TUGAS AKHIR - TE141599

RANCANG BANGUN SISTEM PENGUKURAN POSISI TARGET DENGAN KAMERA STEREO UNTUK PENGARAH SENJATA OTOMATIS

Anas Maulidi Utama
NRP 2213100341

Dosen Pembimbing
Dr. Ir. Djoko Purwanto, M.Eng.
Ronny Mardiyanto, S.T., M.T., Ph.D.

JURUSAN TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2016

FINAL PROJECT - TE141599

SYSTEM DEVELOPMENT FOR TARGET POSITION MEASUREMENT USING STEREO CAMERA FOR AUTOMATIC DIRECTIONAL WEAPON

Anas Maulidi Utama
NRP 2213100341

Supervisor
Dr. Ir. Djoko Purwanto, M.Eng.
Ronny Mardiyanto, S.T., M.T., Ph.D.

ELECTRICAL ENGINEERING DEPARTMENT
Faculty of Industrial Technology
Sepuluh Nopember Institute of Technology
Surabaya 2016

RANCANG BANGUN SISTEM PENGUKURAN POSISI TARGET DENGAN KAMERA STEREO UNTUK PENGARAH SENJATA OTOMATIS

TUGAS AKHIR

Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Sarjana Teknik
Pada
Bidang Studi Elektronika
Jurusan Teknik Elektro
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember

Menyetujui

Dosen Pembimbing I	Dosen Pembimbing II
Dr. Ir. Djoko Purwanto, M.Eng.	Ronny Mardiyanto, S.T., M.T., Ph.D.
NIP. 196512111990021002	NIP. 198101182003121003

SURABAYA
JUNI, 2016

RANCANG BANGUN SISTEM PENGUKURAN POSISI TARGET DENGAN KAMERA STEREO UNTUK PENGARAH SENJATA OTOMATIS

Nama : Anas Maulidi Utama
Pembimbing I : Dr. Ir. Djoko Purwanto, M.Eng.
Pembimbing II : Ronny Mardiyanto, S.T., M.T., Ph.D.

ABSTRAK

Salah satu hal yang penting dalam mengarahkan senjata secara otomatis ke target adalah informasi posisi dari target terhadap senjata. Terdapat banyak metode untuk mengetahui posisi target. Salah satunya adalah dengan metode pengukuran triangulasi. Metode ini membutuhkan minimal dua citra untuk medapatkan informasi jarak target terhadap kamera. Kemudian, informasi jarak tersebut bisa diolah untuk mengetahui posisi target terhadap senjata.

Pada tugas akhir ini, *stereo visual* digunakan untuk mendukung proses pengukuran triangulasi. *Stereo visual* menggunakan dua kamera untuk menghasilkan dua citra. Dalam sistem ini, salah satu kamera bertindak sebagai pemilih target. Citra yang ditangkap dua kamera tersebut akan diproses oleh *processing unit* untuk mendapatkan informasi posisi target terhadap senjata. Informasi ini digunakan untuk menggerakkan motor pada *platform* senjata agar senjata mengarah ke target. Sistem ini juga memiliki fitur tambahan berupa *tracking* dengan metode *optical flow* yang membuat sistem ini dapat mengikuti target yang dipilih.

Hasil pengujian yang dilakukan pada tugas akhir ini adalah sistem dapat menentukan posisi target yang dipilih oleh operator dan juga dapat mengarahkan senjata ke arah target tersebut. Agar senjata bisa diarahkan ke target, pengukuran posisi target menjadi hal terpenting dalam sistem ini. Akurasi tertinggi dalam penentuan posisi target dicapai ketika jarak antar dua kamera sekitar 30 cm, yaitu dengan kesalahan pengukuran bernilai kurang dari 5%.

Kata kunci: pengukuran posisi, *stereo visual*, teknologi senjata

SYSTEM DEVELOPMENT FOR TARGET POSITION MEASUREMENT USING STEREO CAMERA FOR AUTOMATIC DIRECTIONAL WEAPON

Name	**: Anas Maulidi Utama**
Supervisor	**: Dr. Ir. Djoko Purwanto, M.Eng.**
Co-Supervisor	**: Ronny Mardiyanto, S.T., M.T., Ph.D.**

ABSTRACT

One of the important thing in directing the weapon automatically to target is the position of target to the weapon. There are many methods to determine the position of target. One of method is triangulation method. This method requires at least two image to obtain target distance to the camera. Then, the information of target distance can be processed to determine the position of target to the weapon.

In this final project, stereo visual is used to support the process of triangulation method. Stereo visual uses two cameras to produce two images. In this system, one camera acts as a target selector. The image captured by two cameras will be processed by processing unit to obtain the target position to the weapon. This information is used to drive the motor on the platform that leads the weapon to the target. This system also has additional feature such as tracking using optical flow method that makes this system can follow the choosen target.

The results of tests performed in this project is the system can determine the position of a target selected by operator and also direct the weapon toward the target. If we want to direct the weapon to the target, position measurement become the most important thing in this system. The highest accuracy in the position measurement was achieved when the distance between cameras was about 30 cm, which the error was less than 5%.

Key words: position measurement, stereo visual, weapon technology

DAFTAR ISI

ABSTRAK i
ABSTRACT iii
KATA PENGANTAR v
DAFTAR ISI vii
DAFTAR GAMBAR ix
DAFTAR TABEL xi
BAB I PENDAHULUAN 1
1.1. Latar Belakang 1
1.2. Perumusan Masalah 2
1.3. Batasan Masalah 2
1.4. Tujuan 2
1.5. Metodologi 3
1.6. Sistematika Penulisan 4
1.7. Relevansi dan Manfaat 5
BAB II TINJAUAN PUSTAKA DAN TEORI PENUNJANG 7
2.1. Citra Digital 7
2.1.1. Citra Warna 8
2.1.2. Citra *Grayscale* 9
2.1.3. Citra Biner 10
2.2. *Stereo Vision* 11
2.2.1. *Stereo Calibration* 12
2.2.2. *Stereo Rectification* 14
2.3. *Stereo Matching* 15
2.4. Metode Pengukuran Triangulasi 16
2.5. *Optical Flow* 19
2.6. *Kalman filter* 21
2.7. *Pulse Width Modulation* 22
2.8. Mikrokontroler STM32F4 22
2.9. *Driver* motor 22
2.10. Kontrol PID 24
2.11. Komunikasi Serial 24
2.12. OpenCV 25
2.13. Qt Creator 25
BAB III PERANCANGAN SISTEM 27
3.1. *Graphical User Interface* Sistem 29
3.2. *Stereo Calibration* dan *Stereo Rectidication* 31
3.3. Pengukuran Posisi Target 32

3.2.1. *Preprocessing* Citra 32
3.2.2. *Template Matching* 33
3.2.3. Pengukuran Posisi Target dengan Metode Triangulasi ... 35
3.4. Penghitungan Sudut Posisi Motor *Platform* Senjata 35
3.5. Pengarahan Senjata ke Target 37
3.5.1. Kontrol PID 37
3.5.2. *Driver* Motor 39
3.5.3. Motor DC 40
3.5.4. *Rotary Encoder* 40
3.5.5. Sensor *Proximity* 41
3.5.6. *Limit Switch* 42
3.6. Fitur Pendukung 42
3.6.1. Komunikasi Serial 42
3.6.2. GUI Pengujian Kontrol Senjata 45
3.6.3. *Optical Flow* 47
3.6.4. *Kalman filter* 47
BAB IV PENGUJIAN DAN ANALISA 49
4.1. Pengujian *Stereo Calibration* dan *Stereo Rectification* 49
4.2. Pengujian Pengukuran Posisi Target dengan Metode Triangulasi 51
4.3. Pengujian Pengukuran Posisi Target berdasarkan Matriks Proyeksi 53
4.4. Pengujian Pengukuran Posisi Target dengan Jarak Antar Kamera yang Berbeda-beda 59
4.5. *Tuning* PID dan Pengujian Kontrol *Platform* Senjata Berdasarkan Sudut Horisontal dan Sudut Vertikal 61
4.6. Pengujian Kontrol *Platform* Senjata Berdasarkan Posisi Kartesian 63
4.7. Pengujian *Optical Flow* dan *Kalman filter* 63
4.8. Pengujian Pengarahan Target 64
4.9. Pengujian Performa Sistem 67
4.10. *Real Application* 72
BAB V PENUTUP 73
5.1. Kesimpulan 73
5.2. Saran 73
DAFTAR PUSTAKA 75
LAMPIRAN 77
BIODATA PENULIS 85

DAFTAR GAMBAR

Gambar 2.1 Konsep *pixel* 8
Gambar 2.2 Kombinasi warna RGB 9
Gambar 2.3 Rentang *gray level* 10
Gambar 2.4 Citra RGB dan Citra *Grayscale* 10
Gambar 2.5 Citra Biner 11
Gambar 2.6 Peletakan kamera stereo yang sejajar (a) dan peletakan kamera stereo dengan kesalahan vertikal (b) dan (c) 12
Gambar 2.7 Proses Pengambilan Citra dari Dua Kamera 16
Gambar 2.8 Pengambilan citra oleh kamera kiri 17
Gambar 2.9 Pengambilan citra oleh kamera kanan 18
Gambar 2.10 Ilustrasi *Optical Flow* 19
Gambar 2.11 Algoritma *Pyramidal Lucas Kanade* 20
Gambar 2.12 *Duty cycle* pada PWM 23
Gambar 2.13 *Board* STM32F4-*Discovery* 23
Gambar 2.14 Skematik sederhana *H-Bridge* 23
Gambar 2.15 Logo OpenCV 25
Gambar 2.16 Logo Qt *Creator* 26

Gambar 3.1 Ilustrasi sistem 28
Gambar 3.2 Diagram Blok Sistem 29
Gambar 3.3 GUI Utama 30
Gambar 3.4 GUI pendukung 30
Gambar 3.5 *Stereo Calibration* dan *Stereo Rectification* 31
Gambar 3.6 Diagram Blok Pengukuran Posisi Target 32
Gambar 3.7 Hasil *Remap* Citra 33
Gambar 3.8 Proses *Template Matchimg* 34
Gambar 3.9 Ilustrasi Konversi Koordinat Kartesian menjadi Sudut Motor 36
Gambar 3.10 Diagram Blok Pengarah Senjata 37
Gambar 3.11 *Driver* Motor BTN7970 39
Gambar 3.12 Motor DC Penggerak Vertikal 40
Gambar 3.13 Motor DC Penggerak Horisontal 40
Gambar 3.14 *Rotary Encoder* 41
Gambar 3.15 Sensor *Proximity* 41
Gambar 3.16 *Limit Switch* 42
Gambar 3.17 GUI kontrol *platform* senjata berdasarkan sudut horisontal dan vertikal 46

Gambar 3.18 GUI kontrol *platform* senjata berdasarkan posisi kartesian 46

Gambar 4.1 Hasil Kalibrasi yang Buruk..... 49
Gambar 4.2 Hasil Kalibrasi yang Cukup Baik 50
Gambar 4.3 Posisi Kalibrasi 50
Gambar 4.4 Pengukuran Target Jarak 20 cm..... 52
Gambar 4.5 Citra Kalibrasi Jarak 50 cm 54
Gambar 4.6 Citra Kalibrasi Jarak 100 cm 54
Gambar 4.7 Citra Kalibrasi Jarak 70 cm 54
Gambar 4.8 Target standar ISSF jarak 50 m untuk penembakan dengan pistol..... 64

DAFTAR TABEL

Tabel 4.1 Hasil Pengukuran Posisi dengan Fungsi OpenCV 52
Tabel 4.2 Hasil Pengukuran Posisi dengan Penghitungan Manual..... 53
Tabel 4.3 Data Titik-Titik Pola Papan Catur di Kamera Kiri 55
Tabel 4.4 Data Titik-Titik Pola Papan Catur di Kamera Kanan 57
Tabel 4.5 Hasil Pengukuran dengan Matriks Proyeksi 59
Tabel 4.6 Hasil Pengukuran Posisi dengan Jarak Kamera 10 cm..... 60
Tabel 4.7 Hasil Pengukuran Posisi dengan Jarak Kamera 20 cm..... 60
Tabel 4.8 Hasil Pengukuran Posisi dengan Jarak Kamera 30 cm..... 60
Tabel 4.9 Hasil Pengukuran Posisi dengan Jarak Kamera 40 cm..... 61
Tabel 4.10 Hasil Tuning PID untuk Motor Penggerak Horisontal 61
Tabel 4.11 Hasil Tuning PID untuk Motor Penggerak Vertikal..... 62
Tabel 4.12 Pengujian Pengarahan Senjata Melalui Komputer 62
Tabel 4.13 Pengujian Kontrol Platform Senjata Berdasarkan Posisi Kartesian 63
Tabel 4.14 Pengujian *Optical Flow* dan *Kalman Filter*..... 63
Tabel 4.15 Pengujian Pengarahan Target 65
Tabel 4.16 Pengujian Performa Sistem Malam Hari dengan Lampu ... 67
Tabel 4.17 Pengujian Performa Sistem Pagi Hari 70

BAB I
PENDAHULUAN

1.1. Latar Belakang

Otomatisasi telah menjadi hal yang terus dikembangkan dalam beberapa dekade ini seiring dengan perkembangan teknologi yang begitu cepat. Hal ini bertujuan untuk membantu bahkan untuk menggantikan peran manusia dalam melakukan sesuatu. Pengembangan otomatisasi juga terjadi di dunia militer. Penggunaan senjata dan alat-alat militer pada saat ini sudah mulai mengarah agar peran manusia dikurangi di dalamnya, walaupun masih dalam tahap semi otomatis. Salah satu contoh otomatisasi dalam dunia militer adalah pengarah senjata otomatis.

Otomatisasi senjata dalam beberapa aspek semakin diminati karena memiliki beberapa keuntungan. Beberapa keuntungan yang bisa didapatkan dalam penggunaan senjata otomatis di antaranya adalah mengurangi *human error* dan mengurangi korban manusia di sisi militer. Dengan berkurangnya *human error*, kejadian salah tembak atau jatuhnya korban yang tidak diinginkan dapat dicegah atau diminimalkan.

Selain itu, otomatisasi senjata juga dapat digunakan utnuk melawan beberapa macam bentuk terorisme, misalnya, penyanderaan oleh sekelompok teroris. Dengan menggunakan senjata yang telah diotomatisasi, pelumpuhan teroris dapat dengan cepat dilakukan dan dapat mengurangi kejadian salah tembak terhadap korban penyanderaan.

Mengarahkan senjata secara otomatis artinya senjata akan mengarahkan bidikannya ke arah target jika menerima perintah dari suatu *processing unit*. Beberapa metode telah dikembangkan untuk mengarahkan senjata secara otomatis, seperti pemasangan sensor-sensor untuk mendeteksi target. Namun, metode-metode tersebut akan membutuhkan biaya yang besar jika senjata yang dibutuhkan cukup banyak karena satu senjata membutuhkan satu sensor pendeteksi target.

Dalam mengarahkan senjata ke target, *processing unit* pada suatu sistem senjata otomatis hanya membutuhkan informasi posisi target terhadap senjata. Jika posisi target dalam koordinat kartesian bisa diketahui, *processing unit* bisa mengendalikan beberapa senjata sekaligus untuk mengarahkan senjata ke target. Dalam beberapa dekade ini, telah dikembangkan kamera stereo untuk mengukur posisi jarak objek dalam koordinat kartesian. Kamera stereo hanya membutuhkan dua kamera untuk memetakan posisi objek yang dipilih sebagai target. Dengan

informasi posisi tersebut, senjata dapat diarahkan secara otomatis dan sudut elevasi senjata untuk penembakan dapat ditentukan. Diharapkan agar sistem ini dapat menutupi kekurangan yang ada pada manusia untuk melakukan tugas serupa dan dapat mencegah timbulnya korban yang tidak diinginkan maupun terjadinya kesalahan yang terjadi akibat keterbatasan yang dimiliki oleh manusia.

1.2. Perumusan Masalah

Permasalahan yang dibahas dalam tugas akhir ini adalah:

1. Perancangan sistem pengarahan senjata untuk meringankan beban penjaga perbatasan.
2. Perancangan sistem senjata dan kamera yang terpisah untuk mengalihkan perhatian pelaku terror saat terjadi penyanderaan.
3. Pengukuran posisi target berdasarkan data yang didapatkan dari 2 kamera untuk pengarahan senjata secara otomatis.
4. Pengintegrasian perangkat lunak pengolahan citra dengan perangkat keras pengarah senjata.

1.3. Batasan Masalah

Batasan masalah dalam tugas akhir ini adalah

1. Sistem ini hanya diaplikasikan untuk sistem pertahanan statis.
2. Perancangan sistem ini menggunakan kamera yang hanya menerima panjang gelombang cahaya tampak.
3. Sistem ini bekerja pada lokasi dengan sistem pencahayaan yang tepat, tidak ada *backlight* dan tidak terlalu redup.
4. Motor untuk pergerakan vertikal hanya terbatas untuk elevasi sudut -25^0 sampai 25^0 karena keterbatasan sistem mekanik.
5. Jarak minimal target adalah 100 cm dan jarak maksimal target adalah 800 cm.
6. Target dipilih secara manual di *frame* kamera kiri.

1.4. Tujuan

Tujuan yang ingin dicapai dalam perancangan ini adalah:

1. Target yang dipilih pada satu kamera referensi bisa dikenali di kamera lain.
2. Sistem dapat mengukur posisi target terhadap senjata.
3. Sistem dapat mengarahkan senjata ke target secara otomatis.

4. Sistem dapat mengetahui sudut elevasi yang tepat agar penembakan oleh senjata dapat mencapai target.

1.5. Metodologi

Langkah-langkah yang dikerjakan pada tugas akhir ini adalah sebagai berikut:

1. **Studi Literatur**

   Pada tahap ini dilakukan pengumpulan dasar teori yang menunjang dalam penulisan Tugas Akhir. Dasar teori ini dapat diambil dari buku-buku, jurnal, dan artikel-artikel di internet dan forum-forum diskusi internet.

2. **Perancangan *Software***

   Software dirancang dengan pembuatan *source code* yang meliputi penentuan target, pencocokan target di kamera lain, pengukuran jarak target, dan perintah untuk mengarahkan senjata secara otomatis. Untuk pemilihan target, dilakukan oleh *user* secara *manual*. Untuk pencocokan target di kamera lain akan digunakan salah satu algoritma pencocokan objek pada dua citra. Selanjutnya, dilakukan pengukuran posisi target terhadap senjata dengan menggunakan metode triangulasi. Terakhir, *source code* yang dibuat akan mengirimkan perintah kepada *hardware* untuk bergerak sesuai dengan arah target.

3. **Perancangan *Hardware***

   Perancangan *hardware*, secara umum, meliputi kamera stereo dan senjata. Posisi kamera stereo dan senjata terpisah dengan jarak yang tetap. Kamera stereo terdiri dari dua kamera dengan jarak yang tetap. Posisi dua kamera harus berada pada level yang sama dan sebisa mungkin mengarah pada arah yang sejajar. Untuk senjata, digunakan dua motor agar senjata dapat digerakkan dalam dua derajat kebebasan. Dua kamera dan senjata tersebut dihubungkan dengan *processing unit* agar bisa dikontrol secara otomatis. Terakhir, *platform* dari sistem dibuat tetap.

4. **Pengujian Sistem**

   Pengujian alat dilakukan untuk menentukan keandalan dari sistem yang telah dirancang. Pengujian dilakukan untuk melihat apakah *software* dan *hardware* dapat bekerja secara baik.

Untuk pengujian dapat dilakukan dalam tiga tahap. Pertama adalah menggunakan citra stereo untuk menguji kemampuan pencocokan sistem terhadap dua citra. Kedua adalah pengujian sistem untuk pengukuran jarak. Dan terakhir adalah pengujian sistem untuk mengarahkan senjata kea rah target beserta sudut elevasinya.

5. **Analisa**

   Analisa dilakukan terhadap hasil dari pengujian sehingga dapat ditentukan karakteristik dari *software* dan *hardware* yang telah dibuat. Apabila karakteristik pencocokan target pada dua kamera dan pengarahan senjata dari *software* dan *hardware* yang telah dibuat masih belum sesuai, maka perlu dilakukan perancangan ulang pada sistem dan diuji kembali.

6. **Penyusunan Laporan Tugas Akhir**

   Tahap penulisan laporan tugas akhir adalah tahapan terakhir dari proses pengerjaan tugas akhir ini. Laporan tugas akhir berisi seluruh hal yang berkaitan dengan tugas akhir yang telah dikerjakan yaitu meliputi pendahuluan, tinjauan pustaka dan teori penunjang, perancangan sistem, pengujian, dan penutup.

1.6. Sistematika Penulisan

Dalam buku tugas akhir ini, pembahasan mengenai sistem yang dibuat terbagi menjadi lima bab dengan sistematika penulisan sebagai berikut:

- BAB I : PENDAHULUAN

  Bab ini meliputi penjelasan latar belakang, rumusan masalah, batasan masalah, tujuan, metodologi, sistematika penulisan, dan relevansi.

- BAB II : TINJAUAN PUSTAKA DAN TEORI PENUNJANG

  Bab ini menjelaskan tentang teori penunjang dan *literature* yang dibutuhkan dalam pengerjakan tugas akhir ini. Dasar teori yang menunjang meliputi citra digital, *stereo vision, stereo matching*, metode pengukuran triangulasi, *optical flow, pulse width modulation*, Mikrokontroler STM32F4, *driver* motor, dan Motor DC. Bagian ini memaparkan mengenai beberapa teori penunjang dan beberapa literatur yang berguna bagi pembuatan Tugas Akhir ini.

- BAB III : PERANCANGAN SISTEM

Bab ini menjelasakan tentang perencanaan sistem baik perangkat keras (*hardware*) maupun perangkat lunak (*software*) untuk sistem pengukuran posisi target dan pengarahan senjata secara otomatis.

- BAB IV : PENGUJIAN

  Pada bab ini akan menjelaskan hasil uji coba sistem beserta analisanya.
- BAB V : PENUTUP

  Bagian ini merupakan bagian akhir yang berisikan kesimpulan yang diperoleh dari pembuatan Tugas Akhir ini, serta saran-saran untuk pengembangan lebih lanjut.

1.7. Relevansi dan Manfaat

Hasil yang diharapkan dari tugas akhir ini diharapkan mampu meringankan pekerjaan manusia khususnya pada bidang militer khususnya untuk pertahanan statis dan juga penyelematan sandera oleh kelompok teroris. Pengembangan lebih lanjut dari sistem ini adalah penambahan motor penggerak pada *platform* sistem ini sehingga sistem ini dapat bergerak secara dinamis.

Halaman ini sengaja dikosongkan

BAB II
TINJAUAN PUSTAKA DAN TEORI PENUNJANG

Teori penunjang dalam bab ini menjelaskan tentang teori penunjang yang berhubungan dengan keseluruhan sistem yang akan dibuat pada tugas akhir ini. Sedangkan tinjauan pustaka dalam bab ini menjelaskan tentang sistem-sistem yang berhubungan dengan tugas akhir ini dan pernah diimplementasikan oleh penulis-penulis sebelumnya.

2.1. Citra Digital

Untuk menampilkan sebuah gambar pada layar-layar digital, gambar tersebut harus melalui proses digitalisasi citra untuk diubah menjadi citra digital. Proses-proses tersebut meliputi proses *sampling* dan proses *quantization*. Proses *sampling* membagi gambar menjadi beberapa kotak kecil. Proses *quantization* memberi nilai kecerahan dari setiap kotak tersebut. Nilai hasil *quantization* juga sering disebut nilai kedalaman.

Hasil dari proses digitalisasi citra adalah sebuah matriks yang berisi nilai-nilai riil. Elemen-elemen dari matriks tersebut disebut *picture element* atau sering disbut *pixel*. Sebuah *pixel* mempunyai dua property yaitu koordinat posisi dari *pixel* tersebut dan nilai dari *pixel* tersebut[1]. Sehingga, sebuah *pixel* dapat dinyatakan sebagai fungsi dua dimensi $f(x, y)$ dengan titik asal x dan y ada di pojok kiri atas dari citra. Misalkan, sebuah *pixel* dinyatakan $f(2, 3) = 5$, berarti *pixel* tersebut berada di posisi $x = 2$ dan $y = 3$ dari pojok kiri atas citra dengan tingkat kecerahan 5. Gambar 2.1 menunjukkan citra yang dilihat oleh manusia merupakan kumpulan matriks data yang dibaca oleh komputer. Dari gambar tersebut bisa dilihat bahwa, pada potongan citra bagian spion mobil memiliki nilai kecerahan sebesar 194 pada koordinat $x = 0$ dan $y = 0$. Sedangkan pada koordinat $x = 1$ dan $y = 0$, kecerahan citra adalah 210.

Kedalaman *pixel* sering disebut juga kedalaman warna. Citra digital yang memiliki kedalaman *pixel* n bit disebut juga citra n-bit. Dalam pemrosesan citra digital, citra digital bisa dibedakan berdasarkan batas nilai kecerahan atau nilai kedalaman dari suatu citra. Beberapa jenis citra berdasarkan kedalaman citra, di antaranya adalah citra warna atau sering disebut sebagai citra RGB (*Red Green Blue*), citra *grayscale*, dan citra biner.

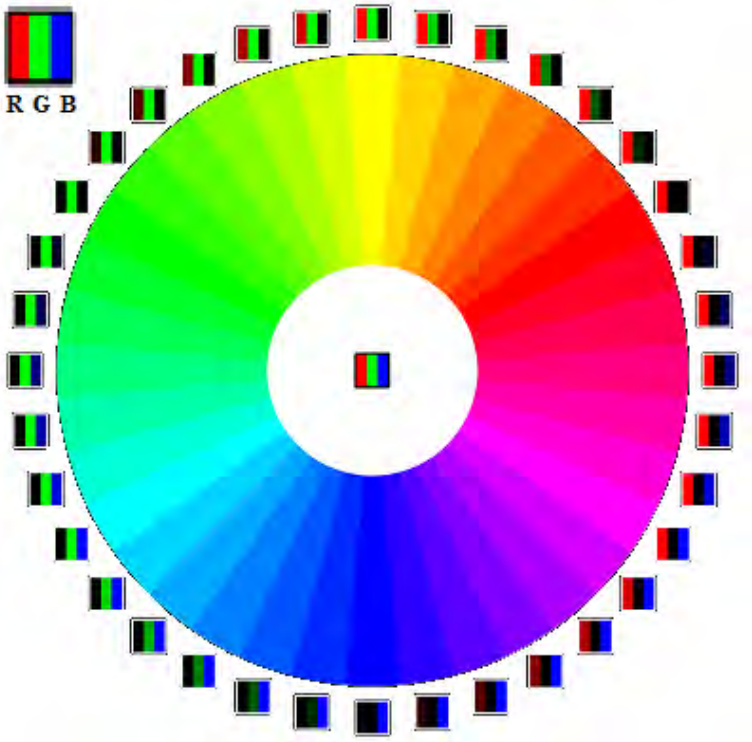
Gambar 2.1 Konsep *pixel*

2.1.1. Citra Warna

Citra warna adalah citra yang setiap *pixel*-nya ditentukan oleh tiga nilai masing-masing untuk komponen merah, biru, dan hijau[2]. Komponen warna lebih dikenal sebagai kanal. Sehingga, nilai dalam satu *pixel*-nya terdapat 3 kanal yang mewakili nilai dari warna merah, warna hijau, dan warna biru.

Kombinasi nilai dari ketiga warna tersebut akan membentuk suatu warna tertentu. Gambar 2.2 menunjukkan warna-warna yang dihasilkan dari kombinasi nilai warna merah, hijau, dan biru.

Nilai dari masing-masing komponen merah, hijau, dan biru memiliki rentang tertentu, tergantung tipe data yang dipakai. Jika tipe data tiap kanal warna tersebut berupa tipe data 8 bit, rentang nilai setiap kanal berada pada rentang nilai antara 0 sampai 255. Jika tipe datanya 16 bit, rentang nilai setiap kanal adalah antara 0 sampai 65535. Jumlah bit yang lebih banyak akan memberi variasi warna yang lebih banyak pula. Tipe data 8 bit per kanal akan memberi variasi warna sebanyak 16777216 warna, sedangkan tipe data 16 bit per kanal akan membentuk 281474976710656 jenis warna. Akan tetapi, dalam proses pengolahan citra, jumlah bit yang lebih sedikit memberi keuntungan pengolahan citra yang lebih cepat.



Gambar 2.2 Kombinasi warna RGB

2.1.2. Citra *Grayscale*

Citra *grayscale* disebut juga citra intensitas atau citra *gray level*[2]. Berbeda dengan citra warna yang terdiri dari tiga kanal warna, citra *grayscale* hanya terdiri satu kanal saja. Jika kanal tersebut merupakan data 8 bit, akan ada 256 *gray level*, sehingga setiap *pixel* akan memiliki nilai antara 0 sampai 255. Nilai 0 merupakan warna hitam dan 255 merupakan warna putih. Di antara 0 dan 255 merupakan warna abu-abu tergantung kecerahan dari *pixel* tersebut. Semakin tinggi nilainya, semakin cerah *pixel* tersebut. Gambar 2.3 menunjukkan rentang *gray level* dari hitam yang bernilai 0 hingga putih yang bernilai 255 dan identik dengan terang.

Gambar 2.3 Rentang *gray level*

Gambar 2.4 Citra RGB dan Citra *Grayscale*

Tipe citra ini sangat sering digunakan di dalam pengolahan citra. Hal ini dikarenakan jumlah datanya yang tidak terlalu besar yang akan mempercepat proses pengolahan data, tapi data *grayscale* tidak menghilangkan informasi penting dari citra. Oleh karena itu, jika citra asli yang didapatakan berupa citra warna dan akan dilakukan proses pengolahan citra, citra tersebut biasanya dikonversi dahulu ke citra *grayscale* dengan cara mencari nilai rata-rata dari nilai-nilai dari kanal-kanal merah, hijau, dan biru. Gambar 2.4 menunjukkan perbandingan citra warna dan citra *grayscale*.

2.1.3. Citra Biner

Citra biner merupakan *array* logika yang hanya berisi 0 dan 1, yang diartikan hitam dan putih[2]. Untuk citra biner yang normal, objek biasanya ditandai dengan warna hitam. Sedangkan warna putih menunjukkan *background*.

Citra biner berasal dari citra *grayscale* yang di-*threshold*. Sebagai contoh, suatu citra *grayscale* di-*threshold* menggunakan

Gambar 2.5 Citra Biner

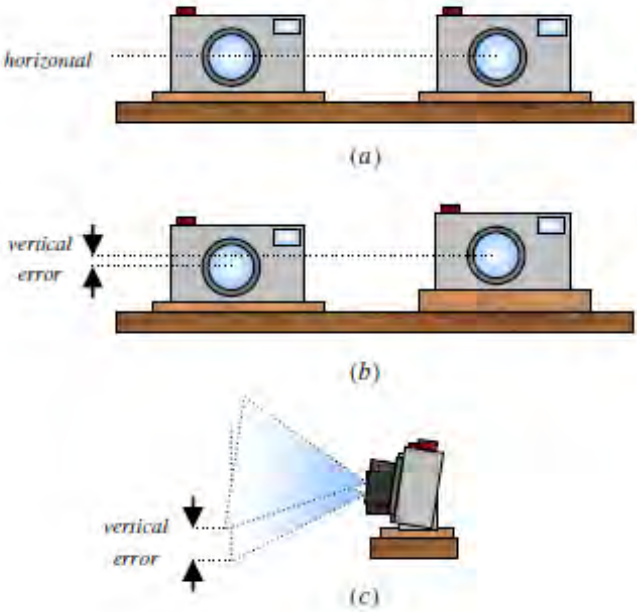
nilai *threshold* 100, maka *pixel* yang memiliki nilai *gray level* kurang dari 100 akan diberi nilai 0 (warna hitam), sedangkan *pixel* yang memiliki nilai *gray level* di atas 100 akan diberi nilai 255 (warna putih). Gambar 2.5 menunjukkan hasil konversi menjadi citra biner.

Walaupun citra warna lebih disukai banyak orang, namun citra biner tetap dipakai, terutama dalam pemrosesan citra. Hal ini dikarenakan kebutuhan memori citra biner kecil sehingga waktu pemrosesan lebih cepat dibandingkan dengan citra *grayscale* ataupun warna.

2.2. *Stereo Vision*

Stereo vision merupakan cabang penting dalam penilitian di area *computer vision*[3]. *Stereo vision* menggunakan dua citra yang diambil dalam waktu yang hampir bersamaan pada suatu area dana arah yang sama. Dengan menggunakan *stereo vision*, jarak objek-objek yang ada di kedua citra bisa diukur dengan menggunakan beberapa metode pengukuran jarak yang berdasarkan *stereo vision*, seperti metode triangulasi. Untuk melakukan pengukuran jarak objek menggunakan metode triangulasi, ada syarat yang harus dipenuhi, yaitu

- Kamera harus benar-benar sejajar dan mengarah pada arah yang sama.seperti pada Gambar 2.6.
- Pengambilan dua citra harus bersamaan.



Gambar 2.6 Peletakan kamera stereo yang sejajar (a) dan peletakan kamera stereo dengan kesalahan vertikal (b) dan (c)[4]

Peletakan dua kamera yang tidak sejajar akan menyebabkan posisi horisontal dari dua citra yang diambil oleh dua kamera tersebut tidak sama. Hal ini akan menyebabkan proses pengukuran jarak dengan metode triangulasi tidak dapat dilakukan. Namun, pada praktiknya, peletakan dua kamera agar benar-benar sejajar memang sangat sulit untuk dilakukan.

Untuk mengatasi hal tersebut, dibutuhkan beberapa proses komputasi yang dapat membuat posisi horisontal dari dua citra menjadi sama. Proses-proses yang harus dilakukan sebelum melakukan pengukuran jarak dengan metode triangulasi adalah *Stereo Calibration* dan *Stereo Rectification*

.

2.2.1. *Stereo Calibration*

Stereo Calibration merupakan proses penghitungan hubungan geometris dari dua kamera di suatu ruang[5]. Hasil dari

stereo calibration adalah parameter-parameter intrinsik dan ektrinsik dari dua kamera tersebut. Parameter intrinsik merupakan parameter yang mengoreksi kesalahan-kesalahan yang dapat dimunculkan oleh kondisi fisik internal dari kamera, seperti distorsi akibat pemasangan lensa di dalam kamera atau distorsi akibat kecacatan lensa kamera itu sendiri. Sedangkan parameter ekstrinsik mengoreksi ketidak sejajaran dari dua kamera yang berupa kesalahan translasi maupun kesalahan rotasi.

Untuk melakukan proses ini, dibutuhkan citra suatu objek yang sudah diketahui ukuran 3D-nya yang diambil oleh dua kamera tersebut dalam beberapa pose. Objek yang biasa dipakai adalah objek-objek yang berpola dengan jarak yang tetap, seperti papan catur. Titik-titik yang ada di papan catur akan diambil koordinatnya dan dilakukan beberapa proses transformasi matematika.

Seperti yang telah dijelaskan sebelumya, *stereo calibration* akan menghasilkan parameter eksterinsik dari sistem ini. Parameter-parameter tersebut berupa matriks rotasi (R_1) dan translasi (T_1) dari kamera 1 serta matriks rotasi (R_2) dan translasi (T_2) dari kamera 2. Jika koordinat 3D dari suatu titik diketahui (hasil ekstrak data titik di papan catur) yaitu P dan proyeksi titik tersebut di kamera 1 adalah P_1 serta proyeksi titik tersebut di kamera 2 adalah P_2, hubungan P_1 dan P_2 terhadap P adalah sesuai pada persamaan (2.1) dan (2.2).

$$P_1 = R_1 P + T_1 \quad (2.1)$$

$$P_2 = R_2 P + T_2 \quad (2.2)$$

Titik P pada dua kamera bisa ditransformasi dengan persamaan (2.3).

$$P_1 = R^T (P_2 - T) \quad (2.3)$$

dengan R dan T merupakan matriks rotasi dan vektor translasi antara dua kamera. Dari persamaan-persamaan di atas bisa diperoleh matriks R dan T, yaitu tertulis seperti pada persamaan (2.4) dan persamaan (2.5)

$$R = R_2 (R_1) T \quad (2.4)$$

$$T = T_2 - R T_1 \quad (2.5)$$

Dengan demikian, *stereo calibration* terselesaikan melalui proses yang telah dijelaskan di atas.

2.2.2. *Stereo Rectification*

Telah dijelaskan sebelumnya, proses penghitungan triangulasi akan lebih mudah jika citra dua gambar diambil dari dua kamera yang benar-benar sejajar. Namun, konfigurasi yang sangat ideal tersebut hampir tidak mungkin dicapai. Karena dibutuhkan citra yang sejajar secara horisontal, maka diperlukan reproyeksi kedua citra. Dengan demikian, proses penghitungan jarak dengan metode triangluasi lebih mudah dilakukan.

Ada banyak metode *rectification*, namun ada dua metode yang terkenal. Yang pertama adalah algortima Hartley yang menggunakan matriks dasar dari citra sehingga tidak perlu dilakukan *stereo calibration*. Metode kedua adalah algoritma Bouguet yang menggunakan parameter rotasi dan translasi dari kamera yang telah terkalibrasi

Tujuan dari algoritma Hartley adalah mencari matriks homografi yang memetakan garis epipolar menuju ke titik tak hingga. Sehingga, metode ini hanya menghitung matriks fundamental. Keuntungan dari algoritma ini adalah proses *stereo calibration* dilakukan secara *real time*. Namun, metode ini tidak dapat mendeteksi adanya penskalaan citra.

Sedangkan algoritma Bouguet secara sederhana adalah meminimalkan proyeksi ulang dari dua citra stereo. Dengan demikian, sudut pandang yang lebih luas bisa diperoleh.

Masing-masing metode memiliki kelebihan dan kekurang masing-masing. Jika kamera stereo sudah dikalibrasi, maka lebih baik menggunakan algoritma Bouguet. Namun, jika belum dikalibrasi dan butuh metode yang *real time*, algoritma Hartley menjadi metode yang lebih baik untuk digunakan.

Setelah proses *stereo rectification*, citra akan menjadi sejajar pada bagian horisontal. Selain itu, matriks pemetaan untuk proses *rectification* didapatkan. Tanpa mengganti posisi relative kedua kamera serta parameter intrinsik dari keduanya, proses pengukuran jarak menggunakan metode triangulasi bisa didapatkan. Walaupun citra baru dari dua kamera akan mengganti citra yang lama, citra kedua kamera akan tetap horisontal karena kedua citra yang baru bisa dipetakan menggunakan matriks pemetaan yang telah didapatkan setelah proses *stereo rectification*.

2.3. *Stereo Matching*

Stereo matching, mencocokkan sebuah titik 3D di dua kamera dengan sudut pandang berbeda, bisa dihitung hanya jika terdapat area pandang dari dua kamera tersebut yang *overlap*[3]. Jika objek di kamera kiri ditentukan, dengan *stereo matching*, objek yang sama yang terekam oleh kamera kanan akan ditemukan.

Pada dasarnya, s*tereo matching* menggunakan metode-metode yang ada di *template matching*. *Template matching* tidak berdasarkan pada histogram, namun lebih kepada pencocokan potongan gambar terhadap suatu citra masukan yang digeser-geser dan dicocokkan dengan metode-metode tertentu[6]. *Template matching* adalah sebuah masalah pada pengolahan citra untuk mencari lokasi dari suatuobjek menggunakan gambar template kemudian mencari gambar *template* tersebut pada gambar lainnya ketika posisinya tidak diketahui.

Template matching secara *inheren* merupakan masalah yang berat dikarenakan masalah kecepatan dan kehandalannya. Solusi yang dihasilkan harus kuat terhadap perubahan kecerahan ketika suatu objek sebagian terlihat atau dicampur dengan benda lain. Dan hal yang paling penting adalah algoritma yang dimiliki harus dapat dikomputasi secara efisien. Terdapat dua jenis pendekatan untuk memecahkan masalah *template matching*. Pertama adalah berdasarkan nilai abu-abu (*grey value based matching*) atau pencocokan berdasarkan area dan pencocokan berbasis fitur (*feature based matching*) atau tidak berdasarkan *area*.

Pada pendekatan berbasis *grey value*, algoritma *Normalized Cross Correlation (NCC)* telah diketahui sebelumnya. Hal ini biasanya dilakukan pada setiap langkah dengan mengurangi rata-rata dan membaginya dengan deviasi standar. Korelasi silang dari *template* t(x,y) dengan gambar sub f(x,y) seperti yang tertulis pada persamaan (2.6).

$$NCC = \frac{1}{n-1}\sum_{x,y}\frac{(f(x,y)-\mu f)(t(x,y)-\mu t)}{\sigma f.\sigma t} \qquad (2.6)$$

Dimana n adalah angka nilai pixel di t(x,y) dan f(x,y). Meskipun metode ini kuat menghadapi iluminasi perubahan linear, algoritma ini akan gagal ketika objek terlihat secara parsial atau objek tersebut bercampur dengan objek lainnya. Lebih dari itu, algoritma ini secara komputasi mahal karena membutuhkan komputasi untuk korelasi antara semua *pixel* pada gambar *template* dan untuk mencari gambar.

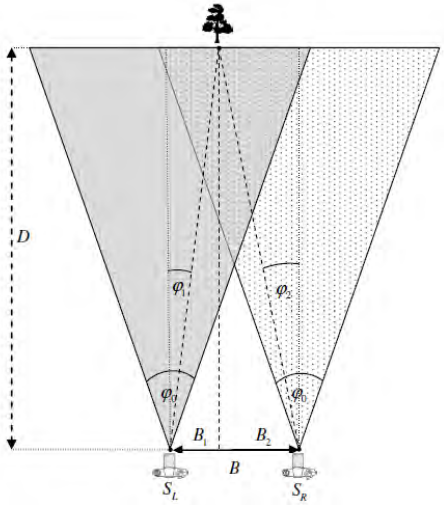
2.4. Metode Pengukuran Triangulasi

Metode ini dikenal juga dengan istilah *stereoscopy*. *Stereoscopy* merupakan teknik yang digunakan untuk merekam dan merepresentasikan citra *stereoscopic* (3D)[4]. Dari teknik ini, citra stereo dapat diolah untuk menentukan kedalaman ataupun jarak dari suatu objek yang ada di dalam citra.

Citra stereo merupakan sepasang citra atau lebih dari satu objek dengan sudut pengambilan gambar yang berbeda. Agar bisa mendapatkan citra *stereoscopic* yang akurat, biasanya digunakan dua kamera atau lebih dari posisi yang berbeda untuk mengambil citra objek dalam waktu yang hampir bersamaan. Khusus untuk dua kamera, seperti yang telah dijelaskan sebelumnya, keakuratan jarak terukur dari citra *stereoscopic* juga ditentukan oleh peletakan kamera yang harus terpasang sejajar secara horisontal. Berikut ini adalah penjelasan bagaimana cara mendapatkan jarak objek hanya dengan menggunakan citra objek yang diambil oleh dua kamera dengan sudut pandang yang berbeda.

Dari gambar 2.7, hubungan B_1, B_2, dan jarak kamera (B) seperti pada persamaan (2.7)

$$B = B_1 + B_2 \tag{2.7}$$



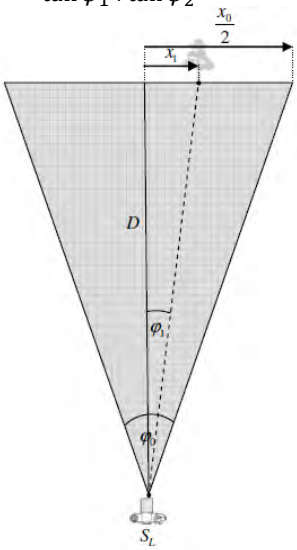
Gambar 2.7 Proses Pengambilan Citra dari Dua Kamera[4]

Dengan demikian, hubungan antara jarak objek (D) dengan jarak kamera ada pada persamaan (2.8), (2.9), dan (2.10).

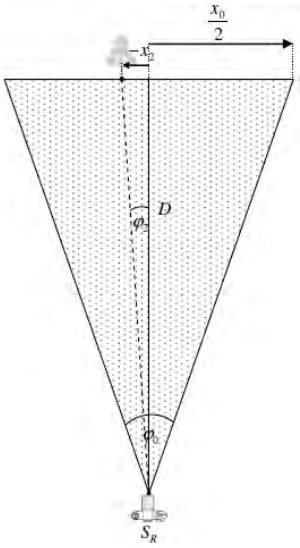
$$B = D \tan \varphi_1 + D \tan \varphi_2 \quad (2.8)$$

$$B = D(\tan \varphi_1 + \tan \varphi_2) \quad (2.9)$$

$$D = \frac{B}{\tan \varphi_1 + \tan \varphi_2} \quad (2.10)$$



Gambar 2.8 Pengambilan citra oleh kamera kiri[4]



Gambar 2.9 Pengambilan citra oleh kamera kanan[4]

Dengan menggunakan trigonometri untuk citra yang diambil oleh kamera kiri dan kanan seperti yang ada pada Gambar 2.8 dan Gambar 2.9, bisa didapatkan hubungan seperti pada persamaan (2.11) dan (2.12) berikut:

$$\frac{x_1}{\frac{x_0}{2}} = \frac{\tan \varphi_1}{\tan \frac{\varphi_0}{2}} \tag{2.11}$$

$$-\frac{x_2}{\frac{x_0}{2}} = \frac{\tan \varphi_2}{\tan \frac{\varphi_0}{2}} \tag{2.12}$$

Sehingga, jarak D bisa didapatkan, seperti pada persamaan (2.13).

$$D = \frac{B x_0}{2 \tan\left(\frac{\varphi_0}{2}\right)(x_L - x_D)} \tag{2.13}$$

Dari persamaan di atas, B merupakan jarak antar kamera, x_0 merupakan jumlah *pixel* horisontal, φ_0 merupakan *field of view* dari kamera dan (x_L-x_D) merupakan perbedaan *pixel* objek pada dua citra. Data-data tersebut bisa diukur, sehigga jarak D bisa diperoleh.

Namun, di dalam pengukuran, nilai x_0 dan $\tan\left(\frac{\varphi_0}{2}\right)$ memiliki hubungan seperti pada persamaan (2.14).

$$f = \frac{x_0}{2\tan\left(\frac{\varphi_0}{2}\right)} \tag{2.14}$$

Dengan f merupakan panjang fokus kamera.

Sehingga, nilai D bisa didapatkan dari persamaan (2.15).

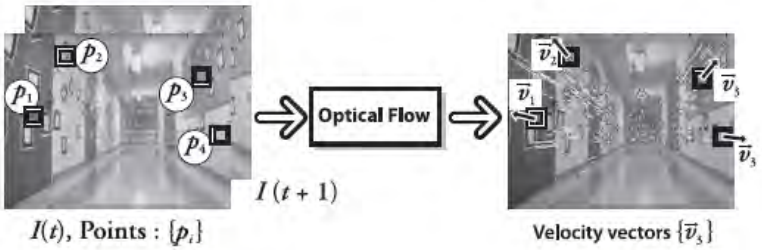
$$D = \frac{fB}{x_L - x_D} \tag{2.15}$$

2.5. *Optical Flow*

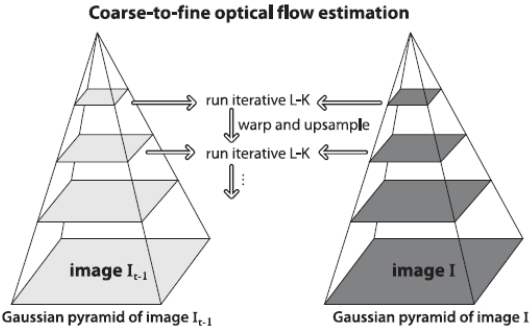
Optical Flow memungkinkan untuk memperkirakan gerak antara dua *frame* (atau sederet *frame*) tanpa pengetahuan sebelumnya tentang isi *frame-frame* tersebut. Biasanya, pergerakan itu sendiri lah yang menjadi hal yang menarik di *frame-frame* tersebut. *Optical flow* bisa diilustrasikan seperti pada Gambar 2.10.

Pada gambar pojok kanan atas di Gambar 2.10 menunjukkan fitur target yang diikuti setiap waktu. *Optical flow* mendeteksi pergerakan fitur-fitur tersebut dan mengubahnya menjadi sebuah vector kecepatan dari setiap fitur, seperti pada gambar pojok kiri atas yang ada di Gambar 2.10. Dengan demikian, objek-objek yang menjadi target tersebut bisa diikuti selama ada di dalam *frame*.

Ada beberapa algoritma *tracking* objek yang berdasar pada *optical flow*. Salah satunya adalah metode Lucas Kanade *Optical Flow*. Metode Lucas Kanade *Optical Flow* ini pun mengalami pengembangan menjadi *Pyramidal* Lucas Kanade *Optical Flow*.



Gambar 2.10 Ilustrasi *Optical Flow* [7]



Gambar 2.11 Algoritma *Pyramidal Lucas Kanade*[7]

Metode Lucas-Kanade merupakan metode turunan dari proses estimasi *optical flow* yang diajukan oleh Bruce D. Lucas dan Takeo Kanade. Metode ini mendeteksi pergerakan "kecil" dengan asumsi titik-titik sekitarnya koheren. Sehingga, Metode Lucas-Kanade bisa digunakan untuk *window* berukuran kecil.

Sedangkan pada penggunaan *window* berukuran besar, gerakan-gerakan yang ditangkap sering tidak koheren. Sehingga, merode Lukas-Kanade tidak dapat menangkap dan memperhitungkan pergerakan yang "besar". Pada tahun 2000, Jean Yves Bouguet mengajukan metode *Pyramidal* Lucas-Kanade untuk menyelesaikan keterbatasan dari metode Lucas-Kanade konvensional.

Metode *Pyramidal* Lucas-Kanade menggunakan prinsip piramida, yaitu detail citra paling rendah lalu terus diiterasi hingga detail citra yang paling tinggi. Metode ini menggunakan implementasi iteratif dari komputasi *optical flow* Lucas-Kanade yang memberikan akurasi *tracking* yang cukup. Hal ini seperti yang ditunjukkan pada Gambar 2.11.

Bouguet menjelaskan algoritma *Pyramidal* Lucas-Kanade di dalam makalahnya seperti penjelasan berikut ini. Anggap I dan J merupakan citra *grayscale* 2D. I(x) = I(x,y) dan J(x) = J(x,y) merupakan nilai *grayscale* dari dua citra tersebut dengan lokasi x=$[x\ y]^T$, dengan x dan y merupakan koordinat dari *pixel* dari titik x. Citra I menjadi referensi pertama dan J sebagai referensi kedua[8].

Misalkan terdapat suatu titik citra u = $[u_x\ u_y]^T$ pada citra I. Tujuan dari *tracking* adalah mencari lokasi v = u + d = $[u_x+d_x\ \ u_y+d_y]^T$ pada cita J dimana I(u) dan J(v) hampir sama. Vektor d = $[d_x\ d_y]^T$ merupakan kecepatan citra di titik x, yang juga dikenal dengan *optical flow* di titik x.

Karena masalah *aparture*, penting untuk mendefinisikan dugaan kemiripan pada titik citra 2D. Misalkan w_x dan w_y merupakan bilangan bulat. Kita mendefinisikan kecepatan citra d sebagai vektor yang meminimumkan fungsi residual ε seperti yang didefinisikan pada persamaan (2.16) sebagai berikut[8]:

$$\varepsilon(d) = \varepsilon(d_x, d_y) = \sum_{x-u_x-w_x}^{u_x+w_x} \sum_{y-u_y-w_y}^{u_y+w_y} \left(I(x,y) - J(x+d_x, y+d_y)\right)^2 \quad (2.16)$$

2.6. *Kalman filter*

Kalman filter muncul untuk kepentingan dalam konteks pemrosesan sinyal dengan perubahan yang besar[7]. *Kalman filter* sering dianggap sebagai *estimator*. Hal ini dikarenakan penggunaan *filter* ini sering digunakan dalam pengestimasian keluaran sensor atau pengukuran yang memiliki *noise* yang sangat besar. Penamaan *filter* dalam *estimator* ini dikarenakan pem-*filter*-an *noise* dari sensor atau hasil pengukuran.

Terdapat dua variabel yang penting dalam penggunaan *kalman filter*. Yang pertama adalah *Kalman gain* dan *prediction error*. *Kalman gain* merupakan variabel yang di-*update* dalam satu siklus yang mempengaruhi nilai baru hasil estimasi *kalman filter*. Sedangkan *prediction error* merupakan sebuah variabel yang mempengaruhi *update* dari *Kalman gain*. Dalam satu siklus, *prediction error* juga di-*update* berdasarkan karakteristik dari pengukuran.

Dalam menggunakan *kalman filter*, terdapat dua langkah utama yang ada di dalamnya, yaitu prediksi atau estimasi dan *update* nilai. Proses prediksi meliputi persamaan (2.17) dan (2.18).

$$x_k = a\, x_{k-1} \quad (2.17)$$

$$p_k = a^2 p_{k-1} \quad (2.18)$$

x merupakan nilai hasil estimasi *kalman filter*, a merupakan karakteristik dari sistem. Dalam pembacaan sensor atau pengukuran nilai, niali a adalah 1. Nilai p merupakan $prediction\ error$, indeks k merupakan kondisi saat ini, dan $k-1$ merupakan kondisi sebelumnya. Sedangkan pada proses *update*, nilai pengukuran sensor saat itu mengoreksi nilai hasil prediksi dari *kalman filter* ini. Proses ini meliputi persamaan (2.19), (2.20), dan (2.21).

$$g_k = \frac{p_k}{p_k + r} \quad (2.19)$$

$$x_k = x_k + g_k(z_k - x_k) \quad (2.20)$$

$$p_k = (1 - g_k)p_k \quad (2.21)$$

Dengan nilai g merupakan *Kalman gain*.

2.7. *Pulse Width Modulation*

Dasar *pulse width modulation* (PWM) secara luas digunakan di dalam aplikasi elektronika daya untuk pengaturan pengkonversian daya (DC/DC, DC/AC, dll.)[9]. Secara sederhana, PWM merupakan sinyal yang lebar pulsa yang bernilai "HIGH" dalam satu periode mewakili suatu tegangan DC, tergantung nilai *duty cycle*. *Duty cycle* merupakan perbandingan lama waktu sinyal bernilai "HIGH" dengan satu periode. Gambar 2.12 menggambarkan *duty cycle* dari PWM.

Selain digunakan dalam pengaturan pengkonversian daya, ada beberapa aplikasi lain dari PWM. Contoh aplikasi umum yang lain adalah pengendalian kecepatan motor DC, pengendalian motor servo, pengaturan nyala terang LED dan lain sebagainya.

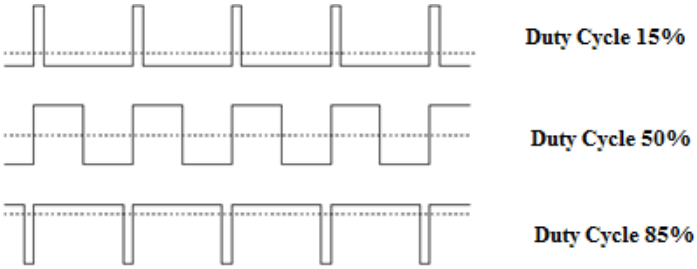
2.8. Mikrokontroler STM32F4

Mikrokontroler STM32F4 termasuk dalam keluarga mikrokontroler 32 bit dengan arsitektur ARM. Mikrokontroler jenis STM32F4 merupakan seri pertama dalam kelompok STM32 dengan ARM core-M4F. Seri F4 juga merupakan seri pertama dari STM32 yang dapat melakukan *digital signal processing* (DSP). Seri STM32F4 memiliki pin-pin yang sama dengan pin-pin STM32F2 dengan penambahan kecepatan clock yang lebih tinggi dari STM32F2, 64K CCM static RAM, full duplex I^2S, dan memiliki kecepatan konversi ADC.

Berdasarkan chip STM32F407VGT6, *board* mikrokontroler ini memiliki *ST-LINK/V2 embedded debug tool*, accelerometer digital, mikrofon digital, satu *audio DAC* dengan driver speaker *class* D terintegrasi, beberapa LED dan *push button*s dan sebuah konektor USB OTG micro-AB. Gambar 2.13 merupakan gambar dari *Board Stm32f4-Discovery* yang sudah dilengkapi dengan *downloader* tipe ST-Link untuk memasukkan program dari komputer ke mikrokontroler STM32F4.

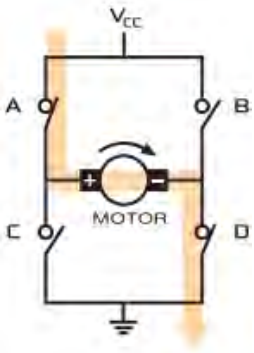
2.9. *Driver* motor

Driver motor merupakan rangkaian yang memperkuat arus yang akan masuk ke motor dari pengontrol atau mikrokontroler sehingga arus yang masuk ke motor cukup besar untuk menggerakkan motor. Rangkaian motor yang sering dipakai adalah rangkaian *H-Bridge*.



Gambar 2.12 Duty cycle pada PWM

Gambar 2.13 *Board* STM32F4-*Discovery*[10]



Gambar 2.14 Skematik sederhana *H-Bridge*[11]

Rangkaian *H-Bridge* namanya berasal dari rangkaian *full-bridge* yang ditunjukkan oleh gambar 2.14[11]. Dengan rangkaian ini, arah gerak motor bisa diatur. Jika saklar A dan saklar D disambungkan dan saklar B dan saklar C tidak disambungkan, arus akan mengalir dari tegangan sumber ke sisi kiri motor. Hal ini akan membuat gerakan motor searah jarum jam. Sebaliknya, jika saklar yang tersambung adalah saklar B dan saklar C, arus akan mengalir dari tegangan sumber ke sisi kanan motor. Hal ini akan membuat motor bergerak berlawanan arah jarum jam.

2.10. Kontrol PID

Kontroler *Proportional-Integral-Derivative* (PID) secara luas digunakan di dalam sistem pengaturan industri karena pengurangan jumlah parameter-parameter yang akan di-*tuning*[13]. Kontrol PID menghasilkan sinyal kontrol yang sebanding dengan sinyal *error* (aksi proporsional), sebanding dengan total sinyal *error* (aksi integral), dan sebanding dengan turunan dari kesalahan yang sekarang dengan kesalahan yang sebelumnya (aksi *derivative*). Sinyal *error* merupakan selisih antara *set point* dengan nilai keluaran aktual. Kontrol PID dapat ditulis menjadi persamaan (2.22) berikut ini,

$$u(t) = Kp.e(t) + Kd.\frac{d}{dt}e(t) + Ki.\int e(t)dt \qquad (2.22)$$

dengan $u(t)$ merupakan sinyal kontrol PID dan $e(t)$ merupakan sinyal *error*. Kp, Ki, dan Kd masing-masng merupakan koefisien Untuk sistem diskrit, persamaan (2.23) di atas dimodifikasi menjadi,

$$u[n] = Kp.e[n] + Kd.(e[n] - e[n-1]) + Ki.\sum e[n] \quad (2.23)$$

Nilai Kp, Ki, dan Kd pada Tugas Akhir ini akan dicari menggunakan *tuning* manual. Sinyal kontrol yang akan dihasilkan berupa PWM untuk menggerakkan motor agar mengarah ke posisi yang diinginkan.

2.11. Komunikasi Serial

Salah satu hal terpenting ketika menggunakan dua *processor* dalam satu sistem adalah komunikasi serial. Komunikasi serial memungkinkan dua *processor* untuk saling bertukar data hanya dengan menggunakan satu kawat konduktor saja. Hal ini dikarenakan data yang dikirimkan berupa deretan bit dari data yang dikirimkan.

Komunikasi serial terbagi menjadi dua jenis, yaitu *universal asynchronous receiver-transmitter* (UART) dan *universal synchronous*

asynchronous receiver-transmitter (USART)[14]. USART dapat melakukan komunikasi asinkron dan sinkron, sedangkan UART hanya dapat melakukan komunikasi asinkron saja. Komunikasi sinkron mengharuskan dua divais yang berkomunikasi harus memiliki sistem *clock* yang sama. Sedangkan komunikasi asinkron tidak mengharuskan sistem *clock* dari dua divais berasal dari sistem *clock* yang sama.

2.12. OpenCV

OpenCV merupakan suatu *library* dari *computer vision* yang *open source* (gratis digunakan baik untuk urusan akademik ataupun komersil) dan tersedia di *http://SourceForge.net/projects/opencvlibrary*[15]. *Library* ini bisa digunakan dalam bahasa C, C++, Python, dan beberapa bahasa pemrograman lain. Dengan *library* OpenCV, program yang dibuat bisa digunakan untuk mengambil citra hingga mengolah citra. *Library* ini juga memungkinkan *programmer* untuk melakukan manipulasi video ataupun *real-time* video. Gambar 2.15 merupakan logo OpenCV yang terdapat di situs OpenCV yaitu *http://opencv.org*. Di dalam situs tersebut terdapat pula hasil dokumentasi beberapa orang tentang pengolahan citra menggunakan OpenCV.

2.13. Qt Creator

Qt merupakan *framework* C++ yang komperhensif untuk mengembangkan aplikasi GUI (*Graphical User Interface*) lintas-*platform* menggunakan pendekatan "tulis sekali, *compile* di mana saja"[17]. Dengan Qt, *programmer* yang sudah membuat program *user interface* di *platform* Windows bisa meng-*compile* program tersebut di Linux. Logo dari Qt ada pada Gambar 2.16.

Gambar 2.15 Logo OpenCV[16]

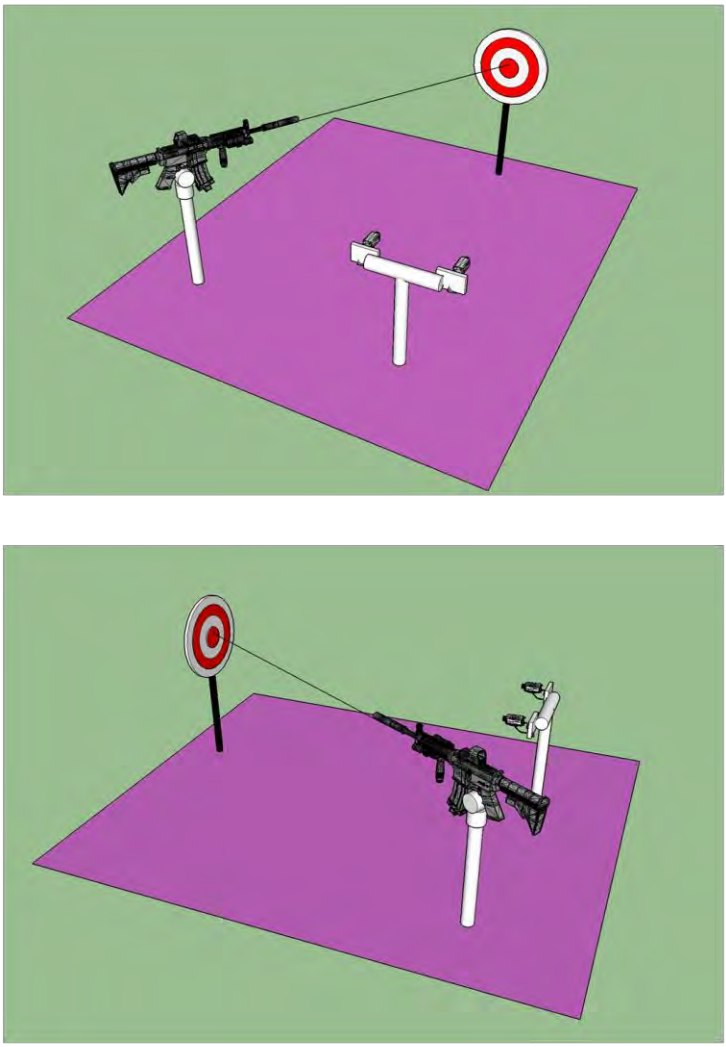
Gambar 2.16 Logo Qt Creator

BAB III
PERANCANGAN SISTEM

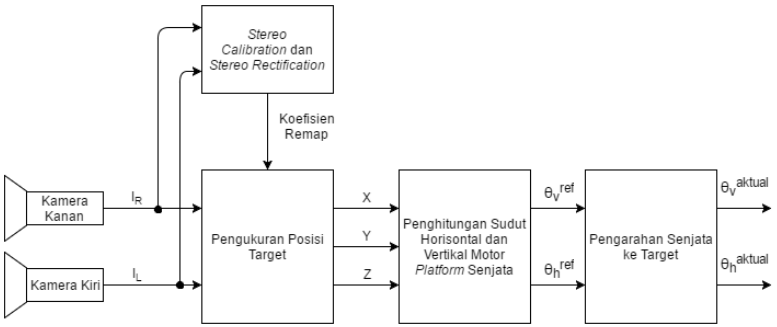
Pada bab ini akan dijelaskan perancangan sistem mulai dari perancagan *software* yang meliputi perancangan pengukuran jarak dan perancangan GUI hingga perancangan *hardware* yang meliputi perancangan senjata beserta perancangan gerak senjata baik gerak vertikal dan juga gerak horisontal. Dalam bab ini juga akan dijelaskan bagaimana proses komunikasi antara *software* yang dibuat di computer dengan *software* yang ada di mikrokontroler yang akan mengontrol pergerakan motor pada *platform* senjata.

Pada tugas akhir ini, teknologi pengolahan citra digunakan untuk mengetahui posisi target terhadap kamera. Dengan informasi target tersebut, senjata yang diletakkan pada suatu *platform* yang dapat dikontrol oleh mikrokontroler dapat digerakkan untuk mengarah pada target tersebut. Pada sistem ini, dibutuhkan *operator* yang bertugas untuk memilih target dan mengarahkan senjata sesuai dengan target yang dipilih oleh *operator*. Gambar 3.1 mengilustrasikan sistem yang akan dibuat. Seperti yang terlihat pada gambar tersebut, *platform* senjata dan kamera dibuat terpisah. *Platform* kamera dibuat tetap, sedangkan *platform* senjata bisa berotasi secara horisontal dan juga vertikal dengan batasan sudut tertentu. Dari gambar tersebut juga bisa dilihat bahwa senjata mengarah pada target yang berada di daerah pengamatan kamera yang diam tersebut.

Dalam perencanaannya, sistem ini seharusnya diletakkan untuk mempertahankan suatu daerah yang akan dipertahankan. Peletakan *platform* sistem ini statis secara translasi, artinya baik senjata maupun kamera dibuat tetap di suatu tempat. Sehingga, sistem ini disebut juga sebagai sistem pertahanan statis atau titik. Sistem ini lebih difungsikan sebagai *observer* bagi *operator*, sehingga senjata tidak akan bergerak jika tidak diperintahkan oleh *operator*. Saat dioperasikan, sistem ini akan menampilkan citra dari kamera kanan dan kamera kiri. Saat *operator* mengamati dari kamera tersebut ada sesuatu yang mencurigakan, *operator* dapat menarget objek tersebut dengan cara memilih objek di citra kamera kiri. Kemudian, operator dapat mengaktifkan senjata agar mengarah ke target tersebut. Jika target bergerak, senjata akan selalu mengikuti target yang telah dipilih. Sistem ini juga akan menampilkan posisi target ada pada koordinat berapa terhadap sistem kamera.



Gambar 3.1 Ilustrasi sistem



Gambar 3.2 Diagram Blok Sistem

Diagram blok dari sistem ini ditunjukkan seperti pada Gambar 3.2. Secara garis besar, sistem terbagi menjadi empat sub sistem, yaitu *Stereo Calibration* dan *Stereo Rectification*, Pengukuran Posisi Target, Penghitungan Sudut Horisontal dan Vertikan Motor Platform Senjata, dan Pengarahan Senjata ke Target. Bagian pengarahan senjtata ke target dikerjakan oleh mikrokontroler STM32F4. Selain bagian tersebut, semua proses dilakukan di komputer. Sehingga, dibutuhkan komunikasi serial antara komputer dan mikrokontroler STM32F4 untuk mengirimkan data berupa sudut horisontal dan vertikal referensi untuk motor penggerak senjata.

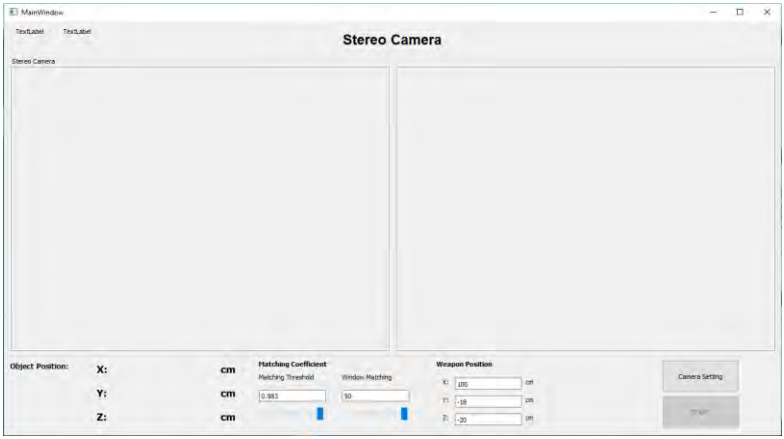
Terdapat beberapa fitur tambahan yang dirancang dalam sistem ini. Untuk mempermudah kerja operator, *Graphical User Interface* (GUI) dibuat untuk sistem ini dengan menggunakan *Qt Creator*. Selain itu, terdapat fitur tambahan berupa *tracking* dengan metode *optical flow* yang membuat senjata dapat mengikuti target yang telah dipilih.

3.1. *Graphical User Interface* Sistem

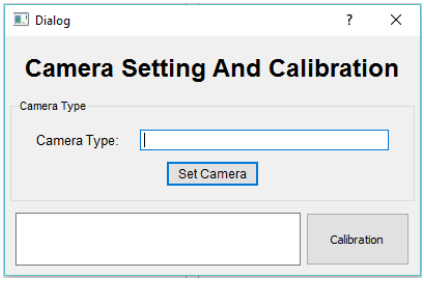
Sebelum masuk ke dalam sistem utama, GUI dari sistem akan dijelaskan terlebih dahulu karena semua proses yang ada di dalam komputer akan berkaitan dengan *user interface* ini. Rancangan GUI yang dibuat terlihat seperti pada Gambar 3.3 dan Gambar 3.4.

Gambar 3.3 merupakan GUI utama dari sistem ini. Di dalamnya terdapat dua kotak berukuran 640x480 *pixel* yang digunakan sebagai penampil citra yang diambil oleh kamera kiri dan kanan. Di GUI utama ini juga terdapat *label* X, Y, dan Z yang menunjukkan nilai posisi dari target yang dipilih. *Group box* dengan nama *matching coefficient*

digunakan untuk proses *template matching* antara kamera kanan dan kamera kiri yang akan dijelaskan nanti. *Group box weapon position* merupakan letak dari senjata terhadap kamera yang harus dimasukkan secara manual oleh *operator* untuk mendukung dalam penghitungan sudut vertikal dan horisontal referensi dari senjata. Tombol *Camera Setting* merupakan tombol untuk membuka GUI yang ada Gambar 3.4. Dan terakhir, tombol *Start* berfungsi untuk memulai dan menghentikan proses penghitungan posisi target, penghitungan sudut referensi senjata, serta pengiriman data melalui komunikasi serial ke mikrokontroler STM32F4.



Gambar 3.3 GUI Utama



Gambar 3.4 GUI pendukung

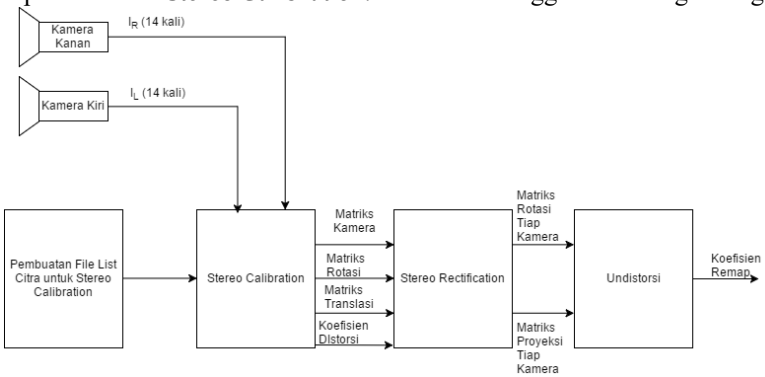
Gambar 3.4 merupakan GUI pendukung yang bertujuan untuk memasukkan tipe kamera yang dipakai serta dapat digunakan untuk proses *stereo calibration* dan *stereo rectification* dari kamera stereo. Jika tipe kamera yang dipakai pernah dikalibrasi dan konfigurasi kamera stereo tidak berubah, proses kalibrasi tidak perlu dilakukan lagi.

3.2. *Stereo Calibration* dan *Stereo Rectidication*

Pada bagian ini, semuanya diekseskusi di dalam komputer dalam bentuk perangkat lunak yang tergambar seperti pada Gambar 3.5. Proses Stereo Calibration dan Stereo Rectification ini sebagian besar diotomatisasi. Faktor manusia hanya dibutuhkan untuk memutar-mutar papan catur yang digunakan untuk proses stereo calibration. Selain hal tersebut, sub sistem ini secara otomatis akan memproses tanpa campur tangan manusia.

Hal pertama yang dilakukan di dalam sub sistem ini adalah pembuatan file xml yang berisi list citra yang akan dikalibrasi. Proses ini hanya membutuhkan tipe kamera yang dipakai agar nama file disesuaikan dengan nama kamera yang dipakai. Selanjutya, kedua kamera yang ada dalam sistem kamera stereo akan mengambil citra masing-masing 14 citra yang dinamakan sesuai dengan nama yang ada di file list citra yang sebelumnya telah dibuat. Proses-proses ini dilakukan otomatis oleh komputer.

Selanjutnya, *file* list citra dan 14 pasang citra yang telah dibuat akan diproses dalam *Stereo Calibration*. Proses ini menggunakan fungsi-fungsi



Gambar 3.5 *Stereo Calibration* dan *Stereo Rectification*

yang telah tersedia di *library* OpenCV. Keluaran dari sistem ini adalah matriks masing-masing kamera, matriks koefisien distorsi dari masing-masing kamera, matriks rotasi kamera pertama terhadap kamera kedua, dan matriks translasi kamera pertama terhadap kamera kedua.

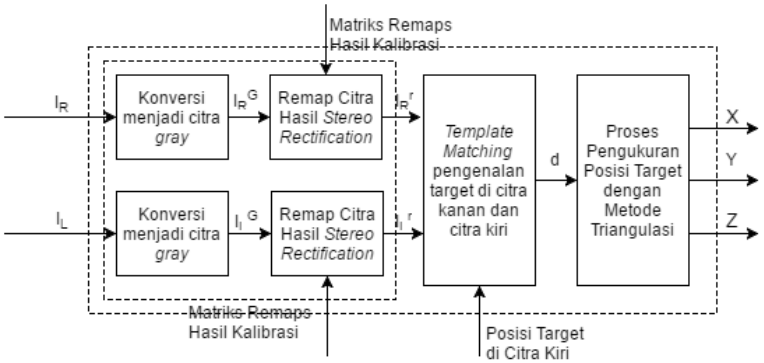
Keluaran dari proses *Stereo Calibration* digunakan didalam proses *Stereo Rectification*. Di dalam proses ini, matriks-matriks masukan akan diproses untuk menghasilkan matriks rotasi dari setiap kamera dan matriks proyeksi dari setiap kamera. Matriks-matriks tersebut digunakan untuk meng-*undistortion* citra yang keluarannya berupa matriks yang berisi koefisien *remap*.

3.3. Pengukuran Posisi Target

Seperti yang ditunjukkan pada Gambar 3.6, diagram blok dari pengukuran posisi target meliputi *preprocessing* citra, *template matching*, dan pengukuran posisi target dengan metode triangulasi. Semua proses ini dilakukan di komputer dalam bentuk perangkat lunak.

3.2.1. *Preprocessing* Citra

Agar dapat mengukurn posisi target dengan mudah, citra yang diambil dari tiap kamera harus diolah terlebih dahulu melalui dua tahapan, yaitu pengkorvesian citra menjadi *grayscale* dan *remap* citra.



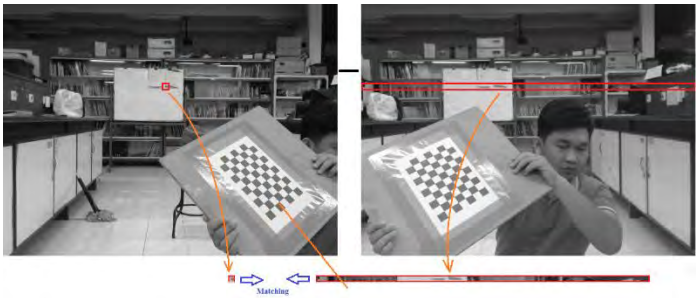
Gambar 3.6 Diagram Blok Pengukuran Posisi Target

Gambar 3.7 *Hasil* Remap *Citra*

Pengkorversian citra menjadi *grayscale* bertujuan untuk memepercepat proses-proses berikutnya karena *grayscale* hanya memiliki lebar data sebesar 8 bit. Dalam proses ini, digunakan *library* OpenCV, yaitu `cvtColor` dengan koefisien `BGR2GRAY`. Proses selanjutnya adalah *remap* citra. Proses ini bertujuan untuk memproyeksi ulang citra dari kamera kanan dan citra dari kamera kiri agar sejajar dalam arah horisontal seperti yang ditunjukkan pada Gambar 3.7. Proses ini memanfaatkan koefisien *remap* yang didapatkan dari proses *stereo calibration* dan *stereo rectification*.

3.2.2. *Template Matching*

Dalam memilih target, *operator* hanya memilih target di *frame* kamera kiri. Oleh karena itu, diperlukan algoritma agar *frame* kamera kanan mendeteksi target yang sama. Cara yang digunakan di dalam tugas akhir ini adalah *template matching*. Karena kedua citra sudah sejajar dalam arah horisontal di setelah mengalami proses *remap* citra, proses *template matching* bisa dibuat lebih cepat dengan menjadikan titik yang di-*click* sebagai *template* dan satu baris di citra kedua dengan posisi horisontal yang sama dengan letak *template* sebagai tempat pencocokan *template matching*. Gambar 3.8 menunjukkan proses *template matching*.



Gambar 3.8 Proses *Template Matchimg*

Langkah-langkah dan *syntax* dari proses ini adalah sebagai berikut:

- Pengambilan *template* dari citra kamera kiri

```
if(camera1_pos.x() < (imgG1.cols-tmp.cols) &&
     camera1_pos.y() < (imgG1.rows-tmp.rows))
{
     for(int i=0; i < tmp.rows; i++){//row
          for(int j=0; j < tmp.cols; j++){//col
             tmp.at<uchar>(i,j) =
                   imgG1.at<uchar>(i+camera1_pos.y(),j
                   +camera1_pos.x());
          }
     }
}
```

Di dalam pengambilan *template*, algoritma yang dipakai adalah memindahkan nilai kecerahan dari posisi yang di-*click* ke suatu *frame* baru yang seukuran. Pada *syntax* di atas, *frame* baru memiliki nama *variable* `tmp`.

- Pengambilan tempat pencarian proses *template matching* yang ada di citra kamera kanan

```
if(camera1_pos.y() < (imgG2.rows-match.rows)){
        for(int i=0; i < match.rows; i++){//row
            for(int j=0; j < match.cols; j++){//col
                match.at<uchar>(i,j) =
                   imgG2.at<uchar>(i+camera1_pos.y(),
                   j);
            }
        }
}
```

Algoritma yang dipakai sama dengan algoritma pengambilan *template*. Namun, lebarnya sama dengan lebar citra, yaitu 640 *pixel*.

- Proses *template matching*

```
dst_size = cv::Size(match.size().width-
tmp.size().width+1,match.size().height-
tmp.size().height+1);
dst_img = cv::Mat(dst_size,CV_32FC1);

cv::matchTemplate(match,tmp,dst_img,CV_TM_CCORR_NORME
D);
cv::minMaxLoc(dst_img,&min_val,&max_val,&min_loc,&max
_loc);
```

Proses *template matching* yang digunakan adalah *normalized cross correlation*. Hal ini tampak di dalam `cv::matchTemplate` terdapat *syntax* `CV_TM_CCORR_NORMED` yang merujuk pada metode tersebut. Posisi *matching* ada di posisi dengan nilai hasi *cross correlation* maksimum. Nilai kecocokan metode ini bisa dilihat di variabel `max_val` yang nilainya antara 0 dan 1 dengan 0 merupakan nilai yang sangat tidak cocok dan 1 merupakan nilai yang benar-benar cocok.

3.2.3. Pengukuran Posisi Target dengan Metode Triangulasi

Pengukuran posisi target di dalam sistem ini menggunakan fungsi `triangulatePoints` yang ada di *library* OpenCV. Metode ini membutuhkan matriks proyeksi setiap kamera P1 dan P2 yang telah dihitung dari proses *stereo rectification*. Keluaran yang dihasilkan fungsi ini adalah matriks $[X\ Y\ Z\ W]^T$. Sehingga, posisi yang didapatkan adalah x=X/W, y=Y/W, dan z=Z/W.

3.4. Penghitungan Sudut Posisi Motor *Platform* Senjata

Untuk menggerakkan senjata agar mengarah ke target, sudut vertikal dan horisontal dibutuhkan. Padahal, informasi yang diperoleh dari hasil pengukuran triangulasi merupakan informasi posisi dalam koordinat kartesian. Sehingga, dibutuhkan konversi dari koordinat kartesian menjadi posisi sudut vertikal dan horisontal seperti pada Gambar 3.9.

Dari ilustrsi tersebut, persamaan sudut horisontal dan sudut vertikal bisa didapatkan dari persamaan (3.1) dan (3.2) berikut:

$$\theta_h = \tan^{-1}\left(\frac{X_{target}-X_{senjata}}{Z_{target}-Z_{senjata}}\right) \quad (3.1)$$

$$\theta_v = \tan^{-1}\left(\frac{Y_{target}-Y_{senjata}}{\sqrt{(X_{target}-X_{senjata})^2+(Z_{target}-Z_{senjata})^2}}\right) \quad (3.2)$$

Proses ini dilakukan di dalam komputer dalam bentuk perangkat lunak.

Syntax dari proses ini tergabung di dalam pemrograman GUI. Berikut ini *syntax* yang berkaitan dengan penghitungan sudut referensi motor.

- Penghitungan sudut horizontal

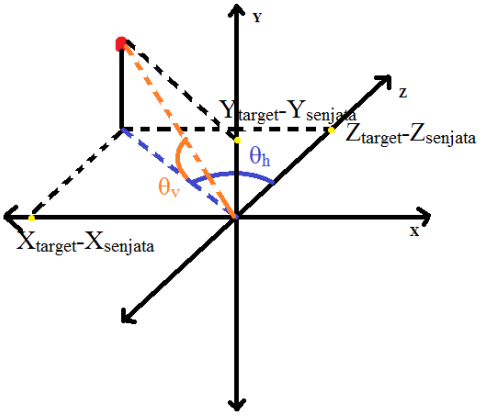
```
long double theta_h = atan((long double)xx/(long double)zz)
                      *180/PI;
```

`xx` dan `zz` merupakan selisih antara posisi target dengan posisi senjata dalam sumbu x dan sumbu z.

- Penghitungan sudut vertical

```
long double r = (long double)sqrt((long double)(xx*(long
                double) xx+(long double)zz*(long double)zz
                ));
long double theta_v = atan((long double)(yy)/r)*180/PI;
```

`xx`, `yy`, dan `zz` merupakan selisih antara posisi target dengan posisi senjata dalam sumbu x, sumbu y, dan sumbu z.



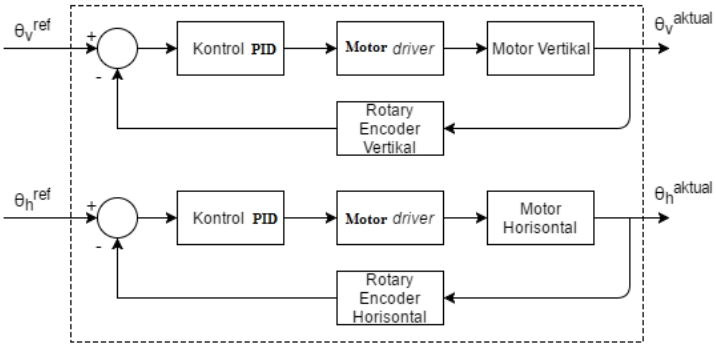
Gambar 3.9 Ilustrasi Konversi Koordinat Kartesian menjadi Sudut Motor

3.5. Pengarahan Senjata ke Target

Pada bagian ini, semua proses dilakukan di dalam Mikrokontroler STM32F4. Sudut vertikal dan sudut horisontal yang dihasilkan dari proses konversi dari koordinat kartesian dikirimkan melalui komunikasi serial ke STM32F4. Di dalam STM32F4, sudut hasil perhitungan akan dikurangi dengan sudut aktual motor yang akan menjadi sinyal *error* bagi kontrol PID. Kontrol PID akan menentukan berapa keluaran PWM yang akan menjadi input bagi motor *driver*. *Motor driver* yang dipakai akan menghasilkan arus untuk menggerakkan motor. Pembacaan posisi actual motor dilakukan oleh *Rotary Encoder* dengan bantuan sensor *proximity* dan *limit switch* untuk menentukan posisi nol dari setiap encoder. Diagram blok dari sistem ada pada Gambar 3.10.

3.5.1. Kontrol PID

Keluaran kontrol PID dalam Tugas Akhir ini adalah sinyal PWM yang akan mengontrol jumlah arus yang dihasilkan *driver* motor untuk menggerakkan motor ke posisi yang diinginkan. Setelah dilakukan *tuning* Kp, Ki, dan Kd secara manual, nilai masing-masing koefisien tersebut yang membuat respon motor cepat dan meminimumkan timbulnya osilasi adalah $Kp = 0.9$, $Ki = 0.0$, dan $Kd = 1.1$.



Gambar 3.10 Diagram Blok Pengarah Senjata

Implementasi kontrol PID di dalam Tugas Akhir ini berada di Mikrokontroler STM32F4. Algoritma yang digunakan untuk mengontrol posisi motor vertikal dan motor horisontal sama. Perbedaan hanya terletak pada *threshold* kecepatan maksimum dan *threshold* sinyal *error* minimum motor diperbolehkan bergerak untuk menghindari motor menjadi panas akibat bergetar ketika terjadi perubahan yang kecil.

Algoritma dari kontrol PID yang dirancang untuk sistem ini meliputi pencarian nilai sinyal *error*, dilanjutkan dengan pencarian nilai *integral error* dan nilai *derivative error*. Setelah ketiga nilai tersebut didapatkan, nilai keluaran PWM dapat diperoleh dengan menggunakan perumusan kontrol PID. Algoritma ini diimplementasikan di STM32F4 dengan *syntax* seperti yang tertulis berikut ini.

- Mencari *proportional error*

```
error_h = setpoint_h-rotasi_h;
err1=error_h;
```

- Mencari *integral error*

```
sumerr1+=err1;
if(err1>65535)err1=0;
```

Syntax di atas ini bertujuan agar jika *integral error* mengalami *overflow*, nilai *integral error* dinolkan kembali.

- Mencari *derivative error*

```
err1-errs1
errs1=err1;
```

- Keluaran kontrol PID

```
PWM1_PID=(float)0.9*err1+ki1*sumerr1+
         (float)1.1*(err1-errs1);
```

- Keluaran PWM

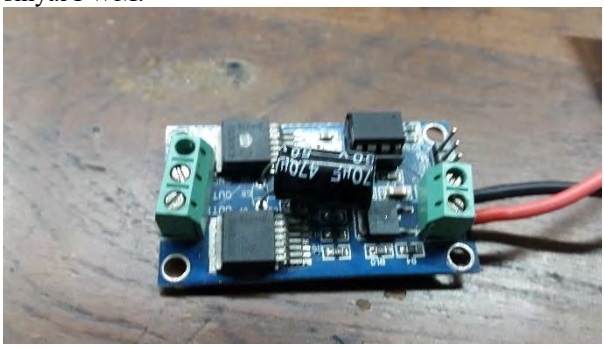
```
if(PWM1_PID < 0){
   if(PWM1_PID < -350)PWM1_PID=-350;
   GPIO_WriteBit(GPIOE, GPIO_Pin_6,0);
   GPIO_WriteBit(GPIOC, GPIO_Pin_13,1);
   PWM1=-1*PWM1_PID;
}
else if(PWM1_PID>0){
   if(PWM1_PID > 350)PWM1_PID=350;
```

```
    GPIO_WriteBit(GPIOE, GPIO_Pin_6,1);
    GPIO_WriteBit(GPIOC, GPIO_Pin_13,0);
    PWM1=PWM1_PID;
  }
  else if(PWM1_PID==0){
    GPIO_WriteBit(GPIOE, GPIO_Pin_6,0);
    GPIO_WriteBit(GPIOC, GPIO_Pin_13,0);
    PWM1=0;
  }
```

3.5.2. *Driver* Motor

Driver motor yang digunakan pada tugas akhir ini adalah BTN7970 sebanyak 2 buah. Satu *driver* untuk menggerakkan motor penggerak vertikal dan yang lain sebagai motor penggerak horisontal. Gambar 3.11 menunjukkan bentuk dari *driver* motor BTN7970.

Tegangan suplai yang digunakan disesuaikan dengan motor yang digunakan, yang dalam tugas akhir ini digunakan sebesar 24 Volt. Terdapat 3 input lain, yaitu IN1, IN2, dan VIN. IN1 dan IN2 merupakan pengatur arah gerak motor yang menentukan kutub positif dan negatif dari OUT1 dan OUT2. Sedangkan VIN merupakan input tegangan DC antara 0-5 Volt yang berfungsi untuk mengatur kecepatan motor. Di pin VIN juga bisa diinputkan sinyal PWM.



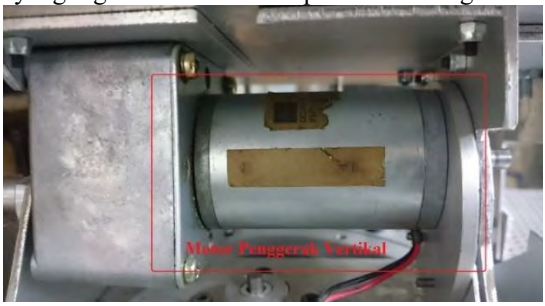
Gambar 3.11 *Driver* Motor BTN7970

3.5.3. Motor DC

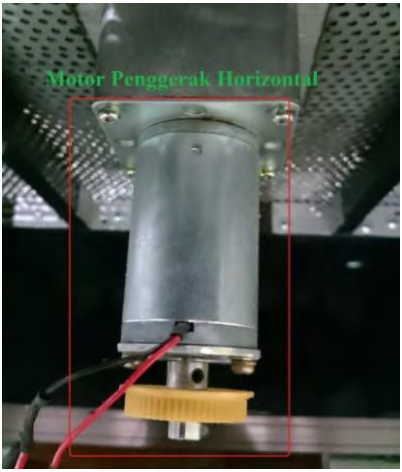
Penggerak vertical dan horizontal system senjata berupa motor DC dengan tegangan 24 Volt. Arah pergerakan motor ditentukan oleh pemberian tegangan DC pada kedua input motor. Gambar 3.12 dan Gambar 3.13 memberikan gambaran motor DC yang digunakan pada tugas akhir ini beserta peletakannya.

3.5.4. *Rotary Encoder*

Dengan bantuan sensor *proximity* untuk posisi horisontal dan *limit switch* untuk posisi vertikal, *rotary encoder* bisa digunakan sebagai *feedback* posisi dari motor penggerak horisontal. *Rotary encoder* yang digunakan memiliki spesifikasi sebagai berikut:



Gambar 3.12 Motor DC Penggerak Vertikal



Gambar 3.13 Motor DC Penggerak Horisontal

Gambar 3.14 *Rotary Encoder*

- *Manufacture* : Omron
- *Model* : E6B2-CWZ6C
- Suplai Daya : 5-24VDC ± 5%
- Jumlah pulsa : 200 pulsa per rotasi

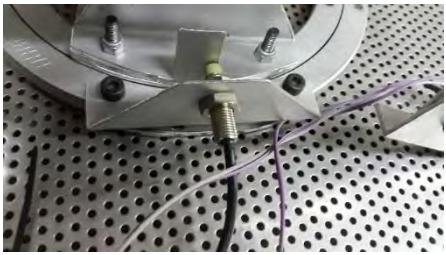
Gambar 3.14 menunjukkan *rotary encoder* yang akan dipakai pada tugas akhir ini.

3.5.5. Sensor *Proximity*

Sensor *proximity* dalam tugas akhir ini digunakan sebagai titik 0 dari *rotary encoder* yang digunakan sebagai pengukur posisi motor horisontal. Sensor *proximity* yang digunakan adalah Autonics PR8-2DN. Sensor ini memiliki parameter sebagai berikit:

- Jarak deteksi : 2mm±10%
- *Power Supply* : 12-24V
- Jenis Deteksi : Logam
- Output sensor : *Active Low*

Gambar 3.15 menunjukkan sensor *proximity* yang akan dipakai pada tugas akhir ini.



Gambar 3.15 Sensor *Proximity*

3.5.6. *Limit Switch*

Sensor *proximity* dalam tugas akhir ini digunakan sebagai pembantu mencari titik 0 dari *rotary encoder* yang digunakan sebagai pengukur posisi motor vertikal. *Limit switch* yang digunakan memiliki 3 pin seperti pada Gambar 3.16. Parameter dari *limit switch* yang dipakai adalah

- Pin : *Normally Open (NO), Normally Closed (NC), Ground*
- Batas Arus :
  - 250 VAC, 15 A
  - 125 VAC, 15 A
  - 250 VDC, 0.3 A
  - 125 VDC, 0.6 A

3.6. Fitur Pendukung

Fitur pendukung di dalam sistem ini bertujuan agar sistem ini bisa berjalan baik. Terdapat empat fitur yang dirancang di dalam sistem ini, yaitu komunikasi serial antara komputer dan mikrokontroler, pembuatan GUI pengujian kontrol *platform* senjata, dan metode *tracking* menggunakan metode *optical flow*, dan *kalman filter*.

3.6.1. Komunikasi Serial

Perancangan komunikasi serial merupakan komponen penting di dalam sistem ini. Hal ini dikarenakan informasi yang telah diolah di komputer yang berupa sudut posisi motor untuk menggerakkan senjata harus diterima oleh miktrokontroler yang bertugas menggerakkan motor pengarah senjata. Karena sistem ini terdiri dari dua *processor*, maka perancangan komunikasi serial terbagi menjadi dua, yaitu bagian pengiriman dari komputer dan bagian penerimaaan di mikrokontroler. Pengaturan komunikasi serial yang dipakai adalah

Gambar 3.16 *Limit Switch*

- *Baud rate* : 9600 bps
- Jumlah data tiap pengiriman : 8 bit
- *Stop* bit : 1 bit
- *Parity* bit : tidak ada
- *Flow control* : tidak ada

3.6.1.1. Pengiriman Data dari Komputer

Syntax pengaturan pengiriman dari komputer agar sesuai dengan aturan komunikasi serial adaalah sebagai berikut:

```
SerialCom->setBaudRate(QSerialPort::Baud9600);
SerialCom->setDataBits(QSerialPort::Data8);
SerialCom->setParity(QSerialPort::NoParity);
SerialCom->setStopBits(QSerialPort::OneStop);
SerialCom->setFlowControl(QSerialPort::
          NoFlowControl);
```

Data pengiriman berupa nilai sudut vertikal dan horisontal. Tipe data yang dipakai berupa `qint32` yang merupakan tipe data di Qt dengan panjang data 32 bit. Karena dalam sekali pengiriman hanya mengirim 8 bit data, data sudut dibagi menjadi 4 bagian dengan *syntax*:

```
send_serial.append(th_h>>24);
send_serial.append(th_h>>16);
send_serial.append(th_h>>8);
send_serial.append(th_h);
send_serial.append(th_v>>24);
send_serial.append(th_v>>16);
send_serial.append(th_v>>8);
send_serial.append(th_v);
```

`th_h` merupakan sudut horisontal dan `th_v` merupakan sudut vertikal. Untuk menghindari kesalahan penggabungan data di sisi penerima, maka sebelum data sudut dikirim, sebuah penanda dikirim yaitu berupa huruf ‘a’ dengan *syntax*:

```
send_serial.append('a');
```

Setelah penyusunan data yang akan dikirim telah selesai, pengiriman data pada Qt menggunakan *syntax*:

```
SerialCom->write(send_serial);
```

3.6.1.2. Penerimaan Data pada STM32F4

Pengaturan komunikasi serial pada STM32F4 agar dapat menerima data dari komputer adalah sebagai berikut:

```
RxTx.USART_BaudRate = 9600;
RxTx.USART_WordLength = USART_WordLength_8b;
RxTx.USART_StopBits = USART_StopBits_1;
RxTx.USART_Parity = USART_Parity_No;
RxTx.USART_HardwareFlowControl=
            USART_HardwareFlowControl_None;
```

Karena pengiriman menggunakan penanda awal berupa huruf ‘a’ dan data 32 bit dibagi menjadi empat kali pengiriman, maka algoritma penerimaan harus sesuai dengan hal tersebut agar tidak ada kesalahan data saat penerimaan. Oleh karena itu, di dalam algoritma penerimaan data di mikrokontroler, terdapat variabel penanda yang merupakan apakah data siap diambil atau belum. Ketika huruf ‘a’ diterima, penanda akan bernilai ‘1’ dan data berikutnya merupakan data pertama yang akan diambil. Karena data yang dikirim adalah sudut horisontal dan sudut vertikal yang masing-masing memilki panjang data 32 bit dan sekali pengiriman adalah 8 bit, maka pengambilan data setelah huruf ‘a’ dilakukan sebanyak (32+32)/8 = 8 kali. *Syntax* dalam proses penerimaan di mikrokontroler adalah:

```
if(penanda==0){
  if(USART6->DR=='a')penanda=1;
  else penanda=0;
}
else if(penanda==1){
 switch(datake){
  case 0:rec[datake]=USART6->DR;datake++;break;
  case 1:rec[datake]=USART6->DR;datake++;break;
  case 2:rec[datake]=USART6->DR;datake++;break;
  case 3:rec[datake]=USART6->DR;datake++;break;
  case 4:rec[datake]=USART6->DR;datake++;break;
  case 5:rec[datake]=USART6->DR;datake++;break;
  case 6:rec[datake]=USART6->DR;datake++;break;
  case 7:rec[datake]=USART6->DR;
         datake=0;penanda=0;break;
 }
}
```

Setelah data diterima, data tersebut dikonversi kembali menjadi nilai sudut horisontal dan sudut vertikal dengan *syntax*:

```
data1 =
      (float)(rec[0]<<24)+(float)(rec[1]<<16)+
      (float)(rec[2]<<8)+(float)(rec[3]);
data2 =
      (float)(rec[4]<<24)+(float)(rec[5]<<16)+
      (float)(rec[6]<<8)+(float)(rec[7]);
```

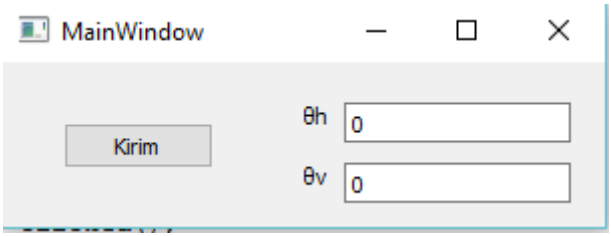
dengan data1 merupakan sudut horisontal dan data2 merupakan sudut vertikal.

3.6.2. GUI Pengujian Kontrol Senjata

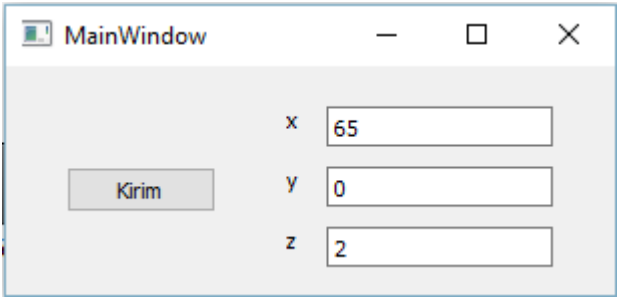
Untuk menguji keberhasilan komunikasi antara komputer dan mikrokontroler, GUI pengujian dibuat untuk menguji pengontrolan sistem senjata melalui komputer. Terdapat dua buah GUI yang dibuat untuk mendukung keberhasilan GUI ini. Yang pertama adalah GUI untuk pengontrolan *platform* senjata berdasarkan sudut vertikal dan horisontal. GUI yang kedua adalah pengontrolan *platform* senjata berdasarkan posisi kartesian.

3.6.2.1. GUI Kontrol Platform Senjata Berdasarkan Sudut Vertikal dan Horisontal

Gambar 3.17 menunjukkan GUI pengontrolan *platform* senjata berdasarkan sudut vertikal dan sudut horisonal. Pada GUI ini terdapat dua kotak yang dapat diisikan nilai sudut horisontal dan sudut vertikal yang diinginkan. Ketika tombol kirim ditekan, senjata akan bergerak sesuai dengan sudut yang telah dituliskan di kotak yang tersedia. *Syntax* pengiriman sudut vertikal dan horisontal pada bagian ini sama dengan *syntax* pada bagian perancangan komunikasi serial untuk pengiriman dari komputer.



Gambar 3.17 GUI kontrol *platform* senjata berdasarkan sudut horisontal dan vertikal



Gambar 3.18 GUI kontrol *platform* senjata berdasarkan posisi kartesian

3.6.2.2. GUI Kontrol Platform Senjata Berdasarkan Posisi Kartesian

Seperti halnya pada bagian sebelumnya, GUI yang tergambar pada Gambar 3.18 juga akan mengontrol *platform* senjata. Hanya saja, pengontrolan ini berdasarkan posisi kartesian suatu target. *Syntax* komunikasi serial yang digunakan sama. Data yang dikirim juga hanya dua. Karena nilai masukan berupa koordinat kartesian, sebelum dikirim, tiga nilai koordinat kartesian diubah terlebih dahulu menjadi data sudut vertikal dan horisontal. *Syntax* konversi dari koordinat kartesian menjadi sudut horisontal dan vertikal adalah sebagai berikut.

```
long double theta_h = atan((long double)x/(long
                double)z)*180/PI;
```

```
long double r = (long double)sqrt((long
            double)(x*(long double)x+(long
            double)z*(long double)z));
long  double  theta_v  =  atan((long  double)
            (y)/r)*180/PI;
```

Dari *syntax* di atas bisa dilihat bahwa konversi dari kartesian menjadi sudut mirip dengan penghitungan sudut pada sub bab 3.4. Perbedaannya, pada bagian ini, titik asal berada pada titik pusat penggerakan *platform* snejata.

3.6.3. *Optical Flow*

Di dalam sistem ini, setelah senjata diarahkan ke target, algoritma *tracking* ditambahkan agar senjata bisa terus mengarah ke target. Metode yang dipakai adalah *Pyramidal Lucas-Kanade Optical Flow*. Agar tidak menimbulkan keambiguan target, jumlah maksimal objek yang bisa di-*tracking* adalah satu. *Syntax* yang digunakan adalah fungsi yang ada di *library* OpenCV, yaitu:

```
calcOpticalFlowPyrLK(prevGray,     gray,     points[0],
          points[1], status, err, winSize, 3, termcrit,
          0, 0.001);
```

`prevGray` merupakan citra sebelumnya yang sudah dikonversi menjadi *grayscale*. Sedangkan `gray` merupakan citra saat ini yang dikonversi menjadi *grayscale*. `Points[0]` merupakan titik yang ingin di-*tracking* di citra `prevGray`. Dan `points[1]` merupakan prediksi titik `points[0]` pada citra `gray`.

3.6.4. *Kalman filter*

Agar motor tidak bergetar saat target diam, *kalman filter* dibuat di dalam sistem ini. *Kalman filter* berfugsi untuk mem-*filter noise* pengukuran dari pengukuran posisi target. Posisi algoritma dari *kalman filter* adalah setelah pengukuran. *Syntax* dari *kalman filter* adalah sebagai berikut.

```
//Predict Kalman X
Xk=ax*(float)Xk;
pkx=ax*pkx*ax;
//Update Kalman X
gkx=pkx/(pkx+1);
Xk=(qint32)((float)Xk+gkx*(float)(X-Xk)/100);
pkx=(1-gkx)*pkx;
```

```
//Predict Kalman Y
Yk=ay*(float)Yk;
pky=ay*pky*ay;
//Update Kalman Y
gky=pky/(pky+1);
Yk=(qint32)((float)Yk+gky*(float)(Y-Yk)/100);
pky=(1-gky)*pky;

//Predict Kalman Z
Zk=az*(float)Zk;
pkz=az*pkz*az;
//Update Kalman Z
gkz=pkz/(pkz+1);
Zk=(qint32)((float)Zk+gkz*(float)(Z-Zk)/100);
pkz=(1-gkz)*pkz;
```

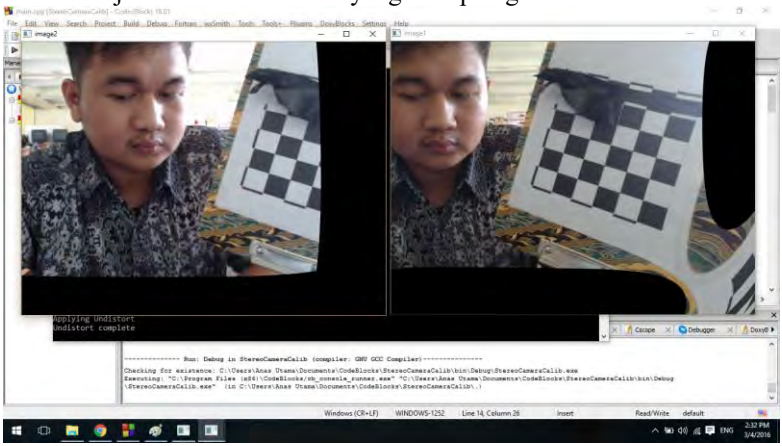
Pada *syntax* tersebut, nilai a (ax, ay, dan az) diberi nilai 1.0. Nilai awal dari *prediction* error, pk (pkx, pky, dan pkz) diberi nilai 1.0. Sedangkan gk (gkx, gky, dan gkz) diberi nilai awal 0.0.

BAB IV
PENGUJIAN DAN ANALISA

Pada bab ini akan dibahas mengenai pengujian dari sistem yang telah dirancang. Bab ini bertujuan untuk mengetahui apakah tujuan dalam perancangan sistem pada tugas akhir ini telah terlaksana atau tidak. Pengujian pada bab ini terdiri dari pengujian pengukuran posisi target dengan metode triangulasi, pengukuran posisi target berdasarkan matriks proyeksi, pengukuran posisi target dengan jarak antar kamera yang berbeda.

4.1. Pengujian *Stereo Calibration* dan *Stereo Rectification*

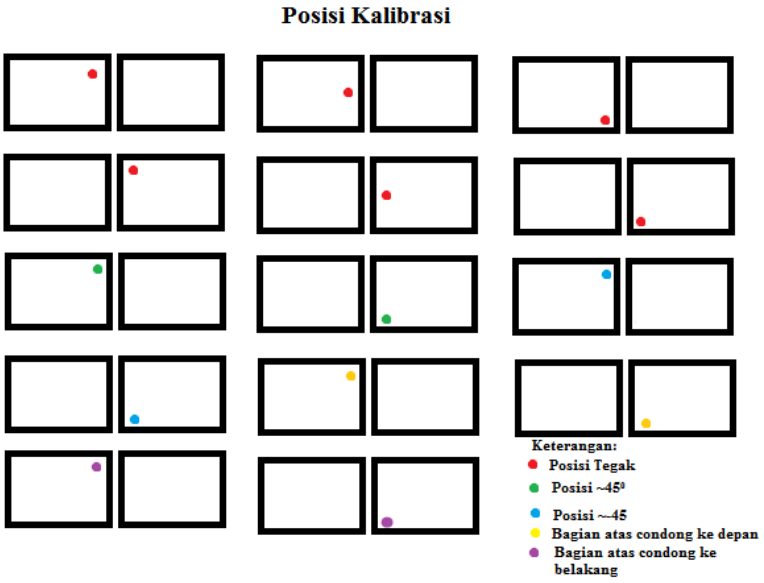
Pengujian ini dilakukan karena dapat menentukan luas *frame* kamera stereo yang bisa dipakai. Semakin bagus posisi kalibrasi, semakin luas *frame* kamera stereo serta menentukan keakuratan pengukuran posisi. Gambar 4.1 menunjukkan hasil kalibrasi yang buruk dan Gambar 4.2 menunjukkan hasil kalibrasi yang cukup bagus.



Gambar 4.1 Hasil Kalibrasi yang Buruk

Gambar 4.2 Hasil Kalibrasi yang Cukup Baik

Setelah lebih dari 10 kali pengujian, posisi pola papan catur yang menghasilkan hasil pengukuran terbaik dalah seperti pada Gambar 4.3.



Gambar 4.3 Posisi Kalibrasi

4.2. Pengujian Pengukuran Posisi Target dengan Metode Triangulasi

Pada pengujian ini, dilakukan pengukuran posisi target yang berupa buku yang memiliki fitur unik. Dalam pengujian ini, pengukuran triangulasi yang dilakukan menggunakan dua cara yang berbeda. Cara pertama adalah menggunakan fungsi `triangulatePoints` yang ada dalam *library* OpenCV. Cara kedua adalah dengan penghitungan *disparity* (perbedaan koordinat) dari target yang sama di dua citra dan kemudian dihitung menggunakan rumus triangulasi seperti pada persamaan (4.1), (4.2), dan (4.3) berikut ini,

$$X = \frac{a(x_R + x_L)}{x_R - x_L} \tag{4.1}$$

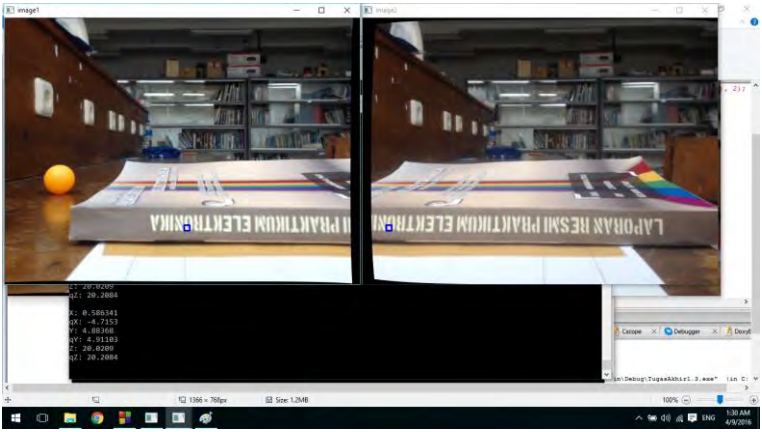
$$Y = \frac{2ay_R}{x_R - x_L} \tag{4.2}$$

$$Z = \frac{2fa}{x_R - x_L} \tag{4.3}$$

Keterangan:
a: jarak antar kamera
x_R, y_R: koordinat target pada citra kamera kanan
x_L, y_L: koordinat target pada citra kamera kiri
f: panjang fokus kamera
X, Y, Z: koordniat ruang 3D target
y_R dan y_L harus sama (citra stereo sudah dikalibrasi, di-*rectify*, dan di-*remap*)

Gambar 4.4 menunjukkan target berupa buku yang nantinya jarak target diubah-ubah. Jarak target bervariasi mulai dari 20 cm hingga 100 cm dengan interval jarak 10 cm. Buku yang menjadi target akan dipilih bagian dari buku tersebut yang memiliki fitur unik. Posisi fitur unik yang dipilih berada di sekitar titik tengah citra kamera kiri.

Sebagai catatan, pengukuran posisi dengan OpenCV memiliki titik asal di titik tengah citra kamera kiri. Sedangkan dengan pengukuran menggunakan rumus triangulasi, titik asal koordinat berada di titik tengah antara kamera kanan dan kiri



Gambar 4.4 Pengukuran Target Jarak 20 cm

Tabel 4.1 Hasil Pengukuran Posisi dengan Fungsi OpenCV

Jarak (cm)	Posisi (cm)			Error Z (%)
	X	Y	Z	
20	0.5863	4.8837	20.021	4.8215
30	0.6001	4.8771	29.3015	0.6924
40	0.7892	4.6494	38.5335	1.4488
50	1.1444	4.5772	47.6770	2.8982
60	1.1758	4.0960	57.4070	2.8646
70	1.1225	4.0231	66.1869	3.7981
80	1.5363	4.0311	75.0118	4.8074
90	1.7455	3.8302	85.2407	4.0082
100	1.5902	3.8830	93.7648	5.0964

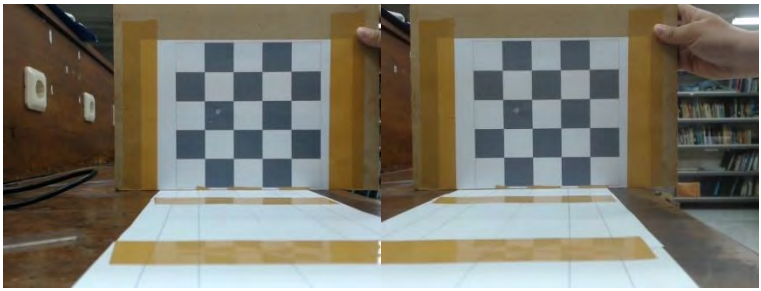
Tabel 4.2 Hasil Pengukuran Posisi dengan Penghitungan Manual

Jarak (cm)	Posisi (cm)			Error Z (%)
	X	Y	Z	
20	-4.7153	4.9110	20.2084	5.8031
30	-4.8438	4.8958	29.5758	1.6351
40	-4.7945	4.6575	38.8942	0.5263
50	-4.5763	4.5763	49.1233	0.0475
60	-4.6939	4.0816	57.9444	1.9553
70	-4.8824	4.0000	66.8064	2.8977
80	-4.6000	4.0000	75.7140	3.9162
90	-4.5455	3.7879	86.0386	3.1097
100	-4.8333	3.8333	94.6424	4.2081

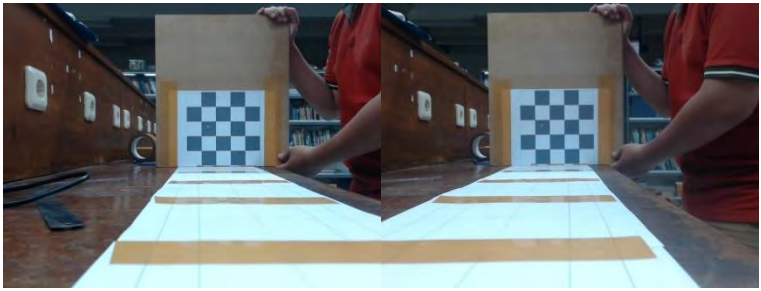
Dari data yang diperoleh pada Tabel 4.1 dan Tabel 4.2, baik menggunakna fungsi `triangulatePoints` yang ada di *library* OpenCV maupun dengan menggunakan persamaan metode triangulasi, kesalahan pengukuran jarak target (Z) kurang dari 6%. Sebagai catatan, konfigurasi stereo kamera saat pengujian ini dilakukan masih belum menggunakan *casing*, sehingga ada kemungkinan kamera tergeser yang bisa mengurangi akurasi dari pengukuran ini.

4.3. Pengujian Pengukuran Posisi Target berdasarkan Matriks Proyeksi

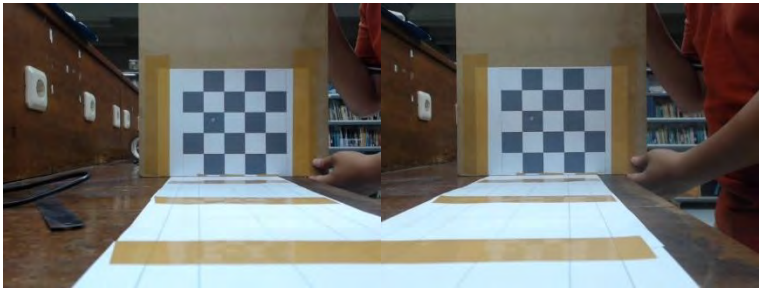
Dengan cara ini, yang harus dicari terlelbih dahulu adalah matriks proyeksi dari masing-masing kamera. Untuk mendapatkan matriks proyeksi dari setiap kamera, kalibrasi kamera tunggal perlu dilakukan. Untuk itu, pengambilan gambar dengan objek yang memiliki titik-titik yang diketahui koordinat ruang 3D harus dilakukan. Objek yang bisa digunakan adalah papan catur.



Gambar 4.5 Citra Kalibrasi Jarak 50 cm



Gambar 4.6 Citra Kalibrasi Jarak 100 cm



Gambar 4.7 Citra Kalibrasi Jarak 70 cm

Gambar 4.5, Gambar 4.6, dan Gambar 4.7 merupakan citra pola papan catur yang diambil oleh kamera kanan dan kiri dengan jarak terhadap kamera yang bervariasi, yaitu 50 cm, 70cm, dan 100 cm. Setelah diekstrak, data titik-titik yang ada di papan catur terpapar di Tabel 4.3 untuk kamera kiri dan Tabel 4.4 untuk kamera kanan.

Tabel 4.3 Data Titik-Titik Pola Papan Catur di Kamera Kiri

Kamera Kiri					
Titik	Koordinat Kamera		Koordinat 3D		
	x_i	y_i	x_c	y_c	z_c
1	341.185	114.763	-4.050	9.804	49.200
2	389.852	114.584	0.000	9.804	49.200
3	438.701	114.418	4.050	9.804	49.200
4	488.151	113.973	8.100	9.804	49.200
5	341.370	163.872	-4.050	5.754	49.200
6	389.906	163.716	0.000	5.754	49.200
7	438.718	163.507	4.050	5.754	49.200
8	488.045	163.248	8.100	5.754	49.200
9	341.534	212.670	-4.050	1.704	49.200
10	390.037	212.603	0.000	1.704	49.200
11	438.854	212.459	4.050	1.704	49.200
12	487.966	212.335	8.100	1.704	49.200
13	341.702	261.363	-4.050	-2.346	49.200
14	390.378	261.330	0.000	-2.346	49.200
15	438.890	261.198	4.050	-2.346	49.200
16	488.230	261.137	8.100	-2.346	49.200
17	339.164	148.761	-4.050	9.804	69.200
18	373.983	148.670	0.000	9.804	69.200
19	408.895	148.592	4.050	9.804	69.200
20	444.087	148.499	8.100	9.804	69.200
21	339.276	183.681	-4.050	5.754	69.200
22	374.169	183.654	0.000	5.754	69.200
23	409.146	183.554	4.050	5.754	69.200
24	444.219	183.413	8.100	5.754	69.200
25	339.415	218.594	-4.050	1.704	69.200

Lanjutan Tabel 4.3 Data Titik-Titik Pola Papan Catur di Kamera Kiri

Kamera Kiri					
Titik	Koordinat Kamera		Koordinat 3D		
	x_i	y_i	x_c	y_c	z_c
26	374.297	218.600	0.000	1.704	69.200
27	409.257	218.522	4.050	1.704	69.200
28	444.303	218.504	8.100	1.704	69.200
29	339.478	253.567	-4.050	-2.346	69.200
30	374.477	253.562	0.000	-2.346	69.200
31	409.417	253.541	4.050	-2.346	69.200
32	444.658	253.453	8.100	-2.346	69.200
33	335.584	174.460	-4.050	9.804	99.200
34	360.159	174.395	0.000	9.804	99.200
35	384.863	174.359	4.050	9.804	99.200
36	409.465	174.219	8.100	9.804	99.200
37	335.730	199.312	-4.050	5.754	99.200
38	360.257	199.077	0.000	5.754	99.200
39	385.141	198.780	4.050	5.754	99.200
40	409.601	198.649	8.100	5.754	99.200
41	335.860	223.803	-4.050	1.704	99.200
42	360.384	223.735	0.000	1.704	99.200
43	385.272	223.649	4.050	1.704	99.200
44	409.681	223.613	8.100	1.704	99.200
45	336.082	248.495	-4.050	-2.346	99.200
46	360.531	248.441	0.000	-2.346	99.200
47	385.409	248.394	4.050	-2.346	99.200
48	409.973	248.245	8.100	-2.346	99.200

Tabel 4.4 Data Titik-Titik Pola Papan Catur di Kamera Kanan

Kamera Kanan					
Titik	Koordinat Kamera		Koordinat 3D		
	x_i	y_i	x_c	y_c	z_c
1	205.565	112.249	-4.050	9.804	49.200
2	254.736	111.807	0.000	9.804	49.200
3	303.708	111.450	4.050	9.804	49.200
4	352.810	110.842	8.100	9.804	49.200
5	206.451	161.590	-4.050	5.754	49.200
6	255.495	161.353	0.000	5.754	49.200
7	304.360	160.730	4.050	5.754	49.200
8	353.364	160.352	8.100	5.754	49.200
9	207.362	210.607	-4.050	1.704	49.200
10	256.159	210.316	0.000	1.704	49.200
11	304.788	209.760	4.050	1.704	49.200
12	353.597	209.454	8.100	1.704	49.200
13	208.166	259.385	-4.050	-2.346	49.200
14	256.741	258.957	0.000	-2.346	49.200
15	305.365	258.572	4.050	-2.346	49.200
16	354.151	258.282	8.100	-2.346	49.200
17	238.125	146.296	-4.050	9.804	69.200
18	273.159	145.997	0.000	9.804	69.200
19	308.196	145.725	4.050	9.804	69.200
20	343.323	145.482	8.100	9.804	69.200
21	238.455	181.425	-4.050	5.754	69.200
22	273.495	181.179	0.000	5.754	69.200
23	308.487	180.782	4.050	5.754	69.200
24	343.568	180.485	8.100	5.754	69.200
25	238.799	216.510	-4.050	1.704	69.200

Lanjutan Tabel 4.4 Data Titik-Titik Pola Papan Catur di Kamera Kanan

Kamera Kanan					
Titik	Koordinat Kamera		Koordinat 3D		
	x_i	y_i	x_c	y_c	z_c
26	273.821	216.340	0.000	1.704	69.200
27	308.909	215.854	4.050	1.704	69.200
28	344.101	215.620	8.100	1.704	69.200
29	239.369	251.559	-4.050	-2.346	69.200
30	274.448	251.408	0.000	-2.346	69.200
31	309.486	250.994	4.050	-2.346	69.200
32	344.557	250.555	8.100	-2.346	69.200
33	260.042	171.857	-4.050	9.804	99.200
34	284.679	171.655	0.000	9.804	99.200
35	309.605	171.521	4.050	9.804	99.200
36	334.261	171.314	8.100	9.804	99.200
37	260.556	196.798	-4.050	5.754	99.200
38	284.964	196.545	0.000	5.754	99.200
39	309.894	196.309	4.050	5.754	99.200
40	334.499	195.803	8.100	5.754	99.200
41	260.742	221.502	-4.050	1.704	99.200
42	285.416	221.310	0.000	1.704	99.200
43	310.282	221.076	4.050	1.704	99.200
44	334.751	220.808	8.100	1.704	99.200
45	261.123	246.421	-4.050	-2.346	99.200
46	285.685	246.220	0.000	-2.346	99.200
47	310.485	245.725	4.050	-2.346	99.200
48	335.388	245.467	8.100	-2.346	99.200

Dengan data-data tersebut dan proses pengolahan data untuk medapatkan matriks proyeksi kamera, didapatkan matriks proyeksi kamera kiri yaitu:

Tabel 4.5 Hasil Pengukuran dengan Matriks Proyeksi

Jarak (cm)	Kamera Kiri		Kamera Kiri		Koordinat 3D (cm)			Error (%)
	x_R	y_R	x_L	y_L	X	Y	Z	
30	434	323	219	319	4.948	-4.711	26.863	10.456
80	379	273	282	271	5.792	-4.785	71.895	10.132
100	365	265	284	263	5.142	-4.614	89.739	10.261

$$M_{kiri} = \begin{bmatrix} 127.5661 & -1.4905 & 1.5615 & 694.6692 \\ 0.1519 & -128.1993 & -0.7274 & -46.8174 \\ -0.0142 & -0.0128 & -0.2011 & 1 \end{bmatrix}$$

$$M_{kanan} = \begin{bmatrix} 107.4755 & -0.8312 & -0.5700 & -541.3215 \\ -0.9994 & -106.9837 & -0.9627 & -42.7535 \\ -0.0002 & -0.0181 & 0.1654 & 1 \end{bmatrix}$$

Setelah matriks proyeksi diperoleh, koordinat ruang 3D dari target bisa diperoleh. Dari Tabel 4.5 bisa dilihat bahwa *error* hasil pengukuran dengan metode ini bernilai lebih dari 10%. Dengan demikian, pengukuran posisi target dengan metode triangulasi lebih akurat daripada menggunakan matriks proyeksi.

4.4. Pengujian Pengukuran Posisi Target dengan Jarak Antar Kamera yang Berbeda-beda

Pengujian ini dilakukan dengan tujuan mencari jarak antar kamera yang dapat memberikan hasil pengukuran posisi target yang optimum dengan menggunakan metode triangulasi. Metode triangulasi yang digunakan adalah fungsi `triangulatePoints` yang ada di *library* OpenCV. Tabel 4.6 hingga Tabel 4.9 menunjukkan hasil pengukuran dengan jarak antar kamera yang berbeda-beda, yaitu: 10 cm, 20 cm, 30 cm, dan 40 cm. Dari tabel-tabel tersebut bisa dilihat pengukuran dengan *error* terkecil terjadi saat jarak antar kamera pada sistem kamera stereo sebesar 30 cm.

Tabel 4.6 Hasil Pengukuran Posisi dengan Jarak Kamera 10 cm

Jarak Objek (cm)	Pengukuran Posisi (cm)			
	X	Y	z	Error z (%)
150	`-38.64	47.28	143.57	4.29
278	9.94	77.58	274.65	1.21
398	-23.98	52.21	394.81	0.80
511.5	-5.18	25.01	485.92	5.00
678	-7.75	-136.88	631.69	6.83

Tabel 4.7 Hasil Pengukuran Posisi dengan Jarak Kamera 20 cm

Jarak Objek (cm)	Pengukuran Posisi (cm)			
	X	Y	z	Error z (%)
150	-36.86	48.03	145.43	3.05
278	20.18	77.82	277.93	0.03
398	-16.00	53.01	390.83	1.80
511.5	-1.00	35.39	500.26	2.20
678	-5.31	-139.37	625.33	7.77

Tabel 4.8 Hasil Pengukuran Posisi dengan Jarak Kamera 30 cm

Jarak Objek (cm)	Pengukuran Posisi (cm)			
	X	Y	Z	Error z (%)
150	-16.17	40.56	152.06	1.37
278	27.86	78.86	277.28	0.26
398	-20.99	49.99	392.82	1.30
511.5	9.80	-40.00	523.75	2.39
678	5.04	-141.04	673.40	0.68

Tabel 4.9 Hasil Pengukuran Posisi dengan Jarak Kamera 40 cm

Jarak Objek (cm)	Pengukuran Posisi (cm)			
	X	Y	Z	Error z (%)
150				
278			278.80	0.29
398	-2.01	62.97	423.56	6.42
511.5	18.32	-55.30	595.28	16.38
678	-134.28	-204.74	815.75	20.32

4.5. *Tuning* PID dan Pengujian Kontrol *Platform* Senjata Berdasarkan Sudut Horisontal dan Sudut Vertikal

Di dalam pengujian ini, nilai Kp, Ki, dan Kd diubah-ubah dan dicari agar pergerakan motor senjata berosilasi seminimum mungkin ketika diberi suatu *setpoint*. Nilai *setpoint* berasal dari GUI yang telah dirancang sebelumnya yang hanya mengirimkan data sudut vertikal dan sudut horisontal motor.

Tabel 4.10 Hasil Tuning PID untuk Motor Penggerak Horisontal

No.	Kp	Ki	Kd	Perubahan Sudut	Duty Cycle Maksimum	Simpangan Osilasi	Lama Osilasi
1.	1.0	0.0	0.0	50^o	80%	Besar	2s
2.	3.0	0.0	0.0	50^o	80%	Besar	1s
3.	3.0	0.0	2.0	50^o	80%	Besar	2s
4.	3.0	1.0	2.0	50^o	80%	Sangat besar	>10s
5.	0.9	0.0	0.5	50^o	80%	Kecil	1s
6.	1.9	0.0	0.5	50^o	80%	Kecil	5s
7.	0.9	0.0	1.1	50^o	80%	Kecil	<1s
8.	0.9	0.0	1.1	50^o	70%	Sangat kecil	<1s

Tabel 4.11 Hasil Tuning PID untuk Motor Penggerak Vertikal

No.	Kp	Ki	Kd	Perubahan Sudut	Duty Cycle Maksimum	Simpangan Osilasi	Lama Osilasi
1.	0.9	0.0	1.1	50^o	70%	Kecil	2s
2.	0.9	0.0	1.1	50^o	40%	Sangat kecil	<1s

Tabel 4.12 Pengujian Pengarahan Senjata Melalu Komputer

No.	Sudut Horisontal	Sudut Vertikal	Keterangan
1.	10^o	10^o	Respon kedua motor cepat dan osilasi sangat kecil di akhir
2.	-50^o	-15^o	Respon kedua motor cepat dan osilasi sangat kecil di akhir
3.	70^o	20^o	Respon kedua motor cepat dan osilasi sangat kecil di akhir
4.	-70^o	0^o	Respon kedua motor cepat dan osilasi sangat kecil di akhir

Tabel 4.10 menunjukkan pengujian nilai Kp, Ki, dan Kd dengan *tuning* manual untuk motor penggerak horisontal *platform* kamera. Hasilnya adalah, nilai Kp, Ki, dan Kd terbaik adalah masing-masing 0.9, 0.0, dan 1.1. Dengan demikian, kontroler terbaik untuk motor jenis yang dipakai pada TA ini adalah kontroler PD. Selain itu, nilai *duty cycle* maksimum dibatasi 70%. Dengan nilai-nilai ini, ketika motor mencapai *setpoint*, motor tidak bergetar.

Untuk motor penggerak vertikal, nilai Kp, Ki, dan Kd yang dipakai sama dengan yang dipakai untuk motor penggerak horisontal, seperti yang tertera pada Tabel 4.11. Namun, *duty cycle* maksimum untuk motor penggerak vertikal diturunkan menjadi 40%. Dengan nilai ini, osilasi tidak terjadi.

Setelah nilai Kp, Ki, dan Kd serta *duty cycle* maksimum untuk kedua motor penggerak *platform* senjata didapatkan, sistem diuji untuk menggerakkan motor vertikal dan motor horisontal bersama-sama dengan

setpoint sudut horisontal dan vertikal yang diatur melalui komputer. Hasil pengujian tertera pada Tabel 4.12. Dari pengamatan yang dilakukan, sistem memiliki respon yang cukup cepat dan osilasi hampir tidak terjadi ketika senjata sudah ada di posisi *setpoint*.

4.6. Pengujian Kontrol *Platform* Senjata Berdasarkan Posisi Kartesian

Pada pengujian ini, posisi X, Y, dan Z dimasukkan di GUI yang telah dirancang khusus untuk pengujian ini. Koordinat ini akan diproses menjadi data sudut horisontal dan vertikal. Nilai Z yang dimasukkan dibatasi agar selalu bernilai positif. Sehingga, pengujian ini difokuskan pada pengarahan senjata dengan koordinat kartesian berada di depan sistem. Tabel 4.13 berikut ini menunjukkan hasil pengujian kontrol *platform* senjata berdasarkan posisi kartesian.

Dari pengujian yang dilakukan, bisa diamati bahwa arah senjata yang dikontrol menggunakan posisi kartesia sudah sesuai dengan keinginan. Hal ini bisa dilihat dari kuadran yang ditunjuk oleh arah senjata.

4.7. Pengujian *Optical Flow* dan *Kalman filter*

Pengujian ini bertujuan untuk menghilangkan getaran pada motor akibat *noise* pengukuran. Tabel 4.14 menunjukkan hasil pengujian dari sistem ini dengan tambahan fitur *optical flow dan kalman filter*.

Tabel 4.13 Pengujian Kontrol Platform Senjata Berdasarkan Posisi Kartesian

No.	X	Y	Z	Arah Senjata
1.	3	4	10	Kanan Atas
2.	-5	3	10	Kiri Atas
3.	-4	-4	10	Kiri Bawah
4.	0	0	10	Tengah

Tabel 4.14 Pengujian Optical Flow dan *Kalman filter*

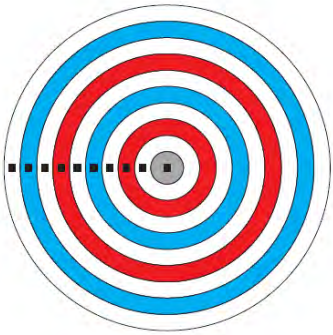
No.	Penggunaan *Kalman filter*	Osilasi Motor Saat Target Diam	*Tracking Optical Flow*
1.	Tidak	Ya	Respon cepat
2.	Ya	Tidak	Respon lambat

Dari pengujian ini, bisa diamati bahwa penggunaan *kalman filter* sangat bagus ketika target diam. Dengan menggunakan *kalman filter*, motor tidak bergetar sehingga motor tidak cepat panas. Namun, penggunaan *kalman filter* memperlambat respon dalam *tracking* target. Walaupun *tracking* di frame tetap cepat, namun motor penggerak bergerak lambat karena pergerakan motor menggunakan informasi koordinat kartesian yang telah diproses menggunakan *kalman filter*.

Untuk mengatasi hal ini, dilakukan sedikit perubahan algoritma agar kedua metode bekerja optimal. *Kalman filter* aktif terus, namun nilai prediksi posisi dari Kalman dibuat sama dengan pengukuran hasil triangulasi ketika pergeseran posisi target pada *frame* sebesar 30 pixel atau lebih. Setelah diuji lagi dengan algoritma yang dimodifikasi ini, motor tidak bergetar saat target diam dan respon *tracking optical flow* cukup cepat sehingga senjata bisa mengikuti target ketika target bergerak.

4.8. Pengujian Pengarahan Target

Dengan menggunakan metode-metode yang telah dijelaskan pada peracangan sistem, pengujian pengarahan target dilakukan untuk menganalisa keakuratan pengarahan target. Pengujian dilakukan pada malam hari di ruangan tertutup dengan pencahayaan yang cukup. Sasaran yang digunakan adalah papan target yang sesuai standar ISSF (*International Shooting Sport Federation*) untuk target 50m dengan menggunakan pistol. Gambar 4.8 di bawah ini menggambarkan papan target yang akan menjadi objek yang akan disasar oleh senjata.

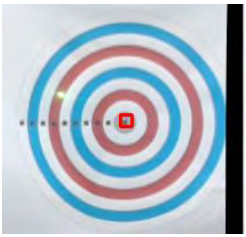


Gambar 4.8 Target standar ISSF jarak 50 m untuk penembakan dengan pistol.

Papan target jenis ini memiliki 10 ring. Ring 1 merupakan ring terluar dengan diameter 500 mm. Sedangkan Ring 10 merupakan ring terdalam dengan diameter 50 mm. Semakin tinggi nomor ring, diameter ring semakin mengecil sebesar 50 mm.

Pada pengujian ini, jarak target diubah-ubah, yaitu: 160 cm, 250 cm, 370 cm, 500 cm, dan 650 cm. Kotak merah menandakan target yang dipilih, yang dalam pengujian ini adalah Ring 10 pada papan target. Sedangkan arah dari senjata ditandai oleh laser hijau yang terpasang pada senjata. Tabel 4.15 berikut ini menunjukkan hasil pengujian pengarahan target dari sistem ini.

Tabel 4.15 Pengujian Pengarahan Target

Jarak	Target yang dipilih	Target yang disasar	Gambar
160 cm	Ring 10	Ring 4	
250 cm	Ring 10	Ring 7	

Lanjutan Tabel 4.15 Pengujian Pengarahan Target

Jarak	Target yang dipilih	Target yang disasar	Gambar
370 cm	Ring 10	Ring 8	
500 cm	Ring 10	Ring 6	
650 cm	Ring 10	Ring 3	

Dari pengujian yang telah dilakukan, bisa diamati bahwa arah senjata masih mengarah ke papan target. Namun, arah senjata tersebut tidak ada yang mengenai target yang dipilih yaitu Ring 10 pada papan target. Arah paling akurat yang dicapai adalah ketika jarak target 370 cm, yaitu saat senjata mengarah ke Ring 8 pada papan target. Target yang

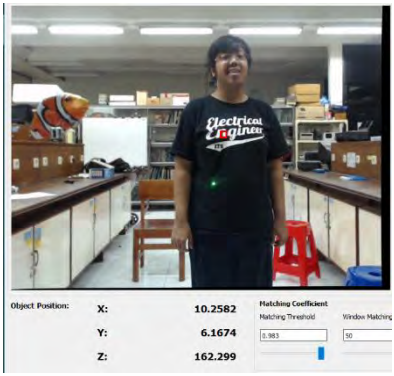
terlalu dekat menyebabkan pengarahan senjata semakin tidak akurat. Begitu pula jika target berjarak cukup jauh.

4.9. Pengujian Performa Sistem

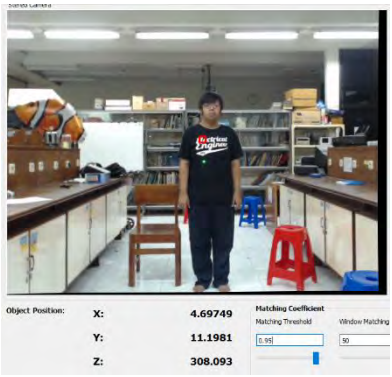
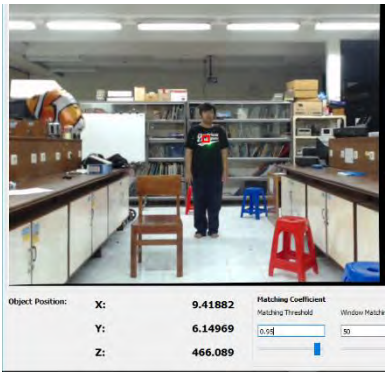
Pada pengujian ini, sistem diuji untuk mengarahkan senjata ke target berupa manusia. Sistem diuji pada dua kondisi, yakni pagi hari dan malam hari dengan pencahayaan lampu. Di dalam pengujian ini, jarak target berubah-ubah, yakni 150 cm, 300 cm, 450 cm, dan 600 cm. Laser yang ada pada senjata dinyalakan agar keakuratan pengarahan senjata bisa diamati dengan melihat posisi kotak merah, yang merupakan titik target yag dipilih, dan laser.

Pengujian malam hari dengan lampu
Tanggal Pengujian : 16 Juni 2016
Pukul : 01.09-01.16
Tempat : Lab B202 Jurusan Teknik Elektro ITS

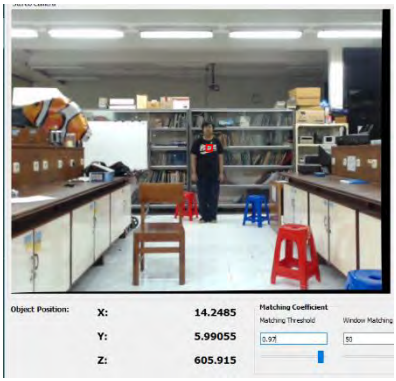
Tabel 4.16 Pengujian Performa SIstem Malam Hari dengan Lampu

Jarak	Pengamatan	Keterangan
150 cm		- Target yang dipilih ada di tulisan *Engineer* di pakaian subjek. - Arah senjata ditandai dengan laser hijau dan tidak di tulisan *Engineering* namun masih di pakaian subjek. - Pengarahan senjata kurang akurat namun masih mengarah ke target

Lanjutan Tabel 4.16 Pengujian Performa Sistem Malam Hari

Jarak	Pengamatan	Keterangan
300 cm		- Target yang dipilih ada di tulisan *Electrical* di pakaian subjek. - Arah senjata ditandai dengan laser hijau dan tidak di tulisan *Engineering* namun masih di pakaian subjek. - Pengarahan senjata kurang akurat namun masih mengarah ke target
450 cm		- Target yang dipilih ada di tulisan *Electrical* di pakaian subjek. - Arah senjata ditandai dengan laser hijau dan berada di posisi yang sama dengan kotak merah - Pengarahan senjata cukup akurat

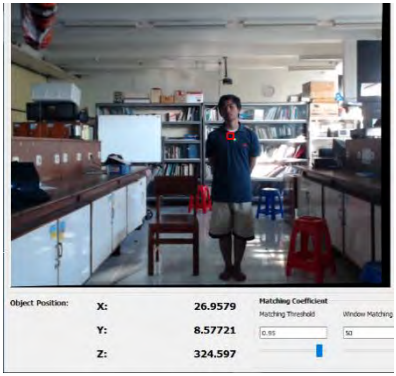
Lanjutan Tabel 4.16 Pengujian Performa Sistem Malam Hari

Jarak	Pengamatan	Keterangan
600 cm		- Target yang dipilih ada di tulisan *Electrical* di pakaian subjek. - Arah senjata ditandai dengan laser hijau dan berada di posisi yang sama dengan kotak merah - Pengarahan senjata cukup akurat

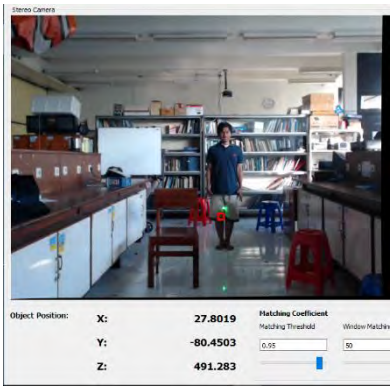
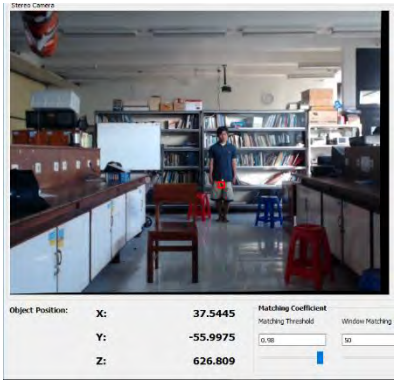
Dari pengujian yang dilakukan pada malam hari dengan kondisi pencahayaan yang terang mengguakan lampu, seperti yang tertera pada Tabel 4.16, bisa diamati bahwa arah horisontal motor masih mengarah ke target yang diinginkan. Namun, untuk arah vertikal, terjadi kesalahan ketika target berada pada jarak yang cukup dekat dengan sistem. Semakin jauh target, arah senjata semakin mengarah ke titik yang diinginkan di tubuh target.

Pengujian pagi hari
Tanggal Pengujian : 16 Juni 2016
Pukul : 01.09-01.16
Tempat : Lab B202 Jurusan Teknik Elektro ITS

Tabel 4.17 Pengujian Performa Sistem Pagi Hari

Jarak	Pengamatan	Keterangan
150 cm		- Target yang dipilih ada di dekat kerah pakaian subjek. - Arah senjata ditandai dengan laser hijau dan mengarah ke bagian pusar subjek. Pengarahan - senjata kurang akurat namun masih mengarah ke target
300 cm		- Target yang dipilih ada di dekat kerah pakaian subjek. - Arah senjata ditandai dengan laser hijau dan mengarah dekat kotak merah. - Pengarahan senjata cukup akurat

Lanjutan Tabel 4.17 Pengujian Performa Sistem Pagi Hari

Jarak	Pengamatan	Keterangan
450 cm		- Target yang dipilih ada di bagian lutut subjek. - Arah senjata ditandai dengan laser hijau dan mengarah dekat kotak merah. - Pengarahan senjata cukup akurat
600 cm		- Target yang dipilih ada di bagian pangkal paha subjek. - Arah senjata ditandai dengan laser hijau dan mengarah dekat kotak merah. - Pengarahan senjata cukup akurat

Dari pengujian yang telah dilakukan, terlihat pada Tabel 4.17, pengarahan senjata dengan target dekat, kurang dari 300 cm, menjadi kurang akurat. Namun, arah senjata masih mengarah ke subjek yang ditarget. Sedangkan pengarahan senjata yang semakin jauh megarah ke dekat titik yang ditarget.

4.10. *Real Application*

Setelah pengujian-pengujian dilakukan, akurasi dari sistem tidak cukup baik. Hal-hal yang bisa meningkatkan akurasi pengarahan senjata ke target di antaranya adalah

1. Posisi senjata terhadap kamera harus akurat agar pengukuran sudut horisontal dan vertikal senjata menjadi lebih akurat.
2. Posisi senjata terhadap kamera tidak boleh berubah sedikit pun agar akurasi pengarahan senjata tidak menurun.
3. Proses kalibrasi kamera stereo harus dilakukan berulang kali dan dicari hasil kalibrasi terbaik untuk meningkatkan akurasi pengukuran posisi target yang akan berdampak pada akurasi pengarahan senjata ke target.
4. Kamera stereo yang sudah dikalibrasi dengan baik tidak boleh berubah konfigurasi antar kedua kamera tersebut sedikit pun sehingga akurasi pengukuran posisi target tidak menurun.

Dengan beberapa syarat di atas, aplikasi nyata (*real application*) dari sistem ini di antaranya adalah untuk

1. Pertahanan Daerah Sendiri

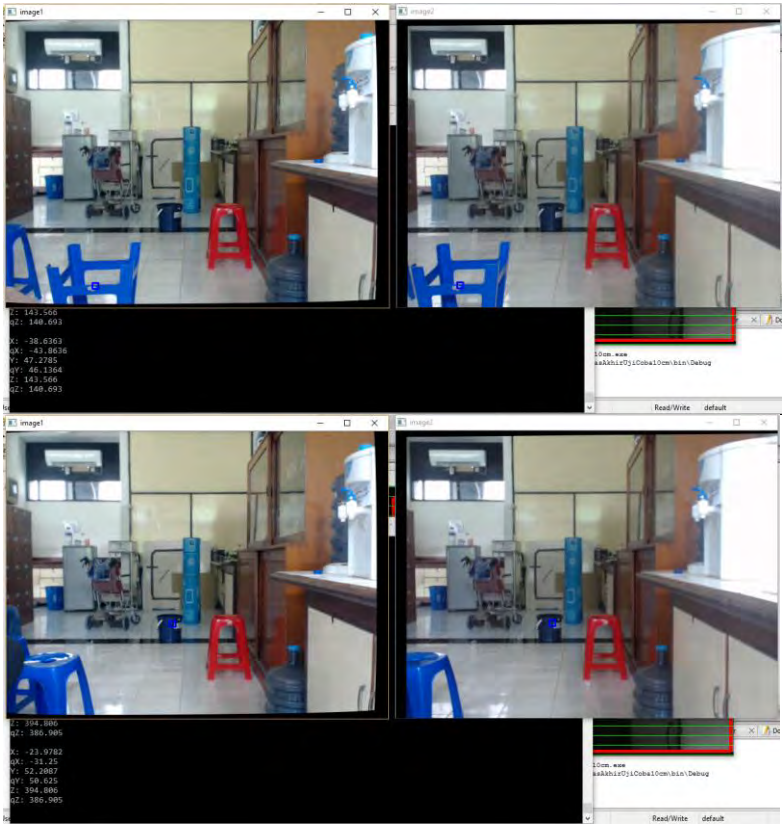
   Pemasangan sistem ini pada pertahanan statis menjadi aplikasi yang paling masuk akal untuk sistem ini. Karena adanya syarat-syarat di atas untuk meningkatkan akurasi pengarahan senjata, tentu pemasangan sistem ini oleh *engineer* akan lebih baik ketika berada di pertahanan sendiri. Karena para *engineer* akan lebih tenang memasang sistem ini dan bisa meningkatkan akurasi terutama dalam peletakan posisi senjata terhadap kamera.

2. Pengarah Senjata Bagian Depan untuk Kapal Perang

   Pemasangan sistem ini di kapal perang akan masuk akal ketika pemasangan dilakukan di pelabuhan dan tidak terjadi perang. Dengan kondisi ini, para *engineer* akan bisa meningkatkan akurasi sistem. Selain itu, pemasangan sistem pada kapal perang menjadi aplikasi yang masuk akal karena peperangan di laut tidak secepat di darat. Sehingga, penargetan musuh menggunkan sistem ini masih masuk akal untuk diaplikasikan.

LAMPIRAN

Gambar Pengujian Pengukuran Posisi dengan Pengukuran
Jarak Kamera 10 cm



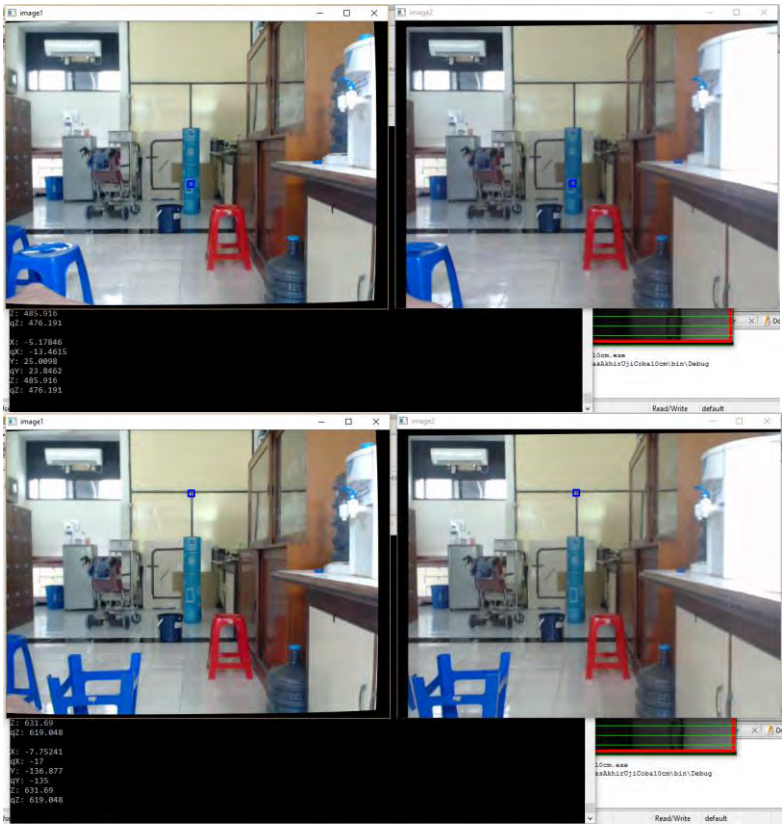
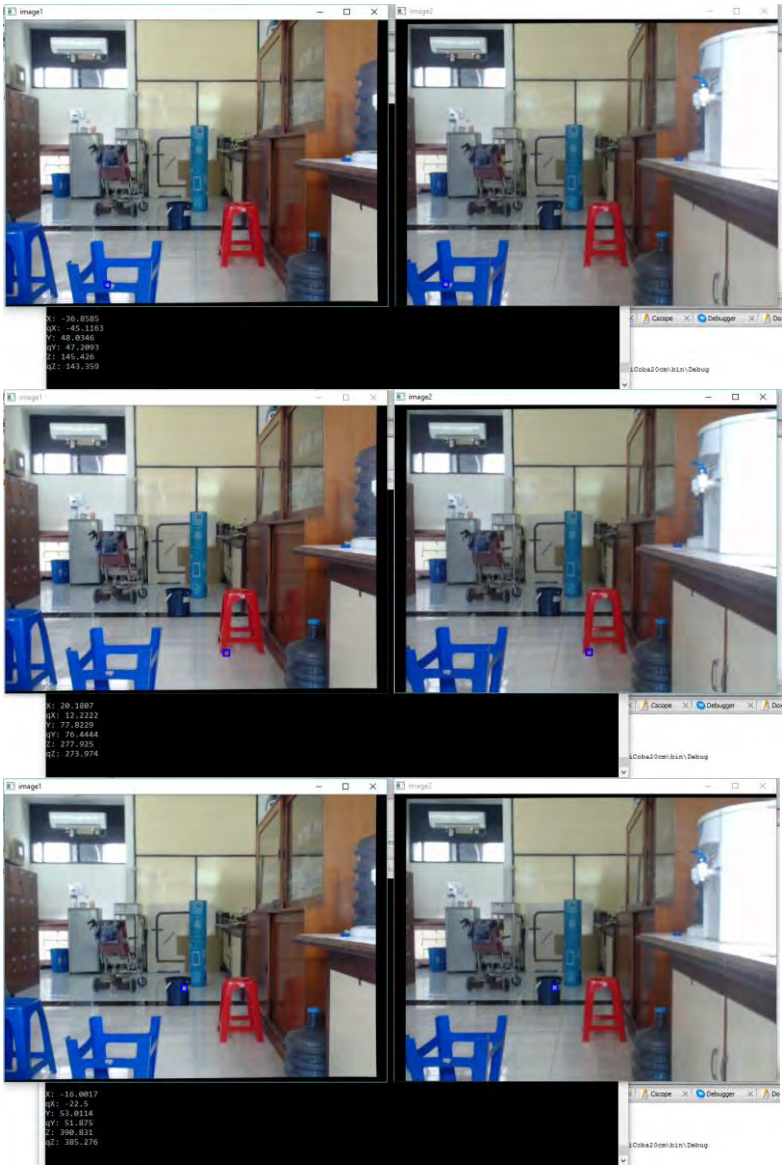


image1
image2
Z: 485.916
qZ: 476.191
X: -5.17846
qX: -13.4615
Y: 25.8898
qY: 23.8462
Z: 485.916
qZ: 476.191
Read/Write default
image1
image2
Z: 631.69
qZ: 619.048
X: -7.75241
qX: -17
Y: -136.877
qY: -135
Z: 631.69
qZ: 619.048
Read/Write default

Jarak Kamera 20 cm



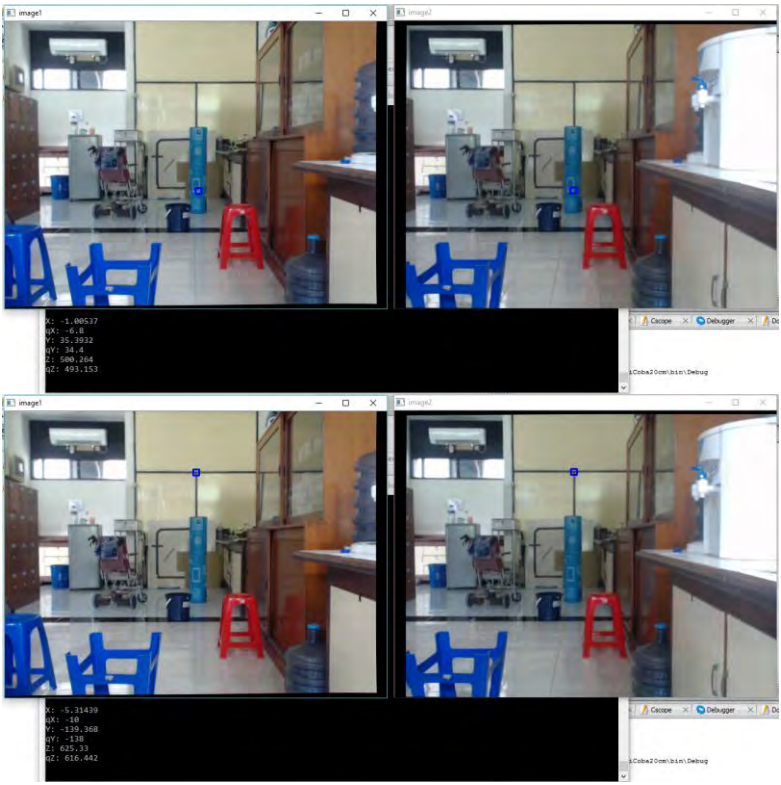
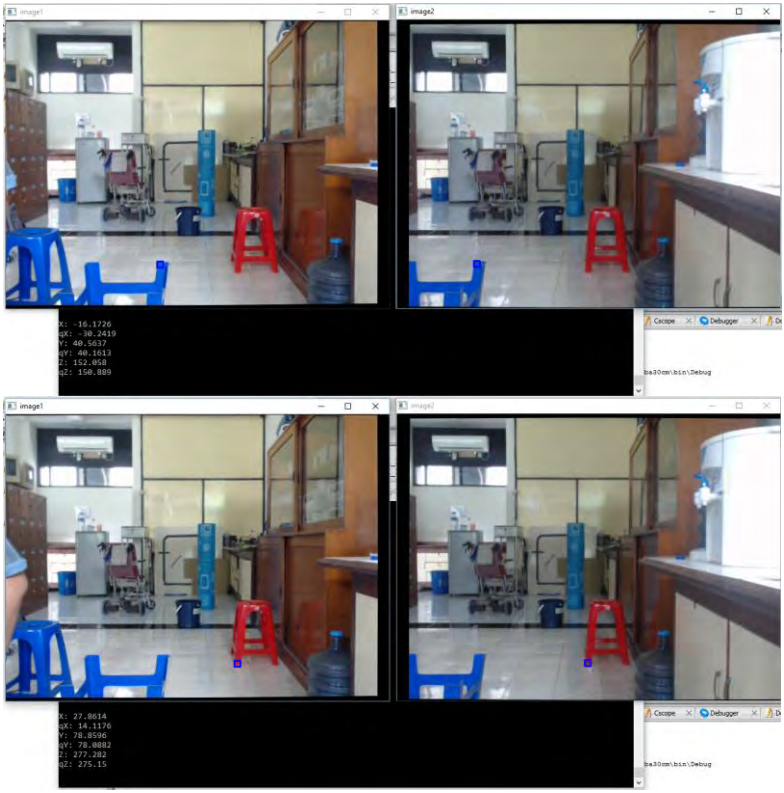


image1
image2
X: -1.00537
qX: -6.8
Y: 35.3932
qY: 34.4
Z: 500.264
qZ: 493.153
Cscope
Debugger
image1
image2
X: -5.31439
qX: -10
Y: -139.368
qY: -138
Z: 625.33
qZ: 616.442
Cscope
Debugger

Jarak Kamera 30 cm



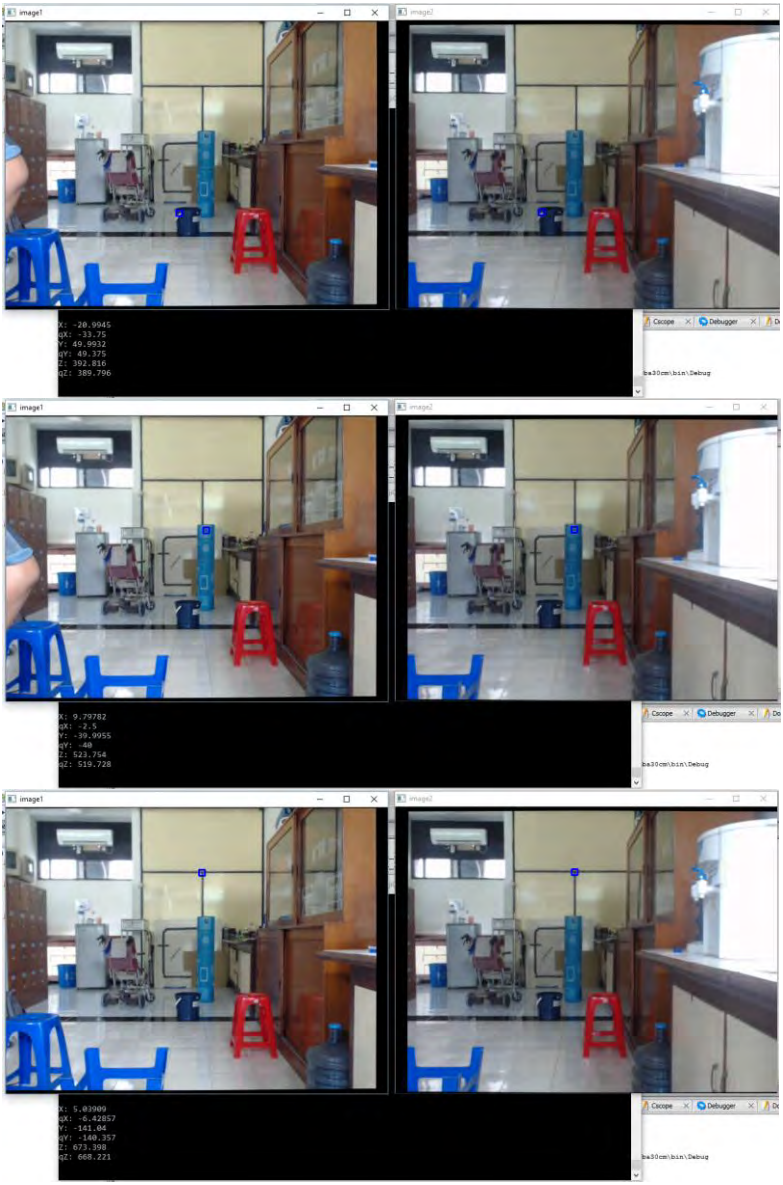


image1
image2
X: -20.9945
gX: -33.75
Y: 49.9932
gY: 49.375
Z: 392.816
gZ: 389.796
Cscope
Debugger
ba30cm\bin\Debug
image1
image2
X: 9.79782
gX: -2.5
Y: -39.9055
gY: -40
Z: 523.754
gZ: 519.728
Cscope
Debugger
ba30cm\bin\Debug
image1
image2
X: 5.03909
gX: -6.42857
Y: -141.04
gY: -140.357
Z: 673.398
gZ: 668.221
Cscope
Debugger
ba30cm\bin\Debug

Jarak Kamera 40 cm

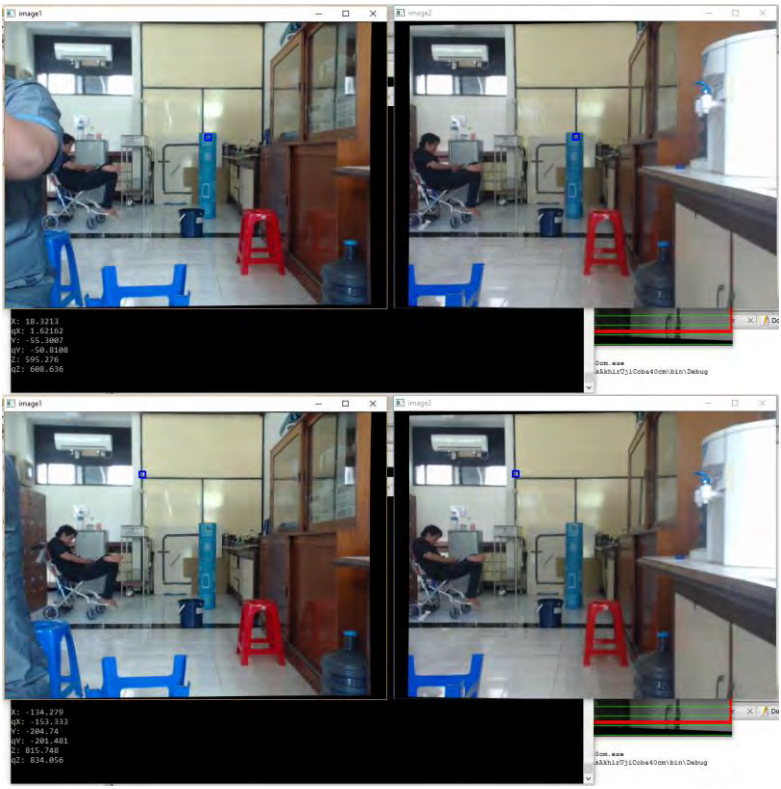


image1
image2
x: 18.3213
qX: 1.62162
Y: -55.3007
qY: -50.8108
Z: 595.276
qZ: 608.636
image1
image2
x: -134.279
qX: -153.333
Y: -204.74
qY: -201.481
Z: 815.748
qZ: 834.056

BAB V
PENUTUP

5.1. Kesimpulan

Kesimpulan yang bisa diambil oleh penulis dari tugas akhir ini adalah:

1. Pengukuran posisi objek menggunakan kamera stereo dengan metode triangulasi, baik yang ada di *library* OpenCV ataupun perhitungan manual menghasilkan kesalahan pengukuran kurang dari 6%.
2. Metode pengukuran posisi objek menggunakan kamera stereo dengan menggunakan matriks proyeksi hasil kalibrasi kamera tunggal untuk masing-masing kamera menghasilkan kesalahan pengukuran lebih dari 10%.
3. Jarak antar kamera pada stereo kamera yang menghasilkan pengukuran posisi objek yang optimum adalah 30 cm dengan kesalahan pengukuran kurang dari 2.5%.
4. Pengarahan senjata sesuai target yang dibidik di tubuh target akurat pada jarak lebih dari 300 cm.

5.2. Saran

Beberapa saran yang penulis bisa berikan untuk pengembangan tugas akhir adalah:

1. Metode pencocokan objek antara kamera kiri dan kanan harus dicari yang lebih baik dan lebih cepat.
2. Kamera yang dipakai memiliki resolusi lebih tinggi agar jarak yang terukur bisa lebih jauh.
3. Jika ingin menambahkan metode *tracking*, bisa dicari metode *tracking* yang respon terhadap pergerakan yang cepat dan dapat tidak terpengaruh halangan.
4. Kamera yang digunakan memiliki mode *night vision* atau memiliki respon rentang panjang gelombang hingga infra merah sehingga bisa dicoba untuk kondisi gelap.

DAFTAR PUSTAKA

[1] Hermawati, F. A., "Pengolahan Citra Digital Konsep & Teori", Penerbit Andi, Yogyakarta, Bab 2, 2013.

[2] Kumar, T. dan Verma, K., "A Theory Based on Conversion of RGB Image to Gray Image", International Journal of Computer Applications, vol. 7 - no. 2, pp. 7-10, September, 2010.

[3] Zou, L. dan Li, Y., "A Method of Stereo Vision Matching Based on OpenCV", International Conference on Audio, Language, and Image Processing (ICALIP), pp. 185-190, Shanghai, November, 2010.

[4] Mrovlje, J. dan Vrancic, D., "Distance Measuring Based on Stereoscopic Pictures", 9th International PhD Workshop on System and Control: Young Generation Viewpoint, Izola, October, 2008.

[5] Bradski, G. dan Kaehler, A., "Learning OpenCV Computer Vision with the OpenCV Library", O'Reilly, Sebastopol, Ch.12, 2008.

[6] Bradski, G. dan Kaehler, A., "Learning OpenCV Computer Vision with the OpenCV Library", O'Reilly, Sebastopol, Ch.7, 2008.

[7] Bradski, G. dan Kaehler, A., "Learning OpenCV Computer Vision with the OpenCV Library", O'Reilly, Sebastopol, Ch.10, 2008.

[8] Bouget, J. Y., "Pyramidal Implementation of The Lucas-Kanade Feature Tracker", Intel Coorporation, 2000.

[9] Koutroulis, E., dkk., "High-Frequency Pulse Width Modulation Implementation using FPGA and CPLD ICs", Journal of Systems Architecture 52, pp. 332-344, 2006.

[10] "UM1472 User Manual", STMicroelectronics, 2016.

[11] "Controlling DC Brush Motors with H-Bridge Driver ICs", ROHM Semiconductor, 2009.

[12] Chapman, S.J., "Electric Machinery Fundamentals 4th Edition", Mc Graw Hill, New York, Ch. 9, 2005.

[13] Basilio, J.C. dan Matos, S.R., "Design of PI and PID Controllers with Trancient Performance Specification", IEEE Transaction on Education, vol. 45 – no.4, pp. 364-370, November, 2002.

[14] Syahrul, "Pemrograman Mikrokontroler AVR Bahasa ASSEMBLY dan C", INFORMATIKA, Bandung, Bab 11, 2014.

[15] Bradski, G. dan Kaehler, A., "Learning OpenCV Computer Vision with the OpenCV Library", O'Reilly, Sebastopol, Ch.1, 2008.

[16] "OPENCV (Open Source Computer Version)" <URL:http://opencv.org>, 2016.

[17] Blanchette, J. dan Summerfield, M., “C++ GUI Programming with Qt 4”, Safari Enabled, Stoughton, 2006.

BIODATA PENULIS

Anas Maulidi Utama lahir di Sumenep pada 17 September 1992, yang merupakan anak kedua dari tiga bersaudara dari pasangan Moh. Rais dan Rafika. Penulis menyelesaikan pendidikan dasar di SDN Pajagalan 1 Sumenep dan dilanjutkan dengan pendidikan menengah di SMPN 1 Sumenep dan SMAN 5 Surabaya. Penulis sempat mengenyam bangku kuliah di salah satu institut di Bandung. Namun, karena alasan medis, penulis tidak mampu menyelesaikannya. Pada tahun 2013, penulis memulai pendidikan di jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember (ITS) Surabaya. Selama kuliah, penulis aktif membantu penyelenggaraan kegiatan dan aktif sebagai asisten laboratorium Elektronika Dasar.

Email:
anas.utama.1992@gmail.com