



**APLIKASI *TRACKING OBJECT* PADA SISTEM *WEB STREAMING*
DENGAN PROTOKOL TCP/IP SEBAGAI SISTEM NAVIGASI *MOBILE*
ROBOT BERBASIS MINI PC**

SKRIPSI

Oleh

**AWALUDDHIN CHOLIQ AZIS
NIM 101910201058**

**PROGRAM STUDI STRATA-1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2015**



**APLIKASI *TRACKING OBJECT* PADA SISTEM *WEB STREAMING*
DENGAN PROTOKOL TCP/IP SEBAGAI SISTEM NAVIGASI *MOBILE*
ROBOT BERBASIS MINI PC**

SKRIPSI

diajukan guna melengkapi skripsi dan memenuhi syarat-syarat
untuk menyelesaikan Program Studi Teknik Elektro (S1)
dan guna mencapai gelar Sarjana Teknik

Oleh

**AWALUDDHIN CHOLIQ AZIS
NIM 101910201058**

**PROGRAM STUDI STRATA-1 TEKNIK ELEKTRO
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS JEMBER
2015**

PERSEMBAHAN

Alhamdulillah, segala puji bagi Allah SWT, yang telah memberikan limpahan nikmat yang sangat luar biasa kepada penulis, dan tidak lupa juga sholawat dan salam selalu tercurahkan kepada Nabi besar Muhammad SAW yang telah membawa kita semua menuju peradaban manusia yang lebih baik. Dengan kerendahan hati, penulis mempersembahkan tugas akhir ini untuk:

1. Ibuk/Ibu Siti Dwi Ningsih dan Bapak/Bapak Sukadi yang selalu mendoakan dan selalu mendukung baik secara moral dan materi.
2. Keluarga besar penulis yang telah memberikan doa serta menjadi motivasi penulis untuk sukses.
3. Semua Dosen Jurusan Teknik Elektro Fakultas Teknik Universitas Jember yang telah membimbing dan memberikan ilmu. Terutama Bapak Widya Cahyadi, S.T., M.T. selaku DPU dan Bapak Catur Suko Sarwono, S.T. selaku DPA yang telah meluangkan waktu dan pikirannya serta perhatiannya guna memberikan bimbingan dan pengarahan demi terselesainya skripsi ini.
4. Almamater Jurusan Teknik Elektro Universitas Jember
5. Pihak-pihak yang membaca serta menjadikan skripsi ini sebagai referensi penelitiannya.
6. Serta pihak-pihak lain yang telah membantu penulis untuk menyelesaikan skripsi ini yang tidak bisa penulis sebutkan satu persatu.

MOTTO

“Do not judge me by my successes, judge me by how many times I fell down and got back up again”

(Nelson Mandela)

“Jenius adalah 1 % inspirasi dan 99 % keringat. Tidak ada yang dapat menggantikan kerja keras”

(Thomas Alva Edison)

“Cobalah untuk tidak menjadi orang yang berhasil. Tetapi jadilah orang yang berguna”

(Albert Einstein)

“Barang siapa yang menghabiskan waktu berjam-jam lamanya untuk mengumpulkan harta karena takut miskin, maka dialah sebenarnya orang yang miskin”

(Imam Al Ghazali)

“Ketika seseorang menghina kamu, itu adalah sebuah pujian bahwa selama ini mereka menghabiskan banyak waktu untuk memikirkan kamu, bahkan ketika kamu tidak memikirkan mereka”

(BJ Habibie)

“Barang siapa yang tidak mau duduk dengan orang beruntung, bagaimana mungkin ia akan beruntung dan barang siapa yang duduk dengan orang beruntung bagaimana mungkin ia tidak akan beruntung”

(Habib Umar Bin Hafidz)

PERNYATAAN

Saya yang bertanda tangan dibawah ini :

Nama : Awaluddhin Choliq Azis

NIM : 101910201058

Menyatakan dengan sesungguhnya bahwa skripsi yang berjudul “Aplikasi *Tracking Object* Pada Sistem *Web Streaming* Dengan Protokol TCP/IP Sebagai Sistem Navigasi *Mobile Robot* Berbasis Mini PC” adalah benar – benar hasil karya sendiri, kecuali jika dalam pengutipan subtransi disebutkan sumbernya dan belum pernah diajukan pada institusi manapun, serta bukan karya jiplakan. Saya bertanggung jawab atas keabsahan dan kebenaran isinya sesuai dengan sikap ilmiah yang harus dijunjung tinggi.

Demikian pernyataan ini saya buat dengan sebenarnya, tanpa adanya tekanan dan paksaan dari pihak manapun serta bersedia mendapat sanksi akademik jika ternyata dikemudian hari pernyataan ini tidak benar.

Jember, 1 April 2015

Yang menyatakan,

Awaluddhin Choliq Azis

NIM 101910201058

SKRIPSI

**APLIKASI *TRACKING OBJECT* PADA SISTEM *WEB STREAMING*
DENGAN PROTOKOL TCP/IP SEBAGAI SISTEM NAVIGASI *MOBILE*
ROBOT BERBASIS MINI PC**

Oleh

Awaluddhin Choliq Azis

NIM 101910201058

Pembimbing :

Dosen Pembimbing Utama : Widya Cahyadi, S.T., M.T.

Dosen Pembimbing Anggota : Catur Suko Sarwono, S.T.

PENGESAHAN

Skripsi dengan judul “Aplikasi *Tracking Object* Pada Sistem *Web Streaming* Dengan Protokol TCP/IP Sebagai Sistem Navigasi *Mobile Robot* Berbasis Mini PC “ telah diuji dan disahkan oleh Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember pada :

hari,tanggal : Rabu, 1 April 2015

tempat : Fakultas Teknik, Universitas Jember.

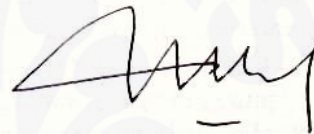
Tim Penguji

Ketua,



Widya Cahyadi, S.T., M.T.
NIP 19851110 201404 1 001

Sekretaris,



Catur Suko Sarwono, S.T.
NIP 19680119 199702 1 001

Anggota I,



Satryo Budi Utomo, S.T.,M.T.
NIP 19850126 200801 1 002

Anggota II,



Dodi Setiabudi, S.T.,M.T.
NIP 19840531 200812 1 004

Mengesahkan,
Dekan,



Ir. Widyono Hadi, M.T.
NIP. 19610414 198902 1 001

**APLIKASI *TRACKING OBJECT* PADA SISTEM *WEB STREAMING*
DENGAN PROTOKOL TCP/IP SEBAGAI SISTEM NAVIGASI *MOBILE*
ROBOT BERBASIS MINI PC**

Awaluddhin Choliq Azis

Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember

ABSTRAK

Perkembangan dunia robotika saat ini telah menjadi suatu hal yang menarik untuk dibicarakan. Penelitian dalam hal sistem navigasi robot juga sudah banyak dilakukan. Sistem navigasi robot merupakan sebuah sistem yang digunakan untuk mengendalikan pergerakan robot. *Image Processing* atau sering disebut dengan pengolahan citra digital merupakan metode yang digunakan untuk mengolah atau memproses dari gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. *Video web streaming* merupakan salah satu layanan video berbasis *web* yang sekarang ini semakin digemari oleh user. Layanan *video web streaming* ini banyak dimanfaatkan sebagai sarana hiburan, telekomunikasi, pembelajaran dan pemantauan (*monitoring*) keadaan suatu tempat. Tujuan penelitian ini adalah menggabungkan teknologi *image processing* dengan robotika. Merancang sebuah navigasi robot berdasarkan pergerakan bola. Sebagai input sistem digunakan *webcam* untuk input video (*real time*) kemudian ditampilkan pada halaman website. Deteksi bola dengan menggunakan metode segmentasi warna. Secara keseluruhan sistem dapat berjalan dengan baik sesuai algoritma yang dirancang. Dengan algoritma yang dirancang, protokol TCP pada layer transport berhasil mengirimkan data. *Delay* pengiriman antar paket dengan menggunakan protokol TCP/IP sebesar 0.026 s.

Kata kunci: *Navigasi Mobile Robot, Webcam, Segmentasi Warna, TCP/IP, Raspberry Pi.*

***THE APPLICATION OF TRACKING OBJECT ON WEB STREAMING
SYSTEM WITH TCP/IP PROTOCOL AS MOBILE ROBOT NAVIGATION
SYSTEM IN MINI PC BASE***

Awaludhin Choliq Azis

Department of Electrical Engineering, Engineering Faculty, University of Jember

ABSTRACT

The development of robotica world has been an interesting topic today. The Research of Robot navigation system has been conducted by many people. Robot navigation system is a system which is used for controlling robot movement. Image processing also well known as digital image processing is a methode that is used for managing or processing from the real picture becomes another picture suited with the needed. Video web streaming is one of web based video service which is today highly used by the user. This video web streaming service is very usefull for entertainment, telecommunication, learning activity and monitoring a place situation. The purpose of this research is to combine processing image technology with robotica. Designing a robot navigation based on the ball movement. The webcam is used as the system input to input the video (real time) then it is displayed in the website page. Ball detection by using colour segmentation method. Overall the system works suited with the plan algorhtym. By the plan algorithm , the TCP Protocol in the transport layer can deliver the data. The delay of inter package transfer with TCP / IP protocol is 0,026 s.

Keywords: mobile robot navigation, webcam, colour segmentation, tcp/ip, raspberry pi.

RINGKASAN

Aplikasi *Tracking Object* Pada Sistem *Web Streaming* Dengan Protokol TCP/IP Sebagai Sistem Navigasi *Mobile Robot* Berbasis Mini PC; Awaluddin Choliq Azis; 101910201058; 2015; 44 halaman; Program Studi Strata Satu Teknik, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember.

Perkembangan dunia robotika saat ini telah menjadi suatu hal yang menarik untuk dibicarakan. Penelitian dalam hal sistem navigasi robot juga sudah banyak dilakukan. Sistem navigasi robot merupakan sebuah sistem yang digunakan untuk mengendalikan pergerakan robot. Hasil dari keputusan sebuah navigasi robot diantaranya maju, mundur, belok kanan, belok kiri maupun berhenti.

Image Processing atau sering disebut dengan pengolahan citra digital merupakan metode yang digunakan untuk mengolah atau memproses dari gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Pada awalnya pengolahan citra (*image processing*) dilakukan untuk memperbaiki kualitas citra, namun seiring berkembangnya dunia komputasi yang memungkinkan manusia mengambil informasi dari suatu citra.

Halaman *web* merupakan salah satu pemanfaatan teknologi dalam bidang jaringan komputer. Halaman *web* memudahkan kita untuk mengakses suatu informasi yang ada dalam jaringan komputer tersebut. Dengan bantuan *search engine*, pencarian informasi didalam jaringan komputer semakin mudah. Kita dapat mencari dan mengakses suatu informasi dari mana pun kita berada dengan bantuan *search engine* dan *web browser*.

Video web streaming merupakan salah satu layanan video berbasis *web* yang sekarang ini semakin digemari oleh user. Layanan *video web streaming* ini banyak dimanfaatkan sebagai sarana hiburan, telekomunikasi, pembelajaran dan pemantauan (*monitoring*) keadaan suatu tempat. Contoh sarana hiburan dalam penerapan *video streaming* pada situs you tube. Sedangkan dalam bidang telekomunikasi contohnya adalah *video call* dan *video conference*.

Tujuan penelitian ini adalah menggabungkan teknologi *image processing* dengan robotika. Merancang sebuah navigasi robot berdasarkan pergerakan bola. Sebagai input sistem digunakan *webcam* untuk input video (*real time*) kemudian ditampilkan pada halaman website. Deteksi bola dengan menggunakan metode segmentasi warna. Secara keseluruhan sistem dapat berjalan dengan baik sesuai algoritma yang dirancang. Dengan algoritma yang dirancang, protokol TCP pada layer transport berhasil mengirimkan data. *Delay* pengiriman antar paket dengan menggunakan protokol TCP/IP sebesar 0.026 s.

PRAKATA

Bismillahirrohmanirrohim

Puji syukur kehadiran Allah SWT atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “*aplikasi tracking object pada sistem web streaming dengan protokol tcp/ip sebagai sistem navigasi mobile robot berbasis mini pc*”. Skripsi ini disusun untuk memenuhi salah satu syarat menyelesaikan pendidikan strata satu (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Penyusunan skripsi ini tidak lepas dari bantuan beberapa pihak. Oleh karena itu, penulis menyampaikan terima kasih kepada :

1. Allah SWT yang telah melimpahkan rahmat dan rizki-Nya serta memberi kelancaran dan kemudahan sehingga terseleassikannya skripsi ini.
2. Bapak Ir. Widyono Hadi, M.T., selaku Dekan Fakultas Teknik Universitas Jember yang memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
3. Bapak Widya Cahyadi, S.T.,M.T., selaku Dosen Pembimbing Utama dan Bapak Catur Suko Sarwono, S.T., selaku Dosen Pembimbing Anggota yang telah meluangkan waktu, pikiran guna memberikan bimbingan dan pengarahan dalam penyusunan skripsi ini.
4. Bapak Satryo Budi Utomo, S.T., M.T. dan Dodi Setiabudi, S.T., M.T. selaku Tim Penguji yang telah meluangkan banyak waktu, pikiran dan perhatiannya guna memberikan pengarahan demi terselesaikannya penulisan skripsi ini.
5. Seluruh Dosen yang ada di Fakultas Teknik khususnya Teknik Elektro beserta karyawan.
6. Ibuk/Siti Dwi Ningsih, dan Bapak/Sukadi, serta adik saya Hanifah Banu Rohmi. Terima kasih atas semua dukungan baik secara materi maupun moral dan kasih sayang serta doa restunya yang tulus kepada saya. Kalian

semua tetap sabar dan selalu menyempatkan do'a untuk saya disetiap shalat, hingga terselesaikannya skripsi ini.

7. Teman-teman atau bisa dikatakan keluarga/dulur “Angkatan 18 / LOWO / 2000X” Azhu, Oyek, Parto, Mbah Mujib, Gerbong, jembot, Bryan, Resan, jessica. Dengan kalian senang, susah, canda tawa apapun keadaan di jember kita selalu bersama, semoga kebersamaan dan kekompakan kita tetep terjalin selamanya. Terima kasih atas dukungan dan bantuannya selama proses perkuliahan hingga penyusunan skripsi ini.
8. *My special one*, A'thi Fauzani Wisudawati yang selalu memberi motivasi dan semangat selama ini.
9. Teman-teman PATEK UJ 2010, yang dengan bangga mengusung slogan “Sing Penting Wani Disek !” Bangga menjadi bagian dari kalian. #SPWD.
10. Teman-teman Riki, Angga, Bono dan Budi, terima kasih telah membantu dalam penyelesaian skripsi ini.
11. Pihak-pihak yang tidak dapat saya sebutkan satu persatu, terima kasih atas dukungan dan motivasi kalian dalam penyusunan skripsi ini.

Semoga skripsi ini dapat bermanfaat dalam mengembangkan ilmu pengetahuan khususnya untuk disiplin ilmu teknik elektro. Kritik dan saran yang membangun diharapkan terus mengalir untuk lebih menyempurnakan skripsi ini dan dapat dikembangkan untuk penelitian selanjutnya.

Jember, 1 April 2015

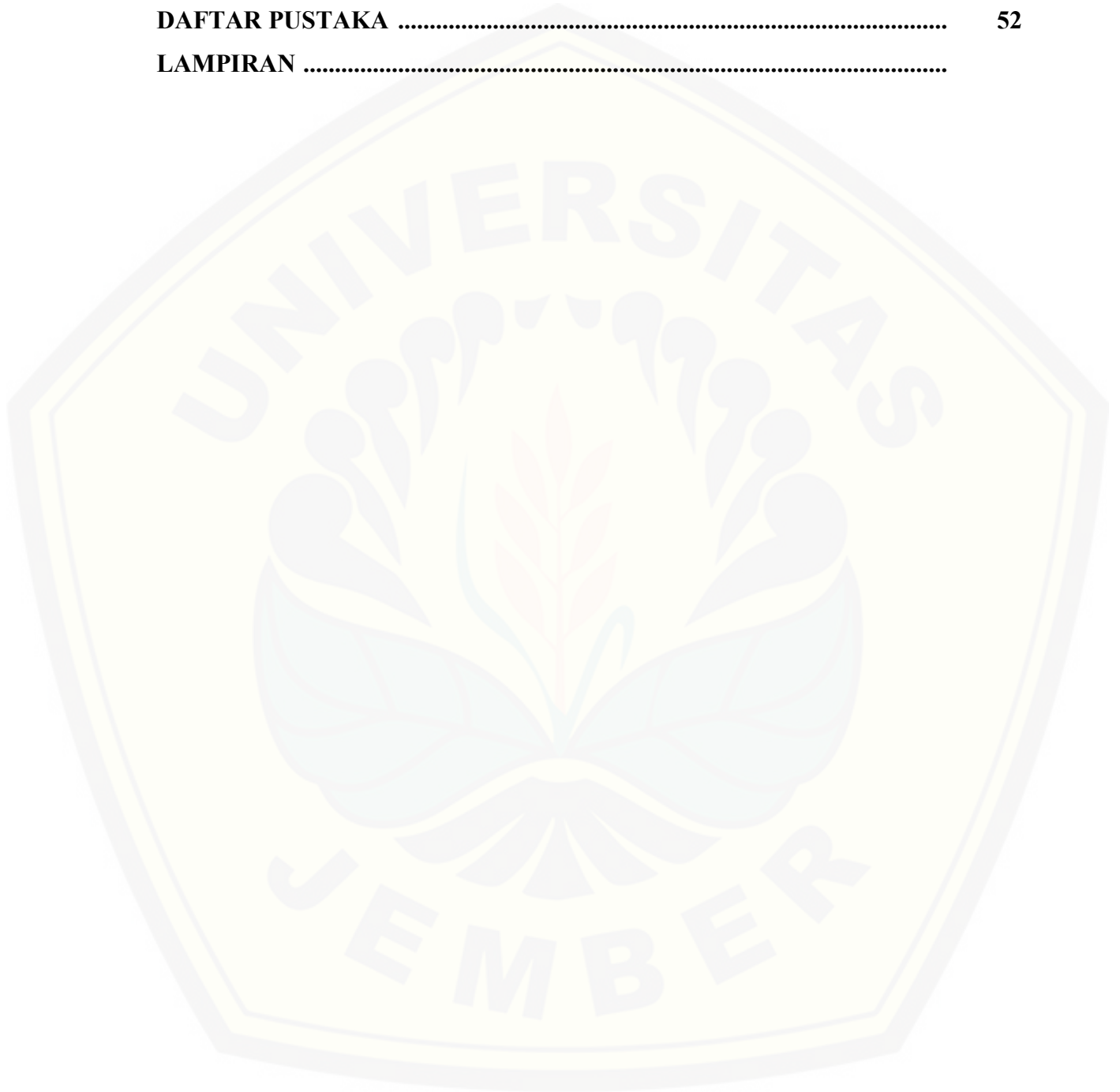
Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PERSEMBAHAN	ii
HALAMAN MOTTO	iii
HALAMAN PERNYATAAN	iv
HALAMAN PEMBIMBING	v
HALAMAN PENGESAHAN	vi
ABSTRAK	vii
RINGKASAN	ix
PRAKATA	x
DAFTAR ISI	xii
DAFTAR TABEL	xv
DAFTAR GAMBAR	xvi
BAB 1 PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah	3
1.4 Tujuan	3
1.5 Manfaat	3
1.6 Sistematika Penulisan	4
BAB 2 TINJAUAN PUSTAKA	5
2.1 Pengertian Citra Digital	5
2.1.1 Pengolahan Citra Warna	5
2.1.2 Citra HSV (<i>Hue, Saturation, Value</i>)	8
2.2 Jaringan Komputer	8
2.2.1 TCP/IP (<i>Transfer Control Protokol/Internet Protokol</i>).....	9

2.2.2	Arsitektur Protokol TCP/IP	10
2.2.3	Sistem <i>Wireless Local Area Network</i>	11
2.2.4	Standarisasi <i>Wireless Local Area Network</i>	12
2.3	Protokol <i>Streaming</i> HTTP	13
2.4	MJPEG	13
2.5	Raspberry Pi	14
BAB 3	METODOLOGI PENELITIAN	16
3.1	Jadwal Dan Tempat Penelitian	16
3.2	Tahap Perencanaan	16
3.3	Alat dan Bahan	17
3.3.1	<i>Hardware</i>	17
3.3.2	<i>Software</i>	17
3.4	Gambaran Umum Sistem	17
3.5	Blok Diagram Alat	18
3.5.1	Blok <i>Mobile Robot</i>	18
3.5.2	Blok Server	19
3.6	<i>Flowchart Image Processing</i>	20
3.7	<i>Flowchart</i> Menampilkan Gambar Pada <i>Website</i>	21
3.8	<i>Flowchart</i> Mengirim Nilai Koordinat Bola	22
BAB 4	HASIL DAN PEMBAHASAN	23
4.1	Pengujian Program Server	23
4.1.1	Pengujian Program Deteksi Objek	23
4.1.2	Program Pengiriman Nilai Koordinat Objek	30
4.2	Program <i>Mobile Robot</i>	31
4.2.1	Program Penerima Nilai Koordinat Objek	31
4.2.2	Program Navigasi <i>Mobile Robot</i>	31
4.3	Pengujian Keberhasilan Koneksi TCP/IP	33
4.4	Pengujian Method GET dan POST Pada Protokol HTTP	38
4.5	Pengujian Sistem Secara Keseluruhan	42
4.6	Pengujian <i>Delay</i> Waktu Kerja Sistem	48

BAB 5 PENUTUP	51
5.1 Kesimpulan.....	51
5.2 Saran	51
DAFTAR PUSTAKA	52
LAMPIRAN	



DAFTAR TABEL

	Halaman
Tabel 4.1 Hasil <i>Capture</i> Paket TCP dengan <i>Tcpdump</i>	35
Tabel 4.2 Pengujian Pengiriman Nilai Koordinat Bola Dengan Protokol TCP/IP	38
Tabel 4.3 Pengujian Pendeteksian Objek Bola Terhadap Aksi <i>Mobile</i> <i>Robot</i>	44
Tabel 4.4 Hasil Pengujian Didalam Ruangan Dengan Intensitas Cahaya 19,8 lumen	46
Tabel 4.5 Hasil Pengujian Didalam Ruangan Dengan Intensitas Cahaya 47,1 lumen	47
Tabel 4.6 Hasil Pengujian Diluar Ruangan Dengan Intensitas Cahaya 1470 lumen	47
Tabel 4.7 Hasil Pengujian Diluar Ruangan Dengan Intensitas Cahaya 21500 lumen	48
Tabel 4.8 Hasil Pengujian Respon Server Dalam Melayani Method GET	49

DAFTAR GAMBAR

	Halaman
Gambar 2.1 Nilai Warna RGB dalam Hexadesimal	6
Gambar 2.2 Komposisi Warna RGB	6
Gambar 2.3 TCP/IP <i>Layer</i>	10
Gambar 2.4 Penyisipan <i>Header</i> Pada Tiap <i>Layer</i>	10
Gambar 2.5 <i>Standart</i> IEEE 802.11	13
Gambar 3.1 Gambaran Umum Sistem	18
Gambar 3.2 Diagram Blok Alat	22
Gambar 3.3 <i>Flowchart Image Processing</i>	21
Gambar 3.4 <i>Flowchart</i> Menampilkan Gambar Pada <i>Website</i>	22
Gambar 3.5 <i>Flowchart</i> Mengirimkan Nilai Koordinat Bola	25
Gambar 4.1 Citra RGB Dan Citra HSV	24
Gambar 4.2 Nilai Citra HSV Bola	25
Gambar 4.3 Segmentasi Warna	27
Gambar 4.4 Membagi Frame Menjadi 9 Area	27
Gambar 4.5 Tampilan Gambar Pada <i>Website</i> Lokal	26
Gambar 4.6 Ketika Gagal dan Berhasil Koneksi	30
Gambar 4.7 <i>Capture</i> Paket TCP Dengan <i>Tcpdump</i>	34
Gambar 4.8 Tampilan Terminal Sisi Server	39
Gambar 4.9 Kualitas Gambar <i>Low</i>	40
Gambar 4.10 Kualitas Gambar <i>Medium</i>	40
Gambar 4.11 Kualitas Gambar <i>High</i>	41
Gambar 4.12 Method POST dan GET Terminal Server	42
Gambar 4.13 <i>Mobile Robot</i>	43

BAB 1. PENDAHULUAN

Pada bab ini berisi tentang latar belakang penulis melakukan penelitian. Latar belakang penulis mengambil judul penelitian. Bab ini berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika pembahasan.

1.1 Latar Belakang

Perkembangan dunia robotika saat ini telah menjadi suatu hal yang menarik untuk dibicarakan. Penelitian dalam hal sistem navigasi robot juga sudah banyak dilakukan. Sistem navigasi robot merupakan sebuah sistem yang digunakan untuk mengendalikan pergerakan robot. Hasil dari keputusan sebuah navigasi robot diantaranya maju, mundur, belok kanan, belok kiri maupun berhenti. Untuk jenis robot otomatis, pergerakannya berdasarkan kondisi masukan sensor-sensor yang digunakan. Selain robot otomatis, ada juga robot yang navigasinya berdasarkan perintah-perintah yang dikirimkan secara manual melalui PC, *remote control* atau *joystick* yang dinamakan *teleoperated*. (Reda Anggra Distira: 2012).

Image Processing atau sering disebut dengan pengolahan citra digital merupakan metode yang digunakan untuk mengolah atau memproses gambar asli sehingga menghasilkan gambar lain yang sesuai dengan kebutuhan. Penggunaan *image processing* ini sudah cukup berkembang sejak orang mengerti bahwa komputer tidak hanya mampu menangani data teks, melainkan juga data citra. Pada awalnya pengolahan citra (*image processing*) dilakukan untuk memperbaiki kualitas citra, namun seiring berkembangnya dunia komputasi yang memungkinkan manusia mengambil informasi dari suatu citra, maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision*. Dalam perkembangan lebih lanjut, *image processing* dan *computer vision* digunakan sebagai pengganti mata manusia, dengan perangkat *input image capture* seperti kamera dan *scanner* dijadikan sebagai mata dan mesin komputer (dengan program

komputasinya) digunakan sebagai otak yang mengolah informasi. (Reda Anggra Distira: 2012).

Halaman *web* merupakan salah satu pemanfaatan teknologi dalam bidang jaringan komputer. Halaman *web* memudahkan kita untuk untuk mengakses suatu informasi yang ada dalam jaringan komputer tersebut. Dengan bantuan *search engine* pencarian informasi didalam jaringan komputer semakin mudah. Kita dapat mencari dan mengakses suatu informasi dari mana pun kita berada dengan bantuan *search engine* dan *web browser*.

Video web streaming merupakan salah satu layanan video berbasis *web* yang sekarang ini semakin digemari oleh user. Layanan *video web streaming* ini banyak dimanfaatkan sebagai sarana hiburan, telekomunikasi, pembelajaran dan pemantauan (*monitoring*) keadaan suatu tempat. Contoh sarana hiburan dalam penerapan *video streaming* pada situs you tube. Sedangkan dalam bidang telekomunikasi contohnya adalah *video call* dan *video conference*.

Dari kedua bidang tersebut akhirnya muncul teknologi baru yakni *robot vision*. *Robot Vision* adalah robot yang menggunakan teknologi *image processing* dan *computer vision* dalam tugasnya, baik untuk pengenalan objek maupun navigasi robot. Berdasarkan penjelasan diatas, penulis merancang dan membuat aplikasi *tracking object* pada *web streaming* untuk navigasi *mobile robot* dengan menggunakan *raspberry pi*.

1.2 Rumusan Masalah

Berdasarkan latar belakang diatas maka dapat dirumuskan masalah sebagai berikut :

- a. Bagaimana merancang sistem navigasi *mobile robot* berbasis *web streaming* dengan menggunakan protokol TCP/IP ?
- b. Bagaimana mendeteksi objek yang berada pada *video web streaming* berbasis *image processing* dengan menggunakan metode segmentasi warna?

- c. Bagaimana merancang komunikasi dan koneksi antara server web streaming dengan *mobile robot* secara *wireless* menggunakan perangkat WLAN dengan standart 802.11?

1.3 Batasan Masalah

Agar dalam penulisan skripsi ini dapat mencapai sasaran dan tujuan yang diharapkan, maka dalam pembahasan penelitian ini membatasi pembahasan menjadi beberapa permasalahan berikut :

- a. Sistem program pengendalian piranti menggunakan *Raspberry Pi* tipe B+ dan pemrogramnya menggunakan *python* untuk pengolahan citra.
- b. Metode pendeteksian pergerakan objek menggunakan metode segmentasi warna.
- c. Objek yang digunakan adalah sebuah bola dengan warna merah.
- d. Hanya terdapat 1 objek dengan warna merah.

1.4 Tujuan

Adapun tujuan yang akan diperoleh dari penelitian ini adalah sebagai berikut :

- a. Membuat sistem *tracking object* pada *web video streaming* sebagai sistem navigasi *mobile robot* dengan menggunakan protokol TCP/IP.
- b. Merancang algoritma pendeteksian objek yang berbasis *image processing* dengan metode segmantasi warna.
- c. Dapat mengaplikasikan protokol HTTP untuk *video web streaming* sebagai sistem navigasi *mobile robot*.

1.5 Manfaat

Adapun manfaat yang diperoleh dari penulisan skripsi ini adalah sebagai berikut:

- a. Dengan sistem yang digunakan, dapat diterapkan sebagai sistem navigasi *mobile robot* berbasis *video web steaming*.
- b. Dengan metode yang digunakan pada *image processing* dapat diimpementasikan sebagai proses dalam sistem navigasi *mobile robot*.

- c. Skripsi ini dapat dijadikan sebagai referensi tentang pengendalian *mobile robot* menggunakan sistem *web streaming* dengan protokol HTTP.

1.6 Sistematika Penulisan

Secara garis besar penyusunan proposal skripsi ini adalah sebagai berikut:

BAB 1 PENDAHULUAN

Berisi tentang latar belakang, perumusan masalah, batasan masalah, tujuan, manfaat, dan sistematika pembahasan.

BAB 2 TINJAUAN PUSTAKA

Berisi penjelasan tentang teori yang berhubungan dengan penelitian.

BAB 3 METODOLOGI PENELITIAN

Menjelaskan tentang metode yang digunakan untuk menyelesaikan skripsi.

BAB 4 HASIL DAN PEMBAHASAN

Berisi hasil penelitian dan analisa hasil penelitian.

BAB 5 PENUTUP

Berisi tentang kesimpulan dan saran dari penulis.

BAB 2. TINJAUAN PUSTAKA

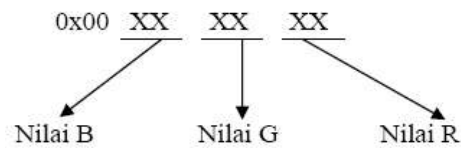
Bab ini berisi penjelasan tentang teori yang berhubungan dengan penelitian. Yaitu pada penelitian ini menggunakan teori tentang *image processing*, jaringan komputer, *standarisasi jaringan wireless local area network*, format video mjpeg dan penjelasan tentang mini komputer raspberry pi.

2.1 Pengertian Citra Digital

Citra dapat berbentuk foto hitam putih atau berwarna, sinyal-sinyal video seperti gambar pada monitor televisi, atau bersifat digital yang dapat langsung disimpan pada suatu pita magnetik. Menurut presisi yang digunakan untuk menyatakan titik-titik koordinat pada ranah waktu atau bidang dan untuk menyatakan nilai keabuan atau warna suatu citra, maka secara teoritis citra dapat dikelompokkan menjadi empat kelas citra, yaitu cara kontinu-kontinu, kontinu-diskret, diskret-kontinu, dan diskret-diskret; dengan label pertama menyatakan presisi dari titik-titik koordinat pada bidang citra sedangkan label kedua menyatakan presisi nilai keabuan atau warna. Kontinu dinyatakan dengan presisi takhingga, sedangkan diskret dinyatakan dengan presisi angka berhingga. Perubahan citra yang bersifat kontinu menjadi citra yang bersifat diskret memerlukan pembuatan kisi-kisi arah vertikal dan horisontal, sehingga diperoleh citra dalam bentuk larik dua dimensi. Proses tersebut dikenal sebagai proses digitisasi atau pencuplikan (*sampling*). Setiap elemen larik tersebut dikenal sebagai elemen gambar atau piksel. 7

2.1.1 Pengolahan Citra Warna

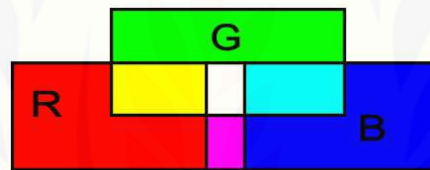
Dasar dari pengolahan citra adalah pengolahan warna RGB pada posisi tertentu. Dalam pengolahan citra warna dipresentasikan dengan nilai hexadesimal dari 0x00000000 sampai 0x00ffffff. Warna hitam adalah 0x00000000 dan warna putih adalah 0x00ffffff. Definisi nilai warna di atas seperti gambar 2.1, variabel 0x00 menyatakan angka dibelakangnya adalah hexadecimal.



Gambar 2.1 Nilai warna RGB dalam hexadesimal

(Sumber : Reda Anggra Distira, 2012)

Gambar 2.1 Nilai warna RGB dalam hexadesimal Terlihat bahwa setiap warna mempunyai range nilai 00 (angka desimalnya adalah 0) dan ff (angka desimalnya adalah 255), atau mempunyai nilai derajat keabuan $256 = 2^8$. Dengan demikian range warna yang digunakan adalah $(2^8)(2^8)(2^8) = 2^{24}$ (atau yang dikenal dengan istilah True Colour pada Windows). Nilai warna yang digunakan di atas merupakan gabungan warna cahaya merah, hijau dan biru seperti yang terlihat pada gambar 2.2. Sehingga untuk menentukan nilai dari suatu warna yang bukan warna dasar digunakan gabungan skala kecerahan dari setiap warnanya.



Gambar 2.2 Komposisi warna RGB

(Sumber : Reda Anggra Distira, 2012)

Dari definisi diatas untuk menyajikan warna tertentu dapat dengan mudah dilakukan, yaitu dengan mencampurkan ketiga warna dasar RGB.

Pengolahan citra dimulai dengan proses *thresholding*, yaitu proses pemisahan citra berdasarkan batas nilai tertentu, dalam proses *thresholding* citra warna diubah menjadi citra biner. Tujuan proses *thresholding* adalah untuk membedakan objek dengan latar belakangnya. Setelah proses *thresholding* proses selanjutnya adalah proses penghitungan nilai-nilai parameter antara lain R, G, B, RGB rata-rata (*color value*), dan indeks R (Ired), indeks G (Igreen), indeks B (Iblue), dari tiap-tiap *pixel*.

a. Pengukuran Parameter RGB (*Red*, *Green* dan *Blue*)

Parameter RGB diperoleh dari tiap-tiap *pixel* warna pada citra yang merupakan nilai intensitas untuk masing-masing warna merah, hijau, dan biru. Nilai rata-rata dari R,G dan B dijumlahkan untuk mendapatkan *color value* atau RGB rata-rata.

b. Pengukuran parameter Indeks R, Indeks G dan Indeks B

Perhitungan indeks warna merah/indeksR (I_{Red}), indeks warna hijau/indeks G (I_{Green}), dan indeks warna biru/indeks B (I_{Blue}) menggunakan rumus pada persamaan (1), (2) dan (3). Intensitas warna merah dibagi dengan penjumlahan dari nilai intensitas warna merah, hijau, dan biru sehingga menghasilkan nilai parameter indeks R. Intensitas warna hijau dibagi dengan penjumlahan dari nilai intensitas warna merah, hijau, dan biru sehingga menghasilkan nilai parameter indeks G. Intensitas warna biru dibagi dengan penjumlahan dari nilai intensitas warna merah, hijau, dan biru sehingga menghasilkan nilai parameter indeks B. Perhitungan parameter Indeks R, G, dan B diperoleh dari tiap-tiap *pixel* pada citra kopi. Model warna RGB dapat juga dinyatakan dalam bentuk indeks warna RGB dengan rumus sebagai berikut (Ahmad, 2005; Arimurthy, dkk., 1992):

Indeks warna merah:

$$(I_{Red}) = \frac{R}{R+G+B} \dots\dots\dots(2.1)$$

Indeks warna biru:

$$(I_{Blue}) = \frac{B}{R+G+B} \dots\dots\dots(2.2)$$

Indeks warna hijau:

$$(I_{Green}) = \frac{G}{R+G+B} \dots\dots\dots(2.3)$$

Dengan R, G dan B masing-masing merupakan besaran yang menyatakan nilai intensitas warna merah, hijau, dan biru.

2.1.2 Citra *HSV*

Model warna HSV mendefinisikan warna dalam terminologi Hue, Saturation dan Value. Definisi untuk Hue dan Saturation sama dengan definisi pada mode HSL, sedangkan untuk value memiliki arti kecerahan dari warna yang ada variasi dengan warna saturation. Nilainya berkisar antara 0 sampai 100%. Apabila nilainya 0 maka warnanya akan menjadi hitam dan apabila nilainya dinaikkan maka kecerahan akan menaik dan akan muncul variasi-variasi baru dari warnatersebut. Model warna ini dibuat berdasarkan system warna Ostwald (1931).

Variasi dari roda HSV digunakan untuk memilih warna yang diinginkan . Hue diwakiloleh lingkaran/keliling dalam roda. Sumbu horizontal menunjukkan saturation dan sumbu vertikal menunjukkan value.

Untuk mengambil suatu warna tertentu kita perlu menentukan dahulu hue dan kemudian kita baru memilih nilai saturation dan untuk brightness kita bisa memilihnya dari nilai value.Keuntungan dari model warna HSV & HSL ini adalah terdapat warna-warna yang sama dengan warna yang biasanya ditangkap oleh indra manusia. Sedangkan warna-warna yang dibentuk pada model lainnya merupakan hasil campuran dari warna-warna primer/ dasar untuk membentuk warna lain.

2.2 Jaringan Komputer

Network atau jaringan, dalam bidang komputer dapat diartikan sebagai dua atau lebih komputer yang dihubungkan sehingga dapat berhubungan dan berkomunikasi, sehingga menimbulkan suatu efisiensi, sentralisasi dan optimasi kerja. Pada jaringan komputer yang dioptimalkan adalah data. Suatu komputer dapat berhubungan dengan komputer yang lain tanpa harus membawa disket/media penyimpanan lain seperti yang biasa dilakukan.

Ada beberapa jenis jaringan komputer dilihat dari cara pemrosesan data dan pengaksesannya.

1. *Host-Terminal*. Di mana terdapat sebuah atau lebih server yang dihubungkan dalam suatu *dumb terminal*. Karena *Dumb Terminal* hanyalah sebuah monitor yang dihubungkan dengan menggunakan kabel

RS232, maka pemrosesan data dilakukan dalam *server*. Maka dari itu suatu *server* haruslah sebuah sistem komputer yang memiliki kemampuan pemrosesan data dan penyimpanan yang sangat besar.

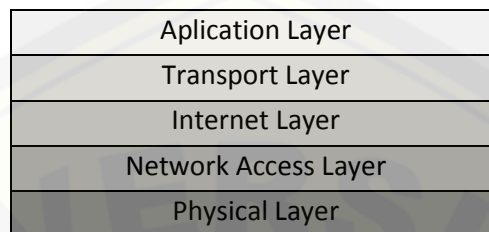
2. *Client-Server*. Di mana sebuah *server* atau lebih dihubungkan dengan beberapa *client*. *Server* bertugas menyediakan berbagai macam layanan misalnya pengaksesan *file*, *database*, dan sebagainya. *Client* adalah sebuah terminal yang menggunakan layanan tersebut. Perbedaannya dengan hubungan *dumb terminal*, sebuah terminal *client* melakukan pemrosesan data di terminalnya sendiri dan hal itu menyebabkan spesifikasi *server* tidaklah harus memiliki performa yang sangat tinggi, karena hampir semua pemrosesan data dilakukan di *client*.
3. *Peer to Peer*. Di mana terdapat beberapa terminal komputer yang dihubungkan dengan media kabel. Secara prinsip, hubungan *p2p* ini adalah bahwa setiap komputer dapat berfungsi sebagai *server* (penyedia layanan) dan *client*. Kedua fungsi tersebut dapat berjalan secara bersamaan.

2.2.1 TCP/IP (Transfer Control Protocol/Internet Protocol)

TCP/IP (*Transmission Control Protocol/Internet Protocol*) adalah sekelompok *protocol* yang mengatur komunikasi data komputer di internet. Komputer-komputer yang terhubung ke internet berkomunikasi dengan *protocol* ini. Karena menggunakan bahasa yang sama, yaitu *protocol* TCP/IP, perbedaan jenis komputer dan system operasi tidak menjadi masalah. Komputer PC dengan *system* operasi Windows dapat berkomunikasi dengan komputer Macintosh atau dengan Sun SPARC yang menjalankan Solaris. Jadi, jika sebuah komputer menggunakan *protocol* TCP/IP dan terhubung langsung ke Internet, maka komputer tersebut dapat berhubungan dengan komputer di belahan dunia manapun yang juga terhubung ke internet.

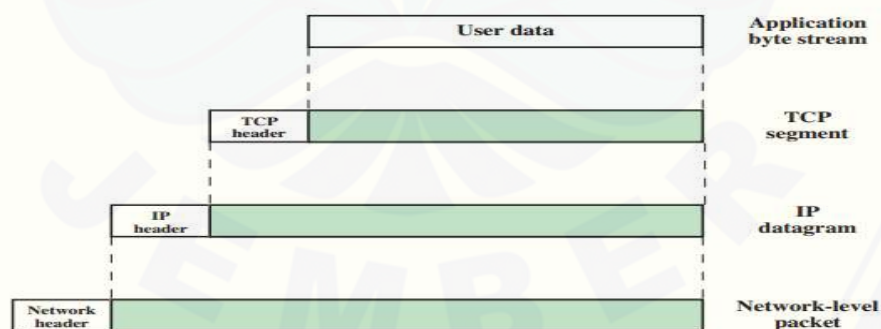
2.2.2 Arsitektur Protokol TCP/IP

Karena tidak ada perjanjian umum tentang bagaimana melukiskan TCP/IP dengan model layer, biasanya TCP/IP didefinisikan dalam 3-5 level fungsi dalam arsitektur protocol. Kali ini kita akan melukiskan TCP/IP dalam 4 layer model, yaitu seperti digambarkan dalam diagram di bawah ini :



Gambar 2.3 TCP/IP Layer

Jika suatu *protocol* menerima data dari *protocol* lain di layer atasnya, ia akan menambahkan informasi tambahan miliknya ke data tersebut, informasi ini memiliki fungsi yang sesuai dengan fungsi *protocol* tersebut. Setelah itu, data ini diteruskan lagi ke *protocol* pada layer di bawahnya. Hal yang sebaliknya terjadi jika suatu *protocol* menerima data dari *protocol* lain yang berada pada layer di bawahnya. Jika data ini dianggap valid, *protocol* akan melepas informasi tambahan tersebut untuk kemudian meneruskan data itu ke *protocol* lain yang berada pada layer di atasnya.



Gambar 2.4 Penyisipan Header Setiap Layer

(Sumber : William Stallings, 2010)

TCP mengatur masalah perintah-perintah pengiriman data, mengawasi jalannya data dan memastikan data tersebut sampai ke tujuannya. Apabila ada bagian dari data yang tidak mencapai tujuan maka TCP akan mengirimkan lagi. Proses tersebut akan terus diulang sampai data yang dikirim sampai ke tujuannya. Apabila ada data yang sangat besar untuk dimuat dalam sekali pengiriman maka TCP akan memecahnya menjadi beberapa paket dan kemudian mengirimkan data ke tujuan dan memastikan data itu sampai dengan benar. IP adalah protokol yang memuat semua kebutuhan aplikasi dalam berhubungan antar terminal. Seperti telah disampaikan sebelumnya bahwa TCP bertanggungjawab pada masalah pengiriman dan pemecahan data menjadi bagian-bagian kecil, maka IP merupakan pembuka jalan sehingga data dapat sampai ke terminal tujuan. Pelapisan-pelapisan protokol tersebut berguna untuk menjaga agar data dapat sampai dengan sempurna.

Beberapa layanan dasar tapi merupakan layanan yang penting diberikan oleh TCP/IP adalah :

1. *File Transfer Protocol* (FTP)
2. *Remote Login* (SSH/Telnet)
3. *E-Mail*
4. *Web*

2.2.3 Sistem *Wireless Local Area Network*

Wireless adalah teknologi tanpa kabel, dalam hal ini adalah melakukan hubungan telekomunikasi dengan menggunakan gelombang elektromagnetik sebagai pengganti kabel. Saat ini teknologi *wireless* berkembang dengan pesat, secara kasatmata dapat dilihat dengan semakin banyaknya pemakaian telepon selular, selain itu berkembang pula teknologi *wireless* yang digunakan untuk akses internet.

Wireless LAN menggunakan gelombang elektromagnetik (radio dan inframerah) untuk melakukan komunikasi data menyalurkan data dari satu point ke point yang lain tanpa melalui fasilitas fisik. Koneksi ini menggunakan frekuensi tertentu untuk menyalurkan data tersebut, kebanyakan *Wireless LAN*

menggunakan frekuensi 2,4 GHz. Frekuensi inilah yang disebut dengan Industrial, Scientific and Medical Band atau sering disebut ISMBand.

Teknologi *wireless*, memungkinkan satu atau lebih peralatan untuk berkomunikasi tanpa koneksi fisik, yaitu tanpa membutuhkan jaringan atau peralatan kabel. Teknologi *wireless* menggunakan transmisi frekuensi radio sebagai alat untuk mengirimkan data, sedangkan teknologi kabel menggunakan kabel. Teknologi *wireless* berkisar dari sistem kompleks seperti *Wireless Local Area Network* (WLAN) dan telepon selular hingga peralatan sederhana seperti *headphone wireless*, *microphone wireless* dan peralatan lain yang tidak memproses atau menyimpan informasi. Disini juga termasuk peralatan infra merah seperti *remote control*, *keyboard* dan *mouse komputer wireless*, dan *headset stereo hi-fi wireless*, semuanya membutuhkan garis pandang langsung antara transmitter dan receiver untuk membuat hubungan. (Janner Simamarta)

2.2.4 Standarisasi Wireless LAN

Setiap perangkat keras yang berhubungan dengan komputer dan teknologi selalu mengacu pada suatu standar. Begitu juga dengan *Wireless LAN*. Ada beberapa organisasi yang telah menetapkan *standard wireless LAN* diantaranya yaitu *Federal Communication Commission (FCC)*, *Institute of Electrical and Electronic Engineers (IEEE)*, *Wireless Ethernet Compatibility Alliance (WECA)*, dan *European Telecommunications Standards Institute (ETSI)*. Namun standar *wireless* yang paling sering digunakan adalah standar dari IEEE. IEEE merupakan organisasi pembuat standarisasi untuk hampir semua hal yang berhubungan dengan teknologi informasi di Amerika Serikat. IEEE menciptakan standar dengan aturan yang dibuat FCC. IEEE telah menetapkan berbagai standar teknologi seperti IEEE 802.3 untuk ethernet dan 802.11 untuk Wireless LAN. Salah satu tugas IEEE adalah mengembangkan standar untuk Wireless LAN dengan mengacu pada peraturan yang dikeluarkan FCC. Spesifikasi yang digunakan dalam WLAN adalah 802.11 dari IEEE dimana ini juga sering disebut dengan WiFi (Wireless Fidelity) standar yang berhubungan dengan kecepatan akses data. Berikut ini adalah standart IEEE yang pernah dikeluarkan untuk WirelessLAN:

	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g	IEEE 802.11n
<i>Frequency band</i>	5.7 GHz	2.4 GHz	2.4 GHz	2.4 / 5 GHz
<i>Average Theoretical speed</i>	54 Mbps	11 Mbps	54 Mbps	600 Mbps
<i>Modulation</i>	OFDM	CCK modulated with QPSK	DSSS, CCK, OFDM	OFDM
<i>Channel bandwidth</i>	20 MHz	20 MHz	20 MHz	20 / 40 MHz
<i>Coverage radius</i>	35 m	38 m	38 m	75 m
<i>Unlicensed spectrum</i>	Yes (it depends on countries)	Yes	Yes	Yes (it depends on countries)
<i>Radio Interference</i>	Low	High	High	Low
<i>Introduction cost</i>	Medium-Low	Low	Low	High-medium
<i>Device cost</i>	Medium-Low	Low	Low	Medium
<i>Mobility</i>	Yes	Yes	Yes	Yes
<i>Current use</i>	Medium	High	High	High
<i>Security</i>	Medium	Medium	Medium	High

Gambar 2.5 Standart IEEE 802.11

(Sumber : Sandra Sendra, 2011)

2.3 Protokol Streaming HTTP

HTTP protokol digunakan dalam *streaming* karena protokol ini lebih mudah diakses dari manapun. Menyediakan movie dari standart web server dengan nama lain *pseudo streaming* atau *progressive download* dikenal juga dengan *fast start*. Jika *file* telah di *download* oleh *user* tetapi bisa di *play* sebelum *download* selesai. Terlihat seperti *true streaming*. Bisa memiliki *data rate* yang lebih tinggi, sehingga memungkinkan kualitas lebih tinggi juga, *file* yang telah di *download* mudah untuk di *play* berulang – ulang. HTTP tidak bisa *live* tetapi bisa *streaming* semua jenis data

2.4 MJPEG

Motion Joint Photographic Expert Group Format Video ini memiliki kualitas gambar atau image yang dihasilkan suatu kamera sangat baik dikarenakan pixel gambar lebih rapat dan lebih banyak. (*Anonymous, 2006*)

Motion JPEG (M-JPEG) adalah nama informal untuk kelas di mana masing-masing format video yang berhubungan dalam sebuah video digital urutan, secara terpisah dikompresi sebagai *JPEG image*. Awalnya dikembangkan untuk aplikasi *PC multimedia*, M-JPEG sekarang digunakan oleh banyak perangkat *portable* dengan kemampuan *video capture*, seperti kamera digital.

Karakteristik MJPEG:

Pada *bandwidth* yang rendah, prioritas diberikan untuk resolusi gambar (gambar yang ditransmisi akan mempertahankan kualitas gambarnya, walaupun beberapa gambar akan rendah kualitasnya).

1. *Latency* minimum dalam pemrosesan gambar.
2. Gambar memiliki ukuran *file* yang konsisten.
3. Merupakan format kompresi yang paling banyak digunakan sekarang ini.

Keuntungan dari MJPEG:

1. *Frame* demi *frame* menawarkan lebih banyak *frame* untuk dilihat pada saat pemutaran.
2. Teknologi sederhana.
3. Mengurangi waktu *delay* ketika digunakan bersamaan dengan *audio*.
4. Pada *bandwidth* yang rendah, prioritas ditujukan pada resolusi gambar.
5. Lebih mudah digunakan jika menggunakan *photo editing*.

2.5 Raspberry Pi

Raspberry Pi, sering juga disingkat dengan nama Raspi, adalah komputer papan tunggal (*Single Board Circuit /SBC*) yang memiliki ukuran sebesar kartu kredit. Raspberry Pi bisa digunakan untuk berbagai keperluan, seperti *spreadsheet*, *game*, bahkan bisa digunakan sebagai *media player* karena kemampuannya dalam memutar video *high definition*. Raspberry Pi dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris.

Ide dibalik komputer mungil ini diawali dari keinginan untuk mencetak generasi baru programer, pada 2006 lalu. Seperti disebutkan dalam situs resmi Raspberry Pi Foundation, waktu itu Eben Upton, Rob Mullins, Jack Lang, dan Alan Mycroft, dari Laboratorium Komputer Universitas Cambridge memiliki

kekhawatiran melihat kian turunnya keahlian dan jumlah siswa yang hendak belajar ilmu komputer. Mereka lantas mendirikan yayasan Raspberry Pi bersama dengan Pete Lomas dan David Braben pada 2009. Tiga tahun kemudian, Raspberry Pi Model B memasuki produksi massal. Dalam peluncuran pertamanya pada akhir Februari 2012 dalam beberapa jam saja sudah terjual 100.000 unit.

Raspberry Pi memiliki dua model yaitu model A dan model B. Secara umum Raspberry Pi Model B, 512MB RAM. Perbedaan model A dan B terletak pada memory yang digunakan, Model A menggunakan memory 256 MB dan model B 512 MB. Selain itu model B juga sudah dilengkapi dengan *ethernet port* (kartu jaringan) yang tidak terdapat di model A. Desain Raspberry Pi didasarkan seputar SoC (*System-on-a-chip*) Broadcom BCM2835, yang telah menanamkan prosesor ARM1176JZF-S dengan 700 MHz, VideoCore IV GPU, dan 256 Megabyte RAM (model B). Penyimpanan data di desain tidak untuk menggunakan *hard disk* atau *solid-state drive*, melainkan mengandalkan kartu SD (*SD memory card*) untuk *booting* dan penyimpanan jangka panjang. Raspberry Pi merupakan komputer mini yang sangat murah, harganya hanya 25 dollar AS untuk Model A adapun 35 dollar AS untuk Model B per unit

Hardware Raspberry Pi tidak memiliki *real-time clock*, sehingga OS harus memanfaatkan *timer* jaringan *server* sebagai pengganti. Namun komputer yang mudah dikembangkan ini dapat ditambahkan dengan fungsi *real-time* (seperti DS1307) dan banyak lainnya, melalui saluran GPIO (*General-purpose input/output*) via antarmuka IC (*Inter-Integrated Circuit*).

Raspberry Pi bersifat *open source* (berbasis Linux), Raspberry Pi bisa dimodifikasi sesuai kebutuhan penggunaannya. Sistem operasi utama Raspberry Pi menggunakan Debian GNU/Linux dan bahasa pemrograman Python. Salah satu pengembang OS untuk Raspberry Pi telah meluncurkan sistem operasi yang dinamai Raspbian, Raspbian diklaim mampu memaksimalkan perangkat Raspberry Pi. Sistem operasi tersebut dibuat berbasis Debian yang merupakan salah satu distribusi Linux OS.

BAB 3. METODELOGI PENELITIAN

Dalam bab ini berisi tentang jadwal penelitian, tempat penelitian dan waktu penelitian. Selain itu bab ini juga berisi tentang *flowchart* sistem yang akan dirancang dan juga menjelaskan tentang metode yang digunakan untuk menyelesaikan skripsi.

3.1 Jadwal dan Tempat Penelitian

Tempat penelitian dilakukan di Laboratorium Jaringan Komputer, Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember, Jl. Slamet Riyadi No.62 Patrang, Kabupaten Jember.

3.2 Tahap Perencanaan

Dalam penelitian ini langkah-langkah yang akan dilakukan dalam melakukan perancangan *mobile robot* adalah sebagai berikut:

1. Studi literatur.

Tahap awal dari penelitian ini yaitu mencari literatur dari penelitian sebelumnya. Sehingga diharapkan penelitian dapat dilaksanakan dan memberikan arahan untuk mengurangi kesalahan.

2. Perancangan Alat.

Tahap kedua adalah perancangan alat untuk membuat sistem navigasi *mobile robot* menggunakan *web streaming* dengan alat dan bahan yang sudah dibeli.

3. Pembuatan Alat.

Pada tahap ini, alat yang telah direncanakan akan dibuat menjadi sebuah sistem yang bekerja dengan baik.

4. Pengujian Alat dan Analisis.

Pada tahap ini dilakukan pengujian alat yang sudah dibuat. Hal yang akan diuji dan dianalisis yaitu pengaruh nilai *lumen* terhadap keberhasilan deteksi dan pembacaan koordinat dari objek. Keberhasilan koneksi secara TCP/IP antara *mobile robot* dengan server yang menggunakan protokol TCP pada layer Transport.

3.3 Alat dan Bahan

Alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

3.3.1 Hardware

- a. Raspberry pi tipe B+.
- b. Driver motor.
- c. *Wireless* Edimax WN7211Un.
- d. Kamera Webcam.

3.3.2 Software

- a. Python.
- b. Ubuntu 14.04 yang digunakan sebagai *server web steaming*.
- c. Hostapd.
- d. Tcpdump.
- e. Iptraf

3.4 Gambaran Umum Sistem



Gambar 3.1 Gambaran Umum Sistem

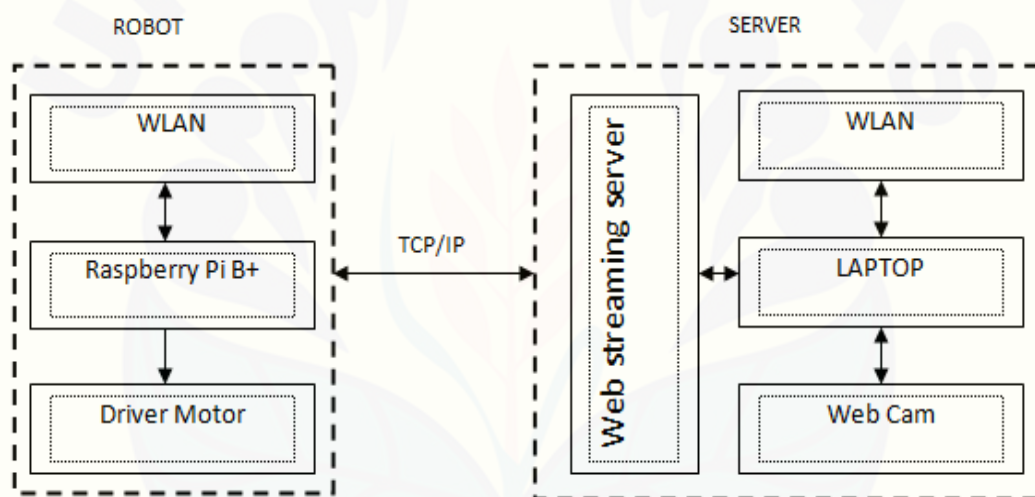
Gambar 3.1 merupakan gambaran umum sistem yang akan dirancang. Pertama server akan mencari kamera webcam yang digunakan. Kemudian server akan mengambil gambar dari kamera webcam tersebut. Server akan mengolah gambar dari kamera webcam untuk mendeteksi bola. Kemudian setelah berhasil mendeteksi koordinat bola, maka server memberi tanda berupa lingkaran kecil dengan warna hijau pada koordinat tersebut. Lingkaran ini sebagai tanda titik tengah dari bola tersebut. Kemudian hasil dari pengolahan citra yang dilakukan server akan ditampilkan pada halaman *website* lokal server.

Server menggunakan port 7070 untuk menampilkan gambar dari kamera webcam pada *website* lokal. Gambar yang ditampilkan pada *website* lokal ini

memiliki ekstensi .mjpeg. Gambar *mjpeg* kemudian akan distreamingkan melalui protokol *HTTP* kepada *user*. Pada gambar 3.1, *access point* digunakan sebagai media koneksi antara server dengan *mobile robot*.

Pada sistem yang dibuat ini, protokol yang digunakan adalah protokol TCP/IP. Setelah koneksi secara TCP/IP terjalin, maka server akan mengirim data kepada *mobile robot*. Data yang dikirim server adalah nilai dari koordinat bola yang terdeteksi server. Untuk menerima koordinat yang dikirimkan server, raspberry akan membuka port 2020. Kemudian nilai koordinat bola diproses oleh raspberry. Mulai dari maju, mundur, putar kanan dan putar kiri akan menggunakan nilai koordinat bola yang dikirim oleh server melalui protokol TCP/IP.

3.5 Blok Diagram Alat



Gambar 3.2 Diagram Blok Alat

Gambar 3.2 merupakan gambar diagram blok dari sistem yang akan dibuat, diagram blok diatas terdiri dari 2 buah diagram blok sistem, yaitu sebagai berikut :

3.5.1 Blok *Mobile Robot*

Dalam diagram blok *mobile robot*, terdapat raspberry pi sebagai pengendali. Kemudian bagian kedua adalah perangkat WLAN, perangkat ini digunakan sebagai media transmisi data dari server menuju *mobile robot* ataupun sebaliknya. Perangkat WLAN yang digunakan adalah IEEE 802.11. Setelah

komunikasi antar *mobile robot* dengan server terjalin, raspberry pi akan mengolah data yang dikirimkan server. Data yang dikirim server adalah nilai koordinat dari bola yang telah dideteksi.

Bahasa pemrograman yang digunakan untuk koneksi dan pengendalian *mobile robot* adalah python. *Mobile robot* akan membuka port 2020 untuk proses pengiriman nilai koordinat bola.

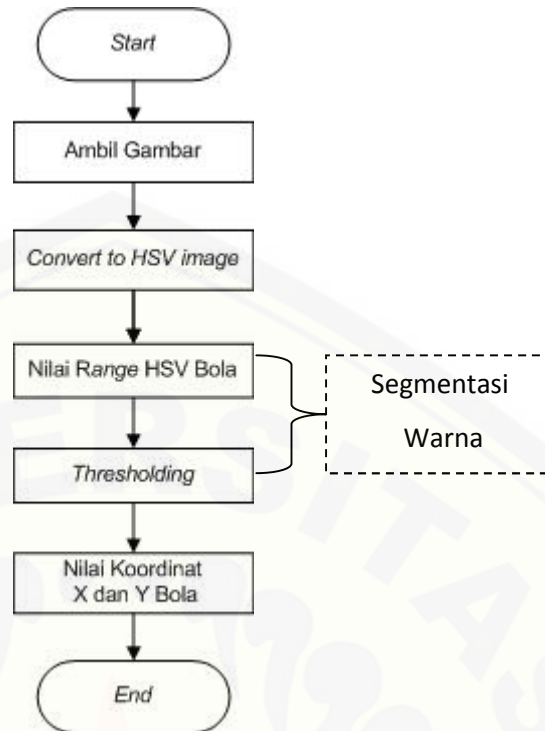
Raspberry pi memiliki dua fungsi utama dalam sistem ini. Fungsi pertama adalah untuk mengontrol motor, mulai dari gerak maju, mundur, putar kanan dan putar kiri. Kemudian fungsi kedua raspberry pi adalah membuka port 2020. Port 2020 ini digunakan untuk menerima nilai koordinat bola yang dikirimkan server.

3.5.2 Blok Server

Dalam diagram blok sisi server, laptop digunakan sebagai server yang akan menyediakan sebuah halaman website. Website ini berguna sebagai *interfaces* sekaligus sebagai media *video web streaming* hasil *capturing image* dari kamera webcam. Format *video streaming* yang digunakan adalah *mjpeg*. Server akan melakukan method POST untuk menampilkan *web video streaming* pada halaman website. Selain melakukan method POST, server juga melayani method GET yang dikirimkan *user* untuk mengakses *video web streaming*. Perangkat WLAN dengan standart 802.11 digunakan sebagai media koneksi antara *mobile robot* dengan server.

Pada sisi server, bahasa pemrograman python juga digunakan untuk *capturing video* dari kamera webcam. Hasil *capturing* dari kamera webcam akan diubah menjadi format *mjpeg*. Format gambar dengan ekstensi *mjpeg* ini kemudian akan ditampilkan pada website lokal server.

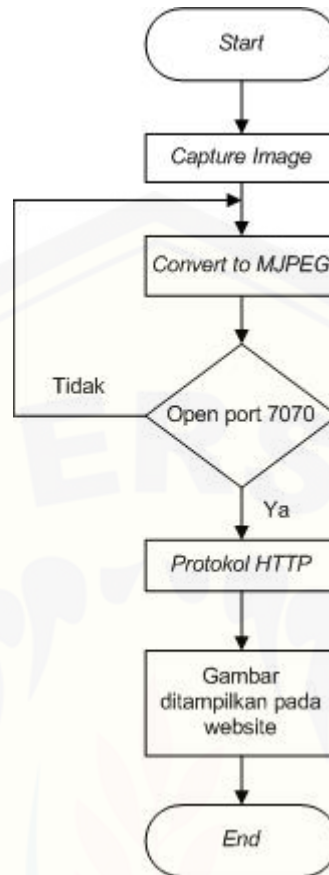
3.6 Flowchart Image Processing



Gambar 3.3 Flowchart Image Processing

Proses *image processing* dimulai dengan pengambilan gambar dari kamera webcam. Hasil pengambilan gambar adalah citra RGB. Citra RGB ini kemudian diubah menjadi citra HSV. Setelah berhasil mengubah menjadi citra HSV maka selanjutnya adalah menentukan *range* nilai citra HSV bola yang akan dideteksi. Selanjutnya adalah *thresholding* yang menghasilkan gambar hitam dan putih. Objek akan berwarna putih dan *background* berwarna hitam. Kemudian dicari nilai tengah dari objek putih tersebut. Nilai tengah ini menghasilkan nilai koordinat X dan koordinat Y bola.

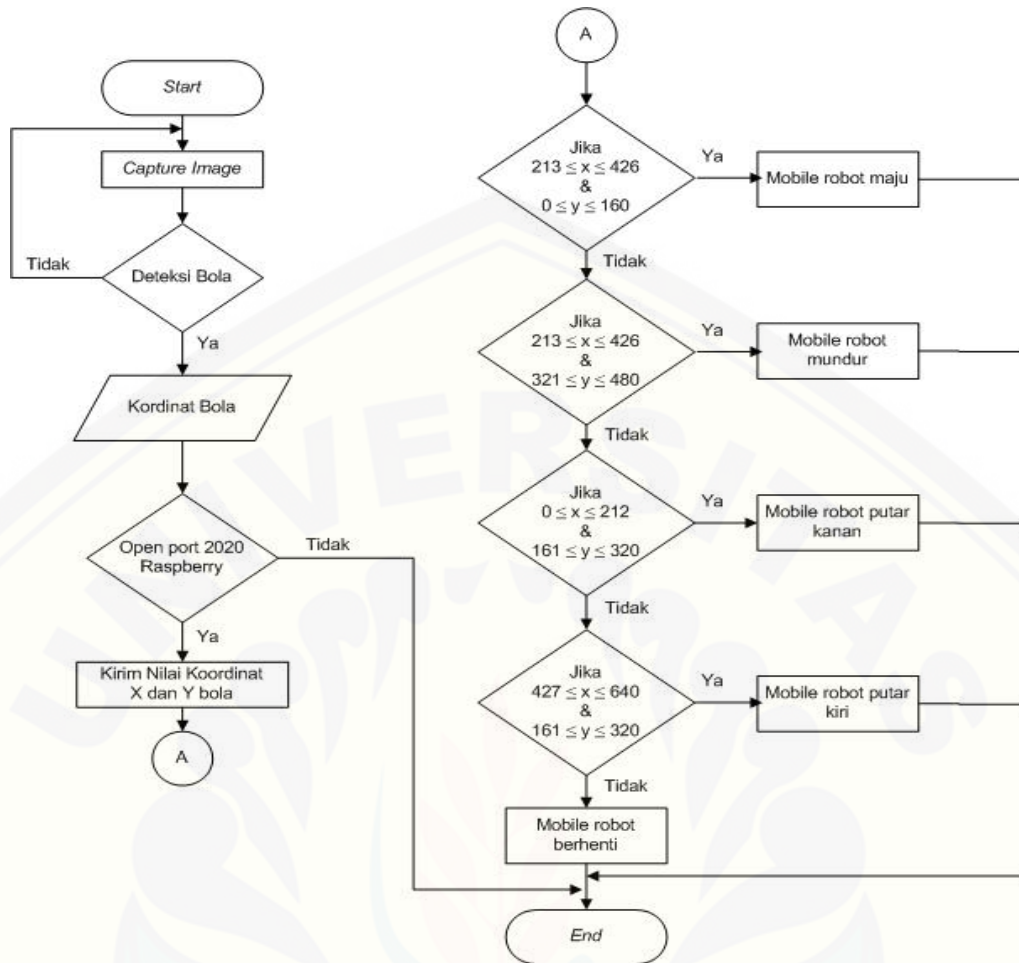
3.7 Flowchart Menampilkan Gambar pada Website



Gambar 3.4 Flowchart Menampilkan Gambar Pada Website

Gambar 3.4 adalah gambar *flowchart* untuk menampilkan gambar dari kamera webcam pada website lokal. Ketika *start* maka server akan melakukan *capture image* menggunakan kamera webcam. Setelah itu, server akan mendeteksi keberadaan bola dan juga membaca nilai koordinatnya. Hasil *capture image* ini akan dirubah ke dalam format *mjpeg*. Setelah berhasil merubah format video ke *mjpeg*, maka *server* akan membuka port yang digunakan untuk menampilkan *video streaming* pada halaman website. Ketika *server* berhasil membuka *port*, maka *server* akan menampilkan *video streaming* pada *port 7070*.

3.8 Flowchart Mengirim Nilai Koordinat Bola



Gambar 3.5 Flowchart Pengiriman Koordinat Bola

Gambar 3.5 merupakan *flowchart* untuk mengirim koordinat bola menuju raspberry yang dilakukan server. Hasil pengolahan citra yang dilakukan oleh server menghasilkan nilai koordinat bola yang sudah terdeteksi. Kemudian nilai ini dikirimkan menuju port 2020 raspberry pi. Pada saat mengirim, server akan mengecek apakah port 2020 raspberry sudah dibuka atau belum. Jika sudah dibuka maka pengiriman nilai koordinat akan berlangsung. Pengiriman nilai koordinat dengan menggunakan protokol TCP pada layer Transport TCP/IP.

Ketika bola berada pada range koordinat $213 \leq X \leq 426$ dan $0 \leq Y \leq 160$ maka *mobile robot* akan gerak maju. Jika bola berada pada range koordinat $213 \leq X \leq 426$ dan $321 \leq Y < 480$ maka *mobile robot* akan gerak mundur. Jika bola berada pada range koordinat $0 \leq X \leq 212$ dan $161 \leq Y \leq 320$ maka *mobile robot* putar kanan. Jika bola berada pada range koordinat $427 \leq X \leq 640$ dan $161 \leq Y \leq 320$ maka *mobile robot* putar kiri.

BAB 4. HASIL DAN PEMBAHASAN

Pada bab ini dilakukan pengujian alat yang sudah dibuat. Hal yang akan diuji dan dianalisis yaitu pengaruh nilai *lumen* terhadap keberhasilan deteksi dan pembacaan koordinat dari bola. Protokol yang digunakan untuk pengiriman koordinat bola menggunakan TCP pada layer Transport. Keberhasilan penggunaan protokol TCP pada layer Transport akan diuji dengan keberhasilan proses *hand shaking*.

Untuk mengetahui respon *mobile robot*, maka digunakan parameter *packet loss* dan *delay* antar paket yang dikirimkan server. Pengukuran respon *mobile robot* ini bertujuan untuk mengetahui *delay* antara gerakan objek dengan gerakan *mobile robot*. Perhitungan *delay* dengan melakukan pengamatan waktu kedatangan tiap paket pada *mobile robot*.

4.1 Pengujian Program Server

4.1.1 Pengujian Program Deteksi Objek

Langkah pertama dalam melakukan proses pendeteksian objek dengan metode segmentasi warna menggunakan bahasa program python dengan library *opencv* adalah melakukan pengambilan gambar dari kamera webcam yang digunakan. Untuk mengambil gambar dari kamera webcam, kita harus mendeteksi kamera webcam dahulu. Berikut listing program untuk mengambil gambar dengan kamera webcam pada server dengan menggunakan *python* dan *opencv*.

```
import string, cgi, time
from os import curdir, sep
from BaseHTTPServer import BaseHTTPRequestHandler,
HTTPServer
from SocketServer import ThreadingMixIn
import cv
import re

capture = cv.CaptureFromCAM(1)
```

```

frame =cv.QueryFrame (capture)
test =cv.CreateImage (cv.GetSize (frame) , 8, 3)
while True:
    frame=cv.QueryFrame (capture)
    cv.Flip (frame, frame, 1)

```

Kemudian hasil dari listing program diatas adalah seperti pada gambar 4.1.

Gambar 4.1 merupakan gambar dengan format citra RGB, untuk melakukan proses segmentasi warna maka, citra RGB ini akan diubah dahulu menjadi citra HSV. Proses segmentasi warna ini diperlakukan pada citra dengan format HSV. Dibawah ini merupakan perintah program untuk mengubah citra RGB menjadi citra HSV.

```

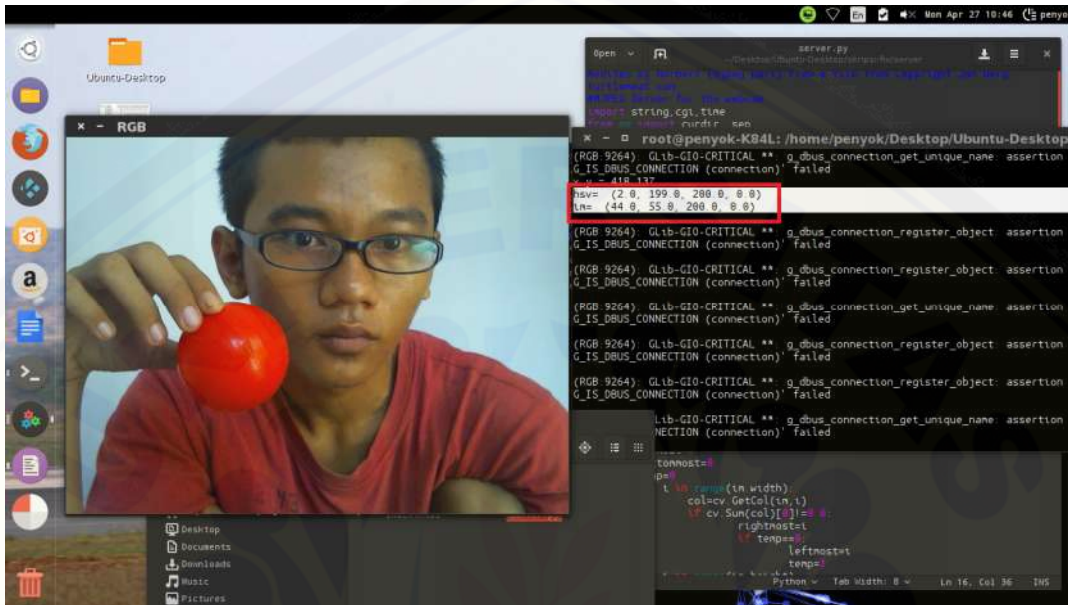
def getthresholdedimg (im) :
    imghsv=cv.CreateImage (cv.GetSize (im) , 8, 3)
    cv.CvtColor (im, imghsv, cv.CV_BGR2HSV)

```



Gambar 4.1 Citra RGB Dan Citra HSV

Kemudian setelah citra hasil dari kamera webcam memiliki format HSV, selanjutnya adalah menentukan nilai warna HSV objek yang akan dideteksi. Pada penelitian ini, objek yang digunakan berupa bola dengan warna merah. Maka kita akan mencari nilai dari warna HSV bola tersebut. Gambar dibawah ini merupakan gambar pada saat mencari nilai citra HSV bola yang digunakan.



Gambar 4.2 Nilai Citra HSV Bola.

Dari gambar 4.2 nilai HSV bola adalah 2,199,200. Setelah kita mengetahui nilai dari citra HSV bola yang digunakan, maka proses selanjutnya adalah melakukan proses untuk membedakan objek dengan background. Metode yang digunakan adalah segmentasi warna. Kemudian setelah melakukan proses segmentasi warna, selanjutnya adalah mendeteksi titik tengah dari bola tersebut. Titik tengah ini yang akan menjadi titik koordinat x dan y dari bola yang digunakan. Di bawah ini merupakan listing program untuk proses segmentasi warna dan juga mencari titik koordinat objek bola.

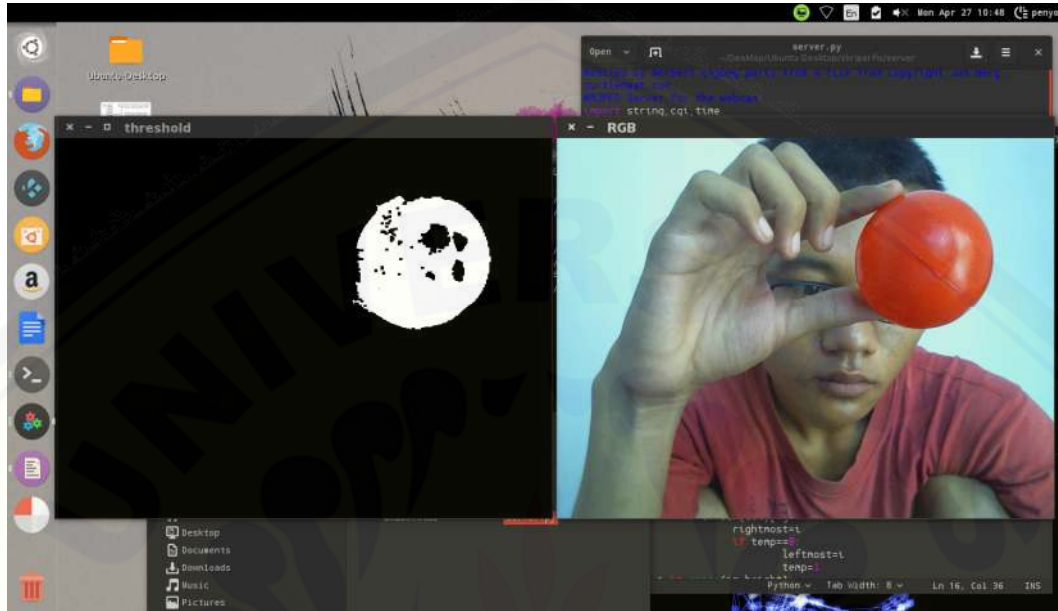
```
def getthresholdedimg(im):
    imgHSV=cv.CreateImage(cv.GetSize(im),8,3)
    cv.CvtColor(im,imgHSV,cv.CV_BGR2HSV)
    imgthreshold=cv.CreateImage(cv.GetSize(im),8,1)
    cv.InRangeS(imgHSV,(2,199,200),(12,255,255),imgthres
hold)
    cv.Dilate(imgthreshold,imgthreshold, None, 9)
    cv.Erode(imgthreshold, imgthreshold, None, 6)
    storage = cv.CreateMemStorage(0)
    contour = cv.FindContours(imgthreshold, storage,
cv.CV_RETR_LIST, cv.CV_CHAIN_APPROX_SIMPLE)
```

```
    return imgthreshold

def getpositions(im):

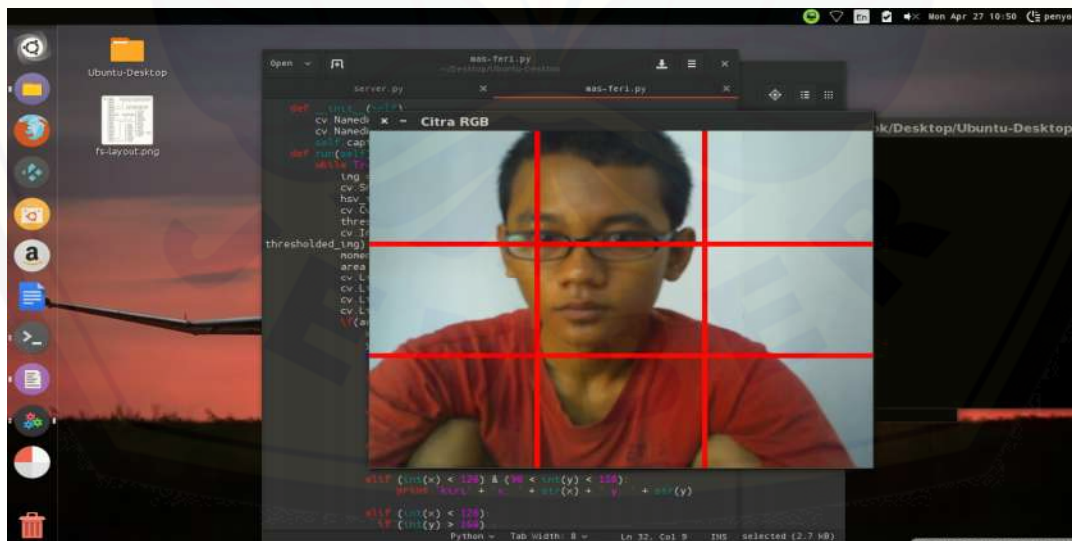
    leftmost=0
    rightmost=0
    topmost=0
    bottommost=0
    temp=0
    for i in range(im.width):
        col=cv.GetCol(im,i)
        if cv.Sum(col)[0]!=0.0:
            rightmost=i
            if temp==0:
                leftmost=i
                temp=1
    for i in range(im.height):
        row=cv.GetRow(im,i)
        if cv.Sum(row)[0]!=0.0:
            bottommost=i
            if temp==1:
                topmost=i
                temp=2
    return (leftmost,rightmost,topmost,bottommost)
```

Gambar 4.3 adalah gambar asli hasil dari *capture* dengan kamera webcam. Kemudian dengan metode segmentasi warna, maka warna bola dan warna *background* dipisahkan. Dimana warna *background* diubah menjadi warna hitam dan warna bola diubah menjadi warna putih. Kemudian setelah bola terdeteksi, selanjutnya server akan membaca koordinat dari bola tersebut. Server akan mengirimkan nilai koordinat bola kepada *mobile robot*. Pengiriman ini menggunakan protokol TCP/IP dan protokol TCP pada layer transport. Nilai koordinat akan diterima raspberry pada port 2020.



Gambar 4.3 Segmentasi Warna.

Secara default frame video dengan menggunakan opencv yaitu 640 X 480. Kemudian frame ini dibagi menjadi 9 kotak kecil. Hal ini ditujukan untuk memudahkan *user* dalam melakukan navigasi *mobile robot*.



Gambar 4.4 Membagi Frame Menjadi 9 Area.

Kemudian setelah berhasil mengambil gambar dari kamera webcam, maka langkah selanjutnya adalah mengubah format video tersebut kedalam format mjpeg. Format mjpeg ini nanti akan ditampilkan dalam website lokal. Berikut ini adalah listing program untuk mengubah video menjadi format mjpeg.

```
if self.path.endswith(".mjpeg") :
    self.send_response(200)
self.wfile.write("Content-Type:          multipart/x-mixed-
replace; boundary=--aaboundary")
self.wfile.write("\r\n\r\n")
```

Kemudian untuk menampilkan video tersebut kedalam website lokal, kita harus menentukan alamat ip dan port yang akan digunakan. Berikut ini adalah listing program untuk menampilkan frame video pada website lokal menggunakan *python* dan *opencv*.

```
def main():
try:
server = ThreadedHTTPServer(('127.0.0.1', 7070),
MyHandler)
print 'started httpserver...'
server.serve_forever()
time.time (10)
except KeyboardInterrupt:
print '^C received, shutting down server'
server.socket.close()
```

Fungsi utama untuk menampilkan video dalam website adalah *server = ThreadedHTTPServer(('127.0.0.1', 7070), MyHandler)*. Hasil video akan ditampilkan pada alamat website “127.0.0.1:7070/camera.mjpeg”. Berikut ini hasil video frame yang dihasilkan pada website lokal. 127.0.0.1 adalah alamat *localhost* dari suatu komputer. *User* dapat mengakses dengan menggunakan alamat ip lokal server pada jaringan yang digunakan. Gambar 4.5 merupakan hasil dari program untuk menampilkan gambar pada website lokal.



Gambar 4.5 Tampilan Gambar Pada Website Lokal

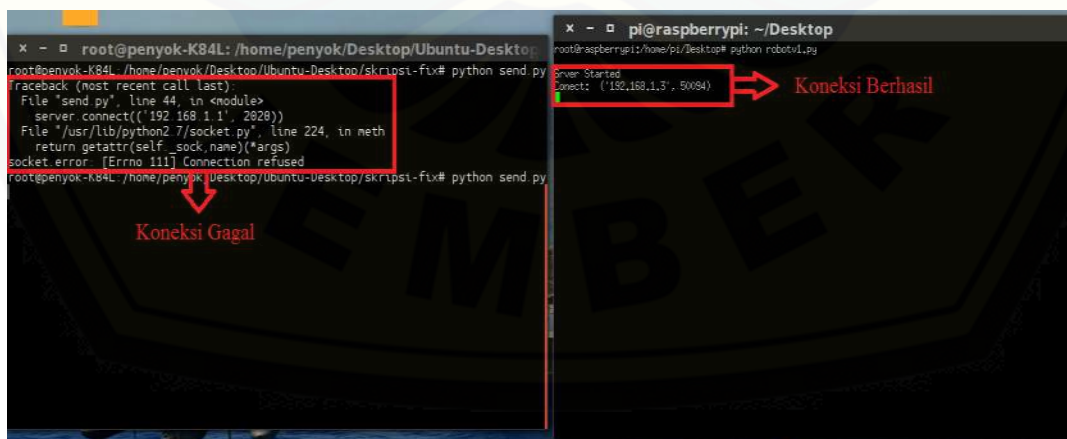
Gambar 4.5 diatas merupakan tampilan dari website lokal server dengan alamat IP *127.0.0.1:7070/index.html*. Pada halaman website tersebut, *user* dapat memilih kualitas gambar yang diinginkan. Terdapat 3 pilihan kualitas gambar, yaitu mulai dari *low*, *medium* dan *high*. Terdapat 9 kotak dalam gambar 4.1, kotak no 1,3,5,7, dan 9 digunakan untuk memberi perintah berhenti pada *mobile robot*. Jika bola yang terdeteksi berada pada kotak 1, 3, 5, 7 dan 9 maka *mobile robot* tidak akan melakukan aksi. Kemudian jika bola terdeteksi berada pada kotak nomor 2 maka *mobile robot* akan bergerak maju. Jika bola terdeteksi berada pada kotak nomor 4, maka *mobile robot* bergerak putar kiri. Jika bola terdeteksi pada kotak nomor 6 maka *mobile robot* akan melakukan aksi putar kanan. Kemudian jika bola terdeteksi pada kotak nomor 8, maka *mobile robot* akan bergerak mundur.

4.1.2 Program Pengirim Nilai Koordinat Objek.

Setelah nilai koordinat diketahui, maka proses selanjutnya adalah mengirimkan nilai koordinat objek tersebut kepada *mobile robot*. Pengiriman nilai koordinat objek dengan menggunakan protokol TCP/IP dengan layer transport menggunakan protokol TCP. Sebelum melakukan pengiriman, maka harus ditentukan dahulu port yang digunakan oleh *mobile robot*. Port ini nanti akan dibuka oleh *mobile robot* untuk menerima nilai koordinat yang dikirimkan server. Pada penelitian ini *mobile robot* akan membuka port 2020 untuk menerima nilai yang dikirimkan server. Maka alamat tujuan pada pengiriman yang dilakukan oleh server adalah alamat IP *mobile robot* dengan port 2020. Berikut ini adalah listing program untuk mengirimkan nilai koordinat objek.

```
server = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
server.connect(('192.168.1.1', 2020))

x = int (posx)
y = int (posy)
listed = [x,y]
telem = ','.join(map(str, listed))
server.send(telem)
```



Gambar 4.6 Ketika Gagal Dan Berhasil Koneksi.

4.2 Program Mobile Robot.

4.2.1 Program Penerima Nilai Koordinat Objek.

Agar server dapat mengirimkan nilai koordinat bola kepada *mobile robot*, maka terlebih dahulu *mobile robot* haruslah membuka port 2020 sebagai alamat tujuan dari nilai koordinat bola yang dikirimkan server. Jika *mobile robot* gagal membuka port 2020, maka koneksi antara server dengan *mobile robot* tidak akan terjalin. Berikut ini adalah listing program pada raspberry pi untuk membuka port 2020.

```
import sys
import socket
from subprocess import call
import os

server = socket.socket(socket.AF_INET,
socket.SOCK_STREAM)
server.bind(('0.0.0.0', 2020))
server.listen(1)
conn, addr = server.accept()
print "\nServer Started"
print 'Connect: ', addr
data = 'nothing'
```

4.2.2 Program Navigasi Mobile Robot.

Setelah pengiriman nilai koordinat objek berhasil, maka nilai koordinat ini akan diolah terlebih dahulu oleh *mobile robot*. Pengolahan ini bertujuan untuk menyamakan range nilai koordinat untuk pergerakan *mobile robot* yang ada pada *website*. Untuk aksi *maju mobile robot* ditentukan nilai koordinat X antara 213 sampai 426 dan koordinat Y antara 0 sampai 160. Untuk aksi *mundur mobile robot* ditentukan nilai koordinat X antara 213 sampai 426 dan koordinat Y antara 321 sampai 480. Untuk aksi *putar kanan mobile robot* ditentukan nilai koordinat X antara 0 sampai 212 dan koordinat Y antara 161 sampai 320. Untuk aksi *putar kiri mobile robot* ditentukan nilai koordinat X antara 427 sampai 640 dan koordinat Y antara 161

sampai 320. Selain dari range nilai koordinat tersebut maka *mobile robot* akan stop.

Berikut ini adalah listing program untuk navigasi *mobile robot*.

```
if data :
    got = data.split(',')
    x = int (got[0])
    y = int (got[1])
    if (213 < x < 426) & (0 < y < 160):
        GPIO.setup(7,GPIO.OUT)
        GPIO.setup(11,GPIO.OUT)
        GPIO.setup(13,GPIO.OUT)
        GPIO.setup(15,GPIO.OUT)
        GPIO.output(7,True)
        GPIO.output(11,False)
        GPIO.output(13,False)
        GPIO.output(15,True)

    elif (213 < x < 426) & (321 < y < 480):
        GPIO.setup(7,GPIO.OUT)
        GPIO.setup(11,GPIO.OUT)
        GPIO.setup(13,GPIO.OUT)
        GPIO.setup(15,GPIO.OUT)
        GPIO.output(7, False)
        GPIO.output(11,True)
        GPIO.output(13,True)
        GPIO.output(15,False)

    elif ( 0 < x < 212) & (161 < y < 320):
        GPIO.setup(7,GPIO.OUT)
        GPIO.setup(11,GPIO.OUT)
        GPIO.setup(13,GPIO.OUT)
        GPIO.setup(15,GPIO.OUT)
        GPIO.output(7,False)
        GPIO.output(11,False)
        GPIO.output(13,False)
        GPIO.output(15,True)

    elif (427 < x < 640) & (161 < y < 320):
        GPIO.setup(7,GPIO.OUT)
        GPIO.setup(11,GPIO.OUT)
        GPIO.setup(13,GPIO.OUT)
        GPIO.setup(15,GPIO.OUT)
        GPIO.output(7,True)
```

```
GPIO.output(11, False)
GPIO.output(13, False)
GPIO.output(15, False)

else:
    GPIO.setup(7, GPIO.OUT)
    GPIO.setup(11, GPIO.OUT)
    GPIO.setup(13, GPIO.OUT)
    GPIO.setup(15, GPIO.OUT)
    GPIO.output(7, False)
    GPIO.output(13, False)
    GPIO.output(11, False)
    GPIO.output(15, False)
```

4.3 Pengujian Keberhasilan Koneksi TCP/IP.

Protokol yang digunakan dalam pengiriman data antara server dengan *mobile robot* menggunakan TCP pada layer transport dengan koneksi secara TCP/IP. Keberhasilan penggunaan protokol TCP pada layer transport dapat dilihat dengan ada tidaknya proses *handshaking*. Proses *handshaking* ini dapat dilihat dengan menganalisa tiap paket yang dikirim dan diterima oleh robot. *Tools* yang digunakan dalam menganalisa paket adalah *tcp dump*.

Untuk mengetahui respon *mobile robot*, maka digunakan parameter *packet loss* dan *delay* antar paket yang dikirim oleh server ke *mobile robot* dengan menggunakan protokol TCP/IP. Paket yang dikirim adalah koordinat dari bola yang telah terdeteksi. Server mengirim nilai koordinat ke alamat IP *mobile robot* dengan port 2020. Nilai yang dikirim berupa nilai koordinat x dan koordinat y bola. Pengukuran respon *mobile robot* ini bertujuan untuk mengetahui *delay* antara gerakan bola dengan gerakan *mobile robot*. Perhitungan *delay* dengan melakukan pengamatan waktu kedatangan tiap paket dengan menggunakan *tcpdump*. Gambar 4.3 merupakan hasil dari *tcpdump* dengan menggunakan perintah *tcpdump -i wlan0*.

```

root@penyok-K84L: /home/penyok
[ -r file ] [ -s snaplen ] [ -T type ] [ -V file ] [ -w file ]
[ -M #filecount ] [ -y dataLinktype ] [ -z command ]
[ -i user ] [ expression ]

root@penyok-K84L: /home/penyok# tcpdump -i wlan0
tcpdump verbose output suppressed, use -v or -vv for full protocol decode
listening on wlan0, link-type EN10MB (Ethernet), capture size 65535 bytes
14.58.14.043651 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [P], seq 7892332, win 229, options [nop,nop,TS val 1464985 ecr 592838], length 6
14.58.14.658258 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 6, win 453, options [nop,nop,TS val 592851 ecr 1464985], length 0
14.58.14.198572 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 6, win 1, win 229, options [nop,nop,TS val 1465822 ecr 592851], length 4
14.58.14.197446 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 10, win 453, options [nop,nop,TS val 592866 ecr 1465822], length 0
14.58.14.324327 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 10, ack 1, win 229, options [nop,nop,TS val 1465856 ecr 592866], length 4
14.58.14.336768 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 14, win 453, options [nop,nop,TS val 592879 ecr 1465856], length 0
14.58.14.467729 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 14, win 1, win 229, options [nop,nop,TS val 1465891 ecr 592879], length 6
14.58.14.474689 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 28, win 453, options [nop,nop,TS val 592884 ecr 1465891], length 8
14.58.14.618574 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 20, ack 1, win 229, options [nop,nop,TS val 1465127 ecr 592884], length 6
14.58.14.617807 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 26, win 453, options [nop,nop,TS val 592988 ecr 1465127], length 0
14.58.14.748396 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 26, ack 1, win 229, options [nop,nop,TS val 1465162 ecr 592988], length 6
14.58.14.754759 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 32, win 453, options [nop,nop,TS val 592922 ecr 1465162], length 0
14.58.14.894548 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 32, ack 1, win 229, options [nop,nop,TS val 1465198 ecr 592922], length 6
14.58.14.901129 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 38, win 453, options [nop,nop,TS val 592937 ecr 1465198], length 0
14.58.15.007997 IP penyok-K84L.local.65155 > 8.8.8.8.domain.53292 PTR [ ], seq 1, win 1, win 192, len=addr arpa (42)
14.58.15.098443 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 38, ack 1, win 229, options [nop,nop,TS val 1465232 ecr 592937], length 3
14.58.15.099992 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 42, win 453, options [nop,nop,TS val 592958 ecr 1465232], length 0
14.58.15.108382 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 43, ack 1, win 229, options [nop,nop,TS val 1465267 ecr 592958], length 6
14.58.15.174798 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 49, win 453, options [nop,nop,TS val 592964 ecr 1465267], length 0
14.58.15.307766 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 49, ack 1, win 229, options [nop,nop,TS val 1465301 ecr 592964], length 6
14.58.15.314581 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 55, win 453, options [nop,nop,TS val 592978 ecr 1465301], length 0
14.58.15.451852 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 55, ack 1, win 229, options [nop,nop,TS val 1465337 ecr 592978], length 5
14.58.15.458251 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 60, win 453, options [nop,nop,TS val 592992 ecr 1465337], length 0
14.58.15.584721 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 60, ack 1, win 229, options [nop,nop,TS val 1465371 ecr 592992], length 6
14.58.15.591286 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 66, win 453, options [nop,nop,TS val 593086 ecr 1465371], length 0
14.58.15.727794 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 66, ack 1, win 229, options [nop,nop,TS val 1465406 ecr 593086], length 5
14.58.15.734178 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 71, win 453, options [nop,nop,TS val 593020 ecr 1465406], length 0
14.58.15.871159 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 71, ack 1, win 229, options [nop,nop,TS val 1465442 ecr 593020], length 4
14.58.15.877569 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 75, win 453, options [nop,nop,TS val 593034 ecr 1465442], length 0
14.58.16.004436 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 75, ack 1, win 229, options [nop,nop,TS val 1465476 ecr 593034], length 6
14.58.16.019947 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 81, win 453, options [nop,nop,TS val 593048 ecr 1465476], length 0
14.58.16.147998 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 81, ack 1, win 229, options [nop,nop,TS val 1465511 ecr 593048], length 7
14.58.16.154327 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 88, win 453, options [nop,nop,TS val 593862 ecr 1465511], length 0
14.58.16.298716 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 88, ack 1, win 229, options [nop,nop,TS val 1465547 ecr 593862], length 5
14.58.16.297211 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 93, win 453, options [nop,nop,TS val 593876 ecr 1465547], length 0
14.58.16.424347 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 93, ack 1, win 229, options [nop,nop,TS val 1465581 ecr 593876], length 4
14.58.16.438815 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 97, win 453, options [nop,nop,TS val 593890 ecr 1465581], length 0
14.58.16.575196 IP penyok-K84L.local.59328 > 192.168.1.1.2028 Flags [P], seq 97, ack 1, win 229, options [nop,nop,TS val 1465618 ecr 593890], length 6
14.58.16.581584 IP 192.168.1.1.2028 > penyok-K84L.local.59328 Flags [ ], ack 103, win 453, options [nop,nop,TS val 593885 ecr 1465618], length 8

```

Gambar 4.7 Capture Paket TCP Dengan *Tcpdump*

Tabel 4.1 merupakan hasil dari *capture* data dengan menggunakan *tcpdump* seperti ditunjukkan gambar 4.7. Tabel 4.1 menunjukkan komunikasi antara server dengan *mobile robot*. Hasil dari *tcpdump* pada baris pertama sampai dengan baris ke 7 tujuh ditunjukkan pada tabel 4.1. Koneksi pertama pada protokol TCP yaitu dengan pengiriman paket SYN(*synchronization*) dari pengirim ke penerima. Kemudian penerima membalas dengan mengirimkan paket SYN (*synchronization*) dan ACK (*acknowledgement*) kepada pengirim. Setelah itu pengirim membalas dengan paket ACK (*acknowledgement*), yang berarti bahwa transmisi dengan protokol TCP siap dilakukan. Ketika seluruh data yang ditransmisikan sudah selesai, maka penerima akan mengirim paket FIN(*Finish*). Dengan menggunakan tools *tcpdump* kita dapat mengcapture seluruh paket yang dalam suatu jaringan komputer. Dengan melihat waktu kedatangan dan waktu pengiriman paket kita dapat mengetahui *delay* tiap paket yang dikirim. Selain itu kita juga dapat melihat alamat tujuan dari tiap paket-paket tersebut.

Tabel 4.1 Hasil *capture* paket TCP dengan *tcpdump*

No	Tcpdump
1	16:02:53.182494 IP penyok-K84L.local.59733 > 192.168.1.1.2020: Flags [S], seq 1173556807, win 29200, options [mss 1460,sackOK,TS val 2434770 ecr 0,nop,wscale 7], length 0
2	16:02:53.188848 IP 192.168.1.1.2020 > penyok-K84L.local.59733: Flags [S.], seq 2870595692, ack 1173556808, win 28960, options [mss 1460,sackOK,TS val 216590 ecr 2434770,nop,wscale 6], length 0
3	16:02:53.188942 IP penyok-K84L.local.59733 > 192.168.1.1.2020: Flags [.], ack 1, win 229, options [nop,nop,TS val 2434772 ecr 216590], length 0
4	16:02:53.235779 IP penyok-K84L.local.59733 > 192.168.1.1.2020: Flags [P.], seq 1:8, ack 1, win 229, options [nop,nop,TS val 2434783 ecr 216590], length 7
5	16:02:53.242232 IP 192.168.1.1.2020 > penyok-K84L.local.59733: Flags [.], ack 8, win 453, options [nop,nop,TS val 216596 ecr 2434783], length 0
6	16:02:53.261997 IP penyok-K84L.local.59733 > 192.168.1.1.2020: Flags [P.], seq 8:14, ack 1, win 229, options [nop,nop,TS val 2434790 ecr 216596], length 6
7	16:02:53.268412 IP 192.168.1.1.2020 > penyok-K84L.local.59733: Flags [.], ack 14, win 453, options [nop,nop,TS val 216598 ecr 2434790], length 0

Tabel 4.1 menunjukkan komunikasi secara TCP/IP antara server dengan *mobile robot* dengan menggunakan protokol TCP pada layer transportnya. Dari tabel 4.1 diatas dapat diketahui bahwa :

- Baris 1, pada pukul 16:02:53 : time stamp 182494 Sever (penyok-K84L.local.59733) melakukan koneksi ke mobile robot (192.168.1.1) dengan port 2020. [S] menandakan SYN pada control/code bit diaktifkan yang menandakan sebuah koneksi TCP/IP akan segera dilakukan dengan sequence number awal 1173556807.

- Baris 2, pada pukul 16:02:53 time stamp 188848
Mobile robot merespon koneksi dari Server. Mobile robot membalas dengan mengirimkan kodebit SYN dan ACK dengan sequence number 2870595692, lagi-lagi tidak ada datayang dikirimkan. Selain itu Server juga mengirimkkan ACK 1173556808 yaitu sequence number SYN dariServer ditambah 1.
- Baris 3, pada pukul 16:02:53 time stamp 188942
Server mengirim kode ACK 1. 1 merupakan angka relatif setelah persiapan koneksi selesai. ACK 1 berartiserver mengkonfirmasi bahwa paket data no 1 dari *mobile robot* siap diterima oleh server.
- Baris 4, pada pukul 16:02:53 time stamp 235779
Server mengirimkan 7 paket (sequence awal 1 dan sequence akhir 8). Selain mengirimkan data, Server juga mengirimkan signal ACK 1 yang berarti Server siap untuk menerima paket data ke 1 dari *mobile robot*.
- Baris 5, pada pukul 16:02:53 time stamp 242232
Mobile robot memberitahu kepada Server bahwa *mobile robot* sudah siap menerima paket ke 8 (sebelumnya mobile robot sudah menerima 7paket)
- Baris 6, pada pukul 16:02:53 time stamp 261997
Server mengirimkan 6 paket (sequence awal 8 dan sequence akhir 14). Selain mengirimkan data, server juga mengirimkan signal ACK 1 yang berarti Server siap untuk menerima paket data ke 1 dari *mobile robot*.
- Baris 7, pada pukul 16:02:53 time stamp 268412
Mobile robot memberitahu kepada server bahwa *mobile robot* sudah siap menerima paket ke 14 (sebelumnya Client sudah menerima 13 paket).

Delay pengiriman paket dapat diketahui dengan melihat waktu kedatangan tiap paket pada *mobile robot*. Dari data diatas,*delay* antar paket dapat diketahui dengan melihat waktu kedatangan paket pertama dan waktu kedatangan paket kedua. Pada tabel 4.1 baris ke 4, server mengirim paket pertama pada jam 16:02:53.235779. Kemudian pada tabel 4.1 baris ke 5, paket diterima oleh *mobile*

robot pada jam 16:02:53.242232. Selisih waktu paket dikirim dan diterima sebesar 0.006453 detik. Kemudian pada tabel 4.1 baris ke 6, server mengirim paket kedua pada jam 16:02:53.261997. Selisih waktu antara paket pertama dan kedua yang dikirimkan server adalah sebesar 0,026218 detik. Pada tabel 4.1 koneksi pertama untuk pengiriman paket secara TCP terjadi pada jam 16:02:53.182494 dan baris ke 4 menunjukkan paket pertama dikirim pada jam 16:02:53.235779. Selisih waktu saat proses pertama koneksi sampai dengan pengiriman paket pertama sebesar 0.053285 detik.

Setelah koneksi dengan protokol TCP berhasil dilakukan, maka selanjutnya dilakukan pengamatan terhadap nilai koordinat bola yang dikirimkan oleh server dan nilai koordinat bola yang diterima oleh *mobile robot*. Pengamatan ini dilakukan dengan cara melihat nilai koordinat bola yang dikirimkan server, kemudian dicocokkan dengan nilai koordinat bola yang diterima *mobile robot*. Jika nilai koordinat bola yang diterima *mobile robot* sama dengan nilai koordinat bola yang dikirimkan server, maka dalam pengiriman nilai koordinat bola tersebut tidak terjadi adanya *packet loss* atau pun *error rate* pada saat pengiriman. Nilai koordinat bola yang akan dikirim yaitu nilai dari koordinat x dan koordinat y.

Tabel 4.2 merupakan tabel pengiriman nilai koordinat bola dari server kepada *mobile robot*. Pengujian dilakukan sebanyak 20 kali pengiriman nilai koordinat bola dari server menuju *mobile robot*. Tujuan dari pengujian pengiriman nilai ini adalah untuk melihat apakah server dapat mengirimkan nilai koordinat objek bola atau tidak. Selain itu juga untuk melihat apakah data yang dikirimkan server berhasil diterima oleh *mobile robot*. Pengujian ini juga untuk melihat ada atau tidaknya kesalahan dalam proses pengiriman. Jika terjadi perbedaan antara nilai yang dikirimkan server dengan nilai yang diterima *mobile robot* maka terdapat kesalahan dalam proses pengiriman yang dilakukan. Kemudian jika tidak terdapat kesalahan dalam pengiriman maka nilai yang diterima oleh *mobile robot* sama dengan nilai yang dikirimkan oleh server.

Tabel 4.2 Pengujian pengiriman nilai koordinat bola dengan protokol TCP/IP

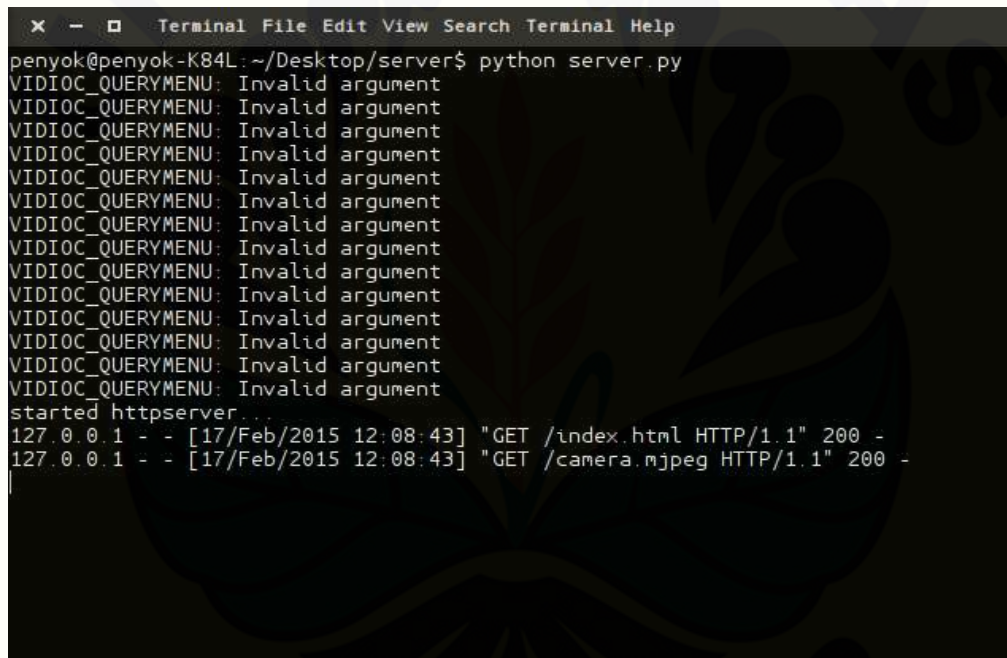
No	Pengirim (server)		Penerima (<i>mobile robot</i>)		Keterangan
	Koordinat X	Koordinat Y	Koordinat X	Koordinat Y	
1	441	177	441	177	Benar
2	362	125	362	125	Benar
3	345	115	345	115	Benar
4	327	89	327	89	Benar
5	281	50	281	50	Benar
6	245	33	245	33	Benar
7	193	85	193	85	Benar
8	159	119	159	119	Benar
9	131	147	131	147	Benar
10	113	203	113	203	Benar
11	107	255	107	255	Benar
12	103	285	103	285	Benar
13	135	313	135	313	Benar
14	187	333	187	333	Benar
15	257	333	257	333	Benar
16	291	355	291	355	Benar
17	347	379	347	379	Benar
18	427	393	427	393	Benar
19	453	392	453	392	Benar
20	489	375	489	375	Benar

Tabel 4.2 merupakan data hasil pengamatan nilai koordinat bola yang dikirimkan oleh server kepada *mobile robot*. Percobaan pengiriman nilai koordinat bola ini dilakukan sebanyak 20 kali. Nilai koordinat bola yang diterima oleh *mobile robot* sama dengan nilai yang dikirimkan oleh server. Hal ini menunjukkan bahwa tidak ada paket yang hilang atau *error* pada saat proses pengiriman.

4.4 Pengujian method GET dan POST pada protokol HTTP

Pengujian ini adalah untuk mengetahui keberhasilan dari method GET dan POST pada server. Method GET terjadi pada saat *user* mengakses halaman website server. Ketika *user* berhasil mengakses video pada server, maka server akan

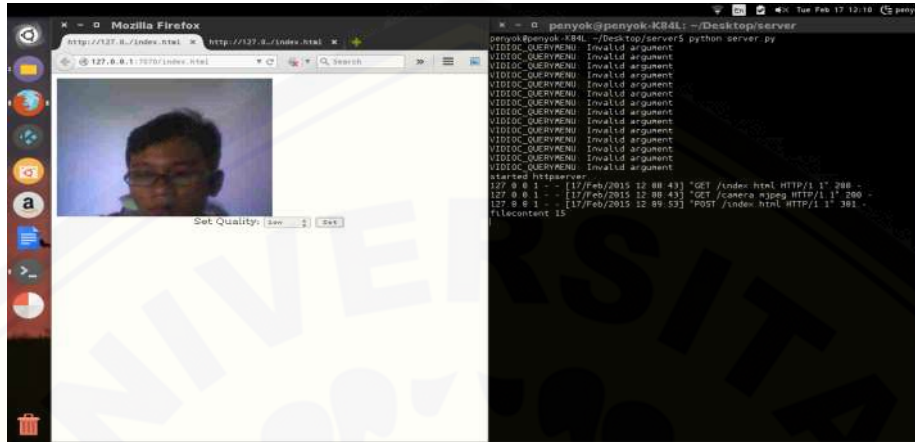
menampilkan pesan pada terminal server bahwa *user* telah berhasil mengakses video. Gambar 4.8 merupakan tampilan pesan yang ditampilkan oleh server ketika *user* berhasil mengakses video. Dari gambar 4.8 dapat diketahui bahwa *user* dengan IP 127.0.0.1 telah berhasil mengakses video dari server. Hal itu ditunjukkan dengan adanya pesan pada terminal server. Pesan yang ditampilkan adalah 127.0.0.1 - - [17/Feb/2015 12:08:43] "GET /index.html HTTP/1.1" 200 - . format pesan yang ditampilkan adalah IP CLIENT - -[tanggal/bulan/tahun jam:menit:detik] "GET /index.html HTTP/1.1" 200 -



```
penyok@penyok-k84L:~/Desktop/server$ python server.py
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
VIDIOC_QUERYMENU: Invalid argument
started httpserver...
127.0.0.1 - - [17/Feb/2015 12:08:43] "GET /index.html HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2015 12:08:43] "GET /camera.mjpeg HTTP/1.1" 200 -
```

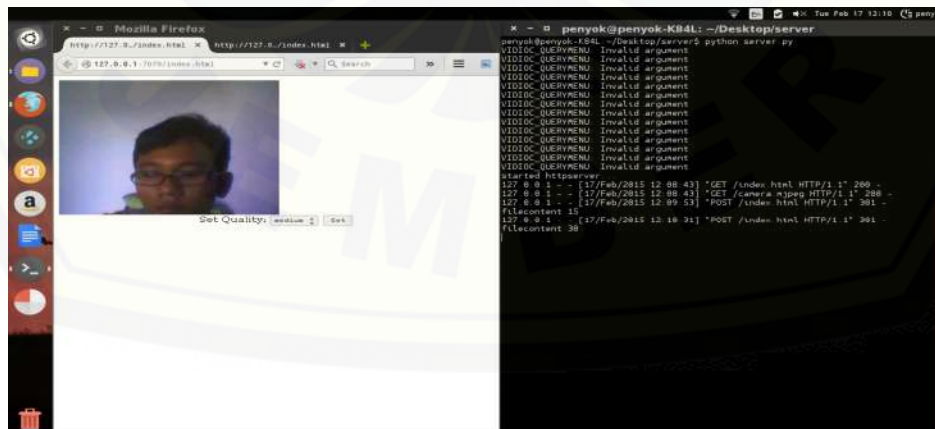
Gambar 4.8 Tampilan Terminal Sisi Server

Kemudin method POST terjadi pada saat *user* memilih kualitas gambar yang ditampilkan pada website. Terdapat 3 pilihan untuk memilih kualitas gambar, yaitu *low*, *medium* dan *high*. Gambar 4.9 merupakan tampilan pada saat *user* memilih kualitas gambar *low*.

Gambar 4.9 Kualitas Gambar *Low*

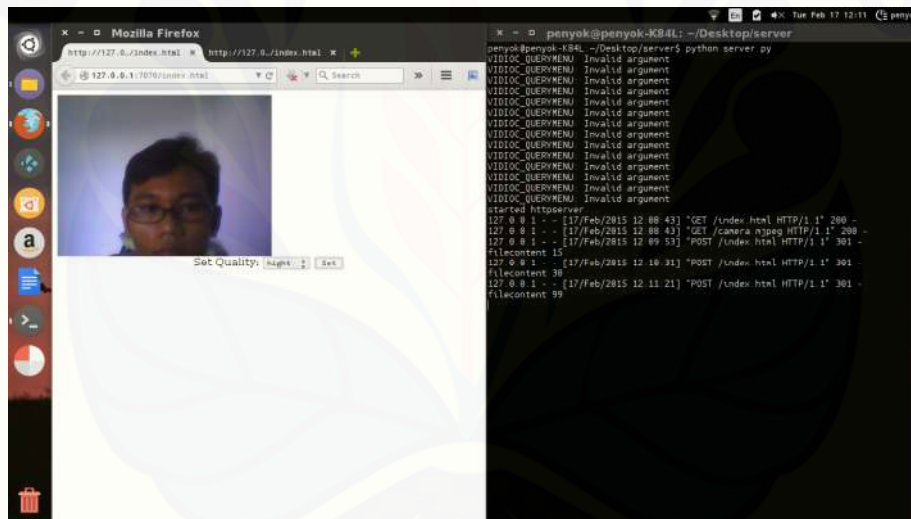
Pada saat *user* memilih kualitas gambar *low*, maka server akan menampilkan pesan bahwa *user* telah melakukan POST untuk memilih kualitas gambar *low*. Pesan ini ditampilkan pada terminal server. Pesan yang ditampilkan adalah `127.0.0.1 - [17/Feb/2015 12:08:53] "POST /index.html HTTP/1.1" 301 - filecontent 15`. Pada open cv untuk kualitas dari gambar memiliki range 2 sampai dengan 99. File content 15 menunjukkan nilai dari kualitas gambar jpeg.

Kemudian method POST pada saat *user* memilih kualitas gambar *medium*. Maka server akan menampilkan pesan pada terminal server bahwa method POST telah dilakukan oleh *user*. Gambar 4.10 merupakan tampilan saat *user* memilih kualitas gambar *medium*.

Gambar 4.10 Kualitas Gambar *Medium*

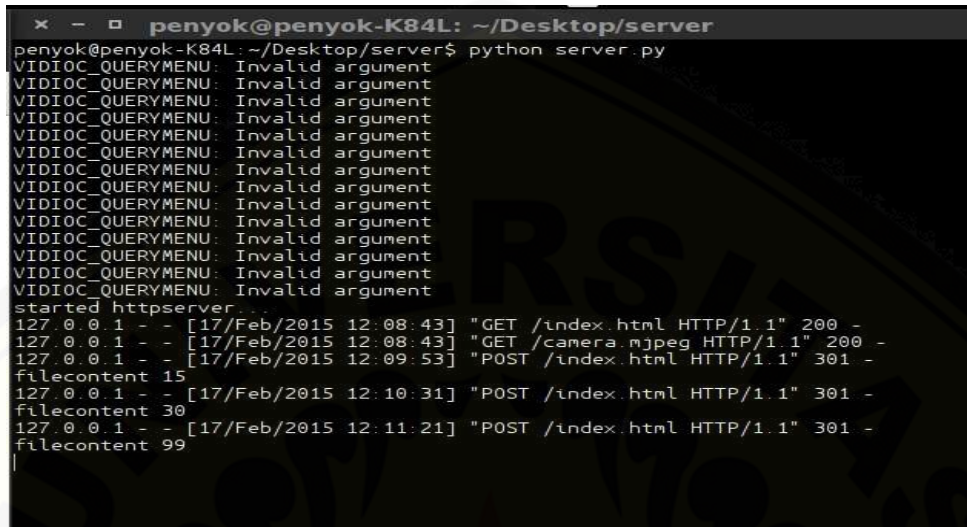
Pada saat *user* melakukan pemilihan kualitas gambar *medium*, maka server akan menampilkan pesan bahwa *user* telah melakukan POST untuk memilih kualitas gambar *medium*. Pesan ini ditampilkan pada terminal server. Pesan yang ditampilkan adalah `127.0.0.1 - [17/Feb/2015 12:10:31] "POST /index.html HTTP/1.1" 301 - filecontent 30`. Pada open cv untuk kualitas dari gambar memiliki range 2 sampai dengan 99. File content 30 menunjukkan nilai dari kualitas gambar jpeg.

Kemudin method POST pada saat *user* memilih kualitas gambar *high*. Maka server akan menampilkan pesan bahwa *user* memilih kualitas gambar *high*. Gambar 4.11 merupakan tampilan dari pada saat *user* memilih kualitas gambar *high*.



Gambar 4.11 Kualitas Gambar High

Pada saat *user* melakukan pemilihan kualitas gambar *high*, maka server akan menampilkan pesan bahwa *user* telah melakukan method POST untuk memilih kualitas gambar *high*. Pesan yang ditampilkan adalah `127.0.0.1 - [17/Feb/2015 12:11:21] "POST /index.html HTTP/1.1" 301 - filecontent 99`. Pada open cv untuk kualitas dari gambar memiliki range 2 sampai dengan 99. File content 99 menunjukkan nilai dari kualitas gambar jpeg.



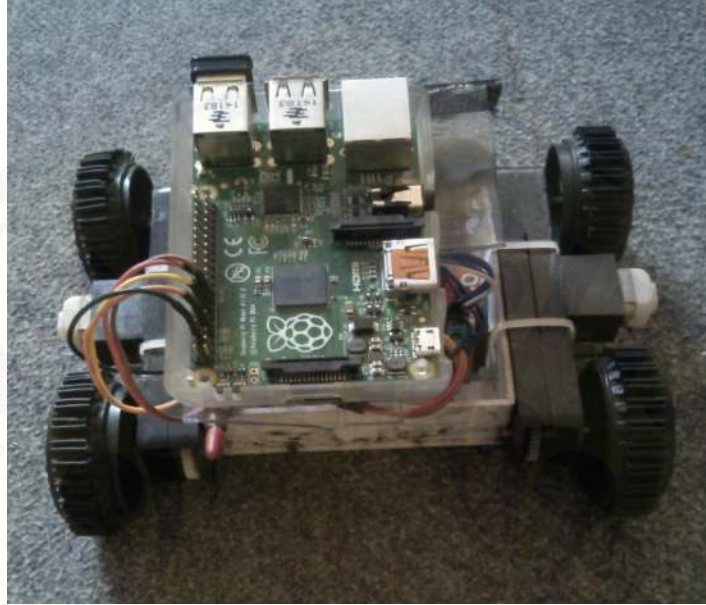
```
penyok@penyok-K84L: ~/Desktop/server
penyok@penyok-K84L:~/Desktop/server$ python server.py
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
VIDI0C_QUERYMENU: Invalid argument
started httpserver...
127.0.0.1 - - [17/Feb/2015 12:08:43] "GET /index.html HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2015 12:08:43] "GET /camera.mjpeg HTTP/1.1" 200 -
127.0.0.1 - - [17/Feb/2015 12:09:53] "POST /index.html HTTP/1.1" 301 -
filecontent 15
127.0.0.1 - - [17/Feb/2015 12:10:31] "POST /index.html HTTP/1.1" 301 -
filecontent 30
127.0.0.1 - - [17/Feb/2015 12:11:21] "POST /index.html HTTP/1.1" 301 -
filecontent 99
```

Gambar 4.12 Method POST Dan GET Terminal Server

Gambar 4.12 merupakan tampilan pesan pada terminal server saat method POST dan GET pada protokol HTTP berlangsung. Method GET menunjukkan bahwa video telah berhasil ditampilkan. Method POST pertama ditampilkan pada saat *user* memilih kualitas gambar *low* (*filecontent* 15). Kemudian method POST yang kedua menunjukkan bahwa *user* memilih kualitas gambar *medium* (*filecontent* 30). Method POST ketiga menunjukkan bahwa *user* memilih kualitas gambar *high* (*filecontent* 90).

4.4 Pengujian Sistem Secara Keseluruhan.


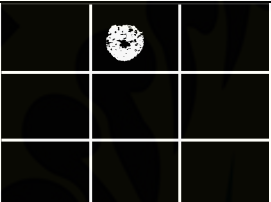

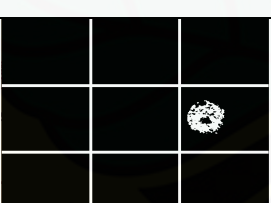
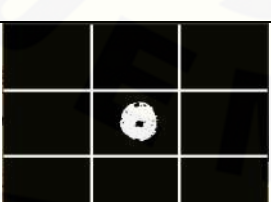
Gambar 4.13 adalah gambar dari *mobile robot*. *Mobile robot* ini terdiri dari raspberry pi sebagai pengendali gerakan *mobile robot* dan untuk menerima dan mengolah nilai koordinat bola yang dikirim server. Perangkat *wireless* berfungsi untuk media koneksi antara *mobile robot* dengan server. Dan driver motor untuk mengendalikan motor. *Mobile robot* terdiri dari satu buah raspberry pi tipe B+, wireless usb dan driver motor.







Gambar 4.13 *Mobile Robot*

Pengujian deteksi bola terhadap aksi *mobile robot* dilakukan untuk mengetahui keberhasilan gerak *mobile robot* terhadap posisi obyek bola yang sedang diamati. Pengujian dilakukan sebanyak 9 posisi pengujian yaitu pojok kiri atas atau kotak no 1, ujung kiri tengah atau kotak no 4, ujung bawah kiri tengah atau kotak no 7, tengah atau kotak no 5, ujung atas tengah atau kotak no 2, ujung bawah tengah atau kotak no 8, ujung kanan tengah atau kotak no 6, ujung atas tengah atau kotak no 3 dan ujung kanan bawah atau kotak no 9. Kesembilan kotak ini digunakan untuk pergerakan *mobile robot*. Tujuan dari mebagi frame menjadi 9 area ini agar mempermudah *user* dalam melakukan navigasi *mobile robot*. Ketika titik tengah bola sudah berada pada area, maka akan ditandai dengan perubahan warna dari garis dari merah menjadi warna putih. Gambar yang ditambahkan pada tabel 4.3 adalah gambar hasil proses segmentasi warna yang telah dilakukan. Data hasil pengujian yang dilakukan diperlihatkan pada tabel 4.3

Tabel 4.3 Pengujian pendeteksian obyek bola terhadap aksi mobile robot

No	Gambar	Nilai Koordinat Objek		Aksi Robot
		Koordinat X	Koordinat Y	
1		149	77	Stop
2		294	91	Maju
3		494	81	Stop
4		489	237	Putar Kanan
5		327	235	Stop

6		115	228	Putar Kiri
7		95	394	Stop
8		299	393	Mundur
9		488	406	Stop

Tabel 4.3 merupakan hasil dari pengujian posisi objek terhadap aksi *mobile robot*. No 1 memberikan perintah kepada *mobile robot* untuk berhenti. No 2 memberikan perintah kepada *mobile robot* untuk gerak maju. No 3 memberikan perintah kepada *mobile robot* untuk berhenti. No 4 memberikan perintah kepada *mobile robot* untuk putar kanan. No 5 memberikan perintah kepada *mobile robot* untuk berhenti. No 6 memberikan perintah kepada *mobile robot* untuk putar kiri. No 7 memberikan perintah kepada *mobile robot* untuk berhenti. No 8 memberikan perintah kepada *mobile robot* untuk gerak mundur. No 9 memberikan perintah kepada *mobile robot* untuk berhenti.

Kemudian setelah pengujian posisi objek terhadap aksi *mobile robot* . Pengujian selanjutnya bertujuan untuk mengetahui keberhasilan gerak *mobile robot*. Pengujian pertama dilakukan di dalam ruangan dengan intensitas cahaya 19,8 lumen. Dalam pengujian ini, objek menggunakan bola warna merah dengan diameter 7 cm. Hasil di dalam ruangan dengan intensitas cahaya 19,8 lumen ditunjukkan pada tabel 4.4.

Tabel 4.4 Hasil pengujian didalam ruangan dengan intensitas cahaya 19,8 lumen.

No	Jarak Kamera Dengan Bola	Aksi Robot			
		Maju	Mundur	Putar Kanan	Putar Kiri
1	50 cm	Gagal	Gagal	Gagal	Gagal
2	100 cm	Gagal	Gagal	Gagal	Gagal
3	200 cm	Gagal	Gagal	Gagal	Gagal
4	300 cm	Gagal	Gagal	Gagal	Gagal
5	400 cm	Gagal	Gagal	Gagal	Gagal
6	500 cm	Gagal	Gagal	Gagal	Gagal
7	600 cm	Gagal	Gagal	Gagal	Gagal
8	700 cm	Gagal	Gagal	Gagal	Gagal

Dari tabel 4.4, robot gagal melakukan aksi maju, mundur, putar kanan dan putar kiri. Hal ini dikarenakan bola tidak dapat dideteksi server. Hal ini dikarenakan kondisi pencahayaan ruangan pada saat pengujian terlalu gelap. Hal ini ditunjukkan dengan tingkat intensitas cahaya sebesar 19,8 lumen.

Pengujian selanjutnya dilakukan di dalam ruangan dengan nilai intensitas cahaya sebesar 47,1 lumen. Hasil dari pengujian yang dilakukan di dalam ruangan dengan nilai intensitas cahaya sebesar 47,1 lumen ditunjukkan pada tabel 4.5.

Tabel 4.5 Hasil pengujian didalam ruangan dengan intensitas cahaya 47,1 lumen.

No	Jarak Kamera Dengan Bola	Aksi Robot			
		Maju	Mundur	Putar Kanan	Putar Kiri
1	50 cm	Berhasil	Berhasil	Berhasil	Berhasil
2	100 cm	Berhasil	Berhasil	Berhasil	Berhasil
3	200 cm	Berhasil	Berhasil	Berhasil	Berhasil
4	300 cm	Berhasil	Berhasil	Berhasil	Berhasil
5	400 cm	Berhasil	Berhasil	Berhasil	Berhasil
6	500 cm	Berhasil	Berhasil	Berhasil	Berhasil
7	600 cm	Gagal	Gagal	Gagal	Gagal
8	700 cm	Gagal	Gagal	Gagal	Gagal

Dari tabel 4.5, *mobile robot* berhasil melakukan aksi maju, mundur, putar kanandan putar kiri pada saat objek berada pada jarak 50-500 cm. Hal ini dikarenakan pada jarak 50-500cm bola dapat terdeteksi oleh server. Kemudian pada jarak 600 cm dan 700 cm bola tidak terdeteksi server, hal ini menyebabkan robot gagal dalam melakukan aksi maju, mundur, putar kanan dan putar kiri.

Pengujian selanjutnya dilakukan diluar ruangan. Intensitas cahaya pada tempat ini adalah 1470 lumen. Hasil pengujian diluar ruangan dengan intensitas cahaya sebesar 1470 lumen ditunjukkan pada tabel 4.6.

Tabel 4.6 Hasil pengujian diluar ruangan dengan intensitas cahaya 1470 lumen.

No	Jarak Kamera Dengan Bola	Aksi Robot			
		Maju	Mundur	Putar Kanan	Putar Kiri
1	50 cm	Berhasil	Berhasil	Berhasil	Berhasil
2	100 cm	Berhasil	Berhasil	Berhasil	Berhasil
3	200 cm	Berhasil	Berhasil	Berhasil	Berhasil
4	300 cm	Berhasil	Berhasil	Berhasil	Berhasil
5	400 cm	Berhasil	Berhasil	Berhasil	Berhasil
6	500 cm	Berhasil	Berhasil	Berhasil	Berhasil
7	600 cm	Berhasil	Berhasil	Berhasil	Berhasil
8	700 cm	Berhasil	Berhasil	Berhasil	Berhasil

Dari tabel 4.6, robot berhasil melakukan aksi maju, mundur, putar kanandan putar kiri pada saat objek berada pada jarak 50-700 cm. Hal ini dikarenakan bola dapat terdeteksi server dan server berhasil mengirimkan nilai koordinat bola.

Pengujian terakhir juga berada diluar ruangan. Intensitas cahaya ditempat ini adalah sebesar 21500 lumen. Hasil dari pengujian yang dilakukan ditunjukkan pada tabel 4.7.

Tabel 4.7 Hasil pengujian diluar ruangan dengan intensitas cahaya 21500 lumen.

No	Jarak Kamera Dengan Bola	Aksi Robot			
		Maju	Mundur	Putar Kanan	Putar Kiri
1	50 cm	Berhasil	Berhasil	Berhasil	Berhasil
2	100 cm	Berhasil	Berhasil	Berhasil	Berhasil
3	200 cm	Berhasil	Berhasil	Berhasil	Berhasil
4	300 cm	Berhasil	Berhasil	Berhasil	Berhasil
5	400 cm	Berhasil	Berhasil	Berhasil	Berhasil
6	500 cm	Berhasil	Berhasil	Berhasil	Berhasil
7	600 cm	Berhasil	Berhasil	Berhasil	Berhasil
8	700 cm	Berhasil	Berhasil	Berhasil	Berhasil

Dari tabel 4.6, robot berhasil melakukan aksi maju, mundur, putar kanandan putar kiri pada saat objek berada pada jarak 50-700 cm. Hal ini dikarenakan bola dapat terdeteksi server dan server berhasil mengirimkan nilai koordinat bola.

4.5 Pengujian *delay* waktu kerja system.

Pengujian ini bertujua untuk mengetahui waktu yang dibutuhkan oleh server untuk memberikan respon ketika *mobile robot* mengirimkan method GET. Tiap Pengujian dilakukan sebanyak 10 kali percobaan. Pada pengujian ini menggunakan variasi jumlah pengiriman method GET, dimulai dengan mengirim 1 method GET kepada server sampai dengan mengirim 10 method GET secara bersamaan. Pengujian ini menggunakan program untuk menghitung respons server dalam melayani method GET dari user. Listing program ini berada pada sisi *mobile robot*. Dibawah ini merupakan listing program yang digunakan.

```

from urllib import urlopen
import time

result = []
sites =
('http://192.168.1.3:7070/index.html', 'http://192.168.1.3:7070
/index.html')
for site in sites :
    start = time.time()
    req = urlopen(site)
    resultappend((site, req.read()))
    end = time.time()
    print "respond %s second" % (end-start)

```

Program diatas menggunakan bahasa pemrograman *python* dengan *library urllib* dan *time*. Program diatas dapat dijalankan dengan perintah *python req.py* pada terminal *mobile robot*. Hasil dari pengujian yang dilakukan ditampilkan pada tabel 4.8.

Tabel 4.8 Hasil pengujian respon server dalam melayani method GET

Jumlah User	Percobaan respon method GET ke -										Rata-rata Respons (s)
	1	2	3	4	5	6	7	8	9	10	
1 User	0.088	0.086	0.087	0.086	0.087	0.087	0.094	0.112	0.095	0.099	0.092
2 User	0.109	0.109	0.108	0.107	0.109	0.122	0.111	0.11	0.108	0.114	0.111
3 User	0.14	0.13	0.137	0.132	0.132	0.13	0.197	0.133	0.141	0.125	0.140
4 User	0.15	0.172	0.153	0.309	0.161	0.144	0.256	0.246	0.151	0.157	0.190
5 User	0.169	0.272	0.213	0.175	0.186	0.231	0.181	0.199	0.176	0.285	0.209
6 User	0.187	0.204	0.2	0.256	0.195	0.198	0.232	0.208	0.187	0.195	0.216
7 User	0.195	0.228	0.214	0.246	0.218	0.224	0.238	0.214	0.206	0.222	0.221
8 User	0.241	0.248	0.239	0.242	0.257	0.261	0.244	0.235	0.226	0.346	0.254
9 User	0.257	0.27	0.248	0.256	0.271	0.258	0.254	0.266	0.265	0.354	0.270
10 User	0.297	0.43	0.286	0.282	0.273	0.282	0.28	0.425	0.297	0.262	0.311

Dari tabel 4.8 dapat dilihat bahwa rata-rata waktu yang dibutuhkan server untuk merespon method GET yang diirinkan oleh *user* semakin besar seiring dengan bertambahnya jumlah *user*. Hal ini dikarenakan server tidak dapat melayani method

GET yang dikirimkan *user* secara bersamaan. Contohnya pada saat ada 10 user yang mengirimkan method GET, maka server akan melayani satu user dahulu dan user yang lain menunggu. Hal inilah yang menyebabkan respon server lebih lama ketika 10 user, dikarenakan server melayani secara bergantian. Selain hal diatas respon juga dipengaruhi oleh kurang optimalnya listing program yang digunakan untuk menampilkan gambar pada *website*.



BAB 5. PENUTUP

5.1 Kesimpulan

Dari hasil penelitian yang telah dilakukan, maka dapat diambil kesimpulan sebagai berikut :

1. Secara keseluruhan sistem dapat berjalan dengan baik sesuai algoritma yang dirancang. (Tabel 4.5 dan Tabel 4.6)
2. Dengan algoritma yang dirancang, protokol TCP pada layer transport berhasil mengirimkan data. (Tabel 4.1 dan Tabel 4.2)
3. *Delay* pengiriman antar paket dengan menggunakan protokol TCP/IP sebesar 0.026 s.
4. Respons server dalam melayani method GET yang dikirimkan oleh 1 *user* sebesar 0.092 s, 2 *user* sebesar 0.111 s, 3 *user* sebesar 0.140 s, 4 *user* sebesar 0.190 s, 5 *user* sebesar 0.209 s, 6 *user* sebesar 0.216 s, 7 *user* sebesar 0.221 s, 8 *user* sebesar 0.254 s, 9 *user* sebesar 0.270 s, 10 *user* sebesar 0.311 s. (Tabel 4.8)

5.2 Saran

1. Sistem ini mempunyai kelemahan pada pengaruh cahaya, dikarenakan dapat merubah warna objek yang dideteksi, objek tidak dapat terdeteksi pada saat kondisi pencahayaan gelap.
2. Untuk pengembangan, dapat ditambahkan kamera pada *mobile robot* dan ditampilkan pada *website* lokal serta pengontrolan kecepatan motor.

DAFTAR PUSTAKA

- Distira, Reda Anggra. 2012. *Desain Sistem Navigasi Robot Dengan Isyarat Mata Menggunakan Metode Canny Dan Hough Transform*. Teknik Elektro Universitas Jember.
- Firmansyah, Iman. 2008. *A Prototype of Web-Based, Modular and Wireless Monitoring Robot*. FMIPA UI
- Grannel, Craig. 2007. *The Essential guide to CSS and HTML Web Design*. Friends of.
- Mufadhol. 2012. *Simulasi Jaringan Komputer Menggunakan Cisco Packet Tracer*. JURNAL TRANSFORMATIKA, Volume, No. 2
- Nielsen, Finn Arup. 2013. *Python programming - Web serving*. Department of Informatics and Mathematical Modelling Technical University of Denmark
- Putra, Samuel Mahatma. 2010. *Analisis Dan Perancangan Aplikasi Monitoring Ip Camera Menggunakan Protokol Http Pada Mobile Phone*. Seminar Nasional Aplikasi Teknologi Informasi 2010 (SNATI 2010). ISSN: 1907-5022
- Sendra, Sandra. 2011. *WLAN IEEE 802.11a/b/g/n Indoor and Interface Performance Study*. International Journal on Advances in Network and Services, Vol 4 No 1 & 2
- Stallings, William. 2010. *Appendix L TCP/IP and OSI*. Cryptography and Network Security, Fifth Edition. ISBN-10: 0136097049
- (https://github.com/abidrahmank/OpenCV-python/tree/master/Other_Examples)
- Diakses pada 11 Februari 2015

LAMPIRAN

1. Menampilkan Gambar Pada Website Lokal

```
import string,cgi,time
from BaseHTTPServer import BaseHTTPRequestHandler, HTTPServer
from SocketServer import ThreadingMixIn
import socket
import cv
import re

def getthresholdedimg(im):
    imgHSV=cv.CreateImage(cv.GetSize(im),8,3)
    cv.CvtColor(im,imgHSV,cv.CV_BGR2HSV)
    imgThreshold=cv.CreateImage(cv.GetSize(im),8,1)
    cv.InRangeS(imgHSV,(0,40,40),(34,100,100),imgThreshold)
    cv.Dilate(imgThreshold,imgThreshold,None,9)
    cv.Erode(imgThreshold,imgThreshold,None,6)
    storage = cv.CreateMemStorage(0)
    contour = cv.FindContours(imgThreshold, storage,
cv.CV_RETR_LIST, cv.CV_CHAIN_APPROX_SIMPLE)
    return imgThreshold

def getpositions(im):

    leftmost=0
    rightmost=0
    topmost=0
    bottommost=0
    temp=0
    for i in range(im.width):
        col=cv.GetCol(im,i)
        if cv.Sum(col)[0]!=0.0:
            rightmost=i
            if temp==0:
                leftmost=i
                temp=1
    for i in range(im.height):
        row=cv.GetRow(im,i)
        if cv.Sum(row)[0]!=0.0:
            bottommost=i
            if temp==1:
                topmost=i
                temp=2

    return (leftmost,rightmost,topmost,bottommost)
capture = cv.CaptureFromCAM(1)
cameraQuality=15
```

```
class MyHandler(BaseHTTPRequestHandler):
#method GET HTTP
    def do_GET(self):
        posx=0
        posy=0
        global cameraQuality
        try:
            self.path=re.sub('[^a-zA-Z0-9]', '',str(self.path))
            if self.path=="" or self.path==None or
self.path[:1]==".":
                return
            #menyiapkan ekstensi .html dlm 1 direktori
            if self.path.endswith(".html"):
                f = open(curdir + sep + self.path)
                self.send_response(200)
                self.send_header('Content-type',      'text/html')
                self.end_headers()
                self.wfile.write(f.read())
                f.close()
                return
            #menyiapkan ekstensi .jpeg dlm 1 direktori
            if self.path.endswith(".jpeg"):
                self.send_response(200)
                self.wfile.write("Content-Type: multipart/x-
mixed-replace; boundary=--aaboundary")
                self.wfile.write("\r\n\r\n")
                while 1:
                    img = cv.QueryFrame(capture)
                    cv.Flip(img, img, 1)
                    imdraw=cv.CreateImage(cv.GetSize(img), 8, 3)
                    imgyellowthresh=getthresholdedimg(img)
                    cv.Erode(imgyellowthresh, imgyellowthresh, None, 1)

                    (leftmost, rightmost, topmost, bottommost)=getpositions(imgyellowthr
esh)
                    if (leftmost-rightmost!=0) or (topmost-
bottommost!=0):
                        lastx=posx
                        lasty=posy
                        posx=cv.Round((rightmost+leftmost)/2)
                        posy=cv.Round((bottommost+topmost)/2)

                        if lastx!=0 and lasty!=0:

                            cv.Line(img, (posx, posy), (lastx, lasty), cv.CV_RGB(0, 255, 0), 5)

                            cv.Circle(img, (posx, posy), 5, cv.CV_RGB(0, 255, 0), -1)
```



```
        x = int (lastx) - int (posx)
        y = int (lasty) - int (posy)

cv2mat=cv.EncodeImage(".jpeg",img, (cv.CV_IMWRITE_JPEG_QUALITY,cameraQuality))

        JpegData=cv2mat.tostring()
        self.wfile.write("--aaboundary\r\n")
        self.wfile.write("Content-Type:
image/jpeg\r\n")
        self.wfile.write("Content-length:
"+str(len(JpegData))+"\r\n\r\n" )
        self.wfile.write(JpegData)
        time.sleep(0.01)
        return
    if self.path.endswith(".jpeg"):
        f = open(curdir + sep + self.path)
        self.send_response(200)
        self.send_header('Content-type','image/jpeg')
        f.close()
        return
    return
except IOError:
    self.send_error(404,'File Not Found: %s' % self.path)
def do_POST(self):
    global rootnode, cameraQuality
    try:
        ctype, pdict =
cgi.parse_header(self.headers.getheader('content-type'))
        if ctype == 'multipart/form-data':
            query=cgi.parse_multipart(self.rfile, pdict)
            self.send_response(301)

            self.end_headers()
            upfilecontent = query.get('upfile')
            print "filecontent", upfilecontent[0]
            value=int(upfilecontent[0])
            cameraQuality=max(2, min(99, value))
            self.wfile.write("<HTML>POST OK. Camera Set
to<BR><BR>");
            self.wfile.write(str(cameraQuality));
            print upfile
            #print value
        except :
            pass

class ThreadedHTTPServer(ThreadingMixIn, HTTPServer):
    #class ThreadedHTTPServer(HTTPServer):
```

```
        """Handle requests in a separate thread."""

def main():
    try:
        server = ThreadedHTTPServer(('127.0.0.1', 7070),
MyHandler)
        print 'started httpserver...'
        server.serve_forever()
    time.time (10)
    except KeyboardInterrupt:
        print '^C received, shutting down server'
        server.socket.close()

if __name__ == '__main__':
    main()
```

2. Program Index.html

```
<HTML>
<BODY>

<div id="hiddenDiv" style="display:none;visible:hidden;">
  <iframe src="hidden.html" height="1" width="1" border="0"
scrolling="no" name="hiddenFrame" id="hiddenFrame" >< /iframe>
</div>



<form target="hiddenFrame" method='POST' enctype='multipart/form-
data'>

  <div align="center">
    Set Quality:
    <select name="upfile" width="50">
      <option value="15"> low </option>
      <option value="30"> medium </option>
      <option value="99"> hight </option>
    </select>
    <input type=submit value=Set>
  </div>
</form>

</BODY>
</HTML>
```

3. Program Untuk Mengirim Nilai Koordinat Bola.

```
import cv
import time
import socket
#setup GPIO pin
posx=0
posy=0
def getthresholdedimg(im):
    imgHSV=cv.CreateImage(cv.GetSize(im),8,3)
    cv.CvtColor(im,imgHSV,cv.CV_BGR2HSV)
    imgthreshold=cv.CreateImage(cv.GetSize(im),8,1)
    cv.InRangeS(imgHSV,(115,100,98),(130,100,100),imgthreshold)
    cv.Dilate(imgthreshold,imgthreshold, None, 9)
    cv.Erode(imgthreshold, imgthreshold, None, 6)
    storage = cv.CreateMemStorage(0)
    contour = cv.FindContours(imgthreshold, storage,
cv.CV_RETR_LIST, cv.CV_CHAIN_APPROX_SIMPLE)
    return imgthreshold
def getpositions(im):
    leftmost=0
    rightmost=0
    topmost=0
    bottommost=0
    temp=0
    for i in range(im.width):
        if cv.Sum(col)[0]!=0.0:
            rightmost=i
            if temp==0:
                leftmost=i
                temp=1
    for i in range(im.height):
        if cv.Sum(row)[0]!=0.0:
            bottommost=i
            if temp==1:
                topmost=i
                temp=2
    return (leftmost,rightmost,topmost,bottommost)
capture=cv.CaptureFromFile("http://127.0.0.1:7070/camera.mjpeg")
frame =cv.QueryFrame(capture)
test =cv.CreateImage(cv.GetSize(frame),8,3)
server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.connect(('192.168.1.1', 2020))
while True:
    frame=cv.QueryFrame(capture)
    cv.Flip(frame,frame,1)
```

```
imdraw=cv.CreateImage(cv.GetSize(frame),8,3)
imgyellowthresh=getthresholdedimg(frame)
cv.Erode(imgyellowthresh,imgyellowthresh,None,1)
(leftmost,rightmost,topmost,bottommost)=getpositions(imgyellow
thresh)
if (leftmost-rightmost!=0) or (topmost-bottommost!=0):
    lastx=posx
    lasty=posy
    posx=cv.Round((rightmost+leftmost)/2)
    posy=cv.Round((bottommost+topmost)/2)
    if lastx!=0 and lasty!=0:

cv.Line(frame,(posx,posy),(lastx,lasty),cv.CV_RGB(0,255,0),5)
    cv.Circle(frame,(posx,posy),5,cv.CV_RGB(0,255,0),-
1)

    x = int (posx)
    y = int (posy)
    listed = [x,y]
    telem = ','.join(map(str, listed))
    print 'x = ' + str(x) + ' y = ' + str(y)
    server.send(telem)
#cv.Add(test,imdraw,test)
#cv.Add(test,frame,test)
#cv.ShowImage("Object Tracking",imdraw)
#cv.ShowImage("RGB", frame)

if cv.WaitKey(10) == 27:
    break
cv.DestroyWindow("output")
```


4. Program Pada Raspberry Pi

```
import sys
import socket
from subprocess import call
import os
import RPi.GPIO as GPIO
import time

server = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
server.bind(('0.0.0.0', 2020))
server.listen(1)
conn, addr = server.accept()
print "\nSrver Started"
print 'Conect: ', addr
data = 'nothing'
GPIO.setmode(GPIO.BOARD)
while True:
    if (data == 'quit') or (data == ''):
        print "\nClosing Socket"
        GPIO.cleanup()
        server.close()
        break
    data = conn.recv(1024)
    #print 'RECV : ', data
    if data :
        got = data.split(',')
        x = int (got[0])
        y = int (got[1])
        print "x = " + str(x) + "   y = " + str(y)
        if (0 < y < 160)  & (213 < x < 426):

            GPIO.setup(7,GPIO.OUT)
            GPIO.setup(11,GPIO.OUT)
            GPIO.setup(13,GPIO.OUT)
            GPIO.setup(15,GPIO.OUT)
            GPIO.output(7,True)    #//motor A maju
            GPIO.output(11,False)
            GPIO.output(13,False) #//motor A maju
            GPIO.output(15,True)
            #print "maju"
            #time.sleep (1.0)
        elif (320 < y < 480) & (213 < x < 426):
            GPIO.setup(7,GPIO.OUT)
            GPIO.setup(11,GPIO.OUT)
            GPIO.setup(13,GPIO.OUT)
```

```
GPIO.setup(15,GPIO.OUT)
GPIO.output(7, False)
GPIO.output(11,True)  #//motor A maju
GPIO.output(13,True)
GPIO.output(15,False)
#print "mundur"
#time.sleep (1.0)
elif (160 < y < 320) & ( 0 < x < 213):
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
GPIO.output(7,False)  #//motor A maju
GPIO.output(11,False)
GPIO.output(13,False) #//motor A maju
GPIO.output(15,True)
#print "kanan"
#time.sleep (1.0)
elif (160 < y < 320) & (426 < x < 640):
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
GPIO.output(7,True)  #//motor A maju
GPIO.output(11,False)
GPIO.output(13,False) #//motor A maju
GPIO.output(15,False)
#print "kiri"
#time.sleep (1.0)
else:
GPIO.setup(7,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
GPIO.setup(15,GPIO.OUT)
GPIO.output(7,False)  #//motor A maju
GPIO.output(13,False)
GPIO.output(11,False) #//motor A maju
GPIO.output(15,False)
#print "stop"
#time.sleep (1.0)

sys.exit()
```