



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - TE 145561**

***PROTOTYPE SISTEM PENGAWASAN PADA LAMPU LALU  
LINTAS DI PERSIMPANGAN JALAN MENGGUNAKAN  
RASPBERRY PI UNTUK MENCATAT PELANGGARAN LAMPU  
MERAH***

Daniel Nugraha  
NRP 2213030064  
Simeon Dinar Kurniawan  
NRP 2213030094

Dosen Pembimbing  
Suwito, ST., MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - TE 145561**

**PROTOTYPE MONITORING SYSTEM AT CROSSROAD  
TRAFFIC LIGHT USING RASPBERRY PI FOR CAPTURE RED  
LIGHT VIOLATORS**

Daniel Nugraha  
NRP 2213030064  
Simeon Dinar Kurniawan  
NRP 2213030094

Advisor  
Suwito, ST., MT.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM  
Faculty of Industrial Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016

## **PERNYATAAN KEASLIAN TUGAS AKHIR**

Dengan ini kami menyatakan bahwa isi sebagian maupun keseluruhan Tugas Akhir kami dengan judul **“Prototype Sistem Pengawasan Pada Lampu Lalu Lintas di Persimpangan Jalan Menggunakan Raspberry Pi Untuk Mencatat Pelanggaran Lampu Merah”** adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang kami akui sebagai karya sendiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka.

Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, (1 Juni 2016)

Daniel Nugraha  
NRP 2213030064

Simeon Dinar Kurniawan  
NRP 2213030094



**PROTOTYPE SISTEM PENGAWASAN PADA LAMPU LALU  
LINTAS DI PERSIMPANGAN JALAN MENGGUNAKAN  
RASPBERRY PI UNTUK MENCATAT PELANGGARAN  
LAMPU MERAH**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Ahli Madya  
Pada  
Program Studi D3 Teknik Elektro  
Jurusan Teknik Elektro  
Institut Teknologi Sepuluh Nopember**



**SURABAYA  
JUNI, 2016**



-----Halaman ini sengaja dikosongkan-----

# PROTOTYPE SISTEM PENGAWASAN PADA LAMPU LALU LINTAS DI PERSIMPANGAN JALAN MENGGUNAKAN RASPBERRY PI UNTUK MENCATAT PELANGGARAN LAMPU MERAH

**Nama** : Daniel Nugraha, Simeon Dinar Kurniawan  
**Pembimbing** : Suwito, ST., MT.

## ABSTRAK

Lalu lintas dan angkutan jalan mempunyai peran strategis dalam mendukung pembangunan dan integrasi nasional sebagai bagian dari upaya memajukan kesejahteraan umum. Seiring dengan perkembangan dan pembangunan masyarakat, timbul masalah pelanggaran lalu lintas yang mengakibatkan timbulnya ketidaktertiban dan kecelakaan.

Pada Tugas Akhir ini akan dibuat sistem pemantau menggunakan kamera dan sensor jarak untuk mendeteksi pelanggaran lampu lalu lintas. Terdapat 2 buah sensor jarak yang akan mendeteksi pengendara bermotor. Sensor pertama akan bekerja dengan memberi peringatan berupa *alarm* bunyi apabila pengendara bermotor melewati atau berhenti tepat di *stopping line*. Saat pengendara melewati sensor kedua, maka kamera akan mengambil gambar kendaraan yang melanggar tersebut secara otomatis. Untuk mengambil data gambar kendaraan pelanggar digunakan metode *background subtraction*. Selanjutnya gambar kendaraan yang melanggar tersebut akan dikirim ke *database* kantor polisi.

Hasil percobaan menunjukkan bahwa dalam sistem pemantau memerlukan kamera dengan resolusi yang baik sehingga menghasilkan gambar yang jelas serta dalam mengaktifkan sensor jarak memerlukan penyesuaian waktu selama lampu merah aktif.

Dengan menggunakan sistem pemantau ini, diharapkan polisi sebagai pihak yang berwajib untuk menindak pelanggaran lalu lintas dapat melaksanakan tugasnya secara maksimal.

**Kata Kunci** : Lalu Lintas, *Background Subtraction*, Sistem Pemantau, *Database*, Sensor Jarak



***PROTOTYPE MONITORING SYSTEM AT CROSSROAD TRAFFIC LIGHT USING RASPBERRY PI FOR RECORDING PERFORMER RED LIGHT VIOLATORS***

**Name : Daniel Nugraha, Simeon Dinar Kurniawan**  
**Advisor : Suwito, ST., MT.**

***ABSTRACT***

*Traffic and road transport has a strategic role in supporting development and national integration as part of an effort to advance the general welfare. Along with the development and construction of community, a problem was traffic violations which resulted in the incidence of ketidaktertiban and*

*. On this final assignment will be made using a camera monitoring systems and proximity sensor for detecting violations of traffic lights. There are 2 proximity sensor detects a motor rider. The first sensor will work with either a warning alarm sounds when motor riders pass through or stop right in stopping the line. When motorists passing through the second sensor, then the camera will take pictures of the vehicle who violates these automatically. To retrieve image data vehicle background subtraction method used offenders. Further pictures of the vehicle in violation will be sent to the database of the police station. The experiment results show that the monitoring system requires a camera with good resolution so as to produce a clear image as well as to enable proximity sensor requires an adjustment time during red lights on.*

*Using this monitoring system, the police are expected as the party authorities to crack down on traffic violations can carry out its work to its full potential.*

***Keywords : Traffic, Background Subtraction, Monitoring Systems, Databases, Proximity Sensor***



## **KATA PENGANTAR**

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa yang selalu memberikan rahmat dan anugerah-Nya sehingga Tugas Akhir ini dapat terselesaikan dengan baik.

Tugas Akhir ini disusun untuk memenuhi sebagian persyaratan guna menyelesaikan pendidikan Diploma pada Bidang Studi Komputer Kontrol, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya dengan judul:

### **PROTOTYPE SISTEM PENGAWASAN PADA LAMPU LALU LINTAS DI PERSIMPANGAN JALAN MENGGUNAKAN RASPBERRY PI UNTUK MENCATAT PELANGGARAN LAMPU MERAH**

Dalam Tugas Akhir ini dirancang sistem pemantau menggunakan kamera dan sensor jarak untuk mendeteksi pelanggar lampu lalu lintas.

Penulis mengucapkan terima kasih kepada Ibu dan Bapak penulis yang memberikan berbagai bentuk doa serta dukungan tulus tiada henti, Bapak Suwito, ST., MT. atas segala bimbingan ilmu, moral, dan spiritual dari awal hingga terselesaikannya Tugas Akhir ini. Penulis juga mengucapkan banyak terima kasih kepada semua pihak yang telah membantu baik secara langsung maupun tidak langsung dalam proses penyelesaian Tugas Akhir ini.

Penulis menyadari dan memohon maaf atas segala kekurangan pada Tugas Akhir ini. Akhir kata, semoga Tugas Akhir ini dapat bermanfaat dalam pengembangan keilmuan di kemudian hari.

Surabaya, (1 Juni 2016)

Penulis



-----Halaman ini sengaja dikosongkan-----

# DAFTAR ISI

	HALAMAN
HALAMAN JUDUL.....	i
HALAMAN JUDUL.....	i
PERNYATAAN KEASLIAN TUGAS AKHIR.....	v
HALAMAN PEESAHAN.....	<b>Error! Bookmark not defined.</b>
ABSTRAK.....	ix
ABSTRACT.....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR.....	xvii
DAFTAR TABEL.....	xix
<b>BAB I PENDAHULUAN.....</b>	<b>1</b>
1.1 Latar Belakang.....	1
1.2 Permasalahan.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Laporan.....	3
1.7 Relevansi.....	4
<b>BAB II TEORI DASAR.....</b>	<b>5</b>
2.1 Citra Digital[6].....	5
2.2 Pengolahan Citra[1].....	6
2.2.1 Operasi Pengolahan Citra[7].....	6
1. <i>Image enhancement</i> .....	6
2. <i>Image restoration</i> .....	7
3. <i>Image compression</i> .....	7
4. <i>Image segmentation</i> .....	7
5. <i>Image analysis</i> .....	7
6. <i>Image reconstruction</i> .....	7
2.2.2 Proses Pengolahan Citra[7].....	8
1. <i>Grayscale</i> .....	8
2. <i>Thresholding</i> .....	8
3. <i>Background Subtraction</i> .....	9
2.3 <i>Computer Vision</i> [1].....	9
2.4 PHP[2].....	10

2.5 OpenCV[1] .....	10
2.6 Sistem Terdistribusi .....	11
2.6.1 Client-Server .....	11
2.7 Raspberry Pi.....	14
2.8 Webcam.....	16
2.9 Sensor Ultrasonik HC-SR04 .....	16
2.10 Arduino Uno R3.....	18
2.11 MySQL .....	19
<b>BAB III PERANCANGAN DAN PEMBUATAN SISTEM.....</b>	<b>21</b>
3.1 Perancangan <i>Software</i> .....	21
3.1.1 Perancangan <i>Database</i> .....	22
3.1.2 <i>Background Subtraction</i> .....	23
3.2 Perancangan <i>Hardware</i> .....	24
3.2.1 Sensor Ping .....	25
<b>BAB IV HASIL SIMULASI DAN IMPLEMENTASI .....</b>	<b>27</b>
4.1 Perangkat Keras .....	27
4.2 Perangkat Lunak .....	29
4.2.1 Pengolahan Citra.....	29
4.2.2 <i>Database</i> .....	31
<b>BAB V PENUTUP .....</b>	<b>33</b>
5.1 Kesimpulan .....	33
5.2 Saran .....	33
<b>DAFTAR PUSTAKA .....</b>	<b>35</b>
<b>LAMPIRAN A .....</b>	<b>37</b>
<b>LAMPIRAN B .....</b>	<b>43</b>
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>45</b>
<b>DAFTAR RIWAYAT HIDUP .....</b>	<b>47</b>

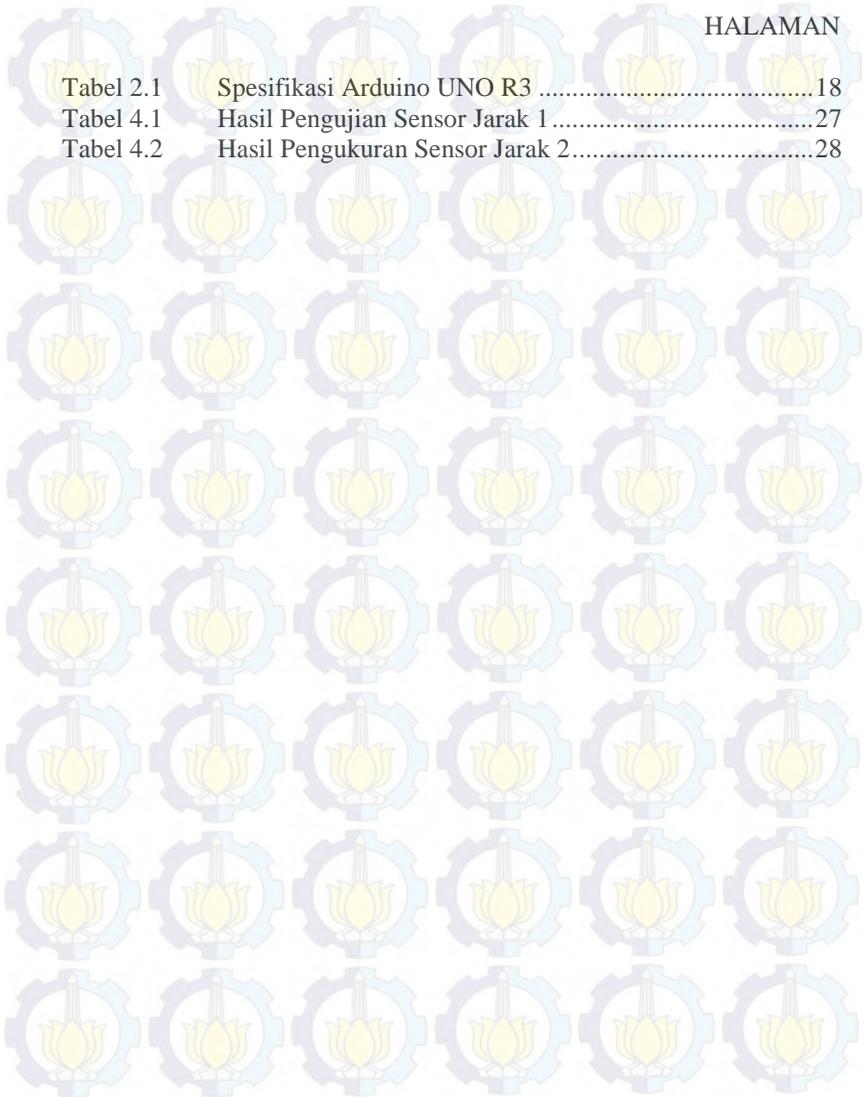
## DAFTAR GAMBAR

	HALAMAN
Gambar 1.1 Ilustrasi Penempatan Kamera pada Pertigaan .....	2
Gambar 2.1 Citra Digital .....	5
Gambar 2.2 Gambar Derajat Keabuan.....	8
Gambar 2.3 Gambar <i>Thin Client</i> .....	12
Gambar 2.4 Sistem <i>Client Server Two Tier</i> .....	13
Gambar 2.5 <i>Hardware</i> Raspberry Pi Model B+ .....	14
Gambar 2.6 Konfigurasi Pin GPIO Raspberry Pi .....	15
Gambar 2.7 <i>Webcam</i> Snowwolf M12.....	16
Gambar 2.8 Sensor Ultrasonik HC-SR04 .....	17
Gambar 2.9 <i>Timing Diagram</i> HC-SR04 .....	17
Gambar 2.10 Arduino UNO R3 .....	18
Gambar 3.1 Blok Diagram Sistem .....	21
Gambar 3.2 Sketsa Miniatur Persimpangan Jalan .....	24
Gambar 3.3 Konfigurasi Arduino dengan Sensor Ping.....	25
Gambar 4.1 Grafik Perbandingan Antara Hasil Pengukuran Sensor HC-SR04 Dengan Penggaris.....	28
Gambar 4.2 Gambar Citra Saat Proses <i>Thresholding</i> .....	29
Gambar 4.3 Hasil <i>Frame Differencing</i> .....	30
Gambar 4.4 Hasil Pemantauan .....	31
Gambar 4.5 Tampilan data gambar pada phpMyAdmin.....	31
Gambar 4.6 Tampilan Gambar pada <i>Website</i> .....	32



## DAFTAR TABEL

	HALAMAN
Tabel 2.1 Spesifikasi Arduino UNO R3 .....	18
Tabel 4.1 Hasil Pengujian Sensor Jarak 1 .....	27
Tabel 4.2 Hasil Pengukuran Sensor Jarak 2.....	28





-----Halaman ini sengaja dikosongkan-----

**LEMBAR PERNYATAAN  
PERSETUJUAN PUBLIKASI KARYA ILMIAH  
UNTUK KEPENTINGAN AKADEMIS**

Sebagai mahasiswa Institut Teknologi Sepuluh Nopember Surabaya, yang bertanda tangan di bawah ini saya :

Nama : Simeon Dinar Korniawan  
Nrp. : 2213030099  
Jurusan / Fak. : D3 T. Elektro / FTI  
Alamat kontak : Jl. Asem Payung I No.39, Surabaya  
a. Email : simeondkorniawan@gmail.com  
b. Telp/HP : 0856 55896309

Menyatakan bahwa semua data yang saya *upload* di Digital Library ITS merupakan hasil final (revisi terakhir) dari karya ilmiah saya yang sudah disahkan oleh dosen penguji. Apabila dikemudian hari ditemukan ada ketidaksesuaian dengan kenyataan, maka saya bersedia menerima sanksi.

Demikian perkembangan ilmu pengetahuan, saya menyetujui untuk memberikan **Hak Bebas Royalti Non-Eksklusif (Non-Exclusive Royalti-Free Right)** kepada Institut Teknologi Sepuluh Nopember Surabaya atas karya ilmiah saya yang berjudul :

Prototype Sistem Pengawasan Pada Lampu Lalu - Lintas di Perempatan Jalan menggunakan Rbpberry Pi untuk Mencatat Pelanggaran Lampu Merah

Dengan Hak Bebas Royalti Non-Eksklusif ini, Institut Teknologi Sepuluh Nopember Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (*database*), mendistribusikannya, dan menampilkan/mempublikasikannya di internet atau media lain untuk kepentingan akademis tanpa meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta. Saya bersedia menanggung secara pribadi, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya Ilmiah saya ini tanpa melibatkan pihak Institut Teknologi Sepuluh Nopember Surabaya.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Surabaya  
Pada tanggal : 27 Juni 2016  
Yang menyatakan,

Dosen Pembimbing 1

Sawir, S.T., M.P., Ph.D.  
Fakultas Teknik  
Institut Teknologi Sepuluh Nopember  
Surabaya

Simeon Dinar K.

NIP. 19710105 200501 1009

Nrp. 2213030099

# BAB I

## PENDAHULUAN

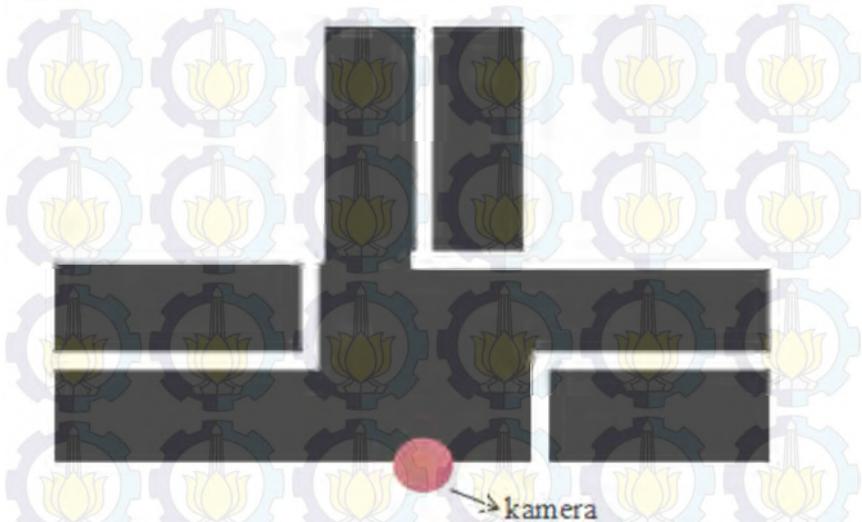
### 1.1 Latar Belakang

Lalu lintas jalan merupakan sarana masyarakat yang memegang peranan penting dalam memperlancar pembangunan yang pemerintah laksanakan, karena merupakan sarana untuk masyarakat maka sudah sepatutnya masyarakat berpartisipasi dalam menjaga ketertiban umum di jalan. Timbulnya masalah lalu lintas merupakan salah satu masalah yang berkembang seiring dengan perkembangan dan pembangunan masyarakat. Antara lain adalah masalah pelanggaran lalu lintas yang cenderung mengakibatkan timbulnya ketidaktertiban dan kecelakaan dalam masyarakat. Pelanggaran lalu lintas merupakan suatu keadaan dimana terjadi ketidakseuaian antara aturan dan pelaksanaan. Aturan dalam hal ini adalah piranti hukum yang telah ditetapkan dan disepakati oleh negara sebagai undang-undang yang berlaku secara sah, sedangkan pelaksanaannya adalah manusia atau masyarakat suatu negara yang terikat oleh piranti hukum tersebut. Dalam Undang-Undang Nomor 22 Tahun 2009 Tentang Lalu Lintas dan Angkutan Jalan pasal 106 ayat 4 menyebutkan bahwa setiap pengendara kendaraan bermotor wajib mematuhi rambu lalu-lintas, dan isyarat lalu-lintas.

Pengaturan yang dilakukan oleh pemerintah menyangkut setiap pengguna jalan, dimana setiap individu diharapkan dapat melaksanakan peraturan dalam berlalu lintas, tidak terkecuali siapapun selama mereka berada di jalan mereka tidak sekedar berjalan atau mengemudi, tetapi juga memperhatikan adanya aturan dalam berlalu lintas guna kelancaran bersama. Pihak yang bertanggung jawab untuk mengawasi para pengendara agar mentaati aturan yang telah dibuat adalah polisi. Namun tidak semua tempat ada polisi yang mengawasinya. Untuk itu dibuat sistem pengawasan yang dibangun menggunakan perangkat Raspberry Pi, kamera, serta sensor jarak yang berguna untuk mendeteksi adanya pelanggaran garis henti pada alat pemberi isyarat lalu-lintas di pertigaan jalan. Kamera diletakkan pada bagian tengah pertigaan seperti pada gambar 2.1.

Untuk mencatat pelanggaran digunakan metode *background subtraction*. *Background subtraction* adalah proses untuk mendeteksi pergerakan atau perbedaan signifikan yang terjadi didalam *frame* video ketika dibandingkan dengan citra referensi. Harapan kami alat ini nantinya akan membantu kepolisian untuk mengawasi persimpangan

jalan yang tidak dijaga. Serta akan mengurangi tingkat pelanggaran lalu-lintas.



**Gambar 1.1** Ilustrasi Penempatan Kamera pada Pertigaan

## 1.2 Permasalahan

Tingkat pelanggaran lalu-lintas semakin hari semakin meningkat. Dibutuhkan petugas untuk mengawasi pengendara agar peraturan yang telah dibuat oleh pemerintah terlaksana dengan baik. Namun kenyataannya tidak semua area terdapat petugas untuk mengawasi. Selain itu tingkat kesadaran pengendara akan pentingnya ketertiban berlalu-lintas semakin menurun.

## 1.3 Batasan Masalah

Pengambilan gambar dilakukan menggunakan modul *webcam* yang memiliki resolusi VGA menggunakan metode *background subtraction* dan Pengambilan gambar dilakukan hanya saat keadaan terang.

## 1.4 Tujuan

Perancangan Alat ini bertujuan untuk mengawasi area persimpangan jalan, mencatat pelanggaran lampu merah dan meningkatkan kesadaran pengguna jalan untuk mematuhi rambu lalu lintas.

## 1.5 Metodologi Penelitian

Penelitian ini dilakukan melalui beberapa tahapan metodologi, yaitu, studi literatur, perancangan alat, simulasi hasil desain alat, implementasi dan analisis data yang diperoleh, dan yang terakhir adalah penyusunan laporan berupa buku Tugas Akhir.

Pada tahap studi literatur akan dipelajari mengenai *image processing* menggunakan software OpenCV, *background subtraction*, mini komputer Raspberry Pi dan komunikasi data antara Raspberry Pi dengan laptop *server*. Pada tahap pemodelan sistem, dilakukan perancangan *hardware* setelah semua komponen lengkap dengan data cara pembuatan yang diperoleh dari studi literatur. Data percobaan yang telah diperoleh selanjutnya akan dianalisis. Selanjutnya dilakukan pembuatan program pada Raspberry Pi untuk bisa mengambil gambar menggunakan metode *background subtraction* dengan *library* OpenCV. Pada tahap pengujian dan analisa akan dilakukan pengujian alat, menganalisa kesalahan atau kegagalan pada alat serta mengatasi masalah tersebut. Dari hasil analisis, akan ditarik kesimpulan dari penelitian yang telah dilakukan. Tahap akhir penelitian adalah penyusunan laporan penelitian.

## 1.6 Sistematika Laporan

Pembahasan Tugas Akhir ini akan dibagi menjadi lima Bab dengan sistematika sebagai berikut:

### Bab I      **Pendahuluan**

Bab ini meliputi latar belakang, permasalahan, tujuan penelitian, metodologi penelitian, sistematika laporan, dan relevansi.

### Bab II     **Teori Dasar**

Bab ini menjelaskan tentang tinjauan pustaka, konsep dari *image processing* dengan metode *background subtraction*, Raspberry Pi, OpenCV, sensor jarak .

### Bab III    **Perancangan Sistem**

Bab ini membahas desain dan perancangan sistem kamera pengawas berdasarkan teori dasar pada Bab II.

### Bab IV    **Simulasi, Implementasi dan Analisis Sistem**

Bab ini memuat hasil simulasi dan implementasi serta analisis dari hasil tersebut.

## **Bab V Penutup**

Bab ini berisi kesimpulan dan saran dari hasil pembahasan yang telah diperoleh.

### **1.7 Relevansi**

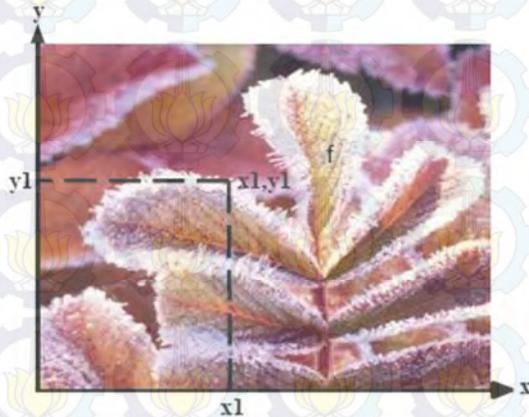
Hasil yang diperoleh dari Tugas Akhir ini diharapkan menjadi referensi pembuatan sistem kamera pengawas, perbandingan dan pengembangan *image processing* menggunakan tipe *processor* dan kamera tertentu.

## BAB II TEORI DASAR

Dalam bab ini akan dibahas mengenai hal-hal yang mendukung terlaksananya tugas akhir dengan baik, khususnya dalam hal landasan teori.

### 2.1 Citra Digital[6]

Citra digital adalah citra dua dimensi yang dapat ditampilkan pada layar monitor komputer sebagai himpunan berhingga (diskrit) nilai digital yang disebut pixel (picture elements). Pixel adalah elemen citra yang memiliki nilai yang menunjukkan intensitas warna. Berdasarkan cara penyimpanan atau pembentukannya, citra digital dapat dibagi menjadi dua jenis. Jenis pertama adalah citra digital yang dibentuk oleh kumpulan pixel dalam array dua dimensi. Citra jenis ini disebut citra bitmap atau citra raster. Jenis citra yang kedua adalah citra yang dibentuk oleh fungsi-fungsi geometri dan matematika. Jenis citra ini disebut grafik vektor. Citra digital (diskrit) dihasilkan dari citra analog (kontinu) melalui digitalisasi. Gambar 2.1 menunjukkan ilustrasi dari citra digital



**Gambar 2.1** Citra Digital

Proses digitalisasi citra analog terdiri *sampling* dan *quantitazion*. *Sampling* adalah pembagian citra ke dalam elemen-elemen diskrit (pixel), sedangkan *quantitazion* adalah pemberian nilai intensitas warna pada setiap pixel dengan nilai yang berupa bilangan bulat. Citra digital dapat didefinisikan sebagai fungsi dua variabel,  $f(x,y)$ , dimana  $x$  dan  $y$

adalah koordinat spasial dan nilai  $f(x,y)$  adalah *brightness* level dari citra pada koordinat tersebut. Teknologi dasar untuk menciptakan dan menampilkan warna pada citra digital berdasarkan pada penelitian bahwa sebuah warna merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru (*Red, Green, Blue - RGB*).

## 2.2 Pengolahan Citra[1]

Pengolahan citra atau *image processing* adalah suatu sistem dimana proses dilakukan dengan masukan (*input*) berupa citra (*image*) dan hasilnya (*output*) juga berupa citra (*image*). Pada awalnya pengolahan citra ini dilakukan untuk memperbaiki kualitas citra, namun dengan berkembangnya dunia komputasi yang ditandai dengan semakin meningkatnya kapasitas dan kecepatan proses komputer, serta munculnya ilmu-ilmu komputer yang memungkinkan manusia dapat mengambil informasi dari suatu citra maka *image processing* tidak dapat dilepaskan dengan bidang *computer vision*. Meskipun sebuah citra kaya informasi, namun seringkali citra yang kita miliki mengalami penurunan mutu (degradasi), misalnya mengandung cacat atau derau (*noise*), warnanya terlalu kontras, kurang tajam, kabur (*blurring*) dan sebagainya. Tentu saja citra semacam ini menjadi lebih sulit diinterpretasikan karena informasi yang disampaikan oleh citra tersebut menjadi berkurang. Agar citra yang mengalami gangguan mudah diinterpretasikan (baik oleh manusia maupun mesin), maka citra tersebut perlu dimanipulasi menjadi citra lain yang kualitasnya lebih baik. Pengolahan citra memiliki 2 sub, yaitu operasi dan proses.

### 2.2.1 Operasi Pengolahan Citra[7]

Operasi yang dilakukan dalam pengolahan citra ada berbagai jenis. Secara umum, operasi pengolahan citra dapat dikelompokkan dalam beberapa jenis sebagai berikut,

#### 1. *Image enhancement*

Operasi ini bertujuan untuk memperbaiki kualitas citra dengan cara memanipulasi berbagai parameter di dalam citra. Dengan melakukan operasi ini, ciri khusus yang ada dalam citra bisa lebih ditonjolkan. Contoh operasi perbaikan citra atau *image enhancement*:

- a. Perbaikan kontras gelap/terang
- b. Perbaikan tepian objek (*edge enhancement*)
- c. *Sharpening*

- d. *Grayscaleing*
- e. *Noise filtering*

## 2. *Image restoration*

Operasi ini digunakan untuk mengurangi cacat pada citra. Tujuan operasi *image restoration* hampir sama dengan *image enhancement*. Bedanya pada *image restoration*. Contoh operasi *image restoration* adalah:

- a. *Deblurring*
- b. Penghilangan *noise*

## 3. *Image compression*

Jenis operasi ini dilakukan untuk memampatkan citra sehingga dapat direpresentasikan dalam bentuk yang lebih kompak dan memerlukan memori yang lebih sedikit. Kualitas citra setelah proses pemampatan harus sangat diperhatikan. Citra yang telah dimampatkan harus tetap memiliki kualitas gambar yang bagus. Salah satu contoh metode pemampatan adalah metode JPEG.

## 4. *Image segmentation*

Jenis operasi ini bertujuan untuk memecah suatu citra ke dalam beberapa *segment* dengan suatu kriteria tertentu. Operasi ini biasanya dilakukan untuk pengenalan pola.

## 5. *Image analysis*

Jenis operasi ini bertujuan menghitung besaran kuantitatif dari citra untuk menghasilkan deskripsinya. Teknik analisis citra mengekstraksi ciri- ciri tertentu yang membantu dalam identifikasi objek. Proses segmentasi kadangkala diperlukan untuk melokalisasi objek yang diinginkan dari sekelilingnya. Contoh operasi pengorakan citra:

- a. *Edge detection*
- b. *Boundary*
- c. *Region*

## 6. *Image reconstruction*

Jenis operasi ini bertujuan untuk membentuk ulang objek dari beberapa citra hasil proyeksi. Operasi rekonstruksi citra banyak

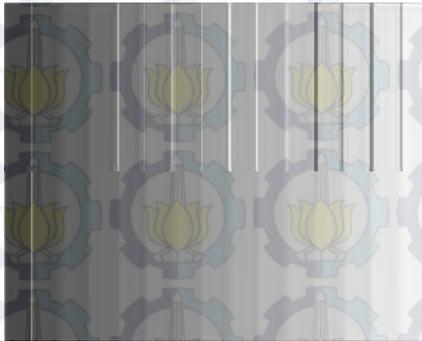
digunakan dalam bidang medis. Misalnya beberapa foto rontgen dengan sinar X digunakan untuk membentuk ulang gambar organ tubuh.

### 2.2.2 Proses Pengolahan Citra[7]

#### 1. *Grayscale*

Proses awal yang banyak dilakukan dalam Pengolahan Citra adalah mengubah citra berwarna menjadi citra *grayscale*. Hal ini digunakan untuk menyederhanakan model citra.

Citra terdiri dari 3 layer matrik yaitu *R-layer*, *G-layer* dan *B-layer*. Ketiga layer ini perlu diperhatikan untuk melakukan proses selanjutnya. Jika setiap proses perhitungan dilakukan menggunakan tiga layer, maka akan dilakukan tiga perhitungan yang sama, sehingga konsep itu diubah dengan mengubah 3 layer menjadi 1 layer matrik *grayscale* dan hasilnya adalah citra *grayscale*. Proses ini menghasilkan citra yang bukan berupa warna, namun berupa derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matrik masing-masing *r*, *g* dan *b* menjadi citra *grayscale* dengan nilai *s*, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai *r*, *g* dan *b*. Gambar 2.2 menunjukkan contoh dari derajat keabuan.



**Gambar 2.2** Gambar Derajat Keabuan

#### 2. *Thresholding*

*Thresholding* digunakan untuk mengatur jumlah derajat keabuan yang ada pada citra. *Thresholding* dapat mengubah derajat keabuan sesuai dengan keinginan. Proses *thresholding* pada dasarnya adalah proses pengubahan kuantisasi pada citra, sehingga untuk melakukan *thresholding* dengan derajat keabuan dapat digunakan persamaan (1)

$$x = \frac{w}{b} \dots\dots\dots (1)$$

Dimana:

$x$  adalah nilai derajat keabuan setelah *thresholding*

$w$  adalah nilai derajat keabuan sebelum *thresholding*

$b$  adalah nilai derajat keabuan yang diinginkan

### 3. *Background Subtraction*

Subtraksi latar memiliki peranan penting dalam visi komputer terutama dalam hal sistem pemantauan. Peranan yang dilakukan dalam subtraksi latar adalah untuk mengetahui atau membedakan bagian latar dan objek yang ada pada sebuah citra. Untuk mengetahui adanya objek di dalam citra, hal yang harus dilakukan adalah mempelajari dan mengetahui model dari latar. Ide dasar dari subtraksi latar yaitu:

$$\begin{aligned} |background(r) - image(r)| &= frame(r) \\ |background(g) - image(g)| &= frame(g) \\ |background(b) - image(b)| &= frame(b) \\ frame(r) &> threshold \\ frame(g) &> threshold \\ frame(b) &> threshold \end{aligned}$$

### 2.3 *Computer Vision*[1]

*Computer vision* merupakan sebuah proses otomatis yang mengintegrasikan sejumlah besar proses persepsi visual, seperti pengolahan citra, klasifikasi citra, pengenalan citra dan akuisisi citra. *Computer vision* didefinisikan sebagai salah satu cabang ilmu pengetahuan yang mempelajari bagaimana komputer dapat mengenali obyek yang diamati atau diobservasi. Cabang ilmu ini bersama kecerdasan buatan (*Artificial Intelligence*) akan mampu menghasilkan sistem kecerdasan visual (*Visual Intelligence System*).

Proses-proses dalam *computer vision* dapat dibagi menjadi tiga aktivitas:

1. Memperoleh atau mengakuisisi citra digital.
2. Melakukan teknik komputasi untuk memproses atau memodifikasi data citra.
3. Menganalisis dan menginterpretasi citra dan menggunakan hasil pemrosesan untuk tujuan tertentu, misalnya memandu robot, mengontrol peralatan, memantau proses manufaktur, dan lain-lain.

## 2.4 PHP[2]

PHP adalah bahasa pemrograman *script server-side* yang didesain untuk pengembangan *web*. PHP banyak dipakai untuk memprogram situs web dinamis. PHP disebut bahasa pemrograman *server-side* karena PHP diproses pada komputer *server*.

PHP hanya mengeksekusi kode yang ditulis dalam pembatas sebagaimana ditentukan oleh dasar sintaks PHP. Apapun di luar pembatas tidak diproses oleh PHP (meskipun teks PHP ini masih mengendalikan struktur yang dijelaskan dalam kode PHP. Pembatas yang paling umum adalah "<?php" untuk membuka dan "?>" Untuk menutup kode PHP. Tujuan dari pembatas ini adalah untuk memisahkan kode PHP dari kode di luar PHP, seperti HTML, Javascript. PHP memiliki 3 jenis sintaks sebagai komentar pada kode yaitu tanda blok, /\* \*/ komentar 2 baris // serta tanda pagar # digunakan untuk komentar satu baris. Komentar bertujuan untuk meninggalkan catatan pada kode PHP dan tidak akan diterjemahkan ke program.

## 2.5 OpenCV[1]

OpenCV merupakan *open source computer vision library*. Library ini ditulis dalam bahasa pemrograman C dan C++ dan bisa dioperasikan di Linux, Windows, dan MacOS X. OpenCV dirancang untuk meningkatkan efisiensi komputasi dan dengan fokus pada aplikasi *realtime*. Salah satu tujuan dari OpenCV adalah untuk menyediakan infrastruktur *computer vision* yang mudah digunakan untuk membantu orang membangun aplikasi pencitraan yang cukup canggih dengan cepat[..]

OpenCV didesain untuk aplikasi *real-time*, memiliki fungsi-fungsi akuisisi yang baik untuk *image/video*. OpenCV juga menyediakan *interface* ke *Integrated Performance Primitives* (IPP) Intel sehingga jika anda bisa mengoptimasi aplikasi *computer vision* anda jika menggunakan prosesor Intel.

Fitur yang dimiliki OpenCV antara lain :

1. Manipulasi data citra (*allocation, copying, setting, convert*).
2. Citra dan *video* I/O (file dan kamera *based input, image/video file output*).
3. Manipulasi Matriks dan Vektor beserta rutin-rutin aljabar linear (*products, solvers, eigenvalues, SVD*).
4. Data struktur dinamis (*lists, queues, sets, trees, graphs*).

5. Pemroses citra *fundamental* (*filtering, edge detection, corner detection, sampling and interpolation, color conversion, morphological operations, histograms, image pyramids*).
6. Analisis struktur (*connected components, contour processing, distance Transform, various moments, template matching, Hough Transform, polygonal approximation, line fitting, ellipse fitting, Delaunay triangulation*).
7. Kalibrasi kamera (*calibration patterns, estimasi fundamental matrix, estimasi homography, stereo correspondence*).
8. Analisis gerakan (*optical flow, segmentation, tracking*).
9. Pengenalan obyek (*eigen-methods, HMM*).
10. *Graphical User Interface* (*display image/video, penanganan keyboard dan mouse handling, scroll-bars*).  
 OpenCV terdiri dari 3 *library*, yaitu:
  1. CV : Untuk algoritma *Image Processing* dan *Vision*
  2. Highgui : Untuk GUI, *Image* dan *Video I/O*
  3. CXCORE : Untuk struktur data, *support XML* dan fungsi-fungsi grafis.

## 2.6 Sistem Terdistribusi

Sistem terdistribusi adalah sistem dimana pemrosesan informasi didistribusikan pada beberapa komputer dan tidak terbatas hanya pada satu mesin saja. Dengan kata lain sistem terdistribusi adalah suatu perangkat lunak sistem yang berjalan pada kelompok prosesor yang bekerja sama dan terintegrasi secara *loggar*, dengan dihubungkan oleh jaringan. Pada sistem terdistribusi ada dua tipe *generic* , yaitu:

### 1. Arsitektur Client Server

Sistem berperan sebagai satu set layanan yang disediakan untuk *client*. Ada perbedaan antara *server* dan *client*, pembagian tugasnya pun berbeda.

### 2. Arsitektur Objek terdistribusi.

Pada sistem ini tidak ada perbedaan antara *server* dan *client*. Sistem dapat berperan sebagai satu set objek yang berinteraksi yang aplikasinya tidak relevan, sehingga pada arsitektur ini *server* dan *client* mempunyai derajat yang sama.

### 2.6.1 Client-Server

Pada sistem ini terdapat satu atau lebih proses *client* dan satu atau lebih proses *server*, dan sebuah proses *client* dapat mengirimkan *query*

ke sembarang proses *server*. *Client* bertanggung jawab pada antar muka untuk *user*, sedangkan *user* bertanggung jawab untuk mengatur data dan mengeksekusi transaksi. Hal ini diperlukan agar suatu proses *client* berjalan pada sebuah personal komputer dan mengirimkan *query* ke sebuah *server* pada *mainframe*.

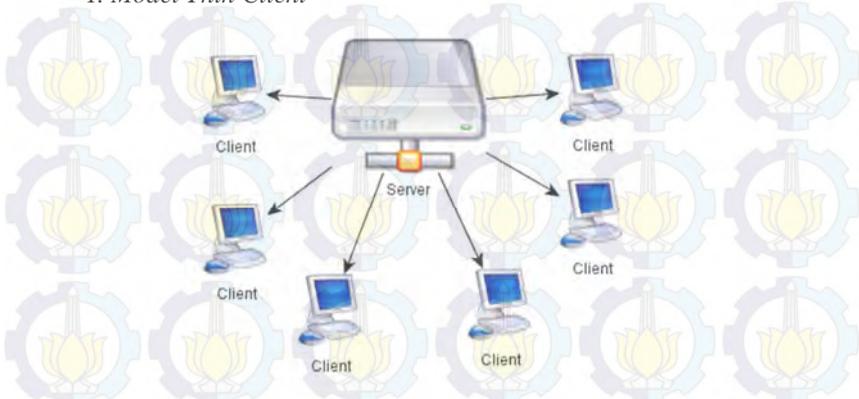
Arsitektur ini banyak digunakan karena implementasinya yang sederhana, *server* tidak terpengaruh pada interaksi *user* pada *client*, selain itu *user* dapat menjalankan *interface* berbasis grafis.

Perancangan arsitektur *client server* harus mempertimbangkan proses logika aplikasi yang dibagi menjadi menjadi tiga lapisan, yaitu:

- Lapisan presentasi, yang berhubungan dengan penyajian informasi ke *user* dengan semua interaksi *user*
- Lapisan pemrosesan aplikasi, lapisan yang berhubungan dengan implementasi logika aplikasi.
- Lapisan manajemen data, lapisan yang berhubungan dengan operasi *database*.

Arsitektur *client server* yang paling sederhana adalah *client server* two tier, dimana aplikasi diorganisir seperti *server* dan satu set *client*. *Arsitektur Two Tier* memiliki dua bentuk, yaitu:

#### 1. Model *Thin Client*



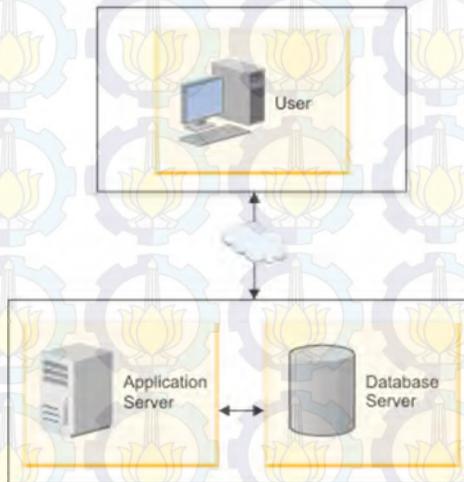
**Gambar 2.3** Gambar *Thin Client*

Pada model ini semua pemrosesan aplikasi dan manajemen data dilakukan pada *server*. Klien bertanggung jawab menjalankan *interface* sistem atau bisa berupa GUI. Arsitektur ini mempunyai beberapa keunggulan, diantaranya biaya yang digunakan lebih sedikit selain itu arsitektur ini sangat cocok jika digunakan pada model jaringan yang

sederhana. Tetapi arsitektur ini juga mempunyai kelemahan, pemrosesan yang berat dibebankan pada *server*. Gambar 2.3 merupakan gambar dari model *thin Client*.

## 2. Model *Fat Client*

Pada model ini *server* hanya bertanggung jawab pada manajemen data. *Client* bertanggung jawab pada logika aplikasi dan interaksi dengan *user*. Keunggulan arsitektur ini adalah *server* hanya menangani seluruh transaksi database. Menggunakan pemrosesan yang besar dan mendistribusikan pemrosesan logika aplikasi dan presentasi pada *client*. Tetapi dengan menggunakan sistem ini, sistem menjadi kompleks dan biayanya cukup mahal.



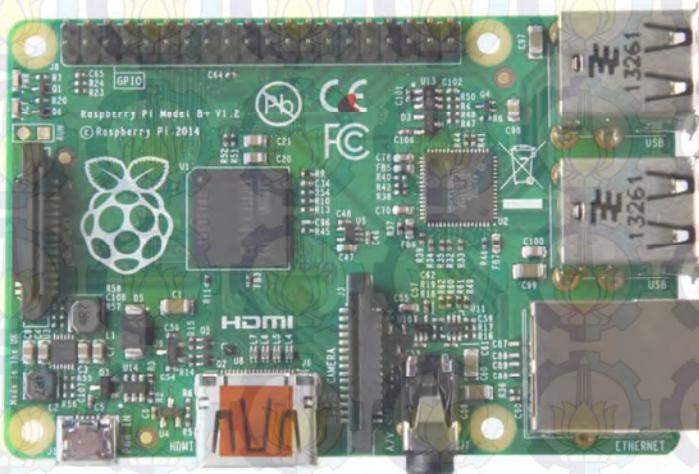
**Gambar 2.4** Sistem *Client Server Two Tier*

Penerapan arsitektur *client server two tier* dapat dijumpai pada jaringan mesin ATM. Pada jaringan ATM, ATM tidak terhubung langsung dengan *database* nasabah, tetapi melalui monitor teleprocessing. Dalam hal ini monitor teleprocessing berperan sebagai middleware yang mengatur komunikasi dengan *client* jarak jauh dan meneruskan transaksi *client* untuk diproses di *database*. Dengan transaksi seri maka apabila terjadi pada *client* maka sistem akan pulih kembali tanpa merusak data sistem. Hal terpenting pada arsitektur *two tier* adalah ketiga logika harus dipetakan kedalam dua sistem komputer. Masalah skalabilitas mungkin bisa terjadi jika arsitektur ini diterapkan

pada model *thin client*. Gambar 2.4 menampilkan gambar dari sistem *client server two tier*.

## 2.7 Raspberry Pi

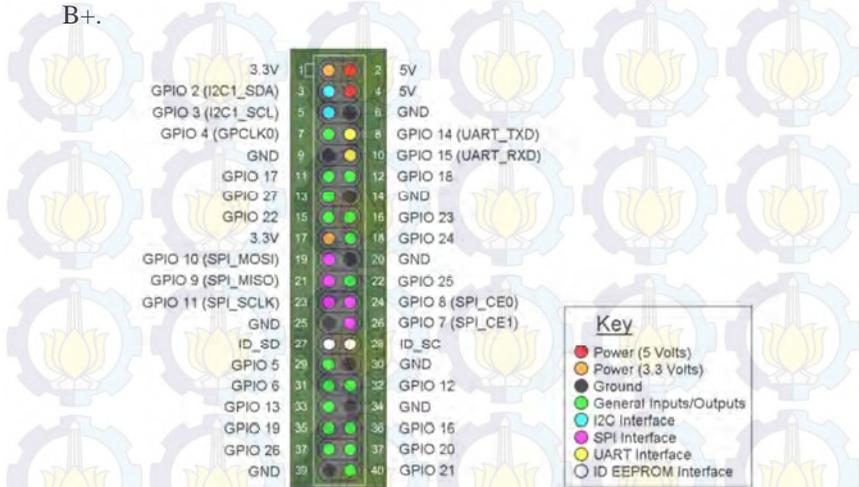
Raspberry Pi adalah komputer papan tunggal (*Single Board Circuit*) yang dikembangkan oleh yayasan nirlaba, Raspberry Pi Foundation yang digawangi sejumlah developer dan ahli komputer dari Universitas Cambridge, Inggris. RasPi menggunakan *system on a chip* (SoC) dari Brodcom BCM2835. Selain itu Raspi menggunakan *processor* ARM1176JZF-S 700 MHz, GPU *VideoCore* IV dan RAM sebesar 512MB. Gambar 2.5 menunjukkan bentuk fisik dari Raspberry Pi Model B+.



**Gambar 2.5** Hardware Raspberry Pi Model B+

Berbeda dengan komputer pada umumnya, mini komputer ini menggunakan *SD Card* untuk proses *booting* dan penyimpanan pada proses yang lama. Perangkat ini menggunakan *operating system*(OS) yang berbeda dengan komputer pada umumnya. OS yang digunakan berbasis linux dengan berbagai jenis , beberapa diantaranya yang sudah sering dipakai di masyarakat umum adalah raspbian dan Fedora. Dalam Raspberry Pi, terdapat beberapa fitur yang dapat diakses yaitu, dua *port* USB, satu *port* HMI, satu *port* RCA untuk video *output*, satu *port*

SDCARD, satu port RJ45 (*Ethernet*), delapan GPIO, UART,SPI BUS.  
 Gambar 2.6 menunjukkan konfigurasi dari pin GPIO Raspberry Pi model B+.



**Gambar 2.6** Konfigurasi Pin GPIO Raspberry Pi

Spesifikasi dari Raspberry Pi model B+ adalah sebagai berikut:

- a. Broadcom BCM2835 SoC
- b. 700 MHz ARM1176JZF-S core CPU
- c. Broadcom VideoCore IV GPU
- d. 512 MB RAM
- e. 4 x USB2.0 Ports with up to 1.2A output
- f. Expanded 40-pin GPIO Header
- g. Video/Audio Out via 4-pole 3.5mm connector, HDMI, or Raw LCD (DSI)
- h. Storage: microSD
- i. 10/100 Ethernet (RJ45)
- j. 27 x GPIO
- k. UART
- l. I2C bus
- m. SPI bus with two chip selects
- n. Power Requirements: max 5V and minimal @ 700 mA via MicroUSB or GPIO Header
- o. Supports Debian GNU/Linux

## 2.8 Webcam

*Web camera* atau yang biasa dikenal dengan *webcam*, adalah kamera yang gambarnya dapat di akses menggunakan *world wide web* (www), program *instant messaging*, atau aplikasi komunikasi dengan tampilan video pada PC.

*Webcam* juga digambarkan sebagai kamera video digital yang sengaja didesain sebagai kamera dengan resolusi rendah. *webcam* dapat digunakan untuk sistem keamanan. Pada beberapa *webcam*, ada yang di lengkapi dengan *software* yang mampu mendeteksi pergerakan dan suara. Dengan *software* tersebut, memungkinkan PC yang terhubung ke kamera untuk mengamati pergerakan dan suara, serta merekamnya ketika terdeteksi. Hasil rekaman ini bisa disimpan pada komputer, e-mail atau di *upload* ke internet. *Webcam* sangat bermanfaat dalam bidang telekomunikasi, bidang keamanan dan bidang industri. Sebagai contoh *webcam* digunakan untuk *videocall chatting*, *surveillance camera*, dan sebagai *video conference* oleh beberapa *user*. Gambar 2.7 menunjukkan bentuk fisik dari *webcam* Snowwolf M12.



**Gambar 2.7** *Webcam* Snowwolf M12

## 2.9 Sensor Ultrasonik HC-SR04

Sensor ini merupakan sensor ultrasonik siap pakai, satu alat yang berfungsi sebagai pengirim, penerima, dan pengontrol gelombang ultrasonik. Alat ini bisa digunakan untuk mengukur jarak benda dari 2cm - 4m dengan akurasi 3mm.

Sensor ini memiliki 4 pin, pin Vcc, Gnd, Trigger, dan Echo. Pin Vcc untuk listrik positif dan Gnd untuk ground-nya. Pin Trigger untuk trigger keluarnya sinyal dari sensor dan pin Echo untuk menangkap

sinyal pantul dari benda. Cara menggunakan alat ini yaitu: ketika kita memberikan tegangan positif pada pin Trigger selama 10 $\mu$ s, maka sensor akan mengirimkan 8 step sinyal ultrasonik dengan frekuensi 40kHz. Selanjutnya, sinyal akan diterima pada pin Echo. Untuk mengukur jarak benda yang memantulkan sinyal tersebut, maka selisih waktu ketika mengirim dan menerima sinyal digunakan untuk menentukan jarak benda tersebut. Gambar 2.8 menunjukkan bentuk fisik dari modul sensor ultrasonik HC-SR04. Proses agar sensor ini dapat menentukan jarak dari suatu objek dapat dilihat pada gambar 2.9.



**Gambar 2.8** Sensor Ultrasonik HC-SR04



**Gambar 2.9** Timing Diagram HC-SR04

## 2.10 Arduino Uno R3

Arduino UNO adalah sebuah *board* mikrokontroler yang didasarkan pada ATmega328. Arduino UNO mempunyai 14 pin digital *input/output* (6 di antaranya dapat digunakan sebagai *output* PWM), 6 *input* analog, sebuah osilator Kristal 16 MHz, sebuah koneksi USB, sebuah power jack, sebuah ICSP *header*, dan sebuah tombol *reset*. Spesifikasi dari Arduino UNO R3 dapat dilihat pada tabel 2.1 dan bentuk fisik dari Arduino UNO R3 dapat dilihat pada gambar 2.10.



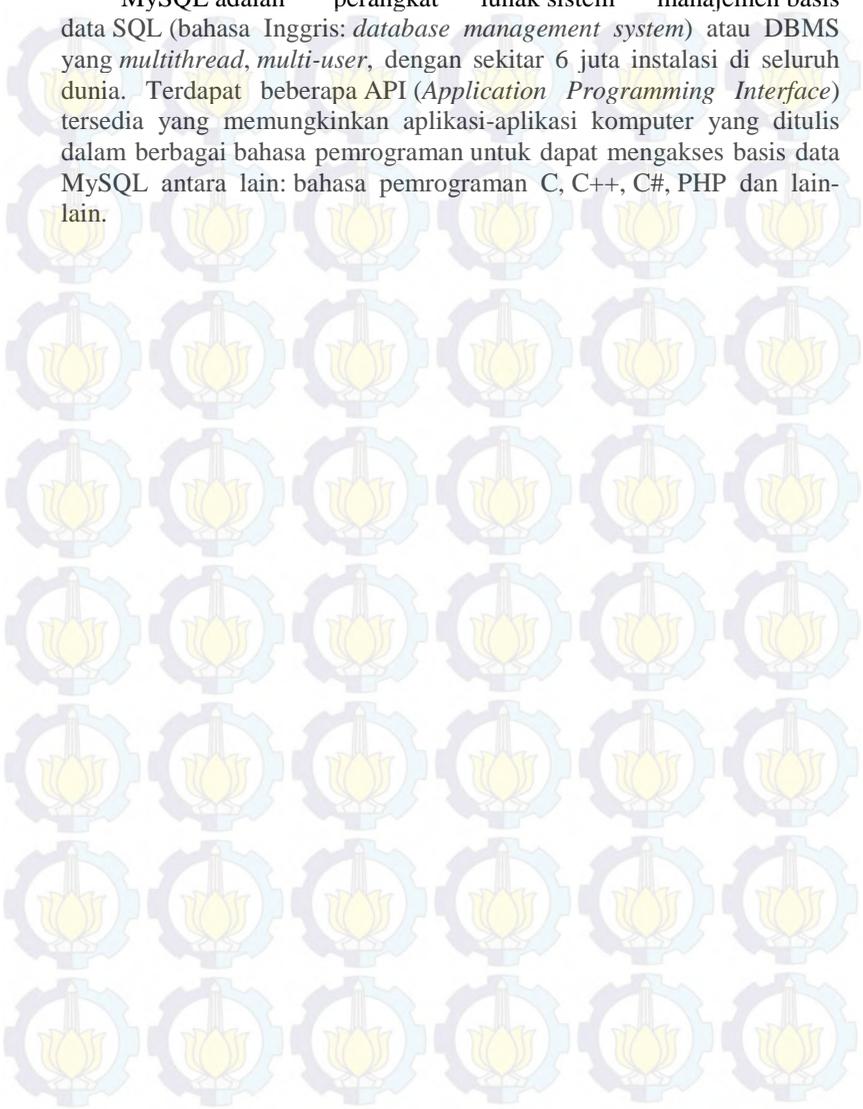
Gambar 2.10 Arduino UNO R3

Tabel 2.1 Spesifikasi Arduino UNO R3

Mikrokontroler	ATmega328
Tegangan pengoperasian	5 V
Tegangan <i>input</i> yang disarankan	7-12 V
Batas tegangan <i>input</i>	6-20 V
Jumlah pin I/O digital	14 (6 diantaranya menyediakan keluaran PWM)
Jumlah pin <i>input</i> analog	6
Arus DC tiap pin I/O	20 mA
Arus DC untuk pin 3.3 V	50 mA
Memori flash	32 KB (ATmega328), sekitar 0.5 KB digunakan oleh bootloader
SRAM	2 KB
EEPROM	1 KB
Clock speed	16 MHz

## 2.11 MySQL

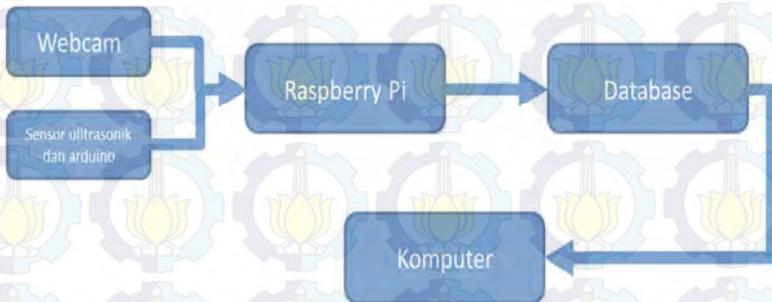
MySQL adalah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: *database management system*) atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. Terdapat beberapa API (*Application Programming Interface*) tersedia yang memungkinkan aplikasi-aplikasi komputer yang ditulis dalam berbagai bahasa pemrograman untuk dapat mengakses basis data MySQL antara lain: bahasa pemrograman C, C++, C#, PHP dan lain-lain.



## BAB III

### PERANCANGAN DAN PEMBUATAN SISTEM

Pada bab ini akan dibahas mengenai perancangan sistem kamera pengawas yang meliputi *client* dan *server*. Sistem *client* berfungsi sebagai perangkat pengolah gambar kemudian menyimpan gambar hasil olahan ke *database* yang telah tersambung dengan *server* melalui Wi-Fi. *Server* berfungsi untuk menerima dan menampilkan gambar yang telah diolah kepada *user* pengawas. Secara garis besar, keseluruhan sistem dapat diilustrasikan pada gambar 3.1.



**Gambar 3.1** Blok Diagram Sistem

Pada bagian *client* terdapat Raspberry Pi, modul kamera Raspi, *buzzer*, Arduino dan relay. Arduino berfungsi untuk membuat simulasi rangkaian lampu merah. Relay tersambung dengan Arduino yang berfungsi untuk mengaktifkan program setelah sensor jarak mendeteksi kendaraan yang melanggar lampu merah. Pada bagian *server* hanya terdapat sebuah komputer yang memuat *database* yang menggunakan Wi-Fi lokal untuk berkomunikasi.

Saat lampu berwarna merah maka program akan aktif. Terdapat 2 sensor jarak pada lampu merah, sensor 1 terletak sebelum garis batas berhenti kendaraan yang tersambung dengan *buzzer*. Sensor ke-2 terletak setelah garis batas yang akan mengaktifkan kamera saat ada kendaraan yang melanggar lampu merah.

#### 3.1 Perancangan Software

Perancangan *software* yang dilakukan pada Tugas Akhir ini yaitu perancangan *database* dan perancangan *background subtraction*

### 3.1.1 Perancangan Database

Tugas utama bagian *client* adalah untuk menerima gambar hasil *image processing* dari *server* dan menampilkannya kepada *user* pengawas.

Pertama-tama *client* melakukan koneksi dengan *server* supaya data gambar di *database server* dapat diakses *client*. Untuk menyambung keduanya, digunakan perintah program MySQL sebagai berikut :

```
GRANT ALL PRIVILEGES ON *.* TO 'USERNAME'@'IP'  
IDENTIFIED BY 'PASSWORD';
```

```
FLUSH PRIVILEGES;
```

Setelah *database* antara komputer dengan Raspberry Pi tersambung, setiap gambar yang telah diambil oleh Raspberry dapat di akses komputer operator. Untuk melihat database di *browser* digunakan phpMyAdmin. Untuk mengakses phpMyAdmin, maka menggunakan IP local dari *server*. Pada tahap selanjutnya setiap gambar yang diambil akan ditampilkan di internet menggunakan PHP. Untuk menghubungkan dengan *user* pada MySQL, menggunakan *script* berikut:

```
$conn = mysql_connect('host','user','password');
```

Sebelum memasukkan data gambar, *database* akan memeriksa apakah dalam *database* sudah dimasukkan gambar. Hal ini untuk menghindari penumpukkan data gambar yang sama. *Script* yang digunakan yaitu sebagai berikut:

```
$sql = 'SELECT nama FROM images WHERE nama =  
"'.$name_file[0]."';
```

```
$result = mysql_query($sql);
```

```
$row = mysql_fetch_array($result, MYSQL_ASSOC);
```

Apabila belum ada gambar yang dimasukkan ke dalam database, maka gambar yang baru saja diambil akan dimasukkan. Scriptnya adalah sebagai berikut:

```
if(!$row){  
    mysql_select_db('db');  
    $sql_insert = 'INSERT INTO images (nama)  
VALUES ("'.$name_file[0]."' );  
    $query = mysql_query($sql_insert);  
}
```

```
echo ' <br>
```

### 3.1.2 Background Subtraction

Metode untuk segmentasi *moving region* secara *real time* adalah *background subtraction*. *Background subtraction* merupakan batas kesalahan antara *background* statis (tidak ada pergerakan) dengan gambar itu sendiri. *Background image* bisa diartikan sebagai *scene* atau adegan tanpa adanya objek yang bergerak. *Background image* harus selalu diperbaharui sehingga dapat beradaptasi dengan perubahan kondisi seperti perubahan pencahayaan. Perubahan pencahayaan akan mempengaruhi proses pendeteksian objek. Pada *background subtraction*, gambar saat ini dibandingkan dengan gambar referensi untuk mendeteksi adanya perubahan *pixel*. Gambar referensi sebaiknya disesuaikan dengan kondisi pencahayaan dari suatu kejadian. Karena kesederhanaan dan karena lokasi kamera tetap dalam banyak konteks, *background subtraction* mungkin adalah gambar paling mendasar pengolahan operasi untuk aplikasi keamanan video. Untuk melakukan *background subtraction*, pertama harus mempelajari model latar belakang. Setelah dipelajari, model latar belakang ini dibandingkan terhadap gambar saat ini dan kemudian bagian latar belakang akan dikurangi. Objek yang tersisa setelah pengurangan bisa disebut sebagai objek *foreground*. Proses dari *background subtraction* terdiri dari akses input, proses *grayscale*, dan frame *differencing*.

Pada proses akses input, digunakan kamera dengan pengaturan resolusi sebesar 320 x 240. Perintah yang digunakan untuk memilih kamera yaitu `cvCaptureFromCAM( 0 )`. *Delay* diberikan untuk memberikan waktu kepada kamera agar menyesuaikan kondisi pencahayaan disekitarnya. Sedangkan `cvQuery` berfungsi untuk menampilkan hasil capture dari kamera kedalam matriks gambar yang bernama *frame*.

Proses kedua yaitu proses *grayscale* yang termasuk dalam *preprocessing*. Langkah pertama dalam *preprocessing* adalah mengubah gambar dari RGB kedalam bentuk *grayscale*. Proses ini dibutuhkan karena gambar RGB mempunyai nilai 24 bit, yang masing-masing warna bernilai delapan bit. Jika tidak dikonversi dahulu kedalam bidang gambar *gray* yang hanya bernilai delapan bit saja, maka proses

perhitungan piksel akan sangat sulit, karena sistem akan melakukan perhitungan matriks dengan dimensi yang sangat besar.

```
cv::cvtColor(frame1,grayImage1,COLOR_BGR2GRAY);
```

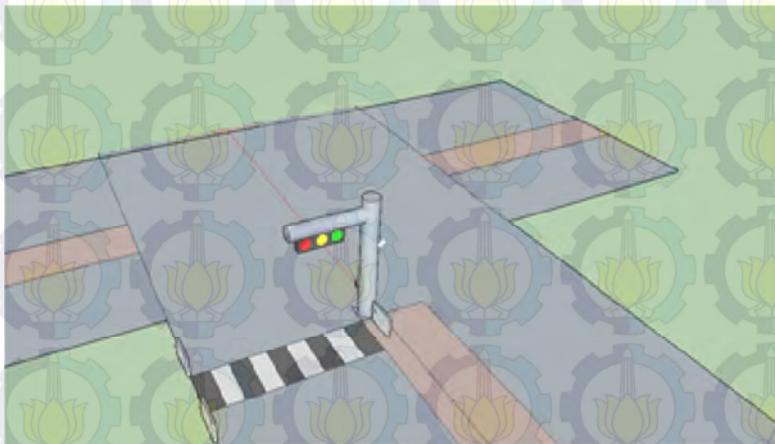
Dalam tugas akhir ini, digunakan fungsi yang sudah ada dalam opencv untuk mgubah gambar dari RGB ke *grayscale*. Library di atas adalah prosedur yang digunakan untuk mengubah gambar dari RGB ke gray.

Proses terakhir yaitu *frame differencing*. Tahap ini mendeteksi obyek dengan cara membandingkan antara *background* dan obyek. Obyek akan terdeteksi jika ada perbedaan antara *frame 1* dan *frame* sebelumnya. Berikut adalah kode untuk mendeteksi obyek.

```
cv::absdiff(grayImage1,grayImage2,differenceImage);
```

### 3.2 Perancangan *Hardware*

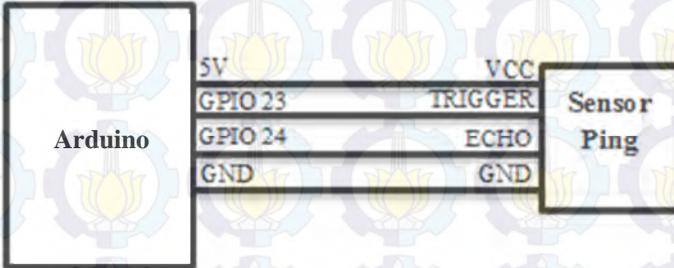
*Hardware* sistem ini adalah miniatur persimpangan jalan. Dalam miniatur tersebut dipasang sistem kamera pengawas persimpangan jalan. Sistem dari kamera pengawas terdiri dari perangkat komputer *mini* Raspberry Pi, kamera, serta sensor ping. Gambar 3.2 menunjukkan sketsa miniatur dari persimpangan jalan.



**Gambar 3.2** Sketsa Miniatur Persimpangan Jalan

### 3.2.1 Sensor Ping

Sensor ping di sistem ini berfungsi untuk mendeteksi adanya kendaraan yang melewati *stopping line* pada persimpangan jalan. Ketika pengemudi melewati sensor pertama, maka akan membunyikan peringatan agar pengemudi mundur sampai berada di belakang *stopping line*. Gambar 3.3 menunjukkan konfigurasi Arduino dengan sensor Ping



**Gambar 3.3** Konfigurasi Arduino dengan Sensor Ping

Sensor ping diprogram untuk memberikan perintah ketika mendeteksi benda dengan jarak kurang dari 20 cm. untuk sensor pertama hanya membunyikan peringatan. Sedangkan sensor kedua untuk mengaktifkan program *background subtraction* itu sendiri. Penghitungan jarak dilakukan dengan menggunakan persamaan (2)

$$jarak = \frac{waktu_{pantul} \times 344 \text{ m/s}}{2} \dots\dots\dots (2)$$



-----Halaman ini sengaja dikosongkan-----

## BAB IV HASIL SIMULASI DAN IMPLEMENTASI

Pada bab ini akan dibahas hasil pengujian perangkat keras dan perangkat lunak yang telah dirancang sebelumnya. Perangkat keras meliputi sensor jarak sedangkan perangkat lunak meliputi kamera.

### 4.1 Perangkat Keras

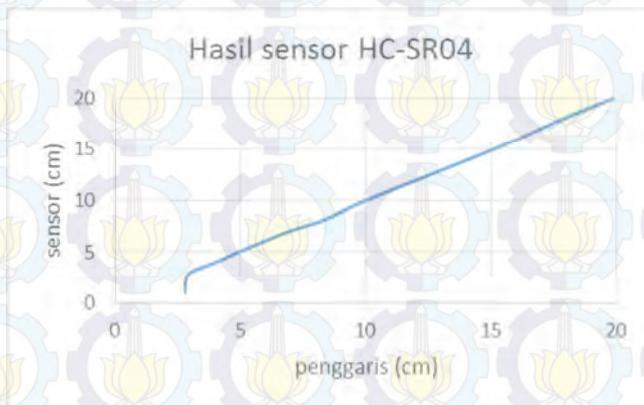
Pengujian pada sensor jarak yaitu mengukur perbandingan antara jarak sebenarnya (cm) dengan hasil (cm) dari sensor jarak. Tabel 4.1 menunjukkan hasil pengujian dari sensor jarak 1.

**Tabel 4.1** Hasil Pengujian Sensor Jarak 1

Sensor 1		
Penggaris (cm)	Sensor (cm)	Error(%)
1	2.8	180
2	2.8	40
3	3	0
4	4	0
5	4.9	2
6	5.9	1.7
7	6.9	1.43
8	8.2	2.5
9	9,1	1.11
10	10	0
15	15.1	0.67
18	17.9	0.56
20	19.9	0.5

**Tabel 4.2** Hasil Pengukuran Sensor Jarak 2

Sensor 2		
Penggaris (cm)	Sensor (cm)	Error(%)
1	2.8	180
2	2.8	40
3	3.1	3.33
4	4.1	2.50
5	5.2	4.00
6	6	0
7	7.1	1.43
8	8.2	2.50
9	9.3	3.33
10	10.2	2.
15	14.9	0.67
18	18	0.00
20	19.9	0.50



**Gambar 4.1** Grafik Perbandingan Antara Hasil Pengukuran Sensor HC-SR04 Dengan Penggaris

Pada tabel 4.1 dan tabel 4.2 di atas bisa dilihat bahwa masih ada *error* antara hasil pengukuran dari program dan hasil pengukuran menggunakan penggaris. *Error* atau selisihnya antara 1 sampai dengan 3 cm. Jika dalam persentase adalah 0.5% sampai dengan 2% untuk sensor 1, dan 0.5% sampai dengan 3.33% untuk sensor 2. Namun pada jarak pengukuran dibawah 3 cm sensor tidak dapat mengukur jarak benda tersebut. Hal itu disebabkan karena sensor hanya bisa mengukur jarak benda antara 3 – 400 cm saja. Gambar 4.1 menunjukkan Grafik Perbandingan Antara Hasil Pengukuran Sensor HC-SR04 Dengan Penggaris Penghitungan error dilakukan menggunakan persamaan (3).

$$error = \left( 1 - \left( \frac{pengukuran_{sensor}}{pengukuran_{penggaris}} \right) \right) \times 100\% \dots\dots\dots (3)$$

## 4.2 Perangkat Lunak

Pengujian perangkat lunak menghasilkan data sebagai berikut:

### 4.2.1 Pengolahan Citra

Pada pengujian *software* pengolahan citra, dilakukan dengan menggunakan objek yang digerakkan secara manual. Karena saat pengujian dalam keadaan gelap, maka ditambahkan pencahayaan agar hasil pengujian terlihat lebih jelas. Pengujian dilakukan sekali dengan output citra yang berbeda. Yaitu *thresholding* dan *frame deferancing*.



**Gambar 4.2** Gambar Citra Saat Proses *Thresholding*

Yang pertama adalah *thresholding*. Pada tahap ini citra yang berderajat keabuan akan diubah menjadi citra hitam putih. Sehingga dapat diketahui daerah mana yang termasuk obyek dan *background*. Citra hasil *thresholding* akan digunakan lebih lanjut untuk mendeteksi obyek. Seperti pada gambar 4.2, bisa dilihat bahwa obyek yang terdeteksi mempunyai tepian putih. Pada gambar 4.2 terjadi pergerakan kamera, sehingga hasil *threshold* kurang sempurna terekstraksi.

Kemudain proses *frame differencing*. Pada tahap ini citra akan dibandingkan antara *frame* 1 dan *frame* sbelumnya. Pada gambar 4.2 terlihat ada perbedaan antara *frame* 1 dan *frane* sebelumnya. Tahap ini untuk mendeteksi adanya perbedaan obyek yang ada dalam *frame*. Setelah perbedaan terdetreksi maka bisa dipastikan ada obyek yang bergerak di dalam *frame*.

Namun pada gambar 4.3 terjadi pergerakan kamera, sehingga seolah olah semua yang berada di dalam *frame* tersebut bergerak. Sedangkan pada gambar 4.4 menampilkan *frame* utama yang mendeteksi adanya pergerakan.



**Gambar 4.3** Hasil *Frame Differencing*

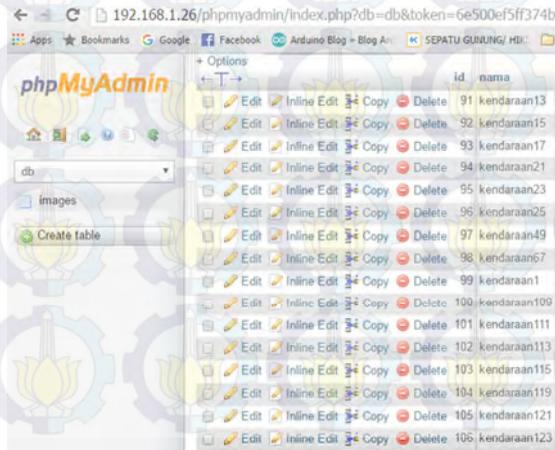
Gambar 4.4 merupakan hasil dari proses sebelumnya. Setelah dilakukan *frame differencing* maka akan diketahui jika ada obyek bergerak. Jika terdapat obyek yang bergerak, maka akan muncul tulisan pelanggaran pada *frame* tersebut.



**Gambar 4.4** Hasil Pemantauan

#### 4.2.2 Database

Pada pengujian terhadap database yang telah dibuat dilakukan dengan cara mengisi direktori yang telah ditentukan dengan gambar hasil dari pengolahan citra secara *real time*. Selama proses pengisian direktori dengan gambar, *database* dapat dipantau pada aplikasi phpMyAdmin. Tampilan phpMyAdmin adalah seperti pada gambar 4.5 dibawah ini. Gambar yang telah diambil akan langsung di-*upload* pada database ini.



**Gambar 4.5** Tampilan data gambar pada phpMyAdmin



**Gambar 4.6** Tampilan Gambar pada *Website*

Gambar 4.6 menunjukkan tampilan dari *database* yang diakses dari *browser* internet pada komputer. Untuk mengakses *database* dapat dilakukan dengan cara memasukkan *IP address* perangkat Raspberry Pi yang digunakan ke dalam kolom alamat pada *browser* yang digunakan. Dalam hal ini *IP address* dari perangkat tersebut adalah 192.168.1.26.

## **BAB V PENUTUP**

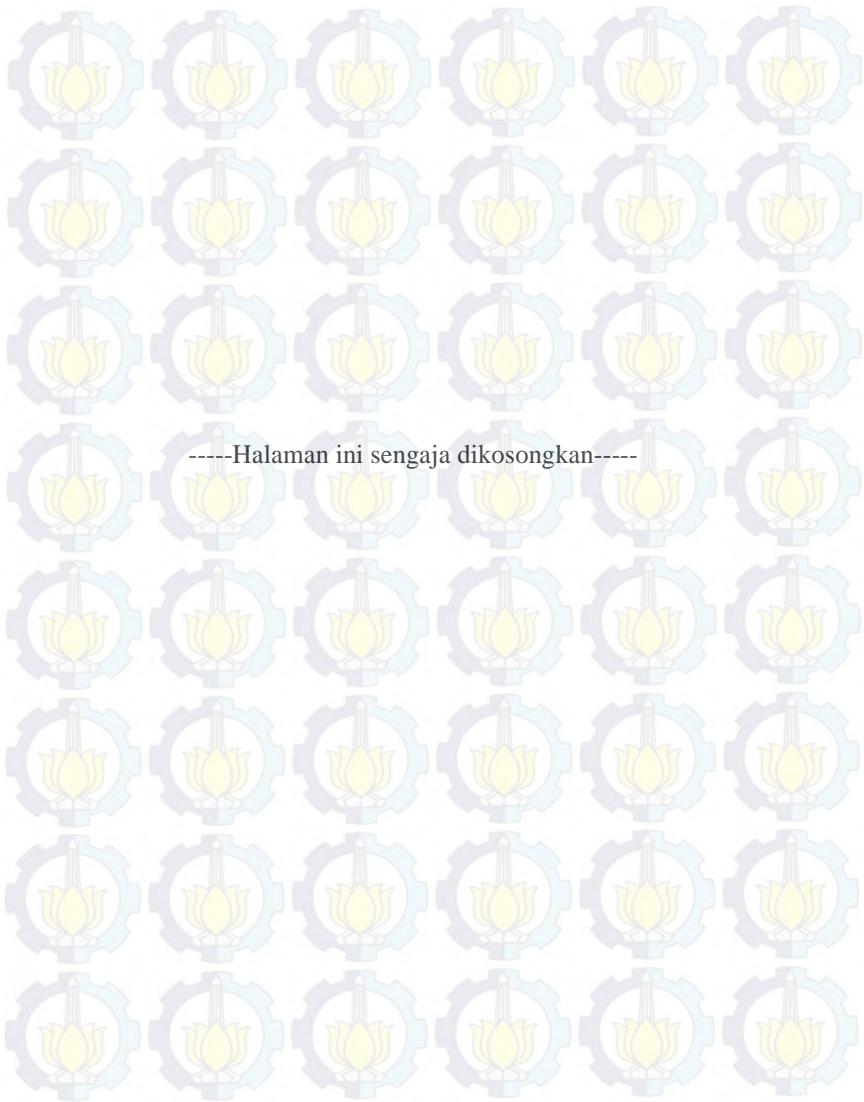
### **5.1 Kesimpulan**

Dari pengujian dan hasil analisis tugas akhir ini dapat diambil kesimpulan untuk sistem yang dibangun

1. Sensor ultrasonik yang digunakan hanya bisa mengukur jarak antara 3 – 400 cm. Apabila pengukuran dibawah 3 cm maka hasil pengukuran akan menghasilkan *error* sangat besar hingga 180% yang menunjukkan pengukuran kurang akurat.
2. Kamera yang digunakan hanya beresolusi VGA. Kamera dengan resolusi 5MP keatas menghasilkan gambar yang lebih baik.
3. Kecepatan pemrosesan di perangkat Raspberry Pi B+ yaitu 700 MHz adalah cukup cepat karena mampu menangkap 2 gambar setiap detiknya.

### **5.2 Saran**

1. Karena metode yang digunakan sangat terpengaruh cahaya, maka diharapkan digunakan metode yang lebih stabil dari pengaruh cahaya.
2. Untuk kamera yang digunakan diharapkan menggunakan kamera dengan resolusi yang lebih bagus agar gambar yang dihasilkan menjadi lebih jelas



## DAFTAR PUSTAKA

- [1] Bradski, Gary, dan Kaehler. Adrian, "*Learning OpenCV*", O'Reilly Media, Sebastopol, 2008.
- [2] Kadir. Abdul, "Belajar Database Menggunakan MySQL", ANDI OFFSET, Yogyakarta, 2008.
- [3] Kroenke. David M., Dolan. Kathleen A., "Database Processing: Fundamentals, Design, Implementation: Third Edition", Science Research Associates Inc., Singapore, 1990.
- [4] Lindley. Craig A., "Practical Image Processing in C: acquisition, Manipulation, and Storage", John Wiley & Sons Inc., Canada, 1991.
- [5] Parker. James R., "Algorithms for Image Processing and Computer Vision", John Wiley & Sons Inc., Canada, 1997.
- [6] Putra. Darma, "Pengolahan Citra Digital", ANDI OFFSET, Yogyakarta, 2010.
- [7] Sid-Ahmed. Maher A., "Image Processing: Theory, Algorithms, and Architectures", McGraw-Hill Book Co., Singapore, 1995.
- [8] Abdelkader. Mohamed F., Chellappa. Rama, Qinfen Zheng, "Integrated Motion Detection and Tracking for Visual Surveillance", the Fourth IEEE International Conference on Computer Vision Systems, 2006.
- [9] Brijesh. Kodinaria, Tiwari. Vineeta, "*Intelligent Surveillance System For Motion Detection Using Raspberry Pi*", International Journal on Recent and Innovation Trends in Computing and Communication Volume: 3 Issue: 5, India, Mei, 2015.
- [10] Irianto. Kurniawan Dwi, Ariyanto. Gunawan, Ary P. Dedi, "*Motion Detection Using OpenCV With Background Subtraction and Frame Differencing Technique*". Symposium Nasional RAPI VIII, 2009.
- [11] Senthilkumar. G, Gopalkhrisnan. K, Kumar. V.Sathish, "Embedded Image Capturing System Using Raspberry Pi system", International Journal of Emerging Trends and Technolog in Computer Science, India, March-April, 2014.
- [12] Shilpashree. K.S., H. Loksha, Shivkumar. Hadimani, "Implementation of Image Processing on Raspberry Pi", International Journal of Advance Research in Computer and Comunnication Engineering, India, Mei, 2015.
- [13] Singh. Alam Inder, Kaur. Gagandeep, "Motion Detection to

Compensate Camera Flicker Using an Algorithm”, International Journal Of Computational Engineering Research, Juni, 2012.

- [14] Solichin. Achmad, Harjoko. Agus, “Metode Background Subtraction untuk Deteksi Obyek Pejalan Kaki pada Lingkungan Statis”, Seminar Nasionalaplikasi Teknologi Informasi (SNATI), Yogyakarta, Juni, 2015.
- [15] Suryatali. Abhijeet, Dharmadhikari. V.B., “*Computer Vision Based Vehicle Detection for Toll Collection System Using Embedded Linux*”, International Conference on Circuit, Power and Computing Technologies [ICCPCT], India, 2015.
- [16] Zhang. Lijing, Liang. Yingli, “*Motion Human Detection Based On Background Subtraction*”, Second International Workshop on Education Technology and Computer Science, China, 2010.

## LAMPIRAN A

### Program Background Subtraction

```
#include <cv.h>
#include <highgui.h>
#include <time.h>
#include <sstream>
#include <wiringPi.h>

using namespace std;
using namespace cv;

const static int SENSITIVITY_VALUE = 40;
const static int BLUR_SIZE = 10;
bool debugMode;
bool trackingEnabled;

string intToString(int number){
    stringstream ss;
    ss << number;
    return ss.str();
}

bool detectMotion(Mat thresholdImage, Mat &cameraFeed){
    bool motionDetected = false;
    Mat temp;
    thresholdImage.copyTo(temp);
    vector< vector<Point> > contours;
    vector<Vec4i> hierarchy;
    findContours(temp,contours,hierarchy,CV_RETR_CCOMP,C
V_CHAIN_APPROX_SIMPLE );// retrieves all contours
    findContours(temp,contours,hierarchy,CV_RETR_EXTERNAL,CV_CHAIN_APPROX_SIMPLE );
    if(contours.size(>0)motionDetected=true;
    else motionDetected = false;

    return motionDetected;
```

```

}
int main(){
    wiringPiSetup();
    pinMode(4,INPUT);
    pinMode(5,INPUT);
    PinMode(6,INPUT);
    bool motionDetected = false;
    //set debug mode and trackingenabled initially to false
    //these can be toggled using 'd' and 't'
    debugMode = false;
    trackingEnabled = true;
    //set up the matrices that we will need
    //the two frames we will be comparing
    Mat frame1,frame2;
    //their grayscale images (needed for absdiff() function)
    Mat grayImage1,grayImage2;
    //resulting difference image
    Mat differenceImage;
    //thresholded difference image (for use in findContours()
function)
    Mat thresholdImage;
    //video capture object.
    VideoCapture capture;
    capture.open(0);
    VideoWriter oVideoWriter;//create videoWriter object, not
initialized yet
    double dWidth =
capture.get(CV_CAP_PROP_FRAME_WIDTH); //get the width of
frames of the video
    double dHeight =
capture.get(CV_CAP_PROP_FRAME_HEIGHT); //get the height of
frames of the video
    //set framesize for use with videoWriter
    Size frameSize(static_cast<int>(dWidth),
static_cast<int>(dHeight));
    if(!capture.isOpened()){
        cout<<"ERROR ACQUIRING VIDEO FEED\n";
        getch();
    }
}

```

```

        return -1;
    }
    while(1){
        capture.read(frame1);

        cv::cvtColor(frame1,grayImage1,COLOR_BGR2GRAY);
        capture.read(frame2);

        cv::cvtColor(frame2,grayImage2,COLOR_BGR2GRAY);
        cv::absdiff(grayImage1,grayImage2,differenceImage);

        cv::threshold(differenceImage,thresholdImage,SENSITIVITY_
        VALUE,255,THRESH_BINARY);
        if(debugMode==true){
            //show the difference image and threshold
            image cv::imshow("Difference
            Image",differenceImage);
            cv::imshow("Threshold
            Image",
            thresholdImage);
        }else{
            //if not in debug mode, destroy the windows
            so we don't see them anymore
            cv::destroyWindow("Difference Image");
            cv::destroyWindow("Threshold Image");
        }
        //blur the image to get rid of the noise. This will
        output an intensity image
        cv::blur(thresholdImage,thresholdImage,cv::Size(BLUR_SIZE,
        BLUR_SIZE));
        //threshold again to obtain binary image from blur
        output
        cv::threshold(thresholdImage,thresholdImage,SENSITIVITY_
        VALUE,255,THRESH_BINARY);
        if(debugMode==true){

```

```

    imshow("Final
Image",thresholdImage);
    }
    else {
        cv::destroyWindow("Final
Image");
    }
    if(trackingEnabled){
        motionDetected
detectMotion(thresholdImage,frame1);
    }else{
        motionDetected = false;
    }
    unsigned int ct=ct+1;
    if(digitalRead(4)==HIGH,digitalRead(6),motionDetected)
    {
        putText(frame1,"PELANGGARAN",cv::Point(0,420),2,2,cv::S
calar(0,255,0));
        char file[100];
        sprintf(file,"kendaraan%d.jpg",ct);
        imwrite("//var/www/file", frame1);
        ct++;
    }
    imshow("lulus pokoke",frame1);
    switch(waitKey(10)){

```

```
case 27: //'esc' key has been pressed, exit program.
    return 0;
case 116: //'t' has been pressed. this will toggle
tracking (disabled for security cam)
    /*trackingEnabled = !trackingEnabled;
    if(trackingEnabled == false)
    cout<<"Tracking disabled."<<endl;
    else cout<<"Tracking enabled."<<endl;*/
    break;
case 100: //'d' has been pressed. this will debug mode
    debugMode = !debugMode;
    if(debugMode == false) cout<<"Debug mode
disabled."<<endl;
    else cout<<"Debug mode enabled."<<endl;
    break;
}
}
return 0;
}
```



## LAMPIRAN B

Program php:

```
<?php
$conn = mysql_connect('localhost','root','123');
$dir = "image/";

$images = glob($dir."*.{jpg,png}",GLOB_BRACE);

foreach($images as $image){
    $name = basename($image);
    $name_file = explode(".", $name);

    mysql_select_db('db');
    // check di database gambar ada atau belum
    $sql = 'SELECT nama FROM images WHERE nama =
    ".$name_file[0]."';
    $result = mysql_query($sql);
    $row = mysql_fetch_array($result, MYSQL_ASSOC);

    // jika tidak ada, masukkan nama gambar ke database
    if(!$row){
        mysql_select_db('db');
        $sql_insert = 'INSERT INTO images (nama)
        VALUES ("'.$name_file[0].'")';
        $query = mysql_query($sql_insert);
    }
    echo ' <br>
    ';
}

mysql_close($conn);
?>
```

## DAFTAR RIWAYAT HIDUP



Nama : Daniel Nugraha  
TTL : Ambon, 20 Desember 1995  
Jenis Kelamin : Laki-Laki  
Agama : Kristen  
Alamat : Jl. Mulyosari Utara IV/25,  
Mulyorejo, Surabaya  
Telp/HP : 083856475902  
E-mail : xz2545.dn@gmail.com

### RIWAYAT PENDIDIKAN

1. 2001 – 2004 : SDK YBPK Surabaya
2. 2004 – 2007 : SDN Kalisari 1 Surabaya
3. 2007 – 2010 : SMP Negeri 19 Surabaya
4. 2010 – 2013 : SMA Negeri 3 Surabaya
5. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di PT. PLN (Persero) APD Jatim Surabaya

### PENGALAMAN ORGANISASI

1. Staff Departemen Hubungan Luar Periode 2014/2015  
HIMAD3TEKTRO, FTI - ITS



-----Halaman ini sengaja dikosongkan-----

## DAFTAR RIWAYAT HIDUP



Nama : Simeon Dinar Kurniawan  
TTL : Lamongan, 21 April 1995  
Jenis Kelamin : Laki-laki  
Agama : Kristen  
Alamat : Jl. Asem Payung I No. 34,  
Surabaya  
Telp/HP : 085655846309  
E-mail : simeondkurniawan@gmail  
.com

### RIWAYAT PENDIDIKAN

1. 2001 – 2007 : SDN 1 Ngasemlemahbang
2. 2007 – 2010 : SMPN 3 Ngimbang
3. 2010 – 2013 : SMAN 1 Bluluk
4. 2013 – 2016 : D3 Teknik Elektro, Program Studi Teknik Elektro Komputer Kontrol - FTI Institut Teknologi Sepuluh Nopember (ITS)

### PENGALAMAN KERJA

1. Kerja Praktek di PT.CNC Controller Indonesia Bekasi, Jawa Barat

### PENGALAMAN ORGANISASI

1. Kepala Staff Urusan Khusus UKM MENWA 802-ITS Periode 2014/2015
2. Dewan Pertimbangan UKM MENWA 802-ITS Periode 2015/2016

