TUGAS AKHIR – KM184801

KLASIFIKASI DAN PENGHITUNGAN KENDARAAN BERGERAK PADA MALAM HARI DENGAN TEKNIK *PAIRING* LAMPU DEPAN

EKO ANDI PRASTIYO
NRP 06111440000080

Dosen Pembimbing :
Dr. Budi Setiyono, S.Si, MT

DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2019

TUGAS AKHIR – KM184801

KLASIFIKASI DAN PENGHITUNGAN KENDARAAN BERGERAK PADA MALAM HARI DENGAN TEKNIK *PAIRING* LAMPU DEPAN

EKO ANDI PRASTIYO
NRP 06111440000080

Dosen Pembimbing :
Dr. Budi Setiyono, S.Si, MT

DEPARTEMEN MATEMATIKA
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember
Surabaya 2019

FINAL PROJECT – KM184801

CLASSIFICATION AND COUNTING MOVING VEHICLES IN NIGHT TIME USING HEADLAMP PAIRING TECHNIQUE

EKO ANDI PRASTIYO
NRP 06111440000080

***Supervisor* :**
Dr. Budi Setiyono, S.Si, MT

DEPARTMENT OF MATHEMATICS
Faculty of Mathematics Computations and Data Science
Sepuluh Nopember Institute of Technology
Surabaya 2019

LEMBAR PENGESAHAN

KLASIFIKASI DAN PENGHITUNGAN KENDARAAN BERGERAK PADA MALAM HARI DENGAN TEKNIK *PAIRING* LAMPU DEPAN

CLASSIFICATION AND COUNTING MOVING VEHICLES IN NIGHT TIME USING HEADLAMP PAIRING TECHNIQUE

TUGAS AKHIR

Diajukan untuk memenuhi salah satu syarat
Untuk memperoleh gelar Sarjana Sains
Pada bidang studi Ilmu Komputer
Program Studi S-1 Departemen Matematika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember Surabaya

Oleh :
EKO ANDI PRASTIYO
NRP. 06111440000080

Menyetujui,
Dosen Pembimbing

Dr. Budi Setiyono, S.Si, MT
NIP. 19720207 199702 1 001

Mengetahui,
Ketua Departemen Matematika
FMKSD ITS

Dr. Imam Mukhlash, S.Si, MT
NIP. 19700831 199403 1 003

Surabaya, 14 Juni 2019

KLASIFIKASI DAN PENGHITUNGAN KENDARAAN BERGERAK PADA MALAM HARI DENGAN TEKNIK *PAIRING* LAMPU DEPAN

Nama Mahasiswa	:	EKO ANDI PRASTIYO
NRP	:	06111440000080
Departemen	:	Matematika FMKSD ITS
Pembimbing	:	Dr. Budi Setiyono, S.Si, MT

Abstrak

Pengumpulan data terkait kendaraan yang melintasi jalan raya dapat dilakukan dengan berbagai cara. Salah satu cara adalah dengan mendeteksi kendaraan melalui media video. Teknik deteksi kendaraan pada siang hari tidak bisa digunakan untuk deteksi kendaraan pada malam hari karena pada malam hari penerangan buruk menyebabkan banyak *noise*, gambar kamera juga memiliki kontras yang sangat rendah dan sensitivitas cahaya yang lemah. Pada Tugas Akhir ini dilakukan klasifikasi dan penghitungan kendaraan bergerak pada malam hari dengan menggunakan teknik *pairing* lampu depan kendaraan. Proses *pairing* diawali dengan pencarian kontur dan pengoneksian piksel kemudian membentuk *boundingrect* sehingga dapat ditentukan titik pusat dari lampu depan. Lampu depan di*pairing* jika titik pusat dua lampu depan memenuhi jarak horizontal dan vertikal tertentu serta memiliki perbedaan luas tertentu. Lampu yang di*pairing* diklasifikasi sebagai roda 4 dan yang tidak di*pairing* diklasifikasi sebagai roda 2. Dari uji coba pada empat video didapatkan hasil klasifikasi dan penghitungan dengan rata-rata akurasi sebesar 91,5%.

Kata-kunci : klasifikasi kendaraan, deteksi lampu depan, *pairing*, *boundingrect*

CLASIFICATION AND COUNTING MOVING VEHICLES IN NIGHT TIME USING HEADLAMP PAIRING TECHNIQUE

Name	:	EKO ANDI PRASTIYO
NRP	:	06111440000080
Department	:	Matematika FMKSD-ITS
Supervisor	:	Dr. Budi Setiyono, S.Si, MT

Abstract

Data collection related to vehicles crossing the highway can be done in various ways. One way is to detect vehicles through video media. Vehicle detection techniques during the day cannot be used for vehicle detection at night because bad night lighting causes a lot of noise, camera images also have very low contrast and weak light sensitivity. In this Final Project, moving vehicles classification and calculation are carried out at night using vehicle headlamps pairing techniques. The pairing process begins with the search for contours and connecting pixels then forming boundingrect so that the center point of the headlamps can be determined. The headlamps are paired if the center of the two headlights meet certain horizontal and vertical distances and have a certain area difference. The paired headlamps are classified as 4 wheels and which are not classified as 2 wheels. From the test results of four videos obtained calculations with an average accuracy of 91.5%.

Keywords **:** *vehicles classification, headlamp detection, pairing, boundingrect*

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillaahirobbil'aalamiin, segala puji dan syukur penulis panjatkan kehadirat Allah SWT yang telah memberikan limpahan rahmat, nikmat, taufik dan hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir dengan baik yang berjudul

"KLASIFIKASI DAN PENGHITUNGAN KENDARAAN BERGERAK PADA MALAM HARI DENGAN TEKNIK *PAIRING* LAMPU DEPAN"

sebagai salah satu syarat kelulusan Program Sarjana Departemen Matematika FMKSD Institut Teknologi Sepuluh Nopember (ITS) Surabaya.

Tugas Akhir ini dapat diselesaikan dengan baik berkat bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis menyampaikan ucapan terima kasih dan penghargaan kepada:

1. Bapak Dr. Budi Setiyono, S.Si, MT selaku dosen pembimbing yang telah memberikan motivasi dan pengarahan dalam penyelesaian Tugas Akhir ini.
2. Bapak Dr. Imam Mukhlash, S.Si, MT selaku Kepala Departemen Matematika FMKSD ITS yang telah memberikan dukungan dan motivasi selama perkuliahan hingga terselesaikannya Tugas Akhir ini.
3. Bapak Dr. Didik Khusnul Arif, S.Si, M.Si selaku Ketua Program Studi Sarjana dan Bapak Drs. Iis Herisman, M.Si selaku Sekretaris Program Studi Sarjana Departemen Matematika FMKSD ITS yang telah memberikan arahan akademik selama penulis kuliah di Departemen Matematika FMKSD ITS.

4. Bapak dan Ibu dosen penguji atas semua saran dan masukan yang telah diberikan.
5. Ibu Dr. Dra. Mardlijah, M.T selaku dosen wali yang telah memberikan arahan akademik selama penulis kuliah di Departemen Matematika FMKSD ITS.
6. Bapak dan Ibu dosen serta para staf Departemen Matematika ITS yang tidak dapat penulis sebutkan satu-persatu.
7. Kedua orang tua saya, Bapak Agung Prastijo dan Ibu Kartini atas dukungan dan semangat yang telah diberikan.
8. Keluarga AKSIOM14 dan sahabat-sahabat yang saya sayangi yang telah membantu dan memotivasi saya.
9. Seluruh pihak yang tidak dapat penulis sebutkan satu persatu, yang turut membantu dalam penyusunan Tugas Akhir ini.

Penulis menyadari bahwa dalam penyusunan Tugas Akhir ini masih terdapat banyak kekurangan. Oleh karena itu, kritik dan saran dari berbagai pihak yang bersifat membangun sangat diharapkan sebagai bahan perbaikan di masa yang akan datang.

Surabaya, 14 Januari 2019

Penulis

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

BAB I
PENDAHULUAN

Pada bab ini, dibahas mengenai latar belakang adanya penelitian ini. Selanjutnya, disebutkan rumusan masalah berdasarkan latar belakang yang telah dibahas beserta batasan-batasan masalah yang ada. Selain itu, disebutkan juga tujuan dan manfaat mengenai adanya penelitian ini dan sistematika penulisannya.

1.1 Latar Belakang Masalah

Di masa kini teknologi berkembang semakin pesat seiring dengan perkembangan zaman. Salah satunya teknologi yang berkembang dibidang transportasi. Pertumbuhan transportasi di Indonesia sangat tinggi, terutama transportasi darat. Berdasarkan data dari Badan Pusat Statistik jumlah kendaraan bermotor pada tahun 2017 mencapai 138.556.669 unit meningkat 9.275.590 unit dari tahun 2016 [1].

Pengumpulan data terkait kendaraan yang melintasi jalan raya dapat dilakukan dengan berbagai cara. Cara yang pernah dilakukan adalah secara manual dengan langsung melakukan perhitungan di suatu lokasi dengan kemampuan manusia ala kadarnya[2]. Namun, saat ini dengan berkembang pesatnya teknologi, perhitungan kendaraan yang melintasi suatu jalan raya dapat dilakukan secara online dengan alat yang beroperasi *real time*[3]. *Intelligent Transportation System* (ITS) atau Sistem Tranportasi Pintar merupakan penerapan teknologi, informasi dan komunikasi di bidang transportasi. ITS dapat memberikan manfaat pada sistem transportasi seperti mendapatkan informasi mengenai keadaan lalu lintas, mengurangi kemacetan dan antrian, meningkatkan sarana dan prasarana transportasi, mengurangi polusi lingkungan dan permasalahan lalu lintas lainnya.

Pengaplikasian ITS dalam sistem transportasi dapat dijadikan sebagai alat efektif sehingga dapat meningkatkan nilai mobilitas perjalanan dan mengurangi dampak permasalahan lalu lintas.

Pengambilan data secara *real time* harus menghadapi semua kondisi geografis yang terjadi pada saat pengambilan data seperti hujan, malam hari, bersalju dan kodisi alam lainnya [4]. Data informasi untuk sistem pemantauan kendaraan bergerak secara *real time* dapat diperoleh dari berbagai sumber seperti detektor lingkaran, detektor ultrasonik, sensor microwave, sensor radar atau kamera video [5]. Dengan adanya kemajuan teknologi terbaru yang berguna untuk teknik pengolahan citra. Kamera video yang telah ditemukan adalah cara yang efisien untuk mengumpulkan dan menganalisis data lalu lintas dikarenakan sistem kamera berbasis video yang lebih canggih dan kuat karena informasi yang berhubungan dengan urutan gambar yang hadir dalam video memungkinkan untuk mengidentifikasi dan mengklasifikasikan kendaraan dengan cara yang paling efektif [5].

Teknik pengolahan video untuk mendapat data dan dapat mendeteksi kendaraan di jalan raya telah dilakukan beberapa peneliti. Salah satunya yaitu pada penelitian Zamroji Hariyanto yang pendeteksiannya dilakukan pada saat pagi dan siang hari[6]. Penelitian ini menggunakan data berupa video yang berdurasi 1 menit dengan keadaan jalan raya pada siang hari. Penetitian Zamroji H yang berjudul Klasifikasi Jenis Kendaraan Bergerak Berbasis *Geometric Invariant Moment* mengklasifikasikan kendaraan menjadi 3 jenis kendaraan yaitu motor, mobil dan truk.

Teknik yang digunakan untuk deteksi kendaraan pada siang hari tidak bisa digunakan untuk deteksi kendaraan pada malam hari karena berbagai faktor[5]. Pada malam hari, penerangan yang buruk menyebabkan *noise* yang kuat sehingga meningkatkan kompleksitas dari pendeteksian. Refleksi bergerak dari lampu depan kendaraan bisa menyebabkan adanya kesalahan dalam

proses deteksi karena keambiguan antara *foreground* dan *background*. Selain itu pada malam hari gambar kamera memiliki kontras yang sangat rendah dan sensitivitas cahaya yang lemah sehingga menyulitkan teknik deteksi kendaraan pada waktu normal.

Lingkungan pada malam hari bisa dikelompokkan dalam dua jenis[5]. Pertama, pemandangan gelap dimana tidak ada lampu jalan dan satu-satunya yang terlihat adalah lampu depan kendaraan dan refleksinya. Lampu depannya tampak seperti gumpalan bulat terang yang kontras dengan lingkungan yang gelap. Kedua, pemandangan terang dimana jalanan diterangi oleh lampu seperti pada daerah perkotaan. Pada kondisi ini, *background*, pejalan kaki, dan objek lain terlihat dengan jelas sehingga kerumitan dalam mengekstraksi lampu depan kendaraan meningkat.

Ekstraksi lampu depan dari suatu kendaraan menghasilkan beberapa objek. Posisi masing-masing objek tersebut berdekatan sehingga dapat dipasangkan antara yang satu dengan yang lainnya dengan suatu kriteria tertentu. Jika dua objek berdekatan berhasil dipasangkan maka objek tersebut menjadi kandidat lampu depan kendaraan roda 4 sedangkan jika tidak berhasil dipasangkan maka menjadi kandidat lampu depan kendaraan roda 2. Oleh karena itu, penulis memilih tugas akhir yang dapat mengklasifikasi dan menghitung kendaraan pada malam hari dengan teknik *pairing* lampu depan.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka didapat rumusan masalah sebagai berikut,

1. Bagaimana menerapkan teknik *pairing* untuk mengklasifikasi suatu kendaraan yang melintasi jalan raya pada malam hari?

2. Bagaimana menghitung banyak kendaraan yang melintasi jalan raya pada malam hari?
3. Bagaimana mengembangkan program yang mampu mengklasifikasi dan menghitung kendaraan yang melintasi jalan raya pada malam hari melalui video?

1.3 Batasan Masalah

Batasan permasalahan dalam tugas akhir ini yaitu,

1. Pengklasifikasian dan penghitungan kendaraan dilakukan secara *offline* melalui media video
2. Video diambil pada waktu malam hari di lajur satu arah
3. Perekaman video saat keadaan cuaca normal
4. Teknik yang digunakan yaitu teknik *pairing*
5. Klasifikasi dua jenis kendaraan yaitu roda 2 dan roda 4

1.4 Tujuan

Tujuan yang ingin dicapai dalam tugas akhir ini yaitu,

1. Menerapkan teknik *pairing* untuk mengklasifikasi suatu kendaraan yang melintasi jalan raya pada malam hari
2. Menghitung banyak kendaraan yang melintasi jalan raya pada malam hari
3. Mengembangkan program yang mampu mengklasifikasi dan menghitung kendaraan yang melintasi jalan raya pada malam hari melalui video

1.5 Manfaat

Manfaat yang dapat diperoleh dari tugas akhir ini adalah dihasilkannya sebuah program yang dapat mengklasifikasi dan menghitung banyak kendaraan yang melintasi jalan pada saat malam hari melalui media video

1.6 Sistematika Penulisan

Penulisan Tugas Akhir ini, disusun dalam enam bab yaitu,

1. BAB I PENDAHULUAN

   Pada bab ini, dibahas mengenai latar belakang adanya penelitian ini. Selanjutnya, disebutkan rumusan masalah berdasarkan latar belakang yang telah dibahas beserta batasan-batasan masalah yang ada. Selain itu, disebutkan juga tujuan dan manfaat mengenai adanya penelitian ini dan sistematika penulisannya.

2. BAB II TINJAUAN PUSTAKA

   Pada bab ini, dibahas mengenai penelitian-penelitian yang telah dilakukan oleh peneliti terdahulu. Penelitian-penelitian tersebut akan menjadi acuan pada penelitian ini. Selain itu, dibahas pula mengenai teori-teori penunjang yang akan digunakan dalam penelitian ini.

3. BAB III METODOLOGI PENELITIAN

   Pada bab ini, dijelaskan mengenai tahapan-tahapan dalam pengerjaan penelitian ini secara rinci dan sistematis. Tahapan-tahapan tersebut juga disajikan dalam bentuk diagram alir (*flowchart*).

4. BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM

   Pada bab ini dijelaskan perancangan dan implementasi sistem dimulai dari pembahasan proses pengambilan data masukan, pengolahan data masukan untuk mendapatkan data keluaran yang sesuai serta dilanjutkan dengan implementasi sistem.

5. BAB V UJI COBA DAN PEMBAHASAN

   Pada bab ini membahas tentang hasil uji coba program dari beberapa rekaman video arus lalu lintas pada malam hari. Kemudian dicatat sebagai bahan untuk merumuskan kesimpulan dan saran dari Tugas Akhir ini.

6. BAB VI PENUTUP

Pada bab ini berisi kesimpulan yang diperoleh dari pembahasan pada bab sebelumnya serta saran untuk pengembangan penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

Pada bab ini menjelasakan teori-teori penunjang dalam melakukan penelitian Tugas Akhir ini. Berisi teori-teori yang menjelaskan *image processing* dan operasi-operasi pada suatu citra

2.1 Penelitian Terdahulu

Penelitian tentang deteksi dan perhitungan jumlah kendaraan yang berada di jalan raya sebelumnya telah diteliti oleh beberapa peneliti yaitu Zamroji Hariyanto pada tahun 2014 melakukan penelitian mendeteksi dan mengklasifikasi kendaraan yang melewati jalan raya dengan menggunakan *Geometric Invariant Moment*. Dalam penelitiannya, perangkat lunak yang dirancang untuk mendeteksi kendaraan memiliki tingkat akurasi 91,81% dan terdeteksi 202 dari 220 kendaraan untuk pagi hari kemudian akurasi sebesar 94,79% dan terdeteksi 364 dari 284 kendaraan untuk siang hari [6]. Tetapi pada penelitian ini belum melakukan deteksi dan perhitungan kendaraan pada malam hari.

Selanjutnya yaitu penelitian Kostia Robert yang berjudul *Video-based traffic monitoring at day and night time*[8]. Pada penelitian yang dilakukan Kostia Robert yaitu membandingkan pendeteksian yang terjadi pada siang hari dan pada malam hari. Dari hasil penelitian ini menghasilkan tingkat akurasi pendeteksian sebesar 94,8% untuk sore hari dan 97% untuk malam hari. Tetapi untuk perhitungan belum dilakukan pada penelitian ini. Kemudian yaitu penelitian yang dilakukan oleh S. Padmavathi dkk yang berjudul *Vision based Vehicle Counting for Traffic Congestion Analysis during Night Time* yang melakukan perhitungan jumlah kendaraan di jalan raya pada malam hari[7].

Selanjutnya penelitian Gery Dias Claudio yang berjudul Deteksi dan Perhitungan Kendaraan Bergerak pada Waktu Malam

Hari. Pada penelitian ini, dilakukan perhitungan dengan tingkat akurasi 33.33% dan terdeteksi 20 dari 12 kendaraan untuk video pertama, tingkat akurasi 96% dan terdeteksi 26 dari 25 kendaraan untuk video kedua, dan tingkat akurasi 73.68%, terdeteksi 14 dari 19 kendaraan. Tetapi pada penelitian ini hanya mampu untuk melakukan pendeteksian terhadap mobil saja dan juga tidak bisa mengklasifikasikan jenis kendaraan.

2.2 Citra Digital

Citra digital merupakan merupakan fungsi dua dimensi yang dapat dinyatakan dengan fungsi $f(x, y)$ dimana x dan y merupakan titik koordinat spasial. Sedangkan amplitudo dari fungsi f pada sembarang koordinat (x, y) merupakan nilai intensitas cahaya, yang merupakan representasi dari warna cahaya yang ada pada citra analog, yang dijelaskan dalam dalam buku Digital Image Processing yang ditulis oleh Rafael. C. Gonzales dan Richard. E. Woods [9].

2.3 Video Digital

Video adalah teknologi untuk menangkap, merekam, memproses, menyimpan, dan merekonstruksi suatu urutan dari beberapa citra. Alan C. Bovik menjelaskan bahwa video digital merupakan hasil sampling dan kuantisasi dari video analog[10]. Secara mendasar, tidak ada perbedaan proses sampling dan kuantisasi antara citra digital dan video digital.

Bagaimanapun juga, video analog yang kita lihat sehari-hari seperti tampilan pada TV analog, sebenarnya bukan sesuatu yang benar-benar kontinu, melainkan terdiri dari beberapa frame yang ditampilkan dengan kecepatan tertentu. Setiap frame merupakan citra analog dan kecepatan untuk menampilkan citra-citra yang ada disebut sebagai frame rate dengan satuan fps (frame per

second).Jika frame rate cukup tinggi, maka akan telihat sebagai rangkaian yang kontinu sehingga tercipta ilusi gerak yang halus.

Video analog dapat dinyatakan dengan fungsi $I(x, y, t)$ dimana (x, y) adalah nilai kontinu dari fungsi I dan t menyatakan waktu. Sebenarnya tampilan video analog di TV maupun monitor merupakan representasi dari fungsi sinyal elektrik satu dimensi $V(t)$ yang terdiri dari beberapa citra analog $I(x, y, t)$ dengan jumlah citra (x, y) tertentu dan waktu (t) tertentu. Proses pemisahan video ke beberapa unit frame citra disebut sebagai scanning [7].

2.4 *Noise* Pada Citra

Noise dapat diartikan sebagai gangguan atau kecatatan dari suatu citra yang tidak kita harapkan yang menyebabkan citra menjadi rusak, tidak jelas, blur, bintik-bintik dan sebagainya. Ada banyak noise yang dapat mempengaruhi kualitas dari citra.

Gambar 2.1. *Noise* pada citra

1. Salt and Peper

   Pada citra akan nampak seperti titik-titik. Noise ini disebabkan oleh gangguan tajam atau tiba-tiba pada sinyal citra. Untuk citra *RGB* titik-titik muncul dalam tiga warna yakni merah (red), hijau (green) dan biru (blue), sedangkan pada citra gray noise akan muncul dalam dua warna yakni

hitam (black) dan putih (white) yang tersebar pada citra. Noise ini memberikan efek ”on dan off” pada piksel.

2. Gaussian

Disebut juga *Gaussian White Noise*. Noise ini dikatakan *White Noise* karena mempunyai distribusi normal. Nilai rata-rata dan variasinya besar maka citra seolah-olah hanya terlihat seperti citra putih saja.

3. Poisson

Poisson noise bukan merupakan noise buatan. Poisson merupakan noise yang ditambahkan langsung pada citra tanpa kita menambahkan parameter apapun, sehingga efeknya pada citra pun tetap, berbeda dengan tipe noise yang sudah dijelaskan sebelumnya.

4. Speckle

Speckle merupakan noise ganda. Noise ini ditambahkan pada citra menggunakan persamaan $J = I + n * I$, dimana n terdistribusi random seragam dengan mean 0 dan varian V. V adalah konstanta non negative yang besarnya dapat berubah-ubah. Default nilai untuk V adalah 0.04. Makin besar nilai V maka citra akan semakin kabur. Noise speckle sering dijumpai pada aplikasi radar.

5. Localvar

Localvar merupakan Gaussian noise dengan mean 0, dengan variance noise adalah fungsi dari intensitas citra yang nilainya berada dalam matrik citra. Vektor intensitas citra tidak boleh bernilai sama karena citra akan nampak sebagai layar putih.

2.5 Thresholding

Thresholding adalah proses mengubah citra berderajat keabuan menjadi citra biner atau hitam putih sehingga dapat diketahui daerah mana yang termasuk objek dan background dari

citra secara jelas. Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra grayscale, dan citra warna.

1. Citra Biner (Monokrom).

   Banyaknya dua warna, yaitu hitam dan putih. Dibutuhkan 1 bit di memori untuk menyimpan kedua warna ini.

2. Citra *Grayscale* (Skala Keabuan).

   Banyaknya warna tergantung pada jumlah bit yang disediakan di memori untuk menampung kebutuhan warna ini. Citra 2 bit mewakili 4 warna, citra 3 bit mewakili 8 warna, dan seterusnya. Semakin besar jumlah bit warna yang disediakan di memori, semakin halus gradasi warna yang terbentuk.

3. Citra Warna (*True Color*).

   Setiap piksel pada citra warna mewakili warna yang merupakan kombinasi dari tiga warna dasar (RGB = *Red Green Blue*). Setiap warna dasar menggunakan penyimpanan 8 bit = 1 byte, yang berarti setiap warna mempunyai gradasi sebanyak 255 warna. Berarti setiap piksel mempunyai kombinasi warna sebanyak 28 x 28 x 28 = 224 =16 juta warna lebih. Itulah sebabnya format ini dinamakan *true color* karena mempunyai jumlah warna yang cukup besar sehingga bisa dikatakan hampir mencakup semua warna di alam.

Citra hasil thresholding biasanya digunakan lebih lanjut untuk proses pengenalan objek serta ekstraksi fitur. Metode thresholding secara umum dibagi menjadi dua, yaitu:

1. *Thresholding* global

   Thresholding dilakukan dengan mempartisi histogram dengan menggunakan sebuah threshold (batas ambang) global T, yang berlaku untuk seluruh bagian pada citra.

2. *Thresholding* adaptif

Thresholding dilakukan dengan membagi citra menggunakan beberapa sub citra. Lalu pada setiap sub citra, segmentasi dilakukan dengan menggunakan *threshold* yang berbeda.

2.6 Operasi Morfologi

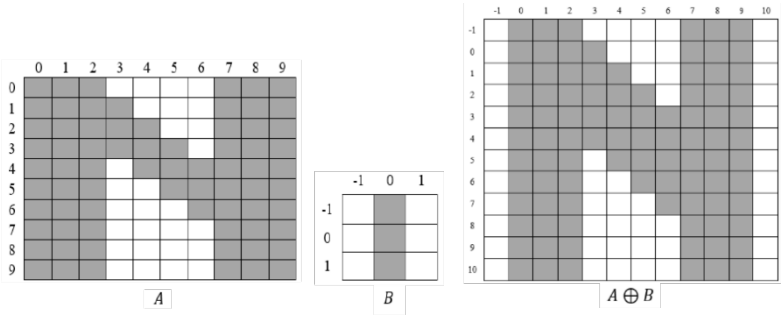
Kata morfologi merupakan sebuah cabang biologi yang berhubungan dengan bentuk dan struktur dari hewan dan tumbuhan. Kata yang sama digunakan dalam konteks morfologi matematika yaitu untuk mengekstraksi komponen gambar yang berguna dalam mendeskripsikan dan merepresentasikan bentuk wilayah. Morfologi matematika menggunakan teori himpunan dalam memroses suatu citra. Himpunan dalam morfologi matematika mewakili objek dalam suatu citra. Sebagai contoh, himpunan semua piksel putih dalam suatu citra biner adalah deskripsi morfologi lengkap dari citra tersebut. Dalam gambar biner, himpunan yang dimaksud adalah anggota ruang bilangan bulat 2 dimensi Z^2 di mana setiap elemen himpunan menunjukkan koordinat (x, y) dengan $x, y \in Z$ piksel putih pada suatu citra[9].

2.6.1 Dilasi

Pada dilasi, setiap piksel dari *background* yang sesuai dengan ukuran dan bentuk dari s*tructuring element* sehingga menyentuh piksel-piksel objek diubah menjadi piksel objek, hal tersebut membuat objek menjadi terlihat lebih gemuk. Operasi pada dilasi dilakukan dengan cara berikut, misal A dan B himpunan pada Z^2 berturut-turut menyatakan objek dari suatu citra dan *structuring element*, dilasi dari A oleh B dinyatakan dengan

$$A \oplus B = \{a + b\ , a \in A, b \in B\}$$

Berikut contoh operasi dilasi pada suatu objek A dengan *structuring element B*



$A = \{(0,0), (0,1), (0,2), (0,7), (0,8), (0,9), (1,0), (1,1), (1,2), (1,3), (1,7), (1,8), (1,9), (2,0), (2,1), (2,2), (2,3), (2,4), (2,7), (2,8), (2,9), (3,0), (3,1), (3,2), (3,3), (3,4), (3,5), (3,7), (3,8), (3,9), (4,0), (4,1), (4,2), (4,4), (4,5), (4,6), (4,7), (4,8), (4,9), (5,0), (5,1), (5,2), (5,5), (5,6), (5,7), (5,8), (5,9), (6,0), (6,1), (6,2), (6,6), (6,7), (6,8), (6,9), (7,0), (7,1), (7,2), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9)\}$

$B = \{(-1,0), (0,0), (1,0)\}$

$A \oplus B = \{(-1,0), (0,0), (1,0), (-1,1), (0,1), (1,1), (-1,2), (0,2), (1,2), (-1,7), (0,7), (1,7), (-1,8), (0,8), (1,8), (-1,9), (0,9), (1,9), (2,0), (2,1), (2,2), (0,3), (1,3), (2,3), (2,7), (2,8), (2,9), (3,0), (3,1), (3,2), (3,3), (1,4), (2,4), (3,4), (3,7), (3,8), (3,9), (4,0), (4,1), (4,2), (4,3), (4,4), (2,5), (3,5), (4,5), (4,7), (4,8), (4,9), (5,0), (5,1), (5,2), (5,4), (5,5), (3,6), (4,6), (5,6), (5,7), (5,8), (5,9), (6,0), (6,1), (6,2), (6,5), (6,6), (6,7), (6,8), (6,9), (7,0), (7,1), (7,2), (7,6), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9), (10,0), (10,1), (10,2), (10,7), (10,8), (10,9)\}$

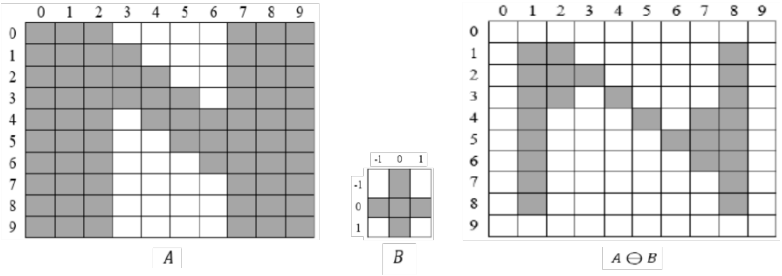
2.6.2 Erosi

Operasi erosi, sama seperti pada dilasi, dilakukan secara biner. Tetapi untuk erosi akan menghasilkan objek pada citra menjadi lebih tipis atau terkikis. Operasi pada erosi dilakukan dengan cara berikut, misal A dan B himpunan pada Z^2 berturut-

turut menyatakan objek dari suatu citra dan *structuring element*, erosi dari A oleh B dinyatakan dengan

$$A \ominus B = \{z|(B)_z \subseteq A\}$$

di mana erosi A oleh B adalah semua himpunan z yaitu translasi dari B oleh z, ditulis $(B)_z$, terdapat di dalam A. Berikut contoh operasi erosi pada suatu objek A dengan *structuring element B*



$A = \{(0,0), (0,1), (0,2), (0,7), (0,8), (0,9), (1,0), (1,1), (1,2), (1,3), (1,7), (1,8), (1,9), (2,0), (2,1), (2,2), (2,3), (2,4), (2,7), (2,8), (2,9), (3,0), (3,1), (3,2), (3,3), (3,4), (3,5), (3,7), (3,8), (3,9), (4,0), (4,1), (4,2), (4,4), (4,5), (4,6), (4,7), (4,8), (4,9), (5,0), (5,1), (5,2), (5,5), (5,6), (5,7), (5,8), (5,9), (6,0), (6,1), (6,2), (6,6), (6,7), (6,8), (6,9), (7,0), (7,1), (7,2), (7,7), (7,8), (7,9), (8,0), (8,1), (8,2), (8,7), (8,8), (8,9), (9,0), (9,1), (9,2), (9,7), (9,8), (9,9)\}$

$B = \{(-1,0), (0,-1), (0,0), (0,1), (1,0)\}$

$A \ominus B = \{(1,1), (1,2), (1,8), (2,1), (2,2), (2,3), (2,8), (3,1), (3,2), (3,4), (3,8), (4,1), (4,5), (4,7), (4,8), (5,1), (5,6), (5,7), (5,8), (6,1), (6,7), (6,8), (7,1), (7,8), (8,1), (8,8)\}$

2.6.3 Opening

Opening merupakan suatu operasi morfologi citra berupa gabungan dari erosi dan dilasi, opening merupakan operasi dilasi terhadap erosi yang terjadi pada suatu himpunan A dimana A adalah suatu citra masukkan.

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

Bersama dengan operasi closing, opening berperan dalam pemrosesan citra sebagai alat dasar untuk menghilangkan *noise* morfologi.

2.6.4 Closing

Pada morfologi matematika, closing dari suatu himpunan A oleh structuring element B adalah erosi dari dilasi A oleh B.

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

Set A terlebih dahulu di dilasi oleh B, kemudian dilanjutkan dengan melakukan erosi ke proses tersebut. Closing akan menghilangkan lubang kecil pada citra sedangkan opening akan menghilangkan objek yang kecil.

2.7 *Pairing* 2 Objek

Posisi titik pusat dari setiap objek yang telah didapatkan digunakan untuk melakukan pengecekan *pairing* dua objek. Berdasar pada posisi titik pusat tersebut, kemudian dilakukan perhitungan jarak antar titik pusat. Jarak antar titik pusat yang dihitung merupakan jarak horizontal dan vertikal. Jika jarak horizontal dan vertikalnya memenuhi suatu kriteria yang telah ditentukan selanjutnya akan dihitung selisih luas kedua objek. Jika selisih luasnya memenuhi kriteria yang telah ditentukan maka kedua objek tersebut akan di*pairing*.

$$y_1 - y_2 \leq \triangle y$$
$$\triangle x_1 \leq x_1 - x_2 \leq \triangle x_2$$
$$a_1 - a_2 \leq \triangle a$$

dengan :

- y_1 merupakan koordinat titik pusat objek pertama pada sumbu y
- y_2 merupakan koordinat titik pusat objek kedua pada sumbu y
- $\triangle y$ merupakan batas jarak vertikal

- x_1 merupakan koordinat titik pusat objek pertama pada sumbu x
- x_2 merupakan koordinat titik pusat objek kedua pada sumbu x
- $\triangle x_1$ merupakan batas minimum jarak horizontal
- $\triangle x_2$ merupakan batas maksimum jarak horizontal
- a_1 merupakan luas objek pertama
- a_2 merupakan luas objek kedua
- $\triangle a$ merupakan batas selisih luas

BAB III
METODOLOGI PENELITIAN

Pada bab ini dijelaskan mengenai metodologi sistem yang digunakan untuk menyelesaikan Tugas Akhir ini. Metodologi penelitian yang digunakan berfungsi sebagai acuan sehingga penelitian dapat berjalan sistematis.

3.1 Studi Literatur dan Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang diperlukan pada penulisan tugas akhir ini. Data yang dikumpulkan berupa video keadaan jalan raya pada waktu malam hari. Video diambil dari atas jembatan penyeberangan dengan arah kendaraan satu arah. Kemudian melakukan penelitian video processing yang diperlukan sebagai metode dalam menyelesaikan permasalahan pada Tugas Akhir ini dengan cara mempelajari literatur-literatur ilmiah yang memiliki hubungan dengan topik penelitian yang sedang penulis lakukan guna mendapat data teoritis sehubungan dengan pembahasan penelitian ini.

3.2 Perancangan dan Implementasi Program

Pada tahap ini akan dilakukan perancangan program berupa memilih dan menganalisis video yang telah didapatkan. Kemudian akan dibuat sebuah program dengan menggunakan bahasa pemrograman Java. Tahap implementasi program akan dijelaskan dengan menggunakan diagram alir serta diberikan penjelasan pada setiap tahapan.

Gambar 3.1. Blok diagram proses

1. *Pre-processing*

Melakukan *pre-processing* pada video yang telah dimasukkan. *Pre-processing* dilakukan dengan cara menghilangkan *noise* yang ada pada citra tersebut. Penghilangan *noise* sangat penting agar objek lampu depan kendaraan sangat jelas. Penghilangan *noise* menggunakan filter *Gaussian*[11].

2. Ekstraksi Objek Terang

Untuk mengekstraksi lampu depan menggunakan teknik thresholding. Di malam hari, lampu depan akan memiliki intensitas tinggi. Sehingga bisa menyegmentasi lampu depan. Misalkan histogram tingkat abu-abu sesuai dengan

gambar $f(x, y)$ terdiri dari objek gelap dalam latar belakang yang terang, sedemikian rupa sehingga piksel objek dan latar belakang memiliki tingkat abu-abu yang dikelompokkan menjadi dua mode dominan. Salah satu cara yang jelas untuk mengekstrak objek dari latar belakang adalah dengan memilih ambang batas 'T' yang memisahkan mode ini. Kemudian setiap titik (x, y) yang $f(x, y) > T$ disebut titik objek, jika tidak disebut titik latar balakang[11].

3. Operasi Morfologi

Setelah ekstraksi benda terang maka perlu dilakukan mengekstrak lampu depan kendaraan secara terpisah. Pada malam hari, refleksi jalan menimbulkan masalah besar, sehingga perlu untuk menghilangkan refleksi tersebut menggunakan operasi morfologi. Refleksi ini dapat mengurangi keakuratan penghitungan. Ada beberapa jenis operator morfologi khususnya operator *opening* dan *closing* yang akan digunakan. Sehingga hanya akan didapatkan lampu depan kendaraan[11].

4. *Tracking* dan *Pairing* Objek

Kendaraan roda 4 biasanya memiliki dua lampu depan, oleh karena itu, perlu memasangkan setiap lampu depan dengan lampu depan lainnya dalam frame yang sama. Ketentuan berikut digunakan untuk menghubungkan antar lampu depan :

1) Jarak horizontal dan vertikal antar titik pusat
2) Perbedaan area antar titik pusat

Misalkan titik pusat dan ukuran dari lampu depan yang pertama dinotasikan (x_1, y_1) dan a_1 sedangkan titik pusat dan ukuran dari lampu depan kedua dinotasikan (x_2, y_2) dan a_2. Dua lampu depan dianggap sebagai objek lampu depan dari kendaraan yang sama jika memenuhi kondisi :

$$y_1 - y_2 \leq \triangle y$$
$$\triangle x_1 \leq x_1 - x_2 \leq \triangle x_2$$
$$a_1 - a_2 \leq \triangle a$$

dimana $\triangle a$ mengontrol perbedaan area, $\triangle x_1$ dan $\triangle x_2$ mengontrol jarak horizontal, dan $\triangle y$ mengontrol jarak vertikal. Dua lampu depan dianggap sebagai objek lampu depan yang sama jika terdapat irisan / perpotongan bagian dari objek pada frame sebelumnya dengan objek pada frame saat ini.

5. Klasifikasi dan Perhitungan

Lampu depan roda 4 akan berada dekat secara horizontal atau sedikit perbedaan pada jarak vertikal karena variasi garis penglihatan dan kelurusan kendaraan[7]. Oleh karena itu antara titik pusat objek satu dengan titik pusat objek di dekatnya memiliki jarak horizontal minimal ke kanan dan dengan perbedaan kecil pada jarak vertikal keduanya. Untuk setiap titik pusat objek dari kiri ke kanan, semua titik pusat pada jarak horizontal minimal ditemukan dan perbedaan jarak vertikal mereka dihitung. Semua kecocokan tersebut diklasifikasikan sebagai kendaraan roda 4 dan jika tidak ditemukan kecocokan seperti itu, kendaraan ini diklasifikasikan sebagai roda 2. Lampu depan roda 2 yang berada di dekat roda 4 dipasangkan dengan algoritma yang disebutkan di atas. Sehingga memberikan hasil yang salah.

Setiap lampu depan kendaraan yang telah keluar dari ROI melalui garis BL – BR akan dilakukan perhitungan dengan memperhatikan informasi / atribut yang dibawa oleh lampu tersebut.

3.3 Simulasi Program

Pada tahap ini dilakukan pembuatan implementasi dari tugas akhir ini berupa simulasi program sesuai dengan perancangan

program yang telah dilakukan dengan menggunakan bahasa pemrograman Java.

3.4 Pengujian Program

Pada tahap ini dilakukan uji coba simulasi perogram yang telah dibuat. Setelah itu akan dilihat apakah pada video yang didapatkan dapat mendeteksi kendaraan yang lewat dan apakah dapat menghitung jumlah kendaraan yang melewati jalan tersebut. Pada penelitian ini akan dilakukan pendeteksian dan perhitungan kendaraan berdasarkan lampu kendaraannya. Apabila terjadi terjadi kesalahan dan error, akan dilakukan perbaikan program sehingga didapatkan hasil yang baik, akurat dan sesuai dengan yang diinginkan.

3.5 Penarikan Kesimpulan

Pada tahap ini dilakukan penarikan kesimpulan terhadap dari hasil pengerjaan, uji coba yang telah dilakukan dan pembahasan dari pengerjaan Tugas Akhir ini serta disampaikan saran untuk pengembangan berikutnya.

3.6 Pembuatan Laporan

Bagian terakhir dalam Tugas Akhir ini adalah membuat laporan seluruh tahapan/proses yang sudah dilakukan.

BAB IV
PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini dibahas mengenai perancangan dan implementasi sistem dimulai dari pembahasan proses pengambilan data masukan, pengolahan data masukan dengan menggunakan pengolahan citra digital serta penjelasan mengenai cara untuk mendapatkan data keluaran yang sesuai dengan tujuan dari penelitian Tugas Akhir ini. Hasil dari analisis perancangan sistem dilanjutkan dengan implementasi sistem. Sehingga perangkat lunak dapat memproses informasi dari rekaman video digital.

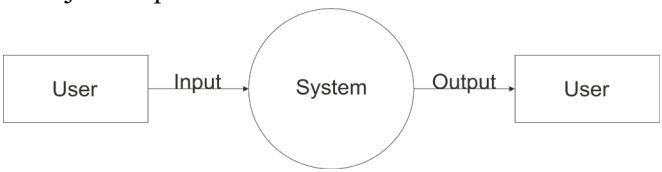
4.1 Analisis Sistem

Untuk membangun perangkat lunak yang dapat mengklasifikasi dan menghitung jumlah kendaraan bergerak pada waktu malam hari berbasis video digital diperlukan sistem yang dapat mengolah data masukan berupa rekaman arus kendaraan. Pada Tugas Akhir ini menggunakan data masukan berupa video digital arus kendaraan satu arah secara *offline*. Dengan menggunakan pengolahan citra digital dan melakukan pengolahan video pada data yang diperoleh, dapat digunakan untuk mengklasifikasi kendaraan serta menghitung jumlah kendaraan yang melintasi suatu jalan sehingga tercipta sistem yang sesuai dengan tujuan pada Tugas Akhir ini.

4.1.1 Analisis Sistem Perangkat Lunak

Perangkat lunak yang dibangun dapat digunakan oleh dinas perhubungan untuk mengamati kondisi arus lalu lintas di suatu jalan sehingga dapat menganalisa kebutuhan infrastruktur jalan tersebut. Selain itu, masyarakat juga dapat memperoleh informasi mengenai kondisi lalu lintas pada suatu jalan. Proses alur dari tugas akhir ini ditunjukkan pada *Data Flow Diagram* dengan sistem perangkat lunak Klasifikasi dan Penghitungan Kendaraan

Bergerak Pada Malam Hari dengan Teknik *Pairing* Lampu Depan yang ditunjukkan pada Gambar 4.1.



Gambar 4.1. Data *Flow* Diagram

Sistem perangkat lunak yang dibangun memiliki beberapa tahapan sebagai berikut,

a. Akuisisi Video

Data masukan berupa rekaman video digital *offline* arus kendaraan yang kemudian di ekstraksi, mengubah video menjadi rangkaian citra yang sering disebut sabagai *frame*.

Gambar 4.2. Proses ekstraksi video

Pengambilan video arus lalu lintas kendaraan dilakukan di beberapa jalan di daerah Surabaya dengan menggunakan kamera digital. Pada prosesnya, terdapat beberapa ketentuan dalam pengambilan video yaitu sebagai berikut,

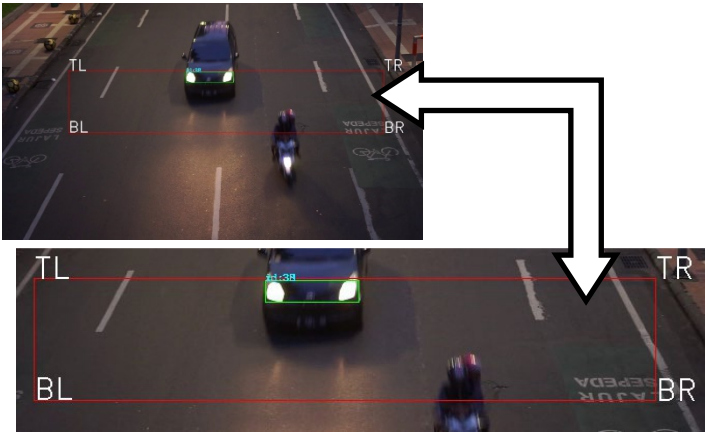
- Sudut pandang perekaman video dari sisi atas dengan jangkauan pandangan selebar jalan satu arah.
- Video diambil dalam waktu malam hari.

b. Penentuan *Region of Interest* (ROI)

Langkah pertama dalam penentuan ROI yaitu dengan membuat bentuk segi empat pada bidang citra *frame* video. Dari bentuk segi empat tersebut akan didapatkan posisi titik

TL atau *top left* yaitu (x_1, y_1) dan BR atau *bottom right* yaitu (x_2, y_2). Sehingga dapat ditentukan posisi kedua titik lainnya yaitu TR (*top right*) pada (x_2, y_1) dan BL (*bottom left*) pada (x_1, y_2) seperti pada Gambar 4.3.

Garis dari titik TL hingga TR akan digunakan sebagai awal dari pendeteksian objek yang selanjutnya akan dipasangkan dan dilacak pada setiap *frame*. Sedangkan garis dari titik BL hingga BR akan digunakan sebagai garis akhir dari pendeteksian yang selanjutnya dilakukan penghitungan. Sehingga pendeteksian kendaraan bergerak hanya dilakukan pada area ROI.



Gambar 4.3. Citra dengan ROI

c. *Bilateral Filtering*

Lampu utama kendaraan umumnya memiliki intensitas tertinggi dalam gambar malam hari. Namun, mungkin ada refleksi yang kuat di permukaan jalan yang mungkin memiliki intensitas setinggi lampu dan sangat menurunkan kinerja deteksi lampu. Sehingga perlu dilakukan filtering agar refleksi tidak memiliki intensitas setinggi lampu.

Pada *Bilateral Filtering* ini nilai piksel hasil diperoleh dari rata-rata pembobotan piksel-piksel tetangga melalui proses konvolusi. Semakin kecil bobot spasial suatu piksel berarti semakin besar jarak piksel tersebut terhadap piksel yang sedang menjadi pusat analisis pada citra, begitupun sebaliknya. Semakin besar perbedaan intensitas antara dua piksel, maka semakin kecil pula bobot fotometriknya, sehingga kontribusinya pada pembobotan pun kecil, begitupun sebaliknya. Perhitungan bobot spasial pada tiap piksel ditunjukkan pada persamaan (1)

$$w_S[x,u] = \exp\left\{\frac{-d^2[x],[x-u]}{2\sigma^2{}_S}\right\} = \exp\{\frac{-u^2}{2\sigma^2{}_S}\} \qquad (1)$$

dengan :

- $w_S[x,u]$ adalah bobot spasial tiap piksel pada kernel
- x merupakan titik tengah kernel ($w(0,0)$)
- u merupakan elemen-elemen tetangga pada kernel
- σ_S merupakan standar deviasi untuk pembobotan spasial, yang besarnya dapat berubah-ubah sesuai *input* dari *user*
- d jarak *euclidean*

Perhitungan bobot fotometrik pada tiap piksel ditunjukkan pada persamaan (2)

$$w_R[x,u] = \exp\left\{\frac{-d^2\{g[x],g[x-u]\}}{2\sigma^2{}_R}\right\} = \exp\left\{\frac{-\{g[x]-g[x-u]\}^2}{2\sigma^2{}_R}\right\} \qquad (2)$$

dengan:

- $w_R[x,u]$ adalah bobot fotometrik tiap piksel pada kernel
- x merupakan titik tengah kernel ($w(0,0)$)
- u merupakan elemen-elemen tetangga pada kernel
- $g[x]$ adalah piksel yang menjadi pusat analisis pada citra terdegradasi

- $g[x-u]$ adalah piksel tetangga dari piksel $g[x]$
- σ_R merupakan standar deviasi untuk pembobotan fotometrik, yang besarnya tergantung *input* dari *user*
- d jarak *euclidean*

Kedua bobot tersebut dinormalisasi menjadi satu nilai bobot (W) seperti pada persamaan (3)

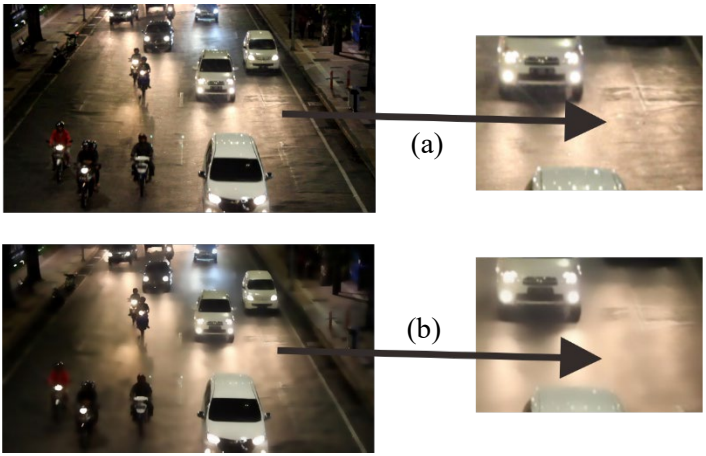
$$W[x,u] = W_S[x,u] * W_R[x,u] \quad (3)$$

Setelah pembobotan terhadap piksel-piksel dilakukan, maka nilai piksel hasil dapat dicari dengan persamaan (4)

$$f[x] = \frac{\sum_{u=-N}^{N} W[x,u]g[x-u]}{\sum_{u=-N}^{N} W[x.u]} \quad (4)$$

dengan :

- N = dimensi kernel
- $W[x,u]$ merupakan nilai bobot tetangga pada matriks bobot W
- $g[x,u]$ merupakan piksel tetangga dari piksel $g[x]$
- $f(x)$ merupakan nilai piksel hasil



Gambar 4.4. *Bilateral filtering*, (a) Citra sebelum di filter, (b) Citra setelah di filter

d. *Grayscale*

Citra hasil *filtering* masih dalam bentuk citra RGB kemudian akan diubah ke dalam bentuk citra *grayscale*. Untuk mendapatkan citra *grayscale* dari citra RGB, harus melalui proses pengubahan nilai dari citra RGB tersebut. Dikarenakan nilai dari suatu citra memiliki nilai matriks yang sangat besar maka untuk menunjukkan perhitungan pembentukan citra *grayscale*, diambil sebagian citra pada salah satu frame video 1 berupa matriks 5 x 5. Didapatkan matriks R,G,B dari citra tersebut seperti pada matriks dibawah ini.

$$r = \begin{bmatrix} 209 & 203 & 202 & 199 & 198 \\ 210 & 203 & 197 & 194 & 196 \\ 196 & 185 & 178 & 174 & 178 \\ 157 & 139 & 133 & 132 & 140 \\ 111 & 91 & 88 & 90 & 101 \end{bmatrix}, \; g = \begin{bmatrix} 179 & 178 & 177 & 176 & 175 \\ 185 & 187 & 181 & 182 & 184 \\ 171 & 196 & 162 & 162 & 167 \\ 135 & 127 & 122 & 127 & 135 \\ 88 & 80 & 76 & 85 & 96 \end{bmatrix}$$

$$b = \begin{bmatrix} 154 & 154 & 153 & 152 & 151 \\ 157 & 163 & 157 & 160 & 162 \\ 143 & 146 & 139 & 140 & 144 \\ 106 & 103 & 97 & 108 & 116 \\ 59 & 55 & 51 & 66 & 76 \end{bmatrix}$$

Proses *grayscaling* dilakukan dengan menggunakan metode Luminosity yang mengalihkan setiap nilai RGB dengan konstanta yang telah ditentukan. Secara matematis dapat ditulis sebagai berikut.

$$grayscale = 0.21 * r + 0.72 * g + 0.07 * b$$

Citra yang akan diubah menjadi citra *grayscale* didapatkan sebagai berikut

$$grayscale = 0.21 \begin{bmatrix} 209 & 203 & 202 & 199 & 198 \\ 210 & 203 & 197 & 194 & 196 \\ 196 & 185 & 178 & 174 & 178 \\ 157 & 139 & 133 & 132 & 140 \\ 111 & 91 & 88 & 90 & 101 \end{bmatrix} +$$

$$+0.72 \begin{bmatrix} 179 & 178 & 177 & 176 & 175 \\ 185 & 187 & 181 & 182 & 184 \\ 171 & 196 & 162 & 162 & 167 \\ 135 & 127 & 122 & 127 & 135 \\ 88 & 80 & 76 & 85 & 96 \end{bmatrix} + 0.07 \begin{bmatrix} 154 & 154 & 153 & 152 & 151 \\ 157 & 163 & 157 & 160 & 162 \\ 143 & 146 & 139 & 140 & 144 \\ 106 & 103 & 97 & 108 & 116 \\ 59 & 55 & 51 & 66 & 76 \end{bmatrix}$$

$$= \begin{bmatrix} 44 & 43 & 42 & 42 & 42 \\ 44 & 43 & 41 & 41 & 41 \\ 41 & 39 & 37 & 37 & 37 \\ 33 & 29 & 97 & 28 & 29 \\ 23 & 19 & 51 & 19 & 21 \end{bmatrix} + \begin{bmatrix} 129 & 128 & 127 & 127 & 126 \\ 133 & 135 & 130 & 131 & 132 \\ 123 & 122 & 117 & 117 & 120 \\ 97 & 91 & 88 & 91 & 97 \\ 63 & 58 & 55 & 61 & 69 \end{bmatrix} +$$

$$+ \begin{bmatrix} 11 & 11 & 11 & 11 & 11 \\ 11 & 11 & 11 & 11 & 11 \\ 10 & 10 & 10 & 10 & 10 \\ 7 & 7 & 8 & 8 & 8 \\ 4 & 4 & 5 & 5 & 5 \end{bmatrix} = \begin{bmatrix} 184 & 182 & 180 & 180 & 179 \\ 188 & 189 & 182 & 183 & 184 \\ 174 & 171 & 164 & 164 & 167 \\ 137 & 127 & 123 & 127 & 134 \\ 90 & 81 & 77 & 85 & 95 \end{bmatrix}$$

Sehingga citra tersebut telah berubah menjadi citra *grayscale* yang dapat dilihat pada gambar dibawah ini

(a) (b)

Gambar 4.5. Proses *grayscaling*, (a) Citra RGB, (b) Citra *grayscale*

e. *Noise Filtering*

Citra hasil perekaman video memungkinkan masih memiliki *noise-noise* yang muncul, baik karena teknis pengambilan video maupun keadaan lingkungan. Sehingga perlu untuk menghilangkan *noise* yang muncul. *Filter* yang digunakan untuk menghilangkan *noise* adalah *Gaussian filtering*.

Gaussian filtering didapat dari operasi konvolusi. Operasi perkalian yang dilakukan ialah perkalian antara matriks kernel dengan matriks gambar asli. Matriks kernel gauss didapat dari fungsi komputasi dari distribusi gaussian, seperti pada persamaan (5)

$$G(i,j) = c.e - \frac{(i-u)^2 + (j-v)^2}{2\sigma^2} \qquad (5)$$

keterangan :

- c dan σ adalah konstanta
- $G(i,j)$ adalah elemen matriks kernel gauss pada posisi (i,j)
- (u,v) = indeks tengah dari matriks kernel gauss

Contoh konvolusi matriks gambar asli (A) dengan matrik kernel gauss (G) :

$$\begin{bmatrix} 0 & 5 & 70 & 45 & 0 & 0 & 0 \\ 0 & 20 & 55 & 30 & 10 & 5 & 0 \\ 30 & 35 & 40 & 15 & 60 & 45 & 30 \\ 0 & 15 & 50 & 85 & 23 & 37 & 51 \\ 44 & 52 & 66 & 16 & 33 & 44 & 55 \\ 46 & 47 & 28 & 36 & 7 & 51 & 3 \\ 0 & 62 & 0 & 56 & 0 & 56 & 0 \end{bmatrix} x \begin{bmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

A G

$$= \begin{bmatrix} 390 & 6585 & 510 & 325 & 215 \\ 440 & 570 & 523 & 503 & 483 \\ 529 & 660 & 602 & 579 & 615 \\ 644 & 663 & 556 & 468 & 568 \\ 627 & 468 & 415 & 435 & 649 \end{bmatrix}$$

B

$$\frac{1}{15} \times \begin{bmatrix} 390 & 6585 & 510 & 325 & 215 \\ 440 & 570 & 523 & 503 & 483 \\ 529 & 660 & 602 & 579 & 615 \\ 644 & 663 & 556 & 468 & 568 \\ 627 & 468 & 415 & 435 & 649 \end{bmatrix}$$

$$= \begin{bmatrix} 26 & 439 & 34 & 22 & 14 \\ 29 & 38 & 35 & 34 & 32 \\ 35 & 44 & 40 & 39 & 41 \\ 43 & 44 & 37 & 31 & 38 \\ 42 & 31 & 28 & 29 & 43 \end{bmatrix}$$

C

keterangan :

A = Matriks gambar asal

G = Matriks kernel gauss

B = Matriks hasil perkalian

C = Matriks gambar hasil

(a) (b)

Gambar 4.6. *Noise filtering*, (a) Citra dengan *noise*, (b) Citra dengan *noise* yang telah dihilangkan

f. Ekstraksi Objek Terang

Setelah menghilangkan *noise*, selanjutnya dilakukan proses *thresholding* pada citra untuk mendapatkan citra biner. Pada proses ini kecerahan warna dari setiap piksel dibandingkan ke dalam sebuah nilai *threshold* dan piksel hasilnya dinyatakan sebagai satu atau dua kategori, putih atau hitam, tergantung dari apakah nilai piksel aslinya melampaui atau tidak nilai *threshold*nya. Seleksi dari angka *threshold* biasanya didapat dari histogram.

Jika sebuah gambar terdiri dari dua area, satu adalah daerah terang utama yang terang dan yang lain adalah daerah gelap, maka bisa diasumsikan sebuah histogram mempunyai dua puncak. Dan nilai *threshold* bisa dipilih berdasarkan nilai antara dua puncak pada histogramnya. *Threshold* dapat didefinisikan seperti pada persamaan (6)

$$g(x,y) = \begin{cases} 1 & jika\ f(x,y) \geq T \\ 0 & jika\ f(x,y) < T \end{cases} \quad (6)$$

Sehingga mendapatkan bagian cahaya lampu dari kendaraan dalam citra tersebut.

(a) (b)

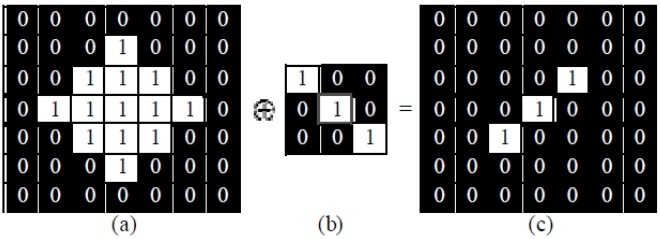
Gambar 4.7. Proses ekstraksi objek terang, (a) Citra RGB, (b) Citra biner

g. Operasi Morfologi

Operasi morfologi citra merupakan suatu operasi pemrosesan citra yang mengolah citra berdasarkan bentuknya. Pada sistem ini digunakan morfologi *opening* pada citra biner yang telah didapatkan. *Opening* merupakan suatu proses dimana suatu gambar terlebih dahulu mengalami proses erosi, disusul dengan proses dilatasi dengan menggunakan struktur elemen yang sama.

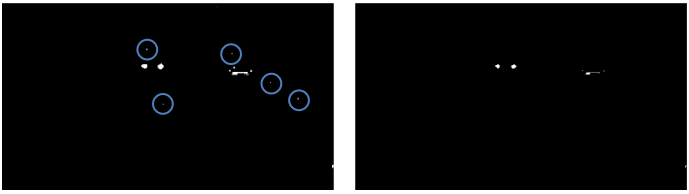
Erosi merupakan proses untuk mengecilkan ataupun menipiskan objek pada gambar, proses ini juga dipengaruhi oleh *structuring element* yang digunakan pada pengolahan gambar. Secara algoritma dapat didefinisikan erosi sebagai :

- Tiap nilai yang ada pada keluaran $A \ominus B$ akan dijadikan nilai 0
- B ditempatkan pada tiap lokasi A yang memiliki nilai 0
- Bila pada A terdapat B maka pada keluaran akan digunakan nilai B
- Hasil keluaran akhir adalah setiap elemen dari A yang tidak terkena elemen B pada proses diatas.



Gambar 4.8. Operasi erosi, (a) Gambar awal, (b) *Structuring element*, (c) Hasil akhir operasi erosi

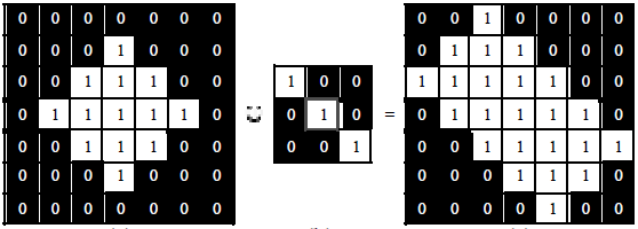
Pada Gambar 4.9 terlihat bahwa dengan dilakukannya proses erosi ini pada citra biner, mampu menghilangkan objek-objek kecil sehingga sangat berguna dalam menghilangkan objek-objek yang tidak diteliti. Karena pada deteksi lampu depan kendaraan, refleksi lampu di jalanan dapat meningkatkan *error* dan mengurangi akurasi sehingga sangat mengganggu dalam deteksi lampu depan kendaraan.



Gambar 4.9. Pemrosesan citra biner dengan operasi erosi, (a) Citra biner, (b) Citra hasil operasi erosi

Dilatasi merupakan proses mempertebal objek pada gambar, proses ini dilakukan dengan memberikan tmabahan piksel pada batasan suatu objek pada suatu gambar. Jumlah piksel yang ditambahkan tergantung dari ukuran dan bentuk dari *structuring element* yang digunakaan pada pengolahan gambar. Secara algoritma dapat didefinisikan dilatasi sebagai berikut :

- Asumsikan bahwa B merupakan sebuah *mask*
- Tempatkan titik referensi dari *structuring element* pada piksel yang memiliki nilai 1 pada gambar A
- Bila terdapat nilai yang berbeda pada piksel A dan B yang memiliki lokasi yang sama, maka nilai dari keluarannya adalah 1. Namun, jika pada piksel A dan B sama maka nilai keluarannya akan sama dengan nilai tersebut.
- Proses diulang terus hingga nilai titik referensi B telah berada di semua titik A yang memiliki nilai 1



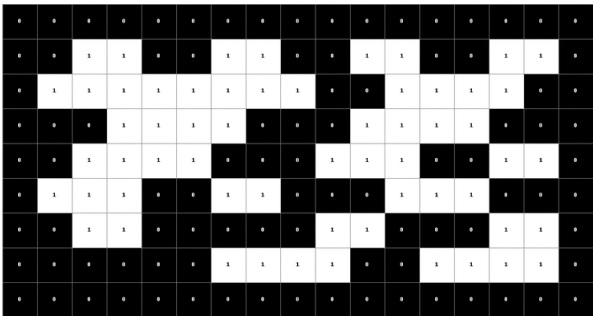
Gambar 4.10. Operasi dilasi, (a) Gambar awal, (b) *Structuring element*, (c) Hasil akhir operasi dilatasi

Pada Gambar 4.11 terlihat bahwa objek-objek yang semula kecil dapat diperbesar dan objek-objek yang besar akan memiliki tepian yang halus. Sehingga dari proses *opening* akan dapat menghasilkan citra yang jelas dengan tepian halus serta mampu menghilangkan objek yang tidak diamati.

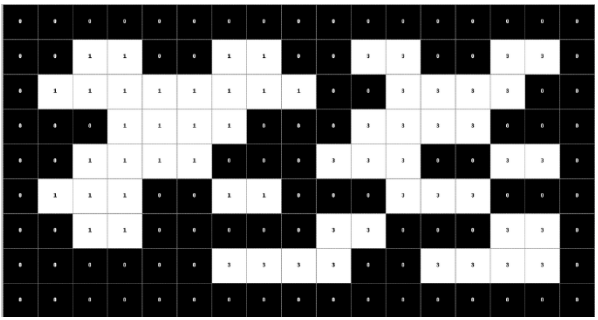
Gambar 4.11. Pemrosesan citra biner dengan operasi dilasi, (a) Citra biner, (b) Citra hasil operasi dilasi

h. Deteksi Objek

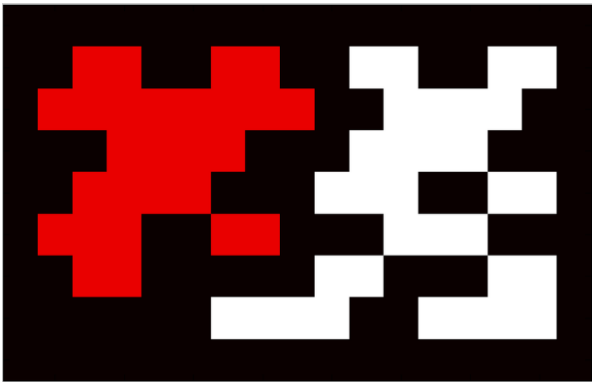
Dari operasi morfologi didapatkan citra biner yang jelas. Kemudian piksel-piksel dikelompokkan menurut nilai intensitasnya (memisahkan antara yang putih atau objek dengan yang hitam atau bukan objek). Dengan demikian didapatkan suatu objek yang merupakan lampu depan dari suatu kendaraan. Lebih jelas perhatikan ilustrasi deteksi objek pada Gambar 4.12.



(a)



(b)

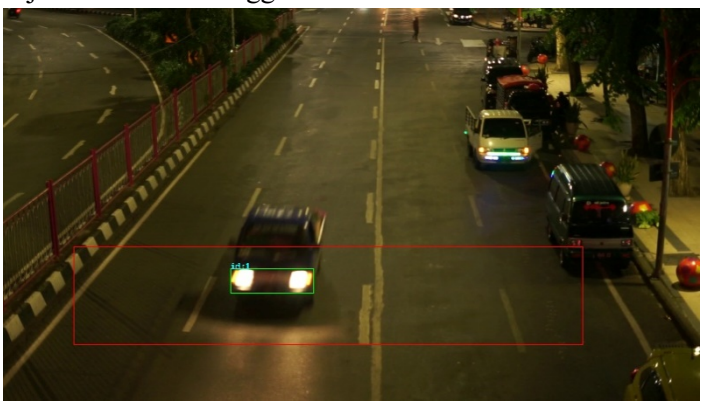


(c)

Gambar 4.12. Proses deteksi objek, (a) Objek dikelompokkan berdasarkan nilai piksel, (b) Objek dipisahkan yang tidak memiliki hubungan ketetanggaan, (c) Hasil pengelompokkan objek

i. *Pairing*

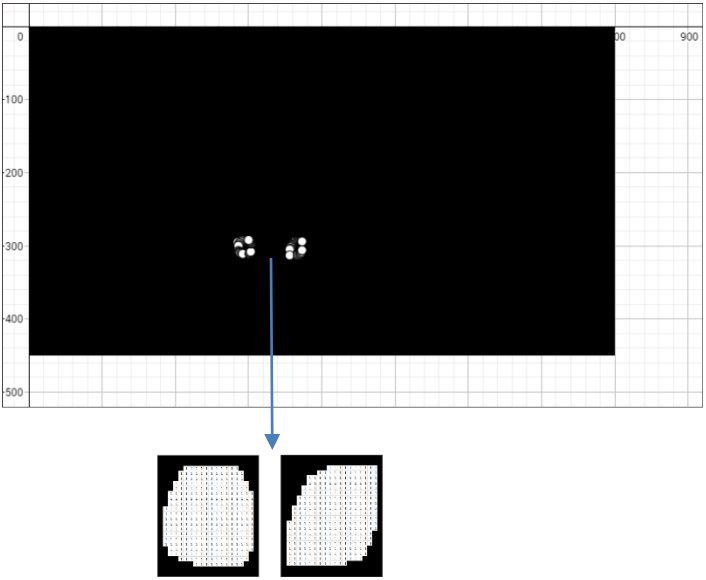
Proses *pairing* mulai berjalan ketika terdapat suatu objek yang telah memasuki area ROI dan berhenti ketika objek tersebut meninggalkan area ROI.



Gambar 4.13. Pasangan objek di dalam ROI

Setelah memasuki area ROI kemudian menentukan letak dari objek tersebut pada suatu citra biner. Letak dari

suatu objek didapatkan dengan mencari kontur dari objek tersebut terlebih dahulu. Kontur yang akan didapatkan menunjukkan posisi objek yang 37elative terhadap citra biner berukuran 450×800 piksel yang merupakan salah satu *frame* dari video.



Gambar 4.14. Pencarian kontur pada citra biner

Citra pada Gambar diatas berukuran 450x800 piksel yang menunjukkan bahwa citra tersebut adalah suatu matriks yang memiliki 450 baris dan 800 kolom piksel. Kemudian diambil sebagian matriks dari matriks berukuran 450x800 piksel tersebut yang hanya memuat objek terdeteksi (nilai pikselnya 1). Sehingga didapatkan matriks seperti yang ditunjukkan pada Gambar 4.15

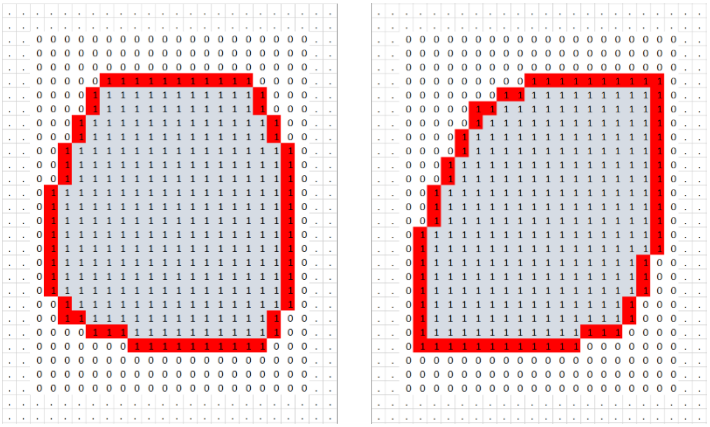
Gambar 4.15. Matriks dari piksel yang bernilai 1

Akan dilakukan koneksi piksel yang bernilai 1 dengan menggunakan algoritma *Connected Component Labelling* (CCL). Hasil pengoneksian menggunakan metode CCL ditentukan oleh jumlah konektivitas yang digunakan pada saat melakukan pengecekan label tetangga terdekat. Terdapat dua jenis konektivitas yang dapat digunakan yaitu 4 konektivitas dan 8 konektivitas.

Konektivitas dengan 4 konektivitas melakukan pengecekan pada $(x+1, y), (x-1, y), (x, y+1), (x, y-1)$. Sedangkan 8 konektivitas melakukan pengecekan pada $(x \pm 1, y \pm 1)$. Berdasarkan matriks yang didapatkan, dapat dilakukan pengoneksian piksel menggunakan 4 konektivitas sehingga didapatkan hasil seperti pada Gambar 4.16

Gambar 4.16. Hasil pengoneksian piksel dengan CCL

Hasil pengoneksian piksel, selanjutnya akan digunakan untuk menentukan piksel terluar dari objek tersebut, yaitu suatu piksel yang memiliki tetangga dengan perbedaan nilai piksel, sehingga didapatkan hasil seperti pada Gambar 4.17



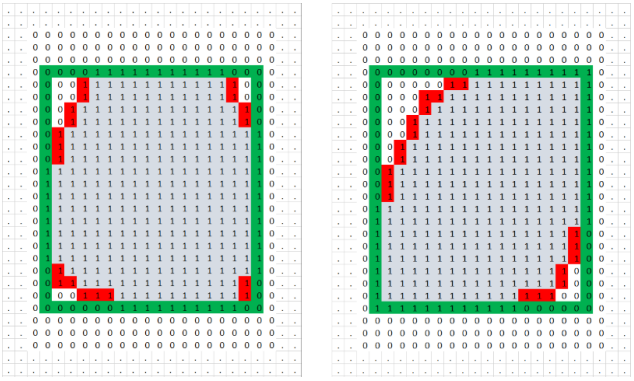
Gambar 4.17. Piksel terluar dari objek

Piksel terluar telah didapatkan sehingga dapat dilakukan pembentukan *boundingrect* yaitu suatu kotak terkecil yang memuat seluruh piksel pada suatu objek. Pembentukan *boundingrect* dimulai dengan pengecekan letak piksel-piksel terluar tersebut, letak TL (*top left*)

boundingrect didapatkan dari letak piksel pada indeks baris dan indeks kolom terkecil. Sedangkan letak BR (*bottom right*) *boundingrect* didapatkan dari letak piksel pada indeks baris dan indeks kolom terbesar.

Pada objek pertama, didapatkan indeks baris terkecil yaitu pada baris ke-294 dan indeks kolom terkecil yaitu pada kolom ke-356 sehingga dapat dituliskan letak TL *boundingrect*nya pada matriks adalah [294, 356]. Sedangkan indeks baris terbesar yaitu pada baris ke-313 dan indeks kolom terbesar yaitu pada kolom ke-373 sehingga dapat dituliskan letak BR *boundingrect*nya pada matriks adalah [313, 373].

Pada objek kedua, didapatkan indeks baris terkecil yaitu pada baris ke-292 dan indeks kolom terkecil yaitu pada kolom ke-286 sehingga dapat dituliskan letak TL *boundingrect*nya pada matriks adalah [292, 286]. Sedangkan indeks baris terbesar yaitu pada baris ke-311 dan indeks kolom terbesar yaitu pada kolom ke-303 sehingga dapat dituliskan letak BR *boundingrect*nya pada matriks adalah [311, 303]. Dari letak TL dan BR *boundingrect*nya yang sudah didapatkan selanjutnya dapat dibentuk masing-masing *boundingrect*nya seperti pada Gambar 4.18



Gambar 4.18. Pembentukan b*oundingrect*

Selanjutnya untuk dapat dilakukan *pairing* maka perlu didapatkan titik pusat dan luas dari setiap *boundingrect* yang terbentuk. Dari segi empat *boundingrect* dilakukan pencarian letak titik pusatnya dan luas areanya dengan persamaan berikut.

$$pusat_{baris} = (TL_{baris} + BR_{baris}) \,/\, 2 \quad (7)$$

$$pusat_{kolom} = (TL_{kolom} + BR_{kolom}) \,/\, 2 \quad (8)$$

$$luas = (BR_{baris} - TL_{baris}) * (BR_{kolom} - TL_{kolom}) \quad (9)$$

Berdasarkan letak TL dan BR *boundingrect* yang telah didapatkan akan ditentukan letak dari masing-masing titik pusatnya dengan persamaan (7) dan (8) serta luas dari objek tersebut dengan persamaan (9). Letak TL dan BR dari objek pertama akan digunakan untuk mendapatkan letak titik pusat dan luas objek id 0 (pada Gambar 4.18) yaitu

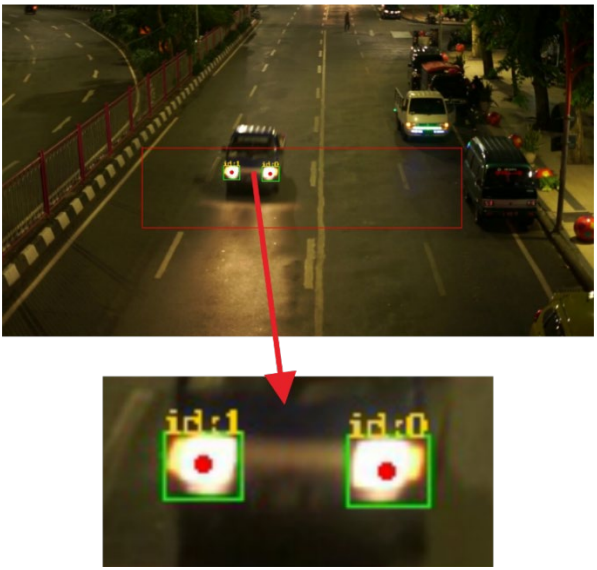
$$pusat0_{baris} = (294 + 313) \,/\, 2 \;=\; 303.5$$
$$pusat0_{kolom} = (356 + 373) \,/\, 2 \;=\; 364.5$$
$$luas0 = (313 - 294) * (373 - 356) \;=\; 323$$

Didapatkan letak titik pusat dari objek id 0 yaitu $[303.5, 364.5]$ karena citra dalam satu satuan piksel sehingga dibulatkan ke satuan terdekat dengan fungsi *ceiling* menjadi $[304, 365]$ dan luas objek id 0 sebesar 323 piksel. Sedangkan letak TL dan BR dari objek kedua akan digunakan untuk mendapatkan letak titik pusat dan luas objek id 1 (pada Gambar 4.19) yaitu

$$pusat1_{baris} = (292 + 311) \,/\, 2 \;=\; 301.5$$
$$pusat1_{kolom} = (286 + 303) \,/\, 2 \;=\; 294.5$$
$$luas1 = (311 - 292) * (303 - 286) \;=\; 323$$

Didapatkan letak titik pusat dari objek id 1 yaitu $[301.5, 294.5]$ dibulatkan ke satuan terdekat dengan fungsi *ceiling* menjadi $[302, 295]$ dan luas objek id 1 sebesar 323 piksel. Setelah didapatkan $[pusat_{baris}, pusat_{kolom}]$ dan luas dari masing-masing objek dalam satu *frame*. Selajutnya akan dilakukan pengecekan pada letak titik-titik pusat dan luas

objek tersebut agar dapat diketahui apakah objek-objek tersebut merupakan sepasang lampu dari kendaraan yang sama atau tidak.



Gambar 4.19. Pencarian titik pusat dari setiap objek

Dua objek akan dikenali sebagai lampu dari kendaraan yang sama jika memenuhi kriteria *pairing* yaitu memiliki perbedaan jarak titik pusat secara vertikal yang sangat kecil sebesar $\triangle y$ dan perbedaan jarak titik pusat secara horizontal minimal sebesar $\triangle x_1$, maksimal sebesar $\triangle x_2$ serta memiliki perbedaan luas objek sebesar $\triangle A$, lebih jelasnya ada pada persamaan (10), (11), dan (12).

$$y_1 - y_2 \leq \triangle y \quad (10)$$

$$\triangle x_1 \leq x_1 - x_2 \leq \triangle x_2 \quad (11)$$

$$a_1 - a_2 \leq \triangle a \quad (12)$$

dengan :

- y_1 merupakan indeks baris titik pusat objek pertama pada matriks
- y_2 merupakan indeks baris titik pusat objek kedua pada matriks
- x_1 merupakan indeks kolom titik pusat objek pertama pada matriks
- x_2 merupakan indeks kolom titik pusat objek kedua pada matriks
- a_1 merupakan luas objek pertama
- a_2 merupakan luas objek kedua

Nilai parameter yang dipilih didapatkan dari berbagai hasil uji coba, yaitu $\triangle y$ = 4 satuan, parameter $\triangle x_1$ = 40 satuan, parameter $\triangle x_2$ 100 satuan, dan parameter $\triangle a$ 50 piksel sehingga didapatkan hasil *pairing* terbaik.

Dari hasil perhitungan didapatkan letak titik pusat objek id 0 pada [304, 365] dengan luas 323 piksel. Letak titik pusat objek id 1 pada (302, 295) dengan luas 323 piksel. Letak titik pusat tersebut relatif terhadap *frame* sehingga di berbeda *frame* akan didapatkan letak titik puast yang mungkin berbeda. Dengan menggunakan persamaan (10), (11), dan (12) didapatkan hasil

$$y_1 - y_2 = 2 \leq \triangle y$$
$$\triangle x_2 \leq x_1 - x_2 = 70 \leq \triangle x_2$$
$$a_1 - a_2 = 0 \leq \triangle a$$

karena ketiga kriteria *pairing* terpenuhi maka kedua lampu tersebut merupakan merupakan kandidat lampu depan dari kendaraan yang sama sehingga akan digabungkan menjadi satu objek seperti pada Gambar 4.20 dibawah ini.

Gambar 4.20. Hasil *pairing* lampu depan roda 4

Saat ini sering dijumpai kendaraan roda 4 dengan lampu depan sebanyak 2 pasang, seperti pada Gambar 4.21. Kondisi seperti ini dapat mempengaruhi proses *pairing* serta hasil penghitungan banyak kendaraan dari hasil klasifikasi sehingga perlu adanya penanganan.

Ide dasar untuk mengatasi kondisi satu mobil dengan 2 pasang lampu adalah dengan mengecek kembali objek yang saat ini memenuhi kriteria *pairing* pada proses *pairing* dengan objek-objek sebelumnya yang telah memenuhi kriteria *pairing* terlebih dahulu dalam 1 *frame*. Jika suatu kendaraan roda 4 memiliki 2 pasang lampu, maka letak titik pusat hasil *pairing*nya akan memiliki perbedaan jarak secara horizontal yang sangat kecil yaitu $\triangle x_{pairing}$ dan memiliki perbedaan jarak secara vertikal tidak lebih dari $\triangle y_{pairing}$, lebih jelasnya pada persamaan (13) dan (14). Jika persamaan (13) dan (14) terpenuhi maka objek tersebut di*pairing* dengan warna berbeda yang menunjukkan bahwa tidak akan dilakukan pelacakan, klasifikasi dan penghitungan pada objek hasil *pairing* tersebut sehingga akan dapat dikenali sebagai satu kendaraan roda 4.

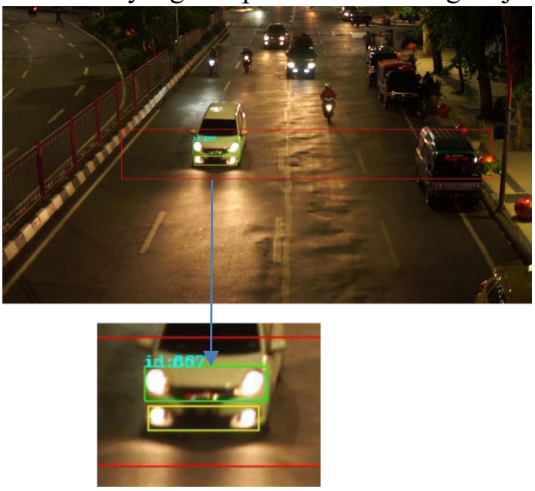
$$y_1 - y_2 \leq \triangle y_{pairing} \quad (13)$$

$$x_1 - x_2 \leq \triangle x_{pairing} \quad (14)$$

dengan :

- y_1 merupakan indeks baris titik pusat objek pertama pada matriks
- y_2 merupakan indeks baris titik pusat objek kedua pada matriks
- x_1 merupakan indeks kolom titik pusat objek pertama pada matriks
- x_2 merupakan indeks kolom titik pusat objek kedua pada matriks

Parameter $\triangle y_{pairing}$ yang dipilih adalah 25 satuan dan $\triangle x_{pairing}$ 10 satuan yang didapatkan dari berbagai uji coba.



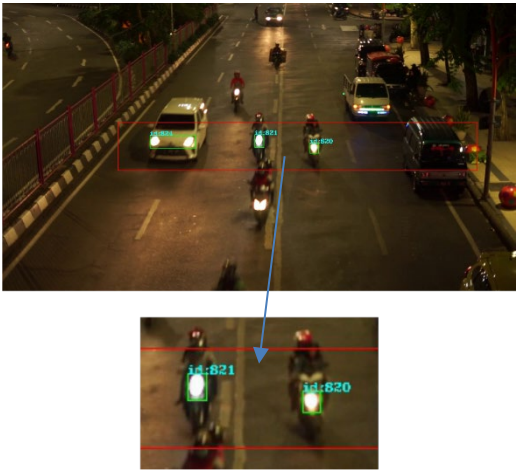
Gambar 4.21. Kendaraan roda 4 dengan 2 pasang lampu

Pada Gambar 4.21 terdapat dua pasang lampu dalam 1 kendaraan roda 4 yaitu pasangan lampu dengan kotak berwarna hijau dan kuning. Pasangan lampu dengan kotak berwarna hijau merupakan pasangan lampu yang akan diproses selanjutnya sedangkan pasangan lampu dengan kotak berwarna kuning

merupakan pasangan lampu yang tidak akan diproses pada tahap berikutnya karena pasangan tersebut memenuhi persamaan (13) dan (14) terhadap pasangan lampu dengan kotak berwarna hijau dengan hasil perhitungan sebagai berikut

$$y_1 - y_2 = 20 \leq \triangle y_{pairing}$$
$$x_1 - x_2 = 1 \leq \triangle x_{pairing}$$

Sehingga dua pasang lampu ini akan dikenali program sebagai dua pasang lampu depan dari kendaraan roda 4 yang sama.

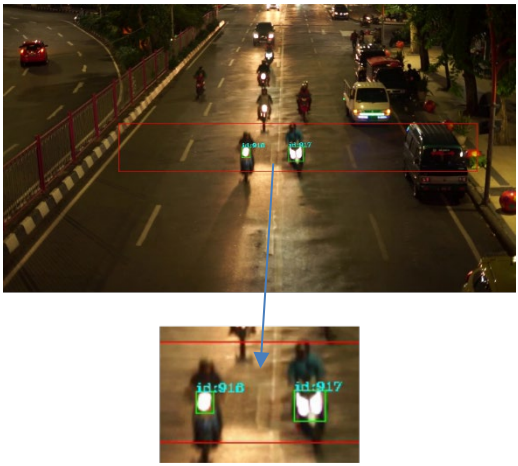


Gambar 4.22. Dua lampu gagal *pairing*

Pada Gambar 4.22 terdapat dua lampu kendaraan yang hampir bersebelahan namun tidak memenuhi salah satu kriteria *pairing*. Pada lampu id 820, titik pusat terletak pada [226, 485] dengan luas 208 dan pada lampu id 821 titik pusatnya terletak pada [215, 400] dengan luas 240 didapatkan :

$$y_1 - y_2 = 11 \not\leq \triangle y$$
$$\triangle x_2 \leq x_1 - x_2 = 85 \leq \triangle x_2$$
$$a_1 - a_2 = 32 \leq \triangle a$$

Dua lampu tersebut memenuhi dua kriteria *pairing* yaitu persamaan (11) dan (12) namun gagal memenuhi persamaan (10) sehingga tidak dilakukan *pairing*.



Gambar 4.23. Dua lampu gagal *pairing*

Pada Gambar 4.23 terdapat dua lampu kendaraan yang bersebelahan namun tidak memenuhi salah satu kriteria *pairing*. Pada lampu id 916, titik pusat terletak pada [231, 378] dengan luas 208 dan pada lampu id 917 titik pusatnya terletak pada [233, 456] dengan luas 529 didapatkan :

$y_1 - y_2 = 2 \leq \triangle y$

$\triangle x_2 \leq x_1 - x_2 = 78 \leq \triangle x_2$

$a_1 - a_2 = 321 \not\leq \triangle a$

Dua lampu tersebut memenuhi dua kriteria *pairing* yaitu persamaan (10) dan (11) namun gagal memenuhi persamaan (12) sehingga tidak dilakukan *pairing*.

Gambar 4.24. Dua lampu gagal *pairing*

Pada Gambar 4.24 terdapat dua lampu kendaraan yang bersebelahan namun tidak memenuhi salah satu kriteria *pairing*. Pada lampu id 1254, titik pusat terletak pada [233, 357] dengan luas 294 dan pada lampu id 1257 titik pusatnya terletak pada [232, 210] dengan luas 336 didapatkan :

$y_1 - y_2 = 1 \leq \triangle y$

$\triangle x_2 \leq x_1 - x_2 = 147 \not\leq \triangle x_2$

$a_1 - a_2 = 42 \leq \triangle a$

Dua lampu tersebut memenuhi dua kriteria *pairing* yaitu persamaan (10) dan (12) namun gagal memenuhi persamaan (11) sehingga tidak dilakukan *pairing*.

Alur proses pengecekan *pairing* pada suatu *frame* adalah sebagai berikut :

- Dilakukan pengecekan pada objek ke-1 dengan objek ke-2
- Jika kriteria *pairing* terpenuhi maka objek ke-1 dan ke-2 di*pairing*kan, menjadi kandidat lampu dari kendaraan roda 4 dan dilanjutkan pengecekan pada objek ke-3 dengan ke-4
- Jika kriteria *pairing* tidak terpenuhi maka pengecekan dilanjutkan pada objek ke-1 dengan

objek ke-3 begitu seterusnya hingga objek terakhir. Jika tidak diketemukan yang memenuhi kriteria *pairing* maka objek ke-1 menjadi kandidat lampu depan dari kendaraan roda 2.

Setelah proses *pairing* selesai akan didapatkan pasangan lampu jika memenuhi ketiga kriteria *pairing*. Pasangan lampu yang telah didapatkan menjadi kandidat lampu depan dari kendaraan roda 4 sedangkan yang tidak berpasangan menjadi kandidat lampu depan dari kendaraan roda 2. Disebut sebagai kandidat karena pada *frame-frame* berikutnya tidak selalu terjadi *pairing* lampu seperti pada *frame* saat ini. Pada *frame* pertama, dua objek memenuhi kriteria *pairing* sehingga dapat di*pairing*kan namun pada *frame* kedua dua objek tersebut ternyata tidak memenuhi kriteria *pairing* sehingga gagal di*pairing*kan dan pada *frame* ketiga dua objek kembali memenuhi kriteria *pairing* sehingga dapat di*pairing*kan kembali. Beberapa kondisi yang dapat mempengaruhi proses *pairing* ini telah disebutkan sebelumnya.

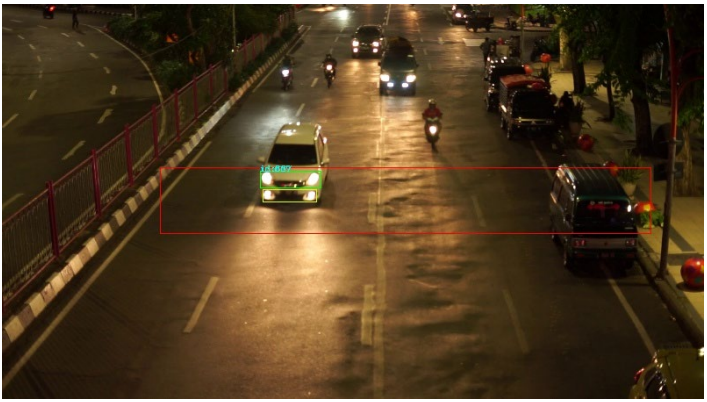
Letak objek lampu relatif terhadap citra *frame*nya, sehingga suatu objek lampu depan dari kendaraan bergerak tidak selalu berada di posisi yang sama pada setiap *frame* dari video. Untuk dapat mengenali suatu objek dan pergerakannya diperlukan adanya pelacakan (*tracking*) objek tersebut pada setiap *frame*.

j. *Tracking*

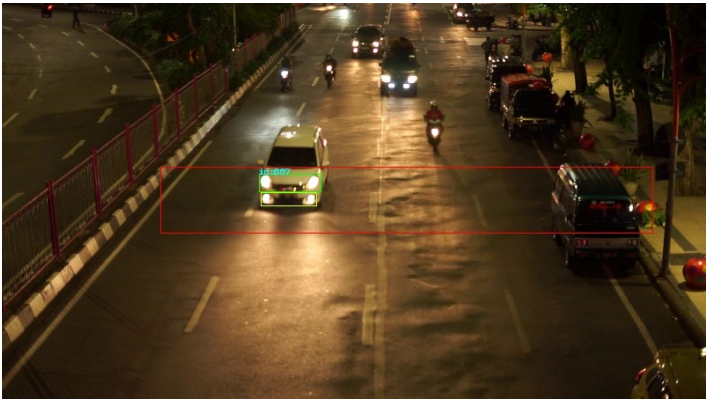
Pelacakan (*tracking*) suatu objek bergerak sangat berguna dalam aplikasi *computer vision* seperti untuk pengenalan gerakan, pelacakan kendaraan, dan penghitungan jumlah kendaraan. Pada tahap ini dilakukan manajemen objek, yaitu objek yang sudah dilacak diperiksa, masih ada di dalam ROI atau sudah keluar dari ROI yang telah ditentukan.

Objek yang telah melalui proses *pairing* selanjutnya akan diberikan label (id objek). Id objek bersifat unik, setiap objek akan mendapatkan id yang berbeda mulai dari angka 0. Id objek ini berguna untuk membedakan antara objek yang satu dengan objek yang lainnya dalam satu *frame*. Kemudian dilanjutkan dengan *tracking* masing-masing objek tersebut pada *frame-frame* berikutnya.

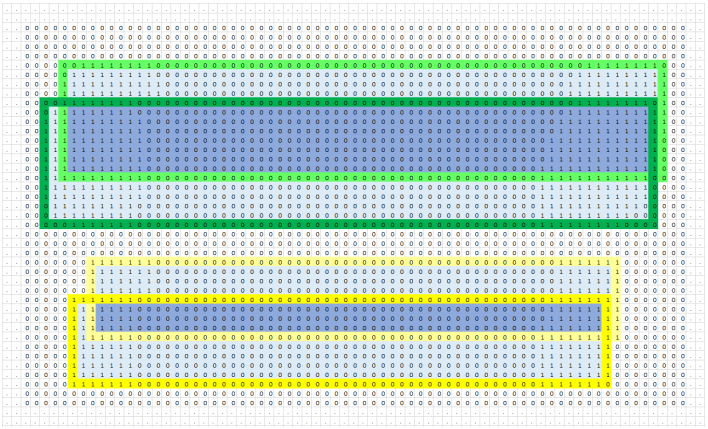
Tracking suatu objek dilakukan dengan memanfaatkan informasi letak relatif objek tesebut terhadap *frame*nya. Sehingga dapat dilakukan pembandingan letak suatu objek antara *frame* saat ini dengan *frame* sebelumnya. Jika didapatkan suatu area / titik yang merupakan irisan dari letak suatu objek pada *frame* saat ini dengan letak suatu objek pada *frame* sebelumnya maka objek tersebut merupakan suatu objek yang sama sehingga pada *frame* saat ini objek tersebut akan diberikan id objek sesuai dengan id objek pada *frame* sebelumnya dari objek yang letaknya beririsan tersebut. Namun, jika tidak didapatkan suatu area / titik irisan dari letak suatu objek pada *frame* saat ini dengan letak suatu objek pada *frame* sebelumnya, maka objek tersebut akan diberikan id baru sesuai dengan urutan id yang telah digunakan untuk memberi label pada objek sebelumnya.



(a)



(b)



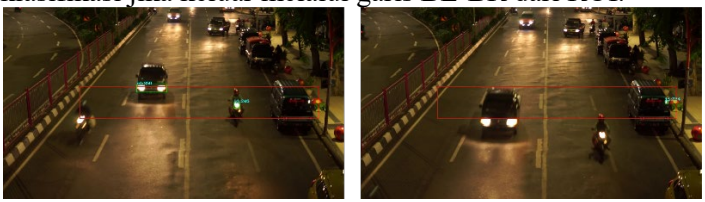
(c)

Gambar 4.25. Proses *tracking* objek, (a) *frame* ke-(i-1), (b) *frame* ke-i, (c) letak objek relatif terhadap *frame*

Pada Gambar 4.25 (c) didapatkan area yang merupakan irisan dari dua objek, area yang dibatasi warna hijau dan kuning yang lebih terang menunjukkan area objek pada *frame* sebelumnya, sedangkan area yang dibatasi warna hijau dan kuning yang lebih gelap menunjukkan area objek pada *frame* saat ini. Area berwarna biru yang lebih gelap menunjukkan area hasil irisan dua objek. Sehingga dapat disimpulkan bahwa dua objek ini merupakan objek yang

sama. Oleh karena itu pada Gambar 4.25 (a) objek diberi id 667 dan pada Gambar 4.25 (b) objek juga diberi id 667.

Proses *tracking* objek akan terus dilakukan sejak suatu objek memasuki area ROI hingga objek tersebut keluar dari area ROI. Id yang diberikan pada suatu objek saat memasuki area ROI akan menjadi identitas objek tersebut hingga keluar dari area ROI. Ketika objek sudah keluar maka pemberian id akan dihentikan dan akan dilakukan penghitungan terhadap objek tersebut berdasar hasil klasifikasi jika keluar melalui garis BL-BR dari ROI.



(a) (b)

Gambar 4.26. Pemberian id pada objek yang ditracking, (a) Objek memasuki area ROI, (b) Objek meninggalkan area ROI

k. *Classification*

Objek yang telah melalui proses *pairing* akan diberikan suatu informasi pada id dari objek tersebut. Jika kriteria *pairing* terpenuhi maka id objek akan menyimpan informasi bahwa objek tersebut merupakan kandidat lampu depan kendaraan roda 4, sedangkan jika kriteria *pairing* tidak terpenuhi maka id objek akan menyimpan informasi bahwa objek tersebut merupakan kandidat lampu depan kendaraan roda 2.

l. *Counting*

Program akan melakukan penghitungan jika terdeteksi terdapat suatu objek yang keluar dari area ROI melalui garis BL – BR. Jika objek tersebut telah menjadi kandidat lampu

depan kendaraan roda 4 maka banyak kendaraan roda 4 akan bertambah 1. Namun. Jika objek tersebut menjadi kandidat lampu depan kendaraan roda 2 maka banyak kendaraan roda 2 yang akan bertambah 1.

4.1.2 Analisis Kebutuhan Sistem

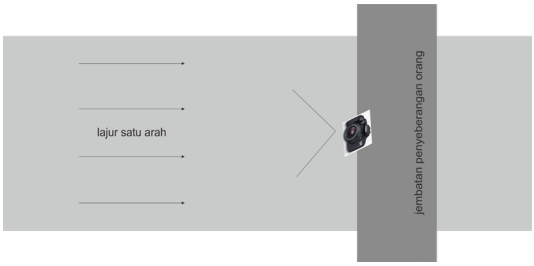
Perangkat lunak ini dibangun menggunakan bahasa pemrograman JAVA dengan library OPENCV. Selain itu juga diperlukan kamera digital untuk merekan arus kendaraan di jalan. Program ini dikembangkan menggunakan komputer dengan processor inter core-i5 dan RAM 8GB

4.2 Perancangan Sistem

4.2.1 Perancangan Data Sistem

a. Data Masukan

Data masukan sistem ini berupa video *offline* rekaman kendaraan di jalan yang diambil menggunakan kamera digital. Video diambil dengan posisi lensa kamera menjangkau lebar area jalan satu lajur. *Layout* untuk pengambilan video rekaman dapat dilihat pada Gambar 4.27



Gambar 4.27. Layout untuk perekaman video

File video hasil perekaman memiliki spesifikasi sebagai berikut,

- Format ekstensi rekaman video *offline* berupa .mp4

- Memiliki kemampuan merekam video dengan kecepatan minimal 25 *frame* per-*second*
- Memiliki resolusi 1280 x 720

b. Data Proses

Data proses merupakan data yang digunakan dalam proses pengolahan data masukan. Data proses ini diperoleh dari hasil pengolahan data masukan sesuai dengan tahapan algoritma dan metode yang telah disusun. Tahapan dari data proses dijelaskan pada Tabel 4.1

Tabel 4.1. Tabel data proses

No	Tahapan	Masukan	Keluaran
1	Akuisisi	Video	Citra RGB
2	*Grayscale*	Citra RGB	Citra *Grayscale*
3	*Noise Filtering*	Citra *Grayscale*	Citra *Grayscale* tanpa *noise*
4	Ekstraksi Objek Terang	Citra *Grayscale* tanpa *noise*	Citra hitam putih (*Biner*)
5	Operasi Morfologi	Citra *Biner*	Citra B*iner* hasil operasi morfologi
6	Deteksi Objek	Citra *Biner* hasil operasi morfologi	Citra dengan objek terdeteksi
8	*Tracking* dan *Pairing*	Citra dengan objek terdeteksi	Citra dengan objek telah dipasangkan
9	*Clasification*	Citra dengan objek telah dipasangkan	Pengklasifikasian jenis kendaraan
10	*Counting*	Pengklasifikasian jenis kendaraan	Penghitungan banyak kendaraan

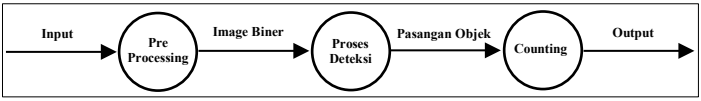
c. Data Keluaran

Data keluaran merupakan hasil pengklasifikasian kendaraan pada waktu malam hari dan hasil perhitungan jumlah kendaraan yang melintas. Program dalam Tugas Akhir ini menghasilkan video dengan menampilkan hasil *pairing* lampu depan kendaraan.

4.2.2 Perancangan Proses

a. Perancangan Proses Algoritma

Proses algoritma ini dimulai dengan proses *segmentasi* citra yaitu proses *grayscale, noise filtering*, ekstraksi objek terang, dan operasi morfologi. Setelah melakukan proses segmentasi, dilanjutkan ke proses pendeteksian bagian lampu kendaraan (objek) yang sehingga didapatkan banyak objek kemudian dilakukan *pairing* untuk memasangkan 2 objek yang berdekatan sesuai dengan kriteria *pairing.* Proses selanjutnya yaitu mengklasifikasikan jenis kendaraan berdasar pasangan objek yang terdeteksi dan melakukan penghitungan jumlah kendaraan yang terdeteksi.



Gambar 4.28. *Data Flow Diagram* sistem klasifikasi dan penghitungan kendaraan bergerak pada malam hari

b. Perancangan Proses *Pre-processsing* dan Deteksi Lampu

Pada proses *pre-processing* dilakukan perubahan warna citra yaitu perubahan citra RGB menjadi citra *grayscale*. Setelah mendapatkan citra *grayscale*, kemudian dilakukan *noise filtering* untuk menghilangkan noise yang ada pada citra. Selanjutnya citra dikenakan *thresholding* untuk mendapatkan citra biner, sehigga objek yang terang akan

berwarna putih dan yang lain berwarna hitam. Kemudian, operasi morfologi dilakukan untuk memperbaiki objek yang berwarna putih agar lebih jelas.

c. Perancangan Proses *Pairing*

Setelah mendapatkan bagian lampu kendaraan, proses *pairing* dilakukan dengan memasangkan lampu-lampu kendaraan yang dekat secara horizontal dan perbedaan sangat kecil pada jarak vertikal serta memiliki luas objek yang identik.

d. Perancangan Proses *Classification*

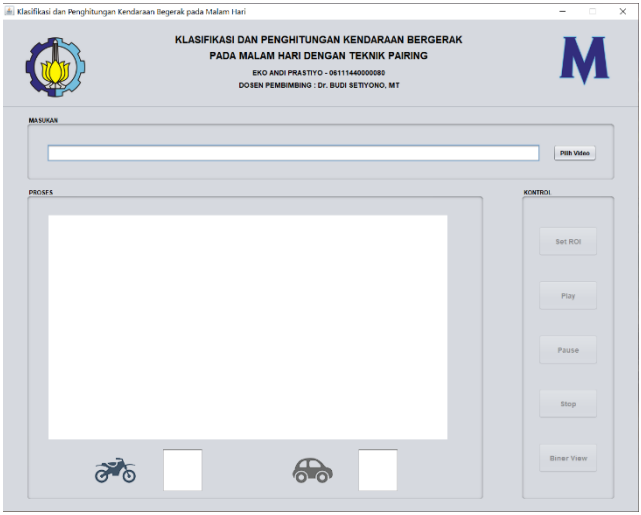
Lampu-lampu kendaraan yang berhasil dipasangkan akan diklasifikasikan sebagai kendaraan roda empat dan yang tidak berhasil dipasangkan adalah kendaraan roda dua.

e. Perancangan Proses *Counting*

Setelah objek diklasifikasikan kemudian akan dilakukan proses penghitungan jumlah kendaraan di jalan raya pada malam hari berdasarkan lampu kendaraan yang dideteksi.

4.2.3 Perancangan Antar Muka Sistem

Desain antarmuka sistem dibutuhkan untuk memudahkan pengguna dalam mengoperasikan perangkat lunak yang dibangun. Pada gambar 4.29 menunjukkan desain antarmuka sistem. Secara detail pada desain antarmuka sistem ini terdapat tombol “pilih video”, tombol “set ROI”, tombol “Play”, tombol “Pause”, tombol “Stop”, tombol “Biner View”, label dengan logo ITS, label logo Matematika, label nama program, label nama pembuat, label nama pembimbing, label *frame* dan *textfield* yang menunjukkan lokasi penyimpanan video serta *textfield* yang menunjukkan banyak kendaraan roda 2 atau 4 yang tedeteksi.



Gambar 4.29. Desain antar muka sistem

4.3 Implementasi Sistem

4.3.1 Implementasi *Input* Video

Pada proses ini ditentukan video yang digunakan sebagai data masukan untuk diproses. Penjabaran tentang proses pemilihan video adalah sebagai berikut,

Fungsi : menginputkan video bertipe *.mp4 atau *.mov pada sistem

Input : video bertipe *.mp4 atau *.mov

Deskripsi : mengambil video bertipe *.mp4 atau *.mov yang telah disimpan pada perangkat

Kode program untuk menerima *input* video adalah sebagai berikut :

```
JFileChooser fc = new
JFileChooser("C:\\Users\\Eko\\Desktop\\semhas\\vide
o");
fc.setFileSelectionMode(JFileChooser.FILES_ONLY);
FileNameExtensionFilter filter = new
FileNameExtensionFilter("Video Files", "avi",
"mp4", "mpg", "mov", "mkv", "mpeg");
fc.setFileFilter(filter);
fc.setMultiSelectionEnabled(false);
fc.setAcceptAllFileFilterUsed(false);
```

4.3.2 Implementasi Proses *Grayscale*

Pengubahan citra video yang awalnya adalah citra dalam bentuk RGB menjadi citra *grayscale*. Berikut ini adalah kode program untuk mengubah citra RGB menjadi citra *grayscale*.

```
Imgproc.cvtColor(foregroundImage, foregroundImage,
Imgproc.COLOR_BGR2GRAY);
```

4.3.3 Implementasi *Noise Filtering*

Citra *grayscale* dihilangkan *noise*-nya pada proses *noise filtering* menggunakan metode *gaussian filter*. Berikut ini adalah kode program untuk *filtering noise*.

```
Imgproc.GaussianBlur(foregroundClone,
foregroundImage, new Size(5, 5), 0, 0);
```

4.3.4 Implementasi Ekstraksi Objek Terang

Setelah didapatkan citra *grayscale* yang bersih dari *noise*, kemudian memasuki proses pengubahan citra *grayscale* menjadi citra biner. Objek diubah menjadi warna putih dan selain objek akan diubah menjadi warna hitam. Berikut ini adalah kode program untuk ekstraksi objek terang.

```
Imgproc.threshold(foregroundImage, foregroundThres,
200, 255, Imgproc.THRESH_BINARY);
```

4.3.5 Implementasi Operasi Morfologi

Operasi morfologi dilakukan setelah mendapatkan hasil dari ekstraksi objek terang. Pada operasi morfologi ini dilakukan erosi dan dilasi untuk membentuk objek agar lebih jelas. Berikut ini adalah kode program untuk operasi morfologi.

```
Mat kernel1 = Mat.ones(3, 3, CvType.CV_8U);
Imgproc.erode(foregroundThres, foregroundThres,
kernel1);
Mat kernel2 = Mat.ones(7, 7, CvType.CV_8U);
Imgproc.dilate(foregroundThres, foregroundThres,
kernel2);
```

4.3.6 Implementasi Deteksi Objek

Pada tahap ini akan dilakukan proses mendapatkan kontur objek sehingga objek akan terdeteksi dengan jelas.

```
Mat hierarchy = new Mat();
List<MatOfPoint> contours = new ArrayList<>();
Imgproc.findContours(foregroundThres, contours,
hierarchy, Imgproc.RETR_LIST,
Imgproc.CHAIN_APPROX_SIMPLE);
Rect r = null;
Vector<Rect> rect_array = new Vector<Rect>();

for (int i = 0; i < contours.size(); i++) {
   r = Imgproc.boundingRect(contours.get(i));
   rect_array.add(r);
}
hierarchy.release();
array = rect_array;
```

4.3.7 Implementasi *Pairing*

Untuk melakukan *pairing* diperlukan titik tengah dari lampu kendaraan. Setelah diperoleh titik tengah dari lampu kendaraan, dapat ditentukan jarak lampu dengan lampu tetangga yang terdekat dari kiri ke kanan.

```
int pusatX0 = (int) ((startarray.get(i).tl().x +
startarray.get(i).br().x) / 2);
int pusatX1 = (int) ((startarray.get(i - 1).tl().x
+ startarray.get(i - 1).br().x) / 2);
int pusatY0 = (int) ((startarray.get(i).tl().y +
startarray.get(i).br().y) / 2);
int pusatY1 = (int) ((startarray.get(i - 1).tl().y
+ startarray.get(i - 1).br().y) / 2);

if (Math.abs(pusatY0 - pusatY1) <= 4 &&
Math.abs(pusatX0 - pusatX1) >= 40 &&
Math.abs(pusatX0 - pusatX1) <= 100 &&
Math.abs(startarray.get(i).area() -
startarray.get(i - 1).area()) <= 105) {
   Rect newrect = new Rect(startarray.get(i).br().x
> startarray.get(i - 1).br().x ?
startarray.get(i).br() : startarray.get(i -
1).br(), startarray.get(i).tl().x <
startarray.get(i - 1).tl().x ?
startarray.get(i).tl() : startarray.get(i -
1).tl());
startarray.set(i-1, newrect);

startarray.remove(i);
objekdihapus++;

ObjectCenterX = (int) (startarray.get(i-1).tl().x +
startarray.get(i-1).br().x) / 2;
ObjectCenterY = (int) (startarray.get(i-1).tl().y +
startarray.get(i-1).br().y) / 2;
Point ptpairing = new Point(ObjectCenterX,
ObjectCenterY);
pairingdetections.add(ptpairing);
```

4.3.8 Implementasi *Tracking*

Pelacakan objek yang sama dalam setiap frame dilakukan dengan mengecek apakah ada titik yang beririsan dari objek yang terdeteksi pada *frame* saat ini dengan objek yang telah terdeteksi pada *frame* sebelumnya.

```
for (Pairing.Obj obj : listObj) {
   Rect objRect = obj.getRect();
   if (isOverLap(objRect, startarray.get(i-1)) &&
!obj.isOut){

      check = true;
      listObj.set(obj.getId(), new
Pairing.Obj(startarray.get(i-1), obj.getId(), 4,
obj.getRoda(), false));
      obj.addRoda(4);
      listObjIn.add(obj.getId());
      Imgproc.rectangle(imag, startarray.get(i-
1).tl(), startarray.get(i-1).br(), new Scalar(0,
255, 0));
      Imgproc.putText(imag, "id:" + obj.getId(),
startarray.get(i-1).tl(),
Core.FONT_HERSHEY_COMPLEX_SMALL, 0.5, new
Scalar(255, 255, 0));
      break;
   }
}
```

4.3.9 Implementasi Klasifikasi dan Penghitungan Kendaraan

Objek yang telah diklasifikasikan sebagai kendaraan roda 4 ataupun roda 2 akan diberikan label sehingga dapat dilakukan penghitungan setelah objek tersebut keluar dari ROI melalui garis ROI yang sejajar dengan garis masuk ROI.

```
for (Pairing.Obj obj : listObj){
   Rect objRect = obj.getRect();
   if (objRect.x >= ROI.tl().x && objRect.x <=
ROI.br().x){
      if ((objRect.tl().y+objRect.br().y)/2 >=
ROI.br().y){
         if (!idcounting.contains(obj.id)){
            idcounting.add(obj.id);
            int roda4 = 0, roda2 = 0;
            for (int i = 0; i <
obj.getRoda().size(); i++){
               if
(obj.getRoda().get(i).equals(2))
                  roda2++;
               else
                  roda4++;
            }
            if (roda4 >= roda2){
               cars++;
               txroda4.setText(""+cars);
            }
            else{
               motors++;
               txroda2.setText(""+motors);
            }
         }
      }
   }
}
```

BAB V
UJI COBA DAN PEMBAHASAN

Pada bab ini dijelaskan mengenai pengujian program dan pembahasan dari hasil uji coba. Pengujian yang dilakukan adalah pengujian prgram dengan *input* video rekaman kendaraan bergerak di jalan raya. Pengambilan video dilakukan pada waktu malam hari dengan waktu yang berbeda-beda.

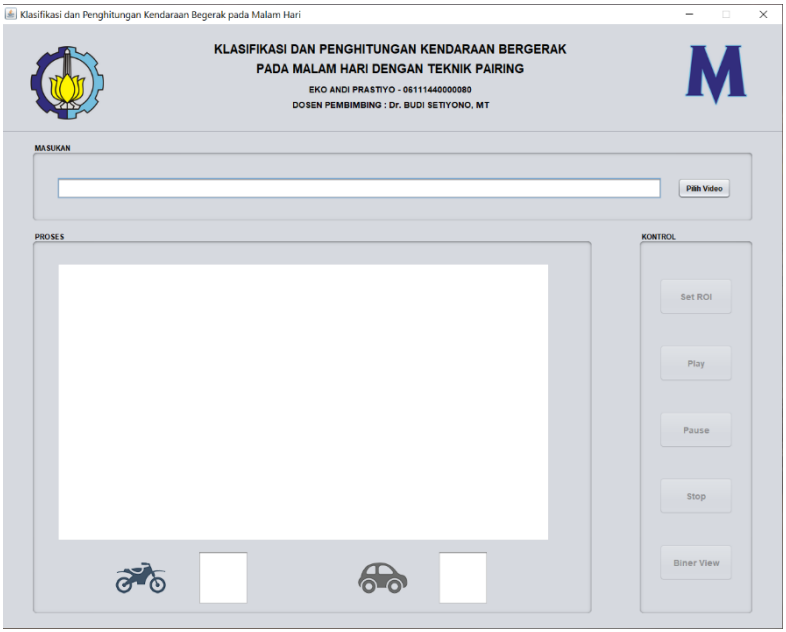
5.1 Data Uji Coba

Uji coba pada program dalam Tugas Akhir ini dilakukan terhadap video *.mp4 atau *.mov. Video uji coba sudah tersimpan dalam penyimpanan perangkat dan diperoleh dari hasil rekaman. Video yang digunakan memiliki piksel 1280 × 720 dengan *frame rate* minimal sebesar 25 *frame*/detik. Daftar input uji coba tersebut disajikan dalam Tabel 5.1

Tabel 5.1. Daftar data uji

No	NamaVideo	Durasi	Tempat
1	Video1	1 menit	Jembatan penyeberangan di Jalan A. Yani
2	Video2	1 menit	Jembatan penyeberangan di Jalan St. Wonokromo
3	Video3	1 menit	Jembatan penyeberangan di Jalan Urip Sumoharjo
4	Video4	1 menit	Jembatan penyeberangan di Surabaya Plaza

5.2 *Graphic User Interface (GUI)* Program



Gambar 5.1. Tampilan GUI

Halaman utama GUI ini memiliki desain antar muka yang ditunjukkan pada Gambar 5.1. Pada halaman ini dilakukan proses pemilihan video dan penentuan ROI. Halaman ini juga menampilkan hasil deteksi kendaraan. Antarmuka halaman utama terdiri dari :

- Tombol "pilih video", berfungsi untuk memilih video input yang akan ditampilkan dan digunakan untuk proses klasifikasi dan penghitungan
- Tombol "set ROI", berfungsi untuk membentuk area pendeteksian kendaraan bergerak pada video
- Tombol "Play" berfungsi untuk memulai menjalankan video per *frame* nya

- Tombol “Pause” berfungsi untuk menjeda jalannya video per *frame*
- Tombol “Stop”, berfungsi untuk menghentikan jalannya video per *frame*
- Tombol “Biner View”, berfungsi untuk menampilkan video dalam citra biner
- Label dibagian atas GUI, untuk menampilkan logo ITS, logo Matematika, nama program, dan identitas pengembang dan pembimbing
- Label di proses berfungsi untuk menampilkan video per *frame*
- *Textfield* di masukan berfungsi untuk menampilkan tempat penyimpanan video
- *Textfield* di proses berfungsi untuk menampilkan jumlah kendaraan roda 4 atau roda 2 yang telah terdeteksi

5.3 Hasil Proses Tahapan

5.3.1 Proses *Preprocessing*

a. Proses Grayscale

Citra video yang telah dimasukkan akan di ekstraksi menjadi kumpulan *frame*. Citra *frame* ini kemudian akan dikenakan *bilateralfilter* untuk mengurangi intensitas cahaya pada citra yang bukan menjadi objek deteksi. Selanjutnya citra *frame* dalam RGB akan diubah menjadi citra *grayscale* untuk memudahkan dalam pendeteksian objek pada citra. Dengan memanfaatkan *library* OpenCV untuk mengubah citra RGB menjadi citra *grayscale*.

(a) (b)

(c)

Gambar 5.2. Hasil proses *preprocessing*, (a) Citra sebelum difilter, (b) Citra setelah diberi *bilateral filtering*, (c) Citra *grayscale*

b. Proses *Noise Filtering*

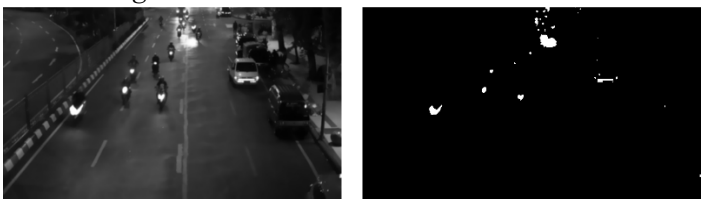
Pada Citra *grayscale* yang didapatkan dilakukan proses *Noise Filtering* untuk menghilangkan gangguan (*noise*) yang terdapat pada citra yang akan diteliti sehingga didapatkan hasil citra yang memiliki *noise* sedikit atau tidak ada sama sekali. Filter yang digunakan adalah *gaussianblur* karena filter ini sangat efektif untuk menghilangkan *noise* pada citra yang terlalu kontras.

(a) (b)

Gambar 5.3. Hasil proses *noise filtering*, (a) Citra sebelum *noise filtering*, (b) Citra setelah *noise filtering*

c. Proses Ekstraksi Objek Terang

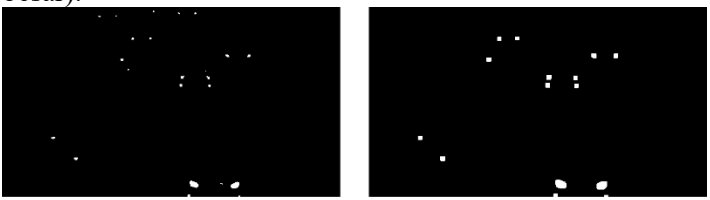
Dalam proses ini dibutuhkan suatu nilai batas yang disebut dengan nilai *threshold*. Nilai intensitas citra yang lebih dari atau sama dengan nilai *threshold* akan diubah menjadi 1 (berwarna putih) sedangkan nilai intensitas citra yang kurang dari nilai threshold akan diubah menjadi 0 (berwarna hitam). Sehingga citra keluaran dari hasil *thresholding* adalah citra biner.



Gambar 5.4. Hasil proses ekstraksi objek terang

d. Proses Morfologi

Proses ini bertujuan untuk memperbaiki hasil segmentasi pada citra biner. Proses morfologi yang digunakan adalah *opening* (pembukaan). *Opening* adalah proses erosi yang diikuti dengan dilasi. Dimulai dengan melakukan erosi pada citra kemudian hasil tersebut dilakukan dilasi. *Opening* ini digunakan untuk menghilangkan objek-objek kecil dan kurus serta dapat membuat tepi citra lebih *smooth* (untuk citra berukuran besar).



Gambar 5.5. Hasil proses morfologi

5.3.2 Proses Deteksi Objek

Setelah proses *preprocessing* kemudian dengan fungsi *findContours* pada OpenCV didapatkan kontur atau struktur dari setiap citra berwarna putih. Untuk setiap contour yang telah terbentuk, dilakukan *bounding* sehingga didapatkan beberapa bentuk objek dalam suatu *frame*.

5.3.3 Proses *Pairing*

Pada setiap objek yang telah terbentuk bisa didapatkan titik pusatnya. Dilakukan penghitungan selisih koordinat x dan y dari titik pusat serta selisih luas dua objek yang berdekatan. Maka dari itu dibuat batas selisih koordinat x, y dan luas untuk menghasilkan *pairing* yang baik. Batas yang telah ditentukan yaitu untuk selisih koordinat x antara 40 sampai 100 , untuk selisih koordinat y tidak lebih dari 4, dan selisih luasnya tidak lebih dari 50.

5.3.4 Proses *Tracking*

Untuk dapat mempertahankan informasi dari suatu objek maka perlu adanya proses pelacakan agar mampu mengenali objek yang sama pada *frame* yang berbeda. Proses pelacakan yang digunakan adalah dengan mengecek apakah terdapat titik irisan antara suatu objek pada *frame* sekarang dengan objek yang telah dideteksi pada *frame* sebelumnya.

5.3.5 Proses *Classification*

Objek yang telah memenuhi kriteria *pairing* akan diproses dan diberi nilai atribut bahwa objek itu kendaraan roda 4, begitu pula dengan objek yang gagal memenuhi kriteria *pairing*, akan diproses dan diberi informasi namun nilai atributnya adalah kendaraan roda 2.

5.3.6 Proses Counting

Sebelum dilakukan penghitungan, akan dilakukan pengecekan pada objek yang telah memasuki ROI. Jika objek tersebut sudah keluar dari ROI selanjutnya akan dilakukan

penghitungan berdasarkan nilai atribut yang telah diberikan. Jika nilai atributnya adalah 2 maka banyak kendaraan yang bertambah adalah motor dan jika nilai atributnya adalah 4 maka banyak kendaraan yang bertambah adalah mobil.

5.4 Uji Coba Video 1

Berikut adalah video kendaraan yang melewati jalan raya pada waktu malam hari. Video ini diambil di jembatan penyeberangan yang berada di Jalan A. Yani, Surabaya pada sekitar pukul 10 malam. Uji coba dilakukan sebanyak 4 kali dengan lebar ROI berbeda, yaitu 10 piksel, 20 piksel, 30 piksel, dan 40 piksel.

5.4.1 Hasil Proses Preprocessing

Pada Gambar 5.6 menunjukkan hasil dari salah satu *frame* pada video 1 setelah dilakukan proses *preprocessing*. Terlihat bahwa hasil dari proses *preprocessing* pada *frame* tersebut terdapat bagian berbentuk bulatan berwarna putih yang merupakan lampu depan dari kendaraan.

Gambar 5.6. Hasil *preprocessing* salah satu *frame* pada video 1

5.4.2 Hasil Deteksi dan *Pairing*

Pada Gambar 5.7 menunjukkan hasil deteksi dan *pairing* dari lampu kendaraan yang telah didapatkan dari proses *preprocessing*.

Gambar 5.7. Hasil deteksi dan *pairing* salah satu *frame* pada video 1

5.4.3 Hasil Tracking

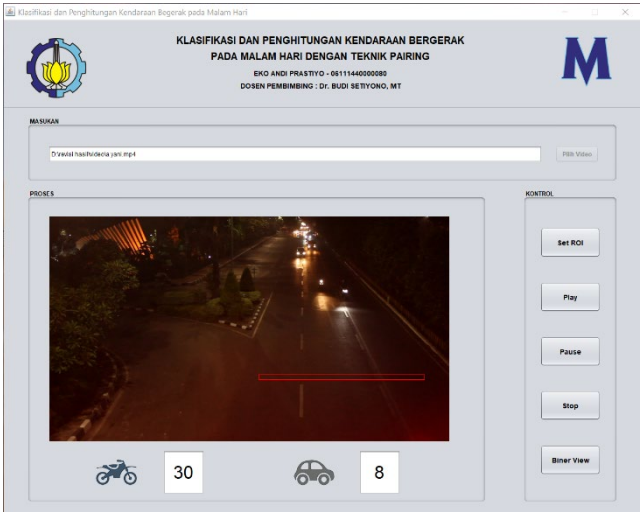
Pada Gambar 5.8 menunjukkan hasil *tracking* objek lampu depan kendaraan bergerak, sehingga objek yang sama akan tetap dikenali sebagai satu objek di setiap *frame* di dalam ROI.

Gambar 5.8. Hasil *tracking* salah satu *frame* pada video 1

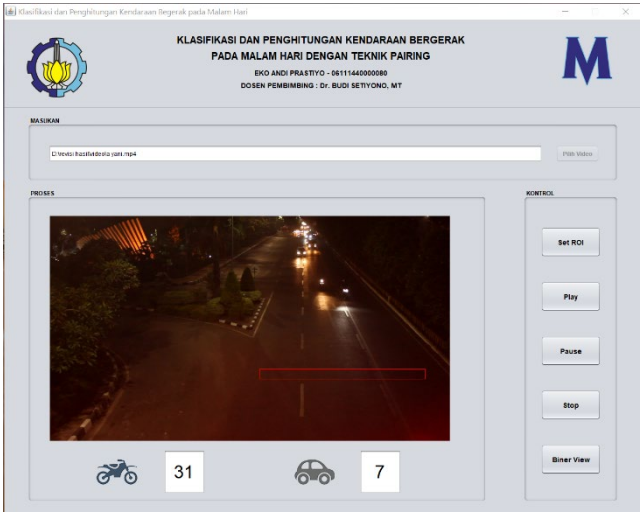
5.4.4 Hasil Classification dan Counting

Lampu depan yang telah berhasil dideteksi akan diberikan label dan identitas untuk mengklasifikasikan lampu depan tersebut apakah lampu depan dari kendaraan roda 2 atau lampu depan dari

kendaraan roda 4. Kemudian dilakukan penghitungan kendaraan setelah lampu depan tersebut keluar dari area ROI melalui garis BL- BR.



Gambar 5.9. Penghitungan video 1 dengan lebar ROI 10 piksel



Gambar 5.10. Penghitungan video 1 dengan lebar ROI 20 piksel

Gambar 5.11. Penghitungan video 1 dengan lebar ROI 30 piksel

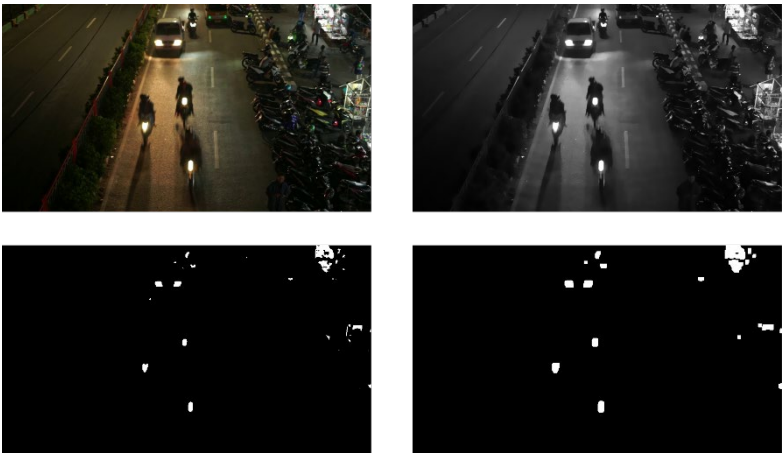
Gambar 5.12. Penghitungan video 1 dengan lebar ROI 40 piksel

5.5 Uji Coba Video 2

Berikut adalah video kendaraan yang melewati jalan raya pada waktu malam hari. Video ini diambil di jembatan penyeberangan yang berada di Jalan St. Wonokromo, Surabaya pada sekitar pukul 12 malam. Uji coba dilakukan pada video 2 dengan lebar ROI sebesar 40 piksel.

5.5.1 Hasil Proses Preprocessing

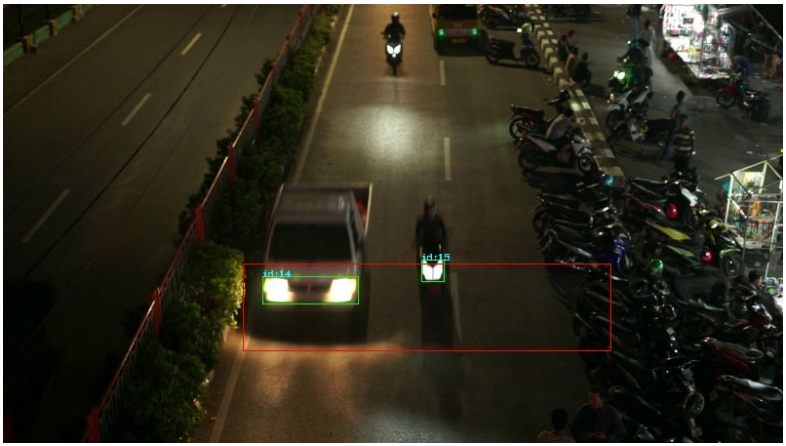
Pada Gambar 5.13 menunjukkan hasil dari salah satu *frame* pada video 2 setelah dilakukan proses *preprocessing*. Terlihat bahwa hasil dari proses *preprocessing* pada *frame* tersebut terdapat bagian berbentuk bulatan berwarna putih yang merupakan lampu depan dari kendaraan.



Gambar 5.13. Hasil *preprocessing* salah satu *frame* pada video 2

5.5.2 Hasil Deteksi dan *Pairing*

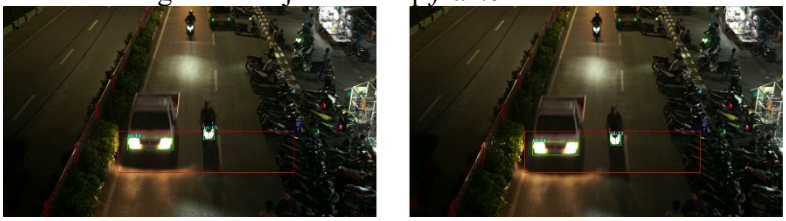
Pada Gambar 5.14 menunjukkan hasil deteksi dan *pairing* dari lampu kendaraan yang telah didapatkan dari proses *preprocessing*.



Gambar 5.14. Hasil deteksi dan *pairing* salah satu *frame* pada video 2

5.5.3 Hasil Tracking

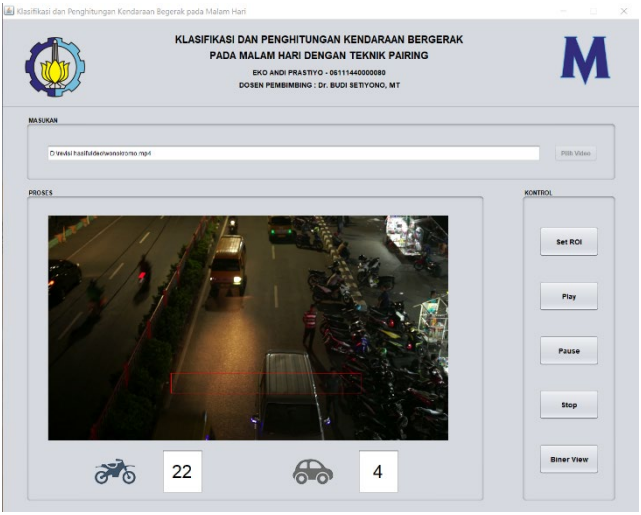
Pada Gambar 5.15 menunjukkan hasil *tracking* objek lampu depan kendaraan bergerak, sehingga objek yang sama akan tetap dikenali sebagai satu objek di setiap *frame* di dalam ROI.



Gambar 5.15. Hasil *tracking* salah satu *frame* pada video 2

5.5.4 Hasil Klasifikasi dan Penghitungan

Lampu depan yang telah berhasil dideteksi akan diberikan label dan identitas untuk mengklasifikasikan lampu depan tersebut apakah lampu depan dari kendaraan roda 2 atau lampu depan dari kendaraan roda 4. Kemudian dilakukan penghitungan kendaraan setelah lampu depan tersebut keluar dari area ROI melalui garis BL- BR.



Gambar 5.16. Penghitungan video 2 dengan lebar ROI 40 piksel

5.6 Uji Coba Video 3

Berikut adalah video kendaraan yang melewati jalan raya pada waktu malam hari. Video ini diambil di jembatan penyeberangan yang berada di Jalan Urip Sumoharjo, Surabaya pada sekitar pukul 1 malam. Uji coba dilakukan pada video 3 dengan lebar ROI sebesar 40 piksel.

5.6.1 Hasil Proses *Preprocessing*

Pada Gambar 5.17 menunjukkan hasil dari salah satu *frame* pada video 3 setelah dilakukan proses *preprocessing*. Terlihat bahwa hasil dari proses *preprocessing* pada *frame* tersebut terdapat bagian berbentuk bulatan berwarna putih yang merupakan lampu depan dari kendaraan.

Gambar 5.17. Hasil *preprocessing* salah satu *frame* pada video 3

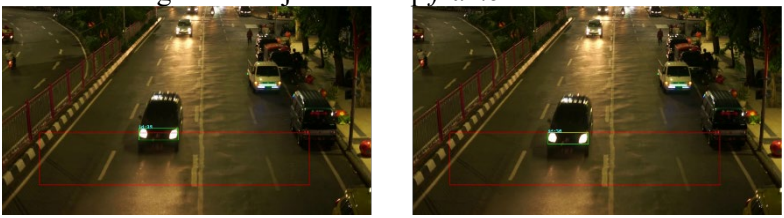
5.6.2 Hasil Deteksi dan *Pairing*

Pada Gambar 5.18 menunjukkan hasil deteksi dan *pairing* dari lampu kendaraan yang telah didapatkan dari proses *preprocessing*.

Gambar 5.18. Hasil deteksi dan *pairing* salah satu *frame* pada video 3

5.6.3 Hasil Tracking

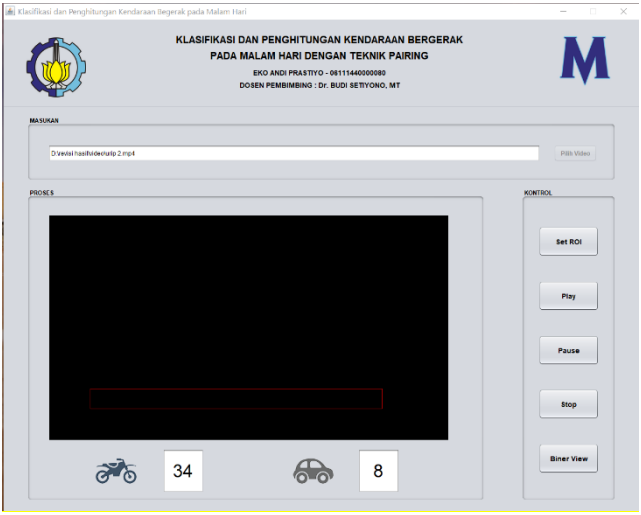
Pada Gambar 5.19 menunjukkan hasil *tracking* objek lampu depan kendaraan bergerak, sehingga objek yang sama akan tetap dikenali sebagai satu objek di setiap *frame* di dalam ROI.



Gambar 5.19. Hasil *tracking* salah satu *frame* pada video 3

5.6.4 Hasil Classification dan Counting

Lampu depan yang telah berhasil dideteksi akan diberikan label dan identitas untuk mengklasifikasikan lampu depan tersebut apakah lampu depan dari kendaraan roda 2 atau lampu depan dari kendaraan roda 4. Kemudian dilakukan penghitungan kendaraan setelah lampu depan tersebut keluar dari area ROI melalui garis BL- BR.



Gambar 5.20. Penghitungan video 3 dengan lebar ROI 40 piksel

5.7 Uji Coba Video 4

Berikut adalah video kendaraan yang melewati jalan raya pada waktu malam hari. Video ini diambil di jembatan penyeberangan yang berada di Surabaya Plaza pada sekitar pukul 7 malam. Uji coba dilakukan pada video 4 dengan lebar ROI sebesar 40 piksel.

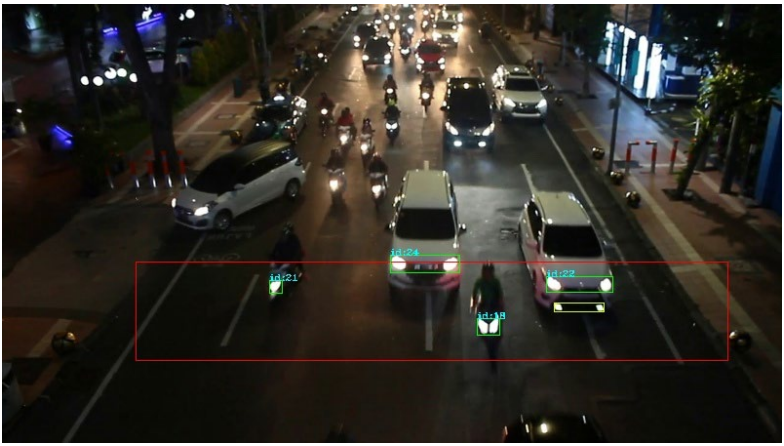
5.7.1 Hasil Proses *Preprocessing*

Pada Gambar 5.21 menunjukkan hasil dari salah satu *frame* pada video 4 setelah dilakukan proses *preprocessing*. Terlihat bahwa hasil dari proses *preprocessing* pada *frame* tersebut terdapat bagian berbentuk bulatan berwarna putih yang merupakan lampu depan dari kendaraan.

Gambar 5.21. Hasil *preprocessing* salah satu *frame* pada video 4

5.7.2 Hasil Deteksi dan *Pairing*

Pada Gambar 5.22 menunjukkan hasil deteksi dan *pairing* dari lampu kendaraan yang telah didapatkan dari proses *preprocessing*.



Gambar 5.22. Hasil deteksi dan *pairing* salah satu *frame* pada video 4

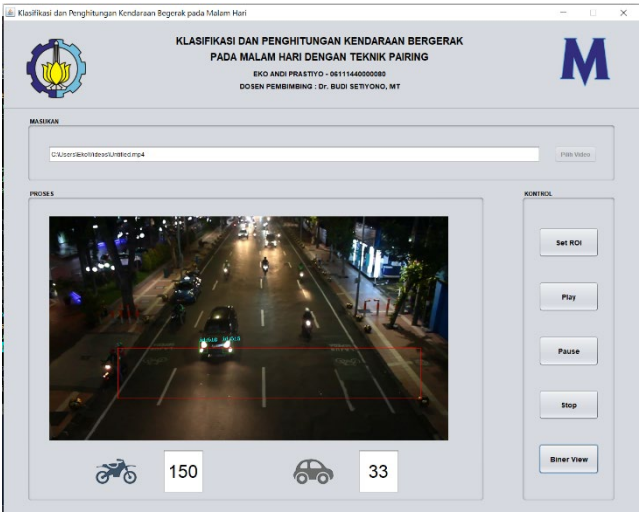
5.7.3 Hasil Tracking

Pada Gambar 5.23 menunjukkan hasil *tracking* objek lampu depan kendaraan bergerak, sehingga objek yang sama akan tetap dikenali sebagai satu objek di setiap *frame* di dalam ROI.

Gambar 5.23. Hasil *tracking* salah satu *frame* pada video 4

5.7.4 Hasil Classification dan Counting

Lampu depan yang telah berhasil dideteksi akan diberikan label dan identitas untuk mengklasifikasikan lampu depan tersebut apakah lampu depan dari kendaraan roda 2 atau lampu depan dari kendaraan roda 4. Kemudian dilakukan penghitungan kendaraan setelah lampu depan tersebut keluar dari area ROI melalui garis BL- BR.



Gambar 5.24. Penghitungan video 4 dengan lebar ROI 40 piksel

5.8 Pembahasan Hasil Uji Coba

Uji coba dilakukan pada 4 video yang direkam di tempat yang berbeda. Dalam uji coba tersebut, pada video 1 diberikan empat parameter berbeda yaitu berupa lebar ROI sebesar 10 piksel, 20 piksel, 30 piksel, dan 40 piksel. Hasil pengujian video 1 tersebut disajikan pada tabel 5.2.

Hasil klasifikasi dan penghitungan kendaraan disusun ke dalam tabel hasil dengan merujuk pada paper [14]. Data pada kolom hasil deteksi didapatkan dari hasil penghitungan oleh program yang telah dibuat. *False negative* menunjukkan banyaknya kendaraan yang tidak terdeteksi oleh program. Salah klasifikasi menunjukkan banyaknya hasil klasifikasi pada kelas yang salah. *False positive* menunjukkan kesalahan penghitungan kendaraan, seperti bukan objek yang terdeteksi sebagai objek dan objek yang terhitung dua kali.

Performa klasifikasi ditunjukkan pada kolom *recall* dan kolom *precision*. Kolom *recall* menunjukkan persentase dari data kategori positif yang diklasifikasi dengan benar oleh sistem. Kolom *precision* menunjukkan persentase dari jumlah data

kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif.

- $recall = \frac{\text{hasil deteksi} - \text{salah klasifikasi} - false\ positive}{\text{banyak kendaraan}} \times 100\%$
- $precision = \frac{\text{hasil deteksi} - \text{salah klasifikasi} - false\ positive}{\text{banyak kendaraan} - \text{salah klasifikasi}} \times 100\%$

Performa deteksi ditunjukkan pada kolom tingkat deteksi, kolom tingkat kesalahan deteksi, dan kolom rasio deteksi. Kolom tingkat deteksi (akurasi) menunjukkan persentase dari data yang terdeteksi oleh sistem. Kolom tingkat kesalahan deteksi menunjukkan persentase dari kesalahan deteksi oleh sistem. Kolom rasio deteksi menunjukkan persentase perbandingan dari banyak kendaraan hasil deteksi oleh sistem terhadap banyak kendaraan sebenarnya.

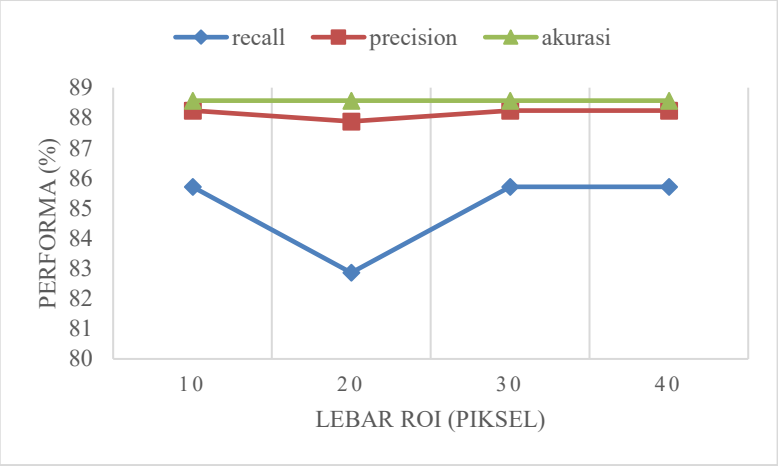
- tingkat deteksi = $1 - \frac{false\ negative}{\text{banyak kendaraan}} \times 100\%$
- tingkat kesalahan deteksi = $\frac{false\ positive}{\text{banyak kendaraan}} \times 100\%$
- rasio deteksi = $\frac{\text{hasil deteksi}}{\text{banyak kendaraan}} \times 100\%$

Tabel 5.2. Hasil klasifikasi dan penghitungan pada video 1

				Hasil			
Video	Lebar ROI	Klasifikasi	Banyak Kendaraan	Hasil Deteksi	False Negative	Salah Klasifikasi	False Positive
Video 1	10 piksel	roda 2	27	30	4	1	6
		roda 4	8	8	0	0	1
		total	**35**	**38**	**4**	**1**	**7**
	20 piksel	roda 2	27	31	4	2	6
		roda 4	8	7	0	0	1
		total	**35**	**38**	**4**	**2**	**7**
	30 piksel	roda 2	27	30	4	1	6
		roda 4	8	8	0	0	1
		total	**35**	**38**	**4**	**1**	**7**
	40 piksel	roda 2	27	30	4	1	6
		roda 4	8	8	0	0	1
		total	**35**	**38**	**4**	**1**	**7**

Video	Lebar ROI	Klasifikasi	Performa Klasifikasi		Performa Deteksi		
			Recall (%)	Precision (%)	Tingkat Deteksi (%)	Tingkat Kesalahan Deteksi (%)	Rasio Deteksi (%)
Video 1	10 piksel	roda 2	85,19	88,46			
		roda 4	87,5	87,5			
		total	**85,71**	**88,24**	**88,57**	**20**	**108,57**
	20 piksel	roda 2	85,19	92			
		roda 4	75	75			
		total	**82,86**	**87,88**	**88,57**	**20**	**108,57**
	30 piksel	roda 2	85,19	88,46			
		roda 4	87,5	87,5			
		total	**85,71**	**88,24**	**88,57**	**20**	**108,57**
	40 piksel	roda 2	85,19	88,46			
		roda 4	87,50	87,50			
		total	**85,71**	**88,24**	**88,57**	**20**	**108,57**

Berikut disajikan diagram garis yang menunjukkan keterhubungan lebar ROI dengan nilai *recall*, *precision*, dan akurasi.



Dari hasil diatas, pada lebar ROI 20 piksel didapatkan nilai *recall* dan *precision* terendah dibandingkan dengan nilai pada lebar ROI 10 piksel, 30 piksel, dan 40 piksel karena terjadi kesalahan klasifikasi pada satu kendaraan yaitu roda 4 yang terklasifikasikan sebagai roda 2.

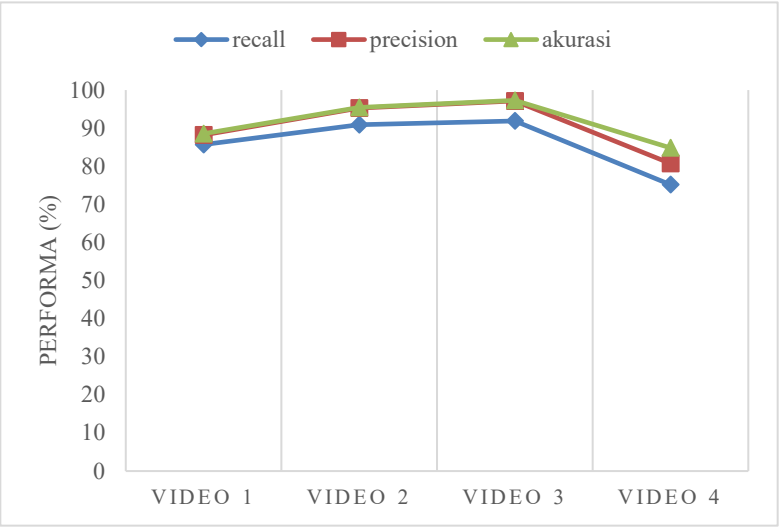
Dilakukan uji coba pada video 2, video 3, dan video 4 dengan lebar ROI 40 piksel yang pada hasil uji coba video 1 memiliki nilai *recall* dan *precision* serta akurasi tinggi.

Tabel 5.3. Hasil klasifikasi dan penghitungan video 1, video 2, video 3, dan video 4

				Hasil			
Video	Lebar ROI	Klasifikasi	Banyak Kendaraan	Hasil Deteksi	False Negative	Salah Klasifikasi	False Positive
Video 1	40 piksel	roda 2	27	30	4	1	6
		roda 4	8	8	0	0	1
		total	**35**	**38**	**4**	**1**	**7**
Video 2	40 piksel	roda 2	17	22	1	1	5
		roda 4	5	4	0	0	0
		total	**22**	**26**	**1**	**1**	**5**
Video 3	40 piksel	roda 2	27	34	1	2	6
		roda 4	10	8	0	0	0
		total	**37**	**42**	**1**	**2**	**6**
Video 4	40 piksel	roda 2	101	150	21	10	64
		roda 4	44	33	1	0	0
		total	**145**	**183**	**22**	**10**	**64**

			Performa Klasifikasi		Performa Deteksi		
Video	Lebar ROI	Klasifikasi	Recall (%)	Precision (%)	Tingkat Deteksi (%)	Tingkat Kesalahan Deteksi (%)	Rasio Deteksi (%)
Video 1	40 piksel	roda 2	85,19	88,46			
		roda 4	87,50	87,50			
		total	**85,71**	**88,24**	**88,57**	**20**	**108,57**
Video 2	40 piksel	roda 2	94,12	100			
		roda 4	80	80			
		total	**90,91**	**95,24**	**95,45**	**22,73**	**118,18**
Video 3	40 piksel	roda 2	96,30	104			
		roda 4	80	80			
		total	**91,89**	**97,14**	**97,30**	**16,22**	**113,51**
Video 4	40 piksel	roda 2	75,25	83,51648			
		roda 4	75	75			
		total	**75,17**	**80,74**	**84,83**	**44,14**	**126,21**

Berikut disajikan diagram garis yang menunjukkan keterhubungan lebar ROI dengan nilai *recall*, *precision*, dan akurasi dari sistem.



Dari hasil diatas, pada video 4 didapatkan hasil dengan performa terendah pada *recall*, *precision*, dan akurasi. Hal tersebut dikarenakan pada video 4 terjadi iluminasi yaitu pantulan lampu dari jembatan pada atap mobil sehingga sangat mempengaruhi hasil klasifikasi dan penghitungan kendaraan, terutama dalam penghitungan roda 2. Selain itu, ada beberapa kendaraan roda 2 dengan kecepatan tinggi sehingga tidak ter*tracking* dengan baik oleh sistem yang menyebabkan nilai akurasi sistem turun jika dibandingkan dengan akurasi pada video 1, video 2, dan video 3.

BAB VI
PENUTUP

Pada bab ini berisi kesimpulan yang diperoleh dari pembahasan pada bab sebelumnya serta saran untuk pengembangan penelitian selanjutnya.

6.1 Kesimpulan

Berdasarkan analisis terhadap hasil pengujian program dapat diambil kesimpulan sebagai berikut :

1. Teknik *pairing* berhasil diterapkan untuk mengklasifikasi suatu kendaraan yang melintasi jalan raya pada malam hari. Dua lampu depan di*pairing* jika titik pusat dua lampu depan memenuhi jarak horizontal dan vertikal tertentu serta memiliki perbedaan luas tertentu. Lampu yang di*pairing* diklasifikasi sebagai roda 4 dan yang tidak di*pairing* diklasifikasi sebagai roda 2.
2. Penghitungan banyak kendaraan yang melintasi jalan raya pada malam hari memberikan hasil yang baik berdasar pada objek lampu depan yang terdeteksi. Hasil penghitungan pada video 1 didapatkan 38 kendaraan dari 35 kendaraan pada kondisi sebenarnya, pada video 2 didapatkan 26 kendaraan dari 22 kendaraan pada kondisi sebenarnya, pada video 3 didapatkan 42 kendaraan dari 37 kendaraan pada kondisi sebenarnya, dan pada video 4 didapatkan 183 kendaraan dari 145 kendaraan pada kondisi sebenarnya.
3. Berhasil dikembangkan suatu program dengan menggunakan bahasa pemrograman Java dan *library* opencv untuk mengklasifikasi dan menghitung kendaraan yang melintasi jalan raya pada malam hari melalui video berdasarkan lampu depan kendaraan.

Tahapan program tersebut dalam mengklasifikasi dan menghitung yaitu *preprocessing* (*grayscale*, *noise filtering*, ekstraksi objek terang, dan operasi morfologi), *pairing*, *tracking*, *classification*, dan *counting*.

6.2 Saran

Terdapat beberapa hal yang penulis sarankan untuk pengembangan selanjutnya dengan melihat hasil yang dicapai pada penelitian ini yaitu :

1. Memilih *hardware* kamera digital yang memiliki fitur *illumination reduction* agar didapatkan gambar dengan lampu kendaraan yang jelas tanpa banyak terganggu oleh iluminasi.
2. Menambahkan metode untuk menghindari perbedaan ukuran lampu pada kendaraan yang sama.
3. Melakukan eksplorasi terhadap sudut pengambilan video agar didapatkan deteksi yang lebih baik

DAFTAR PUSTAKA

[1] Anonim. (2019). **Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis, 1949 - 2017**. https://www.bps.go.id/linkTableDinamis/view/id/1133, *posted* : 2019. Diakses pada tanggal 19-03-2019 pukul 23.13 WIB.

[2] Anonim. (2013). **Kisah Edi Warsito: Hitung Kendaraan Pemudik Secara Manual**. http://news.liputan6.com/read/662163/kisah-edi-warsito-hitung-kendaraan-pemudik-secara-manual, *posted* : 10/08/2013 14:43. Diakses pada tanggal 18-10-2017 pukul 19.03 WIB.

[3] Anonim. (2013). **Ada Alat Canggih untuk Hitung Kendaraan di Posko Nagreg**. http://ramadan.okezone.com/read/2013/08/05/335/847739/ a da-alat-canggih-untuk-hitung-kendaraan-di-poskonagreg, *posted* : 05/08/2013 15:40. Diakses pada tanggal 18-10-2017 pukul 20.14 WIB.

[4] Robert, K. (2009). **Night-Time Traffic Surveillance A Robust Framework for Multi-Vehicle Detection, Classification and Tracking**. International Journal of Computer Technologi and Application, Volume 5 Issues 2.

[5] Padmavathi, S. dan Gunasekaran, K. (2014). **Night Time Vehicle Detection for Real Time Traffic Monitoring Systems: A Review**. International Journal Computer Technologi and Applications, Volume 5(2),451-456.

[6] Hariyanto, Z. (2014). **Klasifikasi Jenis Kendaraan Bergerak Berbasis *Geometric Invariant Moment***. Tugas Akhir. Jurusan Matematika ITS.

[7] Padmavathi,S., Naveen, C. R. dan Kumari, V., H. (2016). **Vision based Vehicle Counting for Traffic Congestion Analysis during Night Time**. Indian Journal of Science and Technology, Vol 9(20).

[8] Robert, K. (2009). **Video-based Traffic Monitoring at Day and Night Time Vehicle Detection and Tracking**. IEEE

International Conference on Intelligent Transportation Systems, pp. 1-6.

[9] Gonzalez, R.C. dan Woods, R.E. (2002). **Digital Image Processing**. United States of America: Tom Robbins Publisher.

[10] Al Bovik. (2000). **Handbook of Image and Video Processing**. San Diego: Academic Press Publisher.

[11] Kanagamalliga,S. , Dr. Vasuki,S. , Kanimozhidevi,A. , Priyadharshmi,S. , dan Rajeswari,S. (2014). **Tracking and Counting The Vehicles in Night Scenes**. International Journal of Information Sciences and Techniques (IJIST), Vol. 4, No. 3

[12] Wei Zhang, Q.M.J. (2012). **Tracking and Pairing Vehicle Headlight in Night Scenes**. IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 1

[13] Satoshi Suzuki dan Keiichi Abe (1985). **Topological Structural Analysis of Digitized Binary Images by Border Following.** Computer Vision, Graphics, and Image Processing 30, 32-46

[14] Alain Crouzil, Louahdi Khoudour, dan Paul Valiere (2016). **Automatic Vehicle Counting System for Traffic Monitoring.** Journal of Electronic Imaging, vol. 25

BIODATA PENULIS

Penulis bernama lengkap Eko Andi Prastiyo, lahir di Tuban, 22 Juni 1996. Penulis menempuh pendidikan di SDN Rangkah 1 Surabaya, SMP Negeri 9 Surabaya, dan SMA Negeri 6 Surabaya. Kemudian penulis melanjutkan studi di Departemen Matematika FMKSD ITS dengan bidang ilmu komputer. Semasa menempuh perkuliahan di ITS, penulis turut aktif dalam beberapa kegiatan kemahasiswaan sebagai Konseptor OMITS 10th Himatika ITS Periode 2015/2016 dan Koordinator Konseptor OMITS 11th Himatika ITS Periode 2016/2017. Aktif di Lembaga Dakwah Jurusan Ibnu Muqlah sebagai staff Tim Big Event Periode 2015/2016 dan Ketua Tim Big Event Periode 2016/2017. Selama penulisan Tugas Akhir ini penulis tidak lepas dari kekurangan, kritik dan saran mengenai Tugas Akhir ini dapat dikirimkan melalui e-mail ke ekoandi@hotmail.com