



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC 184801

PREDIKSI KECEPATAN KERETA API BERBASIS TIME DIFFERENCE MENGGUNAKAN IP-CAMERA SECARA REAL-TIME

Arifandi Putrawardana
NRP 07211440000037

Dosen Pembimbing
Eko Pramunanto, ST., MT.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

TUGAS AKHIR - EC 184801

**PREDIKSI KECEPATAN KERETA API BERBASIS TIME
DIFFERENCE MENGGUNAKAN IP-CAMERA
SECARA REAL-TIME**

Arifandi Putrawardana
NRP 07211440000037

Dosen Pembimbing
Eko Pramunanto, ST., MT.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

DEPARTEMEN TEKNIK KOMPUTER
Fakultas Teknologi Elektro
Institut Teknologi Sepuluh Nopember
Surabaya 2020



ITS
Institut
Teknologi
Sepuluh Nopember

FINAL PROJECT - EC 184801

REAL-TIME TRAIN SPEED PREDICTION USING IP-CAMERA BASED ON TIME DIFFERENCE

Arifandi Putrawardana
NRP 07211440000037

Advisor
Eko Pramunanto, ST., MT.
Dr. Eko Mulyanto Yuniarno, S.T., M.T.

DEPARTMENT OF COMPUTER ENGINEERING
Faculty of Electrical Technology
Sepuluh Nopember Institute of Technology
Surabaya 2020

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir saya dengan judul “**Prediksi kecepatan Kereta Api berbasis *Time Difference* menggunakan IP-Camera secara Real-Time**” adalah benar-benar hasil karya intelektual sendiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya orang lain yang saya akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai dengan peraturan yang berlaku.

Surabaya, Juli 2020

Arifandi Putrawardana
NRP. 07211440000037

LEMBAR PENGESAHAN

Prediksi kecepatan Kereta Api berbasis *Time Difference* menggunakan IP-Camera secara *Real-Time*

Tugas Akhir ini disusun untuk memenuhi salah satu syarat memperoleh gelar Sarjana Teknik di Institut Teknologi Sepuluh Nopember Surabaya

Oleh: Arifandi Putrawardana (NRP: 07211440000037)

Tanggal Ujian: 08 Juli 2020

Periode Wisuda: September 2020

Disetujui oleh:

Eko Premunanto, ST., MT.
NIP: 196612031994121001

(Pembimbing I)

Dr. Eko Mulyanto Yuniarno, S.T., M.T.
NIP: 196806011995121009

(Pembimbing II)

Mochamad Hariadi, ST., M.Sc., Ph.D..
NIP: 19691209199703100

(Penguji I)

Dr. I Ketut Eddy Purnama, S.T., M.T
NIP. 196907301995121001

(Penguji II)

Dr.Reza Fuad Rachmadi, S.T., M.T
NIP: 19850403201212100

(Penguji III)



Mengetahui
Kepala Departemen Teknik Komputer

Dr. Supeno Mardhi Susiki Nugroho, ST., MT.
NIP. 197003131995121001

ABSTRAK

Nama Mahasiswa : Arifandi Putrawardana
Judul Tugas Akhir : Prediksi Kecepatan Kereta Api berbasis *Time Difference* menggunakan *IP-Camera* secara *Real-Time*.
Pembimbing : 1. Eko Pramunanto, ST., MT.
2. Dr. Eko Mulyanto Yuniarno, S.T., M.T.

Pada tahun 2014, PT. KAI telah menambahkan perangkat pendeteksi kecepatan lokomotif jarak jauh atau locotrack dalam operasional lokomotif yang berguna untuk meningkatkan keselamatan perjalanan. Monitor panel indikator akan berubah menjadi merah apabila laju kereta yang dipantau tersebut melebihi kecepatan yang telah ditentukan (*Over Speed*)[1]. Pemantauan tersebut hanya dapat tertuju pada satu *Over Speed* saja, sementara perlu diberikan *Over Speed* khusus pada tempat-tempat seperti pintu-pintu kereta api, persimpangan, dan pemberhentian (stasiun). Pada tugas akhir kali ini, akan dibuat sistem pengamanan untuk Monitoring Kecepatan Kereta Api pada titik tertentu (yang perlu di pantau), seperti pintu-pintu kereta api, persimpangan, dan pemberhentian (stasiun). Deteksi kehadiran dan kecepatan kereta api merupakan solusi yang efektif dan terjangkau untuk diaplikasikan pada kasus ini. Sistem ini menggunakan *IP-camera* untuk menangkap citra input dan *SBC* untuk mengolah citra agar menghasilkan *output* data arah dan kecepatan. Hasil tangkapan kamera digunakan untuk mendeteksi kereta, kemudian dihitung kecepatannya berdasarkan perbedaan frame pada citra *input* tersebut.

KATA PENGANTAR

Puji dan syukur kehadiran Allah SWT atas segala limpahan berkah dan hidayah-Nya, penulis dapat menyelesaikan penelitian ini dengan judul "**Prediksi Kecepatan Kereta Api berbasis *Time Difference* menggunakan *IP-Camera* secara *Real-Time***" untuk menghitung kecepatan kereta api dan monitoring kecepatan kereta api pada titik tertentu (yang perlu di pantau). Penelitian ini disusun dalam rangka pemenuhan bidang riset di Departemen Teknik Komputer, FTEIC-ITS serta digunakan sebagai persyaratan menyelesaikan pendidikan S1. Penelitian ini dapat terselesaikan tidak lepas dari bantuan berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Keluarga, Ibu, Bapak dan Saudara tercinta yang telah memberikan dorongan spiritual dan material dalam penyelesaian buku penelitian ini.
2. Bapak Dr. Supeno Mardi Susiki Nugroho, ST., MT. selaku Kepala Departemen Teknik Komputer, Fakultas Teknologi Elektro, Institut Teknologi Sepuluh Nopember.
3. Secara khusus penulis mengucapkan terima kasih yang sebesar-besarnya kepada Bapak Eko Premunanto, ST., MT. dan Bapak Dr. Eko Mulyanto Yuniarno, ST., MT. atas bimbingan selama pengerjaan tugas akhir ini.
4. Bapak-ibu dosen pengajar Departemen Teknik Komputer, atas pengajaran, bimbingan, serta perhatian yang diberikan kepada penulis selama ini.
5. Teman-teman Tekkom 2014, khususnya Adrian, Hana, Yayak, 2015, 2016, dan 2017. Teman-teman SMA saya yang turut serta berjuang di ITS dan selalu memberi saya semangat khususnya Sarah, Hafiz, Rahman, Ega. Serta teman-teman lain yang tidak dapat saya sebutkan satu-persatu.

Kesempurnaan hanya milik Allah SWT, untuk itu penulis memohon segenap kritik dan saran yang membangun. Semoga penelitian ini dapat memberikan manfaat bagi kita semua. Amin.

Surabaya, Juni 2020

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

Abstrak

KATA PENGANTAR

DAFTAR ISI

DAFTAR GAMBAR

DAFTAR TABEL

NOMENKLATUR

1	PENDAHULUAN	1
1.1	Latar belakang	1
1.2	Permasalahan	2
1.3	Tujuan	2
1.4	Batasan masalah	2
1.5	Sistematika Penulisan	3
1.6	Relevansi	4
2	TINJAUAN PUSTAKA	5
2.1	Dasar Teori	5
2.1.1	Computer Vision	5
2.1.2	Pengukuran Kecepatan	5
2.1.3	<i>Single Board Computer Raspberry Pi</i>	7
2.1.4	<i>Image Pre-Processing</i>	7
2.1.5	Thresholding	9
2.1.6	Deteksi Objek	10
2.1.7	Deteksi Objek dengan Gerakan	10
2.1.8	<i>Background Subtraction</i>	11
2.1.9	Feature Extraction	11
2.1.10	Python	11
2.1.11	<i>OpenCV</i>	13
2.1.12	Action Cam	13
2.2	Penelitian Terkait	14

2.2.1	Vehicle Speed detection by using Camera and image processing software	14
3	DESAIN DAN IMPLEMENTASI SISTEM	15
3.1	Desain Sistem	15
3.2	Alur Kerja Sistem	19
3.3	Akuisisi Data	19
3.3.1	Persiapan dan Peletakan Kamera	19
3.3.2	Lokasi Akuisisi Citra	20
3.3.3	Pemasangan Penanda RoI	22
3.3.4	Spesifikasi Kamera	24
3.3.5	Pengambilan Data Video	24
3.4	Deteksi Rel	26
3.5	Deteksi Kereta Api	27
3.6	Perhitungan Kecepatan Kereta Api berdasarkan Time Difference	27
4	PENGUJIAN DAN ANALISA	33
4.1	Pengujian dengan Perbedaan Jumlah <i>Frame Rate</i>	34
4.2	Pengujian dengan Perbedaan Jarak RoI	36
4.3	Pengujian dengan Perbedaan Waktu	38
4.4	Pengujian dengan Perbedaan Resolusi	40
4.5	Pengujian secara Real Time	41
4.6	Pengujian menggunakan video dari Youtube	43
5	PENUTUP	45
5.1	Kesimpulan	45
5.2	Saran	46
	DAFTAR PUSTAKA	47
	LAMPIRAN	49
	Biografi Penulis	51

DAFTAR GAMBAR

2.1	Raspberry Pi 3 Model B+	7
2.2	Contoh hasil dari <i>adaptive thresholding</i> [2]	9
2.3	Gopro Hero 4 Silver [3]	13
2.4	Ilustrasi jarak perhitungan kecepatan [15]	14
3.1	Blok Diagram Sistem	15
3.2	Diagram proses Railway detection menggunakan <i>adaptive thresholding</i>	16
3.3	Diagram proses Perhitungan Kecepatan	18
3.4	Skema peletakan Kamera	19
3.5	(a) Pengukuran sudut kemiringan camera ; (b) Peletakan camera	20
3.6	Lokasi Akusisi Data [5]	21
3.7	Jembatan Penyebrangan Ahmad Yani	22
3.8	Pemasangan RoI	22
3.9	Ilustrasi pemasangan RoI	23
3.10	RoI berjarak 10 meter	23
3.11	Pengambilan video siang hari	25
3.12	Pengambilan video malam hari	25
3.13	Adaptive Threshold	26
3.14	Hasil rel setelah menggunakan adaptive thresholding	26
3.15	Deteksi kereta api pada saat memasuki RoI	27
3.16	Perhitungan frame berhenti setelah keluar RoI	27
3.17	Hasil arah dan kecepatan kereta api	28
3.18	Hasil arah dan kecepatan kereta api	29
3.19	Tampilan GUI untuk pengujian	29
3.20	Tampilan pada saat setting crop area roi	31
3.21	Tampilan pada saat setting threshold	32
4.1	Hasil dari virtual marker yang saya buat	34
4.2	Grafik hasil galat dengan perbedaan frame rate	36
4.3	Grafik hasil galat dengan perbedaan roi (arah 1)	37
4.4	Grafik hasil galat dengan perbedaan roi (arah 2)	38
4.5	Grafik hasil galat dengan perbedaan waktu	39
4.6	Grafik hasil galat dengan perbedaan resolusi	41

4.7	Resolusi 1080p	41
4.8	Resolusi 720p	41
4.9	41
4.10	Hasil pengujian menggunakan video dari youtube . .	43

DAFTAR TABEL

3.1	Spesifikasi Kamera Aksi Gopro Hero 4	24
3.2	Jadwal kehadiran kereta yang akan dipantau	25
4.1	Tabel Pengujian <i>Kecepatan Kereta Api</i> dengan Perbedaan Jumlah <i>Frame Rate</i> (Arah Selatan)	35
4.2	Tabel Pengujian Galat <i>Kecepatan Kereta Api</i> dengan Perbedaan Jumlah <i>Frame Rate</i> (Arah Utara)	35
4.3	Tabel Pengujian <i>Kecepatan Kereta Api</i> dengan Perbedaan jarak RoI	36
4.4	Tabel Pengujian Galat dengan Perbedaan jarak RoI	37
4.5	Tabel Pengujian <i>Kecepatan Kereta Api</i> dengan Perbedaan Waktu (Arah Selatan)	38
4.6	Tabel Pengujian Galat <i>Kecepatan Kereta Api</i> dengan Perbedaan Waktu (Arah Utara)	39
4.7	Tabel Pengujian <i>Kecepatan Kereta Api</i> dengan Perbedaan Resolusi	40
4.8	Tabel Pengujian Galat <i>Kecepatan Kereta Api</i> dengan Perbedaan Resolusi	40
4.9	Tabel Pengujian <i>Kecepatan Kereta Api</i> secara real time	42
4.10	Tabel Pengujian Galat secara real time	42

Halaman ini sengaja dikosongkan

NOMENKLATUR

V	: Kecepatan
D	: Jarak Tempuh
T	: Waktu Tempuh
x_{bw}	: Jarak <i>box</i> x dari pojok kiri atas
$PanjangGaris$: Nilai jarak didapatkan berdasarkan jarak RoI (<i>Region of Interest</i>)
fps	: Jumlah frame per second pada video

Halaman ini sengaja dikosongkan

BAB 1

PENDAHULUAN

Penelitian ini di latar belakang oleh berbagai kondisi yang menjadi acuan. Selain itu juga terdapat beberapa permasalahan yang akan dijawab sebagai luaran dari penelitian.

1.1 Latar belakang

Pada tahun 2014, PT. KAI telah menambahkan perangkat pendeteksi kecepatan lokomotif jarak jauh atau locotrack dalam operasional lokomotif yang berguna untuk meningkatkan keselamatan perjalanan. Alat tersebut bekerja menggunakan *Global Positioning System* (GPS) melalui jaringan seluler yang dapat memberikan laporan kecepatan di sepanjang lintasan secara langsung kepada Pusat Pengendali Operasi Kereta Api Pusat. Monitor panel indikator akan berubah menjadi merah apabila laju kereta yang dipantau tersebut melebihi kecepatan yang telah ditentukan (*Over Speed*)[1]. Pemantauan tersebut hanya dapat tertuju pada satu *Over Speed* saja, sementara perlu diberikan *Over Speed* khusus pada tempat-tempat seperti pintu-pintu kereta api, persimpangan, dan pemberhentian (stasiun).

Teknologi *image processing* merupakan salah satu bidang riset yang berkembang pesat beberapa tahun belakangan ini. Kemunculan metode-metode pengenalan pola, bentuk benda, dan pengukuran berbasis pengolahan citra telah dimanfaatkan pada beragam aplikasi kehidupan. Salah satu contoh implementasi pengukuran adalah perhitungan kecepatan berdasarkan video atau rekaman langsung dari kamera. Untuk dapat menghitung kecepatan objek bergerak yang melewati suatu jalur maka sistem harus mampu melakukan deteksi terhadap objek yang terekam kamera[6].

Berdasarkan referensi paper oleh N. H. Tsani dari Universitas Telkom, metode frame difference merupakan bentuk pengurangan latar belakang yang sederhana. Proses metode ini mengurangi frame yang sekarang dengan frame sebelumnya sepanjang jumlah urutan frame video. Jika nilai mutlak piksel dari hasil pengurangan ini lebih besar dari threshold yang ditentukan maka akan dipertimbangkan

sebagai latar depan atau objek bergerak[6]. Pada tugas akhir kali ini latar depan atau objek bergerak tersebut adalah kereta api, sedangkan latar belakangnya adalah rel kereta api.

Oleh karena itu, saya mencoba melakukan penelitian dengan membuat perangkat yang dapat mendeteksi arah dan kecepatan kereta api di titik tertentu berbasis pengolahan citra digital. Perangkat ini akan membantu pemantauan serta dapat mendukung sistem keamanan kereta api menggunakan teknologi pada bidang *image processing* untuk mendeteksi arah dan menghitung kecepatan. Sistem ini akan mengambil gambar pada saat Kereta Api melewati daerah tertentu untuk menjadi masukan dalam proses *speed detection* yang kemudian hasilnya akan berupa *integer*.

1.2 Permasalahan

Berdasarkan data yang telah dipaparkan di latar belakang, sistem pengamanan yang sudah ada tidak dapat memantau lokasi-lokasi tertentu seperti pada pintu-pintu kereta api, persimpangan, terowongan dan pemberhentian (stasiun). Sistem tersebut juga memiliki kekurangan seperti pengaruh human error, ketahanan sensor (hardware), terdapat delay pada gps sehingga pembacaan kecepatan yang salah, dan kehilangan sinyal GPS seperti pada saat kereta api melewati terowongan yang membuat kereta api tidak dapat terpantau, sehingga diperlukan sebuah perangkat untuk mendeteksi dan menghitung kecepatan Kereta Api pada suatu daerah tertentu.

1.3 Tujuan

Adapun tujuan dari penelitian Tugas Akhir ini adalah sebagai berikut.

1. Melakukan penelitian alat prediksi kecepatan kereta api di titik tertentu berbasis time difference. Melakukan analisis performa terhadap sistem deteksi kehadiran dan kecepatan kereta api di titik tertentu sehingga dapat digunakan sebagai pengembangan sistem keamanan.

1.4 Batasan masalah

Batasan masalah pada penelitian Tugas Akhir ini adalah:

1. Daerah atau tempat untuk mendeteksi Kereta Api yaitu di

- Jembatan Penyebrangan Jalan Ahmad Yani.
2. Waktu pengujian sistem dilakukan pada waktu siang, sore, dan malam hari.
 3. Pengujian hanya mendeteksi arah dan kecepatan secara sistem (estimasi).
 4. Untuk pengujian pemrosesan citra secara real time menggunakan *single board computer* (SBC) dan kamera action cam.

1.5 Sistematika Penulisan

Laporan penelitian Tugas akhir ini tersusun dalam sistematika dan terstruktur sehingga mudah dipahami dan dipelajari oleh pembaca maupun seseorang yang ingin melanjutkan penelitian ini. Alur sistematika penulisan laporan penelitian ini yaitu:

1. BAB I Pendahuluan Bab ini berisi uraian tentang latar belakang permasalahan, penegasan dan alasan pemilihan judul, sistematika laporan, tujuan dan metodologi penelitian.
2. BAB II Tinjauan Pustaka Bab ini berisi tentang uraian teori yang berkaitan maupun yang digunakan pada penelitian ini secara sistematis. Teori-teori yang disebutkan digunakan sebagai dasar dalam penelitian ini antara lain informasi singkat mengenai Visi Komputer, Pengukuran Kecepatan, *SBC*, *Adaptive Thresholding*, serta teori penunjang lainnya.
3. BAB III Perancangan Sistem dan Implementasi Bab ini berisi tentang penjelasan-penjelasan terkait eksperimen yang akan dilakukan dan langkah-langkah data diolah hingga menghasilkan output kecepatan. Guna mendukung eksperimen pada penelitian ini, digunakanlah blok diagram atau *work flow* agar sistem yang akan dibuat dapat terlihat dan mudah dibaca untuk implementasi pada pelaksanaan tugas akhir.
4. BAB IV Pengujian dan Analisa Bab ini menjelaskan tentang pengujian eksperimen yang dilakukan terhadap data dan analisisnya. Beberapa teknik visualisasi akan ditunjukkan hasilnya pada bab ini dan dilakukan analisa terhadap hasil visualisasi dan informasi yang didapat dari hasil mengamati visualisasi yang tersaji
5. BAB V Penutup

Bab ini merupakan penutup yang berisi kesimpulan yang diambil dari penelitian dan pengujian yang telah dilakukan. Saran dan kritik yang membangun untuk mengembangkan lebih lanjut juga dituliskan pada bab ini.

1.6 Relevansi

Penelitian mengenai visi komputer merupakan bidang riset yang sangat dibutuhkan dan dipakai dalam pemenuhan kebutuhan komputasi yang semakin beragam saat ini. Dalam tugas akhir ini, dilakukan penelitian mengenai pengembangan sistem pengukuran kecepatan kereta api berbasis pengolahan citra.

BAB 2

TINJAUAN PUSTAKA

Demi mendukung penelitian ini, dibutuhkan beberapa teori penunjang sebagai bahan acuan dan referensi. Dengan demikian penelitian ini menjadi lebih terarah.

2.1 Dasar Teori

2.1.1 Computer Vision

Computer vision merupakan suatu ilmu di bidang komputer yang dapat membuat mesin atau robot untuk ‘melihat’. Kata ‘melihat’ disini berarti mesin atau robot dapat bekerja seperti memiliki panca indera layaknya manusia. Terdapat beberapa klasifikasi dari vision itu sendiri, yaitu *Low Level Vision*, *Medium Level Vision*, dan *High Level Vision*[7]. *Low Level Vision* meliputi *Sensing*, yaitu pengambilan input berupa gambar, dan *Preprocessing*, yaitu memperoleh suatu gambar sebelum di proses. *Medium Level Vision* meliputi proses *Segmentation*, *Description*, *Recognition*. *Segmentation* adalah proses pemisahan gambar digital ke dalam beberapa region. *Description* merupakan proses mendeskripsikan suatu gambar, sedangkan *Recognition* merupakan pengenalan terhadap suatu gambar. Pada level yang lebih tinggi (*High Level Vision*) terdapat proses *Interpretation*, dimana *Interpretation* merupakan suatu kemampuan untuk memperkirakan bentuk asli dari gambar yang didapat, hal ini dapat dilakukan cara mendapatkan berbagai informasi yang diperlukan pada gambar tersebut. Maka proses *Interpretation* memerlukan deteksi, identifikasi, dan pengukuran dari fitur-fitur pada suatu gambar.

2.1.2 Pengukuran Kecepatan

Perhitungan kecepatan pada objek yang bergerak dapat dihitung dengan mencari total jarak yang ditempuh dari awal objek bergerak sampai akhir objek bergerak dibagi dengan waktu yang ditempuh dalam melakukan perpindahan menggunakan persamaan

berikut.

$$V = \frac{D}{T} \quad (2.1)$$

Pada persamaan 2.1 tersebut, dimana:

V merupakan *velocity* atau kecepatan yang dicapai dalam melakukan perpindahan dengan Satuan Internasional (SI) adalah meter/sekon (m/s).

D merupakan *distance* atau jarak yang ditempuh dengan Satuan Internasional (SI) adalah meter (m)

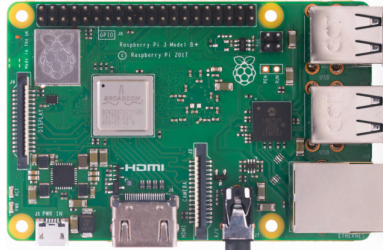
T merupakan *time* atau waktu yang ditempuh dalam melakukan perpindahan dengan Satuan Internasional (SI) adalah sekon (s).

Proses perhitungan untuk mendapatkan kecepatan laju kendaraan dari hasil tangkapan video dengan objek yang bergerak, dapat dihitung dengan menggunakan persamaan berikut [4]:

$$Kecepatan = \frac{PanjangGaris \times fps \times 3600}{JumlahFrame \times 1000} km/h \quad (2.2)$$

Pada persamaan 2.2 tersebut, dimana:

1. Pajang garis merupakan nilai jarak didapatkan berdasarkan jarak RoI (*Region of Interest*) yang telah ditentukan.
2. *fps* merupakan *frame per second* yaitu kemampuan kamera dalam mengambil citra tiap detiknya.
3. 3600 merupakan konversi dari sekon ke jam.
4. Jumlah *frame* merupakan banyak *frame* yang ditempuh oleh objek dalam batas garis yang sudah ditentukan.
5. 1000 merupakan konversi dari kilometer ke meter.



Gambar 2.1: Raspberry Pi 3 Model B+ [?]

2.1.3 Single Board Computer Raspberry Pi

Raspberry Pi merupakan single board computer yang dapat digunakan untuk menjalankan perangkat lunak maupun proses pengolahan data. Selain itu *Raspberry Pi* dapat melakukan proses visualisasi data seperti halnya komputer personal. *Raspberry Pi Model B* berbasis sistem operasi *Raspbian*, yaitu sistem operasi yang dikembangkan oleh perusahaan *Raspberry Pi* yang merupakan turunan dari sistem operasi *Debian*. Pada penelitian ini akan digunakan *Raspberry Pi model 3B+* memiliki spesifikasi SoC Cortex-A53 (ARMv8) 64-bit @ 1.4GHz, 1 GB LPDDR2 SDRAM dengan Catu Daya 5V/2.5A DC.

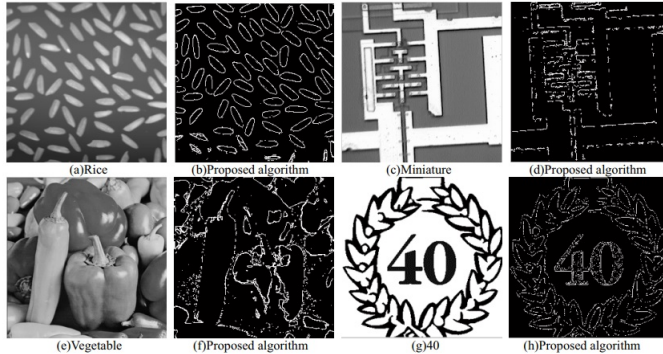
2.1.4 Image Pre-Processing

Menurut (Krig, 2014) image pre-processing metode yang digunakan untuk mempersiapkan gambar untuk analisa lanjut, termasuk deteksi interest point dan feature extraction. Image pre-processing dapat membuat informasi asli pada gambar terdistorsi. Akan tetapi penggunaan yang tepat terhadap image pre-processing sangat membantu untuk mendapatkan hasil yang lebih global pada feature detection. Jika dianalogikan maka, image pre-processing adalah seperti sound system dengan kontrol yang bervariasi, seperti suara tanpa kontrol volume; kontrol volume beserta tune knob; kontrol volume ditambahkan dengan treble, bass dan mid; atau kontrol volume dengan graphics equalizer dan pengeras suara yang berkualitas bagus. Begitu juga dengan image pre-processing. Gambar yang diberi koreksi dan perbaikan yang tepat merupakan bagian penting

dari computer vision. Ada banyak metode untuk melakukan image pre-processing dan beberapa dari metode tersebut berkaitan. Akan tetapi tidak semua metode ini dapat memberikan hasil yang terbaik dalam semua kondisi. Tetap perlu adanya pemilihan metode yang tepat untuk kondisi dan tujuan tertentu [8]. Berikut beberapa metode image pre-processing:

1. Illumination corrections: sangat dibutuhkan untuk memproses gambar biner, karena ketika cahaya terlalu terang maka outline benda juga semakin tidak terlihat.
2. Blur and focus correction: koreksi untuk membuat gambar semakin kabur atau semakin tajam. Biasanya koreksi ini dibutuhkan ketika kita ingin mendapat gradien dari sebuah garis di gambar.
3. Filtering and noise removal: koreksi untuk mengurangi element yang dapat membiaskan komputasi pada gambar.
4. Thresholding: metode ini mengubah gambar ke warna hitam dan putih saja. Tujuan digunakannya thresholding agar data pada gambar lebih sederhana. Gambar dengan warna hitam atau putih saja akan lebih cepat diproses dibandingkan gambar berwarna-warni. *frame* yang ditempuh oleh objek dalam batas garis yang sudah ditentukan.
5. Morphology: metode ini dapat mengubah bentuk sebuah gambar.
6. Segmentation: metode untuk membagi-bagi gambar menjadi poligon-poligon yang terhubung.

2.1.5 Thresholding



Gambar 2.2: Contoh hasil dari *adaptive thresholding* [2]

Threshold merupakan teknik yang sederhana dan efektif untuk segmentasi citra dengan melakukan pengolahan piksel pada suatu citra atau menghilangkan beberapa piksel dan juga mempertahankan beberapa piksel sehingga menghasilkan suatu citra baru hasil sortir piksel yang telah dilakukan. Dengan dilakukannya komputasi threshold maka dapat dengan mudah mendapatkan edge atau tepi dari suatu citra.[4]

Threshold dilakukan agar mempermudah dalam proses identifikasi ataupun perbandingan dari dua atau lebih citra. Terdapat 5 tipe threshold, yaitu binary threshold, binary inverted, truncate, threshold to zero and threshold to zero inverted. Binary threshold merupakan citra yang telah melalui proses pemisahan piksel-piksel berdasarkan derajat keabuan yang dimiliki. Pembentukan citra biner memerlukan nilai batas keabuan yang akan digunakan sebagai nilai patokan. Piksel dengan derajat keabuan lebih besar dari nilai batas akan diberi nilai 1 sebagai warna putih dan sebaliknya piksel dengan derajat keabuan lebih kecil dari nilai batas akan diberi nilai 0 sebagai warna hitam. Warna putih biasanya digunakan untuk warna foreground sedangkan warna hitam adalah warna backgro-

und. Maka dapat dilihat dalam persamaan berikut.

$$x_{bw} = \begin{cases} 1 & \text{jika } x \geq \bar{x} \\ 0 & \text{jika } x < \bar{x} \end{cases} \quad (2.3)$$

Berdasarkan referensi paper oleh Payel Roy, teknik thresholding konvensional menggunakan threshold global untuk semua piksel, dimana *adaptive thresholding* mengubah nilai threshold secara dinamis pada gambar. Penulis paper ini membandingkan berbagai macam metode adaptive thresholding yang telah ada, sehingga mendapatkan kesimpulan metode Entropy merupakan yang terbaik dari semua pembandingnya yaitu metode Otsu, Rosin, dan Kapur. Pada penelitian kali ini, adaptive thresholding digunakan untuk mendeteksi rel kereta api. Metode ini sangat efektif dan dapat digunakan dalam semua kondisi waktu (pagi, siang, sore, malam). Adaptive thresholding tersedia pada library opencv sehingga dapat langsung digunakan[9].

2.1.6 Deteksi Objek

Menurut (Szeliski, 2010), deteksi objek adalah detektor yang dibuat secara khusus yang digunakan untuk secara cepat mendeteksi letak ciri pada gambar yang menunjukkan sebuah benda dapat berada. Sebagai contohnya adalah algoritma yang dibuat untuk kamera digital agar dapat memiliki fungsi auto-focus. Deteksi objek dibagi menjadi beberapa jenis. Berikut beberapa deteksi objek yang sudah dikembangkan [10].

2.1.7 Deteksi Objek dengan Gerakan

Menurut (Martín & Pobil, 2012), deteksi gerakan adalah inti dari beberapa aplikasi visual dengan menyediakan stimulus untuk mendeteksi objek perhatian yang berada pada lingkup pandang sebuah sensor. Deteksi gerakan biasanya dilakukan dengan menggunakan background subtraction, terutama untuk aplikasi pada kamera statis. Ide utama dari metode ini adalah membuat background model dari beberapa deretan gambar untuk mencari perbedaan latar sebelumnya dengan latar gambar terbaru. Oleh karena itu, akurasi dari segmentasi gambar pada background subtraction ini sangat

bergantung pada model latarnya. Pada kasus nyata, model latar harus disesuaikan terhadap perubahan yang secara natural terjadi, misalnya daun yang bergerak maupun perubahan intensitas cahaya. Untuk itu, model latar harus selalu disesuaikan agar hasil segmentasi objeknya baik [11].

2.1.8 *Background Subtraction*

Background Subtraction adalah sebuah metode untuk mendapatkan foreground image dengan cara mengurangi gambar asli dengan background image. Menurut (Bouwmans, Porikli, Höferlin, & Vacavant, 2015), background subtraction dilakukan berdasarkan perbedaan background image dengan gambar yang sekarang akibat benda yang bergerak. Pixels yang tidak bergerak dapat dianggap sebagai background sedangkan pixels yang bergerak dapat mewakili benda yang bergerak. Dalam kepustakaan sains biasanya benda yang bergerak disebut sebagai foreground [12].

2.1.9 Feature Extraction

Feature Extraction adalah proses pengambilan ciri-ciri yang unik dari data yang akan diolah, yang bertujuan untuk memperkecil jumlah data mengambil informasi yang terpenting dari data yang diolah dan mempertinggi presisi pengolahan data. Dalam metode bounding box gambar di visualisasikan melalui dimensi objek yang dihasilkan dari gambar sebelumnya yaitu dengan menggunakan hasil output frame difference. Namun, terlebih dahulu melakukan proses pengambilan ekstraksi ciri benda yang akan diolah untuk menentukan setiap koordinat dari suatu objek menggunakan feature extraction agar proses bounding box dapat dilakukan. Scanning dilakukan untuk mencari batas-batas nilai edge berupa nilai non zero di setiap frame yang dimulai dari baris atas sampai baris bawah setiap frame, kolom kiri sampai dengan kolom kanan setiap frame.[4]

Dalam hal ini manfaat feature extraction digunakan untuk mendeteksi objek yang akan diidentifikasi agar lebih jelas, dimana dengan cara menggambarkan nilai intensitas warna pixel tersebut.

2.1.10 Python

Tugas akhir ini dikembangkan dalam bahasa pemrograman Python. Menurut (Romano, 2015) bahasa pemrograman Python di-

buat oleh Guido Van Rossum, seorang ahli ilmu teknologi komputer dan ahli matematika Belanda. Bahasa pemrograman Python secara resmi diterbitkan pada tahun 1991, semenjak itu bahasa pemrograman Python telah berubah menjadi bahasa pemrograman yang digunakan secara global [13]. Beberapa keunggulan dari bahasa pemrograman Python adalah sebagai berikut:

1. **Portabel.** Bahasa pemrograman Python dapat melakukan *porting* dari Linux ke Windows atau Mac hanya dengan menyesuaikan *path*. Bahasa pemrograman Python juga tidak tergantung pada sistem operasi sebuah komputer, hal ini memungkinkan *programmer* untuk membuat program yang tidak harus spesifik dengan sistem operasinya.
2. **Koherensi.** Bahasa python memiliki struktur yang logis dan koheren sehingga membantu programmer untuk cepat mengerti bagaimana sebuah *method* dipanggil. Dengan adanya keunggulan seperti ini, programmer tidak perlu menghabiskan banyak waktu untuk membaca buku panduan.
3. **Produktivitas *developer*.** Menurut Mark Lutz (Lutz, 2013), ukuran sebuah program Python adalah seperlima sampai dengan sepertiga program Java atau C++. hal ini berarti proses pekerjaan dapat berjalan lebih cepat. Program yang pendek juga membuat *programmer* lebih cepat dalam mengurus program Python yang sudah ada. Aspek penting lainnya adalah program Python dapat dijalankan juga secara cepat.
4. ***Library* yang luas.** Bahasa pemrograman Python didukung dengan standard library yang luas. Komunitas Python dari seluruh dunia juga menyediakan third party library yang dapat diakses dengan mudah melalui Python Package Index (PyPI).
5. **Kualitas *software*.** Bahasa pemrograman Python memiliki sifat intrinsic multi-paradigm yang berarti bahasa pemrograman Python dapat digunakan sebagai bahasa scripting, bahasa pemrograman berbasis objek, bahasa pemrograman imperatif maupun fungsional.

6. **Integrasi *software*.** Bahasa pemrogram Python dapat di ekstensikan untuk digunakan dengan bahasa pemrograman lain, sehingga sistem yang sudah menggunakan bahasa pemrograman lain sebagai bahasa pemrograman utama tidak perlu melakukan migrasi ke bahasa pemrograman Python.

2.1.11 *OpenCV*

OpenCV (*Open Source Computer Vision*) adalah sebuah *library* yang menyediakan infrastruktur umum untuk aplikasi Computer Vision. OpenCV berada di bawah lisensi BSD (*Berkeley Software Distribution*) sehingga memungkinkan pengguna untuk memodifikasi code dan resmi dipakai secara komersial. OpenCV dibuat menggunakan bahasa C++, namun OpenCV juga menyediakan interface untuk bahasa pemrograman lain seperti Python dan Java. OpenCV juga dapat dipakai oleh beberapa OS seperti Windows, Linux, Android dan MacOS. Penelitian ini menggunakan *library* OpenCV karena terdapat *function* untuk mengatur operasi kamera, *preprocessing* gambar, *processing* gambar dan juga operasi matematika pada *array*. Selain itu OpenCV juga memiliki dokumentasi yang cukup lengkap sehingga sangat membantu dalam proses pembuatan program untuk sistem deteksi gerakan objek.

2.1.12 Action Cam



Gambar 2.3: Gopro Hero 4 Silver [3]

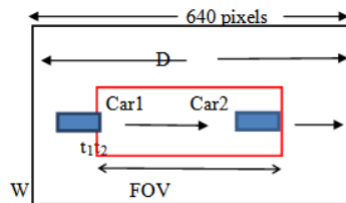
Kamera aksi atau action-cam adalah kamera digital yang di-

rancang untuk merekam aksi atau kegiatan di luar ruangan, olahraga, dan kegiatan yang bersifat ekstrem lainnya[14]. Kamera ini mampu merekam foto atau video kegiatan ekstrem yang tidak bisa dilakukan oleh kamera biasa. Karenanya, kamera tindakan biasanya kompak dan kasar, dan tahan air pada tingkat permukaan. Pada penelitian kali ini action-cam digunakan untuk mengambil video kereta api karena kemampuan kamera yang dapat mengambil video dengan frame rate hingga 60 fps dengan resolusi 1080 dan ukuran yang compact sehingga memudahkan pada saat pengambilan data.

2.2 Penelitian Terkait

2.2.1 Vehicle Speed detection by using Camera and image processing software

Penelitian terkait dilakukan oleh Hakan Koyuncu dan Baki Koyuncu untuk menghitung kecepatan kendaraan dengan metode image processing sederhana yaitu Linear Motion Detection, peneliti melakukan perhitungan terhadap frame terdeteksi bergerak. Jarak asli didapatkan dengan rumus $(PCF) = D / (640)$ meter/pixel, karena jarak untuk perhitungan kecepatan sebesar 640 pixel. Camera di setting dengan mengambil gambar dari samping jalan sehingga terlihat kendaraan bergerak secara horizontal terhadap frame. Kecepatan didapatkan dengan cara menghitung rata-rata dari semua jumlah kecepatan yang telah dihitung sehingga akan diperoleh nilai tengah. [15].



Gambar 2.4: Ilustrasi jarak perhitungan kecepatan [15]

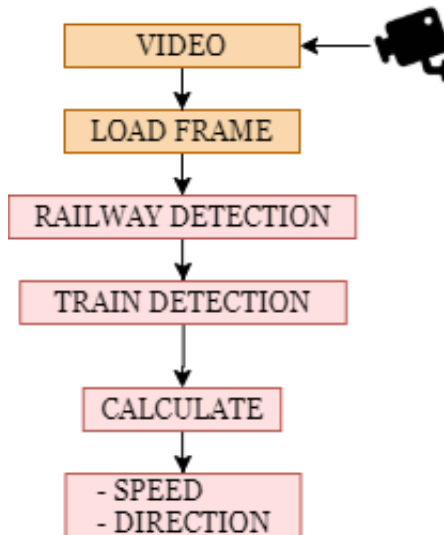
BAB 3

DESAIN DAN IMPLEMENTASI SISTEM

Penelitian ini dilaksanakan sesuai dengan desain sistem berikut dengan implementasinya. Desain sistem merupakan konsep dari pembuatan dan perancangan infrastruktur kemudian diwujudkan dalam bentuk blok-blok alur yang harus dikerjakan. Pada bagian implementasi merupakan pelaksanaan teknis untuk setiap blok pada desain sistem.

3.1 Desain Sistem

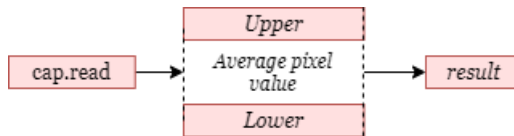
Tugas akhir ini merupakan salah satu bentuk dari pemanfaatan image processing untuk mengukur kecepatan Kereta Api yang bertujuan untuk mendapatkan data kecepatan guna memantau kecepatan kereta pada titik tertentu. Proses kerja dari sistem ditunjukkan pada 3.1.



Gambar 3.1: Blok Diagram Sistem

Data yang dibutuhkan untuk menghitung kecepatan adalah berupa video, pertama-tama video akan diambil menggunakan ip camera kemudian data video yang telah diambil diolah dan dibaca frame demi frame oleh program.

Frame dibaca setelah melakukan perintah `cap.read` pada program python. Frame akan di-threshold menggunakan metode *adaptive thresholding* dimana semua nilai pixel akan dirata-rata kemudian program akan menentukan lower dan upper berdasarkan rata-rata tersebut sehingga gambar akan berubah menjadi 8 bit. Sehingga akan keluar hasil dari threshold untuk mendeteksi rel kereta api (result) seperti pada diagram 3.2.



Gambar 3.2: Diagram proses Railway detection menggunakan *adaptive thresholding*

Setelah program mendeteksi rel, akan dibuat garis posisi awal dan akhir sesuai dengan letak RoI yang telah ditentukan. Kedua garis ini berfungsi untuk menentukan kapan program dapat memulai perhitungan, dalam hal ini menggunakan crop area, yaitu ketika kereta telah melewati garis pertama, program akan mendeteksi adanya objek bergerak kemudian memulai menentukan arah dan perhitungan kecepatan sampai kereta melewati garis selanjutnya. Proses perhitungan lebih jelas terdapat pada diagram 3.3 berikut.

Proses mendeteksi objek bergerak dalam urutan frame dikenal sebagai pelacakan. Pelacakan ini dapat dilakukan dengan menggunakan ekstraksi ciri benda dan mendeteksi objek atau benda bergerak diurutan frame. Dengan menggunakan nilai posisi objek disetiap frame kita dapat menghitung posisi dan kecepatan objek bergerak. Jarak yang ditempuh oleh objek ditentukan dengan menggunakan perbedaan nilai pixel. Jarak yang dihitung dengan menggunakan rumus jarak euclidean. Rumus jarak euclidean pada dua dimensi merupakan perpindahan sebuah simbol antara satu titik ke titik lainnya pada sumbu X dan sumbu Y [4].

$$D = \sqrt{(X_2 - X_1)^2 + (Y_2 - Y_1)^2} \quad (3.1)$$

Dimana:

D = jarak yang ditempuh

X1 = posisi titik 1 pada sumbu X

X2 = posisi titik 2 pada sumbu X

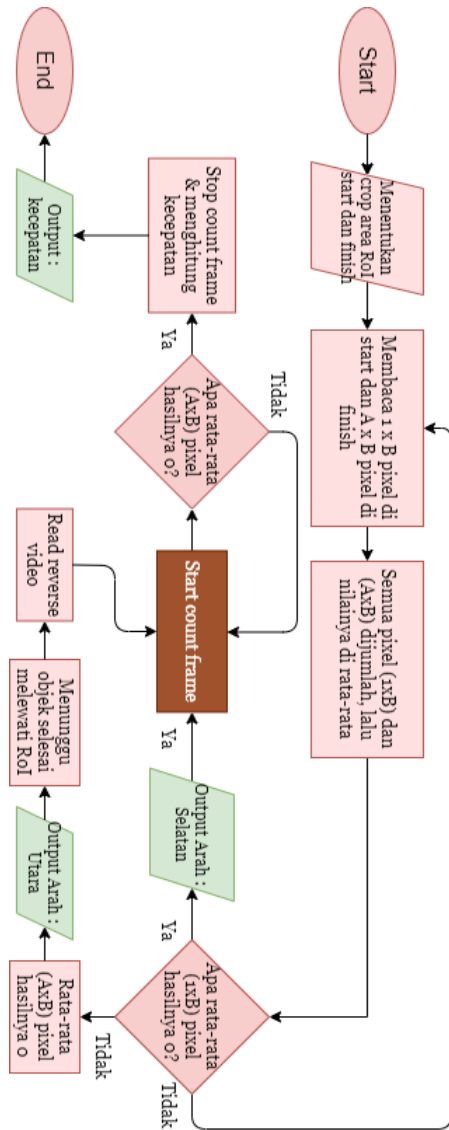
Y1 = posisi titik 1 pada sumbu Y

Y2 = posisi titik 2 pada sumbu Y

Nilai D digunakan untuk mencari jumlah frame yang terdapat pada persamaan 1.

Area crop yang ditentukan untuk mendeteksi objek bergerak berada pada area 1xB sampai AxB

Dalam tugas akhir ini output arah yang dihasilkan adalah sebanyak dua arah, yaitu arah Selatan dan arah Utara. Dengan posisi kamera tertentu, program akan mendeteksi arah dengan sesuai.



Gambar 3.3: Diagram proses Perhitungan Kecepatan

3.2 Alur Kerja Sistem

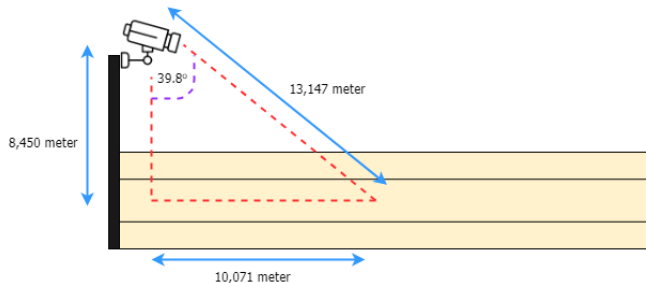
Pengerjaan tugas akhir ini dibagi menjadi beberapa tahapan berdasarkan metodologi penelitian, yaitu :

1. Akusisi Data.
2. Deteksi Rel.
3. Deteksi kehadiran Kereta Api.
4. Perhitungan Kecepatan Kereta Api berdasarkan Time Difference.

3.3 Akusisi Data

Pada tahap akusisi data, terdapat data video yang di proses dalam penelitian ini yaitu data video yang ditangkap dari posisi atas lalu berada di tengah objek.

3.3.1 Persiapan dan Peletakan Kamera



Gambar 3.4: Skema peletakan Kamera

Akusisi data video pada penelitian ini menggunakan camera action cam yang dipasang di atas dan berada persis di tengah rel kereta. Berdasarkan pada gambar 3.4, camera dipasang pada ketinggian 8,45 meter dari jalur rel hingga tepat pada posisi kamera diletakkan yaitu di JPO (Jembatan Penyebrangan Orang) dan camera membentuk sudut kemiringan sebesar 39,8 derajat. Tinggi jembatan dan sudut kamera diukur menggunakan alat ukur laser yang memiliki tingkat error hingga 2mm saja sehingga pengukuran menjadi lebih akurat.



(a)

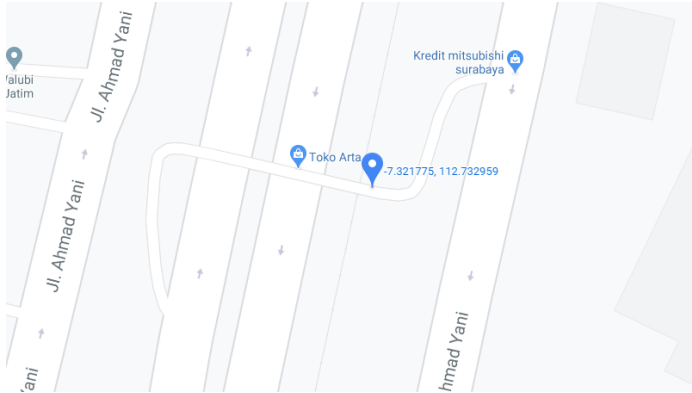


(b)

Gambar 3.5: (a) Pengukuran sudut kemiringan camera ; (b) Peletakan camera

3.3.2 Lokasi Akusisi Citra

Lokasi akuisisi data berada di rel kereta api sebelah Jalan Ahmad Yani, Kota Surabaya. Berada di sebelah jalan depan kampus Universitas Islam Negeri Sunan Ampel Surabaya, dapat dilihat pada gambar 3.6 dengan latitude di koordinat -7.321775 dan longtitude di koordinat 112.732959 pada Google Maps. Dengan pertimbangan berikut yang dijelaskan dibawah ini, menurut penulis merupakan lokasi yang strategis untuk akuisisi data.



Gambar 3.6: Lokasi Akusisi Data [5]

Rel kereta api dapat dilihat pada gambar 3.7 merupakan jalur utama kereta api dari Stasiun Gubeng Surabaya menuju kota di Jawa timur yang berada di timur dan selatan seperti kota Pasuruan, Jember, Malang, dan yang lainnya. Rel kereta api yang berada di sebelah Jalan Ahmad Yani dan Frontage Ahmad Yani ini memberikan efek pada lalu lintas kendaraan lain yaitu mobil, motor, truk dan bus pada saat kereta api lewat melalui pintu kereta api dimana terdapat persimpangan jalan raya.

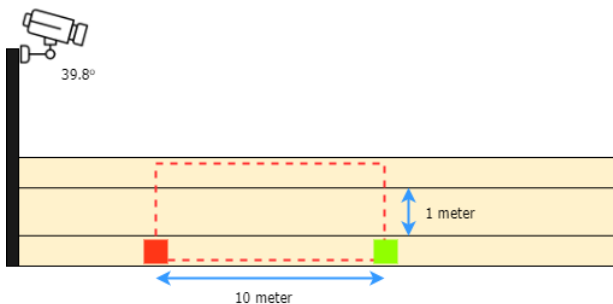
Pada lokasi ini terdapat JPO (Jembatan Penyebrangan Orang) yang memudahkan pengambilan gambar karena memanfaatkan ketinggianannya agar hasil tangkapan video dapat menjangkau untuk merekam daerah RoI dengan jarak yang sudah ditentukan, sepanjang 10 meter dengan lebar rel sepanjang 1,069 meter dan memiliki satu jalur rel sehingga mendukung untuk pengambilan data video kereta yang melewati jalur ini. Pengambilan data video dilakukan pada waktu siang hari dengan kondisi cuaca cerah dalam hal ini yang dimaksudkan adalah tidak sedang turun hujan. Pengambilan data video pada siang hari dilakukan pada pukul 13.15 dan 13.45 WIB tepat ketika kereta api dari stasiun gubeng sedang menuju ke arah selatan dan sebaliknya.



Gambar 3.7: Jembatan Penyebrangan Ahmad Yani

3.3.3 Pemasangan Penanda RoI

RoI (Region of Interest) digunakan sebagai acuan titik awal dan titik akhir dalam proses perhitungan kecepatan kendaraan. RoI diletakkan sepasang di sebelah rel dengan jarak 10 meter seperti pada gambar 3.11. Jarak RoI tersebut digunakan sebagai variabel pembanding dalam pengujian sistem ini.

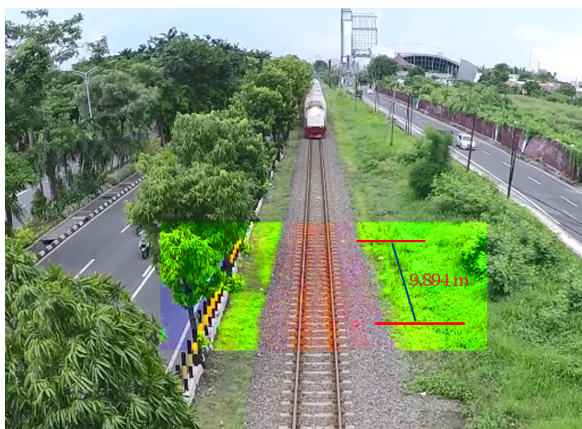


Gambar 3.8: Pemasangan RoI

Penanda RoI merupakan kertas origami berwarna berjumlah dua buah berwarna hijau dan merah yang diletakkan di sebelah rel. Pemasangan penanda RoI dimaksudkan untuk mendapatkan waktu tempuh dari kendaraan yang melewati jalan tersebut. Waktu tem-



Gambar 3.9: Ilustrasi pemasangan RoI



Gambar 3.10: RoI berjarak 10 meter

puh dalam objek yang bergerak di dalam sebuah tangkapan video didapatkan dari jumlah frame yang melewatinya. Jumlah frame dapat dihitung dengan melakukan selisih pada frame yang masuk di titik awal dan frame yang keluar di titik akhir. Pada gambar 3.12, penanda RoI yang telah terpasang di sebelah rel (dapat dilihat dari

area yang berwarna hijau) dan diletakkan sesuai dengan jarak yang ditentukan. Jarak ini juga digunakan dalam proses perhitungan kecepatan. Adapun untuk menghitung kecepatan dapat dilihat pada persamaan 2.1.

3.3.4 Spesifikasi Kamera

Berdasarkan gambar 2.4, kamera yang digunakan adalah Gopro Hero 4 Silver, Dalam penelitian ini, konfigurasi jumlah *frame rate* yang digunakan yaitu 60 fps (*frame per second*) dengan resolusi Full HD (1920 x 1080 piksel).

Tabel 3.1: Spesifikasi Kamera Aksi Gopro Hero 4

SPESIFIKASI GOPRO HERO 4 SILVER	
System Information	
Storage	micro-SDHC, micro-SDXC
Feature	
Image sensor	1/2.3" (5.6x) CMOS
Maximum Resolution	3840 x 2160 (12MP)
Lens type	Fixed-focal
Focal length	3.0mm f/2.8 (16mm equiv.)
Shutter time	1/2 sec to 1/8000 sec
General connectors	mini-USB for USB connection
Video compression	H.264 and MPEG
Maximum frame rate	60 fps @ 1920x1080

3.3.5 Pengambilan Data Video

Pengambilan data video dilakukan pada waktu pagi, siang, sore, dan malam hari dengan beberapa parameter sesuai kebutuhan untuk pengujian, yaitu arah yang berbeda. Pengambilan video membutuhkan penanda yang sudah terpasang sebagai RoI. Setelah itu dilakukannya perekaman video dilakukan pada saat kereta melintasi jalur lalu melewati JPO. Ukuran resolusi yang diajukan adalah 1080x1920 piksel dengan kecepatan merekam standar sebesar 60 fps (*frame per second*). Posisi kamera yang vertikal dapat



Gambar 3.11: Pengambilan video siang hari



Gambar 3.12: Pengambilan video malam hari

Tabel 3.2: Jadwal kehadiran kereta yang akan dipantau

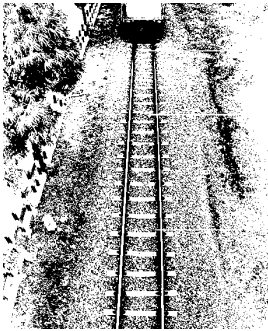
Jadwal Kehadiran	Arah Selatan		Arah Utara	
	Nama KA	Waktu	Nama KA	Waktu
Pagi	Penataran 449	7.35	Penataran 449	7.11
Siang	Probowangi 338	14.35	Probowangi 338	13.50
Sore	Penataran 453	18.15	Penataran 453	17.56
Malam	Probowangi 338	21.54	Probowangi 338	21.10

menangkap gambar rel yang lebih panjang. Berikut jadwal dan jenis kereta yang diambil. Tabel 3.2 merupakan jadwal kereta api pada saat pengambilan data, waktu yang tertera merupakan waktu pada saat melewati jembatan penyebrangan orang di jalan ahmad yani.

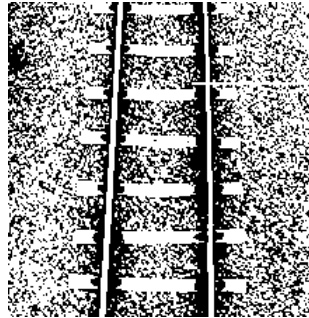
Pengambilan video setiap waktu mendapatkan data video dua arah yaitu selatan dan utara, pertama-tama kereta api menuju utara melewati JPO terlebih dahulu, selang beberapa menit kemudian, kereta api arah selatan akan melewati JPO kembali.

3.4 Deteksi Rel

Proses deteksi rel dilakukan dengan cara Adaptive Thresholding. Pertama-tama semua frame nilainya akan dirata-rata, kemudian program akan menentukan upper dan lower berdasarkan dari nilai rata-rata tersebut. Kemudian hasil threshold akan terlihat berupa gambar 2 bit. Setelah rel telah terlihat seperti pada gambar 3.13 maka dapat menjadi acuan perhitungan rata-rata nilai pixel seperti metodologi pada gambar 3.3.



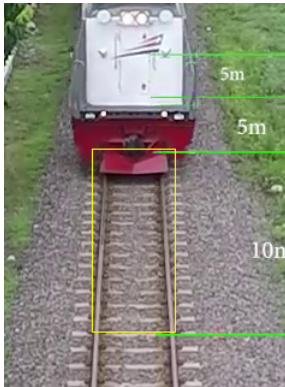
Gambar 3.13: Adaptive Threshold



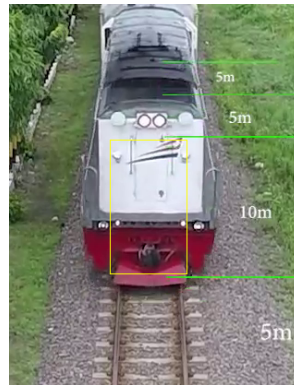
Gambar 3.14: Hasil rel setelah menggunakan adaptive thresholding

3.5 Deteksi Kereta Api

Langkah berikutnya adalah membuat crop area box dengan letak sesuai dimana tempat Roi berada. Crop area pada program dapat di ubah-ubah sesuai dengan RoI yang diinginkan pada GUI. Pada saat kereta api memasuki crop area dari bagian atas (1xB Pixel), maka program akan menentukan arah yaitu kereta berjalan mengarah ke selatan. Jika kereta api memasuki crop area dari bagian bawah (AxB Pixel), maka program akan menentukan arah yaitu kereta berjalan mengarah ke utara. Kemudian hasil arah akan langsung tampil sebagai output. Kemudian pada saat menentukan arah kereta api dan program memulai perhitungan, setelah keluar dari garis tersebut maka proses perhitungan akan berakhir.



Gambar 3.15: Deteksi kereta api pada saat memasuki RoI



Gambar 3.16: Perhitungan frame berhenti setelah keluar RoI

3.6 Perhitungan Kecepatan Kereta Api berdasarkan Time Difference

Pada saat kereta memasuki RoI sampai keluar dari RoI, jumlah frame akan dihitung. Jumlah frame akan dijadikan perbandingan berdasarkan dengan jumlah fps pada video yang digunakan. Karena pada video tersebut 30 frame dapat diambil dalam 1 detik, maka dengan mengetahui jumlah frame pada saat melewati RoI akan didapatkan berapa detik kereta melewati RoI tersebut.

Kecepatan dihitung dengan cara persamaan (1). Hasil jumlah frame didapatkan dengan cara persamaan (2) pada saat kereta berada di dalam RoI. Pada gambar 3.18, merupakan hasil running program di python shell, menggunakan Laptop MSI dengan CPU intel core i7 dan GPU nvidia GTX960. Kecepatan yang ditunjukkan adalah hasil dengan satuan km/jam, count frame yang didapatkan adalah berapa jumlah frame pada saat kereta masuk sampai keluar RoI yang telah ditentukan.

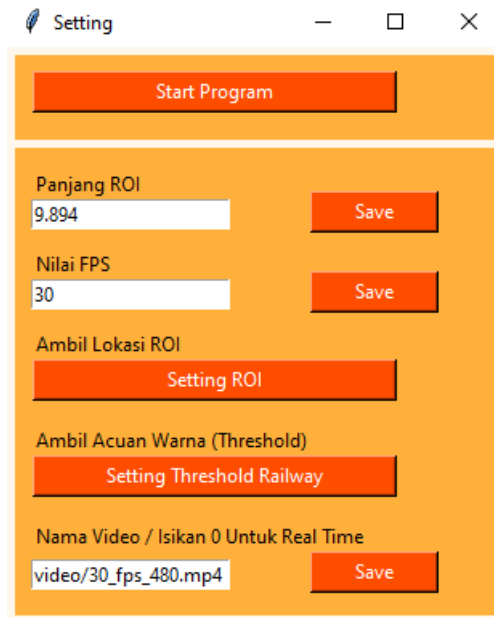


Gambar 3.17: Hasil arah dan kecepatan kereta api


```
Kecepatan : 77.14 Count frame 14  
[INFO] elapsed time: 35.77  
[INFO] approx. FPS: 15.79  
>>>>
```

Gambar 3.18: Hasil arah dan kecepatan kereta api

Hasil akhir arah dan kecepatan ditunjukkan pada gambar 3.17 menggunakan library tkinter untuk GUI nya. Parameter RoI dan jumlah fps kamera dapat diatur menggunakan GUI seperti pada gambar 3.19.

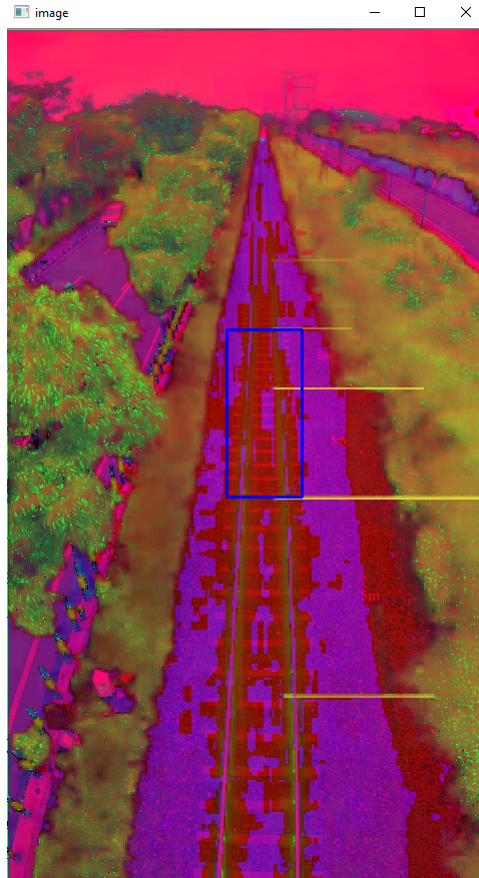


Gambar 3.19: Tampilan GUI untuk pengujian

Panjang ROI dapat diatur sesuai dengan pengukuran yang telah dilakukan. Nilai FPS adalah kemampuan berapa frame per second kamera dapat mengambil video. Kemudian Nama video adalah path file untuk video yang digunakan untuk penelitian. Pada

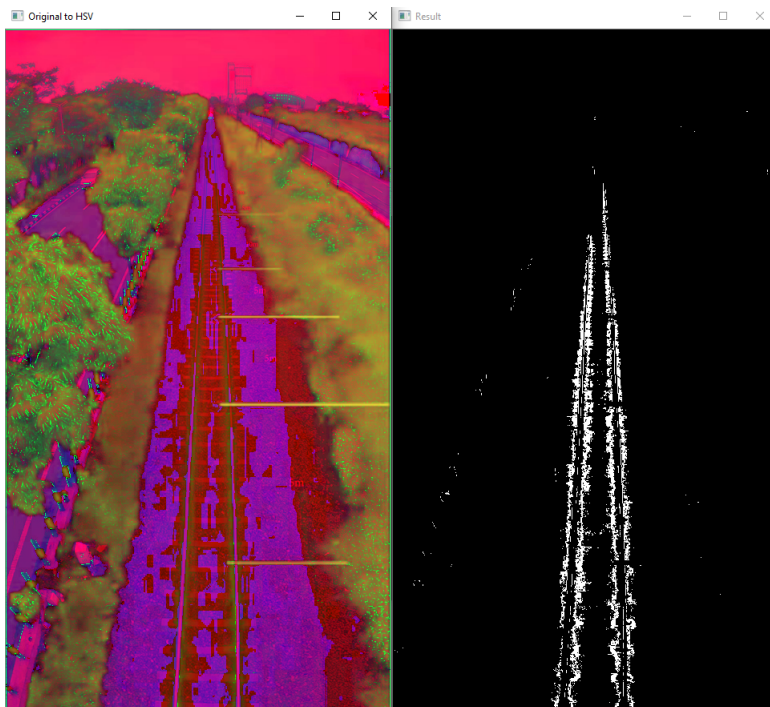
percobaan real time diisi angka 0 untuk menggunakan ip camera.

ROI dapat diatur menggunakan Setting ROI dengan cara menentukan area acuan rel untuk perhitungan kecepatan sesuai dengan panjang ROI yang telah diukur dan dimasukkan. Crop area dapat ditentukan sesuai dengan posisi rel. Tampilan setting ROI ditunjukkan pada gambar 3.20.



Gambar 3.20: Tampilan pada saat setting crop area roi

Pada Setting Threshold dapat diatur threshold untuk menentukan hasil threshold rel sehingga perhitungan kecepatan dapat berjalan. Penentuan acuan nya dapat menggunakan cursor seperti color picker. Tampilan Setting Threshold ditunjukkan pada gambar 3.21.



Gambar 3.21: Tampilan pada saat setting threshold

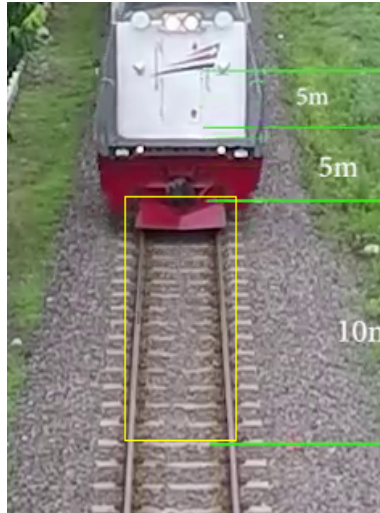
BAB 4

PENGUJIAN DAN ANALISA

Pada bab ini dipaparkan hasil pengujian serta analisa dari desain sistem dan implementasi. Data yang digunakan dalam pengujian program diperoleh dengan merekam video kereta api di JPO Jalan Ahmad Yani yang diambil menggunakan kamera action cam dari atas rel. Data video yang sudah direkam, kemudian diolah oleh program dan dilakukan pengujian dengan beberapa parameter yang berbeda. Pengujian yang dilakukan di bagi menjadi beberapa bagian yaitu sebagai berikut :

1. Pengujian dengan perbedaan jumlah *frame rate*.
2. Pengujian dengan perbedaan jarak RoI.
3. Pengujian dengan perbedaan waktu.
4. Pengujian dengan perbedaan resolusi video.
5. Pengujian secara *real-time*.
6. Pengujian menggunakan video dari Youtube.

Pada pengujian ini diperlukan data berupa video 60, 30, 25, 15 fps. Pada saat pengambilan data, video yang diambil adalah berupa 60 fps, sehingga perlu adanya data sintesis untuk menghasilkan 30,25, dan 15 fps. Video hasil pengambilan data di render ulang menggunakan aplikasi Adobe Premiere Pro sehingga dapat menghasilkan video dengan fps yang diinginkan. Kemudian RoI dibuat dengan cara membuat virtual marker, data aktual diambil dengan cara mengukur panjang jarak antara bantalan kayu rel, dimana pada saat saya ukur semuanya hasilnya sama yaitu 58.4 cm. Jarak antara bantalan kayu dihitung karna terlihat pada frame sehingga dapat disesuaikan untuk mendapatkan jarak RoI yang diinginkan.



Gambar 4.1: Hasil dari virtual marker yang saya buat

4.1 Pengujian dengan Perbedaan Jumlah *Frame Rate*

Pengujian dengan perbedaan jumlah *frame rate* bertujuan untuk mengetahui pengaruh jumlah *frame rate* pada hasil deteksi arah dan perhitungan kecepatan kereta api. Jumlah *frame rate* diatur pada aplikasi video editing Premiere Pro, dengan resolusi video yang digunakan dalam pengujian ini yaitu 1920x1080 piksel. Video yang digunakan adalah pada saat pengambilan data dengan RoI 10 meter. Ada 4 parameter jumlah *frame rate* yang digunakan, yaitu 60 fps, 30 fps, 25 fps, dan 15 fps.

Pada tabel 4.1, merupakan perbandingan antara data perhitungan jumlah kecepatan secara manual menggunakan penglihatan indera berupa kedua mata dengan perhitungan kecepatan secara sistem menggunakan program. Ada 4 jenis kereta dengan perbedaan waktu yang dihitung, P = pagi, Si = Siang, So = Sore, M = Malam dan J= Jumlah.

Setelah perhitungan jumlah kendaraan secara manual dan sistem dibandingkan, maka selanjutnya adalah menghitung nilai ga-

Tabel 4.1: Tabel Pengujian *Kecepatan Kereta Api* dengan Perbedaan Jumlah *Frame Rate* (Arah Selatan)

FPS	Manual (km/jam)					Sistem (km/jam)				
	P	Si	So	M	J	P	Si	So	M	J
60	65	71	68	90	294	68	72	70	0	210
30	65	71	68	90	294	72	67	73	0	212
25	65	71	68	90	294	75	69	77	0	221
15	65	71	68	90	294	77	67	83	0	227

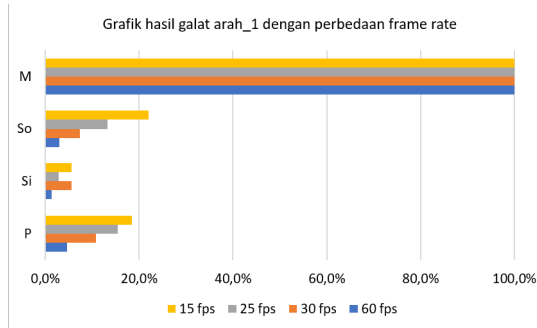
lat. Nilai galat diperoleh dari selisih jumlah manual dan jumlah sistem kemudian dibandingkan dengan jumlah yang sebenarnya dikali 100%. Perhitungan nilai galat untuk setiap pengujian dapat ketahui dengan menggunakan persamaan 4.1.

$$\bar{G} = \frac{JumlahManual - JumlahSistem}{JumlahManual} * 100\% \quad (4.1)$$

Tabel 4.2: Tabel Pengujian Galat *Kecepatan Kereta Api* dengan Perbedaan Jumlah *Frame Rate* (Arah Utara)

FPS	Galat (%)				
	P	Si	So	M	J
60	4,6	1.4	2.9	100	109
30	10.8	5.6	7.4	100	123,8
25	15.4	2.8	13.2	100	131,4
15	18.5	5.6	22.1	100	146,2

Pada tabel 4.2 masing-masing arah kereta dilakukan perhitungan galat. Untuk jumlah galat secara keseluruhan pada deteksi kecepatan kereta sebesar 1.38% pada kecepatan kamera 60 fps, 8.25% pada kecepatan kamera 30 fps, 10.13% pada kecepatan kamera 25 fps dan 15.63% pada kecepatan 15 fps. Semakin kecil jumlah *frame rate* berbanding terbalik dengan galat, maka galat kecepatan kereta api juga akan semakin besar.



Gambar 4.2: Grafik hasil galat dengan perbedaan frame rate

4.2 Pengujian dengan Perbedaan Jarak RoI

Pengujian dengan perbedaan jarak RoI bertujuan untuk mengetahui pengaruh jarak RoI pada hasil deteksi arah dan perhitungan kecepatan kereta api. RoI dipasang dan diletakkan sepasang sesuai dengan jarak yang telah ditentukan di pinggir rel sebelum melakukan pengambilan data dengan resolusi yang digunakan dalam pengujian ini yaitu 1920x1080 piksel dan kecepatan kamera sebesar 60fps. Video yang digunakan adalah pada saat pengambilan data siang hari dan dengan frame rate 60 fps. Ada 4 parameter jarak RoI yang digunakan, yaitu 10m, 15m, 20m, dan 25m.

Tabel 4.3: Tabel Pengujian *Kecepatan Kereta Api* dengan Perbedaan jarak RoI

Meter	Manual (km/jam)			Sistem (km/jam)		
	arah 1	arah 2	Jumlah	arah 1	arah 2	Jumlah
10	71.00	69.67	140.67	71.88	69.67	141.55
15	77.14	68.94	146.08	75.13	68.74	143.87
20	75.65	67.50	143.15	75.65	67.37	143.02
25	74.90	68.35	143.25	74.90	67.85	142.75

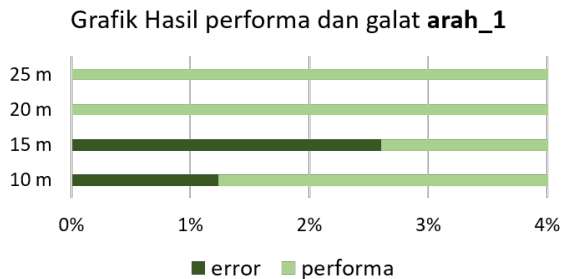
Pada tabel 4.3, merupakan perbandingan antara data perhitungan jumlah kendaraan secara manual menggunakan penglihatan indera berupa kedua mata dengan perhitungan jumlah kendaraan secara sistem menggunakan program. Ada 2 jenis kereta yang dihi-

tung, arah 1 = kereta arah selatan, arah 2 = kereta arah utara, dan J= Jumlah.

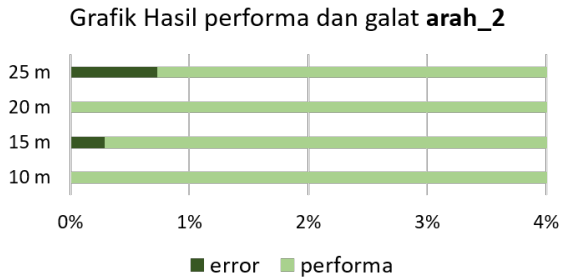
Tabel 4.4: Tabel Pengujian Galat dengan Perbedaan jarak RoI

Meter	Galat (%)		
	arah 1	arah 2	Jumlah
10	1.24	0	1.24
15	0.60	0.29	0.89
20	0	1.07	1.07
25	0	0.73	0.73

Pada tabel masing-masing jenis kendaraan dilakukan perhitungan galat. Untuk jumlah galat secara keseluruhan pada perhitungan kecepatan sebesar 1.24% pada jarak RoI 10 meter, 0.89% pada jarak RoI 15 meter, 1.07% pada jarak RoI 20 meter, dan 0.73% pada jarak RoI 25 meter. Semakin besar jarak RoI, maka galat untuk mendeteksi jenis kendaraan juga akan semakin rendah.



Gambar 4.3: Grafik hasil galat dengan perbedaan roi (arah 1)



Gambar 4.4: Grafik hasil galat dengan perbedaan roi (arah 2)

4.3 Pengujian dengan Perbedaan Waktu

Pengujian dengan perbedaan waktu bertujuan untuk mengetahui pengaruh waktu pada hasil deteksi arah dan perhitungan kecepatan kereta api. Jumlah *frame rate* diatur pada aplikasi video editing Premiere Pro, dengan resolusi video yang digunakan dalam pengujian ini yaitu 1920x1080 piksel. Video yang digunakan adalah pada saat pengambilan data dengan RoI 10 meter. Ada 4 parameter jumlah *frame rate* yang digunakan, yaitu P = pagi, Si = Siang, So = Sore, M = Malam.

Tabel 4.5: Tabel Pengujian *Kecepatan Kereta Api* dengan Perbedaan Waktu (Arah Selatan)

Waktu	Manual (km/jam)					Sistem (km/jam)				
	60	30	25	15	J	60	30	25	15	J
P	65	65	65	65	260	68	72	75	77	292
Si	71	71	71	71	284	72	67	69	67	275
So	68	68	68	68	272	70	73	77	83	303
M	90	90	90	90	360	0	0	0	0	0

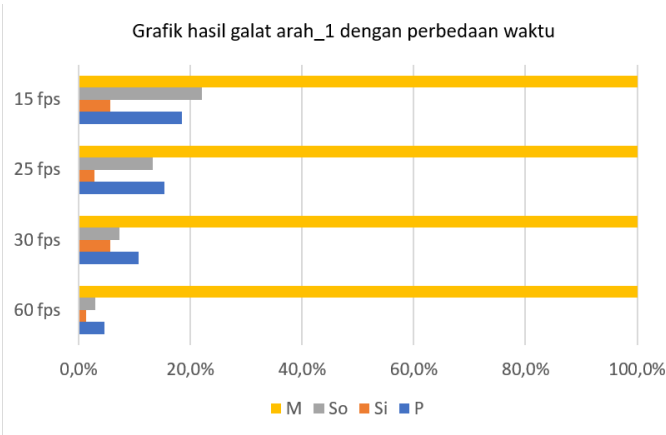
Pada tabel 4.5, merupakan perbandingan antara data perhitungan jumlah kecepatan secara manual menggunakan penglihatan indera berupa kedua mata dengan perhitungan kecepatan secara sistem menggunakan program. Ada 4 jenis kereta dengan perbedaan waktu yang dihitung, yaitu kereta dengan video 60 fps, 30 fps, 25

fps, 15 fps, dan J= Jumlah.

Tabel 4.6: Tabel Pengujian Galat *Kecepatan Kereta Api* dengan Perbedaan Waktu (Arah Utara)

Waktu	Galat (%)				
	60	30	25	15	J
P	4,6	10.8	15.4	18.5	49.2
Si	1.4	5.6	2.8	5.6	15.5
So	2.9	7.4	13.2	22.1	45.6
M	100	100	100	100	400

Pada tabel 4.6 masing-masing arah kereta dilakukan perhitungan galat. Untuk jumlah galat secara keseluruhan pada deteksi kecepatan kereta sebesar 49.2% pada pagi hari, 15.5% pada siang hari, 45.6% pada sore hari dan 400% pada malam hari. Hasil kecepatan terbaik didapatkan pada siang hari, namun pada malam hari terdapat galat sebesar 100% dimana artinya kecepatan tidak dapat terbaca.



Gambar 4.5: Grafik hasil galat dengan perbedaan waktu

4.4 Pengujian dengan Perbedaan Resolusi

Pengujian dengan perbedaan resolusi bertujuan untuk mengetahui pengaruh resolusi pada hasil deteksi arah dan perhitungan kecepatan kereta api. Resolusi video diatur pada aplikasi video editing Premiere Pro, dengan frame rate video yang digunakan dalam pengujian ini yaitu 60 fps. Video yang digunakan adalah pada saat pengambilan data dengan RoI 10 meter. Ada 3 parameter resolusi yang digunakan, yaitu 1080p, 720p, dan 480p.

Tabel 4.7: Tabel Pengujian *Kecepatan Kereta Api* dengan Perbedaan Resolusi

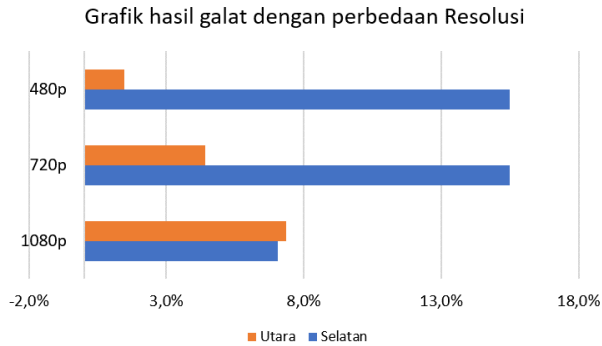
FPS	Manual (km/jam)			Sistem (km/jam)		
	U	S	J	U	S	J
1080p	68	71	139	63	76	139
720p	68	71	139	65	82	147
480p	68	71	139	67	82	149

Pada tabel 4.7, merupakan perbandingan antara data perhitungan jumlah kecepatan secara manual menggunakan penglihatan indera berupa kedua mata dengan perhitungan kecepatan secara sistem menggunakan program. Ada 2 jenis kereta dengan perbedaan arah yang dihitung, U = Utara, S = Selatan, dan J= Jumlah.

Tabel 4.8: Tabel Pengujian Galat *Kecepatan Kereta Api* dengan Perbedaan Resolusi

FPS	Galat (%)		
	U	S	J
1080p	7.0	7.4	14.4
720p	15.5	4.4	19.9
480p	15.5	1.5	17.0

Pada tabel 4.8 masing-masing arah kereta dilakukan perhitungan galat. Untuk jumlah galat secara keseluruhan pada deteksi kecepatan kereta sebesar 14.4% pada resolusi 1080p, 19.9% pada resolusi 720p, dan 17.0% pada resolusi 480p.



Gambar 4.6: Grafik hasil galat dengan perbedaan resolusi



Gambar 4.7: Resolusi 1080p



Gambar 4.8: Resolusi 720p



Gambar 4.9

4.5 Pengujian secara Real Time

Pengujian sistem secara real time bertujuan untuk mengetahui seberapa mampu performa sistem pada hasil deteksi arah dan perhitungan kecepatan kereta api secara real time. Pada saat pengambilan data, resolusi yang digunakan dalam pengujian ini yaitu

480x852 piksel dan kecepatan kamera sebesar 30fps. Video akan diproses di SBC raspberry pie 3B+ kemudian akan memberikan hasil arah dan kecepatan secara langsung. Ada 2 parameter jarak RoI yang digunakan, yaitu 10m, dan 15m.

Tabel 4.9: Tabel Pengujian *Kecepatan Kereta Api* secara real time

Meter	Manual (km/jam)	Real Time (km/jam)
10	71.00	90.00
15	68.74	90.00

Pada tabel 4.9, merupakan perbandingan antara data perhitungan jumlah kendaraan secara manual menggunakan penglihatan indera berupa kedua mata dengan perhitungan jumlah kendaraan secara sistem menggunakan sbc secara real time. Kereta yang di hitung kecepatannya merupakan kereta yang mengarah ke selatan sebanyak 2 kali pengambilan data dengan menggunakan parameter RoI yang berbeda.

Tabel 4.10: Tabel Pengujian Galat secara real time

Meter	Galat (%)
10	26.8
15	30.9

Pada tabel 4.10 kedua kereta dilakukan perhitungan galat. Untuk jumlah galat secara pada perhitungan kecepatan sebesar 26.8% pada jarak RoI 10 meter, dan 30.9% pada jarak RoI 15 meter. SBC tidak dapat melakukan pengukuran kecepatan secara akurat dikarenakan performa yang terbatas untuk pengolahan citra.

4.6 Pengujian menggunakan video dari Youtube

Pada pengujian ini saya mengambil video keamanan kereta api yang terdapat di youtube, kemudian saya melakukan konfigurasi untuk menyesuaikan terhadap video tersebut. Setelah melakukan konfigurasi, hasil dari pengujian seperti pada gambar 4.10.



Gambar 4.10: Hasil pengujian menggunakan video dari youtube

Hasil arah dan kecepatan yang didapatkan bernilai salah dan tidak terhitung, hal ini dikarenakan resolusi video yang rendah, angle kamera yang berada disamping sehingga menghasilkan perspektif, dan threshold dengan kondisi cahaya video sehingga tidak memungkinkan untuk melakukan perhitungan.

Dengan dilaksanakannya beberapa pengujian tersebut, sehingga dapat ditarik kesimpulan dari pelaksanaan tugas akhir ini.

Halaman ini sengaja dikosongkan

BAB 5

PENUTUP

5.1 Kesimpulan

Dalam penelitian ini, telah diimplementasikan serangkaian prosedur untuk menentukan deteksi arah dan kecepatan kereta api. Setelah dilakukan beberapa pengujian data, didapatkan kesimpulan sebagai berikut:

1. Pengujian secara real-time pada SBC membutuhkan waktu untuk memproses data dengan hanya mendapatkan video dengan frame rate 5 – 10 fps (tidak menentu) sehingga mempengaruhi perhitungan kecepatan dengan galat yang tinggi yaitu mencapai 30.9% .
2. Sistem tidak mendapatkan kendala pada saat mendeteksi arah. Tidak terdapat galat pada semua pengujian sehingga program dapat mendeteksi arah dengan akurat.
3. Pengujian jumlah FPS pada kamera, semakin kecil FPS berbanding terbalik dengan galat yang semakin besar, galat terkecil terdapat pada kereta siang hari dengan perekaman 60 fps yaitu sebesar 1.4%.
4. Semakin panjang jarak RoI, semakin berbeda kecepatan dengan perhitungan manual, namun galat perhitungan kecepatan akan lebih rendah.
5. Berdasarkan pengujian dengan perbedaan waktu, hasil perhitungan kecepatan terbaik adalah pada saat siang hari, yaitu menghasilkan galat terendah sebesar 1.4%.
6. Intensitas cahaya pada waktu malam hari mempengaruhi perhitungan kecepatan dengan galat terbesar mencapai 100%. Hal ini disebabkan karena keterbatasan kamera mendapatkan gambar yang jelas pada malam hari.

5.2 Saran

Demi pengembangan lebih lanjut mengenai tugas akhir ini, disarankan beberapa langkah lanjutan sebagai berikut:

1. Penggunaan SBC selain raspberry pie dengan memory dan gpu yang lebih tinggi dapat membantu pemrosesan data lebih cepat.
2. Penelitian selanjutnya dapat menggunakan kamera infrared atau cctv night vision, sehingga data pada malam hari dapat di proses oleh program.

DAFTAR PUSTAKA

- [1] “Tingkatkan keselamatan, kai pasang alat pendeteksi kecepatan.” <https://ekonomi.bisnis.com/read/20140110/98/196539/tingkatkan--kai--alat-pendeteksi-kecepatan>. Terakhir Diakses pada tanggal 1 September 2019. (Dikutip pada halaman , 1).
- [2] “Adaptive thresholding: A comparative study,” 2014. Payel Roy. (Dikutip pada halaman , 9).
- [3] “Spesifikasi gopro hero 4 silver.” <https://ngonoo.com/2016/02/081220/harga-dan--kamera--hero4-silver-edition-di->. (Dikutip pada halaman , 13).
- [4] “Vehicle speed detection by using camera and image processing software,” 2018. Hakan Koyuncu, Baki Koyuncu. (Dikutip pada halaman , 14).
- [5] “Lokasi pengambilan data.” <https://goo.gl/maps/se6bCcFBbQzrTNvr8>. (Dikutip pada halaman , 21).
- [6] “Pengukuran kecepatan kendaraan berbasis deep learning,” 2019. Selvy Andy Wijaya. (Dikutip pada halaman 1, 2).
- [7] Computer Science Handbook. Massachusetts: CRC Press, 2004. A. B. Tucker. (Dikutip pada halaman 5).
- [8] “The implementation of vehicle speed detection using webcam with frame difference method,” 2017. N. H. Tsani. (Dikutip pada halaman 6, 9, 11, 16).
- [9] Computer Vision Metrics: Survey, Taxonomy, and Analysis. Apress; 1st ed. edition, 2014. Scott Krig. (Dikutip pada halaman 8).
- [10] “Adaptivethreshold opencv documentation.” https://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html?highlight=threshold. (Dikutip pada halaman 10).

- [11] Computer Vision: Algorithms and Applications (Texts in Computer Science). Springer; 2011 edition, 2010. Richard Szeliski. (Dikutip pada halaman 10).
- [12] Robust Motion Detection in Real-Life Scenarios (SpringerBriefs in Computer Science). Springer; 2012 edition, 2012. Ester Martinez-Martin, Angel P.del Pobil. (Dikutip pada halaman 11).
- [13] Background Modeling and Foreground Detection for Video Surveillance. Chapman and Hall/CRC; 1 edition, 2015. Thierry Bouwmans, Fatih Porikli, Benjamin Höferlin, Antoine Vacavant. (Dikutip pada halaman 11).
- [14] Learning Python: Learn to code like a professional with Python - an open source, versatile, and powerful programming language. Packt Publishing, 2015. Fabrizio Romano. (Dikutip pada halaman 12).
- [15] “Kamera aksi.” https://id.wikipedia.org/wiki/Kamera_aksi. (Dikutip pada halaman 14).

LAMPIRAN

Berikut lampiran dari hasil output video berdasarkan jadwal kereta api.

Sheet 1

Video size 1920 x 1080 – FPS 60

1:Panataran 449 – Arah Utara – 07.11 WIB

Kecepatan: 73 km/jam

2:Panataran 449 – Arah Selatan – 07.35 WIB

Kecepatan: 65 km/jam

3:Probowangi 338 – Arah Utara – 13.50 WIB

Kecepatan: 70 km/jam

4:Probowangi 338 – Arah Selatan – 14.35 WIB

Kecepatan: 71 km/jam

5:Probowangi 338 – Arah Utara – 13.50 WIB

Kecepatan: 70 km/jam

6:Probowangi 338 – Arah Selatan – 14.35 WIB

Kecepatan: 71 km/jam

7:Probowangi 338 – Arah Utara – 13.50 WIB

Kecepatan: 68 km/jam

8:Probowangi 338 – Arah Selatan – 14.35 WIB

Kecepatan: 71 km/jam

9:Panataran 453 – Arah Utara – 17.56 WIB

Kecepatan: 70 km/jam

10:Panataran 453 – Arah Selatan – 18.15 WIB

Kecepatan: 68 km/jam

11:Probowangi 338 – Arah Utara – 21.10 WIB

Kecepatan: 72 km/jam

12:Probowangi 338 – Arah Selatan – 21.54 WIB

Kecepatan: 90 km/jam

BIOGRAFI PENULIS



Arifandi Putrawardana, lahir pada 18 Juni 1996 di Jakarta. Penulis lulus dari SMA Negeri 3 Malang pada tahun 2014. Penulis kemudian melanjutkan pendidikan S-1 ke Departemen Teknik Komputer, FTE-ITS Surabaya. Saat kuliah penulis aktif menjadi staff UKM Click dan menjadi Koordinator big event Electra 2017. Bagi pembaca yang memiliki kritik, saran atau pertanyaan mengenai tugas akhir ini dapat menghubungi penulis melalui email fandi.vip@gmail.com.

Halaman ini sengaja dikosongkan