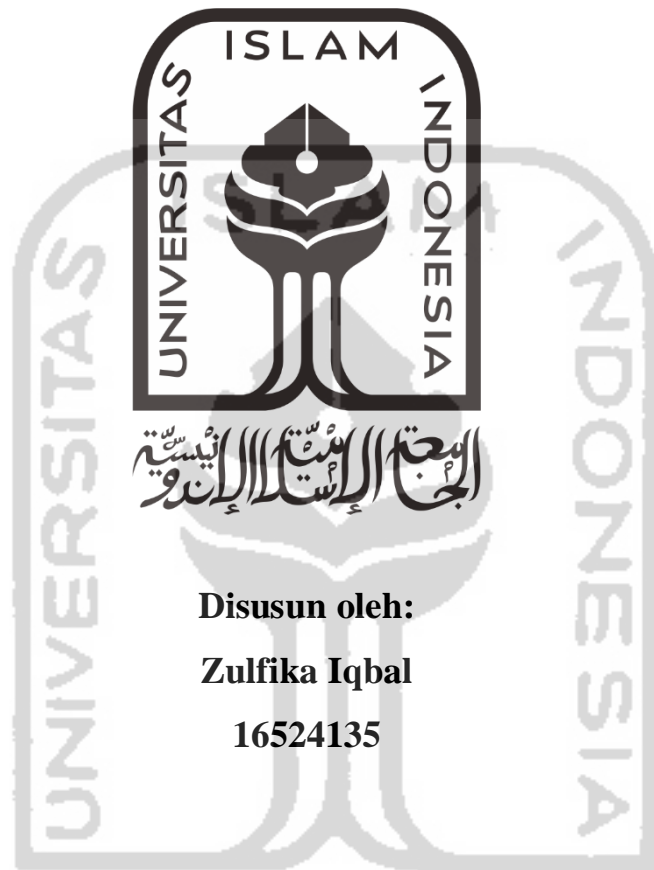


**DETEKSI ARAH PERGERAKAN MANUSIA BERBASIS VIDEO
OFFLINE DENGAN MENGGUNAKAN DETEKSI *OPTICAL FLOW*
SKRIPSI**

untuk memenuhi salah satu persyaratan
mencapai derajat Sarjana S1



**Disusun oleh:
Zulfika Iqbal
16524135**

**Jurusan Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia
Yogyakarta**

2020

LEMBAR PENGESAHAN

DETEKSI ARAH PERGERAKAN MANUSIA BERBASIS VIDEO OFFLINE DENGAN MENGGUNAKAN DETEKSI *OPTICAL FLOW*

TUGAS AKHIR

Diajukan sebagai Salah Satu Syarat untuk Memperoleh
Gelar Sarjana Teknik
pada Program Studi Teknik Elektro
Fakultas Teknologi Industri
Universitas Islam Indonesia

Disusun oleh:



Zulfika Iqbal
16524135

Yogyakarta, 30 Juli 2020

Menyetujui,

Pembimbing



Elvira Sukma Wahyuni, S.Pd., M.Eng
155231301

LEMBAR PENGESAHAN

SKRIPSI

DETEKSI ARAH PERGERAKAN MANUSIA BERBASIS VIDEO OFFLINE DENGAN MENGGUNAKAN DETEKSI *OPTICAL FLOW*

Dipersiapkan dan disusun oleh:

ISLAM

Zulfika Iqbal

16524135

Telah dipertahankan di depan dewan penguji

Pada tanggal: 10 Juli 2020

Susunan dewan penguji

Ketua Penguji : Elvira Sukma Wahyuni, S.Pd., M.Eng. 

Anggota Penguji 1: Firdaus, S.T., M.T., Ph.D. 

Anggota Penguji 2: Ida Nurcahyani, S.T., M.Eng. 

Skripsi ini telah diterima sebagai salah satu persyaratan
untuk memperoleh gelar Sarjana

Tanggal: 3 Agustus 2020

Ketua Program Studi Teknik Elektro



Yusuf Aziz Amrullah, S.T., M. Eng., Ph. D.

045240101

PERNYATAAN

Dengan ini Saya menyatakan bahwa:

1. Skripsi ini tidak mengandung karya yang diajukan untuk memperoleh gelar kesarjanaan di suatu Perguruan Tinggi, dan sepanjang pengetahuan Saya juga tidak mengandung karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis diacu dalam naskah ini dan disebutkan dalam daftar pustaka.
2. Informasi dan materi Skripsi yang terkait hak milik, hak intelektual, dan paten merupakan milik bersama antara tiga pihak yaitu penulis, dosen pembimbing, dan Universitas Islam Indonesia. Dalam hal penggunaan informasi dan materi Skripsi terkait paten maka akan diskusikan lebih lanjut untuk mendapatkan persetujuan dari ketiga pihak tersebut diatas.

Yogyakarta, 8 April 2020



Zulfika Iqbal

KATA PENGANTAR

Bismillahirrahmanirrahim

Assalamu 'alaikum Warahmatullahi Wabarakatuh

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang telah memberikan rahmat dan karunianya, sehingga penulis dapat menyelesaikan Laporan Tugas Akhir yang berjudul DETEKSI ARAH PERGERAKAN MANUSIA BERBASIS VIDEO OFFLINE DENGAN MENGGUNAKAN DETEKSI *OPTICAL FLOW*. Tidak lupa juga shalawat serta salam kita panjatkan kepada Nabi Muhammad SAW yang telah membimbing umatnya menjadi pribadi yang lebih baik.

Atas petunjuk dan ridho-Nya jualah Laporan Tugas Akhir ini dapat diselesaikan dengan baik dan lancar. Kelancaran dalam mempersiapkan dan menyelesaikan Laporan Tugas Akhir ini tidak terlepas dari bantuan berbagai pihak. Oleh karena itu dengan rasa hormat dan terima kasih yang sebesar-besarnya penulis haturkan kepada :

1. Ayahku Drs. Dewirman dan Ibuku Fitri Aplitawati S.Sos yang selalu memberikan doa, semangat, nasihat, serta dukungan selama proses penelitian dan laporan tugas akhir.
2. Kakakku Zulfamina Anugrah, S.Sc yang selalu mendukung dan memberikan bantuan dalam mengerjakan laporan tugas akhir ini.
3. Adik-adiku Ifan, Fani, dan Faira yang selalu menjadi motivasi penulis dalam mengerjakan penelitian dan laporan tugas akhir.
4. Semua keluargaku yang ada di Kepulauan Riau.
5. Ibu Elvira Sukma Wahyuni, S.Pd., M.Eng. selaku Dosen Pembimbing yang telah memberi bantuan dan pengarahan hingga terselesaikan laporan Tugas Akhir ini.
6. Bapak Yusuf Aziz Amrullah, S.T., M.Sc., Ph.D. selaku Ketua Jurusan Teknik Elektro, Universitas Islam Indonesia
7. Para dosen dan karyawan program studi Teknik Elektro Universitas Islam Indonesia.
8. Sahabat seperjuanganku Abid, Septian, Doge, Nizam yang selalu menjadi rekan terbaik selama masa perkuliahan.
9. Teman-teman Divisi lapangan slayer merah 2015, 2016, 2017 yang telah mengajarkan apa arti kebersamaan, pengorbanan, dan solidaritas.

10. Sahabat-sahabatku di Pohon Mangga club yang menjadi motivasi keras dalam mengerjakan penelitian dan laporan tugas akhir ini,
11. Sahabat- sahabat KKN unit 222 yang selalu menjadi salah satu sahabat terbaik dan motivator penulis ketika di semester tua.
12. Teman-teman mahasiswa Jurusan Teknik Elektro Universitas Islam Indonesia angkatan 2016.
13. Semua pihak lain yang tidak dapat disebutkan satu persatu yang telah memberikan masukan, dorongan dan semangat dalam menyelesaikan Laporan tugas akhir ini.

Wassalamu 'alaikum Warahmatullahi Wabarakatuh

Yogyakarta, 8 April 2020

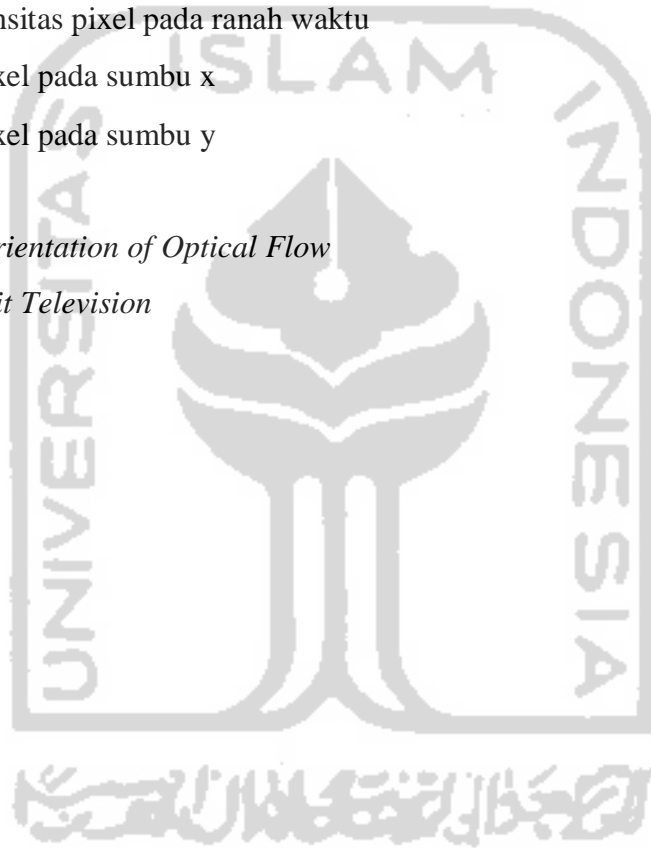


Zulfika Iqbal



ARTI LAMBANG DAN SINGKATAN

m	:Baris pada pixel
n	:Kolom pada pixel
θ	:Orientasi
$^{\circ}$:Derajat
$I_{(x,y,t)}$:Intensitas <i>Optical Flow</i>
I_x	:Gradient intensitas pixel pada sumbu x
I_y	:Gradient intensitas pixel pada sumbu y
I_t	:Gradient intensitas pixel pada ranah waktu
v_x	:Kecepatan pixel pada sumbu x
v_y	:Kecepatan pixel pada sumbu y
q_n	:Pixel ke-n
HOOF	: <i>Histogram Orientation of Optical Flow</i>
CCTV	: <i>Closed Circuit Television</i>



ABSTRAK

Pergerakan anak-anak secara aktif memiliki resiko keamanan ketika anak-anak tersebut bergerak ke tempat-tempat yang berbahaya ditambah dengan tanpa adanya pengawasan dari orang dewasa. Untuk mengurangi resiko kecelakaan pada anak-anak tanpa pengawasan orang tua dibutuhkan sebuah sistem deteksi pergerakan secara otomatis. Pada penelitian ini dilakukan deteksi arah pergerakan manusia menggunakan pengolahan citra dengan input video *offline* dari 4 kamera CCTV yang masing-masing kamera CCTV menghasilkan 4 video objek manusia bergerak ke arah depan, belakang, kanan dan kiri dengan menggunakan metode deteksi *Optical Flow*. *Optical Flow* akan menghitung nilai arah atau orientasi dari pergerakan sebuah objek. Arah pergerakan objek yang didapatkan akan dilakukan *preprocessing* untuk mempermudah perhitungan dan visualisasi. Orientasi akan diakumulasikan dengan metode klasifikasi orientasi HOOF yang akan mengumpulkan orientasi objek pada keseluruhan *frame* pada suatu inputan video sesuai dengan bagian-bagian sudut kartesius 8 bagian. Hasil dari orientasi dengan *Optical Flow* akan dibandingkan dengan arah deteksi yang diukur secara manual menggunakan diagram kartesius untuk mengetahui apakah deteksi arah pergerakan menggunakan *Optical Flow* berjalan dengan baik. Hasil dari penelitian ini adalah 16 video inputan dari kamera CCTV berhasil dideteksi dengan *Optical Flow* dengan keberhasilan pendeteksian 100%. Pada nilai akurasi yaitu membandingkan nilai deteksi arah *Optical Flow* dengan deteksi manual didapatkan akurasi sebesar 87,5%.

Kata Kunci : Pengolahan citra, Kamera CCTV, *Optical Flow*, HOOF.



DAFTAR ISI

LEMBAR PENGESAHAN	i
LEMBAR PENGESAHAN	ii
PERNYATAAN.....	iii
KATA PENGANTAR.....	iv
ARTI LAMBANG DAN SINGKATAN.....	vi
ABSTRAK.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	x
DAFTAR TABEL.....	xi
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	2
1.5 Manfaat Penelitian.....	2
BAB 2 TINJAUAN PUSTAKA.....	3
2.1 Studi Literatur	3
2.2 Tinjauan Teori.....	5
2.2.1 Representasi Citra Digital.....	5
2.2.2 Jenis Citra Digital.....	5
2.2.3 <i>Optical Flow</i>	6
2.2.4 Algoritma <i>Lucas-Kanade Optical FLOW</i>	8
BAB 3 METODOLOGI	11
3.1 Alur Penelitian.....	11
3.1.1 Pengambilan Data Video	11
3.1.2 Sistem Deteksi Arah Pergerakan Manusia Menggunakan <i>Optical FLOW</i>	12

3.1.3 Hasil dan Pembahasan.....	18
BAB 4 HASIL DAN PEMBAHASAN	21
4.1 Hasil pengujian sistem pendeteksian arah pergerakan menggunakan <i>Optical FLOW</i> ...	21
4.2 Hasil orientasi arah pergerakan dengan klasifikasi HOOFF dan akurasi deteksi arah menggunakan <i>Optical Flow</i>	22
4.3 Perbandingan dengan metode lain.....	25
BAB 5 KESIMPULAN DAN SARAN	27
5.1 Kesimpulan	27
5.2 Saran	27
DAFTAR PUSTAKA.....	28
LAMPIRAN.....	29



DAFTAR GAMBAR

Gambar 2. 1 Representasi citra digital dalam 2 dimensi [7]	5
Gambar 2. 2 <i>Pyramida Lucas-kanade Optical FLOW</i> [10]	9
Gambar 2. 3 Vektor pergerakan hasil pendeteksian <i>Optical Flow</i>	9
Gambar 2. 5 Contoh Orientasi pada sebuah <i>frame</i> [11]	10
Gambar 3. 1 Diagram Blok Alur Penelitian.....	11
Gambar 3. 2 Data video yang telah diambil pada setiap kamera CCTV	12
Gambar 3. 3 Diagram Alir Deteksi Arah Pergerakan Objek	12
Gambar 3. 4 Nilai <i>Greyscale</i> pada <i>frame</i> terakhir	13
Gambar 3. 5 Hasil Orientasi <i>Optical Flow</i>	14
Gambar 3. 6 Deteksi dan Tracking pergerakan manusia dengan GMM dan <i>kalman filter</i>	15
Gambar 3. 7 Orientasi pada objek manusia	15
Gambar 3. 8 Orientasi pada objek manusia yang ditelaah dikodekan	16
Gambar 3. 9 Bagian- bagian orientasi	16
Gambar 3. 10 Pengodean orientasi kedalam bentuk histogram HOOF.....	17
Gambar 3. 11 Jumlah orientasi terbanyak pada setiap bagian sudut	18
Gambar 3. 12 Visualisasi hasil perhitungan Orientasi objek	18
Gambar 3. 13 Histogram orientasi <i>Optical Flow</i>	19
Gambar 3. 14 Hasil pengukuran arah secara manual menggunakan diagram kartesius	19
Gambar 3. 15 Visualisasi deteksi arah menggunakan <i>Optical Flow</i>	20
Gambar 4. 1 Hasil pengujian deteksi arah menggunakan <i>Optical Flow</i> pada arah depan kamera 1	21
Gambar 4. 2 Hasil orientasi arah pergerakan <i>Optical Flow</i> dengan HOOF pada arah depan kamera 1.....	22
Gambar 4. 3 Hasil perhitungan secara manual	24

DAFTAR TABEL

Tabel 4. 1 Hasil Pengujian Pendeteksian Arah menggunakan <i>Optical FLOW</i>	21
Tabel 4. 2 Hasil Deteksi Arah menggunakan Optical Flow dalam bentuk Orientasi	23
Tabel 4. 3 Perbandingan deteksi arah <i>Optical Flow</i> dengan deteksi secara manual.....	24
Tabel 4. 4 Perbandingan metode yang berkaitan dengan penelitian	25



BAB 1

PENDAHULUAN

1.1 Latar Belakang Masalah

Populasi jumlah penduduk Indonesia pada tahun 2019 mencapai 266,91 juta jiwa. Jumlah penduduk usia anak-anak (0-14 tahun) menempati posisi kedua jumlah penduduk terbanyak setelah usia produktif (15-64 tahun) dengan populasi 66,17 juta jiwa atau sekitar 24,8 % dari total populasi [1]. Dengan jumlah penduduk usia anak yang tergolong banyak tersebut cukup menjadi perhatian. Sesuatu yang harus diperhatikan adalah gaya bermain usia anak-anak dimana pada usia ini mulai rasa ingin tahu terhadap sesuatu hal baik yang dilihat maupun yang didengar, rasa ingin tahu tersebut memacu anak-anak bergerak secara aktif [2].

Pergerakan anak-anak secara aktif memiliki resiko terhadap kemana anak-anak akan bergerak dan beraktifitas, ditambah dengan tanpa adanya pengawasan dari orang dewasa anak-anak akan bergerak ke arah yang berbahaya seperti bangunan berlantai yang bisa menyebabkan resiko terjatuh, bergerak ke arah sumber listrik yang bertegangan, bergerak ke arah jalan raya yang akan menyebabkan kecelakaan dan tempat-tempat yang seharusnya tidak dilalui oleh anak-anak. Upaya yang dilakukan adalah dengan memasang kamera CCTV (*Closed Circuit Television*) pada tempat-tempat yang sering ditempati oleh anak-anak seperti ruangan didalam rumah, teras rumah, halaman belakang rumah, ruang sekolah, halaman sekolah, dan taman bermain. Namun kamera CCTV hanya merekam kejadian saja tidak bisa menganalisis sesuatu pergerakan, maka perlu adanya sistem yang mengetahui arah pergerakan manusia melalui kamera CCTV secara otomatis.

Dalam penelitian ini dilakukan deteksi arah pergerakan manusia menggunakan video *offline* dari kamera CCTV, penggunaan kamera dalam penelitian ini dimanfaatkan dengan menambahkan teknologi *video processing* di dalamnya. Adapun salah satu parameter yang dapat dipantau melalui sistem deteksi pergerakan manusia adalah arah manusia bergerak. Arah manusia bergerak adalah hal yang bisa dilihat oleh mata manusia dan bisa diproses oleh otak manusia kemana manusia tersebut bergerak, tapi tidak dengan kamera CCTV yang hanya merekam kejadian saja tanpa mengetahui kemana arah objek dalam kejadian tersebut bergerak.

1.2 Rumusan Masalah

1. Bagaimana cara mendeteksi arah pergerakan manusia berbasis *video processing* menggunakan video rekaman CCTV?
2. Bagaimana unjuk kerja deteksi arah pergerakan manusia berbasis *video processing* menggunakan video rekaman CCTV?

1.3 Batasan Masalah

1. Kamera CCTV yang digunakan berjumlah 4 buah masing-masing berada pada setiap sudut ruangan.
2. Objek yang berada didalam ruangan berjumlah satu orang.
3. Sistem pencahayaan didalam ruangan percobaan harus memadai sehingga video yang akan diambil terdeteksi dengan jelas.
4. Video CCTV yang digunakan ialah secara *offline* (rekaman video).

1.4 Tujuan Penelitian

1. Menemukan metode yang dapat mendeteksi arah pergerakan manusia berbasis *Video Processing* berdasarkan hasil rekaman video CCTV.
2. Untuk mengetahui unjuk kerja metode deteksi arah pergerakan manusia berbasis *video processing*.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah sistem yang dianalisis dapat membantu dalam mencegah kondisi yang tidak diinginkan pada anak-anak ketika bergerak ke arah yang berbahaya seperti jalan raya, rumah berlantai lebih dari satu ketika tanpa pengawasan orang tua.

BAB 2

TINJAUAN PUSTAKA

2.1 Studi Literatur

Penelitian yang berkaitan secara tidak langsung tentang deteksi arah pergerakan manusia sudah ada dilakukan.

Seperti pada penelitian [3], penelitian ini mendeteksi arah pergerakan dengan objek kendaraan yang bergerak seperti mobil, menurut penelitian ini sebelum mencari arah pergerakan objek yang bergerak maka dibutuhkan metode pendeteksian objek bergerak terlebih dahulu yaitu memakai metode deteksi *You Only Look Once Version 3* (YOLOv3). Metode ini bekerja secara *real time* untuk mendeteksi dan *men-tracking* objek yang bergerak dengan algoritmanya berdasarkan *centroid tracking* dimana terdapat *multiple proses* yang ada proses deteksi utk mendeteksi objek dengan Ids atau nama objek dalam sistem, untuk *men-tracking bounding box* objek lebih dari satu membutuhkan metode *hungarian algorithm* dan *kalman filter* untuk terus mengupdate *bounding box* objek, setelah objek terdeteksi dan dilakukan tracking maka akan dicari arah pergerakannya, pada metode ini untuk mencari arah pergerakannya adalah dengan *mentraining pixel* yang pada suatu *frame*. Pixel-pixel tersebut diatur apakah pixel tersebut boleh dilewati oleh kendaraan atau tidak sehingga arah pergerakan yang dideteksi pada metode ini adalah dengan parameter arah pergerakan yang salah dan arah pergerakan yang benar dengan cara *mentraining pixel-pixel* pada sebuah *frame*. Maka didapatkan rata-rata akurasi adalah 91,98%. Akan tetapi penelitian ini terdapat beberapa metode yang kompleks dan untuk menyelesaikan beberapa fungsi maka harus digunakan metode- metode lain.

Penelitian berikutnya yang berhubungan secara langsung dengan deteksi arah pergerakan manusia adalah penelitian [4]. Pada penelitian ini untuk mendeteksi arah pergerakan objek dengan menggunakan metode deteksi *Optical Flow* khususnya *Optical Flow-Horn Schunk* yang tidak perlu lagi memakai tambahan metode deteksi objek dan metode training arah pergerakan karena *Optical Flow* sendiri adalah metode deteksi objek yang sekaligus bisa menjadi metode deteksi arah pergerakan yang bekerja dengan cara memanfaatkan prinsip pergerakan intensitas cahaya *grey* pada sebuah pixel. Maka *Optical Flow* bisa menghitung kecepatan sebuah pixel, magnitud dan arah atau yang disebut orientasi pada penelitian ini. Untuk mencari arah dari keseluruhan video *Optical Flow* tidak memakai metode training data pixel melainkan mempunyai metode klasifikasi orientasi pixel sendiri yaitu *Histogram Orientation of Optical Flow* atau HOOF yang bekerja dengan cara mengumpulkan orientasi yang ada pada setiap pixel pada setiap *frame* dikumpulkan sesuai dengan bagian-bagian sudut kartesius yang pada paper

disarankan memakai 8 bagian sudut yaitu sudut 0° , 45° , 90° , 135° , 180° , 225° , 270° , 315° , 360° . Histogram dengan jumlah orientasi atau arah pada setiap pixel dengan jumlah terbesar maka akan menjadi arah akhir pendeteksian arah. Paper ini mendapatkan rata-rata akurasi 98,1 %. Akan tetapi *Optical Flow* adalah metode yang sangat sensitif dengan *noise* dan juga metode pendeteksian arah menggunakan *Optical Flow-Horn Schunk* mempunyai algoritma yang sedikit kompleks.

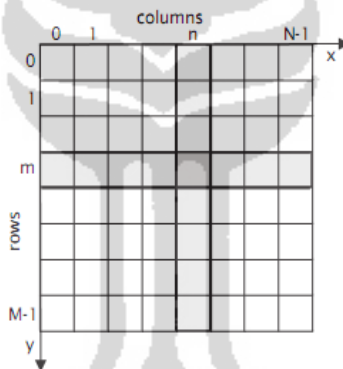
Penelitian berikutnya yang berkaitan secara tidak langsung dalam pendeteksian arah pergerakan objek adalah pada penelitian [5]. Pada penelitian ini terdapat metode untuk mendeteksi tingkah laku manusia yang berdasarkan perubahan arah pergerakan manusia yaitu *Cubic Higher-order Local Auto-Correlation (CHLAC)*. Metode ini adalah metode yang mendeteksi gerakan abnormal pada manusia dengan memanfaatkan perubahan arah gerakan manusia. Arah pergerakan yang digunakan pada paper ini adalah dengan membagi *frame* menjadi beberapa grid. Untuk mencari arah pergerakan penelitian tersebut menandai bahwa grid pertama yang ditempati objek menjadi titik pertama objek dan setelah objek bergerak ke grid berikutnya maka arah pergerakannya akan didapat bahwa objek bergerak antara grid pertama dan grid berikutnya. Untuk mencari perubahan tingkah laku manusia maka pixel akan ditraining untuk gerakan atau tingkah laku yang normal dan abnormal pada sebuah pixel dalam sebuah grid. Akan tetapi penelitian tersebut belum bisa menunjukkan arah pergerakan objek secara terperinci dengan parameter sudut melainkan hanya dengan parameter arah yang berpindah dan juga terdapat banyak metode untuk menyelesaikan masing-masing fungsi.

Dari ketiga paper yang berisikan tentang metode deteksi arah pergerakan objek yang berhubungan secara langsung maupun tidak langsung dengan deteksi arah pergerakan manusia yang paling sesuai dengan penelitian ini adalah deteksi arah pergerakan objek dengan menggunakan *Optical Flow*, karena deteksi arah pergerakan dengan *Optical Flow* tidak perlu lagi memakai deteksi pergerakan objek yang digunakan pada metode yang pertama dan untuk klasifikasi arah pergerakan objek pun tidak perlu menggunakan sistem *training data* seperti pada paper pertama dan ketiga karena *Optical Flow* menggunakan metode klasifikasi arah sendiri yaitu menggunakan klasifikasi *Histogram Orientation of Optical Flow* yang mengumpulkan orientasi atau arah pergerakan per pixel pada setiap *frame* ke dalam bentuk bagian sudut kartesius, namun deteksi arah pergerakan pada penelitian tersebut menggunakan *Optical Flow-Horn Schunk* mempunyai algoritma yang sedikit kompleks. Maka pada penelitian ini digunakan metode *Optical Flow-Lucas Kanade* yang mempunyai algoritma yang sangat sederhana.

2.2 Tinjauan Teori

2.2.1 Representasi Citra Digital

Citra merupakan istilah dari gambar yang merupakan informasi berbentuk visual. Citra juga bisa dikatakan gambaran atau representasi dari suatu objek. Citra dibedakan menjadi dua jenis yaitu citra analog dan digital. Citra analog adalah citra yang bersifat kontinu yang jika harus diolah harus diubah ke dalam bentuk digital dan alat akuisisi citra analog adalah mata manusia, sedangkan citra digital adalah citra yang dapat diolah oleh komputer dan diakuisisi oleh sensor optik seperti kamera digital, *smartphone*, *webcam*, *scanner*, mikroskop digital, pesawat radiodiagnostik seperti *CT Scan (Computerized Tomography)*, *MRI (Magnetic Resonance Imaging)*, dan *USG (Ultrasonography)* [6]. Setelah diakuisisi citra digital akan membentuk bidang dua dimensi yaitu koordinat pixel (x,y) seperti pada Gambar 2. 1 dan nilai besar intensitas yang pixel disimbolkan $f(x,y)$ yang terdiri dari M kolom dan N baris. Gambar 2. 1 merupakan representasi citra digital, dimana perpotongan antara kolom M dan baris N disebut pixel (pixel= picture element) atau element terkecil dari sebuah citra.



Gambar 2. 1 Representasi citra digital dalam 2 dimensi [7]

Dalam bidang pengolahan citra dapat diaplikasikan dalam bidang pengolahan video (video processing) hal ini karena sejatinya video adalah sekumpulan citra yang dijalankan secara sekuensial [8].

2.2.2 Jenis Citra Digital

Pada umumnya pengolahan citra digital dibagi menjadi 3 yaitu citra warna, citra Biner dan citra Grayscale [6].

❖ Citra warna

Citra warna adalah jenis citra yang memiliki kombinasi tiga warna dasar yaitu RGB (*Red Green Blue*). Setiap masing-masing warna memiliki 256 kemungkinan nilai yang berada

pada rentang 0-256 sehingga untuk tiga warna RGB akan menghasilkan kombinasi warna sebanyak $256 \times 256 \times 256 = 16.581.375$ warna.

❖ Citra Biner

Pada citra biner warna yang ada hanya terdapat dua warna yaitu hitam dan putih yang diwakili angka 0 dan 1.

❖ Citra Grayscale

Merupakan citra yang hanya memiliki satu channel warna dan juga merupakan citra yang nilai intensitas pikselnya berdasarkan derajat keabuan. Nilai yang dihasilkan citra ini hanyalah nilai intensitas atau lebih sering disebut grayscale.

2.2.3 Optical Flow

Untuk mencari nilai arah pergerakan objek dibutuhkan suatu metode yaitu metode *Optical Flow*. *Optical Flow* adalah metode yang digunakan untuk mencari nilai vektor pergerakan. *Optical Flow* pada deteksi objek bergerak adalah memperkirakan gerakan suatu bagian dari sebuah citra berdasarkan turunan intensitas cahaya atau $I_{(x,y,t)}$ pada sebuah sekuen citra. Metode *Optical Flow* mempunyai dua asumsi dalam pendeteksi vektor pergerakannya. Asumsi pertama *Optical Flow* mengasumsikan bahwa aliran/flow atau intensitas cahaya pada pixel yang bergerak ke *frame* selanjutnya adalah konstant [9] seperti pada persamaan (2. 1) :

$$I_{(x(t),y(t),t)} = Constant \tag{2.1}$$

Keterangan :

$$I_{(x(t),y(t),t)} = \text{Intensitas cahaya Optical FLOW}$$

Dalam ruang dua dimensi *Optical FLOW* berarti seberapa jauh suatu pixel citra berpindah diantara dua *frame* citra yang berurutan. Sebuah pixel pada posisi awal (x, y, t) dengan intensitas $I_{(x,y,t)}$ akan bergerak sejauh dx , dy dan dt antara dua *frame* gambar seperti pada persamaan (2. 2) :

$$\begin{aligned} 0 &= I_{(x+\Delta x,y+\Delta y,t+\Delta t)} - I_{(x,y,t)} \\ I_{(x,y,t)} &= I_{(x+\Delta x,y+\Delta y,t+\Delta t)} - I_{(x,y,t)} \end{aligned} \tag{2.2}$$

Keterangan :

$$I_{(x(t),y(t),t)} = \text{Intensitas cahaya Optical FLOW}$$

$$I_{(x+\Delta x,y+\Delta y,t+\Delta t)} = \text{Intensitas cahaya Optical FLOW yang bergerak sejauh } \Delta x, \Delta y \text{ dan } \Delta t$$

Asumsi kedua *Optical Flow* mengasumsikan bahwa pergerakan adalah sangat kecil [9], maka persamaan pada $I_{(x,y,t)}$ dapat dikembangkan untuk memperoleh persamaan (2. 3) :

$$I_{(x+\Delta x,y+\Delta y,t+\Delta t)} = I_{(x,y,t)} + \frac{dI}{dx}\Delta x + \frac{dI}{dy}\Delta y + \frac{dI}{dt}\Delta t + H.O.T \quad (2.3)$$

H.O.T (*Higher-Order-Terms*) pada persamaan (2. 3) merupakan solusi pendekatan, karena nilainya sangat kecil maka bisa diabaikan seperti pada persamaan (2. 4) yang merupakan penyederhanaan persamaan (2. 3).

$$I_{(x+\Delta x,y+\Delta y,t+\Delta t)} = I_{(x,y,t)} + \frac{dI}{dx}\Delta x + \frac{dI}{dy}\Delta y + \frac{dI}{dt}\Delta t$$

$$\frac{dI}{dx}\Delta x + \frac{dI}{dy}\Delta y + \frac{dI}{dt}\Delta t = I_{(x+\Delta x,y+\Delta y,t+\Delta t)} - I_{(x,y,t)} \quad (2.4)$$

$$\frac{dI}{dx}\Delta x + \frac{dI}{dy}\Delta y + \frac{dI}{dt}\Delta t = 0$$

Didiferensialkan terhadap waktu, maka persamaan (2. 4) menjadi :

$$\frac{dI}{dx} \frac{\Delta x}{dt} + \frac{dI}{dy} \frac{\Delta y}{dt} + \frac{dI}{dt} \frac{\Delta t}{dt} = 0 \quad (2.5)$$

Karena turunan fungsi posisi terhadap waktu adalah kecepatan maka persamaan (2. 5) menjadi :

$$\frac{dI}{dx} V_x + \frac{dI}{dy} V_y + \frac{dI}{dt} = 0$$

$$\frac{dI}{dx} V_x + \frac{dI}{dy} V_y = -\frac{dI}{dt} \quad (2.6)$$

Dengan V_x , V_y adalah komponen kecepatan dari *Optical FLOW* $I_{(x,y,t)}$, serta $\frac{dI}{dx}$ $\frac{dI}{dy}$ $\frac{dI}{dt}$ merupakan derivatif parsial dari gambar $I_{(x,y,t)}$ yang berubah menjadi I_x , I_y , I_t [9] maka persamaan menjadi :

$$I_x V_x + I_y V_y = -I_t \quad (2.7)$$

Keterangan :

I_x = Gradien intensitas pixel pada sumbu x

I_y = Gradien intensitas pixel pada sumbu y

I_t = Gradien intensitas pixel pada ranah waktu

V_x = Kecepatan pada pixel sumbu x

V_y = Kecepatan pada pixel sumbu y

Persamaan (2. 7) adalah persamaan akhir *Optical FLOW* untuk mendapatkan kecepatan pixel pada sumbu X dan pada sumbu Y. Namun persamaan tersebut tidak dapat diselesaikan

karena terdapat dua variabel dalam satu persamaan. Permasalahan ini disebut *aperture problem* dari *Optical Flow*. Maka dibutuhkan metode pengembangan untuk bisa menentukan V_x dan V_y pada persamaan diatas dengan metode *Lucas-Kanade* [9] .

2.2.4 Algoritma *Lucas-Kanade Optical Flow*

Algoritma Lucas-Kanade merupakan algoritma *Optical Flow* yang menyelesaikan persamaan *Optical Flow* sebelumnya yaitu dengan cara menggabungkan informasi dari pixel-pixel yang berdekatan [9] . Metode ini mengasumsikan vektor pada pixel sama dengan vektor pixel tetangganya. Maka persamaan pada metode ini menjadi :

$$\begin{aligned} I_x(q_1)V_x + I_y(q_1)V_y &= -I_t(q_1) \\ &\vdots \\ I_x(q_2)V_x + I_y(q_2)V_y &= -I_t(q_2) \\ I_x(q_n)V_x + I_y(q_n)V_y &= -I_t(q_n) \end{aligned} \tag{2. 8}$$

Dimana q_1, q_2, q_3 adalah pixel dalam suatu *frame*, sedangkan $I_x(q_i), I_y(q_i), I_t(q_i)$ adalah turunan parsial dari gambar yang berhubungan dengan posisi x, y dan waktu t. Persamaan diatas ditulis dalam bentuk matriks menjadi persamaan (2. 9):

$$A v = (-b) \tag{2. 9}$$

$$A = \begin{bmatrix} I_x(q_1) & I_y(q_1) \\ \vdots & \vdots \\ I_x(q_n) & I_y(q_n) \end{bmatrix}, \quad v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}, \quad b = \begin{bmatrix} -I_t(q_1) \\ -I_t(q_2) \\ \vdots \\ -I_t(q_n) \end{bmatrix}$$

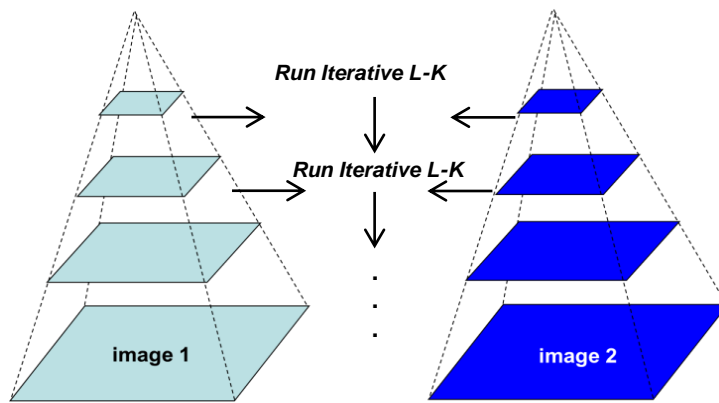
Ketika ingin mencari nilai v maka persamaan matriks akan berubah menjadi :

$$v = (A^T A)^{-1} A^T (-b) \tag{2. 10}$$

Persamaan dalam bentuk matriks akan menjadi :

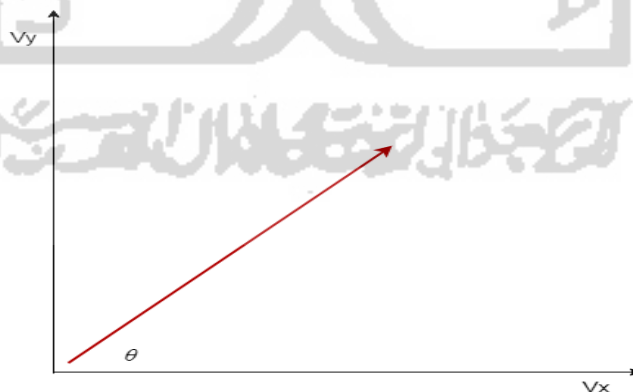
$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i I_x(q_i)^2 & \sum_i I_x(q_i)I_y(q_i) \\ \sum_i I_x(q_i)I_y(q_i) & \sum_i I_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i I_x(q_i)I_t(q_i) \\ -\sum_i I_y(q_i)I_t(q_i) \end{bmatrix} \tag{2. 11}$$

Dengan persamaan (2. 11) diatas maka akan didapatkan nilai V_x dan V_y pada sebuah gambar. Akan tetapi asumsi *Optical Flow* tentang pergerakan pixel yang kecil memiliki kelemahan yaitu tidak terdeteksinya gerakan yang besar. Permasalahan ini dapat diatasi dengan prinsip pyramid yaitu *Pyramida Lucas-Kanade* [10].



Gambar 2. 2 *Pyramida Lucas-kanade Optical FLOW* [10]

Prinsip pyramida ini bekerja mulai dari detail citra paling rendah hingga detail citra paling tinggi seperti pada Gambar 2. 2, tujuannya adalah agar gerakan-gerakan yang besar dapat ikut diperhitungkan. Dengan kata lain prinsip ini bekerja dengan *frame* yang diperbesar setiap layer pyramidanya. Pada level atas maka akan dihitung vektor pergerakannya seperti pada persamaan diatas, pada level berikutnya maka *frame* diperbesar dan otomatis pixel yang diperhitungkan vektor nya akan dilipatgandakan untuk menghitung vektor pergerakan yang kedua tetapi tetap dalam pixel yang sama, pada level berikutnya pixel dilipatgandakan lagi dan *frame* diperbesar untuk menghitung vektor pergerakan yang ketiga tetapi tetap dalam pixel yang sama, begitu seterusnya sampai jarak atau pergerakan yang besar sudah tercapai. Ketika iterasi telah mencapai jarak yang telah dicapai maka nilai kecepatan pixel setiap iterasi nya akan ditambahkan dengan kecepatan pixel pada iterasi pertama kali. Maka nilai V_x dan V_y y pada setiap pixel akan didapatkan [9] seperti pada Gambar 2. 3 berikut.



Gambar 2. 3 Vektor pergerakan hasil pendeteksian *Optical Flow*

Dengan adanya nilai V_x dan V_y maka arah atau yang biasa disebut orientasi atau θ pada *Optical Flow* bisa di dapat [4]. Pada penelitian ini nilai yang digunakan untuk mencari orientasi adalah nilai kecepatan pixel pada sumbu y dan nilai kecepatan pixel pada sumbu x atau direpresentasikan pada persamaan (2. 12):

$$\theta(V_x, V_y) = \tan^{-1} \frac{V_y}{V_x} \quad (2.12)$$

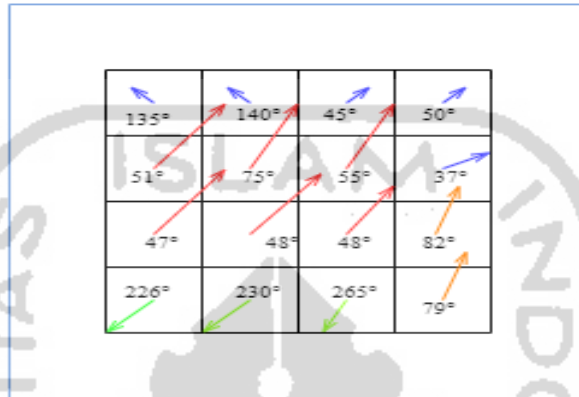
Keterangan :

θ = Orientasi

V_x = Kecepatan pada pixel sumbu x

V_y = Kecepatan pada pixel sumbu y

Maka orientasi yang didapat seperti pada gambar Gambar 2. 4 yang merupakan contoh orientasi pada suatu *frame*.



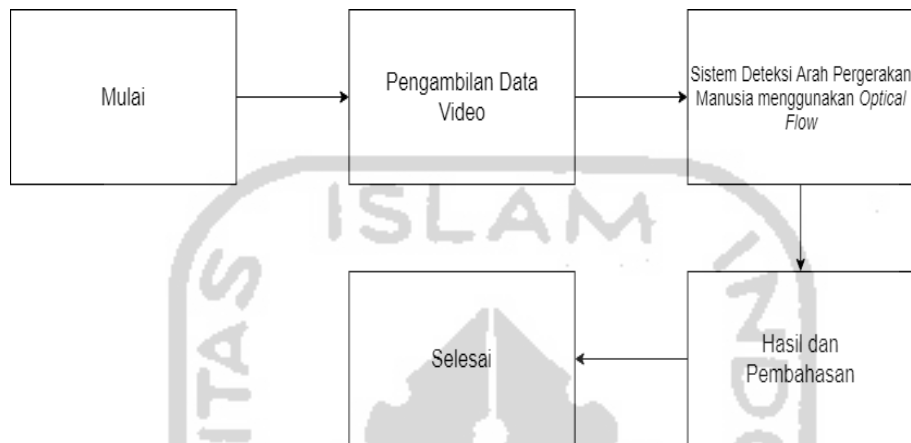
Gambar 2. 4 Contoh Orientasi pada sebuah *frame* [11]

BAB 3

METODOLOGI

3.1 Alur Penelitian

Proses penelitian yang dilakukan dijelaskan pada Gambar 3. 1 yang merupakan diagram blok alur penelitian ini.



Gambar 3. 1 Diagram Blok Alur Penelitian

3.1.1 Pengambilan Data Video

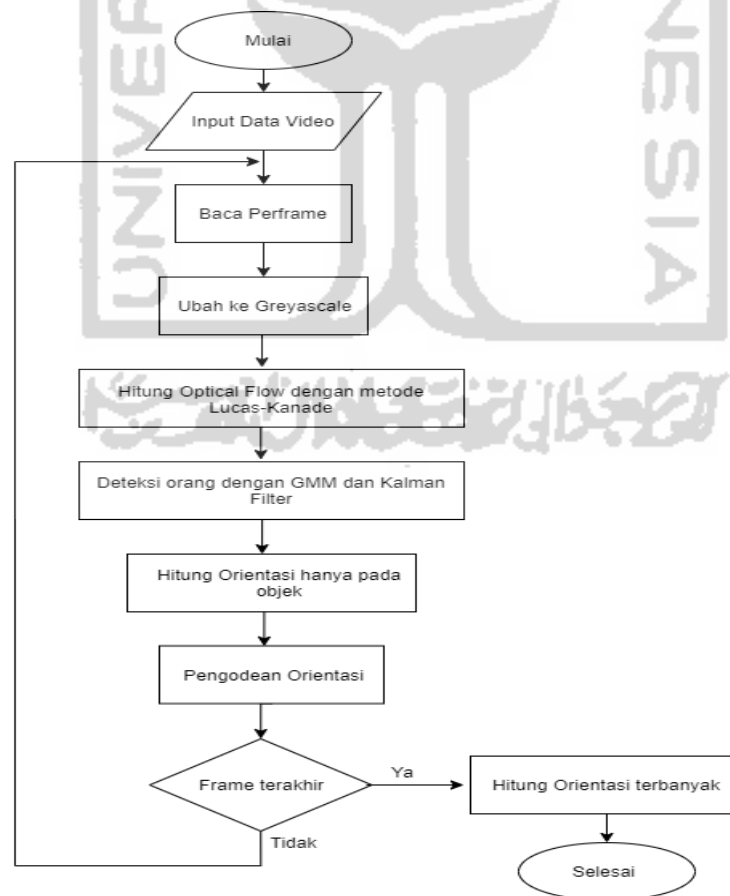
Tahap pertama dalam penelitian ini adalah proses pengambilan data video sebagai data input yang akan diproses dan dideteksi oleh pendeteksi. Data video tersebut diambil menggunakan kamera CCTV (*Closed Circuit Television*) yang telah terpasang pada setiap sudut ruangan berbentuk persegi panjang pada ruangan Laboratorium Pemrograman Komputer UII dengan resolusi video yang dihasilkan adalah 432x240 pixel. Pada pengambilan data video, jumlah data video yang digunakan berjumlah 16 video, dengan jumlah empat video pada tiap-tiap kamera CCTV yang terpasang. Empat video pada masing-masing kamera berisikan data video manusia berjalan ke arah depan, berjalan ke arah belakang, berjalan ke arah kiri, dan berjalan ke arah kanan. Arah pergerakan manusia tersebut relatif pada masing-masing kamera seperti pada Gambar 3. 2 berikut :



Gambar 3. 2 Data video yang telah diambil pada setiap kamera CCTV

3.1.2 Sistem Deteksi Arah Pergerakan Manusia Menggunakan *Optical Flow*

Data video yang telah diambil pada tahap pertama akan dideteksi menggunakan algoritma *Lucas-Kanade Optical Flow* yaitu berdasarkan vektor arah. Pada tahap ini proses deteksi objek menggunakan algoritma *Lucas-Kanade Optical Flow* dirancang menggunakan *software Matlab (Matrix Laboratory) 2016* dimana setiap tahap deteksi menggunakan fungsi masing-masing untuk menyelesaikan penelitian ini seperti pada Gambar 3. 3 berikut ini:



Gambar 3. 3 Diagram Alir Deteksi Arah Pergerakan Objek

- Input data video

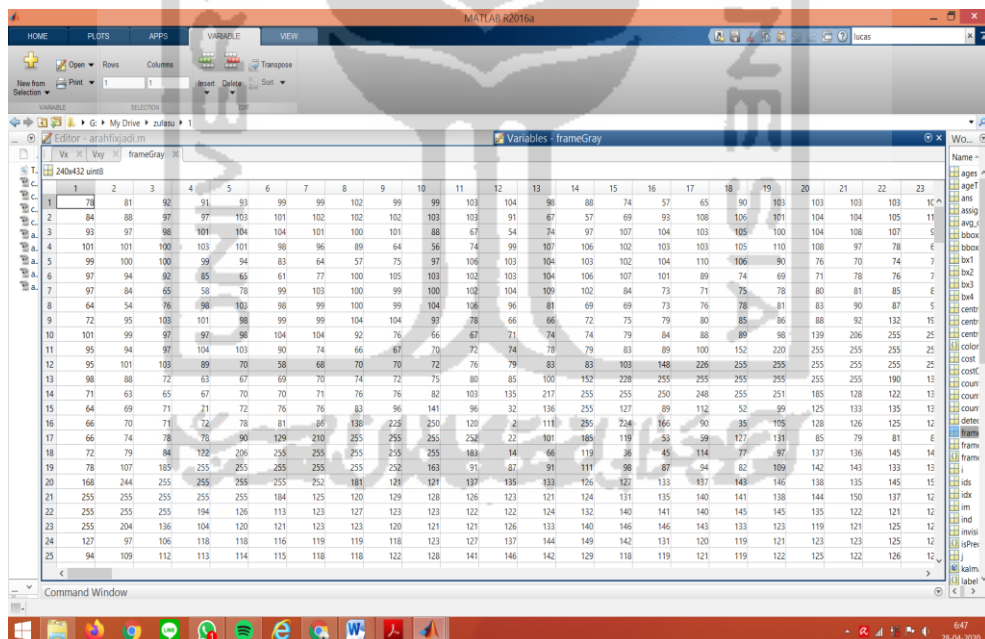
Pada analisis arah pergerakan manusia input yang digunakan adalah data video telah direkam dari kamera CCTV pada tahap sebelumnya. Data dari masing kamera yang berjalan ke arah depan, ke arah belakang, ke arah kiri dan ke arah kanan dimasukkan secara bergantian pada program Matlab untuk menghasilkan parameter arah pergerakan atau yang disebut *Orientation* pada *Optical FLOW*.

- Baca perframe

Proses yang dilakukan setelah video dimasukkan ke dalam program Matlab adalah program akan membaca video secara per-frame, karena pada hakikatnya video adalah sekumpulan *frame* yang bergerak.

- Ubah ke *Greyscale*

Pada proses ini citra yang awalnya berbentuk *Red, Green, Blue* atau RGB akan diubah ke citra *Greyscale*, karena *Optical FLOW* hanya bisa membaca inputan perubahan intensitas cahaya derajat keabuan yang terdapat pada *Greyscale*. Konversi RGB ke *Greyscale* pada penelitian ini tidak ditampilkan dalam bentuk gambar melainkan dalam bentuk angka seperti pada Gambar 3. 4 :



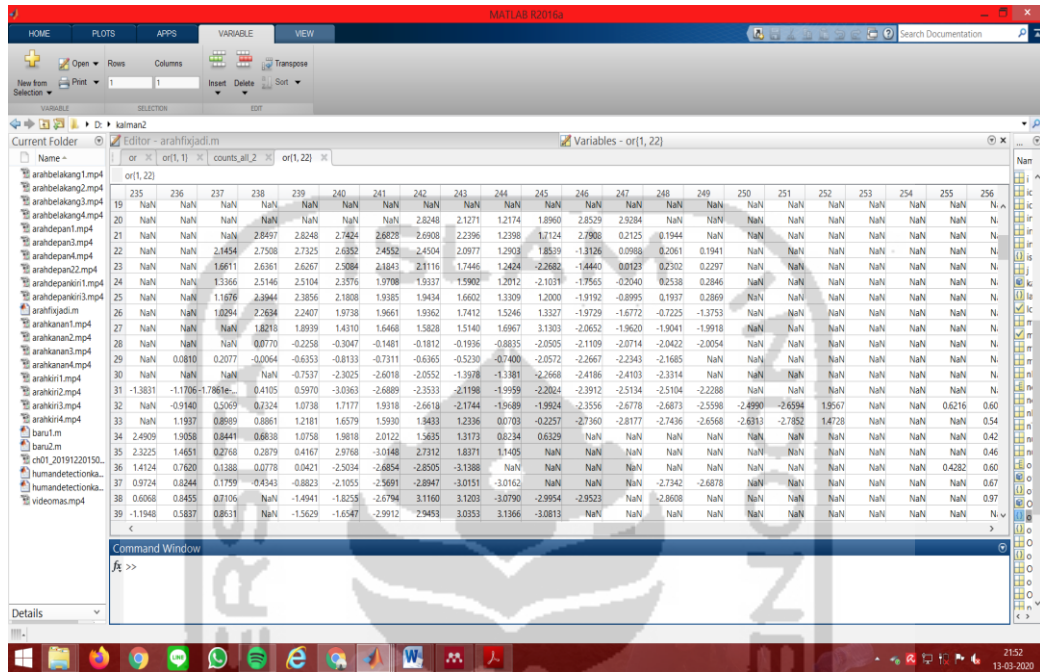
Gambar 3. 4 Nilai *Greyscale* pada *frame* terakhir

Nilai diatas merupakan nilai *Greyscale* yang mempunyai rentang 0-255 pada derajat keabuan. Rentang tersebut berasal dari *Greyscale* 8 bit.

- Hitung orientasi dengan algoritma *Lucas-Kanade Optical FLOW*

Setelah citra RGB diubah ke citra *Greyscale* maka *Optical FLOW* melalui metode *Lucas-Kanade* mulai menghitung orientasi yang ada pada citra seperti pada Gambar 3. 5,

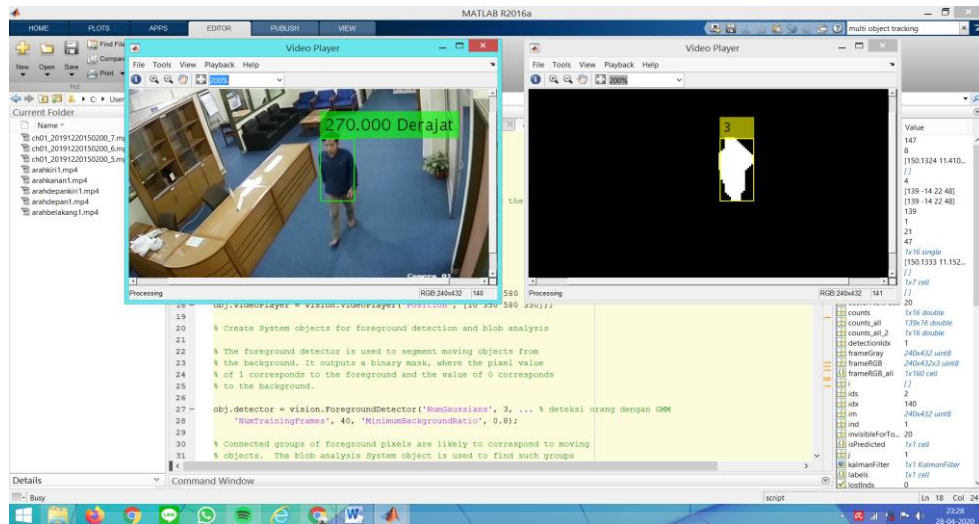
orientasi yang dihitung masih orientasi pada gambar belum diseleksi orientasi hanya pada objek ,orientasi yang bernilai negatif ditambahkan 360 untuk memudahkan dalam perhitungan. Orientasi *Not a Number* atau NaN adalah nilai orientasi jika nilai V_x dan V_y sama dengan nol. Pengkodean NaN bertujuan agar orientasi berasal dari V_x dan V_y sama dengan nol tersebut tidak termasuk didalam HOOF karena HOOF mengambil nilai terbanyak sesuai dengan bagian-bagian sudut atau arahnya.



Gambar 3. 5 Hasil Orientasi *Optical Flow*

- Deteksi orang dengan GMM dan Kalman Filter

Fungsi GMM dan Kalman Filter pada penelitian ini adalah untuk memudahkan dalam memvisualisasi orientasi pergerakan manusia agar dan juga mempercepat dalam menganalisis orientasi pergerakan manusia. Deteksi orang dengan *Gaussian Mixture model* atau GMM yang berfungsi untuk mengsegmentasi objek atau mendeteksi *foreground* (latar depan/objek) dengan cara memisahkan nilai pixel yang tidak sesuai dengan distribusi *Gaussian* pada *background*. *Kalman filter* berfungsi untuk mentracking suatu objek yang telah didefinisikan oleh GMM. Pada saat objek bergerak atau *frame* berjalan maka *kalman filter* akan memprediksi pergerakan objek setiap *frame*-nya. *Frame* objek yang ditracking oleh *kalman filter* dan GMM akan disegmentasi dengan *Region of Interest* berupa segi empat untuk membatasi pixel *frame* objek dan sebagai visualisasi *tracking*. Jika *frame* objek yang di-tracking dari area awal telah melewati *detecting lane*, maka objek akan didefinisikan secara otomatis karena objek telah berada di area berikutnya [12]. Sistem deteksi ini bekerja secara otomatis mulai dari identifikasi *foreground* dan *background* citra, *tracking*, dan visualisasi arah dalam bentuk derajat.

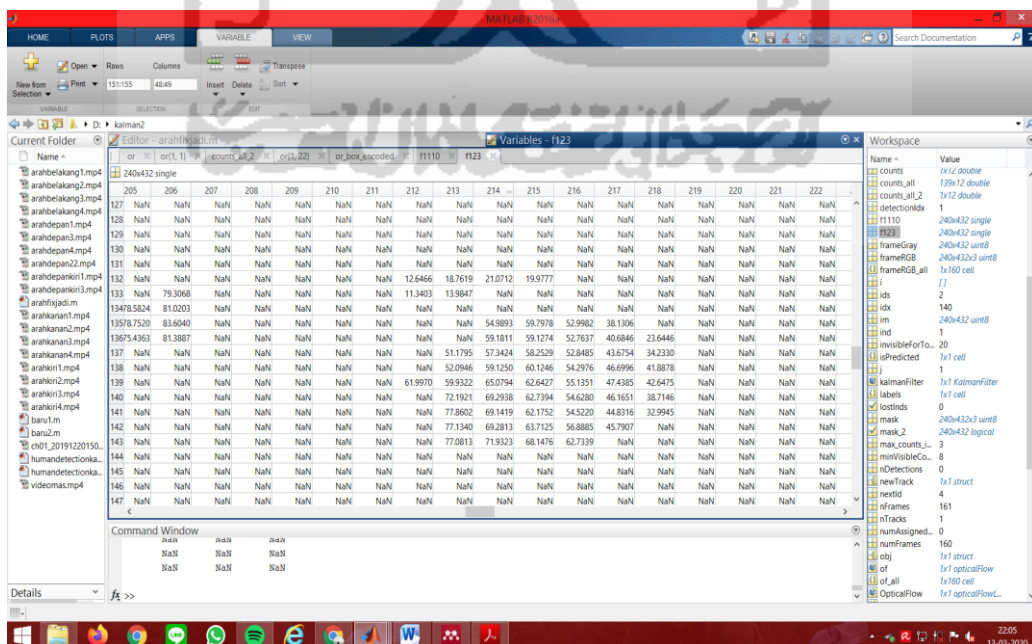


Gambar 3. 6 Deteksi dan Tracking pergerakan manusia dengan GMM dan kalman filter

Pada Gambar 3. 6 adalah gambar hasil visualisasi tracking pergerakan manusia dengan deteksi dan tracking objek GMM dan kalman filter. Pada output visualisasi yang berbentuk biner menunjukkan hasil fungsi dari GMM dan kalman filter.

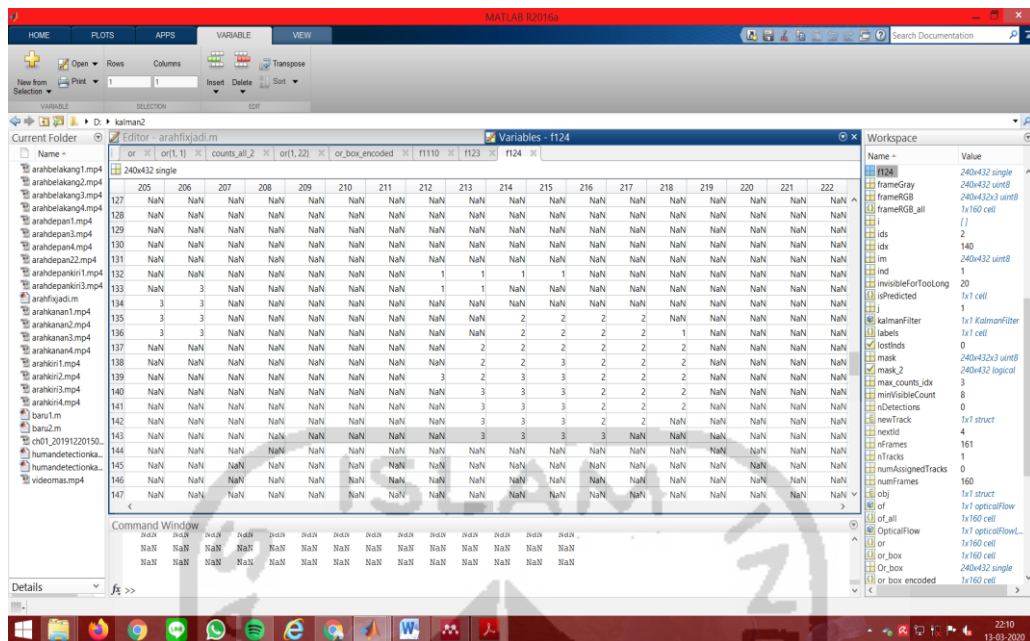
- Hitung orientasi hanya pada objek (pergerakan manusia)

Pada penelitian ini arah yang dianalisis adalah arah pergerakan manusia. Orientasi yang dihitung dan dianalisis adalah orientasi pada objek. Orientasi pada objek dibedakan dengan adanya *bbox* atau kotak penanda pada objek untuk membedakan objek manusia dengan objek bergerak lainnya sehingga bisa dengan mudah untuk menganalisis orientasi yang ada pada objek manusia. Pada Gambar 3. 7 yang merupakan orientasi hanya pada objek yaitu orientasi yang ada hanya pada *foreground video*.



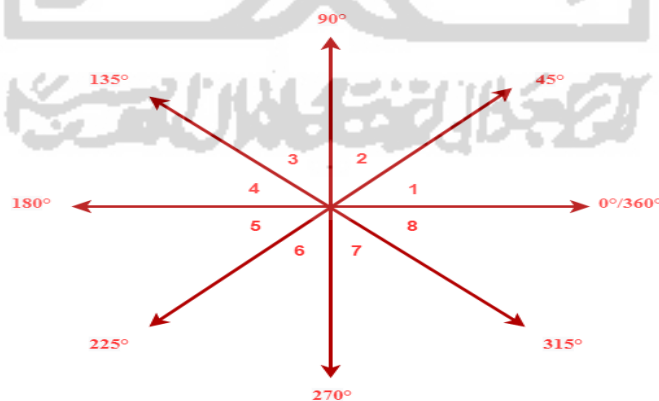
Gambar 3. 7 Orientasi pada objek manusia

- Pengodean orientasi



Gambar 3. 8 Orientasi pada objek manusia yang ditelah dikodekan

Pada proses ini orientasi yang didapat akan dikodekan atau diubah kedalam klasifikasi *Histogram Orientation of Optical Flow* atau HOOF. *Histogram Orientation of Optical Flow* adalah histogram yang berfungsi untuk mengumpulkan nilai orientasi setiap pixel pada setiap *frame* dalam rentang sudut atau arah tertentu [4]. Pada penelitian ini orientasi dibagi menjadi 8 bagian seperti pada Gambar 3. 9 berikut :

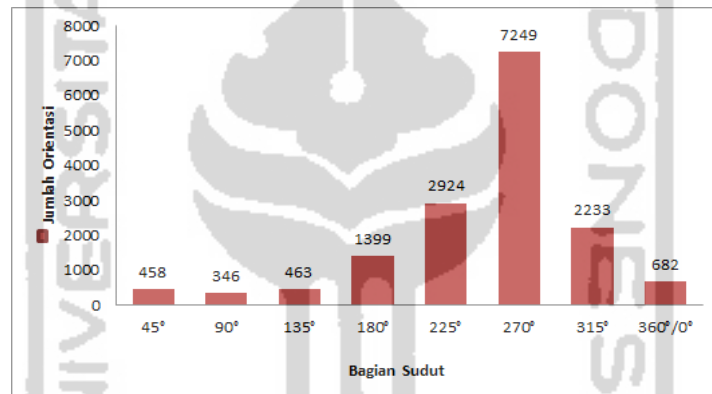


Gambar 3. 9 Bagian- bagian orientasi

Pada Gambar 3. 9 yang merupakan bagian sudut orientasi yang berjumlah 8 bagian dengan deskripsi sebagai berikut :

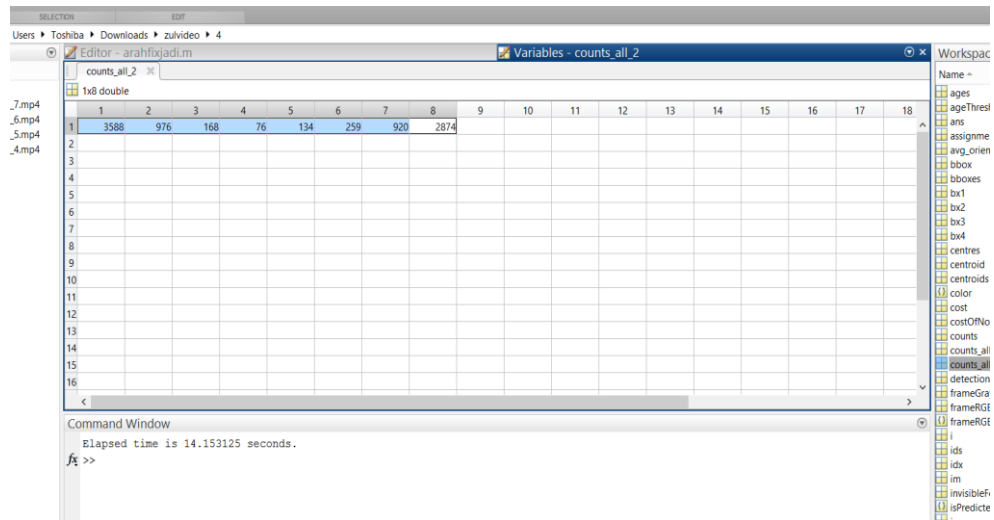
Bagian sudut	Rentang Sudut
1	$0^{\circ}/360^{\circ} < \theta \leq 45^{\circ}$
2	$45^{\circ} < \theta \leq 90^{\circ}$
3	$90^{\circ} < \theta \leq 135^{\circ}$
4	$135^{\circ} < \theta \leq 180^{\circ}$
5	$180^{\circ} < \theta \leq 225^{\circ}$
6	$225^{\circ} < \theta \leq 270^{\circ}$
7	$270^{\circ} < \theta \leq 315^{\circ}$
8	$315^{\circ} < \theta \leq 0^{\circ}/360^{\circ}$

Maka hasil pengodean tersebut akan berbentuk histogram dimana histogram dengan jumlah terbesar yang akan menjadi arah pergerakan manusia seperti pada Gambar 3. 10 berikut :



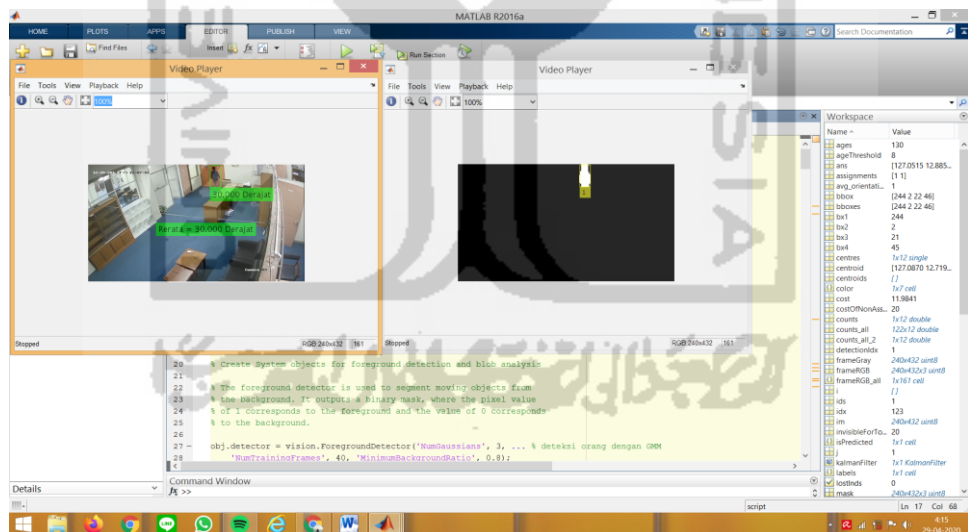
Gambar 3. 10 Pengodean orientasi kedalam bentuk histogram HOOF

- *Frame* terakhir
Ketika proses penjalaran video dan perhitungan arah pergerakan manusia memasuki *frame* terakhir maka sistem penjalaran video pada *Matlab* akan berakhir.
- Hitung Orientasi terbanyak
Ketika sistem penjalaran pada *Matlab* selesai untuk *frame* terakhir maka sistem akan mengumpulkan orientasi pada setiap pixel di setiap *frame* dan akan diurutkan sesuai dengan HOOF atau mengumpulkan orientasi dengan rentan nilai terbanyak.



Gambar 3. 11 Jumlah orientasi terbanyak pada setiap bagian sudut

Pada Gambar 3. 11 yaitu orientasi terbanyak pada setiap bagian sudut atau arah. Pada penelitian ini jumlah bagian sudut atau arah yang digunakan adalah berjumlah 8 bagian seperti pada Gambar 3. 9 Jumlah orientasi yang paling banyak akan menjadi hasil akhir dalam penentuan arah pergerakan manusia. Gambar 3. 12 merupakan hasil akhir yang divisualisasikan untuk memudahkan menganalisis arah hanya dengan melihat hasil akhir.

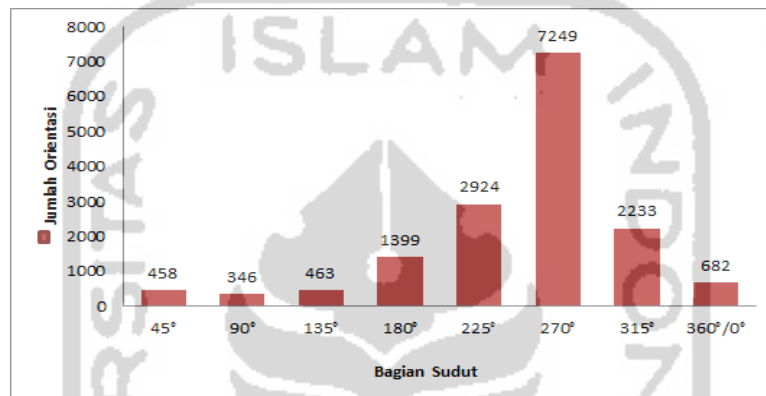


Gambar 3. 12 Visualisasi hasil perhitungan Orientasi objek

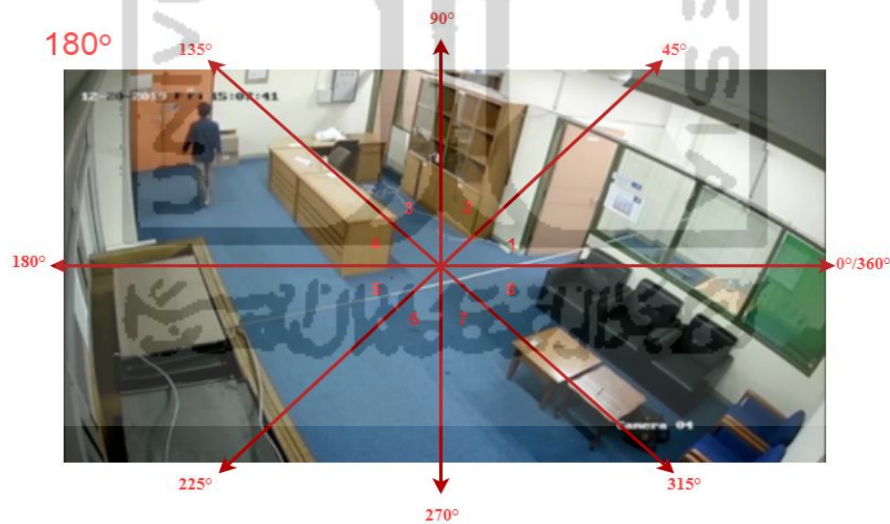
3.1.3 Hasil dan Pembahasan

Hasil dan pembahasan pada penelitian ini adalah menampilkan hasil deteksi arah pergerakan manusia dengan histogram HOOF menggunakan sistem deteksi *Optical Flow* pada setiap kamera dengan masing-masing arah ke depan, belakang, kanan dan kiri. Hasil deteksi arah berbentuk histogram tersebut menggunakan sistem deteksi *Optical Flow* akan dibandingkan

dengan arah yang diukur secara manual dengan diagram kartesius untuk menentukan apakah hasil deteksi arah dengan *Optical Flow* terdeteksi dengan benar. Arah yang diukur secara manual dengan diagram kartesius dibuat dengan cara mengukur manual arah pergerakan objek dengan diagram kartesius. Perbandingan tersebut akan dilakukan pembahasan untuk mengetahui apakah deteksi arah pergerakan manusia dengan *Optical Flow* tersebut berjalan dengan baik. Arah hasil pendeteksian arah dengan pengukuran secara manual mempunyai kartesius yang berbeda-beda terhadap sudut pandang kamera atau relatif terhadap setiap kamera hal ini karena deteksi arah dengan *Optical Flow* mendeteksi arah pergerakan secara relatif terhadap masing-masing kamera [4]. Pada hasil dengan menggunakan sistem deteksi *Optical Flow* juga akan ditampilkan histogram HOOOF.



Gambar 3. 13 Histogram orientasi *Optical Flow*



Gambar 3. 14 Hasil pengukuran arah secara manual menggunakan diagram kartesius



Gambar 3. 15 Visualisasi deteksi arah menggunakan *Optical Flow*

Pada Gambar 3. 13 merupakan histogram HOOF atau histogram arah dari *Optical Flow* dengan hasil orientasi terbanyak pada histogram yang akan menjadi arah pergerakan. Gambar 3. 14 yang merupakan deteksi arah pergerakan manusia menggunakan hasil pengukuran manual dengan diagram kartesius yang akan dibandingkan dengan hasil deteksi arah pergerakan manusia dengan histogram menggunakan *Optical Flow* . Gambar 3. 15 adalah visualisasi yang didapat dari sebuah HOOF dan akan divisualisasikan untuk memudahkan menganalisis arah pergerakan dengan hanya melihat hasil akhir.

BAB 4

HASIL DAN PEMBAHASAN

4.1 Hasil pengujian sistem pendeteksian arah pergerakan menggunakan *Optical Flow*

Pada bagian ini sistem deteksi *Optical Flow* diuji untuk mengetahui apakah sistem deteksi optical flow bekerja dengan baik, tahap pengujiannya adalah menguji input *video offline* dari 4 kamera CCTV yang berjumlah total 16 *video offline*. Maka hasil pengujian deteksi arah menggunakan *Optical Flow* akan menjadi seperti pada Gambar 4. 1 dibawah ini :



Gambar 4. 1 Hasil pengujian deteksi arah menggunakan *Optical Flow* pada arah depan kamera 1

Gambar 4. 1 adalah visualisasi hasil pengujian pendeteksian deteksi arah *Optical Flow* pada salah satu video dan hasil keseluruhan hasil pengujian pendeteksian arah pergerakan manusia menggunakan *Optical Flow* pada semua inputan video adalah seperti Tabel 4. 1 dibawah ini :

Tabel 4. 1 Hasil Pengujian Pendeteksian Arah menggunakan *Optical FLOW*

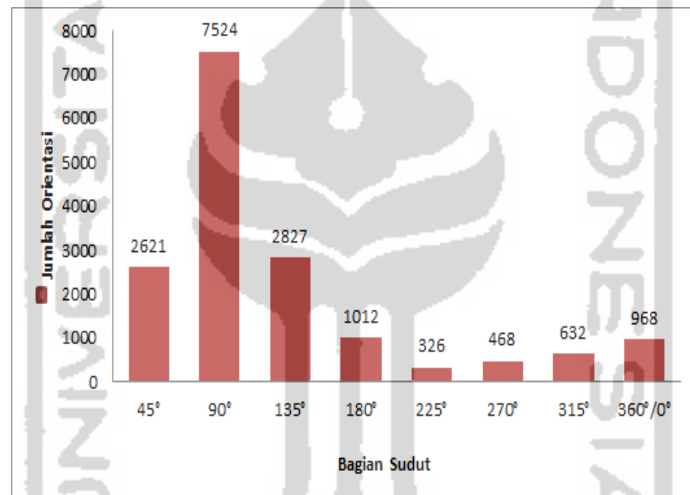
No	Kamera	Arah	Terdeteksi Optical Flow
1	1	Depan	Terdeteksi
2		Belakang	Terdeteksi
3		Kanan	Terdeteksi
4		Kiri	Terdeteksi
5	2	Depan	Terdeteksi
6		Belakang	Terdeteksi
7		Kanan	Terdeteksi
8		Kiri	Terdeteksi
9	3	Belakang	Terdeteksi
10		Depan	Terdeteksi
11		Kiri	Terdeteksi
12		Kanan	Terdeteksi
13	4	Belakang	Terdeteksi
14		Depan	Terdeteksi

No	Kamera	Arah	Terdeteksi Optical Flow
15		Kiri	Terdeteksi
16		Kanan	Terdeteksi

Pada Tabel 4. 1 diatas bahwa deteksi arah pergerakan manusia menggunakan *Optical Flow* mampu mendeteksi arah pergerakan manusia pada penelitian ini dengan 100% keberhasilan pendeteksian dari 16 video yang diuji coba pada 4 kamera.

4.2 Hasil orientasi arah pergerakan dengan klasifikasi HOOF dan akurasi deteksi arah menggunakan *Optical Flow*.

Hasil orientasi arah pergerakan *Optical Flow* yang diklasifikasi menggunakan histogram HOOF dengan diagram kartesius 8 bagian seperti pada Gambar 4. 2 berikut :



Gambar 4. 2 Hasil orientasi arah pergerakan *Optical Flow* dengan HOOF pada arah depan kamera 1

Gambar 4. 2 adalah histogram HOOF deteksi *Optical Flow* pada arah depan kamera 1 maka hasil orientasi arah pergerakan menggunakan klasifikasi HOOF pada semua video akan ditampilkan dalam bentuk tabel seperti pada Tabel 4. 2 berikut :

Tabel 4. 2 Hasil Deteksi Arah menggunakan Optical Flow dalam bentuk Orientasi

No	Kamera	Arah	Jumlah orientasi dengan akumulasi HOOF								Bagian Sudut dengan Orientasi Terbanyak
			Bagian sudut								
			45°	90°	135°	180°	225°	270°	315°	360°	
1	1	Depan	2621	7524	2827	1012	326	468	632	968	90°
2		Belakang	458	346	463	1399	2924	7249	2233	682	270°
3		Kanan	193	33	30	0	21	16	65	220	360°
4		Kiri	238	54	67	678	135	311	99	315	180°
5	2	Depan	1248	1865	2181	6468	2107	1586	733	986	180°
6		Belakang	2408	966	249	259	218	503	2396	8266	360°
7		Kanan	2968	800	308	80	48	93	636	2872	45°
8		Kiri	265	217	353	650	487	565	68	304	180°
9	3	Belakang	563	875	1108	2646	5075	3665	1293	919	225°
10		Depan	10776	4166	623	116	173	281	545	2466	45°
11		Kiri	589	1175	1389	3704	2118	1324	571	585	180°
12		Kanan	2126	586	144	115	239	173	755	3630	360°
13	4	Belakang	1779	425	249	298	457	1237	8397	225°	315°
14		Depan	2143	3557	5487	5605	1365	2100	257	45°	180°
15		Kiri	436	535	714	1899	1134	1050	316	180°	180°
16		Kanan	5817	1591	287	125	249	446	1447	360	45°

Pada awalnya orientasi yang dihasilkan oleh optical flow pada keseluruhan *frame* dalam suatu video belum bisa menjadi arah akhir dalam deteksi arah pergerakan karena belum ada proses klasifikasi orientasi, maka dalam proses akhir pendeteksian arah pada optical flow dibutuhkan klasifikasi arah dengan *Histogram Oriented of Optical Flow* (HOOF) yang mengumpulkan semua orientasi yang didapat pada setiap pixel dalam setiap *frame* ke bagian-bagian sudut pada diagram kartesius 8 bagian. Bagian sudut dengan orientasi terbanyak akan menjadi arah akhir dalam pendeteksian arah pergerakan pada *Optical Flow* seperti pada Tabel 4. 2 yang merupakan hasil orientasi yang dikumpulkan ke dalam HOOF 8 bagian. Bagian sudut atau histogram yang mempunyai orientasi terbanyak akan menjadi hasil akhir arah pergerakan.

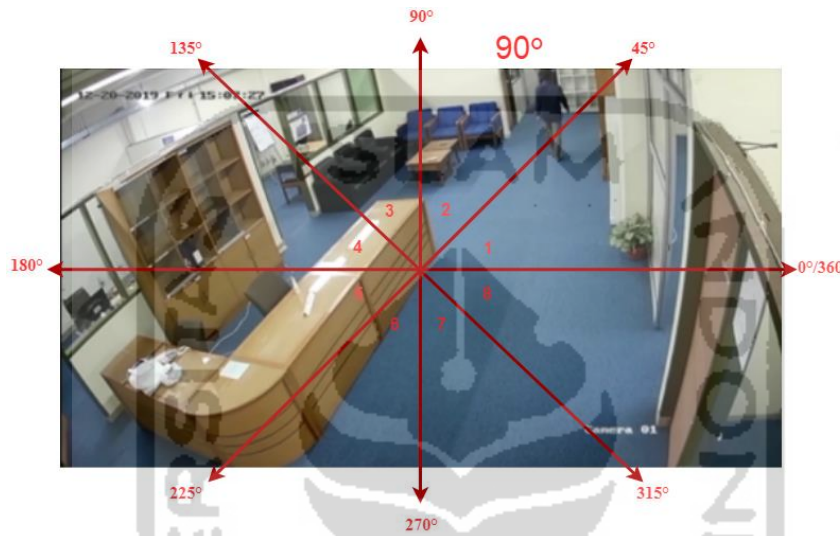
Pada kamera 1 dengan arah depan, belakang, kanan dan kiri mempunyai orientasi terbanyak atau hasil akhir arah deteksi pergerakan pada arah 90°, 270°, 360°, 180° dengan jumlah orientasi sebesar 7524, 7249, 220, 678.

Pada kamera 2 dengan arah depan, belakang, kanan dan kiri mempunyai orientasi terbanyak atau hasil akhir arah deteksi pergerakan pada arah 180°, 360°, 45°, 180° dengan jumlah orientasi sebesar 6468, 8266, 2968, 650.

Pada kamera 3 dengan arah belakang, depan, kiri dan kanan mempunyai orientasi terbanyak atau hasil akhir arah deteksi pergerakan pada arah 225°, 45°, 180°, 360° dengan jumlah orientasi sebesar 5075, 10776, 3704, 3630.

Pada kamera 4 dengan arah belakang, depan, kiri dan kanan mempunyai orientasi terbanyak atau hasil akhir arah deteksi pergerakan pada arah 315°, 180°, 180°, 45° dengan jumlah orientasi sebesar 8397, 5605, 1899, 5817.

Untuk mengetahui akurasi hasil akhir arah pergerakan tersebut maka akan dibandingkan dengan hasil pendeteksian secara manual dengan menggunakan diagram kartesius 8 bagian. Pada masing-masing kamera perhitungan secara manual dengan diagram kartesius mempunyai sudut pandang yang berbeda atau arah pada setiap kamera adalah relatif. Gambar 4. 3 merupakan perhitungan secara manual :



Gambar 4. 3 Hasil perhitungan secara manual

Gambar 4. 3 adalah hasil perhitungan arah secara manual menggunakan diagram kartesius 8 bagian yang akan dibandingkan dengan arah hasil pendeteksian *Optical Flow* seperti pada Tabel 4. 3 berikut :

Tabel 4. 3 Perbandingan deteksi arah *Optical Flow* dengan deteksi secara manual

No	Kamera	Arah	Hasil arah pergerakan menggunakan <i>Optical Flow</i>	Hasil Deteksi Manual	Hasil Perbandingan
1	1	Depan	90°	90°	Benar
2		Belakang	270°	270°	Benar
3		Kanan	360°	45°	Salah
4		Kiri	180°	180°	Benar
5	2	Depan	180°	180°	Benar
6		Belakang	360°	360°	Benar
7		Kanan	45°	45°	Benar
8		Kiri	180°	180°	Benar
9	3	Belakang	225°	225	Benar
10		Depan	45°	90°	Salah
11		Kiri	180°	180°	Benar
12		Kanan	360°	360°	Benar
13	4	Belakang	315°	315°	Benar

No	Kamera	Arah	Hasil arah pergerakan menggunakan <i>Optical Flow</i>	Hasil Deteksi Manual	Hasil Perbandingan
14		Depan	180°	180°	Benar
15		Kiri	180°	180°	Benar
16		Kanan	45°	45°	Benar

Pada Tabel 4. 3 diatas adalah perbandingan hasil arah deteksi *Optical FLOW* dengan deteksi manual didapatkan akurasi pendeteksian *Optical FLOW* pada penelitian ini sebesar 87,5 % atau dari 16 video yang diuji coba 14 diantaranya sesuai dengan deteksi manual sedangkan 2 tidak sesuai yaitu pada kamera 1 arah kanan dan kamera 3 arah depan.

4.3 Perbandingan dengan metode lain

Pada bagian ini metode, hasil, cara kerja pada penelitian akan dibandingkan dengan penelitian-penelitian yang sudah ada yang berkaitan secara langsung maupun tidak langsung seperti pada Tabel 4. 4 berikut :

Tabel 4. 4 Perbandingan metode yang berkaitan dengan penelitian

No.	Penelitian	Metode	Capaian akurasi	Objek yang digunakan	Pendeteksian arah berdasarkan
1	Saidasul Usmanhujaev, Shokhrukh Baydadaev and Kwon Jang Woo	Menggunakan metode <i>You Only Look Onve Version 3 (YOLOv 3)</i>	91.98 %	Kedaraan roda empat	Training pixel apakah pixel tersebut boleh dilewati oleh kendaraan atau tidak sehingga arah pergerakan yang dideteksi pada metode ini adalah dengan parameter arah pergerakan yang salah dan arah pergerakan yang benar
2	Solichin, Achmad Harjoko, Agus Putra, Agfianto Eko	Menggunakan metode <i>Optical Flow-Horn Schunk</i>	98.1 %	Manusia	Hasil orientasi arah dalam bentuk sudut menggunakan HOOOF berdasarkan diagram kartesius 12 bagian
3	Ishii, Takeyuki Murakami, Hitomi Koike, Atsushi	<i>Cubic Higher-order Local Auto-Correlation (CHLAC)</i>	82 %	Manusia	Training pixel pada sebuah grid bergerak dalam setiap <i>frame</i>
4	Penelitian pada skripsi ini	<i>Optical Flow-Lucas Kanade</i>	87,5 %	Manusia	Hasil orientasi arah dalam bentuk sudut menggunakan HOOOF berdasarkan diagram kartesius 8 bagian

Seperti pada Tabel 4. 4 diatas bahwa perbandingan dengan metode lain memiliki hasil dan cara pengerjaan yang berbeda. Ada yang berhubungan secara langsung maupun tidak langsung dalam penelitian ini. Pada penelitian 1 yaitu mendeteksi arah pergerakan kendaraan roda empat dengan parameter arah berdasarkan arah yang benar dan arah yang salah, pada penelitian 2 penelitian yang mendekati penelitian yang ditulis mendeteksi arah dengan parameter sudut, pada penelitian 3 mendeteksi tingkah laku manusia berdasarkan perubahan arah pergerakan manusia. Seperti pada tabel diatas metode yang dipakai mempunyai akurasi, cara perhitungan akurasi dan data perhitungan akurasi yang berbeda. Seperti pada penelitian 1 perhitungan akurasi menggunakan 12 data video dengan cara membandingkan hasil deteksi arah pergerakan objek oleh sistem dengan hasil deteksi manual. Pada penelitian 2 data video adalah data yang terbaik yang terdapat beberapa objek didalamnya, menghitung akurasi dengan cara membandingkan hasil deteksi arah pergerakan objek oleh sistem dengan hasil deteksi aktual. Pada penelitian 3 perhitungan akurasi adalah dengan metode yang diusulkan dan juga dengan cara membandingkan hasil deteksi arah pergerakan objek oleh sistem dengan hasil deteksi aktual. Sedangkan pada penelitian ini data video yang digunakan adalah berasal dari 4 kamera CCTV yang berjumlah total 16 video, serta perhitungan akurasinya adalah dengan membandingkan hasil deteksi arah pergerakan objek oleh sistem dengan hasil deteksi aktual. Maka dari ke empat penelitian ini didapatkan akurasi pada penelitian 1 sebesar 91.8 %, penelitian 2 sebesar 98,1%, penelitian 3 sebesar 82 %. Sedangkan pada penelitian ini sebesar 87,5 % berada dibawah penelitian 2 dan penelitian 1.

Akan tetapi dalam hal algoritma pendeteksian arah menggunakan *Optical Flow* khususnya *Lucas-kanade* ini mempunyai algoritma pendeteksian arah sederhana yang tidak perlu menggunakan training data pada video seperti pada penelitian 1 dan penelitian 3. Karena *Optical Flow* adalah metode yang digunakan untuk mendeteksi objek dengan cara mendeteksi vektor pergerakan yang salah satunya adalah arah dalam bentuk sudut dibandingkan penelitian 1 dan penelitian 3 yang metode pendeteksian dan metode deteksi arah yang berbeda. Penelitian ini secara langsung mirip seperti pada penelitian 2 tetapi metode turunan *Optical Flow* yang berbeda yaitu *Lucas-kanade* yang mempunyai formula pendeteksian objek yang lebih sederhana dibanding penelitian 2 yang menggunakan *Horn Schunk*. Penelitian ini juga menggunakan beberapa kamera CCTV yang membuat sudut pandang pendeteksian yang lebih banyak dibanding pada penelitian 1, 2 dan 3 yang menggunakan satu kamera CCTV.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka didapatkan kesimpulan sebagai berikut:

1. Sistem deteksi arah pergerakan manusia menggunakan deteksi *Optical Flow* pada penelitian ini mampu mendeteksi semua arah pergerakan manusia dengan 100% keberhasilan pendeteksian dari 16 video yang diuji coba pada 4 kamera.
2. Hasil akhir arah pergerakan manusia berbentuk orientasi yang diklasifikasi dengan *Histogram Oriented of Optical Flow* atau HOOF dengan cara mengumpulkan arah pergerakan objek pada setiap pixel yang bergerak dalam setiap *frame* ke rentan bagian sudut atau arah pada diagram kartesius 12 bagian, bagian sudut dengan jumlah orientasi terbanyak merupakan hasil akhir arah pergerakan.
3. Hasil arah pergerakan manusia menggunakan deteksi *Optical Flow* pada 16 video yang dibandingkan dengan arah hasil pengukuran secara manual menggunakan diagram kartesius 8 bagian didapatkan akurasi sebesar 87,5 %.

5.2 Saran

Berdasarkan penelitian yang telah dilakukan, penulis memiliki beberapa saran untuk penelitian kedepannya agar performa dari sistem menjadi lebih baik lagi, yaitu:

1. Menggunakan *video online* atau secara *real time* agar deteksi arah pergerakan manusia bisa direalisasikan secara langsung pada sistem keamanan rumah, keamanan anak-anak, lanjut usia dan lain-lain.
2. Penambahan filter *noise* lain untuk menghilangkan benda bergerak selain objek agar tidak ikut terhitung dalam perhitungan orientasi atau arah pergerakan objek.

DAFTAR PUSTAKA

- [1] Kementerian Perencanaan Pembangunan Nasional (Bappenas), “Jumlah Penduduk Indonesia 2019 Mencapai 267 Juta Jiwa |,” *Dkatadata.co.id*, 2019. [Online]. Available: <https://databoks.katadata.co.id/datapublish/2019/01/04/jumlah-penduduk-indonesia-2019-mencapai-267-juta-jiwa>.
- [2] Windiarto and Yanto, “PROFIL ANAK INDONESIA,” *Kementeri. Pemberdaya. Peremp. dan Perlindungan Anak ii*, vol. 5, no. 1, pp. 12–21, 2018.
- [3] S. Usmanhujaev, S. Baydadaev, and K. J. Woo, “Real-time, deep learning based wrong direction detection,” *Appl. Sci.*, vol. 10, no. 7, pp. 1–13, 2020.
- [4] A. Solichin, A. Harjoko, and A. E. Putra, “Movement direction estimation on video using optical flow analysis on multiple frames,” *Int. J. Adv. Comput. Sci. Appl.*, vol. 9, no. 6, pp. 174–181, 2018.
- [5] T. Ishii, H. Murakami, and A. Koike, “Human behavior detection method with direction change invariant features,” *Int. Symp. Med. Inf. Commun. Technol. ISMICT*, vol. 1, no. 1, pp. 257–261, 2013.
- [6] M. B. Frasetyo, J. T. Elektro, F. T. Industri, and U. I. Indonesia, “Sistem pendeteksi jatuh pada manusia berbasis pengolahan citra,” 2018.
- [7] O. N. Shpakov and G. V. Bogomolov, “Technogenic activity of man and local sources of environmental pollution,” *Stud. Environ. Sci.*, vol. 17, no. C, pp. 329–332, 1981.
- [8] Z. Wang, L. Zheng, Y. Liu, and S. Wang, “Towards Real-Time Multi-Object Tracking,” *Comput. Sci.*, vol. 1, pp. 1–9, 2019.
- [9] Y. Mi, P. K. Bipin, and R. K. Shah, “Using Lucas-Kanade Algorithms to Measure Human Movement,” *Commun. Comput. Inf. Sci.*, vol. 835, no. 1, pp. 118–130, 2019.
- [10] N. Sharmin and R. Brad, “Optimal filter estimation for Lucas-Kanade optical flow,” *Sensors (Switzerland)*, vol. 12, no. 9, pp. 12694–12709, 2012.
- [11] R. V. H. M. Colque, C. Caetano, M. T. L. De Andrade, and W. R. Schwartz, “Histograms of Optical Flow Orientation and Magnitude and Entropy to Detect Anomalous Events in Videos,” *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 3, pp. 673–682, 2017.
- [12] H. Vazirani, A. Kautsar, J. Fisika, F. Sains, and U. Diponegoro, “Implementasi Object Tracking Untuk Mendeteksi Dan Menghitung Jumlah Kendaraan Secara Otomatis Menggunakan Metode Kalman Filter Dan Gaussian Mixture Model,” *Youngster Phys. J.*, vol. 5, no. 1, pp. 13–20, 2016.

LAMPIRAN

Lampiran 1 – Rincian Biaya Skripsi

No	Rincian	Frekuensi (Kali)	Volume (Unit)	Satuan (Rp)	Jumlah (Rp)
(1)	(2)	(3)	(4)	(5)	(6) = (3) × (4) × (5)
1	Survey lokasi	2	1	Rp20,000.00	Rp40,000.00
2	Ram 4 Gb	1	1	Rp475,000.00	Rp475,000.00
3	Buku Matlab untuk pemrosesan citra digital	1	1	Rp80,000.00	Rp80,000.00
Jumlah					Rp 595,000.00

Lampiran 2 – Source Code Penelitian

```

% function 1()
clc; clear; close all
%read video
videolink = [cd, '\arahkanan1.mp4'];
% obj = setupSystemObjects(videolink);
% function obj = setupSystemObjects(videolink)
% Initialize Video I/O
% Create objects for reading a video from a file, drawing
the tracked
% objects in each frame, and playing the video.

% Create a video file reader.
obj.reader = VideoReader(videolink);

% Create two video players, one to display the video,
% and one to display the foreground mask.
obj.maskPlayer = vision.VideoPlayer('Position', [600 350
580 350]);
obj.videoPlayer = vision.VideoPlayer('Position', [10 350
580 350]);

% Create System objects for foreground detection and blob
analysis

% The foreground detector is used to segment moving objects
from
% the background. It outputs a binary mask, where the pixel
value

```

```

% of 1 corresponds to the foreground and the value of 0
corresponds
% to the background.

obj.detector = vision.ForegroundDetector('NumGaussians', 3,
... % deteksi orang dengan GMM
    'NumTrainingFrames', 40, 'MinimumBackgroundRatio',
0.8);

% Connected groups of foreground pixels are likely to
correspond to moving
% objects. The blob analysis System object is used to find
such groups
% (called 'blobs' or 'connected components'), and compute
their
% characteristics, such as area, centroid, and the bounding
box.

obj.blobAnalyser =
vision.BlobAnalysis('BoundingBoxOutputPort', true, ... %
hasil deteksi dengan GMM diperbaiki dengan blob Analyser
    'AreaOutputPort', true, 'CentroidOutputPort', true, ...
    'MinimumBlobArea', 100);
% end

% create optical flow
OpticalFlow = opticalFlowLKDoG('NoiseThreshold',0.0039);
% shapeInserter = vision.ShapeInserter('Shape', 'lines',
'BorderColor', 'Custom', 'CustomBorderColor', 255);

nFrames = 1;

color = {'green', 'blue', 'cyan', 'red', 'magenta',
'black', 'white'};

% tracks = initializeTracks(); % Create an empty array of
tracks.

% function tracks = initializeTracks()
% create an empty array of tracks
tracks = struct(...
    'id', {}, ...
    'bbox', {}, ...
    'kalmanFilter', {}, ...
    'age', {}, ...
    'totalVisibleCount', {}, ...
    'consecutiveInvisibleCount', {});
% end

```

```

nextId = 1; % ID of the next track

numFrames = 0;
v = VideoReader(videolink);
while hasFrame(v)
    readFrame(v);
    numFrames = numFrames+1;
end

idx = 1;
while hasFrame(obj.reader)
    % read frame
    frameRGB = readFrame(obj.reader); % read the next video
frame
    frameRGB = imresize(frameRGB,240/size(frameRGB,1));
    frameGray = rgb2gray(frameRGB);
    im = frameGray;

    % calculate orientation of velocity
    of = estimateFlow(OpticalFlow, im);
    of_all{nFrames} = of;
    % konversi dari koordinat optical flow ke koordinat
citra
    Vx = of.Vx;
    Vy = of.Vy;
    Vxy = Vx+Vy;
%     Vx(Vx==0) = nan;
%     Vy(Vy==0) = nan;

    orientation = -of.Orientation;
    orientation(Vxy==0) = nan;

    or{nFrames} = orientation;
%     vy{nFrames} = Vy;

    % detect people
    % [centroids, bboxes, mask] =
detectObjects(obj,imresize(frameRGB,0.25));
    % function [centroids, bboxes, mask] =
detectObjects(obj,frame)

    % Detect foreground.
    mask = obj.detector.step(imresize(frameRGB,0.5));

    % Apply morphological operations to remove noise and
fill in holes.
%     mask = medfilt2(mask);
    mask = imopen(mask, strel('disk',3));
    mask = imclose(mask, strel('disk',15));

```

```

mask = imfill(mask, 'holes');

% Perform blob analysis to find connected components.
[~, centroids, bboxes] = obj.blobAnalyser.step(mask);
%     end
bboxes = 2*(bboxes);
mask = imresize(mask,2);

%     tracks = predictNewLocationsOfTracks(tracks);
%     function tracks =
predictNewLocationsOfTracks(tracks)
    for i = 1:length(tracks)
        bbox = tracks(i).bbox;

        % Predict the current location of the track.
        predictedCentroid =
predict(tracks(i).kalmanFilter);

        % Shift the bounding box so that its center is at
        % the predicted location.
        predictedCentroid = int32(predictedCentroid) -
bbox(3:4) / 2;
        tracks(i).bbox = [predictedCentroid, bbox(3:4)];
    end
% end

%     [assignments, unassignedTracks,
unassignedDetections] = ...
%         detectionToTrackAssignment(tracks,centroids);

%     function [assignments, unassignedTracks,
unassignedDetections] = ...
%         detectionToTrackAssignment(tracks,centroids)

nTracks = length(tracks);
nDetections = size(centroids, 1);

% Compute the cost of assigning each detection to each
track.
cost = zeros(nTracks, nDetections);
for i = 1:nTracks
    cost(i, :) = distance(tracks(i).kalmanFilter,
centroids);
end

% Solve the assignment problem.
costOfNonAssignment = 20;
[assignments, unassignedTracks, unassignedDetections] =
...

```

```

        assignDetectionsToTracks(cost,
costOfNonAssignment);
    %     end

    %     tracks =
updateAssignedTracks(tracks,centroids,bboxes,assignments);
    %     function tracks =
updateAssignedTracks(tracks,centroids,bboxes,assignments)
    numAssignedTracks = size(assignments, 1);
    for i = 1:numAssignedTracks
        trackIdx = assignments(i, 1);
        detectionIdx = assignments(i, 2);
        centroid = centroids(detectionIdx, :);
        bbox = bboxes(detectionIdx, :);

        % Correct the estimate of the object's location
        % using the new detection.
        correct(tracks(trackIdx).kalmanFilter, centroid);

        % Replace predicted bounding box with detected
        % bounding box.
        tracks(trackIdx).bbox = bbox;

        % Update track's age.
        tracks(trackIdx).age = tracks(trackIdx).age + 1;

        % Update .
        tracks(trackIdx).totalVisibleCount = ...
            tracks(trackIdx).totalVisibleCount + 1;
        tracks(trackIdx).consecutiveInvisibleCount = 0;
    end
    % end

    %     tracks =
updateUnassignedTracks(tracks,unassignedTracks);
    %     function tracks =
updateUnassignedTracks(tracks,unassignedTracks)
    for i = 1:length(unassignedTracks)
        ind = unassignedTracks(i);
        tracks(ind).age = tracks(ind).age + 1;
        tracks(ind).consecutiveInvisibleCount = ...
            tracks(ind).consecutiveInvisibleCount + 1;
    end
    % end

    %     tracks = deleteLostTracks(tracks);
    %     function tracks = deleteLostTracks(tracks)
    %     if isempty(tracks)

```

```

%         return;
%     end

invisibleForTooLong = 20;
ageThreshold = 8;

% Compute the fraction of the track's age for which it
was visible.
ages = [tracks(:).age];
totalVisibleCounts = [tracks(:).totalVisibleCount];
visibility = totalVisibleCounts ./ ages;

% Find the indices of 'lost' tracks.
lostInds = (ages < ageThreshold & visibility < 0.6) |
...
    [tracks(:).consecutiveInvisibleCount] >=
invisibleForTooLong;

% Delete lost tracks.
tracks = tracks(~lostInds);
% end

%     [tracks,nextId] =
createNewTracks(tracks,centroids,bboxes,unassignedDetection
s,nextId);
%     function [tracks,nextId] =
createNewTracks(tracks,centroids,bboxes,unassignedDetection
s,nextId)
    centroids = centroids(unassignedDetections, :);
    bboxes = bboxes(unassignedDetections, :);

for i = 1:size(centroids, 1)

    centroid = centroids(i,:);
    bbox = bboxes(i, :);

    % Create a Kalman filter object.
    kalmanFilter =
configureKalmanFilter('ConstantVelocity', ...
        centroid, [200, 50], [100, 25], 100);

% Create a new track.
newTrack = struct(...
    'id', nextId, ...
    'bbox', bbox, ...
    'kalmanFilter', kalmanFilter, ...
    'age', 1, ...
    'totalVisibleCount', 1, ...
    'consecutiveInvisibleCount', 0);

```

```

    % Add it to the array of tracks.
    tracks(end + 1) = newTrack;

    % Increment the next id.
    nextId = nextId + 1;
end
% end

%% Display Tracking Results
% The |displayTrackingResults| function draws a
bounding box and label ID
% for each track on the video frame and the foreground
mask. It then
% displays the frame and the mask in their respective
video players.

% Convert the frame and the mask to uint8 RGB.
frameRGB = im2uint8(frameRGB);
mask_2 = mask;
mask = uint8(repmat(mask, [1, 1, 3])) .* 255;

minVisibleCount = 8;
if ~isempty(tracks)

    % Noisy detections tend to result in short-lived
tracks.
    % Only display tracks that have been visible for
more than
    % a minimum number of frames.
    reliableTrackInds = ...
        [tracks(:).totalVisibleCount] >
minVisibleCount;
    reliableTracks = tracks(reliableTrackInds);

    % Display the objects. If an object has not been
detected
    % in this frame, display its predicted bounding
box.
    if ~isempty(reliableTracks)
        % Get bounding boxes.
        bboxes = cat(1, reliableTracks.bbox);

        % Get ids.
        ids = int32([reliableTracks(:).id]);

        % Create labels for objects indicating the ones
for

```

```

% which we display the predicted rather than
the actual
% location.

labels = cellstr(int2str(ids'));
predictedTrackInds = ...

[reliableTracks(:).consecutiveInvisibleCount] > 0;
isPredicted = cell(size(labels));
isPredicted(predictedTrackInds) = {'
predicted'};
labels = strcat(labels, isPredicted);

for j = 1 : size(bboxes,1)
    bx1 = max(1,bboxes(j,1));
    bx2 = max(1,bboxes(j,2));
    bx3 = bboxes(j,3);
    bx4 = bboxes(j,4);

    bx3 = min(bx1+bx3,size(frameRGB,2))-bx1-1;
    bx4 = min(bx2+bx4,size(frameRGB,1))-bx2-1;

    Or_box = orientation*180/pi;
    Or_box(Or_box<0) = Or_box(Or_box<0) + 360;
    Or_box(mask_2~=1) = nan;
%     Vy_box = Vy(bx2:bx2+bx4,bx1:bx1+bx3);
    or_box{j,nFrames} = Or_box;
%     vy_box{j,nFrames} = Vy_box;

% create HOOFF
    Or_box_encoded = ceil(Or_box/45);
    or_box_encoded{j,nFrames} = Or_box_encoded;

%     Or_box_encoded(Or_box_encoded==0) = 1;
    [counts,centres] =
hist(Or_box_encoded(:),8);
    [~,max_counts_idx] = max(counts);
    Orientation(nFrames,j) = max_counts_idx*45;
%     VY(j) = mean(Vy_box(:),'omitnan');
%     Orientation(nFrames) =
atan2(VY(j),VX(j))/pi*180;
%     if Orientation(nFrames) < 0
%         Orientation(nFrames) =
Orientation(nFrames) + 360;
%     end
    frameRGB =
insertObjectAnnotation(frameRGB,'rectangle',bboxes(j,:),[nu
m2str(Orientation(nFrames,j),'%.3f'),'
Derajat'],'FontSize',16,'Color',color{j});

```



```

%           VX_all(nFrames,j) = VX(j);
%           VY_all(nFrames,j) = VY(j);
counts_all(idx,:,j) = counts;
if nFrames == numFrames
    counts_all_2 = sum(counts_all);
    [~,avg_orientation(j)] =
max(counts_all_2);
%           if avg_orientation(j) < 0
%           avg_orientation(j) =
avg_orientation(j) + 360;
%           end

    frameRGB =
insertText(frameRGB,[size(frameRGB,2)/2-
80,size(frameRGB,1)/2],['Arah =
',num2str(45*avg_orientation(j),'%.3f'),'
Derajat'],'BoxColor',color{j},'FontSize',16);
end
end
frameRGB_all{nFrames} = frameRGB;
% Draw the objects on the frame.
%           frameRGB = insertObjectAnnotation(frameRGB,
'rectangle', ...
%           bboxes, labels);

% Draw the objects on the mask.
mask = insertObjectAnnotation(mask,
'rectangle', ...
    bboxes, labels);
idx = idx+1;
end
end

% Display the mask and the frame.
obj.maskPlayer.step(mask);
obj.videoPlayer.step(frameRGB);

nFrames = nFrames + 1;

end
%release video reader and writer
release(obj.maskPlayer);
release(obj.videoPlayer);
% end

```