



**PERANCANGAN *MONITORING AND CONTROLLING TRAFFIC LIGHT*  
PADA *DIFFERENT STREET CONDITION* MENGGUNAKAN  
JARINGAN INTERNET**

**SKRIPSI**

**Oleh :**

**Cries Avian  
NIM 121910201033**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**



**PERANCANGAN *MONITORING AND CONTROLLING TRAFFIC LIGHT*  
PADA *DIFFERENT STREET CONDITION* MENGGUNAKAN  
JARINGAN INTERNET**

**SKRIPSI**

diajukan guna melengkapi tugas akhir dan memenuhi salah satu syarat untuk menyelesaikan Program Pendidikan S1 Teknik Elektro dan mencapai gelar Sarjana Teknik

Oleh :

**Cries Avian  
NIM 121910201033**

**PROGRAM STUDI STRATA 1 TEKNIK ELEKTRO  
JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS JEMBER  
2016**

## PERSEMBAHAN

*Proyek akhir ini merupakan sebuah awal, langkah kecil menuju lompatan besar guna menggapai kesuksesan yang lebih baik lagi. Untuk itu saya ucapkan rasa syukur dan terima kasih sebesar-besarnya kepada ...*

*Allah SWT, dengan segala Keagungan dan Keajaiban-Nya yang senantiasa mendengar do'a ku, menuntunku dari dari kegelapan, serta senantiasa menaungiku dengan rahmat dan hidayah-Nya dan junjunganku Nabi Besar Muhammad SAW yang telah menjadi penerang di dunia dan suri tauladan bagi kita semua;*

*Bunda Suti'ah, Ayahanda Suswanto, Adekku Chella Tiarani terima kasih atas segala kasih sayang, dukungan, semangat, dan doa selama ini;*

*Seluruh teman dan sahabat seperjuangan Teknik Elektro angkatan 2012, kalian sebagai tempat berbagi suka dan duka yang tidak akan terlupakan. Aku menjadikan kalian semua bagian dari diriku dan aku sangat menyayangi kalian semua;*

*Seluruh Dosen Jurusan Teknik Elektro yang terhormat, terima kasih telah banyak memberikan ilmu dan mendidik dengan penuh kesabaran;*

*Buat semua teman – teman Teknik Elektro semua angkatan, Serta semua pihak yang belum tertulis dalam lembar persembahan ini, Terima kasih atas segalanya;*

***Almamater Fakultas Teknik Universitas Jember.***

MOTTO

“Demi masa. Sesungguhnya manusia itu benar-benar dalam kerugian, kecuali orang-orang yang beriman dan mengerjakan amal saleh dan nasehat menasehati supaya mentaati kebenaran dan nasehat menasehati supaya menetapi kesabaran”

(QS: Al Ashr 1-3)

“Jika kamu menolong agama Allah, maka Allah akan menolongmu dan meneguhkan kedudukanmu di Bumi”

(QS: Muhammad 7)

“Tidak perlu menjadi professor untuk menjadi orang bodoh “

(Habib Rizieq, 2015)

“Salah satu cara termudah untuk menyelesaikan masalah adalah dengan menyelesaikan masalah orang lain”

(Cries Avian)

“Tergila - gilalah pada ilmu, jangan tergila - gila pada harta, karena harta hanya akan membawa kepada keindahan dunia, sedangkan ilmu akan membawa dirimu menuju keindahan surga “

(Cries Avian)

## PERNYATAAN

Saya yang bertanda tangan di bawah ini:

Nama : Cries Avian

NIM : 121910201033

Menyatakan dengan sesungguhnya bahwa proyek akhir yang berjudul: *“Perancangan Monitoring And Controlling Traffic Light Pada Different Street Condition Menggunakan Jaringan Internet”* adalah benar-benar hasil karya sendiri, kecuali jika dalam pengutipan substansi disebutkan sumbernya dan belum pernah diajukan pada institusi mana pun 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 di kemudian hari pernyataan ini tidak benar.

Jember, 29 Juni 2016

Yang menyatakan,

Cries Avian

NIM 121910201033

**TUGAS AKHIR**

**PERANCANGAN *MONITORING AND CONTROLLING TRAFFIC LIGHT*  
PADA *DIFFERENT STREET CONDITION* MENGGUNAKAN  
JARINGAN INTERNET**

Oleh  
Cries Avian  
NIM 121910201033

Pembimbing :

Dosen Pembimbing Utama : Widjonarko, S.T., M.T.

Dosen Pembimbing Anggota : Satryo Budi Utomo, S.T., M.T.

## LEMBAR PENGESAHAN

Laporan proyek akhir berjudul “*Perancangan Monitoring And Controlling Traffic Light Pada Different Street Condition Menggunakan Jaringan Internet*” oleh Cries Avian NIM: 121910201033 telah diuji dan disahkan oleh Fakultas Teknik Elektro Universitas Jember pada;

Hari : Rabu  
Tanggal : 29 Juni 2016  
Tempat : R. Sidang 2 Lantai III Dekanat Fakultas Teknik

Pembimbing Utama,

Widjonarko, S.T., M.T.  
NIP 197109081999031001

Pembimbing Anggota,

Satryo Budi Utomo  
NIP 198501262008011002

Penguji I,

Bambang Supeno, S.T., M.T.  
NIP 196906301995121001

Penguji II,

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

Mengesahkan  
Dekan,

Dr.Ir. Entin Hidayah M.U.M  
NIP. 19661215 199503 2 001

Perancangan *Monitoring and Controlling Traffic Light* pada *Different Street Condition* Menggunakan Jaringan Internet (*Design of Monitoring and Controlling Traffic Light on Different Street Condition Using Internet Network* )

**Cries Avian**

**Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember**

## **ABSTRAK**

Kemacetan lalu lintas menjadi permasalahan yang sering kita jumpai di berbagai wilayah di Indonesia terutama di kota – kota besar dengan beraneka ragam penyebabnya. Di antara penyebab yang sering kita jumpai adalah ketidaksiplinan dari pengguna jalan, selain itu pengaturan waktu nyala lampu lalu lintas juga sangat berpengaruh. Pada tugas akhir ini dirancanglah sebuah alat yang mampu melakukan monitoring kepadatan jalan serta mampu mengontrol waktu nyala dari lampu lalu lintas sesuai dengan skenario dan dapat diakses dengan menggunakan jaringan internet. Untuk proses monitoring alat ini menggunakan kamera *webcam*, sedangkan untuk mengendalikan lama waktu lampu lalu lintas digunakanlah *Relay* yang berfungsi sebagai pengganti saklar untuk mengendalikan PLC yang menjadi kontrol utama dari lampu lalu lintas. Semua aktivitas monitoring dan *controlling* akan diproses dan ditampilkan pada *web interface* dengan menggunakan Rapsberry Pi sebagai komponen pemroses utama. Dari hasil pengujian, didapatkan waktu tunda rerata atau delay pada alat tidak lebih dari 150 ms, diantaranya 3 tampilan kamera bersamaan sebesar 0,006934s untuk kamera 1, 0,006012s untuk kamera 2 dan 0,005486s untuk kamera 3. Sedangkan untuk waktu tunda kamera 1 sebesar 0,143407s, kamera 2 sebesar 0,145678s dan kamera 3 sebesar 0,140676s. Sedangkan untuk kontrol *relay* didapatkan waktu tunda tombol 1 sebesar 0,031789s, tombol 2 memiliki waktu tunda sebesar 0,069339s, tombol 3 memiliki waktu tunda sebesar 0,041896s dan tombol 4 memiliki waktu tunda sebesar 0,050381s. Dengan demikian maka dapat dikatakan bahwa pengujian pada waktu tunda rerata pada proses *monitoring* dan *controlling*, dapat dikatakan baik sesuai dengan standart ITU G.114. Sedangkan untuk kendali dari *Relay* suara, keberhasilan mencapai 100% dari 20 percobaan.

**Kata kunci : *Monitoring, Controlling, Traffic Light, Web, Internet***



Perancangan *Monitoring and Controlling Traffic Light* pada *Different Street Condition* Menggunakan Jaringan Internet (*Design of Monitoring and Controlling Traffic Light on Different Street Condition Using Internet Network* )

**Cries Avian**

**Jurusan Teknik Elektro, Fakultas Teknik, Universitas Jember**

## **ABSTRACT**

*The Traffic jams become so many problem that frequently encountered in various regions in Indonesia, especially in cities and big cities with different causes. Among the causes that had often encountered is unndiscipline of road users. In addition to the timing of the traffic lights are also very influential. In this final project we should designed a device that have a capabled of monitoring the density of the road and be able to control the timing of traffic lights flame in accordance with the scenario and can be accessed by using the Internet networks. For process of the monitoring tool is using the web camera, whereas for a long time to control traffic lights that the function of Relay is used as a replacement switch for controlling the PLC which is the main control of traffic lights. The Monitoring and controlling all activities will be processed and displayed on the web interface by using the Raspberry Pi as a component of the main processor. Monitoring and controlling all activities will be processed and displayed on the web interface using the Raspberry Pi as a component of the main processor. From the test results, we obtained an average delay time or delay on the appliance is not more than 150 ms. Including three camera views simultaneously for 0,006934s of camera 1, camera 2 0,006012s and 0,005486s for the camera 3. As for the time delay camera 1 by 0,143407s ,camera 2 for 0,145678s and camera 3 of 0,140676s. As for the control relay delay time obtained by 0,031789s button 1, button 2 has a time delay of 0,069339s, the key 3 has a time delay of 0,041896s and button 4 has a delay time of 0,050381s. Thus it can be said that testing on the average time delay in the process of monitoring and controlling, can be said to be good according to the ITU G.114 standard. And then for sound relay's controlling, we have succeded of 100% from 20 experiment*

**Keywords : Monitoring, Controlling, Traffic Light, Web, Internet**

## RINGKASAN

**“Perancangan *Monitoring and Controlling Traffic Light* pada *Different Street Condition* Menggunakan Jaringan Internet”**; Cries Avian 121910201033; 2016: 85 halaman; Program Studi Strata 1 (S1) Teknik , Jurusan Teknik Elektro, Fakultas Teknik Universitas Jember.

Kemacetan lalu lintas menjadi permasalahan yang sering kita jumpai di berbagai wilayah di Indonesia terutama di kota – kota besar dengan beraneka ragam penyebabnya. Di antara penyebab yang sering kita jumpai adalah ketidaksiplinan dari pengguna jalan, selain itu pengaturan waktu nyala lampu lalu lintas juga sangat berpengaruh. Pada tugas akhir ini dirancanglah sebuah alat yang mampu melakukan monitoring kepadatan jalan serta mampu mengontrol waktu nyala dari lampu lalu lintas sesuai dengan skenario dan dapat diakses dengan menggunakan jaringan internet. Pada alat ini sistem *monitoring* terletak pada fungsi pengawasan yang dilakukan dengan cara melihat gambar yang berasal dari webcam dan dikirim melalui internet pada *web interface*. Sedangkan fungsi kontrol terletak pada proses perintah perpindahan aktif dan mati pada *relay*. *Relay* ini berfungsi untuk mengontrol *input* dari PLC dan *relay* yang digunakan untuk mengontrol suara.

Pengujian dilakukan dengan cara melihat waktu tunda yang dihasilkan oleh fungsi *monitoring* dimana pengujian pada kali ini dilakukan untuk mengecek dua mode *monitoring*, yaitu ketika gambar ditampilkan secara bersamaan dan ketika gambar ditampilkan secara satu persatu pada *web interface*. Sedangkan untuk pengujian kedua adalah pengujian fungsi kontrol dimana untuk menguji fungsi ini dilakukan dengan cara menekan tombol yang berada *web interface* dan dilihat seberapa lama respon yang terjadi sampai merubah kondisi *relay*. Kemudian untuk pengujian selanjutnya adalah pengujian kontrol *relay* suara dimana pada pengujian kali ini adalah untuk menguji berapa kali keberhasilan *relay* suara merespon perubahan perintah yang dikirim melalui *web interface*.

## PRAKATA

*Bismillahirrohmanirrohim*

Puji syukur ke hadirat Allah SWT atas segala rahmat dan hidayah-Nya sehingga laporan proyek akhir yang “*Perancangan Monitoring And Controlling Traffic Light Pada Different Street Condition Menggunakan Jaringan Internet*” dapat terselesaikan dengan baik. Laporan tugas akhir ini disusun untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan Strata 1 (S1) pada Jurusan Teknik Elektro Fakultas Teknik Universitas Jember.

Terselesaikannya laporan tugas akhir ini tidak terlepas dari bantuan berbagai pihak, oleh karena itu disampaikan ucapan terima kasih kepada:

1. Allah Swt, Tuhan pencipta alam semesta yang telah melimpahkan rahmat dan rizki-Nya serta memberi pertolongan sehingga tugas akhir ini dapat saya selesaikan dengan lancar dan mudah atas izin – Nya.
2. Shalawat serta salam kepada Nabi baginda Muhammad SAW yang telah menjadi suri tauladan terbaik bagi umatnya serta yang telah menjadi penghubung antara manusia dengan Rabb pecipta alam semesta untuk mendapatkan tuntunan yang diharapkan oleh Dia.
3. Ayah (Suswanto) dan Ibu (Suti'ah) serta Adekku Chella Tiarani yang saya cintai, terima kasih atas doa dan dukungannya sehingga anakmu ini mampu meraih gelar yang selama ini ia cita – citakan
4. Terimakasih kepada program Bidikmisi, alhamdulillah dengan program ini saya dapat mewujudkan keinginan saya untuk terus belajar dan terus berbakti kepada negeri dengan ilmu dan karya yang telah saya dapatkan saat di masa perkuliahan ini.
5. Ibu Dr. Ir. Entin Hidayah M.U.M selaku Dekan Fakultas Teknik Universitas Jember.
6. Bapak Widjonarko, A.Md., S.T., M.T.. selaku dosen pembimbing utama dan Bapak Satryo Budi Utomo, ST., MT. selaku dosen pembimbing anggota yang telah meluangkan waktu dan pikiran guna memberikan bimbingan dan pengarahan dalam penyusunan tugas akhir ini.

7. Bapak Bambang Supeno., S.T., M.T.. selaku dosen penguji satu dan Bapak Dodi Setio Budi, ST., MT. selaku dosen pembimbing dua yang telah memberi masukan kepada saya sehingga saya dapat menciptakan sebuah karya berupa tugas akhir menjadi lebih baik dan lebih baik lagi.
8. Seluruh guru – guru saya dan dosen – dosen saya yang telah memberikan ilmunya kepada saya, beribu – ribu ucapan terimakasih atas ilmu yang selama ini telah diberikan kepada saya, sehingga tangga – tangga kesuksesan kini telah tersusun dan menjulang semakin tinggi menuju puncak cita – cita saya. Saya hanya bisa mengucapkan terimakasih dan berdoa agar Allah memberikan pahala jariyah yang tak pernah terputus atas ilmu yang sangat amat bermanfaat ini.
9. Keluarga besar Teknik Elektro khususnya angkatan 2012 SATE UJ, terimakasih atas dukungan dan motivasi yang kalian berikan.
10. Teman – teman seperjuangan dan keluarga besar Habib Mbul, master Awang, Badrul, *soulmate* saya Sandi, Prof Rizaldy, Ilham, Sabta, Agam, Umam, Rahmanu, Linda dan Arif yang selalu mendukung, membantu dan momen yang sangat berharga selama ini ketika menjalani masa kuliah sampai terselesaikannya tugas akhir ini, kenangan dan pengalaman tak akan pernah terlupakan.
11. Kepada Dewi Nurhaji Meivita, terimakasih atas omelan-nya yang membuat keluarga besar Habib Mbul semakin semangat dalam mengerjakan skripsinya dan juga Aisyatin Masluhah yang selama ini selalu menjadi malaikat terbaik bagi saya.
12. Kepada seluruh keluarga besar Laboratorium Kendali, terimakasih atas dukungannya dan ilmunya yang telah diberikan selama ini.
13. Terimakasih kepada musyrif tercinta dan teman – teman seperjuangan yang tidak bisa saya sebutkan satu persatu namanya.
14. Semua pihak yang tidak dapat saya sebutkan satu persatu, terima kasih atas dukungan dan motivasi kalian dalam penyusunan proyek akhir ini.

Penulis menyadari bahwa sebagai manusia biasa tidak terlepas dari keterbatasan, yang biasanya akan mewarnai kadar ilmiah dari tugas akhir ini. Oleh karena itu penulis selalu terbuka terhadap masukan dan saran dari semua pihak yang sifatnya membangun untuk mendekati kesempurnaan. Tidak lupa penulis menyampaikan permohonan maaf yang sebesar-besarnya jika terdapat kesalahan dan kekeliruan. Akhir kata penulis berharap laporan ini dapat memberikan manfaat bagi pembaca dan dapat menjadi bahan acuan yang bermanfaat di kemudian hari.

Jember, 29 Juni 2016

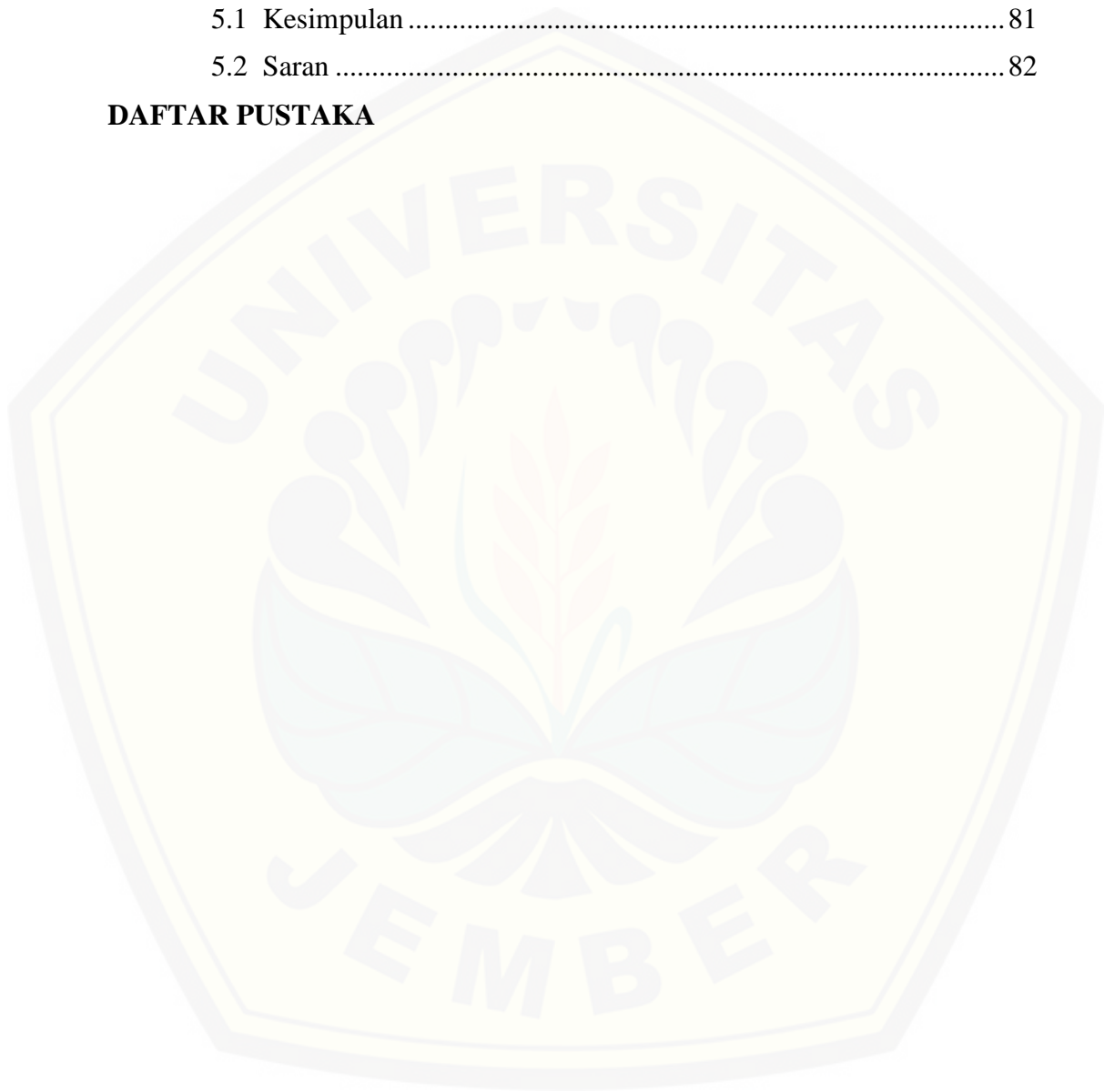
Penulis

DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	<b>i</b>
<b>HALAMAN PERSEMBAHAN</b> .....	<b>ii</b>
<b>HALAMAN MOTTO</b> .....	<b>iii</b>
<b>HALAMAN PERNYATAAN</b> .....	<b>iv</b>
<b>HALAMAN PEMBIMBING</b> .....	<b>v</b>
<b>HALAMAN PENGESAHAN</b> .....	<b>vi</b>
<b>ABSTRAK</b> .....	<b>vii</b>
<b>ABSTRACT</b> .....	<b>viii</b>
<b>RINGKASAN</b> .....	<b>ix</b>
<b>KATA PENGANTAR</b> .....	<b>x</b>
<b>DAFTAR ISI</b> .....	<b>xiii</b>
<b>DAFTAR GAMBAR</b> .....	<b>xvi</b>
<b>DAFTAR TABEL</b> .....	<b>xviii</b>
<b>BAB 1. PENDAHULUAN</b> .....	<b>1</b>
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	3
1.4 Tujuan .....	3
1.5 Manfaat .....	3
<b>BAB 2. TINJAUAN PUSTAKA</b> .....	<b>5</b>
2.1 Rapsberry PI .....	5
2.2 Webcam .....	8
2.3 PLC Zelio SR2 B201 BD <i>Smart Relay</i> .....	9
2.4 <i>Relay</i> .....	12
2.5 <i>Driver Relay</i> .....	15
2.6 HTML .....	16
2.7 PHP .....	17
2.8 QOS ( <i>Quality of Service</i> ) .....	17

2.8 Wireshark .....	18
<b>BAB 3. METODE PENELITIAN.....</b>	<b>21</b>
3.1 Tempat dan Waktu Penelitian.....	21
3.2 Alat dan Bahan.....	21
3.2.1 Alat .....	21
3.2.2 Bahan.....	21
3.3 Perancangan Sistem .....	22
3.3.1 Diagram Blok .....	22
3.3.2 <i>Flow chart</i> .....	24
3.4 Cara Kerja .....	25
3.5 Skenario .....	26
3.5.1.1 Skenario Pertama .....	27
3.5.1.2 Skenario Kedua.....	28
3.5.1.3 Skenario Ketiga.....	29
3.6 Desain <i>Web Interface</i> .....	30
3.6 Pengaturan IP Rapsberry Pi.....	31
3.7 Membangun Jaringan <i>Streaming</i> Kamera.....	32
3.8 Membangun Kontrol Utama .....	36
3.9 Membangun Kontrol Suara.....	38
3.10 Membangun Miniatur <i>Traffic Light</i> .....	39
3.11 Analisa Sistem dengan Wireshark .....	41
<b>BAB 4. HASIL DAN ANALISA DATA .....</b>	<b>43</b>
4.1 Pengukuran <i>Delay</i> Kamera .....	43
4.2 Sistem Kontrol Utama .....	56
4.2.1 Pengukuran pada Raspberry Pi .....	56
4.2.2 Pengecekan <i>Relay</i> .....	56
4.2.3 Pengukuran Tegangan pada PLC.....	57
4.2.4 Pengujian Kontrol Lampu Lalu Lintas .....	58
4.2.5 Pengujian Waktu Tunda Kontrol <i>Relay</i> .....	63
4.2.6 Pengujian Sistem.....	68
4.3 Pengujian Miniatur Lampu Lintas .....	74

4.4 Pengujian Waktu Lampu Lintas .....	77
4.5 Pengujian <i>Relay</i> Suara .....	80
<b>BAB 5. PENUTUP.....</b>	<b>81</b>
5.1 Kesimpulan .....	81
5.2 Saran .....	82
<b>DAFTAR PUSTAKA</b>	





## DAFTAR GAMBAR

Gambar 2.1 Bentuk dari Rapsberry Pi .....	5
Gambar 2.2 Diagram blok Rapsberry PI.....	6
Gambar 2.3 Bagian dari Raspberry PI .....	7
Gambar 2.4 Bagian fungsi dari GPIO dari Raspberry PI.....	7
Gambar 2.5 Webcam .....	8
Gambar 2.6 Bentuk PLC Zelio SR2 B210 BD .....	9
Gambar 2.7 Tampilan Awal Software ZelioSoft 2 .....	10
Gambar 2.8 Pilih PLC .....	11
Gambar 2.9 Pilih Bahasa Pemrograman .....	11
Gambar 2.10 Buka atau Buat Program Ladder .....	12
Gambar 2.11 Proses Flashing ke PLC .....	12
Gambar 2.12 Bentuk – bentuk Relay .....	13
Gambar 2.13 Struktur sederhana Relay .....	14
Gambar 2.14 Gambar rangkaian driver Relay .....	15
Gambar 2.15 Tampilan Wireshark .....	19
Gambar 2.16 Struktur Wireshark .....	20
Gambar 3.1 Diagram blok sistem kontrol dan monitoring .....	22
Gambar 3.2 Diagram blok sistem monitoring dan pengendali pada remote.....	22
Gambar 3.3 Flow chart sistem kontrol utama .....	24
Gambar 3.4 Flow chart PLC .....	25
Gambar 3.5 Skenario perempatan .....	26
Gambar 3.6 Desain Web Interface .....	31
Gambar 3.7 Tampilan Kamera pada Web Interface.....	36
Gambar 3.8 Konfigurasi Pemasangan Pin .....	37
Gambar 3.9 Skema Kontrol Suara .....	39
Gambar 3.10 Rangkaian LED Traffic Light .....	40
Gambar 3.11 Menu Utama Wireshark .....	41

Gambar 3.12 Gambar Hasil Capture.....	42
Gambar 4.1 Tampilan Software Wireshark .....	43
Gambar 4.2 Kamera dalam Keadaan 3 Tampilan .....	44
Gambar 4.3 Grafik Waktu Tunda untuk ke 3 Kamera dalam 1 Tampilan.....	48
Gambar 4.4 Tampilan pada Kamera 1 .....	49
Gambar 4.5 Tampilan pada Kamera 1 (lanjutan).....	50
Gambar 4.6 Grafik Waktu Tunda pada Kamera 1 .....	50
Gambar 4.6 Tampilan pada Kamera 2 .....	51
Gambar 4.7 Grafik Waktu Tunda untuk Kamera 2.....	52
Gambar 4.8 Tampilan pada Kamera 3 .....	53
Gambar 4.9 Grafik Waktu Tunda pada Kamera 3 .....	55
Gambar 4.10 Mode Monitoring .....	61
Gambar 4.11 Grafik Waktu Tunda Tombol.....	67
Gambar 4.12 Kondisi Jalan untuk Mode 1 ke Mode 2 .....	68
Gambar 4.13 Kondisi Jalan untuk Mode 1 ke Mode 3 .....	71
Gambar 4.14 Kondisi Jalan untuk Mode 2 ke Mode 3 .....	73
Gambar 4.15 Port Input Miniatur Jalan.....	76
Gambar 4.16 Miniatur Jalan.....	76
Gambar 4.17 Pengujian Nyala LED.....	77
Gambar 4.18 Mode Monitoring Timer.....	78

**DAFTAR TABEL**

Tabel 2.1 Parameter <i>Delay</i> berdasarkan ITU-T G.114 .....	18
Tabel 3.1 Skenario Sistem <i>Timing Diagram</i> Mode 1 .....	28
Tabel 3.2 Skenario <i>Timing Diagram</i> Sistem Mode 2 .....	29
Tabel 3.3 Skenario <i>Timing Diagram</i> Sistem Mode 3 .....	29
Tabel 4.1. Waktu Tunda untuk Kamera 1 pada Tampilan 3 Kamera .....	45
Tabel 4.2. Waktu Tunda untuk Kamera 2 pada Tampilan 3 Kamera .....	46
Tabel 4.3. Waktu Tunda untuk Kamera 3 pada Tampilan 3 Kamera .....	47
Tabel 4.4. Waktu Tunda untuk Kamera 1 .....	49
Tabel 4.5. Waktu Tunda untuk Kamera 1 (lanjutan) .....	50
Tabel 4.6. Waktu Tunda untuk Kamera 2 .....	51
Tabel 4.7. Waktu Tunda untuk Kamera 2 (Lanjutan) .....	52
Tabel 4.8. Waktu Tunda untuk Kamera 3 .....	53
Tabel 4.9. Waktu Tunda untuk Kamera 3 (Lanjutan) .....	54
Tabel 4.10 Pengukuran Tegangan pada Pin GPIO Rapsberry Pi.....	56
Tabel 4.11 Pengecekan Fungsi Relay pada Port Keluaran Relay .....	57
Tabel 4.12 Pengukuran Tegangan pada Port I/O PLC .....	58
Tabel 4.13 Tabel Pengujian Kerja Relay .....	59
Tabel 4.14 Tabel Pengujian Kerja Relay Tombol 1.....	61
Tabel 4.15 Tabel Pengujian Kerja Relay Tombol 2.....	61
Tabel 4.16 Tabel Pengujian Kerja Relay Tombol 3.....	62
Tabel 4.17 Tabel Pengujian Kerja Relay Secara Acak .....	62
Tabel 4.18 Tabel Pengukuran Delay Tombol 1 .....	64
Tabel 4.19 Tabel Pengukuran Delay Tombol 2 .....	64
Tabel 4.20 Tabel Pengukuran Delay Tombol 2 (lanjutan).....	65
Tabel 4.21 Tabel Pengukuran Delay Tombol 3 .....	65
Tabel 4.22 Tabel Pengukuran Delay Tombol 3 (Lanjutan) .....	66
Tabel 4.23 Tabel Pengukuran Delay Tombol 4 .....	66

Tabel 4.24 Tabel Pengujian Putaran Skenario 1 Menuju Skenario 2 .....	69
Tabel 4.25 Tabel Waktu Tunda Putaran Skenario 1 ke 2 dan 2 ke 1 .....	69
Tabel 4.26 Tabel Waktu Tunda Putaran Skenario 1 ke 2 dan 2 ke 1 (Lanjutan) ...	70
Tabel 4.27 Tabel Pengujian Putaran Skenario 1 Menuju Skenario 3 .....	71
Tabel 4.28 Tabel Waktu Tunda Putaran Skenario 1 ke 3 dan 3 ke 1 .....	71
Tabel 4.29 Tabel Waktu Tunda Putaran Skenario 1 ke 3 dan 3 ke 1 (Lanjutan) ...	72
Tabel 4.30 Tabel Pengujian Putaran Skenario Kedua Menuju Skenario Ketiga ...	73
Tabel 4.31 Tabel Waktu Tunda Putaran Skenario 2 ke 3 dan 3 ke 2 .....	73
Tabel 4.32 Tabel Waktu Tunda Putaran Skenario 2 ke 3 dan 3 ke 2 (Lanjutan) ...	74
Tabel 4.33 Pengujian Respon LED terhadap Tegangan .....	75
Tabel 4.34 Pengujian Timer pada Mode 1 .....	78
Tabel 4.35 Pengujian Timer pada Mode 2 .....	79
Tabel 4.36 Pengujian Timer pada Mode 3 .....	79
Tabel 4.37 Tabel Pengujian Relay Suara .....	80

## BAB 1. PENDAHULUAN

### 1.1 Latar Belakang

Kemacetan lalu lintas, merupakan masalah yang sering kita jumpai terutama di kota – kota besar di Indonesia. Kemacetan adalah keadaan tersendat atau terhentinya lalu lintas yang disebabkan oleh banyaknya jumlah kendaraan melebihi kapasitas jalan dan kurang disiplinnya pengguna jalan dalam berkendara. Daerah yang sering terjadi kemacetan seperti di jalan – jalan raya yang sangat padat lalu lintasnya misal persimpangan jalan baik persimpangan tiga ataupun persimpangan empat, sering kita jumpai lampu lalu lintas yang secara garis besar lampu lalu lintas atau *traffic light* ini berfungsi untuk dapat mengurai kemacetan serta menertibkan kendaraan-kendaraan yang berada pada persimpangan jalan. Lampu lalu lintas (menurut UU no. 22/2009 tentang lalu lintas dan angkutan jalan : alat pemberi isyarat lalu lintas atau APILL) adalah lampu yang mengendalikan arus lalu lintas yang terpasang di persimpangan jalan, tempat penyebrangan jalan kaki (*zebra cross*), dan tempat arus lalu lintas lainnya (Republik Indonesia, 2009).

Berdasarkan fungsi lampu lalu lintas atau *traffic light* diatas, maka dapat dikatakan bahwa salah satu instrumen paling penting dalam kelancaran lalu lintas terletak pada penyalaaan lampu lalu lintas. Pengaturan lampu lalu lintas di persimpangan jalan utamanya di kota – kota besar masih banyak menggunakan sistem pengaturan waktu tetap (*fixed time signals*) yaitu sistem kontrol dengan menggunakan suatu program yang telah ditentukan terlebih dahulu dengan referensi waktu yang dibuat menggunakan komponen diskrit yang dirangkai di atas papan rangkaian kemudian digabung dengan komponen tambahan seperti *timer*, *Relay* dan lainnya yang pengaturannya hanya berlaku pada tiap-tiap lampu lalu lintas yang ada pada persimpangan itu saja (Suryadi, 2009). Pada sistem lampu lalu lintas di atas, petugas lalu lintas tidak dapat mengendalikan secara manual untuk penyalaaan lampu lalu lintas jika kondisi dan keadaan lalu lintas membutuhkan penanganan khusus seperti mengatur arus lalu lintas atau kendaraan jika terjadi kemacetan di persimpangan atau melancarkan perjalanan kendaraan yang sifatnya penting (*urgent*) misalnya kendaraan ambulance, pemadam kebakaran dan iring-iringan

kendaraan pejabat (Asri, 2010). Dalam perkembangannya di Indonesia, penggunaan teknologi dalam mengatasi permasalahan transportasi khususnya di persimpangan telah dilakukan seperti dengan penerapan *Area Traffic Control Sistem (ATCS)* dan *Intelligence Transport Sistem (ITS)*. Yaitu dengan menggunakan pengaturan lampu lalu lintas menggunakan *SMS gateway* (Joko Triyono, 2010). Namun, teknologi tersebut masih difokuskan pada pengaturan pengguna jalan dan lalu lintasnya dan belum mengakomodir kebutuhan akan informasi kondisi jalan secara *real time*, sehingga pengatur lampu lalu lintas terlebih dahulu harus berada di lokasi untuk dapat memberikan keputusan perpindahan pengaturan waktu lampu lalu lintas yang seakan cenderung lamban (Wijayanto, 2015).

Berdasarkan permasalahan diatas, maka disusun sebuah perancangan *monitoring and controlling traffic light* pada *different street condition* menggunakan jaringan internet sebagai pengembangan dari teknologi sistem informasi dan *monitoring* alat pemberi isyarat lalu lintas (SIMAPILL). Perancangan SIMAPILL kali ini dirancang agar dapat memberikan informasi kondisi jalan secara *real time* yang diinformasikan melalui tangkapan kamera yaitu berupa gambar jalan dan sebuah tombol yang digunakan untuk mengatur nyala lampu lalu lintas dengan cara mengakses alamat tertentu pada jaringan internet secara *real time*, sehingga proses pengambilan keputusan APILL dapat dilakukan dengan cepat.

## 1.2 Rumusan Masalah

Berdasarkan uraian yang terdapat pada latar belakang di atas, maka dalam rumusan masalah ini adalah :

1. Bagaimana cara merancang alat yang mampu memonitoring keadaan jalan secara *real time*.
2. Bagaimana cara merancang alat yang mampu mengontrol lampu lalu lintas sesuai dengan deskripsi yang telah ditentukan oleh sistem secara *real time*.
3. Bagaimana cara merancang alat yang mampu memberikan suara kepada pengendara yang tidak disiplin dengan berbantuan alat tanpa langsung terjun menuju lapangan yang dikendalikan nyala

### 1.3 Batasan Masalah

Mengacu pada rumusan masalah, peneliti membuat prototipe pengatur lampu lalu lintas dengan menggunakan internet dengan batasan – batasan masalah yang akan dilakukan penelitian, sebagai berikut :

1. Dalam pentransmision data, media yang digunakan adalah jaringan internet.
2. Dalam pengendalian lampu lalu lintas, peneliti menggunakan PLC dengan tipe PLC Zelio SR B201BD.
3. Dalam lampu lalu lintas yang dirancang, peneliti meniadakan lampu kuning dikarenakan keterbatasan *input* dan *output* dari PLC.
4. Dalam prototipe kali ini, peneliti membuat skenario dengan 4 arah arus lalu lintas.
5. Peneliti menggunakan 3 *Webcam* sebagai kamera pengawas lalu lintas.
6. Dalam pengontrolan lama waktu nyala lampu, dikendalikan melalui internet.
7. Dalam deskripsi atau mode lama lampu menyala, peneliti hanya menggunakan 4 deskripsi tugas.
8. Tidak memperhitungkan perhitungan lama nyala lampu sesuai dengan kondisi aktual.

### 1.4 Tujuan

Adapun tujuan dari penelitian ini adalah membuat suatu pengontrol purwarupa pengatur lampu lalu lintas dengan menggunakan internet serta mampu memberikan instruksi suara kepada pengguna jalan yang diberikan oleh operator.

### 1.5 Manfaat

Manfaat yang ingin dicapai dalam melakukan penelitian ini adalah :

1. Digunakan untuk penerapan ilmu elektronika terutama di bidang transportasi.
2. Digunakan sebagai alat bantu untuk proses pengaturan lalu lintas oleh polisi lalu lintas.

3. Digunakan sebagai riset bagi teknologi – teknologi pengatur transportasi yang memanfaatkan dunia elektronika sebagai alternatif penyelesaian masalah.

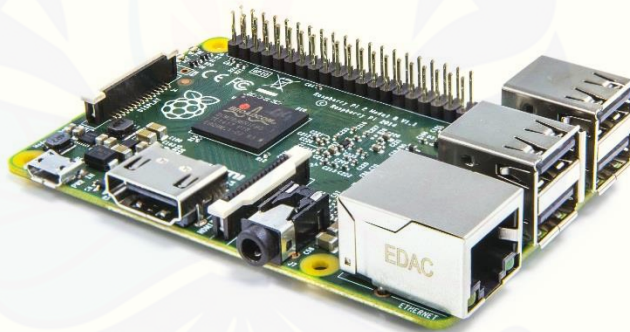




## BAB 2. TINJAUAN PUSTAKA

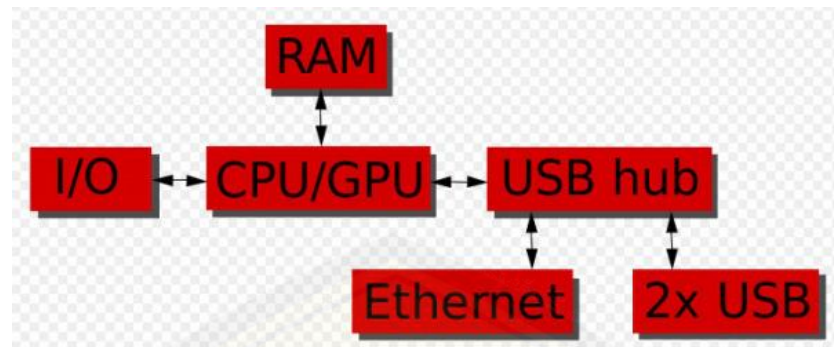
### 2.1 Rapsberry PI

Raspberry Pi adalah modul mikrokomputer yang memiliki *input output digital port* seperti pada *board* mikrokontroler. Diantara kelebihan Rapsberry Pi dibanding dengan *board* mikrokontroler adalah pada Rapsberry mempunyai *Port / koneksi* untuk *display* berupa TV atau Monitor PC serta koneksi USB untuk *keyboard* serta *mouse*. Jadi Raspberry Pi bisa dikatakan lebih mirip sebuah modul mikrokomputer yang memiliki kemampuan yang hampir sama dengan mikrokontrol. Raspberry Pi dibuat di Inggris oleh sebuah perusahaan yang bernama Raspberry Pi Foundation yang sebenarnya pada awal pembuatan dari Raspberry Pi ditunjukkan untuk modul pembelajaran ilmu komputer di sekolah.



Gambar 2.1 Bentuk dari Rapsberry Pi  
([www.raspberrypi.org](http://www.raspberrypi.org))

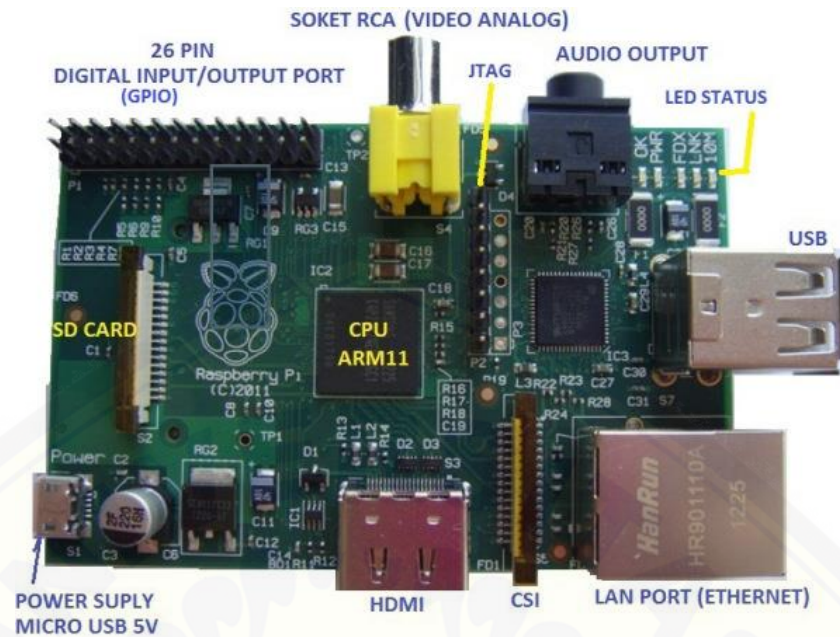
Raspberry Pi *board* dibuat dengan 2 tipe yang berbeda, yaitu di antaranya adalah Rapsberry Pi dengan tipe A dan Rpsberry Pi dengan tipe B. Perbedaannya antara lain terletak pada RAM dan *port* LAN. Tipe A memiliki RAM = 256 Mb dan tanpa *port* LAN (ethernet), sedangkan tipe B memiliki RAM = 512 Mb dan terpasang *port* untuk LAN. Berikut ini blok diagram dari Raspberry Pi pada Gambar 2.2.



Gambar 2.2 Diagram blok Rapsberry PI  
(<https://pccontrol.wordpress.com>)

Raspberry PI *board* mempunyai *input* dan *output* antara lain:

1. HDMI, dihubungkan ke LCD TV yang mempunyai *port* HDMI atau dengan kabel *converter* HDMI to VGA dapat dihubungkan ke monitor PC.
2. Video analog (RCA port), dihubungkan ke televisi sebagai alternatif jika tidak memiliki monitor PC .
3. Audio output
4. 4 buah port USB digunakan untuk *keyboard* dan *mouse*
5. 40 pin I/O digital
6. CSI port (*Camera Serial Interface*)
7. DSI (*Display Serial Interface*)
8. LAN *port* (*network*)
9. SD Card *slot* untuk SD Card memori yang menyimpan sistem operasi dan berfungsi seperti harddisk pada PC.



Gambar 2.3 Bagian dari Raspberry PI  
 (<https://pccontrol.wordpress.com>)

Sedangkan untuk GPIO yang merupakan sederet pin yang terdiri dari 26 pin memiliki konfigurasi dengan berbagai fungsi di antaranya:

P1: The Main GPIO connector						
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin
		3.3v	1 2	5v		
8	Rv1:0 - Rv2:2	SDA	3 4	5v		
9	Rv1:1 - Rv2:3	SCL	5 6	0v		
7	4	GPIO7	7 8	TxD	14	15
		0v	9 10	RxD	15	16
0	17	GPIO0	11 12	GPIO1	18	1
2	Rv1:21 - Rv2:27	GPIO2	13 14	0v		
3	22	GPIO3	15 16	GPIO4	23	4
		3.3v	17 18	GPIO5	24	5
12	10	MOSI	19 20	0v		
13	9	MISO	21 22	GPIO6	25	6
14	11	SCLK	23 24	CE0	8	10
		0v	25 26	CE1	7	11
WiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	WiringPi Pin

Gambar 2.4 Bagian fungsi dari GPIO dari Raspberry PI  
 ([http://wikipedia.co.id/wiring\\_pi](http://wikipedia.co.id/wiring_pi))

Selain sebagai *input output* pada beberapa pin GPIO juga berfungsi sebagai komunikasi serial diantaranya I2C, SPI dan serial komunikasi UART.

## 2.2 Webcam

*Webcam* adalah kamera video sederhana berukuran relatif kecil yang sering digunakan untuk konferensi video jarak jauh atau sebagai kamera pemantau.

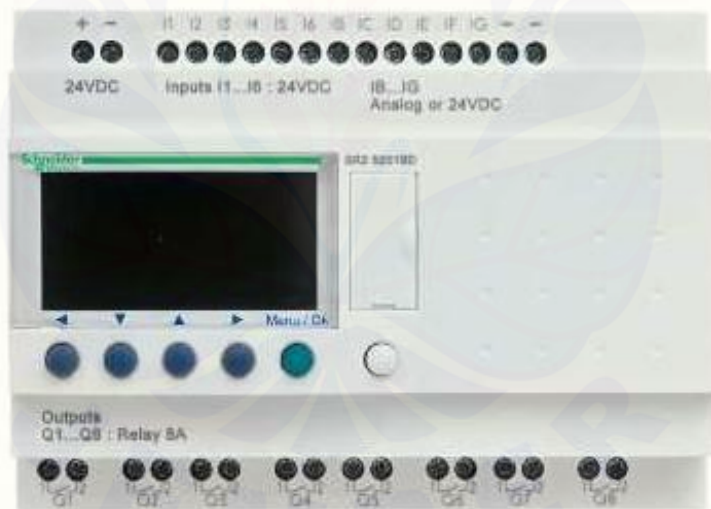


Gambar 2.5 Webcam  
(<http://idkf.bogor.net>)

Definisi yang lain tentang *Webcam* adalah sebuah periferan berupa kamera sebagai pengambil citra/gambar dan mikrofon ( *optional* ) sebagai pengambil suara/audio yang dikendalikan oleh sebuah komputer atau oleh jaringan komputer. Gambar yang diambil oleh *Webcam* ditampilkan ke layar monitor, karena dikendalikan oleh komputer maka ada *interface* atau *port* yang digunakan untuk menghubungkan *Webcam* dengan komputer atau jaringan. Ada beberapa orang mengartikan *Webcam* sebagai *Web pages + Camera*, karena dengan menggunakan *Webcam* untuk mengambil gambar video secara aktual bisa langsung di *upload* bila komputer yang mengendalikan terkoneksi internet.

### 2.3 PLC (*Programmable Logic Controller*) Zelio SR2 B201 BD *Smart Relay*

PLC (*Programmable Logic Controller*) Zelio SR2 B201 BD merupakan *smart Relay* yang terdiri dari 12 *input*, 8 *output* dan 16 *timer Relay*. *Smart Relay* adalah suatu alat yang dapat diprogram oleh bahasa tertentu yang biasa digunakan pada proses *automasi*. *Smart Relay* memiliki ukuran yang kecil dan relatif ringan. *Zelio logic smart Relay* memang didesain untuk otomasi sistem yang biasa digunakan pada aplikasi industri dan komersial. Untuk keperluan industri biasanya digunakan untuk aplikasi *small finishing*, *packaging* dan juga proses produksi. Selain itu juga digunakan untuk mesin-mesin yang berskala kecil sampai dengan yang berskala besar dan terkadang juga digunakan untuk *home industry*. Untuk sektor komersial atau bangunan biasanya digunakan untuk alat penggulung, pintu masuk, instalasi listrik, *compressor* dan lain-lain yang menggunakan sistem otomasi (Arief, 2015).



Gambar 2.6 Bentuk PLC Zelio SR2 B210 BD  
(<https://ariefeeiiiggeennblog.wordpress.com>)

Untuk spesifikasi dari PLC Zelio SR2 B210 BD adalah sebagai berikut

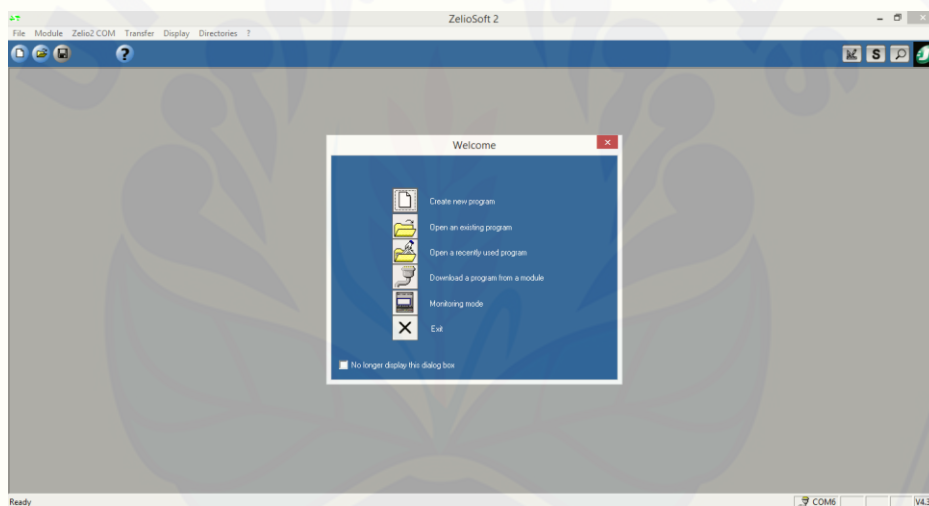
1. PLC bekerja dengan tegangan 24 VDC.
2. Terdapat 12 *port input* dengan pembagian *port* I1 – I6 adalah *port* I/O biasa dengan tegangan masukan sebesar 24 VDC dan *port* IB – IG adalah *port*

*analog* dengan masukan tegangan *analog* dengan tegangan berupa tegangan *analog* atau tegangan 24VDC

3. Terdapat 8 *port output* dengan arus maksimum yang melewati kontaktor sebesar 8A.
4. Terdapat 16 kontaktor *timer* dengan 11 mode *timer* dan 16 kontaktor *counter*.

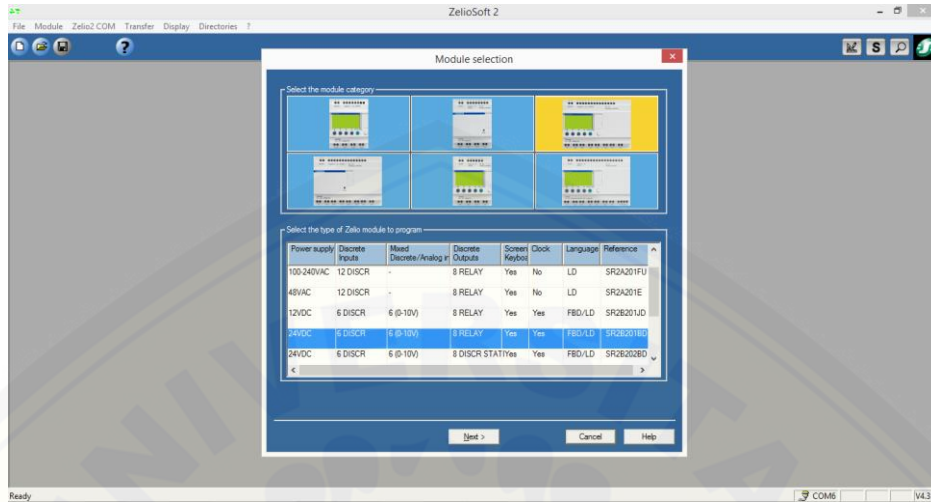
Untuk dapat memprogram PLC Zelio SR2 B201 BD, dapat digunakan *software* Zelio Soft 2 yang merupakan *software* bawaan dari PLC ini. Untuk langkah – langkah pemrograman atau *flashing* ke PLC adalah sebagai berikut

1. Memilih “*Create New Program*” untuk membuat program baru.



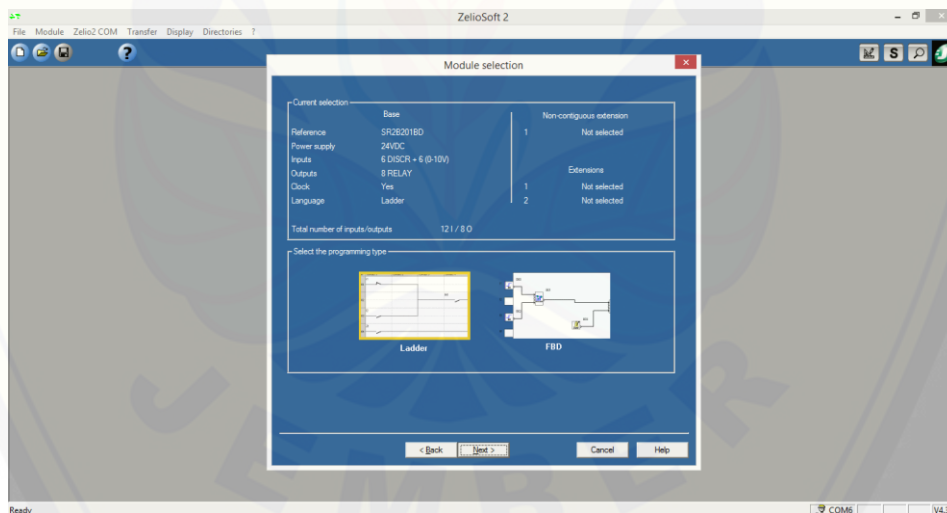
Gambar 2.7 Tampilan Awal *Software* ZelioSoft 2

2. Memilih model PLC yang akan digunakan yaitu SR2 B201 BD.



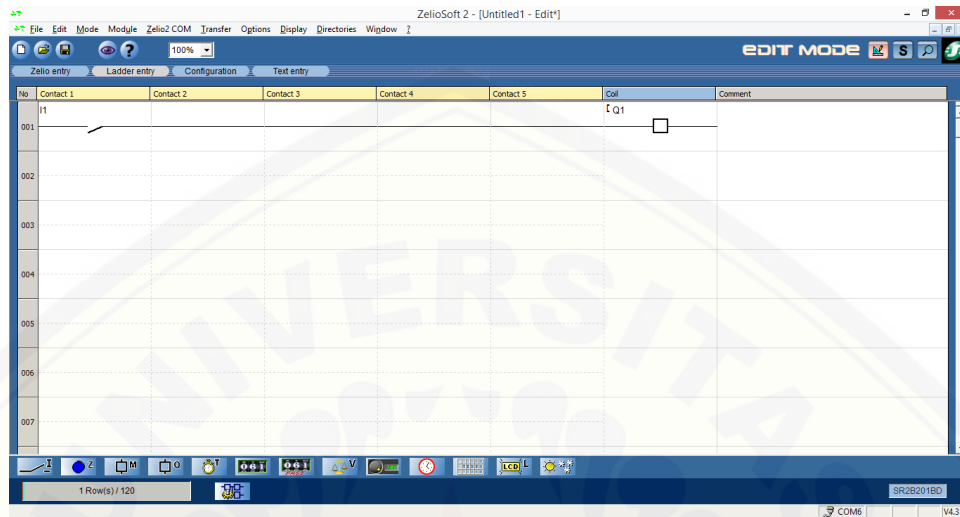
Gambar 2.8 Pilih PLC

3. Mengelik *Next* lalu memilih Ladder.



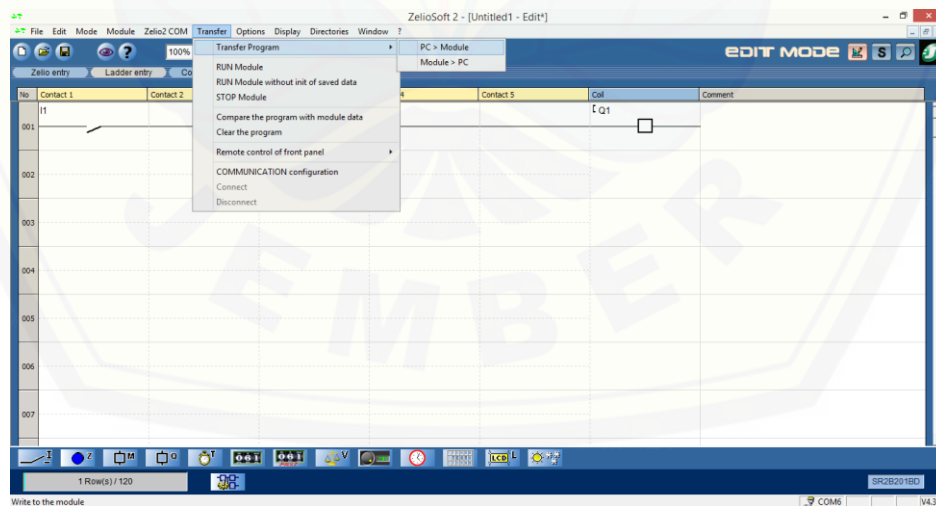
Gambar 2.9 Pilih Bahasa Pemrograman

#### 4. Membuat atau Membuka Ladder.



Gambar 2.10 Buka atau Buat Program Ladder

#### 5. Memilih menu pada *toolbar transfer*, *transfer program*, *PC to Module* lalu menekan Ok maka proses *upload* atau *flashing* bekerja.



Gambar 2.11 Proses *Flashing* ke PLC



## 2.4 Relay

*Relay* adalah saklar (*switch*) yang dioperasikan secara listrik dan merupakan komponen *electromechanical* (elektromekanikal) yang terdiri dari 2 bagian utama yakni elektromagnet (*coil*) dan mekanikal (seperangkat kontak saklar/*switch*). *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *Relay* yang menggunakan elektromagnet 5V dan 50 mA mampu menggerakkan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A (Arief, 2015).

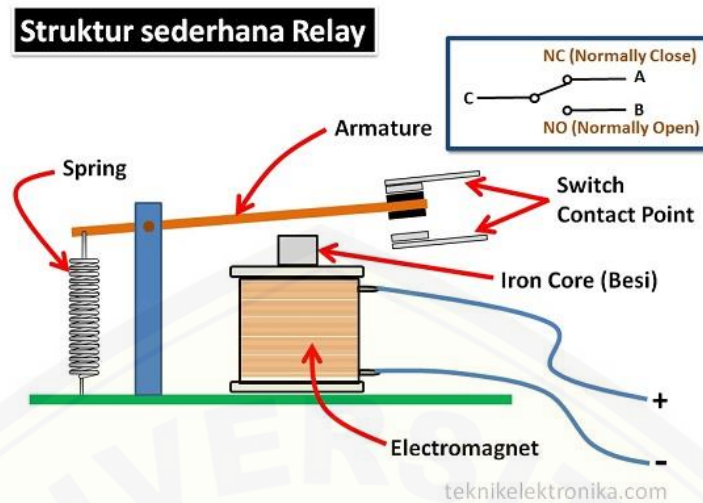


Gambar 2.12 Bentuk – bentuk *Relay*  
(<http://teknikelektronika.com/pengertian-Relay-fungsi-Relay/>)

Pada dasarnya, *Relay* terdiri dari 4 komponen dasar yaitu :

1. *Electromagnet (Coil)*
2. *Armature*
3. *Switch Contact Point (Saklar)*
4. *Spring*

Berikut ini merupakan gambar dari bagian-bagian *Relay* :



Gambar 2.13 Struktur sederhana *Relay*  
 (<http://teknikelektronika.com/pengertian-Relay-fungsi-Relay/>)

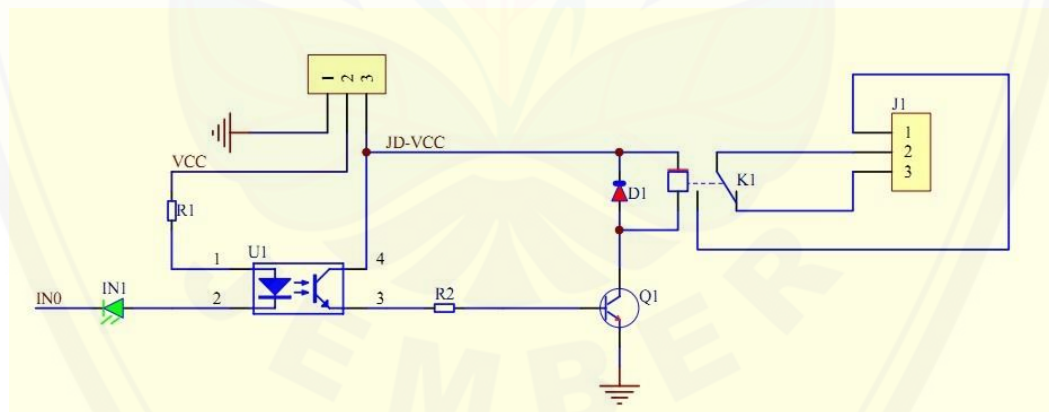
Kontak poin (*contact point*) *Relay* terdiri dari 2 jenis yaitu :

- *Normally Close* (NC) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *CLOSE* (tertutup)
- *Normally Open* (NO) yaitu kondisi awal sebelum diaktifkan akan selalu berada di posisi *OPEN* (terbuka)

Berdasarkan gambar di atas, sebuah besi (*iron core*) yang dililit oleh sebuah kumparan koil yang berfungsi untuk mengendalikan besi tersebut. Apabila Kumparan koil diberikan arus listrik, maka akan timbul gaya elektromagnet yang kemudian menarik armature untuk berpindah dari posisi sebelumnya (NC) ke posisi baru (NO) sehingga menjadi saklar yang dapat menghantarkan arus listrik di posisi barunya (NO). Posisi dimana armature tersebut berada sebelumnya (NC) akan menjadi *OPEN* atau tidak terhubung. Pada saat tidak dialiri arus listrik, armature akan kembali lagi ke posisi awal (NC). koil yang digunakan oleh *Relay* untuk menarik *contact poin* ke posisi *close* yang pada umumnya hanya membutuhkan arus listrik yang relatif kecil.

## 2.5 Driver Relay

*Driver Relay* adalah sebuah rangkaian elektronik yang digunakan untuk mengendalikan kerja *Relay* dengan memanfaatkan cara kerja dari rangkaian transistor sebagai *switch* dimana pada umumnya komponen yang terdapat didalamnya adalah Resistor, Transistor dengan tipe NPN, Dioda dan *Relay*. Alasan penggunaan *driver Relay* pada sebuah rangkaian elektronik disebabkan karena daya dari pengendali lebih kecil dibandingkan dengan daya yang dibutuhkan *Relay* untuk bekerja sehingga peran *driver Relay* sangat dibutuhkan untuk dapat menggerakkan *Relay*. Contoh perangkat yang sering menggunakan *driver Relay* adalah sirkuit papan logika (mikrokontrol, IC logika, mini komputer seperti Raspberry Pi dan lain sebagainya) yang keluaran pada setiap pin *out* hanya 5V dan arus maksimal sekitar 20mA, jika pin *output* dari sirkuit papan logika langsung disambungkan dengan *Relay*, maka *Relay* tidak akan bekerja karena pada umumnya *Relay* yang beredar dipasaran harus di suplai dengan tegangan minimal 12V ataupun ada yang *Relay* yang hanya di suplai dengan tegangan 5V namun membutuhkan arus yang cukup besar yaitu berkisar 50 – 100mA tergantung dari ukuran *Relay*.



Gambar 2.14 Gambar rangkaian *driver Relay*

(<https://www.raspberrypi.org/forums/viewtopic.php?f=37&t=43625>)

Berdasarkan Gambar 2.14, pada *driver Relay* terdapat tambahan yaitu berupa komponen Optocoupler yang sengaja dipasang pada rangkaian *driver Relay* yang berfungsi sebagai *switch* dan juga sebagai pengaman bilamana terjadi hubungan singkat pada rangkaian *driver Relay*, sehingga ketika terjadi hubung

singkat, arus tidak masuk ke dalam pin *output* dari sirkuit papan logika sehingga dikhawatirkan akan terjadi kekacauan logika yang menyebabkan sistem rusak atau bahkan kerusakan pada komponen.

## 2.6 HTML

HTML adalah singkatan dari *HyperText Markup Language* yaitu bahasa pemrograman standar yang digunakan untuk membuat sebuah halaman *Web*, yang kemudian dapat diakses untuk menampilkan berbagai informasi di dalam sebuah penjelajah *Web* Internet (*Browser*). HTML dapat juga digunakan sebagai *link - link* antara *file - file* dalam situs atau dalam komputer dengan menggunakan *localhost*, atau *link* yang menghubungkan antar situs dalam dunia internet. Supaya dapat menghasilkan tampilan wujud yang terintegrasi Pemformatan *hypertext* sederhana ditulis dalam berkas format ASCII sehingga menjadi halaman *Web* dengan perintah-perintah HTML. HTML merupakan sebuah bahasa yang bermula bahasa yang sebelumnya banyak dipakai di dunia percetakan dan penerbitan yang disebut *Standard Generalized Markup Language* (SGML) (Wempen, 2011).

Sekarang ini HTML merupakan standar Internet yang dikendalikan dan didefinisikan pemakaiannya oleh *World Wide Web Consortium* (W3C). Pada tahun 1989, HTML dibuat oleh kolaborasi Berners-lee Robert dengan Caillau TIM pada saat mereka bekerja di CERN (CERN merupakan lembaga penelitian fisika energi tinggi di Jenewa). HTML (HyperText Markup Language) adalah suatu bahasa yang menggunakan tanda-tanda tertentu (*tag*) untuk menyatakan kode-kode yang harus ditafsirkan oleh *browser* agar halaman tersebut dapat ditampilkan secara benar. Secara umum, fungsi HTML adalah untuk mengelola serangkaian data dan informasi sehingga suatu dokumen dapat diakses dan ditampilkan di Internet melalui layanan *Web*. Fungsi HTML yang lebih spesifik yaitu membuat halaman *Web*, menampilkan berbagai informasi di dalam sebuah *browser* Internet, membuat *link* menuju halaman *Web* lain dengan kode tertentu (*hypertext*) (Wempen, 2011).

## 2.7 PHP

PHP adalah bahasa pemrograman *script server - side* yang didesain untuk pengembangan *Web*. Selain itu, PHP juga bisa digunakan sebagai bahasa pemrograman umum (wikipedia, 2015). PHP dikembangkan pada tahun 1995 oleh Rasmus Lerdorf, dan sekarang dikelola oleh The PHP Group. Situs resmi PHP beralamat di <http://www.php.net>. Pada awalnya PHP merupakan singkatan dari *Personal Home Page*. Sesuai dengan namanya, PHP digunakan untuk membuat *Website* pribadi. Dalam beberapa tahun perkembangannya, PHP menjelma menjadi bahasa pemrograman *Web* yang *powerful* dan tidak hanya digunakan untuk membuat halaman *Web* sederhana, tetapi juga *Website* populer yang digunakan oleh jutaan orang seperti wikipedia, wordpress, joomla, dan lain - lain (Hanan, 2012).

PHP adalah singkatan dari PHP: *Hypertext Preprocessor*, sebuah kepanjangan *rekursif*, yakni permainan kata dimana kepanjangannya terdiri dari singkatan itu sendiri: PHP: *Hypertext Preprocessor*. PHP dapat digunakan dengan gratis (*free*) dan bersifat *open source*. Fungsi PHP Dalam Pemrograman *Web* adalah Untuk membuat halaman *Web*, sebenarnya PHP bukanlah bahasa pemrograman yang wajib digunakan. Kita bisa saja membuat *website* hanya menggunakan HTML saja. *Web* yang dihasilkan dengan HTML (dan CSS) ini dikenal dengan *Website* statis, dimana konten dan halaman *Web* bersifat tetap. Sebagai perbandingan, *Website* dinamis yang bisa dibuat menggunakan PHP adalah situs *web* yang bisa menyesuaikan tampilan konten tergantung situasi. *Website* dinamis juga bisa menyimpan data ke dalam *database*, membuat halaman yang berubah-ubah sesuai *input* dari *user*, memroses *form*, dll. Untuk pembuatan *web*, kode PHP biasanya disisipkan kedalam dokumen HTML. Karena fitur inilah PHP disebut juga sebagai *Scripting Language* atau bahasa pemrograman *script* (Hanan, 2012).

## 2.8 QoS (*Quality of Service*)

QoS (*Quality of Service*) adalah kemampuan suatu jaringan untuk menyediakan layanan yang lebih baik pada trafik data tertentu pada berbagai jenis *platform* teknologi. QoS tidak diperoleh secara langsung dari infrastruktur yang

ada, melainkan diperoleh langsung dengan mengimplementasikannya pada jaringan bersangkutan. QoS (*Quality of Service*) merupakan parameter-parameter yang menunjukkan kualitas paket data jaringan. Beberapa parameter yang mempengaruhi QoS antara lain *throughput*, *delay*, dan *packet loss* pada jaringan, selain itu QoS juga dipengaruhi oleh pemenuhan kebutuhan *bandwidth*. (Mukarram, 2010).

*Delay* merupakan suatu permasalahan yang harus diperhitungkan karena bagus tidaknya suatu layanan tergantung dari waktu *delay*. Besarnya *delay* maksimum yang direkomendasikan sesuai dengan standar ITU-T G.114 adalah 150 ms, sedangkan penerima akan mulai merasakan terjadinya *delay* jika > 250 ms. Pada tabel 2.1 menunjukkan parameter *delay* sesuai standar ITU-T G.114. (Mukarram, 2010).

Tabel 2.1 Parameter *Delay* berdasarkan ITU-T G.114

<b>Delay</b>	<b>Kualitas</b>
0 – 150 ms	Baik
150 – 300 ms	Cukup
> 500 ms	Buruk

Ada beberapa penyebab terjadinya *delay* antara lain:

1. Kongesti (kelebihan beban data).
2. Kekurangan pada metode *traffic shapping*.
3. Penggunaan paket-paket data dengan ukuran-ukuran berbeda.
4. Perubahan kecepatan antar jaringan WAN pematat *bandwidth* secara tiba-tiba.

## 2.9 Wireshark

Wireshark merupakan *software* untuk melakukan analisa lalu-lintas jaringan komputer, yang memiliki fungsi-fungsi yang amat berguna bagi profesional jaringan, administrator jaringan, peneliti, hingga pengembang piranti lunak jaringan. Wireshark dapat membaca data secara langsung dari Ethernet,

Token-Ring, FDDI, serial (PPP and SLIP), 802.11 *wireless* LAN dan koneksi ATM. (Kholid, tanpa tahun).

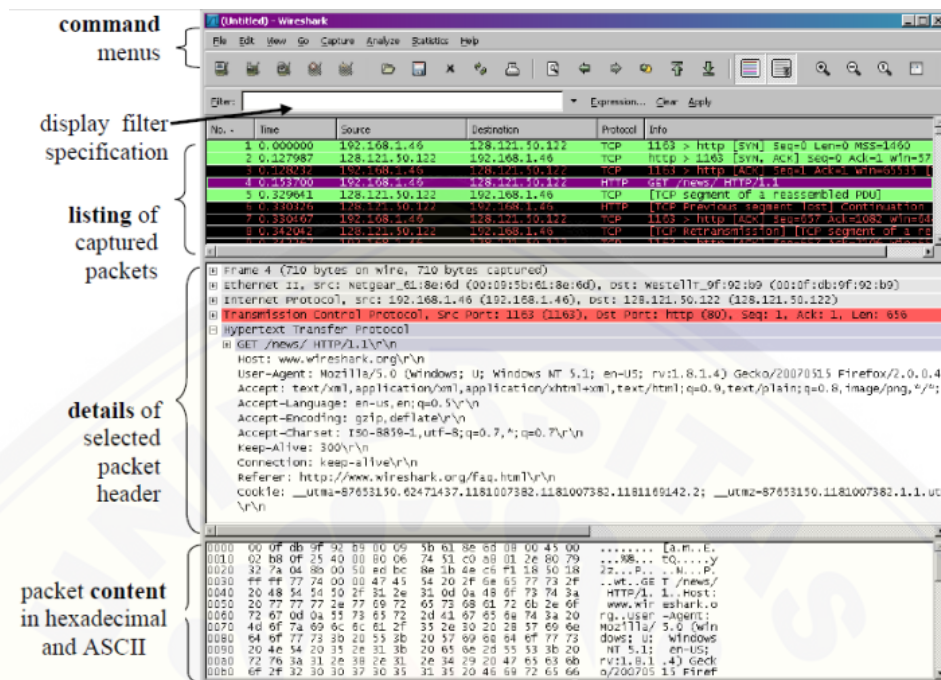
No.	Time	Source	Destination	Protocol	Info
1	0.000000	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
2	0.898984	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
3	1.899045	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
4	2.970706	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
5	18.900010	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.124? Tell 0.0.0.0
6	18.900294	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 239.255.255.250 for any sources
7	18.912888	0.0.0.0	239.255.255.250	DHCP	DHCP Request - Transaction ID 0x3b571104
8	19.400094	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 239.255.255.250 for any sources
9	19.900093	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.124? Tell 0.0.0.0
10	20.361292	167.205.67.124	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
11	20.900138	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.124? Tell 0.0.0.0
12	21.136266	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
13	21.900198	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
14	21.931784	167.205.67.124	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
15	21.946727	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any sources
16	21.980052	167.205.67.124	167.205.67.127	NBNS	Registration NB SYSADMIN-PC<00>
17	21.992460	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
18	22.027172	167.205.67.124	224.0.0.252	UDP	Source port: 61492 Destination port: 1199
19	22.127344	167.205.67.124	224.0.0.252	UDP	Source port: 61492 Destination port: 1199
20	22.327957	167.205.67.124	224.0.0.252	UDP	Source port: 49914 Destination port: 1199
21	22.400255	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.252 for any sources
22	22.428390	167.205.67.124	224.0.0.252	UDP	Source port: 49914 Destination port: 1199
23	22.729293	167.205.67.124	167.205.67.127	NBNS	Registration NB SYSADMIN-PC<00>
24	22.900256	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
25	23.395198	167.205.67.124	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
26	23.479327	167.205.67.124	167.205.67.127	NBNS	Registration NB SYSADMIN-PC<00>
27	23.480302	0.0.0.0	255.255.255.255	DHCP	DHCP Request - Transaction ID 0x3b571104
28	23.900303	Intelcor_ba:57:6c	Broadcast	ARP	who has 167.205.67.17 Tell 167.205.67.124
29	24.229377	167.205.67.124	167.205.67.127	NBNS	Registration NB SYSADMIN-PC<00>
30	24.385057	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.251 for any sources
31	24.400362	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.251 for any sources
32	24.411194	167.205.67.124	224.0.0.22	IGMP	V3 Membership Report / Join group 224.0.0.251 for any sources
33	24.912701	167.205.67.124	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1

Gambar 2.15 Tampilan Wireshark  
(Kholid, tanpa tahun)

*Tools* ini bisa menangkap paket - paket data / informasi yang berjalan dalam jaringan. Semua jenis paket informasi dalam berbagai format protokol pun akan dengan mudah ditangkap dan dianalisa. Karenanya tak jarang *tool* ini juga dapat dipakai untuk *sniffing* (memperoleh informasi penting seperti *password email* atau *account* lain) dengan menangkap paket-paket yang berjalan di dalam jaringan dan menganalisisnya. Namun *tools* ini hanya bisa bekerja didalam jaringan melalui LAN / Ethernet Card yang ada di PC. (Kholid, tanpa tahun).

Struktur dari Wireshark *graphical user interface* adalah sebagai berikut :

- Command menu*
- Display filter specification* : untuk men - *filter packet data*
- Listing of captured packets* : paket data yang tertangkap oleh Wireshark
- Details of selected packet header* : data lengkap tentang *header* dari suatu *packet*
- Packet contents* : isi dari suatu *packet data*



Gambar 2.16 Struktur Wireshark (Kholid, tanpa tahun)

Wireshark akan menampilkan semua informasi tentang *packet* yang keluar dan masuk dalam *interface* yang telah dipilih. Beberapa tampilan hasil capture Wireshark meliputi *Frame*, *Ethernet*, *Internet Protokol*, *User-Datagram Protokol*, *Link - local Multicast Name Resolution*.



### BAB 3. METODE PENELITIAN

#### 3.1 Tempat dan Waktu Penelitian

Pembuatan “Perancangan *Monitoring and Controlling Traffic Light* pada *Different Street Condition* dengan Menggunakan Jaringan Internet” dilakukan di laboratorium sistem kendali Fakultas Teknik, Teknik Elektro.

#### 3.2 Alat dan Bahan

##### 3.2.1 Alat

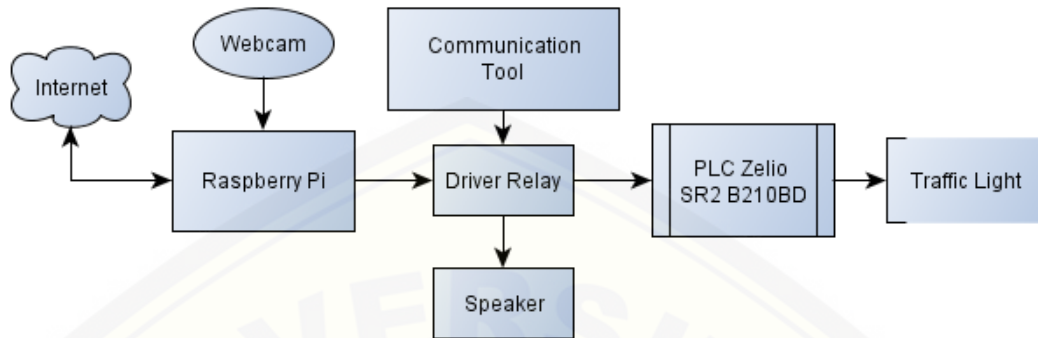
- |                |              |
|----------------|--------------|
| 1. Solder      | 6. Bor PCB   |
| 2. Timah       | 7. Antraktor |
| 3. Tang potong | 8. Gunting   |
| 4. Tang jepit  | 9. Lem bakar |
| 5. Multimeter  |              |

##### 3.2.2 Bahan

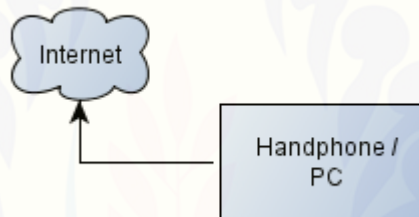
- |                               |   |
|-------------------------------|---|
| 1. Rapsberry Pi               | 6. Kabel LAN                                      |
| 2. Webcam                     | 7. Kabel pelangi                                  |
| 3. PLC Zelio SB2 B210<br>BD   | 8. Mini Amplifier                                 |
| 4. Modul <i>traffic light</i> | 9. <i>Header female</i> dan<br><i>header male</i> |
| 5. <i>Driver Relay</i>        |   |

### 3.3 Perancangan Sistem

#### 3.3.1 Diagram Blok



Gambar 3.1 Diagram blok sistem kontrol dan monitoring



Gambar 3.2 Diagram blok sistem *monitoring* dan pengendali pada *remote*

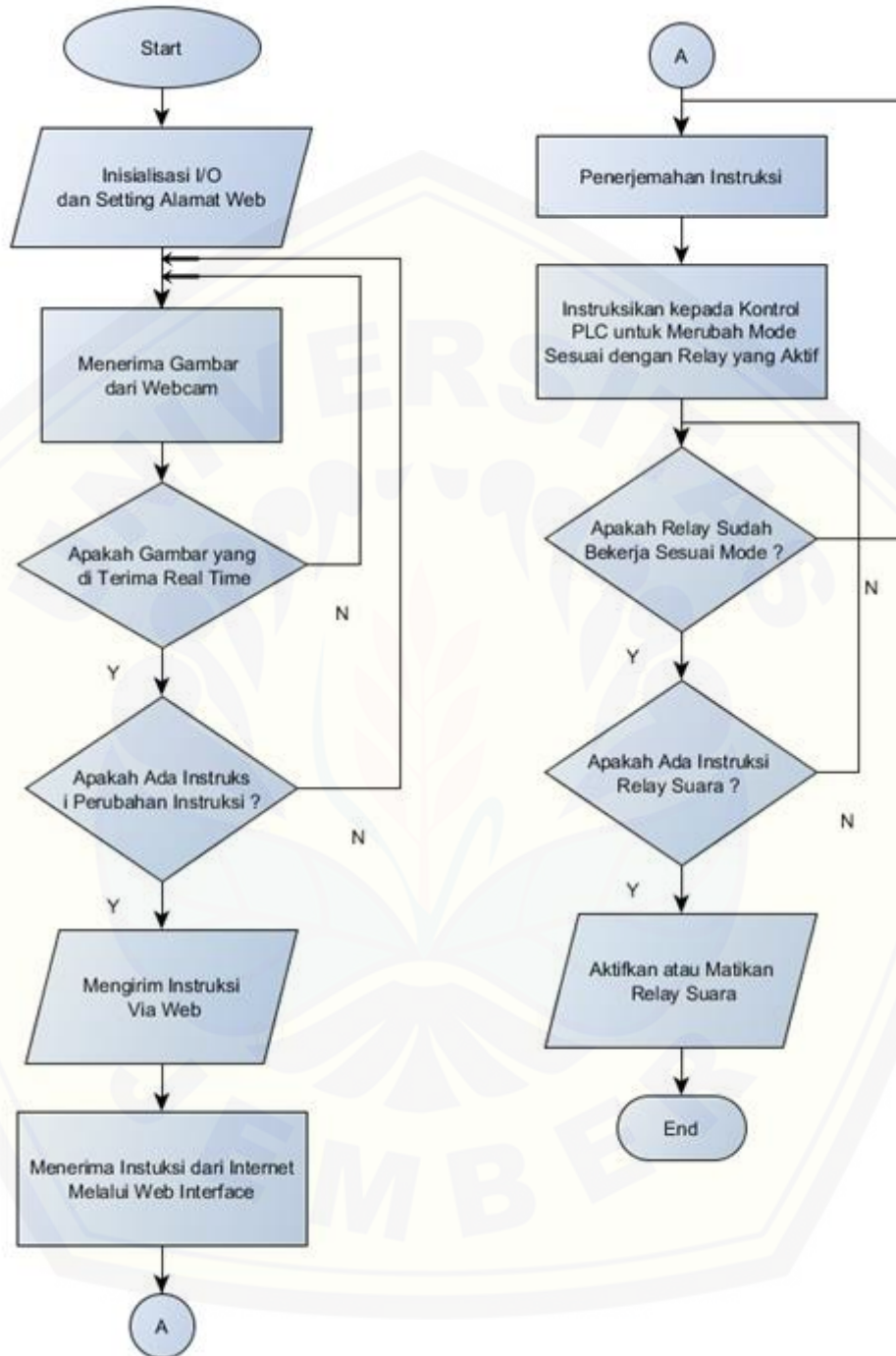
Diagram blok pada perancangan ini terdiri dari dua blok, blok pertama adalah blok kontrol dan monitoring utama yang terdiri dari *Webcam*, *Raspberry Pi*, *driver Relay* dan terakhir tersambung dengan PLC yang menjadi pengendali utama dari *traffic light*, sedangkan blok kedua adalah blok sistem monitoring dan pengendali pada *remote* yang berupa gadget baik itu berupa *handphone* ataupun PC yang tersambung dengan internet.

Pada blok pertama, yaitu blok kontrol dan monitoring, komponen bagian pengirim gambar situasi arus lalu lintas berupa *Webcam* yang tersambung dengan *Raspberry Pi*. *Webcam* sendiri pada blok ini berfungsi sebagai penangkap dari situasi arus lalu lintas, sedangkan *Raspberry Pi* berfungsi sebagai komponen pengirim hasil yang ditangkap oleh *Webcam* dan gambar tersebut dikirimkan melalui jaringan internet untuk ditampilkan melalui *Web*. Untuk bagian penerima instruksi, masih menggunakan komponen yang sama yaitu *Raspberry Pi* dimana

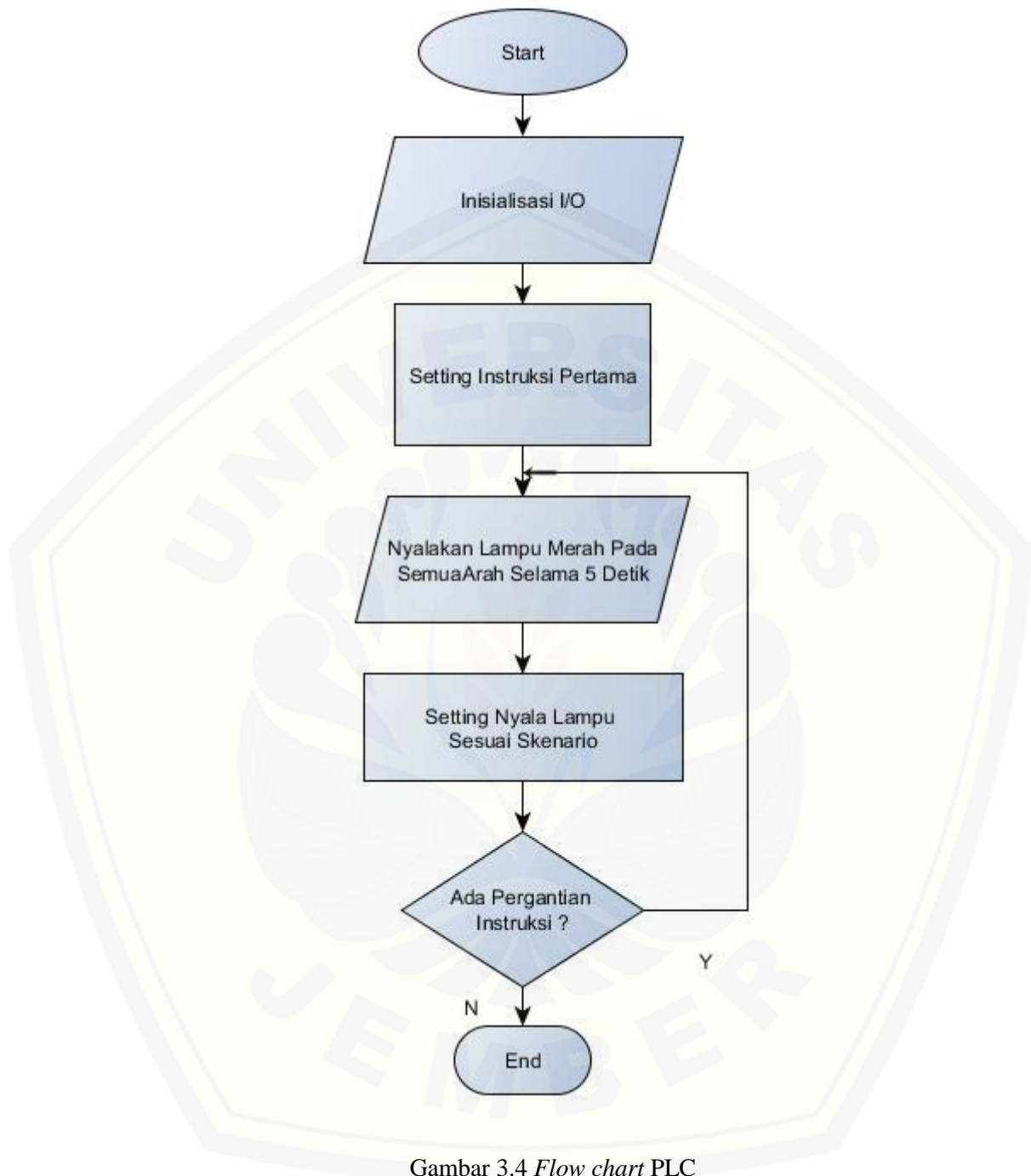
instruksi yang dikirimkan petugas melalui *interface* berupa tampilan *Web* selanjutnya akan diterima dan diproses oleh Raspberry Pi yang kemudian akan dilanjutkan untuk di olah oleh sistem pengontrol lampu lalu lintas. Untuk pengontrol lampu lalu lintas sendiri terdiri dari pin pada GPIO Raspberry Pi yang disambungkan pada *driver Relay* dan pin keluaran dari *driver Relay* disambungkan pada pin *input* PLC Zelio SR2 B210 BD. Pada bagian awal, data yang di terima oleh Raspberry Pi akan diproses kemudian didapatkan instruksi yang di maksud oleh pengirim. Setelah data tersebut berhasil di dekodekan, selanjutnya Raspberry Pi akan merespon dan mengaktifkan pin GPIO yang digunakan sebagai pin penerjemah yang mewakili saklar. Dikarenakan tegangan *output* dari pin GPIO Raspberry Pi hanya 3,3V, maka untuk dapat mengoneksikan ke perangkat utama yaitu berupa PLC perlu diberikan *driver* berupa *driver Relay* yang berfungsi untuk saklar perpindahan mode. *Relay* tersebut masuk ke dalam pin *input* dari PLC dan tersambung secara langsung dengan PLC yang berfungsi sebagai *main control* dari lampu lalu lintas. *Relay* pada sistem ini terdiri dari 4 *Relay* yang masing – masing *Relay* mewakili satu instruksi kerja dari lampu lalu lintas, dengan menghubungkan *Relay* tersebut ke PLC, maka PLC secara otomatis akan merespon bahwa instruksi kerja yang dimaksud oleh pengirim adalah mode 1, mode 2, mode 3 atau mode *speaker* aktif.

Sedangkan pada blok kedua adalah blok untuk sistem *monitoring* dan *controlling* pada *remote*. Pada blok ini, hanya terdiri dari satu komponen, yaitu *handphone* atau PC dimana gadget tersebut berfungsi sebagai gadget pengakses dari alamat *Web* yang telah disediakan oleh blok pertama sehingga operator dapat mengakses segala fasilitas yang telah disediakan oleh sistem.

3.3.2 Flow chart



Gambar 3.3 Flow chart sistem kontrol utama



Gambar 3.4 Flow chart PLC

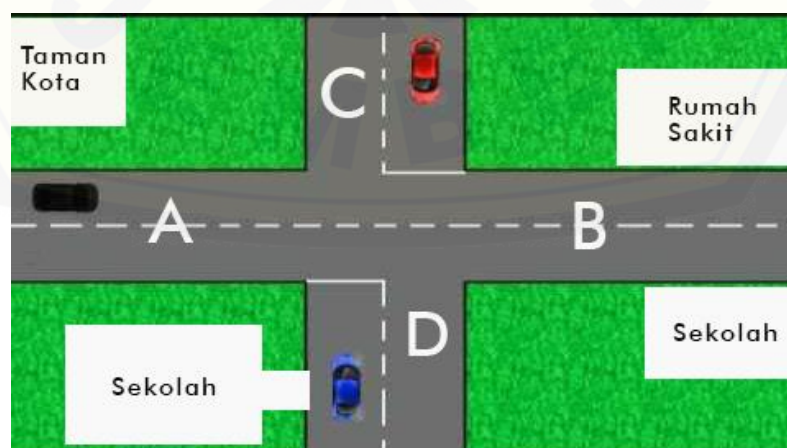
### 3.4 Cara Kerja

Cara kerja alat ini dapat dilihat pada gambar *flowchart* dimana modul ini terdiri dua bagian, yaitu untuk bagian yang pertama adalah bagian yang pengontrol dan bagian yang kedua adalah bagian PLC. Pada bagian pertama adalah bagian dari

sistem kontrol utama yang komponennya terdiri dari *Webcam*, *Raspberry Pi*, *driver Relay*. Cara kerja dari bagian pertama ini adalah sebagai berikut, pertama sistem menerima gambar yang ditangkap oleh *Webcam* yang kemudian gambar yang didapat dari *Webcam* tersebut di kirim melalui jaringan internet dengan *interface* berupa *Web*. Sistem mengirim gambar dengan kecepatan 20 FPS (*frame per second*) sehingga gambar mendekati *real time*. Setelah itu, sistem menunggu instruksi dari *Web* berupa instruksi perubahan mode apakah mode tersebut berganti. Jika ada pergantian instruksi maka instruksi tersebut dikirim melalui *Web* dan kemudian instruksi itu diterima yang selanjutnya kode tersebut dikodekan atau di terjemahkan oleh *Raspberry* dengan cara memberikan respon *high* ataupun *low* pada pin *GPIO Raspberry Pi* yang kemudian diumpankan ke *Relay* yang pada akhirnya *Relay* bergerak sesuai dengan instruksi kerja yang sudah ditentukan. *Relay* yang tersambung pada sistem ada 4 buah, dimana 3 *Relay* mewakili mode instruksi kerja dari lampu lalu lintas dan satu *Relay* mewakili instruksi kerja untuk menyalakan *Relay* suara sehingga setelah proses penerjemahan kode instruksi kerja pada 3 *Relay*, langkah selanjutnya sistem akan melakukan mengecek apakah ada instruksi *Relay* suara aktif atau tidak.

### 3.5 Skenario

Skenario yang digunakan peneliti adalah skenario jalan dengan 4 arus kendaraan, yaitu arah A, B, C dan D dengan gambar skenario seperti berikut,



Gambar 3.5 Skenario perempatan

pada skenario di atas jalan terbagi menjadi 4 arus, yaitu dua arus utama A dan B serta dua arus kecil yaitu arus C dan D. Jalan A adalah arus utama yang padat dan menjadi gerbang utama untuk masuk kota, pada jalan ini terdapat taman kota yang menjadi destinasi yang selalu ramai pengunjung disaat malam hari terutama malam Minggu, Minggu pagi dan pada *event - event* tertentu yang sengaja diselenggarakan oleh pemerintah kota yang di tempatkan di taman kota, sehingga kepadatan lalu lintas sering terjadi penumpukan pada jam – jam tersebut. Sedangkan jalan B selain menjadi penghubung utama yang mengarahkan pengguna jalan menuju luar kota, pada jalan ini terdapat dua bangunan penting yaitu sekolahan dan rumah sakit dimana pada pagi hari lalu lintas padat dikarenakan aktivitas lalu lintas kendaraan orang tua yang menghantarkan anaknya ke sekolah, begitu pula pada saat siang hari ketika orang tua menjemput anaknya. Selain itu kepadatan tersebut dikhawatirkan mengganggu aktivitas lalu lalang yang keluar masuk rumah sakit sehingga penanganan medis terhambat karena lalu lalang pada jam – jam tersebut. Sedangkan pada jalan C memiliki karakteristik yang hampir sama dengan karakteristik kepadatan lalu lintas pada jalan B, hanya saja pada jalan C ini arus tidak sebegitu padat karena jalan C adalah jalan yang kecil dan hanya padat pada jam – jam pagi dan siang hari saat orang tua beraktivitas antar jemput anaknya di sekolahan. Sedangkan jalan D hanyalah jalan kecil yang minim arus kendaraan, jalan ini hanya menghubungkan ketiga arus jalan A, B, C menuju sebuah perumahan kecil pada pinggiran kota. Dikarenakan arus yang sangat pada lalu lintas hanya terletak pada ketiga arus lalu lintas, maka *Webcam* yang terpasang diarahkan menuju arus A, B dan C yang menjadi penentu keputusan pergantian mode sesuai dengan arahan yang diberikan operator.

Berdasarkan skenario jalan tersebut, maka peneliti menggunakan tiga skenario pengaturan sebagai berikut,

### 3.5.1.1 Skenario Pertama

Skenario pertama adalah bagaimana mengatur agar arus lalu lintas dari arah B memiliki waktu nyala hijau yang lebih lama dibandingkan dengan jalan yang lain yang pada skenario ini menyala selama 12 detik. Pada skenario ini, untuk dapat

mengaktifkannya maka *Relay* 1 haruslah dalam kondisi menyala dan *Relay* yang lain yaitu *Relay* 2 dan 3 harus mati. Pada detik – detik awal yaitu 5 detik awal, lampu merah pada setiap arus akan menyala selama 5 detik dan setelah itu lampu akan bekerja sesuai dengan skenario.

Tabel 3.1 Skenario Sistem *Timing Diagram* Mode 1

		Detik																	
		5	7	9	11	13	15	17	19	21	23	25	27	29	31	33	35	37	39
Arus	A	Red										Green				Red			
	B	Red	Green										Red						
	C	Red										Green							
	D	Red										Green							
		Green										Red							
		Red										Green							
		Green										Red							
		Red										Green							

### 3.5.1.2 Skenario Kedua

Skenario kedua adalah kebalikan dari skenario pertama dimana pada skenario kedua kali ini bertujuan mengatur agar arus lalu lintas dari arah A memiliki waktu nyala hijau yang lebih lama dibandingkan dengan jalan yang lain yang pada skenario ini menyala selama 12 detik. Pada skenario ini, untuk dapat mengaktifkannya maka *Relay* 2 haruslah dalam kondisi menyala dan *Relay* yang lain yaitu *Relay* 1 dan 3 harus mati. Pada detik – detik awal yaitu 5 detik awal, lampu merah pada setiap arus akan menyala selama 5 detik dan setelah itu lampu akan bekerja sesuai dengan skenario.



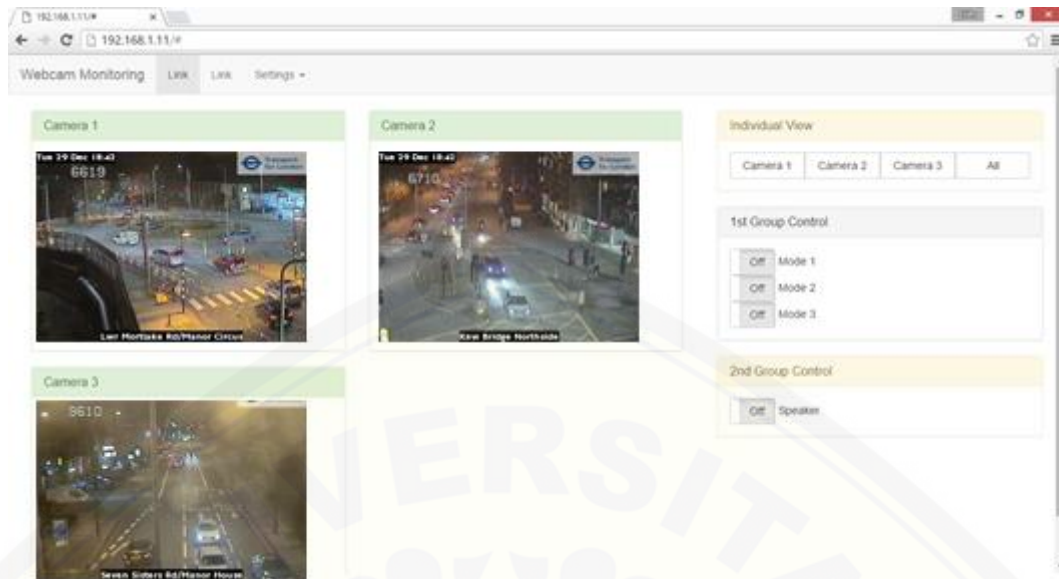


Sedangkan untuk *flowchart* yang kedua adalah *flowchart* pada bagaikan PLC. Pada saat pertama kali dinyalakan, sistem akan mengatur sistem agar berjalan sesuai dengan skenario pertama (dapat dilihat pada tabel 3.1) namun didahului dengan nyala lampu merah di setiap arah selama 5 detik agar di setiap arah berhenti sejenak dan mengantisipasi kebingungan pengendara karena sistem yang baru menyala. Setelah sistem memberikan instruksi menyalakan lampu merah pada setiap arah selama 5 detik dan menyalakan lampu sesuai dengan skenario pertama, maka sistem akan membaca apakah ada perintah perubahan skenario. Jika ada, maka sistem akan menyalakan kembali lampu merah di setiap arah selama 5 detik lalu menjalankan skenario yang telah dipilih oleh petugas ataupun polisi lalu lintas. Mode satu dan mode dua ditujukan untuk pengaturan dengan prioritas yang berbeda, mode satu untuk prioritas arus B dan mode dua prioritas arus A, sedangkan mode tiga adalah penggunaan untuk arus kepadatan yang normal. Perubahan dari mode satu ke mode dua dan ke mode tiga dikendalikan oleh *Relay* yang dihubungkan ke PLC sebagai pengganti dari saklar sedangkan untuk *relay* ke empat difungsikan hanya untuk menyambungkan ke *mini amplifier* suara apabila dibutuhkan untuk memberikan instruksi kepada pengguna jalan dengan memberikan aba – aba yang bersumber suara yang berasal dari Handy Talkie ataupun Walky Talkie.

Penentuan waktu didasarkan atas arus kendaraan yang memasuki persimpangan jalan, semakin padat tingkat kendaraan lalu lintas pada ruas jalan tertentu, maka semakin lama nyala lampu hijau pada *traffic light* tersebut. (Hayun dan Sundari, 2005).

### **3.6 Desain Web Interface**

Untuk desain dari *Web interface* sistem ini dirancang dengan memunculkan tiga gambar hasil dari *capture* yang ditangkap oleh *Webcam* dari 3 arah yang telah ditentukan. Selain itu pada *Web interface* ini disediakan kontrol panel yang berupa 4 buah tombol yang 3 tombol berfungsi untuk mengganti mode serta satu tombol berfungsi untuk mengaktifkan speaker pada jalan. Untuk gambar desain *Web Interface* adalah sebagai berikut.



Gambar 3.6 Desain *Web Interface*

### 3.6 Pengaturan IP Raspberry Pi

Untuk mengakses Raspberry Pi dapat dilakukan dengan menggunakan dua cara. Cara pertama dengan memasang monitor, *mouse* serta *keyboard* seperti pada pemasangan komputer dengan CPU seperti biasa. Sedangkan cara kedua yaitu dengan cara mengakses atau memrogramnya dengan cara mengakses melalui jaringan yang tersambung dengan Raspberry Pi. Sebelum menyambungkan Raspberry Pi dengan jaringan internet, maka pengaturan IP *address* perlu diatur. Untuk dapat mengaturnya dapat dilakukan dengan cara memasang Raspberry Pi seperti memasang CPU pada komputer biasa serta menyambungkan Raspberry Pi pada *router*. Setelah Raspberry bekerja ketik *syntax* “*startx*” untuk masuk ke terminal dekstop Raspberry Pi. Selanjutnya adalah mengetik *syntax* untuk pengaturan IP yaitu dengan cara memasukkan kode *syntax* sebagai berikut :

a. Cek koneksi :

```
sudo ifconfig
```

```
sudo nano /etc/network/interfaces
```

b. Mengubah dengan tulisan pada terminal :

```
iface eth0 inet dhcp
```

```
iface eth0 inet static
```

```
address 192.168.1.81
```

```
netmask 255.255.255.0
```

```
network 192.168.1.0
```

```
broadcast 192.168.1.255
```

```
gateway 192.168.1.254
```

c. Setelah itu ditekan ctrl + x untuk menyimpan pengaturan, setelah itu ketik *syntax* berikut untuk menghidupkan ulang Raspberry Pi :

```
sudo reboot
```

d. Setelah Raspberry menyala, mengecek ulang IP dengan mengetik *syntax* berikut :

```
ifconfig
```

### 3.7 Membangun Jaringan *Streaming Kamera*

Pada proses pembangunan *streaming* kamera pada Raspberry Pi, dibutuhkan *software* tambahan yaitu *software motion*. Untuk dapat melakukan *upgrading software* pada Raspberry Pi maka diperlukan koneksi internet yang kemudian ketikkan *syntax* berikut :

a. *Install software*

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install motion
```

- b. Setelah proses *download* dan *installing* selesai, selanjutnya dilakukan pemasangan webcam pada Raspberry Pi kemudian ketikkan *syntax* berikut untuk memastikan webcam sudah terpasang :

```
lsusb
```

- c. Selanjutnya adalah melakukan pengaturan pada software motion dengan cara mengetikkan *syntax* berikut :

```
sudo nano /etc/motion/motion.conf
```

- d. Lalu mengubah tulisan pada layar monitor dengan tulisan :

```
Daemon = OFF menjadi ON
```

```
webcam_localhost = ON menjadi OFF
```

```
control_localhost = ON menjadi OFF
```

- e. Menyimpan dengan menekan ctrl + x.

- f. Lalu untuk menjalankan aplikasi dengan cara mengetikkan :

```
sudo nano /etc/default/motion
```

Dikarenakan kita menggunakan 3 kamera, maka kita perlu memberikan pengaturan ekstra dengan cara :

a. Mematikan *service motion* :

```
sudo etc/init.d/motion stop
```

b. Selanjutnya *installing* apache

```
sudo apt-get update
```

```
sudo apt-get upgrade
```

```
sudo apt-get install motion
```

```
sudo apt-get install apache2
```

```
sudo apt-get install emacs
```

c. Membuat *file* untuk kebutuhan *motion.conf* :

```
sudo nano /home/pi/cam1.conf
```

```
videodevice /dev/video0
```

```
text_left USBWebcam-1
```

```
webcam_port 8081
```

```
sudo nano /home/pi/cam2.conf
```

```
sudo videodevice /dev/video2
```

```
sudo text_left USBWebcam-2
```

```
sudo webcam_port 8082
```

```
sudo nano /home/pi/cam3.conf
```

```
sudo videodevice /dev/video3
```

```
sudo text_left USBWebcam-3
```

```
sudo webcam_port 8083
```

d. Mengubah /etc/motion/motion.conf :

```
thread /home/pi/cam1.conf
```

```
thread /home/pi/cam2.conf
```

```
thread /home/pi/cam3.conf
```

e. Lalu membuat *file* ketikkan program :

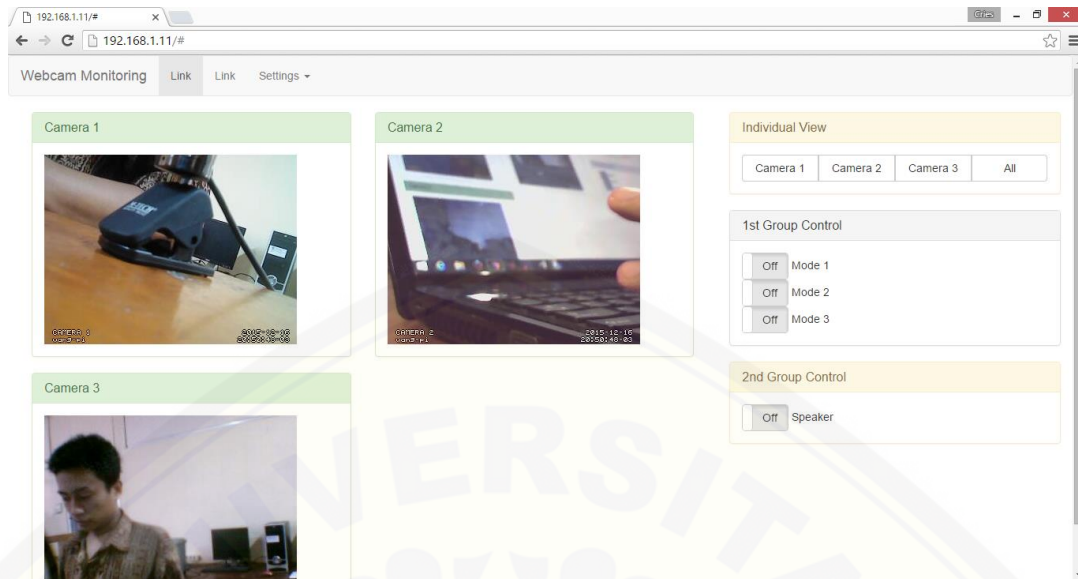
```
sudo nano /ar/www/webcam.html
```

f. Dan mengetik program seperti yang ada pada lampiran 1

g. Menjalankan sistem dengan cara :

```
sudo etc/init.d/motion start
```

Melakukan pemeriksaan kondisi dengan mengakses IP 192.168.1.11 sesuai dengan pengaturan yang telah dibuat sebelumnya.

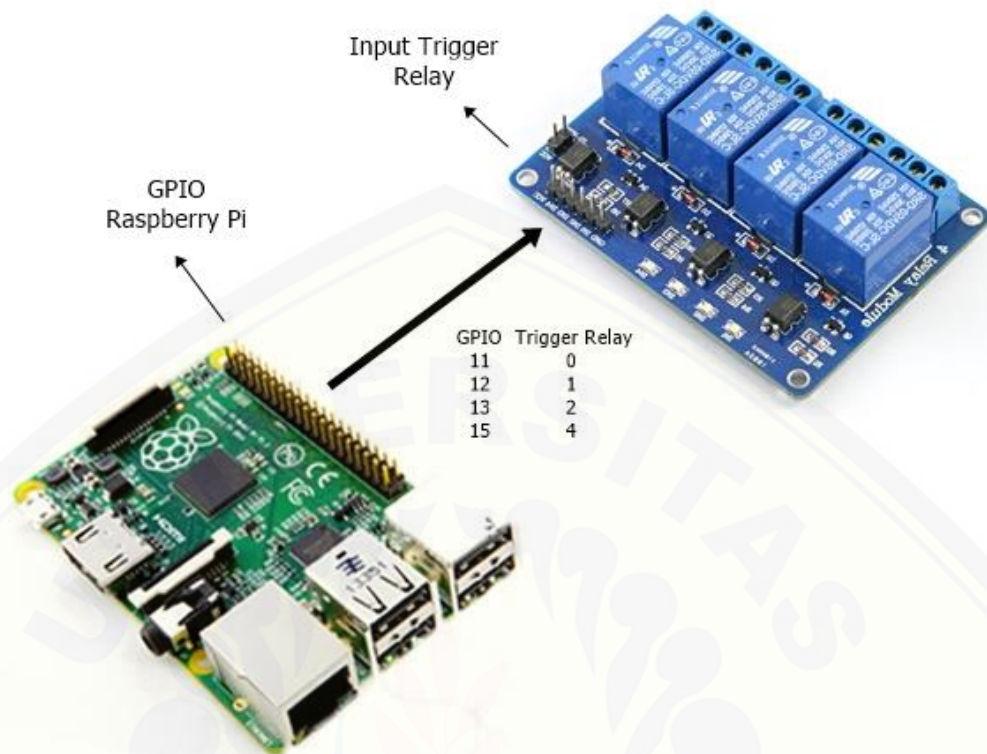


Gambar 3.7 Tampilan Kamera pada *Web Interface*

### 3.8 Membangun Kontrol Utama

Pada alat yang dirancang kali ini, kontrol utama yang berfungsi sebagai pengendali *relay* dan sekaligus sebagai penghubung antara pengguna dengan alat utama menggunakan Raspberry Pi dengan tipe B+ yang disuplai dengan menggunakan adaptor yang memiliki tegangan *output* sebesar 5,15 VDC. Untuk dapat mengendalikan alat utama berupa PLC yang tersambung dengan modul *traffic light* maka dibutuhkan sebuah perangkat penghubung dikarenakan tegangan yang dibutuhkan oleh PLC untuk dapat dikenali logikanya harus menggunakan tegangan 24 VDC dan tidak mungkin disambungkan langsung dengan Raspberry Pi dikarenakan tegangan terlampaui jauh. Oleh karena itu untuk menghubungkan antara PLC dengan Raspberry Pi dipasanglah *relay*, sehingga Raspberry Pi hanya memberikan instruksi berupa *trigger* kepada pin *input relay* yang berfungsi sebagai pengendali *relay* hanya dengan tegangan yang kecil.





Gambar 3.8 Konfigurasi Pemasangan Pin

Pada perancangan kontrol utama, untuk dapat mengendalikan *relay* maka digunakanlah pin GPIO yang berada di Raspberry Pi. Pada penelitian kali ini, pin GPIO yang digunakan adalah pin GPIO 0 pada pin nomor 11, GPIO 1 pada pin nomor 12, GPIO 2 pada pin nomor 13 dan GPIO 3 pada pin nomor 15 seperti yang terlihat pada Gambar 3.8.

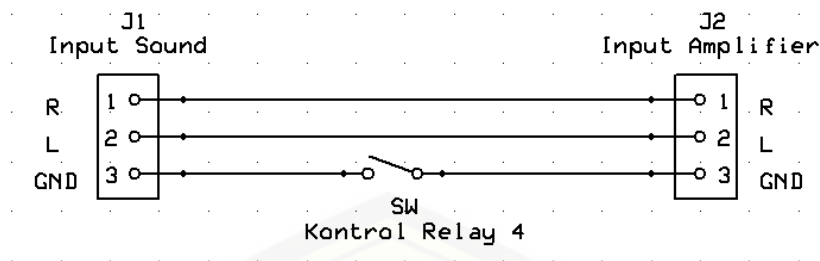
Modul *relay* yang digunakan pada penelitian memiliki 4 pin *input* dengan 4 *relay* yang dikendalikan. Pada modul *relay*, *input* sebelum masuk ke komponen transistor sebagai *switch* terlebih dahulu masuk ke komponen *optocoupler* yang berfungsi sebagai pengaman sekaligus sebagai *switch* dengan tegangan yang rendah sehingga hanya dengan menggunakan tegangan 3V saja *relay* mampu bekerja atau aktif. Pada modul *relay* yang digunakan peneliti, tegangan yang dibutuhkan untuk menggerakkan *relay* sebesar 5V saja sehingga pada penelitian kali ini, peneliti menyambungkan pin VCC dan GND pada relay dengan 4,98V yang didapatkan

dari adaptor. Setelah pin GPIO pada Raspberry Pi tersambung dengan pin *input trigger* dari *relay*, selanjutnya adalah memasukkan program pada Raspberry Pi dan menyimpannya pada direktori `pi\var\www\kontrol\public_html` baik program dengan format html ataupun php agar sistem dapat bekerja dan dikontrol dengan menggunakan jaringan internet. Kedua program tersebut terlampir pada lampiran 1 dan lampiran 2.

Setelah melakukan penyambungan Raspberry Pi dan *relay*, langkah selanjutnya adalah memasang modul *relay* dengan memasukkan atau *port input* dari PLC dimana pada penelitian kali ini, peneliti menyambungkan *port* keluaran dari *relay* menuju PLC pada sambungan *normally close* dari keluaran *relay*. Pada masing masing keluaran *relay* disambungkan pada masing – masing *input* dari PLC dengan konfigurasi, *relay* 1 disambungkan pada *input* PLC I1, *relay* 2 disambungkan pada *input* PLC I2 dan *relay* 3 disambungkan pada *input* PLC I3. Sedangkan untuk *relay* 4 tidak disambungkan dengan *input* PLC dikarenakan *relay* 4 dikhususkan untuk *relay* kontrol suara. *Input* I1 mewakili mode 1, I2 mewakili mode 2 dan I3 mewakili mode 3.

### 3.9 Membangun Kontrol Suara

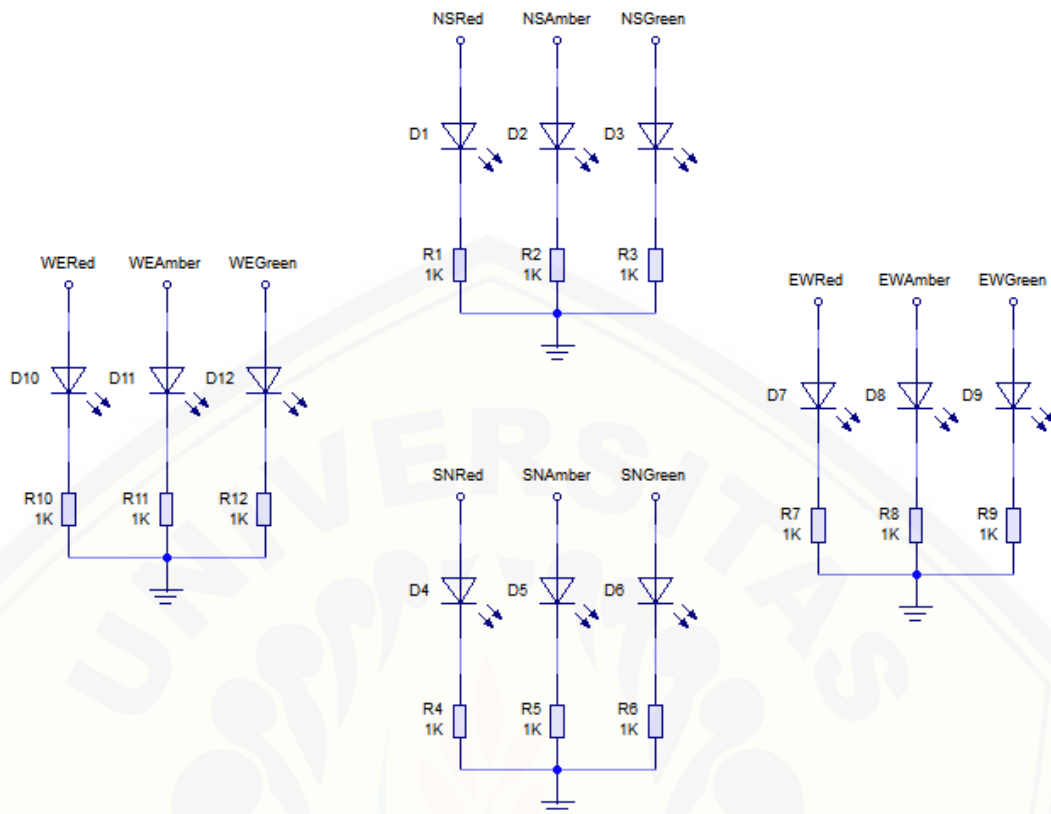
Kontrol suara terdiri dari dua bagian, bagian pertama adalah bagian *connector* atau *jack* suara yang tersambung dengan sumber suara dan bagian kedua adalah *connector* atau *jack* suara yang tersambung dengan *input* dari Amplifier. Pada sistem kontrol suara, peneliti mendesain agar suara aktif atau terdengar ketika *relay* kontrol suara atau *relay* 4 aktif sehingga dapat dilihat pada Gambar 3.9 bahwa peneliti memutus sambungan *ground* pada kedua *input*. SW adalah saklar yang terletak pada *relay* 4 sehingga ketika *relay* 4 aktif, maka *ground* tersambung dan suara dan terdengar.



Gambar 3.9 Skema Kontrol Suara

### 3.10 Membangun Miniatur *Traffic Light*

Untuk dapat membangun miniatur dari lampu lalu lintas atau *traffic light*, dibuatlah miniatur jalan dengan 4 persimpangan. Di antara jalan yang dibuat adalah jalan dengan arah utara ke selatan atau disingkat dengan NS (*north south*), jalan dengan arah selatan ke utara atau disingkat dengan SN (*south north*), jalan dengan arah barat ke timur atau disingkat dengan WE (*west east*) dan jalan dengan arah timur ke barat atau disingkat dengan EW (*west east*). Lampu lalu lintas pada miniatur jalan menggunakan LED dengan tegangan sebesar 24 V dengan jumlah lampu 12 lampu yang pada masing – masing LED diseri dengan resistor sebesar 1K Ohm seperti yang terlihat pada Gambar 3.10.



Gambar 3.10 Rangkaian LED *Traffic Light*

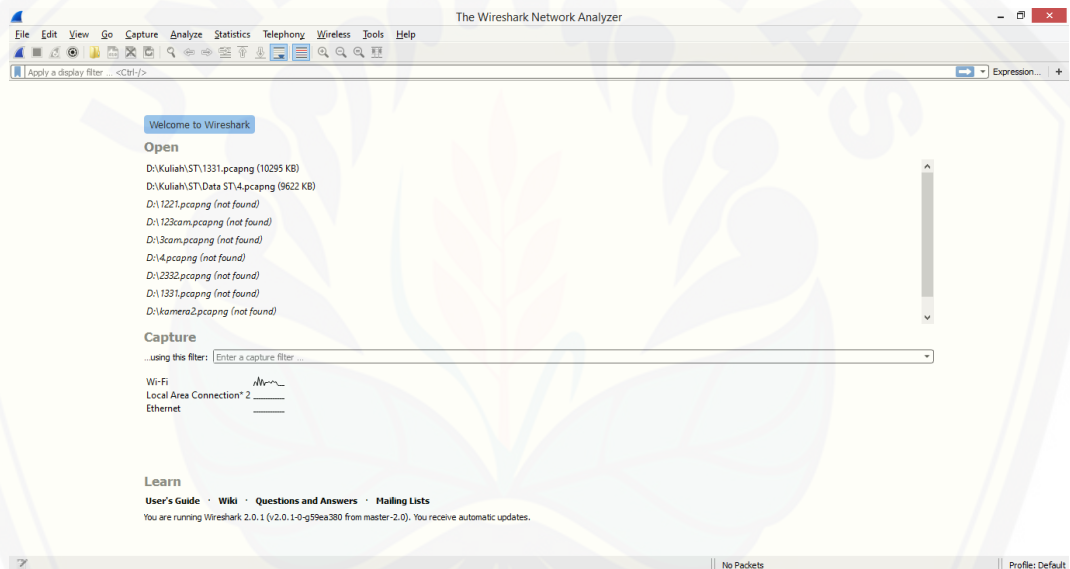
Pada rangkaian Gambar 3.10, masing – masing jalan memiliki tiga buah lampu, yaitu lampu merah atau *red*, kuning atau *amber* dan hijau atau *green*. NSRed adalah lampu merah dari arah utara ke selatan, NSAmber adalah kuning dari arah utara ke selatan dan NSGreen adalah lampu hijau dari arah utara ke selatan, begitu pula untuk lampu pada setiap arah. Pada penelitian yang dilakukan, peneliti menggunakan 2 lampu saja pada setiap arah karena keterbatasan *port* keluaran pada PLC sehingga lampu yang digunakan adalah warna merah dan hijau saja. Pada masing – masing lampu dikontrol oleh satu *port* keluaran PLC. Lampu NS atau arah dari utara ke selatan mewakili lampu A dimana untuk NSRed dikendalikan oleh Q1 sedangkan untuk NSGreen dikendalikan oleh Q2. Lampu SN atau arah dari selatan ke utara mewakili lampu B dimana untuk SNRed dikendalikan oleh Q3 sedangkan untuk SNGreen dikendalikan oleh Q4. Lampu WE atau arah dari barat ke selatan mewakili lampu C dimana untuk WERed dikendalikan oleh Q5 sedangkan untuk WEGreen dikendalikan oleh Q6. Lampu EW atau arah dari timur ke barat mewakili

lampu D dimana untuk EWRed dikendalikan oleh Q7 sedangkan untuk EWGreen dikendalikan oleh Q8.

### 3.11 Analisa Sistem dengan Wireshark

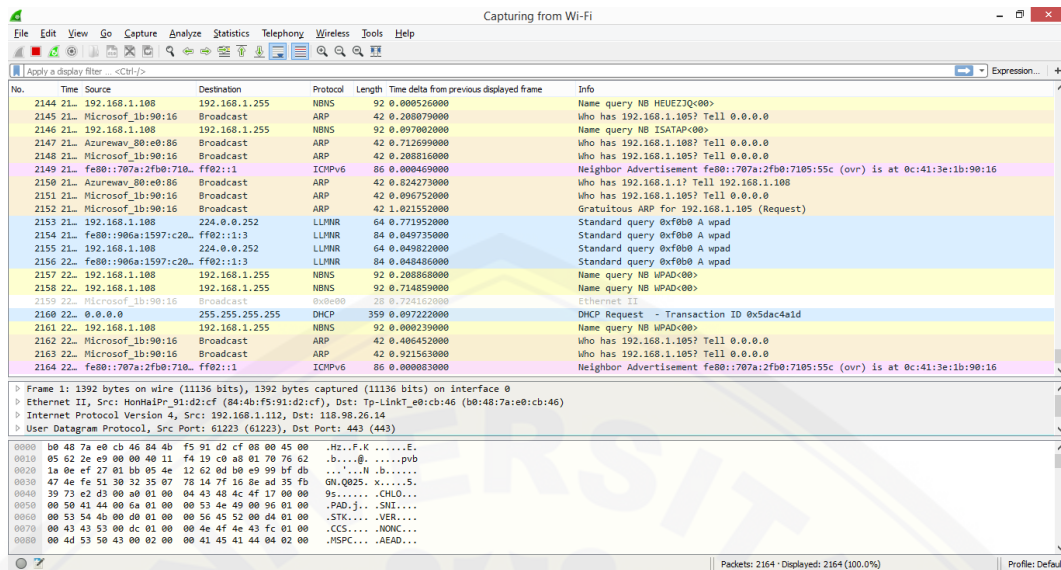
Untuk dapat mengetahui hasil *capture* dari sistem, dapat mengikuti langkah-langkah dibawah ini.

1. Menjalankan Wireshark dengan cara mengklik pada menu *shortcut*.
2. Mulai men-*capture* dengan cara memilih jaringan yang akan kita gunakan.



Gambar 3.11 Menu Utama Wireshark

3. Setelah kita memilih jaringan mana yang akan kita gunakan dengan cara men – klik, maka hasil *capture* akan terlihat.



Gambar 3.12 Gambar Hasil Capture

- Setelah hasil *capture* muncul, data tersebut dapat disimpan dengan cara pilih *File* lalu *save*.

## BAB 5. PENUTUP

### 5.1 Kesimpulan

Dari hasil penelitian pembuatan “*Perancangan Monitoring and Controlling Traffic Light pada Different Street Condition dengan Menggunakan Jaringan Internet*”, dapat diambil beberapa kesimpulan diantaranya :

1. Dari hasil pengujian pengukuran rerata waktu tunda atau *delay* pada kamera didapatkan waktu tunda untuk 3 tampilan kamera bersamaan sebesar 0,006934s untuk kamera 1, 006012 untuk kamera 2 dan 005486s untuk kamera 3. Sedangkan untuk waktu tunda kamera 1 sebesar 0,143407s, kamera 2 sebesar 0,145678s dan kamera 3 sebesar 0,140676s. Dari data tersebut maka sistem dikatakan memiliki standar yang baik menurut ITU G.114.
2. Dari hasil pengujian waktu tunda untuk *relay*, didapatkan waktu tunda tombol 1 sebesar 0,031789s, tombol 2 memiliki waktu tunda sebesar 0,069339s, tombol 3 memiliki waktu tunda sebesar 0,041896s dan tombol 4 memiliki waktu tunda sebesar 0,050381s sehingga dari keempat data tersebut, maka sistem dapat dikatakan memiliki waktu tunda yang baik sesuai dengan standar ITU G.114. Sedangkan untuk pengujian sistem perpindahan antar skenario secara keseluruhan dapat dikatakan baik karena pada keenam perpindahan skenario rerata waktu tunda tidak sampai melebihi 150 ms.
3. Dari hasil pengujian *relay* suara, didapatkan bahwa dari 20 percobaan tidak terdapat kegagalan sama sekali atau probabilitas keberhasilan sebesar 100%.

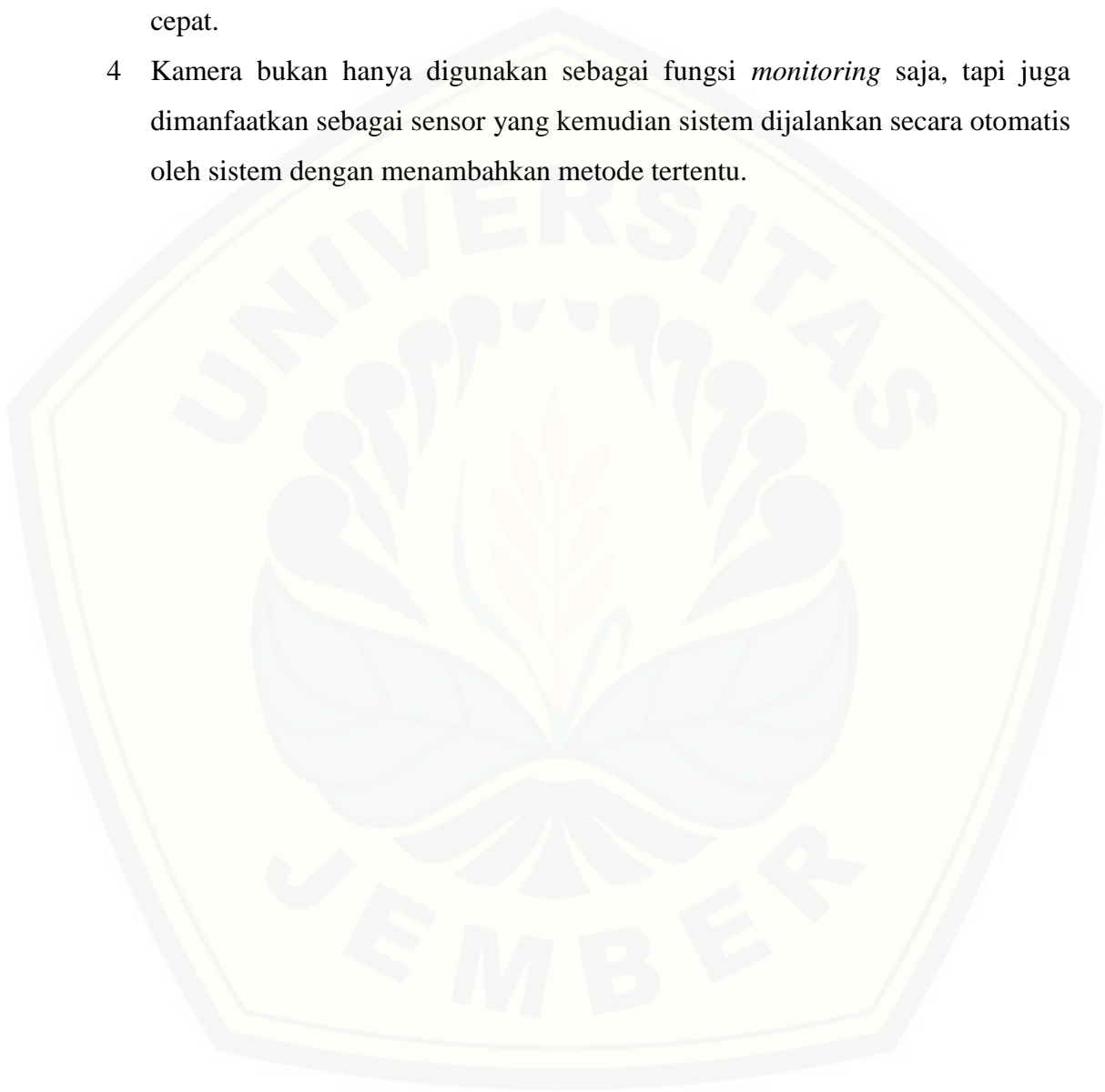
### 5.2 Saran

Berdasarkan hasil penelitian didapatkan beberapa saran sebagai berikut :

- 1 Untuk dapat melakukan implementasi pada kondisi jalan yang sebenarnya, disarankan menggunakan 4 *webcam* dengan kualitas yang sangat bagus.
- 2 Untuk dapat mengganti tipe PLC dengan jumlah port I/O yang lebih banyak dan memori yang lebih besar agar dapat diimplementasikan pada jalan yang

sesungguhnya baik dalam hal penambahan jumlah lampu ataupun jumlah mode yang diinginkan.

- 3 Perlu dipertimbangkan untuk mengganti Rapsberry dengan tipe yang lebih terbaru agar kualitas pemrosesan pengiriman instruksi dapat berjalan lebih cepat.
- 4 Kamera bukan hanya digunakan sebagai fungsi *monitoring* saja, tapi juga dimanfaatkan sebagai sensor yang kemudian sistem dijalankan secara otomatis oleh sistem dengan menambahkan metode tertentu.





## DAFTAR PUSTAKA

- Asri, M. 2010. *Pengembangan Sistem Kontrol dan Monitoring Lampu Lalu Lintas*. Jurnal Ilmiah Universitas Hasanudin. Makasar : Universitas Hasanudin
- Bell, C. 2013. *Beginning Sensor Networks with Arduino and Rapsberry Pi*. Ebook. New York : Apress
- Hanan, A. *Panduan PHP Bahasa Indonesia*. <http://saung-foss.web.id/doku/id-php>. Diakses pada tanggal 30 Desember 2015.
- Hayun, A, Sundari. 2005. *Penentuan Waktu Penyalaan Lampu Lalu Lintas yang Optimal : Kasus Persimpangan Buah Batu Lingkar Selatan*. INASEA, Vol. 90 6 No. 2, Oktober 2005: 77-90. Jakarta : Universitas Binus dan BPPT
- Membrey, P, Hows, D. *Learn Raspberry Pi with Linux*. Ebook. New York : Apress
- Mukkaram, Fauzan Saiful Haq. 2010. *Analisis Implementasi Aplikasi Video Call pada Sinkronisasi Learning Management System Berbasis Moodle sebagai Metode Distance Learning dalam Institusi Pendidikan*. *Proceeding Seminar Tugas Akhir Jurusan Teknik Elektro FTI - ITS*. Surabaya : ITS.
- Triyono, J. 2010. "Pelayanan KRS Online berbasis SMS". *Jurnal Teknologi*, Vol. 3 No. I, Juni 2010, hal. 33-38.
- Tutorial Point. 2015. *PHP Hypertext Preprocessor*. Ebook.
- Republik Indonesia. 2009. *Undang-Undang Nomor 22 Tahun 2009 tentang Lalu Lintas dan Angkutan Jalan*. *Lembaran Negara tahun 2009*. Sekretariat Negara. Jakarta.
- Wempen, F. 2011. *Step by Step HTML 5*. Ebook. California : Microsoft Corporation.
- Widjonarko. 2012. *Petunjuk Praktikum Sistim Kendali (Program Logic Control / PLC Series Zelio Schneider)*. Modul Praktikum. Jember : Universitas Jember.
- Wijayanto, H. 2015. *Rancang Bangun Sistem Informasi dan Monitoring APILL (SIMAPILL) Berbasis Teknologi SMS Gateway*. FSTPT International Symposium. Lampung : Universitas Bandar Lampung.
- Williard, W. 2009. *HTML : A Begineer's Guide Fourth Edition*. Ebook. New York : MCGraw-Hill Companies.

## LAMPIRAN

Lampiran 1 Program HTML :

```

<!DOCTYPE html>
<html>
<head>
    <link href="css/bootstrap.min.css" rel="stylesheet">
    <link href="css/bootstrap-toggle.css" rel="stylesheet">
    <script src="js/jquery-2.1.4.min.js"></script>
    <script src="js/bootstrap.min.js"></script>
    <script src="js/bootstrap-toggle.js"></script>
    <script type="text/javascript">
    //<![CDATA[

    var Webcam = function(cam){
        this.cam = cam;
        this.interval = 200;
        this.width = 320;
        this.height = 240;
        this.thread = 0;
        this.src = "motion/cap" + this.cam + ".jpg";
        this.frame = $('<img width="320" height="240" src="" + this.src +
"/>');
        this.container = $('#cont' + this.cam);

    };

    $.extend(Webcam.prototype, {
        shrink: function(){
            this.container.animate({
                width: 0,

```

```
        height: 0
      }, 1000, function(){
        this.hide();
      });
    },
    grow: function(){
    },
    dimension: function(width, height){
      this.width = width;
      this.height = height;
      return true;
    },
    view: function(){
      return this.img;
    },
    setFPS: function(fps){
      this.interval = 1000/fps;
      return this.interval;
    },
    setInterval: function(interval){
      this.interval = interval;
      return this.interval;
    },
    start: function(){
      this.stop();
      var frame = this.frame;
      var src = this.src;
      var cam = this.cam
```

```

        var width = this.width;
        var height = this.height;
        this.thread = window.setInterval(function(){
            $('<img width="" + width + "" height="" + height + ""
src="" + src +'?' + new Date().getTime() + "">').load(function () {
                $('#cap' + cam).empty();
                $(this).appendTo('#cap' + cam);
            });
        }, this.interval);
    },
    stop: function(){
        window.clearInterval(this.thread);
        this.thread = null;
    }
});

$(document).ready(function () {
    /*var cam0 = setInterval(function(){
        $('<img width="320" height="240" src="" + imgPaht0+'?' + new
Date().getTime() + "">').load(function () {
            $('#cap0').empty();
            $(this).appendTo('#cap0');
        });
    }, 100);
    var cam1 = setInterval(function(){
        $('<img width="320" height="240" src="" + imgPaht1+'?' + new
Date().getTime() + "">').load(function () {
            $('#cap1').empty();
            $(this).appendTo('#cap1');
    }

```

```
    });  
  }, 100);  
  
  var imgPaht0 = "motion/cap0.jpg";  
  var imgPaht1 = "motion/cap1.jpg";  
  */  
});  
  
</script>  
  
</head>  
<body>  
  
<nav class="navbar navbar-default">  
  <div class="container-fluid">  
    <!-- Brand and toggle get grouped for better mobile display -->  
    <div class="navbar-header">  
      <button type="button" class="navbar-toggle collapsed" data-  
toggle="collapse" data-target="#bs-example-navbar-collapse-1" aria-  
expanded="false">  
        <span class="sr-only">Toggle navigation</span>  
        <span class="icon-bar"></span>  
        <span class="icon-bar"></span>  
        <span class="icon-bar"></span>  
      </button>  
      <a class="navbar-brand" href="#">Webcam Monitoring</a>  
    </div>  
  
    <!-- Collect the nav links, forms, and other content for toggling -->
```

```

<div class="collapse navbar-collapse" id="bs-example-navbar-collapse-1">
  <ul class="nav navbar-nav">
    <li class="active"><a href="#">Link <span class="sr-
only">(current)</span></a></li>
    <li><a href="#">Link</a></li>
    <li class="dropdown">
      <a href="#" class="dropdown-toggle" data-toggle="dropdown"
role="button" aria-haspopup="true" aria-expanded="false">Settings <span
class="caret"></span></a>
      <ul class="dropdown-menu">
        <li><a href="#" class="disabled">Pin Mode</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="#" id="link1">Input</a></li>
        <li><a href="#" id="link2">Output</a></li>
      </ul>
    </li>
  </ul>
</div><!-- /.navbar-collapse -->
</div><!-- /.container-fluid -->
</nav>

<div class="container-fluid">
  <div class="row">
    <div class="col-md-8">
      <div class="col-md-6" id="cont0">
        <div class="panel panel-success">
          <div class="panel-heading">
            <h3 class="panel-title">Camera
1</h3>
          </div>
          <div class="panel-body" id="cap0">

```

```

                <img src="" width="320"
height="240" class="img-responsive" />
            </div>
        </div>
    </div>
    <div class="col-md-6" id="cont1">
        <div class="panel panel-success">
            <div class="panel-heading">
                <h3 class="panel-title">Camera
2</h3>
            </div>
            <div class="panel-body" id="cap1">
                <img src="" width="320"
height="240" class="img-responsive" />
            </div>
        </div>
    </div>
    <!--div class="col-md-12">
        <div class="panel panel-info">
            <div class="panel-heading">
                <h3 class="panel-title">GPIO
Reference</h3>
            </div>
            <div class="panel-body">
                
            </div>
        </div>
    </div-->
    <div class="col-md-6" id="cont2">
        <div class="panel panel-success">
```

```
<div class="panel-heading">
  <h3 class="panel-title">Camera
3</h3>
</div>
<div class="panel-body" id="cap2">
  <img src="" width="320"
height="240" class="img-responsive" />
</div>
</div>
</div>
</div>
<div class="col-md-4">
  <div class="panel panel-warning">
    <div class="panel-heading">
      <h3 class="panel-title">Individual
View</h3>
    </div>
    <div class="panel-body">
      <div class="btn-group btn-group-justified"
role="group" aria-label="...">
        <div class="btn-group" role="group">
          <button type="button" class="btn btn-
default" id="btn1">Camera 1</button>
        </div>
        <div class="btn-group" role="group">
          <button type="button" class="btn btn-
default" id="btn2">Camera 2</button>
        </div>
        <div class="btn-group" role="group">
          <button type="button" class="btn btn-
default" id="btn3">Camera 3</button>
        </div>
      </div>
    </div>
  </div>
</div>
```



```

        <div class="btn-group" role="group">
            <button type="button" class="btn btn-
default" id="btn4">All</button>
        </div>
    </div>
</div>
</div>
</div>
</div>
<div class="col-md-4">
    <div class="panel panel-default">
        <div class="panel-heading">
            <h3 class="panel-title">1st Group
Control</h3>
        </div>
        <div class="panel-body">
            <div class="row-md-2"><input id="toggle-
event" type="checkbox" data-toggle="toggle" class="user lock1"> Mode 1</div>
            <div class="row-md-2"><input id="toggle-
event" type="checkbox" data-toggle="toggle" class="user lock1"> Mode 2</div>
            <div class="row-md-2"><input id="toggle-
event" type="checkbox" data-toggle="toggle" class="user lock1"> Mode 3</div>
            <!--div class="row-md-2"><input
id="toggle-event" type="checkbox" data-toggle="toggle" class="user lock1">
Saklar 4 (LED Kuning 2)</div-->
        </div>
    </div>
</div>
<div class="col-md-4">
    <div class="panel panel-warning">
        <div class="panel-heading">
            <h3 class="panel-title">2nd Group
Control</h3>

```

```
</div>
<div class="panel-body">
    <div class="row-md-2"><input id="toggle-
event" type="checkbox" data-toggle="toggle" class="user lock2"> Speaker</div>
</div>
</div>
</div>
</div>
</div>
</body>
<script>
$(document).ready(function() {
    $("#link1").click(function(){
        var i = 0;
        theCheckboxes.each(function(){
            $.get("s.php", {port: i, mode: 1});
            i++;
        });
    });
    $("#link2").click(function(){
        var i = 0;
        theCheckboxes.each(function(){
            $.get("s.php", {port: i, mode: 0});
            i++;
        });
    });
});
```

```
var a = new Webcam(0);
a.start();

var b = new Webcam(1);
b.start();

var c = new Webcam(2);
c.start();

$("#btn1").click(function(){
    b.container.attr("class", "hidden");
    b.stop();
    c.container.attr("class", "hidden");
    c.stop();
    a.dimension(640, 480);
    a.container.attr("class", "col-md-12");
    a.start();
});

$("#btn2").click(function(){
    a.container.attr("class", "hidden");
    a.stop();
    c.container.attr("class", "hidden");
    c.stop();
    b.dimension(640, 480);
    b.container.attr("class", "col-md-12");
    b.start();
});
```

```
$("#btn3").click(function(){
    a.container.attr("class", "hidden");
    a.stop();
    b.container.attr("class", "hidden");
    b.stop();
    c.dimension(640, 480);
    c.container.attr("class", "col-md-12");
    c.start();
});
$("#btn4").click(function(){
    a.stop();
    b.stop();
    c.stop();
    a.dimension(320, 240);
    a.container.attr("class", "col-md-6");
    b.dimension(320, 240);
    b.container.attr("class", "col-md-6");
    c.dimension(320, 240);
    c.container.attr("class", "col-md-6");
    a.start();
    b.start();
    c.start();
});
});
```

```
var theCheckboxes = $("input[type='checkbox'].user");
```

```
$("#input.lock2").change(function() {
```

```
        var index = theCheckboxes.index( $(this) );
        if($(this).is(':checked')) $.get("s.php", {port: index, val: 1 }); else
$.get("s.php", {port: index, val: 0});

});
```

```
$("#input.lock1").change(function() {
    var check2 = theCheckboxes.filter("input.lock1");
    if (check2.filter(":checked").length > 1) {
        //$(this).bootstrapToggle('off') ;
        check2.filter(":checked").bootstrapToggle('off');
        $(this).bootstrapToggle('on');
        return false;
    } else {
        var index = theCheckboxes.index( $(this) );
        //console.log(index);
        if($(this).is(':checked')) $.get("s.php", {port: index, val: 1 }); else
$.get("s.php", {port: index, val: 0});
        //console.log($("#input:checked").val() + ' ' + index);
    }
});
</script>

</body>
</html>
```

Lampiran 2 Program PHP :

```
<?php
error_reporting(-1);
ini_set('display_errors', 'On');

function gpio($port, $value){
    if((int)exec("gpio read " . escapeshellarg($port)) == $value){
        return $value;
    } else {
        exec("gpio write " . escapeshellarg($port) . " " .
escapeshellarg($value));
        return $value;
    }
}

function gpioread($port){
    return (int)exec("gpio read " . escapeshellarg($port));
}

function gpiomode($port, $mode){
    return exec("gpio mode " . escapeshellarg($port) . " " . $mode);
}

if(isset($_GET['port']) && isset($_GET['val'])){
    $port = (int)$_GET['port'];
    $val = (int)$_GET['val'];
    if(is_int($port) && is_int($val)){
        echo gpio($port, $val);
    }
}
```

```
    } else {  
        echo "nub";  
    }  
} elseif(isset($_GET['port']) && isset($_GET['mode'])){  
    $port = (int)$_GET['port'];  
    $mode = (int)$_GET['mode'];  
    if(is_int($port) && is_int($mode)){  
        echo gpiomode($port, ($mode?"in":"out"));  
    } else {  
        echo "nub2";  
    }  
} elseif (isset($_GET['port'])) {  
    $port = (int)$_GET['port'];  
    if(is_int($port)){  
        echo gpioread($port);  
    } else {  
        echo "nub1";  
    }  
} else {  
    echo "gtfo";  
}
```

Lampiran 2 Program PHP :

?>

```
<?php
```

```
error_reporting(-1);
```

```
ini_set('display_errors', 'On');
```

```
function gpio($port, $value){
```

```
    if((int)exec("gpio read " . escapeshellarg($port)) == $value){
```

```
        return $value;
```

```
    } else {
```

```
        exec("gpio write " . escapeshellarg($port) . " " .  
escapeshellarg($value));
```

```
        return $value;
```

```
    }
```

```
}
```

```
function gpioread($port){
```

```
    return (int)exec("gpio read " . escapeshellarg($port));
```

```
}
```

```
function gpiomode($port, $mode){
```

```
    return exec("gpio mode " . escapeshellarg($port) . " " . $mode);
```

```
}
```

```
if(isset($_GET['port']) && isset($_GET['val'])){
```

```
    $port = (int)$_GET['port'];
```

```
    $val = (int)$_GET['val'];
```



```
if(is_int($port) && is_int($val)){
    echo gpio($port, $val);
} else {
    echo "nub";
}
} elseif(isset($_GET['port']) && isset($_GET['mode'])){
    $port = (int)$_GET['port'];
    $mode = (int)$_GET['mode'];
    if(is_int($port) && is_int($mode)){
        echo gpiomode($port, ($mode?"in":"out"));
    } else {
        echo "nub2";
    }
} elseif (isset($_GET['port'])) {
    $port = (int)$_GET['port'];
    if(is_int($port)){
        echo gpioread($port);
    } else {
        echo "nub1";
    }
} else {
    echo "gtfo";
}
?>
```