

SEMINAR HASIL TUGAS AKHIR

PENGEMBANGAN PERANGKAT LUNAK DETEKSI KECEPATAN KENDARAAN BERGERAK BERBASIS PENGOLAHAN CITRA DIGITAL

Oleh :

Farah Fajriyah (1211 100 015)

Dosen Pembimbing :

Dr. Budi Setiyono, S.Si, MT

Jurusan Matematika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Institut Teknologi Sepuluh Nopember

Surabaya

2016

Penelitian ini bertujuan untuk mendeteksi kecepatan kendaraan bergerak di suatu jalan dengan menggunakan pengolahan citra digital. Objek dari penelitian ini adalah video rekaman arus kendaraan. Dimana dalam perekaman video tersebut dilakukan beberapa pengukuran parameter untuk proses pendeteksian kecepatan. Parameter yang harus diukur diantaranya yaitu besar sudut pemasangan kamera, tinggi pemasangan kamera dan jarak horizontal kamera ke daerah titik awal yang tercakup oleh kamera. Kemudian dilakukan pemilihan area yang digunakan sebagai area pendeteksian kecepatan kendaraan bergerak. Selanjutnya yaitu proses *Background Subtraction* dengan metode *Gaussian Mixture Model*. Citra *foreground* difilter dan dilakukan proses morfologi untuk menghilangkan *noise* pada citra. Tahap berikutnya, dicari titik pusat dari kendaraan untuk mengetahui jarak. Selain itu juga dicari waktunya yaitu dengan mencari banyaknya *frame* saat kendaraan berada di dalam ROI. Dengan mengetahui jarak dan waktu, maka dapat dideteksi kecepatan dari kendaraannya.

Kata Kunci--- Pengolahan citra digital, *Gaussian Mixture Model (GMM)*, *Speed Detection*.

I. PENDAHULUAN

II

III

IV

V

M

Latar Belakang

Rumusan
Masalah

Batasan Masalah

Tujuan

Manfaat

Berdasarkan data BPS
pada tahun 2013 jumlah
kendaraan mencapai
104.118.969 unit[1]



Fasilitas jalan tidak
memadahi yang
diperparah dengan
cara berkendara
yang tidak sesuai
dengan aturan



Rambu lalu lintas tanpa
adanya pengawasan
petugas dan kurangnya
pemanfaatan
pengawasan dengan
CCTV



Metode pengamatan
kecepatan kendaraan
dengan LIDAR atau
RADAR



Pengembangan
Perangkat Lunak
untuk mendeteksi
kecepatan
Kendaraan
Berbasis
Pengolahan Citra
Digital

Bagaimana membuat perangkat lunak untuk mendeteksi kecepatan kendaraan bergerak yang memanfaatkan informasi dari video digital?



I. PENDAHULUAN

II

III

IV

V

M

Latar Belakang

Rumusan
Masalah

Batasan
Masalah

Tujuan

Manfaat

Data yang digunakan adalah rekaman video kendaraan yang bergerak di jalan

Sudut pandang yang diambil ketika merekam adalah dari atas

Menggunakan kamera digital dengan resolusi 1280x720

Deteksi kecepatan kendaraan dilakukan pada objek yang masuk pada daerah ROI

Deteksi Kecepatan kendaraan dilakukan untuk *single obyek*

I. PENDAHULUAN

II

III

IV

V

M

Latar Belakang

Rumusan
Masalah

Batasan Masalah

Tujuan

Manfaat

Membuat perangkat lunak untuk mendeteksi kecepatan kendaraan bergerak yang memanfaatkan informasi dari video digital



I. PENDAHULUAN

II

III

IV

V

M

Latar Belakang

Rumusan
Masalah

Batasan Masalah

Tujuan

Manfaat

Memberikan informasi tentang kecepatan kendaraan yang melaju di jalan sehingga nantinya dapat dijadikan bahan pertimbangan bagi instansi pemerintah terkait untuk merumuskan kebijakan rekayasa lalu lintas seperti perencanaan pembangunan jaringan jalan yang tepat dan sesuai dengan beban yang melintasinya.

Penelitian
Sebelumnya

2.2

2.3

2.4

2.5

2.6

2.7

2.8

2.9

2.10

2.11

2.12

Chomtip Pornpanomchai, Kaweepap Kongkittisin (*Vehicle Speed Detection System*)[3]

- Berhasil mendeteksi kecepatan kendaraan dengan tingkat keberhasilan maksimal 95%[3].

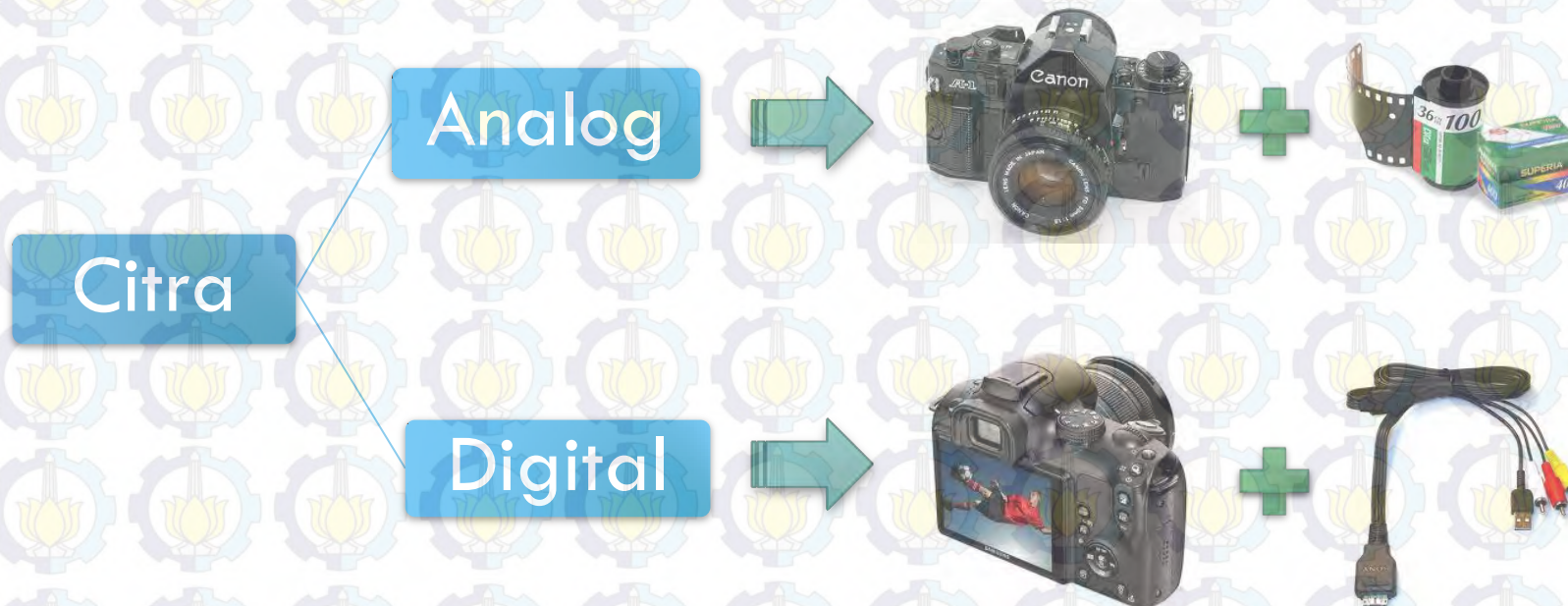
Huei-Yung Lin, Kun-Jhi Li, Chia-Hong Chang (*Vehicle Speed Detection from a Single Motion Blurred Image*)[2]

- Penelitian mengenai kecepatan kendaraan juga telah dilakukan dengan metode *blur image* dengan akurasi keberhasilan maksimal 95%[2].

Arash Gholami Rad, Abbas Dehgani dan Mohamed Rehan Karim (*Vehicle Speed Detection in Video Image Sequences using CVS Method*)[4]

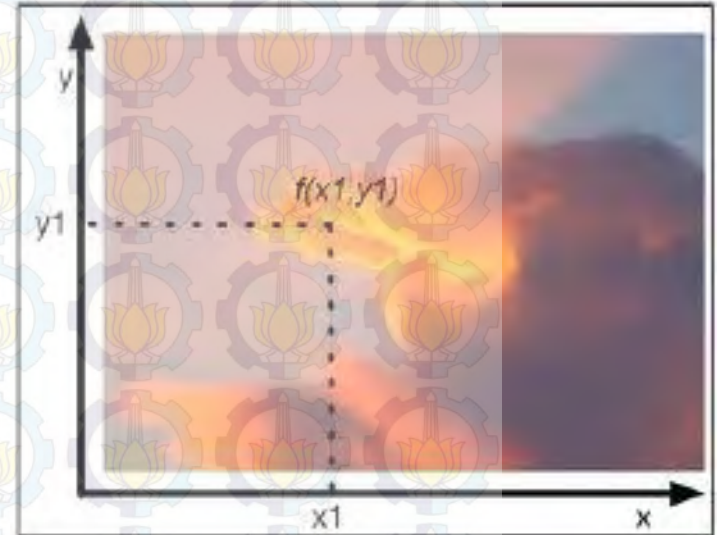
- Pendeteksian kecepatan kendaraan juga berhasil dilakukan dengan metode CVS dengan akurasi 96% dalam penelitian Arash, dkk[4]

- Citra memiliki makna rupa, gambar, atau gambaran (KBBI)
- Citra adalah suatu representasi, kemiripan, atau imitasi dari suatu objek atau benda (Webster)



Rafael C. Gonzales (*Digital Image Processing*)[5]

- Citra digital merupakan fungsi dua dimensi yang dapat dinyatakan dengan fungsi $f(x,y)$, dimana x dan y merupakan titik koordinat spasial. Dan amplitudo dari fungsi f pada setiap titik (x,y) merupakan nilai intensitas cahaya atau tingkat derajat keabuan representasi dari warna cahaya yang ada pada citra analog.
- Ketika (x,y) dan nilai intensitas dari f terbatas (*discrete quantities*), maka citra itu disebut sebagai citra digital.



Alan C. Bovik (*Handbook of Image and Video Processing*)[7]

- Video adalah teknologi untuk menangkap, merekam, memproses, menyimpan, dan merekonstruksi suatu urutan dari beberapa citra.
- Video digital juga merupakan hasil *sampling* dan kuantisasi dari video analog.
- Video digital dapat dinyatakan dengan fungsi 1 dimensi $V(t)$ yang terdiri dari beberapa citra digital $I(x,y,t)$, dimana (x,y) adalah nilai spasial dari fungsi I dan t menyatakan waktu.

Proses ekstraksi video ke beberapa unit *frame* bisa juga disebut dengan *scanning*



$V(t)$



frame 1



frame 2



frame 3



frame 4



frame 5

....

....



frame 1809

$I(x,y,t)$

2.1

2.2

2.3

Algoritma BS

2.5

2.6

2.7

2.8

2.9

2.10

2.11

2.12

Bharti, T. Tejinder (*Background Subtraction Techniques-Review*) [5]

- *Background subtraction (BS)* digunakan untuk mendapatkan objek yang bergerak pada serangkaian image.

Proses mencari obyek bergerak dala urutan frame yang dikenal sebagai pelacakan. Pelacakan ini dapat dilakukan dengan menggunakan ekstraksi ciri benda dan mendeteksi obyek/benda bergerak diurutan *frame*. Dengan menggunakan nilai posisi obyek di setiap *frame*, bisa digunakan untuk menghitung posisi dan kecepatan obyek bergerak[8].

Gaussian Mixture Model (GMM) merupakan metode yang tepat untuk berbagai kondisi yang terdapat pada citra, seperti *background* citra yang selalu statis, multimodal, maupun yang mengandung *noise* (gangguan atau objek yang tidak diinginkan terdapat pada citra)

Analisis *blob* ini menggunakan metode *connected component*, dimana di setiap kumpulan piksel yang tingkat keabuannya bernilai satu, dikategorikan sebagai satu objek. Setiap objek yang terdeteksi akan diberi label untuk mempermudah pengolahan.



Berdasarkan Gambar 2.3, jika diasumsikan tidak ada kendaraan yang lewat, maka : $\theta_1 = \arctan\left(\frac{D}{H}\right)$ (1)

dan sudut untuk daerah yang tidak terlihat oleh kamera adalah : $\theta_2 = \arctan\left(\frac{d_1}{H}\right)$ (2)

dengan :

θ_1 adalah sudut pemasangan kamera;

θ_2 adalah sudut daerah yang tidak terlihat oleh kamera;

θ_3 adalah sudut pandang yang dicakup oleh kamera;

H adalah tinggi kamera dari permukaan jalan;

D adalah jarak horizontal antara kamera dan kendaraan;

L adalah jarak nyata antara kamera dan kendaraan;

d_1 adalah daerah yang tidak terlihat oleh kamera;

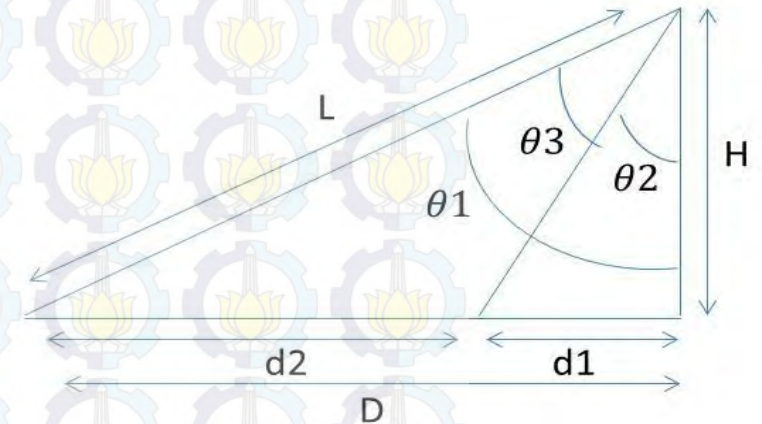
d_2 adalah daerah yang terlihat oleh kamera;

Selain itu juga, dapat diketahui bahwa : $\theta_3 = \theta_1 - \theta_2 \rightarrow \theta_2 = \theta_3 - \theta_1$ (3)

Daerah d_1 bisa diperoleh dengan : $d_1 = H \tan(\theta_1 - \theta_3) = H \tan \theta_2$ (4)

Dan untuk daerah d_2 bisa diperoleh dengan : $d_2 = D - d_1$ (5)

Setting Kamera dengan posisi dari atas



2.1

2.2

2.3

2.4

2.5

2.6

2.7

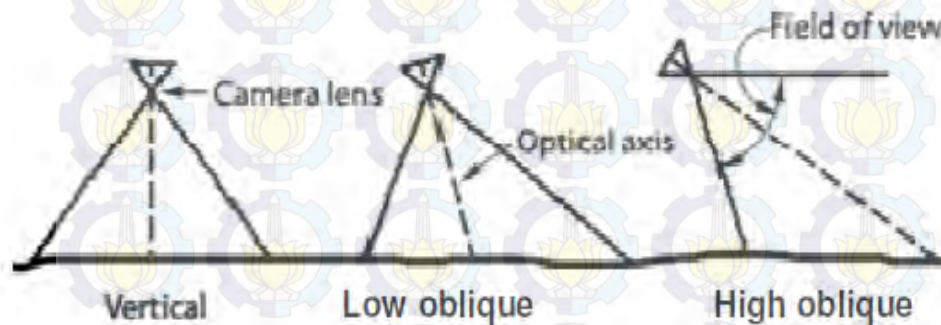
2.8

Kamera di
Atas

2.10

2.11

2.12



Kecenderungan dari kamera dengan posisi diatas akan menghasilkan gambar dari tiga jenis area seperti yang ditunjukkan pada gambar disamping yang bagian atas, sedangkan yang bagian bawah merupakan grid bagian garis yang terlihat pada berbagai jenis sudut kamera[4]

Tendensi Kamera dan tampilan grid hasil rekaman dari atas

Untuk mengetahui jarak yang ditempuh dalam pixel, dapat diperkirakan pixel menjadi suatu koordinat seperti berikut:

$$i = (a, b), \quad i - 1 = (e - f).$$

dimana : lokasi *centroid* ditunjukkan dalam *frame* i dan $i - 1$ untuk satu kendaraan;
dengan (a, b) mengkoordinasikan dan (e, f) berkoordinasi[4]

Perbedaan jarak jauh untuk kendaraan yaitu:

$$d_1 = \sqrt{(a - e)^2 + (b - f)^2} \quad (23)$$

dan jika gambar berurutan itu adalah 25 *frame* per detik, maka waktu antara dua *frame* berturut-turut sama dengan 0,04 s[4]

Dan untuk menghitung kecepatan dapat menggunakan persamaan berikut :

$$V = K \frac{\Delta x}{\Delta t} \quad (24)$$

dimana K adalah koefisien uji[4]

Studi Litelatur



- Pengkajian tentang algoritma pendeteksian kecepatan kendaraan.
- Perekaman video kendaraan yang bergerak di jalan.
- Studi ini dilakukan dengan membaca buku, jurnal, ataupun artikel yang terkait serta melakukan pengambilan video di halaman sekolah.
- Video diambil menggunakan kamera digital, yang dimana dalam pengambilan video tersebut dilakukan beberapa pengukuran parameter diantaranya yaitu besar sudut pemasangan kamera, tinggi pemasangan kamera dan jarak horizontal kamera ke daerah titik awal yang tercakup oleh kamera. Rekaman video diambil berdasarkan ketentuan berikut:
- Sudut pandang yang diambil ketika merekam adalah dari atas dengan jangkauan pandangan seluruh jalan arus satu arah.
- Menggunakan kamera digital dengan resolusi 1280x720.

Analisis dan Desain Sistem

Pada tahap ini akan dilakukan :

- analisis video
- penentuan parameter-parameter apa yang akan dibutuhkan dalam pembuatan program.
- Kemudian dibuat desain sistem dari program sesuai dengan hasil analisis.

Pada tahap akan dilanjutkan dengan :

- Implementasi hasil analisis dan desain sistem
- Perancangan program pada software matlab

Uji Coba dan Pembahasan

Pada tahap ini dilakukan:

- pengujian terhadap perangkat lunak yang telah selesai dibuat
- Kemudian dilakukan analisis
- uji coba deteksi kecepatan kendaraan, kemudian dihitung tingkat akurasi dari program menggunakan persamaan (3.1).

$$PK = \frac{KS - \text{abs}(KS - KP)}{KS} \times 100 \quad (3.1)$$

dengan PK adalah prosentase keberhasilan, KP adalah kecepatan kendaraan yang terdeteksi dengan benar, dan KS adalah kecepatan kendaraan sebenarnya.



Penarikan Kesimpulan

- Kesimpulan diambil sebagai hasil uji coba dan evaluasi perbaikan program serta saran pengembangan sistem

Pada tahap ini menuliskan semua hasil yang telah didapatkan selama pengerjaan Tugas Akhir.

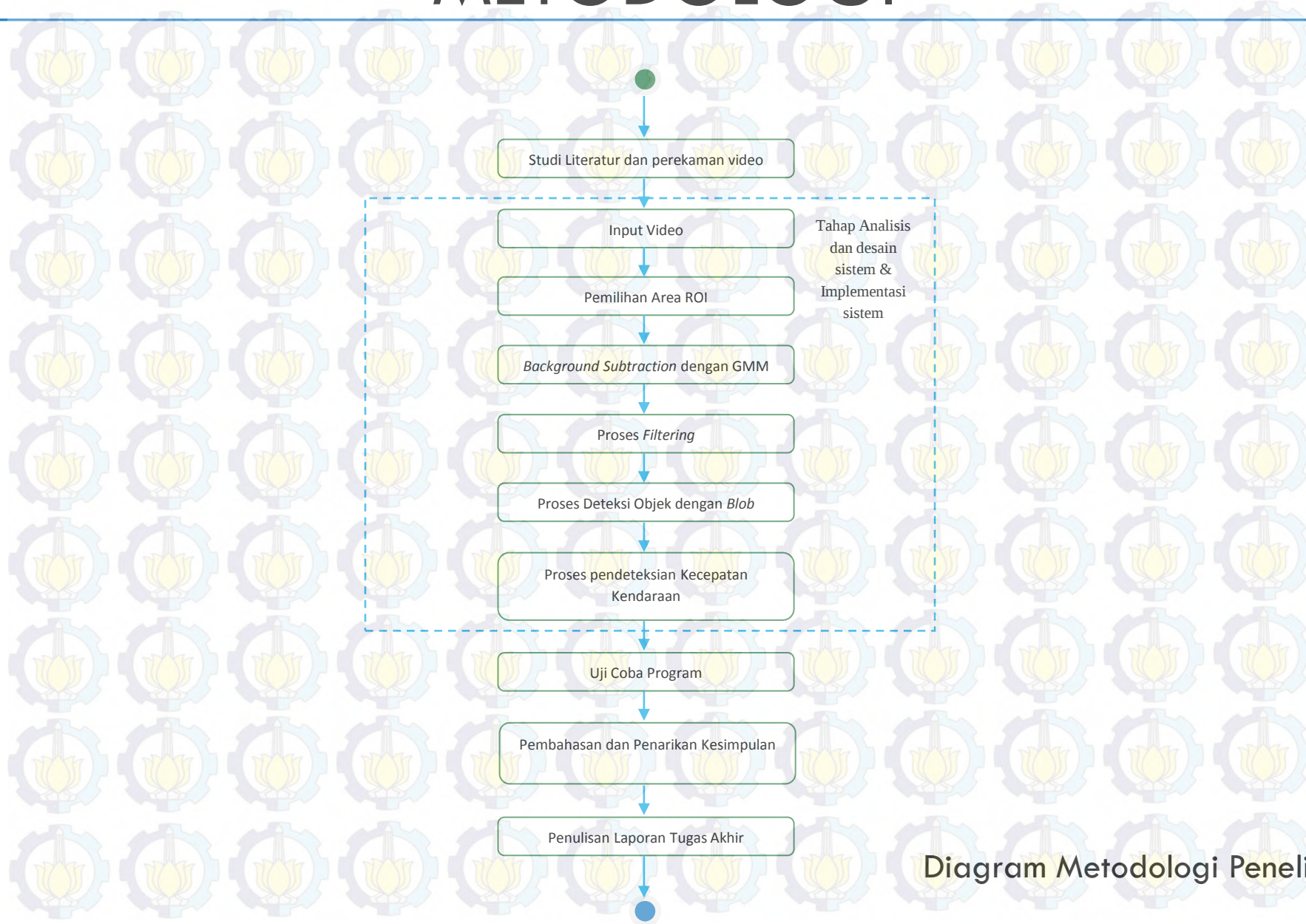


Diagram Metodologi Penelitian

Perancangan dan Implementasi

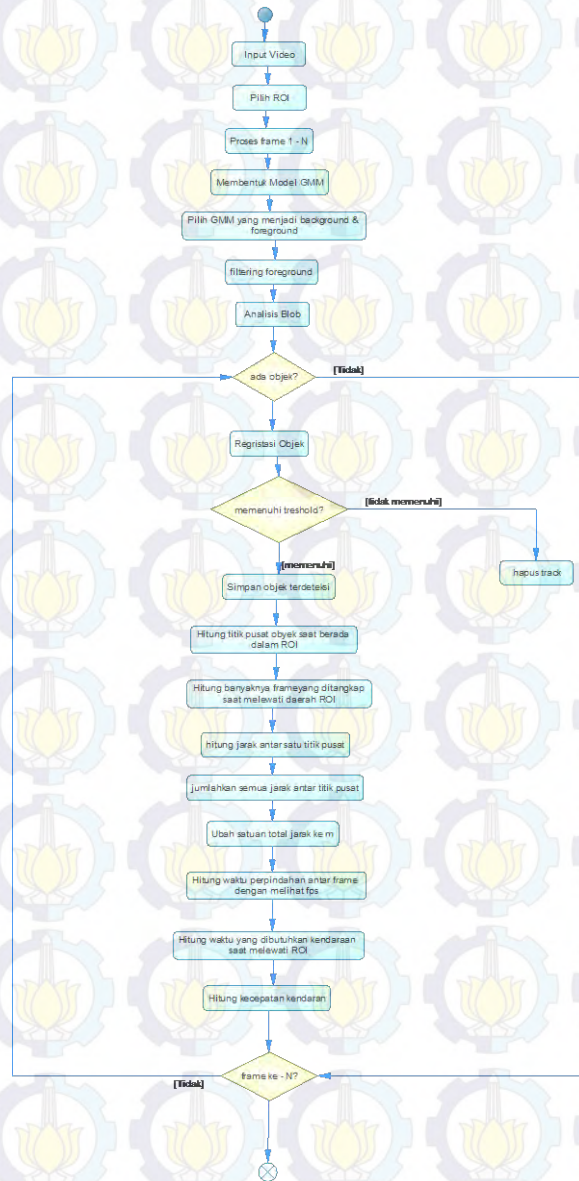


Diagram Alir Sistem Perangkat Lunak

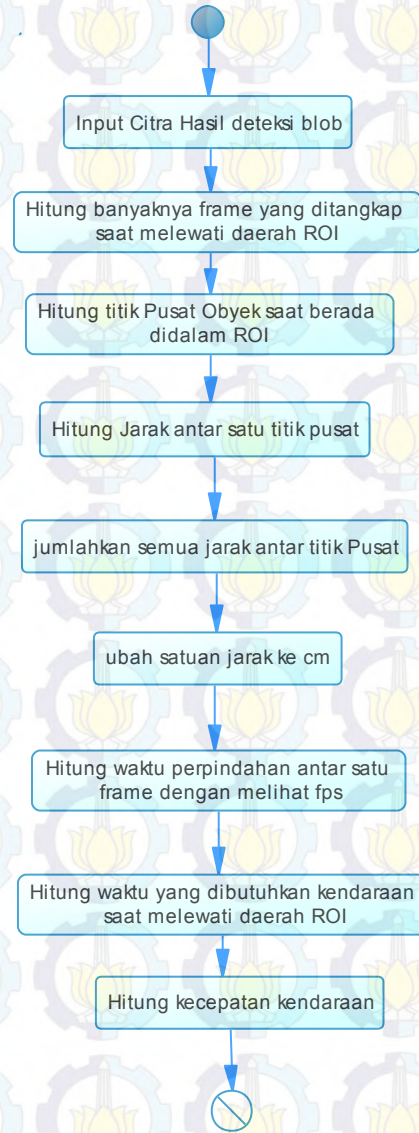


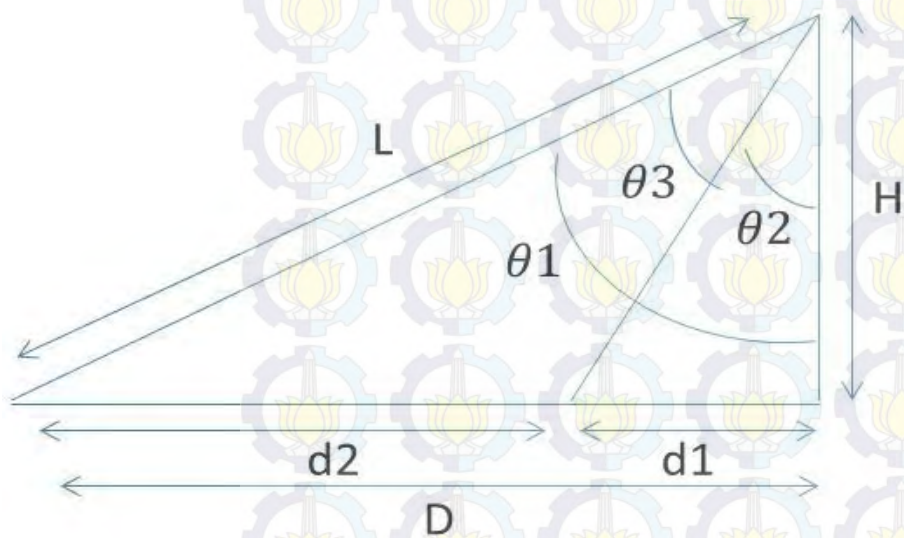
Diagram Alir Deteksi Kecepatan

No	Tahapan	Input	Output
1.	Input Awal	Video	Frame Citra
2.	Pilih Area ROI	Frame Citra	Citra ROI
3.	Hitung Parameter yang dibutuhkan	Parameter	Parameter
4.	Proses GMM	Citra ROI	Citra <i>Foreground</i>
5.	Filtering	Citra <i>Foreground</i>	Citra <i>Foreground</i> yang telah difilter
6.	Deteksi <i>Blob</i>	Citra <i>Foreground</i> yang telah difilter	Area <i>blob</i> (<i>centroid</i> , <i>weight</i> , <i>height</i>)
7.	Hitung koordinat kendaraan pada saat melewati daerah ROI	Objek hasil deteksi <i>blob</i>	Koordinat x dan y
8.	Hitung jarak area yang dilewati kendaraan	Objek hasil deteksi <i>blob</i>	Jarak
9.	Perhitungan waktu yang dibutuhkan	Objek hasil deteksi <i>blob</i>	Waktu
10.	Hitung Kecepatan Kendaraan	Jarak dan Waktu	Kecepatan kendaraan

Tabel data proses



Setting Kamera dengan posisi dari atas



dengan :

θ_1 adalah sudut pemasangan kamera;

θ_2 adalah sudut daerah yang tidak terlihat oleh kamera;

θ_3 adalah sudut pandang yang dicakup oleh kamera;

H adalah tinggi kamera dari permukaan jalan;

D adalah jarak horizontal antara kamera dan kendaraan;

L adalah jarak nyata antara kamera dan kendaraan;

d_1 adalah daerah yang tidak terlihat oleh kamera;

d_2 adalah daerah yang terlihat oleh kamera;

Perancangan Proses Deteksi Kecepatan Kendaraan

i. Mencari Titik Pusat

Proses mencari jarak perpindahan antar pixel dilakukan dengan mencari koordinat titik pusat dan menghitung jarak antar titik pusat. Secara sederhana, untuk menentukan titik pusat, langkah-langkah yang dilakukan adalah :

Menghitung koordinat x , dengan :

$$\text{Posisi } x = \text{bbox}(1) + \left(\frac{\text{bbox}(3)}{2} \right) \quad (8)$$

Menghitung koordinat y , dengan :

$$\text{Posisi } y = \text{bbox}(2) + \left(\frac{\text{bbox}(4)}{2} \right) \quad (9)$$

dimana :

$\text{bbox}(1)$ = nilai matriks *boundingbox* kolom ke 1

$\text{bbox}(2)$ = nilai matriks *boundingbox* kolom ke 2

$\text{bbox}(3)$ = nilai matriks *boundingbox* kolom ke 3

$\text{bbox}(4)$ = nilai matriks *boundingbox* kolom ke 4

ii. Pengukuran Jarak

Variabel jarak didapat dari jarak diantara dua titik pusat obyek dari *frame* satu dengan lainnya yang bisa dihitung dengan menggunakan persamaan (6)[4].

iii. Transformasi jarak ke jarak sebenarnya

Pengukuran jarak yang didapat dari tahap sebelumnya yaitu dalam satuan *pixel*, oleh karena itu jarak tersebut harus diubah ke jarak sebenarnya. Untuk menghitung hubungan antar pixel dengan jarak sebenarnya adalah :

$$z = \frac{1}{x/y} \quad (10)$$

dimana :

x = jarak jangkauan sebenarnya

y = jarak jangkauan pada citra

z = hasil hubungan antar pixel dengan jarak sebenarnya

iv. Menghitung waktu

Untuk mendapatkan variabel waktu maka akan dicari waktu yang dibutuhkan obyek dari mulai *start* ROI hingga *finish* ROI. Untuk menghitung waktu bisa dihitung dengan menggunakan rumus :

$$\Delta t = f \times w \quad (11)$$

dimana :

f = jumlah *frame* (*frame*)

w = waktu yang diperlukan satu *frame* $\left(\frac{\text{detik}}{\text{frame}} \right)$

Δt = waktu (*detik*)

No	Nama	Framerate	Pixel
1.	Movie1.avi	25	1280x720
2.	Movie2.avi	25	1280x720
3.	Movie3.avi	25	1280x720
4.	Movie4.avi	25	1280x720
5.	Movie5.avi	25	1280x720
6.	Movie6.avi	25	1280x720
7.	Movie1.avi	30	1280x720
8.	Movie2.avi	30	1280x720
9.	Movie3.avi	30	1280x720
10.	Movie4.avi	30	1280x720
11.	Movie5.avi	30	1280x720
12.	Movie6.avi	30	1280x720

Tabel Data Video yang Digunakan.

Pengujian dilakukan dengan memasukkan beberapa nilai parameter, diantaranya yaitu :

$D = 2400 \text{ cm}$

$H = 500 \text{ cm}$

$\theta_1 = 80^\circ$

$\theta_2 = 10^\circ$

Frame = sesuai banyaknya *frame* dari hasil rekaman video yaitu 25 atau 30.

Tabel Pendeteksian Kecepatan Kendaraan pada Video
Movie1.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
Motor	25	18	19,52
Motor	30	18	19,18

Tabel Prosentase Keberhasilan Video Movie1.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)	PK
Motor	25	18	19,52	91,56%
Motor	30	18	19,18	93,44%

Tabel Pendeteksian Kecepatan Kendaraan pada Video
Movie2.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
Motor	25	20	21,49
Motor	30	20	21,17

Tabel Prosentase Keberhasilan Video Movie2.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)	PK
Motor	25	20	21,49	86,89%
Motor	30	20	21,17	94,15%

Tabel Pendeteksian Kecepatan Kendaraan pada Video
Movie3.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
Motor	25	24	24,70
Motor	30	24	25,23

Tabel Prosentase Keberhasilan Video Movie3.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)	PK
Motor	25	24	24,70	97,08%
Motor	30	24	25,23	94,87%

Tabel Pendeteksian Kecepatan Kendaraan pada Video
Movie4.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
Jalan	25	7	7,798
Sepeda	25	11	10,38
Jalan	30	7	8.581
Sepeda	30	11	11,23

Tabel Prosentase Keberhasilan Video Movie4.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)	PK
Jalan	25	7	7,798	88,6%
Sepeda	25	11	10,38	94,45%
Jalan	30	7	8,581	77,41%
Sepeda	30	11	11,23	97,90%

Tabel Pendeteksian Kecepatan Kendaraan pada Video
Movie5.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
Jalan	25	7	7,561
Sepeda	25	10	9.629
Jalan	30	7	7,953
Sepeda	30	10	10,38

Tabel Prosentase Keberhasilan Video Movie5.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)	PK
Jalan	25	7	7,561	91,98%
Sepeda	25	10	9,629	96,3%
Jalan	30	7	7,953	86,38%
Sepeda	30	10	10,38	96,2%

Tabel Pendeteksian Kecepatan Kendaraan pada Movie6.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)
Jalan	25	8	8,198
Sepeda	25	11	10,29
Jalan	30	8	9,561
Sepeda	30	11	10,53

Tabel Prosentase Keberhasilan Video Movie6.avi

Kendaraan	<i>Framerate</i>	Sebenarnya (km/jam)	Program (km/jam)	PK
Jalan	25	8	8,198	97,52%
Sepeda	25	11	10,29	93,54%
Jalan	30	8	9,561	80,48%
Sepeda	30	11	10,53	95,72%

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

1. Program dapat mendeteksi kecepatan kendaraan bergerak dari video rekaman dengan membuat model *background* dan *foreground* menggunakan metode *Gaussian Mixture Model*.
2. Untuk kecepatan objek yang paling cepat, tingkat akurasi keberhasilannya lebih baik jika menggunakan parameter *framerate* 25.
3. Rata-rata akurasi keberhasilan program dalam menghitung kendaraan bergerak dengan menggunakan parameter *framerate* 25 dan 30 yaitu prosentasi tertinggi 97,9% dan prosentase terendah adalah 77,41%.
4. Perubahan cuaca dari cahaya matahari dan hembusan angin mempengaruhi performa program dalam mendeteksi kecepatan kendaraan.

- bps.go.id. (2013). “Perkembangan Jumlah Kendaraan Bermotor Menurut Jenis tahun 1987-2013”. http://www.bps.go.id/tab_sub/view.php?tabel=1&id_subyek=17¬ab=12 , posted: 2013. Diakses pada tanggal 16-02-2015
- Lin, Huei-yung. Li, Kun-Jhih. Chang, Chia-Hong. (2008). “Vehicle Speed Detection from a Single Motion Blurred Image”. **Image and Vision Computing** **26**, Hal 1327-1337
- Pornpanomchai, C., and Kongkittisan, K. (2009). “Vehicle Speed Detection System”. **IEEE International Conference on Signal and Image Processing Applications**, Hal. 135-139.
- Rad, A. G, Deghani. A, Karim. M. R. (2010). “Vehicle Speed Detection in Video Image Sequence using CVS Method”. **International Journal of the Physical Sciences Vol. 5(17)**, Hal. 2555-2563
- Rafael C. Gonzales, Richard E. Woods. (2002). “Digital Image Processing”. **United States of America: Tom Robbins Publisher.**

- Al Bovik. (2000). “Handbook of Image and Video Processing”. **San Diego: Academic Press Publisher.**
- Bharti, T. Tejinder. (2013). “Background Subtraction Techniques-Review”. **International Journal of Innovative Technology and Exploring Engineering, Vol. 2, Issue 3, Hal. 166 - 168.**
- Alper Yilmaz, Omar Javed and Mubarak Shah. “Object tracking: A survey”. **ACM Comput. Surv., 38(4):13,2006.**
- Lazuardi, R. Arif Firdaus. (2014). “Penghitungan Kendaraan Bergerak Berbasis Algoritma Background Subtraction Menggunakan Metode Gaussian Mixture Model”. **Tugas Akhir. Jurusan Matematika ITS.**
- C. Stauffer, W.E.L. Grimson. (1999). “Adaptive Background Mixture Models for Real-time Tracking”. **The Artificial Intelligence Laboratory, Cambridge.**
- Hariyanto, Zamroji. (2015). “Klasifikasi Jenis Kendaraan Bergerak Berbasis *Geometric Invariant Moment*”. **Tugas Akhir. Jurusan Matematika ITS**

TERIMAKASIH...

Wassalamu'alaikum

