

**RANCANG BANGUN KONTROL PERGERAKAN *PITCH* DAN *YAW*
PADA GIMBAL KAMERA UNTUK PELACAKAN OBJEK**

(Skripsi)

Oleh

MUHAMMAD ABDUL FATTAH

1715031053



**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS LAMPUNG
BANDAR LAMPUNG
2022**

ABSTRACT

DESIGN AND BUILD PITCH AND YAW MOVEMENT CONTROL ON CAMERA GIMBALS FOR OBJECT TRACKING

By:

MUHAMMAD ABDUL FATTAH

Unmanned Aerial Vehicle (UAV) is widely used in video capture or aerial photography. One part of the camera support on the body of the UAV is a camera gimbal. In this study, a physical model of the gimbal system have two-moving pitch and yaw that is based on the image parameters of the tracking object. The tracking process done by conditioning the object image to always be in the middle of the monitor screen.

This study analyzes the gradation between object color and background on the sensitivity of image processing using a kernelized correlation (KCF) filter. Observing the effect of the image scaling feature on the enlargement of the object image. Observed the performance of the PID control on the gimbal actuator. In addition, observing the percentage comparison of the object's dimension area with the tracking monitor box area.

From this research, a physical model of an object tracking system by a camera gimbal has been built with pitch and yaw motion control. The object color gradation with the background color must have a difference of 140 grayscale to get a better reading for processing images. Image scaling on the camera has the ability to enlarge the image object 177% from the minimum zoom size to the maximum zoom. Then value of the PID control on Y-axis of yaw motion are $K_P_X = 0.114$, $K_I_X = 0.141$, $K_D_X = 0.023$, and the X-axis of pitch movement with the value $K_P_Y = 0.108$, $K_I_Y = 0.112$, and $K_D_Y = 0.026$, produce a rise time response 1.26s with an average error 6 px. The tolerance limit for the object area to be tracked is 0.25% of the monitor tracking box size or if the object area is 720 pixels while the monitor tracking box area is 307200 pixels.

Keywords: Camera Gimbal, PID control, Kernel Correlation Filter (KCF), 2-Axis Movement.

ABSTRAK

RANCANG BANGUN KONTROL PERGERAKAN *PITCH* DAN *YAW* PADA GIMBAL KAMERA UNTUK PELACAKAN OBJEK

Oleh:

MUHAMMAD ABDUL FATTAH

Wahana terbang tanpa awak atau *Unmanned Aerial Vehichel* (UAV) banyak digunakan dalam pengambilan video atau foto udara. Salah satu bagian penyangga kamera di tubuh UAV adalah sebuah perangkat gimbal kamera. Dalam penelitian ini dibangun model fisik sistem gimbal kamera pergerakan dua arah *pitch* dan *yaw* berdasarkan parameter citra objek pelacakan. Proses pelacakan dilakukan dengan mengkondisikan citra objek selalu berada ditengah layar monitor.

Penelitian ini melakukan analisa gradasi antara warna objek dengan warna *background* terhadap sensitifitas pengolahan citra menggunakan filter *kernelized correlation* (KCF). Mengamati pengaruh fitur penskalaan citra terhadap pembesaran citra objek. Mengamati kinerja kontrol PID pada aktuator gimbal. Selain itu mengamati persentase perbandingan luas dimensi objek dengan luas kotak monitor pelacakan.

Dari penelitian ini telah terbangun sebuah model fisik sistem pelacakan objek oleh gimbal kamera dengan pengontrolan pergerakan *pitch* dan *yaw*. Gradasi warna objek dengan warna *background* harus memiliki perbedaan sebesar 140 skala *grayscale* untuk mendapatkan pembacaan yang baik dalam proses pelacakan. Penskalaan citra pada kamera mempunyai kemampuan memperbesar citra objek sebesar 177% dari ukuran *zoom* minimum ke *zoom* maksimum. Kemudian pelacakan objek pada sumbu Y gerak *yaw* dengan nilai konstanta PID ($K_P_X = 0.114$, $K_I_X = 0.141$, $K_D_X = 0.023$) dihasilkan respon gimbal dengan *rise time* sebesar 1,26s dengan nilai *error* rata-rata sebesar 6 px, pada sumbu X gerak *pitch* dengan nilai konstanta PID ($K_P_Y = 0.108$, $K_I_Y = 0.112$, $K_D_Y = 0.026$). Batas toleransi luas objek agar dapat dilacak dengan pelacakan *kernelized correlation filter* (KCF) pada gimbal kamera adalah sebesar 0.25 % dari besar kotak monitor pelacakan atau jika luas objek adalah 720 piksel sedangkan luas kotak monitor pelacakan adalah 307200 piksel.

Kata Kunci: Gimbal Kamera, Kontrol PID, *Kernelized Correlation Filter* (KCF), 2-Sumbu Pergerakan.

**RANCANG BANGUN KONTROL PERGERAKAN *PITCH* DAN *YAW*
PADA GIMBAL KAMERA UNTUK PELACAKAN OBJEK**

Oleh

MUHAMMAD ABDUL FATTAH

Skripsi

**Sebagai Salah Satu Syarat untuk Mencapai Gelar
SARJANA TEKNIK**

pada

**Jurusan Teknik Elektro
Fakultas Teknik Univeritas Lampung**



JURUSAN TEKNIK ELEKTRO

FAKULTAS TEKNIK

UNIVERSITAS LAMPUNG

2022

Judul Skripsi

**: RANCANG BANGUN KONTROL
PERGERAKAN *PITCH* DAN *YAW* PADA
GIMBAL KAMERA UNTUK PELACAKAN
OBJEK**

Nama Mahasiswa

: *Muhammad Abdul Fattah*

Nomor Pokok Mahasiswa

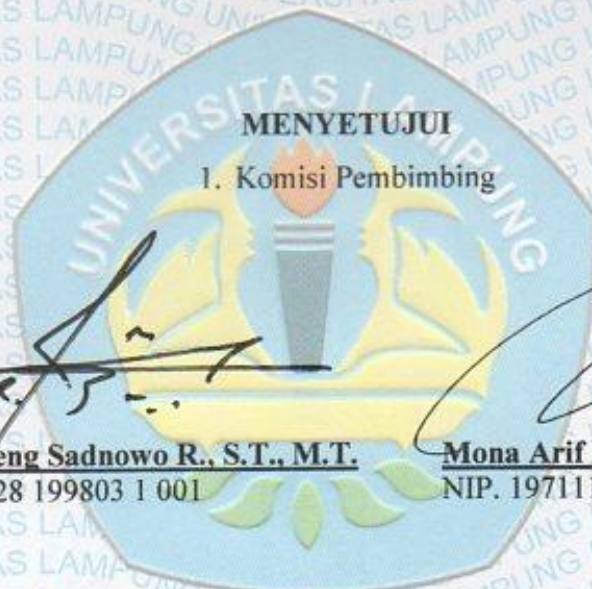
: 1715031053

Jurusan

: Teknik Elektro

Fakultas

: Teknik



Dr. Eng. Ageng Sadnowo R., S.T., M.T.
NIP. 19690228 199803 1 001

Mona Arif Muda, S.T., M.T.
NIP. 19711112 200003 1 002

2. Mengetahui

Ketua Jurusan
Teknik Elektro

Ketua Program Studi
Teknik Elektro

Herlinawati, S.T., M.T.
NIP. 19710314 199903 2 001

Dr. Eng. Nining Purwasih, S.T., M.T.
NIP. 19740422 200012 2 001

MENGESAHKAN

I. Tim Penguji

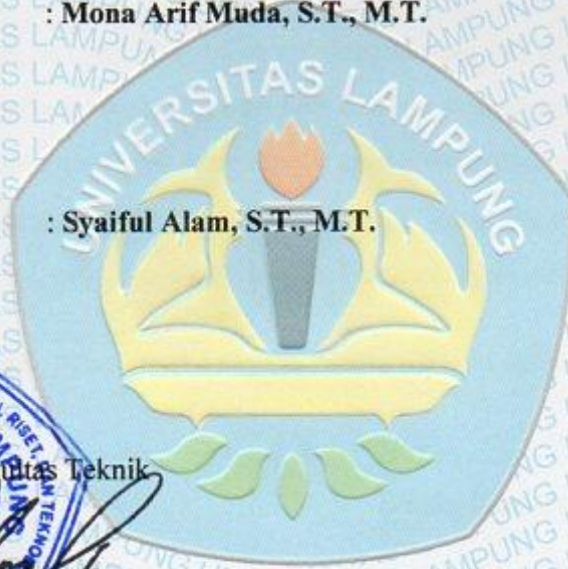
Ketua : Dr. Eng. Ageng Sadnowo R., S.T., M.T.



Sekretaris : Mona Arif Muda, S.T., M.T.



Anggota : Syaiful Alam, S.T., M.T.



Dekan Fakultas Teknik

Dr. Eng. Ir. Helmy Fitriawan, S.T., M.Sc.

NIP. 19750928 200112 1 002

Tanggal Lulus Ujian Skripsi : 13 Juni 2022

SURAT PERNYATAAN

Yang bertanda tangan di bawah ini:

Nama : Muhammad Abdul Fattah

NPM : 1715031053

Dengan ini saya menyatakan bahwa dalam skripsi ini dibuat tidak ada terdapat karya yang pernah dilakukan orang lain dan sepanjang sepengetahuan saya tidak terdapat karya atau pendapat yang ditulis atau diterbitkan orang lain, kecuali secara tertulis diacu dalam naskah ini sebagaimana yang disebutkan di dalam daftar pustaka. Selain itu, saya menyatakan pula bahwa skripsi ini dibuat oleh saya sendiri.

Apabila saya tidak benar, maka saya bersedia dikenai sanksi sesuai dengan hukum yang berlaku.

Bandarlampung, 20 Juni 2022



Muhammad Abdul Fattah
NPM. 1715031053

RIWAYAT HIDUP



Penulis dilahirkan di Seputih Banyak, Lampung Tengah, Lampung, 21 Oktober 1998. Penulis merupakan anak pertama dari dua bersaudara dari pasangan Bapak Iswadi Idris dan Ibu Suciati.

Penulis memulai pendidikan di TK Ma'arif NU pada tahun 2003-2005, SDN 1 Tanjung Harapan pada tahun 2004-2010, SMPN 1 Seputih Banyak pada tahun 2011-2014, dan SMAN 1 Seputih Banyak pada tahun 2014-2017. Penulis menjadi mahasiswa Jurusan Teknik Elektro, Universitas Lampung pada tahun 2017 melalui jalur SBMPTN (Seleksi Bersama Masuk Perguruan Tinggi Negeri) dan mendapatkan beasiswa BIDIKMISI. Selama menjadi mahasiswa, penulis pernah menjadi anggota Himpunan Mahasiswa Teknik Elektro (HIMATRO) sebagai anggota Departemen Pendidikan dan Pengembangan Diri Divisi Pendidikan, pada periode 2018 dan Kepala Divisi Penelitian dan Pengembangan Departemen Pengembangan Keteknikan HIMATRO Unila pada periode 2019. Penulis juga mengikuti lembaga riset Unila Robotika dan Otomasi (URO) sejak 2017 sebagai anggota dan sebagai ketua divisi *Vertical Take Off Landing (VTOL)* URO Unila pada 2019. Pada tanggal 1 Juli 2020 – 10 Agustus 2020, penulis melaksanakan kerja praktik di Pusat Teknologi dan Industri Pertahanan Keamanan (PTIPK) – Badan Penelitian dan Pengembangan Teknologi (BPPT) Puspitek, Buaran, Kec. Serpong, Kota Tangerang Selatan, Banten dengan mengangkat judul “Penggunaan Kamera Hiperspektral *Corning Microshi 410 Shark* dan *Software ENVI (Normalized Difference Vegetation Index)*”.

PERSEMBAHAN

Kupersembahkan karya kecil ini dengan rasa hormat, cinta dan kasih sayang, kepada:

Bapak dan Ibu yang terkasih

ISWADI IDRIS dan SUCIATI

Serta

Saudari:

LUTIFFI DZAKIYYAH

Yang telah menjadi motivasi dan inspirasi serta tiada hentinya memberikan dukungan dan do'a-nya:

HIRO 2017

HIMATRO UNILA

Unila Robotika & Otomasi

Terima kasih atas kekeluargaan, do'a, serta dukungan yang selalu mengiringi hingga merasa lebih baik dari hari ke hari

MOTTO HIDUP

“Maka sesungguhnya bersama kesulitan ada kemudahan,”

(QS. Al-Insyirah 94: ayat 5)

“ dan hanya kepada Tuhanmulah engkau berharap,”

(QS. Al-Insyirah 94: ayat 8)

“Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum,
sebelum mereka mengubah keadaan diri mereka sendiri”

(QS. Ar Rad: ayat 11)

“Diwajibkan atas kamu berperang, padahal itu tidak menyenangkan bagimu. Tetapi boleh jadi kamu tidak menyenangi sesuatu, padahal itu baik bagimu, dan boleh jadi kamu menyukai sesuatu, padahal itu tidak baik bagimu. Allah mengetahui, sedang kamu tidak mengetahui.”

(QS. Al-Baqarah : ayat 216)

“ Sesuatu itu mudah sekali kalau dianalisis dengan ilmu”

(KH. Ahmad Bahauddin Nursalim (Gus Baha))

“Life needs more than solving everyday problems. You need to be
awake and excited about the future”

(Elon Musk - CEO SpaceX, TESLA, Starlink)

“A person who is happy is not because everything is right in his life,
He is happy because of his attitude towards everything in his life is
right”

(Sundar Pichai – CEO Google & Alphabet)

SANWACANA

Bismillahirrahmanirrahim.

Alhamdulillah rabbil'alamin. Puji syukur kehadiran Allah SWT yang telah memberikan rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan penelitian ini.

Skripsi dengan judul “RANCANG BANGUN KONTROL PERGERAKAN *PITCH* DAN *YAW* PADA GIMBAL KAMERA UNTUK PELACAKAN OBJEK ” yang merupakan salah satu syarat untuk memperoleh gelar Sarjana Teknik pada Jurusan Teknik Elektro, Fakultas Teknik, Universitas Lampung. Dalam penulisan skripsi ini, penulis mendapatkan bantuan dan dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu pada kesempatan ini menyampaikan rasa terimakasih kepada:

1. Bapak Prof. Dr. Karomani, M.Si. selaku Rektor Universitas Lampung.
2. Bapak Dr. Eng., Helmy Fitriawan, S.T., M.Sc. selaku Dekan Fakultas Teknik Universitas Lampung.
3. Ibu Herlina, S.T., M.T. selaku Ketua Jurusan Teknik Elektro Universitas Lampung.
4. Unila Robotika Otomasi (URO) dengan pimpinan Bapak Dr.-ing. Ardian Ulvan, S.T., M.Sc. yang telah membantu dalam memberikan ide dan fasilitas dalam penyelesaian pelaksanaan penelitian ini.
5. Bapak Dr. Eng. Ageng Sadnowo R., S.T., M.T. selaku Dosen Pembimbing Utama yang telah memberikan ilmu, bimbingan, bantuan, arahan, masukan, motivasi, dan pandangan kehidupan kepada penulis disetiap kesempatan dengan baik dan ramah.
6. Bapak Mona Arif Muda, S.T., M.T. selaku Dosen Pembimbing Pendamping yang telah memberikan ilmu, bimbingan, bantuan, arahan, masukan, motivasi, dan pandangan kehidupan kepada penulis disetiap kesempatan dengan baik dan ramah.
7. Bapak Syaiful Alam, S.T., M.T. selaku Dosen Penguji yang telah memberikan kritik dan saran yang membangun kepada penulis.

8. Bapak Herri Gusmedi, S.T., M.T. selaku pembimbing akademik dari semester 1 hingga sampai sekarang yang telah memberikan arahan, nasehat, dan bimbingan yang membangun bagi penulis dalam masa perkuliahan.
9. Segenap dosen di Jurusan Teknik Elektro yang telah memberikan ilmu yang bermanfaat kepada penulis selama menempuh pendidikan perkuliahan.
10. Segenap staff di Jurusan Teknik Elektro yang telah membantu penulis baik dalam hal administrasi dan lain-lain.
11. Saudaraku Ahmad Misbah Khoeroni, Mas Iwan dan Pak Teguh sekeluarga yang telah memberikan tempat bernaung serta keramahannya yang telah memberi semangat dan bantuan untuk mengerjakan skripsi ini.
12. Keluarga Besar Unila Robotika dan Otomasi, Bapak Mona Arif Muda, S.T., M.T., Bapak Dr.-ing. Ardian Ulvan, S.T., M.Sc., Ibu Dr. ing. Melvi S.T., M.T., Mas Aris, Kak Dana, Kak Nora, Mas Isol, Mas Anggi, Mbak Nur, Kak Dede, Kak Reza, Kak Ocid, Kak Malik, Mbak Fathimah, Ade, dan Ikhsanuddin, Iqbal, Muhi, Hardiawan, Bobby, Manda, Abdul, Fikri, Lutfi, Hilmi, Ranto, Juan, Rio, Rafif yang memberikan banyak ilmu, masukan dan saran kepada penulis yang tidak bisa dibayarkan.
13. Lelingku Teknik Elektro'17 "HIRO17", terimakasih atas semangat dan kebersamaannya selama menempuh pendidikan di Jurusan Teknik Elektro.
14. Keluarga Besar HIMATRO yang telah memberikan ilmu dan pengalaman kepada penulis selama pendidikan baik secara langsung maupun tak langsung.
15. Teman-teman KKN desa Sri Budaya : Meta, Indri, Dimas, Riyan, Awal, Rizky.
16. Terimakasih juga kepada **KAMU** yang selalu memberi semangat dan motivasi tapi tak bisa ku sebutkan namanya, namun selalu ku bawa dalam do'a.
17. Seluruh pihak yang telah membantu penulis menyelesaikan skripsi ini.

Penulis menyadari bahwa masih banyak kekurangan dalam penulisan skripsi ini. Penulis mengharapkan adanya kritik dan saran yang membangun dari berbagai pihak demi kemajuan bersama. Semoga skripsi ini dapat bermanfaat bagi penulis dan pembaca.

Alhamdulillahil'alamin

DAFTAR ISI

	Halaman
ABSTRACT	ii
ABSTRAK	iii
LEMBAR PERSETUJUAN	v
LEMBAR PENGESAHAN	vi
SURAT PERNYATAAN	vi
RIWAYAT HIDUP	viii
PERSEMBAHAN	ix
MOTTO HIDUP	x
SANWACANA	xii
DAFTAR ISI	xiii
DAFTAR GAMBAR	xiv
DAFTAR TABEL	xvi
I.PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Tujuan Penelitian.....	2
1.3. Rumusan Masalah	2
1.4. Batasan Masalah.....	2
1.5. Manfaat Penelitian.....	3
1.6. Hipotesis.....	3
1.7. Sistematika Penulisan.....	3
II.TINJAUAN PUSTAKA	4
2.1. Penelitian Terdahulu	4
2.2. Gimbal Kamera	5

2.3. Kontrol Kalang Tertutup	7
2.4. Penskalaan Citra pada Kamera.....	9
2.5. Pengolahan Citra	9
2.5.1. Citra <i>Grayscale</i>	10
2.5.2. <i>Region Of Interest</i> (ROI)	11
2.5.3. <i>Kernelized Correlation Filter</i> (KCF).....	12
2.5.4. Lokalisasi Objek pada Citra.....	13
III.METODOLOGI PENELITIAN	15
3.1. Waktu dan Tempat Penelitian	15
3.2. Komponen dan Perangkat Lunak	15
3.3. Skenario Pelacakan Objek dengan Gimbal Kamera.....	16
3.4. Rancangan Sistem	16
3.5. Diagram Alir Sistem.....	18
3.6. Diagram Blok Sistem dan <i>Listing</i> Program.....	19
3.6.1. Pergerakan Manual	20
3.6.2. Pergerakan Otomatis.....	22
3.7. Desain Perancangan Alat.....	37
IV.HASIL DAN PEMBAHASAN	39
4.1. Pengaturan Pengujian	39
4.1.1. Pengaturan Pengujian Kamera yang Terpasang Pada Gimbal Kamera.....	39
4.1.2. Pengaturan Pengujian Batas Toleransi Warna Antara Warna Objek Dengan Warna <i>Background</i> Pada <i>Kernelized Correlation</i> <i>Filter</i> (KCF)	40
4.1.3. Pengaturan Pengujian Penskalaan Citra Pada Kamera yang Terpasang di Gimbal Kamera	41
4.1.4. Pengaturan Pengujian Kontrol PID Gimbal Kamera pada Pergerakan Otomatis.....	42
4.1.5. Pengaturan Pengujian Pengaruh Besar Objek Terhadap Layar Pelacakan <i>Kernelized Correlation Filter</i> (KCF)	45
4.2. Data Hasil	45
4.2.1. Analisis Kamera yang Terpasang pada Gimbal Kamera	45

4.2.2. Analisis Batas Toleransi Warna antara Warna Objek dengan Warna <i>Background</i> pada <i>Kernelized Correlation Filter</i> (KCF).....	46
4.2.3. Analisis Pengujian Penskalaan Citra Pada Kamera yang Terpasang Pada Gimbal Kamera.	48
4.2.4. Analisis Pengujian Kontrol PID Gimbal Kamera pada Pergerakan Otomatis.....	52
4.2.5. Analisis Pengujian Pengaruh Besar Objek Terhadap Layar Pelacakan <i>Kernelized Correlation Filter</i> (KCF)	54
V.PENUTUP	56
5.1. Kesimpulan.....	56
5.2. Saran.....	56
DAFTAR PUSTAKA	58

DAFTAR GAMBAR

Gambar	Halaman
2.1. Gimbal Kamera Dengan Pergerakan 2 Sumbu.	6
2.2. Sinyal <i>Pulse Width Modulation</i> (PWM).	7
2.3. Blok Diagram Kontrol PID.	8
2.4. (A) Citra RGB, (B) Citra <i>grayscale</i>	11
2.5. Contoh Pemilihan Objek Menggunakan ROI.	11
2.6. Sistem Koordinat Kartesian.	14
2.7. Perbandingan Koordinat Kartesian Dengan Koordinat Layar Monitor Pelacakan.	14
3.1. Konsep Penelitian Pada Pengontrol Gimbal Kamera.	17
3.2. Diagram Alir Sistem Pelacakan Gimbal Kamera.....	18
3.3. Diagram Blok Sistem Gimbal Kamera.	19
3.4. Diagram Alir Pengambilan Citra Video.....	23
3.5. Visuali Pelacakan Pada Layar Monitor Pelacakan.....	24
3.6. Ilustrasi Visualisasi Layar Monitor Pelacakan.....	25
3.7. Visualisasi Layar Monitor Pelacakan Pada Citra Video Yang Diambil.	26
3.8. (A) Pergerakan Aktuator Pada Sumbu Y, (B) Pergerakan Aktuator Pada Sumbu X.	29
3.9. Ilustrasi Letak Kotak ROI Visualisasi Layar Monitor Pelacakan.....	30
3.10. Diagram Alir Pergerakan Aktuator A0 Gerak <i>Yaw</i> Gimbal Kamera.....	31
3.11. Diagram Alir Pergerakan Aktuator A1 Gerak <i>Pitch</i> Gimbal Kamera.	32
3.12. Skenario Letak Koordinat Pixel Kotak ROI Pada Layar Monitor.....	33
3.13. Rangkaian Skematik Model Fisik Gimbal Kamera	37
3.14. Model Fisik Gimbal Kamera.....	38
4.1. Membuka Program Melalui Terminal Pada Raspberry Pi 4.	39
4.2. Kamera Yang Terpasang Pada Gimbal Kamera.	40

4.3. Membuat kotak ROI.....	40
4.4. Tombol Pada <i>Keyboard</i> Untuk Mengatur Penskalaan Citra.....	41
4.5. Indikator Nilai Penskalaan Citra Pada Kamera.....	42
4.6. Grafik Respon Letak Objek Terhadap Gerak <i>Yaw</i>	43
4.7. Grafik Respon Letak Objek Terhadap Gerak <i>Pitch</i>	43
4.8. Tampilan Pelacakan Pada Monitor Operator.....	46
4.9. Hasil Pengujian Warna Antara Warna Objek Dengan Warna <i>Background</i> ...	47
4.10. Hasil Pengujian Pembesaran Objek Pada Perubahan Penskalaan Citra.....	50
4.11. Grafik Hubungan Nilai <i>Zoom</i> Step Dengan Pembesaran Objek.....	51
4.12. Grafik Hubungan Nilai Fokus Step Dengan Pembesaran Objek.....	51
4.13. Grafik Respon Letak Objek Terhadap Gerak <i>Yaw</i>	52
4.14. Grafik Respon Letak Objek Terhadap Gerak <i>Pitch</i>	53
4.15. Objek Berada Di Dalam Kotak Pelacakan ROI.....	54
4.16. Objek Terlepas Atau Berada Di Luar Kotak Pelacakan ROI.....	55

DAFTAR TABEL

Tabel	Halaman
3.1. Komponen dan Perangkat Lunak.....	15
3.2. Tombol Pergerakan Gimbal Manual.....	23
3.3. Tombol Penskalaan Citra Pada <i>Keyboard</i>	25
3.4. Pergerakan Aktuator Berdasarkan Letak Titik Koordinat Titik Tengah Kotak ROI Pelacakan.....	30
4.1. <i>Tuning Ziegler-Nichols</i> Berdasarkan Penguatan Kritisal (Kcr) dan Periode Kritisal (Pcr).....	44
4.2. Data Hasil Toleransi Warna Objek Dengan Warna Background Dalam Keadaan Pelacakan Objek.....	47
4.3. Data Hasil Pengujian Penskalaan Citra Pada Kamera Model Fisik Gimbal Kamera.....	50

I. PENDAHULUAN

1.1. Latar Belakang

Teknologi wahana terbang tanpa awak yang dikenal dengan sebutan *Unmanned Aerial Vehichel* (UAV) atau *drone*. UAV didefinisikan sebagai pesawat terbang tanpa awak yang dikontrol dari jarak jauh atau dikontrol secara autopilot. Penggunaan UAV banyak digunakan dalam pengambilan video atau foto udara [1]. Salah satu bagian penting untuk menyangga kamera di tubuh UAV dalam operasinya untuk melakukan pengambilan video atau foto dan melakukan proses pelacakan dibutuhkan sebuah perangkat gimbal. Untuk meningkatkan kemampuan gimbal kamera ditambahkan sebuah kontrol yang masukannya berupa pengolahan citra pelacakan objek yang memiliki fungsi utama membidik objek yang ditentukan sehingga objek itu selalu berada di tengah layar monitor video yang memudahkan operator kamera gimbal untuk mengamati objek secara terus-menerus meskipun UAV dalam keadaan bergerak.

Universitas Lampung dengan pusat unggulan yaitu Unila Robotika & Otomasi (URO) terus mengembangkan riset dalam bidang UAV khususnya dalam pengambilan video monitoring dari udara, sehingga pada penelitian ini dibuat sebuah model fisik sistem gimbal kamera dengan pergerakan dua arah kebebasan *pitch* dan *yaw*. Pergerakan gimbal didukung oleh dua buah aktuator untuk merespon pergerakan gimbal yang dikontrol berdasarkan pada parameter pengolahan citra berupa pelacakan objek. Penelitian ini menghasilkan alat berupa sebuah sistem gimbal kamera yang dapat mempertahankan objek yang ditentukan tetap berada pada tengah layar monitor video.

1.2. Tujuan Penelitian

Adapun tujuan penelitian ini adalah sebagai berikut.

1. Membuat model fisik gimbal kamera dengan pergerakan 2 sumbu yaitu sumbu Y gerak *yaw* dan sumbu X gerak *pitch*.
2. Menganalisis batas toleransi warna antara warna objek dengan warna *background* pada pelacakan *kernelized correlation filter* (KCF).
3. Menganalisis fitur penskalaan citra kamera yang terpasang pada model fisik gimbal kamera.
4. Menganalisis kontrol PID gimbal kamera pada pergerakan otomatis.
5. Menganalisis pengaruh besar objek terhadap layar pelacakan pada pelacakan *kernelized correlation filter* (KCF) di gimbal kamera.

1.3. Rumusan Masalah

Adapun rumusan masalah pada penelitian ini adalah sebagai berikut.

1. Bagaimana membuat pengolahan citra dengan proses pelacakan objek pada *library OpenCV* dari kamera yang bergerak.
2. Bagaimana merancang pergerakan gimbal kamera yang mempunyai 2-sumbu gerak yaitu sumbu X dan sumbu Y agar mempunyai sudut pandang kamera yang mencakup daerah seperti setengah kulit bola.
3. Bagaimana menggabungkan proses pengolahan citra sebagai masukan kontrol gimbal kamera dalam proses pelacakan objek.

1.4. Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah sebagai berikut.

1. Pengolahan citra menggunakan paket yang disediakan oleh *library OpenCV* yaitu *kernelized correlation filter* (KCF).
2. Objek yang dilacak ditentukan secara manual oleh operator.
3. Gimbal dirancang bergerak dalam 2-sumbu, yaitu sumbu X untuk gerak *pitch* dan sumbu Y untuk gerak *yaw*.
4. Gerak aktuator pada sumbu X sebesar 180° ke atas dan ke bawah serta gerak aktuator pada sumbu Y sebesar 180° ke kanan dan ke kiri.

1.5. Manfaat Penelitian

Adapun manfaat penelitian ini adalah menghasilkannya model fisik gimbal kamera yang memudahkan proses pengambilan data citra udara menggunakan UAV. Citra dapat dikoleksi secara terus-menerus dalam radius terbang lintasan melingkar UAV pada saat terbang.

1.6. Hipotesis

Gimbal kamera dapat mempertahankan fokus pada sudut pandang kamera yang selalu mengarah pada objek yang telah ditentukan dengan melakukan aktifitas pergerakan gimbal kamera dalam area yang mencakup hingga setengah kulit bola.

1.7. Sistematika Penulisan

BAB I – PENDAHULUAN

Menjelaskan tentang latar belakang, tujuan penelitian, rumusan masalah, batasan masalah, manfaat penelitian, hipotesis, dan sistematika penulisan laporan tugas akhir.

BAB II – TINJAUAN PUSTAKA

Menjelaskan tinjauan pustaka secara teoretis mengenai landasan dalam penelitian ini dan berisi literatur penelitian terdahulu.

BAB III – METODOLOGI PENELITIAN

Menjelaskan waktu dan tempat penelitian, alat dan bahan, serta langkah-langkah pelaksanaan penelitian.

BAB IV – HASIL DAN PEMBAHASAN

Menjelaskan perancangan dan analisis dari hasil pengujian.

BAB V – KESIMPULAN DAN SARAN

Menjelaskan kesimpulan dari penelitian yang dilakukan serta saran-saran mengenai perbaikan dan pengembangan lebih lanjut agar didapatkan hasil lebih baik.

DAFTAR PUSTAKA

II. TINJAUAN PUSTAKA

2.1. Penelitian Terdahulu

Referensi pertama pada penelitian ini berjudul “*Gimbal Control for Vision-based Target Tracking*” yang dilakukan pada tahun 2019. Penelitian ini membahas masalah pengendalian orientasi gimbal 3-sumbu yang membawa kamera, penggunaan pengukuran gambar dari citra kamera sebagai *feedback* pengendalian gimbal kamera sehingga objek yang ditentukan tetap berada di tengah bidang gambar, untuk melakukan deteksi dan pelacakan objek visual menggunakan metode *Convolutional Neural Networks* (CNNs). Untuk mendapatkan performa *real-time* dengan perangkat keras yang komputasinya terbatas menggunakan detektor *neural* dalam satu tahap yang disusun seperti *Single-Shot Detector* (SSD) dan *You Only Look Once* (YOLO) secara bersamaan menurunkan koordinat piksel dari *Region Of Interest* (ROI) objek dan menetapkan label kelas sehingga dapat menentukan posisi baru dari objek yang ditentukan, pada sistem pengontrolan gimbal menggunakan sistem kontrol *loop* luar, yang menerima pengukuran gambar dan menghitung perintah kecepatan sudut untuk memperbaiki perpindahan gambar [2].

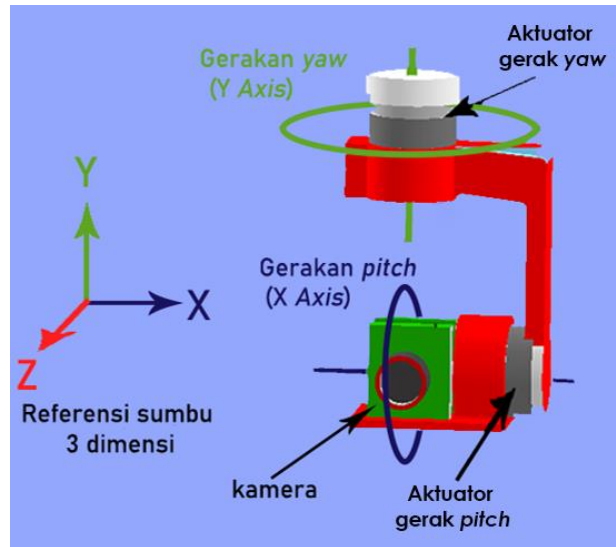
Referensi kedua adalah *paper* berjudul “*Real-Time Visual Tracking of Moving Targets Using a Low-Cost Unmanned Aerial Vehicle with a 3-Axis Stabilized Gimbal System*” yang dilakukan tahun 2020, membahas sistem berbasis visi *onboard* untuk UAV berbiaya rendah yang secara mandiri melacak target bergerak. Pelacakan visual *real-time* dengan menggunakan algoritma deteksi objek berdasarkan metode *Kernelized Correlation Filter* (KCF). Gimbal kamera 3-sumbu dengan *Inertial Measurement Unit* (IMU) terpisah digunakan untuk membidik target yang dipilih selama penerbangan. Algoritma kontrol penerbangan untuk tugas pelacakan diimplementasikan pada *quadrotor* khusus yang dilengkapi dengan komputer *onboard* dan *mikrokontroler* [3].

Referensi ketiga adalah *paper* berjudul “Pemodelan dan Perancangan Sistem Kendali Gimbal Kamera 3-Sumbu Menggunakan Metode *Proportional, Integral, Derivative* (PID) pada *Unmanned Aerial Vehicle* (UAV)”, yang dilakukan tahun 2020. Membahas pemodelan dan perancangan gimbal kamera dengan kendali 3-sumbu berdasarkan hasil simulasi kekuatan rangka menggunakan perangkat lunak desain dan dicetak dengan mesin 3D *printing* serta motor BLDC sebagai aktuator disetiap sumbu. Metode pengontrolan gimbal menggunakan aturan *Ziegler-Nichols* kedua. Hasil pengujian sistem kendali gimbal kamera 3-sumbu menunjukkan nilai kendali PID baru menghasilkan respon yang lebih baik dalam respon peralihannya dibandingkan dengan kendali PID *default* [4].

Gagasan modifikasi pada penelitian mempunyai konsep sama dengan referensi pertama yang membedakan adalah penggunaan divais alat serta metode palacakan objek yang berbeda, pada penelitian ini menggunakan pelacakan objek metode *kernelized Correlation Filter* (KCF) yang mempunyai komputasinya pengolahan data yang lebih kecil dibandingkan dengan metode *Convolutional Neural Networks* (CNNs). Untuk referensi kedua menggunakan metode pelacakan yang sama tetapi penelitian ini adalah hanya membahas gimbal kamera tidak dengan pengontrolan *quadrotor* serta penambahan *keyboard* untuk menggerakkan gimbal secara manual. Untuk referensi ketiga perbedaan berupa konsep pengontrolan gimbal kamera yaitu menggunakan aturan kedua *Ziegler-Nichols*. Sedangkan pada penelitian ini menggunakan pengontrolan gimbal dengan masukan pengolahan citra pelacakan objek.

2.2. Gimbal Kamera

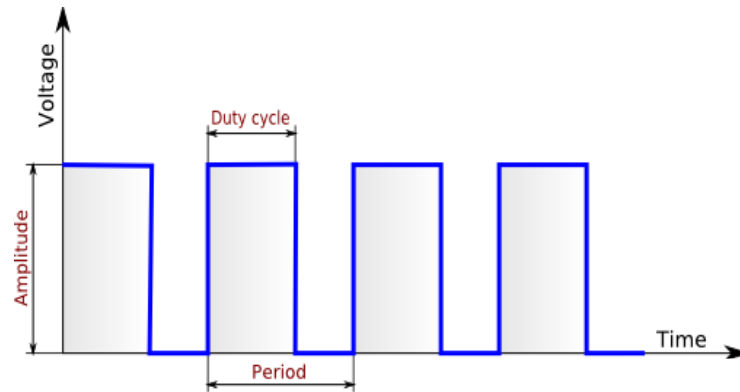
Gimbal kamera adalah perangkat mekanis yang mengkondisikan kamera pada posisi stabil saat pengambilan objek gambar dengan respon pengaturan gerak rotasi aktuator dari lengan-lengan gimbal untuk menghasilkan objek gambar yang stabil. Stabilisasi video mengacu pada algoritma yang digunakan untuk meningkatkan kualitas video dengan menghilangkan guncangan dan getaran kamera yang tidak diinginkan karena guncangan dan bidikan kamera yang tidak disengaja [5], gambar 2.1. merupakan contoh gambar rangka gimbal kamera yang dirancang.



Gambar 2.1. Gimbal Kamera Dengan Pergerakan 2 Sumbu.

Pada gambar 2.1. merupakan gimbal kamera mempunyai 2 sumbu pergerakan, yaitu pada poros sumbu X yang disebut *pitch* dan yang bergerak pada sumbu Y yang disebut *yaw*. Gerakan *yaw* adalah gerakan menyamping dari sisi gimbal kamera ke kiri atau ke kanan. Gerakan *pitch* adalah gerakan ke atas dan ke bawah dari gimbal kamera [6].

Aktuator adalah sebuah alat mekanis yang mengubah tenaga listrik menjadi kuantitas lain sehingga mampu menghasilkan energi kinetik. Energi kinetik yang dihasilkan digunakan untuk menggerakkan atau mengontrol sistem gimbal sehingga aktuator dapat bergerak secara berotasi [7]. Dalam menggerakkan aktuator sinyal yang dikirim dari kontroler adalah sinyal *Pulse Width Modulation* (PWM). PWM secara umum adalah sebuah cara memanipulasi lebar sinyal yang dinyatakan dengan pulsa dalam satu periode, untuk mendapatkan tegangan rata-rata yang berbeda [8]. *Duty cycle* adalah perbandingan lama waktu suatu sinyal berada dalam kondisi *high* dengan lama waktu suatu sinyal tersebut dalam kondisi *low* [8]. Gambar sinyal PWM seperti pada gambar 2.2. berikut.



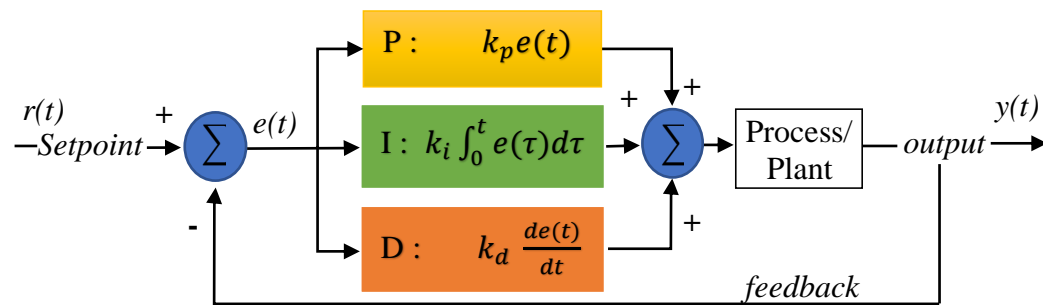
Gambar 2.2. Sinyal *Pulse Width Modulation* (PWM).

2.3. Kontrol Kalang Tertutup

Sistem kontrol adalah suatu susunan komponen fisik yang terhubung atau terkait sedemikian rupa sehingga dapat memerintah, mengarahkan, atau mengatur sistem sendiri atau sistem lain. Sistem kontrol kalang terbuka (*open loop control*) adalah sistem pengontrolan dimana besaran keluaran tidak memberikan efek terhadap besaran masukan, sehingga variabel yang dikontrol tidak dapat dibandingkan terhadap variabel yang diinginkan [9].

Kontrol kalang tertutup (*close loop control*) adalah sistem pengontrolan dimana besaran keluaran memberikan efek terhadap besaran masukan, sehingga besaran yang dikontrol dapat dibandingkan terhadap variabel yang diinginkan. Selanjutnya, perbedaan variabel yang terjadi antara besaran yang dikontrol dengan variabel yang diinginkan digunakan sebagai koreksi yang merupakan sasaran pengontrolan. Kontrol kalang tertutup juga merupakan sistem yang mempertahankan posisi atau keadaan yang diinginkan meskipun terjadi perubahan pada sistem maka akan mengeluarkan umpan balik (*feedback*) untuk melakukan langkah awal pengaturan sehingga sistem kembali bekerja pada keadaan yang diinginkan [9].

Persamaan kontrol PID. Proporsional, Integral, Derivatif (PID) adalah sebuah sistem kontrol untuk menentukan presisi suatu sistem instrumentasi dengan karakteristik adanya umpan balik (*feedback*) pada sistem. Sistem kontrol kalang tertutup seperti pada gambar 2.3. berikut.



Gambar 2.3. Blok Diagram Kontrol PID.

Proporsional adalah komponen yang membandingkan nilai antara *setpoint* dan variabel proses. Perbedaan kedua hal tersebut disebut *error*. Kontrol proporsional adalah kontrol yang memiliki keluaran sebanding dengan masukan *error* nya, secara sederhana bahwa keluaran kontrol proporsional adalah perkalian antara masukan yang berupa *error* dengan konstanta proporsional [9]. Berikut merupakan persamaan untuk mencari nilai proporsional ($P(t)$).

$$P(t) = k_p e(t) \quad (2.1)$$

Keterangan:

$P(t)$ = nilai keluaran kontrol proporsional relatif terhadap waktu (t).

k_p = konstanta proporsional.

$e(t)$ = nilai *error*.

Kontrol integral berfungsi untuk menghilangkan *error steady state* yang tidak bisa dihilangkan oleh kontrol proporsional. *Error steady state* adalah perbedaan jarak dimana beban berhenti dan posisi *setpoint* (posisi yang diinginkan). Keluaran dari kontrol ini adalah menjumlahkan terus menerus dari masuknya yang berupa *error*. Jika sinyal *error* pada masukannya tidak mengalami perubahan, maka keluaran akan menjaga keadaan seperti sebelum terjadinya perubahan masukan [9]. Berikut merupakan persamaan untuk mencari nilai integral ($I(t)$).

$$I(t) = k_i \int_0^t e(\tau) d\tau \quad (2.2)$$

Keterangan:

$I(t)$ = nilai keluaran kontrol integral relatif terhadap waktu (t).

k_i = konstanta Integral.

$e(\tau)$ = nilai *error*.

Derivatif merupakan komponen yang berguna untuk memprediksi besar *error* di masa mendatang. Respon dari derivatif bergantung pada perubahan dari *error* pada setiap waktu yang dikalikan dengan nilai konstan derivatif. Kontrol derivatif adalah kontrol yang memiliki sifat seperti operasi derivatif, perubahan mendadak pada masukan kontrol akan mengakibatkan perubahan yang sangat besar dan cepat. Kontrol derivatif hanya berubah nilai keluarannya ketika ada perubahan *error* sehingga ketika *error* tidak ada perubahan maka praktis kontrol ini tidak akan bereaksi [9]. Berikut merupakan persamaan untuk mencari nilai derivatif ($D(t)$).

$$D(t) = k_d \frac{de(t)}{dt} \quad (2.3)$$

Keterangan:

$D(t)$ = nilai keluaran kontrol derivatif relatif terhadap waktu (t).

k_d = konstanta derivatif.

$e(t)$ = nilai *error*.

Fungsi dari sistem kontrol PID pada penelitian ini adalah untuk merespon cepat dengan presisi yang lebih baik.

2.4. Penskalaan Citra pada Kamera

Penskalaan citra merupakan proses yang dilakukan untuk mengubah resolusi pada citra yang kecil menjadi lebih besar daripada ukuran aslinya [10]. Pembesaran citra atau *zoom* adalah pergerakan pada lensa yang dilakukan secara optik dengan memanjangkan atau memendekkan jarak antara lensa utama dengan sensor kamera. Tujuannya adalah untuk mendapatkan sudut pandang kamera lebar atau sempit, serta dipergunakan untuk memperbesar atau memperkecil sebuah objek [11]. Penskalaan citra terbagi menjadi 2 yaitu *zoom in* merupakan proses rekonstruksi sudut pandang citra ukuran awal ke ukuran yang lebih besar dan *zoom out* merupakan proses rekonstruksi sudut pandang citra ukuran awal ke ukuran yang lebih kecil [12].

2.5. Pengolahan Citra

Pengolahan citra adalah suatu yang digunakan untuk menggambarkan operasi-operasi yang dilakukan pada suatu citra dengan sasaran mencapai suatu tujuan

tertentu. Citra digital didapatkan dari sekumpulan gambar maupun dari video. *Computer vision* merupakan suatu proses untuk mengolah gambar dan video untuk memperoleh suatu hal yang dapat dianalisa [13].

2.5.1. Citra *Grayscale*

Profil warna *Red, Green, Blue* (RGB) digunakan terutama untuk aplikasi layar komputer. Profil warna RGB terdiri dari warna merah, hijau, dan biru yang dapat menciptakan beragam warna berbeda pada saat ketiga warna tersebut dikombinasikan. Profil warna RGB dibentuk melalui proses aditif, warna *primer* RGB pada intensitas penuh berwarna putih, sedangkan pada intensitas terendah berwarna hitam. Citra *grayscale* merupakan suatu citra yang hanya memiliki warna tingkat keabuan. Penggunaan citra *grayscale* dikarenakan citra memiliki sedikit informasi yang diberikan pada tiap piksel dibandingkan dengan citra berwarna. Warna abu-abu pada citra *grayscale* adalah warna R (merah), G (hijau), dan B (biru) yang memiliki intensitas warna yang sama sehingga dalam gambar *grayscale* hanya membutuhkan nilai intensitas tunggal dibandingkan dengan citra berwarna membutuhkan tiga intensitas pada tiap pikselnya.

Intensitas dari citra *grayscale* disimpan dalam bentuk 8bit *integer* dimulai dari level 0 sampai dengan 255 (0 untuk hitam dan 255 untuk putih dan nilai diantaranya adalah derajat keabuan). Proses yang digunakan untuk mendapatkan citra *grayscale* dilakukan dengan mencari nilai rata-rata dari total nilai RGB. Persamaan yang digunakan untuk mendapatkan nilai *grayscale* terdapat pada persamaan 2.4.

$$G = \frac{r+b+g}{3} \quad (2.4)$$

Keterangan:

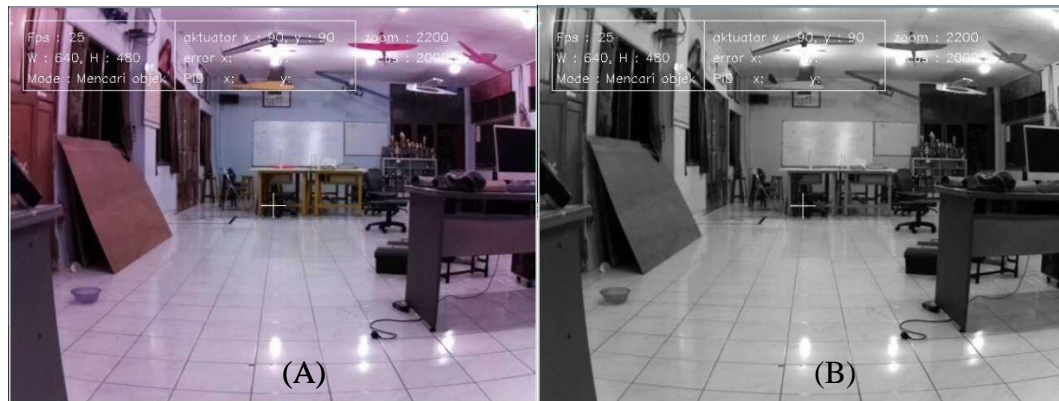
G = nilai *grayscale*

r = nilai intensitas *red* (0 sampai 255) setiap piksel

g = nilai intensitas *green* (0 sampai 255) setiap piksel

b = nilai intensitas *blue* (0 sampai 255) setiap piksel

Perbandingan gambar antara citra RGB dengan citra *grayscale* dapat dilihat pada gambar 2.4. (A) dan (B) berikut.



Gambar 2.4. (A) Citra RGB, (B) Citra *grayscale*

2.5.2. *Region Of Interest (ROI)*

Deteksi objek dalam pengolahan citra digital adalah suatu proses yang digunakan untuk menentukan keberadaan objek tertentu di dalam suatu citra digital. *Region Of Interest (ROI)* adalah sampel bagian dari citra yang ingin diidentifikasi untuk tujuan operasi tertentu dengan kata lain wilayah pilihan poligonal dari video 2D dalam visi komputer dan pengenalan karakter optik. ROI menentukan batas-batas objek yang sedang dipertimbangkan [3]. Menentukan ROI dengan membuat wilayah biner yang akan ditentukan, yang merupakan gambar biner yang berukuran sama dengan gambar yang ingin diproses dengan piksel yang ditetapkan ROI akan diberi nilai 1 dan semua piksel lainnya diberi nilai 0 [6]. Contoh pemilihan citra menggunakan ROI seperti pada gambar 2.5. berikut.



Gambar 2.5. Contoh Pemilihan Objek Menggunakan ROI.

2.5.3. Kernelized Correlation Filter (KCF)

Pelacakan secara harfiah memiliki arti mengikuti jalan, atau dalam arti bebasnya ialah suatu kegiatan untuk mengikuti jejak suatu objek. Sistem pelacakan adalah suatu sistem yang mampu melacak atau mencari suatu hal dengan memberikan informasi tentang hal tersebut. Dalam bidang pengolahan citra teknik pelacakan sering kali diimplementasikan untuk membantu kegiatan manusia dimana diperlukannya suatu sistem yang mampu melacak objek secara otomatis [15]. Pelacakan objek dapat didefinisikan sebagai suatu masalah untuk memperkirakan lintasan dari sebuah objek yang bergerak pada *frame-frame* yang berbeda dalam sebuah video. *Kernel* pada pengolahan citra pada dasarnya adalah himpunan ukuran tetap dari koefisien numerik bersama dengan titik jangkar dalam himpunan itu sendiri, yang biasanya terletak di pusat. Pelacakan objek *kernel* biasanya dilakukan dengan menghitung gerakan objek, yang diwakili oleh wilayah objek tertentu, dari satu *frame* ke *frame* berikutnya.

KCF didasarkan pada kerangka filter korelasi, mengubah solusi filter menjadi masalah pelatihan pengklasifikasi biner. Secara khusus, sampel yang berpusat pada target dipilih sebagai sampel positif dan sampel negatif di sekitar target dipilih untuk melatih pengklasifikasi. Kemudian, gambar *frame* selanjutnya dideteksi dengan *classifier*, dan hasil deteksi tersebut digunakan sebagai lokasi target pada *frame* berikutnya. Dalam KCF, dengan asumsi bahwa jumlah sampel pelatihan dan nilai-nilai yang ditandai diberikan sebagai $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, tujuan dari pelatihan adalah untuk menemukan fungsi $f(z) = w^T z$, yang membuat label $f(x_i)$ dari sampel x_i diprediksi oleh model *classifier* memiliki rata-rata minimum kesalahan persegi dengan tag aslinya y_i [16]. Dapat di nyatakan seperti persamaan 2.5 [16]:

$$E = \min_w \sum_i (f(x_i) - y_i)^2 + \lambda \|w\|^2 \quad (2.5)$$

Keterangan:

x_i = Sampel ke- i

w = Vektor koefisien classifier linier

y_i = Label dari sampel ke- i

λ = Parameter regularisasi yang digunakan untuk mencegah *overfitting*

Metode KCF menghasilkan serangkaian sampel kandidat dengan mengeksploitasi properti matriks siklik pada daerah *frame* kandidat. Ini meningkatkan kecepatan pelacakan dibandingkan dengan pengambilan sampel pada daerah *frame* keseluruhan. Masalah tersebut kemudian diubah menjadi operasi cepat dalam domain frekuensi dengan *transformasi fourier*. Hal ini mengubah masalah *regresi ridge* dalam domain waktu menjadi masalah korelasi silang dalam domain frekuensi [17]. Dalam bahasa pemrograman *python*, *listing* program untuk perintah KCF dapat dituliskan sebagai berikut.

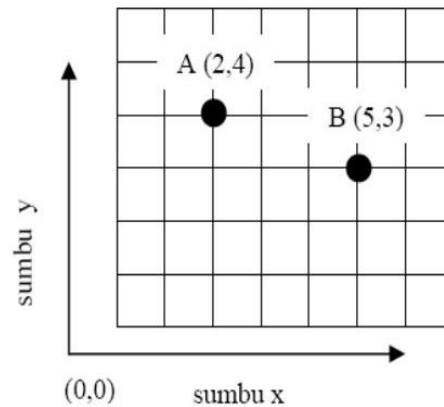
```
Ptr<TrackerKCF> CV :: TrackerKCF::create:(constTrackerKCF:
    :Params & parameters=
    TrackerKCF::Params())
#Python
    Tracker = cv.TrackerKCF_create([,parameters])
```

Sistem sasaran target sebagai informasi masukan untuk mengontrol gimbal. Pelacak KCF digunakan dalam penelitian ini, dengan skala perubahan target. Kotak pembatas pelacakan ROI berubah dengan adaptasi terhadap variasi objek dalam aliran citra video [3].

2.5.4. Lokalisasi Objek pada Citra

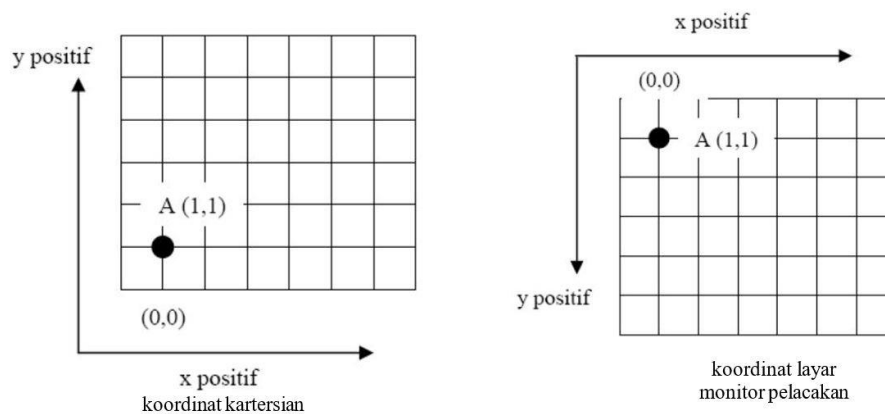
Lokalisasi objek menyediakan letak koordinat objek yang telah ditentukan untuk menghitung posisi objek berdasarkan orientasi dan ukuran piksel objek dalam koordinat layar monitor citra. Sistem koordinat kartesian adalah sistem koordinat berupa susunan garis dan titik dalam dua dimensi yang digunakan untuk menyatakan posisi dari titik dalam bidang menggunakan pasangan bilangan yang disebut dengan *absis* (sumbu x) dan *ordinat* (sumbu y) dari titik tersebut.

Pada gambar 2.6. merupakan sistem koordinat kartesian dimana setiap titik ditentukan lokasinya melalui pasangan nilai sumbu x dan y. Nilai koordinat x bertambah positif dari kiri ke kanan dan nilai y bertambah positif dari bawah ke atas.



Gambar 2.6. Sistem Koordinat Kartesian.

Pada sistem koordinat layar monitor pelacakan dimana memiliki sistem koordinat yang berbeda dengan sistem koordinat kartesian, dimana dapat dilihat seperti pada gambar 2.7. berikut.



Gambar 2.7. Perbandingan Koordinat Kartesian Dengan Koordinat Layar Monitor Pelacakan.

Gambar 2.7. merupakan perbandingan koordinat kartesian dengan koordinat layar monitor pelacakan, sebuah titik pada sistem koordinat kartesian di gambar ulang ke sistem koordinat layar monitor pelacakan maka secara visual, lokasi titik tersebut akan berubah arah sumbu koordinat layar berkebalikan dengan yang digunakan di koordinat kartesian dan pada layar monitor pelacakan sumbu x bertambah positif ke kanan dan sumbu y bertambah positif ke bawah, sehingga koordinat letak objek yang dilacak dihitung menggunakan sistem koordinat layar monitor pelacakan .

III. METODOLOGI PENELITIAN

3.1. Waktu dan Tempat Penelitian

Penelitian tugas akhir ini dilakukan di Laboratorium Terpadu Jurusan Teknik Elektro Universitas Lampung. Sedangkan waktu penelitian tugas akhir ini dimulai dari bulan Juli 2021 sampai dengan bulan Juni 2022.

3.2. Komponen dan Perangkat Lunak

Adapun komponen dan perangkat lunak pada penelitian ini seperti pada tabel 3.1. berikut.

Tabel 3.1. Komponen dan Perangkat Lunak.

No.	Nama Komponen dan Perangkat Lunak	Keterangan Penggunaan
1.	<i>Raspberry pi 4 model BS RAM 4 GB</i>	Sebagai pusat kontrol untuk melakukan pengolahan citra
2.	<i>Kamera Arducam 8MP PTZ</i>	Sebagai modul kamera pada gimbal kamera
3.	<i>Motor servo Tower Pro MG995</i>	Sebagai aktuator gerak pada gimbal kamera
4.	<i>Keyboard</i>	Sebagai <i>tool</i> kontrol penggerak pada gimbal kamera
5.	<i>Mouse</i>	Sebagai <i>tool</i> untuk memilih objek
6.	Perangkat lunak <i>Python</i> versi 3.7.4	Sebagai bahasa pemrograman untuk membuat program gimbal kamera
7.	Perangkat lunak <i>Opencv</i> versi 4.5.1	Sebagai <i>library</i> untuk mengolah citra video pada program gimbal kamera
8.	Perangkat lunak <i>VNC</i> versi 6.5.0	Sebagai pengontrol <i>Raspberry pi 4</i> dari laptop melalui sambungan jaringan

3.3. Skenario Pelacakan Objek dengan Gimbal Kamera

Skenario pelacakan objek pada gimbal kamera sebagai berikut.

1. Operator membuka program yang berada pada *Raspberry pi 4* model B.
2. Operator menggerakkan aktuator gimbal kamera secara manual untuk mencari objek yang akan dilacak dengan *keyboard*;
 - Tombol “D” untuk memutar aktuator sumbu Y ke kanan.
 - Tombol “A” untuk memutar aktuator sumbu Y ke kiri.
 - Tombol “W” untuk memutar aktuator sumbu X ke atas.
 - Tombol “S” untuk memutar aktuator sumbu X ke bawah.
3. Setelah objek ditetapkan, selanjutnya masuk ke mode pemilihan objek dengan memblok objek memposisikan di tengah kotak ROI menggunakan *mouse*.
4. Selanjutnya mode dipindahkan pelacakan otomatis dengan menekan tombol “T” yang berfungsi untuk memulai program pelacakan objek otomatis dan menggerakkan aktuator gimbal kamera untuk memposisikan objek agar objek selalu berada pada tengah layar monitor pelacakan.
5. Jika objek tersebut keluar dari kotak ROI atau ingin membatalkan pelacakan dan memilih objek lain, dapat menekan tombol “R” yang berfungsi untuk membatalkan mode pelacakan otomatis sehingga dapat memilih objek yang dilacak kembali.

3.4. Rancangan Sistem

Pada penelitian ini dirancang gimbal kamera elektrik yang memiliki 2 sumbu pergerakan yaitu *pitch* dan *yaw*. Sistem dirancang dengan dengan masukan berupa citra video yang diambil kamera *Arducam 8MP PTZ*, masukan tombol pada *keyboard* dan *mouse*. Terdapat 2 pengontrolan yaitu pengontrolan secara manual yang dilakukan oleh operator gimbal kamera dan pengontrolan secara otomatis dengan masukan dari citra video dengan penambahan proses pelacakan objek dengan metode *kernelized correlation filter* (KCF) sebagai masukan pengontrolan gimbal kamera disertai kontro PID untuk dapat mempercepat dan menjaga respon sistem agar gimbal kamera dapat bergerak secara otomatis untuk menjaga pengambilan citra video kamera selalu mengarah ke objek yang dilacak.

Sistem pelacakan objek secara otomatis dengan mengurangi selisih atau *error* antara *setpoint* yaitu titik tengah piksel layar monitor pelacakan dengan titik tengah piksel objek pelacakan yang berada dalam kotak *region of interest* (ROI) dalam citra video harus mendekati atau sama dengan 0. Dapat dicontohkan seperti pada persamaan 3.1 dan 3.2.

$$x_c - x_d = 0 \quad (3.1)$$

$$y_c - y_d = 0 \quad (3.2)$$

Keterangan:

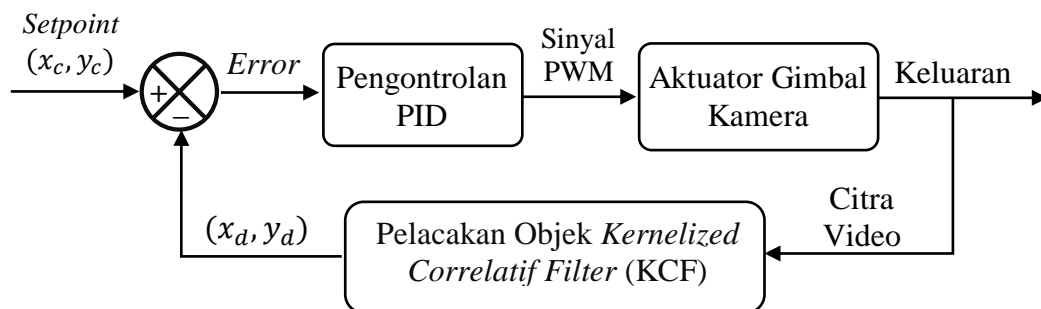
x = Variabel pada sumbu X

y = Variabel pada sumbu Y

x_c, y_c = Titik tengah piksel layar monitor pelacakan atau *setpoint* (px)

x_d, y_d = Titik tengah piksel objek pelacakan yang berada dalam kotak ROI (px)

Pada persamaan 3.1 dan 3.2 menyebabkan kontrol gimbal akan menggerakkan aktuator gimbal kamera hingga *error* mendekati atau sama dengan 0 sehingga citra objek yang dilacak selalu berada di tengah pusat layar monitor pelacakan. Gagasan modifikasi pengontrolan kalang tertutup pada gimbal kamera dengan kontrol PID dapat dilihat pada gambar 3.1. berikut.

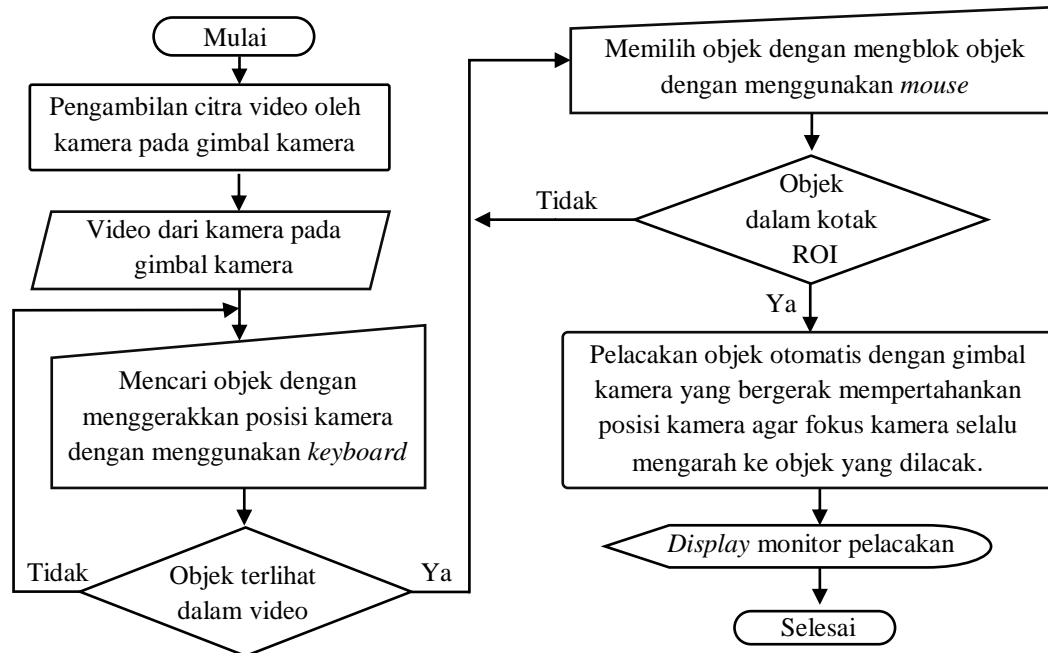


Gambar 3. 1. Konsep Penelitian Pada Pengontrol Gimbal Kamera.

Pada gambar 3.1. merupakan modifikasi dari pengontrolan gimbal kamera yang menggunakan kontrol PID dimana umpan balik (*feedback*) berupa pelacakan objek. Masukan kontrol PID berasal dari *error* antara titik tengah piksel layar monitor pelacakan (*setpoint*) dengan titik tengah piksel objek dilacak yang berada dalam kotak ROI yang telah dilakukan pelacakan KCF, kemudian memberikan masukan sebagai sinyal PWM.

3.5. Diagram Alir Sistem

Tahapan yang dilakukan dalam sistem untuk pengontrolan gimbal kamera dapat direpresentasikan dalam diagram alir seperti pada gambar 3.2. berikut.

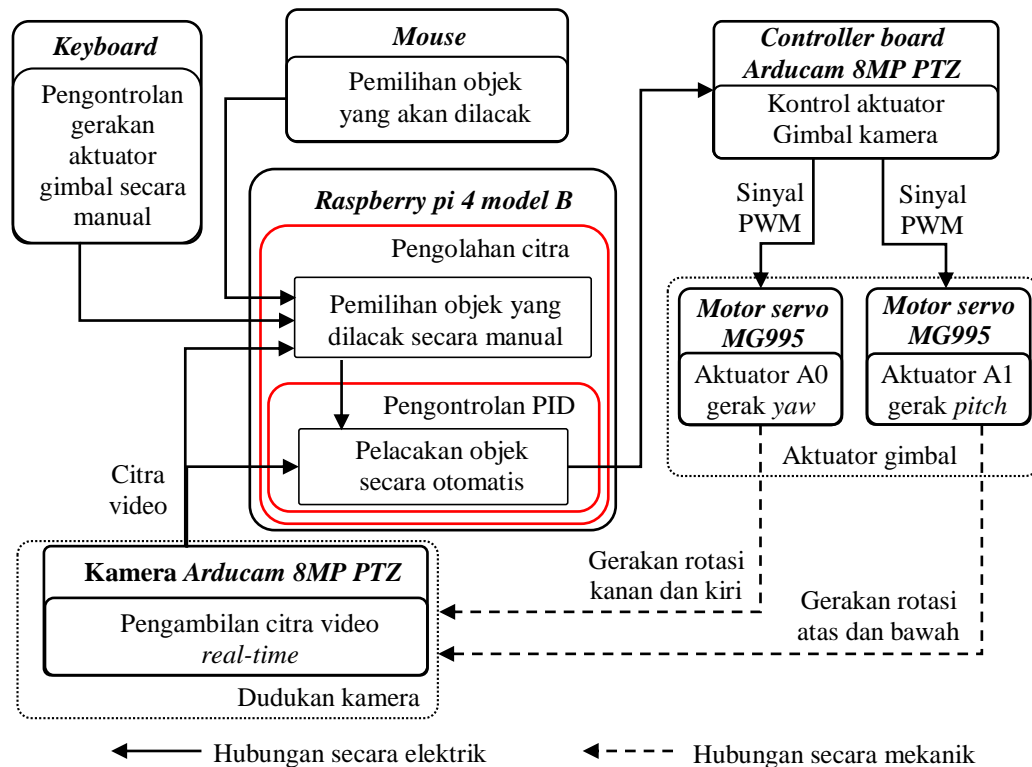


Gambar 3. 2. Diagram Alir Sistem Pelacakan Gimbal Kamera.

Saat gimbal kamera aktif sumbu kamera akan diatur ke sudut inersia kamera. Pengambilan citra video oleh kamera yang terpasang pada gimbal akan mengirimkan video *real time*. Kemudian sistem masuk ke pengontrolan manual, dimana gimbal kamera akan digerakkan untuk mencari objek yang akan dilacak, pencarian dilakukan secara manual oleh operator dengan menggunakan *keyboard*, apabila objek sudah terlihat di dalam video, selanjutnya pemilihan objek dengan cara memblok objek yang dipilih dengan *mouse* secara manual oleh operator, jika objek sudah terpilih dan objek berada dalam kotak pelacakan maka sistem gimbal akan beralih ke pengontrolan otomatis untuk melakukan pelacakan dengan menggerakkan aktuator gimbal kamera untuk mempertahankan posisi kamera agar selalu fokus kearah objek yang ditentukan dan dapat dilihat pada layar *display* monitor objek yang dilacak selalu berada di tengah bidang bingkai video layar monitor pelacakan.

3.6. Diagram Blok Sistem dan Listing Program

Diagram blok sistem pengontrolan gimbal kamera dapat direpresentasikan dalam diagram blok seperti pada gambar 3.3. berikut.



Gambar 3.3. Diagram Blok Sistem Gimbal Kamera.

Pengolahan citra pada *Raspberry Pi 4 Model B* dilakukan pemilihan objek yang dilacak secara manual dengan masukannya berupa citra video dari kamera *Arducam 8MP PTZ*. Pengontrolan gerak aktuator dari *keyboard* dan pemilihan objek dari *mouse*. Setelah objek yang terlacak terpilih, kemudian masuk ke pelacakan otomatis dengan menambahkan pengontrolan PID. Kemudian masuk ke *Controller Board Arducam 8MP PTZ* untuk membagi pergerakan aktuator dengan mengirimkan sinyal PWM ke *motor Servo MG995* pada aktuator A0 gerak *yaw* dan aktuator A1 gerak *pitch*, dimana aktuator A0 gerak *yaw* akan memutar dudukan kamera gimbal bergerak rotasi kanan dan kiri dan aktuator A1 gerak *pitch* akan memutar dudukan kamera gimbal bergerak rotasi atas dan bawah, sehingga posisi kamera selalu mengarah ke objek yang dilacak.

3.6.1. Pergerakan Manual

Pergerakan manual digunakan oleh operator dalam memilih objek yang dilacak dengan menggerakkan aktuator gerak *yaw* dan *pitch* gimbal kamera serta dapat melakukan penskalaan citra video pada kamera yang berfungsi untuk memudahkan operator menyeleksi objek dan memperjelas tampilan visual objek yang dilacak.

3.6.1.1. Pergerakan Aktuator Secara Manual

Aktuator yang digunakan pada penelitian ini adalah motor *servo Tower Pro MG995* yang dapat berotasi sebesar 180° dimana untuk aktuator yang bergerak sumbu Y bernama aktuator A0 gerak *yaw* dan sumbu X bernama aktuator A1 gerak *pitch*. Untuk menggerakkan gimbal secara manual menggunakan *keyboard* dengan aturan tombol dan gerak seperti pada tabel 3.2. berikut.

Tabel 3.2. Tombol Pergerakan Gimbal Manual.

No	Tombol <i>keyboard</i>	Nama aktuator	Gerak	Sumbu gerak
1	D	Aktuator A0	Gerak <i>yaw</i> ke kanan	Sumbu Y
2	A	Aktuator A0	Gerak <i>yaw</i> ke kiri	Sumbu Y
3	W	Aktuator A1	Gerak <i>pitch</i> ke atas	Sumbu X
4	S	Aktuator A1	Gerak <i>pitch</i> ke bawah	Sumbu X

Listing program masukan berdasarkan pada tabel 3.2. untuk pergerakan gimbal secara manual dapat dilihat pada *listing* program berikut.

```
def parseKey(k, actuator):
    global image_count
    set = 90
    yaw_step = 1
    pitch_step = 1
    if k == ord("D"):
        actuator.set(Actuator.OPT_ACTUATOR_X, actuator.get(Actuator.
            OPT_ACTUATOR_Y) + yaw_step)
    elif k == ord("A"):
        actuator.set(Actuator.OPT_ACTUATOR_X, actuator.get(Actuator.
            OPT_ACTUATOR_Y) - yaw_step)
    elif k == ord("W"):
        actuator.set(Actuator.OPT_ACTUATOR_Y, actuator.get(Actuator.
            OPT_ACTUATOR_X) + pitch_step)
    elif k == ord("S"):
        actuator.set(Actuator.OPT_ACTUATOR_Y, actuator.get(Actuator.
            OPT_ACTUATOR_X) - pitch_step)
```

Pada *listing* program untuk pergerakan manual sudut tengah posisi kamera pada setiap sumbu diatur pada sudut 90° , parameter pergerakan disetiap sumbu adalah 1° , pada aktuator A0 sumbu Y posisi 0° adalah batas maksimum gerak *yaw* ke kanan sedangkan posisi 180° adalah batas maksimum gerak *yaw* ke kiri. Sedangkan pada aktuator A1 sumbu X posisi 0° adalah batas maksimum gerak *pitch* ke atas sedangkan posisi 180° adalah batas maksimum gerak *pitch* ke bawah.

Jika menekan tombol “D” maka akan memutar aktuator A0 pada sumbu Y ke kanan sebesar 1° , menekan tombol “A” maka akan memutar aktuator A0 pada sumbu Y ke kiri sebesar 1° , menekan tombol “W” maka akan aktuator A1 pada sumbu X ke atas sebesar 1° , menekan tombol “S” maka akan memutar aktuator A1 pada sumbu X ke bawah sebesar 1° .

3.6.1.2. Penskalaan Citra pada Kamera

Pada penelitian ini menggunakan kamera *Arducam 8MP PTZ* yang memiliki fitur penskalaan citra, fitur tersebut digunakan untuk memperbesar atau memperkecil sudut pandang citra yang diambil oleh kamera yang diatur secara manual dimana pengaturan fokus pada kamera berfungsi untuk menjernihkan citra yang diambil, hal ini dikarenakan setiap melakukan perubahan pada penskalaan citra kamera maka citra tersebut akan buram sehingga fokus kamera harus diubah untuk menjernihkan citra kembali. Untuk mengatur penskalaan citra secara manual menggunakan *keyboard* dengan aturan gerak seperti pada tabel 3.3. berikut.

Tabel 3.3. Tombol Penskalaan Citra Pada *Keyboard*.

No	Tombol <i>keyboard</i>	Penskalaan citra
1	Tombol “UP” (↑)	<i>Zoom in</i>
2	Tombol “DOWN” (↓)	<i>Zoom out</i>
3	Tombol “RIGHT” (→)	<i>Focus in</i>
4	Tombol “LEFT” (←)	<i>Focus out</i>

Listing program masukan berdasarkan pada tabel 3.3. tombol penskalaan citra pada *keyboard* dapat dilihat pada *listing* program berikut.

```
zoom_step = 100
```



```

focus_step = 50
elif k == curses.KEY_UP:
    actuator.set(Actuator.OPT_ZOOM, actuator.get(Actuator.OPT_ZOOM) + zoom_step)
elif k == curses.KEY_DOWN:
    actuator.set(Actuator.OPT_ZOOM, actuator.get(Actuator.OPT_ZOOM) - zoom_step)
elif k == curses.KEY_RIGHT:
    actuator.set(Actuator.OPT_FOCUS, actuator.get(Actuator.OPT_FOCUS) + focus_step)
elif k == curses.KEY_LEFT:
    actuator.set(Actuator.OPT_FOCUS, actuator.get(Actuator.OPT_FOCUS) - focus_step)
    Pass

```

Pada *listing* program untuk mengatur penskalaan citra pada kamera dimana parameter step perubahan pada *zoom* sebesar 0-20000 step atau pada *datasheet Arducam 8MP PTZ* memiliki perubahan pembukaan sensor kamera sebesar 9,1753 mm dan perubahan *focus* sebesar 0-20000 step pada *datasheet Arducam 8MP PTZ* memiliki perubahan jarak lensa dengan sensor kamera sebesar 13,8414 mm. Perubahan step setiap menekan tombol pada *zoom* adalah sebesar 100 step dan pada fokus adalah 50 step. Jika menekan tombol “UP” akan melakukan *zoom in* sebesar 100 step, menekan tombol “DOWN” maka akan melakukan *zoom out* sebesar 100 step dan akan melakukan *zoom out*, menekan tombol “RIGHT” maka akan melakukan *focus in* sebesar 50 step, menekan tombol “LEFT” maka akan melakukan *focus out* sebesar 50 step.

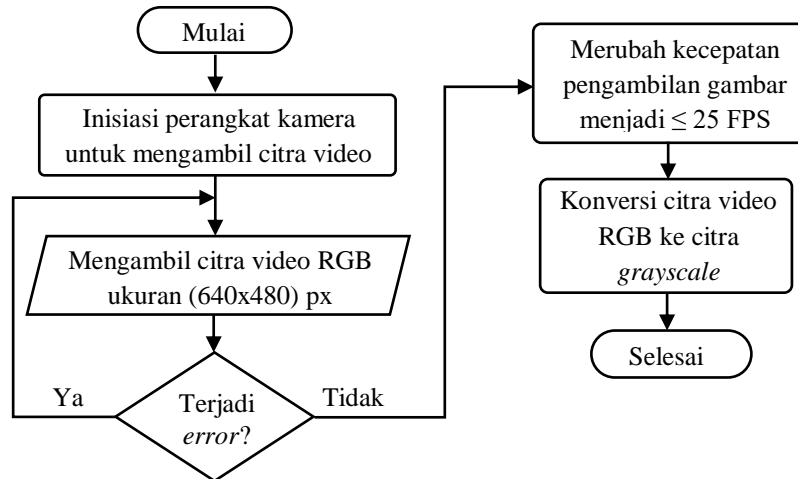
3.6.2. Pergerakan Otomatis

Pergerakan otomatis berfungsi untuk memudahkan operator dalam melakukan pengamatan objek yang bergerak, dimana objek tersebut akan selalu dapat berada pada titik tengah layar monitor pelacakan tanpa operator menggerakkan aktuator gimbal kamera secara manual.

3.6.2.1. Pengolahan Citra

Pengambilan citra menggunakan *Arducam 8MP PTZ* yang dilakukan secara *real-time*, pengolahan citra video menggunakan perangkat lunak *Python 3.7.4* dan *OpenCV 4.5.1*.

Video yang diperoleh dari kamera akan diproses sehingga diperoleh area yang merupakan gambar dan lokasi koordinat objek yang dilakukan di *Raspberry pi 4 model B*. Setelah itu data yang merupakan hasil operasi pengolahan citra dikirim sebagai masukan kontrol PID untuk pergerakan aktuator gimbal kamera. Gambar 3.4. diagram alir pengambilan citra.



Gambar 3.4. Diagram Alir Pengambilan Citra Video.

Gambar 3.4. merupakan diagram alir pengambilan citra video, perangkat kamera diinisialisasi untuk dapat mengambil citra. Selanjutnya kamera mengambil citra video dengan resolusi (640×480) px dan apabila terjadi *error* pada proses pengambilan citra video maka proses pengambilan citra video akan diulang, apabila pengambilan citra video sukses kemudian citra video ditampilkan sebagai antarmuka dengan operator dengan mengubah besar *frame rate* video menjadi kurang atau sama dengan 25 *frame per second* (fps). Selanjutnya citra warna skala RGB diubah menjadi citra skala *grayscale* atau citra abu-abu. Citra video tersebut selanjutnya masuk program pengolahan citra untuk operasi pelacakan objek.

Listing program berdasarkan gambar 3.4. diagram alir pengambilan citra sebagai berikut.

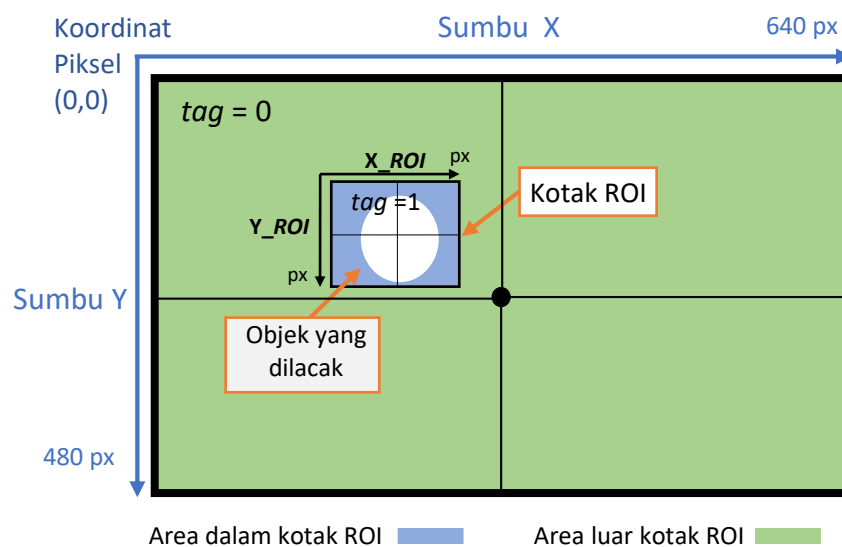
```

cv2.namedWindow("Tracking Object")
video = cv2.VideoCapture(0)
video.set(cv2.CAP_PROP_FPS, 25.0)
Fps = int(video.get(5))
ret, frame = video.read()
frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
  
```

Dimulai dengan menginisiasi tampilan video dengan nama “*Tracking Object*” kemudian membuka pengambilan citra video dengan resolusi tampilan (640×480) px, kemudian mengubah *frame rate* menjadi kurang atau sama dengan 25 fps, kemudian merubah citra masukan menjadi citra *grayscale*.

3.6.2.2. Pemilihan dan Pelacakan Objek yang Dilacak

Pemilihan objek dilakukan dengan memblok objek yang diposisikan di tengah kotak ROI menggunakan *mouse*. Dimana area di dalam kotak ROI ditandai dengan *tag = 1* dan area di luar kotak ROI ditandai dengan *tag = 0*. Pelacakan objek menggunakan metode *kernelized correlation filter* (KCF) yang merupakan paket dari library *Opencv* versi 4.5.1 untuk melakukan pelacakan objek yang berada dalam kotak ROI yang telah dilakukan dari pemilihan objek yang ditandai *tag = 1*. Visualisasi layar pelacakan seperti pada gambar 3.5. berikut.



Gambar 3.5. Visuali Pelacakan Pada Layar Monitor Pelacakan.

Listing program pelacakan objek dengan metode KCF sebagai berikut.

```
while True:
    timer = cv2.getTickCount()
    ret, frame = video.read()
    frame = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    if ret:
        if len(mousePoints) == 2:
            cv2.rectangle(frame, mousePoints[0], mousePoints[1], color, 1)
            KCF_rect = KCF.rectangle(mousePoints[0][0],
```

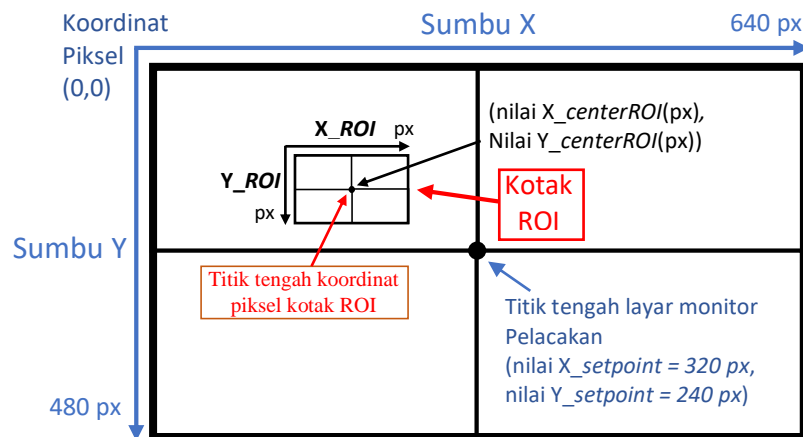
```

mousePoints[0][1],mousePoints[1][0],mousePoints[1][1])
cv2.putText(frame, "Memilih objek", (80, 90),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)
else:
cv2.putText(frame, "Mencari objek", (80, 90),
cv2.FONT_HERSHEY_SIMPLEX, 0.5, color, 1)
if tracked == True:
tracker.update(frame)
track_rect = tracker.get_position()

```

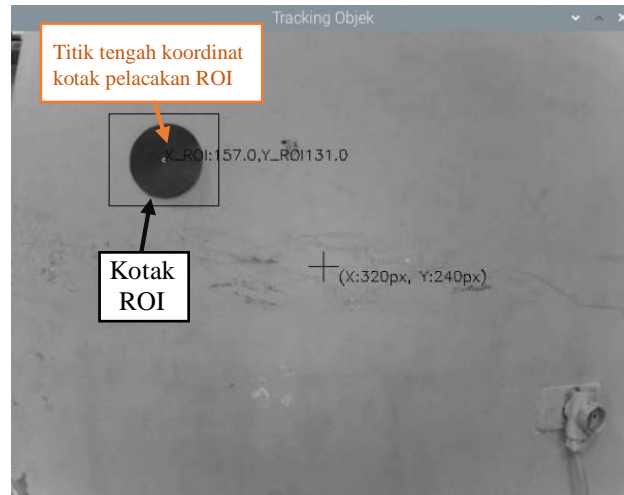
3.6.2.3. Lokalisasi Objek yang Dilacak

Letak koordinat piksel objek pada citra video diketahui melalui pelacakan metode *kernelized correlation filter* (KCF) yang berada di kotak ROI sehingga nilai *error* dari titik tengah layar monitor pelacakan atau *setpoint* dengan letak koordinat piksel titik tengah kotak ROI sebagai masukan kontrol PID untuk menggerakkan aktuator gimbal. Ilustrasi visualisasi pada layar monitor pelacakan (*setpoint*) dan kotak ROI seperti pada gambar 3.6. berikut.



Gambar 3.6. Ilustrasi Visualisasi Layar Monitor Pelacakan.

Pada gambar 3.6 dan gambar 3.7 merupakan layar monitor pelacakan dimana terdapat titik tengah layar monitor pelacakan atau nilai koordinat piksel *setpoint* yang digunakan sebagai acuan pengontrolan gimbal kamera untuk mencari nilai koordinat piksel *setpoint*.



Gambar 3.7. Visualisasi Layar Monitor Pelacakan Pada Citra Video Yang Diambil.

Dimana pada penelitian ini citra video beresolusi (640×480) px maka jumlah nilai koordinat piksel pada sumbu X adalah 640 px dan sumbu Y adalah 480 px, maka untuk mencari nilai koordinat piksel *setpoint* sebagai berikut. Pada koordinat piksel sumbu X:

$$X_setpoint = \frac{(sumbu X)}{2} \quad (3.3)$$

Pada koordinat piksel sumbu Y:

$$Y_setpoint = \frac{(sumbu Y)}{2} \quad (3.4)$$

Nilai koordinat piksel *setpoint* pada kedua sumbu:

$$((X_setpoint) \text{ px}, (Y_setpoint) \text{ px}) \quad (3.5)$$

Untuk mencari nilai koordinat piksel dari titik tengah objek yang dilacak maka dengan mencari nilai koordinat piksel titik tengah kotak *regions of interest* (ROI) sebagai berikut. Pada koordinat piksel sumbu X_ROI:

$$X_centerROI = \frac{(sumbu X_ROI)}{2} \quad (3.6)$$

Pada koordinat piksel sumbu Y_ROI:

$$Y_centerROI = \frac{(sumbu Y_ROI)}{2} \quad (3.7)$$

Nilai koordinat piksel titik tengah kotak ROI pada kedua sumbu:

$$((X_centerROI)\text{px}, (Y_centerROI)\text{px}) \quad (3.8)$$

Untuk mendapatkan nilai *error* antara nilai koordinat piksel *setpoint* atau titik tengah layar monitor pelacakan dengan nilai koordinat piksel titik tengah kotak ROI seperti berikut. *Error* pada sumbu X layar monitor pelacakan:

$$X_error = X_setpoint - X_centerROI \quad (3.9)$$

Error pada sumbu Y layar monitor pelacakan:

$$Y_error = Y_setpoint - Y_centerROI \quad (3.10)$$

Listing program lokalisasi objek untuk menentukan titik tengah koordinat piksel layar monitor pelacakan atau *setpoint* dan titik tengah koordinat piksel kotak ROI dan mencari nilai *error* pada sumbu X dan sumbu Y.

```
x1 = int(track_rect.left())
x2 = int(track_rect.right())
y1 = int(track_rect.top())
y2 = int(track_rect.bottom())
roi1 = (x1, y1)
roi2 = (x2, y2)
X_ROI = (x1 + x2)
Y_ROI = (y1 + y2)
cv2.rectangle(frame, (roi1), (roi2), color, 1)
cv2.putText(frame, "Traking objek", (80, 90), cv2.FONT_HERSHEY_
SIMPLEX, 0.5, color, 1)
X_centerROI = X_ROI / 2
Y_centerROI = Y_ROI / 2
centerROI = (int(X_centerROI), (int(roi_center_y)))
cv2.circle(frame, centerROI, 2, (255, 255, 255), 1)
txtc = "xc:[{}], yc:[{}]".format(X_centerROI, Y_centerROI)
X_error = X_setpoint - X_centerROI
Y_error = Y_setpoint - Y_centerROI
```

3.6.2.4. Algoritma Sistem Kontrol PID

Sistem pengontrolan PID digunakan untuk membuat sistem memiliki *rise time* dan *error* sistem menjadi lebih baik sehingga sistem akan merespon dengan cepat dan presisi. Masukan kontrol yang digunakan adalah *error* yang dihasilkan antara nilai koordinat piksel titik tengah layar monitor pelacakan (*setpoint*) dengan nilai koordinat piksel titik kotak ROI. Keluaran dari kontrol PID adalah kecepatan gerak aktuator A0 *yaw* dan aktuator A0 *pitch*. Kontrol PID merupakan diagram yang digunakan untuk menentukan nilai parameter yang akan dimasukkan pada parameter PID mendefinisikan nilai PID yang akan digunakan di pergerakan aktuator [18].

Nilai Proporsional ($P(t)$) didapat dari nilai konstanta k_p dikalikan dengan nilai *error*. *Listing* program berdasarkan persamaan (2.1) untuk mencari nilai proporsional ($P(t)$) sebagai berikut.

```
#P
if self._last_t == 0 or dt > 1000:
    dt = 0
    self.reset_I()
    self._last_t = self.tnow
    delta_time = float(dt) / float(1000)
    output += self._kp * error
```

Nilai integral ($I(t)$) didapat dari konstanta k_i dikalikan dengan nilai integrator. Nilai integrator adalah penambahan dari nilai *error* sebelum sekarang dengan *error* sekarang dimana *delta time* adalah nilai pada waktu sekarang dikurang nilai pada waktu sebelum sekarang dan nilai *scaler* adalah nilai *error* pada waktu sekarang. *Listing* program berdasarkan persamaan (2.2) untuk mencari nilai integral ($I(t)$) sebagai berikut.

```
#I
if abs(self._ki) > 0 and dt > 0:
    self._integrator += error * scaler * delta_time
    output += self._ki * self._integrator_time
```

Nilai derivatif ($D(t)$) didapat dari nilai *error* diambil dengan *error* terakhir pada waktu sekarang. Nilai derivatif didapat dari konstanta k_d dikalikan dengan nilai derivatif. Nilai derivatif adalah pengurangan dari nilai *error* sekarang dengan *error* sebelum sekarang sedangkan nilai *scaler* adalah nilai *error* pada waktu sekarang. *Listing* program berdasarkan persamaan (2.3) untuk mencari nilai derivatif ($D(t)$) sebagai berikut.

```
#D
if abs(self._kd) > 0 and dt > 0:
    if isnan(self._last_derivative):
        derivative = 0
        self._last_derivative = 0
    else:
        derivative = (error - self._last_error) / delta_time
    derivative = self._last_derivative + (delta_time *
        (derivative - self._last_derivative))
```

```

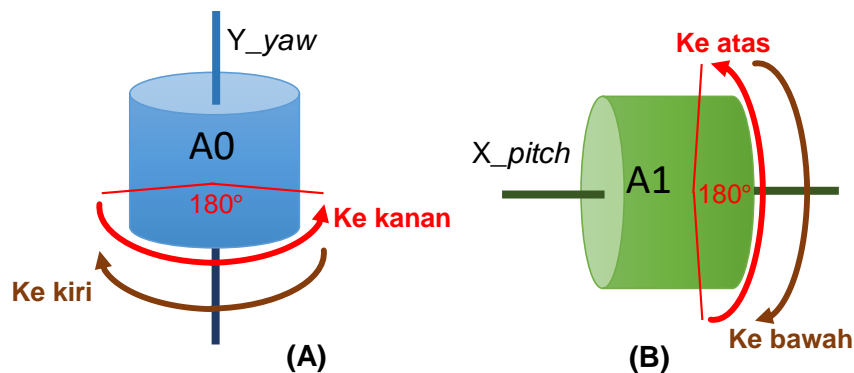
self._last_error = error
self._last_derivative = derivative
output += self._kd * derivative
output *= scaler

```

Setelah nilai proporsional, integral, derivatif (PID) masing-masing ditemukan kemudian menambahkan ketiga nilai tersebut pada keluaran “*output+*” di program “*PID.py*” sehingga didapat nilai dari kontrol PID untuk dimasukan pengontrolan aktuator pada program pergerakan aktuator gimbal kamera.

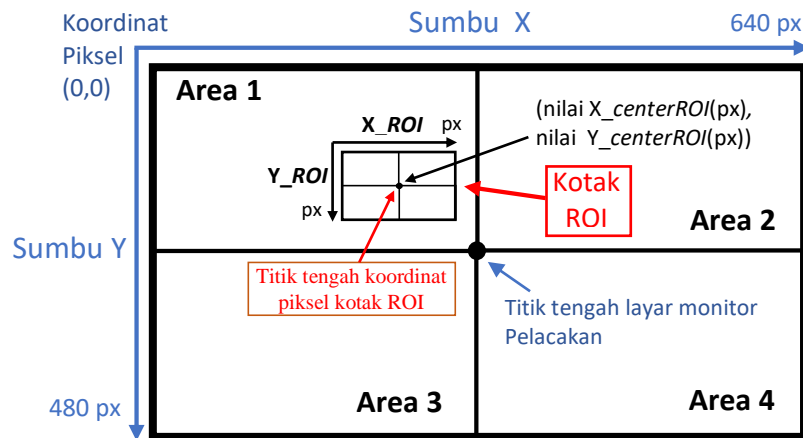
3.6.2.5. Pengontrolan Gerak Aktuator

Pengontrolan pergerakan aktuator pada gimbal kamera yang bergerak otomatis memiliki masukan berupa letak objek pada monitor pelacakan, sehingga letak objek pada monitor pelacakan akan mempegaruhi aktuator gimbal kamera bergerak. Mekanisme gerak aktuator dapat dilihat pada gambar 3.8. berikut.



Gambar 3. 8. (A) Pergerakan Aktuator Pada Sumbu Y, (B) Pergerakan Aktuator Pada Sumbu X.

Aktuator dibagi menjadi 2 yaitu gambar 3.8. (A) aktuator A0 merupakan aktuator pada sumbu Y dengan gerak *yaw* yang dapat bergerak sebesar 180° ke kanan dan ke kiri, sedangkan gambar 3.8. (B) aktuator A1 merupakan aktuator pada sumbu X dengan gerak *pitch* yang dapat bergerak sebesar 180° ke atas dan ke bawah. Letak objek di dalam kotak ROI yang berada pada layar monitor pelacakan diilustrasikan seperti pada gambar 3.9. berikut.



Gambar 3.9. Ilustrasi Letak Kotak ROI Visualisasi Layar Monitor Pelacakan.

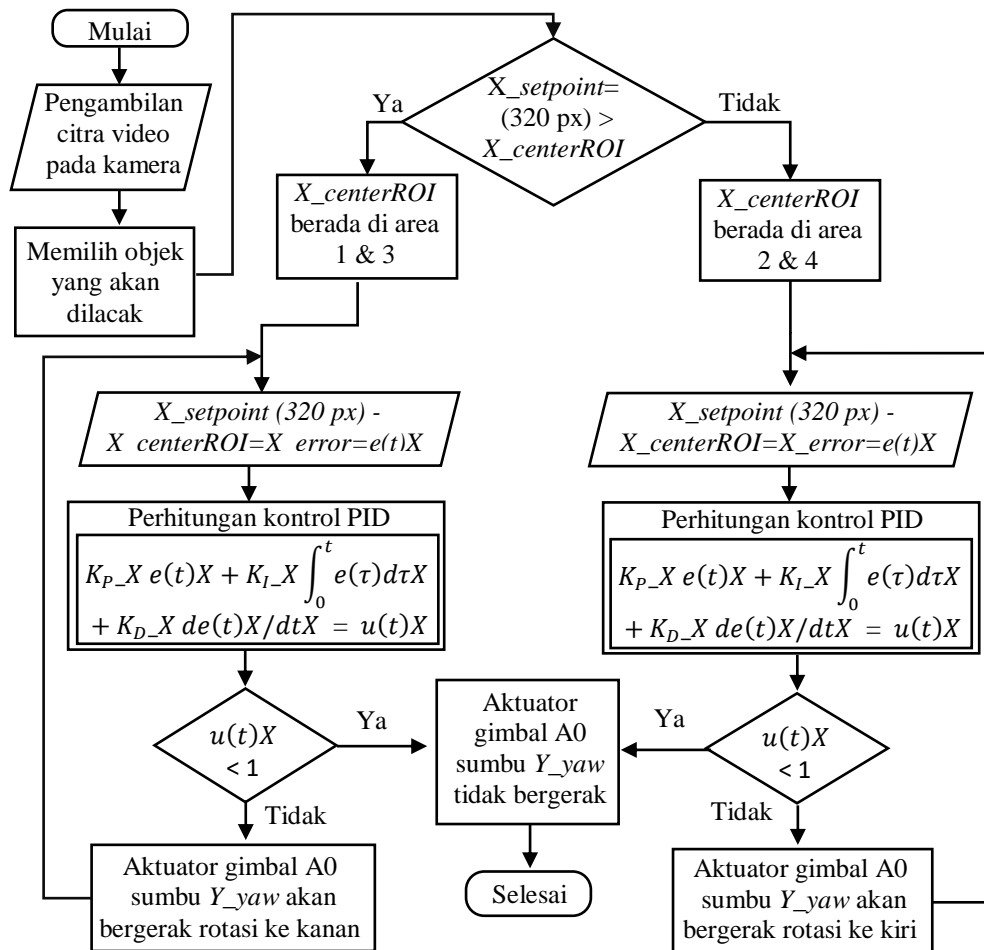
Gambar 3.9. merupakan ilustrasi letak kotak ROI pada layar monitor pelacakan yang membagi menjadi 4 Area, dimana setiap area akan mempengaruhi pergerakan aktuator A0 dan aktuator A1. Pergerakan aktuator berdasarkan letak kotak ROI dapat dilihat pada tabel 3.4. berikut.

Tabel 3.4. Pergerakan Aktuator Berdasarkan Letak Titik Koordinat Titik Tengah Kotak ROI Pelacakan.

No.	Letak kotak ROI	Kondisi berbandingan nilai koordinat piksel	Gerak aktuator untuk menuju <i>setpoint</i>
1	Area 1	nilai $X_{setpoint} >$ nilai $X_{centerROI}$, nilai $Y_{setpoint} >$ nilai $Y_{centerROI}$	Aktuator A0 bergerak ke kanan, Aktuator A1 bergerak ke bawah
2	Area 2	nilai $X_{setpoint} <$ nilai $X_{centerROI}$, nilai $Y_{setpoint} >$ nilai $Y_{centerROI}$	Aktuator A0 bergerak ke kiri, Aktuator A1 bergerak ke bawah
3	Area 3	nilai $X_{setpoint} >$ nilai $X_{centerROI}$, nilai $Y_{setpoint} <$ nilai $Y_{centerROI}$	Aktuator A0 bergerak ke kanan, Aktuator A1 bergerak ke atas
4	Area 4	nilai $X_{setpoint} <$ nilai $X_{centerROI}$, nilai $Y_{setpoint} <$ nilai $Y_{centerROI}$	Aktuator A0 bergerak ke kiri, Aktuator A1 bergerak ke atas

Perhitungan gerak aktuator pada gimbal kamera dimana syarat untuk aktuator dapat bergerak adalah nilai *error* hasil dari operasi perhitungan PID lebih besar dari 1 dan aktuator akan diam atau tidak bergerak jika nilai kurang dari 1.

Diagram alir dari pergerakan aktuator A0 pada sumbu Y gimbal kamera, sumbu X layar monitor pelacakan seperti gambar 3.10. berikut.



Gambar 3.10. Diagram Alir Pergerakan Aktuator A0 Gerak Yaw Gimbal Kamera

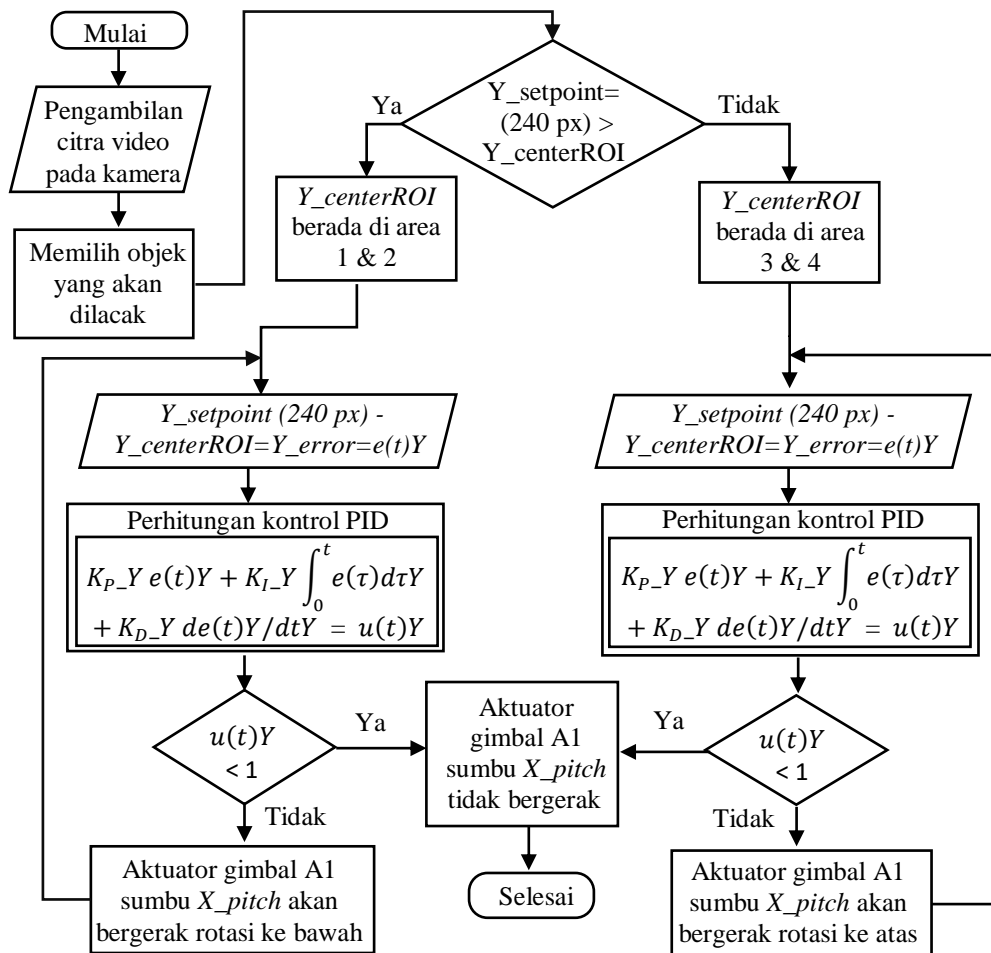
Listing program berdasarkan gambar 3.10. sebagai berikut.

```

XOrY = random.randint(0.1)
if XOrY == 0:
yaw_step = yaw_pid.get_pid(error_x,0.1)
yaw_step_x = round(yaw_step,2)
if ((X_setpoint[0]) > X_centerROI) and ((X_setpoint[0])
< X_centerROI) :
    Pass
elif ((X_setpoint[0]) > X_centerROI):
    actuator.set(Actuator.OPT_ACTUATOR_X,actuator.
get(Actuator.OPT_ACTUATOR_X) - abs(int(yaw_step_x)))
elif ((X_setpoint[0]) > X_centerROI):
    actuator.set(Actuator.OPT_ACTUATOR_X,actuator.
get(Actuator.OPT_ACTUATOR_X) + abs(int(yaw_step_x)))
cv2.putText(frame, "{}".format(yaw_step_x), (275, 90),
cv2.FONT_HERSHEY_SIMPLEX, 0.5 , color, 1)

```

Diagram alir dari pergerakan aktuator A1 pada sumbu X gimbal kamera, sumbu Y layar monitor pelacakan seperti gambar 3.11. berikut.



Gambar 3.11. Diagram Alir Pergerakan Aktuator A1 Gerak *Pitch* Gimbal Kamera.

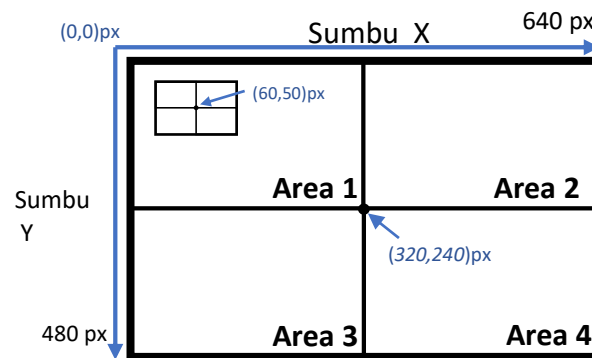
Listing program berdasarkan gambar 3.11. sebagai berikut.

```

else:
pitch_step = pitch_pid.get_pid(error_y,0.1)
pitch_step_y = round(pitch_step,2)
if ((Y_setpoint[1]) > Y_centerROI) and ((Y_setpoint[1]) <
    Y_centerROI):
    Pass
elif ((Y_setpoint[1]) > Y_centerROI):
    actuator.set(Actuator.OPT_ACTUATOR_Y,actuator.
        get(Actuator.OPT_ACTUATOR_Y) + abs(int(pitch_step_y)))
elif ((Y_setpoint[1]) < Y_centerROI):
    actuator.set(Actuator.OPT_ACTUATOR_Y,actuator.
        get(Actuator.OPT_ACTUATOR_Y) - abs(int(pitch_step_y)))
cv2.putText(frame, "{}".format(pitch_step_y), (350, 90),
    cv2.FONT_HERSHEY_SIMPLEX, 0.5 , color, 1)

```

Skenario pergerakan gimbal kamera dengan masukan nilai *error* yang telah dimasukkan dalam proses perhitungan PID. Dimana syarat untuk aktuator dapat bergerak adalah nilai *error* dari operasi perhitungan PID lebih besar dari 1 dan aktuator akan diam atau tidak bergerak jika nilai kurang dari 1. Perhitungan dengan kontrol PID dengan nilai koordinat piksel titik tengah kotak ROI (60,50) px yang memiliki arti pada sumbu X layar monitor sebesar 60 px dan pada sumbu Y sebesar layar monitor 50 px.



Gambar 3.12. Skenario Letak Koordinat Piksel Kotak ROI Pada Layar Monitor.

Pada aktuator A0 sumbu Y_{yaw} dengan koordinat piksel sumbu X layar monitor untuk nilai konstanta pada sumbu Y pergerakan yaw dengan koordinat piksel sumbu X ($K_{p_X} = 0,114$, $K_{I_X} = 0,141$, $K_{D_X} = 0,023$) dengan *setpoint* koordinat piksel sumbu X sebesar 320 px. Nilai *error* didapat dari selisih nilai koordinat piksel *setpoint* layar monitor dengan nilai koordinat piksel titik tengah kotak ROI. Pada program yang dibuat untuk mengurangi kecepatan pergerakan aktuator nilai *error* yang didapat dikalikan dengan 0,1.

$$X_{error} = e(t)X = X_{setpoint} - X_{centerROI} \quad (3.9)$$

$$= 320 - 6 = 26$$

$$u(t)X = X_P + X_I + X_D \quad (3.11)$$

$$= (k_{p_X} e(t)X) + (k_{i_X} \int_0^t e(\tau) d\tau X) + (k_{d_X} \frac{de(t)X}{dtX}) \quad (3.12)$$

$$= (k_{p_X} \times e(t)X) + (k_{i_X} \times (e(t-1)X + e(t)X)) + (k_{d_X} \times (e(t)X - e(t-1)X)) \quad (3.13)$$

$$= (0,114 \times 26) + (0,141 \times 26) + (0,023 \times 26)$$

$$= 2,964 + 3,666 + 0,598 = 7,228$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_{yaw} akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai X_{error} sebesar 7,228.

$$X_{error} = e(t)X = 7,228$$

$$u(t)X = X_P + X_I + X_D$$

$$\begin{aligned} &= (k_p X e(t)X) + (k_i X \int_0^t e(\tau) d\tau X) + (k_d X \frac{de(t)X}{dtX}) \\ &= (k_p X \times e(t)X) + (k_i X \times (e(t-1)X + e(t)X)) + (k_d X \times (e(t)X - e(t-1)X)) \\ &= (0,114 \times 7,228) + (0,141 \times (26+7,228)) + (0,023 \times (7,228-26)) \\ &= 0,823992 + 4,685148 + (-0,431756) = 5,077384 \end{aligned}$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_{yaw} akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai X_{error} sebesar 5,077384.

$$X_{error} = e(t)X = 5,077384$$

$$u(t)X = X_P + X_I + X_D$$

$$\begin{aligned} &= (k_p X e(t)X) + (k_i X \int_0^t e(\tau) d\tau X) + (k_d X \frac{de(t)X}{dtX}) \\ &= (k_p X \times e(t)X) + (k_i X \times (e(t-1)X + e(t)X)) + (k_d X \times (e(t)X - e(t-1)X)) \\ &= (0,114 \times 5,077384) + (0,141 \times (7,228 + 5,077384)) + (0,023 \times \\ &\quad (7,228 - 5,077384)) \\ &= 0,578821776 + 1,735059144 + (-0,049464168) = 2,264416752 \end{aligned}$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_{yaw} akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai X_{error} sebesar 2,264416752.

$$X_{error} = e(t)X = 2,264416752$$

$$u(t)X = X_P + X_I + X_D$$

$$\begin{aligned} &= (k_p X e(t)X) + (k_i X \int_0^t e(\tau) d\tau X) + (k_d X \frac{de(t)X}{dtX}) \\ &= (k_p X \times e(t)X) + (k_i X \times (e(t-1)X + e(t)X)) + (k_d X \times (e(t)X - e(t-1)X)) \\ &= (0,114 \times 2,264416752) + (0,141 \times 2,264416752) + (0,023 \times \\ &\quad 2,264416752) \\ &= 0,25814351 + 1,035193906 + (-0,064698247) = 1,228639169 \end{aligned}$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_yaw akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai X_error sebesar 1,228639169.

$$X_error = e(t)X = 1,228639169$$

$$u(t)X = X_P + X_I + X_D$$

$$\begin{aligned} &= (k_p X e(t)X) + (k_i X \int_0^t e(\tau) d\tau X) + (k_d X \frac{de(t)X}{dtX}) \\ &= (k_p X \times e(t)X) + (k_i X \times (e(t-1)X + e(t)X)) + (k_d X \times (e(t)X - e(t-1)X)) \\ &= (0,114 \times 1,228639169) + (0,141 \times (2,264416752 + 1,228639169)) \\ &\quad + (0,023 \times (1,228639169 - 2,264416752)) \\ &= 0,140064865 + 0,492520885 + (-0,023822884) = 0,608762866 \end{aligned}$$

Nilai $u(t)X < 1$, maka aktuator A0 sumbu Y_yaw berhenti bergerak dengan menghasilkan nilai $X_centerROI$ sebesar 308 px dengan X_error sebesar 12 px dari *setpoint* yaitu 320 px.

Pada aktuator A1 sumbu X_pitch , dengan koordinat piksel sumbu Y layar monitor pelacakan ($K_{pY} = 0,108$, $K_{iY} = 0,112$, $K_{dY} = 0,026$) dengan *setpoint* koordinat piksel sumbu sebesar 240 px.

$$Y_error = e(t)Y = Y_setpoint - Y_centerROI \quad (3.10)$$

$$= 24 - 5 = 19$$

$$u(t)Y = Y_P + Y_I + Y_D \quad (3.14)$$

$$= (k_p Y e(t)Y) + (k_i Y \int_0^t e(\tau) d\tau Y) + (k_d Y \frac{de(t)Y}{dtY}) \quad (3.15)$$

$$\begin{aligned} &= (k_p Y \times e(t)Y) + (k_i Y \times (e(t-1)Y + e(t)Y)) + \\ &\quad (k_d Y \times (e(t)Y - e(t-1)Y)) \quad (3.16) \end{aligned}$$

$$= (0,108 \times 19) + (0,112 \times 19) + (0,026 \times 19)$$

$$= 2,052 + 2,128 + 0,494 = 4,674$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_yaw akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai Y_error sebesar 4,674.

$$Y_error = e(t)Y = 4,674$$

$$u(t)Y = Y_P + Y_I + Y_D$$

$$= (k_p Y e(t)Y) + (k_i Y \int_0^t e(\tau) d\tau Y) + (k_d Y \frac{de(t)Y}{dtY})$$

$$\begin{aligned}
&= (k_{p_Y} \times e(t)Y) + (k_{i_Y} \times (e(t-1)Y + e(t)Y)) + (k_{d_Y} \times (e(t)Y - e(t-1)Y)) \\
&= (0,108 \times 4,674) + (0,112 \times (19 + 4,674)) + (0,026 \times (4,674 - 19)) \\
&= 0,504792 + 2,651488 + (-0,372476) = 2,783804
\end{aligned}$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_yaw akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai Y_error sebesar 2,783804.

$$Y_error = 2,783804.$$

$$u(t)Y = Y_P + Y_I + Y_D$$

$$\begin{aligned}
&= (k_{p_Y} e(t)Y) + (k_{i_Y} \int_0^t e(\tau) d\tau Y) + (k_{d_Y} \frac{de(t)Y}{dt}) \\
&= (k_{p_Y} \times e(t)Y) + (k_{i_Y} \times (e(t-1)Y + e(t)Y)) + (k_{d_Y} \times (e(t)Y - e(t-1)Y)) \\
&= (0,108 \times 2,783804 + (0,112 \times (4,674 + 2,783804))) + (0,026 \times \\
&\quad (2,783804 - 4,674)) \\
&= 0,300650832 + 0,835274048 + (-0,0491451) = 1,086779784
\end{aligned}$$

Nilai $u(t)X > 1$, maka karena kotak ROI di area 1 maka aktuator A0 sumbu Y_yaw akan bergerak berputar ke bawah. Kemudian dilakukan perhitungan kembali dengan persamaan yang sama dengan nilai Y_error sebesar 2,783804.

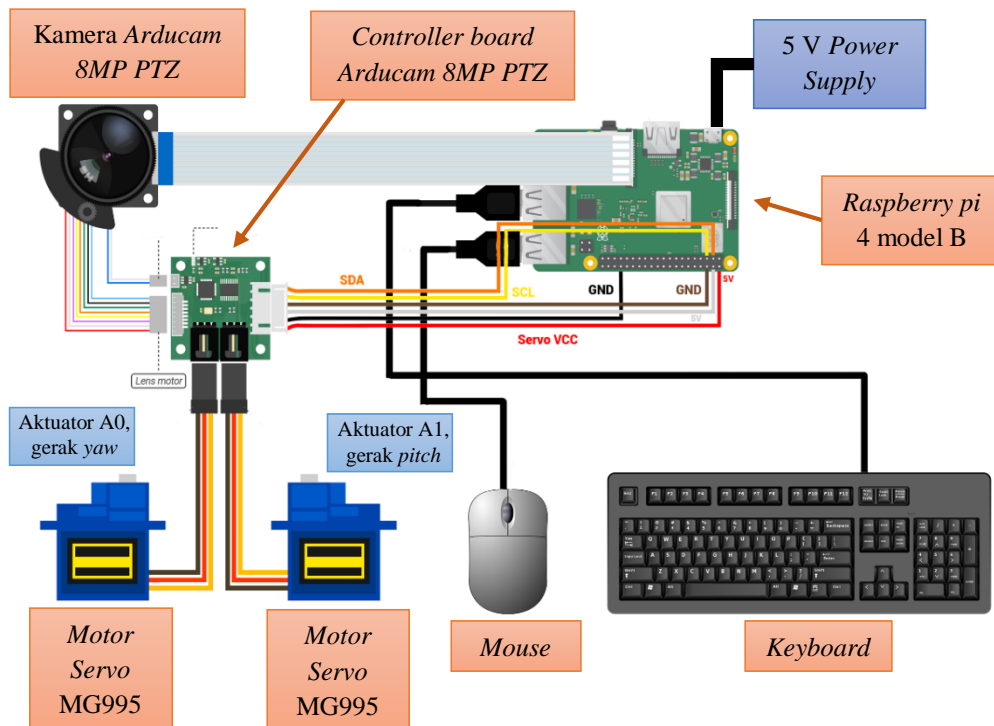
$$Y_error = 2,783804$$

$$u(t)Y = Y_P + Y_I + Y_D$$

$$\begin{aligned}
&= (k_{p_Y} e(t)Y) + (k_{i_Y} \int_0^t e(\tau) d\tau Y) + (k_{d_Y} \frac{de(t)Y}{dt}) \\
&= (k_{p_Y} \times e(t)Y) + (k_{i_Y} \times (e(t-1)Y + e(t)Y)) + (k_{d_Y} \times (e(t)Y - e(t-1)Y)) \\
&= (0,108 \times 1,086779784) + (0,112 \times (2,783804 + 1,086779784)) + \\
&\quad (0,026 \times (1,086779784 - 2,783804)) \\
&= 0,117372217 + 0,433505384 + (-0,0441226) = 0,506754971
\end{aligned}$$

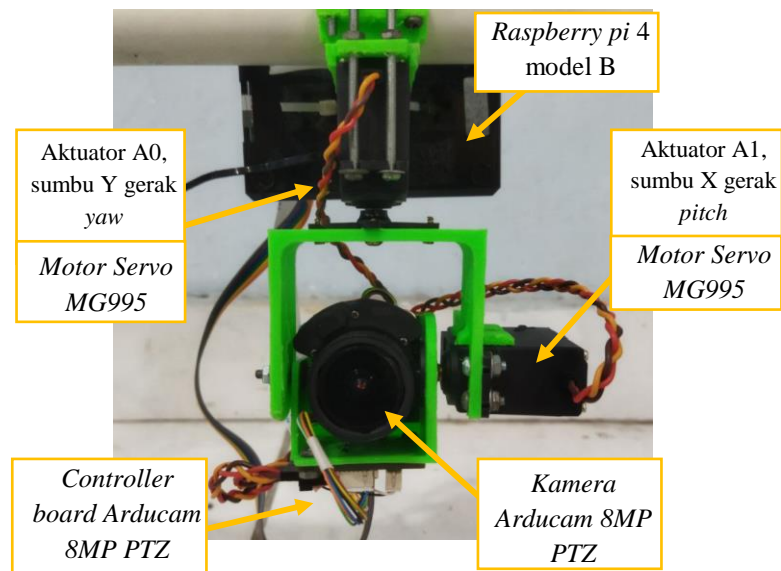
Nilai $u(t)X < 1$, maka aktuator A1 sumbu X_pitch berhenti bergerak dengan menghasilkan nilai $Y_centerROI$ sebesar 229 px dengan Y_error sebesar 12 px dari *setpoint* yaitu 240 px.

3.7. Desain Perancangan Alat



Gambar 3.13. Rangkaian Skematik Model Fisik Gimbal Kamera

Gambar 3.13. desain berdasarkan rangkaian skematik perancangan alat dapat dilihat terdapat *Raspberry pi 4* dihubungkan dengan *controller board Arducam 8MP PTZ* pada pin GPIO 2,3,4,5,6,9, *controller board Arducam 8MP PTZ* dihubungkan dengan kamera *Arducam 8MP PTZ* untuk menggerakkan aktuator lensa untuk melakukan perubahan penskalaan citra (*zoom*) dan fokus, juga dihubungkan dengan 2 aktuator yaitu aktuator A0 untuk gerak rotasi *yaw* dan aktuator A1 untuk gerak rotasi *pitch*, kemudian untuk mengirim citra video dari pengambilan kamera ke *Raspberry pi 4* dihubungkan dengan kabel fleksibel ke konektor *MIPI CSI camera port*. Kemudian terdapat juga *mouse* dan *keyboard* untuk masukan pemilihan objek dan pergerakan manual yang dilakukan oleh operator. Untuk perangkat lunak yang sudah dirangkai menjadi model fisik perangkat gimbal kamera dapat dilihat pada gambar 3.14.



Gambar 3. 14. Model Fisik Gimbal Kamera

Pada gambar 3.14. perangkat keras komponen gimbal kamera yang sudah dirangkai dimana kamera *Arducam 8MP PTZ* dan *Controller board Arducam 8MP PTZ* diletakan pada dudukan gimbal kamera yang ditopang oleh aktuator A0 yaitu *motor servo* MG 995 pergerakan rotasi *yaw* yang dipasang secara horizontal atau bersumbu Y dan aktuator A1 yaitu *motor servo* MG 995 pergerakan rotasi *pitch* yang dipasang secara vertikal atau bersumbu X.

V. PENUTUP

5.1. Kesimpulan

Kesimpulan dari penelitian ini adalah sebagai berikut:

1. Berhasil dibangun model fisik sistem kontrol pergerakan *pitch* dan *yaw* gimbal kamera yang dapat melakukan proses pelacakan objek.
2. Warna objek dengan warna *background* harus memiliki perbedaan sebesar 140 skala grayscale untuk mendapatkan pembacaan yang baik untuk proses pelacakan.
3. Penskalaan citra pada kamera mempunyai kemampuan memperbesar citra objek hingga 177% dari ukuran *zoom* minimum ke *zoom* maksimum.
4. Pelacakan objek pada sumbu Y gerak *yaw* dengan nilai konstanta PID ($K_P_X = 0,114$, $K_I_X = 0,141$, $K_D_X = 0,023$) dihasilkan respon gimbal dengan *rise time_X* sebesar 1,26s dengan nilai *error* rata-rata sebesar 6 px, pada sumbu X gerak *pitch* dengan nilai konstanta PID ($K_P_Y = 0,108$, $K_I_Y = 0,112$, $K_D_Y = 0,026$) dihasilkan respon gimbal dengan *rise time_Y* sebesar 1,36s dengan nilai *error* rata-rata sebesar 4,5 px.
5. Batas toleransi besar objek agar dapat dilacak dengan pelacakan *kernelized correlation filter* (KCF) pada gimbal kamera adalah sebesar 0,25 % dari besar layar pelacakan atau jika besar objek adalah 720 piksel sedangkan besar layar pelacakan adalah 307200 piksel.

5.2. Saran

Adapun saran untuk pekerjaan mendatang adalah sebagai berikut:

1. Menggunakan aktuator berupa motor DC yang memiliki keluwesan derajat pergerakan, sehingga presisi dimungkinkan untuk dicapai lebih baik.
2. Sistem dikembangkan untuk kemampuan melakukan pemilihan objek secara mandiri untuk tugas-tugas tertentu. Sehingga memudahkan operator untuk mendapatkan informasi objek dalam jumlah yang tidak tunggal.

DAFTAR PUSTAKA

- [1] T. Shiino, K. Kawada, T. Yamamoto, M. Komichi and T. Nishioka, "Gimbals Control with The Camera for Aerial Photography in RC Helicopter," *International Conference on Control, Automation and Systems : COEX, Seoul, Korea*, pp.1135-1139, 2008.
- [2] R. Cunha, M. Malaca and V. Sampaion, "Gimbal Control for Vision-based Target Tracking," *EURASIP European Signal Processing Conference (EUSIPCO) : Satellite Workshop, Spain*.2019.
- [3] X. Liu, Y. Yang, C. Ma, J. Li and S. Zhang, "Real-Time Visual Tracking of Moving Targets Using a Low-Cost Unmanned Aerial Vehicle with a 3-Axis Stabilized Gimbal System," *Applied Sciences*,10,5064, 2020.
- [4] S. P. Pertiwi, "Pemodelan dan Perancangan Sistem Kendali Gimbal Kamera 3-Sumbu Menggunakan Metode *Proportional Integral Derivative* (PID) pada *Unmanned Aerial Vehicle* (UAV)," *Skripsi, Teknik Elektro : Universitas Lampung*, 2020.
- [5] V. Naidu and B. Indhu, "Gimballed Camera Control For On-Point Target Tracking," *American Research Journal of Electronics and Communication* ,vol. 1, issue 1, 2015.
- [6] R. T. Yunardi and dkk, "Desain dan Implementasi Visual Object Tracking Menggunakan Pan and Tilt Vision System," *Jurnal ELKHA : Universitas Tanjungpura*, vol. 11, no. 2, pp. 85-92, 2019.
- [7] M.S.Haris and dkk," Sistem Kendali Gimbal 2-Sumbu Sebagai Tempat Kamera Pada Quadrotor Menggunakan PID Fuzzy," *IJEIS, Vol.7, No.2*, pp. 185-196, Oktober 2017.
- [8] Ishak, " Implementasi Metode Pulse Widht Modulation Pengontrol Deras Keluar Air Berdasarkan Jarak Halangan," *Jurnal Teknologi Sistem Informasi dan Sistem Komputer : TGD P:ISSN : 2621-8976 E-ISSN : 2615-5133* , vol. 2, no. 2, pp. 97-105, 2019.
- [9] Y. Ramadhan, "Sistem Kendali Kalang Tertutup dengan Metode PID pada Robot Wayang," *Skripsi, Teknik Elektro : Universitas Islam Indonesi*, 2018.
- [10] D. Jiang, Y. Liang dan G. Xu, "A new Image Zooming via the Split Bregman Method," *6th International Congress on Image and Signal Processing*, 2013.
- [11] A. Ammar, E.M. Saad, I. Ashour dan M. Elzorkany, "Image Zooming and Multiplexing Techniques based on K-Space Transformation," *International Journal of Signal Processing, Image Processing and Pattern Recognition*, vol. 5, no. 4, pp. 31-42, 2012

- [12] D. Zhou, X. Shen dan W. Dong, "Image Zooming Using Directional Cubic Convolution Interpolation," *The Institution of Engineering and Technology*, vol. 6, no. 6, pp. 627-634, 2012.
- [13] A. Fahriannur, R. Mardiyanto and M. Siswanto, "Sistem Pelacakan Objek Menggunakan Kombinasi Algoritma Optical Flow dan Template Matching," *Jurnal Teknologi dan Sistem Komputer*, vol. 6, no. 1, pp. 13-17, 2018.
- [14] D. A. Prabowo, D. Abdullah, A. Manik, "Deteksi dan Perhitungan Objek Berdasarkan Warna Menggunakan Color Object Tracking," *Universitas Muhammadiyah Bengkulu: Jurnal Pseudocode .ISSN 2355-5920* , vol. V, no. 2, 2018.
- [15] A. Yilman and dkk, "Object tracking: A survey.," *A survey. ACM Comput. Surv.* 38, 4, Article 13, pp. , 45 pages. 5, December 2006.
- [16] J. Henriques, R. Caseiro, P. Martins and J. Batista, "High-Speed Tracking with Kernelized Correlation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. KCF, p. 583–596. , 2015.
- [17] X.Wang, G. Wang, Z.Zhao, "An Improved Kernelized Correlation Filter Algorithm for Underwater Target Tracking," *Applied Sciences*,8,2154, 2018.
- [18] M. I. Ma'arif, F. I. Adhim and F. Istiqomah, "Implementasi Metode Pid untuk Mengontrol Posisi Motor Servo pada Sistem Sortir Berat Adonan," *Jurnal Teknik ITS*, vol. 10, no. 2, 2021.
- [19] K.J. Åström and T. Hägglund, "Revisiting the Ziegler-Nichols step response method for PID control," *J. Process Control*, vol. 14, no. 6, pp. 635-650, 2004.
- [20] N. Allu and S. Salu, "Aplikasi Penalaan dengan Metode Ziegler Nichols di Perancangan Pengendali PID pada Putaran Motor DC," *Universitas Kristen Indonesia Paulus Makassar:Sinergitas Multidisiplin Ilmu Pengetahuan dan Teknologi*, ISSN:2622-0520, vol. 1, 2018.