TUGAS AKHIR – KM184801

SEGMENTASI VIDEO BERBASIS OBJEK BERGERAK DENGAN *IMPROVED THREE FRAME DIFFERENCE ALGORITHM*

Praditya Lutvi Charisma
NRP 06111640000055

Dosen Pembimbing
Dr. Budi Setiyono, S.Si., M.T.

DEPARTEMEN MATEMATIKA
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya 2020

TUGAS AKHIR – KM184801

SEGMENTASI VIDEO BERBASIS OBJEK BERGERAK DENGAN *IMPROVED THREE FRAME DIFFERENCE ALGORITHM*

Praditya Lutvi Charisma
NRP 06111640000055

Dosen Pembimbing
Dr. Budi Setiyono, S.Si., M.T.

DEPARTEMEN MATEMATIKA
Fakultas Sains dan Analitika Data
Institut Teknologi Sepuluh Nopember
Surabaya 2020

FINAL PROJECT – KM184801

VIDEO SEGMENTATION BASED ON MOVING OBJECT USING *IMPROVED THREE FRAME DIFFERENCE ALGORITHM*

Praditya Lutvi Charisma
NRP 06111640000055

Supervisor
Dr. Budi Setiyono, S.Si., M.T.

DEPARTMENT OF MATHEMATICS
Faculty of Science and Data Analitics
Institut Teknologi Sepuluh Nopember
Surabaya 2020

LEMBAR PENGESAHAN

SEGMENTASI VIDEO BERBASIS OBJEK BERGERAK DENGAN *IMPROVED THREE FRAME DIFFERENCE ALGORITHM*

VIDEO SEGMENTATION BASED ON MOVING OBJECT USING IMPROVED THREE FRAME DIFFERENCE ALGORITHM

TUGAS AKHIR

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

PRADITYA LUTVI CHARISMA

NRP. 06111640000055

Menyetujui
Dosen Pembimbing,

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

Mengetahui,
Kepala Departemen Matematika
FSAD ITS

Subchan, Ph.D
NIP. 19710513 199702 1 001

Surabaya, 19 Agustus 2020

SEGMENTASI VIDEO BERBASIS OBJEK BERGERAK DENGAN *IMPROVED THREE FRAME DIFFERENCE ALGORITHM*

Nama : **Praditya Lutvi Charisma**
NRP : **06111640000055**
Departemen : **Matematika**
Dosen Pembimbing : **Dr. Budi Setiyono, S.Si., M.T.**

ABSTRAK

Rekaman CCTV dimanfaatkan untuk pengecekan peristiwa di area pengawasan. Namun, proses pelacakan membutuhkan waktu lama karena harus mengamati video dari awal, dan mengamati video tanpa aktifitas objek pada interval waktu tertentu. Oleh karena itu, pada penelitian ini dilakukan pemotongan video yang hanya berisi aktifitas objek. Pada penelitian ini, digunakan *improved three frame difference algorithm* untuk didapatkan interval waktu terjadinya gerakan objek pada video dengan melakukan deteksi objek bergerak. Selanjutnya, dilakukan segmentasi video untuk memudahkan pemilik video dalam melakukan pengamatan atau pengecekan rekaman cctv. Pada penelitian ini, dilakukan uji coba terhadap 4 video dengan spesifikasi 4 parameter antara lain: selang frame sebesar 10, *threshold* sebesar 30, *size strel* sebesar 10x10, dan luas minimum sebesar 100. Dari hasil uji coba, didapatkan nilai *f1 score* terendah sebesar 82,9% dan perhitungan *accuracy* terendah sebesar 97,8%. Rata-rata, nilai *f1 score* segmentasi naik sebesar 21,7% daripada hanya menggunakan algoritma *three frame difference.*

Kata kunci: segmentasi video, deteksi objek bergerak, improved three frame difference algorithm.

VIDEO SEGMENTATION BASED ON MOVING OBJECT USING IMPROVED THREE FRAME DIFFERENCE ALGORITHM

Name : Praditya Lutvi Charisma
NRP : 06111640000055
Department : Mathematics
Supervisor : Dr. Budi Setiyono, S.Si., M.T.

ABSTRACT

CCTV recordings are used to check activities which occur in an area of surveillance. However, the tracking process takes time since we need to play the video from the beginning, and we may check on unimportant at certain time intervals. Therefore, this study aims to trim the video that only contain the moving object. This study employes the improved three frame difference algorithm to obtain the time intervals of moving object on the video by detecting its movement. Furthermore, we performance video segmentation to facilitate the authorities either to observe or to check cctv recordings. In this study, we experiment with 4 videos, by specifying 4 parametes : frame intervals is 10, threshold is 30, size strel is 10x10, and minimum area is 100. In the experimental result we obtain the lowest value of F1 Score which is 82.9% and the lowest value of accuracy which is 97,8%. In average, the f1 score value of segmentation increases 21,7% than only three frame difference algorithm.

Keywords : video segmentation, detect moving objects, improved three frame difference algorithm.

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Alhamdulillaahirobbil'aalamiin, segala puji syukur bagi Allah SWT yang telah memberikan limpahan rahmat, petunjuk serta hidayah-Nya, sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul

"SEGMENTASI VIDEO BERBASIS OBJEK BERGERAK DENGAN *IMPROVED THREE FRAME DIFFERECE ALGORITHM*"

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

Tugas Akhir ini dapat diselesaikan dengan baik berkat kerja sama, bantuan, dan dukungan dari banyak pihak. Sehubungan dengan hal tersebut, penulis ingin mengucapkan terima kasih kepada :

1. Bapak Dr. Budi Setiyono, S.Si., M.T. selaku dosen pembimbing atas segala arahan, dukungan, dan motivasinya kepada penulis, sehingga Tugas Akhir ini dapat terselesaikan dengan baik.
2. Bapak Subchan, Ph.D. selaku Kepala Departemen Matematika Institut Teknologi Sepuluh Nopember Surabaya.
3. Ibu Dr. Dwi Ratna Sulistyaningrum, S.Si., M.T., Ibu Dra. Wahyu Fistia Doctorina, M.Si. dan Bapak Mohammad Iqbal, S.Si., M.Si. selaku dosen penguji yang memberikan saran dan rekomendasi terhadap Tugas Akhir ini.
4. Ibu Dr. Valeriana Lukitosari, S.Si., M.T. selaku dosen wali yang telah memberikan nasihat dan arahan selama penulis menempuh perkuliahan di Departemen Matematika ITS.
5. Bapak dan Ibu dosen, seluruh staf Tata Usaha, dan asisten laboratorium Departemen Matematika ITS.

Penulis menyadari bahwa dalam Tugas Akhir ini masih terdapat kekurangan. Oleh karena itu, kritik dan saran yang bersifat membangun sangat diharapkan oleh penulis. Semoga Tugas Akhir ini dapat bermanfaat bagi penulis dan semua pihak yang berkepentingan.

Wassalamu'alaikum Wr. Wb

Surabaya, Agustus 2020

Penulis

Special thanks to

1. Bapak dan Ibu tercinta, Bapak Mundasir dan Ibu Rusmiati, S.Pd. yang selalu ikhlas memberikan dukungan, semangat, do'a dan nasehat-nasehat yang tiada henti kepada penulis yang tak pernah penulis lupakan.
2. Saudara perempuan yang baik, Pratiwi Citra Safitri, S.Kom. yang selalu memberi semangat dan dukungan kepada penulis sehingga Tugas Akhir ini dapat selesai dengan baik.
3. Pakde Haji Safi'i pemilik warung JOI yang telah memberi fasilitas warung sebagai tempat kami berdiskusi dan bercengkrama selama masa perkuliahan di Surabaya.
4. Warga JOI yang selalu merangkul dengan hangat, Mbah Dim, Agun, Fanani, Dley, Memet, Lul-bab, Fizzion, Galung, Cak Haryo, Mas Junjun, Nafiul gaming, dan Bli Wiryo sebagai teman persambatan dan sepertongkrongan.
5. Ario Fajar, Shafira H, Yuda H, dan Diah Ayu sebagai teman satu dosen pembimbing yang selalu memotivasi saya dalam pengerjaan Tugas Akhir ini.
6. Teman-teman Matematika ITS 2016 yang selalu memberikan dukungan dan doa kepada penulis.
7. Seluruh pihak yang telah memberikan saran, dukungan, dan motivasi dalam menyelesaikan Tugas Akhir ini.

DAFTAR ISI

Halaman

HALAMAN JUDUL i

LEMBAR PENGESAHAN vii

ABSTRAK ix

ABSTRACT xi

DAFTAR ISI xvii

DAFTAR GAMBAR xxi

DAFTAR TABEL xxiii

BAB I PENDAHULUAN 1

1.1 Latar Belakang 1

1.2 Rumusan Masalah 3

1.3 Batasan Masalah 3

1.4 Tujuan 4

1.5 Manfaat 4

BAB II TINJAUAN PUSTAKA 5

2.1 Posisi Penelitian 5

2.2 Pengolahan Citra Digital 9

2.3 Video Digital 9

2.4 Operasi Boolean Citra Digital 10

2.5 Outsu *Thresholding* 12

2.6 Morfologi Citra 12

2.7 Prinsip Metode *Frame Difference* 15

2.8 Algoritma *Three Frame Difference* 15

2.9 *Improved Three Frame Difference Algorithm* 17
BAB III METODOLOGI PENELITIAN 19
3.1 Studi Literatur dan Pengumpulan Data 19
3.2 Perancangan Program 20
3.3 Simulasi Program 22
3.4 Penarikan Kesimpulan 23
3.5 Penulisan Laporan..... 23
BAB IV PERANCANGAN DAN IMPLEMENTASI SISTEM .25
4.1 Analisis Sistem 25
4.1.1 Analisis Sistem Perangkat Lunak 26
4.1.2 Analisis Kebutuhan Sistem 27
4.2 Perancangan Sistem 27
4.2.1 Perancangan Data Sistem..... 27
4.2.2 Perancangan Proses Sistem..... 29
4.2.3 Perancangan Antar Muka Sistem 54
4.3 Implementasi Sistem 54
4.3.1 Implementasi Tahap Input Data 54
4.3.2 Implementasi Tahap *Sampling Frame* 56
4.3.3 Implementasi Tahap *Set Region of Interest* (ROI)..... 56
4.3.4 Implementasi Tahap *Pre-processing*..... 57
4.3.5 Implementasi Tahap Deteksi Objek Bergerak 58
4.3.6 Implementasi Tahap Pemotongan dan Penyimpanan 61
4.3.7 Implementasi Tahap Antarmuka 62

BAB V UJI COBA DAN PEMBAHASAN HASIL 65
5.1 Data Uji Coba 65
5.2 Hasil Uji Coba 65
5.2.1 Uji Coba Video 1 66
5.2.2 Uji Coba Video 2 67
5.2.3 Uji Coba Video 3 69
5.2.4 Uji Coba Video 4 70
5.3 Pembahasan Hasil Uji Coba 72
5.3.1 Pemilihan Nilai Parameter 73
5.3.2 Analisa Hasil Uji coba 80
BAB VI PENUTUP 85
6.1 Kesimpulan 85
6.2 Saran 86
DAFTAR PUSTAKA.....87
Lampiran A : Kode Program Otsu Thresholding 90
BIODATA PENULIS.....93

DAFTAR GAMBAR

Gambar 2. 1 Ilustrasi citra digital dalam bentuk matriks 9
Gambar 2. 2 Operasi boolean *AND* 11
Gambar 2. 3 Operasi boolean *OR*.....11
Gambar 2. 4 Operasi boolean *NOT* 12
Gambar 2. 5 Proses dilasi pada *binary image* 13
Gambar 2. 6 Proses erosi pada *binary image* 14
Gambar 2. 7 Diagram blok algoritma *three frame difference* 16
Gambar 2. 8 Hasil algoritma *three frame difference*.....17
Gambar 2. 9 Perbedaan hasil ekstraksi target objek.....18
Gambar 3. 1 Blok Diagram Pengerjaan Tugas Akhir……..19
Gambar 3. 2 Blok Diagram Perancangan Program 20
Gambar 4. 1 Proses *sampling frame* dengan nilai k = 10……30
Gambar 4. 2 Ilustrasi penentuan ROI.....31
Gambar 4. 3 *Frame Region of Interest (ROI)*.....32
Gambar 4. 4 Ilustrasi citra RGB.....33
Gambar 4. 5 Ilustrasi hasil proses grayscalling.....34
Gambar 4. 6 Proses pemilihan tiga citra grayscale berurutan. 35
Gambar 4. 7 Hasil proses pengurangan.....37
Gambar 4. 8 Input untuk proses *otsu thresholding*.....39
Gambar 4. 9 Ilustrasi proses thresholding dengan T=3.....42
Gambar 4. 10 Histogram keabuan proses *threshold* otsu 43
Gambar 4. 11 Hasil proses binerisasi 44
Gambar 4. 12 Simulai binerisasi dengan kasus lain.....44
Gambar 4. 13 Hasil operasi AND 46
Gambar 4. 14 Hasil operasi xor.....47
Gambar 4. 15 Hasil operasi or dua citra beda 48
Gambar 4. 16 Citra hasil dengan *Improved Three Frame Difference Algorithm*.....49
Gambar 4. 17 Perbedaan citra hasil.....50
Gambar 4. 18 Hasil proses morfologi citra 50
Gambar 4. 19 Ilustrasi segmentasi citra 51
Gambar 4. 20 Proses penentukan status gerakan objek.....52

Gambar 4. 21 Ilustrasi anotasi video 53
Gambar 4.22 Desain Antarmuka Sistem 54
Gambar 4.23 Antarmuka *input* video 55
Gambar 4.24 Antarmuka setelah proses input data 55
Gambar 4.25 Antarmuka halaman utama 63
Gambar 4. 26 Antarmuka pengujian.....63
Gambar 5. 1 Hasil deteksi objek bergerak pada video 1……..66
Gambar 5. 2 Interval waktu gerakan pada video 1 66
Gambar 5. 3 Hasil penyimpanan video gerak dari video 1.....67
Gambar 5. 4 Hasil deteksi objek bergerak pada video 2 68
Gambar 5. 5 Interval waktu gerakan pada video 2 68
Gambar 5. 6 Hasil penyimpanan video gerak dari video 2.....69
Gambar 5. 7 Hasil deteksi objek bergerak pada video 3 69
Gambar 5. 8 Interval waktu gerakan pada video 3 70
Gambar 5. 9 Hasil penyimpanan video gerak dari video 3.....70
Gambar 5. 10 Hasil deteksi objek bergerak pada video 4 71
Gambar 5. 11 Interval waktu gerakan pada video 4 71
Gambar 5. 12 Hasil penyimpanan video gerak dari video 4.....72
Gambar 5. 13 Grafik *F1 Score* uji coba selang *frame* 75
Gambar 5. 14 Grafik *F1 Score* uji coba nilai *threshold* 77
Gambar 5. 15 Grafik *F1 Score* uji coba nilai *strel* 78
Gambar 5. 16 Grafik *F1 Score* uji coba nilai luas minimum.....80
Gambar 5. 17 Grafik *F1 Score* hasil deteksi 81

DAFTAR TABEL

Tabel 2. 1 Tabulansi penelitian 7
Tabel 4. 1 Tabel kebutuhan sistem……...26
Tabel 4. 2 Tabel Data Proses.....28
Tabel 4. 3 Tabel perhitungan metode otsu thresholding 41
Tabel 4. 4 Tabel perbedaan varians tiap nilai *threshold*.....42
Tabel 4. 5 Tabel Kebenaran Gerbang Logika 45
Tabel 5. 1 Daftar data uji coba…….65
Tabel 5. 2 Detail rangkaian parameter proses deteksi 73
Tabel 5. 3 Tabel Uji coba nilai selang *frame*.....74
Tabel 5. 4 Tabel uji coba nilai *threshold*.....76
Tabel 5. 5 Tabel Uji coba *size strel morfologi* 77
Tabel 5. 6 Tabel Uji coba nilai luas minimum 79
Tabel 5. 7 Tabel Hasil Uji coba Dengan Modifikasi.....80
Tabel 5. 8 Selisih hasil perbandingan.....82
Tabel 5. 9 Data perhitungan *accuracy*.....83

BAB I
PENDAHULUAN

1.1 Latar Belakang

Sistem keamanan menggunakan CCTV akhir-akhir ini semakin marak dipergunakan di gedung-gedung, jalan kota, hingga rumah pribadi. CCTV (*Closed Circuit Television*) adalah alat perekaman yang menggunakan satu atau lebih kamera video dan menghasilkan data video atau audio. CCTV adalah penggunaan video kamera yang mentransmisikan sinyal atau penyiaran tertuju kepada lingkup perangkat tertentu, yakni seperangkat monitor "spesifik terbatas" [1]. Fungsi utama dari CCTV adalah merekam atau memonitor segala bentuk aktivitas yang terjadi di lokasi pengamatan dengan menggunakan laptop atau PC yang dapat dipantau secara real time dari mana saja, dan dapat merekam seluruh kejadian secara 24 jam.

Wakil Bupati Mimika, Johannes Rettob menjelaskan karena semakin maraknya kasus tindak kriminal yang terjadi di Kota Timika, Pemerintah Mimika menambah jumlah kamera CCTV pada titik-titik rawan tindak criminal di Kabupaten Mimika [2]. Kasat Lantas Polres Majalengka, AKP Endang Sujana juga menjelaskan, pemasangan cctv di beberapa kawasan rawan pelanggaran lalu lintas dapat memudahkan petugas melakukan pencocokan data pelanggar [3]. Dalam bidang keamanan, CCTV berfungsi untuk membantu mengawasi situasi atau kegiatan dilokasi pengawasan. Selain itu, cctv sangat berperan penting sebagai penunjang penyelidikan atau bukti aksi tindak kejahatan yang terjadi [4].

Akan tetapi, penggunaan CCTV saat ini masih kurang efektif karena dalam proses pemantauan atau pengecekan video harus dilakukan secara runtut dengan cara memutar ulang data video. Hal ini akan membutuhkan waktu yang lama dikarenakan data video berasal dari rekaman kamera CCTV yang selalu merekam aktivitas secara 24 jam. Selain itu, data video sering kali berisi *frame-frame* atau bagian kurang berguna disaat tidak ada aktivitas yang terjadi

di lokasi pengamatan. Hal ini akan memperlambat pengawas dalam mencari bagian *frame* data video yang berisi aktivitas penting terkait bukti tindak kejahatan ataupun yang lain.

Berdasarkan permasalahan tersebut, perlu sebuah model sistem yang dapat memisahkan *frame-frame* data video yang mengandung aktivitas dari data video yang tidak mengandung aktivitas atau kondisi lokasi pengamatan kosong. Dengan pemanfaatan deteksi gerak dapat memperoleh kumpulan *frame*, atau interval waktu video yang berisi aktivitas.

Sebelumya, deteksi gerak objek sudah banyak digunakan. Sebagian besar dapat diklasifikasikan ke empat kategori: pemodelan berdasarkan *background subtraction, trajectory classification, low and sparse matrix decompotion, and object tracking* [5]. Salah satu penerapan tersebut adalah sistem deteksi gerak pada kamera pengawas menggunakan algoritma sobel. Dalam penelitian [6], CCTV tidak merekam kondisi lokasi secara 24 jam. Pada IP kamera pengawas akan diimplementasikan sistem deteksi gerak, sehingga dapat mendeteksi setiap pergerakan yang ditangkap oleh IP kamera. Aktivitas atau pergerakan tersebut yang menjadi awal acuan untuk memulai dan mengakhiri proses perekaman. Hasil penelitian menunjukkan akurasi dari sistem akan menurun ketika kondisi ruangan terang, hal tersebut akan riskan jika ada pergerakan yang luput dari sistem dan aktivitas tersebut tidak terekam oleh kamera CCTV [6]. Selain itu sudah diterapkan juga deteksi gerak berdasarkan multi *frame* video. Pada penelitan [7], variansi piksel pada *frame* berperan penting dalam mendeteksi gerakan. Algoritma yang digunakan akan mendeteksi variansi 0 hanya ketika tidak ada gerakan pada *frame* citra. Sangat sederhana untuk mendeteksi gerakan dalam *frame* objek yang bergerak dengan *type* file AVI. Lingkup makalah ini adalah pelacakan tinggi, artinya melacak seluruh tubuh sebagai satu unit. Dengan adanya perubahan variansi piksel, menandakan perubahan dari *frame* saat ini ke *frame* sebelumnya. Artinya, ada aktivitas gerakan pada *frame* saat ini [7].

Berdasarkan latar belakang serta kajian pada penelitian tersebut, penulis ingin mengembangkan sebuah solusi dalam pemisahan potongan kumpulan *frame* dari data video yang berisi aktivitas atau pergerakan yang terjadi di lokasi pengamatan dengan pemanfaatan algoritma *three frame difference* pada data video hasil rekaman kamera CCTV. Algoritma *three frame difference* telah digunakan untuk mendeteksi objek bergerak [8]. Penerapan juga dilakukan pada target manusia dengan kamera infrared dengan latar statis. Hasil penelitian menunjukkan, dengan menggunakan penggabungan algoritma *three frame difference* dan *background difference* mengoptimalkan kelebihan dari dua algoritma tersebut dan menutupi kekurangan keduanya [9]. Modifikasi dilakukan pada algoritma *three frame difference* untuk mengurangi permasalahan yang muncul dimana target objek tidak sepenuhnya tersegmentasi dengan baik [10]. Penulis akan memanfaatkan *improved three frame difference algorithm* untuk melakukan segmentasi video berdasarkan gerakan objek, sehingga pengawas atau pengamat video tidak perlu untuk mengawasi video secara keseluruhan karena membutuhkan waktu yang lama dan sia-sia pada saat kondisi lokasi kosong atau tidak ada aktivitas yang terekam kamera CCTV.

1.2 Rumusan Masalah

Berdasarkan latar belakang permasalahan di atas, rumusan masalah dalam Tugas Akhir ini yaitu :

1. Bagaimana merancang sistem yang dapat melakukan deteksi objek bergerak pada data video CCTV, dan segmentasi video berdasarkan hasil deteksi.
2. Bagaimana menganalisis kinerja *improved three frame difference algorithm* untuk segmentasi video berbasis objek bergerak

1.3 Batasan Masalah

Batasan permasalahan dalam Tugas Akhir ini yaitu:

1. Data yang digunakan adalah video rekaman CCTV pada area dengan intensitas yang cukup dengan latar belakang yang bersifat statis.
2. Tipe video yang digunakan adalah tipe file .mp4, atau .avi.
3. Kumpulan sampel *frame* yang akan diproses, dipilih dari *frame* video dengan selang sebesar konstanta tertentu.

1.4 Tujuan

Tujuan yang ingin dicapai dalam Tugas Akhir ini yaitu :

1. Merancang sistem untuk segmentasi video berbasis objek bergerak pada rekaman CCTV
2. Melakukan segmentasi video yang hanya mengandung rekaman aktivitas objek.
3. Menganalisa kinerja *improved three frame difference algorithm* dalam segmentasi video berbasis objek bergerak.

1.5 Manfaat

Manfaat yang dapat diperoleh dari Tugas Akhir ini adalah memudahkan untuk melacak aktivitas dari rekaman cctv dengan lebih efektif, sehingga tidak harus melakukan pengecekan secara runtut keseluruhan video rekaman.

BAB II
TINJAUAN PUSTAKA

2.1 Posisi Penelitian

Seiring berkembangnya teknologi, beberapa penelitian tentang deteksi objek bergerak telah dilakukan. Dalam penelitian Neelam Patel (2012), deteksi gerak dilakukan dengan cara menganalisa perbedaan dua *frame* citra yang berturut-turut berdasarkan variansi piksel menggunakan *tracking*, *detection*, dan *validation*. Peningkatan efisiensi dalam pelacakan gerakan dilakukan dengan cara pengimplementasian variansi piksel *frame* objek saat ini dan sebelumnya. Dengan objek penelitian berupa citra video ber-type AVI, penelitian ini sangat mendukung untuk kemajuan sistem pengawasan berbasis citra digital. Algoritma yang digunakan dapat mendeteksi variansi piksel sebesar nol hanya ketika tidak ada gerakan pada *frame* video yang berurutan. Jika ada sedikit gerakan, maka akan sangat mudah untuk terdeteksi dengan melacak variansi piksel dari pasangan beberapa *frame* sebuah video [7].

Penelitian juga dilakukan oleh Widyawan, Muhammad Ihsan Zul, Lukito Edi Nugroho (2012). Penelitian ini menangani masalah deteksi gerakan objek dengan metode *frame* differences dan dynamic template matching. Algoritma *frame* differences dilakukan dengan membandingkan komponen RGB pada setiap piksel *frame* gambar yang ter-capture. Gambar yang dijadikan acuan, untuk metode DTM (*Dynamic Template Matching*), adalah kombinasi dari *frame* gambar yang ter-*capture* pada saat waktu $t - 1$, $t - n$, dan gambar acuan akan diperbarui jika area yang diambil kamera terjadi perubahan drastis, seperti contoh: perubahan kecerahan objek dari gelap menjadi terang, perubahan sudut pandang kamera, dan perubahan kondisi lingkungan seperti keluar masuknya objek di area pengamatan [11].

Selanjutnya penelitian juga dilakukan oleh Fadhli R., Fiky Y.S., Junartho H. (2017). Dalam penelitian tersebut, CCTV tidak merekam kondisi lokasi secara 24 jam. Pada IP kamera pengawas

akan diimplementasikan sistem deteksi gerak, sehingga dapat mendeteksi setiap pergerakan yang ditangkap oleh IP kamera. Aktivitas atau pergerakan tersebut yang menjadi awal acuan untuk memulai dan mengakhiri proses perekaman. Gambar yang direkam akan ditransfer ke *cloud storage* secara real time sehingga dapat diakses dimana saja. Namun, dengan begitu maka butuh daya simpan cloud storage yang besar. Berdasarkan hasil penelitian jurnal tersebut, bahwa akurasi dari sistem akan menurun ketika kondisi ruangan terang, hal tersebut akan riskan jika ada pergerakan yang luput dari sistem dan aktivitas tersebut tidak terekam oleh kamera CCTV ketika intensitas cahaya di area pengamatan cukup tinggi [6].

Penelitian juga telah dilakukan oleh M. Lei, J. Geng (2019). Dalam penelitian tersebut, membahas deteksi target manusia dengan latar belakang yang statis, serta permasalahan dari penggabungan kedua algoritma *three frame difference* dan *background difference* untuk menyelesaikan masalah area yang hilang karena *hybrid gaussian background modeling*. Untuk menjawab permasalahan tersebut, penelitian ini mengambil hasil dari metode *mixed gaussian background modeling* sebagai kontur yang diekstraksi dari target yang akan diuji serta menentukan apakah kontur tersebut valid dengan hasil dari algoritma *three frame difference.* Target dianggap sebagai hasil akhir jika kontur tersebut valid, jika tidak maka dianggap sebagai area tidak perlu dan akan dihapus [8]. Adapun beberapa penelitian di atas apabila dibuat tabulasi pada Tabel 2.1.

Tabel 2. 1 Tabulansi penelitian

Penulis	Judul	Tahun	Objek	Keterangan
Neelam Patel	*Motion Detection based on Multi Frame Video under Suveillance System*	2012	Video orang bergerak	Menggunakan pemanfaatan variansi piksel dari *frame-frame* yang berurutan. Algoritma yang digunakan mampu mendeteksi variansi nol hanya ketika ada sedikit gerakan.
Widyawan, Muhammad Ihsan Zul, Lukito Edi Nugroho	*Adaptive Motion Detection Algoritm using Frame Differences and Dynamic Template Matching Method*	2017	Video Objek bergerak	Pendeteksian Gerakan dilakukan dengan membandingkan komponen RGB pada setiap piksel *frame* gambar yang ter-*capture*. Lalu dilakukan pencocokan *frame* saat $t-1, t-n$ dengan *frame* acuan yang ter-capture.

Fadhli R., Fiky Y.S., Junartho H	*Application of Motion Detection on Security Camera using Sobel Algorithm*	2017	Video Objek bergerak	Pemanfaatan deteksi gerak untuk kamera pengawas dengan algoritma sobel. Ada tidaknya gerakan menjadi pemicu dalam memulai atau mengakhiri perekaman.
M. Lei, J. Geng	*Fusion of Three Frame Difference Method and Baground Difference Method to Achive Infrared Human Target Detection*	2019	Video orang bergerak menggunakan infrared dengan latar statis.	Mengkaji deteksi dengan target manusia pada latar statis, serta menyelesaikan masalah area yang hilang karena *hybrid gaussion background modelling*.
Praditya Lutvi Charisma	*Video Segmentation based on Moving Object using Three Frame Difference Algorithm*	2020	Video cctv	Segmentasi video berdasarkan interval waktu objek bergerak. Sehingga didapatkan video aktifitas objek.

2.2 Pengolahan Citra Digital

Pengolahan citra digital (*Digital Image Processing*) adalah sebuah disiplin ilmu yang mempelajari tentang teknik-teknik mengolah citra. Citra yang dimaksud disini adalah gambar diam (foto) maupun gambar bergerak (video). Sedangkan digital disini mempunyai maksud bahwa pengolahan citra/gambar dilakukan secara digital menggunakan komputer [12].

Proses digitalisasi pada citra digital dibagi menjadi dua proses yakni sampling dan kuantisaisi. Proses sampling merupakan proses pengambilan nilai diskrit koordinat ruang (x, y) secara periodik dengan periode sampling T. Proses kuantisasi merupakan proses pengelompokkan nilai tingkat keabuan citra kontinu kedalam beberapa level atau merupakan proses membagi skala keabuan $(0, L)$ menjadi G buah level yang dinyatakan dengan suatu harga bilangan bulat (*integer*), dinyatakan sebagai $G = 2^m$, dengan G adalah derajat keabuan dan m adalah bilangan bulat positif. Jadi, citra digital dapat diwakili oleh sebuah matriks yang terdiri dari M kolom N baris, dimana pada matriks tersebut terdapat piksel (*picture element*), yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna. Nilai yang terdapat pada koordinat (x, y) adalah $f(x, y)$, yaitu besar intensitas atau warna dari piksel di titik itu. Oleh sebab itu, sebuah citra digital dapat ditulis dalam bentuk matriks pada Gambar 2.1.

$$\begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,N-1) \\ f(1,0) & f(1,1) & \cdots & f(1,N-1) \\ \vdots & \vdots & \vdots & \vdots \\ f(M-1,0) & f(M-1,1) & \cdots & f(M-1,N-1) \end{bmatrix}$$

Gambar 2. 1 Ilustrasi citra digital dalam bentuk matriks

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 [13]. Secara mendasar, tidak ada perbedaan proses sampling dan kuantisasi antara citra digital dan video digital.

Bagaimanapun juga, video analog 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 telihat sebagai rangkaian yang kontinu sehingga tercipta ilusi gerak yang halus.

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* [14].

2.4 Operasi Boolean Citra Digital

Operasi boolean disebut juga sebagai operasi logika dimana citra biner sebagai citra inputan dan beberapa operator yang digunakan adalah operasi NOT, AND, OR, dan XOR. Operasi boolean mempunyai terapan yang penting pada pemrosesan morfologi pada citra biner [15].

Operasi logika yang sering digunakan adalah [16] :

a. Operasi Boolean *AND*

   Operasi *AND* berkaitan dengan operasi *intersection* pada himpunan. Operasi boolean *AND* dinyatakan dengan Persamaan (2.1). Contoh proses operasi boolean *AND* dapat dilihat pada Gambar 2.2.

   $$C(x, y) = A(x, y) \cap B(x, y) \quad (2.1)$$

   Keterangan :

   $C(x, y)$ = Citra hasil operasi *AND*

$A(x, y)$ = Citra biner A
$B(x, y)$ = Citra biner B

(a) (b) (c)

Gambar 2. 2 Operasi boolean *AND*, (a) citra A, (b) citra B, (c) citra hasil operasi boolean AND

b. Operasi Boolean *OR*

Operasi *OR* berkaitan dengan operasi *union* pada himpunan. Operasi *OR* dinyatakan dengan Persamaan (2.2). Contoh proses operasi boolean *OR* dapat dilihat pada Gambar 2.3

$$C(x, y) = A(x, y) \cup B(x, y) \quad (2.)$$

Keterangan :

$C(x, y)$ = Citra hasil operasi *OR*
$A(x, y)$ = Citra biner A
$B(x, y)$ = Citra biner B

(a) (b) (c)

Gambar 2. 3 Operasi boolean *OR*, (a) citra A, (b) citra B, (c) citra hasil operasi boolean *OR*

c. Operasi Boolean *NOT*

Operasi *NOT* berkaitan dengan operasi *complement* pada himpunan. Operasi *NOT* dinyatakan dengan Persamaan

(2.4). Contoh proses operasi boolean *NOT* dapat dilihat pada Gambar 2.4.

$$C(x,y) = I - A(x,y) \quad (2.4)$$

Keterangan :

$C(x,y)$ = Citra hasil operasi *NOT*

$A(x,y)$ = Citra biner A

I = Citra biner dengan semua piksel bernilai 1

(a) (b)

Gambar 2. 4 Operasi boolean *NOT*, (a) citra A, (b) citra A^c

2.5 Outsu *Thresholding*

Threshold adalah ambang batas dari intensitas piksel. Nilai intensitas citra yang lebih dari atau sama dengan nilai *threshold* akan diubah menjadi putih (1) sedangkan nilai intensitas yang kurang dari nilai *threshold* akan diubah menjadi hitam (0). Thresholding akan menghasilkan citra biner atau juga disebut sebagai citra B&W (*black and white*). Metode *binary thresholding* mengharuskan untuk menentukan nilai *threshold* secara manual, yang bisa saja kurang *representatif* dengan objek citra. Dengan metode Outsu *thresholding*, penentuan nilai *threshold* yang optimal dilakukan secara otomatis dengan cara meminimkan varians dalam kelas, atau memaksimalkan varians luar kelas [17].

2.6 Morfologi Citra

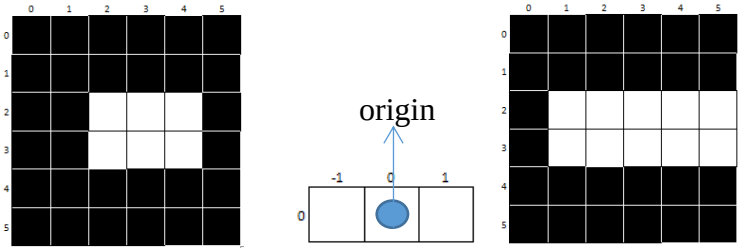
Morfologi adalah teknik pengolahan citra digital dengan menggunakan bentuk (*shape*) sebagai pedoman dalam pengolahan. Nilai dari setiap piksel dalam citra digital hasil diperoleh melalui

proses perbandingan antara piksel yang bersesuaian pada citra digital masukan dengan piksel tetangganya. Operasi morfologi bergantung pada urutan kemunculan dari piksel, tidak memperhatikan nilai *numeric* dari piksel sehingga teknik morfologi sesuai apabila digunakan untuk melakukan pengolahan *binary image* dan *grayscale image*. Dalam gambar biner, himpunan yang dimaksud adalah anggota ruang bilangan bulat 2 dimensi Q^2 dimana setiap elemen himpunan menunjukkan koordinat (x, y) dengan $x, y \in Q$ piksel putih dalam suatu citra [18].

a. Operasi Dilasi

Proses dalam dilasi adalah "penumbuhan" atau "penebalan" dalam citra biner. Setiap piksel dari *background* yang sesuai dengan ukuran dan bentuk dari *structuring element* sehingga menyentuh piksel-piksel objek diubah menjadi piksel objek, hal tesebut membuat objek menjadi lebih "tebal". Jika A dan B adalah anggota Q^2, dilasi antara A (menyatakan objek dari suatu citra) dan B (*structuring element)* dinyatakan dengan Persamaan (2.5).

$$A \oplus B = \{ a + b \mid a + b \in Q^2, \forall a \in A, \forall b \in B\} \tag{2.5}$$



A B $A \oplus B$

Gambar 2. 5 Proses dilasi pada *binary image*

Contoh proses dilasi pada *binary image* dapat dilihat pada Gambar 2.5. Proses penghitungan operasi dilasi pada suatu objek A dengan *structuring element* B adalah sebagai berikut: :

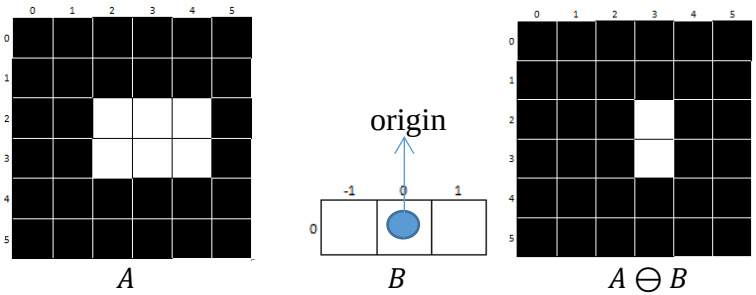
$A = \{(2,2), (2,3), (2,4), (3,2), (3,3), (3,4)\}$
$B = \{(0,-1), (0,0), (0,1)\}$
$A \oplus B = \{(2,1), (2,2), (2,3), (2,4), (2,5), (3,1), (3,2), (3,3), (3,4), (3,5)\}$

b. Operasi Erosi

Erosi adalah operasi morfologi yang mengurangi piksel pada batas antar objek dalam suatu citra digital. Operasi ini menggunakan aturan, yaitu: "Untuk gambar *grayscale* maka nilai hasil operasi (*output pixel*) adalah nilai minimal yang diperoleh dari himpunan piksel tetangganya. Dalam *binary image*, jika ada piksel tetangga yang bernilai 0 maka output piksel diset menjadi 0".

Proses dalam erosi adalah "pengikisan" atau "penipisan" dalam citra biner. Erosi menghasilkan objek pada citra menjadi lebih tipis atau terkikis. Jika A dan B adalah anggota Q^2, erosi antara A (menyatakan objek dari suatu citra) dan B (*structuring element)* adalah himpunan semua elemen $X(x_1, y_1)$. Erosi dari A dan B ditulis $A \ominus B$ didefinisikan dengan Persamaan (2.6).

$$A \ominus B = \{ \boldsymbol{X}(x_1, y_1) \in Q^2 \mid x + b \in A, \forall\, b \in B\} \qquad (2.6)$$



Gambar 2. 6 Proses erosi pada *binary image*

Contoh proses erosi yaitu erosi pada *binary image* yang ditunjukkan pada Gambar 2.6. Proses perhitungan operasi erosi pada suatu objek A dengan *structuring element* B adalah sebagai berikut:

$A = \{(2,2), (2,3), (2,4), (3,2), (3,3), (3,4)\}$
$B = \{(0,-1), (0,0), (0,1)\}$
$A \ominus B = \{(2,3), (3,3)\}$

2.7 Prinsip Metode *Frame Difference*

Prinsip dari algoritma ini sangat sederhana. Metode ini bertujuan untuk mendapatkan perkiraan gerakan pada objek dengan melakukan operasi diferensial pada dua *frame* yang berurutan dari kumpulan *frame* video. Ketika objek bergerak secara tidak normal di area pengawasan, akan tampak jelas perbedaan dari dua *frame* yang berurutan. Untuk analisa karakteristik gerakan di dua *frame* berurutan tersebut, dapat dilakukan pengurangan. Nilai mutlak dari perbedaan nilai pixel dua *frame* berurutan tersebut dapat diperoleh, yaitu dari koordinat yang sama, dibandingkan dengan *threshold* (ambang batas) yang ditetapkan. Rumus matematisnya dijelaskan pada Persamaan (2.7)

$$D_t(x,y) = \begin{cases} 1, & |f_t(x,y) - f_{t-1}(x,y)| \geq T \\ 0, & |f_t(x,y) - f_{t-1}(x,y)| < T \end{cases} \quad (2.7)$$

Keterangan:

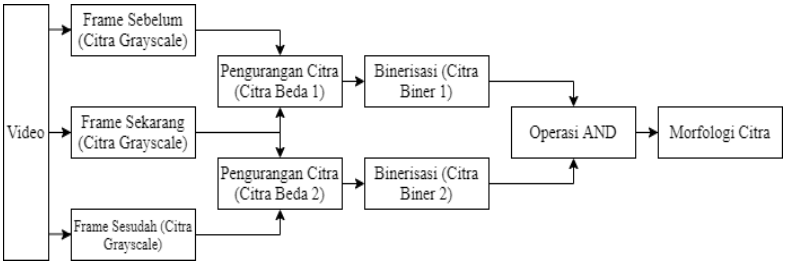
D_t	: Citra beda antara sampel ke t dan $t-1$
(x,y)	: Koordinat piksel
$f_t(x,y)$	: Sampel *frame* ke t
$f_{t-1}(x,y)$	: Sampel *frame* ke $t-1$
T	: Nilai ambang batas (Threshold)

Ketika $D(x,y)$ bernilai 1 mewakili area latar depan atau objek, sedangkan ketika $D(x,y)$ bernilai 0 maka mewakili latar background.

2.8 Algoritma *Three Frame Difference*

Algoritma *three frame difference* merupakan pengembangan dari algoritma *frame difference*. Prinsip dasar dari algoritma *three frame difference* adalah untuk mendapatkan gambar perbedaan dari tiga *frame* berdekatan yang berurutan, kemudian dilakukan operasi AND pada dua gambar biner dengan memilih *threshold* yang

sesuai untuk mendapatkan wilayah dari objek bergerak [8]. Alur kerja algoritma *three frame difference* disajikan pada Gambar 2.7.



Gambar 2. 7 Diagram blok algoritma *three frame difference*

Tiga *frame* berurutan dipilih dari barisan *frame* video, *frame* pertama adalah $f_{t-1}(x,y)$, yang kedua adalah $f_t(x,y)$, dan ketiga adalah $f_{t+1}(x,y)$. Kita dapat meningkatkan kalkulasi dengan cara mengkonversi citra berwarna (RGB) menjadi citra *grayscale*.

Algoritma *frame difference* diterapkan ke citra grayscale untuk mendapatkan citra biner yang bersesuaian dengan cara menghitung perbedaan antara *frame* ke $f_t(x,y)$ dan $f_{t-1}(x,y)$ serta perbedaan antara *frame* ke $f_{t+1}(x,y)$ dan $f_t(x,y)$, didapatkan dua citra beda $D_t(x,y)$ dan $D_{t+1}(x,y)$ antara tiga *frame* yang berurutan.

Dari dua citra beda $D_t(x,y)$ dan $D_{t+1}(x,y)$ antara ketiga *frame* yang berurutan dilakukan operasi boolean AND dan didapatkan citra hasil proses algoritma *three frame difference*, yang merupakan gambar wilayah bergerak sebuah objek. Formula matematika dapat dituliskan pada Persamaan (2.8)

$$Y_t(x,y) = \begin{cases} 1, D_t(x,y) \,\&\, D_{t+1}(x,y) = 1 \\ 0, D_t(x,y) \,\&\, D_{t+1}(x,y) = 0 \end{cases} \qquad (2.8)$$

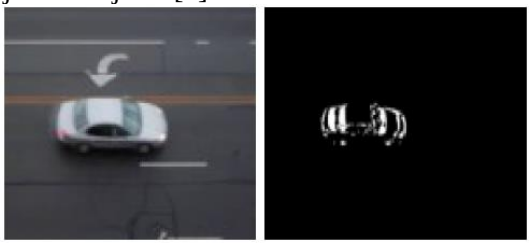
Keterangan :

Y_t : Citra hasil algoritma three *frame* difference

$D_t(x,y)$: Citra beda antara sampel ke t dan $t-1$

$D_{t+1}(x,y)$: Citra beda antara sampel ke $t+1$ dan t

Ketika mendapatkan *intersection* dari dua gambar perbedaan $(D_t(x,y) \& D_{t+1}(x,y))$ seperti pada Gambar 2.8, ada beberapa kesalahan area. Oleh karena itu, menghilangkan *noise* pada citra merupakan hal terpenting. Setelah didapatkan citra biner dengan algoritma *frame difference*, kita dapat menghapus bagian yang tidak perlu dari citra objek yang diekstraksi sehingga membuat *contour* objek lebih jelas [9].



Gambar 2. 8 Hasil algoritma *three frame difference*

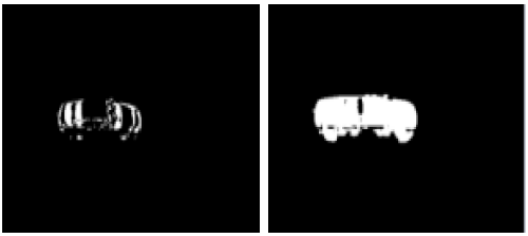
2.9 *Improved Three Frame Difference Algorithm*

Algoritma three *frame* difference mengekstrak target objek bergerak dengan operasi OR, yang dapat mendeteksi objek akan tetapi masih meninggalkan permasalahan. Hasil ekstrak target tidak seluruhnya terdeteksi. Oleh karena itu dilakukan modifikasi untuk menutupi kekurangan tersebut. Target objek bergerak dapat didapatkan dengan melakukan operasi OR pada citra hasil dari algoritma three *frame* difference, dan hasil citra perbedaan kedua. Kemudian operasi OR dilakukan pada $P_t(x,y)$ dan $P_{t+1}(x,y)$, hasil perluasan dari $D_t(x,y)$ dan $D_{t+1}(x,y)$, untuk memaksimalkan area dari target objek. Dan terakhir dilakukan operasi OR dari dua hasil tersebut, sehingga dapat mendeteksi target bergerak secara lengkap. Langkah-langkah algoritma modifikasi sebagai berikut [10] :

1. $D_{t+1}(x,y)$, $D_t(x,y)$, dan $Y_t(x,y)$ merupakan hasil dari algoritma *three frame difference*.
2. Untuk $D_{t+1}(x,y)$ dan $Y_t(x,y)$ dilakukan operasi XOR, sehingga didapatkan citra $M_{t1}(x,y)$.

3. $D_{t+1}(x, y)$ dan $D_t(x, y)$ masing masing diperluas menjadi $P_{t+1}(x, y)$ dan $P_t(x, y)$, untuk selanjutnya dilakukan operasi OR, sehingga didapatkan citra $M_{t2}(x, y)$.
4. Untuk $M_{t1}(x, y)$, dan $M_{t2}(x, y)$ dilakukan operasi OR untuk mendapatkan citra hasil $Z_t(x, y)$. Dimana $Z_t(x, y)$ adalah representasi dari citra hasil dengan *improved three frame difference algorithm.*

Dari Gambar 2.9, dapat dilihat perbandingan hasil ekstraksi objek. Algoritma *three frame difference* mengekstrak target objek secara tidak lengkap dan menjadi masalah selanjutnya jika objek memilik ukuran yang kecil. Algoritma yang telah dimodifikasi dapat diterapkan untuk menutupi kekurangan tersebut [10].

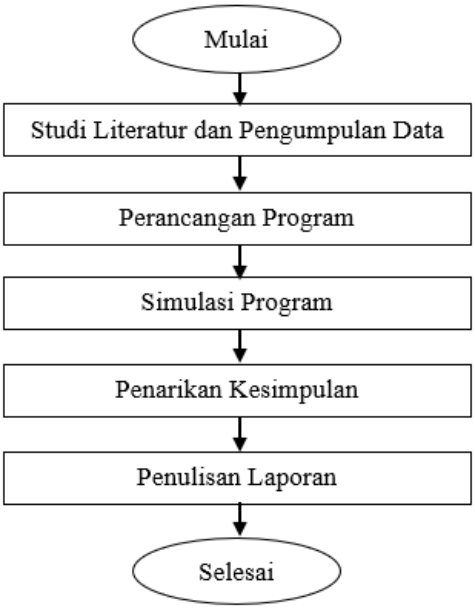


(a) (b)

Gambar 2. 9 Perbedaan hasil ekstraksi target objek, (a) tanpa modifikasi, (b) dengan modifikasi

BAB III
METODOLOGI PENELITIAN

Pada bab ini, akan dijelaskan mengenai metodologi sistem yang digunakan untuk menyelesaian masalah tugas akhir ini agar dapat mencapai tujuan penelitian. Secara umum pengerjaan tugas akhir ini berjalan seperti blok diagram pada Gambar 3.1.



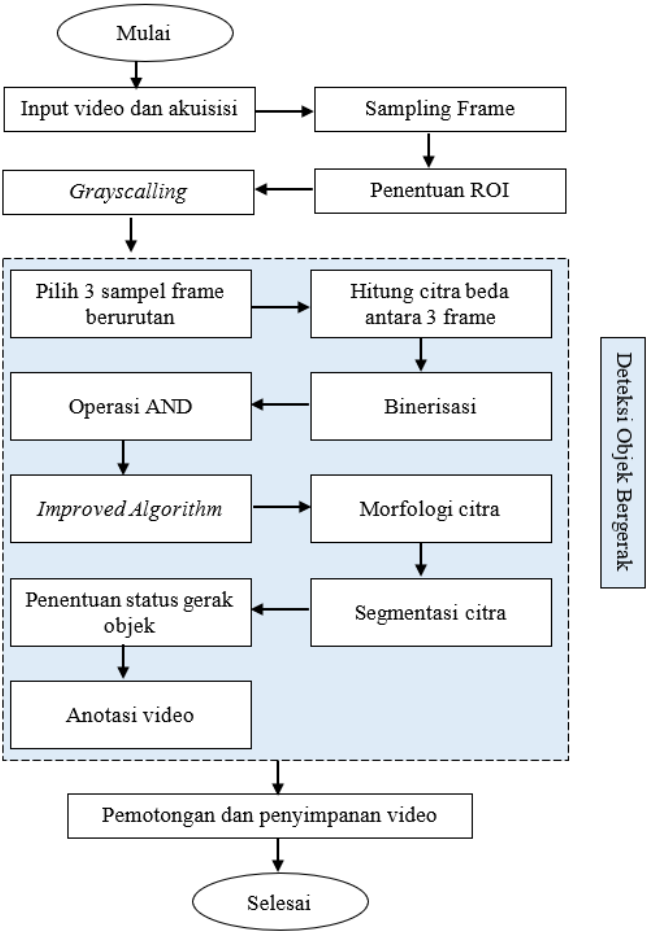
Gambar 3. 1 Blok Diagram Pengerjaan Tugas Akhir

3.1 Studi Literatur dan Pengumpulan Data

Pada tahap ini dilakukan pengumpulan data yang diperlukan untuk pengerjaan tugas akhir ini. Data yang digunakan dalam studi kasus ini berupa data riil dari rekaman cctv yang terpasang di rumah. Kemudian dilakukan studi literatur untuk mendukung pengerjaan penelitian ini dan menambah pemahaman yang lebih mendalam mengenai algoritma *three frame difference* termodifkasi. Literatur yang dipelajari dapat bersumber dari jurnal, buku, internet, maupun bimbingan dengan dosen pembimbing.

3.2 Perancangan Program

Pada tahap ini dilakukan perancangan program. Program yang dibuat menggunakan bahasa pemrograman Java. Proses perancangan program disajikan menggunakan blok diagram pada Gambar 3.2 serta akan diberikan penjelasan pada setiap tahapannya.



Gambar 3. 2 Blok Diagram Perancangan Program

Perancangan program terdiri dari alur program antara lain :

1. Input dan akuisisi video

   Pada tahap ini digunakan video yang telah diambil dalam proses pengumpulan data serta dilakukan akuisisi video menjadi *frame-frame* gambar.

2. Sampling *frame*

   Tahap ini bertujuan untuk memilih kumpulan *frame* secara terutut yang akan digunakan dalam proses berikutnya. Pemilihan sampel dilakukan dengan cara memilih *frame – frame* dengan selang *frame* sebesar konstanta tertentu. Pemilihan sampel ini dilakukan agar waktu pemrosesan video menjadi lebih cepat. Kemudian dilakukan *resizing* yaitu pengubahan dimensi citra menjadi citra dengan dimensi yang lebih kecil untuk mempercepat proses pengolahan citra pada tahap selanjutnya.

3. Penentuan *Region of Interest* (ROI)

   Tahap ini bertujuan untuk mengatur daerah dari *frame* yang nantinya akan di proses. Dengan mengatur ROI, selain mengurangi beban dalam pemrosesan, juga dapat memilih area untuk diproses secara lebih spesifik sehingga mengurangi adanya gangguan yang tidak dibutuhkan.

4. Pre-processing

   Tahap ini bertujuan untuk mempersiapkan komponen-komponen yang digunakan seperti inisialisasi variabel akuisisi data dan perbaikan citra. Pada tahap *pre-processing* dilakukan *grayscalling*.

5. Deteksi objek bergerak

   Tahap deteksi objek bergerak ini bertujuan untuk melacak ada atau tidaknya gerakan suatu objek sebagai tanda suatu aktivitas terjadi. Tahap deteksi objek bergerak dilakukan

dengan *improved three frame difference algorithm*. Berikut adalah langkah-langkahnya :

a. 3 sampel *frame* berurutan dipilih untuk diproses
b. Hitung citra perbedaan antara *frame* 1 dan *frame* 2, serta antara *frame* 2 dan *frame* 3.
c. Binerisasi citra perbedaan dengan cara *thresholding*.
d. Operasi AND dua citra biner
e. *Improved algorithm*
f. Morfologi citra
g. Segmentasi ciitra
h. Tentukan status ada atau tidaknya gerakan berdasarkan besaran area yang tersegmentasi.
i. Anotasi video pada interval waktu objek bergerak.

6. Pemotongan dan penyimpanan video

   Setelah proses deteksi objek bergerak dan mendapatkan interval waktu terjadinya gerakan, dilakukan proses pemotongan video. Tujuan proses ini adalah mendapatkan video yang hanya berisi aktivitas dari objek.

3.3 Simulasi Program

Pada tahap ini dilakukan pembuatan implementasi dari tugas akhir ini berupa simulasi program berdasarkan perancangan program yang telah dilakukan dengan menggunakan bahasa pemrograman Java dengan library OpenCV dan Ffmpeg. Pada tahap ini juga akan dilihat apakah program dapat mendeteksi objek bergerak dan melakukan segmentasi video. Apabila terjadi kesalahan, akan dilakukan perbaikan program sehingga didapat hasil yang sesuai dengan yang diinginkan. Selanjutnya dilakukan pengujian untuk mengetahui kinerja dari *improved three frame difference algorithm*.

3.4 Penarikan Kesimpulan

Pada tahap ini, dilakukan penarikan kesimpulan dari hasil pembahasan sebelumnya, serta diberikan beberapa saran untuk pengembangan penelitian selanjutnya.

3.5 Penulisan Laporan

Pada tahap akhir dalam tugas akhir ini adalah penulisan laporan seluruh tahapan yang telah dilakukan.

BAB IV
PERANCANGAN DAN IMPLEMENTASI SISTEM

Pada bab ini dibahas mengenai analisis dan perancangan sistem pada penelitian ini, diantaranya adalah proses pengambilan data, proses *pre-processing* data masukan berupa video input, proses pendeteksian objek bergerak dengan menggunakan *improved three frame difference algorithm* dan yang terakhir adalah proses pemotongan dan penyimpanan video. Selain proses-proses tersebut, perancangan sistem juga meliputi perancangan antarmuka (*interface*) untuk memudahkan dalam melakukan uji coba dan analisis. Pada bab ini juga dilakukan implementasi terhadap perancangan yang telah dibuat.

4.1 Analisis Sistem

Analisis sistem secara umum adalah memandang, mengamati dan menyimpulkan konsep sistem berdasarkan sistem informasi secara fisik dan konseptual. Dalam penelitian ini, analisis sistem meliputi :

a. Analisis Fungsional

1. Sistem dapat menerima masukan berupa rekaman video yang diambil dari sebuah jembatan penyebrangan yang kemudian citra hasil ekstraksi video tersebut dapat diubah menjadi *binary image*.
2. Sistem mampu mengimplementasikan *improved three frame difference algorithm* untuk proses deteksi objek bergerak.
3. Sistem mampu mendapatkan interval waktu aktivitas objek terjadi berdasarkan proses deteksi objek bergerak.
4. Sistem dapat melakukan pemotongan dan penyimpanan video yang berisi aktivitas objek.

b. Analisis Non-Fungsional

Analisis non-fungsional dari sistem pemotongan video berbasis objek bergerak meliputi perangkat keras dan lunak

komputer. Detail dari perangkat keras dan lunak yang digunakan diberikan pada Tabel 4.1.

Tabel 4. 1 Tabel kebutuhan sistem

Perangkat Keras	Prosesor	: AMD FX-7600P Radeon R7, 12 Compute Cores 4C+8G (4 CPUs), ~2.7GHz
	RAM	: 8192 MB
Perangkat Lunak	Netbeans IDE 8.2 dengan bahasa pemrograman Java, Library OpenCV 4.0.0, dan Ffmpeg 4.2.2	
	SceneBuilder 8.5.0 sebagai tools untuk membuat *user interface*	

4.1.1 Analisis Sistem Perangkat Lunak

Sistem perangkat lunak yang dibangun ini memiliki beberapa tahapan sebagai berikut:

a. Input dan akuisisi video

Data masukan berupa rekaman cctv yang telah diambil kemudian di-*scanning* yaitu proses pemecahan video atau ektraksi video menjadi rangkaian *frame* dan dilakukan *sampling frame* dengan jarak *k* (konstanta) tertentu atau dapat dikatakan *frame-frame* dengan posisi kelipatan dari suatu *k* (konstanta).

b. *Pre-processing*

Pre-processing terdiri dari beberapa proses yang bertujuan untuk mempersiapkan komponen-komponen yang digunakan sebelum diolah ke tahap selanjutnya. Proses dalam *pre-processing* adalah :

1. *Grayscalling*

   Proses pengubahan citra berwarna menjadi citra keabuan untuk pengolahan pada tahap selanjutnya.

c. Deteksi Objek Bergerak

Setelah didapat sampel citra uji, dilakukan deteksi objek bergerak menggunakan *improved three frame difference algorithm*, untuk mendapatkan interval waktu adanya gerakan pada objek.

d. Pemotongan dan Penyimpanan Video
   Setelah mendapatkan interval waktu video, proses pemotongan dilakukan untuk mendapatkan video baru yang berisi aktivitas objek.

4.1.2 Analisis Kebutuhan Sistem

Program dibuat dalam bentuk *desktop* dengan menggunakan bahasa pemrograman JAVA dengan bantuan IDE Netbeans 8.2. *Library* utama yang dibutuhkan untuk membangun sistem adalah OpenCV. OpenCV adalah sebuah *library* untuk pengolahan citra dan bersifat *open source*. Selain OpenCV, Ffmpeg digunakan untuk proses pemotongan video menjadi beberapa video berdasarkan hasil deteksi objek bergerak. Ffmpeg adalah aplikasi command line bersifat open source yang dapat merekam, mengkonversikan dan streaming audio atau video digital.

4.2 Perancangan Sistem

Pada subbab ini dijelaskan mengenai perancangan pada sistem yang telah dianalisis sebelumnya. Diawali dengan perancangan data yang meliputi data masukan, data proses, dan data keluaran kemudian adapun dijelaskan mengenai desain antarmuka sistem yang dibangun sebagai sarana untuk memproses data masukan dan parameter masukan yang ada dan melihat hasil yang diperoleh.

4.2.1 Perancangan Data Sistem

Data yang digunakan untuk deteksi objek bergerak dan pemotongan video terdiri dari data masukan, data proses, dan data keluaran. Data masukan merupakan video digital rekaman cctv yang didapat dari rumah sebagai data sekunder. Data proses adalah data masukan yang dibutuhkan oleh metode-metode dari serangkaian proses yang telah disebutkan sebelumnya. Sedangkan data keluaran adalah informasi mengenai hasil proses deteksi objek bergerak berupa interval waktu dan hasil pemotongan video berupa video yang hanya berisi aktivitas objek.

a. Data Masukan

Data masukan sistem ini berupa video rekaman cctv yang terpasang di rumah. Selain itu pengambilan data juga dilakukan menggunakan kamera *handphone*. File rekaman cctv yang diambil memiliki spesifikasi sebagai berikut:

- Format ekstensi rekaman video berupa *.avi atau *.mp4
- Kecepatan rekaman video cctv sebesar 12fps, sedangkan untuk rekaman handphone sebesar 25fps.

b. Data Proses

Data proses adalah 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. Tabel 4.2 menjelaskan tahapan dari data proses.

Tabel 4. 2 Tabel Data Proses

No.	Tahapan	Input	Output
1.	Inputan Awal	Video	*Frame* citra
2.	Sampling *Frame*	*Frame* citra	Sampel *frame*
3.	Area ROI	Sampel *frame*	Citra ROI
4.	*Pre-processing*	Citra ROI	Citra hasil *pre-processing*
5.	Deteksi Objek Bergerak	Citra hasil *pre-processing*	Interval waktu objek bergerak
6.	Pemotongan Video	Interval waktu	Citra video
7.	Penyimpanan Video	Citra video	Video baru

c. Data Keluaran

Data keluaran merupakan hasil deteksi objek bergerak berupa interval waktu video adanya aktifitas objek dan potongan video baru berdasarkan hasil deteksi objek bergerak. Program ini juga menghasilkan keluaran berupa video yang menampilkan proses deteksi objek bergerak menggunakan *improved three frame difference algorithm.*

4.2.2 Perancangan Proses Sistem

Proses-proses yang dilakukan dalam membangun sistem segmentasi video berbasis deteksi objek bergerak dijelaskan secara rinci setiap tahapan, dimulai dari proses *pre-processing*, kemudian deteksi objek bergerak dengan *improved three frame difference algorithm*, dan yang terakhir proses pemotongan dan penyimpanan video.

a. Sampling *Frame*

Data masukan berupa video yang telah diekstrak menjadi kumpulan *frame* kemudian dilakukan pemilihan *frame* untuk menjadi sampel yang akan digunakan pada sistem segmentasi video berbasis deteksi objek bergerak. Pemilihan sampel dilakukan dengan acuan Persamaan 4.1.

$$A = \{a_0, a_k, a_{2k}, .., a_n\} \tag{4.1}$$

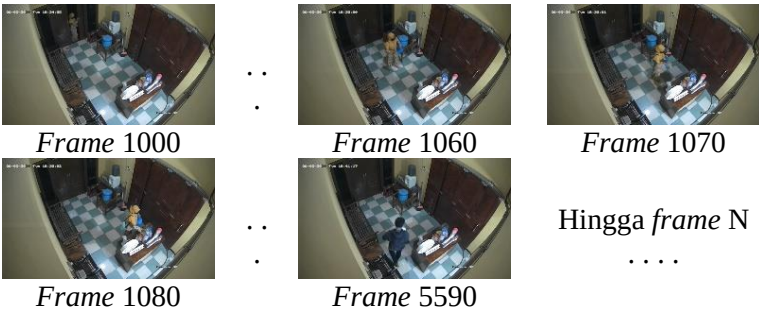
Keterangan :

A : Himpunan sampel *frame*

a_k : *Frame* ke k dari hasil ekstraksi video

k : Konstanta lompat/selang

Tujuan adanya sampling *frame* adalah mengurangi beban komputasi, karena tidak semua *frame* digunakan untuk proses. Proses *resizing* juga dilakukan unutk mengubah dimensi citra menjadi citra dengan dimensi yang lebih kecil. Proses *sampling* dapat dilihat pada Gambar 4.1.



Gambar 4. 1 Proses *sampling frame* dengan nilai k = 10

Maka didapat barisan citra sebanyak n, yang digambarkan dengan $\{f_0(x,y), \dots, f_t(x,y), \dots, f_n(x,y)\}$, dimana $f_t(x,y)$ merepresentasikan sampel *frame* ke t dari barisan *frame*. Pada tahap uji coba dilakukan pemilihan nilai parameter selang *frame* yang optimal dengan nilai sebesar 3, 5, dan 10.

b. Perancangan Proses Penentuan ROI

Data sampel *frame* yang terurut, dilakukan penentuan ROI. ROI berguna untuk mengurangi beban komputasi karena tidak semua piksel dalam *frame* diproses. Selain itu ROI berguna untuk mengurangi gangguan yang ada seperti area pepohonan yang bergerak karena hembusan angin, ataupun gangguan yang lain. Langkah pertama dalam penentuan ROI adalah membuat bentuk segi empat pada bidang citra *frame* video.

Misalkan diambil (x_1, y_1) sebagai titik sudut kiri atas ROI pada (200,50), dengan panjang ROI adalah 850 dan lebar 700, sehingga didapat sebuah titik sudut yang lain yaitu titik sudut kanan bawah (x_2, y_2).

$$(x_2, y_2) = (x_1 + \text{panjang}, y_1 + \text{lebar})$$
$$= (200 + 850, 200 + 700)$$
$$= (1050, 750)$$

Sehingga didapat juga dua titik sudut yang lainnya yaitu titik sudut kiri bawah (x_1, y_2) = (200, 750) dan titik sudut kanan atas (x_2, y_1) = (1050,50). Ilustrasi penentuan ROI diberikan pada Gambar 4.2.

Gambar 4. 2 Ilustrasi penentuan ROI

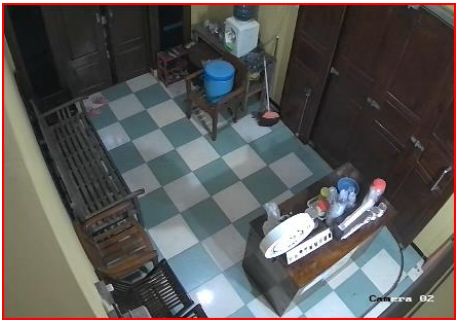
Area ROI dipilih berdasarkan area yang sekiranya akan ada perubahan intensitas yang signifikan. Seperti area seperti dinding ruangan yang secara umum tidak akan ada perubahan intensitas yang signifikan, maka pemilihan roi menghindari area dinging. Contoh proses penentuan *ROI* pada *frame* berukuran 10x4 ditampilkan pada ilustrasi berikut:

$$\left\{\begin{array}{cccc}
(15,33,124) & (16,33,113) & (161.5,11.8,108) & (17,213,114) \\
(13,45,185) & (9,4,165) & (221,12.14,105) & (45,53,107) \\
(40,95,195) & (0,0,253) & (4.31,103.5,170) & (11,43,111) \\
(21,12,206) & (0,0,255) & (9.58,102,255) & (244,5,119) \\
(80,99,232) & (2.8,95,255) & (11.9,32,255) & (5.66,66.52,115) \\
(39,134,214) & (5.86,147,255) & (42.5,1,255) & (0,115.7,174) \\
(68,119,212) & (3.68,150,255) & (0,0,255) & (4.47,107,181) \\
(79,65,190) & (1.28,142.62,236) & (254.5,95,255) & (10.2,83.33,153) \\
(82,15,166) & (12.5,95.27,182) & (9.44,78.23,176) & (247.6,46.54,126) \\
(55,13,139) & (71,9,158) & (45,51.36,139) & (27,30,109)
\end{array}\right\}$$

Titik-titik koordinat sudut *roi* yang dipilih adalah (5,2), (5,4), (9,2), dan (9,4), maka *frame* roi berukuran 5x3 diilustrasikan sebagai berikut:

$$\left\{\begin{array}{ccc}
(2.8,95,255) & (11.9,32,255) & (5.66,66.52,115) \\
(5.86,147,225) & (42.5,1,255) & (0,115.7,174) \\
(3.68,150,255) & (0,0,255) & (4.47,107,181) \\
(1.28,142.62,236) & (254.5,95,255) & (10.2,83.33,153 \\
(12.5,95.27,182) & (9.44,78.23,176) & (247.6,46.54,126)
\end{array}\right\}$$

Gambar 4.2 adalah *frame* gambar yang telah ditentukan ROI. Terlihat pada gambar terdapat batas kotak berwarna merah yang merupakan ROI. Selanjutnya, kotak merah tersebut akan disimpan sebagai *frame* baru seperti Gambar 4.3. *Frame* ROI ini yang akan digunakan dalam proses pendeteksian objek bergerak.



Gambar 4. 3 *Frame Region of Interest (ROI)*

c. Perancangan Proses *Pre-processing*

Untuk proses pre-processing, dilakukan langkah sebagai berikut :

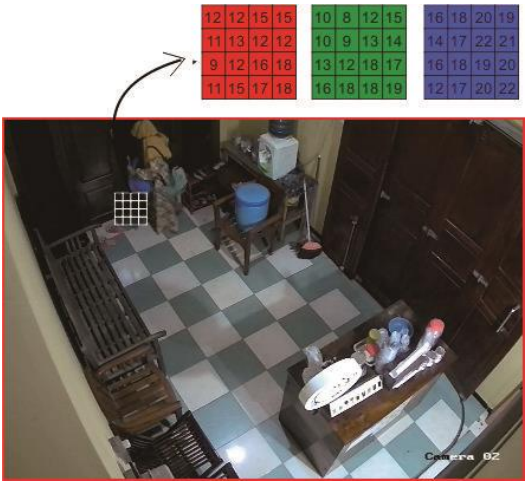
1. *Grayscalling*

   Frame ROI dalam bentuk citra RGB kemudian diubah ke dalam bentuk citra *grayscale* dengan cara perubahan nilai dari citra ROI tersebut. Karena citra memiliki dimensi yang cukup relatif besar, maka untuk menunjukkan perhitungan pembentukkan citra *grayscale* diambil sebagian citra pada salah satu *frame* video berupa matriks 4x4. Perhatikan Gambar 4.4.

   Misal diberikan contoh matriks r, g, b dari citra tersebut seperti pada matriks dibawah ini.

$$r = \begin{pmatrix} 12 & 12 & 15 & 15 \\ 11 & 13 & 12 & 12 \\ 9 & 12 & 16 & 18 \\ 11 & 15 & 17 & 18 \end{pmatrix}, g = \begin{pmatrix} 10 & 8 & 12 & 15 \\ 10 & 9 & 13 & 14 \\ 13 & 12 & 18 & 17 \\ 16 & 18 & 18 & 19 \end{pmatrix}$$

$$b = \begin{pmatrix} 16 & 18 & 20 & 19 \\ 14 & 17 & 22 & 21 \\ 16 & 18 & 19 & 20 \\ 12 & 17 & 20 & 22 \end{pmatrix}$$



Gambar 4. 4 Ilustrasi citra RGB

Proses *grayscalling* dilakukan dengan menggunakan metode Luminosity yang mengalikan setiap nilai RGB dengan konstanta yang telah ditentukan. Secara matematis dapat di tulis sebagai persamaan 4.2.

$$grayscale = 0.21*r + 0.72*g + 0.07*b \quad (4.2)$$

Citra yang akan dirubah menjadi citra grayscale didapat sebagai berikut

$$grayscale = 0.21 \begin{pmatrix} 12 & 12 & 15 & 15 \\ 11 & 13 & 12 & 12 \\ 9 & 12 & 16 & 18 \\ 11 & 15 & 17 & 18 \end{pmatrix} + 0.72 \begin{pmatrix} 10 & 8 & 12 & 15 \\ 10 & 9 & 13 & 14 \\ 13 & 12 & 18 & 17 \\ 16 & 18 & 18 & 19 \end{pmatrix} + 0.07 \begin{pmatrix} 16 & 18 & 20 & 19 \\ 14 & 17 & 22 & 21 \\ 16 & 18 & 19 & 20 \\ 12 & 17 & 20 & 22 \end{pmatrix}$$

$$= \begin{pmatrix} 11 & 10 & 13 & 15 \\ 10 & 10 & 13 & 14 \\ 12 & 12 & 18 & 17 \\ 15 & 17 & 18 & 19 \end{pmatrix}$$

Sehingga citra berubah menjadi citra grayscale yang dapat dilihat pada Gambar 4.5.

Gambar 4. 5 Ilustrasi hasil proses grayscalling

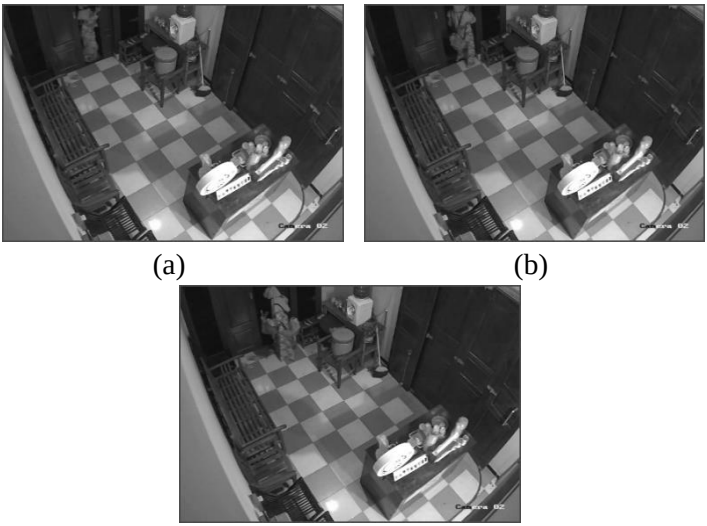
d. Perancangan Proses Deteksi Objek Bergerak

Pada proses deteksi dengan *improved three frame difference algorithm*, dilakukan beberapa proses antara lain:

1. Pengambilan tiga sampel *frame* yang berurutan.

   Tiga buah sampel citra *grayscale* yang berurutan, misalkan $f_{(i-1)}(x, y)$, $f_i(x, y)$, dan $f_{i+1}(x, y)$, dipilih

untuk kemudian diproses. Proses pemilihan dapat dilihat pada Gambar 4.6



(a) (b)

(c)

Gambar 4. 6 Proses pemilihan tiga citra grayscale yang berurutan. (a) citra 1, (b) citra 2, (c) citra 3.

2. Pencarian dua buah citra beda dari tiga sampel

   Citra beda didapatkan dengan cara pengurangan citra grayscale. Pengurangan citra dilakukan antara citra 2 dan citra 1, serta antara citra 3 dan citra 2. Secara umum, proses pengurangan citra dirumuskan seperti pada persamaan 4.3.

$$\begin{aligned} D_t(x,y) &= |f_t(x,y) - f_{t-1}(x,y)| \\ D_{t+1}(x,y) &= |f_{t+1}(x,y) - f_t(x,y)| \end{aligned} \tag{4.3}$$

   Keterangan:

   $D_t(x,y)$: Citra beda antara sampel ke t dan $t-1$

   $D_{t+1}(x,y)$: Citra beda antara sampel ke $t+1$ dan t

(x, y) : Koordinat piksel

$f_{t-1}(x, y)$: Sampel *frame* ke $t - 1$

$f_t(x, y)$: Sampel *frame* ke t

$f_{t+1}(x, y)$: Sampel *frame* ke $t + 1$

Untuk penjelasan proses pengurangan citra, misalkan A, B, dan C adalah tiga buah citra *grayscale* yang direpresentasikan oleh matrik:

$$A = \begin{pmatrix} 35 & 42 & 21 & 15 \\ 21 & 30 & 50 & 14 \\ 18 & 12 & 26 & 41 \\ 17 & 24 & 15 & 19 \end{pmatrix}, B = \begin{pmatrix} 26 & 10 & 13 & 15 \\ 10 & 10 & 13 & 14 \\ 12 & 15 & 18 & 17 \\ 15 & 17 & 18 & 19 \end{pmatrix}$$

$$C = \begin{pmatrix} 11 & 2 & 18 & 50 \\ 34 & 10 & 16 & 45 \\ 41 & 12 & 25 & 20 \\ 50 & 42 & 27 & 31 \end{pmatrix}$$

Maka didapatkan dua buah citra beda yaitu :

$$D_1 = |A - B|$$

$$= \left| \begin{pmatrix} 35 & 42 & 21 & 15 \\ 21 & 30 & 50 & 14 \\ 18 & 12 & 26 & 41 \\ 17 & 24 & 15 & 19 \end{pmatrix} - \begin{pmatrix} 26 & 10 & 13 & 15 \\ 10 & 10 & 13 & 14 \\ 12 & 15 & 18 & 17 \\ 15 & 17 & 18 & 19 \end{pmatrix} \right|$$

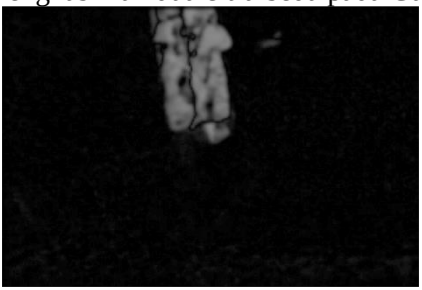
$$= \begin{pmatrix} 9 & 32 & 8 & 0 \\ 11 & 20 & 37 & 0 \\ 6 & 3 & 8 & 24 \\ 2 & 7 & 3 & 0 \end{pmatrix}, \text{dan}$$

$$D_2 = |B - C|$$

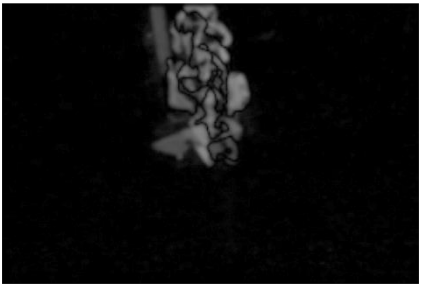
$$= \left| \begin{pmatrix} 26 & 10 & 13 & 15 \\ 10 & 10 & 13 & 14 \\ 12 & 15 & 18 & 17 \\ 15 & 17 & 18 & 19 \end{pmatrix} - \begin{pmatrix} 11 & 2 & 18 & 50 \\ 34 & 10 & 16 & 45 \\ 41 & 12 & 25 & 20 \\ 50 & 42 & 27 & 31 \end{pmatrix} \right|$$

$$= \begin{pmatrix} 15 & 8 & 5 & 35 \\ 24 & 0 & 3 & 31 \\ 29 & 3 & 7 & 3 \\ 35 & 25 & 9 & 12 \end{pmatrix}$$

Sehingga pada proses ini didapatkan dua buah citra beda. Contoh proses pengurangan citra *grayscale* pada Gambar 4.6 akan menghasilkan dua citra beda pada Gambar 4.7



(a)



(b)

Gambar 4. 7 Hasil proses pengurangan; (a) Citra 1 dan 2; (b) Citra 2 dan 3

3. Binerisasi dua buah citra beda dengan *thresholding*

   Binerisasi citra adalah mengubah nilai intensitas menjadi biner dengan cara membandingkan intensitas setiap piksel terhadap suatu nilai. Secara matematis ditunjukkan pada persamaan 4.4.

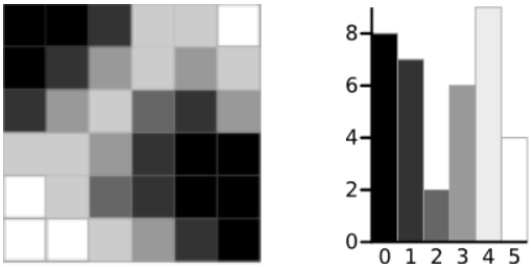
$$H_t(x, y) = \begin{cases} 1, D_t \geq T \\ 0, D_t < T \end{cases}$$

$$H_{t+1}(x,y) = \begin{cases} 1, D_{t+1} \geq T \\ 0, D_{t+1} < T \end{cases} \quad (4.4)$$

Keterangan:

$H_t(x,y)$: Citra biner dari citra beda satu
$H_{t+1}(x,y)$: Citra biner dari citra beda dua
(x, y) : Koordinat piksel
$D_t(x,y)$: Citra beda antara sampel ke t dan $t-1$
$D_{t+1}(x,y)$: Citra beda antara sampel ke $t+1$ dan t
T : Nilai ambang batas (*threshold*)

Citra beda yang berbentuk citra *grayscale* diubah menjadi citra biner dengan metode *otsu thresholding*. Metode Otsu melibatkan iterasi semua kemungkinan nilai ambang yang mungkin dan menghitung ukuran penyebaran tingkat piksel untuk setiap sisi ambang, yaitu piksel di latar objek atau latar *background*. Tujuan iterasi ini adalah menentukan nilai *threshold* dimana jumlah penyebaran piksel latar objek dan latar *background* minimum (atau dapat dikatakan variansinya minim) [17]. Untuk menyederhanakan penjelasan perhitungan, digunakan gambar berukuran 6x6, dan histogram gambar dengan hanya 6 level skala keabuan yang digunakan. Gambar beserta histogram perhitungan dapat dilihat pada Gambar 4.8. Jadi pada kasus ini, metode otsu akan mencoba seluruh kemungkinan nilai *threshold* yang mungkin, yaitu 0,1,2,3,4, dan 5.



(a) (b)

Gambar 4. 8 Input untuk proses *otsu thresholding*;(a) citra 6x6;(b) histogram citra dengan 6 skala keabuan

Maka matrik representasi dari citra 6x6 tersebut adalah sebagai berikut:

$$\begin{pmatrix} 0 & 0 & 1 & 4 & 4 & 5 \\ 0 & 1 & 3 & 4 & 3 & 4 \\ 1 & 3 & 4 & 2 & 1 & 3 \\ 4 & 4 & 3 & 1 & 0 & 0 \\ 5 & 4 & 2 & 1 & 0 & 0 \\ 5 & 5 & 4 & 3 & 1 & 0 \end{pmatrix}$$

Langkah pertama, menghitung varians latar objek dan latar background untuk nilai *threshold* tunggal, masing masing didapat dengan rumus varians pada persamaan 4.5

$$\sigma^2 = \sum_{i=1}^{N} \frac{f_i\,(X_i - \mu)^2}{f_i} \qquad (4.5)$$

dimana $\mu = \sum_{i=1}^{N} \frac{f_i X_i}{f_i}$

X_i adalah data ke-i, f_i adalah banyak data ke-i, μ adalah rata-rata populasi, dan N adalah banyak data. Misal pada kasus perhitungan ini, nilai *threshold* yang digunakan adalah 3, maka matrik untuk analisa pada kasus threshold adalah 3 sebagai berikut:

$$\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

Untuk piksel bernilai 1 mewakili kelas *foreground*, dan bernilai 0 mewakili kelas *background*.

Didapatkan :

Probabilitas $W_b = \frac{8 + 7 + 2}{36} = 0{,}4722$

Rata-rata $\mu_b = \frac{(0\times8)+(1\times7)+(2\times2)}{36} = 0{,}6471$

Varians ${\sigma_b}^2 = \frac{\left((0 - 0{,}6471)^2 \times 8\right)}{17} + \frac{\left((1 - 0{,}6471)^2 \times 7\right)}{17} + \frac{\left((2 - 0{,}6471)^2 \times 2\right)}{17}$

$$= \frac{(0{,}4187 \times 8) + (0{,}1246 \times 7) + (1{,}8304 \times 2)}{17}$$

$= 0{,}4637$

Dengan cara sama, untuk kelas latar objek didapatkan :
$W_f = 0{,}5278$; $\mu_b = 3{,}8947$;dan ${\sigma_b}^2 = 0{,}5152$

Langkah selanjutnya yaitu menghitung nilai varians dalam kelas, yang didefinisikan sebagai jumlah perkalian varians dari dua kelas (latar objek dan latar *background*) dengan probabilitas kelas masing-masing. Secara matematis dirumuskan pada persamaan 4.6.

$$\sigma_W{}^2 = W_b {\sigma_b}^2 + W_f {\sigma_f}^2 \qquad (4.6)$$

Dengan ${\sigma_b}^2$ adalah varians kelas *background*; W_b probabilitas kelas *background*; ${\sigma_f}^2$ adalah varians kelas objek; W_f probabilitas kelas objek, f kelas *foreground*

dan b kelas *background* . Maka untuk kasus ini nilai *threshold* = 3, didapatkan :

$$\sigma_W{}^2 = (0{,}4722 \times 0{,}4637) + (0{,}4722 \times 0{,}4637)$$
$$= 0{,}4909$$

Untuk kemungkinan nilai *threshold* yang lain, pada kasus ini adalah 0 sampai 5, juga dilakukan perhitungan dengan langkah yang sama. Pada Tabel 4.3 dijabarkan proses perhitungan untuk semua kemungkinan nilai *threshold* pada kasus Gambar 4.8.

Tabel 4. 3 Tabel perhitungan metode otsu thresholding

	T=0	T=1	T=2	T=3	T=4	T=5
W_b	0	0,2222	0,4167	0,4722	0,6389	0,8889
μ_b	0	0	0,4167	0,6471	1,2609	2,0313
$\sigma_b{}^2$	0	0	0,2489	0,4637	1,4102	2,5303
W_b	1	0,7778	0,5833	0,5278	0,3611	0,1111
μ_b	2,3611	3,0357	3,7143	3,8947	4,3077	5
$\sigma_f{}^2$	3,1196	1,9639	0,7755	0,5152	0,2130	0
$\sigma_W{}^2$	3,1196	1,5268	0,5561	**0,4909**	0,9779	2,2491

Untuk proses penentuan nilai *otsu threshold* yang optimal, dapat dilakukan pendekatan dengan cara menghitung varians luar kelas, dimana proses perhitungannya jauh lebih cepat. Hal ini berlandaskan, nilai nya dengan nilai varians luar kelas yang maksimum, juga memiliki varians dalam kelas yang minimum. Jadi pendekatan ini dapat digunakan untuk menentukan nilai *threshold* yang optimal karena menjadi lebih sederhana. Nilai varians luar kelas didapat dengan persamaan 4.7.

$$\sigma_B{}^2 = \sigma^2 - \sigma_W{}^2$$
$$= W_b\,(\mu_b - \mu)^2 + W_f\,(\mu_f - \mu)^2;$$
$$\text{dimana } \mu = (W_b\mu_b + W_f\mu_f)$$

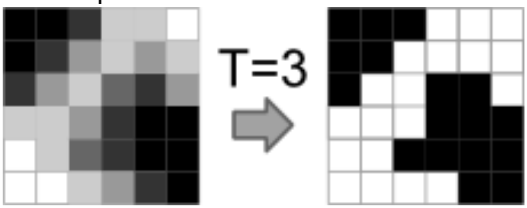
$$= W_b\, W_f\, (\mu_b - \mu_f)^2 \qquad (4.7)$$

Perbedaan varians untuk setiap kemungkinan nilai *threshold* dapat dilihat pada Tabel 4.4.

Tabel 4. 4 Tabel perbedaan varians tiap nilai *threshold*

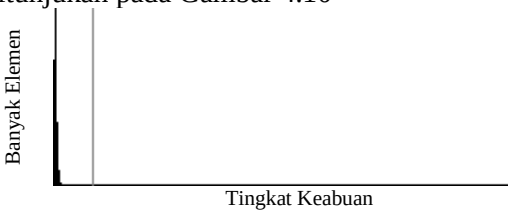
	T=0	T=1	T=2	T=3	T=4	T=5
${\sigma_W}^2$	3,1196	1,5268	0,5561	**0,4909**	0,9779	2,2491
${\sigma_B}^2$	0	1,5928	2,5635	**2,6287**	2,1417	0,8705

Maka untuk contoh penjelasan kasus pada Gambar 4.9, dengan nilai threshold adalah 3, proses thresholding diilustrasikan pada Gambar 4.9.



Gambar 4. 9 Ilustrasi proses thresholding dengan T=3

Sehingga, proses perhitungan untuk citra pada Gambar 4.8(a) akan menghasilkan nilai *threshold* sebesar 22 dan Gambar 4.8(b) akan menghasilkan nilai *threshold* sebesar 17 dengan persebaran tingkat keabuan piksel ditunjukan pada Gambar 4.10



(a)

(b)

Gambar 4. 10 Histogram persebaran tingkat keabuan proses perhitungan *threshold* otsu pada;(a) Citra 4.7a;(b) Citra 4.7b

Jika citra grayscale pada gambar 4.7 diambil sebagian pikselnya dan direpresentasikan menjadi matriks A, maka matriks *thresholding*-nya adalah sebagai berikut :

$$D_1 = \mathrm{T}\begin{pmatrix} 57 & 59 & 50 & 60 & 61 \\ 60 & 62 & 57 & 61 & 59 \\ 55 & 59 & 57 & 62 & 56 \\ 53 & 58 & 52 & 63 & 54 \\ 55 & 57 & 63 & 59 & 57 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 \end{pmatrix}$$

$$D_2 = \mathrm{T}\begin{pmatrix} 30 & 46 & 50 & 60 & 61 \\ 52 & 62 & 60 & 62 & 65 \\ 55 & 59 & 57 & 62 & 59 \\ 61 & 58 & 52 & 63 & 54 \\ 60 & 72 & 45 & 53 & 57 \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$$

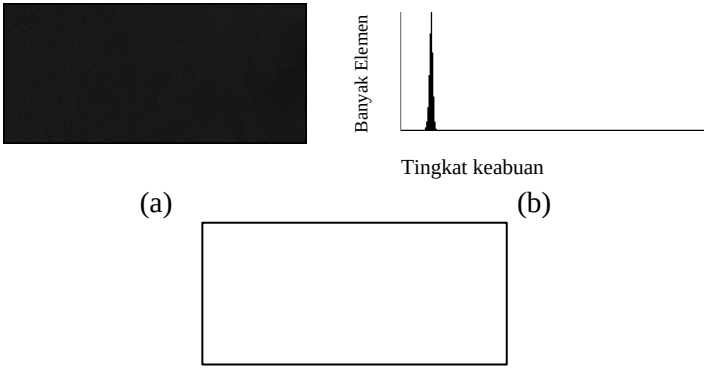
Hasil proses binerisasi untuk citra pada Gambar 4.7 diberikan pada Gambar 4. 11.

(a)

(b)

Gambar 4. 11 Hasil proses binerisasi;(a) Citra biner beda kesatu; (b) Citra biner beda kedua

Selama simulasi, ada kekurangan yang ditemukan. Yaitu, ketika latar objek dan latar background tidak memiliki perbedaan yang jelas, akan mengakibatkan proses penentuan threshold menjadi tidak representatif. Gambar 4.12 menunjukan bahwa, ketika tidak ada objek atau dikatakan selisih antara latar objek dan *background* sangat minim akan membuat proses perhitungan *threshold* tidak representatif.



Gambar 4. 12 Simulai binerisasi dengan kasus lain;(a) citra input;(b) histogram citra a; (c) hasil binerisasi

Dalam hal ini, mengakibatkan metode *otsu thresholding* kurang tepat untuk digunakan dalam proses binerisasi di Tugas Akhir ini. Karena akan mempengaruhi proses penentuan status objek bergerak, dimana penentuan status gerak atau tidaknya objek berdasarkan luas/besarnya kontur piksel warna putih yang mewakili objek.

Sehingga proses binerisasi pada tugas akhir ini menggunakan *basic thresholding* dengan kekurangan pengaturan nilai *threshold* dilakukan secara manual. Pada tahap uji coba dilakukan pemilihan nilai parameter *threshold* sebesar 30, 60, dan 100.

4. Operasi AND antara dua buah citra biner

   Setelah mendapatkan dua citra beda dalam bentuk citra biner, maka dilakukan operasi AND untuk mendapat citra biner yang merepresentasikan daerah gerakan objek. Langkah operasi AND, digunakan untuk mendapatkann irisan dari citra beda pertama dan citra beda kedua. Dimana irisan ini menandakan area objek bergerak. Operasi AND merupakan operasi titik piksel yang bersesuaian antara citra pertama dan citra kedua. Operasi AND hanya dapat dilakukan jika kedua citra memiliki dimensi yang sama. Ketentuan operasi logika dapat dilihat pada Tabel 4.5.

Tabel 4. 5 Tabel Kebenaran Gerbang Logika

X	Y	X & Y	X | Y	X xor Y
0	0	0	0	0
0	1	0	1	1
1	0	0	1	1
1	1	1	1	0

Sehingga jika citra biner pada Gambar 4.11 diambil sebagian piksel-nya menjadi matriks A dan B, maka matriks operasi AND-nya adalah sebagai berikut :

A .& B =

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} .\& \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

Dimana simbol .& adalah operasi AND antara titik-titik yang bersesuaian koordinat pikselnya.

Hasil proses operasi AND diberikan pada Gambar 4.13. Terlihat bahwa area gerakan objek ditandai dengan warna putih yang merupakan irisan dari dua buah citra biner.

Gambar 4. 13 Hasil operasi AND

5. *Improved algorithm*

Karena hasil deteksi objek bergerak menggunakan algoritma *three frame difference* masih meninggalkan kekurangan, dimana algoritma ini mengekstrak target objek secara tidak lengkap, maka dilakukan modifikasi untuk menutupi kekurangan tersebut [10]. Hasil dari algoritma *three frame difference* pada proses sebelumnya, didapatkan citra beda pertama, citra beda kedua, dan citra hasil. Langkah-langkah modifikasi yang dilakukan berdasarkan paper [10] yaitu :

a. Operasi XOR citra hasil dengan citra beda kedua.

Operasi XOR dilakukan dari hasil operasi AND, yang ditunjukan pada Gambar 4.13, dengan citra beda kedua yang ditunjukan pada Gambar 4.11.

Untuk menunjukkan proses perhitungan, jika diambil sebagian piksel menjadi matriks A dan B, maka matriks operasi XOR adalah sebagai berikut :

A .xr B =

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} .\text{xr} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Dimana simbol .xr adalah operasi XOR antara titik-titik yang bersesuaian koordinat pikselnya.

Hasil proses operasi XOR diberikan pada Gambar 4.14. Terlihat bahwa area objek yang tidak menjadi bagian dari citra hasil pada proses sebelumnya.

Gambar 4. 14 Hasil operasi xor

b. Operasi OR antara kedua citra beda.

Operasi OR dilakukan pada dua buah citra beda yang ditunjukkan pada Gambar 4.11, untuk mendapatkan area gerak objek secara maksimum.

Untuk menunjukkan proses perhitungan, diambil sebagian piksel menjadi matriks A dan B, maka matriks operasi OR adalah sebagai berikut :

A .or B =

$$\begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} .\text{or} \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Hasil proses operasi OR diberikan pada Gambar 4.15. Terlihat bahwa area objek yang tidak menjadi bagian dari citra hasil pada proses sebelumnya.

Gambar 4. 15 Hasil operasi or dua citra beda

c. Operasi OR

Langkah modifikasi terakhir yaitu operasi OR antara citra hasil dari dua langkah modifikasi sebelumnya. Untuk menunjukkan proses perhitungan, jika diambil sebagian piksel dari dua citra yang ditunjukan pada Gambar 4.14, dan Gambar 4.15 menjadi matriks A dan B, maka matriks operasi OR adalah sebagai berikut :

A .or B =

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \text{or} \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

Hasil proses operasi OR diberikan pada Gambar 4.16. Terlihat bahwa area objek yang tidak menjadi bagian dari citra hasil pada proses sebelumnya.

Gambar 4. 16 Citra hasil dengan *Improved Three Frame Difference Algorithm*

Terlihat perbedaan antara hasil algoritma modifikasi dan hasil tanpa modifikasi algoritma yang ditunjukkan pada Gambar 4.17.

(a)

(b)

Gambar 4. 17 Perbedaan citra hasil;(a) algoritma *three frame difference*;(b) *improved three frame difference algorithm*

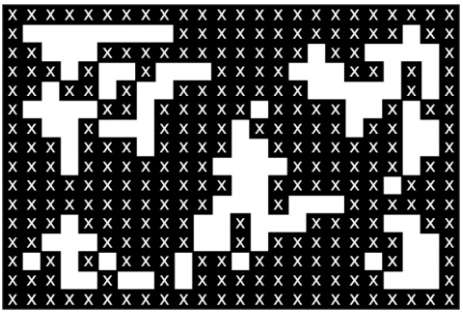
6. Morfologi Citra

Hasil dari proses-proses sebelumnya, meninggalkan *noise* dan lubang pada citra. Proses morfologi citra dilakukan untuk menghilangkan *noise* serta mengisi lubang pada citra dengan tujuan untuk hasil proses segmentasi yang lebih baik. *Strel* berbentuk piringan (*disk*) digunakan untuk proses morfologi closing dan opening. Pada tahap uji coba dilakukan pemilihan nilai parameter *size strel* berukuran 3x3, 5x5, dan 10x10. Hasil proses morfologi ditunjukkan pada Gambar 4.18.

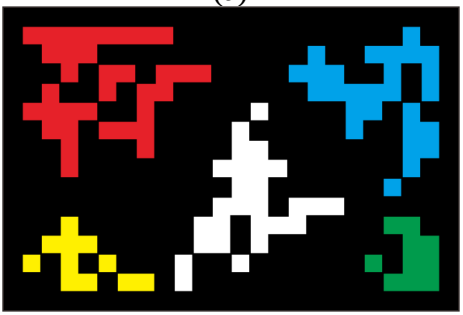
Gambar 4. 18 Hasil proses morfologi citra

7. Segmentasi objek

Dari hasil *improved three frame difference algorithm* didapatkan citra biner yang lebih jelas. Kemudian dilakukan segmentasi yaitu pengelompokan piksel-piksel berdasarkan nilai intensitasnya (memisahkan antara yang putih dan hitam) yang saling berhubungan. Dengan dilakukan segmentasi, didapatkan suatu kontur berwarna putih sebagai representasi area dimana objek telah bergerak seperti pada ilustrasi Gambar 4.19.



(a)



(b)

Gambar 4. 19 Ilustrasi segmentasi citra;(a) Objek dipisahkan yang tidak memiliki ketetanggaan;(b) Hasil pengelompokan objek

8. Penentuan adanya gerakan

   Setelah mendapatkan citra yang merepresentasikan area dimana sebuah objek itu bergerak, masing masing area akan dihitung luasan area kontur. Jika pada citra hasil segmentasi terdapat sedikitnya satu area yang luasnya melebihi luas minimal yang telah ditentukan, maka telah terjadi gerakan pada waktu tersebut. Jika tidak ada satupun, maka dikatakan tidak terjadi aktifitas gerakan.

   Proses penentuan ditunjukan pada Gambar 4.20, terlihat bahwa jika terdapat sedikitnya satu luas area melebihi luas minimal maka dikatakan telah terjadi gerakan. Pada tahap uji coba dilakukan pemilihan nilai parameter luas minimum sebesar 100, 150, dan 250.

Gambar 4. 20 Proses penentukan status gerakan objek

   Jika objek bergerak keluar dari area dan selanjutnya tidak ada objek lain yang bergerak, maka aktifitas gerakan dikatakan telah berhenti. Tetapi jika setelahnya ada objek yang lain yang bergerak menyusul, maka aktifitas gerak dikatakan masih berjalan.

9. Anotasi Video

   Selanjutnya dilakukan anotasi video, yaitu proses melabeli video pada interval waktu tertentu menjadi video objek bergerak. Dari hasil proses penentuan status

gerak, didapatkan status bergerak atau tidaknya objek. Status gerakan ini menjadi acuan untuk melakukan penyimpanan interval waktu. Jika status objek bergerak, maka waktu video disimpan sebagai waktu awal terjadinya gerakan. Sedangkan jika status objek tidak bergerak, maka waktu video disimpan sebagai waktu akhir terjadinya gerakan. Ilustrasi penyimpanan interval waktu ditunjukan pada Gambar 4.21

Gambar 4. 21 Ilustrasi anotasi video

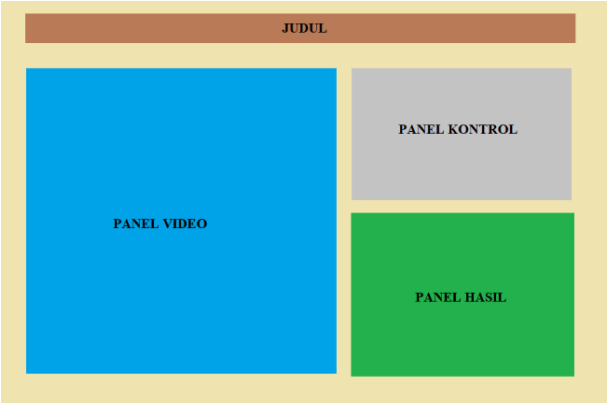
Dari ilustrasi Gambar 4.21 dapat dilihat bahwa terdapat gerakan pada waktu 00:00:50 sampai 00:00:55 dan pada waktu 00:01:16 sampai 00:01:22. Setiap interval waktu yang dihasilkan, selanjutnya dibuat sebuah objek video dengan label tertentu. Contoh : potongan1, potongan2, dan seterusnya. Interval waktu tersebut yang selanjutnya menjadi acuan untuk melakukan anotasi video.

e. Perancangan Proses Pemotongan dan Penyimpanan Video

Dari tahap deteksi objek bergerak, didapatkan kumpulan informasi video dimana telah terjadinya objek bergerak. Interval waktu tersebut menjadi acuan untuk proses pemotongan dan penyimpanan. Pada tahap ini, aplikasi pihak ketiga digunakan untuk membantu proses pemotongan. Ffmpeg digunakan untuk proses pemotongan video menjadi beberapa video berdasarkan interval waktu terjadinya gerakan.

4.2.3 Perancangan Antar Muka Sistem

Dalam mempermudah penelitian, dibutuhkan antar muka dari sistem yang telah dirancang. Desain antar muka dapat dilihat pada Gambar 4.22.



Gambar 4.22 Desain Antarmuka Sistem

4.3 Implementasi Sistem

Pada subbab ini dijelaskan mengenai implementasi kode program pada sistem yang telah dirancang sebelumnya.

4.3.1 Implementasi Tahap Input Data

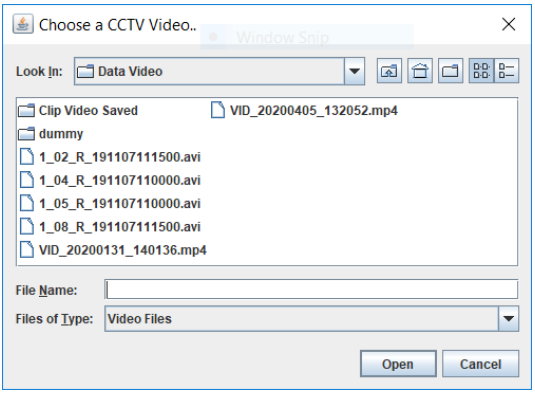
Proses input data berupa tampilan rekaman cctv. Pada proses ini ditentukan video yang nantinya ditampilkan di panel video dan digunakan sebagai data masukan untuk diproses. Penjabaran tentang proses pemilihan video adalah sebagai berikut :

Fungsi : menginputkan video bertipe *.mp4 atau *.avi pada antarmuka simulasi

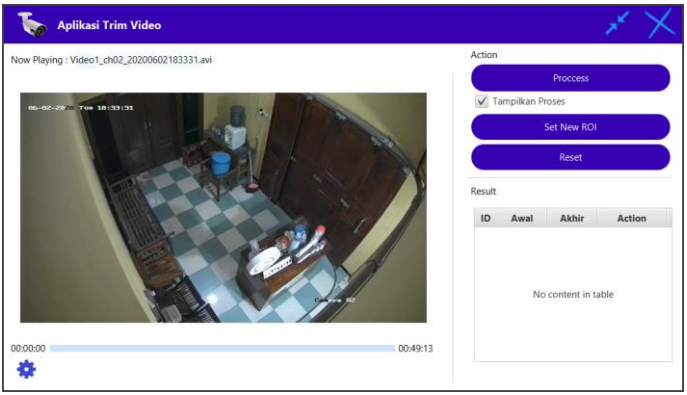
Input : video bertipe *.mp4 atau *.avi

Deskripsi : mengambil video bertipe *.mp4 atau *.avi yang telah disimpan pada komputer dan ditampilkan pada panel video di antarmuka simulasi.

Gambar 4.23 adalah tampilan antar muka pengambilan *input* video yang nantinya akan di-*load* pada tampilan antar muka pengujian yang ditunjukkan pada Gambar 4.24.



Gambar 4.23 Antarmuka *input* video



Gambar 4.24 Antarmuka setelah proses input data

Kode program implementasi tahap input data ini diimplementasikan menjadi sebuah program berikut :

```
JFileChooser fc = new JFileChooser("E:\\KULIAH\\Tugas Akhir\\Data Video\\");
fc.setDialogTitle("Choose a CCTV Video..");
fc.setFileSelectionMode(JFileChooser.FILES_ONLY);
FileNameExtensionFilter filter = new FileNameExtensionFilter("Video Files", "mp4", "avi");
fc.setFileFilter(filter);
fc.setMultiSelectionEnabled(false);
```

```
int returnValue = fc.showOpenDialog(new JFrame());
if(returnValue == JFileChooser.APPROVE_OPTION){
        tabelHasil.getItems().clear();
        videoPanel.setVisible(true);
        btnSetting.setDisable(false);
        path = fc.getSelectedFile().getAbsolutePath();
        nameFile = fc.getSelectedFile().getName();
        txtNowPlaying.setText("Now Playing : "+nameFile);
        . . .
}
```

4.3.2 Implementasi Tahap *Sampling Frame*

Pada tahap pemilihan *frame-frame* dari citra video, *frame – frame* dipilih dengan jarak sebesar konstanta tertentu. *Frame* dengan posisi kelipatan konstanta tertentu akan dipilih sebagai sampel citra. Kode program implementasi tahap *sampling frame* ini diimplementasikan menjadi sebuah program berikut :

```
while(runnable){
    if(video.grab()){
        if (postFrame % (konstantaLoncat) == 0) {
            try{
                video.retrieve(show);
                . . .
            }
        }
    }
}
```

4.3.3 Implementasi Tahap *Set Region of Interest* (ROI)

Penentuan ROI dilakukan secara dinamis dengan cara meng-*click* lalu men-*drag* kursor pada area citra yang diinginkan sebelum proses deteksi dilakukan. Kode program tahap penentuan ROI ini diimplementasikan menjadi sebuah program berikut :

```
private void setMouseListenerOnVideoPanel() {
    x = x2 = y = y2 = 0;
    videoPanel.setOnMousePressed(new EventHandler<MouseEvent>(){
      @Override
      public void handle(MouseEvent event) {
```

```
            if(!roiAda){
                . . .
            }
        }
    });
    videoPanel.setOnMouseDragged(new EventHandler<MouseEvent>(){
        @Override
        public void handle(MouseEvent event) {
            . . .
        }
    });
    videoPanel.setOnMouseReleased(new EventHandler<MouseEvent>(){
        @Override
        public void handle(MouseEvent event) {
            if(!roiAda){
                setEndPointROI((int) event.getX(), (int) event.getY());
                if((int)event.getX() > show.width()){
                    x2 = show.width() - 3;
                } else if((int)event.getX() <= 0){
                    x2 = 3;
                }
                if((int)event.getY() > show.height()){
                    y2 = show.height() - 3;
                } else if((int)event.getY() <= 0){
                    y2 = 3;
                }
                roi = new Rect(new Point(x,y), new Point(x2,y2));
            }
        }
    });
}
```

4.3.4 Implementasi Tahap *Pre-processing*

Pada tahap *pre-processing* dilakukan untuk mempersiapkan komponen-komponen yang digunakan sebelum diolah ke tahap selanjutnya. Proses pada tahap *Pre-processing* adalah :

1. *Grayscalling*

   Pengubahan citra sampel yang awal mulanya berbentuk citra RGB menjadi citra *grayscale*. Berikut ini adalah kode program untuk proses *grayscalling* :

```
Imgproc.cvtColor(frame1, frame1, Imgproc.COLOR_BGR2GRAY);
```

4.3.5 Implementasi Tahap Deteksi Objek Bergerak

Proses selanjutnya adalah tahap deteksi objek bergerak dengan *improved three frame difference algorithm*. Beberapa proses pada tahap deteksi objek bergerak adalah:

1. Pengambilan 3 sampel *frame* berurutan

   Tiga sampel berurutan dipilih untuk digunakan dalam algoritma *three frame difference*. Berikut ini adalah kode program proses pemilihan tiga *frame*:

   ```
   frame3 = show.submat(roi);
   ...
   frame2.copyTo(frame1);
   frame3.copyTo(frame2);
   ```

   *frame*3 merupakan citra yang sudah dibatasi roi, untuk proses saat ini. *frame*2 menggantikan nilai *frame*1 dan *frame*3 menggantikan nilai *frame*2 untuk digunakan pada iterasi proses berikutnya.

2. Pengurangan citra

   Pengurangan citra dilakukan untuk mencari perbedaan antara tiga sampel. Berikut kode program pengurangan citra :

   ```
   Core.absdiff(frame2, frame1, deltaFrame1);
   Core.absdiff(frame3, frame2, deltaFrame2);
   ```

   Dari proses pengurangan citra tiga sampel, didapatkan dua buah citra beda yaitu delta*Frame*1 dan delta*Frame*2.

3. Binerisasi citra

   Pada perancangan dijelaskan bahwa penggunaan *otsu thresholding* untuk proses binerisasi citra kurang cocok untuk diterapkan pada Tugas Akhir ini, sehingga pada permasalahan

Tugas Akhir ini digunakan basic *thresholding* dengan nilai ambang batas ditentukan sebelum proses deteksi dijalankan. Berikut kode program untuk proses binerisasi :

```
ambangBatas = (int) sliderTresh.getValue();
. . .
Imgproc.threshold(deltaFrame1, deltaFrame1, ambangBatas, 255,
Imgproc.THRESH_BINARY);
Imgproc.threshold(deltaFrame2, deltaFrame2, ambangBatas, 255,
Imgproc.THRESH_BINARY);
```

Selain itu, kode program perancangan untuk penentuan nilai *threshold* dengan metode *otsu thresholding* akan diberikan pada halaman Lampiran.

4. Operasi logika AND

   Proses operasi AND digunakan untuk mendapatkan citra irisan dari dua citra biner. Berikut kode program untuk operasi AND :

```
Core.bitwise_and(deltaFrame1, deltaFrame2, matY);
```

   Operasi AND dilakukan antara citra delta*Frame*1 dan citra delta*Frame*2 , matriks hasil operasi disimpan pad matY.

5. Modifikasi algoritma

   Modifikasi algoritma dilakukan untuk menutupi kekurangan algoritma, dimana hasil ekstraksi objek tidak lengkap. Langkah modifikasi dilakukan berdasarkan paper [10]. Kode program proses modifikasi algoritma diberikan di bawah ini :

```
Mat matM1 = new Mat();
Core.bitwise_xor(matY, deltaFrame2, matM1);
Mat matM2 = new Mat();
Core.bitwise_or(deltaFrame1, deltaFrame2, matM2);
Mat matZ = new Mat();
Core.bitwise_or(matM1, matM2, matZ);
```

6. Morfologi citra

   Proses morfologi *closing* dilakukan untuk menghilangkan *noise*, atau kontur yang tidak perlu hasil dari proses sebelumnya. Berikut ini adalah kode program untuk operasi morfologi closing :

```
Mat kernel = Imgproc.getStructuringElement(
Imgproc.MORPH_ELLIPSE, new Size(5,5));
Imgproc.morphologyEx(matZ, matZ, Imgproc.MORPH_CLOSE,
kernel);
Imgproc.morphologyEx(matZ, matZ, Imgproc.MORPH_OPEN,
kernel);
```

7. Segmentasi citra

   Segmentasi dilakukan untuk memisahkan latar objek dan latar *background*. Sehingga didapatkan area-area putih sebagai representasi daerah dimana objek telah bergerak. Berikut ini diberikan kode program untuk segmentasi citra :

```
cnts = new ArrayList<>();
Imgproc.findContours(matZ, cnts, new Mat(),
Imgproc.RETR_EXTERNAL, Imgproc.CHAIN_APPROX_SIMPLE);
```

8. Penentuan status gerak objek

   Ada atau tidaknya gerakan didasarkan pada besarnya area objek-objek yang telah tersegmentasi. Jika sedikitnya ada satu area yang melebihi luas minimal, maka status dikatakan ada

gerakan objek. Berikut ini kode program untuk proses penentuan status gerakan objek :

```
boolean motion = false;
for(int i=0; i < cnts.size(); i++) {
    double area = contourArea(cnts.get(i), false);
    if(area >= minArea) {
        motion = true;
        break;
    }
}
```

9. Anotasi Potongan Video

Status gerakan objek akan digunakan sebagai acuan dalam penyimpanan interval waktu. Jika objek pertama kali melakukan gerakan, maka disimpan sebagai waktu awal bergerak. Sedangkan jika gerakan objek dinyatakan telah selesai (sudah tidak ada gerakan) maka disimpan sebagai waktu akhir bergerak. Berikut kode program untuk proses penyimpanan interval objek bergerak :

```
if(motion){
    if(clip._isnull()){
        clip = new ClipVideo(id, (postFrame1/fps), (postFrame/fps));
    }
} else{
    if(!clip._isnull()){
        clip.setDetikAkhir(postFrame/fps);
        tabelHasil.getItems().add(clip);
        clip = new ClipVideo();
    }
}
```

4.3.6 Implementasi Tahap Pemotongan dan Penyimpanan

Pada tahap pemotongan dan penyimpanan, digunakan aplikasi *command line* pihak ketiga Ffmpeg untuk melakukan

pemotongan dan penyimpanan. Kode program untuk proses pemotongan dan penyimpanan video yang terpilih diberikan di bawah ini :

```
String ffmpegEXE = "C:\\ffmpeg\\bin\\ffmpeg.exe";
List<String> command = new ArrayList<>();
command.add(ffmpegEXE);
command.add("-i");
command.add(vidPathSrc);
command.add("-ss");
command.add(this.detikAwal);
command.add("-to");
command.add(this.detikAkhir);
command.add("-c");
command.add("copy");
command.add(vidPathDesc);

ProcessBuilder builder = new ProcessBuilder(command);
Process proses = builder.start();
```

Sumber video input yang didapat dari direktori ‘vidPathSrc’ pada komputer, akan dipotong pada interval this.detikAwal hingga this.detikAkhir. Hasil pemotongan lalu disimpan di direktori ‘vidPathDesc’ pada computer.

4.3.7 Implementasi Tahap Antarmuka

Pada penelitian ini, antarmuka sistem dibangun dengan menggunakan FXML yang terdapat pada JavaFX. Antarmuka-antarmuka yang diimplementasikan untuk menunjang penelitian Tugas Akhir ini adalah sebagai berikut :

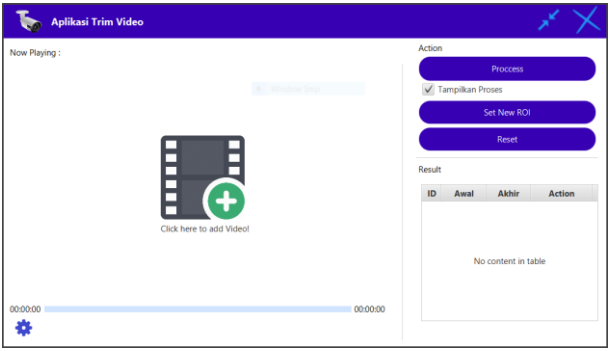
a. Antarmuka Halaman Utama

   Halaman utama merupakan antarmuka yang berisi informasi Tugas Akhir. Hasil implementasi halaman utama dapat dilihat pada Gambar 4.25.

b. Antarmuka Pengujian

Antarmuka pengujian digunakan untuk melakukan pengujian terhadap sistem. Hasil implementasi antarmuka pengujian dapat dilihat pada Gambar 4.26.

Gambar 4.25 Antarmuka halaman utama



Gambar 4. 26 Antarmuka pengujian

Pada antarmuka pengujian terdiri dari :

- *Button* **btnClose,** berfungsi untuk keluar dari program.
- *Button* **btnMinimize,** berfungsi untuk *minimize* tampilan *window* program.

- *Label* **nowPlaying**, berfungsi untuk menampilkan nama video yang diinput.
- *ImageView* **addMedia,** berfungsi untuk memilih video input yang ditampilkan dan digunakan untuk proses deteksi.
- *ImageView* **btnSetting,** berfungsi untuk melakukan pengaturan nilai ambang batas, minimal area, dan konstanta interval sampel *frame*, sebelum proses dilakukan.
- *Button* **btnProses,** berfungsi untuk melakukan proses deteksi objek bergerak pada video input yang sudah diberi ROI.
- *CheckBox* **tampilProses,** berfungsi untuk menampilkan proses yang terjadi selama deteksi objek bergerak.
- *Button* **btnNewROI,** berfungsi untuk mengatur ulang ROI sebelum proses dilakukan.
- *Button* **btnReset,** berfungsi untuk menghapus pekerjaan pada antarmuka pengujian secara keseluruhan.
- *TableView* **tabelHasil,** berfungsi untuk menampilkan hasil dari proses deteksi objek bergerak.
- *TableColumn* **idCol,** berfungsi untuk menampilkan id video hasil dari pemrosesan.
- *TableColumn* **awalCol,** berfungsi untuk menampilkan waktu awal video hasil dari pemrosesan.
- *TableColumn* **akhirCol,** berfungsi untuk menampilkan waktu akhir video hasil dari pemrosesan.
- *TableColumn* **actionCol** terdiri dari :
    - *Button* **play,** berfungsi untuk memutar video yang dipilih.
    - *Button* **save,** berfungsi untuk memilih lokasi untuk penyimpanan video yang dipilih.

BAB V
UJI COBA DAN PEMBAHASAN HASIL

Pada bab ini menjelaskan mengenai pengujian yang dilakukan terhadap program dan pembahasan dari hasil uji coba. Pengujian yang dilakukan adalah pengujian program dengan input video hasil dari menyalin rekaman cctv.

5.1 Data Uji Coba

Uji coba pada program dalam Tugas Akhir ini dilakukan terhadap video bertipe avi dan mp4. Video uji coba telah tersimpan dalam penyimpanan perangkat komputer dan diporeh dari hasil salinan rekaman cctv yang tersimpan pada *hardware* cctv, dan juga data uji yang diambil menggunakan kamera *handphone*. Video yang digunakan sebanyak 4 video. Daftar input uji coba tersebut disajikan dalam Tabel 5.1

Tabel 5. 1 Daftar data uji coba

No.	Nama	*Fps*	Tipe File	Durasi	Tempat
1	Video 1	12	.avi	00:09:15	Dalam ruangan, Malam Hari
2	Video 2	12	.avi	00:13:44	Dalam ruangan, Siang Hari
3	Video 3	12	.avi	00:09:50	Luar ruangan, Malam Hari
4	Video 4	25	.mp4	00:06:34	Luar ruangan, Siang Hari

5.2 Hasil Uji Coba

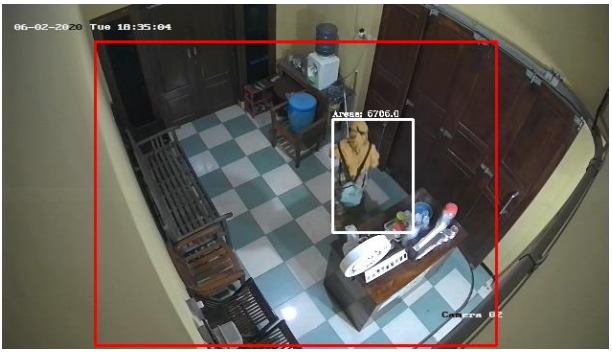
Berikut adalah hasil uji coba yang dilakukan pada tiga video. Pada masing-masing video dilakukan dua pengujian yaitu deteksi objek bergerak menggunakan *improved three frame difference algorithm* dan penyimpanan potongan video yang berisi rekaman objek bergerak.

5.2.1 Uji Coba Video 1

Video rekaman cctv ini diperoleh dari cctv yang terpasang di sudut ruangan, pada kondisi malam hari dengan penerangan lampu yang cukup.

1. Hasil deteksi objek bergerak

   Pada Gambar 5.1 menunjukkan hasil dari salah satu *frame* pada video 1 setelah dilakukan proses deteksi objek bergerak dan proses penentuan status gerakan. Terlihat bahwa jika sedikitnya satu area objek bergerak melebihi luas minimal maka dikatakan telah terjadi gerakan. Selanjutnya data interval waktu akan disimpan, yang ditunjukkan pada Gambar 5.2.



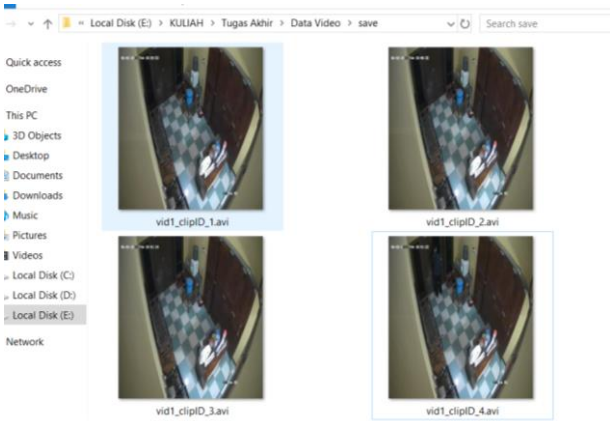
Gambar 5. 1 Hasil deteksi objek bergerak pada video 1

ID	Awal	Akhir	Action
1	00:00:26	00:00:45	
2	00:06:05	00:06:10	
3	00:06:47	00:06:55	
4	00:06:55	00:06:59	

Gambar 5. 2 Interval waktu gerakan pada video 1

2. Hasil penyimpanan potongan video

   Pada Gambar 5.3 menunjukkan hasil penyimpanan potongan video yang berisi rekaman aktifisan yang terjadi pada beberapa interval waktu, hasil dari proses deteksi objek bergerak.



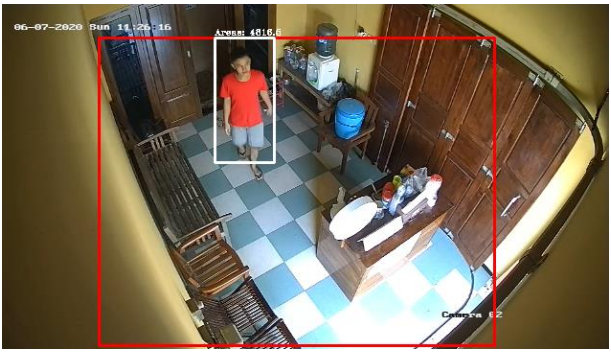
Gambar 5. 3 Hasil penyimpanan video gerak dari video 1

5.2.2 Uji Coba Video 2

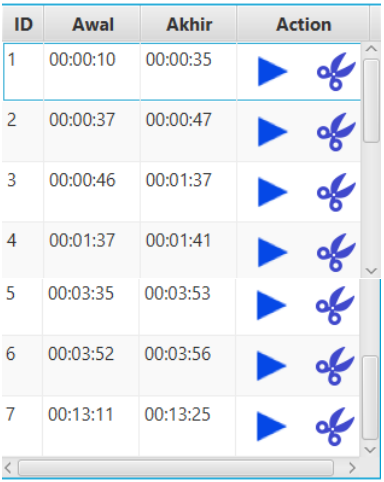
Video rekaman cctv ini diperoleh dari cctv yang terpasang di sudut ruangan, pada kondisi siang hari.

1. Hasil deteksi objek bergerak

   Pada Gambar 5.4 menunjukkan hasil dari salah satu *frame* pada video 2 setelah dilakukan proses deteksi objek bergerak dan proses penentuan status gerakan. Terlihat bahwa jika sedikitnya satu area gerak objek melebihi luas minimal maka dikatakan telah terjadi gerakan. Selanjutnya data interval waktu akan disimpan, yang ditunjukkan pada Gambar 5.5.



Gambar 5. 4 Hasil deteksi objek bergerak pada video 2



ID	Awal	Akhir	Action
1	00:00:10	00:00:35	
2	00:00:37	00:00:47	
3	00:00:46	00:01:37	
4	00:01:37	00:01:41	
5	00:03:35	00:03:53	
6	00:03:52	00:03:56	
7	00:13:11	00:13:25	

Gambar 5. 5 Interval waktu gerakan pada video 2

2. Hasil penyimpanan potongan video

   Pada Gambar 5.6 menunjukkan hasil penyimpanan potongan video yang berisi rekaman aktifisan yang terjadi pada beberapa interval waktu, hasil dari proses deteksi objek bergerak.

Gambar 5. 6 Hasil penyimpanan video gerak dari video 2

5.2.3 Uji Coba Video 3

Video rekaman cctv ini diperoleh dari cctv yang terpasang di sudut depan rumah, pada malam hari dengan penerangan lampu yang redup. Sebelumnya, data cctv bertipe avi dirubah ke dalam tipe mp4. Perubahan tipe video dilakukan untuk menguji fungsionalitas sistem dalam mengolah data video bertipe mp4.

1. Hasil deteksi objek bergerak

   Pada Gambar 5.7 menunjukkan hasil dari salah satu *frame* pada video 3 setelah dilakukan proses deteksi objek bergerak dan proses penentuan status gerakan. Terlihat bahwa jika sedikitnya satu area objek bergerak melebihi luas minimal maka dikatakan telah terjadi gerakan. Selanjutnya data interval waktu akan disimpan, yang ditunjukkan pada Gambar 5.8. Perlu tekankan, pada kasus data uji ini, kendaraan yang melajur di jalan raya tidak menjadi objek pengawasan.

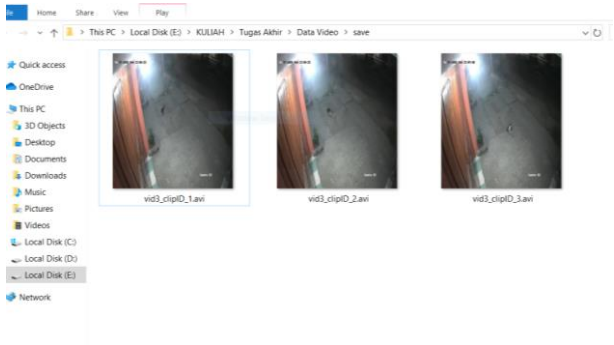
Gambar 5. 7 Hasil deteksi objek bergerak pada video 3

ID	Awal	Akhir	Action
1	00:00:00	00:00:03	
2	00:00:02	00:00:12	
3	00:09:25	00:09:35	

Gambar 5. 8 Interval waktu gerakan pada video 3

2. Hasil penyimpanan potongan video

   Pada Gambar 5.9 menunjukkan hasil penyimpanan potongan video yang berisi rekaman aktifisan yang terjadi pada beberapa interval waktu, hasil dari proses deteksi objek bergerak.



Gambar 5. 9 Hasil penyimpanan video gerak dari video 3

5.2.4 Uji Coba Video 4

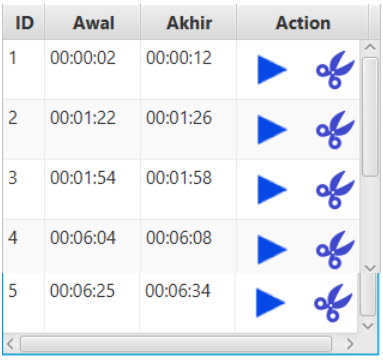
Video rekaman ini diambil dengan kamera *handphone* yang diletakkan diatas sudut area parker rumah, pada siang hari.

1. Hasil deteksi objek bergerak

   Pada Gambar 5.10 menunjukkan hasil dari salah satu *frame* pada video 4 setelah dilakukan proses deteksi objek bergerak dan proses penentuan status gerakan. Terlihat bahwa jika sedikitnya satu area objek bergerak melebihi luas minimal

maka dikatakan telah terjadi gerakan. Selanjutnya data interval waktu akan disimpan, yang ditunjukkan pada Gambar 5.11.

Gambar 5. 10 Hasil deteksi objek bergerak pada video 4

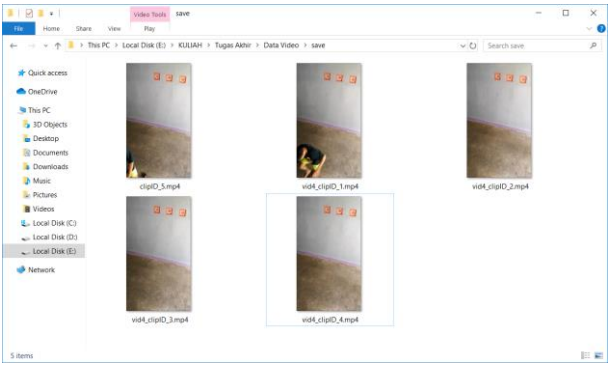


ID	Awal	Akhir	Action
1	00:00:02	00:00:12	
2	00:01:22	00:01:26	
3	00:01:54	00:01:58	
4	00:06:04	00:06:08	
5	00:06:25	00:06:34	

Gambar 5. 11 Interval waktu gerakan pada video 4

2. Hasil penyimpanan potongan video

   Pada Gambar 5.12 menunjukkan hasil penyimpanan potongan video yang berisi rekaman aktifisan yang terjadi pada beberapa interval waktu, hasil dari proses deteksi objek bergerak.



Gambar 5. 12 Hasil penyimpanan video gerak dari video 4

5.3 Pembahasan Hasil Uji Coba

Uji coba dilakukan pada data uji. Dalam uji coba, diketahui bahwa segmentasi video didasarkan pada interval waktu hasil dari pendeteksian objek bergerak. Sehingga banyaknya *frame* pada interval waktu objek bergerak yang harusnya terdeteksi adalah sebanyak jumlah *frame* yang diharapkan, yaitu di interval waktu video sebenarnya terjadinya aktifitas objek. Pengamatan video dilakukan dengan cara mendata waktu terjadinya gerakan untuk digunakan sebagai pembanding hasil dari sistem.

Untuk mengukur kinerja dari sistem yang digunakan pada aplikasi, dilakukan perhitungan *precision rate*, *recall rate*, *dan F1 Score* yang merujuk pada jurnal [19][20]. Persamaan ditunjukan oleh Persamaan (5.1), (5,2) dan (5,3).

$$Precision\ rate = \frac{TP}{TP+FP}\ x\ 100\% \tag{5.1}$$

$$Recall\ rate = \frac{TP}{TP+FN}\ x\ 100\% \tag{5.2}$$

$$F1\ Score = 2\ x \frac{precision\ rate\ x\ recall\ rate}{precision\ rate + recall\ rate} \tag{5.3}$$

Keterangan :

TP : Jumlah *frame* objek bergerak terdeteksi bergerak

FN : Jumlah *frame* objek bergerak terdeteksi tidak bergerak

FP : Jumlah *frame* objek tidak bergerak terdeteksi bergerak

TN : Jumlah *frame* objek tidak bergerak terdeteksi tidak bergerak

Recall rate merupakan tingkat keberhasilam sistem dalam menemukan kembali sebuah informasi. Dalam hal ini maka *recall* merupakan tingkat keberhasilan sistem dalam mendeteksi adanya objek bergerak. *Precision rate* merupakan tingkat ketepatan antara informasi yang diminta oleh pengguna dengan jawaban yang diberikan oleh sistem. Dalam hal ini maka *precision* adalah tingkat ketepatan interval waktu objek bergerak yang diharapkan sebenarnya. *F1 Score* menjadi acuan untuk mengetahui kinerja dari algoritma yang telah digunakan.

5.3.1 Pemilihan Nilai Parameter

Nilai parameter yang digunakan pada penelitian ini berorientasi pada hasil deteksi objek bergerak yang didapatkan dengan tujuan untuk mendapatkan hasil deteksi objek bergerak yang optimal. Hasil yang optimal dalam penelitian ini adalah interval waktu terdeteksi mendekati interval waktu sebenarnya objek bergerak. Atau dapat dikatakan bahwa jumlah *frame* pada interval waktu terdeteksi oleh sistem mendekati atau sama dengan jumlah *frame* pada interval waktu objek bergerak sesungguhnya. Rangkaian proses pemilihan parameter pada proses deteksi objek bergerak disajikan pada Tabel 5.2.

Tabel 5. 2 Detail rangkaian parameter proses deteksi

No	Proses	Parameter	Keterangan
1	*Binary Thresholding*	*Threshold*	Nilai *Threshold* dipengaruhi oleh intensitas cahaya. Threshold pada kondisi siang secara umum lebih besar dari *threshold* malam. Perlu dilakukan kalibrasi agar sistem dapat bekerja optimal baik kasus video siang ataupun video malam.

2	Morfologi	*Size kernel strel*	*Size kernel strel* morfologi mempengaruhi hasil dari segmentasi area target. Perlu dilakukan kalibrasi agar luas kontur objek yang tersegmentasi menjadi lebih tepat atau optimal.
3	Filter area	Luas minimum	Luas minimum perlu dilakukan kalibrasi agar gerakan oleh objek yang tidak diharapkan tidak dikatakan sebagai gerakan. Contoh gerak cicak, semut, daun, ataupun yang lain.

Selain pemilihan parameter untuk proses deteksi, pemilihan parameter juga dilakukan untuk pemilihan konstanta selang *frame* untuk mengurangi beban proses komputasi dan mengurangi resiko gerakan objek yang cenderung sangat pelan.

1. Parameter Konstanta Selang *Frame*.

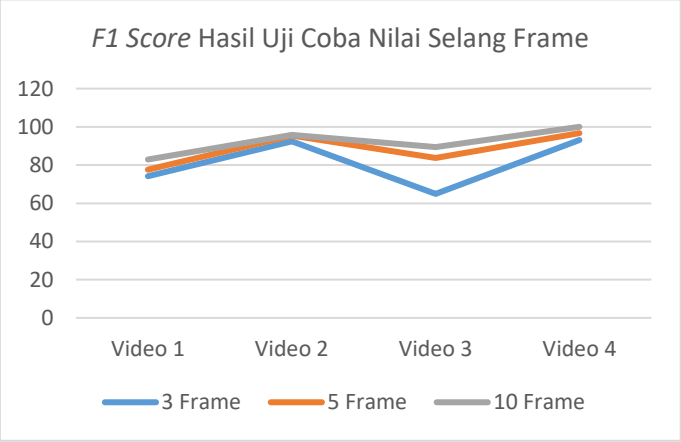
   Parameter selang *frame* digunakan untuk proses *sampling frame* yang bertujuan untuk mengurangi beban proses komputasi dan mengurangi resiko gerakan objek yang cenderung sangat pelan. Hasil analisa penentuan nilai selang *frame* terhadap hasil proses deteksi objek bergerak yang didapatkan disajikan pada Tabel 5.3.

Tabel 5. 3 Tabel Uji coba nilai selang *frame*

Selang *Frame*	Data Uji	TP	FN	FP	TN	R	P	F1
3	Vid1	276	132	60	6196	67,6%	82,1%	74,2%
	Vid2	1176	192	0	8522	86%	100%	92,5%
	Vid3	144	156	0	6780	48%	100%	64,9%
	Vid4	675	100	0	9099	87,1%	100%	93,1%

5	Vid1	312	96	84	6172	76,5%	78,8%	77,6%
	Vid2	1272	96	24	8498	93%	98,1%	95,5%
	Vid3	216	84	0	6780	72%	100%	83,7%
	Vid4	725	50	0	9099	93,5%	100%	96,7%
10	Vid1	348	60	84	6172	85,3%	80,6%	82,9%
	Vid2	1368	0	120	8402	100%	91,9%	95,8%
	Vid3	252	48	12	6768	84%	95,5%	89,4%
	Vid4	775	0	0	9099	100%	100%	100%

Keterangan :
R : *Recall rate*
P : *Precision rate*
F1 : *F1 Score*



Gambar 5. 13 Grafik *F1 Score* uji coba selang *frame*

Grafik *F1 Score* hasil uji coba selang *frame* disajikan pada Gambar 5.13. Dari data hasil uji coba, dipilih konstanta selang *frame* sebesar 10 *frame* sebagai nilai *default* sistem. dengan rataan nilai *precision rate* sebesar 92%; recall rate sebesar 92,3% dan *F1 Score* sebesar 92%.

2. Parameter *Threshold*

   Parameter threshold digunakan untuk proses binerisasi citra yang bertujuan untuk memisahkan antara *background* dengan *foreground*. Nilai *threshold* dipengaruhi oleh kondisi cahaya di area pengawasan. Hasil analisa penentuan nilai *threshold* terhadap hasil proses deteksi objek bergerak yang didapatkan disajikan pada Tabel 5.4.

Tabel 5. 4 Tabel uji coba nilai *threshold*

Threshold	Data Uji	TP	FN	FP	TN	R	P	F1
30	Vid1	348	60	84	6172	85,3%	80,6%	82,9%
	Vid2	1368	0	120	8402	100%	91,9%	95,8%
	Vid3	252	48	12	6768	84%	95,5%	89,4%
	Vid4	31	0	0	9099	100%	100%	100%
60	Vid1	288	120	84	6172	70,6%	77,4%	73,8%
	Vid2	996	372	60	8462	72,8%	94,3%	82,2%
	Vid3	36	264	0	6780	12%	100%	21,4%
	Vid4	31	0	0	9099	100%	100%	100%
100	Vid1	180	228	60	6196	44,1%	75%	55,6%
	Vid2	720	648	36	8486	52,6%	95,2%	67,8%
	Vid3	0	300	0	6780	0%	0%	0%
	Vid4	31	0	0	9099	100%	100%	100%

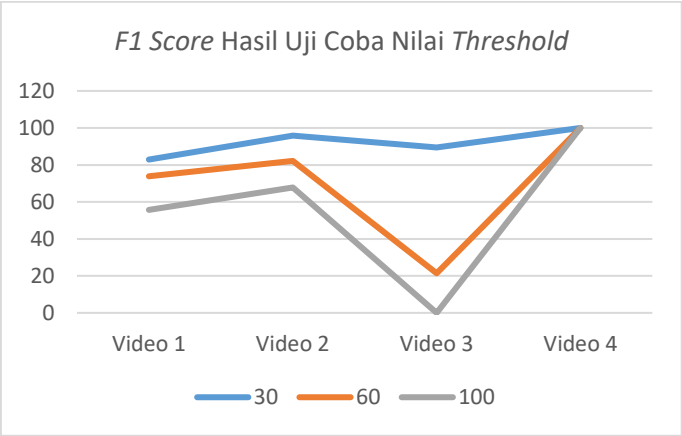
Keterangan :

R : *Recall rate*

P : *Precision rate*

F1 : *F1 Score*

Grafik *F1 Score* hasil uji coba nilai *threshold* disajikan pada Gambar 5.14. Dari data hasil uji coba dipilih *threshold* sebesar 30 sebagai nilai *default* sistem, dengan rataan nilai *precision rate* sebesar 92%; recall rate sebesar 92,3% dan rataan *F1 score* sebesar 92%



Gambar 5. 14 Grafik *F1 Score* uji coba nilai *threshold*

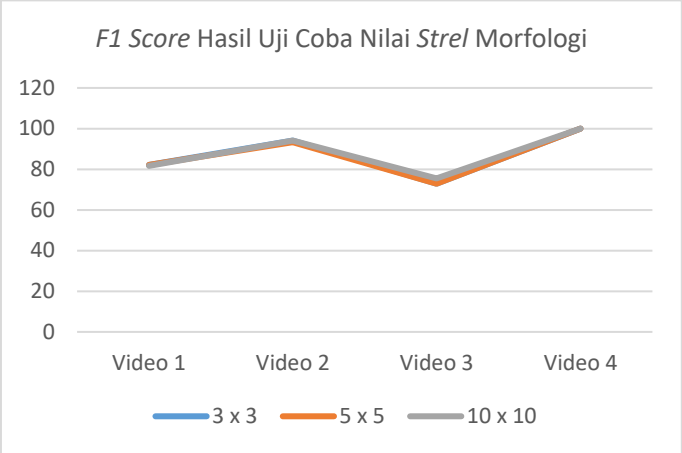
3. Parameter *Size Kernel Strel* (*Struktur Element*)
   Parameter *size strel* digunakan untuk proses morfologi citra yang bertujuan untuk menghilangkan *noise* serta mengisi lubang pada citra dengan tujuan untuk hasil proses segmentasi yang lebih baik. Hasil analisa penentuan nilai *size strel* terhadap hasil proses deteksi objek bergerak yang didapatkan disajikan pada Tabel 5.5.

Tabel 5. 5 Tabel Uji coba *size strel morfologi*

Size Strel	Data Uji	TP	FN	FP	TN	R	P	F1
3x3	Vid1	360	48	108	6148	88,2%	76,9%	82,2%
	Vid2	1368	0	120	8402	100%	91,9%	94,1%
	Vid3	216	60	12	6792	78,3%	68,2%	72,9%
	Vid4	775	0	0	9099	100%	100%	100%
5x5	Vid1	360	48	108	6148	88,2%	76,9%	82,2%
	Vid2	1368	0	120	8402	100%	91,9%	93,3%
	Vid3	216	60	12	6792	78,3%	68,2%	72,9%
	Vid4	775	0	0	9099	100%	100%	100%

10x10	Vid1	348	72	84	6160	82,9%	80,6%	81,7%
	Vid2	1368	0	120	8402	100%	91,9%	94,1%
	Vid3	228	48	12	6792	82,6%	69,6%	75,5%
	Vid4	775	0	0	9099	100%	100%	100%

Keterangan :
R : *Recall rate*
P : *Precision rate*
F1 : *F1 score*



Gambar 5. 15 Grafik *F1 Score* uji coba nilai *strel*

Grafik *F1 Score* hasil uji coba nilai *strel* disajikan pada Gambar 5.15. Dari data hasil uji coba dipilih *size strel* ukuran 10x10 dengan rataan nilai *precision rate* sebesar 85,3%; *recall rate* sebesar 94,5% dan rataan *F1 score* sebesar 89,5%.

4. Parameter Luas Minimum

Parameter luas minimun digunakan untuk proses penentuan status gerakan yang bertujuan untuk mendapatkan status adanya gerak atau tidak objek pada *frame* tersebut. Hasil Analisa penentuan nilai luas

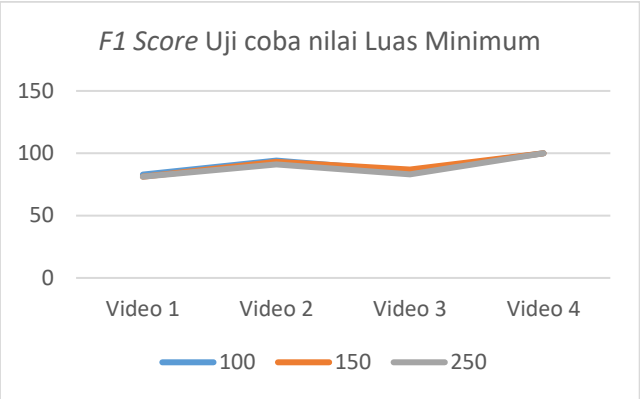
minimum terhadap hasil proses deteksi objek bergerak yang didapatkan disajikan pada Tabel 5.6.

Tabel 5. 6 Tabel Uji coba nilai luas minimum

Luas	Data Uji	TP	FN	FP	TN	R	P	F1
100	Vid1	29	5	7	6172	85,3%	80,6%	82,9%
	Vid2	1368	0	120	8402	100%	88,9%	94,1%
	Vid3	252	48	12	6768	84%	95,5%	89,4%
	Vid4	775	0	0	9099	100%	100%	100%
150	Vid1	28	6	7	6172	82,4%	80%	81,2%
	Vid2	1308	0	96	8426	100%	87,3%	93,2%
	Vid3	240	60	12	6768	80%	95,2%	87%
	Vid4	775	0	0	9099	100%	100%	100%
250	Vid1	28	6	7	6172	82,4%	80%	81,2%
	Vid2	1308	60	96	8426	95,6%	86,8%	91%
	Vid3	240	60	12	6768	80%	95,2%	87%
	Vid4	775	0	0	9099	100%	100%	100%

Keterangan :
R : *Recall rate*
P : *Precision rate*
F1 : *F1 score*

Grafik *F1 Score* hasil uji coba nilai luas minimum disajikan pada Gambar 5.16. Dari data hasil uji coba dipilih luas minimum sebesar 100 sebagai nilai *default* sistem, dengan rataan nilai *precision rate* sebesar 91,2%; *recall rate* sebesar 92,3% dan rataan *F1 score* sebesar 91,6%.



Gambar 5. 16 Grafik *F1 Score* uji coba nilai luas minimum

5.3.2 Analisa Hasil Uji coba

Uji coba dilakukan terhadap data uji untuk mengetahui kinerja dari *improved three frame difference algorithm*. Pada Tabel 5.7 diberikan perbandingan data hasil uji coba pada keempat video dengan algoritma *three frame differe* (tanpa modifikasi) dan dengan *improved three frame difference algorithm* (dengan modifikasi). Data pada tabel didapat dari hasil menghitung secara manual dari video asli dan hasil yang didapat dari sistem yang cocok.

Tabel 5. 7 Tabel Hasil Uji coba Dengan Modifikasi

	Data Uji	TP	FN	FP	TN	R	P	F1
Tanpa modifikas	Vid1	252	156	84	6172	61,8%	75%	67,7%
	Vid2	828	540	36	9398	60,5%	95,8%	74,2%
	Vid3	84	216	0	6588	28%	100%	43,8%
	Vid4	750	25	0	9382	96,8%	100 %	98,4%
Dengan modifikasi	Vid1	348	60	84	6172	85,3%	80,6%	82,9%
	Vid2	1368	0	120	8402	100%	91,9%	95,8%
	Vid3	252	48	12	6768	84%	95,5%	89,4%
	Vid4	775	0	0	9099	100%	100%	100%

Keterangan :

JF : jumlah *frame* terdeteksi bergerak oleh sistem

R : *Recall rate*

P : *Precision rate*

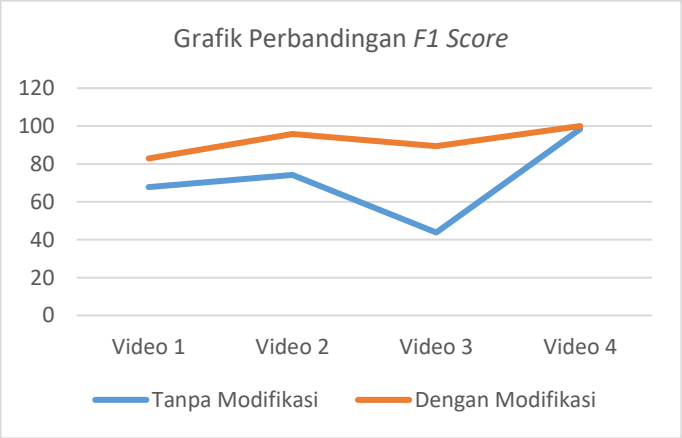
F1 : *F1 score*

Berikut contoh perhitungan untuk data pada tabel 5.7, pada uji coba data video 3 dengan menggunakan *improved three frame difference algorithm*:

$$R = \frac{TP}{TP+FN} \; x \; 100\% = \frac{252}{252+48} \; x \; 100\% = 84\%$$

$$P = \frac{TP}{TP+FP} \; x \; 100\% = \frac{252}{252+12} \; x \; 100\% = 95{,}5\%$$

$$F1\ Score = 2 \; x \; \frac{P \; x \; R}{P + R} = 2 \; x \; \frac{95{,}5\% \; x \; 84\%}{95{,}5\% + 84\%} = 89{,}4\%$$



Gambar 5. 17 Grafik *F1 Score* hasil deteksi

Grafik perbandingan *F1 Score* hasil uji coba dengan *improved three frame difference algorithm* dan algoritma *three frame difference* disajikan pada Gambar 5.17. Dari hasil uji coba di atas, pada data uji 4 didapat nilai tertinggi dibanding dengan 3 data uji yang lain. Hal ini disebabkan karena video diambil pada area luar

ruangan siang hari dengan latar *background* statis yang homogen yaitu dinding rumah.
Nilai *recall rate* terendah pada data uji ke 3 disebabkan karena pada data uji 3 target objek yaitu seekor kucing berwarna hitam, berjalan melalui area yang sedikit lebih gelap daripada area yang tersorot lampu sehingga proses deteksi kurang optimal.
Ketika terdapat suatu objek yang bergerak kemudian diam di area pengawasan selama rentang waktu tertentu, kemudian bergerak lagi, akan mengakibatkan sistem mendeteksi beberapa kegiatan. Sehingga mengakibatkan adanya beberapa potongan video ketika sebuah objek seringkali bergerak lalu diam dan bergerak lagi. Selain itu, nilai *fps* dan kecepatan objek juga mempengaruhi hasil. Ketika nilai fps sangat kecil sedangkan objek bergerak dengan cepat, akan memungkinkan sebuah objek yang melintasi area pengawasan menjadi tidak teramati oleh sistem.

Tabel 5. 8 Selisih hasil perbandingan

Data Uji	$\Delta Recall\ Rate$	$\Delta Precision\ Rate$	$\Delta F1\ Score$
Video 1	23,5%	5,6%	15,1%
Video 2	39,5%	-3,9%	21,6%
Video 3	56,0%	-4,5%	45,6%
Video 4	3,2%	0%	1,6%
Rata-Rata	30,6%	-0,7%	21%

Selisih hasil antara algoritma *three frame difference* dan *improved three frame difference algorithm* disajikan pada Tabel 5.8. Dengan menggunakan *improved three frame difference algorithm*, hasil meningkat cukup signifikan dibandingkan hanya menggunakan algoritma *three frame difference*. Nilai *F1 score* meningkat rata-rata sebesar 21%, dan tingkat kebenaran sistem dalam mendeteksi gerakan (*recall rate*) meningkat rata-rata sebesar 30,6%.
Selain itu, perhitungan *accuracy* juga digunakan untuk mengevaluasi kinerja sistem. Rumus *accuracy* diberikan pada

Persamaan 5.4, dengan data pada Tabel 5.7, didapat hasil perhitungan *accuracy* yang diberikan pada Tabel 5.9.

$$Accuracy = \frac{TP+TN}{TP+FP+FN+TN}\ x\ 100\% \quad (5.4)$$

Tabel 5. 9 Data perhitungan *accuracy*

	Data Video	*Accuracy*
Tanpa Modifikasi	Video 1	96,4%
	Video 2	94,7%
	Video 3	96,9%
	Video 4	99,8%
Dengan Modifikasi	Video 1	97,8%
	Video 2	98,8%
	Video 3	99,2%
	Video 4	100%

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, maka dapat diambil kesimpulan sebagai berikut :

1. Berhasil dikembangkan sistem untuk segmentasi video berbasis objek bergerak dengan tahapan-tahapan yaitu : sampling *frame*, penentuan ROI, *grayscalling*, proses deteksi objek bergerak, dan segmentasi video. Proses deteksi objek bergerak meliputi tahap pengurangan citra, binerisasi, operasi AND, *improved algorithm*, morfologi citra, segmentasi citra, penentuan status gerak objek, dan yang penyimpanan interval waktu gerak.
2. Sistem berhasil melakukan segmentasi video menggunakan *improved three frame difference algorithm* dengan nilai parameter selang frame sebesar 10, *threshold* sebesar 30, *size strel* sebesar 10x10, dan luas minimum sebesar 100 pada sistem, didapatkan nilai *f1 score* terendah sebesar 82,9% dan nilai accuracy terendah sebesar 97,8%. Nilai *f1 score* meningkat rata-rata sebesar 21,7% dibandingkan dengan hasil segmentasi hanya menggunakan algoritma *three frame difference*.
3. Hasil deteksi bergantung pada pemilihan parameter yang tepat karena dipengaruhi beberapa hal, antara lain: perubahan intensitas cahaya ke kamera, bayangan objek, warna objek, dan tinggi pemasangan kamera. Selain itu, pemilihan area ROI juga mempengaruhi tingkat ketepatan interval waktu.

4. Metode Otsu *thresholding* kurang tepat untuk diimplementasikan pada rancangan proses binerisasi citra pada sistem penelitian ini, karena hasil thresholding yang kurang representatif ketika tidak ada objek yang bergerak.

6.2 Saran

Dengan melihat hasil yang dicapai pada penelitian ini, ada beberapa hal yang penulis sarankan untuk pengembangan penelitian selanjutnya:

1. Perlu adanya uji coba dengan metode lain untuk proses thresholding, dengan tujuan sistem dapat bekerja lebih dinamis baik di area gelap ataupun terang, serta lebih *automatically* oleh sitem dalam penentuan nilai *threshold*.
2. Diperlukan suatu pendekatan algoritma lain untuk dapat menentukan nilai luas minimum dan *strel morfologi* agar sesuai dengan kondisi gambar.
3. Perlu adanya penghilangan bayangan di area pengamatan agar hasil deteksi lebih optimal.

DAFTAR PUSTAKA

[1] Surjono, Herman Dwi. Drs.,M.Sc.,MT.,Ph.D. (1996). Eksperimen Pengiriman sinyal televisi dengan pemancar TV dan CCTV serta Pemanfataanya dalam Pendidikan

[2] Tabaos14.com 2020. "Pemerintah Tambah Kamera CCTV Cegah Kejahatan". https://tabaos14.com/baca/pemerintah-tambah-kamera-cctv-cegah-kejahatan.html. Posted : 24/01/2020 23:40 WIT. Diakses tanggal 27/04/2020

[3] Erika Lia. 2019. "Polres Majalengka Sebar CCTV di Titik Rawan Pelanggaran Lalu Lintas". https://www.ayobandung.com/read/2019/12/06/72338/polres-majalengka-sebar-cctv-di-titik-rawan-pelanggaran-lalin. Posted : 06/12/2019. Diakses tanggal 27/04/2020.

[4] Bisnis Online. 2014. "Apa Saja Fungsi dan Tujuan Pemasangan CCTV" https://medium.com/@jualcctvonline/apa-saja-fungsi-dan-tujuan-pemasangan-cctv-aa93ce0225fd. Posted : 28/05/2014. Diakses tanggal 27/04/2020.

[5] Mehran Yazdi, Thierry Bouwmans. New Trends on Moving Object Detection in Video Images Captured by a moving Camera: A Survey. Computer Science Review, Elsevier, 2018.

[6] Fadhli R., Fiky Y. S. , Junartho H. "Application of Motion Detection on Security Camera Using Sobel Algorithm" E-Proceeding of Engineering. ISSN 2355-9365, Volume 4, Number 2, August 2017, Department of Electro Telkom University.

[7] Neelam Patel, "Motion Detection based on Multi *Frame* Video under Surveillance System" in the International Journal of Emerging Technology and Advanced Engineering. ISSN 2250-2459, Volume 2, Issue 1, January 2012, Department of Computer Science Sri Ram Institute of Technology JABALPUR.

[8] M. Lei and J. Geng, "Fusion of Three Frame Difference Method and Background Difference Method to Achieve Infrared Human Target Detection," *2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Kunming, China, 2019, pp. 381-384. doi: 10.1109/ICCASIT48058.2019.8973202.

[9] Zhang Z., Zhang H., Zhang Z. (2020) Using Three *Frame* Difference Algorithm to Detect Moving Objects. In: Xu Z., Choo KK., Dehghantanha A., Parizi R., Hammoudeh M. (eds) Cyber Security Intelligence and Analytics. CSIA 2019. Advances in Intelligent Systems and Computing, vol 928. Springer, Cham

[10] Z. Xu, D. Zhang and L. Du, "Moving Object Detection Based on Improved Three Frame Difference and Background Subtraction," *2017 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII)*, Wuhan, 2017, pp. 79-82. doi: 10.1109/ICIICII.2017.79

[11] Widyawan, M. Ihzan Zul, and L. E. Nugroho, "*Adaptive Motion Detection Algorithm using Frame Differences and Dynamic Template Matching Method*," The 9th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI 2012), Department of Electrical Engineering and Information Technology, Gadjah Mada University.

[12] Gonzalez, R.C. dan Woods, R.E. 2002. “Digital Image Processing”. United States of America: Tom Robbins Publisher.

[13] Al Bovik. 2000. “Handbook of Image and Video Processing”. San Diego: Academic Press Publisher.

[14] 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, N o. 3

[15] Sutoyo, T. & Mulyanto, E. 2009. Teori Pengolahan Citra Digital. Andi: Yogyakarta.

[16] Basuki, A., & Palandi, J. F. F. (2005). Pengolahan Citra Digital Menggunakan Visual Basic. *Yogyakarta: Graha Ilmu*, *1*.

[17] Dr. Andrew Greensted. 2010. “Otsu Thresholding”. http://www.labbookpages.co.uk/software/imgProc/otsuThreshold.html, Terakhir diubah : 17/06/2010. Diakses tanggal 11/05/2020.

[18] Ilwaru, V. Y. I., et al. "Aplikasi Operasi Himpunan dan Matematika Morfologi pada Pengolahan Citra Digital." *Barekeng* 10.2 (2016): 83-96.

[19] Dian Rosdiana. “Tracking Objek Bergerak dengan Algoritma K-Means”. TEDC Vol. 11 No. 1, Januari 2017, Department of Design and Industrial Mechanical Engineering Politeknik TEDC Bandung.

[20] Muhamad, R., Yulianti, T., Purwiyanti, S., & Setyawan, F. (2019). Deteksi Objek Bergerak Pada Video Bawah Air Menggunakan Metode Frame Differencing. *Jurnal EECCIS*, *13*(2),100-104.
Dari https://jurnaleeccis.ub.ac.id/index.php/eeccis/article/view/598

Lampiran A : Kode Program Otsu Thresholding

```
private int getOtsuThresh(Mat gray){
        Mat coba = new Mat();
        gray.copyTo(coba);
        // prepare untuk proses menentukan nilai T dengan metode otsu
        MatOfByte temp = new MatOfByte();
        Imgcodecs.imencode(".bmp", coba, temp);

        BufferedImage grayImg = null;
        try {
            grayImg = ImageIO.read(new ByteArrayInputStream(temp.toArray()));
        } catch (IOException ex) {
            System.err.println(ex);
        }
        byte[] srcData = ((DataBufferByte)
grayImg.getRaster().getDataBuffer()).getData(0);
        //proses inti
        int[] histData = new int[256];
        //mencari level dengan nilai tertinggi
        for(int i = 0; i<srcData.length; i++){
            int levelGray = 0xFF & srcData[i];
            histData[levelGray] ++;
        }

        float sum = 0;
        for(int i = 0; i< 256; i++){
            sum += i * histData[i];
        }

        float sumB = 0;
        int wB = 0;
        int wF = 0;
        float varMax = 0;

        //mencari nilai Thresh
        int thresh = 0;
        for(int i = 0; i< 256; i++){
            wB += histData[i]; //weight background
            if(wB == 0){
                continue;
            }
```

```
        wF = srcData.length - wB;   //weight objek (foreground)
        if(wF == 0){
            break;
        }
        sumB += (float) (i * histData[i]);
        float mB = sumB / wB; //mean background
        float mF = (sum - sumB) / wF; //mean objek (foreground)

        //menghitung varians antar kelas
        float varians = (float)wB * (float)wF * (mB - mF) * (mB - mF);

        //mengudpate variansiTertinggi
        if(varians > varMax){
            varMax = varians;
            thresh = i;
        }
    }
    return thresh;
}
```

BIODATA PENULIS

Praditya, lahir pada tanggal 26 April 1998 di Kabupaten Rembang, Jawa Tengah. Memiliki nama lengkap Praditya Lutvi Charisma. Putra kedua dari pasangan Mundasir dan Rusmiati, S.Pd. Penulis menempuh pendidikan formal di SD Wijaya Kusuma, Lasem, Rembang, dan lulus pada tahun 2010. Selanjutnya meneruskan sekolah di SMPN 1 Lasem, Rembang, tamat pada tahun 2013, dan lulus pada tahun 2016 dari SMAN 1 Rembang. Setelah Lulus dari SMA Negeri 1 Rembang, penulis melanjutkan pendidikan tingginya di Departemen Matematika, Fakultas Sains dan Analitika Data, Institut Teknologi Sepuluh Nopember, Surabaya.

Selama kuliah, penulis juga aktif dalam kegiatan non akademik dengan menjadi Kepala Divisi Pengembangan Sumber Daya UKM Sepakbola ITS pada tahun 2018-2019, Koordinator *Equipment* Olimpiade Matematika ITS OMITS12TH 2018, Koordinator Instruktur Padamu HIMATIKA ITS 2018, dan Staff *Sport and Art Development* HIMATIKA ITS serta aktif dalam bermain futsal. Jika ingin memberi saran, kritik dan diskusi mengenai Tugas akhir ini, dapat dikirimkan melalui e-mail: lutvicharisma@gmail.com. Terimakasih.