



UNIVERSITAS INDONESIA

**RANCANG BANGUN SISTEM KOMPUTER VISION PADA
ROBOT PENERIMA TAMU**

SKRIPSI

IRVAN JP ELLIIKA

0806315862

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2012**



UNIVERSITAS INDONESIA

**RANCANG BANGUN SISTEM KOMPUTER VISION PADA
ROBOT PENERIMA TAMU**

SKRIPSI

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik

IRVAN JP ELLIIKA

0806315862

**FAKULTAS TEKNIK
PROGRAM STUDI TEKNIK ELEKTRO
DEPOK
JUNI 2012**

HALAMAN PERNYATAAN ORISINALITAS

**Skripsi ini adalah hasil karya saya sendiri
Dan semua sumber baik yang dikutip maupun dirujuk
telah saya nyatakan dengan benar.**

Nama : Irvan JP Elliika

NPM : 0806315862

Tanda Tangan : 

Tanggal : 13 Juni 2012

HALAMAN PENGESAHAN

Skripsi ini diajukan oleh :
Nama : Irvan JP Elliika
NPM : 0806315862
Program Studi : Teknik Elektro
Judul Seminar : Rancang Bangun Sistem Komputer Vision Pada
Robot Penerima Tamu

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik di Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

Dewan Penguji

Pembimbing :


Dr. Ir. Abdul Muis, ST., M. Eng



Penguji :

Ir. Wahidin Wahab, M. Sc., Ph.D



Prof. Dr.Eng. Drs.Benyamin Kusumoputro, M.Eng. ()

Ditetapkan di : Depok

Hari/Tanggal : 25 Juni 2012

UCAPAN TERIMA KASIH

Segala puji syukur kehadirat Allah SWT penulis haturkan atas limpahan rahmat dan karunia sehingga penyusunan skripsi ini dapat diselesaikan dengan baik. Skripsi ini berjudul Rancang Bangun Sistem Komputer Vision Pada Robot Penerima Tamu diajukan untuk memenuhi salah satu persyaratan memperoleh gelar sarjana teknik. Disamping itu, penulis juga berharap hasil skripsi ini dapat memberikan kontribusi bagi perkembangan di dunia robotika.

Penulis menyadari bahwa tanpa adanya dukungan dan bantuan dari berbagai pihak selama masa perkuliahan sampai pada penyusunan skripsi ini sangatlah sulit bagi penulis untuk menyelesaikan skripsi ini. oleh karenanya, penulis ingin mengucapkan terima kasih kepada :

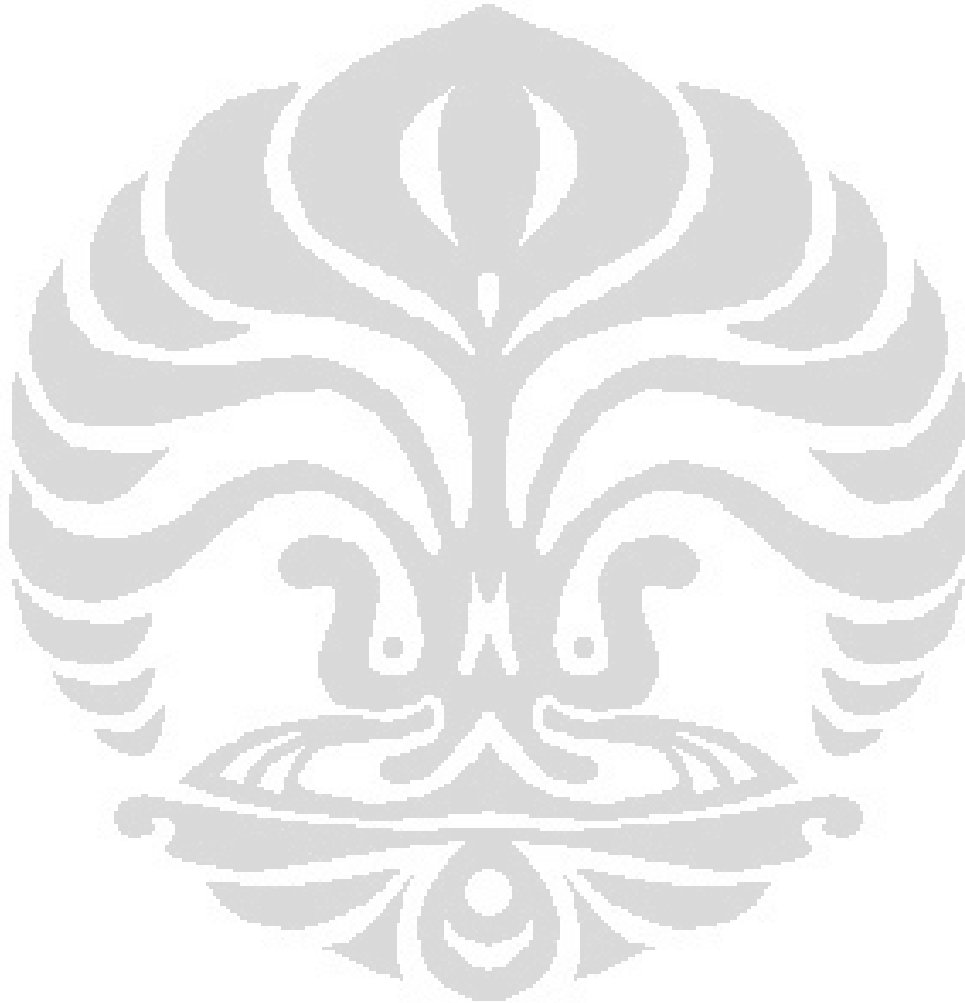
- 1) Dr.Abdul Muis, S.T, M.Eng sebagai pembimbing skripsi yang telah memberikan banyak saran baik berupa tenaga dan pikiran dan peminjaman peralatan elektronika
- 2) Orang tua, Nelli dan Alm. Iskandar Ahmad dan keluarga penulis, Paman, Om, Tante, Bunda, Bang Chandra, Reza, Rizki yang telah memberikan banyak sekali bantuan dukungan material dan moral
- 3) Sahabat – sahabat dari Tim Robot UI khususnya Kholis, Wisnu, Anto, Hadid, Imam, Iqbal, Alan, Nanda, dll yang dalam 4 tahun ini kita selalu bersama – sama dan bekerja sama.
- 4) Sahabat – sahabat dari Tim Robot UI KRCI divisi Battle khususnya Bang Udin, Bang Candra, Ryan, Priza, Budi, Anwar, Niken, Wuri, Fredrik, Crisman, Ihsan, dan lain-lain yang baik secara langsung maupun tidak langsung memberikan bantuan berupa saran dan ilmu yang sangat membantu penyusunan skripsi ini
- 5) Sahabat – sahabat dari Tim Roboboat UI khususnya Uki, Meisar, Ayun, dan haski yang telah bersama-sama bekerja keras untuk menuju negeri Paman Sam.
- 6) Sahabat - sahabat dari keluarga besar Teknik Elektro angkatan 2008 yang selalu ada baik suka maupun duka

7) Sahabat-sahabat dari keluarga besar SAMAN UI yang telah bersama-sama berjuang demi mengharumkan nama besar Tanah Rencong.

Akhirnya, penulis mengharapkan saran yang membangun terhadap skripsi ini sehingga pada kesempatan mendatang akan lebih baik lagi. Demikian, semoga skripsi ini bermanfaat.

Depok, Juni 2012

Penulis



**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS
AKHIR UNTUK KEPENTINGAN AKADEMIS**

Sebagai sivitas akademik Universitas Indonesia, saya yang bertandatangan di bawah ini:

Nama : Irvan JP Elliika
NPM : 0806315862
Program Studi : Teknik Elektro
Departemen : Teknik Elektro
Fakultas : Teknik
Jenis Karya : Skripsi

demikian demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul:

RANCANG BANGUN SISTEM KOMPUTER VISION PADA ROBOT

PENERIMA TAMU

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia/format-kan, mengelolanya dalam bentuk pangkalan data (database), merawat, dan memublikasikan tugas akhir saya tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok

Pada tanggal : 13 Juni 2012

Yang menyatakan



(Irvan JP Elliika)

ABSTRAK

Nama : Irvan JP Elliika
Program Studi : Teknik Elektro
Judul Skripsi : Rancang Bangun Sistem Komputer Vision Pada Robot Penerima Tamu

Salah satu kemampuan robot yang canggih adalah mampu melakukan adaptasi pada lingkungan sekitarnya. Kemampuan ini merupakan analogi terhadap kemampuan manusia secara khusus. Namun, kebanyakan robot yang dibuat masih terbatas dalam hal interaksi secara sentuhan dengan lingkungan sekitarnya. Oleh karenanya diperlukan sistem sensasi non-kontak yang salah satunya adalah sensasi secara visual. Cara ini termasuk salah satu yang paling *advance* karena hampir semua proses manipulasi bisa dilakukan dengan hanya menggunakan sensor visual yaitu kamera walaupun *computational cost*-nya cukup tinggi.

Single Board computer jenis BeagleBoard akan digunakan untuk melakukan komputasi sensasi visual yang meliputi *face detection*, *stereo vision*, dan bahkan lokalisasi nantinya. Wajah manusia yang akan dikenali oleh sistem computer visualnya akan di-*tracking* dan diukur jaraknya secara *real time* melalui teknik *stereo vision*. Koordinat yang didapat akan ditransformasikan dengan persamaan kinematik berupa invers jacobian menuju pusat robot untuk melakukan aktuasi pada aktuator vision dan navigasi robot secara keseluruhan sampai tujuan untuk melakukan interaksi dengan manusia tercapai. Berdasarkan pengujian yang telah dilakukan dapat dinyatakan bahwa sistem komputer vision yang telah dibangun cukup valid dan handal untuk jarak dibawah 100 cm walaupun dengan waktu komputasi yang cukup besar.

Keyword : BeagleBoard, *stereo vision*, sensasi non-kontak, invers jacobian, *face detection*.

ABSTRACT

Name : Irvan JP Elliika
Program : Electrical Engineering
Title : Computer Vision System Development for Receptionist Robot

One of the advance robot's ability is it can adapt into the around environment. This ability itself is the analogy of human's. But now, most of the robots still have limited in contact sensation. So, it's needed to build non-contact sensations and one of them is reached by build visual system. This way belong to one of advance method because almost of manipulation way can be dealt with this visual sensor like camera, even though the computational cost is high enough.

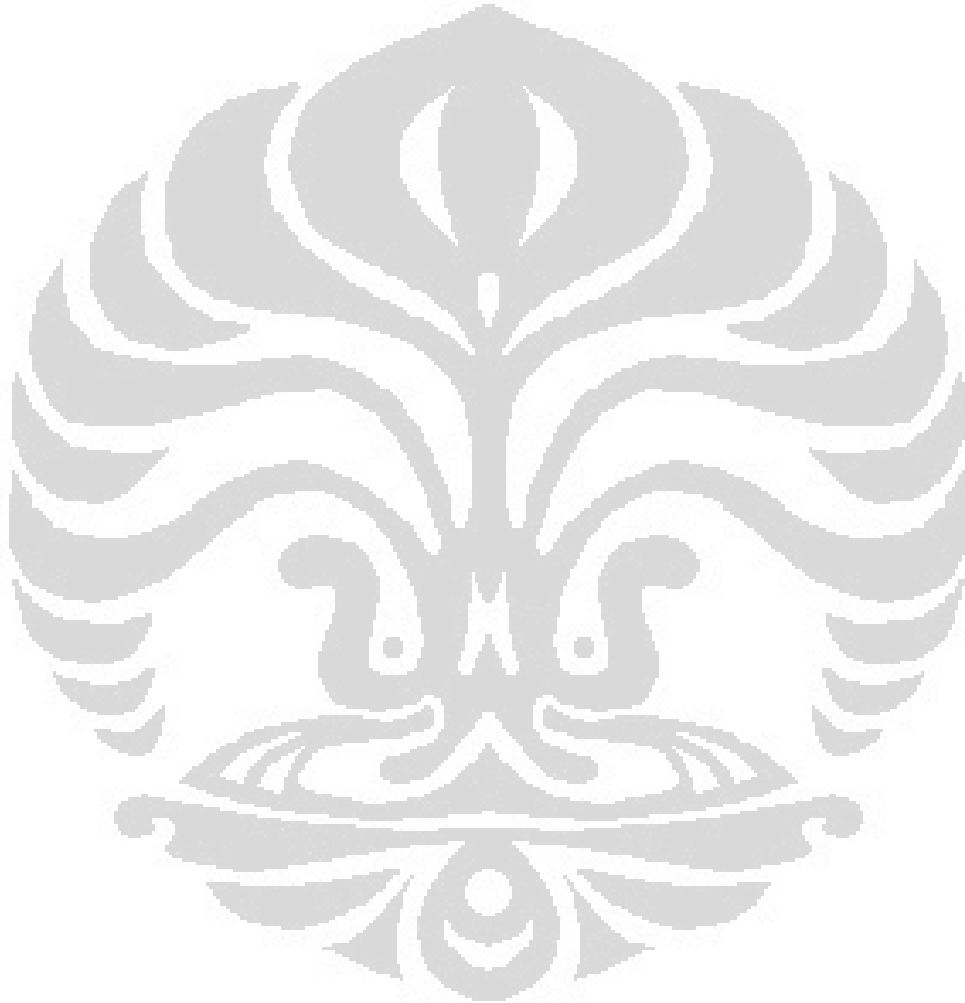
BeagleBoard, a kind of powerful *Single Board computer*, will be use to compute the visual sensation in this receptionist robot include *face detection*, *stereo vision*, and even localization later. The face of human that will be recognized by visual computer system will be tracked and the distant is calculated real time via *stereo vision* system. The coordinate that has been gathered will be transformed by invers jacobian into the center of robot to actuate visual actuation and doing robot navigation until receptionist robot is able to do interaction with human. Based on the result of experiment, it can be stated that the developed computer vision system is valid and reliable enough for distant below 100 cm even though spends high computational time.

Keyword : BeagleBoard, *stereo vision*, non-contact sensation, invers jacobian, *face detection*.

DAFTAR ISI

HALAMAN PERNYATAAN ORISINALITAS ...Error! Bookmark not defined.	
HALAMAN PENGESAHAN	Error! Bookmark not defined.
UCAPAN TERIMA KASIH	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS	
AKHIR UNTUK KEPENTINGAN AKADEMIS. Error! Bookmark not defined.	
ABSTRAK	vii
ABSTRACT	viii
DAFTAR ISI	ix
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiii
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Perumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Pembatasan Masalah.....	5
1.5 Metode Penyelesaian Masalah	5
1.6 Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	6
2.1 BeagleBoard.....	6
2.2 Sistem Operasi Linux	8
2.3 Mikrokontroler	10
2.4 Sistem Debugging dan Remoting.....	11
2.4.1 Koneksi PC dengan BeagleBoard melalui LAN	12
2.4.2 Koneksi BeagleBoard dengan mikrokontroler melalui serial	15
2.5 OpenCV	19
2.6 Model Kamera dan Proyeksi	20
2.6.1 Model Kamera.....	21
2.6.2 Geometri Proyeksi	23
2.7 Face Detection.....	23
2.7.1 Pendahuluan	23
2.7.2 Metode Face Detection	24
2.8 Stereo Vision.....	28
2.8.1 Pendahuluan	28
2.8.2 Metode <i>Stereo Vision</i> pada library OpenCV.....	36
BAB 3 PERANCANGAN SISTEM	37
3.1 Pendahuluan	37
3.2 Blok Diagram Sistem.....	39
3.3 Algoritma	40
3.4 Pengaturan Perangkat Keras	43
3.5 Kinematika Sistem Vision	46
3.5.1 Pemodelan Aktuator Sistem Vision.....	47

3.5.2 <i>Tracking Control</i> Aktuator Sistem Vision.....	48
BAB 4 PENGUJIAN DAN ANALISIS.....	51
4.1 Kalibrasi Dual Kamera	51
4.2 Face detection.....	54
4.3 Stereo Vision.....	59
4.4 Design Output untuk Interaksi dengan Manusia	64
BAB 5 KESIMPULAN	67
DAFTAR ACUAN	69
DAFTAR REFERENSI	70



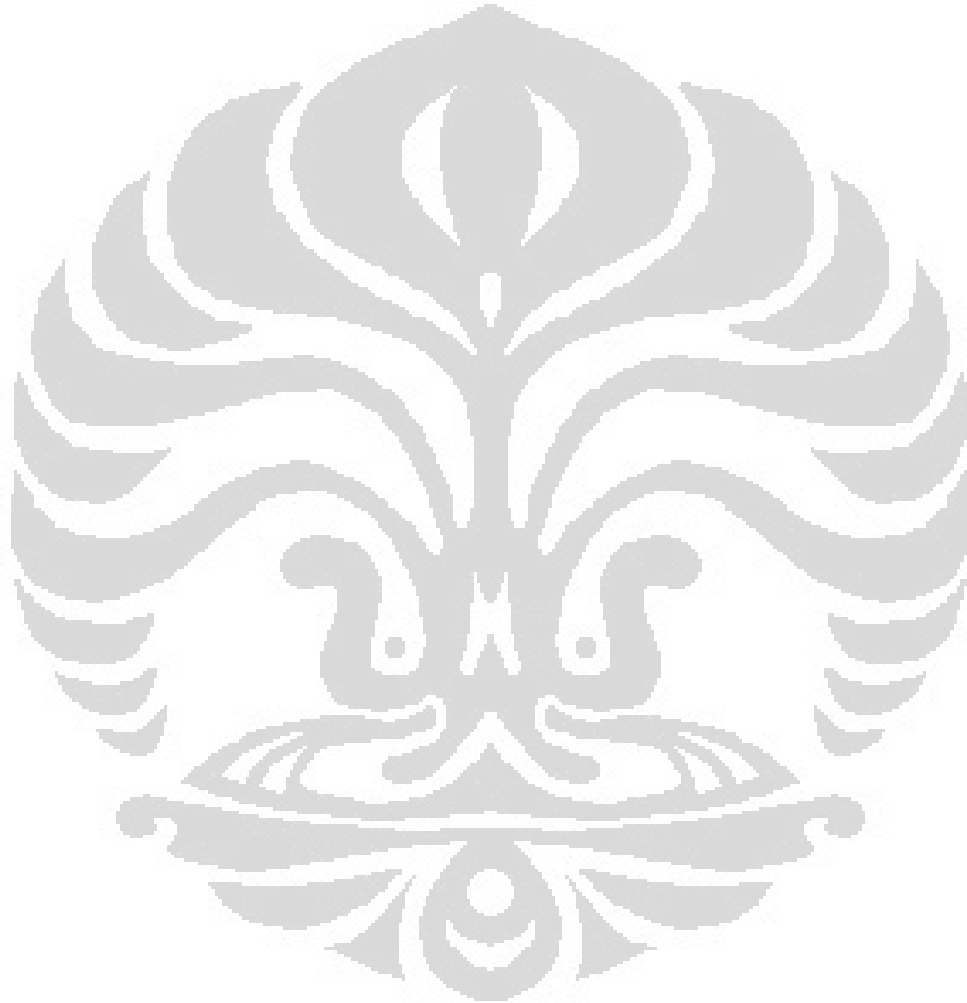
DAFTAR GAMBAR

Gambar 1-1 Robot ASIMO yang Dikembangkan oleh Honda Motor Co., Ltd.	2
Gambar 1-2 Robot Penerima Tamu yang Sedang Dikembangkan.....	3
Gambar 2-1 BeagleBoard.....	6
Gambar 2-2 DB-25 dan DB-9	17
Gambar 2-3 Konverter Serial USB.....	17
Gambar 2-4 Model Kamera dengan <i>Pinhole</i> dan Bidang Image Berada di Belakang <i>Pinhole</i>	21
Gambar 2-5 <i>Front Projection Model</i> Kamera.....	22
Gambar 2-6 Haar Features Pada OpenCV	25
Gambar 2-7 (1) Proses Integrasi (2) <i>Rectangle</i> Dibagi Menjadi Beberapa Segmen	26
Gambar 2-8 <i>Haar Classifier</i> pada OpenCV	27
Gambar 2-9 Hasil <i>Classifier</i>	28
Gambar 2-10 Geometri Ideal dari Sistem <i>Stereo Vision</i>	29
Gambar 2-11 Cara Kerja <i>Stereo Vision</i> pada Mata Manusia	29
Gambar 2-12 Hasil Gambar Akibat <i>Radial Distortion</i>	31
Gambar 2-13 Hasil Gambar Akibat <i>Tangential Distortion</i>	32
Gambar 2-14 Tujuan Akhir dari Proses Rektifikasi	33
Gambar 2-15 a. Image Hasil dari <i>Capture</i> atau Snapshot b. Image yang Telah Dihilangkan Distorsinya, c. Image yang Telah Direktifikasi, d. Image Hasil <i>Crop</i> yang Siap Dilakukan Proses <i>Correspondence</i>	34
Gambar 2-16 Prosedur Umum dari Korespondensi yang Menghasilkan <i>Disparity Map</i>	34
Gambar 2-17 Bagian yang Terang Memiliki Kedalaman yang Lebih Kecil Daripada yang Gelap.....	36
Gambar 3-1 Skematik Perancangan Sistem <i>Stereo Vision</i> Secara Ideal.....	37
Gambar 3-2 Implementasi Sistem <i>Stereo Vision</i>	38
Gambar 3-3 Blok Diagram Robot Penerima Tamu Secara Keseluruhan.....	39
Gambar 3-4 Flowchart Algoritma Sistem	41
Gambar 3-5 Informasi untuk Pergerakan Aktuator Sistem Vision.....	42
Gambar 3-6 Webcam e-1000 Logitech.....	44
Gambar 3-7 Konfigurasi <i>Cross</i> dan <i>Straight</i> RJ-45	44
Gambar 3-8 ATEN USB Serial	45
Gambar 3-9 MAX232 Serial-TTL Konverter	45
Gambar 3-10 LCD <i>Dot Matriks</i>	46
Gambar 3-11 Kabel AWG.....	46
Gambar 3-12 Model Denafit-Hartenberg Aktuator Sistem Vision	47

Gambar 3-13 Blok Diagram <i>Tracking Control</i> Sistem Aktuator Vision.....	49
Gambar 3-14 <i>End-effector</i> Robot Penerima Tamu Terhadap Titik Pusat <i>Base</i>	50
Gambar 3-15 Contoh Pergerakan dari Keseluruhan Robot.....	50
Gambar 4-1 Hasil <i>Snapshot</i> pada Kamera Kanan dan Kiri yang Dilakukan pada Saat Bersamaan.....	52
Gambar 4-2 Grafik Error Terhadap Image <i>Chessboard</i> Hasil dari <i>Snapshot</i>	53
Gambar 4-3 Grafik Error Terhadap Image <i>Chessboard</i> Hasil dari <i>Snapshot</i>	53
Gambar 4-4 Grafik untuk Percobaan <i>Face Detection</i> dengan Jarak 40 cm (Waktu Dikali dengan 20)	55
Gambar 4-5 Grafik untuk Percobaan <i>Face Detection</i> dengan Jarak 60 cm (Waktu Dikali dengan 20)	56
Gambar 4-6 Grafik untuk Percobaan <i>Face Detection</i> dengan Jarak 80 cm (Waktu Dikali dengan 20)	57
Gambar 4-7 Grafik untuk Percobaan <i>Face Detection</i> dengan Jarak 100 cm (Waktu Dikali dengan 20)	58
Gambar 4-8 Grafik untuk Percobaan <i>Face Detection</i> dengan Jarak 120 cm (Waktu Dikali dengan 20)	59
Gambar 4-9 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 40 cm (Waktu Dikali dengan 20)	60
Gambar 4-10 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 60 cm (Waktu Dikali dengan 20).....	61
Gambar 4-11 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 80 cm (Waktu Dikali dengan 20).....	62
Gambar 4-12 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 100 cm (Waktu Dikali dengan 20).....	63
Gambar 4-13 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 120 cm (Waktu Dikali dengan 20).....	64

DAFTAR TABEL

Table 2-1 Spesifikasi BeagleBoard XM rev C.....	7
Table 3-1 Parameter Denavit-Hartenberg Sistem Aktuator Vision.....	47
Table 4-1 Error Hasil Kalibrasi <i>File Image</i>	54



BAB 1 PENDAHULUAN

1.1 Latar Belakang

Perkembangan robot di seluruh belahan dunia saat ini sudah menuju ke arah yang sangat signifikan. Mulai dari negara berkembang sampai Negara maju alias negara super power seperti Amerika, Jepang, Rusia dan lain sebagainya seperti berlomba-lomba dan sangat memberi apresiasi pada bagian *research* ini. Robot-robot yang dihasilkannya pun bukan lagi sekedar hanya untuk menyelesaikan misi-misi khusus dan sederhana seperti pada robot terbang secara semiotomatis untuk *surveillance* dan perlombaan robot baik seperti *fire fighting*, robo sub, robo soccer dan lain sebagainya yang memiliki sedikit interaksi pada lingkungannya (seperti manusia dan benda di sekitar) khususnya pada lingkungan yang dinamikanya cukup tinggi. Namun Penelitian untuk robot sekarang lebih diarahkan pada sistem robot yang mampu dengan baik berinteraksi dengan lingkungannya. Oleh karenanya, para *engineer* robot di seluruh dunia pasti akan sangat senang jika pada kenyataannya robot-robot yang akan dihasilkan sudah memiliki kemampuan seperti pada film I,Robot dimana para robot sudah menjadi bagian dari kehidupan manusia.

Salah satu tipe robot yang harus memiliki interaksi yang bagus adalah *service robot*. *service robot* itu sendiri sebenarnya memiliki tingkat kompleksitas yang sangat tinggi. Para *engineer* yang bergerak di bidang tersebut selalu berusaha agar *service robot*-nya dapat menghampiri kemampuan manusia. Kompleksitas tersebut terdiri dari berbagai macam bidang, misal: tentang struktur dan mekanisme sistem mekanik, kehandalan dan kapabilitas pengendali, sistem vision sensor, sistem navigation dan aktuator, kecerdasan buatan, teknologi mobilitas, teknologi komunikasi dan kerjasama, teknik pentautan sub-sistem, dan sebagainya. Penulis berpendapat jika semua kompleksitas itu dapat diselesaikan, maka bukan tidak mungkin robot-robot seperti dalam film I,Robot dapat menjadi kenyataan nantinya.

Secara khusus, kebanyakan *service robot* memiliki keterbatasan dalam hal interaksi secara bersentuhan (*contact sensation*). Seharusnya sebagaimana

yang dilakukan oleh manusia ketika berinteraksi dengan manusia lainnya, robot-robot tersebut dapat melakukan interaksi secara *non-contact sensation* baik secara diam maupun sedang dalam pergerakan. Namun akan sangat baik jika robot tersebut dapat melakukan kedua tipe interaksi tersebut karena akan membuat robot tersebut lebih *friendly*.

Sistem vision pada *service robot* memainkan peranan yang sangat penting dalam menyelesaikan masalah *non-contact sensation* pada setiap *service robot*. Layaknya penggunaan mata pada manusia, sistem vision pada *service robot* diharapkan juga mampu memiliki kemampuan persepsi terhadap suatu benda, mengenali objek, menentukan jarak relative benda, melakukan lokalisasi terhadap suatu lokasi misalnya ruangan, lorong, dan sebagainya. Oleh karena fungsinya yang vital, sistem vision sudah sangat diminati oleh banyak engineer robot di dunia. Banyak sekali perkembangan baik dari segi metode maupun penemuan baru dalam dunia sistem vision.



Gambar 1-1 Robot ASIMO yang Dikembangkan oleh Honda Motor Co., Ltd.

<http://jeckyglasster.blogspot.com/2011/01/asimo.html>

Wujud nyata dari jenis *service robot* adalah robot asimo yang telah didemokan dengan baik untuk berinteraksi dengan manusia melalui kemampuannya dalam mendeteksi wajah manusia di sekitarnya walaupun masih

terdapat hal-hal yang belum dapat dilakukan oleh Asimo. Perkembangan *service robot* pun akan bertambah lagi dengan adanya kompetisi yang memperlombakan robot yang mampu menjadi pelayan bagi manusia pada tingkat internasional, oleh karena itu sekali lagi, kemampuan sistem vision sangatlah diperlukan untuk perkembangan teknologi jenis robot ini.

Tingkat kesulitan yang tinggi pada penerapan sistem vision di *service robot* menjadi tantangan tersendiri bagi para insinyur robot untuk membuat robot yang mampu mengatasi masalah-masalah yang ditemui pada *service robot*. Dengan demikian, sistem vision menjadi hal yang menarik untuk dikembangkan karena fungsinya yang vital dan gunanya sangat tinggi untuk diaplikasikan. Oleh karenanya, penulis memutuskan untuk berfokus pada bidang ini khususnya penggunaan stereovision pada robot.



Gambar 1-2 Robot Penerima Tamu yang Sedang Dikembangkan

Jenis *service robot* pada penelitian ini adalah robot penerima tamu dan pada penelitian ini penulis akan membahas *stereo vision* dalam implementasi sebagai kepala robot penerima tamu. Gambar 1.2 adalah robot penerima tamu yang sedang dikembangkan. Robot ini nantinya akan mencari dimana letak manusia melalui *face detection* dan mendekatinya sehingga akhirnya dapat berinteraksi dengan manusia seperti melakukan jabat tangan, *says "hello"*, dan

lain sebagainya. Oleh karena itu dalam penelitian ini, sistem vision pada robot penerima tamu meliputi teknik stereovision, *face detection*, transformasi lokasi manusia terhadap robot penerima tamu. Dalam mendesain sebuah sistem vision pada *service robot* yang mampu melakukan persepsi seperti yang telah dijelaskan di atas, banyak hal yang harus diperhatikan seperti kondisi lingkungan baik objek yang ada, perubahan cahaya, dan lain-lain, kecepatan proses, fisik robot secara keseluruhan, komunikasi dengan aktuator, dan lain sebagainya. Oleh karenanya, banyak ahli komputer vision yang telah menerapkan metode untuk menyelesaikan masalah di atas seperti melakukan kalibrasi, mengoptimalkan kode program, membuat sistem komunikasi yang reliable dan cepat, dan sebagainya yang mampu menyelesaikan masalah keterbatasan robot dan meningkatkan kemampuan robot itu sendiri.

1.2 Perumusan Masalah

Salah satu kekuatan manusia dalam melakukan interaksi terhadap lingkungan sekitarnya adalah terletak pada mata. Dengan menggunakan mata, manusia dapat mengetahui setiap objek secara langsung dan dapat melakukan persepsi terhadap apa yang dilihatnya. Oleh karenanya sensor mata ini memiliki peran yang sangat besar dalam diri manusia. Hal inilah yang membuat sistem visual pada robot menjadi salah satu aspek paling penting pada perkembangannya. Dengan menggunakan kamera sebagai analogi dari mata, maka penggunaan sensor *exteroceptive* dasar seperti range finder, proximity, dan lain-lain dapat digantikan dengan sempurna. Dengan kata lain, para insinyur robot memiliki tugas penting untuk memaksimalkan fungsi sensor ini. Berdasarkan paparan yang lalu, maka hal yang dapat dilakukan sekarang adalah menganalisis sistem vision yang dijalankan di *development board* BeagleBoard yang akan berkomunikasi dengan slave mikrokontroler yang mengatur fungsi sensor dasar dan aktuator robot penerima tamu.

1.3 Tujuan Penelitian

Penelitian ini bertujuan untuk menggunakan teknik stereovision dengan dua kamera untuk mencari wajah manusia (*face tracking*) dan mendapatkan variabel yang dapat menentukan kelayakan dan kemampuan sistem vision baik pada BeagleBoard.

1.4 Pembatasan Masalah

Pembahasan dalam skripsi kali ini meliputi penggunaan BeagleBoard, penggunaan program *stereo vision*, *face detection* pada library OpenCV, dan komunikasi serial antara BeagleBoard dan mikrokontroler.

1.5 Metode Penyelesaian Masalah

Metode Penyelesaian masalah yang digunakan pada penyusunan skripsi ini adalah sebagai berikut :

- 1) Studi literature mengenai sistem yang sudah pernah dikembangkan di Universitas Indonesia
- 2) Pendekatan diskusi dengan pembimbing skripsi
- 3) Perakitan *hardware* dan perancangan perangkat lunak
- 4) Percobaan dan analisis hasil

1.6 Sistematika Penulisan

Setelah bab ini, akan dipaparkan penjelasan mengenai perangkat sistem seperti development board yang digunakan yaitu BeagleBoard, sistem operasi Linux, mikrokontroler, proses debugging dan remoting, library OpenCV, program *face detection*, dan *stereovision*; dalam bab 2. Penjelasan tentang rancang bangun dari eksperimen yang akan dilakukan berada pada bab 3, dilanjutkan dengan bab 4 yang berisi tentang pengujian dan analisis hasilnya. setelahnya, kesimpulan akan dipaparkan di bab 5. Daftar acuan dan daftar referensi menjadi bagian akhir dari penyusunan skripsi ini.

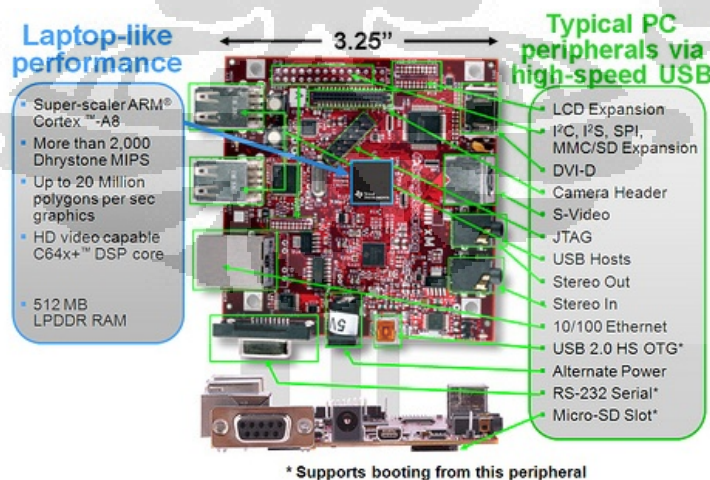
BAB 2

LANDASAN TEORI

Pada bab ini akan dijelaskan seluruh bagian perangkat baik hardware maupun software yang nantinya akan diintegrasikan menjadi satu modul yang dapat merepresentasikan sebagai suatu sistem robot penerima tamu. Perangkat sistem ini adalah sebagai berikut :

2.1 BeagleBoard

BeagleBoard adalah single board computer yang berukuran kecil (3 inci × 3 inci) yang didesain oleh Texas Instruments. BeagleBoard sendiri didesain secara khusus untuk keperluan pengguna open source. Pada dasarnya BeagleBoard adalah sistem yang multipurpose artinya beagleboard dapat digunakan untuk aplikasi apa saja yang membutuhkan komputasi. Pada BeagleBoard sendiri dapat diinstal sistem operasi yang biasa kita instal pada laptop atau komputer PC kebanyakan seperti Linux, Android, dan sebagainya. Namun perlu diperhatikan bahwa BeagleBoard tidak menggunakan prosesor intel seperti kebanyakan komputer standar lainnya sehingga program atau aplikasi yang dikompilasi untuk komputer intel tidak bisa dijalankan pada BeagleBoard. Untuk itu kita harus mendapatkan source code program tersebut dan meng-compilanya terhadap arsitektur prosesor ARM.



Gambar 2-1 BeagleBoard

Sumber : <http://www.piconode.org/lox/img/lox-beagle1.jpg>

BeagleBoard dilengkapi dengan sistem set fitur yang minimum untuk memudahkan pengguna, sehingga secara tersirat BeagleBoard tidak ditujukan untuk pengembangan platform secara penuh seperti yang bisa dilakukan pada komputer PC ataupun laptop.

Secara lengkap fitur dari BeagleBoard dirangkum pada table berikut :

Table 2-1 Spesifikasi BeagleBoard XM rev C

	Feature	
Processor	Texas Instruments Cortex A8 1GHz processor	
POP Memory	Micron 4Gb MDDR SDRAM (512MB) 200MHz	
PMIC TPS65950	Power Regulators	
	Audio CODEC	
	Reset	
	USB OTG PHY	
Debug Support	14-pin JTAG	GPIO Pins
	UART	3 LEDs
PCB	3.1" x 3.0" (78.74 x 76.2mm)	6 layers
Indicators	Power, Power Error	2-User Controllable
	PMU	USB Power
HS USB 2.0 OTG Port	Mini AB USB connector	
	TPS65950 IF	
USB Host Ports	SMSC LAN9514 Ethernet HUB	
	4 FS/LS/HS	Up to 500ma per Port if adequate power is supplied
Ethernet	10/100	From USB HUB
Audio Connectors	3.5mm	3.5mm
	L+R out	L+R Stereo In
SD/MMC Connector	MicroSD	
User Interface	1-User defined button	Reset Button
Video	DVI-D	S-Video
Camera	Connector	Supports Leopard Imaging Module
Power Connector	USB Power	DC Power
Overvoltage Protection	Shutdown @ Over voltage	
Main Expansion Connector	Power (5V & 1.8V)	UART
	McBSP	McSPI
	I2C	GPIO
	MMC2	PWM
2 LCD Connectors	Access to all of the LCD control signals plus I2C	3.3V, 5V, 1.8V
Auxiliary Audio	4 pin connector	McBSP2
Auxiliary Expansion	MMC3	GPIO,ADC,HDQ

Sumber : BeagleBoard-XM Rev C System Reference Manual

BeagleBoard dilengkapi dengan spesifikasi sebagai berikut:

- 1) Konsumsi power yang rendah (maximum of just over 2 Watts).
- 2) Prosesor DM3730CBP system-on-a-chip, yang didalamnya termasuk ARM Cortex-A8 CPU berkecepatan sampai 1GHz

- 3) Memori dapat di-attach secara non-volatile melalui uSD, USB hub, dan USB OTG.
- 4) TMS320C64x+DSP untuk accelerated audio and video encoding and decoding.
- 5) PowerVRSGX530GPU untuk 2D dan 3D graphics acceleration, dengan support dari OpenGL ES 2.0.
- 6) 512 MB MDDR RAM.

Koneksi peripheral pada board terdiri dari:

- 1) An SD(HC)/MMC slot for operating system booting and data storage.
- 2) A USB A host socket, connected to an EHCI chip for USB 2.0 support (but no USB 1.0/1.1 support).
- 3) socket USB OTGmini AB.
- 4) HDMI connector, yang menghasilkan digital DVI video output.
- 5) S-video connector untuk analogue video output PAL atau NTSC.
- 6) 3.5 mm audio jacks untuk stereo audio input dan output.
- 7) DB9 untuk koneksi serial RS-232.
- 8) mm barrel connector power 5V dc.

2.2 Sistem Operasi Linux

Kata “linux” dewasa ini sudah tidak terlalu asing bagi sebagian orang karena kapabilitasnya sebagai satu-satunya sistem operasi yang bersifat open source dan dapat berjalan di berbagai platform bahkan termasuk prosesor intel. Sifat open source ini memungkinkan setiap user dapat menjalankan setiap bagian sistem operasi baik sebagai user biasa maupun sebagai super user.

Sejarah dibuatnya linux pertama kali adalah pada saat salah satu mahasiswa Finlandia yang bernama Linus Torvald. Pria ini telah mengenal bahasa pemrograman sejak berusia 10 tahun. Dan dikarenakan hobinya di dunia computer, pada tahun 1990 Linus Torvald berkuliah di Universitas Helsinki. Hingga pada tahun 1991 Linus mengumumkan sistem operasi linux 0.02. Dia kemudian mempublikasikan sistem operasi buatannya itu lengkap dengan source code programnya dan ternyata sistem operasi linux disambut dengan antusias oleh para programmer dan dikembangkan secara terbuka dan bersama.

Saat itu, linux agak sulit diterima masyarakat umum dikarenakan tampilannya yang mirip Unix, yaitu mode teks, tapi seiring dengan perkembangan pemrograman linux yang bersifat open source dan kini tampilan grafis linux sudah tergolong baik. Hal ini terlihat dari antusias pengguna linux secara keseluruhan. Beberapa distro linux yang cukup digemari masyarakat, seperti Mandrive, Red Hat, Open Suse, Fedora, Winlinux, Ubuntu, CentOS, BlankOn, dan masih banyak lagi.

Berikut adalah beberapa kelebihan sistem operasi linux yang mungkin dapat dipertimbangkan oleh penggunanya, yaitu :

- 1) Bila dilihat dari faktor harga, linux bersifat *free of charge*.
- 2) Bersifat *multiuser* yaitu sistem operasi linux dapat digunakan oleh beberapa user dalam sistem pada saat yang bersamaan.
- 3) Requirement hardwarenya tidak terlalu tinggi baik dari segi kapasitas maupun biayanya
- 4) shared library yang tidak merusak sistem lainnya jika diinstall dengan versi lainnya
- 5) Resistan terhadap virus, tidak seperti windows yang rentan terhadap virus
- 6) pemanfaatan memory secara optimal dengan membuat virtual memory
- 7) dilengkapi dengan firewall
- 8) shell programmable
- 9) Kestabilan sistem yang sangat baik bila dibandingkan dengan sistem operasi yang populer seperti Windows '95, '98, NT.
- 10) Banyak tersedia aplikasi gratis dan legal yang dapat digunakan
- 11) Source code yang tersedia sehingga dapat dibuka dan dipelajari serta mendebugnya
- 12) Dukungan perusahaan besar seperti : Oracle, Netscape, IBM, Corel, Sun, Informix, dan Adaptec.

Dibalik banyaknya keunggulan yang dimiliki oleh linux, terdapat juga beberapa kelemahan dari linux, yaitu antara lain :

- 1) Sistem operasi yang berbeda dengan windows menyebabkan user membutuhkan waktu untuk mempelajarinya
- 2) Penggunaan “what you see is what you get ” belum bisa secara menyeluruh, sehingga dibutuhkan trik tertentu untuk menggunakannya
- 3) sulit bagi pengguna awam untuk menggunakannya
- 4) dukungan perangkat keras dari vendor tertentu yang tidak terlalu baik di linux
- 5) proses instalasi software/aplikasi yang tidak semudah pada windows.

Pada penelitian ini, distro linux yang digunakan adalah Angstrom kernel 3.0.4+ pada BeagleBoard dan ubuntu oneiric ocelot pada PC untuk melakukan remoting dan debugging pada sistem program di BeagleBoard.

2.3 Mikrokontroler

Mikrokontroler adalah salah satu dari bagian dasar dari suatu sistem komputer. Meskipun mempunyai bentuk yang jauh lebih kecil dari suatu komputer pribadi dan komputer mainframe, mikrokontroler dibangun dari elemen-elemen dasar yang sama. Secara sederhana, komputer akan menghasilkan output spesifik berdasarkan inputan yang diterima dan program yang dikerjakan. Seperti umumnya komputer, mikrokontroler adalah alat yang mengerjakan instruksi-instruksi yang diberikan kepadanya. Artinya, bagian terpenting dan utama dari suatu sistem terkomputerisasi adalah program itu sendiri yang dibuat oleh seorang programmer. Program ini menginstruksikan komputer untuk melakukan jalinan yang panjang dari aksi-aksi sederhana untuk melakukan tugas yang lebih kompleks yang diinginkan oleh programmer.

Dalam hal ini, jenis mikrokontroler yang digunakan adalah atmega 2560 keluar ATMEL Corporation. Mikrokontroler ATmega 2560 merupakan mikrokontroler keluarga AVR yang mempunyai kapasitas flash memori 256KB. AVR (*Alf and Vegard's Risc Processor*) merupakan seri mikrokontroler CMOS 8-bit buatan Atmel, berbasis arsitektur RISC (*Reduced Instruction Set*

Computer). Secara umum, AVR dapat terbagi menjadi 4 kelas, yaitu keluarga ATtiny, keluarga AT90Sxx, keluarga AT-Mega, dan AT86RFxx. Pada dasarnya yang membedakan masing-masing kelas adalah memori, *peripheral*, dan fungsinya. Dari segi arsitektur dan instruksi yang digunakan, bisa dikatakan hampir sama. Semua jenis AVR dilengkapi dengan *flash* memori sebagai memori program. Kapasitas dari *flash* memori ini berbeda antara *chip* yang satu dengan *chip* yang lain. Tergantung dari jenis IC yang digunakan. Untuk *flash* memori yang paling kecil adalah 1 kbytes (ATtiny11, ATtiny12, dan ATtiny15) dan paling besar adalah 256 kbytes (AT-Mega2560).

Dilihat dari dalam atau sisi penggunaan, AVR ATMega2560 merupakan mikrokontroler 8 bit. Artinya data-data yang dioleh oleh processor dalam IC tersebut merupakan data berukuran 8 bit (1 byte). Dalam rangka memudahkan pengguna desainer dari ATMega2560 telah menyediakan utilitas-utilitas atau modul-modul seperti yang sudah dijelaskan pada subbab sebelumnya. Sebelum kita mendalami lebih jauh tentang utilitas – utilitas ATMega2560 kita lihat terlebih dahulu arsitektur internal dari ATMega2560. Yang dimaksud arsitektur internal dalam hal ini adalah bagian-bagian IC ATMega2560 yang menyusun sistem IC tersebut dalam menjalankan tugas, misalnya CPU, memory, dan sebagainya.

2.4 Sistem Debugging dan Remoting

Sistem Debugging dan Remoting antar device pada pembahasan kali ini akan dibedakan menjadi dua bagian, yaitu antara PC dengan BeagleBoard dan antara BeagleBoard dengan mikrokontroler. Koneksi antara PC dengan BeagleBoard akan dibangun dalam bentuk komunikasi LAN menggunakan kabel UTP dan konektor RJ-45 mode *crossover*. Koneksi ini diperlukan pada saat sistem sedang dilakukan proses pre-running seperti proses koding dan downloading program. Hal ini diperlukan agar kesalahan fatal akibat kesalahan program dapat dihindari sehingga sistem akan lebih aman jika pada saat percobaan nantinya. Sedangkan koneksi antara BeagleBoard dengan mikrokontroler dibangun dalam bentuk komunikasi serial. Koneksi ini akan dijalankan secara real time oleh sistemnya. Berikut adalah penjelasan mengenai bagaimana proses debugging dan remoting pada masing-masing koneksi.

2.4.1 Koneksi PC dengan BeagleBoard melalui LAN

Untuk membangun suatu koneksi melalui jaringan LAN maka diperlukan suatu protokol jaringan yang nantinya akan menhandel bagaimana setiap jalur data yang masuk dan keluar. Protokol jaringan yang akan digunakan pada kali ini adalah protokol jaringan SSH. SSH adalah protokol jaringan yang memungkinkan proses pertukaran data antara dua perangkat jaringan, bisa dilewatkan dalam satu jalur yang aman. SSH umumnya digunakan di linux atau OS unix lainnya untuk bisa mengakses akun shell di komputer remote. SSH adalah pengganti telnet dan aplikasi remote shell lainnya. Kelemahan telnet adalah dalam proses pertukaran datanya tidak menggunakan jalur yang aman. Misalnya password dikirim dalam format text biasa, sehingga memungkinkan pihak ketiga untuk mencuri data ditengah-tengah komunikasi itu (melakukan proses intercept). Sedangkan proses pengiriman data melalui ssh, dijamin lebih aman karena data yang dilewatkan ke jaringan sudah dienkripsi, sehingga bisa memberikan jaminan keamanan ataupun jaminan integritas data.

OpenSSH adalah satu set aplikasi komputer yang bisa mendukung sesi komunikasi terenkripsi pada jaringan komputer menggunakan protokol SSH. Awalnya aplikasi ini dikembangkan sebagai aplikasi open source yang menjadi alternatif dari aplikasi serupa yang berbayar. OpenSSH dikembangkan dan merupakan bagian dari proyek OpenBSD. Dalam perkembangan selanjutnya, OpenSSH tidak hanya memberikan aplikasi remote shell melalui ssh atau remote eksekusi program saja, tapi bisa digunakan untuk tunnelling atau setup vpn antara dua jaringan.

Sebelum menggunakan jaringan LAN untuk menghubungkan PC dan BeagleBoard maka perlu dilakukan penginstallan OpenSSH terlebih dahulu masing-masing untuk PC dan BeagleBoard, yaitu dengan cara :

```
~# sudo apt-get install openssh-server
```

Setelah terinstal dengan sukses, kita bisa menjalankan servernya menggunakan perintah berikut (harusnya sudah dijalankan secara otomatis saat selesai instalasi).

```
~# sudo /etc/init.d/ssh start
```

Untuk membangun koneksi antara PC dan BeagleBoard diperlukan langkah-langka berikut :

salah satu cara untuk menghubungkan antara BeagleBoard dan PC melalui LAN adalah dengan memberikan IP statik. Caranya adalah sebagai berikut :

- 1) Sambungkan kabel LAN *cross* ke kedua port
- 2) Pada ubuntu di PC masuk ke terminal minicom dan set ip statik di BeagleBoard. Contohnya seperti `ifconfig usb0 192.168.1.2/24`. Jika pada windows di PC dapat menggunakan ssh/serial client software Putty.
- 3) masuk ke terminal pc ubuntu linux dan set ip statik di PC. Contohnya seperti berikut `ifconfig eth0 192.168.1.1/24`.
- 4) kemudian di terminal ubuntu PC, ketik `ssh root@192.168.1.2`
- 5) setelah sukses, maka BeagleBoard langsung diremote dari PC tanpa perlu menggunakan minicom lagi.

Salah satu kelebihan dari koneksi LAN ini adalah kita dapat berinteraksi langsung dari BeagleBoard dan PC tanpa harus terdapat tambahan device misalnya dalam mengcopy dan menjalankan suatu program.

Contoh instruksi mengkopi file dari PC ke BeagleBoard melalui LAN adalah :

```
~# scp /home/irvanjpe/.../jp-nyoba.c root@192.168.1.2:/home/root
```

Contoh menampilkan gambar web kamera di PC yang di-*attach* di BeagleBoard adalah :

```
~# gstreamer-properties
```

Jika ingin menggunakan koneksi WLAN, cara mengkonfigurasinya memiliki kemiripan dengan konfigurasi LAN. Perbedaanya terletak pada interface yang digunakan yaitu wireless adaptor. Karena di BeagleBoard tidak terdapat wireless adaptor, maka dibutuhkan device tambahan tersebut. Dan ketika kita mengetikkan `ifconfig`, akan muncul interface `wlan1` jika kita telah menginstal drivernya. Setelah itu, langkah selanjutnya adalah sama seperti konfigurasi LAN.

Khusus untuk BeagleBoard yang harus melakukan koneksi ke internet dengan cara tertentu diperlukan suatu cara untuk melakukan koneksi ke internet, lain halnya dengan komputer PC yang masing-masing dari kita telah familiar

bagaimana cara melakukan koneksi ke internet. Berikut adalah langkah-langkahnya :

1) Koneksi internet via langsung dari LAN

khusus dalam hal ini, kita akan membuat BB bisa mengakses internet melalui proxy. Akses ini bertujuan untuk mendownload file yang akan diperlukan.

Caranya adalah sebagai berikut :

- a) koneksikan PC ke internet melalui jaringan LAN
- b) setelah login ke proxy dan memastikan akses berhasil dengan membuka suatu website, catat ip dan netmasknya.
- c) kemudian cabut kabel LAN dari PC dan colokkan ke BeagleBoard.
- d) set proxy BeagleBoard dengan instruksi `export http_proxy=http://152.118.24.10:8080`
- e) setelah itu set ip dan netmask secara manual di BeagleBoard sesuai dengan yang berada di PC tadi.
- f) akses pun bisa dilakukan sesuai dengan keinginan

2) Koneksi sharing via LAN

langkah-langkahnya adalah sebagai berikut :

- a) Colok kabel ethernet di pc
- b) `ifconfig eth0`
catat ALAMAT (contoh: 152.118.27.126) dan NETMASK (contoh:255.255.255.128)
- c) `Route -n`, catat GATEWAY (contoh: 152.118.27.1)
- d) Login ke proxy (buka situs luar lalu login)
- e) Cabut kabel lalu colokkan ke colokan RJ45 beagleboard
- f) Di beagleboard: `ifconfig usb0 ALAMAT`, `ifconfig usb0 netmask NETMASK`, `route add default gw GATEWAY`
- g) Salin isi `/etc/resolv.conf` dari pc ke beagleboard(`nameserver 152.118.24.2`)

- h) Setelah selesai maka bisa mencoba akses dengan cara
 http_proxy=http://152.118.24.10:8080 opkg update

2.4.2 Koneksi BeagleBoard dengan mikrokontroler melalui serial

Secara umum terdapat dua jenis komunikasi yang mungkin dilakukan pada dua device, yaitu serial dan parallel. Untuk komunikasi parallel sendiri tidak banyak yang dibahas karena pada komunikasi jenis ini memerlukan resource yang cukup rumit untuk sekelas device yang berukuran kecil. Sedangkan pada komunikasi jenis serial, terdapat banyak jenis yang mendukung seperti UART (Universal Asynchronous Receiver Transmitter), USRT (Universal Synchronous Receiver Transmitter), SPI (Serial Peripheral Interface), dan lain sebagainya. Terdapat juga pengembangan komunikasi lainnya seperti USB, TWI (Two Wire Interface), 1 Wire Interface, dan lain-lain. Seluruh jenis komunikasi tersebut dapat dilakukan, namun biasanya produk device elektronik tertentu lebih memilih jenis komunikasi yang telah familiar dan mudah digunakan oleh orang banyak.

Interface serial lebih kompleks/sulit dibandingkan dengan interface melalui komunikasi parallel hal ini disebabkan karena:

- 1) Dari segi perangkat keras: adanya proses konversi data paralel menjadi serial atau sebaliknya menggunakan piranti tambahan yang disebut UART (*Universal Asynchronous Receiver/Transmitter*)
- 2) Dari Segi perangkat lunak: lebih banyak register yang digunakan atau terlibat

Namun di sisi lain interface serial menawarkan berapa kelebihan dibandingkan secara paralel, antara lain:

- 1) Kabel untuk komunikasi serial bisa lebih panjang dibandingkan dengan paralel; data-data dalam komunikasi serial dikirim-kan untuk logika '1' sebagaitegangan -3 s/d -25 volt dan untuk logika '0' sebagai tegangan +3 s/d +25 volt, dengan demikian tegangan dalam komunikasi serial memiliki ayunan tegangan maksimum 50 volt, sedangkan pada komunikasi paralel hanya 5 volt. Hal ini menyebabkan gangguan pada kabel-kabel panjang lebih mudah diatasi dibandingkan pada paralel;
- 2) Jumlah kabel serial lebih sedikit; Anda bisa menghubungkan dua perangkat komputer yang berjauhan dengan hanya 3 kabel untuk

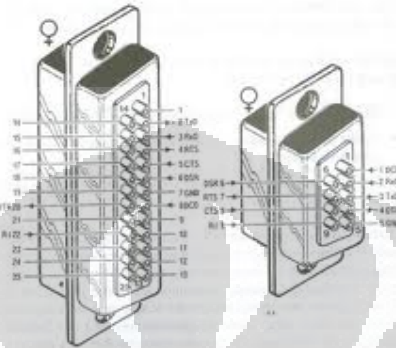
konfigurasi *null modem*, yaitu TXD (saluran kirim), RXD (saluran terima) dan Ground, bayangkan jika digunakan teknik paralel akan terdapat 20 – 25 kabel! Namun pada masing-masing komputer dengan komunikasi serial harus dibayar "biaya" antarmuka serial yang agak lebih mahal;

- 3) Banyaknya piranti saat ini (*palmtop, organizer, hand-phone* dan lain-lain) menggunakan teknologi infra merah untuk komunikasi data; dalam hal ini pengiriman datanya dilakukan secara serial. IrDA-1 (spesifikasi infra merah pertama) mampu mengirimkan data dengan laju 115,2 kbps dan dibantu dengan piranti UART, hanya panjang pulsa berkurang menjadi 3/16 dari standar RS-232 untuk menghemat daya;
- 4) Untuk teknologi *embedded system*, banyak mikrokontroler yang dilengkapi dengan komunikasi serial (baik seri RISC maupun CISC) atau *Serial Communication Interface (SCI)*; dengan adanya SCI yang terpadu pada IC mikrokontroler akan mengurangi jumlah pin keluaran, sehingga hanya dibutuhkan 2 pin utama TxD dan RxD (di luar acuan *ground*).

Pada prinsipnya sendiri, komunikasi serial adalah komunikasi yang terjadi pada dua terminal dimana pengiriman data dilakukan per bit sehingga jika dibandingkan dengan komunikasi paralel seperti pada port printer yang mampu mengirim 8 bit sekaligus dalam satu clock.

Dalam hal ini, komunikasi serial dilakukan antara BeagleBoard yang berfungsi sebagai PC dan mikrokontroler atmega 32. Dalam PC itu sendiri komunikasi serial memiliki protocol tersendiri yaitu sistem transmisi sinyal RS232. Sistem transmisi sinyal RS232 menggunakan level tegangan terhadap sistem ground elektronik. Tipe ini bagus untuk komunikasi data secara point to point (akan dijelaskan kemudian). RS232 port pada PC hanya diperuntukkan untuk satu alat (*single device*). Misal : COM1 digunakan untuk mouse dan COM2 untuk modem untuk di windows sedangkan pada linux digunakan port ttyO1, ttyO2, dan seterusnya. Syarat signal RS232 dapat berfungsi adalah dengan cara menghubungkan ke ground antara PC dengan alat. Jarak maksimal jalur komunikasi sangat terbatas hanya 100 / 200 feet untuk komunikasi data secara asinkron dan hanya 50 feet untuk komunikasi sinkron. Singkatnya, RS232 hanya

untuk komunikasi area local dan hanya untuk satu driver dan satu receiver. RS232 pada PC mempunyai dua jenis konektor yaitu konektor dengan 25 pin (DB25) dan konektor dengan 9 pin (DB9). Pada dasarnya hanya 3 pin yang terpakai yaitu pin TX, RX, dan GND.



Gambar 2-2 DB-25 dan DB-9

Sumber : <http://www.scienceprog.com/wp-content/uploads/RS232/>

Pada komputer saat ini port serial mulai ditinggalkan. Oleh karenanya seringkali kita membutuhkan converter agar masih bisa menggunakan komunikasi serial dan biasanya port USB yang akan digunakan sebagai media komunikasinya. Converter itu seringkali disebut kabel usb-serial.



Gambar 2-3 Konverter Serial USB

Sumber : <http://www.indorobotika.com/image>

Koneksi antara BeagleBoard dan mikrokontroler secara software dilakukan dengan cara sebagai berikut :

- 1) Dari sisi BeagleBoard, kita harus melakukan programming pada serial port dengan cara sebagai berikut (misalnya data dikirim dan diterima adalah array 3 element):

```
void transmit(char MSBx, char LSBx, char MSBy, char LSBBy, char sdata){
    dataT[0] = 0xF5;  dataT[1] = MSBx;  dataT[2] = LSBx;
    dataT[3] = MSBy;  dataT[4] = LSBBy;  dataT[5] = sdata;
    if (write(rs232, dataT, sizeof(dataT)) != sizeof(dataT)) perror(NULL);
        printf("sent\n");
    }
void receive(){
    if(read(rs232, dataR, sizeof(dataR)) != sizeof(dataR){
        perror(NULL);
        read(rs232, dataR, sizeof(dataR));}
    if(dataR[0]!=0xF5) printf("false data\n");

    else{
        printf("read success\n");
        printf("data1=%c data2=%c data3=%c data4=%c data5=%c
        data6=%c\n", dataR[0], dataR[1], dataR[2],
        dataR[3],dataR[4],dataR[5]);}
    }
```

- 2) Dari sisi mikrokontroler, kita hanya melakukan proses putchar dan getchar seperti biasa. Namun dengan sedikit modifikasi pada bagian *override* data karena data yang dikirim dari BeagleBoard adalah data array. caranya digambarkan melalui pseudocode berikut (misalnya data dikirim dan diterima adalah array 3 element) :

```
void transmit4(){
    char i=0,ii=0;
    lcd_gotoxy(0,1);
    lcd_putsf("terkirim");
    putchar(data3[5]); putchar(data3[0]); putchar(data3[1]);
    putchar(data3[2]); putchar(data3[3]); putchar(data3[4]);
    }
void receive(){
    char baca,i=1,ii=0;
    if (getchar()==0xF5) {
        while(1){
            data3[0]=0xF5;  data3[1]=getchar();  data3[2]=getchar();
            data3[3]=getchar();  data3[4]=getchar();  data3[5]=getchar();
            break;
        }
    }
```



```

else lcd_putsf("no entry data");
}

```

Akhirnya kita bisa melakukan proses remoting dan debugging secara robust karena kita setiap kali melakukan pengiriman, maka ada proses feedback yang memberitahukan datanya apakah sampai atau tidak, sehingga proses selanjutnya dapat dilakukan tanpa adanya crash pada proses komunikasi.

2.5 OpenCV

OpenCV adalah sebuah library yang terdiri dari fungsi-fungsi yang mendukung penggunaan image processing secara real time dan menyediakan cross-platform API dari tingkat menengah sampai tinggi. OpenCV memiliki lisensi BSD sehingga dapat digunakan secara bebas baik untuk komersial maupun penelitian. OpenCV sendiri ditulis dalam bahasa C yang dioptimasi dan ditujukan untuk kepentingan aplikasi real time, akan tetapi sekarang OpenCV telah dikembangkan untuk bahasa pemrograman C++ berikut segala perkembangannya. Bahkan sekarang juga terdapat interface Python pada librarinya. OpenCV juga memiliki kemampuan independensi terhadap sistem operasi, hardware, ataupun window-manager tertentu.

Computer vision sendiri adalah proses transformasi data dari sebuah source baik sebagai citra, video dari direktori, maupun video dari kamera yang akan digunakan untuk keperluan *image processing*. Banyak contoh aplikasi yang menggunakan OpenCV sebagai platformnya seperti Human-Computer Interaction (HCI), Object Identification, Segmentation dan Recognition, *Face detection*, Gesture Recognition, Motion Tracking, Ego Motion, Motion Understanding, Mobile Robotics, dan lain-lain.

Terdapat banyak sekali fitur pada OpenCV dalam mendukung kemampuannya sebagai library yang powerful pada computer vision, yaitu antara lain :

- 1) Manipulasi data gambar seperti allocation, release, copying, dan lain-lain
- 2) Input dan output berupa image dan video
- 3) Manipulasi matriks dan vector serta rutin / prosedur pada aplikasi aljabar linear

- 4) Struktur data dinamik seperti lists, queues, sets, dan lain-lain
- 5) Dasar image processing seperti filtering, edge detection, histograms, dan lain-lain
- 6) Analisis struktur seperti countour processing, hough transform, line fitting, dan lain-lain
- 7) Kalibrasi kamera seperti kalibrasi tracking, stereo correspondence, dan lain-lain
- 8) Analisis motion seperti optical flow, motion segmentation, dan lain-lain
- 9) Object recognition seperti eigen-methodes dan HMM
- 10) dasar GUI seperti display image, keyboard, scroll bars, dan lain-lain
- 11) Image labeling seperti line, conic, polygon, dan lain-lain

Penggunaan software ini memiliki beberapa keunggulan dibandingkan dengan yang lain. Antara lain :

- 1) OpenCV dibuat hanya untuk keperluan image processing
- 2) OpenCV dapat 'di-built' dengan Intel® Integrated Performance Primitives (IPP). Hal ini membuat architecturenya lebih ringkas dan cepat.
- 3) OpenCV menggunakan bahasa pemrograman C/C++, sehingga jalannya program yang dibuat lebih cepat dan ringan.
- 4) OpenCV memiliki penggunaan memori yang lebih efisien.
- 5) OpenCV sudah memiliki banyak dokumentasi proyeknya.
- 6) OpenCV lebih mudah diintegrasikan dengan embedded system lain.
- 7) Dan masih banyak lagi jika lebih detail

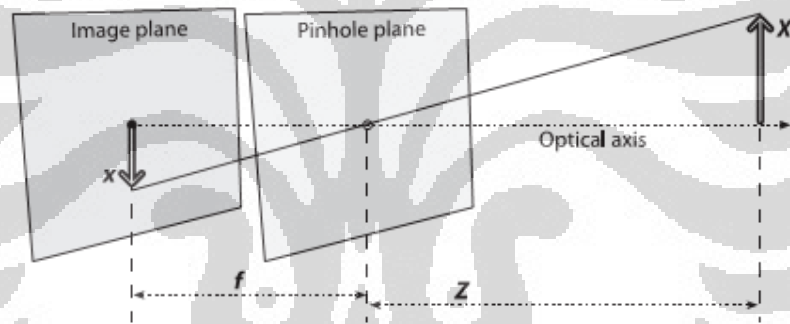
2.6 Model Kamera dan Proyeksi

Sistem vision dimulai dengan adanya cahaya yang dideteksi dari dunia luar cahaya dimulai dengan sebuah sinar yang memancar dari suatu sumber, misalnya lampu atau matahari, yang akan bergerak di suatu ruangan dan mengenai suatu benda. Ketika cahaya tersebut mengenai suatu benda, terdapat sebagian

cahaya yang diserap dan dipantulkan. Jumlah dan intensitas cahaya yang dipantulkan ini akan ditangkap oleh mata kita di retina. Begitu juga dengan cara kerja dari sistem kamera untuk ‘melihat’ objek. Model analitik di atas akan dijadikan sebagai dasar pemodelan kamera pada sistem komputer vision.

2.6.1 Model Kamera

Pada kamera sendiri digunakan model *pinhole* untuk menggambarkan proses di atas. *Pinhole* didefinisikan sebagai dinding imajiner dengan lubang yang sangat kecil di tengahnya yang akan mengeblok semua sinar yang masuk kecuali sinar dari tepat ditengahnya. Akan tetapi, *pinhole* saja tidak cukup untuk membuat suatu image frame karena kurangnya cahaya yang masuk pada kamera. Oleh karena itu, digunakan lensa untuk menyelesaikan masalah di atas. Namun, penggunaan lensa sendiri akan menghasilkan efek distorsi yang akan dijelaskan pada subbab selanjutnya.

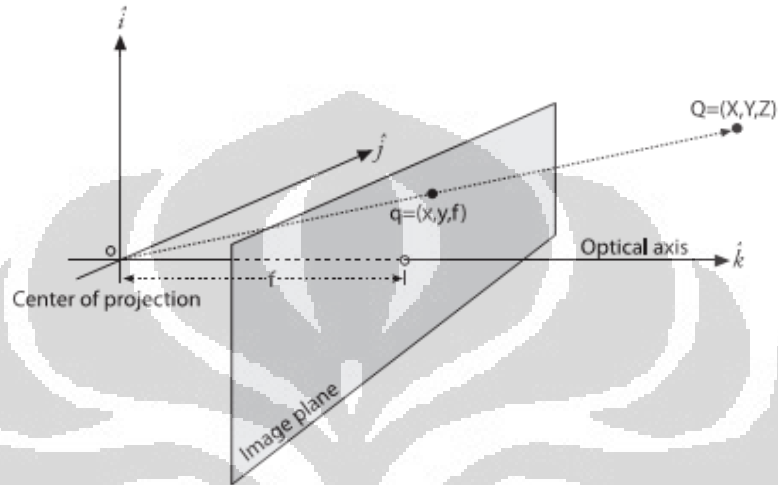


Gambar 2-4 Model Kamera dengan *Pinhole* dan Bidang Image Berada di Belakang *Pinhole*

Cara kerja dari sistem kamera digambarkan dengan jelas pada gambar 2-4 di atas dimana objek yang terdeteksi pada kamera berjarak Z akan diproyeksikan oleh *pinhole* kamera ke bidang image dengan jarak focus yang statis (tergantung jenis kamera) melalui *center of projection* pada *pinhole plane*. Maka dengan menggunakan persamaan dasar matematika, hubungan antara objek dan proyeksinya adalah :

$$-x = f \frac{X}{Z} \quad \text{pers 2-1}$$

Sifat dari proyeksi benda pada model kamera di atas adalah terbalik dan jaraknya minus. Sehingga untuk memudahkan pemodelan digunakan model *front projection model* seperti digambarkan pada gambar



Gambar 2-5 *Front Projection Model* Kamera

Dengan sedikit pengetahuan tentang geometri matematika, maka persamaan modelnya menjadi

$$x = f \frac{X}{Z} \quad \text{pers 2-2}$$

Model ini menggambarkan adanya *optical axis* yang akan berpotongan dengan *center of projection*. Model yang digambarkan sesuai dengan persamaan 2-2 ini akan terus digunakan untuk penjelasan di bagian depan.

Namun, karena ketidakidealan dari proses manufaktur sistem kamera itu sendiri, kemungkinan yang terjadi adalah sumbu optikal terletak tidak pada *center of projection*. Sehingga letak koordinat (x,y) pada bidang image sudah tidak sesuai dengan persamaan 2-2. Maka perlu ditambahkan factor koreksi c_x dan c_y pada persamaan 2-2 untuk menjadi persamaan seperti berikut :

$$x_{screen} = f_x \left(\frac{X}{Z} \right) + c_x, y_{screen} = f_y \left(\frac{Y}{Z} \right) + c_y \quad \text{pers 2-3}$$

Akibat dari pertimbangan di atas, model pixel yang digunakan tidak lagi pada bentuk idealnya yaitu persegi namun sudah pada bentuk persegi panjang.

2.6.2 Geometri Proyeksi

Setelah model kamera diturunkan, persamaan geometri proyeksi secara umum perlu diturunkan juga untuk menghilangkan ambiguitas. Proyeksi kamera secara sederhana dinyatakan dengan bagaimana menghubungkan antara koordinat titik objek $Q(X,Y,Z)$ dengan koordinat titik image $q(x,y)$. Hubungan antar kedua koordinat titik tersebut diistilahkan dengan *projective transform*. Secara ringkas, penurunan formula umum untuk proyeksi tidak dijabarkan secara detil pada kali ini, maka bentuk sederhana dari proyeksinya adalah sebagai berikut :

$$q = MQ \rightarrow q = \begin{bmatrix} x \\ y \\ w \end{bmatrix}, M = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}, Q = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad \text{pers 2-4}$$

Dari persamaan 2-4 tampak bahwa walaupun masukan matriks koordinat objek berdimensi tiga, keluaran dari koordinat imagenya tetap berdimensi dua karena persamaan 2-4 akan menghasilkan $w=Z$.

Akhirnya dari penurunan model kamera dan sistem geometri proyeksi, semua *user* komputer vision dapat melakukan manipulasi objek untuk keperluan sistem vision. Hal ini sangat penting karena pengetahuan dasar tentang *basic* kamera sangat penting untuk keperluan *image processing* selanjutnya

2.7 Face Detection

Berikut akan dijelaskan bagaimana proses *face detection* secara detil dimulai dari awal sejarah dan salah satu metode paling populer yang diimplementasikan pada library OpenCV. Antara lain :

2.7.1 Pendahuluan

Face detection secara umum sebenarnya adalah suatu metode dan konsep yang relative baru. Konsep ini dikembangkan pada tahun 1960-an, dimana *face detection* semiotomatis pertama kali dilakukan pada sebuah foto untuk menentukan letak mata, hidung, mulut, dan lain sebagainya yang hasilnya nanti

akan dibandingkan dengan data referensi. Masalah yang terjadi pada saat itu adalah pengukuran dan pengalokasian tempatnya masih dilakukan secara manual. Pada tahun 1991, Turk dan Pentland menemukan bahwa dengan teknik eigenfaces, besarnya error residual dapat digunakan untuk mendeteksi wajah pada suatu image. Penemuan ini menunjukkan bahwa era *automated face detection* baru dimulai dan terus berkembang ke banyak aplikasi.

Sebagai salah satu aplikasi paling sukses pada bidang image analisis, *face detection* telah mendapat tempat dan perhatian khusus untuk pengembangannya. Sepuluh tahun terakhir, *face detection* menjadi sangat populer dan diimplementasikan di bidang-bidang sebagai berikut :

- 1) Perusahaan keamanan ; computer dan physical access control
- 2) Pemerintahan bagian criminal ; Surveillance
- 3) Keimigrasian ; passport dan ID Card
- 4) Casino ; filter pemain judi yang mencurigakan
- 5) Mainan anak-anak ; Intelligent robotic
- 6) Kendaraan ; sistem keamanan darurat

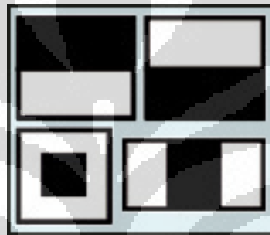
2.7.2 Metode Face Detection

Pendeteksiwajah yang digunakan pada penelitian ini menggunakan metode yang di-*propose* oleh PaulusVioladan Michael Jones yang diterbitkan pada tahun 2001. Metode tersebut biasanya disebut sebagai metode Viola-Jones atau hanya Viola-Jones, pendekatan yang dilakukan untuk mendeteksi objek-objek dalam suatu image adalah dengan menggabungkan empat konsep kunci berikut, yaitu :

- 1) Simple fitur *rectangular*, yang sering disebut sebagai *Haar features*.
- 2) Sebuah Gambar terintegrasi untuk deteksi fitur secara cepat
- 3) Metode *AdaBoost machine-learning*
- 4) Sebuah cascade classifier yang menggabungkan banyak fitur secara efisien

Fitur yang digunakan oleh Viola dan Jones didasarkan pada *Haar wavelet*. *Haar wavelet* adalah gelombang persegi panjang pada gelombang tunggal (satu interval tinggi dan satu interval rendah). Dalam dua dimensi, gelombang persegi adalah pasangan dari persegi panjang yang berdekatan.

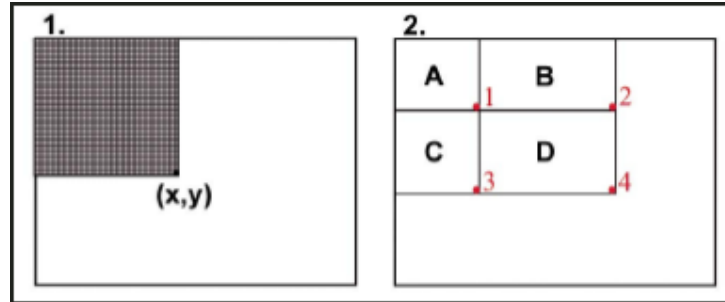
Kombinasi *rectangle* yang digunakan pada object detection sebenarnya adalah bukanlah *Haar wavelets* yang sebenarnya. Melainkan, *Haar wavelets* mengandung kombinasi *rectangle* yang sesuai dengan tugas dari rekognisi visualnya. Karena perbedaan tersebut, maka fitur ini disebut *Haar features* dan bukan *Haar wavelet*. Gambar di bawah menunjukkan fitur yang digunakan oleh OpenCV.



Gambar 2-6 Haar Features Pada OpenCV

Keberadaan *Haar features* ini ditentukan dengan cara mengurangkan rata-rata dari pixel *dark-region* dengan rata-rata dari pixel *light-region*. Jika selishnya berada di atas *threshold*, maka *Haar features* dikatakan sebagai 'ada'.

Selanjutnya untuk menentukan ada atau tidak adanya ratusan *Haar features* pada setiap lokasi di suatu image, maka Viola dan Jones menggunakan teknik yang disebut sebagai *integral image*. Secara umum, "integrasi" maksudnya adalah menambahkan semua unit terkecil. Dalam hal ini, unit terkecil tersebut adalah nilai *value*. Nilai yang sudah terintegrasi tadi merupakan jumlah dari semua pixel di atas *threshold*nya dan berada di sebelah kiri gambar. Dengan memulai prosedur di atas dari sebelah kiri atas image menuju kanan bawah image, maka seluruh gambar dapat diintegrasikan dengan hanya sedikit operasi per pixel.



Gambar 2-7 (1) Proses Integrasi (2) *Rectangle* Dibagi Menjadi Beberapa Segmen

Seperti pada gambar 2-7 menunjukkan bahwa setelah proses integrasi, nilai pada setiap lokasi pixel (x,y) merupakan jumlah dari semua nilai pixel dalam suatu wilayah berbentuk persegi panjang yang memiliki satu sudut di kiri atas gambar dan sudut lainnya di lokasi (x,y) . Untuk menemukan nilai pixel rata-rata dari persegi panjang ini, maka proses yang diperlukan hanyalah dengan membagi nilai pixel pada (x,y) dengan luas persegi panjang tersebut.

Akan tetapi bagaimana jika kita ingin mengetahui nilai-nilai hasil penjumlahan untuk setiap persegi panjang lainnya, yaitu persegi panjang yang salah satu sudutnya berada di kiri atas foto? Gambar 2-7(2) menunjukkan solusi untuk masalah di atas. Misalkan kita ingin nilai-nilai hasil penjumlahan dalam persegi panjang D. Kita bisa menganalogikannya sebagai jumlah dari nilai-nilai piksel pada persegi panjang gabungan $(A+B+C+D)$, dikurangi dengan jumlah dalam persegi panjang $A+B$ dan $A+C$ dan ditambahkan jumlahnya dengan nilai pixel dalam A. Sehingga secara ringkas hasilnya adalah sebagai berikut :

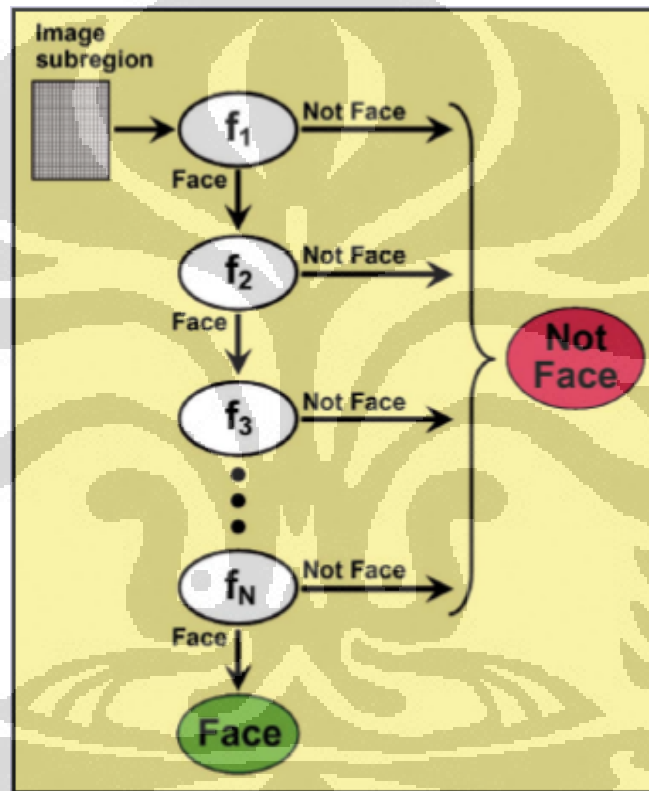
$$D = A + B + C + D - (A + B) - (A + C) + A \quad \text{pers 2-5}$$

Mudahnya, penjumlahan $A+B+C+D$ adalah nilai *integral image* pada lokasi 4, $A+B$ adalah penjumlahan nilai pada lokasi 2, $A+C$ adalah penjumlahan nilai pada lokasi 3, dan A adalah penjumlahan nilai pada lokasi 1. Jadi, dengan konsep *integral image*, kita dapat menemukan jumlah nilai pixel untuk setiap persegi panjang pada gambar asli dengan hanya tiga operasi integer :

$$(x_4, y_4) - (x_2, y_2) - (x_3, y_3) + (x_1, y_1) \quad \text{pers 2-6}$$

Untuk memilih *Haar features* khusus yang digunakan dan untuk mengatur *thresholding*, Viola dan Jones menggunakan metode *machine-learning*

yang disebut dengan metode *AdaBoost*. *AdaBoost* menggabungkan banyak *classifier* yang "lemah" untuk membuatnya menjadi satu *classifier* yang "kuat". Istilah "Lemah" di sini berarti hanya mendapat sedikit *classifier* yang tepat dari pada secara random dan hal tersebut sangat tidak baik. *AdaBoost* memilih satu set *classifier* yang "lemah" untuk menggabungkan dan memberikan bobot untuk masing-masingnya. Viola dan Jones menggabungkan serangkaian *classifier AdaBoost* sebagai suatu rantai filter, sebagaimana ditunjukkan dalam gambar 2-8, metode ini dianggap sangat efisien untuk mengklasifikasikan suatu daerah *image*.



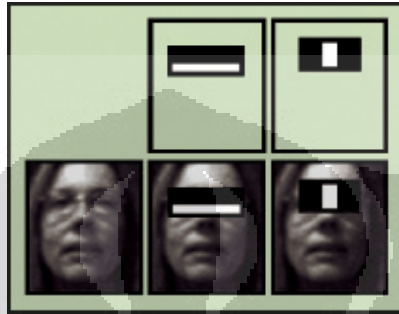
Gambar 2-8 Haar Classifier pada OpenCV

Sumber :

http://www.cognotics.com/opencv/servo_2007_series/part_2/sidebar.html#fig_1

Nilai Threshold yang masih bisa diterima pada setiap level akan diset *low* untuk 'meloloskan' setiap contoh wajah pada training set. Filter pada setiap level tadi dilatih untuk training gambar yang melewati semua tahap sebelumnya. Database pelatihan adalah databas sekumpulan wajah-wajah, mungkin seribu atau lebih. Selama digunakan, jika salah satu dari filter ini gagal untuk melewati

sebuah wilayah gambar, maka daerah tersebut segera diklasifikasikan sebagai "bukan wajah." Ketika filter melewati wilayah gambar, maka gambar tersebut menuju filter berikutnya dalam rantai filter di atas. Gambar daerah yang melewati semua filter dalam rantai tersebut diklasifikasikan sebagai "Wajah." Viola dan Jones memberikan julukan proses ini sebagai *cascade*.



Gambar 2-9 Hasil *Classifier*

Urutan filter dalam *cascade* ini didasarkan pada bobot hasil dari metode AdaBoost. Semakin besar bobot filter pada rantai pertama, maka untuk menghilangkan daerah non-wajah citra juga akan dilakukan dengan cepat. Gambar 2-9 menunjukkan dua fitur pertama dari original Viola-Jones *cascade* ditempatkan pada suatu wajah. Kunci yang pertama adalah pada area pipi yang lebih terang dibandingkan dengan pada wilayah mata dan yang kedua adalah bahwa hubungan antara kening dan hidung lebih terang dari mata.

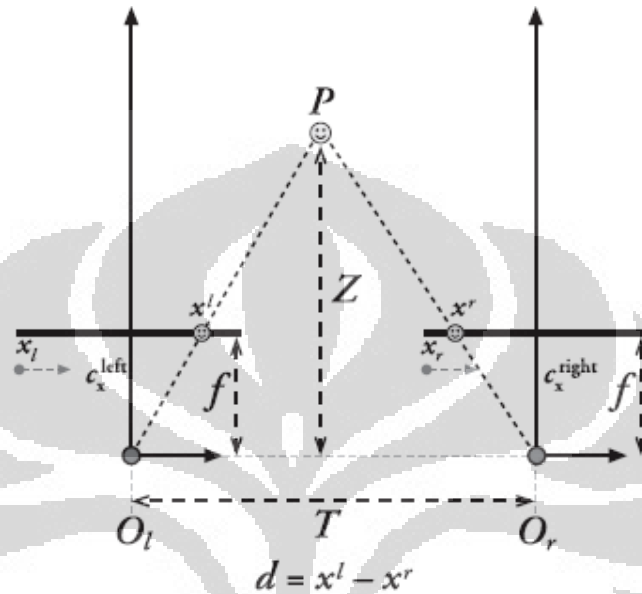
2.8 Stereo Vision

Dalam bab ini akan dijelaskan tentang dasar teori dari teknik menggabungkan dua kamera untuk mendapatkan persepsi 3 dimensi pada gambar yang di-*capture*. Penjelasan dimulai dari awal sejarah dan salah satu metode paling populer yang diimplementasikan pada library OpenCV. Antara lain :

2.8.1 Pendahuluan

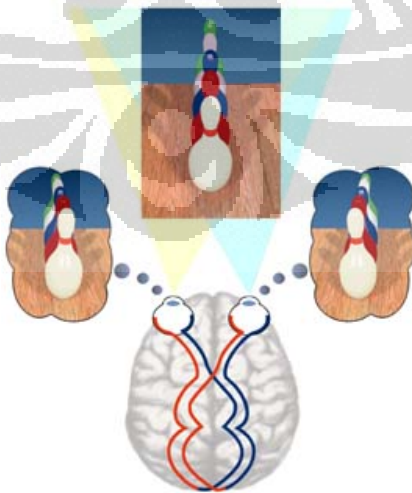
Stereo vision berasal dari kata *stereo* dan *vision* yang masing-masing memiliki arti solid/3-dimensi dan penglihatan. Secara umum *stereo vision* adalah teknik dalam ilmu computer vision untuk melakukan konstruksi 3-dimensi yang berasal dari beberapa image 2 dimensi. Secara fundamental, *stereo vision* diadaptasi dari penggunaan mata pada manusia yaitu masing-masing mata kiri dan mata kanan manusia mendapatkan citra dan kemudian dilakukan proses matching di otak untuk menemukan kedalaman dari suatu citra.

Stereo vision pada computer vision, setiap fitur atau bagian-bagian dari suatu image diambil dalam satu waktu dari beberapa kamera, yang dalam hal ini digunakan 2 kamera untuk dilakukan proses matching sesuai fitur yang berkoresponding satu sama lain.



Gambar 2-10 Geometri Ideal dari Sistem *Stereo Vision*

Hasil dari proses matching tersebut dan informasi mengenai jarak baseline (jarak masing-masing kamera) akan ditemukan perbedaan yang akhirnya didapatkan proses kedalaman dari suatu objek yang dideteksi baik secara parsial maupun keseluruhan image.



Gambar 2-11 Cara Kerja *Stereo Vision* pada Mata Manusia

Sumber : <http://www.vision3d.com/stereo.html>

Proses pengambilan dua gambar dalam satu waktu untuk dilakukan proses pengukuran kedalaman objek gambar sering diistilahkan sebagai *stereoimaging*. Proses *stereo imaging* memiliki empat langkah yaitu antara lain :

- 1) Menghilangkan distorsi radial dan tangential lensa kamera secara matematikal. Proses ini dinamakan *undistortion*. output dari proses ini adalah *undistort image*.
- 2) Mengatur sudut dan jarak di antara masing-masing kamera. proses ini dinamakan *rectification*. Output dari proses ini adalah image yang telah *rectified* dan *row-aligned*.
- 3) Menemukan fitur yang sama pada kamera kanan dan kiri. Proses ini diistilahkan sebagai *correspondence*. Output dari proses ini adalah *disparity map* dimana *disparity* adalah perbedaan koordinat x pada bidang gambar untuk setiap fitur gambar yang sama yang dilihat oleh kaera kanan dan kiri.
- 4) Pengetahuan tentang geometri kamera bisa menghasilkan *depth map* yang berasal dari *disparity map* dengan proses *triangulation*. Proses ini disebut sebagai *reprojection*.

Keempat langkah di atas merupakan prosedur umum yang dilakukan untuk melakukan teknik *stereo vision*. Jika salah satu tidak dilakukan maka akan dipastikan hasil *depth map* tidak akan sesuai dengan aslinya.

Oleh karena itu, ada beberapa hal yang penting yang harus dilakukan untuk melakukan teknik *stereo vision* ini. salah satunya adalah menghilangkan semua hal yang membuat hasil proses *stereo vision* menjadi tidak baik misalnya menghilangkan distorsi pada lensa kamera.

Secara umum, distorsi pada lensa kamera disebabkan oleh ketidaksesuaian pada proses pembuatannya. Karena seperti telah diketahui, untuk mendapatkan lensa secara akurat sangatlah sulit. Berikut adalah dua distorsi utama yang terjadi pada lensa kamera yaitu :

1) *Radial Distortion*

Jenis distorsi ini terjadi dikarenakan oleh bentuk lensa yang *spherical*. Cahaya yang masuk melalui tengah lensa tidak akan mengalami refraksi yang cukup besar, namun untuk cahaya yang masuk melalui tepi lensa akan mengalami *bending* yang sangat besar sehingga bagian pinggir dari lensa adalah bagian paling besar yang terkena *radial distortion*. Berikut adalah contoh gambar akibat dari *radial distortion* dimana gambar tampak melengkung.



Gambar 2-12 Hasil Gambar Akibat *Radial Distortion*

<http://www.aishack.in/2010/07/two-major-physical-defects-in-cameras/>

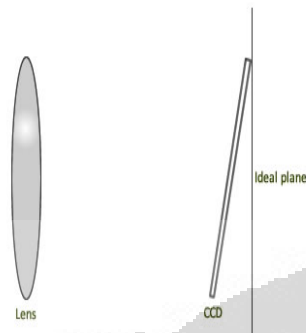
Secara umum, *radial distortion* berasal dari penurunan deret Taylor dan dapat diformulasikan secara umum sampai tiga parameter awal sebagai berikut :

$$x_{corrected} = x(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad \text{pers 2-7}$$

$$y_{corrected} = y(1 + k_1r^2 + k_2r^4 + k_3r^6) \quad \text{pers 2-8}$$

2) *Tangential Distortion*

Ketika letak lensa tidak parallel terhadap bidang gambar, maka akan terjadi distorsi tangential sesuai dengan gambar berikut ini :



Gambar 2-13 Hasil Gambar Akibat *Tangential Distortion*

<http://www.aishack.in/2010/07/two-major-physical-defects-in-cameras/>

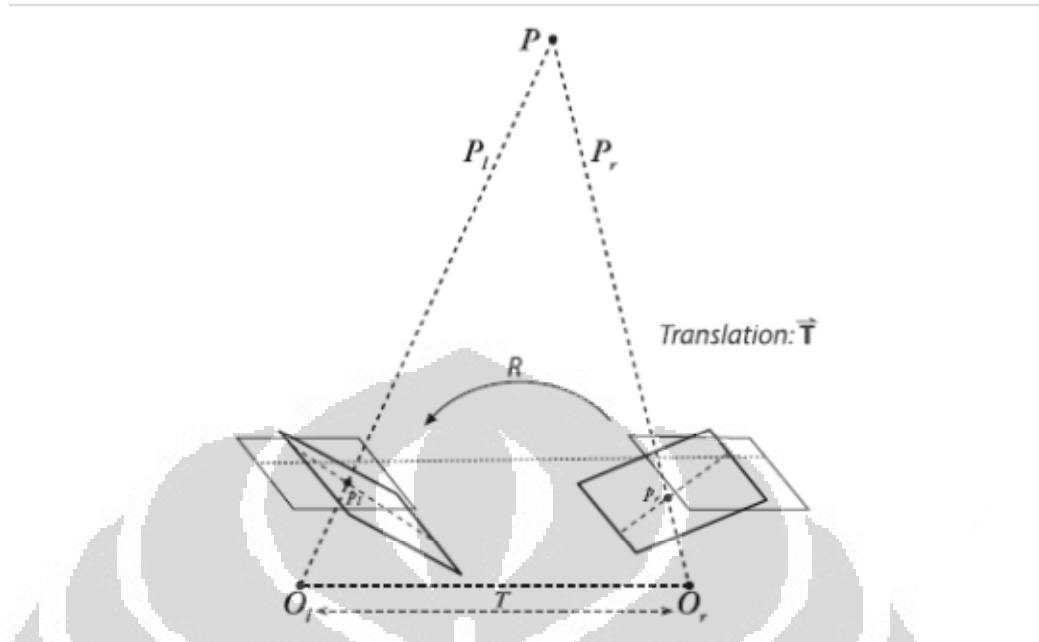
Tangential distortion secara umum diformulasikan sebagai berikut :

$$x_{corrected} = x + (2p_1y + p_2(r^2 + 2x^2)) \quad \text{pers 2-9}$$

$$y_{corrected} = y + (p_1(r^2 + 2y^2) + 2p_2x) \quad \text{pers 2-10}$$

Sebenarnya terdapat jenis distorsi lain selain dua jenis distorsi di atas yaitu *discolouring distortion*, *colour splitting distortion*, *blurring*, dan lain sebagainya. Namun kebanyakan kamera hanya sedikit terganggu oleh jenis distorsi selain *radial distortion* dan *tangential distortion* sehingga pada akhirnya jenis distorsi itu diabaikan saja.

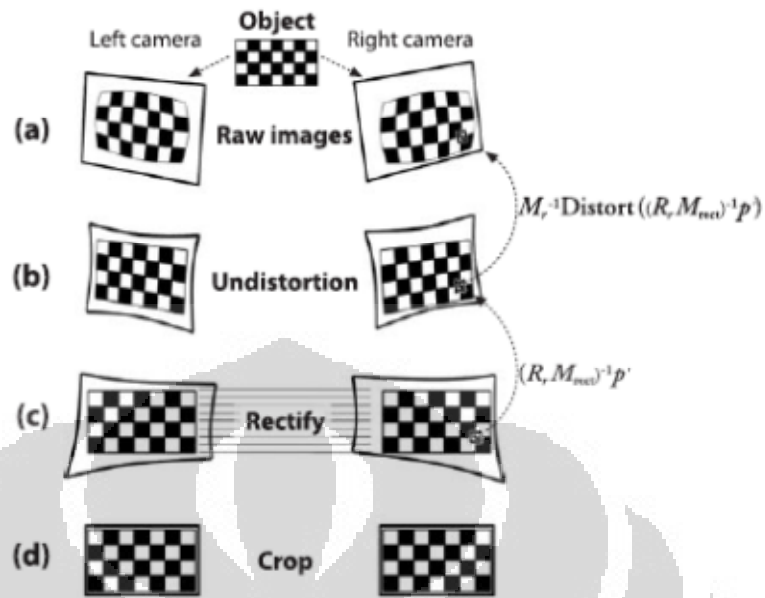
Setelah distorsi dihilangkan, hal selanjutnya yang dilakukan adalah proses rektifikasi yaitu membuat dua image yang berasal dari dua kamera bersekutu atau lurus terhadap satu sama lain sesuai dengan gambar di bawah ini:



Gambar 2-14 Tujuan Akhir dari Proses Rektifikasi

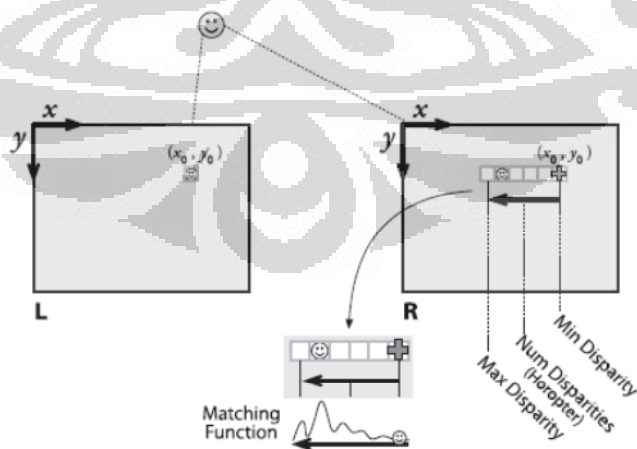
Namun, pada kenyataannya dua kamera tidak akan pernah berseketu dengan sempurna karena penempatan dua kamera tersebut secara fisik tidak pernah coplanar atau terletak pada satu bidang. Tujuan dari proses rektifikasi itu sendiri adalah menempatkan image kamera tepat pada bidang yang sama dengan setiap baris dari masing-masing image berseketu sesuai dengan *frontal parallel configuration*. Dengan melakukan proses ini, proses selanjutnya yaitu stereo corresponding dapat dilakukan dengan lebih mudah secara komputasi.

Pada prakteknya terdapat dua jenis metode stereo rektifikasi, yaitu melalui algoritma Hartley atau algoritma Bouguet. Keduanya memiliki kelebihan dan kekurangan sesuai kebutuhan dari pengguna. Namun dalam penelitian ini digunakan metode dengan algoritma Bouguet karena hasil dari proses rektifikasinya dapat memberikan informasi tentang skala image yang digunakan. Hal ini memungkinkan kita untuk mendapatkan ketelitian ukuran benda yang lebih baik berdasarkan informasi dari image kamera kanan dan kiri.



Gambar 2-15 a. Image Hasil dari *Capture* atau *Snapshot* b. Image yang Telah Dihilangkan Distorsinya, c. Image yang Telah Direktifikasi, d. Image Hasil *Crop* yang Siap Dilakukan Proses *Correspondence*

Pada saat rektifikasi selesai, hal selanjutnya adalah melakukan proses *corresponding* atau *matching* untuk setiap fitur-fitur pada masing-masing image yang berasal dari kamera kanan dan kiri. Algoritma yang digunakan untuk melakukan proses *corresponding* adalah “sum of absolute difference” (SAD) yang dikembangkan oleh Kurt Konolige [konolige97].



Gambar 2-16 Prosedur Umum dari Korespondensi yang Menghasilkan *Disparity Map*

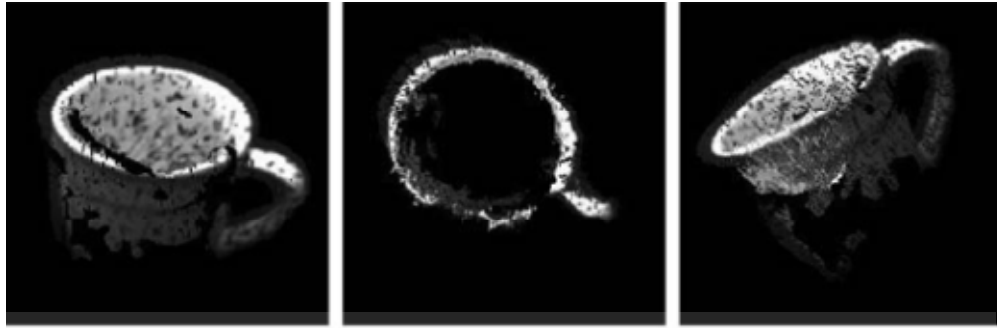
Kekurangan algoritma ini adalah ketika image yang didapat memiliki texture yang rendah seperti dinding yang polos, hanya sedikit titik-titik pada image kamera kiri dan kanan yang dapat dideteksi kedalamannya. Sedangkan pada image dengan texture tinggi seperti suasana di hutan, lukisan, di luar ruangan, dan lain sebagainya dapat menghasilkan depth map yang lebih baik. Untuk proses corresponding sendiri memiliki tiga level proses yaitu antara lain :

- 1) Melakukan proses prefiltering untuk menormalisasikan kecerahan dari suatu image dan meningkatkan texture dari image
- 2) Melakukan pencarian fitur yang saling berkorespondensi sepanjang garis epipolar horizontal dengan menggunakan SAD *window*
- 3) Melakukan postfiltering untuk menghilangkan hasil korespondensi yang buruk

Setelah didapat *disparity map*, maka bagian akhir dari teknik *stereo vision* ini adalah melakukan *reprojection*. Proses reprojection ini akan memetakan data image dua dimensi menjadi titik pada 3 dimensi dengan bantuan informasi disparity dan matriks transformasi Q yang memiliki nilai :

$$Q \begin{bmatrix} x \\ y \\ d \\ 1 \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ W \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 0 & f \\ 0 & 0 & -1/T_x & (c_x - c_y)/T_x \end{bmatrix} \quad \text{pers 2-11}$$

dengan c_x dan c_y adalah *principal point* dan T_x adalah matriks translasi kamera kanan terhadap kamera kiri. Hasil dari proses ini adalah *depth map* seperti gambar berikut ini :



Gambar 2-17 Bagian yang Terang Memiliki Kedalaman yang Lebih Kecil
Daripada yang Gelap

2.8.2 Metode *Stereo Vision* pada library OpenCV

Beruntungnya, keempat proses yang merupakan langkah utama untuk melakukan teknik *stereo vision* di atas sudah dikembangkan oleh library OpenCV sehingga para programmer baik secara langsung maupun tidak langsung telah dibantu oleh dokumentasi yang baik oleh para programmer computer vision sebelumnya. Untuk proses *undistortion*, OpenCV menyediakan fungsi `cvInitUndistortMap()` dan `cvUndistort2()` untuk setiap image. Input fungsi `cvInitUndistortMap` berupa parameter kamera intrinsic dan koefisien distorsi dan outputnya berupa titik-titik yang merupakan hasil mapping dalam bentuk array 32-bit dalam koordinat-x dan koordinat-y.

Proses *Rectifying* sendiri dilakukan dengan fungsi `cvStereoRectify` yang memiliki input berupa kamera intrinsic dan koefisien distorsi pada masing-masing kamera, sedangkan keluaran dari fungsi ini adalah matriks R_l , R_r , P_l dan P_r yang merupakan matriks rotasi dan translasi yang menggambarkan letak kamera kiri relatif terhadap kamera kanan.

Kemudian proses *corresponding* dilakukan oleh fungsi `cvFindStereoCorrespondenceBM` yang memiliki parameter input berupa data image kiri dan image kanan yang akan menghasilkan *disparity map* dan variabel `CvStereoBMState` yang berisi informasi tentang hasil dari proses *block matching*.

Setelah proses *corresponding* selesai, *depth map* bisa didapatkan dari fungsi `cvReprojectImageTo3D` yang parameternya diisi oleh *disparity map* yang merupakan hasil dari fungsi korespondensi, matriks transformasi Q , dan hasil outputnya yaitu *depth map*.

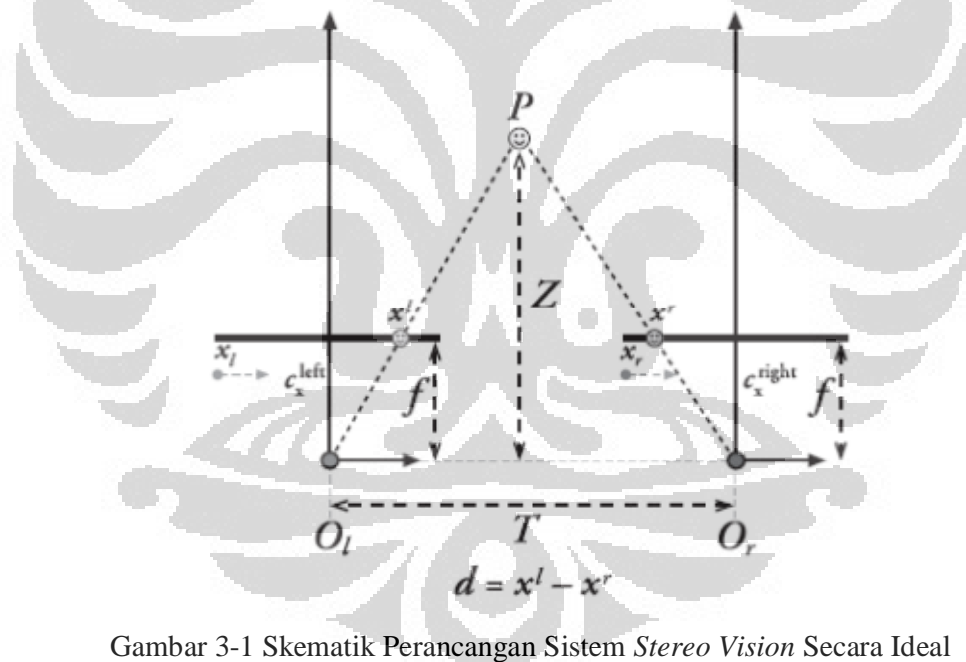
BAB 3

PERANCANGAN SISTEM

Seperti yang telah dijelaskan di bagian atas, sistem stereovision yang dibangun pada penelitian ini memiliki dua kamera yang bersisian secara frontal dan diimplementasikan sebagai satu sistem kepala robot penerima tamu. Namun, pada bab ini akan dibahas lebih detil tentang perancangan sistem *stereo vision* baik dari sisi mekanik, elektrik, program softwarena, maupun hal-hal yang menunjang kebutuhan sistem *stereo vision* secara lengkap. Berikut adalah pembahasannya :

3.1 Pendahuluan

Sistem *stereo vision* yang dibuat pada penelitian kali ini secara skematik diperlihatkan oleh gambar berikut :

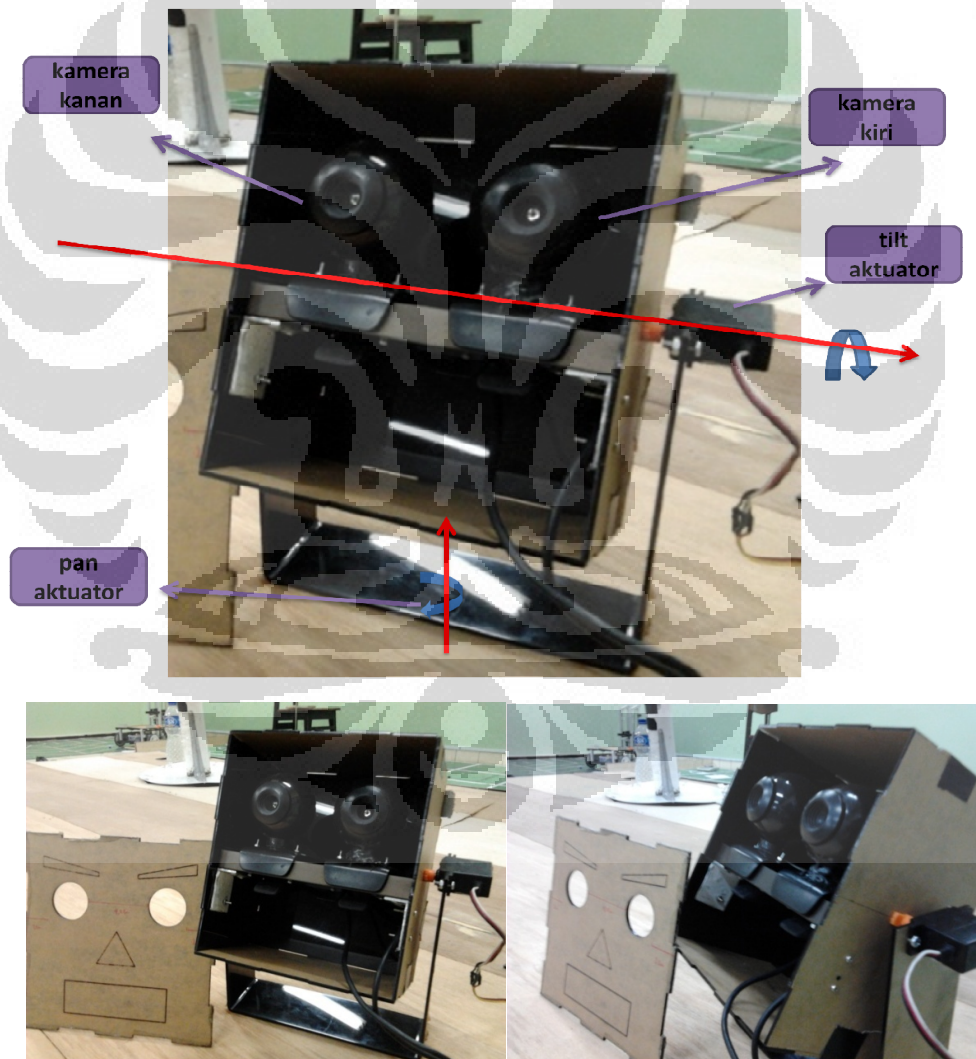


Gambar 3-1 Skematik Perancangan Sistem *Stereo Vision* Secara Ideal

Pada gambar skematik tersebut di atas dapat dijelaskan sebagai berikut, O_l dan O_r masing-masing merupakan titik pusat dari kamera kiri dan kamera kanan dengan asumsi bahwa pusat lensa merupakan titik pusat dari kamera tersebut. f merupakan besar nilai panjang focus dari kamera yang secara ideal akan memiliki nilai yang sama pada masing-masing kamera. Garis yang berasal dari titik pusat kamera yang berpotongan dengan titik c_x^{left} (atau c_x^{right}) merupakan

principal line yang menunjukkan arah frontal dari kamera. Secara ideal kedua garis untuk masing-masing kamera akan saling paralel. Namun pada kenyataannya hal tersebut akan sangat sulit didapat namun dapat diatasi dengan cukup baik pada proses kalibrasi. Titik P sendiri adalah objek gambar yang menjadi *point of interest* pada saat waktu tertentu untuk dicari kedalamannya namun secara umum semua objek pada gambar akan diperhitungkan kedalamannya. Akhirnya Z menunjukkan nilai kedalaman dari suatu objek yang ingin dideteksi.

Implementasi sistem *stereo vision* yang dibuat dalam penelitian ini berdasar pada skematil yang dijelaskan di atas, yaitu :

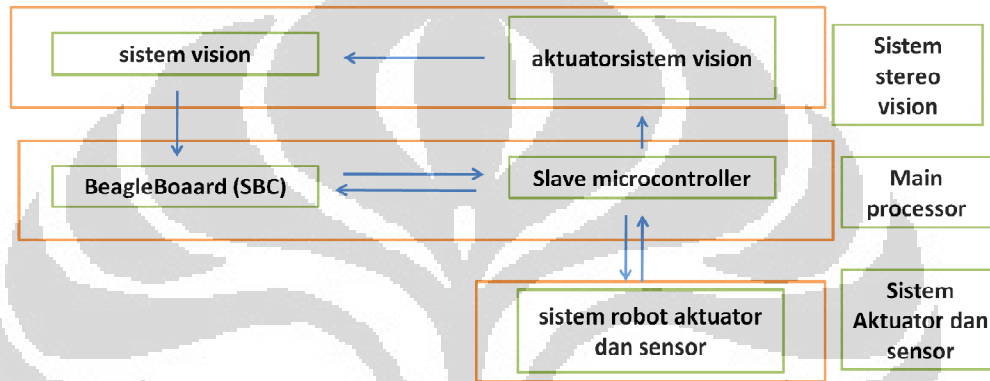


Gambar 3-2 Implementasi Sistem *Stereo Vision*

Pada implementasinya, jarak antara kamera kiri dan kanan adalah 6 cm dan arah dari kedua kamera frontal paralel. Pada bagian kiri dari kepala robot penerima tamu terdapat servo yang dapat menggerakkan kepala secara vertikal. Tujuannya adalah memperluas pandangan tanpa melakukan manuver pada tubuh robot. Hal ini akan mengurangi error akibat melakukan gerakan yang lebih besar.

3.2 Blok Diagram Sistem

Berikut adalah blok diagram sistem *stereo vision* yang telah dibangun pada penelitian ini :



Gambar 3-3 Blok Diagram Robot Penerima Tamu Secara Keseluruhan

Pada sistem ini terdapat 3 bagian besar yang menjadi inti dari sistem rancang bangunnya, yaitu sebagai berikut :

a. Sistem *Stereo vision*

Proses utama yang terjadi pada sistem ini adalah penyediaan data gambar dan melakukan aktuasi pada sistem visionnnya. Data gambar pada masing-masing kamera diberikan kepada main processor BeagleBoard melalui koneksi USB 2.0 dengan kecepatan sekitar 450 Mbit/sekon dengan besar frame gambar secara real time adalah 320 x 240 pixel pada masing-masing kamera. Sehingga untuk masing-masing kamera akan membutuhkan kecepatan transfer minimal sebesar $320 \times 240 \times 3 \text{ channel} = 225 \text{ Mbit/sekon}$ yang masih bisa diakomodir oleh USB 2.0. Jika besar frame image yang digunakan lebih besar, maka data yang akan diterima oleh main processor akan tidak sesuai dengan data sebenarnya. Hal ini akan member hasil yang buruk. Sedangkan aktuasi pada sistem visionnnya digunakan untuk melakukan proses tracking pada program *face detection*.

b. Main Processor (BeagleBoard)

BeagleBoard dalam penelitian ini digunakan sebagai pusat dari proses utama yaitu melakukan image processing mulai dari *face detection*, *stereo vision*, dan lain-lain dan juga melakukan komunikasi 2 arah dengan mikrokontroler slave.

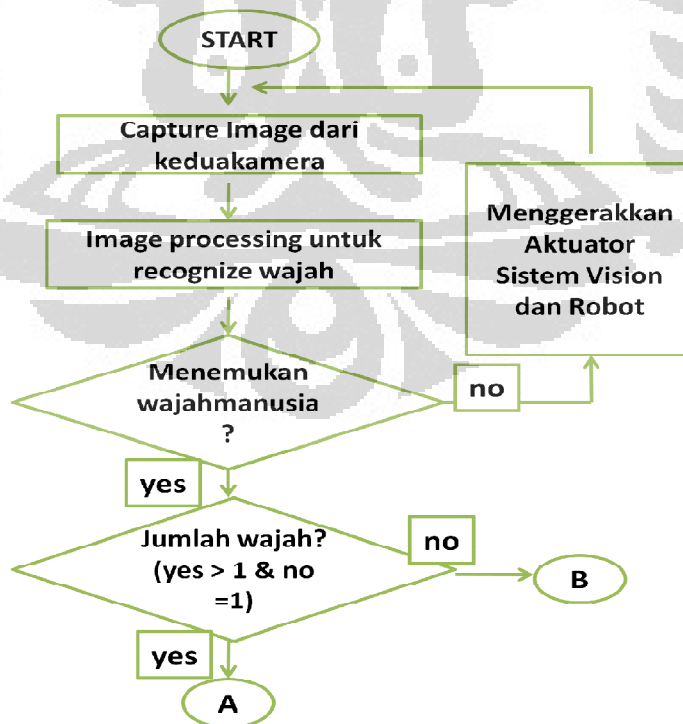
Tipe komunikasi yang antar BeagleBoard dengan mikrokontroler slave adalah komunikasi serial dengan spesifikasi baudrate 115200, 8 bit data no parity 1 stop bit. Sehingga untuk mengirimkan data 5 byte akan memerlukan jumlah bit data sebesar 1 stop bit + 40 bit data = 41 bit yang berarti akan menghabiskan waktu sekitar 0.35 ms. Namun pada kenyataannya waktu yang akan dihabiskan untuk transfer data bisa lebih dari 0.35 ms.

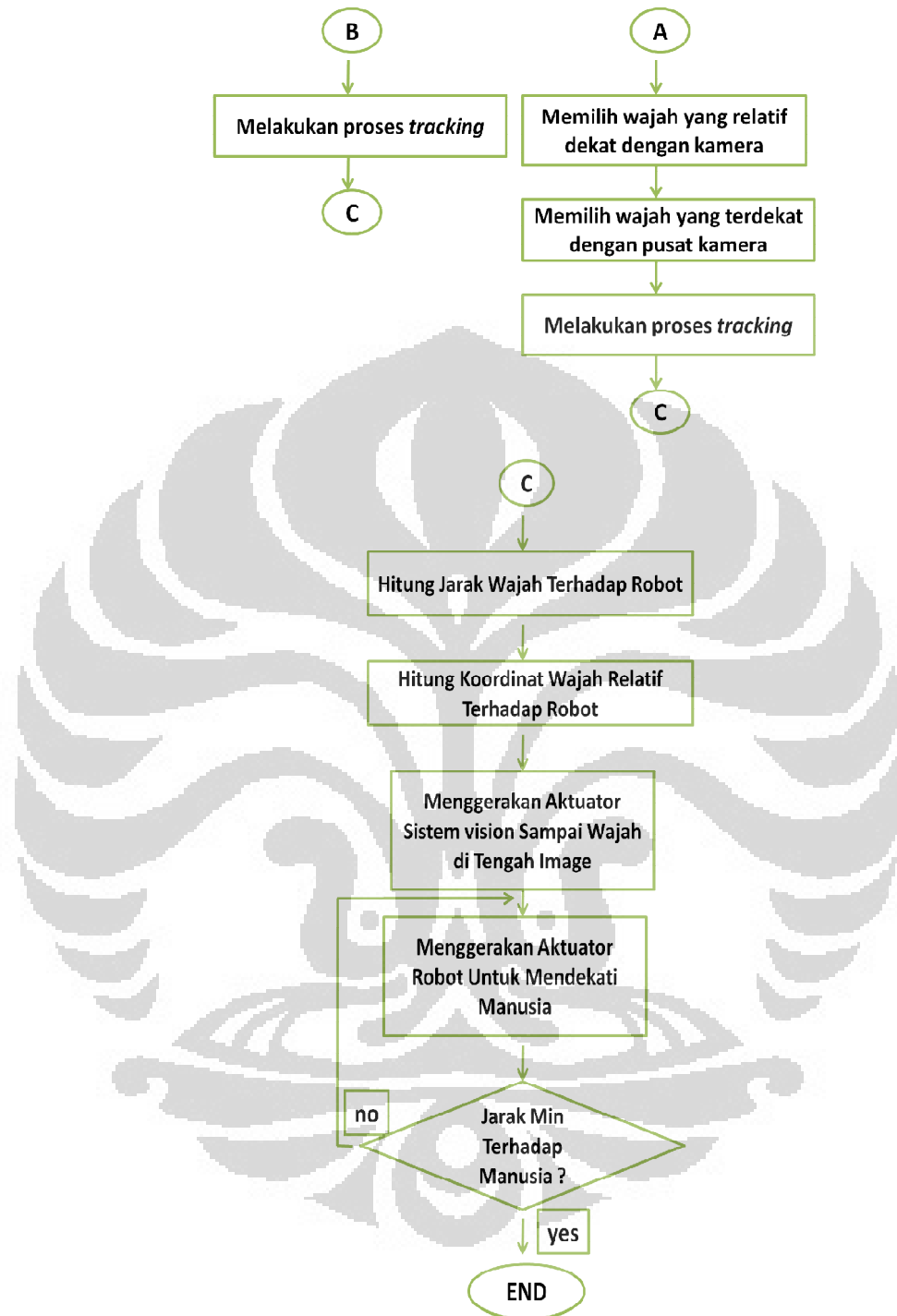
c. Sistem Aktuator dan Sensor

Sistem aktuator dan sensor di sini digunakan secara keseluruhan di tubuh robot untuk melakukan trajectory planning, navigasi, *obstacle avoid*, dan lain sebagainya. Namun cakupan bahasannya di sini hanya mencakup *pan-tilt actuator* saja.

3.3 Algoritma

Berikut adalah flowchart algoritma yang dilakukan pada sistem *stereo vision* yang dibangun pada penelitian ini :



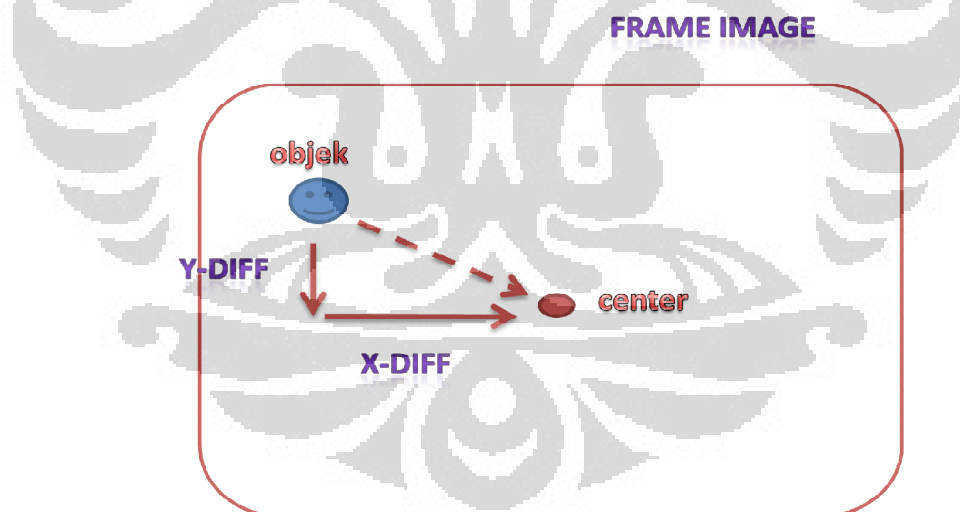


Gambar 3-4 Flowchart Algoritma Sistem

Sistem dimulai dengan melakukan *capture* melalui webcam kiri dan kanan. Proses utama yang dilakukan adalah menjalankan fungsi *face detection* sampai sistem kepala robot penerima tamu menemukan wajah manusia. Jika

belum menemukan, maka aktuator sistem vision (dengan batasan arah putar tertentu) yang berada di kepala robot penerima tamu akan bergerak memutar secara horizontal dan vertikal untuk memperluas pandangan dari kamera. Jika pergerakan dari aktuator juga tidak menemukan wajah manusia, maka aktuator robot (motor base) akan mulai bergerak untuk lebih memperluas pandangan kamera.

Setelah menemukan wajah manusia, maka terdapat dua *case* yaitu menemukan hanya satu wajah manusia dan menemukan lebih dari satu wajah manusia. Jika *case* pertama ditemukan, maka yang dilakukan adalah langsung melakukan proses *tracking* wajah manusia dan jika *case* kedua ditemukan maka yang dilakukan adalah dengan pertama kali melakukan pencarian wajah yang paling terdekat dengan kamera (atau robot) dengan ketelitian tertentu kemudian apabila jarak antara kedua wajah tersebut (dianggap) sama maka posisi wajah yang mendekati pusat image lebih dipilih untuk mengefisienkan gerakan aktuator sistem vision robot. Kemudian langsung melakukan proses *face tracking* sebagaimana pada *case* pertama.



Gambar 3-5 Informasi untuk Pergerakan Aktuator Sistem Vision

Pada saat ini, BeagleBoard akan terus melakukan image processing berupa face tracking dan memberi perintah kepada slave untuk menjalankan aktuator robot guna untuk mendekati ke arah manusia berada dalam waktu bersamaan. Prosedur untuk menggerakkan aktuator baik pada sistem vision maupun pada robot secara keseluruhan berdasarkan data informasi dari jarak

manusia terhadap robot dan perbedaaan koordinat yang didapat dari image processing dengan koordinat pusat kamera.

Apabila dari hasil image processing didapat jarak manusia relative terhadap robot telah sampai pada jaarak minimum, maka robot akan berhenti dan akan melakukan gerakan selanjutnya sebagaimana layaknya penerima tamu yaitu bersalaman, *say hello*, dan lain sebagainya. Setelah proses diatas selesai, maka robot kembali ke tempat semula untuk melakukan gerakan selanjutnya.

3.4 Pengaturan Perangkat Keras

Terdapat beberapa jenis perangkat keras (hardware) yang digunakan pada sistem ini, yaitu antara lain :

a. BeagleBoard

Telah dijelaskan di bagian atas

b. Mikrokontroller

Telah dijelaskan di bagian atas

c. Webcam

Webcam adalah video kamera yang menghasilkan citra secara real time ke komputer baik personal ataupun network melalui USB, Ethernet, ataupun wi-fi. Webcam yang digunakan untuk melakukan penelitian ini adalah logitech tipe e1000 yang memiliki spesifikasi seperti *capture resolution* maksimum 640 x 480 pixel melalui USB 2.0 dengan lensa jenis CMOS. Sedangkan spesifikasi processornya adalah RAM 256 Mb dan frekuensi 1GHz sehingga spesifikasi BeagleBoard sangatlah cukup untuk melakukan image processing yang berasal dari kamera jenis ini. Secara umum, kamera ini memiliki data yang cukup stabil dan jernih walaupun harganya termasuk murah.

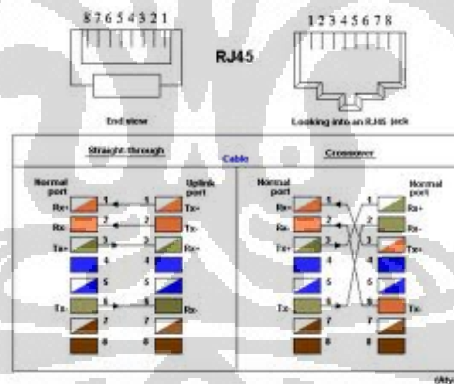


Gambar 3-6 Webcam e-1000 Logitech

http://reviews.cnet.com/webcams/logitech-e1000/4505-6502_7-33462111.html

d. Kabel LAN RJ-45

Terdapat dua jenis koneksi Ethernet menggunakan kabel UTP dan konektor RJ-45 yaitu *straight* dan *crossover*. Dalam hal ini jenis koneksi yang digunakan adalah koneksi *crossover* karena jenis koneksi ini merupakan koneksi yang menghubungkan antara PC-PC, PC-Router, dan Router-PC sedangkan koneksi *straight* menghubungkan device switch. Berikut adalah gambar koneksi tipe *crossover* dan *straight*:



Gambar 3-7 Konfigurasi *Cross* dan *Straight* RJ-45

Sumber : <http://t0.gstatic.com/images>

e. Konverter USB-Serial ATEN

Sesuai dengan bahasan di atas tentang remoting dan debugging program yang sedang berjalan di BeagleBoard melalui PC secara serial, maka tentu saja dibutuhkan interface serial. Namun karena PC sekarang ini sudah tidak terdapat lagi port serial, maka digunakan port USB 2.0 yang akan dikonversi ke serial. Untuk itu dalam penelitian ini, konverter USB-Serial yang digunakan

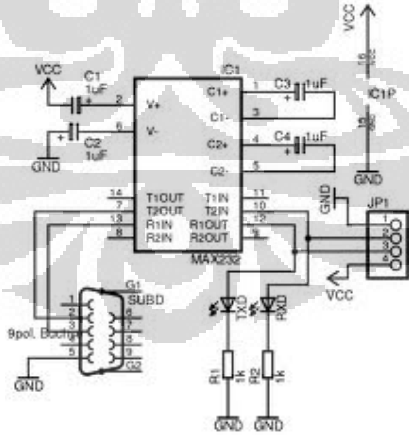
adalah merk ATEN yang berasal dari Negara Taiwan yang memang memiliki kualitas yang cukup baik. Hal ini terlihat dari konektifitas proses debugging dan remoting yang tidak pernah putus secara tiba-tiba dibandingkan dengan jenis converter USB-Serial yang mungkin relatif lebih murah.



Gambar 3-8 ATEN USB Serial
Sumber : <http://www.google.co.id/imgres>

f. Serial Konverter Max232 dan sirkuitnya

Seperti yang telah dijelaskan di subbab 2.4 sebelumnya tentang protokol serial jenis RS232, terlihat bahwa level tegangan antara RS232 dan TTL tidak sama sehingga tidak boleh langsung dilakukan proses *bypass* karena akan merusak sistem elektronik pada level tegangan TTL. Maka diperlukan suatu konverter level tegangan di antara keduanya. Jenis *integrated circuit* yang bisa menyelesaikan masalah ini adalah MAX232. Berikut adalah skematik rangkaian converter RS232-TTL :

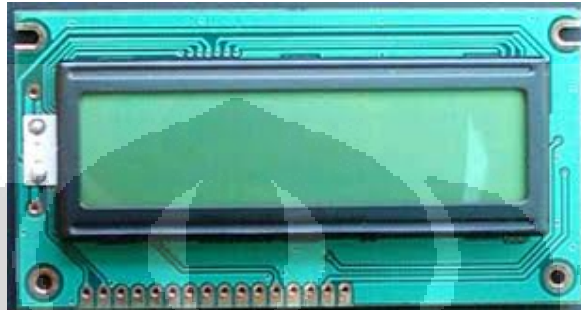


Gambar 3-9 MAX232 Serial-TTL Konverter
Sumber : <http://www.google.co.id/imgres>

g. LCD dotmatriks character 16x2

Modul LCD yang digunakan pada kali ini adalah seri DMC yang

dikembangkan oleh OPTREX CORPORATION. Modul ini terdiri atas liquid crystal panel yang memiliki contrast tinggi dan sudut penglihatan yang luas. Modulnya memiliki kontroler sendiri yang berbentuk CMOS dan beberapa driver tertentu dengan power yang kecil. Kontrolernya dilengkapi dengan karakter yang dapat dibuat sendiri.

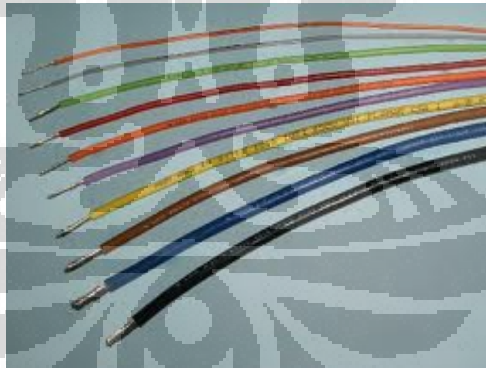


Gambar 3-10 LCD *Dot Matriks*

Sumber : <http://www.google.co.id/imgres?q=lcd+16+x+2&um>

h. Kabel AWG dan AWM

Terdapat banyak jenis kabel yang ada di pasaran, namun jenis kabel AWG dan AWM memiliki tingkat elastisitas yang baik dan tembaga yang tidak mudah rusak.



Gambar 3-11 Kabel AWG

Sumber : <http://www.google.co.id/imgres?q=awg+cable&um=1&hl=id>

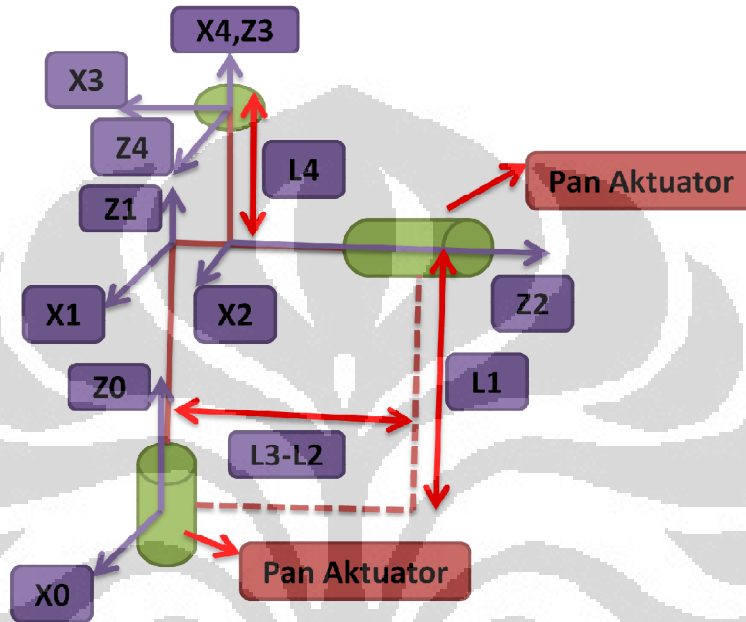
3.5 Kinematika Sistem Vision

Gambar 1.2 menunjukkan gambar keseluruhan dari robot penerima tamu yang dibuat. Jika dilakukan pembahasan secara keseluruhan maka terdapat lebih dari empat aktuator yang akan dihitung kinematikanya. Namun, dalam

pembahasan ini lebih diprioritaskan pada aktuator vision saja yang berjumlah dua aktuator sebagaimana divisualisasikan pada gambar 3-2.

3.5.1 Pemodelan Aktuator Sistem Vision

Berikut adalah gambar model Denavit-Hartenberg aktuator sistem vision pada penelitian kali ini,



Gambar 3-12 Model Denavit-Hartenberg Aktuator Sistem Vision

Berdasarkan gambar 3-12 yang menggambarkan hubungan antara sistem aktuator kamera dengan *end-effector*, yang dalam hal ini berupa frame, dapat diturunkan parameter Denavit-Hartenberg dan persamaan transformasi homogenous untuk masing-masing hubungan jointnya, yaitu :

Table 3-1 Parameter Denavit-Hartenberg Sistem Aktuator Vision

Link(i)	α_{i-1}	a_{i-1}	d_i	θ_i
1	0	0	L1	θ_1
2	-90^0	0	L2	θ_2
3	0	0	L3	0

Dari data parameter sesuai dengan tabel 3-1 dapat diturunkan persamaan matriks transformasi homogenous.

Transformasi kinematika dari base menuju aktuator pan,

$${}^{base}T_1 = \begin{bmatrix} \cos q1 & -\sin q1 & 0 & 0 \\ \sin q1 & \cos q1 & 0 & 0 \\ 1 & 0 & 1 & L1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{pers 3-1}$$

Transformasi kinematika dari aktuator tilt menuju aktuator pan,

$${}^1T_2 = \begin{bmatrix} \cos q2 & -\sin q2 & 0 & 0 \\ 0 & 0 & 1 & L3 - L2 \\ -\sin q2 & \cos q2 & 0 & L0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{pers 3-2}$$

Transformasi kinematika dari aktuator pan menuju *end-effector*,

$${}^2T_{end-effector} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{pers 3-3}$$

Pemodelan transformasi di atas akan digunakan untuk proses *tracking* apabila robot penerima tamu telah menemukan wajah manusia. Sesuai dengan yang telah dijelaskan pada subbab 3.3, proses *tracking* mengharuskan wajah manusia selalu terletak pada tengah image kamera atau dengan kata lain bidang image akan selalu tegak lurus dengan objek wajah sehingga proses perhitungan jarak dengan teknik *stereo vision* dapat dihitung secara langsung tanpa melakukan penghitungan secara geometri akibat objek yang tidak terletak tepat pada tengah image frame. Hal ini dapat mengurangi dan mengefisiensikan beban komputasi.

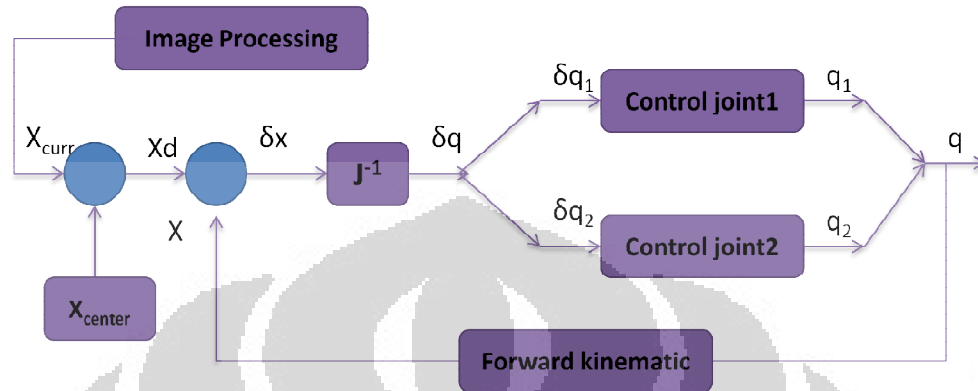
Sesuai dengan subbab 2.6, model proyeksi yang digunakan di sini adalah tipe *perspective projection* dengan formula umum,

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{f}{Z} \begin{bmatrix} X_{cam} \\ Y_{cam} \end{bmatrix} \quad \text{pers 3-4}$$

Tipe model proyeksi ini sesuai dengan algoritma yang dibangun sesuai dengan pembahasan subbab 3-3 karena kedalaman Z_{cam} yang didapat dari proses *stereo vision* bisa langsung diimplementasikan ke persamaan 3-4.

3.5.2 Tracking Control Aktuator Sistem Vision

Setelah pemodelan aktuator sistem vision selesai, maka sekarang proses *tracking control* dapat dengan lebih mudah diturunkan. *Tracking control* yang digunakan adalah *rate motion rate control*. Berikut adalah blok diagramnya :



Gambar 3-13 Blok Diagram *Tracking Control* Sistem Aktuator Vision

Proses *image processing* akan memberikan informasi berupa koordinat objek wajah pada saat sekarang x_{curr} . Kemudian nilai x_{curr} akan langsung dibandingkan dengan x_{center} ,

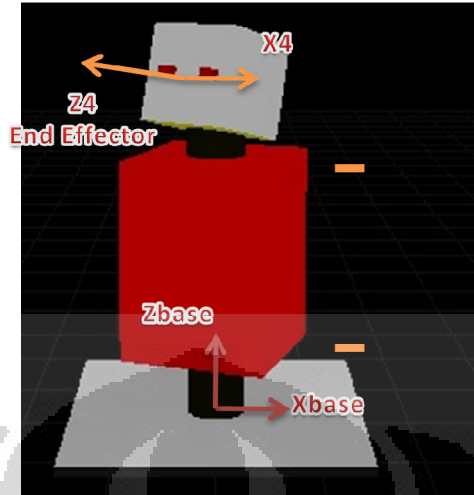
$$x_d = x_{curr} - x_{center} \quad \text{pers 3-5}$$

untuk mendapatkan nilai x_d yang berfungsi sebagai seberapa besar perpindahan sudut yang harus dikerjakan oleh aktuator. Nilai x_d nantinya akan dibandingkan dengan x yang merupakan hasil dari proses forward kinematic untuk mendapatkan update posisi aktuator sekarang,

$$\delta x = x_d - x \quad \text{pers 3-6}$$

$$x = f(\theta) \quad \text{pers 3-7}$$

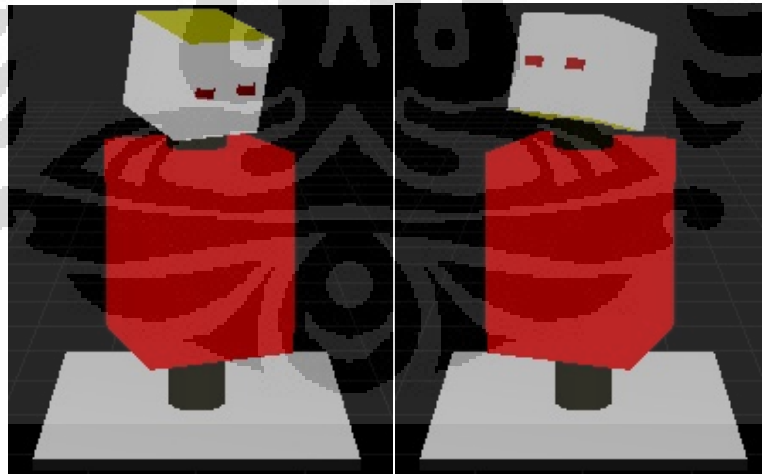
Untuk mendapatkan seberapa besar *rate* sudut δq yang harus ditempuh oleh aktuator berdasarkan masukan *rate* perpindahan koordinat δx digunakan invers jacobian. Setelah didapatkan *rate* sudut δq , maka *rate* sudut δq akan diberikan faktor pengali K_p dan K_v tertentu dan diberikan ke masing-masing aktuatornya, dalam hal ini servo, yang kemudian nilai q yang terakhir akan digunakan untuk keperluan mencari nilai koordinat x melalui persamaan forward kinematik.



Gambar 3-14 *End-effector* Robot Penerima Tamu Terhadap Titik Pusat *Base*

Untuk selanjutnya, setelah proses *tracking* dan informasi tentang objek seperti jaraknya didapatkan, maka dengan bantuan transformasi frame objek ke motor base robot penerima tamu, sesuai dengan gambar 3-14, akan memudahkan proses trajectory planning dari sistem robot penerima tamu untuk mendekati manusia yang dideteksi wajahnya sehingga akhirnya dapat berinteraksi dengan manusia seperti melakukan jabat tangan, *says "hello"*, dan lain sebagainya

Berikut adalah gambar simulasi contoh pergerakan keseluruhan robot :



Gambar 3-15 Contoh Pergerakan dari Keseluruhan Robot

BAB 4

PENGUJIAN DAN ANALISIS

Eksperimen ini bertujuan untuk menguji dan menganalisis sistem komputer vision, khususnya *stereo vision*, berdasarkan percobaan yang telah dilakukan. Pengujian dilakukan sesuai dengan proses berjalannya sistem *stereo vision* mulai dari proses kalibrasi sampai pada proses *reprojection* secara riil. Berikut adalah pembahasannya :

4.1 Kalibrasi Dual Kamera

Sebagaimana telah dijelaskan di bab sebelumnya, penggunaan teknik *stereo vision* dalam penelitian ini membutuhkan dua kamera yang dipasang secara frontal paralel. Secara ideal kedua kamera tersebut akan sama-sama memiliki garis principal yang akan berpotongan pada jarak tak hingga, namun pada kenyataannya sangat sulit mendapatkan konfigurasi seperti itu. Oleh karena itu, dilakukan kalibrasi yang akan menghasilkan parameter ekstrinsik R dan T yang masing-masing menjelaskan orientasi dan translasi kamera kanan relatif terhadap kamera kiri. Matriks R dan T dapat dirumuskan sebagai berikut :

$$R = R_r (R_l)^T \quad \text{pers4-1}$$

$$T = T_r - RT_l \quad \text{pers4-2}$$

Dengan R_r (atau R_l) adalah matriks rotasi yang menjelaskan orientasi objek terhadap kamera kanan (atau kamera kiri) dan T_r (atau T_l) adalah matriks translasi yang menjelaskan jarak objek terhadap kamera kanan (atau kiri). Sedangkan R dan T adalah matriks rotasi dan translasi yang menghubungkan antara kamera kanan dan kiri. Output dari hasil kalibrasi ini akan digunakan oleh proses *undistortion* dan *rectifying*.

Proses kalibrasi dimulai dengan mengambil data-data *snapshot* dari berbagai pose kumpulan bidang hitam putih misalnya *chessboard*. *Chessboard* digunakan karena titik-titik koordinatnya dapat dengan mudah ditemukan di image dengan membedakan warna hitam dan putih. Untuk hasil yang lebih baik, maka pose *chessboard*nya dibedakan di setiap *snapshot*.

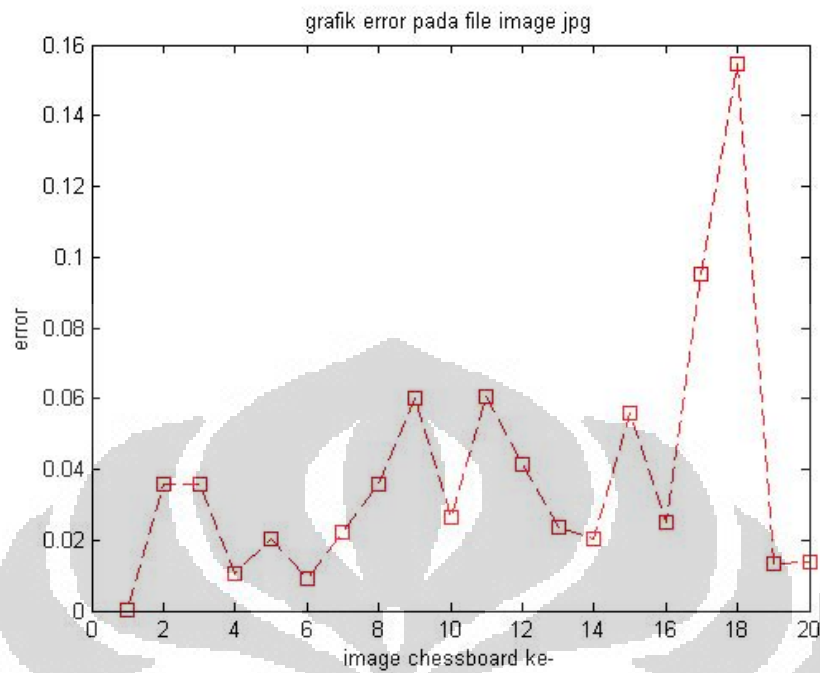


Gambar 4-1 Hasil *Snapshot* pada Kamera Kanan dan Kiri yang Dilakukan pada Saat Bersamaan

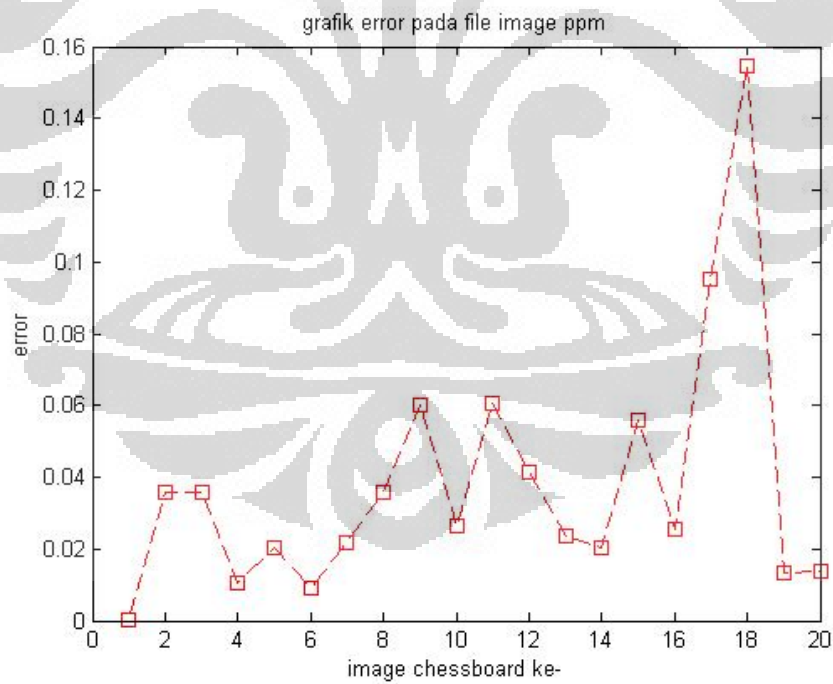
Pada penelitian ini, percobaan proses kalibrasi dilakukan dengan *offline* untuk mengurangi beban komputasi yang tinggi dengan cara membandingkan error hasil dari perbedaan data titik-titik koordinat *chessboard* dengan perhitungan titik-titik berdasarkan gris epipolar yang dihasilkan oleh file jpg dan ppm.

Gambar hasil *snapshot* ini merupakan file berekstensi jpg yang telah dikompresi sedemikian rupa sehingga filenya pun lebih kecil dari yang seharusnya. Sedangkan file gambar berekstensi ppm (*portable pixmap file*) merupakan gambar yang berisi informasi secara “berlimpah” selain informasi warna untuk tujuan kualitas sehingga rata-rata ukuran filenya bisa lebih besar 10 kali dari besaran file jpg. Tipe file image ini cocok untuk proses image yang memerlukan detail yang tinggi.

Untuk data pengambilan gambar sebanyak 40 kali masing-masing 20 kali setiap kamera didapat errornya sesuai grafik berikut :



Gambar 4-2 Grafik Error Terhadap Image *Chessboard* Hasil dari *Snapshot*



Gambar 4-3 Grafik Error Terhadap Image *Chessboard* Hasil dari *Snapshot*

Table 4-1 Error Hasil Kalibrasi *File Image*

Tipe Image	Rata error hasil kalibrasi
Jpg	0.794442
Ppm	0.794261

Dari tabel di atas tampak bahwa penggunaan file gambar berekstensi ppm akan menghasilkan error yang lebih kecil dari pada file gambar berekstensi jpg sebesar 0.022 persen.

Hasil kalibrasi dengan menggunakan file PPM akan memiliki error yang lebih kecil karena data-data pada raw imagenya sendiri sangat *redundant* yang bahkan mata manusia pun tidak cukup baik untuk melihatnya. Tingkat detail yang tinggi ini sangat baik untuk proses kalibrasi kamera khususnya, karena diperlukannya keakuratan untuk menemukan sudut pada kotak-kotak di *chessboard* dan menghubungkannya dengan membentuk garis yang berguna untuk menemukan ukuran intrinsik kamera yang memiliki ketelitian sampai 6 angka di belakang koma.

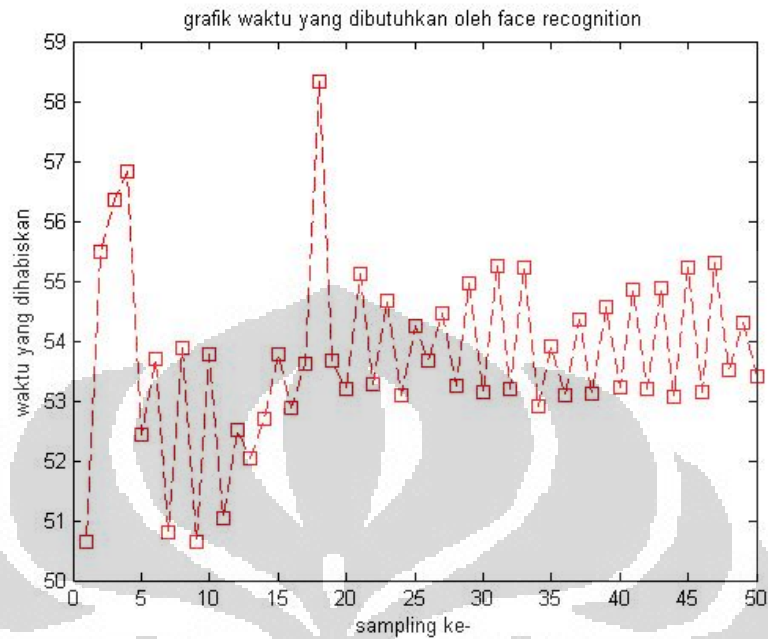
Pada prakteknya, file tipe ppm akan memberikan hasil yang lebih baik secara signifikan dibandingkan dengan file tipe jpg jika hasil snapshot *chessboard* untuk kalibrasinya memiliki pose yang ekstrim, ukurannya kecil relative terhadap seluruh image.

4.2 Face detection

Seperti yang telah dijelaskan di bab sebelumnya, metode yang digunakan untuk *face detection* dalam penelitian ini adalah AdaBoost. Metode ini hanya memerlukan sedikit fitur pada setiap gambar yaitu *Haar features* dan akan digabungkan secara efisien oleh sebuah *cascade classifier*.

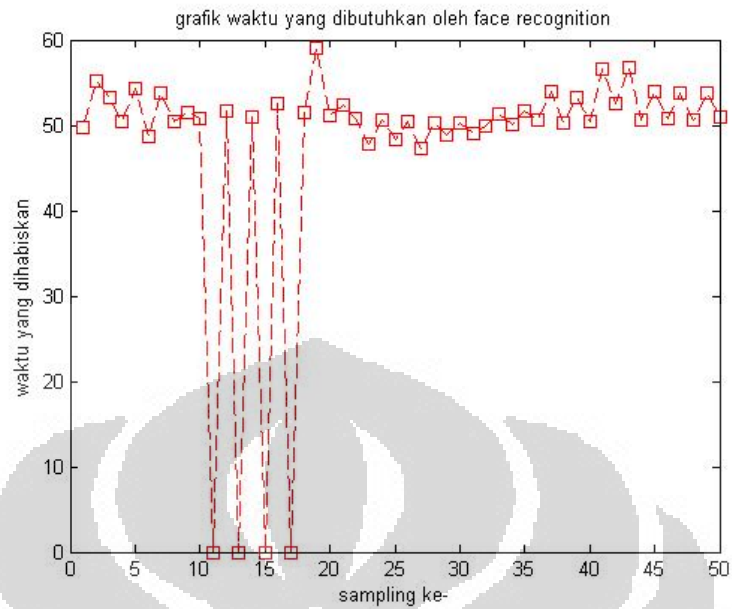
Pada penelitian ini, dilakukan pengujian untuk mengetahui tingkat keberhasilan *face detection* mulai dari jarak paling dekat yaitu 40 cm (kondisi dimana kedua frame kamera sudah dipenuhi oleh wajah) sampai jarak paling jauh yang mungkin masih dicapai oleh metode *face detection* ini. Perhitungan waktu juga dipertimbangkan untuk menunjukkan kekonsistenan algoritmanya.

Berikut adalah gambaran grafiknya :



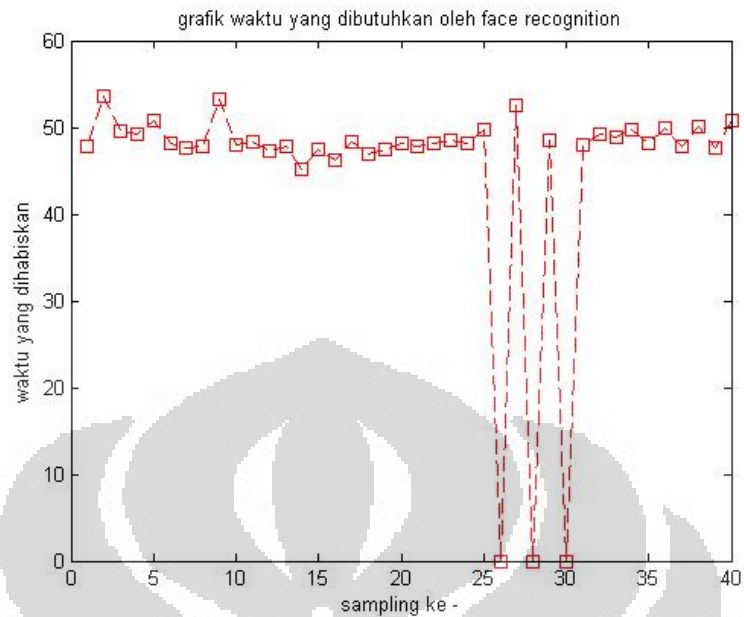
Gambar 4-4 Grafik untuk Percobaan *Face Detection* dengan Jarak 40 cm (Waktu Dikali dengan 20)

Hasil di atas menunjukkan bahwa pada 50 kali sampling data, success rate yang dicapai adalah 100 persen dan waktun rata-ratanya mencapai 1080 milidetik dengan deviasi sekitar 80 milidetik. Hal ini menunjukkan bahwa untuk jarak sekitar 40 cm, algoritma AdaBoost untuk *face detection* berjalan dengan baik dan waktu yang dihabiskan cukup konsisten sesuai data yang telah didapat. Hal ini bisa dilihat dari sampling ke-20 dan di atasnya variasinya mulai mengecil.



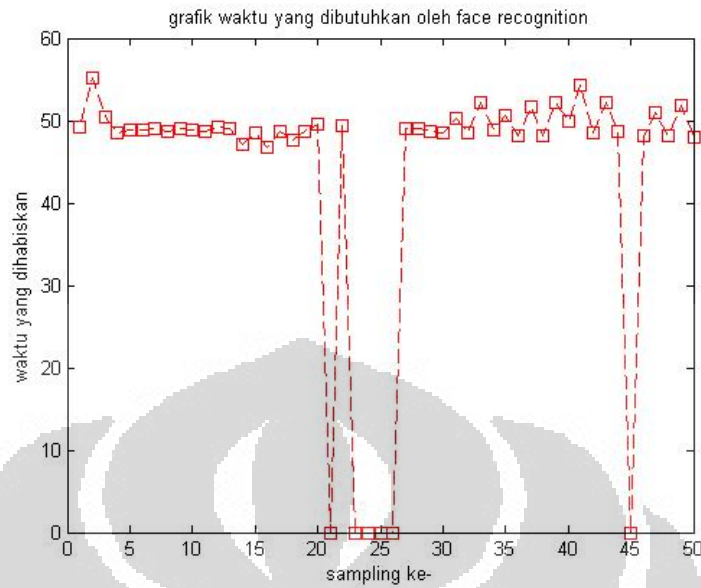
Gambar 4-5 Grafik untuk Percobaan *Face Detection* dengan Jarak 60 cm (Waktu Dikali dengan 20)

Hasil di atas menunjukkan success rate yang dicapai tidak 100 persen karena terdapat 4 kali sampling dengan kegagalan pembacaan wajah. Hal ini kemungkinan terjadi karena ketidakstabilan pada proses pengambilan data pada kamera. waktu rata-ratanya mencapai 1000 milidetik dengan deviasi sekitar 45 milidetik. hasilnya menunjukkan bahwa untuk jarak sekitar 60 cm, metode *face detection* cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten sesuai data yang telah didapat karena dari sampling ke-20 dan di atasnya variasinya mulai mengecil.



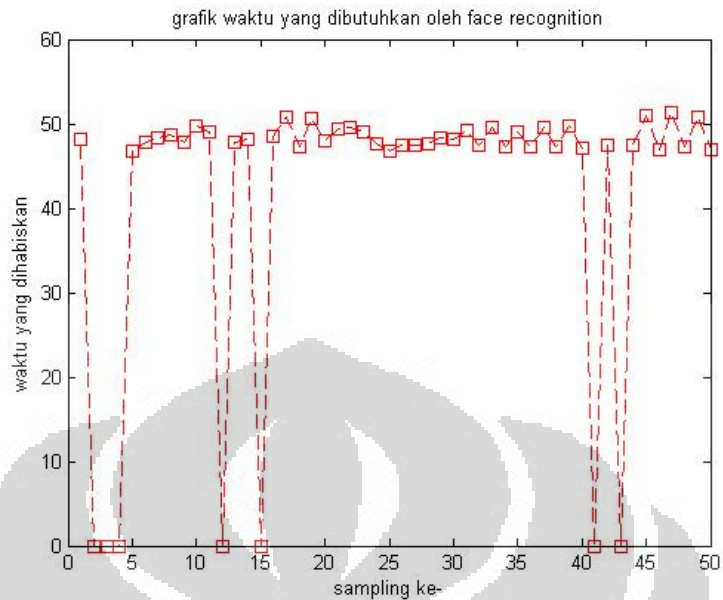
Gambar 4-6 Grafik untuk Percobaan *Face Detection* dengan Jarak 80 cm (Waktu Dikali dengan 20)

Jika pada jarak 60 cm kesalahan pembacaan terjadi pada sampling 10-15, maka pada jarak 80 cm kesalahan terjadi pada sampling sekitar 25-30 yang menunjukkan kesalahannya tidak sama dengan kesalahan pada jarak 60 cm. Semakin jauh jarak wajah terhadap sistem kamera, maka semakin kecil pula informasi yang didapat dari proses *Haar features*-nya. Hal ini menyebabkan terdapat waktu dimana *classifier* tidak dapat mendeteksi wajah. Namun, hasil di atas memiliki success rate yang dicapai tidak 94 persen karena terdapat 3 kali sampling dengan kegagalan pembacaan wajah. waktu rata-ratanya mencapai 1000 milidetik dengan deviasi sekitar 30 milidetik. hasilnya menunjukkan bahwa untuk jarak sekitar 80 cm, metode *face detection* cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten.



Gambar 4-7 Grafik untuk Percobaan *Face Detection* dengan Jarak 100 cm (Waktu Dikali dengan 20)

Hasil pembacaan 100 cm lebih sedikit tidak stabil dibandingkan jarak yang lebih kecil dari 100 cm. Hal ini dapat dimengerti sesuai pembahasan sebelumnya yaitu Semakin jauh jarak wajah terhadap sistem kamera, maka semakin kecil pula informasi yang didapat dari proses *Haar features*-nya. Namun, hasilnya memiliki success rate yang cukup baik yaitu mencapai 90 persen karena terdapat 5-6 kali sampling dengan kegagalan pembacaan wajah. waktu rata-ratanya juga mencapai 1000 milidetik dengan deviasi sekitar 60 milidetik. hasilnya menunjukkan bahwa untuk jarak sekitar 100 cm, metode *face detection* cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten.



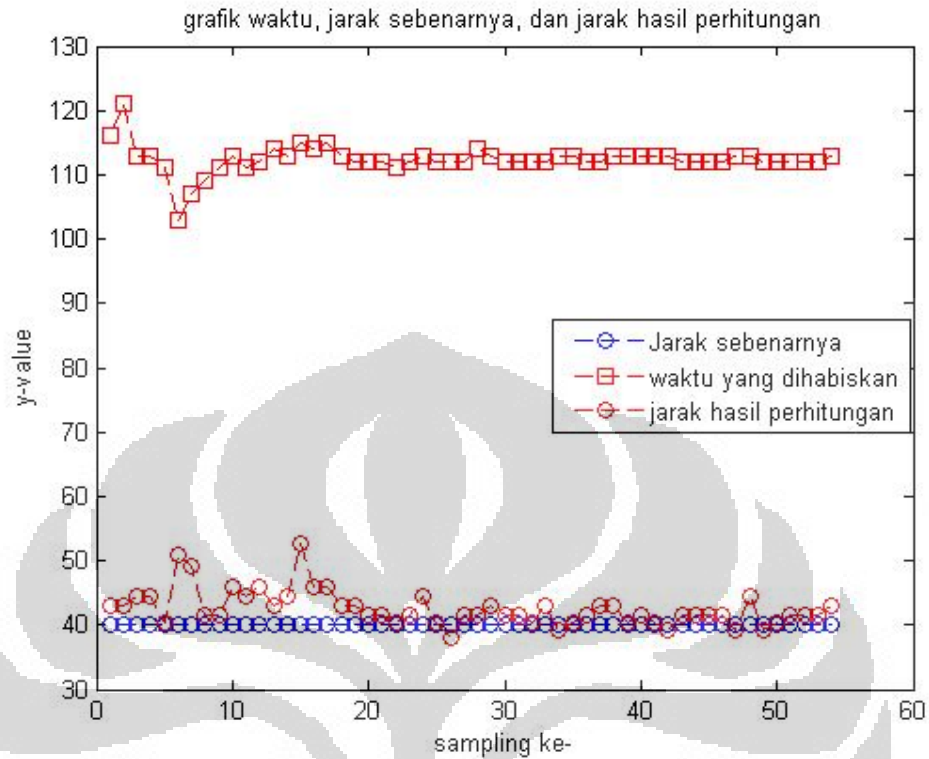
Gambar 4-8 Grafik untuk Percobaan *Face Detection* dengan Jarak 120 cm (Waktu Dikali dengan 20)

Hasil pembacaan 120 cm tentu saja lebih tidak stabil dibandingkan jarak yang lebih kecil dari 120 cm. Hal ini juga dapat dimengerti sesuai pembahasan sebelumnya. Hasilnya memiliki success rate yang lumayan baik yaitu mencapai 85 persen karena terdapat 7-8 kali sampling dengan kegagalan pembacaan wajah. Waktu rata-ratanya juga mencapai 1000 milidetik dengan deviasi sekitar 55. Hasilnya menunjukkan bahwa untuk jarak sekitar 120 cm, metode *face detection* cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten.

4.3 Stereo Vision

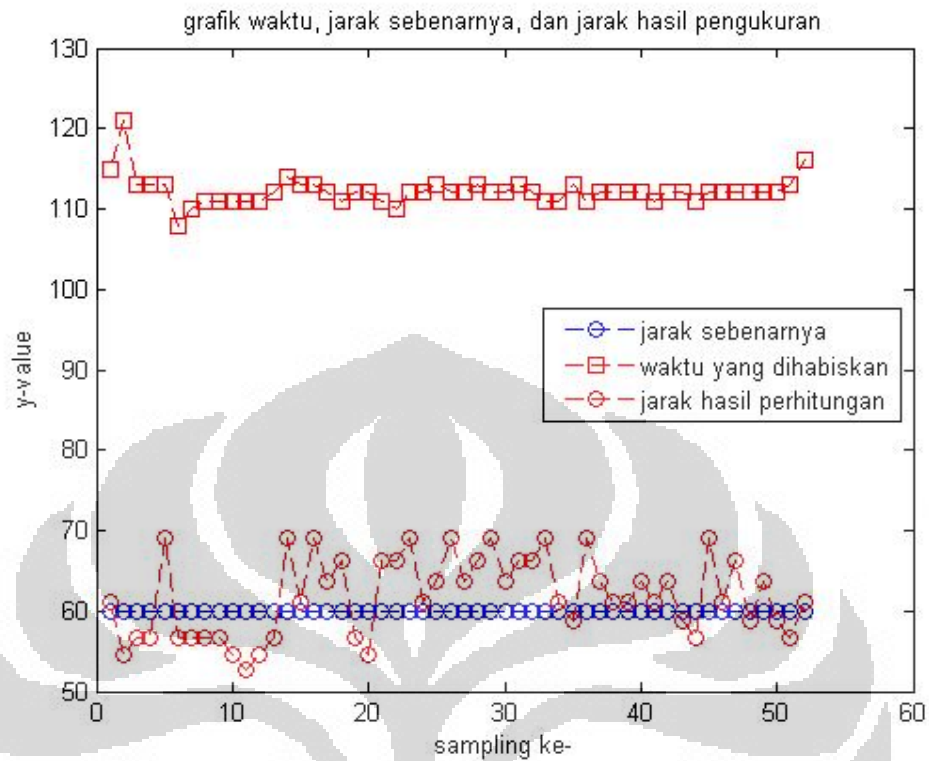
Prosedur untuk melakukan pengujian penghitungan jarak dalam penelitian ini adalah dengan cara mengukur jarak wajah relative terhadap sistem kamera untuk setiap jarak yang *fix*. Hasilnya akan dibandingkan dengan jarak yang sebenarnya untuk mengetahui besarnya error. Pada saat itu juga dihitung waktu yang dihabiskan untuk setiap sampling.

Berikut adalah hasil dari percobaannya :



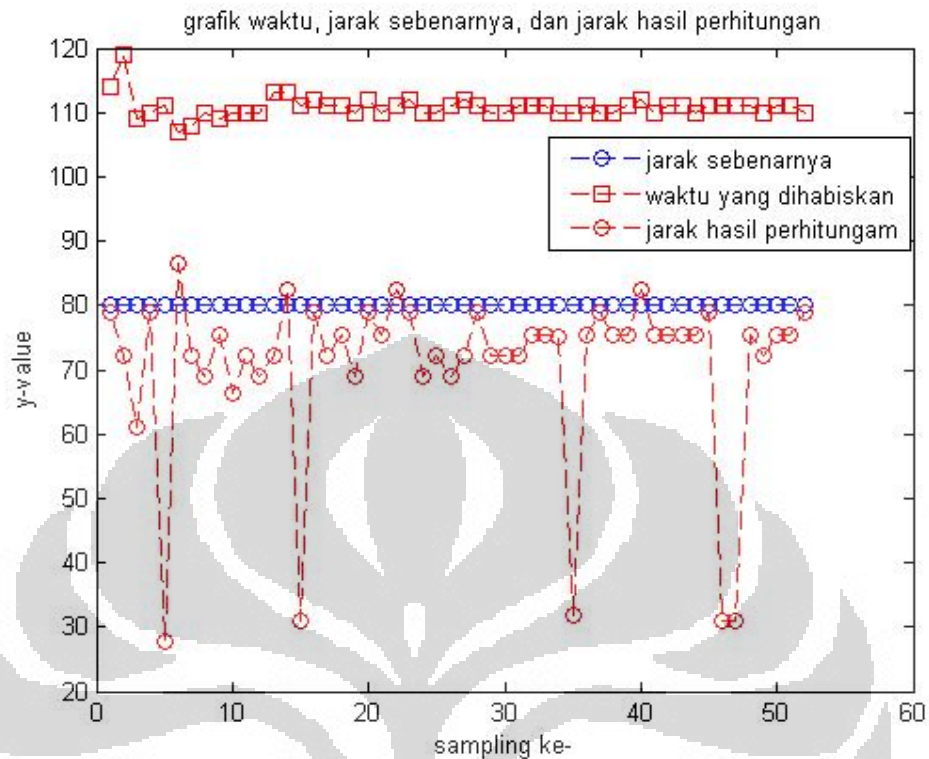
Gambar 4-9 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 40 cm
(Waktu Dikali dengan 20)

Hasil pembacaan jarak 40 cm ditunjukkan pada grafik di atas. Dimana secara umum pembacaannya sudah cukup stabil pada sampling bagian akhir. Ketidakstabilan pada bagian awal dapat terjadi karena data *raw image* dari kedua kamera belum terlalu stabil. Namun, secara keseluruhan hasilnya memiliki error yang cukup kecil. waktu rata-ratanya yang dibutuhkan untuk sekali sampling mencapai 2240 milidetik dengan deviasi sekitar 23 milidetik. hasilnya menunjukkan bahwa untuk jarak sekitar 40 cm, penghitungan jarak cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten.



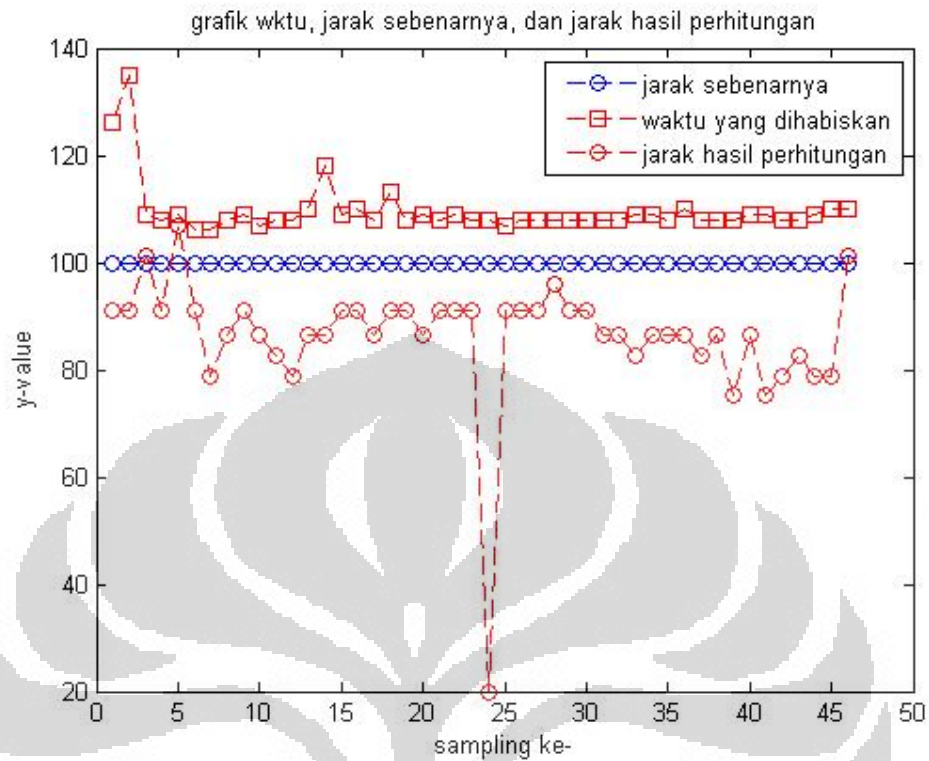
Gambar 4-10 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 60 cm
(Waktu Dikali dengan 20)

Hasil pembacaan jarak 60 cm menunjukkan nilai yang kurang stabil namun, masih dalam jangkauan nilai 60 cm. Ketidakstabilan pembacaan bisa terjadi karena hasil dari kalibrasi yang kurang baik. Namun, secara keseluruhan hasilnya memiliki error yang masih dapat ditoleransi. waktu rata-ratanya yang dibutuhkan untuk sekali sampling mencapai sekitar 2240 milidetik dengan deviasi sekitar 20 milidetik. hasilnya menunjukkan bahwa untuk jarak sekitar 60 cm, penghitungan jarak cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten.



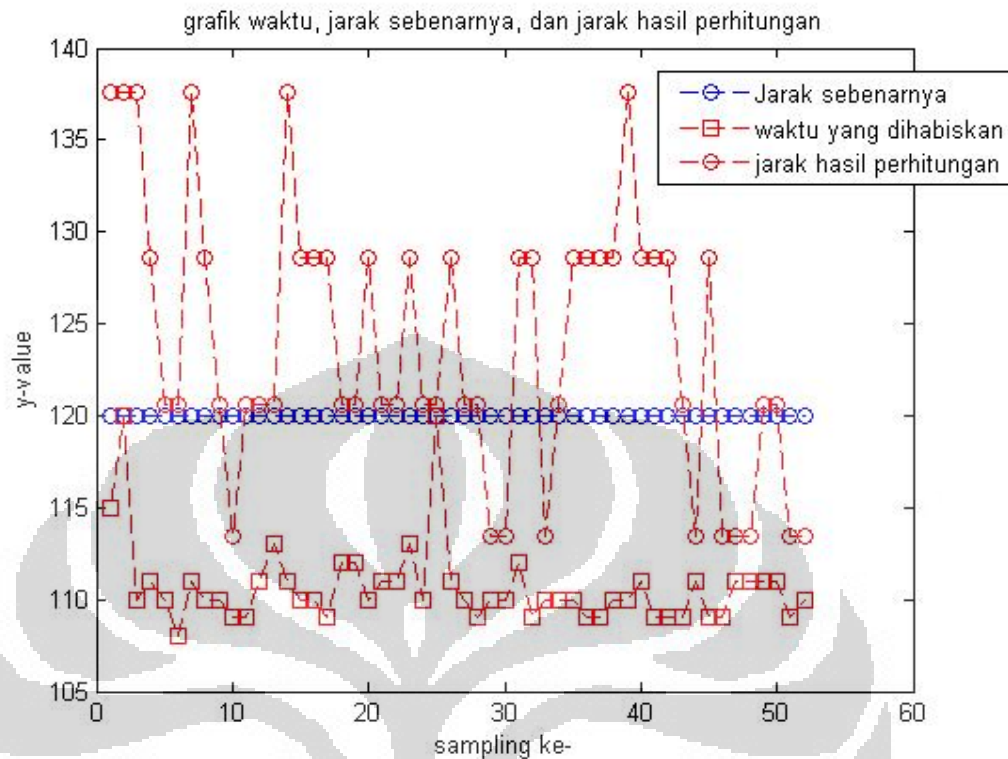
Gambar 4-11 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 80 cm
(Waktu Dikali dengan 20)

Hasil pembacaan jarak 80 cm menunjukkan nilai yang tidak lebih baik dari jarak 60 cm. hal ini terlihat dari beberapa nilai yang berada di luar jangkauan yaitu mencapai 30 cm. Kesalahan pembacaan sangat mungkin berasal dari kalibrasi yang kurang baik. Namun, secara keseluruhan kesalahan pembacaan hanya terjadi sekitar 5 persen dan hasilnya memiliki error yang masih dapat ditoleransi. waktu rata-ratanya yang dibutuhkan untuk sekali sampling mencapai sekitar 2200 milidetik dengan deviasi sekitar 20 milidetik. hasilnya menunjukkan bahwa untuk jarak sekitar 60 cm, penghitungan jarak cukup berjalan dengan baik dan waktu yang dihabiskan cukup konsisten.



Gambar 4-12 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 100 cm
(Waktu Dikali dengan 20)

Kestabilan hasil pembacaan jarak 100 cm sudah cukup baik namun error hasil pembacaannya cukup besar yaitu rata-rata 10 cm. Bahkan terdapat satu *outlayer* sekitar 20 cm. Kesalahan pembacaan kembali sangat mungkin berasal dari kalibrasi yang kurang baik dan ketidakstabilan pengambilan data dari kamera. Pembacaan jarak yang cukup jauh juga mengurangi keakuratan pembacaan akibat terdapat informasi yang mulai hilang. Waktu rata-ratanya yang dibutuhkan untuk sekali sampling mencapai sekitar 2200 milidetik dengan deviasi sekitar 15 milidetik.



Gambar 4-13 Grafik untuk Percobaan Penghitungan Jarak pada Jarak 120 cm
(Waktu Dikali dengan 20)

Kestabilan hasil pembacaan jarak 120 cm adalah yang paling buruk dari yang lain. Hal ini menunjukkan bahwa mulai jarak 120 cm ke atas, perhitungan yang dihasilkan mulai salah. Sehingga dapat dikatakan jarak paling jauh yang bisa dideteksi jaraknya adalah sekitar 100 cm.

4.4 Design Output untuk Interaksi dengan Manusia

Jika Dalam suatu ruangan ataupun tempat dimana jumlah manusia yang berada di tempat itu belum diketahui akan menyulit bagi robot penerima tamu untuk melakukan *decision* misalnya *tracking* ke salah satu manusia tersebut. Sebenarnya para ahli *computer vision* sudah memiliki beberapa metode untuk menyelesaikan masalah tersebut salah satunya adalah melakukan objek matching. Namun, metode ini memiliki beberapa kekurangan yang dapat mengurangi performa sistem secara keseluruhan yakni waktu komputasi yang tinggi karena harus melakukan perhitungan di hamper seluruh image dan untuk objek yang memiliki tekstur rendah sangat kecil kemungkinannya untuk dilakukan proses

matching. Oleh karena itu, metode ini tidak cocok untuk diimplementasikan di sistem robot penerima tamu. Maka, di dalam penelitian digunakan metode lain dengan memanfaatkan metode *face detection*.

Penggunaan *face detection* dengan metode AdaBoost memiliki keunggulan tersendiri untuk beberapa hal. Pengimplementasian metode ini di OpenCV sendiri dilakukan oleh fungsi `cvHaarDetectObjects` yang akan mengembalikan informasi wajah manusia yang didapat. Jadi fungsi ini melakukan klasifikasi apakah sebuah objek dikatakan sebagai sebuah wajah atau bukan. Dengan memanfaatkan informasi tersebut, robot penerima tamu akan mengetahui di suatu ruangan atau tempat tersebut ada atau tidaknya manusia dan ramai atau sepi.

Berikut prosedur untuk membedakan kondisi suatu tempat dimana robot penerima tamu berada :

1. Kondisi sepi atau tidak ada manusia

Dengan menggunakan looping return fungsi `cvHaarDetectObjects` yang bernilai nol, robot penerima tamu akan mengetahui bahwa tidak ada manusia di hadapannya. Ketika kondisi ini terjadi maka robot penerima tamu akan melakukan maneuver pada aktuator visualnya untuk mencari manusia. Jika setelah proses terakhir terakhir masih belum menemukan wajah manusia, maka dilakukan maneuver pada robot *wheel*-nya. Jika setelah itu masih tetap belum menemukan wajah manusia, maka robot penerima tamu akan mengeluarkan output yang menunjukkan bahwa tidak ada manusia yang ditemui.

2. Kondisi ramai atau terdapat beberapa manusia

Berlawanan dengan kondisi pertama, fungsi `cvHaarDetectObjects` akan mengembalikan nilai tidak nol yang menunjukkan ada beberapa wajah manusia yang terdeteksi. Jumlah manusia yang terdeteksi sebanding dengan nilai *return* dari fungsi `cvHaarDetectObjects`. Setelah informasi jumlah manusia terdeteksi, maka langkah selanjutnya adalah melakukan *tracking* ke manusia yang paling dekat.

3. Kondisi robot penerima tamu masih berada jauh dari manusia

Kondisi sekarang adalah robot penerima tamu telah mengetahui ada manusia dan sedang melakukan tracking wajah manusia tersebut. Pada saat itu juga, dengan algoritma *stereo vision* jarak manusia, termasuk orientasi robot terhadap manusia, terhadap robot dihitung untuk mengetahui dimana letak manusia relatif terhadap robot.

4. Kondisi robot penerima tamu sudah berada dekat dengan manusia

Setelah dilakukan tracking dan penghitungan jarak, maka terdapat kondisi dimana jarak robot dengan manusia telah dekat atau dibawah nilai threshold jarak minimal. Pada kondisi ini, robot akan terus melakukan tracking sembari menghitung jarak secara visual namun tidak melakukan maneuver pada *wheel*-nya. Tujuan tracking bagian ini adalah agar jarak manusia dan robot terus bertahan di range tertentu. Pada saat ini juga robot penerima tamu akan melakukan gerakan-gerakan layaknya robot penerima tamu pada umumnya misalnya bersalaman, *say hello*, dan lain sebagainya.

BAB 5

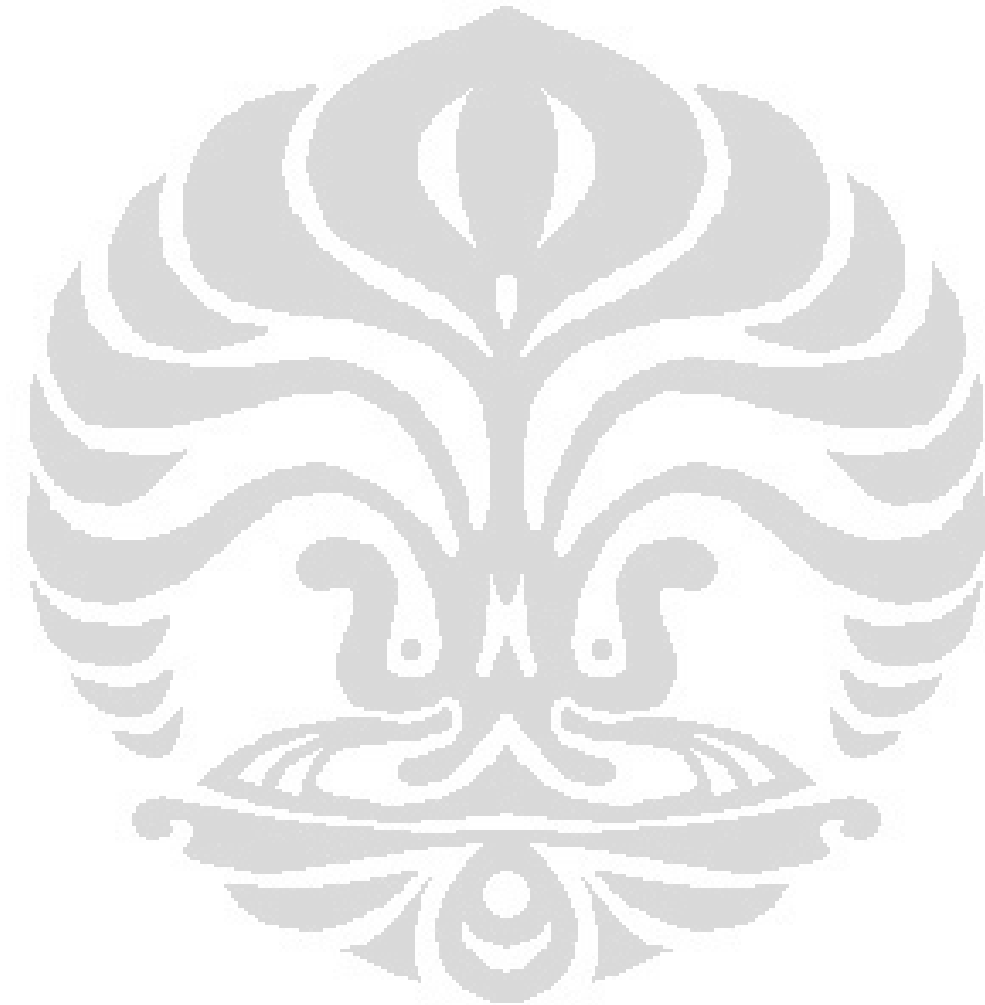
KESIMPULAN

Berdasarkan pembahasan tentang hasil eksperimen dan analisisnya, maka dapat ditarik sejumlah kesimpulan, yaitu sebagai berikut:

- a. Sistem minimum *stereo vision* yang dirancang sesuai dengan tema telah berhasil dilakukan namun pada skripsi kali ini lebih ditekankan pada efisiensi kode programnya.
- b. Proses kalibrasi lebih baik dilakukan secara offline karena dapat mengurangi beban komputasi.
- c. *Image file* PPM yang digunakan pada proses kalibrasi lebih baik dibandingkan dengan JPG dengan selisih error sebesar 0.022.
- d. *Success rate* yang didapat dari proses *face detection* minimal telah mencapai 90 persen sehingga kelayakannya dapat dikatakan valid walaupun perlu dilakukan optimasi kembali.
- e. Percobaan menghitung jarak dengan *stereo vision* menghasilkan error yang cukup besar apabila jarak yang dihitung semakin besar. Kesalahan bisa terjadi karena data yang dikirim dari kamera belum stabil dan proses kalibrasi yang tidak terlalu baik.
- f. Waktu komputasi yang cukup besar terjadi karena kecepatan prosesor tidak optimal yaitu hanya 60 persen dari kemampuan maksimal. Hal ini terjadi karena kernel yang digunakan sekarang belum dapat di-*upgrade* untuk kecepatan prosesor yang lebih tinggi.
- g. Konsep aktuator vision dan desain output untuk interaktif robot telah diajukan pada bab pengujian dan analisis.

Melihat perkembangan dan tren pengaplikasian robot ke arah interaktif terhadap manusia, maka diperlukan suatu sistem yang mampu membuat robot yang dapat mengerti perintah dan setiap gerak – gerak manusia tersebut. oleh karenanya penggunaan kamera di sini sangatlah mutlak diperlukan, karena persepsi dan kedalaman dapat diukur secara efisien dan efektif dengan menggunakan sistem komputer visual. Suatu robot yang canggih harus bisa mengenali lingkungannya baik terhadap objek tidak bergerak dan objek yang

bergerak. Oleh karenanya, algoritma lokalisasi sangat penting bagi sebuah robot yang canggih. Apalagi untuk sebuah robot penerima tamu yang didesain untuk berinteraksi dengan manusia. Diharapkan dengan bantuan algoritma ini, robot dapat lebih menjadi *robust* dan akhirnya bisa memenuhi keinginan manusia yang dapat meringankan kerja manusia nantinya.



DAFTAR ACUAN

Muis, Abdul. 2007. Friendly Motion Control for Robot as Human Partner Beyond Distance.

Sutton, Dean and Green, Richard. Evaluation of Real Time Stereo Vision System Using Web Cameras. Canterbury, New Zealand.

Wan, Lee Chong and Sebastian, Patrick. (2009). Stereo Vision tracking System. IEEE International Conference on Future Computer and Communication. Perak, Malaysia.



DAFTAR REFERENSI

- Buss, S. R. (2009). *Introduction to Inverse Kinematics with Jacobian Transpose, Pseudoinverse, and Damped Least Square*. California: La Jolla.
- Bradsky, Gary and Kaehler, Adrian. (2008). *Learning OpenCV: Computer Vision with the OpenCV Library*. O'Reilly Media, Inc.
- Sutanto, Giovanni and Kanamori, Chisato. (2009). Red Ball Recognizer & Grabber Robot by using *Stereo vision* System for Three-Dimensional (3D) Object Position Sensor. Japan.
- Ben-Tzvi, Pinhas and Xu, Xin. An Embedded Feature-Based Stereo Vision System for Autonomous Mobile Robots. *IEEE Journal*, Washington, DC 20052
- Chen, Chincyang and F. Zheng, Yuan. (1995) . *Passive and Active Stereo Vision for Smooth Surface Detection of Deformed Plates*. *IEEE Journal*.
- K.S. Fu, Gonzalez, R.C., Lee, C.S.G. (1993). *Robotics: Control, Sensing, Vision, and Intelligence*, McGraw-Hill Education.
- Zou, Ling and Li, Yan. A Methode of Stereo Vision Matching Based on OpenCV. National Natural Science Foundation of China.
- Choong Wan, Lee, Sebastian, Patrick, and Vooi Voon, Yap. (2009). *Stereo Vision Tracking System*. International Conference in Future Computer and Communication.
- Ganapathy, Velappa and Oon-Ee, Ng. (2008). *Stereo Vision Based Robot Controller*. IEEE International Conference.
- Yagmour K, Jon Maters, Gilad ben Yossef, Philippe Gerum. (2008). *Building Embedded Linux Systems, 2nd Ed*. O'Reilly.
- <http://nobitakitoyo.blogspot.com/2010/12/robot-pelayan-sabar-dan-tak-doyan-tip.html>. Diakses pada tanggal 9 Juni 2012 Pukul 23.00 WIB