



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - TM 141585

# RANCANG BANGUN INSTRUMEN *FEEDBACK* KONTROL SUDUT MENGGUNAKAN METODE *IMAGE PROCESSING* TERHADAP MEKANISME BANDUL PADA SELF-BALANCING BIKE

SATRIO RAMADHAN  
NRP. 0211124000102

Dosen Pembimbing  
Arif Wahjudi, S.T., M.T., Ph.D

Laboratorium Perancangan dan Pengembangan Produk  
Departemen Teknik Mesin  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2018



**TUGAS AKHIR - TM 141585**

**RANCANG BANGUN INSTRUMEN *FEEDBACK*  
KONTROL SUDUT MENGGUNAKAN METODE  
*IMAGE PROCESSING* TERHADAP MEKANISME  
BANDUL PADA SELF-BALANCING BIKE**

**SATRIO RAMADHAN**  
NRP. 02111240000102

**Dosen Pembimbing**  
Arif Wahjudi, S.T., M.T., Ph.D

Laboratorium Perancangan dan Pengembangan Produk  
Departemen Teknik Mesin  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2018

*[Halaman Ini Sengaja Dikосongkan]*



**FINAL PROJECT - TM 141585**

**DESIGN AND BUILDING OF FEEDBACK  
INSTRUMENT FOR SELF-BALANCING BIKE'S  
TILT ANGLE CONTROL USING IMAGE  
PROCESSING METHODE TO PENDULUM  
MECHANISM**

**SATRIO RAMADHAN**  
NRP. 0211124000102

**Academic Supervisor**  
**Arif Wahjudi, S.T., M.T., Ph.D**

**Product Design and Development Laboratory**  
**Mechanical Engineering Department**  
**Faculty of Industrial Technology**  
**Sepuluh Nopember Institute of Technology**  
**Surabaya**  
**2018**

*[Halaman Ini Sengaja Dikosongkan]*

**RANCANG BANGUN INSTRUMEN FEEDBACK  
KONTROL SUDUT DENGAN METODE *IMAGE*  
PROCESSING TERHADAP MEKANISME BANDUL  
PADA *SELF-BALANCING BIKE***

**TUGAS AKHIR**

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
Pada  
Program studi S-1 Departemen Teknik Mesin  
Fakultas Teknik Industri  
Institut Teknologi Sepuluh Nopember

Oleh :

**SATRIO RAMADHAN**

**NRP. 02111240000102**

Disetujui oleh tim penguji tugas akhir :

1. Arif Wahjudi, ST., MSc., PhD. ..... (Pembimbing)  
NIP. 197303222001121001
2. Prof. Dr.Ing. Ir. I Made Londen B., M.Eng. ..... (Penguji I)  
NIP. 195811061986011002
3. Dr. Eng Unggul Wasiwitono, ST., M.Eng.Sc. ..... (Penguji II)  
NIP. 197805102001121001

**SURABAYA  
JANUARI, 2018**

*[Halaman Ini Sengaja Dikosongkan]*

# PERANCANGAN INSTRUMEN FEEDBACK KONTROL SUDUT MENGGUNAKAN METODE IMAGE PROCESSING TERHADAP MEKANISME BANDUL PADA *SELF-BALANCING BIKE*

**Nama** : Satrio Ramadhan  
**NRP** : 02111240000102  
**Departemen** : Teknik Mesin FTI-ITS  
**Dosen Pembimbing** : Arif Wahjudi, ST., MSc., Ph.D.

## ABSTRAK

Keseimbangan merupakan kebutuhan yang terdapat pada alat transportasi yang mengalami kemiringan sudut seperti pada sepeda. Pengembangan teknologi sebagai solusi permasalahan tersebut memerlukan sistem kontrol keseimbangan dengan menggunakan instrumen *feedback*, sebagai pengukur sudut kemiringan. Berbagai metode pengukuran dapat dimanfaatkan, dimana salah satu perancangan teknologi pengukuran yang dapat dimanfaatkan adalah metode pengolahan citra, atau lebih dikenal dengan *Image Processing*.

Langkah kerja penelitian ini pertama adalah merancang dan membangun hardware dengan memanfaatkan mekanisme bandul sebagai instrumen pengukur sudut. Mekanisme bandul akan diproses oleh komputer melalui kamera Logitech C-170 dengan frekuensi tangkap kamera sebesar 60fps. Setelah kamera berhasil terhubung dengan komputer, selanjutnya menggunakan program *image processing* sebagai pendeteksi pergerakan bandul secara *real-time* yang akan dianalisa sebagai kemiringan sudut, dengan bahasa pemrograman C++ dengan *software library* OpenCV. Program *image processing* memanfaatkan kalibrasi warna HSV yang bertujuan untuk membedakan objek pendulum dengan latar belakangnya. melakukan proses kalibrasi warna dengan memanfaatkan kedalaman warna dengan format HSV yang didapatkan dari *library OpenCV*. Pendeteksian bandul dilakukan dengan mendeteksi warna dari dua buah objek warna yang

ditempatkan pada sumbu putar bandul dan beban bandul. Setelah objek berhasil terdeteksi, nilai koordinat dua dimensi objek dicari dengan memanfaatkan nilai intrinsik kamera. Proses penentuan sudut dari posisi pusat dan beban bandul dihitung menggunakan *trigonometri*, kemudian mencari nilai error dari perbedaan besar sudut hasil pengukuran dengan sensor kalibrator *Inertial Measurement Unit*.

Hasil yang didapat dari Tugas Akhir ini adalah sebuah instrumen pengukur sudut beserta program yang bisa mendeteksi perubahan sudut kemiringan berdasarkan perubahan posisi bandul terhadap sumbu vertikal instrumen. Berdasarkan hasil kalibrasi dengan sensor IMU didapatkan nilai error pengukuran olah citra berkisar antara  $0,01^\circ$  hingga  $0,95^\circ$ . Adapun lamanya waktu *settling time* setelah dilakukan perbaikan pada mekanisme bandul dengan mereduksi panjang batang bandul adalah berkisar pada waktu satu detik. Instrumen hasil rancang bangun Tugas Akhir ini akan diaplikasikan pada tugas akhir rancang bangun *self-balancing bike*.

**Kata kunci: Image Processing, Open CV, Visual Studio, Trigonometri.**

# **DESIGN AND BUILDING OF FEEDBACK INSTRUMENT FOR SELF-BALANCING BIKE'S TILT ANGLE CONTROL USING IMAGE PROCESSING METHODE TO PENDULUM MECHANISM**

**Name** : Satrio Ramadhan  
**Reg. Number** : 0211124000102  
**Department** : Teknik Mesin FTI-ITS  
**Acadmic Supervisor** : Arif Wahjudi, ST., MSc., Ph.D.

## **ABSTRACT**

Balance is a requirement thing which contained in transportation with tilt angle, such as bicycle. The development of technology as a solution to that problem, will need a balance system that use an angle feedback instrument. Various measurement methods can be utilized, for example one of technology that can be used is such as design measurement instrument using Image Processing metode.

The first step of this research is to design and build the hardware by using the mechanism of pendulum as angle measurement insturment. The pendulum mechanism will be processed by a computer via a Logitech C-170 camera with a camera capture frequency of 30fps. After the camera successfully connected to the computer, then we use the image processing program as a real-time pendulum detection of movement, that will be analyzed as the angular slope, with the C++ programming language with the OpenCV library software. The image processing program utilizes HSV color calibration that aims to distinguish pendulum objects against their backgrounds. To perform color calibration process by utilizing color depth with HSV, where the format obtained from OpenCV library. Pendulum detection are done by detecting the color of two color objects placed on the axis

of rotary pendulum and at pendulum's bob. Once the object is successfully detected, the object's coordinate is gained by utilizing the camera's intrinsic value. The process of determining the angle of the central position and the pendulum load is calculated using trigonometry. After that we look for the error of the measurement angle compared with the Inertial Measurement Unit.

The results obtained from this Final Project is an angle measuring instrument and a program that can detect the angle of inclination based on changes in the pendulum position against the vertical axis of the instrument. Based on calibration result with IMU sensor, we get error of image measurement by value ranges from  $0,01^\circ$  to  $0,95^\circ$ . The length of settling time is reduced after the repair of the pendulum mechanism, by reducing the length of the pendulum. The result of this final design will be applied to the final project of self-balancing bike design.

**Keywords: Image Processing, Open CV, Visual Studio, Trigonometri**

## KATA PENGANTAR

Allhamdulillah, puji syukur kehadirat Allah SWT karena dengan karunia-Nya sehingga penulisan Tugas Akhir yang berjudul “**RANCANG BANGUN INSTRUMEN FEEDBACK KONTROL SUDUT MENGGUNAKAN METODE IMAGE PROCESSING TERHADAP MEKANISME BANDUL PADA SELF-BALANCING BIKE**” dapat terselesaikan. Tugas akhir ini disusun untuk memenuhi salah satu persyaratan kelulusan pendidikan sarjana S-1 di Departemen Teknik Mesin, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember, Surabaya.

Pada kesempatan ini, penulis ingin mengucapkan terima kasih kepada:

1. Allah SWT yang telah memberikan semua anugerah, berkah serta hidayah kepada penulis.
2. Ayahanda serta ibunda tercinta, Herry Hermawan dan Farida bin Umartono yang selalu membimbing dan menasehati.
3. Putri Yuanita dan Latifah Hermawati selaku kakak penulis yang selalu mendukung untuk selalu belajar menjadi lebih baik.
4. Bapak Arif Wahjudi, ST., Msc., PhD. selaku dosen pembimbing penulis, yang telah membimbing dan mengarahkan penulis dalam menyelesaikan Tugas Akhir ini.
5. Prof. Dr.Ing. Ir. I Made Londen Batan, M.Eng., MSc., selaku dosen wali selama penulis berkuliah serta menjadi penguji penulis yang telah memberikan saran dan masukan dalam penyusunan Tugas Akhir ini.
6. Dr. Eng Unggul Wasiwitono, ST., M.Eng.Sc. sebagai dosen penguji yang telah memberikan saran dan pembelajaran dalam penyusunan Tugas Akhir ini.
7. Dimas Angga Pradiga, ST. atas bimbingan dan ilmu selama ini.
8. Alm. Haditya Zulkarnain, Alm. Laudy Tirta Madika, tim *Bounty Hunter* dan segenap angkatan M-55 yang tidak bisa saya sebutkan satu-persatu atas pelajaran dan pengalaman yang berharga selama berkuliah di kampus Teknik Mesin ITS.

9. Mario M. H, Bapak Kholiq D, ST., Bapak Hiding Cahyono S., Tedyta N.A.S.L, Abi Nubli., Febriana P.P, ST., Radian F. Rahmam., Tubagus B.P, Rosadila F., Ray Raditya, Mas'ud Asadullah, Hafizh N.P, Agung Wahyu S, Rahmat B.S dan semua warga Lab P3 yang tidak bisa saya sebutkan satu persatu.
10. Pengurus MMC tahun 2013/2014, seluruh civitas akademik Teknik Mesin ITS,

Sekalipun tugas akhir ini selesai melalui proses yang cukup panjang dengan segala keterbatasan kemampuan serta pengetahuan penulis, tidak menutup kemungkinan tugas akhir ini jauh dari kata sempurna. Oleh karena itu penulis berharap kritik dan saran yang membangun dari berbagai pihak untuk penyempurnaan lebih lanjut. Penulis berharap tugas akhir ini dapat bermanfaat bagi semua pihak, khususnya bagi seluruh rakyat Indonesia.

Surabaya, 19 Januari 2018

Penulis

## DAFTAR ISI

<b>JUDUL.....</b>	<b>i</b>
<b>LEMBAR PENGESAHAN.....</b>	<b>v</b>
<b>ABSTRAK.....</b>	<b>vii</b>
<b>KATA PENGANTAR.....</b>	<b>xi</b>
<b>DAFTAR ISI.....</b>	<b>xiii</b>
<b>DAFTAR GAMBAR.....</b>	<b>xvii</b>
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1    Latar Belakang.....	1
1.2    Rumusan Masalah.....	2
1.3    Batasan Masalah.....	2
1.4    Tujuan Penelitian.....	2
1.5    Manfaat Penelitian.....	3
<b>BAB II TINJAUAN PUSTAKA.....</b>	<b>5</b>
2.1    Penelitian Sebelumnya.....	5
2.2    Dasar Teori.....	6
2.2.1    Webcam.....	6
2.2.2    Gambar Digital.....	6
2.2.3    Kedalaman Warna.....	7
2.3    OpenCV.....	10
2.4    Mekanisme Pendulum.....	11
2.5    Getaran Bebas dengan <i>Coulomb Damping</i> .....	12

<b>BAB III METODE PENELITIAN .....</b>	<b>15</b>
3.1 Tahapan Penelitian.....	15
3.2 Diagram Alir Tahapan Penelitian.....	17
3.3 Diagram Alir Pengujian Pengukuran Sudut .....	18
<b>BAB IV PERANCANGAN DAN PEMBUATAN INSTRUMEN .....</b>	<b>19</b>
4.1 Konsep Instrumen .....	19
4.2 Perancangan Komponen .....	20
4.2.1 Perancangan Pelat Dasar Instrumen .....	21
4.2.2 Perancangan Dudukan Bandul Dan Kamera.....	21
4.2.3 Perancangan Batang Bandul.....	22
4.3 Perhitungan Mekanisme Bandul .....	23
4.4 Kalibrasi dengan Inertial Measurement Unit .....	26
4.5 Implementasi Program dalam Sistem .....	28
4.6 Konstruksi Program <i>Image Processing</i> .....	28
4.6.1 Program Pendeteksian Objek.....	29
4.6.2 Program Thresholding .....	31
4.6.3 Program Erode dan Dilate .....	33
4.6.4 Program Pendeteksian Objek.....	35
4.6.5 Pencarian Variabel Pendeteksian Objek .....	36
4.6.6 Penunjukan Objek Terdeteksi.....	38
4.6.7 Program Penentuan Sisi Depan, Sisi Sdan Sudut.....	39
4.7 Perakitan Sistem Kalibrasi Instrumen.....	42

<b>BAB V PENGUJIAN INSTRUMEN .....</b>	<b>43</b>
5.1    Pendeteksian Objek.....	43
5.2    Pengukuran Sudut .....	44
5.3    Respon Time Bandul.....	47
5.4    Evaluasi Desain .....	51
<b>BAB VI PENUTUP .....</b>	<b>53</b>
6.1    Kesimpulan .....	53
6.2    Saran.....	54
<b>LAMPIRAN A .....</b>	<b>57</b>
<b>LAMPIRAN B .....</b>	<b>59</b>
<b>LAMPIRAN C .....</b>	<b>61</b>
<b>BIODATA PENULIS.....</b>	<b>69</b>

*[Halaman Ini Sengaja Dikosongkan]*

## DAFTAR GAMBAR

Gambar 2.1 Logitech C170 .....	6
Gambar 2.2 Warna RGB .....	8
Gambar 2.3 <i>Binary Image</i> .....	9
Gambar 2.4 Model warna HSV .....	9
Gambar 2.5 Sistem Pendulum .....	11
Gambar 3.1 Diagram alir metodologi penelitian .....	17
Gambar 3.2 Diagram alir pengujian pengukuran .....	18
Gambar 4.1 Rancangan instrument .....	19
Gambar 4.2 Plat Dasar Instrumen .....	21
Gambar 4.3 Dudukan Bandul .....	22
Gambar 4.4 Dudukan Kamera .....	22
Gambar 4.5 Rancangan Batang Bandul .....	22
Gambar 4.6 3-DOF Inertial Measurement Unit .....	26
Gambar 4.7 Sinyal Accelerometer .....	27
Gambar 4.8 Diagram konversi sinyal menjadi sudut. ....	27
Gambar 4.9 Diagram alir pengolahan RGB ke HSV .....	30
Gambar 4.10 Konversi RGB menjadi HSV .....	31
Gambar 4.11 Diagram alir proses <i>threshold</i> .....	32
Gambar 4.12 Hasil <i>Threshold</i> dari HSV .....	33
Gambar 4.13 Diagram alir proses <i>erode</i> dan <i>dilate</i> .....	34
Gambar 4.14 Hasil proses <i>erode-dilate</i> .....	35
Gambar 4.15 <i>Trackbar</i> range warna HSV .....	36

Gambar 4.16 Diagram alir pencarian koordinat objek .....	37
Gambar 4.17 Diagram alir pembuatan lingkaran .....	38
Gambar 4.18 Hasil Penentuan sisi segitiga dan sudut .....	40
Gambar 4.19 Diagram alir penentuan sisi dan sudut.....	41
Gambar 4.20 Foto Rangkaian Instrumen.....	42
Gambar 5.1 Hasil pendeteksian objek LED. ....	43
Gambar 5.2 Pembacaan hasil pengukuran.....	44
Gambar 5.3 Pengukuran rentang $-30^\circ \leq \Theta < 0^\circ$ . ....	45
Gambar 5.4 Error vs. Sudut pada $-30^\circ \leq \Theta < 0^\circ$ .....	45
Gambar 5.5 Grafik Pengukuran pada $0^\circ < \Theta \leq 30^\circ$ .....	46
Gambar 5.6 Error vs. Sudut pada $0^\circ < \Theta \leq 30^\circ$ .....	47
Gambar 5.7 <i>Respon Time</i> pada <i>Set-Point</i> $30^\circ$ . ....	48
Gambar 5.8 <i>Respon Time</i> pada <i>Set-Point</i> $15^\circ$ . ....	49
Gambar 5.9 <i>Respon Time</i> Olah Citra $r_{\text{bandul}}= 1\text{cm}$ , <i>Set-Point</i> $15^\circ$ . ....	50
Gambar 5.10 <i>Respon Time</i> Olah Citra $r_{\text{bandul}}= 1\text{cm}$ , <i>Set-Point</i> $30^\circ$ . ....	51

# BAB I

## PENDAHULUAN

### 1.1 Latar Belakang

Pengendalian keseimbangan merupakan permasalahan yang memiliki daya tarik, dimana salah satu solusinya adalah menggunakan giroskop dengan sistem terkontrol. Seperti penelitian *Self-Balancing Bike* yang sedang dikembangkan khususnya pada Laboratorium Perancangan dan Pengembangan Produk Teknik Mesin ITS. Pada keadaan statis, sepeda akan terjatuh akibat tidak memiliki kestabilan keseimbangan, lain halnya saat sepeda melaju dengan arah lurus kedepan sehingga mengalami keseimbangan untuk tetap berdiri. Saat ini telah banyak dikembangkan *self-balancing bike* yang menggunakan *control moment gyroscope* sebagai *actuator* sistem penyeimbangannya. Pada umumnya *gyroscope* banyak diaplikasikan sebagai alat orientasi kemiringan pada industri pesawat terbang.

Sistem penyeimbang sepeda otomatis memerlukan sensor pengukur sudut sebagai subsistem feedback yang menghasilkan pengukuran kemiringan yang tepat. Berdasarkan permasalahan di atas maka muncul lah ide suatu sistem penentuan sudut, menggunakan metode image processing. Dengan memanfaatkan metode pengolahan citra dan perangkat mono kamera, diusulkan sebuah aplikasi pendeteksiian suatu target yang dapat merepresentasikan kemiringan. Aplikasi ini nantinya akan digunakan pada *self balancing bike*. Sehingga alat ini dapat mengetahui kemiringan sudut sepeda secara otomatis ketika kamera menangkap gambar.

Metode pengolahan citra dapat digunakan untuk memperoleh data berupa posisi maupun kemiringan sudut. *Input* yang berupa gambar atau video dapat diperoleh menggunakan mono kamera. Proses pengolahan citra sendiri terbagi menjadi 3 tahap, yaitu *pre-processing*, *processing*, dan *post-processing*. *Pre-processing* sendiri adalah proses kalibrasi warna *HSV* dan *thresholding* dimana gambar disegmentasi agar dapat

membedakan objek yang akan diproses dengan latar belakangnya. Kemudian *processing* merupakan tahap pendeteksian objek pada gambar. Tahap terakhir adalah *post-processing* dimana pengguna mengekstraksi informasi yang dibutuhkan dari gambar. Gambar yang didapat dari kamera nantinya dapat diolah dengan menggunakan bantuan perangkat lunak.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang dari penelitian ini, maka dapat diambil beberapa masalah yang akan dipecahkan antara lain sebagai berikut:

1. Bagaimana mengukur kemiringan sepeda dengan memanfaatkan metode pengolahan citra?
2. Berapa kesalahan pengukuran kemiringan sepeda dengan metode olah citra dibandingkan dengan hasil pengukuran dengan sensor *IMU*?

## 1.3 Batasan Masalah

Untuk membatasi permasalahan yang harus diselesaikan pada tugas akhir ini, maka rumusan masalah ditentukan yang antara lain adalah sebagai berikut:

1. Menggunakan sebuah kamera (mono *camera*) yang posisinya tidak bergerak.
2. Resolusi perhitungan sudut dari kamera sebesar  $1^\circ$
3. Kapasitas Perhitungan sudut  $-30^\circ$  hingga  $30^\circ$
4. Menggunakan objek warna sebagai representasi perubahan posisi bandul.
5. Kamera mengarah tegak lurus terhadap pendulum.

## 1.4 Tujuan Penelitian

Adapun berdasarkan latar belakang dan masalah yang dirumuskan maka penelitian mengambil beberapa tujuan dari tugas akhir ini adalah sebagai berikut:

1. Mengetahui cara mengukur kemiringan sepeda dengan memanfaatkan metode pengolahan citra.

2. Mengetahui kesalahan pengukuran kemiringan sepeda dengan metode olah citra dibandingkan dengan hasil pengukuran dengan sensor *IMU*.

### **1.5 Manfaat Penelitian**

Manfaat dari tugas akhir ini adalah sebagai berikut:

1. Hasil rancang bangun instrumen pengukur sudut dengan metode olah citra dapat digunakan sebagai alat ukur sudut.
2. Dapat dijadikan sebagai referensi pada pengembangan metode olah citra dalam penelitian selanjutnya.
3. Dapat digunakan untuk pengembangan otomatisasi sistem penyeimbang kendaraan transportasi roda dua.

*[Halaman Ini Sengaja Dikosongkan]*

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1 Penelitian Sebelumnya**

Kuihe Yang pada tahun 2013 telah mengembangkan penggunaan teknologi olah citra yang memanfaatkan kamera sebagai kamera pengawas. Metode pengolahan citra ini dilakukan dengan melakukan pendeteksian objek bergerak yang dikombinasikan dengan metode pengurangan *background*. Pendeteksian objek masih terlalu luas dan objek yang dideteksi belum bisa digunakan untuk pengukuran sudut [1].

Sedangkan pengukuran kemiringan secara *real-time* telah diterapkan pada penelitian terdahulu oleh Mukeshkumar Prasad pada tahun 2016. Pada penelitian ini membahas perancangan dan fabrikasi pada *Automatic Balancing Bicycle*. Dalam mencari nilai sudut kemiringan sepeda, penelitiannya menggunakan sensor sudut berupa triple axis Accelerometer ADXL335 dan juga dual axis Gyroscope IDG500, data keduanya diintegrasikan menggunakan *complementary filter*. Sistem tersebut menggunakan PID controller untuk arah putar dengan Motor DC 24V. Seperti accelerometer pada umumnya, resolusi yang dihasilkan oleh sensor *IMU* sangat rentan terjadi error yang dapat mempengaruhi kecermatan perhitungan sensor pengukur sudut [2].

Adapun penelitian tentang pengukuran sudut menggunakan teknik *Image Processing* pernah dilakukan Young-Soo Park pada tahun 2015. Penelitian membuat sistem pengukur kemiringan sudut yang terjadi akibat deformasi pada jembatan. Sistem pengukuran tersebut menghasilkan kemiringan sudut deformasi pada jembatan yang akan dibandingkan dengan hasil pengukuran sudut dengan menggunakan *inclinometer*, melalui uji coba langsung pada jembatan. Penelitian ini belum dapat diterapkan pada sistem *self-balancing bike*, karena sistem yang digunakan adalah untuk mengukur kemiringan sudut akibat deformasi bukan akibat gravitasi. Mekanisme pengukuran tersebut belum sesuai untuk pengukuran kemiringan sepeda [3].

## 2.2 Dasar Teori

### 2.2.1 Webcam

Webcam merupakan kamera digital yang berfungsi sebagai alat input yang dapat memproses gambar secara real time dengan menangkap cahaya melewati lensa yang kemudian diterima oleh *image-sensing microchip*, antara lain bisa berupa *charge-coupled device (CCD)* maupun CMOS image sensor. Umumnya webcam mampu menangkap gambar pada resolusi 640 x 480 dengan kapasitas 30 fps. Sedangkan Kamera yang digunakan pada penelitian ini HD Webcam C170 Logitech seperti yang tampak pada Gambar 2.1, dengan kemampuan tangkap XVGA 1024 x 768 pixels [4].



Gambar 2.1 Logitech C170

### 2.2.2 Gambar Digital

Gambar digital merupakan citra  $[m, n]$  yang dijelaskan dalam ruang *discrete 2D* yang berasal dari citra analog 2D  $(x, y)$  melalui proses pengambilan sampel yang disebut sebagai digitalisasi. Citra *continue 2D*  $a(x, y)$  dibagi menjadi baris  $N$  dan kolom  $M$ . Itu persimpangan baris dan kolom disebut piksel.

Sebenarnya, dalam banyak kasus Citra analog  $(x, y)$  yang biasa dianggap sebagai fisik sinyal yang menimpa sensor 2D, sesungguhnya merupakan fungsi dari banyak variabel

termasuk kedalaman ( $z$ ), warna ( $\lambda$ ), dan waktu ( $t$ ) [5]. Fungsi intensitas cahaya dapat ditulis sebagai  $f(x,y)$  yang mana  $f$  merupakan nilai amplitude pada koordinat  $x$  dan  $y$  [6]. Secara matematis, gambar digital dapat dituliskan sebagai matriks, seperti ditunjukkan pada persamaan:

$$f(x, y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0, N - 1) \\ f(1,0) & f(1,1) & \dots & f(1, N - 1) \\ \vdots & \vdots & & \vdots \\ f(M - 1,0) & f(M - 1,1) & \dots & f(M - 1, N - 1) \end{bmatrix} \quad (2.1)$$

### 2.2.3 Kedalaman Warna

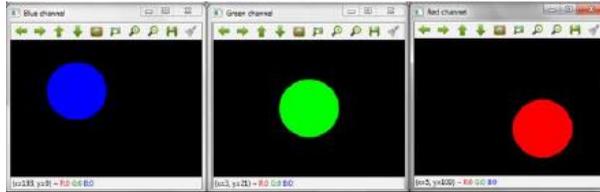
Sistem visual manusia mampu membedakan ratusan ribu warna. Untuk mendapatkan informasi ini, retina manusia memiliki tiga jenis fotoreseptor warna, yang merespons radiasi kejadian. Karena ini, kebanyakan warna manusia Persepsi dapat dihasilkan dengan tiga komponen numerik yang disebut primary. Untuk menentukan warna dalam hal tiga atau lebih karakteristik tertentu, ada metode yang disebut ruang warna atau model warna.

Memilih di antara mereka Untuk mewakili gambar tergantung pada operasi yang akan dilakukan. Pada beberapa Ruang warna seperti RGB, kecerahan mempengaruhi tiga saluran, hal tersebut yang bisa menjadi kekurangan untuk melakukan beberapa operasi pemrosesan gambar [7]. Berdasarkan warna-warna penyusunnya gambar digital memiliki beberapa macam warna penyusun, adapun warna penyusun yang akan dimanfaatkan dalam penelitian ini adalah sebagai berikut:

#### 1. Gambar Berwarna (RGB)

Gambar berwarna merupakan gambar yang memiliki informasi warna di setiap pikselnya. Informasi warna ini dapat dibentuk dari gabungan komponen-komponen satu set *channel* warna. Gambar berwarna memiliki format gambar RGB (*Red Green Blue*) yang

mana warna yang dihasilkan tiap pikselnya terdiri dari gabungan informasi nilai dasar merah, hijau dan biru.



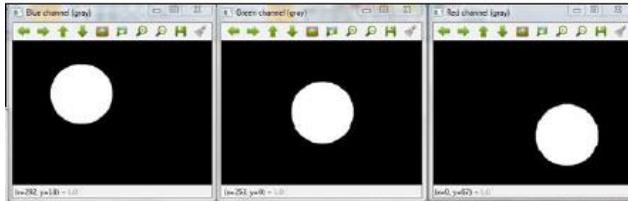
Gambar 2.2 Warna RGB

RGB adalah model warna dimana gambar terdiri dari tiga citra independen yaitu merah, hijau, dan biru seperti yang ditunjukkan pada Gambar 2.2. Dalam menentukan warna tertentu, masing-masing nilai menunjukkan jumlah masing-masing komponen yang ada pada setiap piksel, dengan nilai yang lebih tinggi sesuai dengan piksel yang lebih terang. Ruang warna ini banyak digunakan karena sesuai dengan tiga fotoreseptor mata manusia [7].

## 2. Gambar Biner

Gambar biner merupakan gambar yang hanya memiliki dua macam warna, yaitu putih dan hitam. Pada gambar biner, setiap piksel hanya memiliki 2 macam nilai, yaitu 1 untuk warna hitam dan 0 untuk warna putih. Pada gambar 2.3 dapat dilihat gambar dengan format biner, yang mana gambar hanya memiliki warna hitam dan putih. Gambar biner pada umumnya didapatkan melalui proses *thresholding*. *Thresholding* merupakan proses segmentasi gambar. Segmentasi gambar sendiri digunakan untuk membedakan antara objek dengan latar belakangnya. Proses segmentasi sendiri dilakukan dengan mengisolasi objek pada gambar yang berbentuk format *grayscale* menjadi format biner. Pada saat melakukan proses *thresholding* perlu diperhatikan nilai *threshold* yang akan

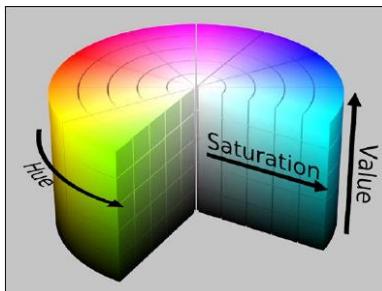
ditentukan. Nilai piksel yang berada di bawah nilai *threshold* akan diubah menjadi warna hitam, sedangkan piksel yang memiliki nilai di atas nilai *threshold* akan diubah menjadi warna putih.



Gambar 2.3 Binary Image

### 3. Gambar HSV

Model warna lain yang dapat dimanfaatkan adalah *hue*, *saturation*, *value* (HSV). HSV banyak digunakan untuk membuat perbandingan warna karena Hue adalah variasi cahaya yang hampir independen.



Gambar 2.4 Model warna HSV

Gambar 2.5 berikut menunjukkan model warna ini mewakili masing-masing nilai warna sebagai bagian dari silinder, yang terdiri dari 3 komponen yaitu Hue, Saturation, Value [7].

Dalam mengubah citra RGB menjadi citra HSV secara manual dapat digunakan persamaan (2.2) dan (2.3).

$$V = \max(r, g, b)$$

$$S = \begin{cases} 0, & v = 0 \\ 1 - \frac{\min(r, g, b)}{v}, & v > 0 \end{cases} \quad (2.2)$$

$$H = \begin{cases} 0, & \text{jika } S = 0 \\ \frac{60*(g-b)}{s*v}, & \text{jika } v = r \\ 60*\left[2 + \frac{b-r}{s*v}\right], & \text{jika } v = g \\ 60*\left[4 + \frac{r-g}{s*v}\right], & \text{jika } v = b \end{cases} \quad (2.3)$$

Jika Saturation  $S=0$ , maka hue tidak terdefinisi atau dengan kata lain tidak memiliki hue berarti *monochrome*. Hue (H) lalu dikonversi menjadi derajat/degrees dengan cara mengalikan dengan 60 sehingga menghasilkan HSV dengan S dan V antara 0 dan 1 dan H antara 0 – 360.

### 2.3 OpenCV

OpenCV (Open Source Computer Vision) adalah *cross-platform library* untuk pemrosesan gambar real-time. Perangkat lunak ini adalah alat standar untuk segala hal yang berhubungan dengan *Computer Vision*. Versi pertama dirilis pada tahun 2000 di bawah lisensi BSD dan sejak saat itu, fungsinya telah sangat banyak dikembangkan.

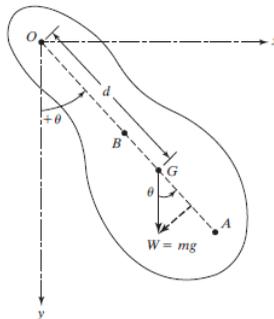
OpenCV (*Open Source Computer Vision*) merupakan sebuah *Open Source BSD-licensed library* yang berisi lebih dari 500 algoritma yang telah dioptimalisasi dengan baik untuk pengolahan gambar dan video analisis. *Library* ini ditulis dengan bahasa C dan C++ dan dapat digunakan dengan *operating system* Windows, Linux, dan Mac OS X. OpenCV dirancang untuk efisiensi komputasional.

Tujuan dari aplikasi ini yaitu agar komputer memiliki kemampuan yang mirip dengan cara pengolahan visual pada manusia. *Library* ini dibuat untuk bahasa C/C++ sebagai

optimasi aplikasi *real-time*. OpenCV memiliki API yang merupakan singkatan dari *Application Programming Interface* yang digunakan untuk pengolahan tingkat tinggi maupun tingkat rendah. Pada OpenCV juga terdapat fungsi – fungsi siap pakai untuk mengambil, menyimpan, serta mengakuisisi gambar dan video.

## 2.4 Mekanisme Pendulum

Mekanisme Pendulum merupakan sistem gerak yang telah diterapkan dalam pengembangan teknologi sejak berabad-abad lamanya. Berbagai aplikasi telah diterapkan berdasarkan prinsip kerja pendulum antara lain pada alat pengukur kecepatan gravitasi *Gravimeter*, *Clock Pendulum*, *Focault Pendulum* sebagai alat demonstrasi gerak putar bumi dan lain sebagainya. Benda apapun yang memiliki pusat putar selain di pusat massa benda itu sendiri akan mengalami osilasi ketika diberikan gaya gravitasi.



Gambar 2.5 Sistem Pendulum

Penelitian ini memanfaatkan mekanisme gerak pendulum sebagai representasi perubahan sudut *self-balancing bike* terhadap arah percepatan gravitasi bumi. Setiap benda *rigid* yang berputar pada titik selain pusat massanya akan berosilasi di bawah gaya gravitasinya sendiri. Sistem seperti ini dikenal sebagai *compound pendulum*.

Persamaan gerak sudut pendulum terhadap porosnya dapat diturunkan dengan menggunakan hukum kedua Newton, sehingga didapatkan persamaan gerak sebagai berikut [8]:

$$J_o \ddot{\Theta} + W.d \sin\Theta = 0 \quad (2.4)$$

dimana  $J_o = \text{moment of inertia (g.cm}^2\text{)}$   
 $\ddot{\Theta} = \text{percepatan sudut. (rad/ s}^2\text{)}$   
 $W = \text{Berat massa (g.cm/s}^2\text{)}$   
 $d = \text{Panjang bandul (cm)}$   
 $\Theta = \text{Sudut.(rad)}$

Dengan frekuensi natural ( $\omega_n$ ) dari *compound pendulum* didapatkan bisa dicari dengan persamaan (2.5)

$$\omega_n = \left( \frac{(W.d)}{J_o} \right)^{1/2} \quad (2.5)$$

Adapun periode bandul dapat dicari dengan menggunakan rumus periode pendulum rigid[11] sebagai berikut:

$$T = 2\pi \left( \frac{I}{m.g R} \right)^{1/2} \quad (2.6)$$

dimana  $T = \text{Periode Pendulum (s)}$   
 $m = \text{massa rigid pendulum. (g)}$   
 $R = \text{Jari-jari pusat massa}$   
 $\text{terhadap pusat putar (cm)}$   
 $g = \text{percepatan gravitasi (cm/s}^2\text{)}$   
 $I = \text{momen inersia total.( g.cm}^2\text{)}$

## 2.5 Getaran Bebas dengan *Coulomb Damping*

Dalam banyak sistem mekanis, *Coulomb* atau *dry-friction damping* digunakan karena kesederhanaan mekanis dan kenyamanan. Juga pada struktur yang bergetar, setiap komponen bergeser relatif terhadap satu sama lain, *dry-friction damping* akan muncul. *Coulomb Damping* muncul saat tubuh meluncur di permukaan yang kering. Hukum *Coulomb*

*Coulomb's Law* menyatakan bahwa ketika dua benda bersentuhan, kekuatan yang dibutuhkan untuk menghasilkan gesekan merupakan proporsional terhadap gaya normal pada permukaan mekanisme yang bekerja. Sehingga gaya gesek didapatkan sebagai berikut:

$$F = \mu \cdot N = \mu \cdot W \quad (2.7)$$

dimana  $\mu = \text{Koefisien Gesek.}$

Apabila gaya gesek bekerja pada sistem pendulum maka berdasarkan persamaan (2.4) dan (2.6) dapat ditentukan persamaan (2.7) gerak rotasional pendulum sebagai berikut:

$$\begin{aligned} J_o \ddot{\Theta} + W \cdot d \sin \Theta - T &= 0 \\ J_o \ddot{\Theta} + W \cdot d \sin \Theta - \mu \cdot W \cdot r &= 0 \\ J_o \ddot{\Theta} + W \cdot d \sin \Theta - \mu \cdot m \cdot g \cdot r &= 0 \end{aligned} \quad (2.8)$$

dimana  $r = \text{Jari-jari benda yang mengalami gesekan (cm)}$   
 $g = \text{Percepatan gravitasi (981 cm/s}^2\text{)}$   
 $T = \text{Torsi Coulomb Damping (g.cm}^2\text{/s}^2\text{)}$   
 $W = \text{Beban Bandul (g)}$   
 $d = \text{Jarak beban terhadap sumbu putar (cm)}$

Sedangkan dari banyak nya siklus osilasi yang dapat dihasilkan dari sistem pendulum perlu diketahui estimasi kapan gerakan akan berhenti yang dapat dicari dengan persamaan (2.8), gerakan akan berhenti pada siklus  $R$  [8].

$$R \geq \left( \frac{\Theta_0 - (T/W \cdot d)}{(2 \cdot T/W \cdot d)} \right) \quad (2.9)$$

dimana  $R = \text{Jumlah setengah siklus.}$   
 $g = \text{Percepatan gravitasi (981 cm/s}^2\text{)}$   
 $\Theta_0 = \text{Simpangan sudut awal. (rad)}$

Adapun untuk menghitung momen inersia ( $J_0$ ) dapat dengan mendefinisikannya dengan integral moment terhadap sumbu putaran [9].

$$J_0 = \int_m \mathbf{r}^2 d\mathbf{m} = \int_V \mathbf{r}^2 \cdot \rho \cdot dV \quad (2.10)$$

dimana  $r$  = Jari-jari putar (cm).

$\rho$  = Massa Jenis ( $\text{g/cm}^3$ ).

$J_0$  = Simpangan sudut awal. (rad).

Sedangkan karakteristik pada sistem gerak Coulomb Damping pada umumnya akan memiliki posisi akhir yang tidak sama dengan posisi *equilibrium*-nya[8] dan mempresentasikan simpangan permanen yang dapat dirumuskan sebagai berikut:

$$\theta_r = \theta_0 - R \frac{2T}{Wd} \quad (2.11)$$

dimana  $R$  = Jumlah Setengah Siklus (cm).

$T$  = Torsi Coulomb Damping ( $\text{g.cm}^2/\text{s}^2$ )

$\theta_0$  = Simpangan sudut awal. ( $^\circ$ )

## **BAB III**

### **METODE PENELITIAN**

Pada Bab ini dijelaskan metode penelitian langkah-langkah sistematis sebagai acuan kerangka penelitian untuk membuat instrument pendeteksi kemiringan yang memanfaatkan kemiringan bandul dengan metode olah citra.

#### **3.1 Tahapan Penelitian**

Dalam mencapai tujuan penelitian, maka digunakan tahapan-tahapan penelitian sebagai berikut:

##### **1. Studi Literatur**

Langkah awal dari penelitian ini adalah melakukan kajian literatur untuk menentukan metode pengolahan citra apa saja yang akan digunakan dan mengumpulkan dasar teori yang menjadi landasan pembuatan sistem pendulum. Studi Literatur diambil dari buku referensi, jurnal, diktat kuliah dan internet.

##### **2. Perumusan Masalah**

Tahap penelitian ini adalah tahap merumuskan masalah yang bertujuan untuk menentukan pemecahan dari masalah perancangan dan pembuatan instrumen pengukur sudut dengan metode olah citra terhadap mekanisme pendulum.

##### **3. Perancangan dan Pembuatan Instrumen**

Dari hasil perhitungan mekanisme bandul dan kemampuan daya tangkap kamera, maka selanjutnya dilakukan perancangan komponen instrument pengukur sudut. Setelah hasil rancangan berhasil dibuat selanjutnya komponen instrument akan dirakit, sesuai kebutuhan.

##### **4. Pembuatan Program**

Setelah Instrumen tersedia selanjutnya melakukan pembuatan program pengolahan citra digital untuk proses pengukuran sudut dengan bahasa program C++, dengan menggunakan perangkat lunak Microsoft Visual Studio 2013 dan pustaka program OpenCV 2.4.10.

##### **5. Pengujian Pendeteksia Objek**

Untuk memastikan apakah program dan rancangan instrument

telah sesuai dan mampu mendeteksi bandul, pada tahap ini dilakukan pengujian program untuk mampu mendeteksi bandul. Pengujian dilakukan dengan menjalankan program dengan berbagai variasi posisi objek. Kemudian dari hasil deteksi objek tersebut didapatkan data berupa koordinat objek bandul pada gambar digital ( $x$ ,  $y$ ) dan  $\Theta$  bandul kemiringan bandul dari hasil perhitungan dua objek pendulum.

#### 6. Kalibrasi Pengujian Sudut

Setelah program berhasil mendeteksi pendulum dilakukan kalibrasi instrument pengukur sudut terhadap sensor kemiringan sudut Motion Node *Inertial Measurement Unit* (IMU). Dalam penelitian ini sensor IMU yang digunakan telah terkalibrasi [10], produk merupakan hasil pengembangan dari Motion Workshop, Seattle Amerika. Batas *error* yang ditoleransikan dalam kalibrasi ini adalah  $error \geq 1^\circ$ , dimana besar toleransi tersebut ditentukan berdasarkan kebutuhan yang disesuaikan dengan kebutuhan feedback dari rancang bangun penelitian selanjutnya yaitu *self-balancing bike*. Apabila besar *error* yang didapatkan masih melebihi dari satu derajat maka akan perbaikan dilakukan baik dengan menambahkan kompensasi *error* pada program pengolahan citra pengukur sudut maupun perbaikan dari mekanisme bandul.

#### 7. Pembahasan Rancangan dan Hasil Pengukuran

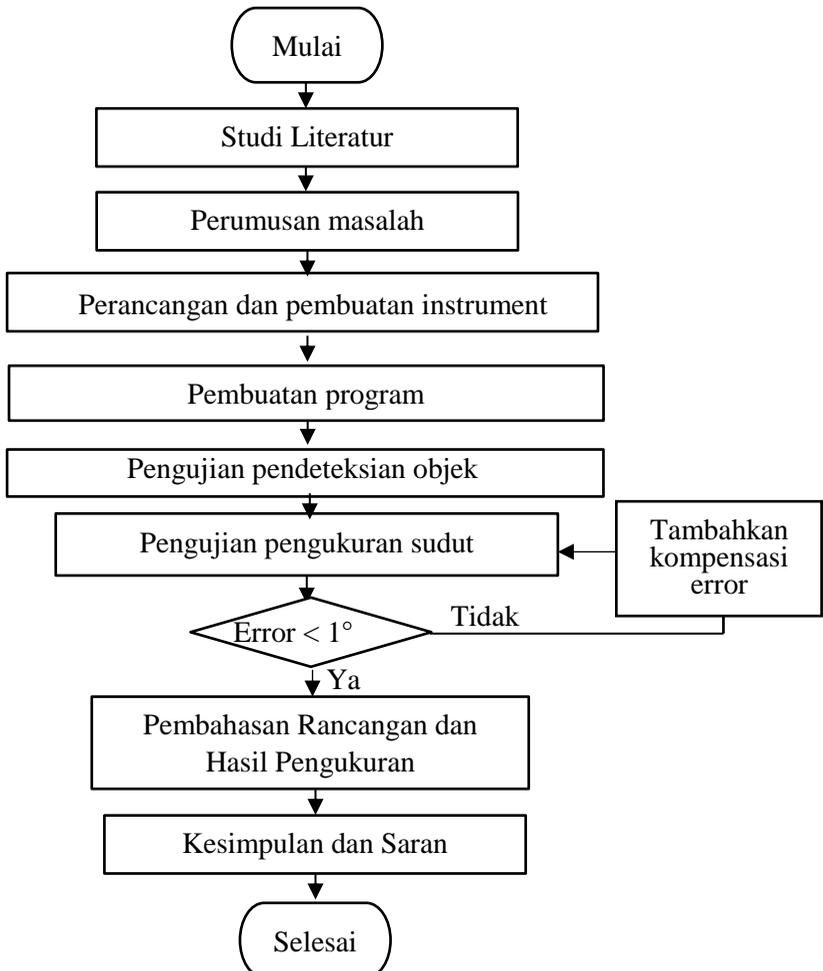
Apabila instrumen dapat menghasilkan pengukuran sudut yang diharapkan maka dilakukan pembahasan dimulai dari perancangan instrument pengukur sudut dan program pengolahan citra hingga pembahasan hasil pengukuran, yaitu seberapa besar *error* yang dihasilkan oleh instrument pengukur sudut dan bagaimana *respon time* mekanisme bandul apabila mengalami osilasi.

#### 8. Kesimpulan dan Saran.

Dari hasil perancangan hingga pembahasan hasil pengukuran instrumen, dibuat suatu kesimpulan yang sesuai dengan tujuan penelitian.

### 3.2 Diagram Alir Tahapan Penelitian

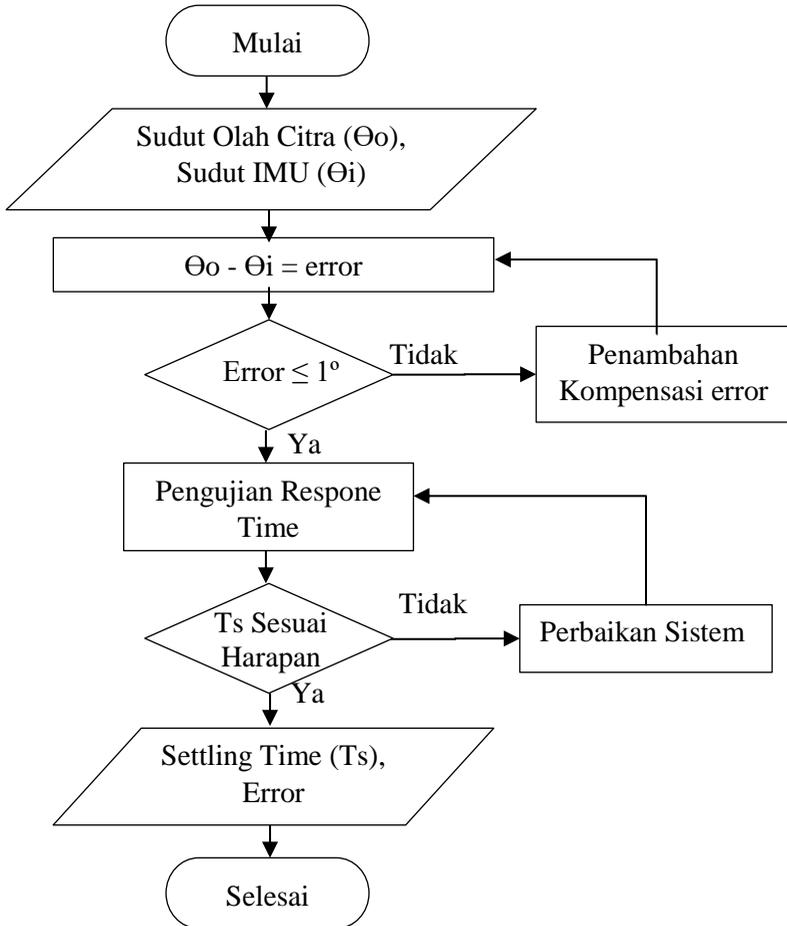
Adapun tahap penelitian yang telah dijelaskan sebelumnya dapat digambarkan dalam diagram alir Gambar 3.1 sebagai berikut,



Gambar 3.1 Diagram alir metodologi penelitian

### 3.3 Diagram Alir Pengujian Pengukuran Sudut

Berikut merupakan tahap pengujian instrument olah citra yang akan dibandingkan dengan pengukuran sudut oleh sensor IMU yang telah dikalibrasi [10]. Berikut adalah tahapan pengujian yang lebih terinci digambarkan dalam diagram alir Gambar 3.2 sebagai berikut.

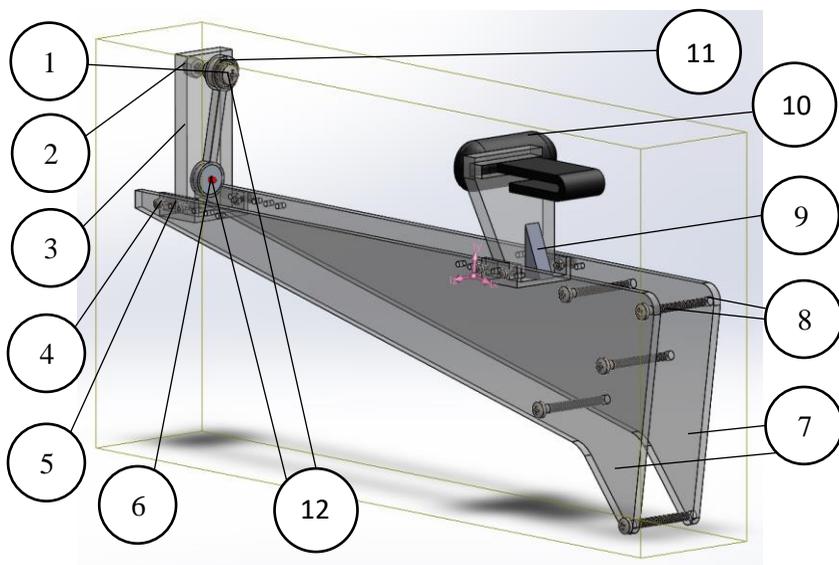


Gambar 3.2 Diagram alir pengujian pengukuran

## BAB IV PERANCANGAN DAN PEMBUATAN INSTRUMEN

### 4.1 Konsep Instrumen

Ide konsep perancangan instrumen berasal dari *rear-fender* kendaraan roda dua yang terhubung langsung dengan rangka kendaraan roda dua. Sistem yang akan dibuat adalah instrumen pengukuran sudut yang digunakan sebagai *feedback* kemiringan sudut pada penelitian *self-balancing bike* menggunakan sistem kontrol sebagai pengatur sistem penyeimbangan sepeda seperti pada Gambar 4.1.



Gambar 4.1 Rancangan instrument

#### Keterangan

- |                      |                            |
|----------------------|----------------------------|
| 1. Baut Poros Bandul | 4. Baut Dudukan Bandul     |
| 2. Mur Poros Bandul  | 5. Mur Dudukan Bandul      |
| 3. Dudukan Bandul    | 6. Batang dan Beban Bandul |

- |                         |                      |
|-------------------------|----------------------|
| 7. Plat Dasar Instrumen | 10. Kamera Instrumen |
| 8. Baut Plat Instrumen  | 11. Bearing          |
| 9. Mur Plat Instrumen   | 12. Objek Warna      |

Rancangan instrumen tersebut merupakan rancangan akhir yang akan digunakan pada penelitian selanjutnya yaitu *self-balancing bike*. Cara kerja alat ini yaitu Batang dan Beban Bandul (6) akan mengalami gerak bebas ketika terjadi perubahan kemiringan sudut pada Plat Dasar Instrumen (7) sehingga bandul akan mempertahankan posisinya sehingga tetap sejajar dengan arah percepatan gravitasi bumi. Posisi sumbu vertikal batang bandul (6) yang membentuk persinggungan dengan sumbu vertikal Plat Dasar Instrumen (7) menghasilkan sudut yang merepresentasikan kemiringan sepeda ketika instrumen telah dipasangkan. Kamera (10) akan mengirimkan perubahan posisi objek warna (12) pada Beban bandul (6) terhadap Baut Poros Bandul (1) secara *real-time*. Perubahan posisi tersebut yang akan digunakan untuk menentukan kemiringan sudut yang diterima oleh instrumen, informasi posisi diolah menjadi sudut oleh program olah citra yang telah dibuat. Setelah kemiringan sudut terbaca oleh program selanjutnya akan dikirim ke bagian kontroler yang akan dikembangkan pada penelitian selanjutnya. Setelah perancangan gambar kerja seluruh komponen mekanisme selesai, dilakukan proses manufaktur komponen instrumen. Komponen yang ada dipasaran dapat dibeli secara langsung, yaitu baut, mur, bearing, web kamera, dan objek warna. Sebagian besar komponen dibuat menggunakan bahan acrylic yang memiliki tebal 3mm dan 5 mm dengan proses *laser cutting* di bengkel acrylic.

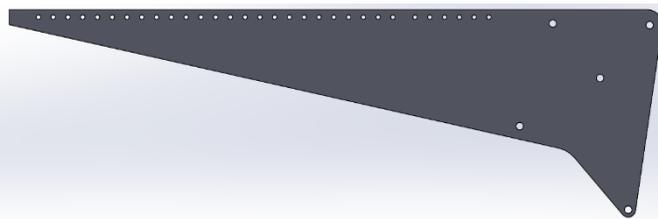
#### 4.2 Perancangan Komponen

Pembuatan instrumen dimulai dengan membuat *list of requirement* untuk menjadi acuan perancangan. Kemudian dilanjutkan dengan tahap perancangan yang dimulai merencanakan dimensi, bentuk dan jenis material komponen. Rancangan mekanisme disesuaikan dengan *list of requirement* pada Tabel 4.1

yang telah ditentukan. Hasil akhir dari perancangan ini adalah gambar detail dan gambar susunan dari mekanisme.

#### 4.2.1 Perancangan Pelat Dasar Instrumen

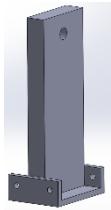
Pelat dasar ini berfungsi untuk menopang komponen lainnya dan akan dipasangkan pada rangka *self-balancing bike*, sehingga sumbu vertikal yang dimiliki instrument pengukur sudut adalah sama dengan sumbu vertikal dari *self-balancing bike*. Pelat dasar ini dibuat dengan bahan acrylic dengan tebal 5mm dan berbentuk memanjang supaya posisi bandul dapat lebih mudah untuk disesuaikan dengan kemampuan tangkap gambar kamera. Ukuran luas pelat dasar instrumen dirancang seminimal mungkin supaya tidak menambahkan gaya berat pada *self-balancing bike*. Pelat dasar memiliki geometri yang sama baik Plat kiri maupun Plat kanan, dapat dilihat pada gambar 4.2.



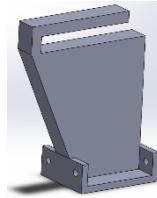
Gambar 4.2 Plat Dasar Instrumen

#### 4.2.2 Perancangan Dudukan Bandul Dan Kamera

Dudukan bandul berfungsi sebagai penopang mekanisme bandul dan memberikan tempat untuk pemasangan poros yang menjadi sumbu putar bandul. Dudukan bandul memiliki plat bagian bawah yang akan langsung dihubungkan dengan plat dasar instrumen. Dudukan bandul dapat dilihat pada gambar 4.3 berikut. Sedangkan dudukan kamera berfungsi sebagai penopang kamera. Dudukan kamera memiliki geometri yang tidak simetris dikarenakan penyesuaian dengan posisi lensa kamera Logitech C-170 yang digunakan tidaklah berada ditengah. Gambar 4.4 merupakan perancangan dudukan kamera.



Gambar 4.3 Dudukan Bandul

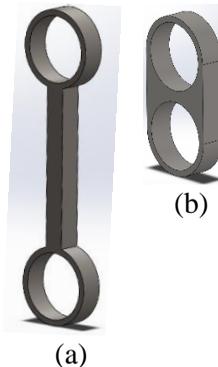


Gambar 4.4 Dudukan Kamera

### 4.2.3 Perancangan Batang Bandul

Batang bandul berfungsi sebagai batang yang menghubungkan beban bandul dengan bearing yang berada pada poros putar bandul. Pemasangannya terletak didepan dudukan batang bandul. Batang ini menggunakan bahan akrilik karena memiliki bahan yang relatif keras, dan mudah dimanufakturing. Gambar 4.5 merupakan rancangan dari batang bandul.

Pada tahap perancangan didapatkan bahwa semakin kecil jari putaran bandul maka akan berdampak semakin sedikitnya siklus osilasi yang dialami oleh bandul. Sehingga dari hasil perhitungan maka desain bandul yang digunakan adalah desain bandul dengan jarak antara pusat beban dengan pusat sumbu putar sebesar 2cm, seperti yang terlihat pada gambar 4.5 (b), begitupun



Gambar 4.5 Rancangan Batang Bandul.

sebaliknya semakin besar jaran beban bandul terhadap sumbu putar maka semakin banyak siklus yang dihasilkan saat berosilasi hingga bandul diam. Adapun dalam merancang bandul perlu memperhatikan pengaruh panjang batang bandul dan berat beban bandul terhadap respon gerak dari bandul tersebut.

#### 4.3 Perhitungan Mekanisme Bandul

Pergerakan mekanisme bandul dapat diketahui dengan menghitung berapa banyak setengah siklus osilasi yang terjadi pada mekanisme gerak bandul dengan redaman terjadi akibat Coulomb atau *dry-friction* damping. Adapun koefisien gesek ( $\mu$ ) dipengaruhi oleh kondisi permukaan material yang terjadi kontak gesekan, dimana akan bernilai 0,1 ketika permukaan gesekan merupakan material logam yang telah diberikan pelumas, 0,3 ketika permukaan gesek merupakan material logam yang tidak mendapatkan pelumas, dan bernilai mendekati 1 ketika plat yang bergesekan merupakan material karet dengan logam [8]. Beberapa perhitungan dilakukan dengan mengubah variable panjang dari batang pendulum yang mempengaruhi momen inersia batang bandul, beban dan berat batang bandul. Dengan perhitungan momen inersia berdasarkan persamaan (2.9) menggunakan massa jenis akrilik sebesar  $1.18\text{g/cm}^3$ , luas penampang batang bandul sebesar  $0,25\text{cm}^2$ , kemudian perhitungan momen inersia dilakukan dengan perubahan panjang batang dari 1cm hingga 5cm dengan kenaikan panjang sebesar 0,5cm. Didapatkan momen inersia sebagai berikut dengan sampel panjang batang 5cm, dan jari-jari dari bearing pada sumbu putar adalah 1cm:

$$\begin{aligned}
 J_{\text{batang}} &= \int_m \mathbf{r}^2 \, d\mathbf{m} = \int_V \mathbf{r}^2 \cdot \rho \cdot dV \\
 &= \int_r \mathbf{r}^2 \cdot \rho \cdot A \cdot dr \\
 &= \int_r \mathbf{r}^2 \cdot \rho \cdot A \cdot dr \\
 &= \rho \cdot A \int_r \mathbf{r}^2 \cdot dr \\
 &= \frac{1}{3} \cdot \frac{1,18\text{g}}{\text{cm}^3} \cdot 0,25\text{cm}^2 \cdot [r^3]_1^5
 \end{aligned}$$

$$J_{\text{batang}} = 12.19333 \text{ g} \cdot \text{cm}^2$$

Sedangkan momen inersia beban pendulum didapatkan dengan mengasumsikan beban pendulum sebagai partikel dengan massa tetap sebesar 12g. Perhitungan momen inersi pada beban bandul pada sampel panjang batang bandul 5cm didapatkan berdasarkan persamaan (2.9) adalah sebagai berikut:

$$\begin{aligned} J_{\text{beban}} &= \int_m \mathbf{r}^2 \, d\mathbf{m} \\ &= m_{\text{beban}} \cdot \mathbf{r}^2 \end{aligned}$$

$$J_{\text{beban}} = 12\text{g} \cdot (5\text{cm})^2 = 48\text{g} \cdot \text{cm}^2$$

Setelah kedua momen inersia batang dan beban pendulum didapatkan maka didapatkan  $J_o = J_{\text{beban}} + J_{\text{batang}}$ , kemudian dimasukkan kedalam persamaan (2.7) dengan koefisien gesek yang digunakan adalah 0,1 karena bearing yang bergesekan dengan poros telah diberi pelumas. Persamaan tersebut dijabarkan dengan sampel panjang batang 5cm sebagai berikut:

$$J_o \ddot{\Theta} + W \cdot d \sin\Theta - \mu \cdot m \cdot g \cdot r \cdot \cos\Theta = 0$$

$$J_o = J_{\text{beban}} + J_{\text{batang}} = 60,19 \text{ g} \cdot \text{cm}^2$$

$$W \cdot d = \left[ \frac{(W_{\text{batang}} \cdot d)}{2} + \{W_{\text{beban}} \cdot (d + r_{\text{beban}})\} \right]$$

$$W \cdot d = [A + B].$$

dimana

$$A = \frac{(1446,97 \text{ g} \cdot \frac{\text{cm}}{\text{s}^2} \cdot 5\text{cm})}{2} = 3617,34 \text{ g} \cdot \frac{\text{cm}^2}{\text{s}^2}$$

$$B = \{11772 \text{ g} \cdot \text{cm}/\text{s}^2 \cdot (5\text{cm} + 1\text{cm})\} = 70632 \text{ g} \cdot \frac{\text{cm}^2}{\text{s}^2}$$

$$W \cdot d = 74.249,44 \text{ g} \cdot \frac{\text{cm}^2}{\text{s}^2}$$

Sehingga didapatkan frekuensi natural dari sistem bandul adalah

$$\omega_n = \left( \frac{W \cdot d}{J_o} \right)^{1/2}$$

$$\omega_n = 12,92 \text{ /s}$$

Perhitungan tersebut dilakukan beberapa kali untuk mengetahui pengaruh panjang batang pendulum terhadap siklus osilasi yang terjadi pada bandul seperti yang ditampilkan pada lampiran A. Berdasarkan hasil perhitungan siklus osilasi dari persamaan (2.8) maka didapatkan dengan panjang bandul 5cm akan menghasilkan siklus sebanyak:

$$R \geq \left( \frac{\Theta_0 - \left( \frac{T}{W \cdot d} \right)}{\left( 2 \cdot \frac{T}{W} \right)} \right)$$

dimana  $T = \mu \cdot W_{total} \cdot r_o = 0,1 \cdot 13218,98 \text{ g} \cdot \text{cm/s}^2 \cdot 0,5\text{cm}$

$$T = 660,94 \text{ g} \cdot \text{cm}^2/\text{s}^2$$

$$\Theta_o = \text{sudut awal (rad)} = (30^\circ \times \pi) / 180^\circ = 0,5236 \text{ rad}$$

Sehingga didapatkan jumlah setengah siklus yang berlalu sebelum akhirnya berhenti adalah sebanyak  $R \geq 29 \text{ half-cycles}$ . Berdasarkan hasil perhitungan dengan variasi panjang bandul didapatkan setengah siklus paling sedikit terjadi pada panjang jari-jari putar batang bandul adalah 1cm dengan didapatkan jumlah setengah siklus yang didapatkan sebanyak  $R \geq 9,78 \text{ half-cycles}$ . Adapun hasil tabel perhitungan dapat dilihat pada lampiran A. Panjang batang bandul dapat dibandingkan dengan jumlah setengah siklus yang dihasilkan pada set point sudut awal kemiringan  $\Theta_o = 30^\circ$ . Berdasarkan hasil perhitungan pada lampiran A dapat dilihat hubungan antara panjang bandul dengan jumlah siklus, hal tersebut yang menjadi dasar bahwa panjang bandul dari rancangan batang bandul 5cm yang ditunjukkan pada gambar 4.5 (a) harus dirubah menjadi batang bandul jari-jari putaran bandul sebesar 1cm seperti

yang ditampilkan pada gambar 4.5 (b). Sedangkan untuk perhitungan periode bandul didapatkan dengan menggunakan rumus (2.6) periode bandul kaku, sebagai berikut:

$$T = 2\pi \left( \frac{I}{m \cdot g R} \right)^{1/2}$$

$$T = 2\pi \left( \frac{12}{(\text{mbatang} + \text{mbeban}) \times 981 \times 1.98} \right)^{1/2}$$

$$T = 2\pi \left( \frac{12}{(12,295) \times 981 \times 1.98} \right)^{1/2}$$

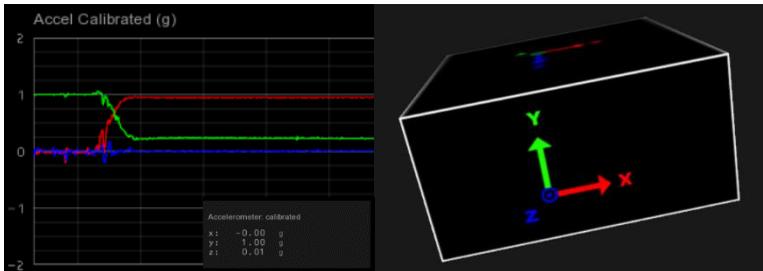
$$T = 0.140560462 \text{ s}$$

#### 4.4 Kalibrasi dengan Inertial Measurement Unit

Dalam merancang dan membangun sebuah instrument pengukur dibutuhkan alat ukur lain yang dijadikan acuan, hal ini bertujuan untuk memenuhi sifat dasar alat ukur yaitu rantai kalibrasi dengan membandingkan alat ukur dengan alat ukur standar. Pada penelitian rancang bangun ini untuk mengetahui keakuratan alat ukur kemiringan yang dibuat dengan metode *Image Processing*, maka perlu dibandingkan dengan sensor kemiringan

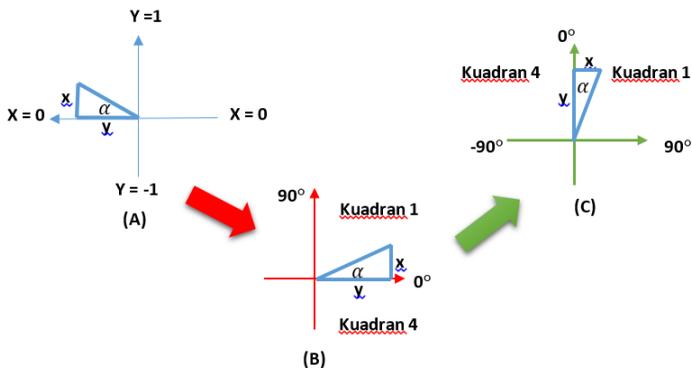


Gambar 4.6 3-DOF Inertial Measurement Unit



Gambar 4.7 Sinyal Accelerometer

yaitu *Inertial Measurement Unit (IMU)*. Sensor kemiringan yang digunakan sebagai kalibrator merupakan produk dari Motion Node seperti yang terlihat pada gambar 4.6 yang merupakan teknologi *inertial motion tracking*, yang mana dapat memberikan hasil pengukuran yang akurat secara *real-time*, dan sudah melalui proses kalibrasi. Sensor ini memiliki *accelerometer* yang mengirimkan sinyal, dimana sinyal tersebut ditentukan dalam satuan  $g$  ( $1g = 9.80665$  meter/second<sup>2</sup>). Dimana dari masing-masing sumbu sensor x, y, z akan bernilai  $1g$  ketika melawan arah gravitasi bumi, dan bernilai  $-1g$  ketika searah dengan gravitasi bumi seperti yang ditunjukkan oleh tampilan perangkat lunak Motion Service pada



Gambar 4.8 Diagram konversi sinyal menjadi sudut. (A) diagram sinyal *IMU*, (B) Quadrant diagram, (C) sumbu sepeda

gambar 4.7. Perbandingan itulah yang akan dikonversi menjadi sudut. Dalam mengkonversi dari nilai sinyal sumbu X dari sensor menjadi nilai pada sumbu y pada diagram sudut, sedangkan sumbu sinyal sumbu Y dari sensor akan menjadi nilai pada sumbu x pada diagram sudut, seperti yang ditunjukkan pada gambar 4.8. Setelah mendapatkan besarnya nilai y dan x maka, perubahan sudut yang dialami oleh *IMU* adalah sebesar:  $\alpha = \arctan \frac{x}{y}$

#### 4.5 Implementasi Program dalam Sistem

Program pendeteksian dan penentuan sudut bandul dibuat dengan menggunakan teknologi *computer vision*. Perangkat lunak yang digunakan dalam pembuatan program adalah Visual Studio Community 2013 dan OpenCV 2.1. Mesin pengolah yang digunakan untuk menjalankan program adalah *personal computer* dengan spesifikasi sebagai berikut:

- a. ASUS G550Jk
- b. *Processor* Intel® Core™ i7-4710HQ CPU @ 2.5GHz
- c. *Memory* 8192 MB RAM
- d. *Operating System* Windows 10 Home (64-bit)

Untuk mengetahui performa dari program pendeteksian dan penentuan koordinat 2 dimensi objek yang dibangun maka perlu dilakukan pengujian. Pengujian dilakukan pada kondisi tertentu. Objek yang digunakan adalah objek berwarna berdiameter 5 mm. Pengambilan data dilakukan didalam ruangan dengan pencahayaan konstan.

#### 4.6 Konstruksi Program *Image Processing*

Dalam mengkonstruksi program pengolahan citra perlu ditetapkan komponen perangkat lunak apa sajakah yang digunakan sebagai landasan pembuatan program, yang mana disesuaikan dengan metode pengukuran sudut terhadap mekanisme bandul. Adapun perangkat lunak yang digunakan pada pembuatan program ini adalah *Visual Studio* 2013 dengan menggunakan bahasa C++, dengan menggunakan *library* OpenCV 2.4.10 sebagai pustaka pengambilan fungsi perintah pengolahan citra. Dalam merancang

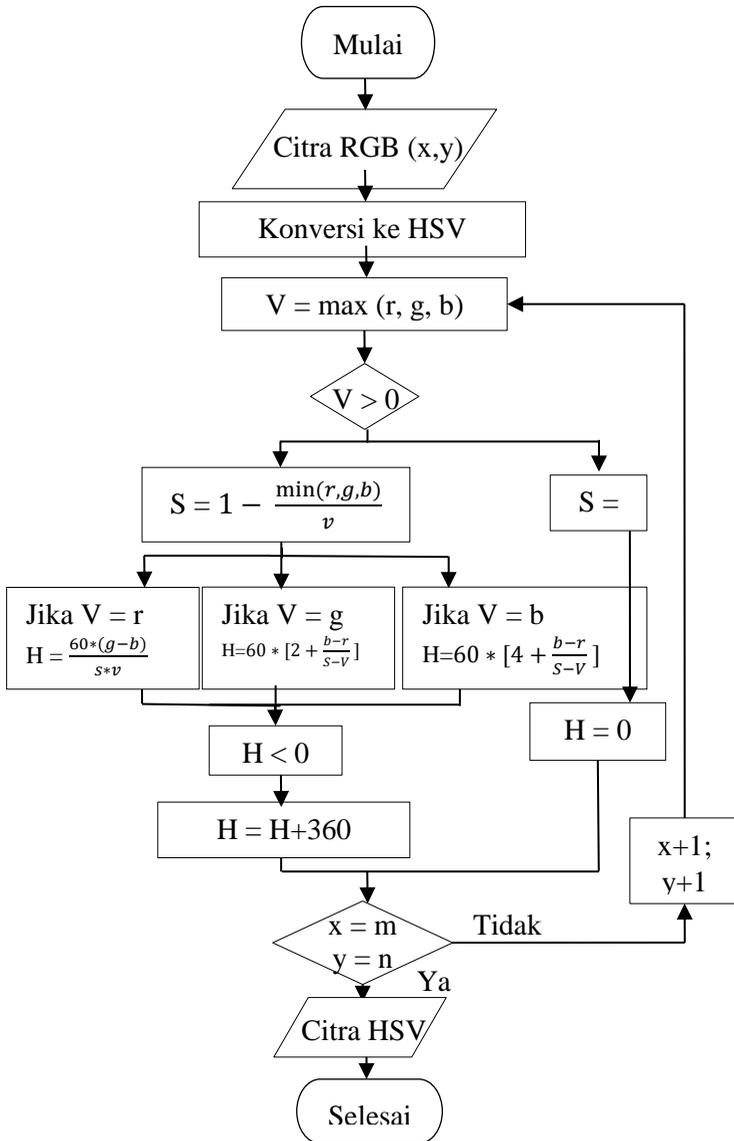
suatu program perlu diperhatikan bahwa banyaknya proses yang dilakukan oleh program akan mempengaruhi lamanya proses pengerjaan, seperti dalam menggunakan OpenCV sebagai *pre-processor* sebelum akhirnya diproses kedalam *editor code* yang akan di-*compile* menjadi *executable* program.

#### 4.6.1 Program Pendeteksian Objek

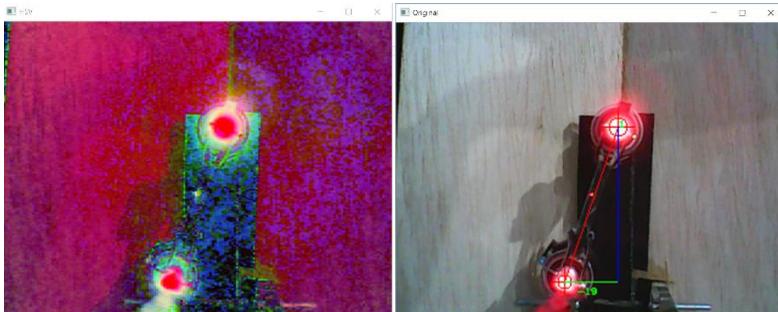
##### Kode program 1: *RGB to HSV*

```
Mat imgOriginal;  
cap.read(imgOriginal);  
Mat imgHSV;  
cvtColor(imgOriginal, imgHSV, COLOR_BGR2HSV);
```

Tahap pertama yang dilakukan adalah menginput gambar secara *real-time* dari kamera kedalam memori pengolahan data dalam bentuk matriks pixel sebesar 320 lebar x 240 tinggi pixel, saat proses ini gambar yang ditangkap oleh kamera secara otomatis merupakan gambar format RGB. Setelah gambar berhasil di ambil, selanjutnya gambar di konversi kedalam format HSV. Hal ini dikarenakan pendeteksian gambar yang dilakukan dalam format HSV lebih baik dalam menentukan objek jika dibandingkan dengan pendeteksian pada format RGB. *Frame* hasil dari tangkapan kamera Logitech C-170 ber resolusi sebesar 320 x 240 *pixel*. Hasil dari *frame* tersebut akan diproses menjadi citra HSV. Gambar 4.9 adalah diagram alir dari proses pengolahan citra RGB ke HSV, sedangkan pada Gambar 4.10 dapat dilihat hasil perubahan citra RGB menjadi citra HSV.



Gambar 4.9 Diagram alir pengolahan RGB ke HSV



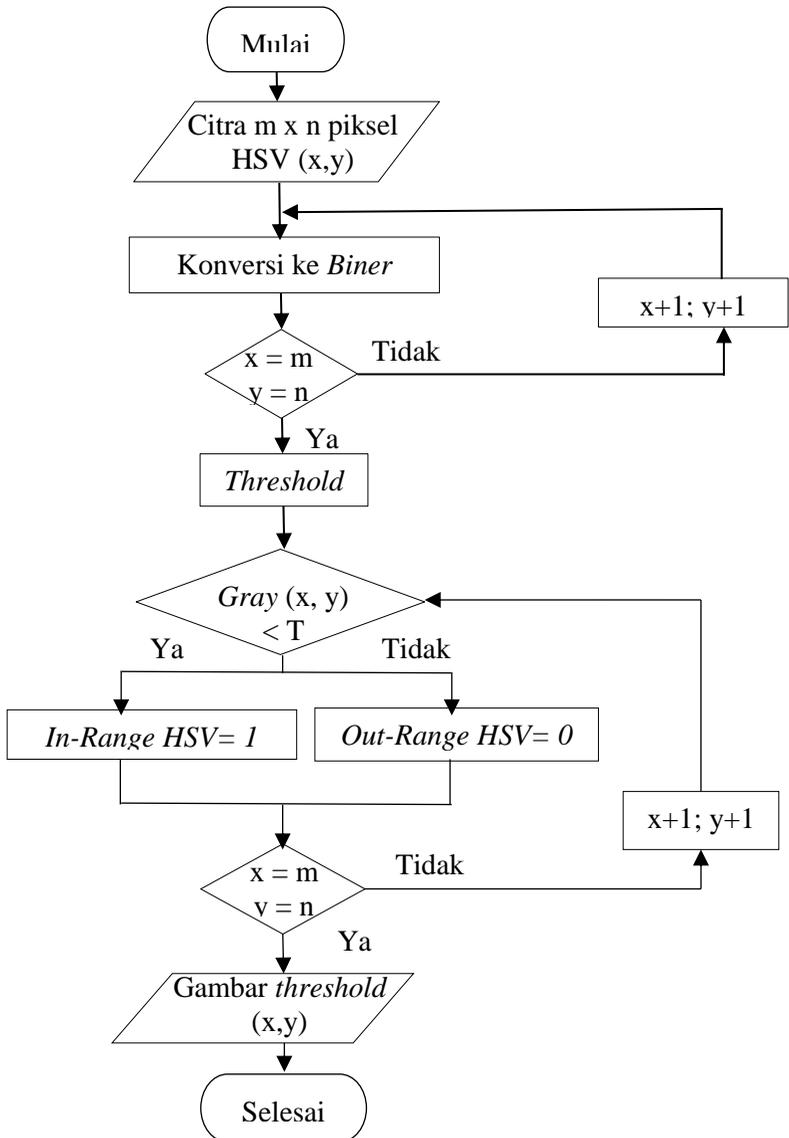
Gambar 4.10 Konversi RGB menjadi HSV

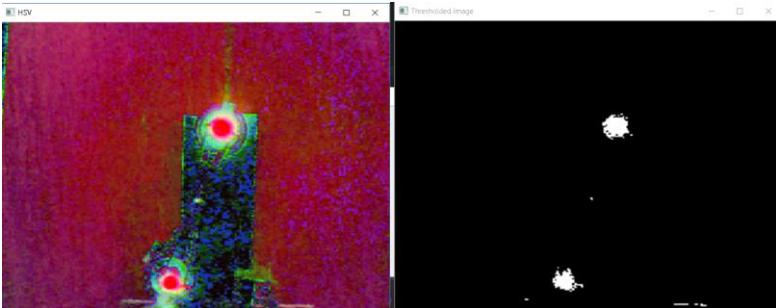
#### 4.6.2 Program Thresholding

##### Kode Program 2: *Threshold*

```
//Threshold the image
Mat imgThresholded;
inRange(imgHSV, Scalar(iLowH, iLowS, iLowV),
Scalar(iHighH, iHighS, iHighV), imgThresholded);
```

Proses kedua dari tahap pendeteksian objek adalah *thresholding* pada gambar HSV. Tipe *threshold* yang digunakan adalah *binary threshold* dengan *range* nilai HSV minimum (0, 0, 0) - maksimum (255, 255, 255) baik untuk HSV *low* dan HSV *high*. Pada gambar 4.11 dapat dilihat diagram alir dari proses *threshold* pada gambar HSV. Hasil perubahan gambar HSV menjadi gambar biner dengan tipe *binary threshold* dapat dilihat pada Gambar 4.12. Pada gambar 4.12 dapat dilihat bahwa warna putih menunjukkan objek yang ingin di deteksi dan warna hitam merupakan background dari objek. Namun terkadang hasil yang didapatkan kurang sempurna. Pada gambar 4.12 dapat dilihat bahwa bentuk hasil *threshold* dari gambar objek warna kurang sempurna.

Gambar 4.11 Diagram alir proses *threshold*



Gambar 4.12 Hasil *Threshold* dari HSV

### 4.6.3 Program Erode dan Dilate

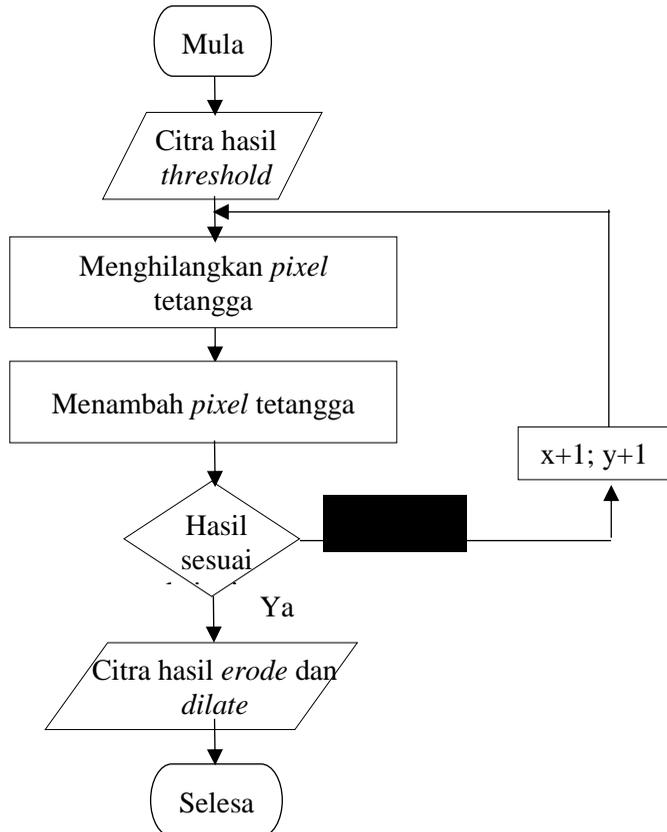
#### Kode Program 3: *Erode and Dilate*

```
//morphological opening (remove small objects from
the foreground)
erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
dilate(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));

//morphological closing (fill small holes in the
foreground)
dilate(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
```

Pada proses *threshold* ini diperlukan juga proses *noise filtering*. Diagram alir dari proses *noise filtering* dapat dilihat pada Gambar 4.13. Proses ini dilakukan karena pada proses *threshold* biasanya terdapat gangguan/*noise* pada hasil gambar yang diperoleh. *Noise filtering* bertujuan untuk membuang atau mengurangi gangguan tersebut agar gambar *threshold* yang diperoleh sesuai dengan yang diinginkan. Proses *noise filtering* terdiri dari *erode* dan *dilate*, dimana *erode* bertujuan untuk

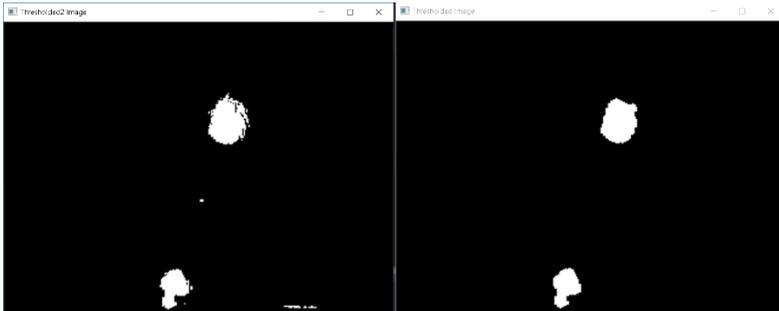
mengurangi *pixel* tetangga sedangkan *dilate* bertujuan untuk menambah *pixel* tetangga.



Gambar 4.13 Diagram alir proses *erode* dan *dilate*

Pada Gambar 4.14 dapat dilihat perubahan pada gambar *threshold* yang dilakukan proses *erode*. *Erode* dilakukan dengan pengecilan 5 *pixel* tetangga. Citra *threshold* yang sudah di-*erode* kemudian dilakukan *dilate*

dengan tiap 5 *pixel* tetangga dari *pixel* terluar. Hasil dari *dilate* tersebut dapat dilihat seperti pada Gambar 4.14.



Gambar 4.14 Hasil proses *erode-dilate*

#### 4.6.4 Program Pendeteksian Objek

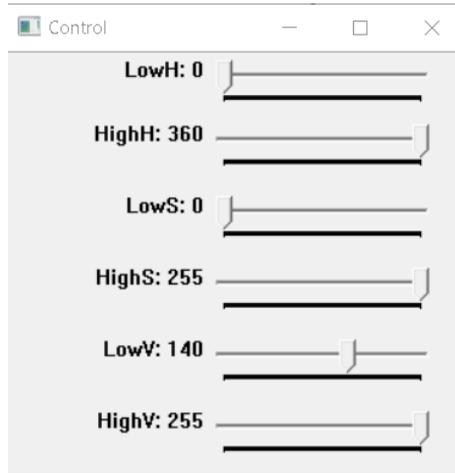
##### Kode Program 3: Kalibrasi untuk Mencari Warna

```
//morphological opening (remove small objects from
the foreground)
erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
dilate(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));

//morphological closing (fill small holes in the
foreground)
dilate(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
```

Proses berikutnya adalah mencari nilai *range* HSV untuk warna objek yang akan dideteksi pada gambar yang telah berformat HSV. Nilai *range* HSV objek warna pada bandul antara lain Hue 0-360, Saturatuion 111-255, dan value 140-255. Nilai *range* HSV yang didapatkan akan

digunakan untuk pendeteksian objek. Penentuan *range* HSV dilakukan beberapa kali percobaan untuk mendapat



Gambar 4.15 *Trackbar* range warna HSV

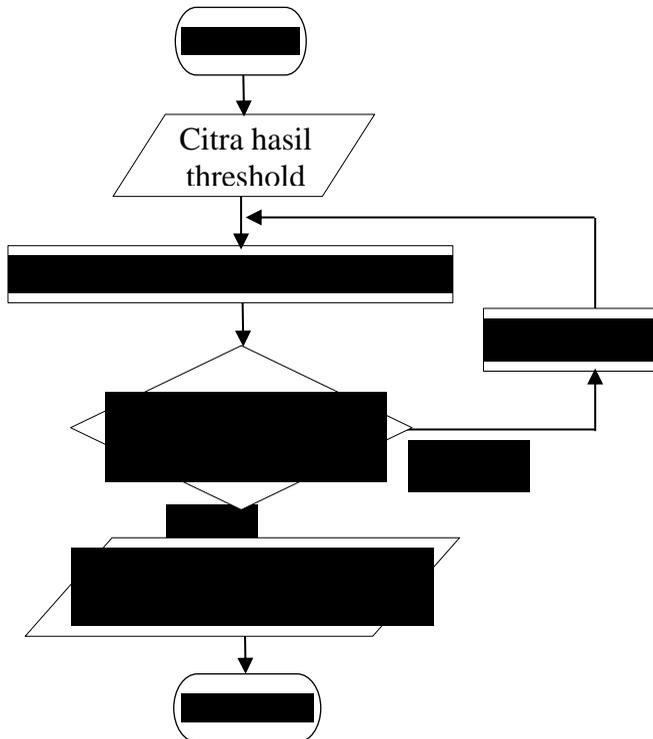
kan hasil pendeteksian objek yang diinginkan. Kondisi lingkungan seperti intensitas cahaya saat dilakukan proses pendeteksian juga sangat berpengaruh, ketidak stabilan pada pendeteksian objek.

#### 4.6.5 Pencarian Variabel Pendeteksian Objek

##### Kode Program 5: Mendeteksi Objek

```
double refArea = 0;
if (hierarchy.size() > 0) {
int numObjects = hierarchy.size();
for (int index = 0; index >= 0; index = hierarchy[index][0]) {
Moments moment = moments((cv::Mat)contours[index]);
double area = moment.m00;
if (area > MIN_OBJECT_AREA){
double x = moment.m10 / area;
double y = moment.m01 / area;
double r = sqrt(area / 3.14);
```

Pendeteksian objek yang ditentukan berdasarkan warna objeknya dapat dilakukan dengan memanfaatkan bentuk gambar biner yang sebelumnya telah dilakukan proses *threshold*. Kemudian dengan menggunakan program, koordinat titik pusat objek dapat diperoleh dari gambar hasil *threshold*. Setelah koordinat x dan y pada pusat objek LED terdeteksi, kemudian dilakukan pencarian kordinat pusat dari LED kedua. Proses dari pencarian koordinat titik pusat objek dapat dilihat pada diagram alir Gambar 4.16.



Gambar 4.16 Diagram alir pencarian koordinat objek

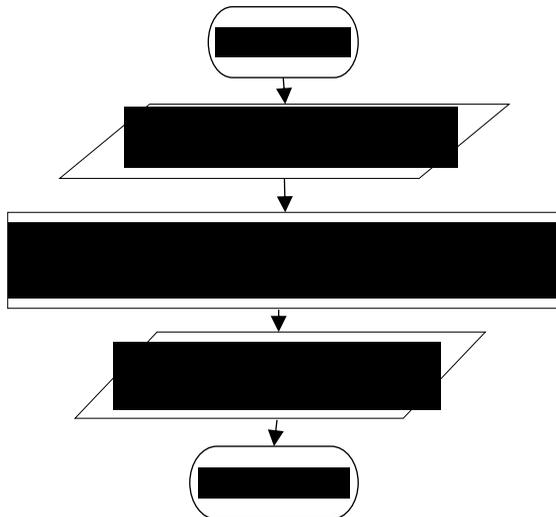
#### 4.6.6 Penunjukan Objek Terdeteksi

##### Kode program 6 : Menunjukkan objek yang terdeteksi

```
//gambar lingkaran luar mengikuti objek
circle(imgOriginal, Point(x, y), r, Scalar(0, 0,
255), 1.5, 8);

//gambar cross line di tengah lingkaran
line(imgOriginal, Point(x, y - r - 5), Point(x, y
+ r + 5), Scalar(0, 0, 255), 1.5, 8);
```

Setelah titik koordinat pusat objek diperoleh, dibuat program yang dapat membuat sebuah lingkaran di sekeliling objek. Lingkaran ini bertujuan untuk menunjukkan posisi dari objek yang dideteksi. Lingkaran ini juga akan deprogram untuk muncul di dalam *frame* gambar hasil tangkapan kamera yang masih memiliki format RGB. Ketika program dijalankan, lingkaran ini juga akan otomatis menandai dan mengikuti objek yang



Gambar 4.17 Diagram alir pembuatan lingkaran

dideteksi ketika objek tertangkap oleh kamera. Langkah-langkah pembuatan lingkaran ini dapat dilihat melalui diagram alir pada Gambar 4.17.

#### 4.6.7 Program Penentuan Sisi Depan, Sisi Sdan Sudut

##### Kode program 7: Menentukan Sisi dan Sudut

```
//gambar lingkaran luar mengikuti objek
circle(imgOriginal, Point(x, y), r, Scalar(0, 0,
255), 1.5, 8);

//gambar cross line di tengah lingkaran
line(imgOriginal, Point(x, y - r - 5),
Point(x, y + r + 5), Scalar(0, 0, 255), 1.5, 8);

line(imgOriginal, Point(x - r - 5, y),
Point(x + r + 5, y), Scalar(0, 0, 255), 1.5, 8);
```

Pada tahap penentuan sudut, langkah pertama yang dilakukan adalah mencari nilai kordinat dua objek warna yang akan merepresentasikan perubahan posisi bandul. Objek yang dideteksi dibatasi berjumlah dua untuk meminimalisir kesalahan penangkapan objek. Setelah kordinat dua objek didapatkan selanjutnya menentukan bahwa segitiga kesebangunan yang akan digunakan untuk menghitung sudut kemiringan bandul. Dan selalu diingat bahwa besar sudut yang didapatkan dalam program C++ adalah dalam satuan radian, sehingga perlu dikonversi kedalam satuan derajat. Seperti yang terlihat dalam diagram alir Gambar 4.19, setelah program mendapatkan sisi depan dari kordinat 2D gambar hasil tangkapan kamera, maka perlu ditentukan sisi depan dan sisi samping yang akan digunakan untuk mengukur sudut segitiga yang akan merepresentasikan kemiringan bandul. Setelah

mendapatkan nilai sisi depan yang didapatkan dari selisih kordinat  $x[1]$  dan  $x[0]$  seperti berikut :

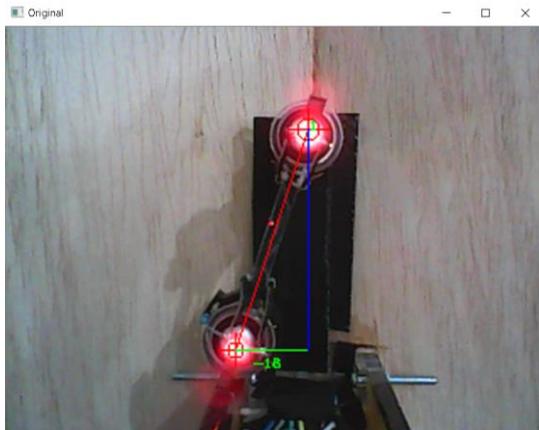
$$depan = x1 - x0$$

Kemudian pencarian sisi samping segitiga didapatkan dengan menghitung selisih kordinat  $y[1]$  dan  $y[0]$  seperti berikut :

$$samping = y1 - y0$$

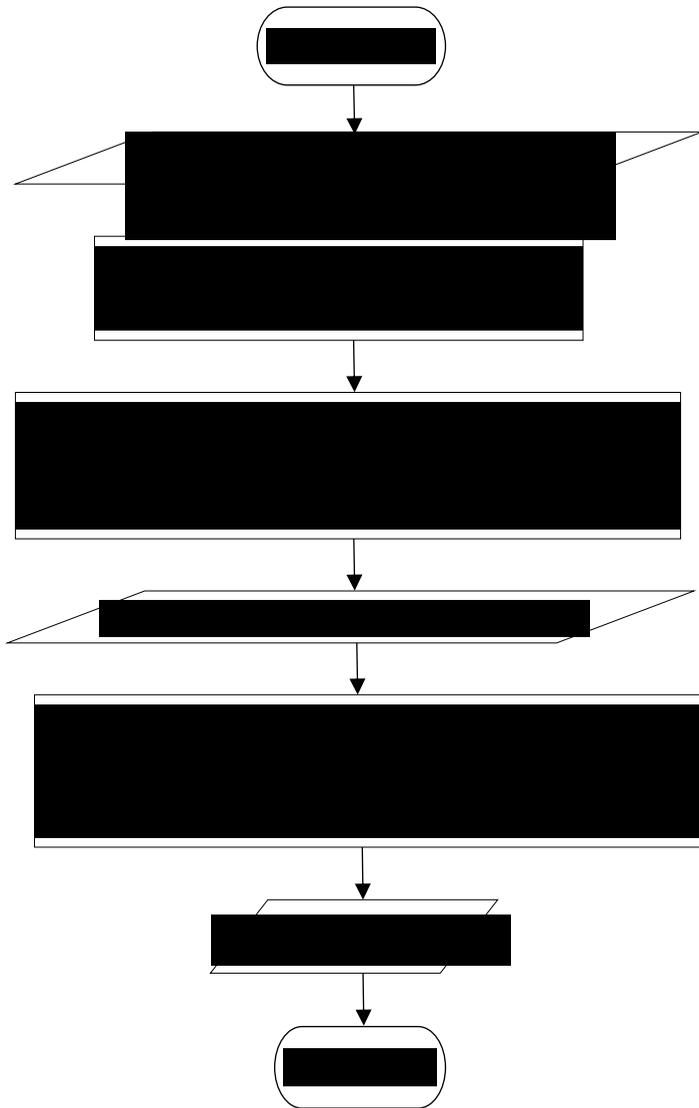
Setelah mendapatkan sisi depan dan sisi samping segitiga, kemudian sudut dihitung dengan persamaan berikut:

$$\alpha = \arctan \frac{depan}{samping}$$



Gambar 4.18 Hasil Penentuan sisi segitiga dan sudut

Adapun hasil dari perhitungan sudut metode olah citra dapat dilihat pada hasil gambar 4.18, dimana sisi yang berwarna biru merupakan sisi samping segitiga yang tegak lurus dengan sepda, dan sisi berwarna hijau merupakan sisi depan.



Gambar 4.19 Diagram alir penentuan sisi dan sudut.

#### 4.7 Perakitan Sistem Kalibrasi Instrumen

Setelah perancangan gambar kerja seluruh komponen mekanisme selesai, dilakukan proses manufaktur komponen mekanisme. Komponen yang ada dipasaran langsung dibeli, yaitu mur, baut, kamera Logitech C-170, Motion Node IMU, bearing, Lampu LED dan Arduino uno untuk proses data akuisisi secara *real-time*. Sebagian besar komponen dibuat menggunakan bahan acrylic yang memiliki tebal 3 mm dan 5 mm dengan proses *laser cutting* di bengkel acrylic. Instrumen pengukur sudut dirakit sesuai desain rancangan instrumen, selebihnya alat-alat penunjang kalibrasi akan ditambahkan hanya untuk proses kalibrasi. Adapun penempatan sensor IMU ditempatkan pada bagian atas plat dasar instrumen dengan posisi datar agar dapat merepresentasikan sumbu vertikal instrument yang akan digunakan sebagai pengukur kemiringan sudut, dengan rangkaian tambahan seperti yang terlihat pada gambar 4.20



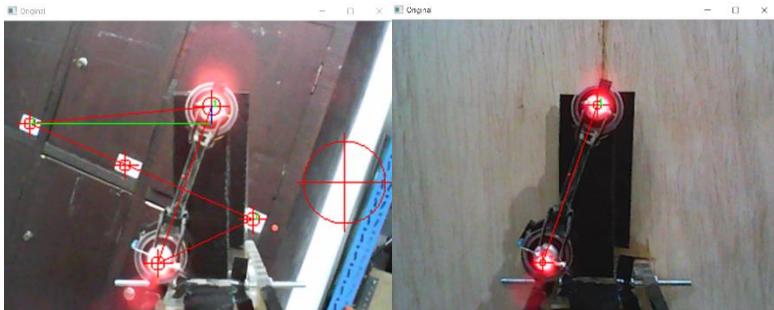
Gambar 4.20 Foto Rangkaian Instrumen.

## BAB V

### PENGUJIAN INSTRUMEN

#### 5.1 Pendeteksian Objek

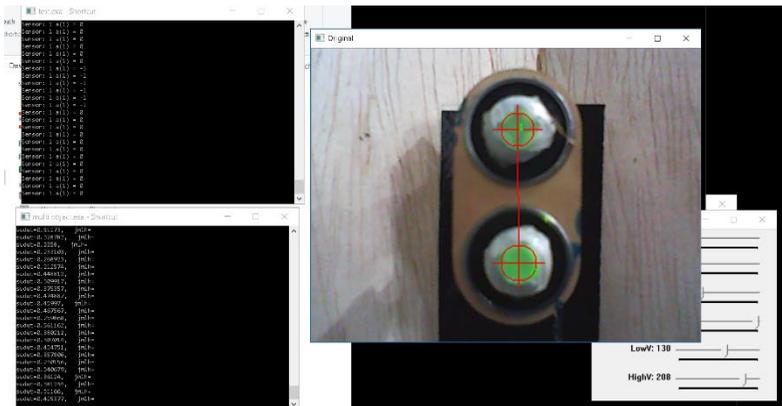
Dalam mendeteksi objek warna ganda yang di tempatkan pada bandul perlu dilakukan kalibrasi sehingga objek yang ingin dideteksi berbeda dengan objek lain yang tertangkap kamera. Pendeteksian objek yang berwarna hijau cenderung memiliki warna cerah yang identik dengan background sehingga proses pendeteksian tidak bisa mendapatkan tepat pada pendeteksian objek bandul. Seperti terlihat pada gambar 5.1 dimana hasil pendeteksian terganggu oleh background atau objek lain yang juga memiliki nilai *value* yang tinggi. Adapun hal yang dapat dilakukan adalah dengan memberikan *background cover*, berhubung objek yang akan dideteksi tidak berganti-ganti, sehingga pendeteksian objek tidak akan terganggu oleh objek lain yang memiliki nilai *value* yang mendekati nilai *value* objek warna pada bandul. Adapun jarak optimal yang ditentukan dari kamera terhadap bandul adalah sejauh 6,5cm.



Gambar 5.1 Hasil pendeteksian objek LED.  
(A) Tanpa Background. (B) Menggunakan Background.

## 5.2 Pengukuran Sudut

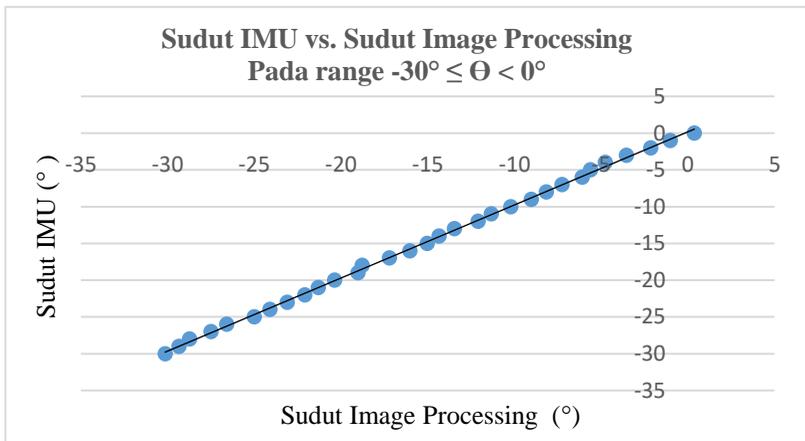
Pengujian program pengukuran sudut instrumen dilakukan dengan meletakkan objek warna pada bandul, dimana jarak tangkap minimum kamera telah diketahui sebelumnya. Pengambilan data dimulai dengan mengacu pada sudut yang didapatkan dari kalibrator *Inertial Measurement Unit*. Selanjutnya pengambilan data sudut yang bernilai negatif merupakan sudut kemiringan ke arah CCW, dan sudut yang bernilai positif menunjukkan arah kemiringan ke arah CW. Pengambilan data dimulai dari sudut  $-30$  hingga sudut  $30$ , dengan interval satu derajat. Adapun contoh pengambilan data dilakukan secara real time seperti yang ditunjukkan pada gambar 5.2. sudut yang telah terdeteksi didapatkan sebagai berikut.



Gambar 5.2 Pembacaan hasil pengukuran.

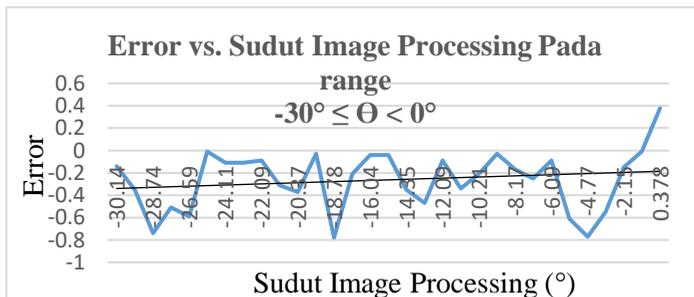
Pada table 1 yang tertera dalam lampiran, dapat dilihat bahwa nilai sudut yang didapat dari metode *Image Processing* berbeda dari sudut yang didapat dari sensor *IMU*. Adapun dari data yang didapat pada *range* pengukuran sudut yang pertama adalah pada rentang  $-30^\circ \leq \Theta < 0^\circ$ . Dari data yang didapatkan membentuk trendline grafik pada Gambar 5.3 berupa garis linier, menunjukkan

bahwa perubahan nilai sudut yang didapat dari *IMU* berbanding lurus dengan perubahan sudut yang dibaca dari metode *Image Processing*. Dari pengambilan data pada *range* pengukuran pertama didapat error paling besar adalah  $-0,77^\circ$  dan yang paling



Gambar 5.3 Pengukuran rentang  $-30^\circ \leq \Theta < 0^\circ$ .

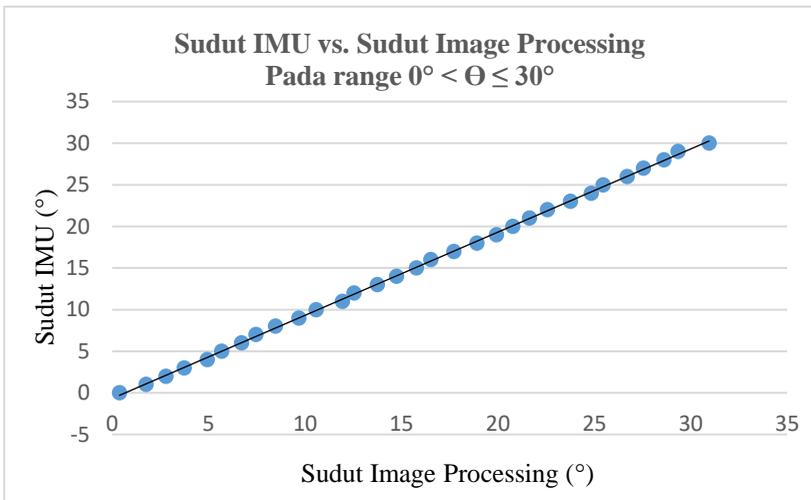
kecil adalah  $-0,01^\circ$ . Nilai Error yang didapat pada pengukuran sudut cenderung fluktuatif seperti yang terlihat pada Gambar 5.4 pada simpangan  $< 1^\circ$  dari besar kemiringan yang didapat dari pengukuran dengan Sensor *IMU*. Adapun kemungkinan error yang



Gambar 5.4 Error vs. Sudut pada  $-30^\circ \leq \Theta < 0^\circ$

terjadi dapat disebabkan oleh perubahan cahaya pada objek *Image Processing* yang dideteksi, yang mengakibatkan objek yang telah di *threshold* berubah bentuk.

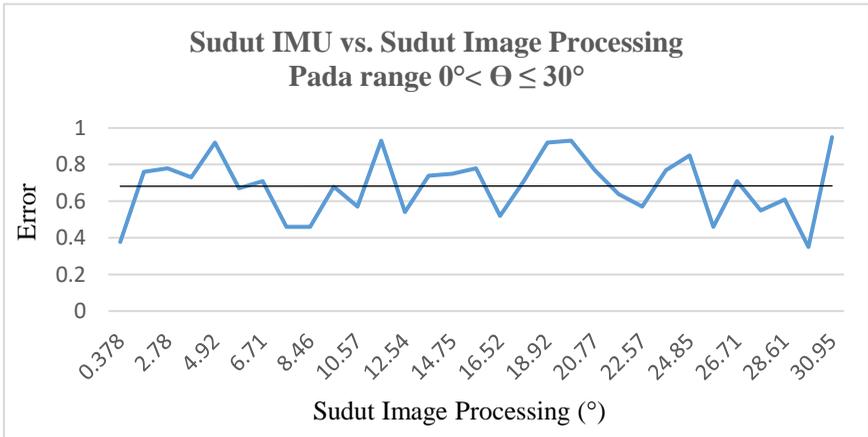
Begitupun pada data yang didapat pada *range* pengukuran sudut yang kedua, dengan rentang pengukuran  $0^\circ < \Theta \leq 30^\circ$ . Dari pengukuran pada *range* tersebut, didapat perbandingan hasil pengukuran antara olah citra dengan pengukuran hasil *IMU* cenderung berbanding lurus, seperti pada grafik Gambar 5.5 dengan trendline *linier*.



Gambar 5.5 Grafik Pengukuran pada  $0^\circ < \Theta \leq 30^\circ$ .

Dari data yang didapatkan membentuk trendline grafik linear, menunjukkan bahwa perubahan nilai sudut pada *range*  $0^\circ < \Theta \leq 30^\circ$  yang didapat dari *IMU* berbanding lurus dengan perubahan sudut yang dibaca dari metode *Image Processing*. Dari pengambilan data pada *range* pengukuran pertama didapat error paling besar adalah  $0,95^\circ$  dan yang paling kecil adalah  $0,378^\circ$ . Nilai Error yang didapat pada pengukuran sudut cenderung fluktuatif seperti yang terlihat pada Gambar 5.6, namun demikian error masih berada pada simpangan yang ditolerir yaitu sebesar

Error  $< 1^\circ$  dari besar kemiringan yang didapat dari pengukuran dengan Sensor IMU.

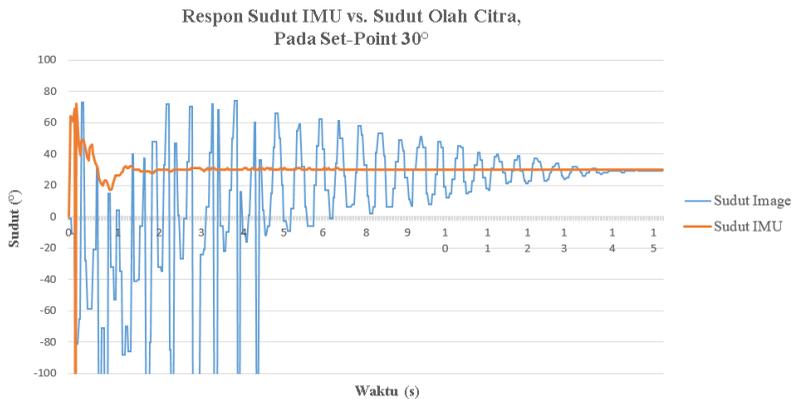


Gambar 5.6 Error vs. Sudut pada  $0^\circ < \Theta \leq 30^\circ$ .

### 5.3 Respon Time Bandul

Dalam membuat instrument pengukuran kita perlu mengetahui performa alat yang kita buat. Informasi ini dapat kita gunakan untuk menentukan apakah alat itu memiliki respon alami seperti yang dibutuhkan. Adapun parameter yang bisa kita gunakan adalah *percent overshoot* dan *settling time*. Pada penelitian kali ini untuk mengetahui transient respon dari instrument pengukur sudut dengan metode *image processing* akan di bandingkan dengan transient respond dari respon sensor *IMU*. Dalam mendapatkan *transient respon* maka digunakanlah perangkat lunak *data acquisition* (DAQ). Perangkat data acquisition digunakan dengan menghubungkan informasi sudut yang didapatkan oleh program pengolahan citra terhadap mikrokontroler. Mikrokontroler akan melanjutkan pengiriman data informasi perubahan sudut kepada perangkat lunak secara *real-time* sehingga besar perubahan sudut dapat dibandingkan langsung dengan perubahan waktu yang terus berjalan.

Set-Point yang digunakan dalam menampilkan transient respon ini diambil kemiringan sudut sebesar  $15^\circ$  dan  $30^\circ$ . Adapun berdasarkan data yang didapatkan menggunakan Software Data Aquisition didapatkan *settling time* untuk instrumen pengukur sudut dengan metode olah citra pada *set-point*  $30^\circ$  ialah 14s seperti yang ditunjukkan oleh grafik pada gambar 5.7, Berbeda dari *settling time* instrumen pengukur sudut yang menunjukkan *respon time* dari sensor *IMU* yang memiliki *settling time* sebesar 1s.

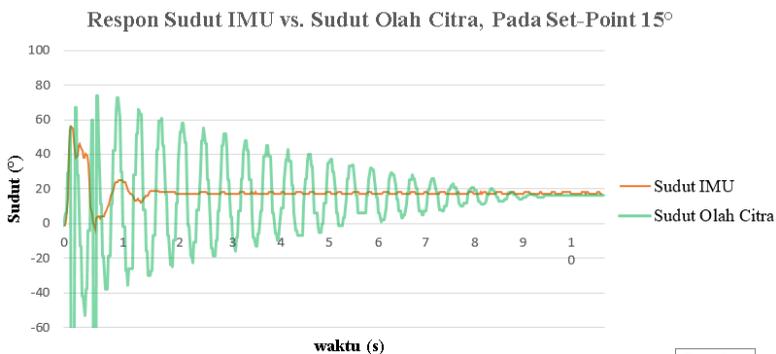


Gambar 5.7 *Respon Time* pada *Set-Point*  $30^\circ$ .

Berdasarkan *transient respon* yang ditunjukkan oleh instrument olah citra memiliki nilai overshoot yang besar, hal tersebut dapat disebabkan oleh gaya eksitasi dari gaya berat sepeda saat mengalami jatuh bebas. Gambar 5.7 juga menunjukkan *Respon Time Sensor IMU*, pada *Set-Point*  $30^\circ$ . Sedangkan pada *set-point*  $30^\circ$ , sensor IMU menunjukkan hasil yang ingin dicapai ketika instrumen pengukur sudut memiliki *settling-time* selama satu detik. Adapun *overshoot* yang masih terjadi pada sensor IMU dapat disebabkan karena sensor IMU yang digunakan adalah accelerometer, dimana sensor tersebut merupakan sensor yang membaca besarnya percepatan yang dialami oleh IMU. Hal itu menunjukkan apabila sensor IMU mengalami gaya eksitasi dari luar,

maka sangat mungkin sensor IMU akan menerima percepatan selain dari percepatan gravitasi. Sensor Motion Node sendiri memiliki besar percepatan yang mampu di tangkap adalah hingga kurang atau lebih 6g atau setara dengan enam kali gravitasi yaitu  $67,5\text{m/s}^2$  [10].

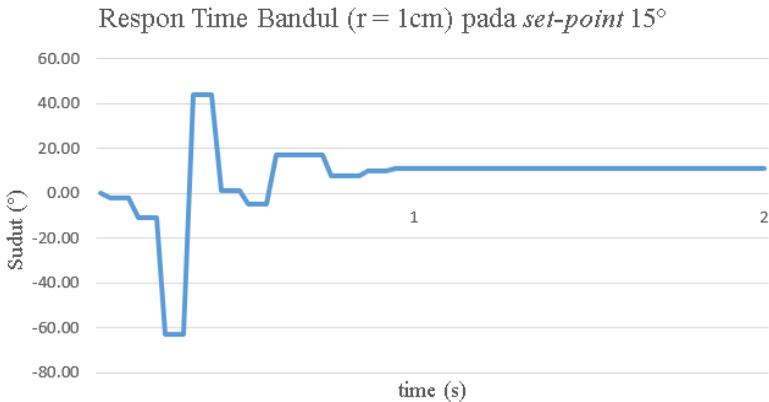
Selanjutnya pengambilan sampling sample time juga dilakukan pada derajat kemiringan yang lebih kecil yaitu pada *set-point*  $15^\circ$  yang ditunjukkan pada gambar 5.8. Berdasarkan transient respon yang dihasilkan oleh instrumen olah citra didapatkan hasil transient respon dengan *settling time* 9s, dan overshoot yang terjadi pada transient respon instrumen olah citra yang lebih sedikit, dimana hal tersebut dipengaruhi oleh *set-point* yang lebih kecil.



Gambar 5.8 *Respon Time* pada Set-Point  $15^\circ$ .

Sedangkan *respon time* dari sensor IMU menunjukkan hasil yang lebih baik seperti yang dapat dilihat pada grafik gambar 5.8 dengan waktu *settling time* yang yaitu satu detik dimana tidak terjadi *overshoot* yang sangat besar seperti yang terjadi pada pengujian transien respon pada *set-point*  $30^\circ$ . Berdasarkan dari hasil kedua *set-point* yang didapatkan menunjukkan bahwa instrumen olah citra perlu diperbaiki dari mekanismenya. Seperti hasil perhitungan yang telah dilakukan sebelumnya pada tahap perancangan, *settling time* yang terlalu lama akibat siklus osilasi yang banyak dapat

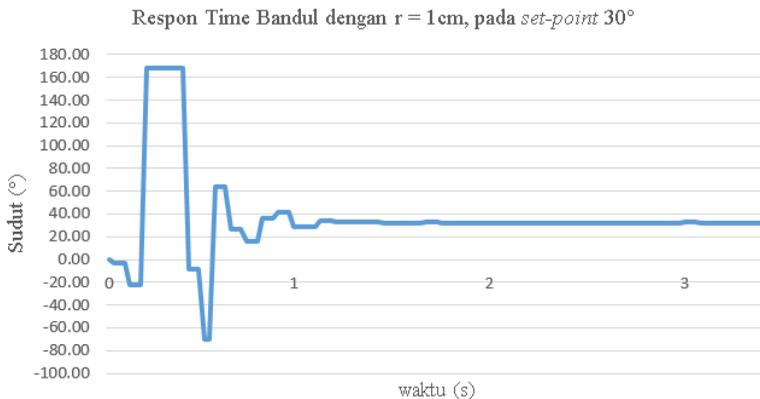
direduksi sesuai dengan hasil data perhitungan pada lampiran A. Berdasarkan hasil analisa pada lampiran A dapat disimpulkan bahwa semakin besar panjang batang bandul sangat berpengaruh pada banyaknya siklus osilasi yang terjadi.



Gambar 5.9 *Respon Time* Olah Citra  $r_{\text{bandul}} = 1\text{cm}$ , *Set-Point*  $15^\circ$ .

Atas dasar hal tersebut maka dilakukan perbaikan pada mekanisme bandul dimana sesuai perhitungan pada tahap perancangan, untuk mendapatkan siklus osilasi yang lebih sedikit maka panjang batang bandul perlu direduksi, sehingga ditentukan panjang bandul yang digunakan untuk perbaikan adalah  $r = 1\text{cm}$ . Massa beban bandul sangat dibutuhkan karna dalam memanfaatkan Coulomb damping, massa digunakan untuk mengurangi simpangan posisi sudut yang pada umumnya sistem tidak akan kembali ke posisi *equilibrium*. Massa beban bandul yang digunakan adalah sebesar 12gram. Dapat dilihat berdasarkan Respon time bandul dengan batang bandul 1cm pada gambar 5.9 dengan *set-point* 15 bahwa walaupun mengalami overshoot yang cukup besar tetapi waktu *settling time* telah tereduksi secara signifikan hingga didapatkan *settling time* sebesar satu detik. Begitu pula yang terjadi pada pengujian bandul *set-point*  $30^\circ$  pada

grafik gambar 5.10, dimana bandul menunjukkan nilai *overshoot* yang sangat tinggi tapi tetap memiliki *settling time* yang diharapkan yaitu berkisar pada waktu satu detik.



Gambar 5.10 *Respon Time* Olah Citra  $r_{\text{bandul}} = 1\text{ cm}$ , Set-Point  $30^\circ$ .

#### 5.4 Evaluasi Desain

Berdasarkan desain yang diperhitungkan dapat dianalisa secara garis besar bahwa semakin kecil batang bandul maka semakin sedikit siklus osilasi yang dihasilkan oleh sistem seperti yang telah diperhitungkan pada lampiran A dengan contoh perhitungan pada subbab 4.3 dan hasil data yang diambil secara *real-time* seperti pada grafik gambar 5.7 dan gambar 5.8. Berdasarkan dari desain sistem instrumen diketahui bahwa untuk mengurangi kesalahan pembacaan objek oleh program olah citra, maka dibutuhkan background untuk menutupi pandangan hasil tangkapan kamera selain objek bandul. Semakin kecil dimensi bandul juga berpengaruh pada volume desain dari instrumen sehingga setelah direduksi menjadi besar bandul satu sentimeter jarak batang bandul terhadap kamera juga dapat direduksi, dan posisi bandul terhadap kamera juga perlu untuk disejajarkan.

*[Halaman Ini Sengaja Dikosongkan]*

## **BAB VI PENUTUP**

### **6.1 Kesimpulan**

Beberapa kesimpulan yang diperoleh pada penelitian tugas akhir ini adalah sebagai berikut:

1. Setelah dilakukan perancangan dan pembuatan perangkat instrumen yang meliputi mekanisme bandul dan konstruksi instrumen selanjutnya membuat program pendeteksian nilai sudut dengan bahasa pemrograman C++. Adapun besar instrumen olah citra juga dipengaruhi oleh panjang bandul yang digunakan. Berdasarkan hasil rancangan dan pembuatan, telah didapatkan cara perhitungan sudut memanfaatkan metode olah citra, yang cara pengukuran sudut dengan menggunakan metode olah citra tersebut akan digunakan sebagai pengukuran sudut pada penelitian rancang bangun *self-balancing bike*.
2. Berdasarkan hasil pengukuran, besar kesalahan yang didapatkan oleh instrumen olah citra dibandingkan dengan kalibrator IMU didapatkan nilai error berkisar antara  $0,01^\circ$  hingga  $0,95^\circ$ , dengan kata lain nilai error yang didapatkan masih berada dibawah batas toleransi. Adapun lamanya waktu yang dibutuhkan agar sistem mencapai *settling time* ( $T_s$ ) pada *set point*  $30^\circ$  adalah sebesar 14s untuk panjang bandul 5cm dan 1s pada panjang bandul 1cm. Sedangkan pada *setpoint*  $15^\circ$  didapatkan  $T_s$  sebesar 10s untuk panjang bandul 5cm dan kurang dari 1s untuk panjang bandul 1cm.

## 6.2 Saran

Berdasarkan hasil pengujian program dan pembahasan pada tugas akhir ini, muncul beberapa saran untuk mengembangkan penelitian rancang bangun instrument pengukur sudut dengan metode olah citra secara *real-time*, antara lain:

1. Berdasarkan sifat dari sistem pendulum dengan menggunakan Coulomb damping yang memiliki *steady state error*, sebaiknya mekanisme bandul menggunakan *rotary viscous damper*. Hal tersebut bertujuan untuk menghilangkan kemungkinan terjadinya *steady state error*, serta mengurangi overshoot yang terjadi pada mekanisme bandul dan mempersingkat waktu *settling time*.
2. Berdasarkan penempatan posisi antara kamera dengan bandul dapat terjadi kesalahan parallax akibat ketidak sejajaran posisi kamera dan bandul sehingga objek bandul yang tertangkap kamera menjadi lebih kecil. Sehingga perlu lakukan pengaturan posisi untuk mendapatkan posisi bandul sejajar terhadap kamera, yang bertujuan untuk mengurangi kesalahan pembacaan posisi objek bandul.

## DAFTAR PUSTAKA

- [1] Yang, Kuihe., Cai, Zhiming. dan Zhao, Lingling., June 2013. “Algorithm Research on Moving Object Detection of Surveillance Video Sequence”, **Optic and Photonics Journal** 3, 308.
- [2] Prasad, Mukeshkumar., dan Nirwan, Nilesh W., Februari 2016. “Design and Fabrication of Automatic Balancing Bicycle”, **International Journal of Science, Engineering and Technology** 5, 4:534.
- [3] Park, Young-Soo., Agbayani, John Arbie., Lee, John-Han., dan Lee, Jong Jae., Januari 2016. “Rotational Angle Measurement of Bridge Support Using Image Processing Techniques”, **Journal of Sensor**.
- [4] Logitech C-170. **Datasheet Logitech International S.A (corporation)**. Lausanne, Switzerland.
- [5] Young, Ian T., Gerbrands, Jan J., dan Vliet, Lucas J. 2007. **Fundamentals of Image Processing**. Nederland: Delft University of Technology.
- [6] Gonzales, Rafael C., dan Woods, Richard E. 2008. **Digital Image Processing**. New Jersey. : Prentice Hall.
- [7] Garcia, Gloria Bueno., Suarez, Oscar Deniz., Aranda, Jose L.E., Tercero, Jesus S., Gracia, Ismael Serrano., dan Enano, Noelia Valez. 2015. **Learning Image Processing with OpenCV**. U.K.: Packt Publishing Ltd.
- [8] Rao, S.S. 2011. **Mechanical Vibrations: Fifth Edition SI Units**. New Jersey: Prentice Hall
- [9] Hibbeler, Russel.C. 2010. **Engineering Mechaics Dynamics 12<sup>th</sup> Edition**. New Jersey: Prentice Hall.
- [10] Motion Workshop., Desember 2017. **Motion Node Technology**.<URL:<https://www.motionnode.com/technology.html>>
- [11] Lestari, Ika Tri., Desember 2017. **Senyawa Pendulum**.,<URL:<https://ikatrilestari.wordpress.com/pendulum/>>

*[Halaman Ini Sengaja Dikosongkan]*

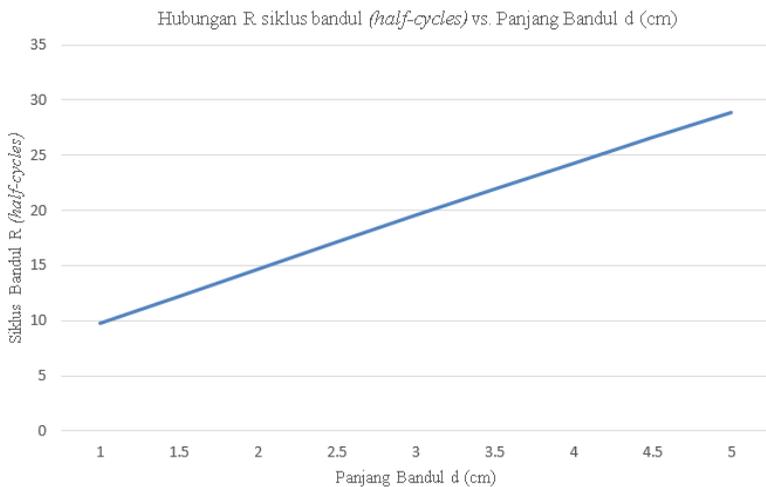
## LAMPIRAN A

Lampiran ini berisikan tabel Hasil perhitungan Mekanisme Bandul yang bertujuan untuk mendapatkan jumlah periode osilasi bandul  $T$ , *half-cycles*, dan simpang sudut saat keadaan *equilibrium*.

mbeban	L	mbatang	R	I batang bandul
12	4	1.18	4.8209408	6.195
$\rho$ batang	3	0.885	3.8969732	2.556666667
1.18	2	0.59	2.9531374	0.688333333
	1	0.295	1.9880033	0
Area	I beban	ltotal	m total	T (s)
0.25	192	198.195	13.18	0.354297515
	108	110.5567	12.885	0.297667836
g (cm/s <sup>2</sup> )	48	48.68833	12.59	0.229563653
981	12	12	12.295	0.140560462

<i>I batang bandul</i>				<i>I beban</i>			w.d		
$\rho$	Area	d	I	m	d	I	wbatang	wbeban	wd
1.18	0.25	1	0	12	2	48	289.395	11772	23688.7
		1.5	0.233542		2.5	75	434.0925		29755.57
		2	0.688333		3	108	578.79		35894.79
		2.5	1.438125		3.5	147	723.4875		42106.36
		3	2.556667		4	192	868.185		48390.28
		3.5	4.117708		4.5	243	1012.883		54746.54
		4	6.195		5	300	1157.58		61175.16
		4.5	8.862292		5.5	363	1302.278		67676.12
		5	12.19333		6	432	1446.975		74249.44

$\mu$	$\omega_n$	$W_{total}$	$T$	$r(\text{half cycle})$	$\Theta(r)$ (rad)	$\Theta(r)$ ( $^\circ$ )
0.1	22.21519	12061.4	603.0698	9.783531	0.025458	1.458643
	19.88742	12206.09	610.3046	12.2641	0.020511	1.175171
ro	18.1729	12350.79	617.5395	14.71722	0.017204	0.985725
0.5	16.84229	12495.49	624.7744	17.14384	0.014838	0.850155
	15.77088	12640.19	632.0093	19.54487	0.013061	0.748321
$\Theta(0)$	14.88423	12784.88	639.2441	21.92119	0.011676	0.66901
0.523599	14.13476	12929.58	646.479	24.27361	0.010568	0.605483
	13.49046	13074.28	653.7139	26.60294	0.009659	0.553445
	12.92887	13218.98	660.9488	28.90993	0.008902	0.510032



## LAMPIRAN B

Lampiran ini berisikan tabel hasil pengukuran sudut menggunakan olah citra dibandingkan dengan hasil pengukuran sensor IMU, yang mana pengukuran ini bertujuan untuk mendapatkan error dari pengukuran olah citra.

Sudut hasil Pengukuran $\Theta$ ( $^{\circ}$ )					
IMU	Olah Citra	IMU	Olah Citra	IMU	Olah Citra
-30	-30.14	-9	-9.03	11	11.93
-29	-29.35	-8	-8.17	12	12.54
-28	-28.74	-7	-7.25	13	13.74
-27	-27.51	-6	-6.09	14	14.75
-26	-26.59	-5	-5.61	15	15.78
-25	-25.01	-4	-4.77	16	16.52
-24	-24.11	-3	-3.55	17	17.71
-23	-23.11	-2	-2.15	18	18.92
-22	-22.09	-1	-1.01	19	19.93
-21	-21.31	0	0.378	20	20.77
-20	-20.37	0	0.378	21	21.64
-19	-19.03	1	1.76	22	22.57
-18	-18.78	2	2.78	23	23.77
-17	-17.21	3	3.73	24	24.85
-16	-16.04	4	4.92	25	25.46
-15	-15.04	5	5.67	26	26.71
-14	-14.35	6	6.71	27	27.55
-13	-13.47	7	7.46	28	28.61
-12	-12.09	8	8.46	29	29.35
-11	-11.34	9	9.68	30	30.95
-10	-10.21	10	10.57		

*[Halaman Ini Sengaja Dikosongkan]*

## LAMPIRAN C

### Konstruksi Program

```
#include <sstream>
#include <string>
#include <iostream>
#include <stdio.h>
#include <windows.h>
#include "opencv2/highgui/highgui.hpp"
#include "opencv2/imgproc/imgproc.hpp"
#include "opencv2/core/core.hpp"
#include <math.h>
#include <stdlib.h>

//image processing
using namespace cv;
using namespace std;

//default capture width and height
int FRAME_WIDTH = 320; //640
int FRAME_HEIGHT = 240; //480

int MIN_OBJECT_AREA = 5 * 5;

int iLowH = 0;
int iHighH = 360;

int iLowS = 0;
int iHighS = 255;

int iLowV = 0;
int iHighV = 255;

int centerX, centerY;

int Modefilter = 1;
```

```

double sudut;

char charCheckForEscKey;

string intToString(int number){

    std::stringstream ss;
    ss << number;
    return ss.str();
}

int main(int argc, char **argv){

    HANDLE hSerial = CreateFile(L"COM3", GENERIC_READ
| GENERIC_WRITE, 0, 0, OPEN_EXISTING,
FILE_ATTRIBUTE_NORMAL, 0);
    if (hSerial != INVALID_HANDLE_VALUE){
        printf("Port opened! \n");
        DCB dcbSerialParams;
        GetCommState(hSerial, &dcbSerialParams);
        dcbSerialParams.BaudRate = CBR_9600;
        dcbSerialParams.ByteSize = 8;
        dcbSerialParams.Parity = NOPARITY;
        dcbSerialParams.StopBits = ONESTOPBIT;
        SetCommState(hSerial, &dcbSerialParams);
    }
    else{
        if (GetLastError() == ERROR_FILE_NOT_FOUND){
            printf("Serial port doesn't exist! \n");
        }
        printf("Error while setting up serial port! \n");
    }
    char outputChars[] = "c";
    DWORD btsIO;

    //capture the video from web cam

VideoCapture cap(1);

```

```

while (1){
// if not success, exit program
if (!cap.isOpened()){
cout << "Cannot open the web cam" << endl;
return -1;
}

//set height and width of capture frame
cap.set(CV_CAP_PROP_FRAME_WIDTH, FRAME_WIDTH);
cap.set(CV_CAP_PROP_FRAME_HEIGHT, FRAME_HEIGHT);

//create a window called "Control"
namedWindow("Control", CV_WINDOW_AUTOSIZE);

/*Create trackbars in "Control" window*/
cvCreateTrackbar("LowH", "Control", &iLowH, 360);
//Hue (0 - 179) 26-125
cvCreateTrackbar("HighH", "Control", &iHighH, 360);
cvCreateTrackbar("LowS", "Control", &iLowS, 255);
//Saturation (0 - 255) 83-255
cvCreateTrackbar("HighS", "Control", &iHighS, 255);

cvCreateTrackbar("LowV", "Control", &iLowV, 223); //
Value(0 - 255) 85
cvCreateTrackbar("HighV", "Control", &iHighV, 255);

Mat imgOriginal;

bool bSuccess = cap.read(imgOriginal); // read a new
frame from video

if (!bSuccess){ //if not success, break loop
cout << "Cannot read a frame from video stream" <<endl;
}

//Convert the captured frame from BGR to HSV
Mat imgHSV;
cvtColor(imgOriginal, imgHSV, COLOR_BGR2HSV);

//find center point

```

```

centerX = FRAME_WIDTH / 2;
centerY = FRAME_HEIGHT / 2;

//Threshold the image
Mat imgThresholded;
Mat imgThresholded2;
inRange(imgHSV, Scalar(iLowH, iLowS, iLowV),
Scalar(iHighH, iHighS, iHighV), imgThresholded);
inRange(imgHSV, Scalar(iLowH, iLowS, iLowV),
Scalar(iHighH, iHighS, iHighV), imgThresholded2);

//morphological opening (remove small objects from the
foreground)
erode(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));
dilate(imgThresholded, imgThresholded,
getStructuringElement(MORPH_ELLIPSE, Size(5, 5)));

//morphological closing (fill small holes in the
foreground)
dilate(imgThresholded, imgThresholded, getStructuringElem
ent(MORPH_ELLIPSE, Size(5, 5)));
erode(imgThresholded, imgThresholded, getStructuringElem
ent(MORPH_ELLIPSE, Size(5, 5)));

//these two vectors needed for output of findContours
vector< vector<Point> > contours;
vector<Vec4i> hierarchy;

Mat imgContour;
imgThresholded.copyTo(imgContour);

//find contours of filtered image using openCV
findContours function
findContours(imgContour, contours, hierarchy,
CV_RETR_CCOMP, CV_CHAIN_APPROX_SIMPLE);

double datax[2], datay[2];
int jmlh = 0;
double depan, sampng;

```

```

//use moments method to find our filtered object
double refArea = 0;
if (hierarchy.size() > 0) {
int numObjects = hierarchy.size();
for (int index = 0; index <= numObjects - 1; index =
hierarchy[index][0]) {

Moments moment = moments((cv::Mat)contours[index]);
double area = moment.m00;
if (area > MIN_OBJECT_AREA){ //jika area kontur lebih
besar dari minimum area object maka gambar lingkaran
dan tulis koordinat
double x = moment.m10 / area;
double y = moment.m01 / area;
double r = sqrt(area / 3.14); //jari2 lingkaran

//gambar lingkaran luar mengikuti objek
circle(imgOriginal, Point(x, y), r, Scalar(0, 0, 255),
1.5, 8);

//gambar cross line di tengah lingkaran
line(imgOriginal, Point(x, y - r - 5), Point(x, y + r +
5), Scalar(0, 0, 255), 1.5, 8);
line(imgOriginal, Point(x - r - 5, y), Point(x + r + 5,
y), Scalar(0, 0, 255), 1.5, 8);

        if (jmlh < 2)
        {
            datax[jmlh] = x;
            datay[jmlh] = y;
            jmlh++;
        }
        else
        {
            jmlh = 0;
        }

    } //end if

```

```

//draw line 2 titik
    if (datay[0] < datay[1])
    {
        line(imgOriginal, Point(datax[0], datay[0]),
Point(datax[1], datay[1]), Scalar(0, 0, 255), 1.5, 8);
putText(imgOriginal, "1", Point(datax[0], datay[0]),
FONT_HERSHEY_COMPLEX, 0.25, Scalar(0, 255, 0), 0.3, 8);
depan = datax[1] - datax[0];
samping = datay[1] - datay[0];
    }
    else
    {
        line(imgOriginal, Point(datax[1], datay[1]),
Point(datax[0], datay[0]), Scalar(0, 0, 255), 1.5, 8);
putText(imgOriginal, "1", Point(datax[1], datay[1]),
FONT_HERSHEY_COMPLEX, 0.25, Scalar(0, 255, 0), 0.3, 8);
depan = datax[0] - datax[1];
samping = datay[0] - datay[1];
    }

    sudut = atan2(depan, samping) * 180 / 3.14159265;
    if (depan < 0)
    {
        putText(imgOriginal, intToString(sudut),
Point(datax[0] + 10, datay[0] + 10),
FONT_HERSHEY_COMPLEX, 0.25, Scalar(0, 255, 0), 0.3, 8);
    }
    else
    {
        putText(imgOriginal, intToString(sudut),
Point(datax[0] + 10, datay[0] + 10),
FONT_HERSHEY_COMPLEX, 0.25, Scalar(0, 255, 0), 0.3, 8);
    }
}

//create triangle line
line(imgOriginal, Point(datax[0], datay[0]),
Point(datax[1], datay[0]), Scalar(0, 255, 0), 1.5);

```

```

line(imgOriginal, Point(datax[1], datay[1]),
Point(datax[1], datay[0]), Scalar(255, 0, 0), 1.5);

/*imshow("Contour", imgContour);*/
cout << "sudut=" << (sudut) << ", " << "    "
    << "jmlh=" << endl;
    if (sudut >= 0 || sudut < 0){
        outputChars[0] = sudut;
        WriteFile(hSerial, outputChars,
        strlen(outputChars), &btsIO, NULL);
    }
    /*show the thresholded image*/
    Mat dstimgThresholded;
    resize(imgThresholded, dstimgThresholded, Size(),
2, 2, INTER_CUBIC);
    imshow("Thresholded Image", dstimgThresholded);
    Mat dstimgThresholded2;
    resize(imgThresholded2, dstimgThresholded2,
Size(), 2, 2, INTER_CUBIC);
    imshow("Thresholded2 Image", dstimgThresholded2);

    //show the original image
    Mat dstimgOriginal;
    resize(imgOriginal, dstimgOriginal, Size(), 2, 2,
INTER_CUBIC);
    imshow("Original", dstimgOriginal);

    //show the HSV image
    Mat dstimgHSV;
    resize(imgHSV, dstimgHSV, Size(), 2, 2,
INTER_CUBIC);
    imshow("HSV", dstimgHSV);

    if (waitKey(1) == 27){ //wait for 'esc' key press
for 30ms. If 'esc' key is pressed, break loop
        //cout << "esc key is pressed by user" << endl;
        break;
    }
}
}
}

```

*[Halaman Ini Sengaja Dikosongkan]*

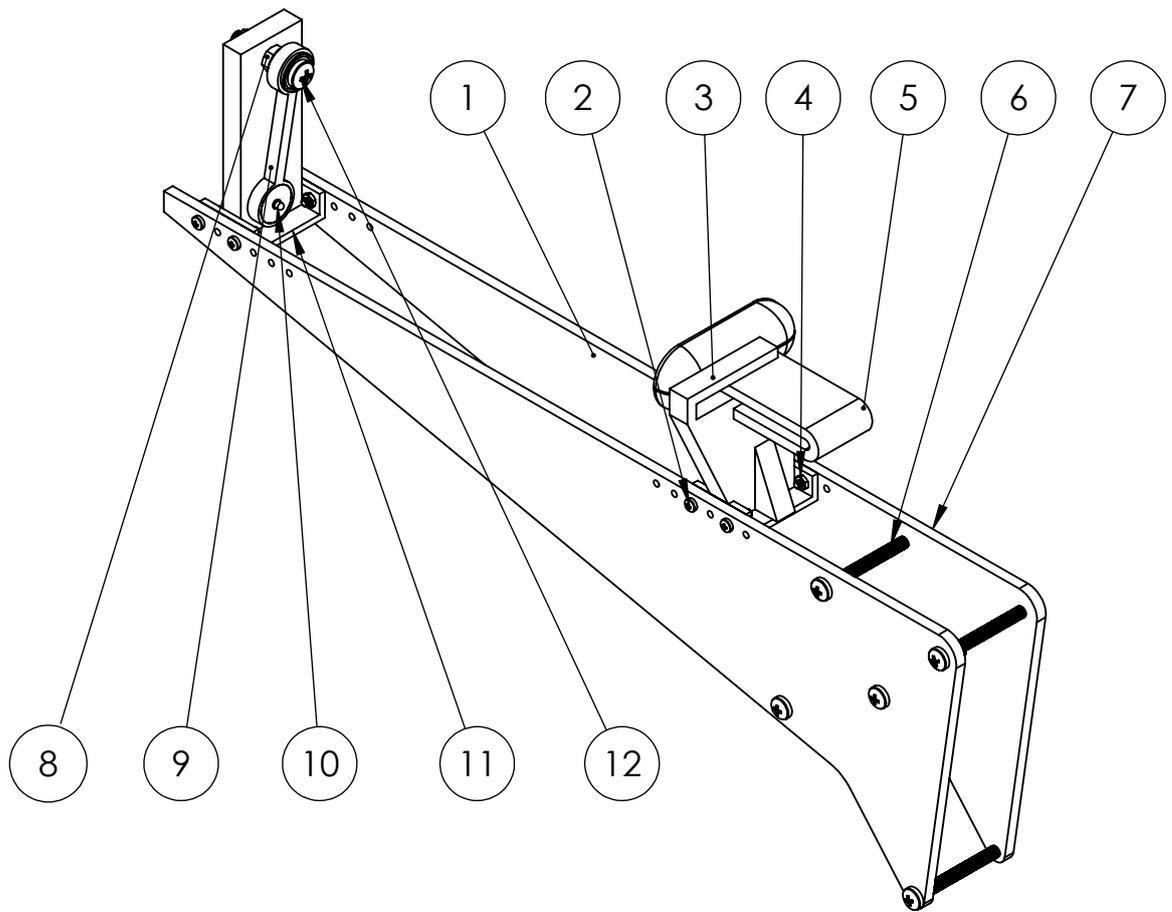
## BIODATA PENULIS



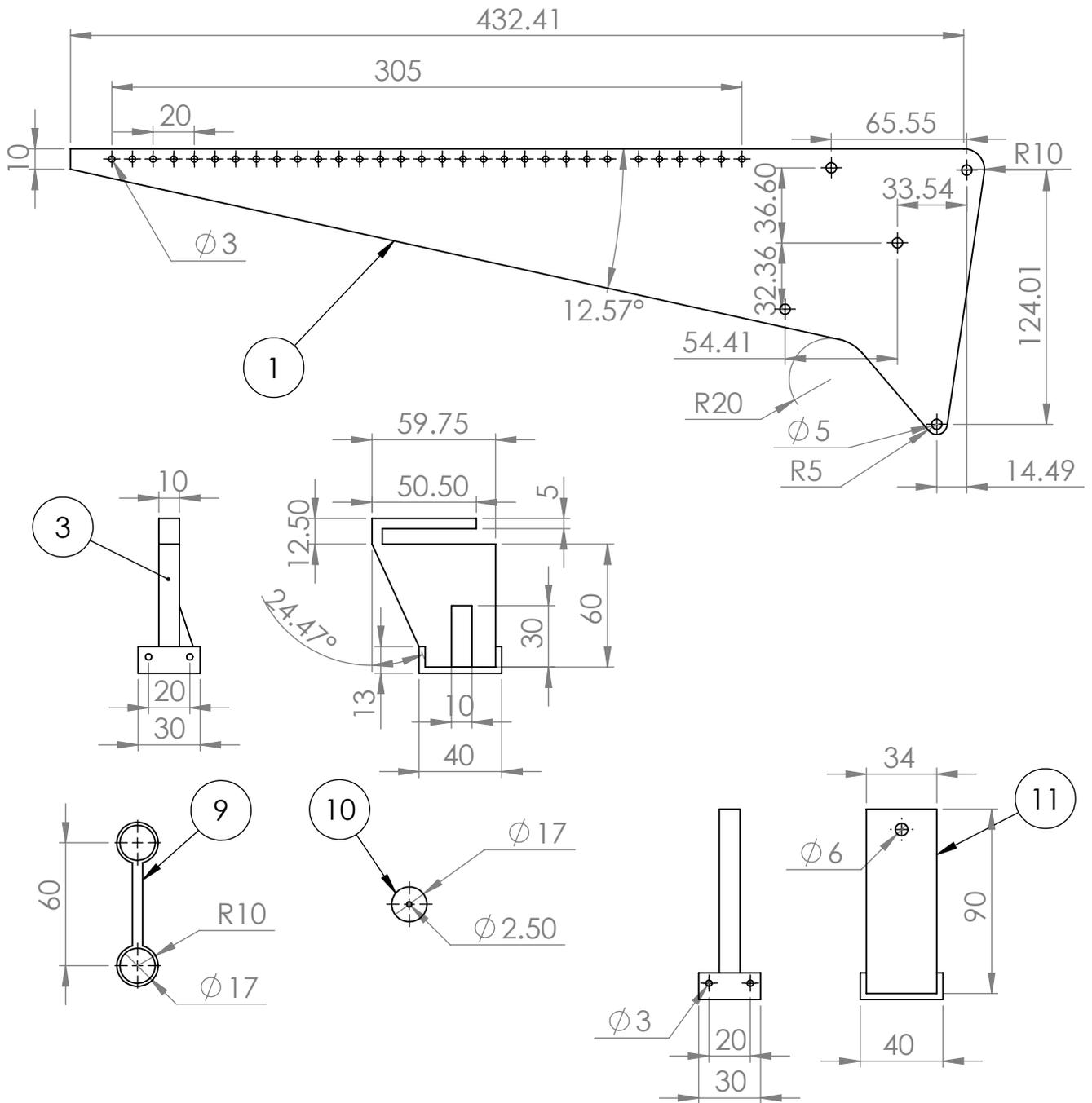
Penulis dilahirkan di Batam, 20 Februari 1994, merupakan anak terakhir dari tiga bersaudara. Penulis telah menempuh pendidikan formal antara lain TK Aisyah Pondok Bambu, SD 08 Pagi Duren Sawit, SMPN 109 Jakarta, dan SMAN 81 Jakarta. Setelah lulus dari SMA pada tahun 2012, penulis mengikuti SNMPTN dan diterima di Jurusan Teknik Mesin FTI-ITS dan terdaftar dengan NRP 2112100102. Di jurusan Teknik Mesin ini penulis

mengambil bidang studi Perancangan dan Pengembangan Produk. Penulis sempat aktif dalam bidang kemahasiswaan baik dalam berbagai kepanitiaan mahasiswa dan juga tergabung dalam organisasi Mesin Music Club kepengurusan 2013/2014. Penulis juga pernah berperan aktif sebagai asisten pengajar kegiatan pengabdian masyarakat di SMK 2 Bangkalan, grader mata kuliah menggambar teknik, asisten di kelas *Computer Aided Design*, dan sebagai asisten praktikum mata kuliah pengukuran teknik. E-mail yang bisa dihubungi yaitu [satrioramadhan2@gmail.com](mailto:satrioramadhan2@gmail.com).

*[Halaman Ini Sengaja Dikosongkan]*



12	1	Round Bolt	Aluminium	M10	
11	1	Dudukan Bandul	Acrylic		
10	1	Beban Bandul	Steel		
9	1	Batang bandul	Acrylic		
8	3	Hex Nut		M10	
7	15	Hex Nut		M8	
6	5	Round Bolt		M8	
5	1	Web Camera			Logitech C-170
4	8	Hex Nut	Aluminium	M3	
3	1	Dudukan Kamera	Acrylic		
2	8	Round Bolt	Aluminium	M3	
1	2	Plat Dasar Instrumen	Acrylic		
NO.	Jml.	NAMA BAGIAN	Bahan	Ukuran	Keterangan
		SKALA : 1:3	DIGAMBAR : Satrio Ramadhan		Peringatan :
		UKURAN : mm	NRP : 2112100102		
		TANGGAL : 1-12-17'	DIPERIKSA : Arif Wahjudi, S.T., M.T.		
TEKNIK MESIN ITS			Instrumen Pengukur Sudut		No. 1
					A4



12	1	Dudukan Bandul	Acrylic		
11	1	Beban Bandul	Steel		
10	1	Batang bandul	Acrylic		
3	1	Dudukan Kamera	Acrylic		
1	2	Plat Dasar Instrumen	Acrylic		
NO.	Jml.	NAMA BAGIAN	Bahan	Ukuran	Keterangan
	SKALA : 1:3		DIGAMBAR : Satrio Ramadhan		Peringatan :
	UKURAN : mm		NRP : 2112100102		
	TANGGAL : 1-12-17'		DIPERIKSA : Arif Wahjudi, S.T., M.T.		
TEKNIK MESIN ITS			Parts Instrumen		No. 2
					A4