

SISTEM DETEKSI POSISI OBJEK ACAK BERBASIS *IMAGE* *PROCESSING* PADA *PLATFORM* MYRIO

SKRIPSI

KEMINATAN TEKNIK KOMPUTER

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Alrynto

NIM: 135150301111082



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

SISTEM DETEKSI POSISI OBJEK ACAK BERBASIS *IMAGE PROCESSING* PADA
PLATFORM MYRIO

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh :

Alrynto

NIM: 135150301111082

Skripsi ini telah diuji dan dinyatakan lulus pada
27 Juli 2018

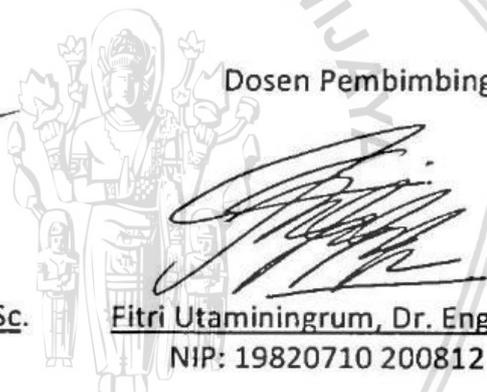
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Dahnial Syauqy, S.T., M.T., M.Sc.

NIK: 201607870423 1 002

Dosen Pembimbing II



Fitri Utamingrum, Dr. Eng., S.T, M.T

NIP: 19820710 200812 2 001

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T., M.T., Ph.D

NIP: 19710518 200312 1 001



PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 12 Juli 2018



Alrynto

NIM: 135150301111082

KATA PENGANTAR

Puji syukur penulis ungkapkan ke hadirat Tuhan Yang Maha Esa sebab oleh penyertaan Nyalah penulisan Skripsi dengan judul “SISTEM PENDETEKSI OBJEK ACAK BERBASIS *IMAGE PROCESSING* PADA *PLATFORM MYRIO*” ini dapat terselesaikan dengan baik.

Penulis dengan sadar menyadari bahwa penyusunan skripsi ini tidak terlepas dari bantuan dan dukungan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada segala pihak yang telah bersedia untuk memberikan bantuan demi kelancaran penyusunan skripsi ini, diantaranya:

1. Bapak **Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D.** selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
2. Bapak **Tri Astoto Kurniawan, S.T., M.T., Ph.D.** selaku Ketua Jurusan Teknik Informatika Fakultas Ilmu Komputer.
3. Bapak **Sabriansyah Rizqika Akbar, S.T, M.Eng** selaku Ketua Prodi Teknik Komputer Fakultas Ilmu Komputer.
4. Bapak **Dahnial Syauqy, S.T., M.T., M.Sc.** selaku dosen pembimbing satu yang telah memberikan ilmu, saran, dan motivasi kepada penulis.
5. Ibu **Fitri Utaminingrum, Dr.Eng., S.T., M.T.**, selaku dosen pembimbing dua yang telah memberikan ilmu, saran, dan motivasi kepada penulis.
6. Kedua orang tua dan seluruh keluarga besar atas segala kasih sayang, bantuan dukungan moril dan kesabaran dalam membesarkan dan mendidik penulis serta senantiasa mendukung dalam doa dan semangat dalam penulisan skripsi ini.
7. Seluruh civitas akademika Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberi bantuan selama penyelesaian skripsi ini.
8. Seluruh dulur Tekom yang selalu mendukung dan berbagi ilmu serta yang telah memberi pengalaman berharga selama kuliah.
9. Teman-teman Himawibu yang terus memberikan dorongan dan semangat kepada penulis.
10. Teman-teman PPGT Jemaat Malang Raya yang tiada henti setia mendoakan penulis dalam menyusun skripsi ini
11. Teman-teman dan saudara sang Torayan yang terus memotivasi dan memberikan dorongan kepada penulis.
12. Semua pihak yang telah membantu dan berbagi ilmu dalam penyelesaian penelitian ini yang tidak dapat penulis sebutkan satu per satu.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih jauh dari kata sempurna, sehingga penulis mengharapkan adanya bantuan dalam hal ini kritik

dan saran yang membangun. Akhir kata penulis berharap skripsi ini bisa membawa
guna dan manfaat bagi semua pihak yang menggunakannya.

Malang, 12 Juli 2018

Penulis

alrynto@live.com



ABSTRAK

Mendeteksi objek dalam hal ini objek acak merupakan salah satu fokus utama dalam pengembangan ilmu komputer. Pendeteksian objek sangat berguna untuk penerapan di bidang ilmu yang lain seperti pemetaan, industri, medis, keamanan dan lain-lain. Untuk dapat diterapkan dengan ilmu yang lain diperlukan keluaran yang dapat berintegrasi dengan ilmu tersebut. Keluaran dari penelitian ini ialah nilai jarak yang sudah terkalibrasi dengan satuan internasional. Penelitian ini melakukan pengembangan terhadap pendeteksian objek acak menggunakan mikrokontroler NI MyRIO-1900 sebagai pemroses citra dan sebuah kamera *webcam* sebagai sensor optik. Kamera mengambil citra dari atas objek. Sistem menggunakan pengolahan citra dengan metode *Geometric Match Pattern* yang akan melakukan pencocokan citra objek dengan citra *template* secara geometri yang kemudian akan mengambil data berupa posisi dan kemiringan objek. Penulisan kode program menggunakan *graphical programming* pada *software* LabVIEW. Hasil dari pengolahan citra ditampilkan pada komputer. Pada pengujian menggunakan satu jenis objek uji berupa penghapus persegi panjang yang diletakkan secara acak. Pengujian dilakukan dengan menguji fungsi dari sistem membaca jarak dan kemiringan serta akurasi koordinat posisi objek. Pengujian jarak mendapatkan persentase keberhasilan sebesar 99.63% dan pengujian kemiringan mendapatkan persentase keberhasilan sebesar 96.8%. Pengujian akurasi koordinat objek menggunakan selisih toleransi pada *pixel* dan didapatkan hasil tertinggi sebesar 100% pada toleransi 5 *pixel*.

Kata kunci: Pengolahan citra, Kamera, Geometric Match Pattern, NI MyRIO

ABSTRACT

Object recognition in this case position of random object is one of the main concern for development of computer science. Object recognition is very useful in another major implementation like maaping, industrial manufacturing, medical, security and many more. For implementation to another major, output of the system need to calibrate for that major. System output on this research are distance and slope value that have been caliberated into international value of length. This research try to develop object recognition using microcontroler by National Instrument MyRIO 1900 as image processor and webcam camera as optical sensor. The camera take object image from above of the object. System using image processing with Geometric Match Pattern method that will match object image with template image geometrically and then will take data position and slope of the object. Output from image processing will show in the computer. In the testing will use one type of object in this case using reactangle and placed randomly. The testing will measure the function of the system for read object position, slope and manualization accuracy. Position test have percentage success 99.63% and slope test have percentage success 96.8%. Accuracy test using pixel tolerance and the highest accuracy 100% on 5 pixel tolerance.

Keywords : Image Processing, Camera, Geometric Match Pattern, NI MyRIO

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI.....	viii
DAFTAR TABEL.....	xi
DAFTAR GAMBAR.....	xii
BAB 1 PENDAHULUAN.....	1
1.1 Latar belakang.....	1
1.2 Rumusan masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan masalah	3
1.6 Sistematika pembahasan.....	4
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Dasar Teori	6
2.2.1 Citra Digital.....	6
2.2.2 Pengolahan Citra Digital.....	7
2.2.3 <i>Geometric Match Pattern</i>	7
2.2.4 Ketetapan Pola dan Acak.....	9
2.2.5 Jarak dan Trigonometri	9
2.2.6 NI MYRIO-1900.....	11
2.2.7 Webcam	14
2.2.8 NI LabVIEW Graphical Programming	14
2.2.9 NI LabVIEW MyRIO Toolkit.....	18
2.2.10 NI LabVIEW <i>Vision Acquisition Software</i>	18
BAB 3 METODOLOGI	20
3.1 Studi Literatur	21
3.2 Rekayasa Kebutuhan.....	21



3.2.1	Rekayasa Kebutuhan Perangkat Keras.....	21
3.2.2	Rekayasa Kebutuhan Perangkat Lunak	21
3.2.3	Kebutuhan Sistem	22
3.3	Perancangan Sistem.....	22
3.3.1	Perancangan Perangkat Keras	22
3.3.2	Perancangan Perangkat Lunak.....	23
3.4	Implementasi	23
3.5	Pengujian Sistem.....	24
3.6	Analisis Pengujian	24
3.7	Penarikan Kesimpulan	24
BAB 4	Rekayasa kebutuhan.....	25
4.1	Gambaran Umum Sistem.....	25
4.2	Tujuan Sistem.....	25
4.3	Analisis Kebutuhan Sistem.....	25
4.3.1	Kebutuhan Pengguna	25
4.3.2	Kebutuhan Fungsional.....	26
4.3.3	Kebutuhan Non-Fungsional.....	26
4.4	Batasan Perancangan Sistem.....	27
4.5	Asumsi dan Ketergantungan.....	28
BAB 5	Perancangan dan implementasi.....	29
5.1	Perancangan Sistem.....	29
5.1.1	Perancangan Perangkat Keras	29
5.1.2	Perancangan Perangkat Lunak.....	30
5.2	Implementasi Sistem	40
5.2.1	Implementasi Perangkat Keras	40
5.2.2	Implementasi Perangkat Lunak.....	41
BAB 6	pengujian dan analisis	49
6.1	Pengujian Fungsional	49
6.1.1	Pengujian Akuisisi Objek	49
6.1.2	Pengujian Penghitungan Jarak Objek.....	51
6.1.3	Pengujian Kemiringan	54
6.2	Pengujian Akurasi Koordinat Objek	56



6.2.1 Tujuan Pengujian.....	56
6.2.2 Prosedur Pengujian	56
6.2.3 Hasil Pengujian	56
6.2.4 Analisis Hasil Pengujian	58
6.3 Pengujian Waktu Pemrosesan	58
6.3.1 Tujuan Pengujian.....	58
6.3.2 Prosedur Pengujian	58
6.3.3 Hasil Pengujian	59
6.3.4 Analisis Hasil Pengujian	60
BAB 7 penutup	61
7.1 Kesimpulan.....	61
7.2 Saran	62
DAFTAR PUSTAKA.....	63



DAFTAR TABEL

Tabel 2.1 Spesifikasi NI MyRIO-1900	11
Tabel 2.2 Kode Program LabVIEW	15
Tabel 2.3 Kode Program Vision Acquisition Software.....	17
Tabel 6.1 Hasil Pengujian Penghitungan Jarak (cm).....	52
Tabel 6.2 Hasil Pengujian Penghitungan Oposit (cm)	55
Tabel 6.3 Hasil Pengujian Koordinat Objek	56
Tabel 6.4 Hasil Pengujian Waktu Pemrosesan Tanpa Highlight	58
Tabel 6.5 Hasil Pengujian Waktu Pemrosesan Dengan Highlight	59



DAFTAR GAMBAR

Gambar 2.1 Algoritma Match Pattern	7
Gambar 2.2 Fitur	8
Gambar 2.3 Algoritma Learning Phase	8
Gambar 2.4 Algoritma Matching Phase	9
Gambar 2.5 Jarak Antara Dua Posisi	10
Gambar 2.6 Segitiga Trigonometri	10
Gambar 2.7 NI MyRIO-1900	11
Gambar 2.8 Detail MyRIO-1900	11
Gambar 2.9 Blok Diagram MyRIO-1900	12
Gambar 2.10 Webcam	14
Gambar 2.11 Tampilan Front Panel LabVIEW	15
Gambar 2.12 Tampilan Block Diagram dan Palette LabVIEW	15
Gambar 2.13 Tipe Data Kabel	17
Gambar 2.14 MyRIO Palette	18
Gambar 3.1 Diagram Alir Metode Penelitian	20
Gambar 3.2 Diagram Blok	22
Gambar 3.3 Rangkaian Letak Sistem	23
Gambar 5.1 Diagram Blok Aliran Data	29
Gambar 5.2 MyRIO Tampak Depan	29
Gambar 5.3 Rancangan Tiang Penyangga	30
Gambar 5.4 Diagram Blok Proses	30
Gambar 5.5 Diagram Alir Perancangan Perangkat Lunak	31
Gambar 5.6 Diagram Alir Akuisisi Citra	32
Gambar 5.7 Diagram Alir Input Template	33
Gambar 5.8 Diagram Alir RGB to Grayscale	34
Gambar 5.9 Diagram Alir Geometric Pattern Matching	35
Gambar 5.10 Diagram Alir Overlaying	36
Gambar 5.11 Titik Tengah Objek	37
Gambar 5.12 Posisi Koordinat Objek	37
Gambar 5.13 Diagram Alir Manualisasi Objek	38
Gambar 5.14 Kemiringan Objek	39



Gambar 5.15 Implementasi Perangkat Keras.....	40
Gambar 5.16 Implementasi Alat Penyangga.....	41
Gambar 5.17 Implementasi Sistem Perangkat Keras.....	41
Gambar 5.18 Implementasi Akuisisi Citra	42
Gambar 5.19 Implementasi Input Template	43
Gambar 5.20 Implementasi RGB to Grayscale	44
Gambar 5.21 Implementasi Geometric Match Pattern	45
Gambar 5.22 Implementasi Overlaying	46
Gambar 5.23 Implementasi Manualisasi Objek	47
Gambar 5.24 Implementasi Front Panel	48
Gambar 6.1 Hasil Pengujian Akuisisi Objek.....	50
Gambar 6.2 Luas Cakupan Kamera	51
Gambar 6.3 Hasil Pengujian Penghitungan Jarak.....	53
Gambar 6.4 Hasil Pengujian Kemiringan.....	55
Gambar 6.5 Hasil Pengujian Akurasi Koordinat Objek.....	57
Gambar 6.6 Grafik Pengujian Waktu Pemrosesan Tanpa Highlight	59
Gambar 6.7 Grafik Pengujian Waktu Pemrosesan Dengan Highlight	60



BAB 1 PENDAHULUAN

1.1 Latar belakang

Seiring dengan perkembangan zaman yang semakin maju, teknologi informasi merupakan salah satu penggerak utama abad ini. Bahkan perkembangan teknologi informasi sudah sangat pesat dan untuk kita melihat satu per satu perkembangannya sudah sangat sulit. Khusus pada bidang Rekayasa Perangkat Cerdas, sudah banyak serangkaian pengembangan dan inovasi keterbaruan. Terlebih untuk bidang *image processing* (pemrosesan citra), perlu dikembangkan lebih lanjut karena tantangan ke depan ialah perangkat cerdas di tuntut untuk mampu menangkap sinyal-sinyal fisis untuk di ubah ke dalam satuan digital dalam hal ini ialah sinyal optik. Salah satu cabang terapan dari pemrosesan citra ialah sistem informasi geografis yang merupakan penggabungan beberapa teknik dalam pemrosesan citra yaitu teknik asembli, integrasi, analisis, tabulasi dan basis data grafis yang berguna pada lingkungan maupun terapan manusia sendiri. Hal ini sangat baik dilakukan pada kegiatan survey maupun *mapping* (Krishnamurthy, 2002).

Dengan adanya pemrosesan citra diperlukan keluaran yang dapat memberikan informasi atau data yang dapat dibaca oleh pengguna sesuai dengan bidang keilmuan masing-masing. Hasil keluaran pemrosesan citra biasanya berupa histogram atau dalam bentuk nilai *pixel*. Diperlukan kalibrasi untuk dapat mengubah nilai *pixel* menjadi ukuran-ukuran dalam dunia nyata misalnya jarak dalam meter. Kalibrasi dapat didapatkan dari perbandingan antara pengukuran biasa dengan pengukuran sistem. Dari situ di dapatkan akurasi dan rasio (Brei, 2013).

Pengolahan citra digital mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri atau *feature images* yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung dalam citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi dan waktu proses data (Sutoyo, 2009).

Pengenalan objek atau *object recognition* merupakan salah satu terapan dari *image processing* untuk mengenali objek-objek yang akan dikenali untuk diolah lebih lanjut agar mendapatkan suatu data informasi. Dengan adanya proses pengenalan objek tersebut, manusia akan di permudah dalam mengenali bentuk dari objek seperti pemetaan geografis, penggunaan sensor pada benda untuk sistem keamanan, pembacaan citra hasil scan medis, dan lain-lain. Terdapat beberapa metode untuk mengenali objek. Salah satu metode itu *pattern matching* yaitu mengenali objek dengan cara melakukan pencocokan antara gambar objek dengan contoh objek yang bisa kita sebut *template*.

Salah satu metode yang dapat digunakan sebagai *pattern matching* ialah *Geometric Match Pattern*. *Geometric Match Pattern* ialah suatu pengenalan objek

dengan menggunakan *template*. *Geometric Match Pattern* dapat mengenali objek yang ukuran, posisi relatif dan sifat ruang yang berbeda dengan *template* selama bentuk dari objek tidak berubah. Pengenalan objek menggunakan metode ini dinilai sangat cocok dengan objek yang posisi, ukuran dan kemiringan yang acak. *Geometric Match Pattern* merupakan hasil dari penelitian yang dilakukan oleh National Instrument Corp (Nair et al, 2008).

Keacakan dinilai penting untuk dikenali karena acak menggambarkan kondisi yang biasanya tidak sesuai dengan pola yang diinginkan. Acak yang dimaksud dalam sistem ini ialah acak secara posisi yaitu posisi peletakan secara *random* dan juga acak secara orientasi kemiringan. Hal ini dinilai perlu karena pada *real world* keacakan itu sering ditemui.

Untuk melakukan suatu pengenalan objek dibutuhkan sebuah sistem yang dapat menerima masukan citra dan memberi keluaran berupa data dari objek. Disamping itu dibutuhkan sistem yang dapat menghasilkan keluaran sistem berupa nilai jarak universal atau meter. Sistem ini dapat dirancangan dalam sebuah rekayasa perangkat cerdas. Dalam melakukan sebuah rekayasa perangkat cerdas tentunya di butuhkan perangkat cerdas itu sendiri. Untuk memenuhi kebutuhan sistem untuk melakukan pengakuisisian citra yang menggunakan kamera maka diperlukan sebuah kontroler yang dapat melakukan proses tersebut. Untuk penelitian ini akan menggunakan Mikrokontroler MyRIO dari National Instrument. Perangkat ini sebagai tempat pemrosesan citra yang dapat di ambil dari sensor optikal berupa kamera dan tempat melakukan penghitungan dan analisis dari data yang telah diolah.

NI MyRIO sendiri memiliki port USB 2.0 yang dapat disambungkan langsung dengan kamera webcam. NI MyRIO sebelumnya harus menginstall beberapa fitur yaitu *NI Vision* sebagai *library* pembacaan kamera. National Instrument sendiri memiliki beberapa metode *pattern matching*. Salah satunya ialah *Geometric Match Pattern* yang dapat diimplementasikan pada NI MyRIO. *Geometric Match Pattern* akan melakukan pencocokan antara citra objek dan citra *template*.

Setelah proses pencocokan citra dilakukan, perlu dilakukan tindakan atau aksi terhadap luaran yang didapatkan. Koordinat objek-objek acak pada citra akan di simpan dan dilakukan analisis. Pada penelitian ini akan berfokus pada pengembangan menggunakan titik koordinat x dan y yang didapatkan dari luaran proses citra. Hasil dari analisis ini akan menilai apakah posisi objek objek ini sudah tertata dengan baik atau acak (tidak teratur).

Berdasarkan latar belakang ini, penulis terinspirasi untuk melakukan sebuah penelitian terhadap pembacaan objek acak dengan menggunakan *platform* NI MyRIO.

1.2 Rumusan masalah

Rumusan masalah pada skripsi ini ialah:

1. Bagaimana menerapkan Object Recognition dan pembacaan koordinat x dan y dengan menggunakan *platform* MyRIO?

2. Bagaimana nilai akurasi akuisisi jarak objek acak dengan menggunakan *platform* MyRIO?
3. Bagaimana nilai akurasi akuisisi kemiringan objek acak dengan menggunakan *platform* MyRIO?
4. Bagaimana nilai akurasi koordinat objek pada masing-masing selisih toleransi (0 *pixel*, 3 *pixel*, 5 *pixel*) menggunakan *platform* MyRIO?
5. Bagaimana waktu pemrosesan *matching* menggunakan *platform* MyRIO?

1.3 Tujuan

Tujuan pada skripsi ini ialah:

1. Menerapkan metode *Object Recognition* dengan keluaran koordinat x dan y ke dalam sistem dengan menggunakan *platform* MyRIO
2. Menguji nilai akurasi akuisisi jarak objek acak dengan menggunakan *platform* MyRIO.
3. Menguji nilai akurasi akuisisi kemiringan objek acak dengan menggunakan *platform* MyRIO.
4. Menguji nilai akurasi koordinat objek pada masing-masing selisih toleransi (0 *pixel*, 3 *pixel*, 5 *pixel*) menggunakan *platform* MyRIO.
5. Menguji waktu pemrosesan *matching* menggunakan *platform* MyRIO.

1.4 Manfaat

Manfaat penelitian dari penelitian ini ialah mahasiswa mampu untuk mengimplementasikan salah satu metode *image processing* dalam hal ini *object recognition* ke dalam sebuah sistem perangkat cerdas berupa platform MyRIO dari National Instrument.

1.5 Batasan masalah

Agar pembahasan di dalam penelitian ini dapat dilakukan secara terarah sehingga mendapatkan hasil sesuai tujuan penelitian, maka perlu diterapkannya batasan penelitian atau batasan masalah. Terdapat beberapa batasan masalah yang dapat dirumuskan, antara lain :

1. Keluaran sistem hanya berupa perintah manggeser.
2. Kamera yang digunakan ialah kamera webcam.
3. Pengambilan gambar dilakukan dari atas objek.
4. Menggunakan software dan bahasa pemrograman Labview.
5. Bentuk dan ukuran bahan uji tetap yaitu penghapus.
6. Sistem melakukan pengambilan gambar pada cahaya 500-750 lux.

1.6 Sistematika pembahasan

Penjelasan singkat mengenai isi dari masing - masing bab adalah sebagai berikut :

BAB 1 Pendahuluan

Bab dari Pendahuluan ini akan dijelaskan mengenai latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan sistematika pembahasan dari penelitian.

BAB 2 Landasan Kepustakaan

Bab dari Landasan Kepustakaan ini berisi tentang tinjauan pustaka dan dasar teori yang dimanfaatkan sebagai rujukan dalam penelitian.

BAB 3 Metode Penelitian

Bab dari Metode Penelitian ini akan menjelaskan tentang langkah kerja atau metode yang digunakan dalam menyelesaikan masalah pada penelitian diantaranya studi literatur, analisis sistem, perancangan, implementasi, pengujian dan analisis pengujian.

BAB 4 Rekayasa Kebutuhan

Bab Rekayasa Kebutuhan meliputi tentang deskripsi kebutuhan kebutuhan umum sistem, kebutuhan fungsional dari sistem, kebutuhan perangkat keras, kebutuhan perangkat lunak dan batasan sistem.

BAB 5 Perancangan dan Implementasi

Bab Perancangan dan Implementasi merupakan penjelasan dari perancangan sistem perangkat keras dan perangkat lunak serta bagaimana cara implementasi sistem dari rancangan perangkat keras dan perangkat lunak yang telah dibuat sebelumnya.

BAB 6 Pengujian dan Analisis

Bab Penujian dan Analisis membahas tentang skenario pengujian yang akan dilakukan, kemudian hasil dari pengujian tersebut akan dianalisis.

BAB 7 Penutup

Bab Penutup merupakan penjelasan kesimpulan dan saran dari hasil penelitian ini berdasarkan rumusan masalah dan hasil dari pengujian dan analisis.

BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan membahas teori, perangkat, dan penelitian-penelitian terdahulu bersangkutan dengan penelitian “Sistem Deteksi Posisi Objek Acak berbasis *Image Processing* pada *platform* MyRIO”. Landasan kepastakaan dilakukan untuk memperkuat pengetahuan penulis tentang teori dan perangkat dalam memulai proses penelitian

Pada penelitian ini, pustaka yang digunakan sebagai rujukan terkait *image processing* bersumber dari *e-book* yang berjudul “*Digital Image Processing*” dari Rafael C. Gonzales dan Richard E. Woods. Kemudian beberapa referensi pendukung dalam perancangan sistem antara lain datasheet dari mikrokontroler MyRIO dan sensor optik.

2.1 Kajian Pustaka

Berdasarkan hasil pengamatan yang dilakukan penulis, dapat dipastikan bahwa penelitian sebelumnya terdapat penelitian yang membahas tentang *object recognition* dan metode tentang *geometric match pattern*. Oleh sebab itu penulis menggunakan beberapa jurnal terkait penelitian sebelumnya.

1. **Sistem Penjejak Objek Berbasis Umpan Balik Citra** (Syahrul & Andy Kurniawan, 2012)

Dalam penelitian ini membahas tentang perancangan dan realisasi sistem penjejak (*tracking*) Posisi Objek menggunakan bantuan pemrosesan citra. Keberadaan benda dideteksi sensor visual. Citra hasil *capture* diproses oleh aplikasi *image processing* menggunakan pemrograman Delphi untuk menghasilkan data koordinat yang akan menentukan keberadaan benda pada *plant*. Data koordinat akan dikirimkan ke mikrokontroler untuk menggerakkan *laser pointer* dengan menggunakan *motor stepper*. Ketepatan posisi penjejukan laser pointer terhadap benda akan ditunjukkan oleh titik koordinat x dan y yang diperoleh dari hasil pemrosesan citra.

2. **Geometric Properties Of Matrices Induced By Pattern Avoidance** (Andrea Frosini, Veronica Guerrini & Simone Renaldi, 2016)

Pada penelitian ini membahas tentang penggunaan *geometric pattern* dalam menghindari permutasi dari polyomino. Polyomino merupakan sebuah bentuk geometrik yang dibentuk dari beberapa persegi sejenis dengan menggabungkan sisi-sisinya. Penelitian ini mempertimbangkan dua jenis kendala geometrik yaitu konveksitas dan keteraturan (*convexity and directedness*) dan penelitian ini mempelajari hal tersebut dengan menggunakan pencegahan secara submatrik.

3. **Understanding Of Indoor Scenes Based On Projection Of Spatial Rectangles** (Hui Wei & Luping Wang, 2018)

Dalam penelitian ini membahas tentang pendeteksi bentuk persegi atau persegi panjang dalam sebuah ruangan *indoor*. Penelitian ini mengambil

persegi sebagai dasar dari pemetaan dari ruangan tersebut. Persegi atau persegi panjang dinilai dapat memberikan informasi tentang isi ruangan, mulai dari pintu, jendela, dinding dan meja. Penelitian ini menggunakan *geometric pattern* sebagai metode pengolahan citra. Hasil dari penelitian ini ialah otomatisasi robot dalam mengenali ruangan.

2.2 Dasar Teori

2.2.1 Citra Digital

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/ diskrit nilai digital yang disebut *pixel* atau *picture elements*. Dalam tinjauan matematis, citra merupakan fungsi kontinu dari intensitas cahaya pada bidang dua dimensi.

Citra digital adalah citra $f(x, y)$ dimana dilakukan diskritisasi koordinat sampling atau spasial dan diskritisasi tingkat kuantisasi (kabuan/ kecemerlangannya). Citra digital merupakan fungsi intensitas cahaya $f(x, y)$, dimana harga x dan harga y adalah koordinat spasial. Harga fungsi tersebut di setiap titik (x, y) merupakan tingkat kecemerlangan citra pada titik tersebut.

Format citra digital dapat dinyatakan dalam sebuah matrik. Matrik yang dinyatakan Citra digital yaitu dengan matriks berukuran N (baris/tinggi) x M (kolom/lebar). Fungsi $f(x, y)$ dapat dilihat pada Persamaan 2.1.

N = jumlah baris $0 \leq y \leq N - 1$

M = jumlah kolom $0 \leq x \leq M - 1$

L = maksimal warna intensitas $0 \leq f(x, y) \leq L - 1$.

gray level/ derajat keabuan).

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

Ukuran citra dapat juga dinyatakan dalam ukuran fisik dalam satuan panjang (misalnya mm atau inch). Dalam hal ini tentu saja harus ada hubungan antara titik penyusunan citra dengan satuan panjang. Hal tersebut dinyatakan dengan resolusi yang merupakan ukuran banyaknya titik untuk setiap satuan panjang. Makin besar resolusinya makin banyak titik yang terkandung dalam citra dengan ukuran fisik yang sama. Hal ini memberikan efek penampakan citra menjadi semakin halus.

Citra yang merepresentasikan informasi hanya bersifat biner untuk membedakan 2 keadaan tertentu tidak sama dengan informasi yang lebih kompleks sehingga memerlukan lebih banyak keadaan yang diwakilinya. Pada citra digital semua informasi tadi disimpan dalam bentuk angka, sedangkan penampakan angka tersebut biasanya dikaitkan dengan warna.

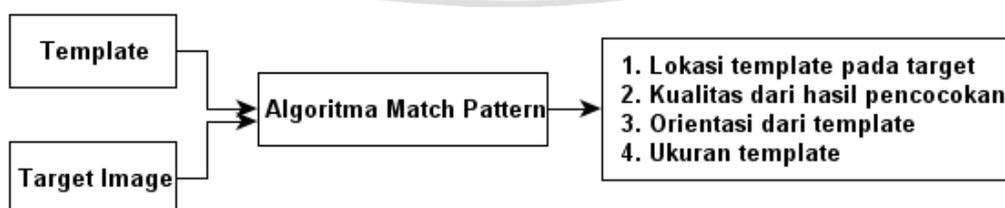


2.2.2 Pengolahan Citra Digital

Citra adalah representasi dari dua dimensi untuk bentuk fisik nyata tiga dimensi. Dalam perwujudannya, citra dibagi menjadi 2 yaitu citra diam (*still images*) dan citra bergerak (*moving images*). Citra diam adalah citra tunggal yang tidak bergerak. Sedangkan citra bergerak adalah rangkaian citra diam yang ditampilkan secara sekuensial sehingga memberi kesan pada mata kita sebagai gambar yang bergerak. Agar citra yang mengalami gangguan mudah diinterpretasi baik oleh manusia maupun mesin, maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik atau disebut pemrosesan citra (*image processing*). Karena pengolahan citra dilakukan dalam komputer digital, maka citra yang akan diolah terlebih dahulu ditransformasikan kedalam bentuk besaran-besaran diskrit pada titik-titik elemen citra, dan bentuk dari citra ini disebut citra digital.

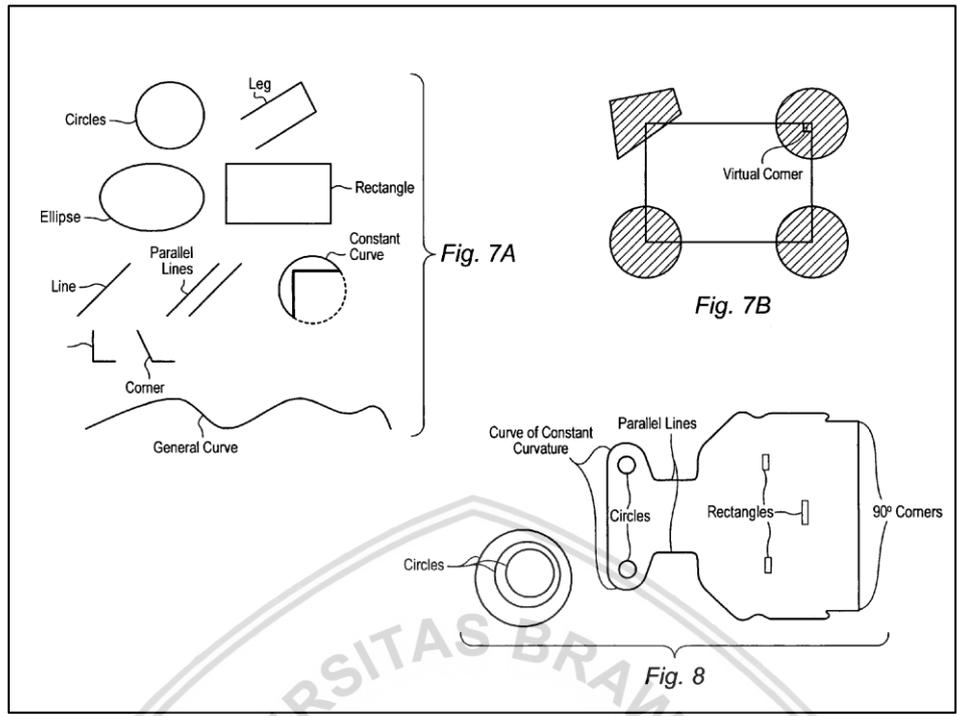
2.2.3 Geometric Match Pattern

Pada jurnal yang diterbitkan oleh National Instrument Corp., dengan judul *Geometric pattern matching using dynamic feature combinations* pada tahun 2008 yang menjelaskan tentang metode *Geometric Match Pattern* itu sendiri. *Geometric match pattern* adalah metode untuk mengetahui lokasi dari objek yang contohnya sudah dijadikan *template* dengan mengenali keberagaman sisi-sisi model dengan menggabungkan beberapa fitur atau ciri atau karakteristik yang berbeda. Fase belajar atau *learning phase* melibatkan *learning geometric information* atau mempelajari informasi geometri tentang citra *template* dan membuat model untuk setiap keberagaman dari perbedaan masing-masing fitur *parent* pada *template*. Setiap model terdiri dari *relationship tree* atau pohon relasi antara *parent* terhadap satu atau lebih fitur *child* dan karenanya setiap model mungkin memiliki beberapa kombinasi fitur yang berbeda. Pada fase pencocokan, metodenya akan memeriksa perbedaan dari model untuk menentukan kecocokan pada target yang di analisis. Pada fase ini mungkin akan menyeleksi perbedaan model secara dinamis berpatokan pada hasil pencocokan sebelumnya. Secara garis besar *Geometric Match Pattern* dapat dilihat pada Gambar 2.1.



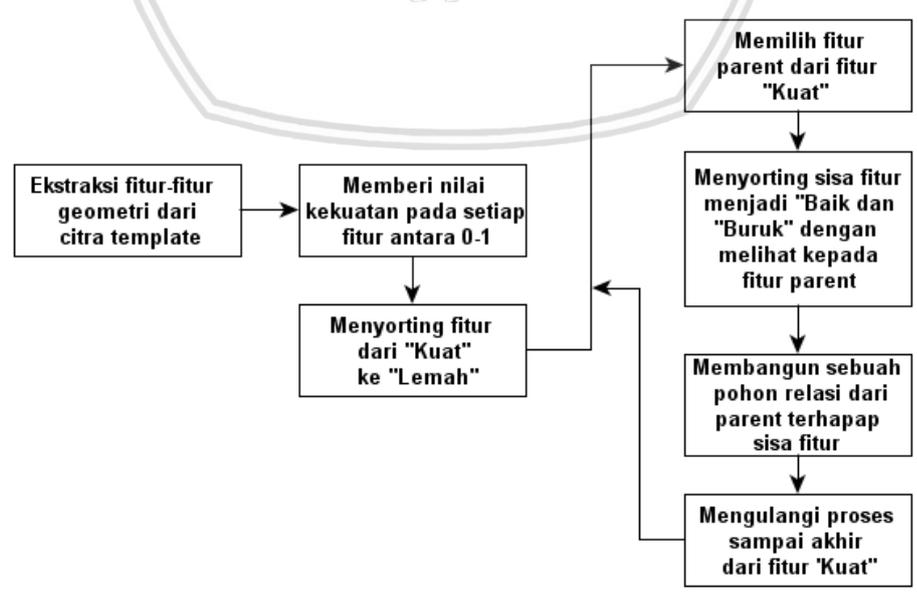
Gambar 2.1. Algoritma *Match Pattern*

Citra target bersama dengan *template* akan dilakukan proses *Match Pattern* yang akan menghasilkan keluaran berupa lokasi *template* pada target, kualitas *template*, orientasi *template* dan ukuran *template* hasil pencocokan.



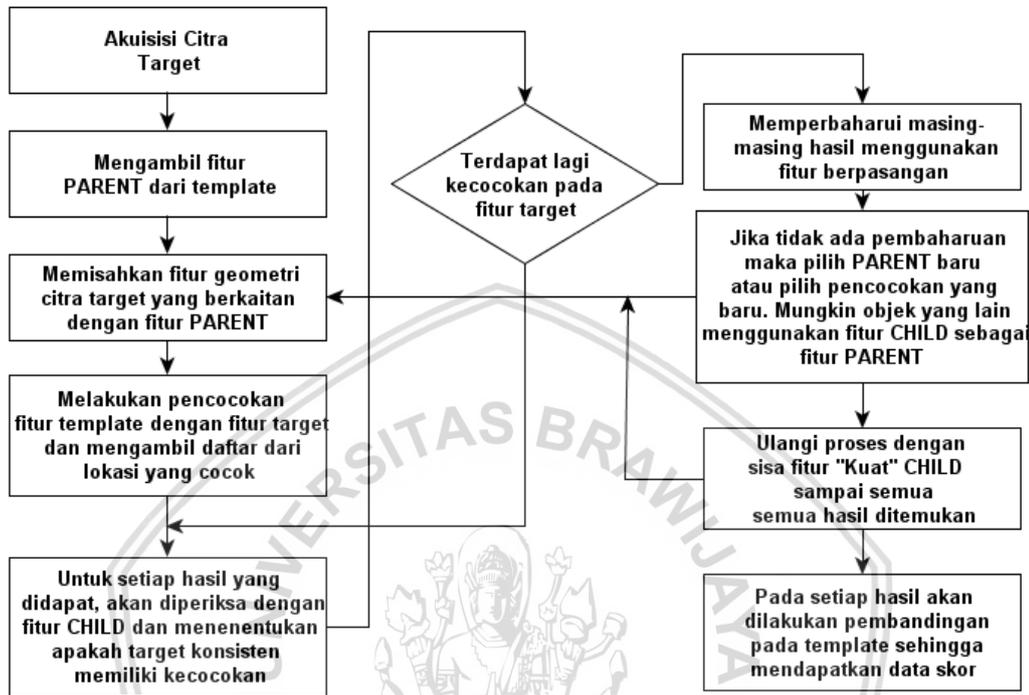
Gambar 2.2. Fitur Geomatic Match Pattern

Fitur merupakan ciri yang unik dan berkarakteristik pada template. Contoh fitur dapat dilihat pada gambar 2.2. Untuk mendapat fitur yang akan diekstraksi dari *template* maka diperlukan algoritma tersendiri. Pada *learning phase* atau fase belajar *template* yang akan ekstraksi fitur yaitu fitur *parent* dan fitur *child*. Fitur *parent* adalah fitur yang berada pada puncak pohon relasi dan fitur *child* merupakan fitur yang di bawah dari fitur *parent*. Untuk lebih jelas, algoritma akan diuraikan pada gambar 2.3.



Gambar 2.3. Algoritma Learning Phase

Setelah dilakukan fase belajar pada *template*, akan dilanjutkan dengan fase pencocokan atau *matching phase*. Pada fase ini akan dijelaskan secara runtut menggunakan algoritma *matching phase* yang diuraikan pada gambar 2.4.



Gambar 2.4. Algoritma Matching Phase

2.2.4 Ketetapan Pola dan Acak

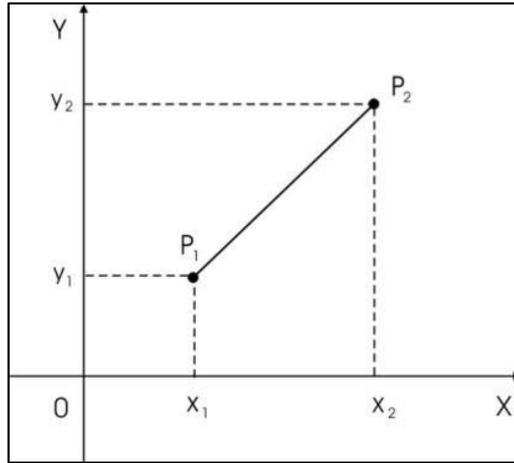
Pola adalah bentuk atau model (lebih abstrak atau suatu set peraturan) yang bisa dipakai untuk membuat atau untuk menghasilkan suatu atau bagian dari sesuatu, khususnya jika sesuatu yang ditimbulkan cukup mempunyai suatu yang sejenis untuk pola dasar yang dapat ditunjukkan atau terlihat, yang mana sesuatu itu dikatakan memamerkan pola. Deteksi pola dasar disebut pengenalan pola.

Definisi kata acak menurut kamus besar bahasa Indonesia ialah tanpa pola atau sebarang. Acak juga bisa di artikan sebagai awur, berantakan, semrawut atau kacau. Pada ilmu lingkungan yang mempelajari tentang populasi. Kata acak bisa di artikan penggambaran suatu pemilihan yg tidak dibatasi atau kalau dibatasi haruslah diwujudkan dengan menggunakan pemilihan peluang.

2.2.5 Jarak dan Trigonometri

Jarak merupakan angka yang menunjukkan selisih antara dua posisi dalam satuan panjang. Jarak biasanya dapat dikatakan besaran vektor ataupun skalar. Vektor adalah besaran-besaran yang memiliki besar (magnitudo) dan arah dan memenuhi aturan-aturan penjumlahan tertentu. Sedangkan besaran yang dinyatakan secara tepat hanya oleh sebuah bilangan dan satuannya disebut skalar. Ilustrasi jarak dapat dilihat pada gambar 2.5.





Gambar 2.5. Jarak antara dua posisi

Dari gambar 2.5 didapatkan beberapa persamaan sebagai berikut :

$$s = P2 - P1 \tag{2.2}$$

Dimana $P1 = (x1, y1)$

$P2 = (x2, y2)$

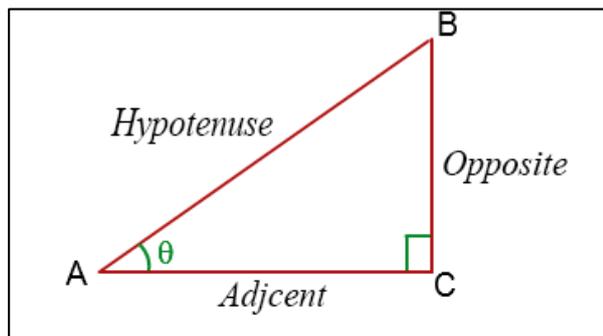
Dengan diketahuinya nilai $x1, y1$ dan $x2, y2$ maka untuk mendapatkan s yaitu jarak digunakan persamaan sebagai berikut :

$$s = \sqrt{\Delta x^2 + \Delta y^2} \tag{2.3}$$

Dimana $\Delta x = x2 - x1$

$\Delta y = y2 - y1$

Trigonometri merupakan ilmu matematika yang mempelajari hubungan antara sisi dan sudut dalam segitiga. Fungsi trigonometri yang terdiri atas sin, cos, tan, cosec, sec, dan cotan dapat digunakan untuk menentukan sisi sebuah segitiga atau sudut yang terbentuk dari dua buah sisi dalam sebuah segitiga. Pada sebuah segitiga terdapat tiga sisi yaitu *hypotenuse*, *opposite* dan *adjent*. *Hypotenuse* biasa disebut sisi miring, *Opposite* biasa disebut sisi depan dan *Adjcent* biasa disebut sisi samping. Gambar segitiga dapat dilihat pada gambar 2.6



Gambar 2.6. Segitiga Trigonometri



Dari gambar 2.6 segitiga tersebut didapatkan beberapa persamaan sebagai berikut

$$\sin \theta = \frac{\textit{Opposite}}{\textit{Hypotenuse}} \tag{2.4}$$

$$\cos \theta = \frac{\textit{Adjcent}}{\textit{Hypotenuse}} \tag{2.5}$$

$$\tan \theta = \frac{\textit{Opposite}}{\textit{Adjcent}} \tag{2.6}$$

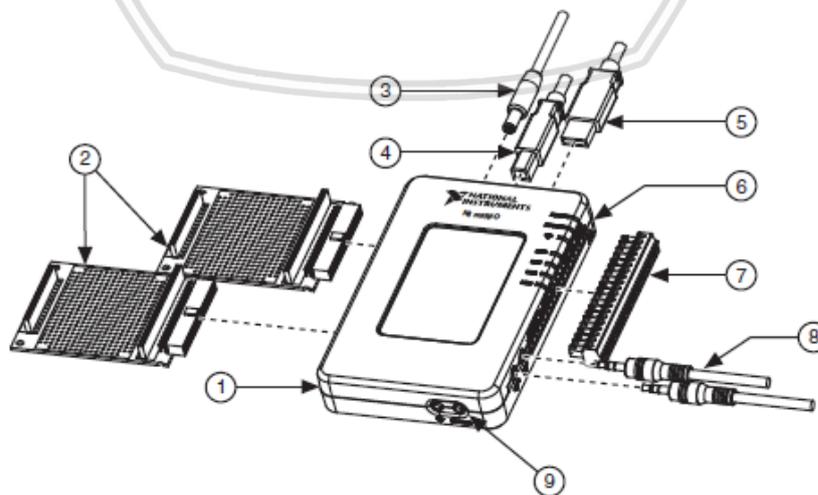
2.2.6 NI MYRIO-1900

NI MyRIO-1900 adalah mikrokontroler dari National Instrument. Dapat menggunakannya untuk merancang sistem kontrol, robot, dan mekatronika. MyRIO memiliki 2 prosesor utama yaitu LabVIEW RT (*Real Time*) dan FPGA. Bentuk fisik NI MyRIO-1900 dapat dilihat pada gambar 2.2.



Gambar 2.7. NI MyRIO-1900

sumber : <http://www.ni.com/en-id/support/model.myrio-1900.html>



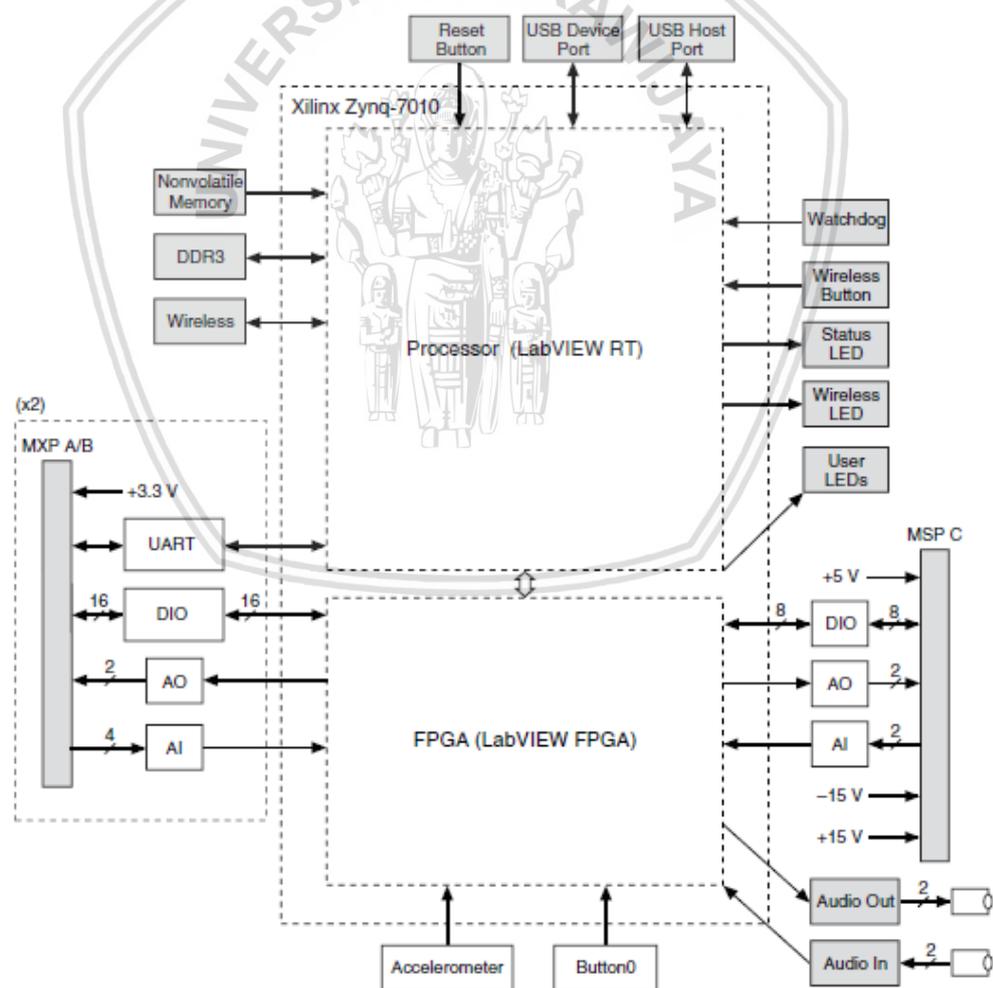
Gambar 2.8. Detail MyRIO 1900

sumber : <http://www.ni.com/en-id/support/model.myrio-1900.html>



Adapun penjelasan perangkat pada gambar 2.6 dijelaskan pada poin-poin berikut:

1. NI MyRIO 1900
2. *MyRIO Expansion Port (MXP)* merupakan pin pin berisi analog output.
3. *Power Input Cable* sebagai masukan kabel power.
4. *USB Device Cable* sebagai port USB menghubungkan MyRIO dengan komputer.
5. *USB Host Cable* sebagai port USB yang bisa digunakan sebagai memori maupun sebagai masukan sensor.
6. LEDs sebagai sinyal keluaran yang dapat di program.
7. Mini System Port (MSP) berisi pin pin analog dan digital.
8. *Audio In/Out Cable* sebagai port audio.
9. *Button 0* yang *input boolean* yang dapat diberi nilai *true* atau *false*.



Gambar 2.9. Blok Diagram MyRIO 1900

sumber : <http://www.ni.com/en-id/support/model.myrio-1900.html>

Tabel 2.1. Spesifikasi NI MyRIO-1900

Processor	
Type	Xilinx Z-7010
Speed	667 MHz
Core	2
Memory	
Non Volatile Memory	512 MB
DDR3 Memory	256 MB
FPGA	
Type	Xilinx Z-7010
Wireless Characteristic	
Radio mode	IEEE 802.11 b,g,n
Frequency band	ISM 2.4 GHz
Channel width	20 MHz
USB Port	USB 2.0 Hi-Speed
Analog Input	
MXP connectors configuration	Four single-ended channels per connector
MSP connector configuration	Two differential channels
Audio input configuration	One stereo input consisting of two AC-coupled, single-ended channels
Analog Output	
MXP connectors configuration	Two single-ended channels per connector
MSP connector configuration	Two single-ended channels
Audio output configuration	One stereo output consisting of two AC-coupled, single-ended channels
Digital I/O	
MXP connectors configuration	2 ports of 16 DIO lines (one port per connector); one UART.RX and one UART.TX line per connector
MSP connector configuration	1 port of 8 DIO lines
Secondary digital function	SPI,PWM, Quadrature encoder, I2C
Power supply voltage range	6 to 16 VDC
Maximum power consumption	14 W
Weight	193 g (6.8 oz)



2.2.7 Webcam

Webcam atau kamera web adalah sebuah kamera video digital kecil yang dihubungkan ke komputer melalui *port* USB ataupun *port* COM. Webcam biasanya dipakai untuk melakukan *video call* atau melakukan dokumentasi penelitian.



Gambar 2.10. Webcam

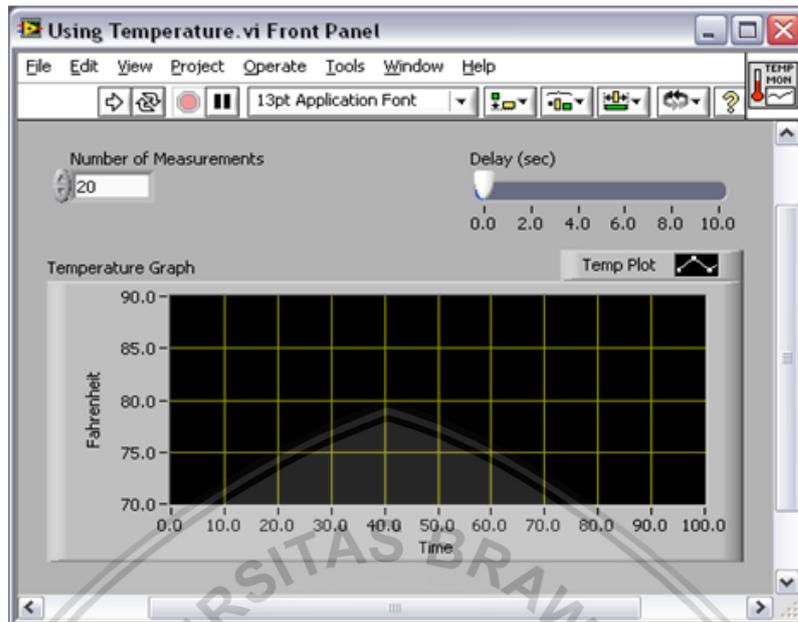
sumber : <https://www.amazon.co.uk/Sogatel-Diamond-Webcam-Windows-Compatible/dp/B003QPJ8JA>

Webcam adalah sebuah *web camera* yang sederhana terdiri dari sebuah lensa standar, dipasang di sebuah papan sirkuit untuk menangkap sinyal gambar, *casing* (*cover*), termasuk *casing* depan dan *casing* samping untuk menutupi lensa standar dan memiliki sebuah lubang lensa di *casing* depan yang berguna untuk memasukkan gambar; kabel *support*, yang dibuat dari bahan yang fleksibel, salah satu ujungnya dihubungkan dengan papan sirkuit dan ujung satu lagi memiliki *connector*, kabel ini dikontrol untuk menyesuaikan ketinggian, arah dan sudut pandang *web camera*. *Webcam* sendiri terdiri dari sensor gambar yaitu CMOS yaitu sensor yang mengubah cahaya optik menjadi sinyal elektrik. Sinyal elektrik kemudian di olah dalam semikonduktor IC yang melakukan proses *encoding* yaitu merubah data dalam sinyal ADC menjadi serial agar dapat di transfer menggunakan USB *port*.

2.2.8 NI LabVIEW Graphical Programming

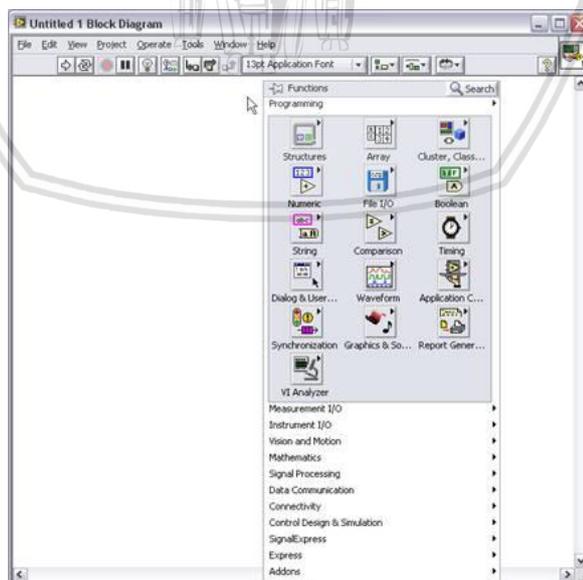
NI LabVIEW merupakan sebuah *software* dimana bahasa pemrogramannya yang digunakan berbasis grafik atau *block*, sehingga berbeda dengan pemrograman-pemrograman lainnya yang menggunakan teks. Program LabVIEW dikenal dengan sebutan VI atau *virtual instruments* karena penampilan dan operasinya dapat meniru sebuah instrument sebenarnya dalam sebuah simbol-simbol atau grafik. *Software* Labview terdiri dari empat komponen utama yaitu *front panel*, *block diagram*, *function palette* dan *control palette*. *Front panel*

merupakan tempat untuk membangun VI, menjalankan program dan melakukan *user control* atau kontrol pengguna. Front Panel dapat dilihat pada gambar 2.9.



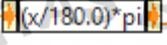
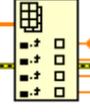
Gambar 2.11. Tampilan Front Panel LabVIEW
sumber : <http://www.ni.com/tutorial/7565/en/>

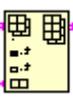
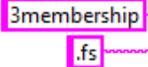
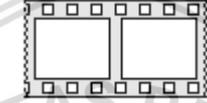
Selanjutnya pada LabVIEW terdapat *block diagram* yang digunakan untuk membangun program. Block diagram terdiri dari *background* putih dan terdapat *function palette* untuk melakukan coding program. *Block diagram* dapat dilihat pada gambar 2.10.



Gambar 2.12. Tampilan Block Diagram dan Palette LabVIEW
sumber : <http://www.ni.com/tutorial/7565/en/>

Tabel 2.2 Kode Program LabVIEW

Nama	Simbol	Fungsi
Array size		Mengembalikan jumlah element pada tiap dimensi <i>array</i>
Build Array		Membuat <i>array</i>
Boolean kontrol		Menngontrol dengan tipe data <i>boolean</i>
Build Path		Membaca <i>file</i> dari sebuah direktori
Bundle		Mengumpulkan data-data yang kemudian akan dieksekusi
Expression Node		Mengubah nilai x sesuai dengan fungsi yang dituliskan
Case Structure		Untuk membuat <i>case</i> yang kemudian dapat dipilih dengan selektornya yang berupa tanda tanya.
For Loop		Untuk membuat <i>looping</i> sesuai dengan banyak nilai i
Wait ms		Memberikan <i>Delay</i> untuk sistem
Tick count		Memberikan nilai <i>timer</i> saat itu
Delete from array		Menghapus <i>element</i> atau sub- <i>array</i> pada <i>array</i> .
Error		Menampilkan pesan jika terjadi <i>error</i>
Fomat into string		Mengubah tipe data ke dalam bentuk <i>string</i> (teks)
Decimal to String		Mengubah nilai <i>decimal</i> menjadi <i>string</i> (teks)
Index array		Mengembalikan <i>element</i> atau sub <i>array</i> pada dimensi <i>array</i> ke dalam suatu <i>index</i>

Indikator		Indikator suatu proses dengan keluaran angka dengan tipe data <i>double</i> .
Insert to array		Memasukan suatu <i>element</i> atau <i>sub array</i> ke dalam dimensi <i>array</i> pada <i>index</i> yang diinginkan
Path		Jalur atau alamat sebuah <i>file</i>
String		Konstanta untuk sebuah tipe data <i>string</i>
Sinus		Merubah x menjadi fungsi $f(\sin x)$
Sequence diagram		Membuat program berjalan secara berurutan
While loop		Untuk melakukan <i>looping</i> pada proses

	Scalar	1D Array	2D Array	Cluster
Numeric	Orange (floating point) Blue (integer)	Orange (floating point) Blue (integer)	Orange (floating point) Blue (integer)	Brown Brown
Boolean	Green	Green	Green	Pink
String	Pink	Pink	Pink	Pink
Path	Dark Green	Dark Green	Dark Green	Pink
Reference	Dark Green	Dark Green	Dark Green	Pink
Hardware Resource	Purple	Purple	Purple	Pink
Variant	Purple	Purple	Purple	Pink
Waveform	Brown	Brown	Brown	Pink
Class	Red	Red	Red	Pink

Gambar 2.13. Tipe Data Kabel

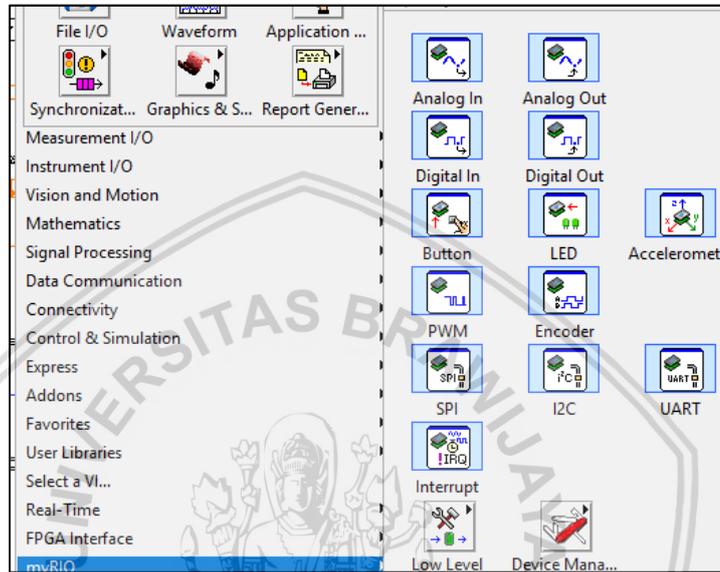
sumber : Nurmaica (2015)

Pada *dataflow programming* atau pemrograman berbasis aliran data di gunakan penghubung berupa kabel-kabel. Pada LabVIEW ada beberapa jenis tipe kabel untuk membedakan masing-masing fungsi dari kabel tersebut.



2.2.9 NI LabVIEW MyRIO Toolkit

Untuk melakukan pemrograman pada NI MyRIO diperlukan perangkat lunak LabVIEW MyRIO toolkit. Toolkit ini diperlukan untuk dapat mengenali MyRIO yang digunakan baik pin-pin, push button, LED dan USB port. Setelah menginstal toolkit ini maka secara otomatis pada block diagram LabVIEW juga akan muncul palatte khusus untuk memprogram MyRIO.



Gambar 2.14. MyRIO Palatte

2.2.10 NI LabVIEW Vision Acquisition Software

Pada Vision Acquisition Software (VAS) hanya berfokus kepada proses akuisisi citra sedangkan untuk melengkapi itu diperlukan Vision Development Module (VDM) untuk melakukan pemrosesan citra. Pada VAS terdapat beberapa tambahan kode program yang sebelumnya tidak ada pada program LabVIEW tanpa VAS. Kode program VAS yang digunakan dapat dilihat pada tabel 2.3.

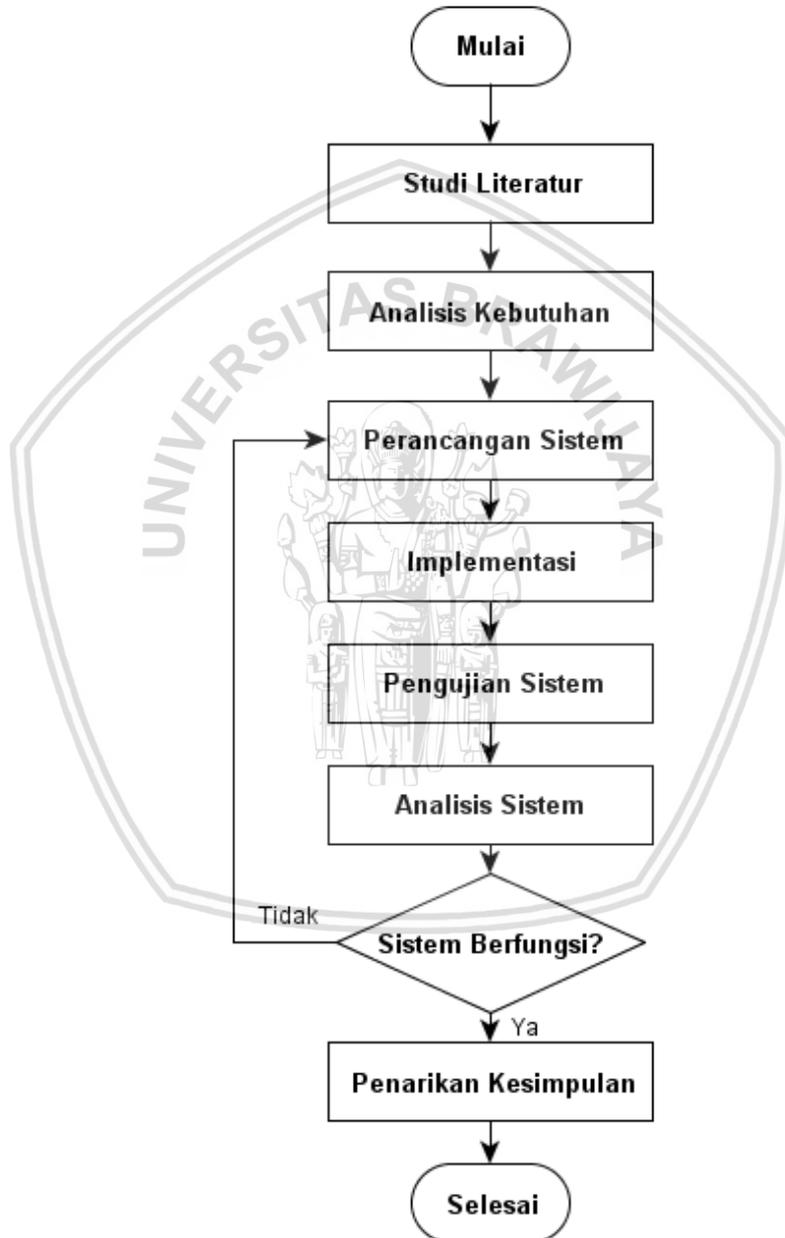
Tabel 2.3 Kode Program Vision Acquisition Software

Nama	Simbol	Fungsi
Open Camera		Memanggil fungsi kamera
Close Camera		Menutup fungsi kamera
Configure Grab		Melakukan konfigurasi pada kamera
Grab Image		Mengambil gambar dari kamera

Create Image		Memanggil alamat yang digunakan untuk menyimpan alamat gambar
Image Dispose		Menghapus gambar
Read Image Info		Mengambil gambar dari memori
Cast Image		Merubah nilai format dari gambar
Extract Single Colour Plane		Mengekstraksi satu jenis warna dari gambar
Learn Geomatric Pattern		Melakukan proses <i>learning</i> pada <i>template</i>
Setup Match Geomatric		Melakukan <i>setup</i> pada <i>geometric pattern</i>
Match Geomatric Pattern		Proses pencocokan <i>tempate</i> dengan target
Format Geomatric Pattern Result		Menampilkan hasil <i>geometric pattern</i> pada sebuah tabel
Overlay Geomatric Pattern Matching Result		Memberi <i>overlay</i> pada target yang sesuai dengan <i>template</i>
Overlay Text		Memberi <i>overlay</i> text atau <i>string</i> pada gambar
Overlay Line		Memberi <i>overlay</i> garis pada gambar

BAB 3 METODOLOGI

Penelitian ini dimulai dengan melakukan studi literatur yang terkait dengan landasan kepustakaan pada bab II. Penelitian yang dilakukan penulis bersifat pengembangan atau *development* pada ilmu terapan *Image Processing* pada platform MyRIO. Proses atau alur penelitian yang dilakukan untuk pembangunan sistem ini dapat dilihat pada diagram alir pada Gambar 3.1.



Gambar 3.1. Diagram Alir Metode Penelitian

3.1 Studi Literatur

Studi literatur adalah tahap dimana penulis mencari, mengumpulkan dan melakukan penyusunan teori yang akan digunakan sebagai rujukan, landasan dan pendukung penelitian. Pada penelitian ini akan menjelaskan metode, perangkat keras dan perangkat lunak. Metode yang berkaitan dengan penelitian ini meliputi:

1. NI MyRIO-1900
2. Kamera *webcam*
3. Citra Digital
4. Metode Geometric Match Pattern

3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan dilakukan dengan tujuan untuk menganalisis kebutuhan yang diperlukan oleh sistem agar sistem dapat berjalan sesuai tujuan. Dalam analisis kebutuhan terdapat beberapa sub bab analisis, diantaranya sebagai berikut.

3.2.1 Rekayasa Kebutuhan Perangkat Keras

Rekayasa kebutuhan perangkat keras yaitu rekaan kebutuhan perangkat keras yang terkait dengan kebutuhan dalam implementasi sistem dalam penelitian ini. Adapun perangkat keras yang digunakan sebagai penunjang penelitian ini antara lain :

1. *Notebook*
 - Model : Lenovo G-40
 - Prosesor : AMD APU A6-6310 1.8 GHz
 - Memori : 4 GB DDR3
2. NI MyRIO-1900
3. *Webcam*
4. Tiang Penyangga
5. Kabel
6. *Power supply*

3.2.2 Rekayasa Kebutuhan Perangkat Lunak

Rekayasa kebutuhan perangkat lunak yaitu rekaan kebutuhan perangkat lunak yang terkait dengan kebutuhan dalam implementasi sistem dalam penelitian ini. Adapun perangkat lunak yang digunakan sebagai penunjang penelitian ini antara lain :

1. Windows 10
2. NI Labview 2015 32 bit

3.2.3 Kebutuhan Sistem

Pada sistem ini ada beberapa kebutuhan yang harus terpenuhi agar sistem dapat berjalan sesuai tujuan penelitian. Kebutuhan sistem akan dijelaskan pada poin-poin di bawah ini.

1. Webcam tersambung dengan NI MyRIO-1900 pada port USB 2.0.
2. Alat Penyangga untuk menaruh webcam.

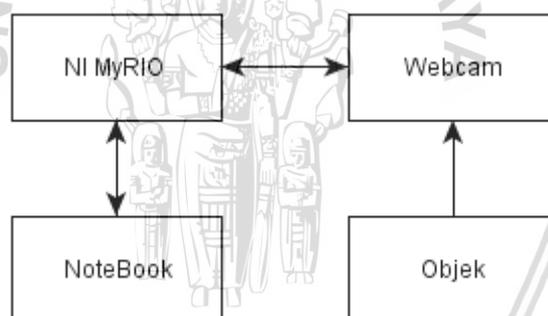
3.3 Perancangan Sistem

Perancangan sistem merupakan tahap dimana dilakukan perancangan awal dimana desain sistem dibuat. Tahap-tahap perancangan akan dijelaskan pada sub bab di bawah ini.

3.3.1 Perancangan Perangkat Keras

3.3.1.1 Diagram Blok

Pada perancangan perangkat keras, perancangan yang dibuat diperlihatkan pada gambar 3.2



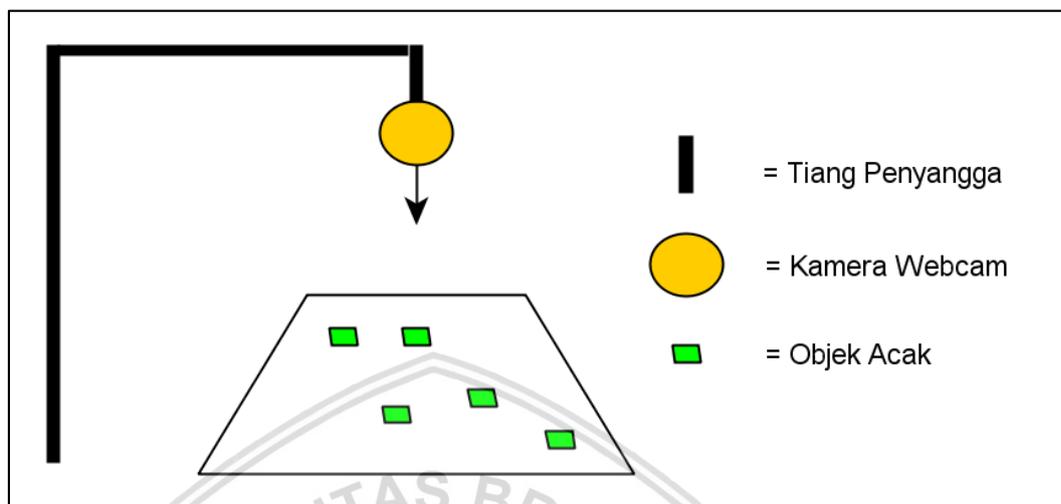
Gambar 3.2. Diagram Blok

Penjelasan mengenai desain perangkat keras pada Gambar 3.2. akan dijelaskan pada poin-poin di bawah ini :

- a. NI MyRIO berfungsi sebagai mikrokontroler yakni menerima data dari *webcam* berupa gambar
- b. *Notebook* disini sebagai tempat untuk menyiapkan *source code* kepada mikrokontroler dan sebagai Display
- c. *Webcam* digunakan sebagai sensor optik yang mengambil gambar dan mengirimkan ke mikrokontroler.

3.3.1.2 Alat Penyangga

Pada perancangan alat penyangga, perancangan yang dibuat diperlihatkan pada gambar 3.2



Gambar 3.3. Rangkaian Letak Sistem

1. Webcam di letakkan pada tiang penyangga. Tiang Penyangga akan menaruh webcam pada tengah bidang datar dan posisi webcam mengarah ke bawah
2. Bidang datar dengan *background* berwarna putih
3. Bahan uji berwarna kontras dengan background

3.3.2 Perancangan Perangkat Lunak

Perangkat Lunak yang digunakan ialah LabVIEW 2015 32 bit yang sudah menginstal *Vision Acquisition Software*. Program akan dirancang dengan menggunakan *Geometric Match Pattern* dan akan menghasilkan keluaran berupa manualisasi pada objek.

3.4 Implementasi

Implementasi merupakan tahap penelitian yang dilakukan setelah proses perancangan. Alur implementasi dimulai dari pembangunan rangkaian perangkat keras hingga tampilan keluaran pada *display* akan dijelaskan pada poin-poin di bawah ini:

1. Pada pembangunan rangkaian akan dilakukan seperti pada gambar 3.2 dan gambar 3.3.
2. Pengambilan dan pengolahan data akan dilakukan pada NI MyRIO-1900.
3. Pembuatan kode program akan dengan menggunakan NI LabVIEW 2015 32 bit.
4. Implementasi dilakukan menggabungkan rangkaian dengan kode program.

5. Hasil keluaran akan di tampilkan pada *display* pada laptop.

3.5 Pengujian Sistem

Pengujian sistem dilakukan untuk memenuhi kebutuhan sistem. Pengujian dilakukan dengan menguji fungsional sistem, menguji akurasi sistem, menguji keluaran manualisasi dan menguji waktu pemrosesan.

1. Pengujian fungsionalitas dari sistem
2. Pengujian akurasi pembacaan jarak, kemiringan dan selisih toleransi *pixel*
3. Pengujian waktu pemrosesan program

3.6 Analisis Pengujian

Analisis pengujian didasarkan pada kebutuhan sistem. Sehingga analisis dilakukan dengan cara sebagai berikut.

1. Analisis fungsionalitas dari sistem
2. Analisis akurasi pembacaan jarak, kemiringan dan selisih toleransi *pixel*
3. Analisis waktu pemrosesan program

3.7 Penarikan Kesimpulan

Penarikan kesimpulan dilakukan dengan mengambil hasil analisis pengujian. Pada kesimpulan akan menjawab rumusan masalah yang telah di rumuskan pada bab pendahuluan. Setelah kesimpulan akan diberikan saran-saran agar penelitian ini dapat berlanjut dan menjadi lebih baik pada penelitian selanjutnya.

BAB 4 REKAYASA KEBUTUHAN

Bab rekayasa kebutuhan berisi tentang penjelasan secara rinci tentang kebutuhan yang mesti dipenuhi untuk perancangan, implementasi hingga pengujian. Sehingga diharapkan sistem deteksi posisi objek acak berbasis *image processing* pada *platform* MyRIO dapat berjalan dengan baik.

4.1 Gambaran Umum Sistem

Sistem ini dapat dikatakan berfungsi dengan baik jika sistem dapat membaca data berupa gambar yang diterima melalui webcam kemudian diteruskan melalui NI MyRIO-1900, lalu kemudian data diolah dalam LabVIEW dan menghasilkan keluaran berupa koordinat objek beserta keacakannya. Sistem akan membaca objek dengan cara mencocokkan objek dengan *template* menggunakan *Geometric Match Pattern*. Proses akuisisi sampai proses penghitungan manualisasi akan di proses pada NI MyRIO kemudian hasil keluaran akan di tampilkan pada LabVIEW *Front Panel* di *Notebook*.

4.2 Tujuan Sistem

Tujuan dari pengembangan sistem ini adalah untuk mendapatkan data posisi dari citra objek menggunakan *Image Processing* dalam hal ini menggunakan metode *Geometric Match Pattern* kemudian akan menghasilkan keluaran berupa arahan manualisasi penyetaraan objek.

4.3 Analisis Kebutuhan Sistem

Kebutuhan sistem menjelaskan tentang kebutuhan-kebutuhan yang digunakan pada sistem yaitu kebutuhan pengguna, kebutuhan fungsional, kebutuhan perangkat keras, dan kebutuhan perangkat lunak.

4.3.1 Kebutuhan Pengguna

Terdapat beberapa kebutuhan yang dibutuhkan pengguna untuk menggunakan sistem yaitu:

1. Tombol untuk mengaktifkan sistem

Sistem akan mulai bekerja ketika pengguna menekan *run* pada *Front Panel*.

2. Notifikasi berupa lampu peringatan ketika terdapat objek acak.

Sistem akan mengeluarkan sebuah peringatan kepada pengguna sesaat ketika sistem mendeteksi sebuah objek yang posisinya tidak beraturan atau tidak pada semestinya.

3. Notifikasi berupa lampu peringatan ketika objek sudah tertata.

Sistem akan memberikan sebuah kode yang menandakan sudah tidak ada lagi objek acak atau objek sudah tertata pada tempat yang seharusnya.

4.3.2 Kebutuhan Fungsional

Agar sistem dapat berjalan sesuai dengan tujuan terdapat beberapa kebutuhan fungsional yaitu:

4.3.2.1 Fungsi mengakuisisi citra

Pada fungsi ini, MyRIO telah diprogram untuk dapat mengambil gambar yang diambil dari akuisisi kamera *webcam*. Program akan mengambil gambar pada saat program dijalankan.

4.3.2.2 Fungsi pengenalan objek

Sistem akan mengenali objek dengan mencocokkan citra objek dan citra template. Pencocokan menggunakan metode *geometric match pattern*. Objek yang dikenali harus benar-benar objek yang sesuai dan diharapkan. Objek kemudian akan di berikan *overlay* agar pengguna dapat melihat apakah fungsi ini berjalan dengan benar.

4.3.2.3 Fungsi menampilkan koordinat objek

Objek yang sudah dikenali sebagai objek akan di ubah menjadi data koordinat yang kemudian akan di simpan terlebih dahulu oleh sistem. Data akan di tunjukkan dalam bentuk tabel. Posisi yang diambil merupakan posisi dalam koordinat x dan y *pixel*.

4.3.2.4 Fungsi penghitungan posisi dan kemiringan

Sistem akan melakukan perhitungan baik dari posisi objek terhadap objek lainnya. Objek akan disetarakan menurut sumbu x atau sumbu y sesuai dengan pilihan pengguna. Sistem dapat mengenali perubahan kemiringan dari objek.

4.3.2.5 Fungsi keluaran penyeteraan posisi

Pada fungsi ini, sistem akan memberikan keluaran berupa perintah menggeser secara manual. Keluaran perintah akan diberikan dalam besaran centimeter. Perintah penggeseran mengikuti objek inti dan sumbu penggeseran sesuai dengan pilihan pengguna.

4.3.3 Kebutuhan Non-Fungsional

Agar sistem dapat berjalan dengan baik sesuai dengan tujuan terdapat beberapa kebutuhan non-fungsional yaitu:

4.3.3.1 Kebutuhan Perangkat Keras

Beberapa kebutuhan perangkat keras yang digunakan sistem yaitu:

1. NI MyRIO-1900

NI MyRIO merupakan sebuah mikrokontroler keluaran National Instrument yang memiliki banyak daya guna baik dalam akuisisi data maupun pengolahan data. Dalam hal ini NI MyRIO akan menjadi prosesor untuk mengambil dan mengolah data.

2. Webcam

Merupakan sensor optik dalam penelitian ini yang memiliki kompatibilitas dengan MyRIO-1900 dalam hal ini menggunakan Havic V-622 dengan resolusi 8 megapixel.

4.3.3.2 Kebutuhan Perangkat Lunak

Terdapat beberapa kebutuhan perangkat lunak yang akan digunakan pada sistem yaitu :

1. NI LabVIEW 2015 32 bit

NI Labview adalah aplikasi yang digunakan untuk memprogram NI MyRIO-1900. NI Labview sendiri merupakan aplikasi yang menggunakan model *dataflow programming* dan menggunakan bahasa pemrograman grafis atau diagram untuk mengontrol aliran data.

2. Library "Vision Acquisition Software"

Pada NI LabVIEW sendiri mempunyai *library* yang memiliki fungsi untuk melakukan pengambilan data berupa gambar. Dalam hal ini yang akan digunakan ialah NI-IMAQ.

4.4 Batasan Perancangan Sistem

Agar penelitian ini dapat berjalan sesuai harapan dan lebih terarah terdapat Batasan untuk merancang sistem yaitu:

1. Sensor yang digunakan ialah sensor optik atau kamera *webcam* berupa Havic V-622.
2. Sensitivitas sensor menggunakan pengaturan *default*.
3. Mikrokontroler yang digunakan adalah NI MyRIO-1900.
4. Aplikasi yang digunakan ialah NI LabVIEW 2015 32-bit.
5. Alat Penyangga memiliki tinggi yang tetap.
6. Bahan uji sebagai objek ialah penghapus berwarna hitam dengan ukuran 3.15 x 14.7 cm.
7. Latar *background* berupa kertas berwarna putih.
8. Keluaran dari sistem hanya akan berupa perintah manual berdasarkan koordinat x dan y .

4.5 Asumsi dan Ketergantungan

Agar sistem dapat berjalan dengan tujuan yang diharapkan terdapat asumsi dan ketergantungan pada sistem ini yaitu :

1. Sudut pengambilan data gambar merupakan 90° tegak lurus terhadap titik tengah dari *background*.
2. Cahaya untuk gambar berdasarkan cahaya ruangan (500-750 lux).



BAB 5 PERANCANGAN DAN IMPLEMENTASI

Pada bab ini akan dijelaskan bagaimana perancangan sistem dan implementasi sistem sesuai dengan perancangan sistem yang telah di buat sebelumnya.

5.1 Perancangan Sistem

Pada tahapan perancangan sistem ini akan ditentukan perilaku sistem, cara kerja sistem, perangkat keras, serta perangkat lunak pendukung dan penyusun sistem.

5.1.1 Perancangan Perangkat Keras

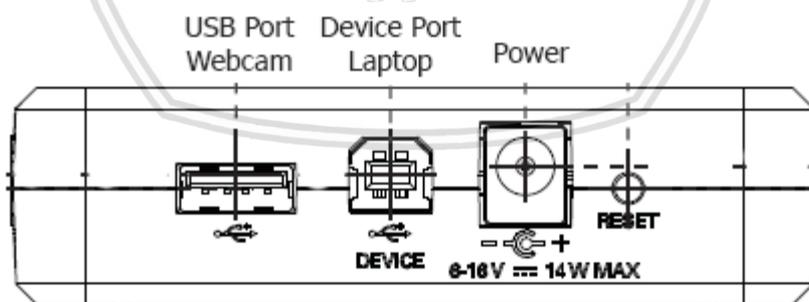
Pada tahapan perancangan perangkat keras ini, dilakukan pembuatan sistem, dengan perancangan yang telah ditentukan sebelumnya.



Gambar 5.1. Diagram Blok Aliran Data

Pada gambar 5.1 di gambarkan tentang diagram blok aliran data, data disini yang dimaksudkan ialah citra yang di ambil melalui webcam kemudian di sambungkan kepada mikrokontroler NI MyRIO-1900.

1. Pada NI MyRIO akan dilakukan pemrosesan citra yang kemudian hasil citra akan lanjut ditampilkan pada Laptop tepat pada aplikasi LabView.
2. Pada laptop akan dilakukan penampilan hasil pemrosesan yang telah dilakukan NI MyRIO.



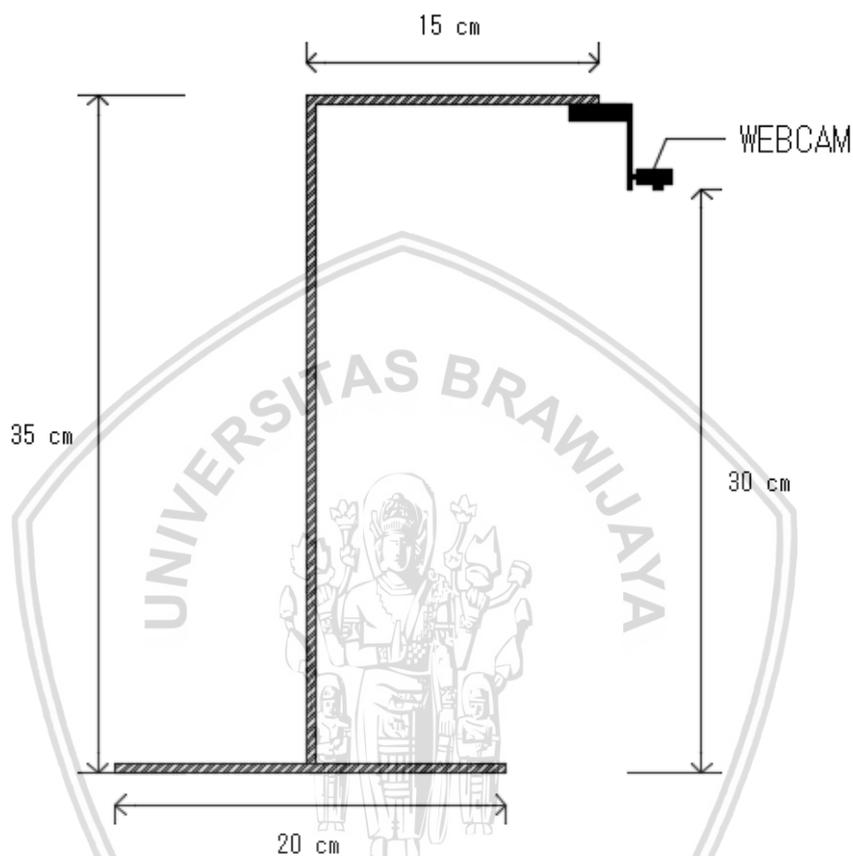
Gambar 5.2. MyRIO Tampak Depan

Pada gambar 5.2. digambarkan tentang MyRIO tampak depan yaitu soket yang akan dihubungkan dengan beberapa bagian seperti berikut:

1. *USB Port* yang akan di hubungkan dengan USB dari kamera *webcam*.
2. *Device Port* yang akan dihubungkan dengan USB dari Laptop menggunakan kabel *device port* itu sendiri
3. *Power* yaitu soket untuk menghubungkan MyRIO dengan sumber daya

5.1.1.1 Perancangan Alat Penyangga

Pada perancangan alat penyangga ini akan dirancang seperti pada gambar 5.3. Alat penyangga ini merupakan *prototype* yang digunakan sebagai penyangga kamera *webcam*. Alat penyangga akan dibuat dari bahan dasar akrilik dengan tebal 0.5cm. Alat penyangga akan membuat kamera berada pada ketinggian 30cm di atas latar *background*.



Gambar 5.3. Rancangan Tiang Penyangga

5.1.1.2 Perancangan Objek

Objek uji yang diakuisisi dan dilakukan pemrosesan citra pada penelitian ini ialah penghapus berukuran 3.15 x 14.7 cm. Objek uji berupa penghapus diambil dengan alasan objek ini memiliki 4 sudut yang dapat digunakan sebagai fitur. Objek ini dapat mewakili beberapa bentuk dari kotak kardus pada pabrik. Objek berbentuk persegi panjang juga dapat dibedakan orientasi kemiringannya. Kemiringan dinilai penting untuk melihat besar rotasi yang dapat dilakukan sebagai keluaran.

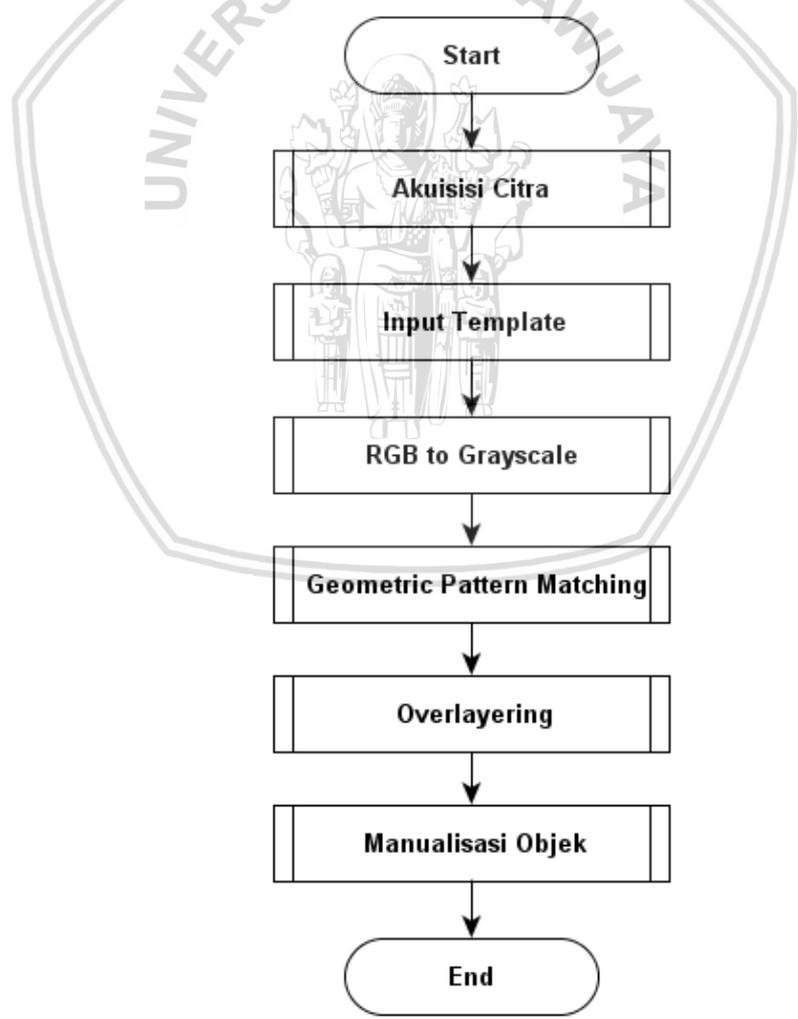
5.1.2 Perancangan Perangkat Lunak

Perancangan perangkat lunak dilakukan pada computer namun di eksekusi oleh mikrokontroler. Hasil program akan tetap ditampilkan pada *Front Panel* yang terdapat di computer karena pada NI MyRIO tidak memiliki *Front Panel*. Untuk penjelasan akan di tampilkan pada gambar 5.4.



Gambar 5.4. Diagram Blok Proses

Pada perancangan perangkat lunak dilakukan dengan tahap-tahap pemrosesan citra atau *image processing* yang diterapkan menggunakan bahasa pemrograman berbasis objek pada aplikasi LabView. Untuk mempermudah implementasi maka perancangan *software* pada aplikasi LabView dibagi menjadi, perancangan akuisisi citra, *input template* atau memasukkan citra *template*, RGB to Grayscale, *Geometric Pattern Matching* atau pencocokan pola secara geometri, *Overlaying* atau pemberian *overlay* pada objek, dan yang terakhir keluaran yaitu manualisasi pada objek. Diagram alir perancangan perangkat lunak dapat dilihat pada gambar 5.5.



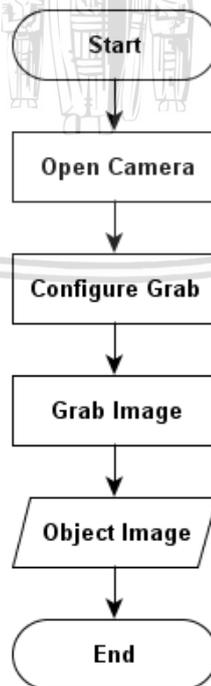
Gambar 5.5. Diagram Alir Perancangan Perangkat Lunak

Penjelasan dari gambar 5.5. akan di jelaskan pada poin poin berikut di bawah ini:

- a. Akuisisi Citra : pada proses ini akan dilakukan pengambilan citra objek secara otomatis oleh kamera *webcam* yang kemudian akan langsung di terima oleh program.
- b. *Input Template* : pada proses ini akan dilakukan pengambilan citra *template* yang telah di simpan di dalam memori MyRIO 1900
- c. *RGB to Grayscale* : pada proses ini akan dilakukan perubahan format citra objek yang sebelumnya RGB menjadi *Grayscale*. Perubahan ini dilakukan untuk memudahkan pencocokan oleh *Geometric Match Pattern*.
- d. *Geometric pattern matching* : pada proses ini akan dilakukan pencocokan antara citra objek dengan citra *template* dan akan menghasilkan keluaran berupa data objek yaitu koordinat, kemiringan dan oklusi.
- e. *Overlaying* : pada proses ini akan dilakukan penambahan *overlay* pada citra objek untuk memberikan penomoran pada objek dan memberikan informasi garis dan jarak objek terhadap objek lainnya
- f. Manualisasi Objek : pada proses ini akan dilakukan perhitungan untuk memberikan keluaran berupa manualisasi untuk menggeser objek.

5.1.2.1 Perancangan Akuisisi Citra

Pada perancangan akuisisi citra akan di gambarkan dalam diagram alir akuisisi citra. Diagram alir akuisisi citra dapat dilihat pada gambar 5.6.



Gambar 5.6. Diagram Alir Akuisisi Citra

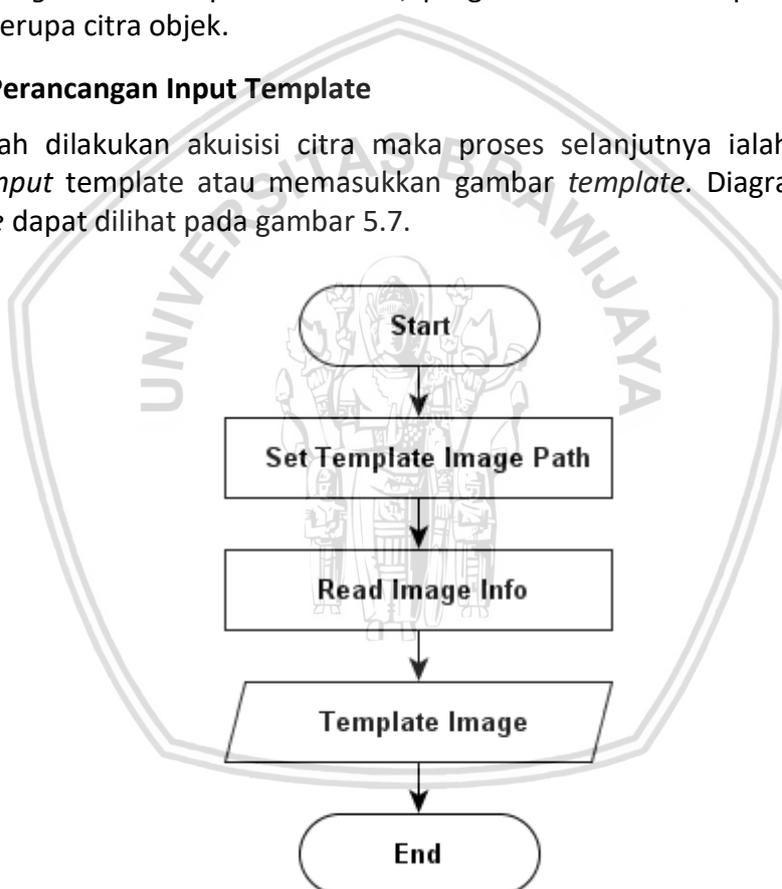


Untuk akuisisi citra sendiri diperlukan diagram alir proses seperti diperlihatkan pada gambar 5.6. Untuk setiap proses akan dijelaskan pada poin poin berikut ini :

1. *Open Camera* : pada proses ini, kamera dipanggil beserta *configuration file* kamera kemudian akan diberikan referensi khusus, selanjutnya juga akan di panggil memori sementara untuk gambar pada proses ini
2. *Configure Grab* : pada proses ini dilakukan konfigurasi mengenai tipe akuisisi apakah dilakukan secara kontinyu atau hanya sekali.
3. *Grab Image* : pada proses ini dimulai akuisisi citra yang telah di konfigurasi sebelumnya menghasilkan citra yang dalam bentuk *video stream*
4. *Get Image* : pada proses ini citra sudah di dapatkan
5. *Image* : setelah proses di atas, program sudah mendapatkan keluaran berupa citra objek.

5.1.2.2 Perancangan Input Template

Setelah dilakukan akuisisi citra maka proses selanjutnya ialah melakukan proses *input template* atau memasukkan gambar *template*. Diagram alir *input template* dapat dilihat pada gambar 5.7.



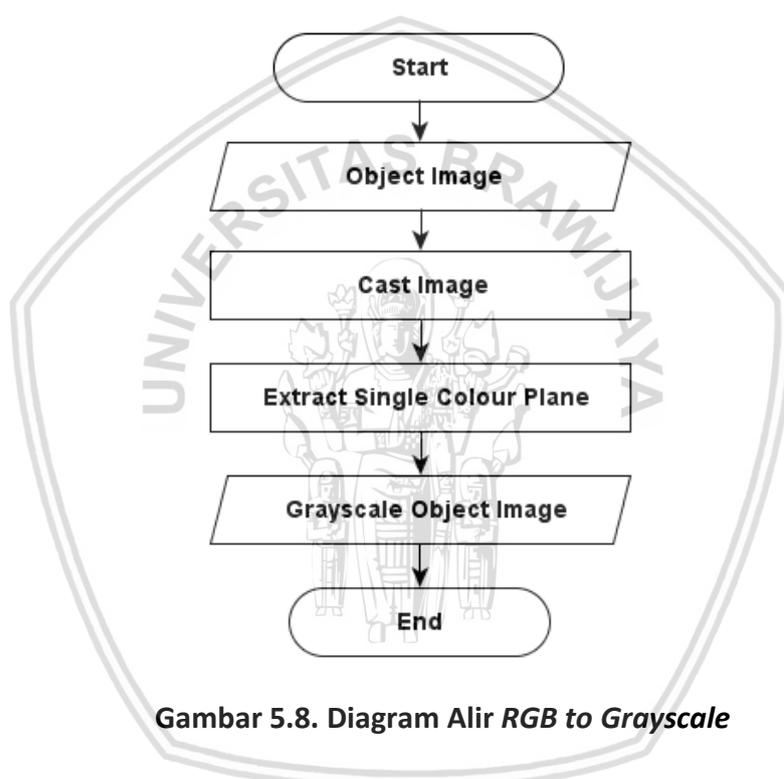
Gambar 5.7. Diagram Alir Input *Template*

Pada proses ini, program akan mengambil data berupa gambar dalam bentuk PNG yang telah di masukkan ke dalam memori MyRIO. Penelitian ini menggunakan bahan uji yang berbentuk persegi panjang oleh sebab itu pada diagram alir pada gambar 5.7. Proses yang ada pada diagram alir di atas akan di jelaskan pada poin poin berikut ini:

1. *Set Template Image Path* : yaitu memberikan alamat citra yang telah di simpan dalam memori MyRIO 1900.
2. *Read Image Info* : pada proses ini, program akan melakukan pembacaan informasi dari citra yang telah di dapat dari MyRIO 1900.
3. *Template Image* : setelah proses di atas, program telah mendapatkan keluaran berupa citra yang digunakan sebagai *template*.

5.1.2.3 Perancangan *RGB to Grayscale*

Pada proses perancangan *RGB to Grayscale* digunakan untuk mengubah citra objek dari *RGB* menjadi *grayscale*. Proses ini dilakukan untuk memudahkan proses pencocokan pola pada proses selanjutnya. Diagram alir *RGB to Grayscale* dapat dilihat pada gambar 5.8.



Gambar 5.8. Diagram Alir *RGB to Grayscale*

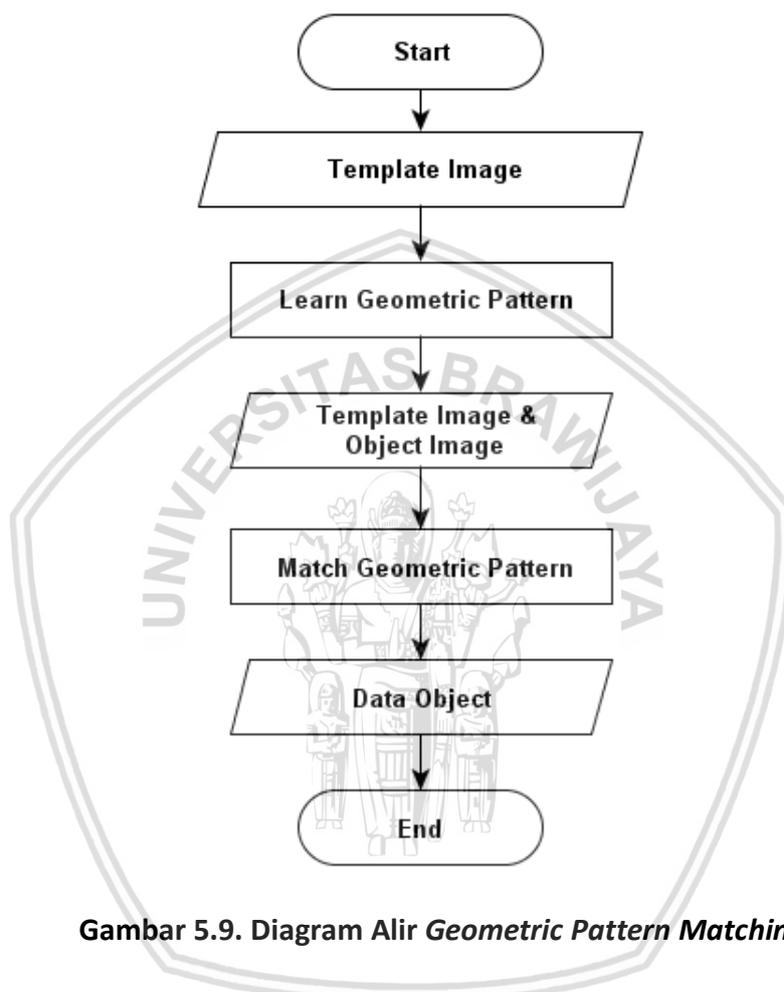
Pada proses ini, program akan mengubah format citra dari *RGB* menjadi *Grayscale* dengan cara melakukan filter dari intensitas citra. Diagram alir di atas akan dijelaskan pada poin-poin berikut :

1. *Object Image* : citra dari objek yang sudah di akuisisi akan digunakan sebagai masukan pada proses ini.
2. *Cast Image* : pada proses ini akan dilakukan perubahan format gambar dari yang sebelumnya berupa video akan di ubah menjadi gambar dengan format RGB 32 bit.
3. *Extract Single Color Plane* : pada proses ini akan dilakukan ekstraksi gambar atau melakukan *filtering*. Untuk mendapatkan objek *grayscale* maka pilihan yang akan di ekstrak adalah intensitas.

4. *Grayscale Object Image* : Setelah proses di atas maka akan di dapatkan sebuah keluaran citra yang sudah menjadi *grayscale*.

5.1.2.4 Perancangan *Geometric Pattern Matching*

Pada proses *Geometric pattern matching* akan dilakukan pencocokan antara citra objek dengan citra template.



Gambar 5.9. Diagram Alir *Geometric Pattern Matching*

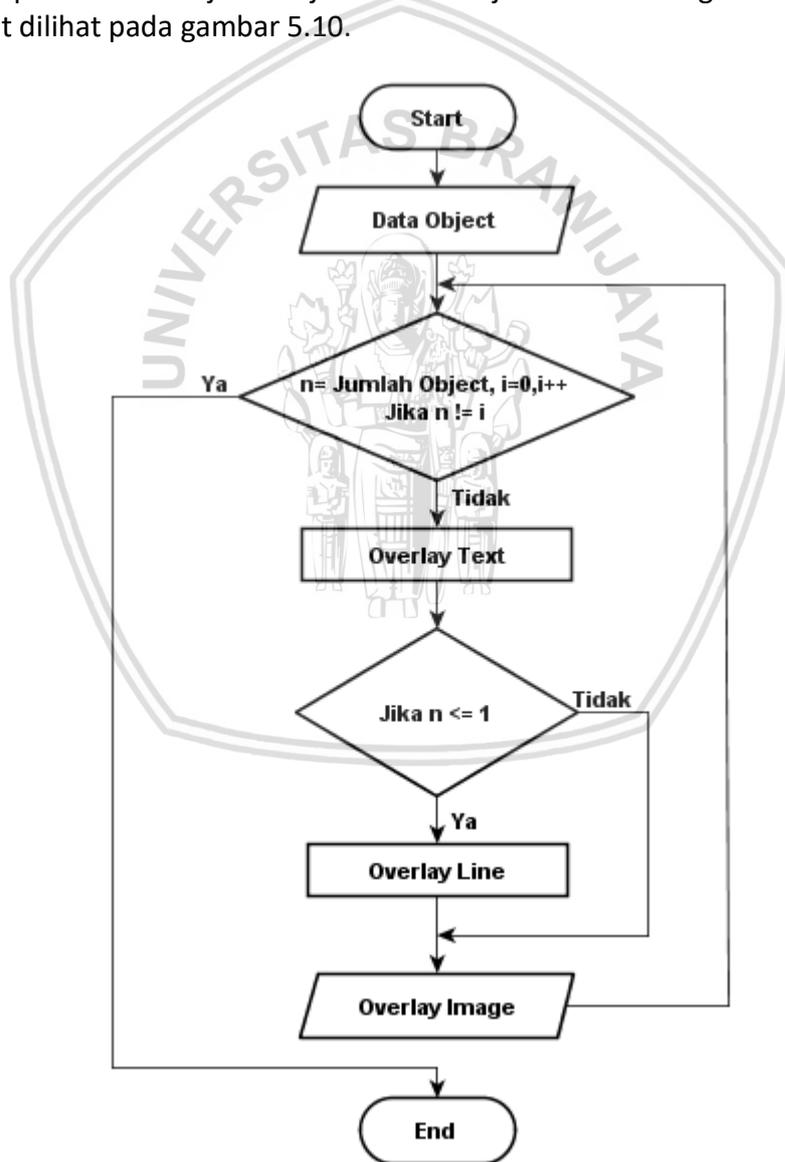
Pada akhir proses ini akan dilakukan pencocokan citra objek dengan citra template menggunakan *Match Geometric Pattern*. Diagram alir di atas akan di jelaskan pada poin poin berikut:

1. *Template Image* : masukan dari proses ini ialah citra objek yang sebelumnya sudah di ubah menjadi format *grayscale* dan citra template yang telah di dapatkan dari memori MyRIO 1900.
2. *Learn Geometric Pattern* : pada proses ini, citra template akan di dilakukan proses *Learning* terlebih dahulu, untuk menentukan apakah citra dapat menjadi *template* pada *Matching Geometric* atau tidak.
3. *Template Image* dan *Object Image* : citra template yang telah di lakukan proses *Learning* dan citra objek dari hasil *RGB to Grayscale* akan menuju pada *Match Geometric Pattern*.

4. *Match Geometric Pattern* : pada proses ini, citra objek dan citra template akan dilakukan pencocokan dengan melakukan *Setup Match Geometric Pattern*. *Setup* yang dilakukan ialah Rotasi, Skala dan Oklusi. Setelah *setup*, akan dilakukan proses *matching*.
5. *Data Object* : Setelah proses di atas, kita akan mendapatkan keluaran berupa data objek dalam bentuk array yaitu koordinat objek, sudut pergeseran, tingkat kecocokan dan oklusi.

5.1.2.5 Perancangan *Overlaying*

Pada perancangan ini, citra akan diberikan *overlay* yang akan di munculkan pada *Front Panel*. *Overlay* yang akan dimunculkan yaitu kotak pembatas yang akan berisikan penomoran objek dan jarak antar objek dari titik tengah objek. Diagram alir dapat dilihat pada gambar 5.10.

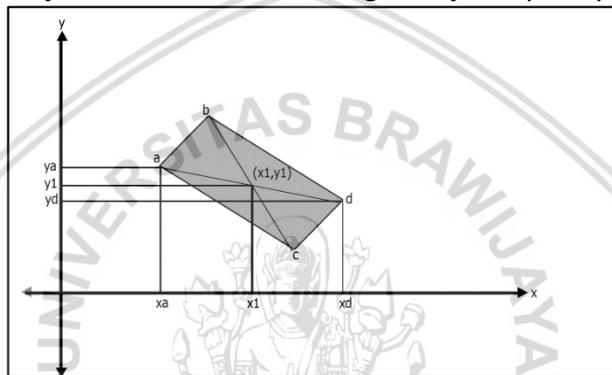


Gambar 5.10. Diagram Alir *Overlaying*



Pada proses ini akan dilakukan proses *overlay text* dan *overlay line*. Diagram alir di atas akan di jelaskan pada poin poin berikut:

1. *Data Object* : pada masukan, proses ini membutuhkan data objek yang sudah tersusun dalam bentuk *array*.
2. *For Loop* : pada *looping* ini akan dilakukan sebanyak jumlah objek. Setiap satu kali penambahan jumlah '*i*', akan dilakukan proses *Overlay Text* dan *Overlay Line*.
3. *Overlay Text* : pada proses ini akan dilakukan penomoran objek
4. *Overlay Line* : Jika objek yang di dapatkan kurang atau sama dengan 1 maka proses akan langsung berlanjut pada keluaran berupa *Overlay Image*, tapi jika objek yang ditemukan 2 atau lebih maka proses akan melakukan penghitungan jarak dari objek 1 ke pada objek lainnya. Titik posisi koordinat objek di ambil dari titik tengah objek seperti pada gambar 5.11.

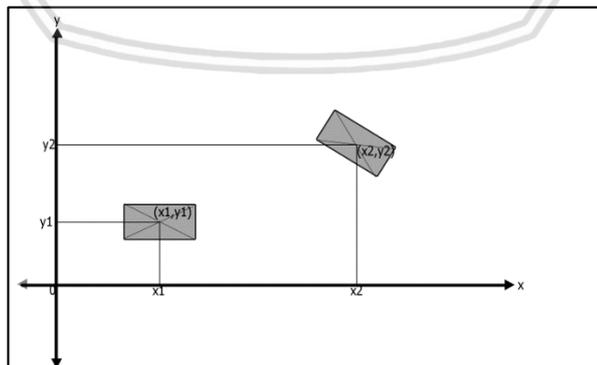


Gambar 5.11. Titik Tengah Objek

Titik tengah di dapatkan dengan persamaan sebagai berikut

$$(x1, y1) = \left(\frac{xd-xa}{2}, \frac{yd-ya}{2} \right) \tag{5.1}$$

Setelah mendapatkan titik tengah dari objek, maka dilanjutkan dengan menghitung jarak antar objek. Gambaran objek dapat dilihat pada gambar 5.12



Gambar 5.12. Posisi Koordinat Objek

Penghitungan jarak akan di lakukan sebagai berikut :

$$\Delta x = x2 - x1 \tag{5.2}$$

$$\Delta y = y2 - y1 \tag{5.3}$$

Pada persamaan di atas dapat kita terapkan pada program yaitu,



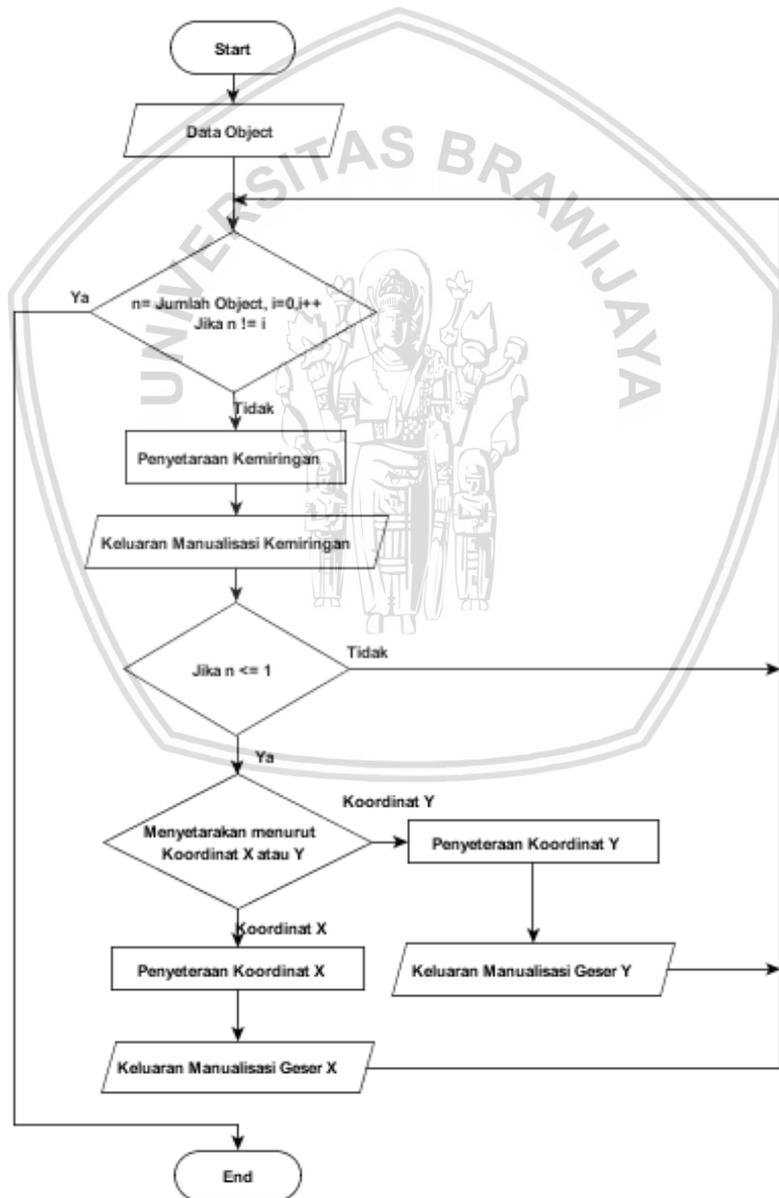
$$\begin{aligned}
 x. Objek [1] - x. Objek [i] &= \Delta x \\
 y. Objek [1] - y. Objek [i] &= \Delta y \\
 s &= \sqrt{\Delta x^2 + \Delta y^2}
 \end{aligned}
 \tag{5.4}$$

Selanjutnya s atau jarak akan ditampilkan pada titik pusat dari objek sebagai jarak dari objek.

5. *Overlay Image* : setelah proses di atas, keluaran yang di dapat ialah citra yang telah terdapat nomor objek dan garis antar objek beserta jaraknya.

5.1.2.6 Perancangan Manualisasi Objek

Pada perancangan ini, akan dilakukan perhitungan untuk menghitung jarak untuk menggeser objek dan untuk membuat objek kembali pada kondisi normal dari kemiringan. Diagram alir manualisasi objek dapat dilihat pada gambar 5.12



Gambar 5.13. Diagram Alir Manualisasi Objek



Pada proses ini akan dilakukan manualisasi dan penghitungan. Diagram alir di atas akan di jelaskan pada poin-poin berikut:

1. *Data Object* : pada masukan, proses ini membutuhkan data objek yang sudah tersusun dalam bentuk *array*.
2. *For Loop* : pada *looping* ini akan dilakukan sebanyak jumlah objek. Setiap satu kali penambahan jumlah '*i*', akan dilakukan penyetaraan kemiringan dan penyetaraan koordinat.
3. Penyetaraan Kemiringan : pada proses ini akan dilakukan penyetaraan kemiringan dengan cara mengambil data sudut dan kemudian akan dilakukan perhitungan dengan mengambil panjang dari objek.

Perhitungan jumlah pergeseran akan dilakukan sebagai berikut :

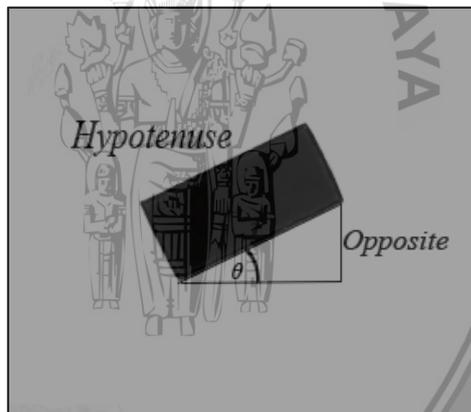
$$\theta = \frac{\alpha}{180.0} \times \pi \quad (5.5)$$

Dimana θ = sudut dalam radian

α = sudut dalam degree hasil pembacaan

π = phi

Setelah mendapatkan sudut dalam radian, perhitungan selanjutnya akan menentukan jumlah pergeseran. Kemiringan objek dapat dilihat pada gambar 5.14



Gambar 5.14. Kemiringan Objek

Pada gambar di atas, kita sudah mendapatkan besar θ dan panjang h atau hipotenus yang didapatkan dari panjang objek. Sehingga untuk mendapatkan O atau oposit akan dilakukan perhitungan sebagai berikut :

$$\text{Opposite} = \sin \theta \times \text{Hypotenuse} \quad (5.6)$$

4. Keluaran Penyetaraan Kemiringan : Keluaran disini ialah besar jarak objek yang akan di geser untuk membuat objek normal.
5. Penyetaraan Koordinat : Jika objek yang di dapatkan kurang atau sama dengan 1 maka proses akan langsung berhenti, tapi jika objek yang ditemukan 2 atau lebih maka proses akan melakukan penghitungan jarak koordinat x atau koordinat y dari objek 1 ke pada objek lainnya. Pilihan akan berlanjut untuk memilih penyetaraan koordinat x atau koordinat y .
6. Penyetaraan Koordinat x : pada proses ini, akan dilakukan perhitungan sebagai berikut :

$$s = \Delta x = x_2 - x_1 \tag{5.7}$$

Dimana $x_2 = x.Objek[1]$

$x_1 = x.Objek[i]$

7. Keluaran Manualisasi Koordinat : Setelah proses penyetaraan koordinat x maka akan menghasilkan keluaran berupa jarak dalam *pixel* dan dalam centimeter.

8. Penyetaraan Koordinat y : pada proses ini, akan dilakukan perhitungan sebagai berikut :

$$s = \Delta y = y_2 - y_1 \tag{5.8}$$

Dimana $y_2 = y.Objek[1]$

$y_1 = y.Objek[i]$

9. Keluaran Manualisasi Koordinat : Setelah proses penyetaraan koordinat y maka akan menghasilkan keluaran berupa jarak dalam *pixel* dan dalam centimeter.

5.2 Implementasi Sistem

Setelah dilakukan perancangan sistem, tahapan selanjutnya dalam pembuatan sistem ini adalah dengan melakukan implementasi berdasarkan spesifikasi dan perilaku sistem yang telah ditentukan sebelumnya.

5.2.1 Implementasi Perangkat Keras

Perancangan sistem ini dilakukan dengan merancang diagram alir akuisisi data dan perancangan rangkaian perangkat keras berupa NI MyRIO 1900, kamera *webcam* dan alat penyangga. Gambar implementasi perangkat keras dapat dilihat pada gambar 5.15.



Gambar 5.15. Implementasi Perangkat Keras

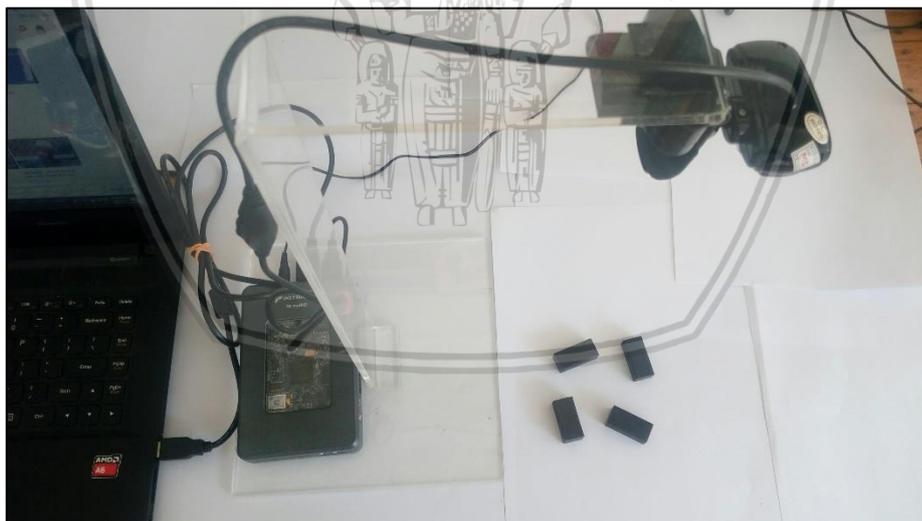
5.2.1.1 Implementasi Alat Penyangga

Pada implementasi alat penyangga, akan di implementasikan sesuai dengan rancangan sebelumnya. Alat penyangga berbahan dasar akrilik dengan tinggi 35 cm dan siku sepanjang 15 cm. Alat penyangga dapat dilihat pada gambar 5.16.



Gambar 5.16. Implementasi Alat Penyangga

Dengan di gabungkannya perangkat keras dan alat pendukung yaitu alat penyangga akan menghasilkan satu-kesatuan seperti gambar 5.17.



Gambar 5.17. Implementasi Sistem Perangkat Keras

5.2.2 Implementasi Perangkat Lunak

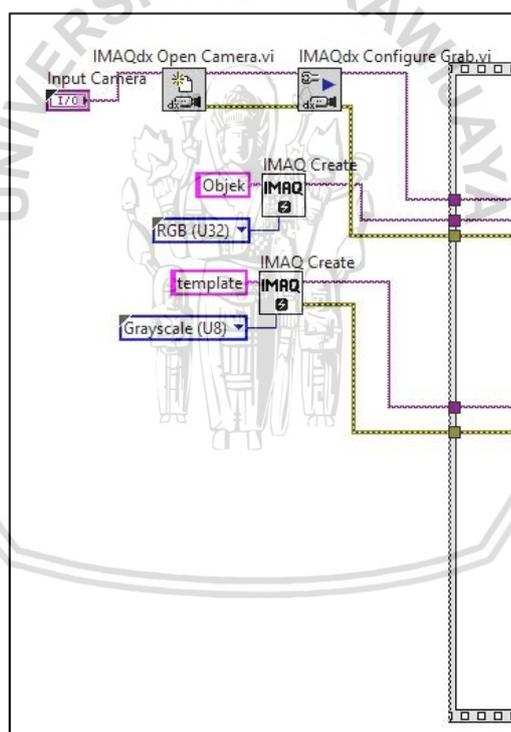
Pada tahapan implementasi perangkat lunak ini dilakukan implementasi sesuai dengan perancangan yang telah dilakukan sebelumnya.

5.2.2.1 Implementasi Akuisi Citra

Implementasi untuk mengakuisisi citra dilakukan dengan langkah sebagai berikut :

1. Memanggil *IMAQ Open Camera.vi* yang memiliki masukan berupa kamera yang di ditentukan pada *Front Panel*.
2. *IMAQdx Configure Grab.vi* akan melakukan konfigurasi yang melakukan *buffer* dan *looping* terhadap gambar yang di akuisisi oleh kamera.
3. *IMAQ Create* digunakan untuk memanggil memori yang akan digunakan untuk menyimpan data citra baik citra objek maupun citra *template*. Proses akan berlanjut dengan aliran data berupa ketersediaan memori citra objek dan citra template. Pada *IMAQ Create* objek menggunakan format RGB U32 yaitu format untuk menyimpan 32 bit gambar. Sedangkan pada *IMAQ Create* template menggunakan format *Grayscale U8* yaitu format untuk 8 bit gambar.

Program dapat dilihat pada gambar 5.18

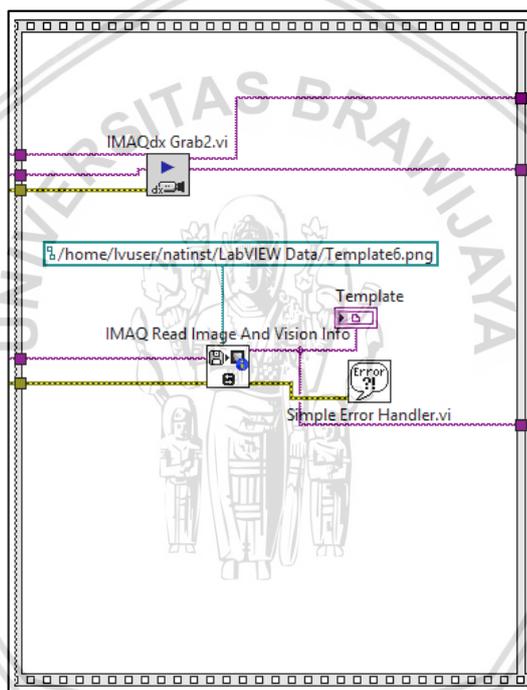


Gambar 5.18. Implementasi Akuisisi Citra

5.2.2.2 Implementasi Input Template

Implementasi untuk memasukkan citra template dilakukan dengan tahapan sebagai berikut:

1. Citra *template* sebelumnya telah tersimpan dalam memori internal dari MyRIO. Untuk dapat mengambil citra *template* di butuhkan *IMAQ Read and Vision Info.vi*
 2. Pada *path* akan diberikan nilai konstan yaitu alamat citra pada memori MyRIO yaitu `"/home/lvuser/natinst/LabVIEW Data/Template6.png"`. Alamat tersebut merupakan alamat yang biasa digunakan untuk menyimpan data pada MyRIO.
 3. Citra *template* kemudian ditampilkan pada *Front Panel* menggunakan *Image.ctl*.
 4. *IMAQdx Grab2.vi* mengambil *session* yang sudah di konfigurasi dan mengambil alamat yang kemudian akan menghasilkan citra objek
- Program dapat dilihat pada gambar 5.19



Gambar 5.19. Implementasi *Input Template*

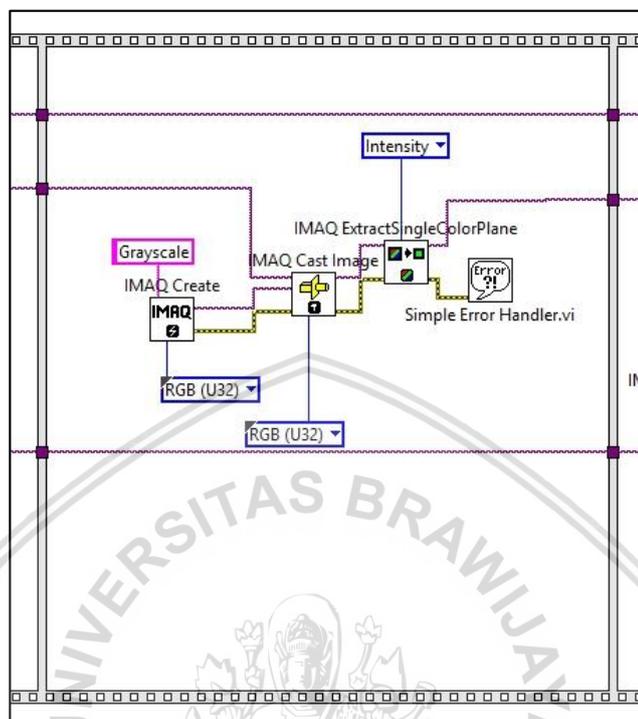
5.2.2.3 Implementasi *RGB to Grayscale*

Pada pengimplementasian ini dilakukan dengan tahapan sebagai berikut :

1. *IMAQ Create* digunakan untuk memanggil memori lagi yang gunanya agar tidak terjadi penumpukan data pada memori sebelumnya.
2. *IMAQ Cast Image* akan memanggil *IMAQ Create* yaitu memori untuk citra objek yang nanti akan berubah menjadi *grayscale*.
3. *IMAQ Extract Single Color Plane* akan dimasukkan citra dari objek yang telah diakuisisi dan memori baru yang telah di panggil yang kemudian akan melewati proses perubahan format citra menjadi *grayscale*. Proses itu di

tentukan dari pilihan pada *IMAQ Extract Single Color Plane* dimana kita memilih *intensity* sebagai pilihan ekstrak pada citra.

Program dapat dilihat pada gambar 5.20



Gambar 5.20. Implementasi *RGB to Grayscale*

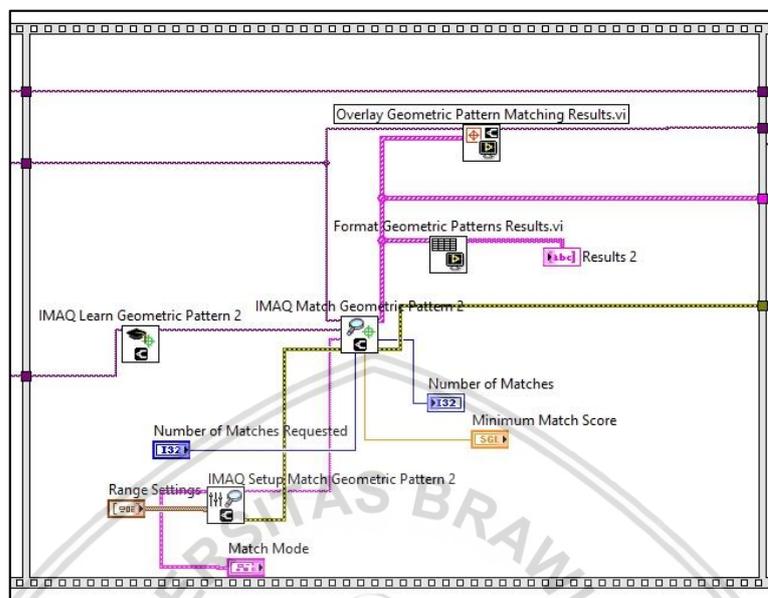
5.2.2.4 Implementasi *Geometric Pattern Matching*

Pada implemtasi *Geometric pattern matching* ini akan dijelaskan pada poin-poin berikut :

1. Sebelumnya dibutuhkan *setup* atau preparasi yang dilakukan pada *IMAQ Setup Match Geometric Pattern2*. Pada preparasi akan dimunculkan pilihan pada *Front Panel* berupa kotak pilihan untuk memilih kesesuaian berupa skala, rotasi dan oklusi. Skala disini ialah perubahan besar kecilnya objek, rotasi merupakan perubahan kemiringan objek, dan oklusi ialah perbedaan yang objek dengan *template*.
2. *IMAQ Learn Geometric Pattern.vi* merupakan proses *learning* untuk menentukan *template* layak sebagai *template* pada *geometric pattern matching*.
3. Setelah melakukan *setup* dan *learning*, maka akan dilakukan pencocokan antara citra objek dan citra *template*. Pada *IMAQ Match Geometric Pattern.vi* juga diberikan pilihan tentang banyak objek yang ingin dilakukan pencocokan dan batas skor kemiripan.
4. Setelah pencocokan selesai, akan keluar data berupa array akan diteruskan *Format Geomatic Pattern Result.vi*

5. *Overlay Geometric Pattern Matching Result.vi* yang akan memberikan tampilan citra yang telah diberikan kotak pembatas.

Program dapat dilihat pada gambar 5.21



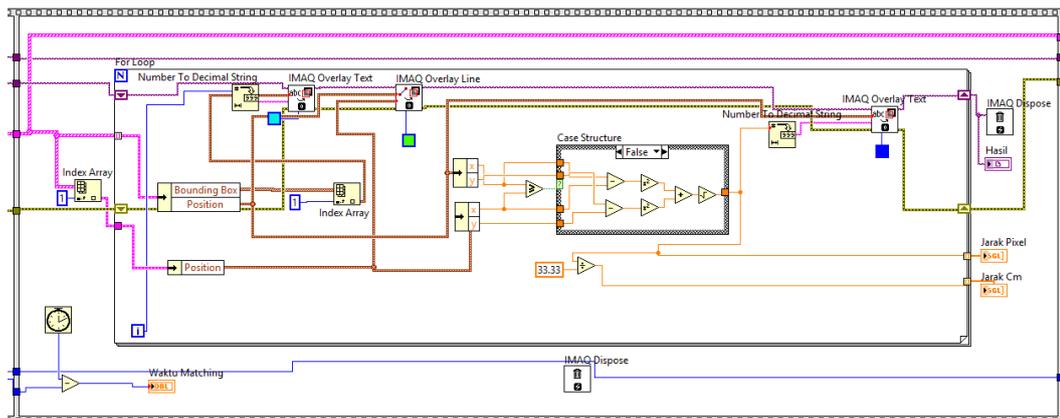
Gambar 5.21. Implementasi *Geometric Pattern Matching*

5.2.2.5 Implementasi *Overlaying*

Implementasi *overlaying* akan dilakukan pada poin-poin berikut :

1. Data *array* hasil pencocokan akan menuju *For Loop* sebanyak jumlah objek.
2. Pada Setiap pengulangan akan dilakukan proses penomoran objek yang menggunakan *IMAQ Overlay Text. String* yang akan masuk kedalam *IMAQ Overlay Text* merupakan hasil proses *Number To Decimal String* dengan inputan *number* dari nilai *For Loop*.
3. Sementara proses memberikan garis antara objek akan menggunakan *IMAQ Overlay Line*. Pada *Overlay Line* juga diberikan informasi berupa jarak antara objek dengan memanggil *IMAQ Overlay Text*. Nilai jarak merupakan hasil penghitungan menggunakan *Case Structure*.
4. Hasil penghitungan jarak dalam *pixel* dan dalam *centimeter* akan masuk kedalam *array*.
5. Hasil *overlaying* akan ditampilkan pada *Image.ctl*.

Program dapat dilihat pada gambar 5.22.



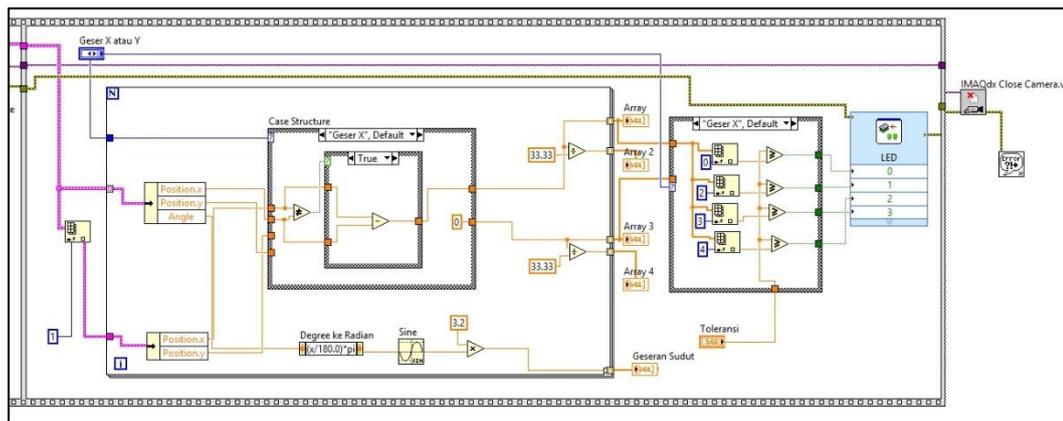
Gambar 5.22. Implementasi *Overlaying*

5.2.2.6 Implementasi Manualisasi Objek

Pada implementasi manualisasi objek ini akan dijelaskan pada poin-poin berikut :

1. Data *array* hasil pencocokan yaitu data berupa posisi dan sudut kemiringan akan masuk ke *For Loop*
2. Kemudian akan dilakukan penghitungan di setiap objek. Disini dilakukan dengan membedakan penyetaraan posisi objek menggunakan koordinat sumbu *x* atau koordinat sumbu *y* yang akan diberikan pilihan pada *Front Panel*. *Case Structure* menggunakan hasil *boolean* dari *Front Panel*.
3. Penghitungan kemiringan akan dilakukan dengan memasukkan nilai *Angle* ke dalam *Expression Node* yang kemudian diterukan menggunakan fungsi Sinus. Dilakukan perkalian untuk mendapatkan nilai oposit yaitu nilai jarak untuk membuat kemiringan objek menjadi normal.
4. Data hasil berupa jarak pergeseran sumbu *x* dan *y* dalam *pixel* dan centimeter akan dimasukkan kedalam *array*.
5. Pada setiap penyesuaian akan diberikan keluaran berupa lampu LED pada MyRIO 1900 sebanyak 4 buah untuk mengindikasikan objek sudah setara sesuai sumbu yang telah dipilih. Nilai toleransi juga dimasukkan dari *Front Panel* sebagai pembanding.
6. Setelah semua proses selesai maka akan dipanggil *Close Camera.vi* untuk menghentikan proses akuisisi citra.

Program dapat dilihat pada gambar 5.23.



Gambar 5.23. Implementasi Manualisasi Objek

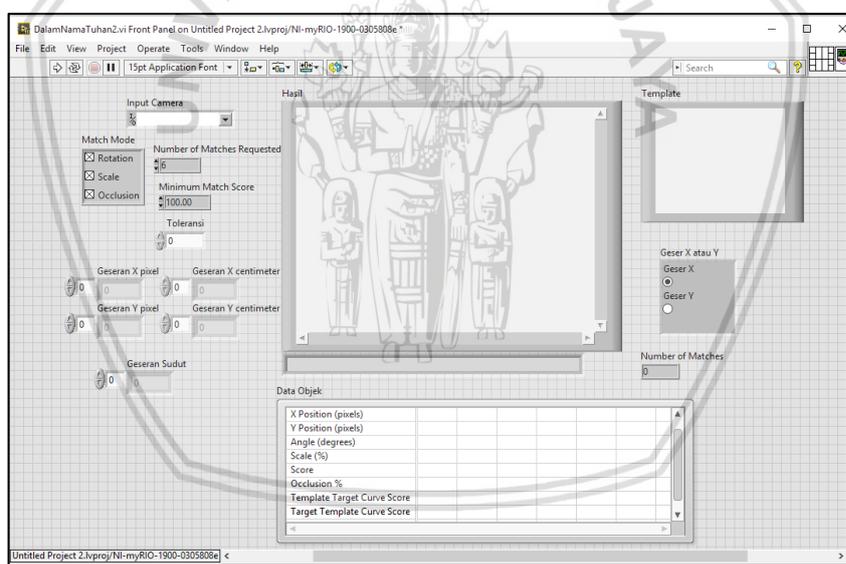
5.2.2.7 Implementasi Front Panel

Setelah melakukan implementasi pada tahap *block diagram* pada MyRIO, data seperti koordinat objek, citra objek yang telah di *overlay* dan citra *template* akan di tampilkan pada *front panel* yang berada pada *Notebook*. Ditampilkan juga kontrol dan indikator yang telah di programkan. Pada implementasi ini, setiap kontrol dan indikator akan diletakkan secara baik agar pengguna dapat dimudahkan dalam pembacaan program. Kontrol dan indikator dalam *front panel* akan dijelaskan pada poin-poin berikut :

1. *Input Camera* yaitu tempat memasukkan pilihan kamera yang digunakan. Sehubungan kamera yang dipakai hanya satu, maka pilihan yang keluar hanya satu yaitu "cam0".
2. *Match Mode* yaitu pilihan pada *setup match geometric pattern*. Pilihan yang keluar akan centang yaitu *Rotation*, *Scale* dan *Occlusion*.
 - a. *Rotation* merupakan pilihan apakah perbedaan rotasi target dengan *template* akan tetap di ambil kemudian di cocokkan.
 - b. *Scale* merupakan pilihan apakah perbedaan ukuran dari target dengan *template* akan tetap di ambil kemudian di cocokkan.
 - c. *Occlusion* merupakan pilihan apakah perbedaan dari fitur-fitur target dengan *template* akan tetap di ambil kemudian di cocokkan.
3. *Number of Match Requested* merupakan jumlah *match* atau hasil pencocokan yang kita inginkan.
4. *Minimum Match Score* merupakan nilai minimal kecocokan yang menjadi penentu kemiripan target dengan *template*.
5. Hasil merupakan *Image.ctl* yang menampilkan hasil dari *overlaying*.
6. *Template* merupakan *Image.ctl* yang menampilkan citra *template*.
7. *Number of Matches* merupakan jumlah hasil pencocokan yang didapat.



8. Data Objek merupakan tabel hasil pencocokan yang menunjukkan koordinat, skala, sudut, skor kemiripan dan skor korelasi.
 9. Geser x atau y merupakan kontrol pilihan yang menentukan apakah objek akan disesuaikan menurut sumbu x atau sumbu y .
 10. Toleransi merupakan nilai masukan yang menentukan nilai toleransi *pixel* antar koordinat.
 11. Geseran x *pixel* merupakan *array* hasil nilai penghitungan untuk menggeser objek sesuai sumbu x menggunakan nilai *pixel*.
 12. Geseran y *pixel* merupakan *array* hasil nilai penghitungan untuk menggeser objek sesuai sumbu y menggunakan nilai *pixel*.
 13. Geseran x centimeter merupakan *array* hasil nilai penghitungan untuk menggeser objek sesuai sumbu x menggunakan jarak centimeter.
 14. Geseran y centimeter merupakan *array* hasil nilai penghitungan untuk menggeser objek sesuai sumbu y menggunakan jarak centimeter.
 15. Geseran Sudut merupakan *array* yang menampilkan nilai Oposit dari objek.
- Diletakkan sedemikian rupa sehingga akan ditampilkan pada gambar 5.24.



Gambar 5.24. Implementasi Front Panel

BAB 6 PENGUJIAN DAN ANALISIS

Pada bab ini menjelaskan tentang pengujian sistem deteksi objek acak serta melakukan analisis pada pengujian yang dilakukan. Proses pengujian ini dilakukan guna mengetahui apakah sistem sudah dapat berjalan sesuai dengan analisis kebutuhan yang telah di tentukan. Proses analisis dilakukan untuk membuat kesimpulan dari penelitian yang telah dilakukan.

6.1 Pengujian Fungsional

Pengujian fungsional bertujuan untuk menguji sistem dapat berjalan sesuai fungsi yang telah di programkan. Sistem akan di uji apakah sistem dapat mengakuisisi objek, penghitungan jarak dan penghitungan kemiringan menggunakan metode *geometric pattern matching*.

6.1.1 Pengujian Akuisisi Objek

6.1.1.1 Tujuan Pengujian

Tujuan dari pengujian pengambilan gambar ialah untuk menguji apakah sistem dapat melakukan pengakuisisian objek dan melakukan proses *overlaying*.

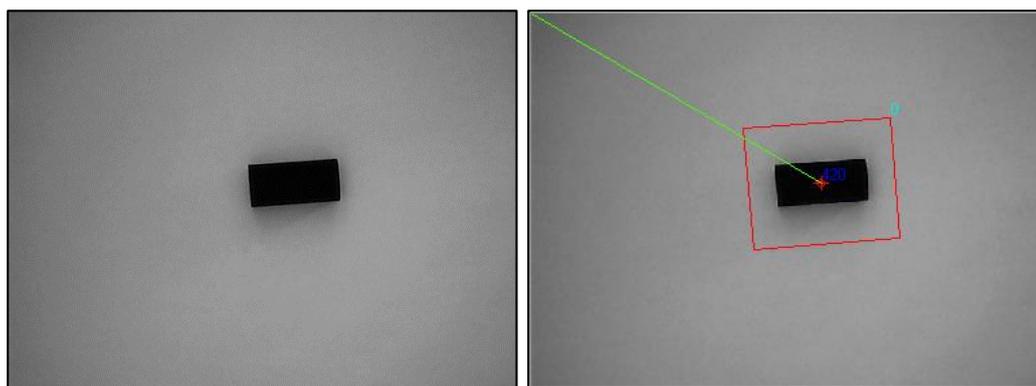
6.1.1.2 Prosedur Pengujian

Pengujian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Rangkaian alat diletakkan pada tempat yang mendapat cahaya yang cukup yaitu 500-700 lux.
2. Objek yaitu penghapus diletakkan secara acak pada kertas putih sebagai *background*.
3. Sistem di jalankan dengan cara mengaktifkan *Highlight Execution* lalu menekan *toolbar Run*.
4. Pengujian dilakukan secara bertahap dimulai dengan satu objek sampai objek yang kelima.

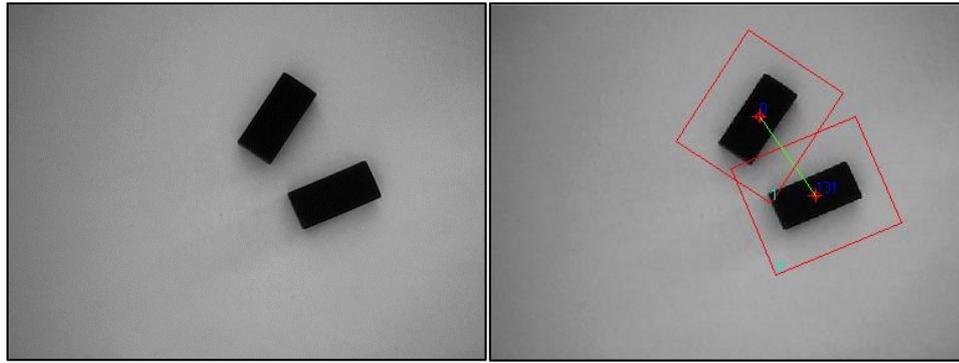
6.1.1.3 Hasil Pengujian

Hasil pengujian ditampilkan melalui gambar 6.1 sebagai berikut :



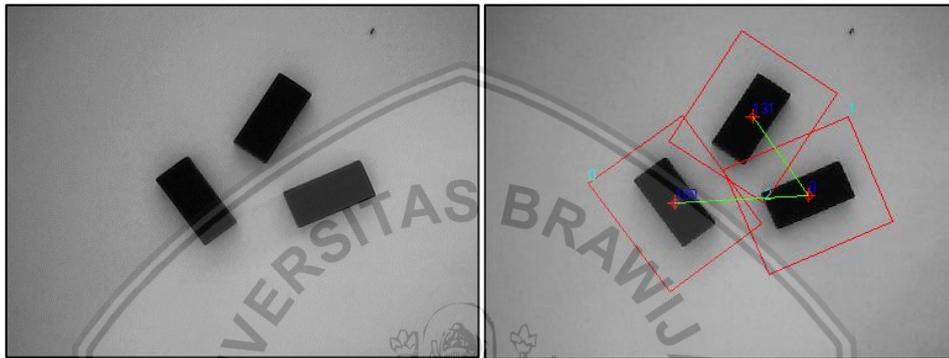
(1a)

(1b)



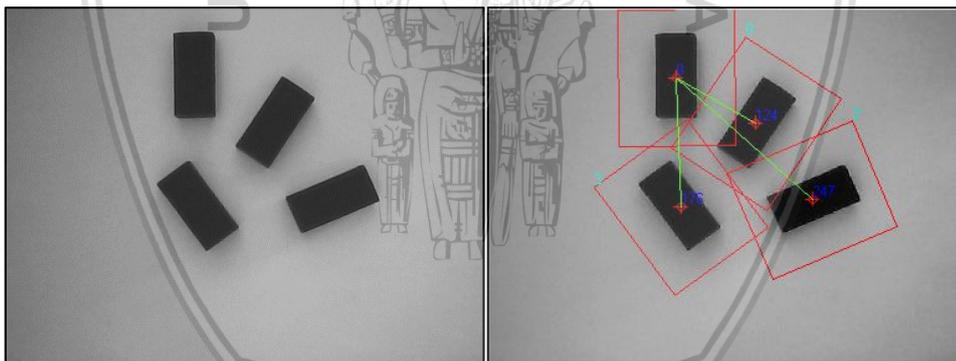
(2a)

(2b)



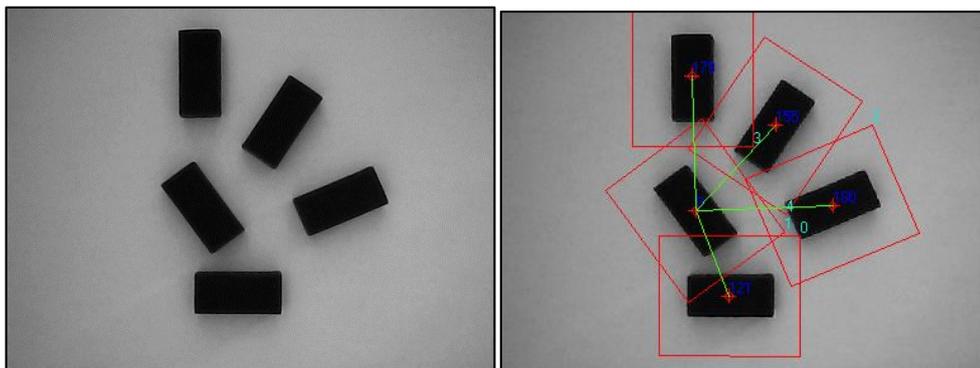
(3a)

(3b)



(4a)

(4b)



(5a)

(5b)

Gambar 6.1. Hasil Pengujian Akuisisi Objek



Pada gambar 6.1 terdapat beberapa gambar yaitu :

- (1a) Akuisisi 1 objek sebelum overlay, (1b) Akuisisi 1 objek sesudah overlay.
- (2a) Akuisisi 2 objek sebelum overlay, (2b) Akuisisi 2 objek sesudah overlay.
- (3a) Akuisisi 3 objek sebelum overlay, (3b) Akuisisi 3 objek sesudah overlay.
- (4a) Akuisisi 4 objek sebelum overlay, (4b) Akuisisi 4 objek sesudah overlay.
- (5a) Akuisisi 5 objek sebelum overlay, (5b) Akuisisi 5 objek sesudah overlay.

6.1.1.4 Analisis Pengujian

Dari hasil pengujian dapat diketahui bahwa sistem sudah dapat mengakuisisi objek yaitu dengan memberikan overlay bounding box yakni memberikan informasi akuisisi objek hasil matching dan overlay garis yaitu memberikan informasi jarak *pixel* antar objek. Sehingga disimpulkan bahwa fungsi sistem untuk mengakuisisi citra dan memberikan overlay sudah berjalan dengan baik.

6.1.2 Pengujian Penghitungan Jarak Objek

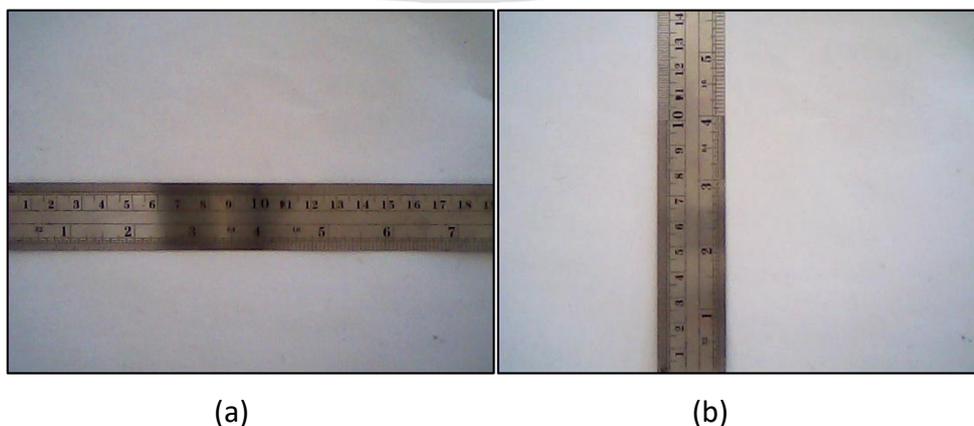
6.1.2.1 Tujuan Pengujian

Pengujian dilakukan untuk mengetahui fungsi jarak berfungsi dengan baik. Pengujian penghitungan jarak dilakukan dengan menguji hasil program dengan hasil pembacaan jarak secara manual.

6.1.2.2 Prosedur Pengujian

Pengujian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Rangkaian alat diletakkan pada tempat yang mendapat cahaya yang cukup yaitu 500-700 lux.
2. Objek yaitu penghapus diletakkan secara acak pada kertas putih sebagai *background*.
3. Menghitung luas cakupan kamera. Luas cakupan citra dihitung dengan mengambil gambar dengan meletakkan alat ukur dalam centimeter lalu mengambil hasil gambar.



Gambar 6.2. Luas Cakupan Kamera

Pada gambar 6.3 di dapatkan data panjang $x = 19.2$ cm dan $y = 14.4$ cm. Sedangkan luas area dalam *pixel* yaitu 640×480 , sehingga menghasilkan persamaan sebagai berikut :

$$\frac{x \text{ cm}}{y \text{ cm}} = \frac{x \text{ pixel}}{y \text{ pixel}} \tag{6.1}$$

$$\frac{19,2}{14,4} = \frac{640}{480} = 1,33$$

Dari hasil penghitungan di atas dapat disimpulkan bahwa data panjang yang di dapatkan dari penghitungan sudah tepat. Penghitungan selanjutnya ialah untuk mendapatkan pembesaran dari *pixel* ke centimeter, sehingga menghasilkan persamaan sebagai berikut :

$$M = \frac{s'}{s} \tag{6.2}$$

dimana M = pembesaran

s' = panjang x *pixel*

s = panjang x centimeter

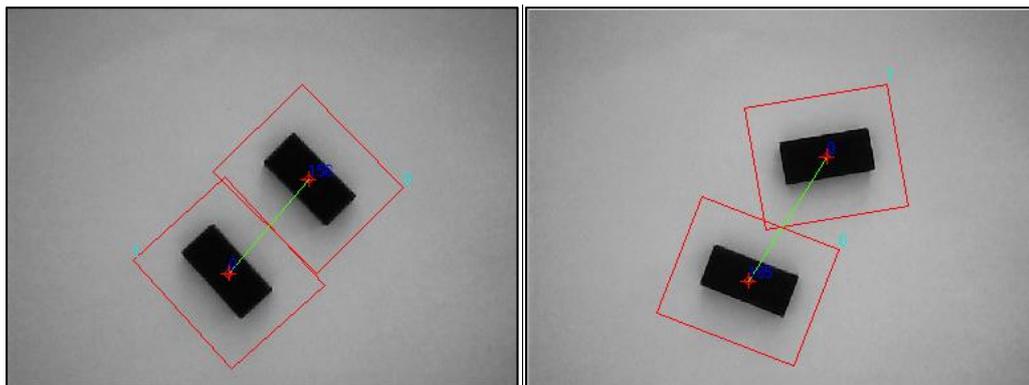
$$\frac{640}{19,2} = 33,33$$

Dari penghitungan di atas di dapatkan besar pembesaran sebesar 33.33. Nilai pembesaran itu akan di masukkan ke dalam program sehingga dari keluaran program berupa *pixel* dapat di ubah menjadi centimeter.

4. Dua objek diletakkan secara acak. Pengujian dilakukan sebanyak 5 kali.
5. Sistem di jalankan dengan cara mengaktifkan *Highlight Execution* lalu menekan *toolbar Run*.
6. Untuk menentukan hasil pengujian, pengujian ini akan menggunakan persamaan sebagai berikut .

$$\text{Persentase Error Jarak} = \frac{\text{Jarak penghitungan sistem} - \text{Jarak penghitungan manual}}{\text{Jarak penghitungan manual}} \times 100\% \tag{6.3}$$

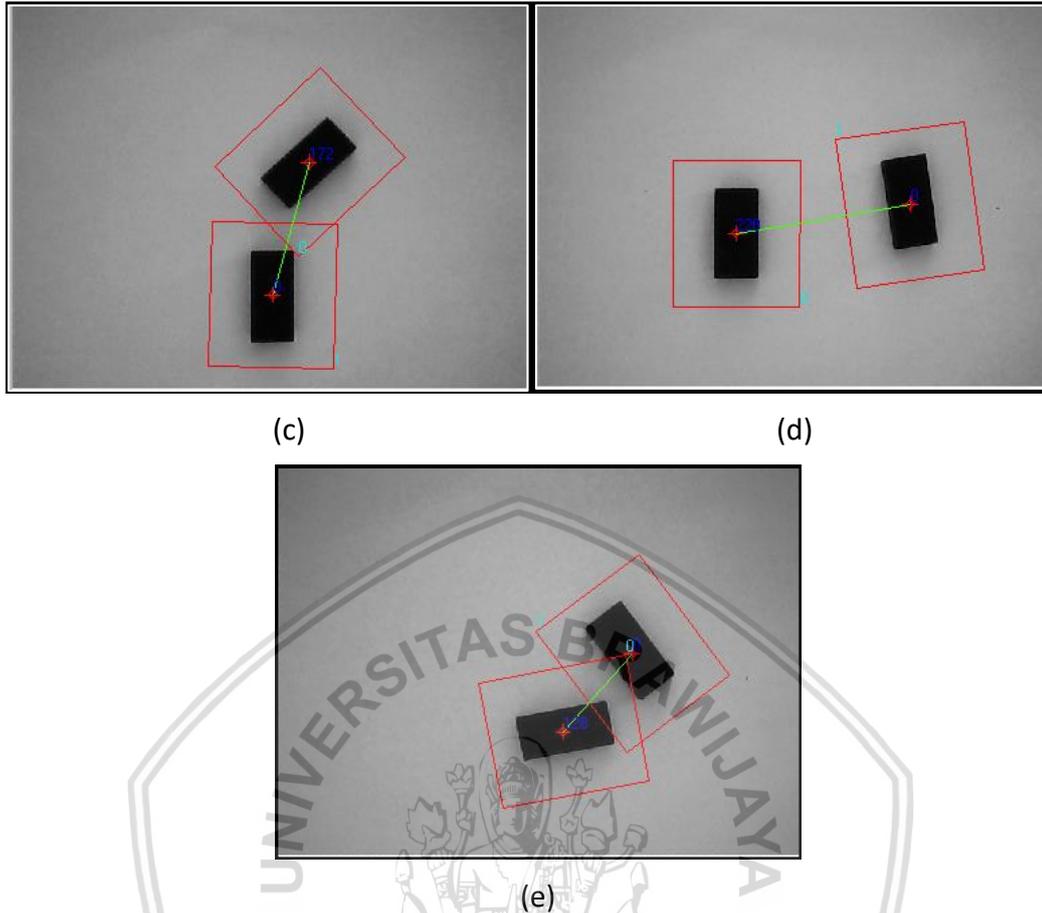
6.1.2.3 Hasil Pengujian



(a)

(b)





Gambar 6.3. Hasil Pengujian Penghitungan Jarak Objek

Dari pengujian penghitungan jarak di dapatkan hasil sebagai berikut:

Tabel 6.1 Hasil Pengujian Penghitungan Jarak Objek(cm)

Pengambilan	Jarak Penghitungan Sistem	Jarak Penghitungan Manual	Persentase Error
Gambar a	4,36	4,35	0,22 %
Gambar b	5,21	5,20	0,19%
Gambar c	4,94	4,89	1,02%
Gambar d	6,35	6,34	0,15%
Gambar e	3,49	3,50	0,28%
Rata - rata			0,37%

6.1.2.4 Analisis Pengujian

Tabel 6.1 merupakan tabel hasil pengujian jarak penghitungan sistem dengan jarak penghitungan manual. Dari tabel tersebut didapatkan hasil rata-rata

persentase eror sebesar 0,37% atau memiliki persentasi keberhasilan sebesar 99.63%. Persentase hasil tidak didapatkan 100% karena proses pengujian penghitungan manual memiliki margin eror 0,005 cm.

6.1.3 Pengujian Kemiringan

6.1.3.1 Tujuan Pengujian

Pengujian fungsi dari guna mendeteksi kemiringan objek akan dilakukan dengan menguji kemiringan objek hasil pembacaan sistem dengan pembacaan manual. Kemiringan objek dapat diukur dengan mengambil nilai oposit.

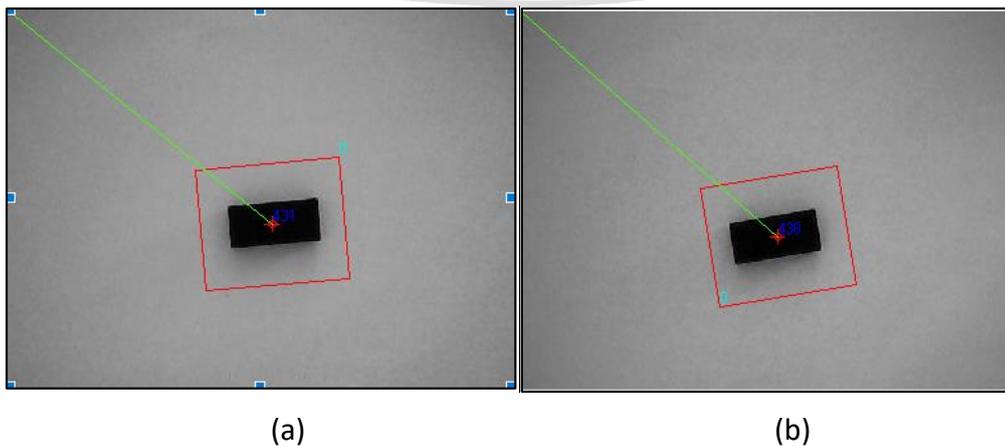
6.1.3.2 Prosedur Pengujian

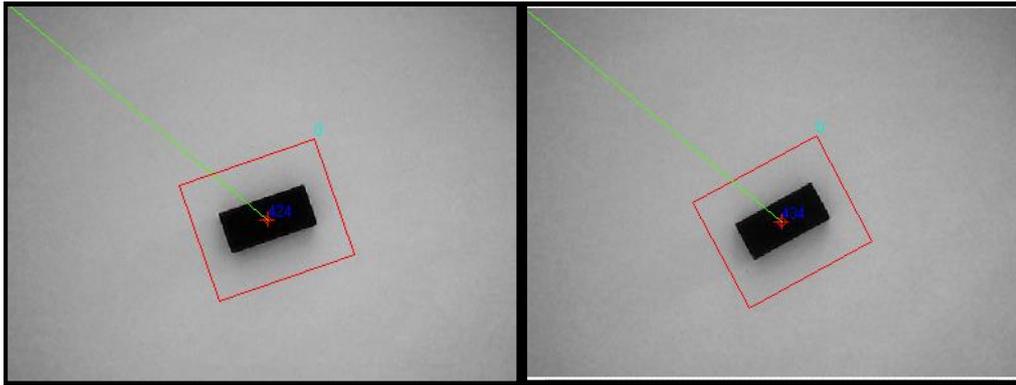
Pengujian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Rangkaian alat diletakkan pada tempat yang mendapat cahaya yang cukup yaitu 500-700 lux.
2. Objek yaitu penghapus diletakkan secara acak pada kertas putih sebagai *background*.
3. Objek diletakkan dengan kemiringan yang sudah ditentukan menurut pembacaan manual kemudian program dijalankan untuk membaca nilai oposit hasil pembacaan program.
4. Sistem di jalankan dengan cara mengaktifkan *Highlight Execution* lalu menekan *toolbar Run*.
5. Pengujian kemiringan dilakukan dengan menguji hasil program dengan hasil pembacaan panjang oposit secara manual. Untuk menentukan hasil pengujian, pengujian ini akan menggunakan persamaan sebagai berikut.

$$\text{Persentase Error Oposit} = \frac{\text{Oposit penghitungan sistem} - \text{Oposit penghitungan manual}}{\text{Oposit penghitungan manual}} \times 100\% \quad (6.4)$$

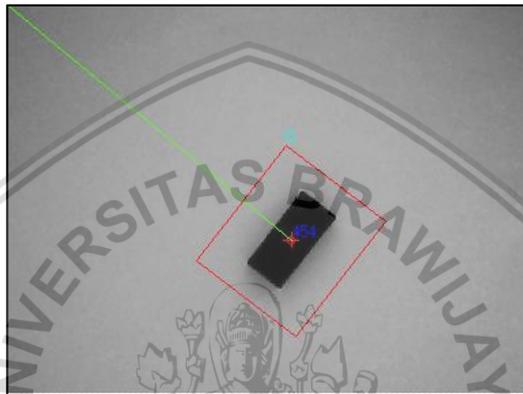
6.1.3.3 Hasil Pengujian





(c)

(d)



(e)

Gambar 6.4. Hasil Pengujian Kemiringan

Dari pengujian penghitungan jarak di dapatkan hasil sebagai berikut:

Tabel 6.2 Hasil Pengujian Penghitungan Oposit (cm)

Pengambilan	Oposit dari Penghitungan Sistem	Oposit dari Penghitungan Manual	Persentase Error
Gambar a	0,19	0,20	5 %
Gambar b	0,48	0,50	4%
Gambar c	1,05	1,00	5%
Gambar d	1,53	1,50	2%
Gambar e	2,50	2,50	0%
Rata - rata			3,2%

6.1.3.4 Analisis Pengujian

Tabel 6.2 merupakan tabel hasil pengujian oposit penghitungan sistem dengan oposit penghitungan manual. Dari tabel tersebut didapatkan hasil rata-rata persentase eror sebesar 3,2% atau persentase keberhasilan sebesar 96,8%.



Persentase hasil tidak didapatkan 100% karena proses pengujian penghitungan manual memiliki margin eror 0,005 cm.

6.2 Pengujian Akurasi Koordinat Objek

6.2.1 Tujuan Pengujian

Pengujian ini akan dilakukan untuk menguji akurasi dari manualisasi yang dilakukan untuk membuat objek tidak acak atau sudah tertata baik tertata sesuai sumbu x maupun sumbu y. Pengujian ini akan menggunakan lima buah objek yang awalnya di letakkan secara acak.

6.2.2 Prosedur Pengujian

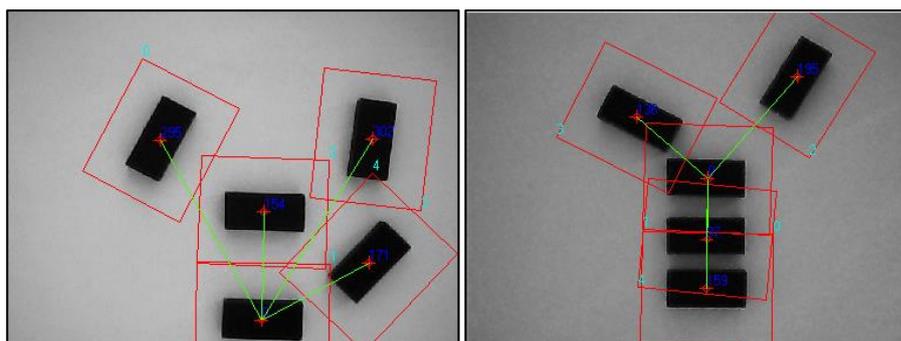
Pengujian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Rangkaian alat diletakkan pada tempat yang mendapat cahaya yang cukup yaitu 500-700 lux.
2. Objek yaitu penghapus diletakkan secara acak pada kertas putih sebagai *background*.
3. Objek diletakkan sejajar menurut sumbu x secara bertahap. Secara bertahap disini ialah dimulai dengan satu objek sejajar, selanjutnya ada 2 objek yang sejajar dan kemudian sampai semua objek sejajar.
4. Sistem di jalankan dengan cara mengaktifkan *Highlight Execution* lalu menekan *toolbar Run*.
5. Pada setiap tahapan objek sejajar dilakukan juga masukan toleransi yang berbeda yaitu pertama *0 pixel*, *3 pixel* dan *5 pixel*.
6. Pada MyRIO telah disediakan 4 buah LED yang pada pengujian ini akan digunakan sebagai indikator manualisasi apakah sudah terjadi penyetaraan atau belum. Persentase keberhasilan dapat dihitung dari kecocokan ada objek sejajar dengan LED menyala, sehingga untuk menghitung persentase keberhasilan digunakan persamaan sebagai berikut.

$$\text{Persentase Keberhasilan} = \frac{\text{Jumlah LED menyala}}{\text{Jumlah Objek Sejajar}} \times 100 \quad (6.5)$$

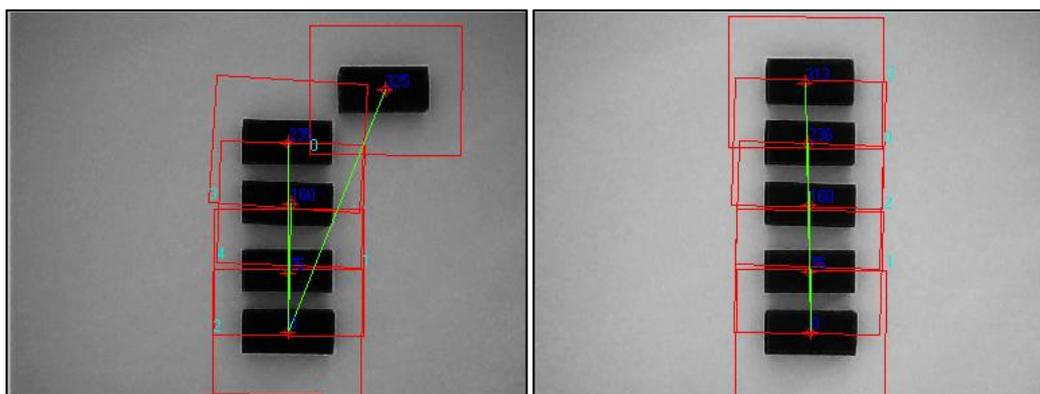
6.2.3 Hasil Pengujian

Hasil pengujian dapat di lihat pada gambar 6.5.



(Percobaan 1)

(Percobaan 2)



(Percobaan 3)

(Percobaan 4)

Gambar 6.5. Hasil Pengujian Akurasi Koordinat Objek

Tabel 6.3 Hasil Pengujian Koordinat Objek

Percobaan	Selisih Toleransi	Objek Sejajar				LED MyRIO				Persentase Keberhasilan
		1	2	3	4	1	2	3	4	
1	0 px	✓	x	x	x	x	x	x	x	0%
2	0 px	✓	✓	x	x	x	x	x	x	0%
3	0 px	✓	✓	✓	x	x	x	x	x	0%
4	0 px	✓	✓	✓	✓	x	x	x	x	0%
Rata-rata Persentase Keberhasilan Toleransi = 0 px										0%
1	3 px	✓	x	x	x	✓	x	x	x	100%
2	3 px	✓	✓	x	x	x	✓	x	x	50%
3	3 px	✓	✓	✓	x	x	✓	✓	x	66.6%
4	3 px	✓	✓	✓	✓	✓	x	✓	✓	75%
Rata-rata Persentase Keberhasilan Toleransi = 3 px										72.75%
1	5 px	✓	x	x	x	✓	x	x	x	100%
2	5 px	✓	✓	x	x	✓	✓	x	x	100%
3	5 px	✓	✓	✓	x	✓	✓	✓	x	100%
4	5 px	✓	✓	✓	✓	✓	✓	✓	✓	100%
Rata-rata Persentase Keberhasilan Toleransi = 5 px										100%



6.2.4 Analisis Hasil Pengujian

Dari hasil pengujian di atas kita dapat diketahui bahwa dari pengujian akurasi dengan toleransi 0 px mendapatkan hasil persentase keberhasilan sebesar 0%, untuk akurasi dengan toleransi 3 px sebesar 72,75% dan sedangkan untuk akurasi dengan toleransi 5 px sebesar 100%. Pembacaan posisi objek yang disejajarkan memiliki akurasi 0% ketika tidak diberikan nilai toleransi *pixel* karena pada keluaran posisi koordinat *x* dan *y* karena pada 0,1 mm terdapat 3 *pixel* atau setiap 1 *pixel* memiliki jarak 0,03mm.

6.3 Pengujian Waktu Pemrosesan

6.3.1 Tujuan Pengujian

Tujuan pengujian ini untuk mengukur waktu yang dibutuhkan sistem selama proses maupun waktu *matching* terhadap jumlah objek. Pada LabVIEW terdapat 2 cara untuk menjalankan sistem yaitu tanpa *highlight* dan dengan *highlight*. Pengujian ini akan menguji waktu pemrosesan pada kedua cara tersebut.

6.3.2 Prosedur Pengujian

Pengujian dilakukan dengan tahapan-tahapan sebagai berikut :

1. Rangkaian alat diletakkan pada tempat yang mendapat cahaya yang cukup yaitu 500-700 lux.
2. Objek yaitu penghapus diletakkan secara acak pada kertas putih sebagai *background*.
3. Sistem di jalankan dengan 2 cara yaitu dengan tidak mengaktifkan *Highlight Execution* dan dengan mengaktifkan *Highlight Execution* lalu menekan *toolbar Run*.
4. Pengujian dilakukan secara bertahap dimulai dengan satu objek sampai objek yang kelima.
5. Tambahkan *time stamp* pada proses akuisisi dan akhir proses untuk menghitung waktu pemrosesan. Tambahkan *time stamp* pada proses *RGB to Grayscale* dan proses *Overlaying* untuk menghitung waktu *matching*.
6. Hitung selisih waktu pada *time stamp* kemudian berikan indikator pada Front Panel.
7. Pengujian di lakukan dengan menghitung waktu total program berjalan dengan variable uji yaitu jumlah objek yang di dikenali. Selisih setiap waktu *matching* akan dihitung kemudian akan di rata-ratakan. Penghitungan selisih waktu akan dilakukan dengan persamaan sebagai berikut.

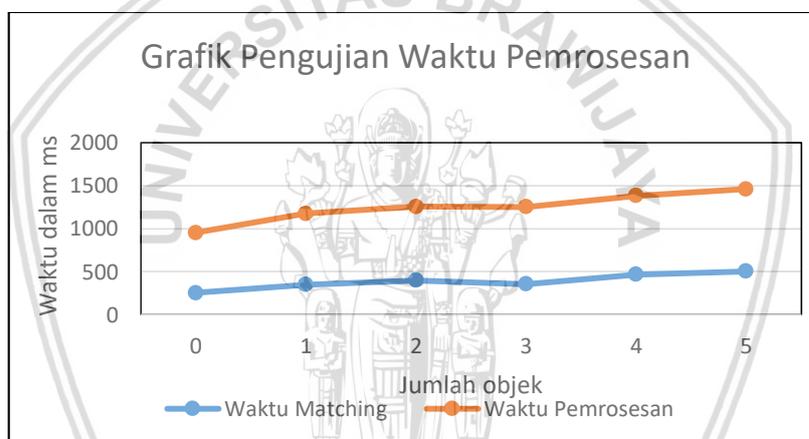
$$\text{Selisih} = \text{Waktu} (n + 1) - \text{Waktu} (n) \quad (6.6)$$

Dimana *n* = jumlah objek.

6.3.3 Hasil Pengujian

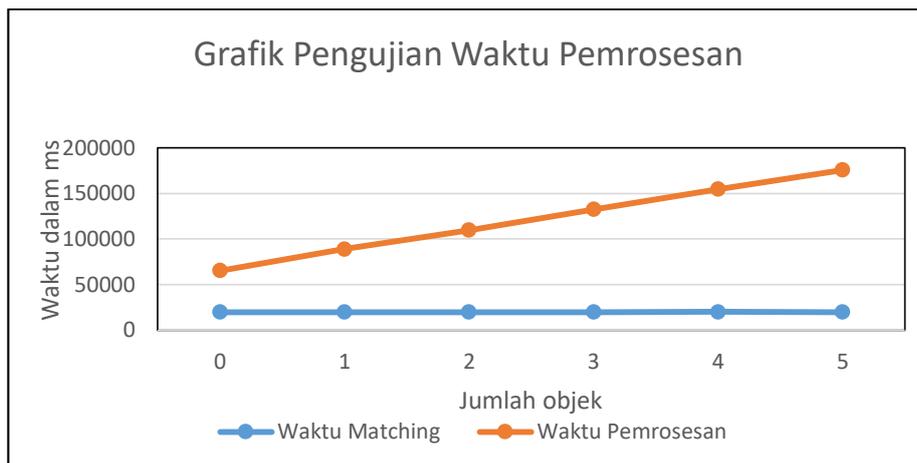
Tabel 6.4 Hasil Pengujian Waktu Pemrosesan Tanpa *Highlight* (ms)

Jumlah Objek	Waktu Matching	Selisih Waktu Matching	Waktu Pemrosesan	Selisih Waktu Pemrosesan
0	253	0	700	0
1	350	97	825	123
2	398	48	858	33
3	455	57	899	41
4	468	13	915	16
5	505	37	955	40
Rata rata Selisih		50.4		51



Gambar 6.6. Grafik Pengujian Waktu Pemrosesan Tanpa *Highlight*
Tabel 6.5 Hasil Pengujian Waktu Pemrosesan Dengan *Highlight* (ms)

Jumlah Objek	Waktu Matching	Selisih Waktu Matching	Waktu Pemrosesan	Selisih Waktu Pemrosesan
0	19740	0	45693	0
1	19701	39	69282	23589
2	19678	23	90240	20958
3	19835	157	112819	22579
4	19960	125	134705	21886
5	19644	316	156286	21581
Rata rata Selisih		132		22186.6



Gambar 6.7. Grafik Pengujian Waktu Pemrosesan Dengan *Highlight*

6.3.4 Analisis Hasil Pengujian

Dari hasil pengujian dapat disimpulkan bahwa jumlah objek berpengaruh terhadap waktu *matching* dan waktu pemrosesan baik dengan *highlight execution* maupun tanpa *highlight execution*. Waktu pemrosesan berbanding lurus dengan jumlah objek. Untuk tanpa *highlight execution* didapatkan rata-rata selisih setiap penambahan setiap satu objek pada waktu *matching* sebesar 50,4 ms sedangkan pada waktu pemrosesan 51 ms. Sedangkan dengan *highlight execution* didapatkan rata-rata selisih setiap penambahan setiap satu objek pada waktu *matching* sebesar 132 ms sedangkan pada waktu pemrosesan 22186,6 ms.

BAB 7 PENUTUP

7.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan untuk memenuhi permasalahan yang ada pada rumusan masalah, hasil dari perancangan dan implementasi, serta pengujian alat yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. Untuk menerapkan *object recognition* atau pengenalan objek pada *platform* MyRIO dan pembacaan koordinat x dan y objek dapat dilakukan dengan menggunakan metode *Geometric Match Pattern* yang tersedia dalam *library Vision Acquisition* pada *software* LabVIEW. Pembacaan dilakukan dengan menggunakan kamera berupa *webcam* yang akan membaca citra objek yang kemudian akan di ubah menjadi *grayscale* lalu kemudian dilakukan pencocokan dengan *template*. Setelah pencocokan selesai maka dilanjutkan dengan memberi *overlay* pada citra objek dan melakukan manualisasi untuk membaca akurasi.
2. Akurasi jarak antar objek yang dibaca setelah pengujian dengan membandingkan nilai jarak hasil keluaran sistem dengan pengukuran jarak secara manual di dapatkan persentase *error* sebesar 0,37% atau persentase keberhasilan sebesar 99,63%.
3. Nilai oposit didapatkan dari hasil penghitungan trigonometri dengan mengambil sudut kemiringan dengan panjang benda uji. Akurasi akuisisi kemiringan didapatkan setelah membandingkan nilai oposit hasil keluaran sistem dengan pengukuran oposit secara manual di dapatkan persentase *error* sebesar 3,2% atau persentase keberhasilan sebesar 96,8%.
4. Akurasi koordinat objek didapatkan dengan melakukan pengujian dengan variabel bebas yaitu toleransi selisih *pixel*. Variabel bebas tersebut ialah 0 *pixel*, 3 *pixel*, dan 5 *pixel*. Setelah pengujian menggunakan 5 objek didapatkan persentase keberhasilan pada 0 *pixel* sebesar 0%, pada 3 *pixel* sebesar 72,75% dan pada 5 *pixel* sebesar 100%. Pembacaan posisi objek yang disejajarkan memiliki akurasi 0% ketika tidak diberikan nilai toleransi *pixel* karena pada keluaran posisi koordinat x dan y karena pada 0.1 mm terdapat 3 *pixel* atau setiap 1 *pixel* memiliki jarak 0,03mm.
5. Pengujian waktu *matching* dan waktu pemrosesan sistem dilakukan untuk melihat pengaruh jumlah objek terhadap waktu *matching* dan waktu pemrosesan sistem. Pengujian dilakukan dengan 2 cara menjalankan sistem yaitu dengan *highlight execution* dan tanpa *highlight execution*. Pada pengujian tanpa *highlight execution* didapatkan nilai rata-rata setiap penambahan satu objek pada waktu *matching* sebesar 50,4ms dan waktu pemrosesan sistem sebesar 51ms. Sedangkan pada pengujian dengan *highlight execution* didapatkan nilai rata-rata setiap penambahan satu objek pada waktu *matching* sebesar 132ms dan waktu pemrosesan sistem sebesar

22118ms. Didapatkan hasil bahwa jumlah objek memiliki pengaruh berbanding lurus dengan waktu *matching* dan waktu pemrosesan sistem.

7.2 Saran

Agar sistem dapat dikembangkan lebih lanjut dan lebih sempurna, maka diperlukan beberapa hal yang disarankan dilakukan yaitu,

1. Pada penelitian ini, sistem hanya menggunakan satu jenis bahan uji. Diharapkan pada penelitian selanjutnya bisa menggunakan lebih dari satu jenis bahan uji
2. Pada penelitian ini, sistem dibatasi dengan melakukan penggesaran atau penyetaraan objek menggunakan manualisasi manusia. Penelitian selanjutnya diharapkan bisa melakukan otomatisasi robotik pada penyetaraan posisi objek.



DAFTAR PUSTAKA

- A. Frosini, V. G. S. R., 2016. *Geometric properties of matrices induced by pattern avoidance*. *Theoretical computer science*, Volume 624, pp. 109-120.
- Brei, Terry M., 2013. *Why Calibration of Your Measuring is Important*. [online] Sure Control Inc. Tersedia di : <<http://www.surecontrols.com/why-calibration-of-your-measuring-instruments-is-important/>> [Diakses 1 Mei 2018].
- Gonzalez, R.C. and Woods, R.E., 1987. *Digital image fundamentals. Digital Imaging Processing Second Edition*, Addison-Wesley Publishing Company.
- Instruments, N., 2015. *User Guide and Specifications NI MyRIO-1900*. National Instruments, p.11.
- Khairunnisa, Afidah., 2014. *Matematika Dasar*. Jakarta. Raja Grafindo Persada.
- Krishnamurthy, N., 2002. *Introduction to Computer Graphics*. Tata McGraw-Hill Publishing Company.
- Liao, H.Q., Qiu, Z.R. and Feng, G.H., 2011. *The design of LDF data acquisition system based on LabVIEW*. *Procedia Environmental Sciences*, 10, pp.1188-1192.
- Nair, D., Fisher, M.S., Kumar, S.V., Smyth, B. and Ramamoorthy, S., National Instruments Corp, 2008. *Geometric pattern matching using dynamic feature combinations*. U.S. Patent 7,340,089.
- Sutoyo, T., Mulanto, E. and Suhartono, V., 2009. *Teori Pengolahan Citra Digital*. Yogyakarta: Penerbit Andi.
- Syahrul, S. and Kurniawan, A., 2012. *Sistem Penjejak Posisi Obyek Berbasis Umpan Balik Citra*. *JSM (Jurnal SIFO Mikroskil)*, 13(1)
- Travis, J. and Kring, J., 2007. *LabVIEW for everyone: graphical programming made easy and fun*. Prentice-Hall.
- Wei, H. and Wang, L., 2018. *Understanding of indoor scenes based on projection of spatial rectangles*. *Pattern Recognition*, 81, pp.497-514.