	2	28	26	7.14	92.86
2	Video2.mp4	2	20	17	15	85
3	Video3.mp4	2	27	23	14.81	85.19
4	Video4.mp4	2	20	15	25	75
5	Video5.mp4	2	16	17	6.25	93.75
6	Video6.mp4	3	40	48	20	80
7	Video7.mp4	3	55	63	14.55	85.45
8	Video8.mp4	3	40	46	15	85
9	Video9.mp4	3	25	21	16	84
10	Video10.mp4	3	95	91	4.21	95.79
Total Kendaraan Mobil			366	367	13.80	86.20

Pengujian pada Tabel 11 mendapatkan nilai rata-rata akurasi sebesar 86.20% dan nilai *error* sebesar 13.80%. Hasil yang didapat pada sistem melebihi hasil pengamatan manual, hal tersebut dikarenakan ramainya lalu lintas kendaraan yang melintas, sehingga objek kendaraan yang melintas dapat terhitung lebih dari satu kendaraan.

Tabel 12 Hasil Pengujian Data Video Kondisi Gelap 30fps

No	Nama Video	Waktu (Menit)	Manual	Sistem	Error (%)	Akurasi (%)
1	Video1.mp4	2	20	12	40	60
2	Video2.mp4	2	21	11	47.62	52.38
3	Video3.mp4	2	36	27	25	75
4	Video4.mp4	2	25	19	24	76
5	Video5.mp4	2	30	20	33.33	66.67
6	Video6.mp4	3	39	20	48.72	51.28
7	Video7.mp4	3	48	37	22.92	77.08
8	Video8.mp4	3	33	24	27.27	72.73
9	Video9.mp4	3	29	19	34.48	65.52
10	Video10.mp4	3	48	33	31.25	68.75
Total Kendaraan Mobil			329	222	33.46	66.54

Pengujian pada Tabel 12 mendapatkan nilai rata-rata akurasi sebesar 66.54% dan nilai *error* sebesar 33.46%. Hasil pengamatan manual dengan hasil sistem pada kondisi gelap menunjukkan banyak kendaraan tidak dapat terhitung dengan tepat. Hal tersebut dikarenakan objek pada video tersebut memiliki kondisi cahaya yang kurang, sehingga objek kendaraan menjadi tersamarkan dan tidak dapat terhitung oleh sistem sebagai kendaraan mobil.

Tabel 13 Hasil Pengujian Data Video Kondisi Terang 30fps

No	Nama Video	Waktu (Menit)	Manual	Sistem	Error (%)	Akurasi (%)
1	Video1.mp4	2	38	25	34.21	65.79
2	Video2.mp4	2	27	33	22.22	77.78
3	Video3.mp4	2	32	25	21.88	78.12
4	Video4.mp4	2	20	18	10	90

5	Video5.mp4	2	27	27	0	100
6	Video6.mp4	3	39	16	58.97	41.03
7	Video7.mp4	3	49	37	24.49	75.51
8	Video8.mp4	3	33	30	9.09	90.91
9	Video9.mp4	3	50	43	14	86
10	Video10.mp4	3	45	32	28.89	71.11
Total Kendaraan Mobil			360	286	22.38	77.63

Pengujian pada Tabel 13 mendapatkan nilai rata-rata akurasi sebesar 77.63% dan nilai *error* sebesar 22.38%. Hasil yang didapat pada kondisi terang menunjukkan bahwa kendaraan mobil dapat terhitung lebih baik dibandingkan dengan kondisi gelap. Namun ada kelemahan pada pengujian kondisi terang, dimana ada banyak objek yang bergerak secara tak beraturan, sehingga sistem mengira bahwa objek yang bergerak merupakan objek kendaraan. Adapun hasil dari objek citra yang telah diuji dapat dilihat pada Gambar 16.



Gambar 16 Hasil Deteksi Kendaraan Mobil

#### 4.6 Pembahasan

Berdasarkan pada penelitian, pengujian yang telah dilakukan diuji data sebanyak 30 video dengan durasi video 2 menit dan 3 menit, serta dari data tersebut memiliki 3 kondisi normal, gelap, dan terang. Adapun tingkat akurasi yang didapat pada kondisi normal sebesar 86.20%, gelap 66.54%, dan terang 77.63%. Adapun kelemahan dalam mendeteksi kendaraan mobil diantaranya adanya pengaruh cahaya, dan padatnya kendaraan yang melintas. Pengaruh cahaya yang dimaksud yaitu kondisi cahaya terang dan gelap, dimana kondisi cahaya terlalu terang objek yang dideteksi menjadi tidak stabil. Sedangkan pada kondisi cahaya gelap objek yang dideteksi menjadi tersamarkan, dikarenakan kurangnya intensitas cahaya pada objek kendaraan mobil. Pengaruh lainnya diakibatkan padatnya kendaraan yang melintas, dimana objek yang berdekatan akan dihitung sebagai 1 objek kendaraan.

## 5. KESIMPULAN

Berdasarkan pengujian yang telah dilakukan pada sistem ini dapat diambil kesimpulan sebagai berikut:

1. Kendaraan mobil dapat dideteksi menggunakan metode *Background Subtraction* ditandai dengan bentuk kontur mobil. Sedangkan untuk menghitung jumlah kendaraan mobil menggunakan minimum lebar dan tinggi kontur mobil yang melewati garis *virtual horizontal*.
2. Pada pengujian yang telah dilakukan didapatkan hasil berupa nilai rata-rata tingkat akurasi pada kondisi cahaya normal sebesar 86.20%, kondisi cahaya gelap 66.54%, dan kondisi cahaya terang 77.63%.
3. Berdasarkan pengujian kondisi video yang digunakan, kondisi optimal didapatkan pada kondisi cahaya normal dengan tingkat rata-rata akurasi sebesar 86.20%.

## 6. SARAN

Terdapat beberapa saran untuk penelitian ini sebagai berikut:

1. Sistem dapat dikembangkan menjadi sistem yang dapat mengklasifikasi jenis kendaraan baik itu kendaraan roda empat maupun roda dua.
2. Pada penelitian selanjutnya sistem ini dapat dilakukan pengujian video 60fps.
3. Pada penelitian selanjutnya, sistem ini dapat dikembangkan berbasis android.
4. Pada penelitian selanjutnya sistem ini dapat dikembangkan menggunakan metode Interpolasi Linear dan *Histogram Equalization*.

## DAFTAR PUSTAKA

- [1] S. Widodo, "Pontianak Tribunnews," 2018. [Online]. Available: <https://pontianak.tribunnews.com/2018/04/11/kendaraan-di-kota-pontianak-tembus-satu-juta-lho-ini-jumlah-detailnya>.
- [2] D. P. Putra, "Analisis Hubungan Antara Kemacetan Dan Polusi Udara Di Jalan Sultan Abdurahman Pontianak," *Fakultas Teknik Sipil Universitas Tanjungpura*, p. 2, 2016.
- [3] N. N. Putri, *APLIKASI OBJEK BERGERAK*, Depok: Teknik Informatika Universitas Gunadarma, 2016.
- [4] K.adi, *Automatic Vehicle counting using background subtraction method on gray scale images and morphology operation*, Semarang: Diponegoro University, 2018.
- [5] P. Apriani, *Aplikasi Deteksi Objek Bergerak Berbasis Citra Dengan Metode Background Subtraction dan Blob Detection (Studi Kasus: Mami Mart Kubu Raya)*, Pontianak: Universitas Tanjungpura, 2019.
- [6] C. A. U. Wahono, "Sistem Monitoring Pendeteksi Komponen Kaleng Pecah dengan Sensor LDR Menggunakan Arduino Nano Berbasis Web Server (Studi Kasus PT Artawena Sakti Gemilang)," *Teknik Informatika*, 2017.
- [7] S. Sukirman, *Dasar-Dasar Perencanaan Geometrik Jalan*, Bandung: Nova, 1994.
- [8] D. Putra, *Pengolahan Citra Digital*, Yogyakarta: Andi, 2010.
- [9] Sutoyo and Dkk, *Pengolahan Citra Digital*, Yogyakarta: Andi, 2009.
- [10] A. Basuki, *Pengolahan Citra Digital Menggunakan Visual Basic*, Yogyakarta: Graha Ilmu, 2005.
- [11] R. Munir, *Pengolahan Citra Digital dengan Pendekatan Algoritmik*, Bandung: Informatika, 2004.
- [12] [Online]. Available: <https://elib.unikom.ac.id/download.php?id=322478>.
- [13] [Online]. Available: <https://pemrogramanmatlab.com/pengolahan-citra-digital/>.
- [14] A. K. d. A. Susanto, *Teori dan Aplikasi Pengolahan Citra*, Yogyakarta: Andi, 2012.
- [15] E. & Ardianto, "Implementasi Metode Image Subtraction dan Metode Regionprops untuk Mendeteksi Jumlah Objek Berwarna RGB pada File Video," *Teknik Informatika*, p. 94, 2013.
- [16] A. Kadir, *Pengolahan Citra, Teori dan Aplikasi*, Yogyakarta: Andi, 2013.