

**DETEKSI TITIK API TERPUSAT MENGGUNAKAN KAMERA  
DENGAN NOTIFIKASI BERBASIS SMS GATEWAY PADA  
RASPBERRY PI**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:

Syahrul Yoga Pradana

NIM: 145150301111013



**PROGRAM STUDI TEKNIK KOMPUTER  
JURUSAN TEKNIK INFORMATIKA  
FAKULTAS ILMU KOMPUTER  
UNIVERSITAS BRAWIJAYA  
MALANG  
2018**

## LEMBAR PENGESAHAN

Deteksi Titik Api Terpusat Menggunakan Kamera Dengan Notifikasi Berbasis SMS Gateway Pada Raspberry Pi

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan memperoleh Sarjana Teknik

Disusun Oleh :  
Syahrul Yoga Pradana  
NIM: 145150301111013

Skripsi ini telah diuji dan dinyatakan lulus pada  
02 Agustus 2018  
Telah diperiksa dan disetujui oleh :

Dosen Pembimbing I

Dosen Pembimbing II



Dr. Eng. Fitri Utamingrum, S.T, M.T  
NIP: 19820710 200812 2 001



Wijaya Kurniawan, S.T, M.T.  
NIP: 19820125 201504 1 002

Mengetahui

Ketua Jurusan Teknik Informatika



Tri Astoto Kurniawan, S.T, M.T, Ph.D  
NIP: 19710518 200312 1 001

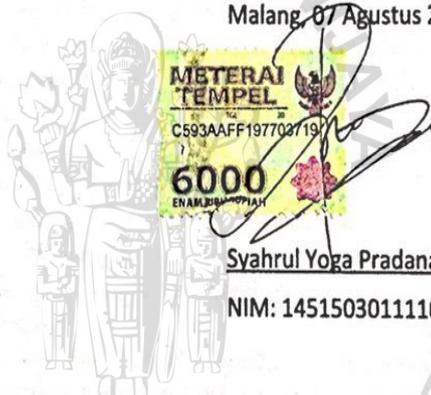


## PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 07 Agustus 2018



Syahrul Yoga Pradana

NIM: 145150301111013

## KATA PENGANTAR

Puja dan puji syukur kami panjatkan kehadirat Allah SWT yang mana atas limpahan rahmat, taufik serta hidayah-Nya, skripsi ini dapat terselesaikan. Skripsi yang berjudul **“Deteksi Titik Api Terpusat Menggunakan Kamera Dengan Notifikasi Berbasis SMS Gateway Pada Raspberry Pi”** ini disusun untuk memenuhi sebagian persyaratan memperoleh gelar Sarjana Komputer.

Penulis menyadari bahwa penyusunan skripsi ini terlepas dari bantuan berbagai pihak. Oleh karena itu penulis mengucapkan terima kasih kepada pihak-pihak yang telah bersedia untuk memberikan bantuan demi kelancaran penyusunan skripsi ini diantaranya:

1. Terimakasih kepada Allah SWT yang telah memberikan nikmat iman dan nikmat islam yang begitu indah, karena dengan rahmatnya saya bisa melakukan penulisan dan membuat alat dengan lancar.
2. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika Universitas Brawijaya Malang.
4. Bapak Sabriansyah Rizqika Akbar, S.t., M.Eng. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang.
5. Ibu Dr. Eng. Fitri Utamingrum, S.T, M.T dan Bapak Wijaya Kurniawan, S.T, M.T. selaku dosen Pembimbing 1 dan Pembimbing 2 skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
6. Bapak Joko Wahono dan Ibu Nuryanti sebagai orang tua yang telah memberikan semangat, do'a dan materi selama penulisan skripsi
7. Seluruh keluarga besar keluarga Bapak Waluyo dan Bapak Soekardjo yang telah memberikan semangat, do'a dan kasih sayang.
8. Teman-teman “D-GENGZ” dan kontrakan keramat yang selalu membantu dan memberika semangat serta motivasi selama proses pengerjaan ini.
9. Sabahat-sabahat “KACO BOS” yang telah memberikan semangat serta memberikan motivasi dalam penulisan ini
10. Kepada Viramuda Tantri Burhan Mubaroq selaku teman, sahabat , sekaligus mentor dalam proses pembuatan pengerjaan ini.
11. Seluruh civitas akademik Teknik Komputer Universitas Brawijaya yang telah banyak memberi bantuan dan dukungan selama penulis menempuh studi di Informatika Universitas Brawijaya dan selama penyelesaian skripsi ini.
12. Rekan-rekan Teknik Komputer 2014 yang selalu memberikan motivasi dalam menyelesaikan skripsi ini.
13. Dan buat kamu yang telah menemani saya selama ini dalam penulisan skripsi.

14. Seluruh pihak yang tidak dapat disebutkan satu persatu yang telah berperan dalam penyelesaian skripsi ini.

Pada akhirnya, penulis menyadari bahwa skripsi ini masih belum sempurna. Oleh karena itu, penulis sangat mengharapkan kritik dan saran yang membangun. Penulis berharap semoga skripsi ini dapat bermanfaat bagi pengembangan ilmu pengetahuan dan teknologi.

Malang, 07 Agustus 2018

Penulis

Syahrul Yoga Pradana

Syahrulyoga0@gmail.com



## ABSTRAK

Kebakaran merupakan peristiwa yang menimbulkan terjadinya api, dimana dapat menimbulkan kerugian yang signifikan akibat dari peristiwa kebakaran yaitu harta benda, bahkan bisa menyebabkan jatuhnya korban jiwa, peristiwa kebakaran lebih banyak terjadi dikarenakan oleh kesalahan manusia. Terjadinya titik api atau kebakaran disebabkan akibat dari 3 elemen yang menyatu yaitu Oksigen, Panas, dan Bahan Bakar yang disebut *Triangle of Fire* (Segitiga Api). Untuk memperkecil kerugian yang disebabkan oleh kebakaran maka perlu adanya sistem untuk mendeteksi titik api sedini mungkin agar tidak meluasnya titik api yang terjadi. Terdapat banyak sistem untuk mendeteksi titik api dipasar teknologi saat ini, maka dari itu dalam penelitian ini akan memberikan solusi alternatif rancangan sebuah sistem untuk mendeteksi titik api sedini mungkin dengan memanfaatkan *image processing* sederhana untuk dapat melakukan deteksi titik api yang memanfaatkan kamera yang terpasang pada setiap ruangan didalam rumah berbasis sms *gateway* sebagai notifikasi layanan pesan singkat berupa sms untuk memudahkan pengguna dalam mendeteksi titik api sedini mungkin. Dimana pada sistem ini menggunakan *Mini PC (Raspberry Pi 3 Model B)* sebagai pusat pengolahan data dan pengontrol sistem, pada *Raspberry Pi 3 Model B* diintegrasikan dengan Kamera *Logitech C525* dengan kamera sebagai *input* yang mengambil gambar titik api, sedangkan *Buzzer* dan *Modul SIM900A* sebagai *output* untuk memberikan notifikasi berupa *alarm* dan berupa layanan pesan singkat (sms). Pada tahap pertama yaitu melakukan pengambilan gambar dari *Kamera Logitech C525* yang kemudian dikonversi ke dalam bentuk *HSV color space*, tahap kedua melakukan proses *Morphological Image Filtering* untuk membedakan objek titik api dan objek lain yang berada disekitar titik api. Tahap ketiga ketika terdeteksi titik api maka melakukan proses penandaan dengan memanfaatkan *Bounding Rectangle*. Tahap keempat sistem akan menghitung jumlah *pixel* biner titik api yang telah terdeteksi. Tahap terakhir adalah mengaktifkan *Buzzer* dan *Modul SIM900A* untuk mengirim notifikasi layanan pesan singkat berupa (sms) ketika telah terdeteksi titik api. Dari Pengujian yang dilakukan akurasi dalam mendeteksi titik api berdasarkan objek yang berpotensi menghasilkan api ketika terbakar yaitu 90%, sedangkan pengujian keakurasian pembacaan antara kamera pada setiap ruangan yaitu sebesar 96,66%. Hasil pengujian kesesuaian pengujian *Buzzer* dan *Modul SIM900A* yaitu sesuai dengan perintah yang dilakukan, yang terakhir adalah hasil pengujian waktu komputasi pemrosesan citra pada setiap ruangan dihasilkan pada kamera ruang 1 dengan rata-rata waktu komputasi sebesar 27,5ms, sedangkan pada kamera ruang 2 sebesar 22,2, yang terakhir adalah waktu komputasi antara ruang 1 & 2 diperoleh rata-rata waktu komputasi sebesar 36,25ms.

**Kata Kunci:** kebakaran, triangle of fire, raspberry pi 3 model b, image processing, kamera, Kamera Logitech c522, buzzer, modul sim900a, morphological filtering.

## ABSTRACT

Fire accident is an event that produce flame, in which fire accident could make a significant loss such as property loss, even casualty, Fire accident mostly caused by human error. Fire happens because of three combined element which are oxygen, heat, and fuel, it is called triangle of fire. In order to minimize loss by the accident, there must be a system to detect fire as soon as possible so that it won't spread. There are many system to detect fire out in the market, And so this research gave an alternate solution which is a design system to detect flame as early as possible by utilizing simple Image Processing to do the detection with camera attached in every room inside a home based on gateway SMS acted as notifier alert in order to help user in detecting flame as soon as possible. This system used Mini PC (Raspberry Pi 3 Model B) integrated with Logitech C525 camera as an input for fire image, as for the Buzzer and Modul SIM900A acted as an output to alarm and send SMS. At the first stage, Logitech C525 Camera took image and then convert it to HSV color space format, second stage was processing Morphological Image Filtering to differentiate fire and other object surrounding the fire. Third stage happened when fire was detected and marked by using Bounding rectangle. at Fourth stage, system would count the fire binary pixel. Last stage was buzzer being activated and Modul SIM900A notified by sending SMS when fire happened. Result obtained for flame detection accuracy was 90%, as for flame detection inter-camera in every room was 96.66%. result obtained for accuracy done by Buzzer and Modul SIM900A was completely following order. The last was result from image processing time test in every room, as for room 1 the mean time was 27.5ms, room 2 was 22.2ms. and mean time between room 1 and room 2 was 36.25ms.

**Keywords :** *fire accident, triangle of fire, raspberry pi 3 model b, image processing, Camera, Camera Logitech c525, buzzer, modul sim900a, morphological image filtering.*

## DAFTAR ISI

LEMBAR PENGESAHAN .....	ii
PERNYATAAN ORISINALITAS .....	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT.....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xii
DAFTAR PERSAMAAN.....	xiii
DAFTAR LAMPIRAN .....	xiv
BAB I .....	1
1.1 Latar belakang .....	1
1.2 Identifikasi Masalah .....	2
1.3 Rumusan Masalah .....	3
1.4 Tujuan.....	3
1.5 Manfaat .....	3
1.6 Batasan Masalah .....	3
1.7 Sistematika Pembahasan .....	4
BAB II .....	6
2.1 Tinjauan Pustaka .....	6
2.2 Dasar Teori .....	7
2.2.1 Definisi Api .....	7
2.2.2 <i>Triangle of Fire</i> (Segitiga Api).....	7
2.2.3 <i>Image Processing</i> (Pengolahan Citra) .....	8
2.2.4 Segmentasi Citra Api .....	8
2.2.5 Jenis Citra .....	9
2.2.6 Pemodelan Warna .....	9
2.2.7 Pengambangan ( <i>Thresholding</i> ) .....	11
2.2.8 <i>Morphological Image Filtering</i> .....	12
2.2.9 <i>Library OpenCV</i> .....	13
2.2.10 Fungsi-Fungsi Dalam <i>Library OpenCV</i> .....	13

2.2.11	Raspberry PI 3 Model B.....	14
2.2.12	Buzzer Aktif .....	15
2.2.13	Module SIM900A.....	15
2.2.14	Kamera Logitech C525.....	16
BAB III .....		17
3.1	Studi Literatur.....	18
3.2	Rekayasa Kebutuhan .....	19
3.2.1	Gambaran Umum Sistem.....	19
3.2.2	Kebutuhan Sistem .....	20
3.2.3	Kebutuhan Perangkat Keras.....	20
3.2.4	Kebutuhan Perangkat Lunak.....	20
3.3	Perancangan Sistem .....	20
3.4	Implementasi Sistem .....	21
3.5	Pengujian dan Analisis.....	22
3.6	Pengambilan Kesimpulan .....	23
BAB IV.....		24
4.1	Gambaran Umum Sistem.....	24
4.1.1	Tujuan .....	24
4.1.2	Kegunaan.....	24
4.1.3	Karakteristik Pengguna .....	24
4.1.4	Lingkungan Operasi.....	24
4.2	Analisi Kebutuhan Sistem.....	25
4.2.1	Kebutuhan Fungsional .....	25
4.2.2	Kebutuhan Non-Fungsional .....	26
4.2.3	Kebutuhan Antarmuka Perangkat Lunak.....	26
4.2.4	Kebutuhan Antarmuka Perangkat Keras.....	26
4.3	Batasan Desain Sistem .....	27
BAB V.....		29
5.1	Perancangan Sistem .....	29
5.2.1	Perancangan <i>Prototype</i> deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms <i>gateway</i> pada <i>raspberry pi</i> .....	29
5.2.2	Perancangan Perangkat Keras .....	30
5.2.3	Perancangan Perangkat Lunak.....	33
5.2	Implementasi Sistem .....	44



5.2.1 Implementasi <i>Prototype</i> deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms <i>gateway</i> pada <i>raspberry pi</i> .....	44
5.2.2 Impementasi Perangkat Lunak .....	47
BAB VI.....	56
6.1 Pengujian Keakurasian dalam Mendeteksi Titik Api Berdasarkan Objek ...	56
6.1.1 Tujuan Pengujian.....	56
6.1.2. Prosedur Pengujian .....	56
6.1.3 Pelaksanaan Pengujian .....	57
6.1.4 Hasil dan Analisis Pengujian.....	58
6.2 Pengujian Keakurasi Pembacaan Antara Kamera pada setiap ruangan .....	62
6.2.1 Tujuan Pengujian.....	62
6.2.2 Prosedur Pengujian .....	62
6.2.3 Pelaksanaan Pengujian .....	63
6.2.4 Hasil dan Analisis Pengujian.....	63
6.3 Pengujian <i>Buzzer</i> dan <i>Modul SIM900A</i> .....	71
6.3.1 Tujuan Pengujian.....	71
6.3.2. Prosedur Pengujian .....	71
6.1.3 Pelaksanaan Pengujian .....	72
6.1.4 Hasil dan Analisis Pengujian.....	72
6.4 Pengujian Waktu Komputasi Pemproses Citra .....	76
6.4.1 Tujuan Pengujian.....	76
6.4.2. Prosedur Pengujian .....	76
6.4.3 Pelaksanaan Pengujian .....	76
6.4.4 Hasil dan Analisis Pengujian.....	76
BAB VII.....	79
7.1 Kesimpulan .....	79
7.2 Saran.....	80
DAFTAR PUSTAKA.....	81
LAMPIRAN .....	84



## DAFTAR GAMBAR

Gambar 2.1 Model Warna RGB.....	10
Gambar 2.1 Model Warna RGB.....	10
Gambar 2.3 <i>Raspberry Pi 3 Model B</i> .....	14
Gambar 2.4 <i>Buzzer</i> .....	15
Gambar 2.5 <i>Modul SIM900A</i> .....	15
Gambar 2.6 <i>Kamera Logitech C525</i> .....	16
Gambar 3.1 Diagram Alir.....	17
Gambar 3.2 Gambaran Sistem Secara Umum .....	21
Gambar 3.3 <i>Flowchart</i> Sistem.....	22
Gambar 5.1 (a) Desain Desain <i>Prototype</i> Ruangan.....	30
Gambar 5.1 (b) Rancangan <i>system</i> Deteksi titik api .....	30
Gambar 5.2 Skematik rangkaian sistem prangkat keras.....	31
Gambar 5.3 <i>Flowchart</i> proses utama .....	34
Gambar 5.4 <i>Flowchart</i> Deteksi Titik Api Di Ruang 1 .....	35
Gambar 5.5 <i>Flowchart</i> Deteksi Titik Api Di Ruang 2 .....	40
Gambar 5.6 Implementasi <i>prototype</i> alat pendeteksi titik api .....	45
Gambar 5.7 Implementasi Perangkat Keras <i>Kamera Logitech C525</i> .....	45
Gambar 5.8 Implementasi Perangkat Keras <i>Modul SIM900A</i> .....	46
Gambar 5.9 Implementasi Perangkat Keras <i>Buzzer</i> .....	47
Gambar 6.1 Screenshot Hasil Pengujian Komputasi Pemrosesan Citra .....	78

## DAFTAR TABEL

Tabel 2.1 Fungsi-Fungsi Dalam OpenCv .....	13
Tabel 5.1 Konfigurasi Kamera Logitech C525 dengan Raspberry Pi .....	32
Tabel 5.2 Koneksi pin Module SIM900A pada kiri dengan Raspberry Pi .....	32
Tabel 5.3 Koneksi pin Buzzer Aktif pada kiri dengan Raspberry Pi .....	32
Tabel 5.4 Menentukan nilai range <i>thresholding</i> citra api pada ruang 1 .....	36
Tabel 5.5 Menentukan nilai range <i>thresholding</i> citra api pada ruang 2.....	40
Tabel 5.6 Kode Program pada Inisialisasi Library Sistem Deteksi Titik api .....	47
Tabel 5.7 Implementasi Proses Deteksi Titik Api Di Ruang 1.....	48
Tabel 5.8 Implementasi Proses Deteksi Titik Api Di Ruang 2.....	50
Tabel 5.9 Implementasi Output sebagai Indikator .....	52
Tabel 5.10 Proses pengenalan variable global .....	54
Tabel 6.1 Data uji dan Hasil pengujian Mendeteksi Titik Api berdasarkan Objek yang telah ditentukan .....	58
Tabel 6.2 Keakurasi Pembacaan Antara Kamera pada ruangan 1.....	63
Tabel 6.3 Keakurasi Pembacaan Antara Kamera pada ruangan 2.....	66
Tabel 6.4 Keakurasi Pembacaan Antara Kamera pada ruangan 1&2 .....	68
Tabel 6.5 Pengujian Buzzer dan Modul SIM900A .....	72
Tabel 6.5 Pengujian Waktu Komputasi Pemrosesan Citra.....	76

## DAFTAR PERSAMAAN

Persamaan 2.1 Konversi RGB ke HSV.....	11
Persamaan 2.2 <i>Thresholding</i> .....	11
Persamaan 2.3 Dilasi.....	12
Persamaan 2.4 Erosi.....	12
Persamaan 2.5 <i>Opening</i> .....	12
Persamaan 2.6 <i>Closing</i> .....	12
Persamaan 6.1 Perhitungan Akurasi.....	57
Persamaan 6.2 Perhitungan Akurasi.....	63



**DAFTAR LAMPIRAN**

Lampiran Kode Program Proses Utama ..... 84



# BAB I

## PENDAHULUAN

### 1.1 Latar belakang

Kebakaran merupakan peristiwa yang menimbulkan terjadinya api, dimana dapat menimbulkan kerugian yang signifikan akibat dari peristiwa kebakaran yaitu hilangnya harta benda dan bahkan bisa menyebabkan jatuhnya korban jiwa, peristiwa kebakaran lebih banyak terjadi dikarenakan oleh kesalahan dari manusia (Setiani, 2015). Dalam hal ini, kebakaran dapat dibedakan menjadi 3 kelas jenis kebakaran berdasarkan penyebabnya, yaitu pada Kelas A disebabkan oleh barang yang mudah terbakar seperti kayu, kertas, plastik, karet, dll. Pada Kelas B disebabkan oleh cairan yang dapat mudah terbakar seperti bensin, oli, solar, spirtus, dan cairan lainnya yang mudah terbakar. Kemudian terakhir adalah Kelas C dimana disebabkan oleh aliran listrik yang konslet (Sigana, 2017). Pada awalnya kebakaran terjadi ketika 3 elemen bertemu yaitu Oksigen, Panas, dan Bahan bakar. Yang nantinya akan membentuk api, 3 elemen itu yang disebut segitiga api (*Triagle of Fire*) (Saberindo, 2017). Dengan adanya teori *Triangle of Fire* apabila tidak ada salah satu unsur yang terdapat pada 3 elemen tersebut maka secara teori api tidak akan terbentuk dan tidak terjadi peristiwa kebakaran. Untuk memperkecil jatuhnya korban jiwa yang disebabkan oleh bencana kebakaran tersebut maka diperlukan sistem yang dapat mendeteksi kebakaran sebagai peringatan dini terhadap indikasi kebakaran.

Terdapat banyak alat untuk mendeteksi kebakaran yang beredar dipasar teknologi saat ini, yang prinsip kerjanya yaitu menggunakan 2 pendeteksian yaitu menggunakan asap dan suhu. Pada saat mendeteksi sebuah kenaikan suhu dan terdapat kepekatan asap, maka prinsip tersebut akan bekerja dengan sesuai. Pada penelitian sebelumnya yang dilakukan oleh Imam Nurromianto dengan judul "Rancangan *Prototipe* Pendeteksian Dini Lokasi Kebakaran Berbasis *Wireless* Dengan *Fuzzy Logic* dan Pemberitahuan Via SMS" yang penggunaannya menggunakan sensor asap dan suhu (Nurromianto, 2015). Sedangkan pada penelitian yang dilakukan oleh Rezak Andri Purnomo "Implementasi Metode *Fuzzy Sugeno* Pada *Embedded System* Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan" dengan menggunakan 3 sensor yaitu sensor suhu, sensor asap, dan sensor *flame* (Purnomo, et all., 2018). Kekurangan dari sistem yang hanya memanfaatkan sensor, sistem tersebut tidak dapat mengetahui akar penyebab dari kebakaran yang terjadi didalam ruangan. Sehingga sistem yang hanya memanfaatkan sensor tidak dapat memberikan informasi secara visual yang terjadi didalam ruangan terutama ketika terjadi peristiwa kebakaran.

Oleh karena itu, penulis ingin mengusulkan solusi alternatif yang berkaitan tentang sistem untuk mendeteksi kebakaran dalam ruangan berbasis rekayasa perangkat cerdas dengan menggunakan *Raspberry PI 3 Model B* dan *Kamera*

*Logitech C525* yang akan mengambil gambar dengan menggunakan *Image Processing*. Penggunaan *Raspberry Pi 3 Model B* berfungsi sebagai pengolahan citra digital dengan menggunakan *Library OpenCV* yang telah terinstall didalam mikroprocessor tersebut, yang nantinya *library* tersebut berfungsi untuk mengolah citra digital. *Raspberry Pi 3 Model B* juga berfungsi untuk mengontrol terhadap modul-modul output yang terintegrasi dengan *Raspberry Pi Model B*. Keunggulan yang di peroleh dengan menggunakan *Raspberry Pi 3 Model B* yang disebut *Mini PC (Personal Computer)* sebagai pusat pengolahan yaitu memudahkan pengguna, *plug-and-play*, serta harga sangat terjangkau dibandingkan dengan menggunakan *Personal Computer (PC)*. Penggunaan *Raspberry Pi 3 Model B* pada sistem ini sendiri dilakukan untuk memudahkan pengguna sebagai pengontrol. Yang kedua adalah penggunaan kamera, dengan menggunakan kamera maka kita lebih mengetahui akar penyebab dari kebakaran tersebut dengan memantau kondisi yang terjadi dimasing-masing ruangan memanfaatkan kamera, dibandingkan hanya dengan memanfaatkan sensor saja.

Pada sistem ini menggunakan *Image Processing* sebagai proses manipulasi citra secara matematis untuk meng-*enhance*, menganalisa dan mengekstraksi fitur dari sebuah citra . Data *input* diambil melalui kamera *Logitech C525* yaitu menangkap gambar ketika terjadi kebakaran kemudian akan diproses oleh *mikroprocessor*, kemudian data baca tersebut akan dideteksi dengan menggunakan *Image Processing* pada *OpenCV* sebagai *filtering* dengan menggunakan metode *Morphological Image Filtering* dengan proses *Opening (Erosion dilanjutkan proses Dilation)* untuk menentukan *outputnya* yang ditampilkan ke dalam layar *monitor*, yang mana pada sistem ini dilengkapi dengan keamanan berupa peringatan dengan pemberitahuan via layanan pesan singkat (sms) dengan memanfaatkan modul *SIM900A* yang diintegrasikan dengan mikroprocessor *Raspberry Pi 3 Model B*. Selain itu pada sistem ini juga memanfaatkan *Buzzer* yang digunakan untuk memberikan *alarm* sebagai peringatan tanda bahaya ketika terdeteksi kebakaran.

## 1.2 Identifikasi Masalah

1. Penanganan kebakaran di Indonesia khususnya ketika terjadi didalam ruangan yang nantinya akan menyebabkan kebakaran yang meluas dinilai masih kurang cepat dan terkendala dengan akses.
2. Penyebab terjadinya bencana kebakaran itu sendiri yaitu konsleting listrik dan kelalaian masyarakat, maka diperlukan pemahaman untuk masyarakat dalam mengetahui penyebab terjadinya bencana kebakaran. Agar bencana kebakaran bisa ditangani dengan baik.
3. Pemahaman masyarakat terhadap cara penanganan dan mendeteksi kebakaran sedini mungkin masih kurang.
4. Bagaimana kontribusi kemampuan sistem dalam mendeteksi titik api kebakaran dalam ruangan dengan cepat serta dalam memberikan *output* informasi dengan tepat terhadap masyarakat dan pihak terkait mengenai kebakaran.

### 1.3 Rumusan Masalah

Melihat dari identifikasi masalah diatas dapat dirumuskanlah suatu permasalahan antara lain :

1. Bagaimana implementasi sistem untuk mendeteksi titik api dalam ruangan?
2. Bagaimana mengenali *object* api dan perbedaan dengan *object* yang terdapat disekitar api dari sebuah citra digital?
3. Bagaimana tingkat akurasi proses *filtering* dalam mendeteksi titik api kebakaran dalam ruangan?
4. Berapa waktu komputasi pemrosesan pada citra ketika titik api terdeteksi?

### 1.4 Tujuan

Tujuan pada penelitian ini yaitu :

1. Merancang sebuah sistem untuk mendeteksi titik api kebakaran secara terpusat dengan menggunakan kamera dengan notifikasi berbasis sms *gateway* pada *raspberry pi* dalam ruangan dengan menggunakan pengolahan citra digital.
2. Menerapkan metode pada pengolahan citra digital untuk mendeteksi titik api kebakaran.
3. Mengukur tingkat keakurasian citra setelah melalui proses Morfologi.

### 1.5 Manfaat

1. Peneliti diharapkan menerapkan ilmu sensor transduser dan rekayasa perangkat cerdas khususnya *Pengolahan Citra Digital* dalam pembuatan sebuah sistem perancangan deteksi titik api kebakaran dalam ruangan.
2. Penelitian ini, diharapkan memberikan solusi alternatif kepada pengguna dan masyarakat untuk menanggulangi dan mendeteksi titik api secara dini khususnya kebakaran yang terjadi didalam ruangan.
3. Penelitian ini, diharapkan fakultas dapat menambah referensi ilmu terutama pada kuliah bidang rekayasa perangkat cerdas khususnya *Pengolahan Citra Digital* dan dapat diterapkan oleh mahasiswa dalam penelitian selanjutnya.

### 1.6 Batasan Masalah

Adapaun batasan masalah yang terdapat pada penelitian ini, yaitu meliputi :

1. Komunikasi antara kamera *Logitech C525* dengan *Raspberrry PI 3 MODEL B*.
2. Pembahasan difokuskan pada keakurasian data *object* gambar sebuah citra yang diambil dan kemudian dilakukan *filtering* menggunakan metode *Morphological Image Filtering*.

3. Mikroprocessor yang dipakai adalah Raspberry PI 2 MODEL B untuk *Kamera Logitech C525*.
4. Pengambilan citra menggunakan webcam dengan resolusi 320x240 *pixels*
5. Penggunaan metode *Morphologi Image Filtering* sebagai *image filtering*.
6. Pengujian dilakukan didalam ruangan dengan pencahayaan tetap.
7. Pengujian dilakukan dengan *prototype* mirip dengan suatu ruangan yang terbuat dari triplek dengan ukuran 40cmx40cm.
8. *Image* citra api yang akan dianalisa adalah citra api yang berasal dari lilin, korek gas, satu lembar kertas terbakar, tisu, kapas, dan barang yang berpotensi terbakar.
9. *SIM900A* merupakan alat untuk mengirim notifikasi layanan pesan singkat berupa (sms)
10. *Buzzer* merupakan alat untuk alarm penanda tanda bahaya.

## 1.7 Sistematika Pembahasan

### BAB I PENDAHULUAN

Pada Bab I dijelaskan latar belakang, identifikasi masalah, rumusan masalah, tujuan dan manfaat penelitian, Batasan dari ruang lingkup dari penelitian ini dan kemudian sistematika penulisan laporan dari penelitian dan jadwal dilaksanakan penelitian ini

### BAB II LANDASAN KEPUSTAKAAN

Pada Bab II menjelaskan teori yang akan melandasi adanya penelitian ini. Dan pada bab Landasan Kepustakaan dicantumkan penelitian sebelumnya yang mendasari terjadinya penelitian ini.

### BAB III METODE PENELITIAN

Pada Bab III dibahas tentang langkah-langkah kerja yang dilakukan pada proses penelitian yaitu studi literatur, kemudian dilakukan sebuah analisis kebutuhan sistem, yang akan diimplementasi serta pengujian seperti apa yang akan dilakukan.

### BAB IV REKAYASA KEBUTUHAN

Pada Bab IV akan dibahas tentang metode penelitian yang memaparkan kebutuhan dari penelitian ini. Antar lain kebutuhan itu adalah kebutuhan non-fungsional, fungsional, serta dari pengguna.

### BAB V PERANCANGAN DAN IMPLEMENTASI

Pada Bab V akan dibahas perancangan serta implementasi dari penelitian ini dengan memaparkan perancangan dari system yang dibuat, dan mengimplementasikan rancangan sistem yang sudah dirancang. Pada pengimplementasikan racangan sistem ini akan dilakukan sampai sesuai dengan rancangan dan mampu bekerja sesuai tujuan awal.

## **BAB VI PENGUJIAN DAN ANALISIS**

Pada Bab VI akan membahas hasil dari pengujian dari *system* yang telah diimplementasikan, dan proses selanjutnya yaitu setelah sistem diimplementasikan adalah menguji dan menganalisis apakah sistem sudah bekerja sesuai yang diharapkan dan tujuan atautkah masih terdapat kendala. Serta dilakukan untuk menganalisis keakurasian dan kecepatan dari sistem yang telah dirancang.

## **BAB VII PENUTUP**

Pada Bab VII adalah penutup bagi penelitian ini, yang memaparkan kesimpulan serta hasil dari penelitian yang sudah dilakukan.



## BAB II

### LANDASAN KEPUSTAKAAN

#### 2.1 Tinjauan Pustaka

Pada kajian pustaka merupakan sub bab dimana menjelaskan mengenai penelitian sebelumnya, yang berkaitan dengan penelitian yang dilakukan oleh penulis. Tujuan dari kajian pustaka yaitu sebagai pembandingan dan sebagai referensi terhadap penelitian yang dilakukan oleh penulis. Berikut merupakan kajian pustaka terhadap penelitian sebelumnya

Pertama, penelitian yang dijadikan acuan adalah penelitian yang diterbitkan pada tahun 2014 oleh Jareerat Seebamrungsat, Suphachai Praising, dan Panomkhawan Riyamongkol dengan judul "*Fire Detection in the Buildings Using Image Processing*" (Seebamrungsat, 2014). Dalam penelitian ini kajian pustaka yang akan dibahas yaitu mengenai *Image Filtering* menggunakan *Morphological Image Filtering* dalam proses pembacaan gambar titik api kebakaran dalam ruangan. Dalam penelitian ini kajian pustaka yang akan dibahas yaitu mengenai *Pengolahan Citra Digital* dalam proses *image filtering*. Dalam, penelitian sebelumnya menggunakan metode *Color Segmentation* dengan menggunakan model warna *YCbCr* untuk memisahkan warna nyala dari latar belakang. Untuk model HSV digunakan untuk mendeteksi informasi berkaitan dengan warna dan kecerahan, sedangkan untuk warna *YCbCr* untuk mendeteksi berkaitan dengan *brightness* karena dapat membedakan gambar terang dengan lebih efisien dari model warna lainnya. Serta pada *output* sistem tersebut menggunakan peringatan berupa *alarm* ketika terjadi kebakaran di dalam ruangan. (Seebamrungsat, 2014)

Kedua, penelitian yang diterbitkan pada tahun 2017 yang berjudul "Implementasi Metode *Fuzzy Sugeno* Pada *Embedded System* Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan" (Purnomo , et all., 2017). Dalam penelitian ini membahas mengenai implementasi *embedder system* untuk mendeteksi kebakaran yang terjadi dalam ruangan dengan memanfaatkan sensor MQ-2 sebagai sensor suhu untuk mendeteksi suhu panas, sensor DHT11 sebagai sensor asap untuk mendeteksi kepekatan asap, dan yang terakhir adalah sensor *flame* untuk mendeteksi adanya api. *Ouput* dari sistem ini yaitu memberikan peringatan berupa *alarm* tanda bahaya serta akan menampilkan pada LCD 16x4 yang terintegrasi dengan sistem. (Purnomo , et all., 2017)

Ketiga, penelitian yang diterbitkan 2016 yang berjudul "*An Automatic Fire Detection and Warning System Under Home Video Surveillance*"(Hasan, et all, 2016). Dalam penelitian ini mendeteksi kebakaran di dalam ruangan dengan menggunakan pengolahan citra digital dengan memanfaatkan ruang warna RGB ketika mendeteksi kebakaran. *Output* pada *system* ini yaitu memberi peringatan berupa layanan pesan singkat berupa (SMS) yang memanfaatkan *Module SIM900 GSM* serta memberikan pesan *email* terhadap *user* ketika *system* menemukan titik api yang berada pada rumah. (Hasan, et all., 2016)

Dengan spesifikasi suatu sistem seperti yang dijelaskan pada penelitian-penelitian sebelumnya yang telah dilakukan maka penelitian ini sebagai solusi alternatif untuk mendeteksi titik api ketika terjadi kebakaran dalam ruangan dengan membedakan *object* api serta *object* disekitar api dan *output* peringatan berupa alarm dan memanfaatkan layanan pesan singkat berupa (sms). Jadi untuk membedakan *object* api dengan *object* disekitar api maka dibutuhkan metode sebagai *filtering* untuk membedakan *object* api dengan *object* disekitar api agar *object* api tersebut dapat terdeteksi yaitu dengan menggunakan metode *Morphological Image Filtering*. Perbedaan pada penelitian ini dengan penelitian sebelumnya tidak terlalu banyak. Namun, penelitian ini memberikan solusi alternatif terhadap *output* yang dihasilkan, perbedaan yang pertama adalah peneliti menerapkan konsep pengolahan citra pada suatu gambar citra digital dengan menggunakan kamera *Logitech C525* dimana sistem ini menggunakan kamera sehingga dapat memantau di masing-masing ruangan yang nantinya ketika terjadi suatu peristiwa kebakaran secara otomatis gambar tersebut terekam dan akan dideteksi. Perbedaan kedua terletak pada penggunaan *output* berupa *alarm* sebagai peringatan dini dan memanfaatkan pengiriman layanan pesan singkat (via sms) terhadap kebakaran sehingga nilai *output* yang diterapkan pada *Rekayasa Perangkat Cerdas* sesuai yang diharapkan. Perbedaan ketiga terletak pada penggunaan mikroprosesor yaitu dengan menggunakan *Raspberry PI*.

## 2.2 Dasar Teori

Pada bagian dasar teori ini, menjelaskan beberapa teori dan komponen yang digunakan dalam penelitian ini. Beberapa hal seperti penjelasan Definisi Api, *Triangle Of Fire (Segitiga Api)*, *Image Processing (Pengolahan Citra)*, *Morphological Image Filtering*, Fungsi-Fungsi Dalam *Library Opencv*, *Raspberry PI 3 Model B*, *Library OpenCV*, modul *Buzzer*, dan modul Kamera *Logitech C525*.

### 2.2.1 Definisi Api

Definisi api menurut NFPA secara umum api sebagai penyebab dari sebuah kebakaran didefinisikan peristiwa oksidasi melibatkan 3 elemen yang harus terpenuhi yaitu Oksigen yang terdapat pada udara, Bahan Bakar yang cepat terbakar, serta energi panas. Maka akan terjadi peristiwa kebakaran, sehingga mempunyai dampak yang signifikan yaitu seperti dapat menimbulkan cedera bahkan lebih parahnya dapat menyebabkan jatuhnya korban jiwa, serta kerugian hilangnya harta dan benda (Darma, 2010).

### 2.2.2 *Triangle of Fire (Segitiga Api)*

*Triangle of Fire* yaitu 3 elemen yang mendukung terjadinya suatu api yang akan menimbulkan peristiwa kebakaran (Saberindo, 2017). 3 elemen menjadi unsur api tersebut terdiri dari :

1. Oksigen (*Oxygen*), merupakan unsur yang terkandung pada udara. Jika, unsur udara atau oksigen tidak ada maka proses pembentukan api tidak terjadi

2. Bahan Bakar (*Fuel*), merupakan unsur bahan bakar yang terdiri dari pada, cair, dan gas yang mudah terbakar.
3. Sumber Panas (*Heat*), merupakan pemicu terjadinya kebakaran dengan energi yang cukup ketika menyalakan campuran antara bahan bakar dengan oksigen dari udara

Proses terjadinya api yang menimbulkan peristiwa kebakaran ketika ketiga unsur tersebut saling berinteraksi atau bereaksi satu dengan lainnya. Namun, ketika salah satu unsur tidak terpenuhi maka proses terjadinya api tidak akan terjadi (Kelvin, et all. 2015).

### 2.2.3 *Image Processing* (Pengolahan Citra)

*Image Processing* (Pengolahan Citra) adalah proses manipulasi citra secara matematis untuk meng-*enhance*, menganalisa dan mengekstraksi fitur dari sebuah citra (Fitriyah, 2016). Pada pengolahan ini dilakukan untuk menemukan nilai intensitas dari sebuah sinyal citra, serta dapat dilakukan dengan menggunakan domain asalnya yakni domain spasial maupun domain frekuensinya.

Dalam pengolahannya terdapat beberapa alasan dalam melakukan *image processing* pada citra digital yaitu :

1. Bertujuan untuk mendapatkan citra yang diperkirakan dapat mendekati citra sesungguhnya. Serta untuk mendapatkan citra yang asli dari citra yang buruk karena derau.
2. Bertujuan mendapatkan citra yang mempunyai karakteristik tertentu serta cocok seara visual yang dibutuhkan dalam tahap proses ketika menganalisis suatu citra ke proses akuisisi, yang dimana citra diolah dengan mentranformasikan ke suatu representasi numerik. Dan proses representasi numeriknya akan diolah secara digital oleh komputer (Fauzi, et all., 2017)

### 2.2.4 Segmentasi Citra Api

Pada beberapa percobaan yang dilakukan untuk mengenali kriteria api dengan menggunakan pengolahan citra. Segmentasi api dilakukan dalam ruang warna RGB, RGB mempunyai kelebihan yaitu pada pencahayaannya (Raharjo, 2014).

Api mempunyai tingkat pencahayaan yang tinggi dan komponen merahnya yang lebih tinggi, dari kriteria api cukup mudah untuk menentukan kriteria piksel pada api dari kelebihan-kelebihan yang terdapat pada ruang warna RGB, namun perlu diketahui bahwasannya api dalam jarak yang dekat dengan kamera bisa berubah maka api tersebut akan semakin besar. Komponen merah, orange dan kuning digunakan dalam segmentasi ini dikarenakan warna dominan dari api. Warna tersebut lebih tinggi dari rata-rata warna komponen warna hijau dan warna biru (Pambudi, 2006).

## 2.2.5 Jenis Citra

Suatu *pixel* memiliki rentang nilai yang bermacam-macam, mulai dari nilai minimal hingga nilai maksimal. Rentang nilai yang digunakan berbeda-beda tergantung jenis warna pada suatu *pixel*. Pada umumnya rentang nilai *pixel* yaitu 0-255. Berdasarkan warna-warna penyusunnya, citra digital dapat dibagi menjadi tiga macam yaitu : (Wijaya, 2007).

### 1. Citra biner

Citra biner yaitu citra yang terdiri dua warna, hitam dan putih. Pada setiap *pixel* pada citra biner cukup direpresentasikan dengan 1 bit. Pada beberapa aplikasi, pada citra biner ini masih tetap digunakan misalnya pada pengolahan citra dengan menggunakan kode barang (*barcode*) yang tertera pada label barang. Pada citra biner, nilai *pixel* suatu objek adalah 1 dan nilai *pixel* pada latar belakang adalah 0, nilai 0 berwarna putih dan sedangkan nilai 1 berwarna hitam.

### 2. Citra *grayscale*

Citra *grayscale* yaitu citra yang memiliki nilai *pixel*-nya dengan mempresentasikan pada derajat keabuan atau intensitas warna putih. Nilai intensitas paling rendah yaitu mempresentasikan warna hitam dan nilai intensitas paling tinggi mempresentasikan warna putih. Pada umumnya citra *grayscale* memiliki kedalaman *pixel* 8 bit, namun citra *grayscale* ini juga terdapat kedalaman *pixel*-nya 16 bit dalam penggunaannya membutuhkan ketelitian yang tinggi.

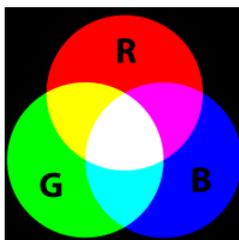
### 3. Citra berwarna

Citra berwarna adalah citra dengan nilai *pixel*-nya mempresentasikan warna tertentu. Pada citra ini memiliki kombinasi dari tiga warna dasar yaitu RGB (*Red, Green, Blue*). Dengan setiap warna dasar menggunakan 8 bit pada kisaran angka 0-255.

## 2.2.6 Pemodelan Warna

### A. Model RGB

Warna-warna yang diterima oleh mata pada manusia yaitu merupakan hasil kombinasi cahaya yang memiliki panjang gelombang yang berbeda-beda. Kombinasi pada warna yang memberikan rentang paling lebar yaitu *red* (R), *green* (G), *blue* (B). Pada ketiga warna itu dinamakan warna pokok (*primaries*), dan singkat pada warna dasar RGB. Pada warna-warna lainnya diperoleh dengan cara mencampurkan pada ketiga warna pokok tersebut dengan perbandingan tertentu (meskipun ketika pencampuran pada ketiga warna tersebut tidak semua kemungkinan warna-warna lain dapat dihasilkan oleh kombinasi RGB saja).



**Gambar 2.1 Model Warna RGB**

Sumber : (Swedia, et all, 2015)

## B. Model HSV

Selain pada RGB, warna juga dapat dimodelkan dengan berdasarkan atribut warnanya. Pada setiap warna memiliki 3 buah atribut, yaitu *Hue (H)*, *Saturation (S)*, dan *Value (V)* (Putra, 2010).

### 1. Hue

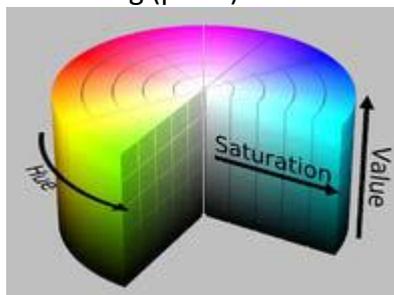
*Hue* menyatakan warna sebenarnya yaitu violet, merah, dan kuning. Pada *Hue* digunakan untuk membedakan warna-warna dan menentukan kemerahan (*redness*), kehijauan (*greenness*), dsb dari cahaya.

### 2. Saturation

*Saturation* menyatakan tingkat kemurnian warna cahaya, dengan mengindikasikan seberapa banyak warna putih yang diberikan pada warna. Sebagai contoh ketika warna merah yaitu 100% warna jenuh (*saturated color*), sedangkan warna pink merupakan warna merah dengan mempunyai tingkat kejenuhan yang rendah (karena didalam warna pink terkandung warna putih). Maka *saturation* menyatakan bahwa seberapa dalam warna tersebut.

### 3. Value

*Value* merupakan atribut dengan menyatakan banyaknya cahaya yang dapat diterima oleh mata tanpa memperdulikan warna. Nilainya berkisar antara gelap (hitam), dan terang (putih).



**Gambar 2.2 Ruang Warna HSV**

Sumber : (Swedia, Cahyani, 2015)

**C. Konversi RGB Ke HSV**

Konversi dari RGB ke HSV pada umumnya akan dilakukan untuk mendapatkan warna yang sebenarnya dari sebuah gambar. Dalam ruang warna RGB, warna yang sama tetapi dengan luminasi yang berbeda akan menghasilkan nilai RGB yang jauh berbeda. Sedangkan pada ruang warna HSV, nilai *Hue* dari warna yang sama akan memiliki nilai yang berdekatan, sedangkan pada *Saturation* dan *Value* akan menunjukkan tingkat luminasi dari sebuah warna. Pada nilai *Hue* adalah sudut dari 0°-360°, sedangkan pada *Saturation* dan *Value* adalah nilai 0-1 (Fitriyah, 2016). Berikut merupakan rumus konversi RGB ke HSV ditunjukkan pada persamaan 2.1 :

$$\begin{aligned}
 R' &= R/255 \\
 G' &= G/255 \\
 B' &= B/255 \\
 C_{max} &= \max(R', G', B') \\
 C_{min} &= \min(R', G', B') \\
 \Delta &= C_{max} - C_{min} \\
 \text{Hue Calculation} & \\
 H &= \begin{cases} 0^\circ & \Delta = 0 \\ 60^\circ \times \left( \frac{G' - B'}{\Delta} \bmod 6 \right) & C_{max} = R' \\ 60^\circ \times \left( \frac{B' - R'}{\Delta} + 2 \right) & C_{max} = G' \\ 60^\circ \times \left( \frac{R' - G'}{\Delta} + 4 \right) & C_{max} = B' \end{cases} \\
 \text{Saturation Calculation} & \\
 S &= \begin{cases} 0 & C_{max} = 0 \\ \frac{\Delta}{C_{max}} & C_{max} \neq 0 \end{cases} \\
 \text{Value Calculation} & \\
 V &= C_{max}
 \end{aligned} \tag{2.1}$$

**2.2.7 Pengambangan (Thresholding)**

*Thresholding* yaitu merupakan metode segmentasi yang paling sederhana, dimana memberikan nilai 1 pada nilai intensitas yang akan masuk dalam batas (*threshold*), sedangkan pada nilai 0 untuk sebaliknya. Untuk memisahkan antar objek yang dideteksi dan *background*.

Proses *thresholding* terhadap citra *grayscale* bertujuan menghasilkan citra biner, secara matematis dapat ditulis seperti persamaan 2.2 :

$$g(x, y) = \begin{cases} 1 & \text{if } f(x, y) \geq T \\ 0 & \text{if } f(x, y) < T \end{cases} \tag{2.2}$$



Untuk menampilkan warna dengan nilai intensitas pixelnya lebih kecil menggunakan persamaan 2.3 sebagai berikut :

$$g(x,y) = \begin{cases} 1 & \text{if } f(x,y) \leq T \\ 0 & \text{if } f(x,y) > T \end{cases} \quad (2.2)$$

Dengan  $g(x,y)$  merupakan citra biner dari citra *grayscale*  $f(x,y)$ , dan  $T$  menyatakan nilai *threshold* (ambang). Nilai  $T$  mempunyai peranan yang penting pada proses pengambangan. Kualitas hasil pada citra biner bergantung pada nilai  $T$  yang akan digunakan (Kumaseh, et al, 2013)

### 2.2.8 Morphological Image Filtering

*Morphological Image Filtering* merupakan proses menganalisa dan mengoperasikan gambar secara *geometric* (bentuk) yang menggunakan teori himpunan (*set theory*). Filter ini menggunakan *structure element* sebagai bentuk yang didefinisikan sebelumnya, *structure element* tersebut diimplementasikan ke dalam gambar menggunakan operasi himpunan *intersection* ( $\cap$ ) dan *union* ( $\cup$ ). *Morphological Filter* pada umumnya diimplementasi menggunakan gambar biner yang mempunyai nilai 0 dan 1. (Fitriyah, 2016).

*Structure Element* diimplementasikan ke dalam bentuk gambar dengan mempunyai 4 operasi *Morphological Filter* dasar, antara lain :

1. *Dilation* merupakan operasi untuk membuat geometris pada gambar menjadi terlihat membesar atau menebal. Pada *dilation* menggunakan operasi *union* antara gambar input dengan *structure element* B, yakni pada persamaan (2.3) :

$$A \oplus B = \bigcup_{b \in B} A_b \quad (2.3)$$

2. *Erosion* merupakan operasi untuk membuat geometris pada gambar menjadi terlihat mengecil atau menipis. Pada *Erosion* yaitu menggunakan operasi *intersection* yang dimana antar gambar A dengan *structure element* B, yakni yakni pada persamaan (2.4) :

$$A \ominus B = \bigcap_{b \in B} A_{-b} \quad (2.4)$$

3. *Opening* merupakan operasi morfologi yaitu *dilation* yang kemudian diikuti dengan *erosion*. *Opening* pada umumnya mengimplementasikan untuk menghilangkan objek yang kecil pada *background* pada sebuah gambar, yakni pada persamaan (2.5) :

$$A \circ B = (A \ominus B) \oplus B \quad (2.5)$$

4. *Closing* merupakan operasi morfologi yaitu *erosion* yang kemudian diikuti dengan *dilation*. *Closing* pada umumnya mengimplementasikan untuk menghilangkan lubang kecil yang terdapat pada *foreground* pada sebuah gambar, yakni pada persamaan (2.6) :

$$A \bullet B = (A \oplus B) \ominus B \quad (2.6)$$

### 2.2.9 Library OpenCV

*Library OpenCV (Open Source Computer Vision Library)* merupakan sebuah *library* perangkat lunak yang ditujukan untuk mengolah pengolahan citra (*image processing*). Pada *library* ini bebas digunakan dan berada didalam naungan sumber BSD, serta memiliki keuntungan menjadi kerangka *multi-platfrom* (Derisma, 2017).

*OpenCv* ditulis dengan menggunakan bahas pemrograman C, C++, dan Python yang dapat berjalan di Linux, Windows, dan MacOS (Denny, 2012). *OpenCV* sendiri terdiri dari 5 *Library*, yaitu :

1. CV : sebagai algoritma pemrosesan gambar (*image processing*) dan *vision*
2. MLL : sebagai *machine learning library*.
3. HighGUI : sebagai GUI, *image* dan Video I/O.
4. CvAux
5. CXCORE : sebagai struktur data, yang support terhadap XML dan fungsi-fungsi grafis.

### 2.2.10 Fungsi-Fungsi Dalam Library OpenCV

*OpenCV* yaitu merupakan sebuah pustaka perangkat lunak yang ditunjukkan sebagai pengolahan citra dinamis secara *real-time* yang dibuat oleh Intel. (Derisma, 2017). *OpenCv* yang digunakan adalah *library opencv 2.4.13*, Berikut merupakan fungsi-fungsi dari *library opencv* :

**Tabel 2.1 Fungsi-Fungsi Dalam Library OpenCV**

No.	Fungsi-Fungsi Dalam Opencv	Penjelasan
1.	<i>cv2.VideoCapture()</i>	Fungsi untuk mengaktifkan kamera
2.	<i>cv2.resize()</i>	Fungsi untuk memperkecil ukuran resolusi pada gambar citra
3.	<i>cv2.cvtColor()</i>	Fungsi untuk konversi gambar citra
4.	<i>cv.inRange()</i>	Fungsi menthreshold gambar citra
5.	<i>cv.erode()</i>	Fungsi untuk melakukan proses erosi pada gambar citra
6.	<i>cv.dilate()</i>	Fungsi untuk melakukan proses dilasi pada gambar citra
7.	<i>cv.findContours()</i>	Fungsi untuk menemukan kontur pada gambar citra
8.	<i>cv.boundingRect()</i>	Fungsi untuk penanda <i>rectangle</i> pada gambar citra

9.	<code>cv.rectangle()</code>	Fungsi untuk menandai dengan <i>rectangle</i> pada gambar citra
10.	<code>cv2.imshow()</code>	Fungsi untuk menampilkan gambar citra

### 2.2.11 Raspberry PI 3 Model B



**Gambar 2.3 Raspberry PI 3 Model B**

Sumber: (Raspberry PI)

*Raspberry Pi 3 Model B* berperan sebagai computing data sensor, di dalam kontroller ini akan digunakan kamera *Logitech C525* yang tertanam didalamnya, sensor ini yang berperan penting memberikan pembacaan data gambar terhadap titik dari sebuah api secara real time atau langsung. Pemrosesan data *sensing* dengan menerapkan sensor terjadi di mikrokontroller ini. (Raspberry PI)

Spesifikasi Raspberry PI 3 MODEL B :

1. A 900MHz quad-core ARM Cortex-A7 CPU
2. 1 GB RAM
3. 4 USB Ports
4. Full HDMI Port
5. 40 GPIO pins
6. Ethernet Port
7. Combined 3.5mm audio jack and compsite video
8. Camera Interface (CSI)
9. Display Interface (DSI)
10. Micro SD Card Slot

### 2.2.12 Buzzer Aktif



**Gambar 2.4 Buzzer**

Sumber : (Instructables, 2017)

*Buzzer* adalah komponen elektronika yang dimana memiliki fungsi untuk mengubah getaran listrik untuk menjadi getaran suara. Prinsip kerja *buzzer* memiliki kemiripan dengan *loud speaker*, jadi *buzzer* terdiri dari kumparan yang dimana kumparan tersebut terpasang diafragma yang kemudian kumparan tersebut dialiri arus sehingga menjadi electromagnet. Kumparan yang sudah menjadi electromagnet akan tertarik kedalam ataupun keluar, tergantung kemana arah arus dan polaritas magnetnya bergerak, karena kumparan dipasang pada diafragma maka gerakan kumparan akan menggerakkan diafragma secara bolak-balik sehingga nantinya membuat udara bergetar dan menimbulkan efek suara (Pratama, et all, 2012).

*Buzzer* banyak sekali digunakan sebagai indikator atau penanda suatu proses ketika proses tersebut telah selesai diproses, atau juga biasanya digunakan ketika untuk penanda kesalahan pada sebuah alat (*alarm*).

### 2.2.13 Module SIM900A



**Gambar 2.5 Module SIM900A**

Sumber : (belajarduino, 2016)

*Module SIM900A* merupakan modul GSM/GPRS untuk uC, Arduino, dan *Raspberry Pi*, yang dapat digunakan untuk mengirim sms, calling, tranfer data dengan melalui GPRS dan DTMF fungsi, *Module GSM-GPRS* memiliki chip SIM900A dari simcomm yang dilengkapi dengan antenna yang nantinya terpasang. Serta

memiliki fitur Quad-Band 850/900/1800/1900 MHz, GPRS multi-slot class 10/8 dan GPRS mobile station class B, perintah yang digunakan yaitu *AT-Commands* dan *SIMCOM AT-Commands* dan dapat memberikan layanan pesan singkat (SMS) (nn-digital, 2016). Spesifikasi *Module SIM900A* sebagai berikut :

1. PCB size : 71.4mm x 66.0mm x 1.6mm
2. Indikator : Power, status LED, net LED
3. Power Supply : 5v
4. Communication Protocol : UART
5. RoHS : Yes

### 2.2.14 Kamera *Logitech C525*



**Gambar 2.6 Kamera *Logitech C525***

Sumber : ([www.logitech.com](http://www.logitech.com))

Kamera *Logitech C525* merupakan sebuah perangkat yang dapat merekam dan meng-*capture* suatu kejadian yang berada disekitar lingkungan yang terpasang perangkat ini. Kamera *Logitech C525* memiliki fitur yang cocok untuk dibuat sebagai pengolahan citra.

Fitur dari Kamera *Logitech C525* sebagai berikut :

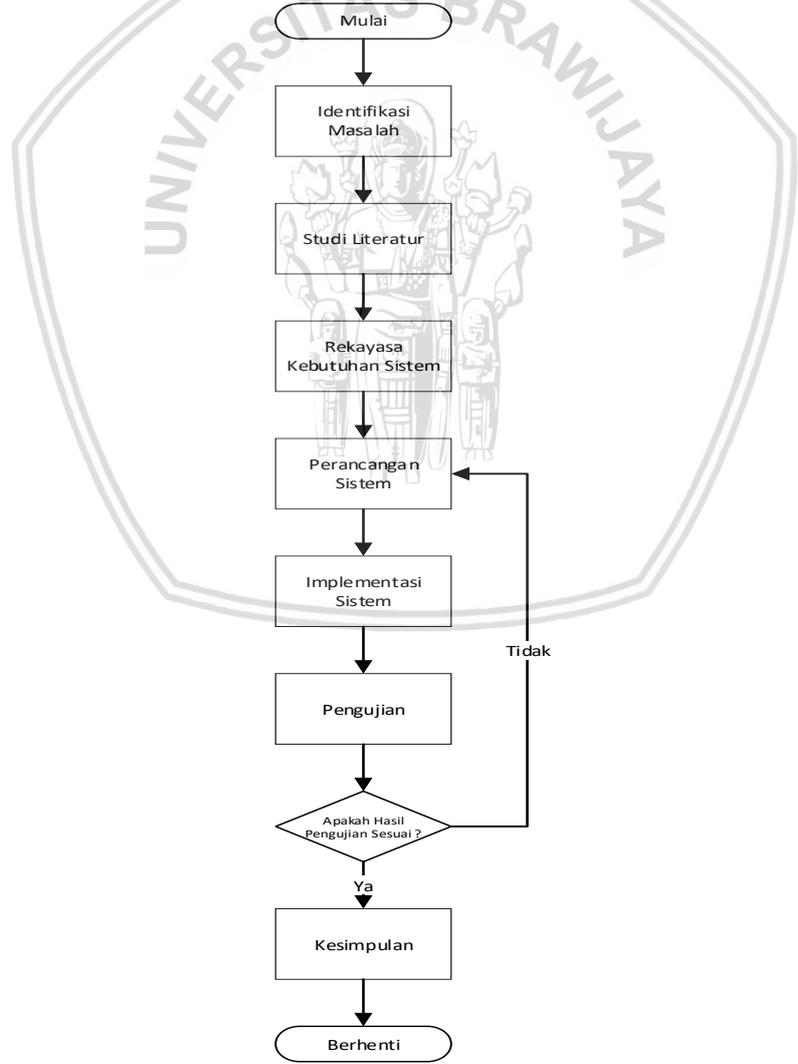
1. *Full HD 720p recording*
2. *H.264 video Standard*
3. *720p HD Video Calls,*
4. *USB 2.0 Compatibility*
5. *RightSound Technology*
6. *Resolution 1280 x 720*
7. *Autofocus*

### BAB III

### METODE PENELITIAN

Pada penelitian ini, langkah awal yang diambil adalah identifikasi masalah yang terdapat pada *Image Processing*. Kemudian dilanjutkan dengan studi literatur tentang tinjauan pustaka dari penelitian sebelumnya serta dasar teori yang menunjang sistem ini. Sifat penelitian ini merupakan implementatif sehingga diperlukan rekayasa kebutuhan dalam menganalisa kebutuhan fungsional dan non-fungsional. Dari rekayasa kebutuhan kemudian sistem dibawa pada bagian perancangan dan implementasi. Pada bagian ini *flowchart* pada masing-masing subsistem yang akan diimplementasikan dan dirancang menjadi satu kesatuan sistem. Pengujian dan analisis dilakukan untuk mengetahui apakah sistem sudah berjalan dengan baik.

Pada penelitian ini memiliki diagram alir penelitian pada Gambar 3.1 berikut ini :



Gambar 3.1 Diagram Alir Penelitian



### 3.1 Studi Literatur

Studi literatur pada penelitian ini didukung dengan adanya acuan atau literatur guna dapat mendukung penelitian ini. Dasar teori yang dipelajari adalah:

1. Definisi Api dan *Triangle of Fire* (segitiga api)  
Studi literatur, terkait dengan definisi api serta untuk mempelajari dan mengetahui proses terjadinya titik api dengan teori *Triangle of Fire* serta jenis-jenis api.
2. Teori dan Konstruksi Sebuah Sistem untuk Mendeteksi Titik Api pada Kebakaran dalam Ruang  
Studi literatur terkait, peneliti mendapatkan beberapa referensi jurnal survey yang menjelaskan mengenai bagaimana membangun sistem atau konstruksi sebuah deteksi titik api pada kebakaran dalam ruangan, dengan beberapa penempatan *actuator* dan sistem modul hardware yang akan di terapkan, serta penerapan *Image Processing*, dan keuntungannya.
3. *Image Processing*  
Studi literatur, dengan mencari jurnal atau penelitian terdahulu yang mempunyai konsep terkait dengan *Image Processing* pada penelitiannya serta menjelaskan bagaimana teori dan implementasi *Image Processing*. Pada tahap ini mempelajari struktur umum dan proses pengolahan citra itu terjadi serta spesifikasi node dan sensor yang akan digunakan pada suatu lingkungan
4. Segmentasi Citra Api  
Studi literatur, dengan mencari dan mempelajari jurnal atau penelitian terlebih dahulu untuk mengetahui segmentasi-segmentasi untuk citra api.
5. Jenis Citra  
Studi literatur, dengan mencari dan mempelajari jurnal atau penelitian terdahulu untuk mengetahui jenis-jenis citra yang terdapat pada suatu objek gambar.
6. Pemodelan Warna  
Studi literatur, dengan mempelajari macam-macam pemodelan warna yang terdapat pada suatu gambar atau *image* khususnya pada pemodelan warna RGB dan HSV.
7. Pengambangan (*Thresholding*)  
Studi literatur, dengan mempelajari tentang pengambangan (*thresholding*) pada suatu citra.

#### 8. *Library OpenCV*

Studi literatur, terkait dengan *OpenCV* dengan mempelajari cara kerja *OpenCv* digunakan untuk mengolah dan mendeteksi suatu *object* gambar yang didapatkan.

#### 9. *Morphological Image Filtering*

Studi literature, terkait dengan *Morphological Image Filtering* dengan mempelajari kerja dari metode ini untuk mengolah dan membedakan *object* api dengan *object* disekitar api.

#### 10. Fungsi-Fungsi Dalam *Library OpenCV*

Studi literatur, terkait dengan fungsi-fungsi *library OpenCv* dan penjelasan fungsi-fungsi yang digunakan.

#### 11. *Raspberry PI 3 Model B*

Studi literatur, terkait dengan mikroprosesor *Raspberry PI* yang nantinya berguna untuk mengetahui karakteristik dan cara menggunakan mikroprosesor tersebut.

#### 12. Buzzer

Studi literatur, terkait modul *Buzzer* yang nantinya modul tersebut dapat digunakan serta dapat diintegrasikan dengan mikrokontroller yang terkait.

#### 13. *Module SIM900A*

Studi literatur, terkait *Module SIM900A* yang nantinya modul tersebut digunakan serta dapat diintegrasikan dengan mikrokontroller yang terkait.

#### 14. Modul Kamera *Logitech C525*

Studi literatur, terkait dengan modul *Kamera Logitech C525* yang digunakan agar modul dapat diintegrasikan dengan mikrokontroller.

### 3.2 Rekayasa Kebutuhan

Rekayasa kebutuhan ini untuk mempermudah dalam melakukan perancangan dan implementasi pada sistem ini. Rekayasa kebutuhan disini dibagi menjadi 4 yaitu gambaran umum sistem, kebutuhan sistem, kebutuhan perangkat keras, dan perangkat lunak.

#### 3.2.1 Gambaran Umum Sistem

Menjelaskan kegunaan pada sistem. Sistem ini mengolah sebuah data dari sebuah *object* gambar yang diambil dari sensor pendeteksi titik api pada kebakaran yaitu kamera *Logitech C525* untuk dilakukan pengolahan

gambar. Kamera *Logitech C525* dipasang pada 2 ruangan yang berbeda, pada setiap ruangan kamera diletakkan dipojok diatas pintu agar dapat menjangkau setiap sudut ruangan. Kegunaan pada sistem ini dilakukan mendeteksi titik api pada kebakaran yang terdeteksi yang dihasilkan oleh pembakaran tersebut, dan dengan mendeteksi titik api serta mendeteksi sistem mengeluarkan peringatan berupa *alarm* dan mengirim notifikasi berupa SMS sebagai tanda telah terjadi sebuah kebakaran diarea tesebut.

### 3.2.2 Kebutuhan Sistem

Beberapa kebutuhan utama sistem yang diperlukan untuk perancangan dan implementasi sistem yaitu :

1. Kamera *Logitech C525* harus mampu menangkap *object* gambar sesuai data yang diperintahkan.
2. Kamera *Logitech C525* harus mampu menangkap *object* gambar sesuai data yang diperintahkan. Kemudian *object* sebuah gambar citra digital tersebut diolah oleh *Rashberry Pi* untuk mengetahui apakah *object* gambar citra digital tersebut sesuai dengan data yang ditentukan.
3. harus mampu melakukan seleksi pada *object* gambar untuk menghindari redudansi data.

### 3.2.3 Kebutuhan Perangkat Keras

Kebutuhan perangkat keras yaitu meliputi:

1. 1 *PC (PERSONAL COMPUTER)*
2. 1 Perangkat *Raspberry Pi 3 Model B*
3. 2 Kamera *Logitech C920 720p HD*
4. 1 Buzzer
5. 1 Modul *SIM900A*
6. 1 Monitor
7. Bearboard dan kabel jumper

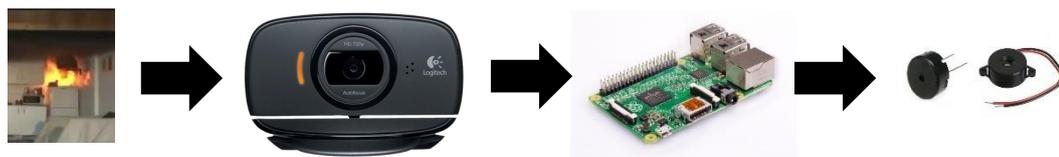
### 3.2.4 Kebutuhan Perangkat Lunak

Perangkat lunak yang digunakan meliputi :

- a. *Windows Remote Desktop.*
- b. *Library OpenCV*
- c. Bahasa pada pemrograman *Python 3.5.*

## 3.3 Perancangan Sistem

Perancangan sistem pada penelitian ini menjelaskan tentang perangkat yang digunakan beserta spesifikasinya. Kemudian diagram blok yang menjelaskan komunikasi sistem secara keseluruhan. Dalam perancangan ini perangkat yang digunakan yaitu terdiri dari beberapa komponen yang dirakit menjadi sedemikian rupa sehingga menjadi sebuah node untuk mendeteksi titik api pada kebakaran dalam ruangan . berikut adalah diagram blok sistem



**Gambar 3.2 Gambaran Sistem Secara Umum**

Pada Gambar 3.2 Menjelaskan Bahwa :

1. Kamera Logitech C525 akan merekam dan mengambil dengan meng-capture object gambar titik api ketika terjadi kebakaran dalam ruangan kemudian object gambar tersebut dikirimkan pada mikroprocessor.
2. Raspberry PI 3 Model B berfungsi sebagai pemberi perintah dan pengolahan data hasil dari object gambar tersebut.
3. Pada Raspberry PI 3 Model B data berupa object gambar titik api pada kebakaran dari kamera Logitech C525 diolah untuk dilakukan pendeteksi titik api dari sebuah citra digital.
4. Setelah object gambar berupa titik api pada kebakaran dideteksi, maka sistem akan mengeluarkan berupa peringatan alarm tanda bahaya dan akan mengirim layanan pesan singkat (sms).

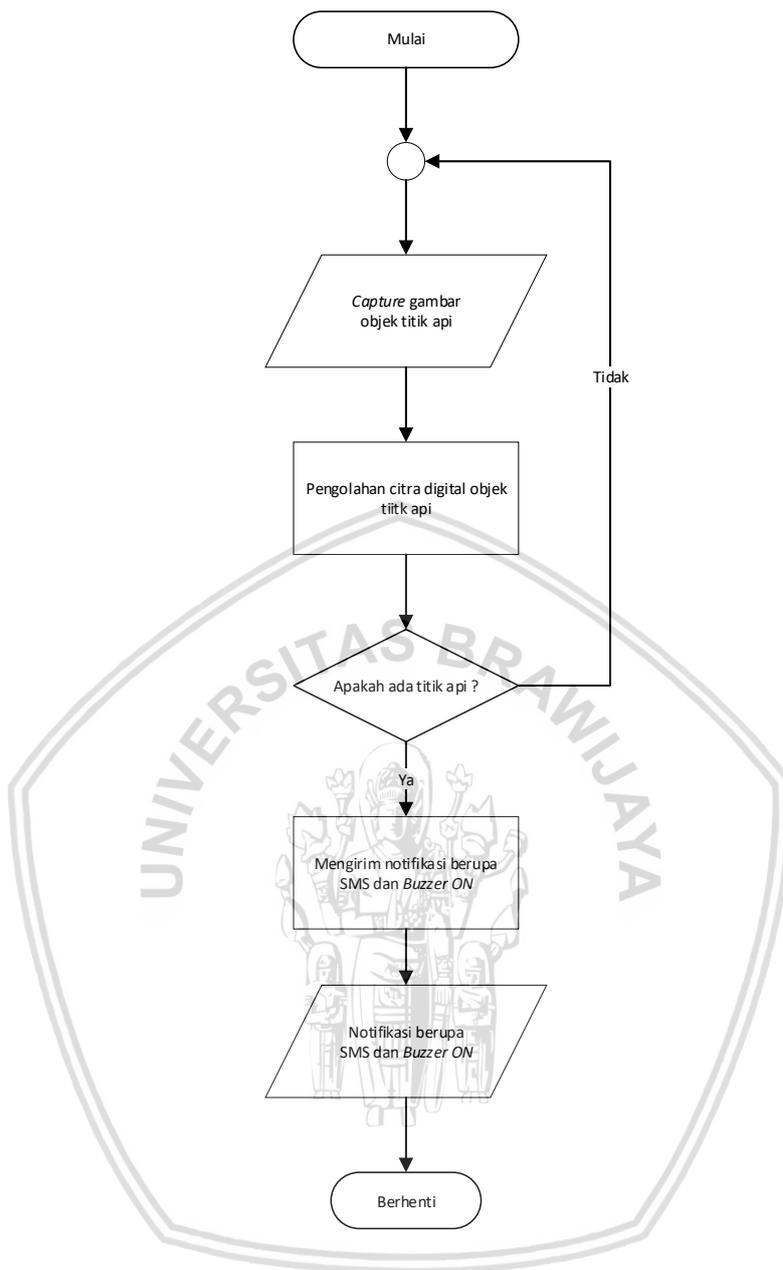
Pada sistem ini menggunakan Image Processing dimana Image Processing berfungsi untuk mengolah object gambar titik api pada kebakaran yang akan di deteksi dengan membedakan antara object api dengan object yang berada disekitar api dari sebuah citra digital, yang dikhususkan untuk mendeteksi titik api pada kebakaran yang terjadi dalam ruangan.

### 3.4 Implementasi Sistem

Pada tahap implementasi merupakan tahapan untuk merakit sistem deteksi titik api pada kebakaran dalam ruangan. Tahapan dari implementasi sistem pada penelitian ini yaitu :

1. Implementasi perangkat Raspberry PI dan modul kamera Logitech C525 untuk node deteksi titik api.
2. Implementasi Image Processing untuk pengolahan data object gambar untuk deteksi titik api pada kebakaran dalam ruangan.
3. Implementasi metode untuk mencegah redundansi data pada pengolahannya.

Untuk flowchart dari sistem dapat dilihat pada gambar 3.3 dibawah ini :



Gambar 3.3 Flowchart sistem

### 3.5 Pengujian dan Analisis

Pada tahap Pengujian dan Analisis menguji beberapa subsistem yang terdapat pada sistem. Subsistem yang diuji yaitu kesesuaian sistem pada saat merekam dan mengambil dengan meng-*capture object* gambar berupa titik api dan pengolahan *object* gambar dengan *Image Processing*. Pengujian yang dilakukan pada sistem ini yaitu :

1. Menguji kemampuan fungsional dari perangkat sensor node yang dibuat dari mikroprocessor *Raspberry PI 3 Model B* dan modul kamera *Logitech C525*.

2. Menguji pengolahan *object* gambar dengan menggunakan *Image Processing* dengan membedakan *object* antara *object* api dengan *object* yang lainnya.
3. Menguji kemampuan node dalam melakukan pemilihan data sehingga tidak terjadi redundansi data.

Dari pengujian diatas, analisis dilakukan sebagaimana untuk menyempurnakan metode pada sistem. Sehingga metode dapat berjalan secara maksimal pada sistem. Dari pengujian diatas juga akan berdampak pada penelitian tentang *Image Processing* yang akan menjadi keakuratan dalam pengolahan data dan dapat meminimalkan sumber daya.

### 3.6 Pengambilan Kesimpulan

Setelah tahap perancangan, implementasi dan pengujian dilakukan, maka tahap selanjutnya adalah dengan tahap kesimpulan. Kesimpulan diambil dikarenakan gambaran dari hasil implementasi yang dilakukan yaitu implementasi untuk mengolah *object* gambar dan mendeteksi titik api pada kebakaran dengan menggunakan *Image Processing*. Setelah kesimpulan diikuti dengan saran yang berguna untuk memperbaiki penelitian ini untuk selanjutnya. Kemudian saran ini digunakan sebagai pertimbangan dalam penerapan pengolahan *object* gambar dan mendeteksi titik api dan kebakaran dengan menggunakan *Image Processing* pada komputasi citra dan suara digital.

## BAB IV

### REKAYASA PERSYARATAN

#### 4.1 Gambaran Umum Sistem

Sistem deteksi titik api kebakaran terpusat secara garis besar untuk mendeteksi titik api ketika terjadi kebakaran agar ditangani secara cepat dan tepat serta meningkatkan tingkat kewaspadaan terhadap deteksi dini titik api kebakaran dalam ruangan. Seringnya terjadi kelalaian terhadap deteksi dini titik api kebakaran pada ruangan, maka diperlukan alat untuk mendeteksi titik api kebakaran dalam ruangan. Ketika sistem mendeteksi titik api kebarakan yang terjadi dalam ruangan, maka nantinya sistem akan mengeluarkan peringatan berupa *alarm*. Dengan menggunakan pengolahan citra digital, citra yang berasal dari titik api kebakaran akan diidentifikasi sehingga titik api dapat terdeteksi. Selanjutnya data tersebut diolah untuk ditampilkan pada komputer yang berada pada ruangan *control*, sehingga informasi akan didapat secara efektif.

##### 4.1.1 Tujuan

Percangan sistem ini bertujuan untuk mendeteksi titik api kebakaran yang terjadi didalam ruangan dengan menggunakan pengolahan citra digital. Pengambilan citra digital menggunakan webcam yang terpasang pada setiap ruangan yang ditentukan. Proses pengolahan citra ini dilakukan untuk mendeteksi titik api kebakaran, apabila sistem mendeteksi titik api kebakaran maka sistem akan mengeluarkan sebuah peringatan berupa *alarm*, mengirim notifikasi berupa layanan pesan singkat (via sms) dan hasilnya akan ditampilkan pada komputer yang terhubung serta akan memunculkan bunyi sebagai notifikasi bahwa terdeteksi titik api.

##### 4.1.2 Kegunaan

Sistem ini berguna untuk memantau ketika terjadi titik api kebakaran. Sistem ini diharapkan akan mempermudah untuk mendeteksi titik api kebakaran sedini mungkin dan mempercepat waktu penanganan titik api kebakaran tersebut sehingga tidak menyebabkan titik api kebakaran tersebut meluas.

##### 4.1.3 Karakteristik Pengguna

Penggunaan sistem bertindak sebagai penerima hasil deteksi titik api kebakaran dari *record webcame* yang terpasang pada setiap ruangan yang berupa bunyi *alarm* sebagai notifikasi sistem dan mengirim notifikasi berupa layanan pesan singkat (via sms).

##### 4.1.4 Lingkungan Operasi

Pada persayatan kebutuhan lingkungan yang mendukung sistem yaitu :

1. Pengambilan gambar citra ketika mendeteksi titik api kebakaran dalam ruangan.
2. Pengambilan sampel dilakukan pada ruangan dengan kondisi cuaca terang dan lampu dinyalakan.
3. Pengambilan gambar citra dilakukan di dalam *prototype* mirip ruangan 40cmx40cm setiap ruangnya.

## 4.2 Analisi Kebutuhan Sistem

Analisis kebutuhan bertujuan untuk menggambarkan kebutuhan-kebutuhan dari sistem. Kebutuhan fungsional serta non-fungsional akan dianalisis sesuai dengan kebutuhan sistem sehingga akan mempermudah dalam mendesain dan mengimplementasikan sistem.

### 4.2.1 Kebutuhan Fungsional

1. Kamera *Logitech C525 Webcam* dapat mengambil citra

Kamera *Logitech C525 Webcam* yaitu mempunyai tugas untuk mengambil citra. Citra tersebut diambil dengan menggunakan *library OpenCV* dalam bentuk format warna BGR.

2. Raspberry Pi dapat membaca citra titik api kebakaran yang ditangkap oleh webcam.

Pada fungsi ini, Raspberry Pi dapat membaca setiap data hasil pengambilan gambar oleh webcam yang merupakan citra titik api kebakaran dan digunakan sebagai *input* sistem.

3. Sistem dapat melakukan proses pengolahan citra digital.

Pada citra titik api kebakaran dilakukan proses pengolahan citra secara digital. Dimulai dari merubah citra BGR ke biner bertujuan memisahkan *object* titik api kebakaran dengan *object* yang berada disekitar titik api atau yang tidak diinginkan pada sebuah gambar citra digital. Selanjutnya dilakukan proses morfologi yaitu merencanakan bagian-bagian dari citra agar mendapatkan citra yang sesuai dan lebih mudah untuk mengidentifikasi kondisi titik api kebakaran nantinya.

4. Sistem dapat mendeteksi titik api kebakaran di setiap ruangan.

Tahap deteksi titik api kebakaran berpengaruh terhadap tahap selanjutnya karena jika salah mendeteksi maka hasil dari *output* yang diterima tidak akan sesuai dengan kondisi nyatanya. Pada sistem ini menggunakan 2 ruangan yang masing-masing ruangan terpasang kamera dengan tujuan untuk mendeteksi titik api kebakaran jika terjadi kebakaran. Letak dari kamera pada ruangan 1 tidak dapat mendeteksi apa yang terjadi pada kamera di ruangan 2 atau sebaliknya. Sehingga masing-masing kamera yang ditempatkan pada setiap ruangan hanya fokus untuk mendeteksi diruangan itu saja.

5. Sistem dapat menampilkan kedalam layar monitor, serta menghidupkan indikator pada berupa *alarm*, layanan pesan singkat (sms) serta menampilkan pada aplikasi monitor.

Setelah ruangan *control* menerima pengolahan tersebut maka hasil selanjutnya akan ditampilkan pada layar monitor, serta menghidupkan indikator berupa *alarm* dengan adanya bunyi sebagai notifikasi atau peringatan dan dapat mengirimkan layanan pesan singkat (sms) yang terintegrasikan pada sistem.

#### 4.2.2 Kebutuhan Non-Fungsional

Kebutuhan non-fungsional sebaiknya menggunakan kamera dengan resolusi 320x240 *pixels* karena dapat mempecepat proses pengolahan citra digital dan pendeteksian titik api kebakaran. Jika resolusi kamera semakin besar maka proses pengolahan citra digital akan semakin lambat.

#### 4.2.3 Kebutuhan Antarmuka Perangkat Lunak

Antarmuka perangkat lunak pada sistem ini yaitu :

1. *Raspbian Operating System*

Sistem operasi yang digunakan pada *Raspberry Pi* yaitu *Raspbian*. Ada beberapa jenis *Raspbian Operating System*, namun yang diggunakan pada penelitian ini berupa *Raspbian Jessie*.

2. *Windows 10 Operating System*

Sistem operasi yang terinstall pada komputer adalah *Windows 10*.

3. *Library OpenCV 2.4.13*

*Library OpenCV* library yang berperan dalam proses pengolahan citra digital.

4. *Python 3.5*

*Python 3.5* yaitu berperan sebagai Bahasa pemrogram untuk memprogram untuk mengontrol semua inputan dan ouput yang dihasilkan.

5. *Real VNC Viewer*

Perangkat lunak ini digunakan mengontrol *Raspberry Pi 3 Model B* pada laptop yang terhubung yang kemudian menampilkan GUI dari *Raspbian Jessie* pada Monitor Laptop.

#### 4.2.4 Kebutuhan Antarmuka Perangkat Keras

Antarmuka perangkat keras pada sistem ini yaitu :

1. Perangkat Komputer

Komputer digunakan untuk menjalankan sistem dan untuk menampilkan GUI *Raspbian* hasil deteksi titik api kebakaran dalam ruangan serta berfungsi sebagai media mengontrol *Raspberry Pi 3 Model B*.

- Model Perangkat : Lenovo
- Processor : Intel(R) Core(TM) i5-4200U CPU
- Sistem Operasi : Windows 10 Enterprise 64-bit

## 2. Kamera *Logitech C525 HD Webcam*

*Webcam* digunakan untuk menangkap citra dari objek yaitu titik api kebakaran pada ruangan.

## 3. *Raspberry Pi*

Mikrokomputer ini berguna sebagai *processing unit*, meliputi proses penerimaan citra yang ditangkap oleh webcam, proses pengolahan citra objek, proses pendeteksiaan objek hingga proses pengiriman hasil deteksi ke layar monitor.

## 4. *Buzzer*

Perangkat ini berguna sebagai indikator output, meliputi proses setelah proses pengolahan citra objek dan objek telah di deteksi maka indikator output ini diaktifkan sebagai *alarm*.

## 5. *Module SIM900A*

Perangkat ini berguna sebagai indikator output yang berguna untuk melakukan proses pengiriman layanan pesan singkat (SMS) peringatan dini terhadap operator ketika telat terdeteksi titik api dalam ruangan.

## 6. Kabel UTP

Kabel UTP dibutuhkan untuk menghubungkan *Raspberry Pi 3 Model B* dengan komputer yang bertindak sebagai monitor.

## 7. Monitor LCD

Monitor LCD dibutuhkan untuk menampilkan hasil deteksi dan GUI dari *Raspbian Jessie* pada *Raspberry Pi 3 Model B*.

### 4.3 Batasan Desain Sistem

Dalam pembuatan Sistem Deteksi Titik Api pada Kebakaran terdapat batasan. Sehingga nantinya pada lingkup pembahasan, perancangan, maupun pada implementasinya tidak terlalu luas. Adapun batas-batas desain sistem ini adalah sebagai berikut :

1. Menggunakan Kamera *Logitech C525 HD Webcam* untuk mengambil citra.
2. Menggunakan *Mini PC Raspberry Pi 3 Model B* untuk mengolah citra.
3. Menggunakan Library *OpenCV* untuk proses deteksi titik api pada kebakaran.
4. Ruangan yang digunakan menggunakan *Prototype* yang terbuat dari triplek yang meyerupai ruangan dengan ukuran 40cmx40cm.
5. *Image* citra api yang akan dianalisa adalah citra api yang berasal dari lilin, korek gas, kapas, satu lembar kertas terbakar, tisu, kain, warna mirip api.

6. Pengujian dilakukan pada *Prototype* yang terbuat dari triplek yang menyerupai ruangan 40cmx40cm.
7. Sistem tidak bergerak pada saat pendeteksian titik api pada kebakaran.
8. Pembahasan difokuskan pada untuk keakuratan untuk mendeteksi titik api pada kebakaran.



## BAB V

### PERANCANGAN DAN IMPLEMENTASI

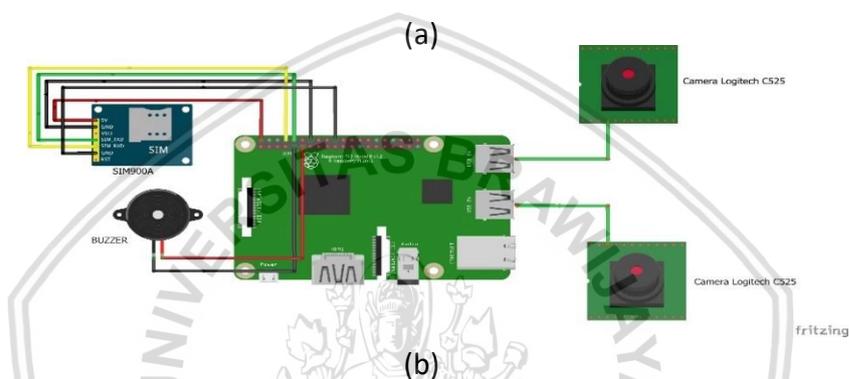
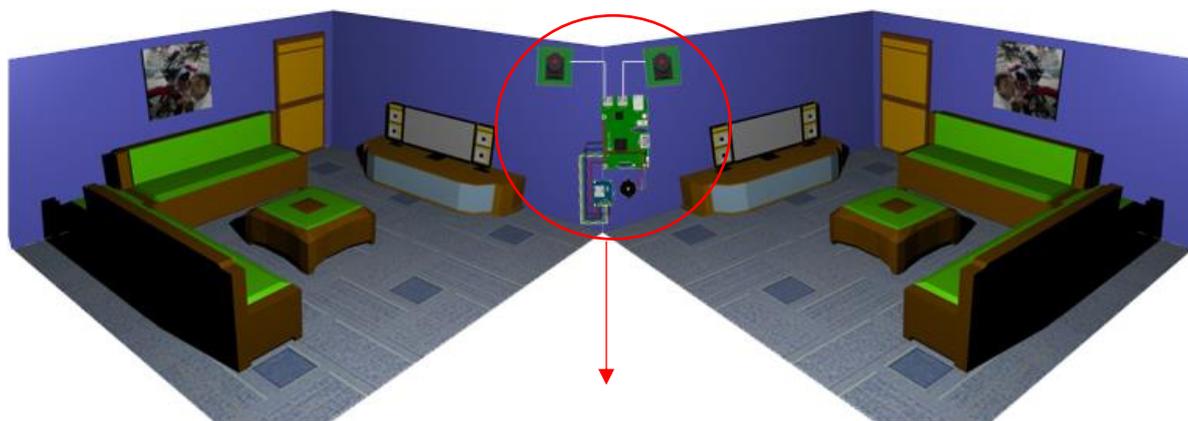
Pada bab perancangan dan implementasi ini dijelaskan mengenai penerapan serta analisi kebutuhan sehingga akan menjadi satu kesatuan sistem yang utuh. Tahap perancangan ini dilakukan perancangan sistem dan implementasi yang akan dibagi kedalam bentuk subsistem perancangan dan implementasi sistem.

#### 5.1 Perancangan Sistem

Pada bagian ini dilakukan perancangan sistem dimana akan diawali dengan perancangan *prototype* sistem, perancangan perangkat keras, serta perancangan perangkat lunak

##### 5.2.1 Perancangan *Prototype* deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms *gateway* pada *raspberry pi*

Sebelum menerapkan implementasi pada deteksi titik api menggunakan Kamera berbasis sms *gateway* pada *raspberry pi* maka akan dilakukan perancangan dan desain alat untuk mendeteksi titik api dengan menggunakan pengolahan citra digital. Pada perancangan sistem ini perlu diperhatikan untuk peletakkan kamera sebagai sensor untuk mengambil citra gambar yang terdapat pada ruangan supaya sistem dapat bekerja dengan maksimal dan mudah digunakan. Perancangan ini sendiri menggunakan aplikasi Fritzing untuk mendesain model *prototype* yang akan digunakan, berikut merupakan desain pada deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms *gateway* pada *raspberry pi* pada gambar 5.1 (a) Desain *Prototype* Ruangan dan (b) Rancangan Sistem Deteksi titik api



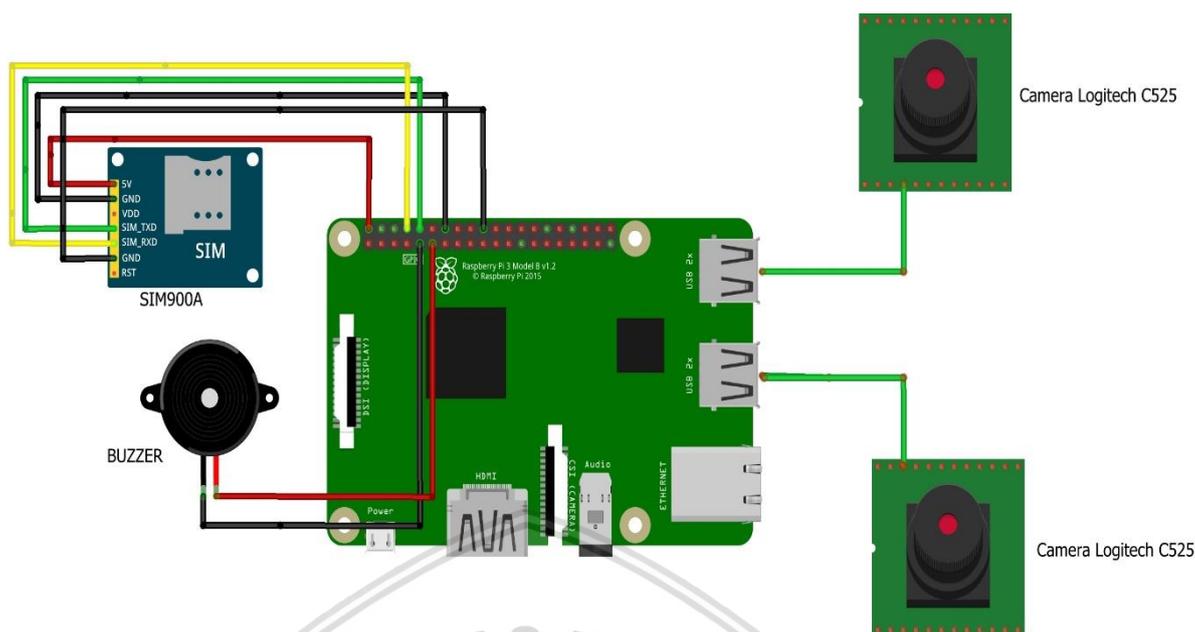
**Gambar 5.1 (a) Desain Desain *Prototype* Ruangan**

**Gambar 5.1 (b) Rancangan Sistem deteksi titik api**

Pada gambar 5.1 (a) merupakan desain *prototype* ruangan dan gambar 5.1 (b) merupakan rancangan *system* deteksi titik api. Pada rancangan tersebut sensor yang digunakan adalah 2 Kamera *Logitech C525* dipasang pada *Raspberry PI 3 Model B* untuk mengambil citra gambar titik api yang terdapat pada ruangan. Untuk *controller* yang digunakan pada *prototypye* ini adalah *Raspberry PI 3 Model B* dan output yang digunakan adalah *Buzzer* untuk mengeluarkan suara untuk peringatan telah terjadi api kecil disekitar ruangan, serta mengirim layanan pesan singkat (via sms) dengan menggunakan modul *SIM900A* untuk memberikan notifikasi terhadap operator telah terdeteksi api yang terdapat disekitar ruangan.

### 5.2.2 Perancangan Perangkat Keras

Pada perancangan perangkat keras dilakukan dengan mengacu pada analisis kebutuhan perangkat keras agar dapat membangun sistem ini sesuai dengan yang diinginkan. Tahap ini, akan dijabarkan skematik rangkaian antar pin pada masing-masing komponen perangkat keras diantara lainnya *Raspberry PI 3 Model 3* sebagai *controller*, 2 buah Kamera *Logitech C525*, *Module SIM900A*, *Buzzer*. Kemudian dirancang rancangan sebuah skematik perangkat keras pada sistem ini. Desain rangkaianya terdapat pada gambar 5.2



**Gambar 5.2 Skematik rangkaian sistem perangkat keras**

Pada gambar 5.2 tersebut dapat dilihat terdapat 2 Kamera *Logitech C525* yang dipasang menggunakan SLOT USB yang terdapat pada *Raspberry Pi 3 Model B*. Kamera tersebut digunakan sebagai sensor untuk mengambil citra gambar yang terdapat pada ruangan untuk mendeteksi titik api. Serta terdapat *Module SIM900A* sebagai indikator untuk mengirim layanan pesan singkat (sms) terhadap operator ketika terdeteksi titik api, komunikasi yang digunakan antara *module SIM900A* dan *Raspberry Pi 3 Model B* yaitu dengan menggunakan komunikasi Serial (UART). Komunikasi ini sebagai pertukaran data dua board yang berbeda ataupun sama, komunikasi ini menggunakan pin *TX (Transmitter)* dan *RX (Receiver)*. Dan juga terdapat *Buzzer* sebagai indikator tanda peringatan sebagai alarm yang terdapat pada sistem yang telah ditentukan. Pada perancangan perangkat keras deteksi titik api menggunakan Kamera berbasis sms gateway pada *Raspberry Pi* menggunakan 2 Kamera *Logitech C525* yang berguna sebagai sensor untuk mengambil citra gambar yang terdapat pada ruangan yang telah dipasang masing-masing kamera, Kamera *Logitech C525* diintegrasikan dengan *Raspberry Pi* yang berfungsi sebagai mikrokontroler.

Pada gambar 5.2 perancangan perangkat keras ini juga menggunakan *Module SIM900A* untuk melakukan pengiriman layanan pesan singkat (sms) sebagai indikator output. Pada perancangan ini juga menggunakan *Module SIM900A* yang dipasang secara serial. Dari spesifikasi *Module SIM900A* tersebut menggunakan komunikasi Serial (UART) yang dimana arsitektur UART terdapat pada master dan perangkat *slave* dipasang secara serial ke modul tersebut. UART sendiri merupakan komunikasi serial yang memanfaatkan *TX(Transmitter)* dan *RX(Receiver)* sebagai pertukaran data untuk mengirim dan menerima data melalui dua bord berbeda ataupun dengan dua bord yang sama. Serta perancangan diatas juga terdapat perancangan perangkat dengan menggunakan *Buzzer* aktif yang dipasang pada *Raspberry Pi* dengan memanfaatkan pin GPIO yang terdapat pada

*Raspberry Pi*. Pada *Buzzer* aktif ini memiliki 2 slot yaitu *VCC* dan *GND*. Berikut adalah konfigurasi PIN yang dapat dilihat pada tabel 5.3

**Tabel 5.1 Konfigurasi Kamera *Logitech C525* dengan *Raspberry Pi***

<i>Kamera Logitech C525</i>	<i>Raspberry pi</i>
USB	SLOT USB
USB	SLOT USB

Dari table 5.1 terlihat Kamera *Logitech C525* sebagai sensor untuk mengambil gambar citra yang terdapat pada masing-masing ruangan dengan dihubungkan dengan *Raspberry Pi* dengan memanfaatkan SLOT USB yang terdapat pada *Raspberry Pi* untuk melakukan komunikasi antara Kamera *Logitech C525* dengan *Raspberry Pi*.

**Tabel 5.2 Koneksi pin Module *SIM900A* pada kiri dengan *Raspberry Pi***

Pin Module <i>SIM900A</i>	Pin <i>Raspberry Pi</i>
5V (VCC)	5V (Pin 1)
GND	Ground (Pin 14)
TX	GPIO 10 RX (Pin 10)
RX	GPIO 8 TX (Pin 8)
GND	Ground (Pin 20)

Dari tabel 5.2 dapat dilihat bahwa pada pin *VCC* modul dihubungkan dengan pin 5V atau pin 1 pada *Raspberry Pi*, kemudian *GND* pada modul dihubungkan dengan pin GND atau pin 14 pada *Raspberry Pi*. TX dan RX merupakan pin untuk komunikasi secara Serial (UART), pin TX pada modul dihubungkan ke pin RX atau pin 10 yang terdapat pada *Raspberry Pi*. Sedangkan pin RX pada modul dihubungkan dengan pin TX atau pin 8 pada *Raspberry Pi*. Untuk pin GND pada modul dihubungkan dengan pin GND atau pin 20 pada *Raspberry Pi*.

**Tabel 5.3 Koneksi pin *Buzzer Aktif* pada kiri dengan *Raspberry Pi***

Pin <i>Buzzer</i>	Pin <i>Raspberry Pi</i>
VCC	HIGH (Pin 9)
GND	Ground (Pin 11)

Pada tabel 5.3 yaitu konfigurasi untuk mengakses *Buzzer* aktif. *Buzzer* aktif digunakan sebagai perangkat indikator peringatan berupa alarm pada output sistem. VCC pada *Buzzer* dihubungkan pada pin 9 pada *Raspberry Pi*. Kemudian

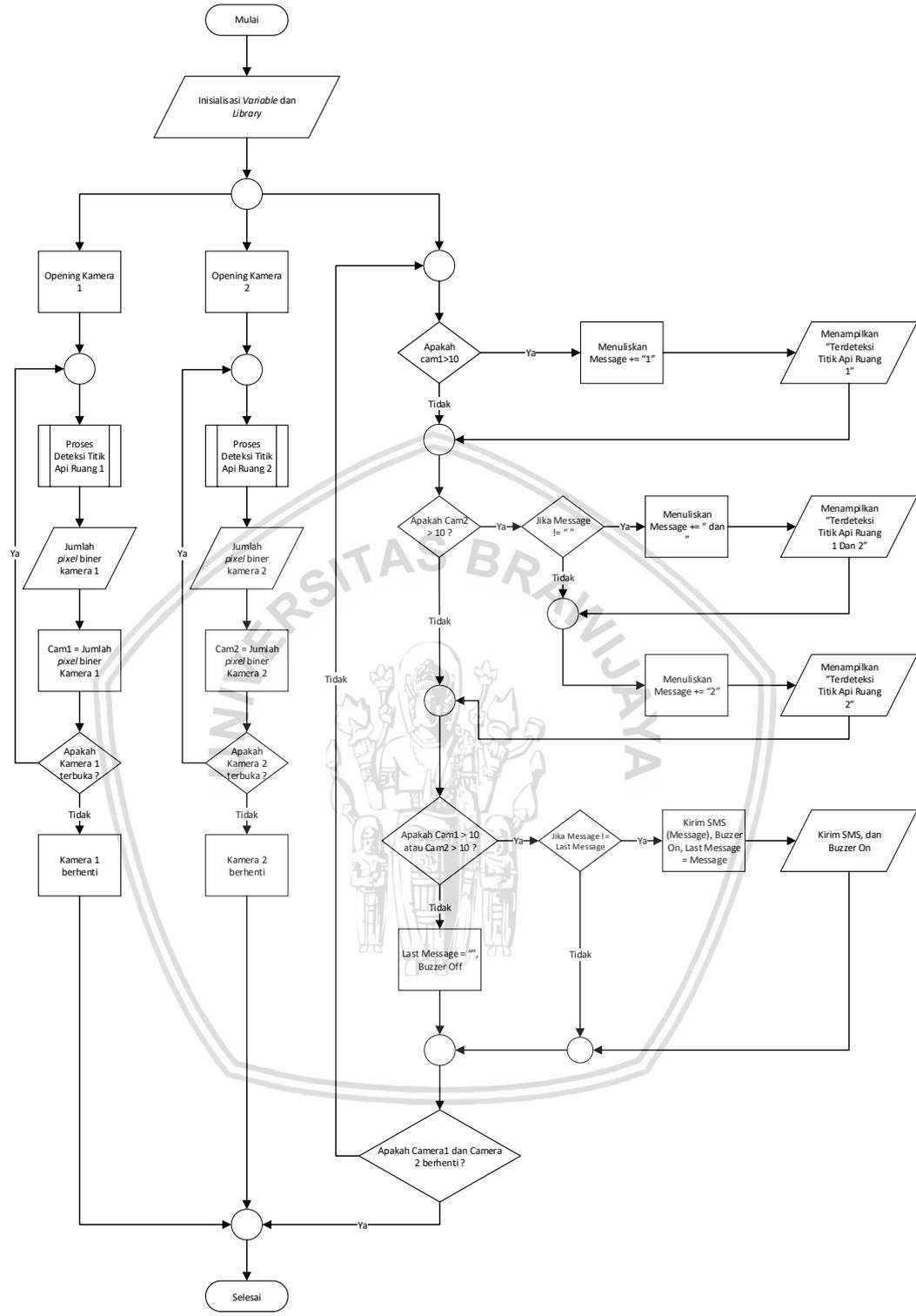
GND dihubungkan dengan Ground ataupun pin 11 pada *Raspberry Pi*. Ketika *Buzzer* diaktifkan maka *Buzzer* akan aktif (*HIGH*) sebaliknya ketika *Buzzer* tidak diaktifkan maka *Buzzer* tidak aktif (*LOW*).

### 5.2.3 Perancangan Perangkat Lunak

Pada pembahasan di perancangan perangkat lunak ini yaitu terdapat 4 bagian perancangan perangkat lunak yang terletak pada perangkat lunak utama yang terdapat pada *Raspberry Pi* yang merupakan *controller* yang merupakan diagram alir program secara umum sistem, bagian kedua yaitu perancangan perangkat lunak dengan mendeteksi objek yaitu objek api. Kemudian bagian ketiga yakni perancangan perangkat lunak dengan menandai objek yang telah terdeteksi dengan *Bounding Rect*, yang terakhir adalah perancangan lunak yaitu *Output* untuk memberikan *Output* berupa notifikasi bunyi *Buzzer* sebagai *alarm* tanda peringatan dan sistem akan memberikan notifikasi berupa layanan pesan singkat (SMS) dengan memanfaatkan *Modul SIM900A*.

#### 5.2.3.1 Perancangan Proses Utama

Proses utama dalam perancangan perangkat lunak yang dapat dilihat pada flowchart 5.3, dalam proses tersebut adalah gambar aliran data dan proses secara umum yang terdapat pada sistem. Pada proses ini akan dilakukan beberapa pekerjaan. Dimulai dengan inialisasi dan mendeklarasikan *variable* dan *library* yang dibutuhkan sistem, kemudian sistem akan memulai untuk mendeteksi objek terutama objek api yang terdeteksi oleh kamera. Jika api terdeteksi maka sistem akan memulai proses pendeteksian dan deteksi tersebut diolah serta akan menandai objek api yang terdeteksi dengan menggunakan *Bounding Rect* atau memberikan kotak sebagai penanda bawah objek api telah terdeteksi, apakah objek api terdeteksi pada kamera di ruangan 1 atau kamera diruangan 2. Jika terdeteksi pada kamera ruangan 1 atau kamera diruangan 2. Maka proses selanjutnya adalah hasil dari penandaan dan jumlah *pixel* biner tersebut kemudian dikirim untuk memberikan *output* terhadap pengguna sistem ini, yaitu dengan memberitahu bahwa telah terdeteksi api baik pada kamera ruangan 1, pada kamera ruang 2, ataupun kamera 1&2, kemudian akan mengaktifkan dan memberikan notifikasi bunyi yang dihasilkan oleh *Buzzer* sebagai *alarm* tanda bahaya, dan yang terakhir adalah akan mengirimkan layanan berupa pesan singkat (SMS) ke pada operator bahwa telah terdeteksi titik api pada ruang 1, ruang 2, dan ruang 1&2.



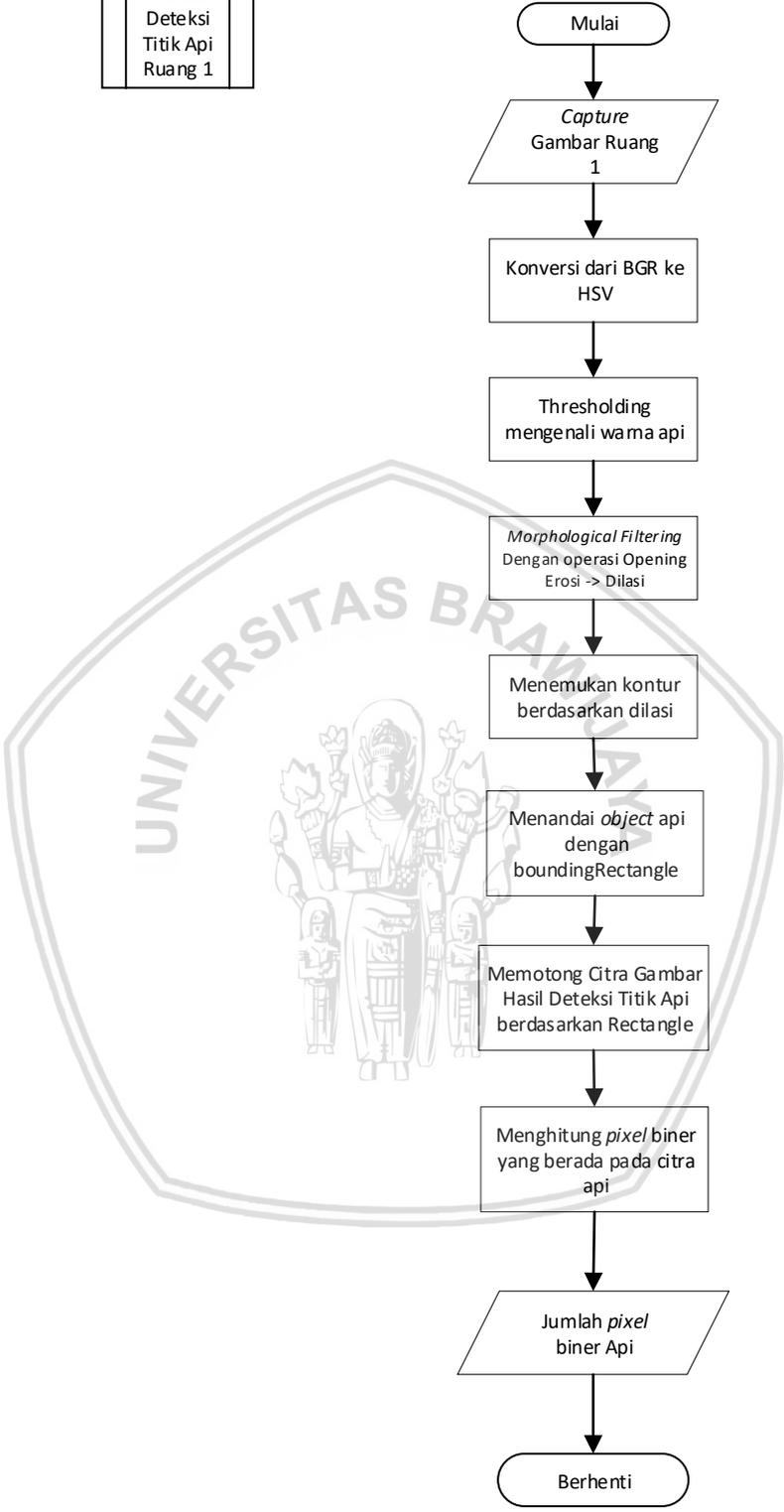
Gambar 5.3 Flowchart proses utama

5.2.3.2 Perancangan Proses Deteksi Titik Api Di Ruang 1

Pada penelitian ini, dilakukan proses perancangan perangkat lunak untuk mendeteksi titik api yang dapat dilihat pada flowchart 5.4 untuk proses deteksi titik api di ruang 1.



Proses Deteksi Titik Api Ruang 1
---

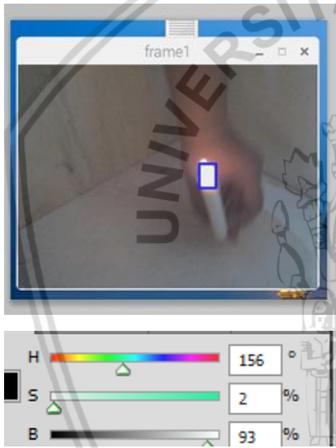


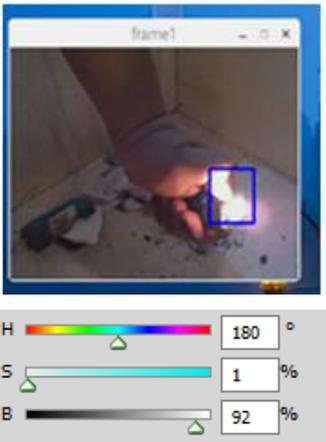
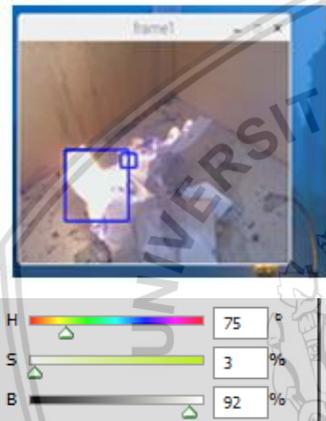
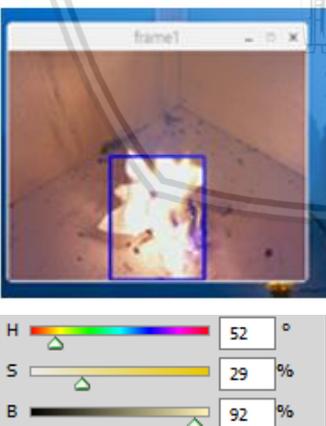
**Gambar 5.4 Flowchart Deteksi Titik Api Di Ruang 1**

Flowchart 5.4 merupakan pemrosesan citra menggunakan *library opencv 2.4.13* yang telah terintegrasi atau terinstall pada *Raspberry Pi 3 Model B*. pada proses pengambilan mendeteksi titik api yaitu dengan menggunakan kamera yang terintegrasi pada *Raspberry Pi*. Proses pertama adalah Pada dasarnya pengambilan

citra yang dilakukan pada *OpenCV* memiliki format *color space* standar yaitu BGR (*Blue Green Red*). *Color space* ini kemudian akan dikonversikan ke dalam bentuk HSV (*Hue Saturation Value*). Pada pendeteksian titik api yang terjadi didalam ruangan, digunakan *color space* HSV karena HSV mempunyai warna-warna yang sama dengan apa yang ditangkap dengan indra penglihatan pada manusia. *Color space* HSV memiliki sensitivitas yang baik terhadap kecerahan cahaya, kemudian citra yang didapat dari hasil konversi akan di-*thresholding* untuk diubah menjadi citra biner. Warna yang tidak sesuai dengan warna yang di-*thresholding* maka akan diubah menjadi nilai 0, sedangkan warna sesuai dengan *thresholding* tersebut maka akan bernilai 1. Dalam hal ini akan mempermudah dalam mendeteksi titik api yang terjadi didalam ruangan. Berikut menentukan batas bawah dan batas atas menggunakan *trackbar* pada tabel 5.4 Berikut :

Tabel 5.4 Menentukan nilai range *thresholding* citra api

No	Gambar Citra Api	Hasil Nilai HSV
1.	<p>Nilai HSV pada Objek 1</p> 	<p><b>Nilai</b>  <math>H = 156</math>  <math>S = 2\% \times 255 = 5.1</math>  <math>B/V = 93\% \times 255 = 237.15</math></p>
2.	<p>Nilai HSV pada Objek 2</p> 	<p><b>Nilai</b>  <math>H = 160</math>  <math>S = 3\% \times 255 = 7.65</math>  <math>B/V = 93\% \times 255 = 237.15</math></p>
3.	<p>Nilai HSV pada Objek 3</p>	<p><b>Nilai</b>  <math>H = 180</math></p>

		<p><math>S = 1\% \times 255 = 2.55</math></p> <p><math>B/V = 92\% \times 255 = 234.6</math></p>
<p>4.</p>	<p>Nilai HSV pada Objek 4</p> 	<p><b>Nilai</b></p> <p><math>H = 75</math></p> <p><math>S = 3\% \times 255 = 7.65</math></p> <p><math>B/V = 92\% \times 255 = 234.6</math></p>
<p>5.</p>	<p>Nilai HSV pada Objek 5</p> 	<p><b>Nilai</b></p> <p><math>H = 52</math></p> <p><math>S = 29\% \times 255 = 73.95</math></p> <p><math>B/V = 92\% \times 255 = 234.6</math></p>

Pada tabel 5.4 merupakan hasil untuk menentukan nilai HSV ( $H = Hue$ ,  $S = Saturation$ ,  $B/V = Brightness / Value$ ) dari hasil 5 kali uji coba untuk mengetahui range *thresholding* nilai HSV pada citra api. Maka disimpulkan range citra api untuk nilai range *thresholding* yaitu  $H = 0-255$ ,  $S = 0-255$ ,  $B/V = 227-225$ .

Pada proses *thresholding* ini masih terdapat beberapa *noise* kecil yang harus dihilangkan. Maka diperlukan untuk menerapkan operasi *Opening* untuk menghilangkan *noise* yang terdapat pada proses *thresholding*, proses ini dilakukan dengan terlebih dahulu meng-erosi kemudian dilanjutkan dengan mendilasi hasil *thresholding*. Setelah *noise* telah diminimalisi kemudian proses pendeteksian citra akan dilanjutkan ke proses memberikan penanda berupa *Rectangle* dengan memanfaatkan fitur *findContours* yang terdapat pada *OpenCv*, Kemudian melakukan penandaan menggunakan fungsi *rectangle*. Setelah proses penandaan maka selanjutnya adalah mengmotong atau *cropped* hasil dari deteksi titik api yang telah di tandai dengan *rectangle*, proses selanjutnya adalah menghitung jumlah *pixel* biner pada titik api yang terdeteksi. Proses terakhir adalah mendapatkan jumlah *pixel* biner titik api yang terdeteksi.

Contoh perhitungan konversi dari BGR to HSV yang menggunakan rumus yang merujuk pada 2.1. semisal menggunakan nilai  $R = 231$ ,  $G = 236$ ,  $B = 234$ .

$$\begin{aligned} R' &= 231/255 \\ &= 0.0958824 \end{aligned}$$

$$\begin{aligned} G' &= 234/255 \\ &= 0.9254902 \end{aligned}$$

$$\begin{aligned} B' &= 236/255 \\ &= 0.9176471 \end{aligned}$$

$$Cmax = Cmax (R', G', B')$$

$$Cmin = Cmin (R', G', B')$$

$$\begin{aligned} \Delta &= Cmax - Cmin \\ &= 0.09254902 - 0.9176471 \\ &= 0.0196078 \end{aligned}$$

$$H = 60 \times \left\{ \frac{(G' - B')}{\Delta} + 2 \right\} \quad Cmax = R'$$

$$H = 60 \times \left\{ \frac{(0.9254902 - 0.9176471)}{0.0196078} + 2 \right\}$$

$$H = 60 \times \left\{ \frac{(0.0117647)}{0.0196078} + 2 \right\}$$

$$H = 60 \times \{0.60001 + 2\}$$

$$H = 60 \times \{2.600001\}$$

$$H = 156.00006$$

$$H = 156$$

Nilai *Hue* yang didapatkan adalah 156.

$$S = \left\{ \frac{\Delta}{C_{max}} \right\}$$

$$S = \left\{ \frac{0.0196078}{0.9254902} \right\}$$

$$S = 0.0211864$$

Nilai *Saturation* yang didapatkan adalah 0.0211864.

$$V = C_{max}$$

$$V = 0.9254902$$

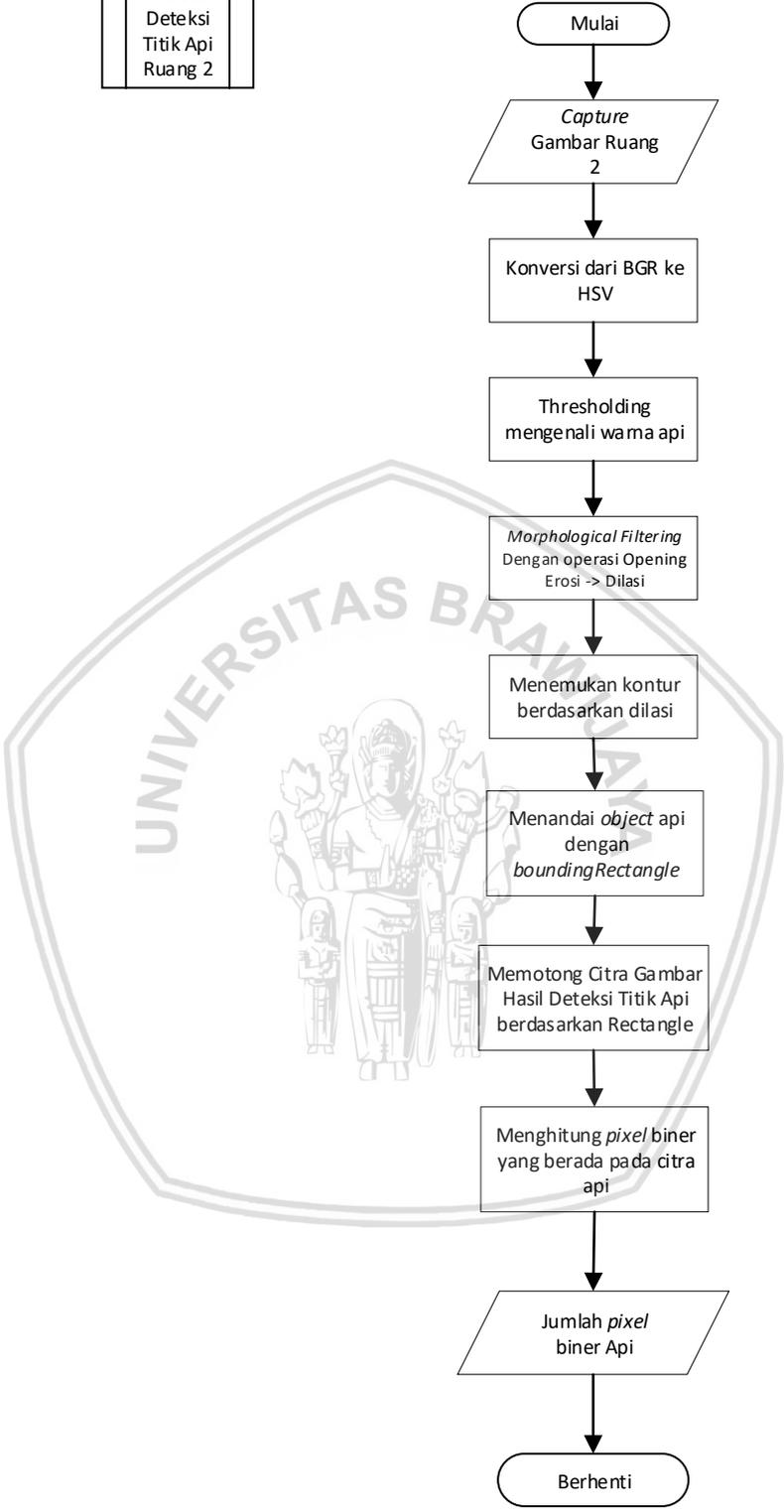
Nilai *Value* yang didapatkan adalah 0.9254902.

### 5.2.3.3 Perancangan Proses Deteksi Titik Api Ruang 2

Proses perancangan penanda menggunakan rectangle ini sebagai penanda terhadap citra api. Berikut merupakan gambar flowchart 5.5 untuk proses deteksi titik api di ruang 2.



Proses Deteksi Titik Api Ruang 2
---



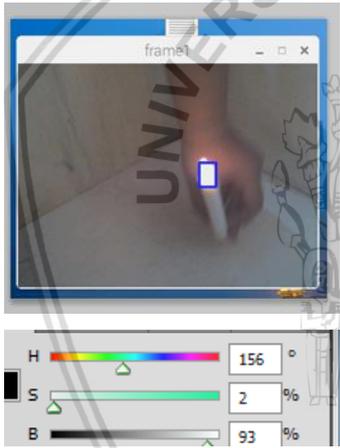
**Gambar 5.5 Flowchart Proses Deteksi Titik Api Ruang 2**

Gambar Flowchart 5.5 proses deteksi titik api ruang 2 yang merupakan pemrosesan citra menggunakan *library opencv 2.4.13* yang telah terintegrasi atau terinstall pada *Raspberry Pi 3 Model B*. pada proses pengambilan mendeteksi titik api yaitu dengan menggunakan kamera yang terintegrasi pada *Raspberry Pi*. Proses

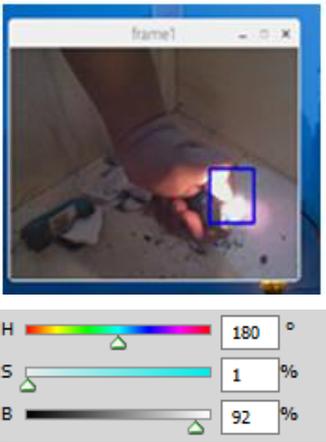
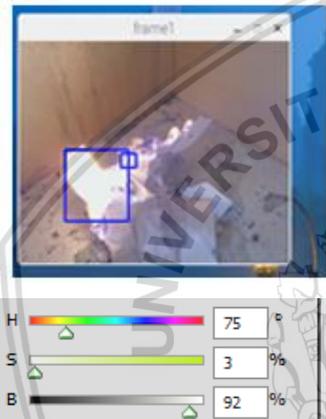


pertama adalah Pada dasarnya pengambilan citra yang dilakukan pada *OpenCV* memiliki format *color space* standar yaitu *BGR (Blue Green Red)*. *Color space* ini kemudian akan dikonversikan ke dalam bentuk *HSV (Hue Saturation Value)*. Pada pendeteksian titik api yang terjadi didalam ruangan, digunakan *color space HSV* karena *HSV* mempunyai warna-warna yang sama dengan apa yang ditangkap dengan indra penglihatan pada manusia. *Color space HSV* memiliki sensitivitas yang baik terhadap kecerahan cahaya, kemudian citra yang didapat dari hasil konversi akan di-*thresholding* untuk diubah menjadi citra biner. Warna yang tidak sesuai dengan warna yang di-*thresholding* maka akan diubah menjadi nilai 0, sedangkan warna sesuai dengan *thresholding* tersebut maka akan bernilai 1. Dalam hal ini akan mempermudah dalam mendeteksi titik api yang terjadi didalam ruangan. Berikut menentukan batas bawah dan batas atas menggunakan *trackbar* pada tabel 5.5 Berikut :

Tabel 5.5 Menentukan nilai range *thresholding* citra api

No	Gambar Citra Api	Hasil Nilai HSV
1.	<p>Nilai HSV pada Objek 1</p> 	<p><b>Nilai</b></p> <p><math>H = 156</math></p> <p><math>S = 2\% \times 255 = 5.1</math></p> <p><math>B/V = 93\% \times 255 = 237.15</math></p>
2.	<p>Nilai HSV pada Objek 2</p> 	<p><b>Nilai</b></p> <p><math>H = 160</math></p> <p><math>S = 3\% \times 255 = 7.65</math></p> <p><math>B/V = 93\% \times 255 = 237.15</math></p>
3.	<p>Nilai HSV pada Objek 3</p>	<p><b>Nilai</b></p> <p><math>H = 180</math></p>



		<p><math>S = 1\% \times 255 = 2.55</math>  <math>B/V = 92\% \times 255 = 234.6</math></p>
<p>4.</p>	<p>Nilai HSV pada Objek 4</p> 	<p><b>Nilai</b>  <math>H = 75</math>  <math>S = 3\% \times 255 = 7.65</math>  <math>B/V = 92\% \times 255 = 234.6</math></p>
<p>5.</p>	<p>Nilai HSV pada Objek 5</p> 	<p><b>Nilai</b>  <math>H = 52</math>  <math>S = 29\% \times 255 = 73.95</math>  <math>B/V = 92\% \times 255 = 234.6</math></p>

Pada tabel 5.4 merupakan hasil untuk menentukan nilai HSV ( $H = Hue$ ,  $S = Saturation$ ,  $B/V = Brightness / Value$ ) dari hasil 5 kali uji coba untuk mengetahui range *thresholding* nilai HSV pada citra api. Maka disimpulkan range citra api yaitu  $H = 0-255$ ,  $S = 0-255$ ,  $B/V = 227-225$ .

Pada proses *thresholding* ini masih terdapat beberapa *noise* kecil yang harus dihilangkan. Maka diperlukan untuk menerapkan operasi *Opening* untuk menghilangkan *noise* yang terdapat pada proses *thresholding*, proses ini dilakukan dengan terlebih dahulu meng-erosi kemudian dilanjutkan dengan mendilasi hasil *thresholding*. Setelah *noise* telah diminimalisi kemudian proses pendeteksian citra akan dilanjutkan ke proses memberikan penanda berupa *Rectangle* dengan memanfaatkan fitur *findContours* yang terdapat pada *OpenCv*, kemudian melakukan penandaan menggunakan fungsi *rectangle*. Setelah proses penandaan maka selanjutnya adalah mengmotong atau *cropped* hasil dari deteksi titik api yang telah di tandai dengan *rectangle*, proses selanjutnya adalah menghitung jumlah *pixel* biner pada titik api yang terdeteksi. Proses terakhir adalah mendapatkan jumlah *pixel* biner titik api yang terdeteksi.

Contoh perhitungan konversi dari BGR to HSV yang menggunakan rumus yang merujuk pada 2.1. semisal menggunakan nilai  $R = 231$ ,  $G = 236$ ,  $B = 234$ .

$$\begin{aligned} R' &= 231/255 \\ &= 0.0958824 \end{aligned}$$

$$\begin{aligned} G' &= 234/255 \\ &= 0.9254902 \end{aligned}$$

$$\begin{aligned} B' &= 236/255 \\ &= 0.9176471 \end{aligned}$$

$$Cmax = Cmax (R', G', B')$$

$$Cmin = Cmin (R', G', B')$$

$$\begin{aligned} \Delta &= Cmax - Cmin \\ &= 0.09254902 - 0.9176471 \\ &= 0.0196078 \end{aligned}$$

$$H = 60 \times \left\{ \frac{(G' - B')}{\Delta} + 2 \right\} \quad Cmax = R'$$

$$H = 60 \times \left\{ \frac{(0.9254902 - 0.9176471)}{0.0196078} + 2 \right\}$$

$$H = 60 \times \left\{ \frac{(0.0117647)}{0.0196078} + 2 \right\}$$

$$H = 60 \times \{0.60001 + 2\}$$

$$H = 60 \times \{2.600001\}$$

$$H = 156.00006$$

$$H = 156$$

Nilai *Hue* yang didapatkan adalah 156.

$$S = \left\{ \frac{\Delta}{C_{max}} \right\}$$

$$S = \left\{ \frac{0.0196078}{0.9254902} \right\}$$

$$S = 0.0211864$$

Nilai *Saturation* yang didapatkan adalah 0.0211864.

$$V = C_{max}$$

$$V = 0.9254902$$

Nilai *Value* yang didapatkan adalah 0.9254902.

## 5.2 Implementasi Sistem

Pada implementasi sistem ini, dilakukan untuk pembuatan sistem dengan merealisasikan dari perancangan sistem yang telah dilakukan sebelumnya pada sub bab 5.1. Bagian implementasi sistem ini, akan dijelaskan dengan rinci mengenai bagaimana implementasi *Prototype* deteksi titik api terpusat menggunakan pengolahan citra digital, perancangan perangkat keras, dan perancangan perangkat lunak

### 5.2.1 Implementasi *Prototype* deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms gateway pada raspberry pi

Pada implementasi ini untuk mendeteksi titik api mengacu pada desain yang telah direncanakan pada sub bab 5.1.1. Dimana *prototype* ini terdiri dari 2 Kamera Logitech C525, Module SIM900A, dan Buzzer. Selain itu juga terdapat *controller* yaitu Raspberry Pi sebagai *controller* yang nantinya akan di intergrasikan menjadi satu didalam Raspberry Pi untuk saling berkomunikasi. Dan pada implementasi ini juga digunakan triplek yang mirip dengan ruangan sebagai *prototype* untuk implementasi sistem ini. Dapat dilihat pada gambar 5.6



**Gambar 5.6 Implementasi prototype alat pendeteksi titik api**

Gambar 5.6 merupakan hasil implementasi perangkat keras yakni dengan menggunakan 2 Kamera *Logitech C525*, *Module SIM900A*, dan *Buzzer*. Nantinya Kamera *Logitech C525* akan di letakkan di masing-masing ruangan. Sedangkan *Module SIM900A* dan *Buzzer* diletakkan dekat dengan *Raspberry Pi* sebagai *controller* pada sistem ini.

#### **5.2.1.1 Implementasi Perangkat Keras Kamera Logitech C525**

Pada implementasi perangkat keras ini merupakan implementasi perangkat keras antar Kamera *Logitech C525* dengan mikrokontroler *Raspberry Pi*. Implementasi ini terdiri dari 2 Kamera *Logitech C525* serta *Raspberry Pi* sebagai mikroprocessor yang mengatur jalannya Kamera tersebut. Implementasi ini dapat dilihat pada gambar 5.7



**Gambar 5.7 Implementasi Perangkat Keras Kamera Logitech C525**

Gambar 5.7 merupakan hasil implementasi dari perangkat keras yakni dengan menggunakan 2 Kamera *Logitech C525* yang terintegrasi dengan *Raspberry Pi* sebagai mikrokontroler. *Raspberry Pi* dan Kamera *Logitech C525* dihubungkan melalui kabel USB yang tersedia pada board *Raspberry Pi*.

### 5.2.1.2 Implementasi Perangkat Keras Module SIM900A

Pada implementasi perangkat keras ini merupakan implementasi perangkat keras antara Module SIM900A dengan mikrokontroler *Raspberry Pi*. Implementasi ini terdiri dari Module SIM900A serta *Raspberry Pi* sebagai mikrokontroler yang mengatur jalannya komunikasi antar 2 board yang berbeda. Implementasi ini dapat dilihat pada gambar 5.8



Gambar 5.8 Implementasi Perangkat Keras Modul SIM900A

Gambar 5.8 adalah hasil implementasi dari perangkat keras ini yakni dengan menggunakan *Module SIM900A* yang terintegrasi dengan *Raspberry Pi*. Dimana implementasi ini menggunakan komunikasi antar 2 board yang berbeda, komunikasi yang digunakan pada implementasi ini adalah Serial (UART) dengan memanfaatkan AT-Commands sebagai bahasa komunikasi antara *Module SIM900A* dengan *Raspberry Pi*.

### 5.2.1.3 Implementasi Perangkat Keras Buzzer

Pada implementasi perangkat keras ini merupakan implementasi perangkat keras antara *Buzzer* dengan mikrokontroler *Raspberry Pi*. Implementasi ini terdiri dari *Buzzer* serta *Raspberry Pi* sebagai mikrokontroler yang mengatur jalannya komunikasi antar 2 board yang berbeda. Implementasi ini dapat dilihat pada gambar 5.9



**Gambar 5.9 Implementasi Perangkat Keras Buzzer**

Gambar 5.9 adalah hasil implementasi dari perangkat keras ini yakni dengan menggunakan *Buzzer* yang terintegrasikan dengan *Raspberry Pi*. Dimana implementasi ini memanfaatkan pin GPIO yang terdapat pada *Raspberry Pi*. Sehingga ketika *Buzzer* aktif akan bernilai 1 (HIGH), namun sebaliknya jika buzzer tidak aktif akan bernilai 0 (LOW).

### 5.2.2 Implementasi Perangkat Lunak

Pada implementasi perangkat lunak dijelaskan realisasi program yang mengacu pada diagram alir yang telah dibuat pada sub bab 5.1.3 implementasi perangkat lunak ini, menulis kode program dengan menggunakan aplikasi *Sublime Text 3* yang kemudian ditambahkan sebuah plugin SFTP Remote, plugin ini memudahkan kita untuk menulis kode program dengan *remote server* pada *Raspberry Pi* melalui protocol SSH. Bahasa yang digunakan dalam implementasi ini merupakan bahasa Python dan memanfaatkan *library OpenCv* yang terinstall pada *Raspberry Pi*. Implementasi ini dimulai dari pembuatan program akuisisi pada citra, *Pre-processing* pada Citra, penanda menggunakan *Rectangle*, dan *Output* sebagai Indikator pada sistem. Berikut merupakan beberapa *library* yang digunakan pada implementasi ini dapat dilihat pada Tabel 5.6

**Tabel 5.6 Kode Program pada Inisialisasi Library Sistem Deteksi Titik api**

Baris	Kode Program
1	<code>import cv2</code>
2	<code>import numpy as np</code>
3	<code>import multiprocessing as mp</code>
4	<code>import time</code>
5	<code>import subprocess as sp</code>
6	<code>import collections</code>
7	<code>import RPi.GPIO as GPIO</code>
8	<code>import serial</code>
9	
10	<code>phone = serial.Serial("/dev/ttyS0", 9600, timeout=5)</code>

Berdasarkan table 5.4 terdapat 9 *library* yang digunakan pada sistem ini. *Library* “*import cv2*” yaitu *library* *OpenCv* untuk mendeteksi titik api pada sistem ini, *library* “*import numpy as np*” merupakan *library* sebagai pengolah angka pada kode program *Python*. Untuk *library* “*import multiprocessing as mp*” merupakan *library* untuk melakukan beberapa proses dalam program *Python* secara *asynchronous*, kemudian terdapat *library* “*import time*” berguna sebagai *delay* atau berguna untuk mengatur *delay* pada program. Kemudian terdapat *library* “*subprocess as sp*” yang berguna untuk memanggil proses lainnya contoh yaitu memanggil *command* pada terminal, *library* “*collections*” merupakan *library* untuk menghitung berapa banyak *pixel* yang terdapat pada *frame* tersebut ketika program dijalankan. *Library* “*import RPi.GPIO as GPIO*” merupakan sebuah *library* yang berguna untuk melakukan pengontrolan terhadap pin-pin *GPIO* yang terdapat pada *board Raspberry Pi*, *library* “*import serial*” merupakan *library* yang berguna melakukan komunikasi secara serial antara *board Raspberry Pi* dengan *board Module SIM900A*. *Library* “*phone = serial.Serial("/dev/ttySO", 9600, timeout=5)*” merupakan *library* untuk melakukan komunikasi serial antara *board Raspberry Pi* dengan *board Module SIM900A* dengan *boudrate* 9600 dan *timeout* sebesar 5ms.

**5.2.2.1 Implementasi Proses Deteksi Titik Api Di Ruang 1**

Setelah melakukan beberapa inisialisasi beberapa *library*, maka akan dilakukan pemrogram untuk dapat menjalankan servis akuisisi pada citra dengan menghidupkan kamera 1 dan kamera 2. Potongan kode programnya dapat dilihat pada Tabel 5.7.

**Tabel 5.7 Implementasi Proses Deteksi Titik Api Di Ruang 1**

Baris	Kode Program
1	<code>def camera01 (args):</code>
2	<code>    cam1 = cv2.VideoCapture(0)</code>
3	
4	<code>    while cam1.isOpened() and not args.cam2stopped:</code>
5	<code>        ret1, frame1 = cam1.read()</code>
6	<code>        frame1 = cv2.resize(frame1, (320,240))</code>
7	
8	<code>        start_time = millis()</code>
9	
10	<code>        frame1hsv = cv2.cvtColor(frame1, cv2.COLOR_BGR2HSV)</code>
11	
12	<code>        thresframe1 = cv2.inRange(frame1hsv, (0,0,227),</code>
13	<code>(255,255,255))</code>
14	
15	<code>        kernel = np.ones((3,3), np.uint8)</code>
16	
17	<code>        erosi1 = cv2.erode(thresframe1, kernel,</code>
18	<code>iterations=5)</code>
19	<code>        dilasi1 = cv2.dilate(erosi1, kernel, iterations=5)</code>
20	
21	<code>        dilasi11 = dilasi1.copy()</code>
22	
23	<code>        _, contours1, _ = cv2.findContours(dilasi11,</code>
24	<code>cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)</code>
25	



Baris	Kode Program
26	temp = 0
27	
28	for cnt in contours1:
29	x,y,w,h = cv2.boundingRect(cnt)
30	cv2.rectangle(frame1, (x,y), (x+w, y+h),
31	(255,0,0), 2)
32	
33	cropped = dilasi1[y:h+y, x:w+x]
34	
35	temp =
36	collections.Counter(np.ravel(cropped))[255]
37	
38	args.cam1 = temp
39	
40	cv2.imshow("frame1", frame1)
41	cv2.imshow("dilasi1", dilasi1)
42	
43	selisih = millis() - start_time
44	args.cam1time = selisih
45	
46	k = cv2.waitKey(30) & 0xff
47	if k==27:
48	break
49	
50	cam1.release()
51	args.cam1stopped = True
52	return

Pada Table 5.7 terdapat inisialisasi variable untuk mengaktifkan kamera. Baris ke-1 adalah fungsi nama atau identitas pada program ini yang nantinya fungsi "camera01" dapat di panggil kembali. Baris ke-2 merupakan proses dimana pengakuisisian data pada citra akan dimulai dengan membuat objek *VideoCapture()* yang dimasukkan ke dalam variable "cam1" dengan argument 0 mengaktifkan kamera 1 pada *Raspberry Pi*. Baris ke-4 merupakan perulangan pada program ini, jika cam1 terbuka dan jika cam2 berhenti, maka akan dibuat variable ret1, fram1 yang nantinya akan diisi nilai dari pembacaan pada kemra. Baris ke-6 merupakan kode program dengan membuat ukuran citra yang diambil dengan 320 x 240 *pixel*. Pada baris ke-8 merupakan fungsi untuk memulai waktu komputasi pada saat pengolahan citra digital berlangsung. Baris ke-10 proses ini dimulai dengan melakukan konversi pada *color space* dari *BGR* ke *HSV*. Konversi ini menggunakan fungsi "cv2.cvtColor" seperti yang terlihat pada tabel 5.5. Baris ke-12 merupakan proses *thresholding* atau *segmentasi* warna api dengan menggunakan fungsi *inRange()* yang nilai tersebut merupakan nilai warna yang segmentasi nantinya nilai tersebut akan dimasukkan kedalam variable "threshframe2". Baris ke-15 merupakan fungsi untuk menghitung atau mengubah kedalam bentuk *integer* 8-bit. Setelah proses *thresholding*, kemudian akan dilakukan proses *Opening* pada baris ke-17 dan ke-19. Operasi *opening* ini merupakan sebuah kombinasi untuk proses dimana suatu citra digital akan dikenai operasi erosi dengan dilanjutkan dengan operasi dilasi, tujuan dari operasi *opening* pada citra ini untuk memperhalus batas-batas pada suatu objek, serta memisahkan objek yang

sebelumnya bergandengan dengan satu dengan yang lainnya, dan menghilangkan objek-objek yang lebih kecil dari ukuran *structuring*.

Selanjutnya pada baris menandai suatu objek dengan menggunakan fungsi *Rectangle()* pada sistem ini. Baris ke-21 merupakan meng-copy nilai variable "dilasi1" kedalam variable "dilasi1". Baris ke-23 merupakan fungsi untuk menemukan *contour* berdasarkan "dilasi1", baris ke-26 adalah variable temp dengan nilai 0 yang nantinya variable ini akan diisi untuk menghitung banyak *pixel* yang terdapat di *frame*. Baris ke-28 sampai ke-36 merupakan perulangan untuk menandai objek dengan *rectangle*, baris ini akan menge-check terlebih dahulu berdasarkan *contours*, maka akan menandai objek pada *frame* dengan *rectangle* dengan fungsi *cv2.boundingRect* berdasarkan nilai panjang dan lebar pada *pixel* tersebut. Kemudian objek tersebut akan di *cropped* berdasarkan panjang dan lebar yang terdapat pada *frame* tersebut. Setelah proses *cropped*, kemudian *variable* temp yang tadi nya bernilai 0 maka akan berisikan nilai banyak *pixel* dengan menggunakan fungsi *collections.Counter()* dengan mengacu pada objek yang di *cropped* tersebut. Baris ke-36 yaitu merupakan hasil dari perhitungan banyaknya *pixel* yang terdapt pada *frame* pada *Kamera 01*.

Pada baris ke-40 yaitu merupakan fungsi *cv.imshow* dimana akan menampilkan *frame1* dengan nama "Frame1", kemudian baris ke-41 juga sama merupakan fungsi *cv.imshow* dimana akan menampilkan dilasi yang telah terjadi pada *frame 1* dengan diberi nama "dilasi1". Baris ke-43 merupakan menghitung waktu komputasi pada pengolahan citra digital ini. Baris ke-46 sampai baris ke-47 merupakan interrupt dengan menggunakan fungsi *cv2.waitKey()*, jika program di-interrupt dengan menggunakan nilai *ascii* 27 atau tombol *esc* maka program akan berhenti dan Baris ke-50 sampai baris ke-51 maka *Kamera 01* secara otomatis akan berhenti dengan otomatis.

### 5.2.2.2 Implementasi Proses Deteksi Titik Api Di Ruang 2

Setelah melakukan proses *pre-processing* pada *Kamera 1* dan *Kamera 2*, selanjutnya akan dilakukan proses penandaan berdasarkan *contour*. Berikut merupakan tabel 5.8

Tabel 5.8 Implementasi Proses Deteksi Titik Api Di Ruang 2

Baris	Kode Program
1	<code>def camera02 (args):</code>
2	
3	<code>    cam2 = cv2.VideoCapture(1)</code>
4	
5	<code>    while cam2.isOpened() and not args.cam1stopped:</code>
6	
7	<code>        ret2, frame2 = cam2.read()</code>
8	<code>        frame2 = cv2.resize(frame2, (320,240))</code>
9	
10	<code>        start_time = millis()</code>
11	
12	<code>        frame2hsv = cv2.cvtColor(frame2, cv2.COLOR_BGR2HSV)</code>
13	



Baris	Kode Program
14	<code>thresframe2 = cv2.inRange(frame2hsv, (0,0,227),</code>
15	<code>(255,255,255))</code>
16	
17	<code>kernel = np.ones((3,3), np.uint8)</code>
18	
19	<code>erosi2 = cv2.erode(thresframe2, kernel,</code>
20	<code>iterations=5)</code>
21	<code>dilasi2 = cv2.dilate(erosi2, kernel, iterations=5)</code>
22	
23	<code>dilasii2 = dilasi2.copy()</code>
24	<code>_, contours2, _ = cv2.findContours(dilasii2,</code>
25	<code>cv2.RETR_LIST, cv2.CHAIN_APPROX_NONE)</code>
26	
27	<code>temp = 0</code>
28	
29	<code>for cnt in contours2:</code>
30	<code>    x,y,w,h = cv2.boundingRect(cnt)</code>
31	<code>    cv2.rectangle(frame2, (x,y), (x+w, y+h),</code>
32	<code>(255,0,0), 2)</code>
33	
34	<code>    cropped = dilasi2[y:h+y, x:w+x]</code>
35	
36	<code>    temp = collections.Counter(np.ravel(cropped))[255]</code>
37	
38	<code>    args.cam2 = temp</code>
39	
40	<code>    cv2.imshow("frame2", frame2)</code>
41	<code>    cv2.imshow("dilasi2", dilasi2)</code>
42	
43	<code>    selisih = millis() - start_time</code>
44	<code>    args.cam2time = selisih</code>
45	
46	<code>    k = cv2.waitKey(30) &amp; 0xff</code>
47	<code>    if k==27:</code>
48	<code>        break</code>
49	
50	<code>cam2.release()</code>
51	<code>args.cam2stopped = True</code>
52	<code>return</code>

Pada Tabel 5.8 terdapat inialisasi variable untuk mengaktifkan kamera. Bari ke-1 adalah fungsi nama atau identitas pada program ini yang nantinya fungsi "camera01" dapat di panggil kembali. Baris ke-3 merupakan proses dimana pengakuisisian data pada citra akan dimulai dengan membuat objek *VideoCapture()* yang dimasukkan ke dalam variable "cam1" dengan argument 0 mengaktifkan kamera 2 pada *Raspberry Pi*. Baris ke-5 merupakan perulangan pada program ini, jika cam2 terbuka dan jika cam2 berhenti, maka akan dibuat variable ret2, fram2 yang nantinya akan diisi nilai dari pembacaan pada kemra. Baris ke-8 merupakan kode program dengan membuat ukuran citra yang diambil dengan 320 x 240 *pixel*. Pada baris ke-10 merupakan fungsi untuk memulai waktu komputasi pada saat pengolahan citra digital berlangsung. Baris ke-12 proses ini dimulai dengan melakukan konversi pada *color space* dari *BGR* ke *HSV*. Konversi ini

menggunakan fungsi “cv2.cvColor” seperti yang terlihat pada tabel 5.6. Baris ke-14 merupakan proses *thresholding* atau *segmentasi* warna api dengan menggunakan fungsi *inRange()* yang nilai tersebut merupakan nilai warna yang segmentasi nantinya nilai tersebut akan dimasukkan kedalam variable “*threshframe2*”. Baris ke-17 merupakan fungsi untuk menghitung atau mengubah kedalam bentuk *integer* 8-bit. Setelah proses *thresholding*, kemudian akan dilakukan proses *Opening* pada baris ke-19 dan ke-21. Operasi *opening* ini merupakan sebuah kombinasi untuk proses dimana suatu citra digital akan dikenai operasi erosi dengan dilanjutkan dengan operasi dilasi, tujuan dari operasi *opening* pada citra ini untuk memperhalus batas-batas pada suatu objek, serta memisahkan objek yang sebelumnya bergandengan dengan satu dengan yang lainnya, dan menghilangkan objek-objek yang lebih kecil dari ukuran *structuring*.

Selanjutnya pada baris menandai suatu objek dengan menggunakan fungsi *Rectangle()* pada sistem ini. Baris ke-23 merupakan meng-copy nilai variable “dilasi1” kedalam variable “dilasi11”. Baris ke-24 merupakan fungsi untuk menemukan *contour* berdasarkan “dilasi11”, baris ke-26 adalah *variable* temp dengan nilai 0 yang nantinya variable ini akan diisi untuk menghitung banyak *pixel* yang terdapat di *frame*. Baris ke-29 sampai ke-38 merupakan perulangan untuk menandai objek dengan *rectangle*, baris ini akan menge-check terlebih dahulu berdasarkan *contours*, maka akan menandai objek pada *frame* dengan *rectangle* dengan fungsi *cv2.boundingRect* berdasarkan nilai panjang dan lebar pada *pixel* tersebut. Kemudian objek tersebut akan di *cropped* berdasarkan panjang dan lebar yang terdapat pada *frame* tersebut. Setelah proses *cropped*, kemudian *variable* temp yang tadi nya bernilai 0 maka akan berisikan nilai banyak *pixel* dengan menggunakan fungsi *collections.Counter()* dengan mengacu pada objek yang di *cropped* tersebut. Baris ke-36 yaitu merupakan hasil dari perhitungan banyaknya *pixel* yang terdapt pada *frame* pada Kamera 02.

Pada baris ke-40 yaitu merupakan fungsi *cv.imshow* dimana akan menampilkan *frame1* dengan nama “Frame2”, kemudian baris ke-41 juga sama merupakan fungsi *cv.imshow* dimana akan menampilkan dilasi yang telah terjadi pada *frame 2* dengan diberi nama “dilasi1”. Baris ke-43 merupakan menghitung waktu komputasi pada pengolahan citra digital ini. Baris ke-46 sampai baris ke-47 merupakan interrupt dengan menggunakan fungsi *cv2.waitKey()*, jika program di-interrupt dengan menggunakan nilai *ascii 27* atau tombol *esc* maka program akan berhenti dan Baris ke-50 sampai baris ke-51 maka Kamera 01 secara otomatis akan berhenti dengan otomatis.

Setelah melakukan pendeteksian dengan menandai objek pada Kamera 01 dan Kamera 02, selanjutnya akan dilakukan proses untuk memberikan peringatan *alarm* dan mengirimkan layanan pesan singkat (sms). Berikut potongan kode program untuk output sebagai indikator pada tabel 5.9

**Tabel 5.9 Impementasi Indikator Output**

Baris	Kode Program
1	<code>def output (args) :</code>



Baris	Kode Program
2	
3	while not args.cam1stopped or not args.cam2stopped:
4	
5	message = ""
6	
7	if args.cam1 > 10:
8	message += "1"
9	
10	if args.cam2 > 10:
11	if message != "":
12	message += " dan "
13	message += "2"
14	
15	if args.cam1 > 10 or args.cam2 > 10:
16	if message != last_message:
17	print ("Ruang "+message+" terdeteksi api")
18	
19	phone.write("AT+CMGF=1\r\n")
20	print (phone.read(512))
21	time.sleep(0.5)
22	
23	phone.write("AT+CMGS=\""+6282233165190+"\r\n")
24	print (phone.read(512))
25	time.sleep(0.5)
26	phone.write("Ruang "+message+" terdeteksi
27	api")
28	time.sleep(0.1)
29	phone.write(chr(26))
30	print (phone.read(512))
31	time.sleep(0.5)
32	
33	GPIO.output(11, GPIO.HIGH)
34	last_message = message
35	else :
36	GPIO.output(11, GPIO.LOW)
37	last_message = ""
38	
39	time.sleep(6)
40	
41	return

Tabel 5.9 implementasi output sebagai indikator, dimana setelah mendapatkan hasil atau setelah objek titik api maka proses selanjutnya adalah memberikan outputan atau memberi peringatan terhadap *user* dengan memberikan *notification* penanda berupa *alarm* dan memberikan *notification* juga terhadap operator berupa layanan pesan singkat (via sms). Pada tabel di atas merupakan potongan kode program, baris ke-1 adalah fungsi dengan nama *output()*. Baris ke-3 merupakan perulangan pada program, jika *Kamera1* dan *Kamera 2* tidak berhenti atau tetap jalan, maka akan menampilkan pesan "". Baris ke-7 sampai ke-13, jika jumlah *pixel* yang terdeteksi pada objek api pada "*cam1*" lebih besar 10 maka akan mengirimkan pesan "Ruang 1 terdeteksi api", jika jumlah *pixel* yang terdeteksi pada objek api pada "*cam2*" lebih besar dari pada 10 maka akan mengirimkan pesan "Ruang 2 terdeteksi api", namun jika "*cam1*" dan "*cam2*"

memiliki jumlah pixel lebih dari pada 10 maka akan mengirimkan pesan “Ruang 1 dan 2 terdeteksi api”. Pada baris ke-15 sampai baris ke-34 merupakan kode program untuk memberikan *notification* berupa layanan pesan singkat (sms) dan *notification* berupa *alarm*, pada baris ke-15 sampai ke-17 jika jumlah *pixel* yang terdeteksi pada objek api pada “cam1” dan “cam2” lebih dari pada 10 maka akan mengirimkan pesan ke operator “Ruang 1 terdeteksi api”, “Ruang 2 terdeteksi api”, dan “Ruang 1 dan 2 terdeteksi api”.

Kemudian pada baris ke-19 sampai baris ke-21 yaitu *Raspberry Pi* akan berkomunikasi dengan *Module SIM900A* dengan mengirimkan pesan OK/ERROR dengan perintah menggunakan AT+Commands “AT+CMGF=1\r\n” ketika *Raspberry Pi* berkomunikasi dengan *Module SIM900A*, jika *Module SIM900A* mengirimkan pesan OK maka komunikasi antar *board Raspberry Pi* dengan *Module SIM900A* berhasil berkomunikasi, kemudian akan mendelay sebanyak 0.5s untuk melanjutkan perintah selanjutnya. Baris ke-23 sampai baris ke-31 dini pada program ini *board Raspberry Pi* akan menyuruh *Module SIM900A* untuk menulis dan sekaligus mengirim layanan pesan singkat ke nomor yang dituju, disini menggunakan nomor +6282233165190, yang isi pesan singkat tersebut adalah “Ruang 1 terdeteksi api”, “Ruang 2 terdeteksi api”, dan “Ruang 1 dan 2 terdeteksi api”. Serta pada baris ke-33 sampai ke-34 yaitu akan mengaktifkan *Buzzer* yang terhubung melalui pin GPIO 11, *Buzzer* akan aktif atau bernilai 1 (HIGH). Jika tidak terdeteksi titik api maka tidak akan melakukan *action* apapun, atau *Module SIM900A* tidak mengirimkan pesan terhadap nomor yang dituju serta *Buzzer* akan tidak aktif atau bernilai 0 (LOW).

Tabel 5.10 dibawah ini merupakan potongan program untuk melakukan proses pengenalan *variable global* ataupun melakukan *multiprocessing* secara *asynchronous* pada sistem ini.

**Tabel 5.10 Proses pengenalan variable golbal**

Baris	Kode Program
1	if __name__ == "__main__":
2	
3	Global = mp.Manager().Namespace()
4	Global.cam1 = 0
5	Global.cam2 = 0
6	Global.cam1stopped = False
7	Global.cam2stopped = False
8	
9	sp.call(['lsusb'])
10	
11	GPIO.setmode(GPIO.BOARD)
12	GPIO.setup(11, GPIO.OUT)
13	
14	pool= mp.Pool(processes=3)
15	pool.apply_async(camera01, (Global, ))
16	pool.apply_async(camera02, (Global,))
17	pool.apply_async(output, (Global,))
18	pool.close()
19	pool.join()
20	



Baris	Kode Program
21	<code>cv2.destroyAllWindows()</code>
22	
23	<code>GPIO.output(11, GPIO.LOW)</code>
24	<code>GPIO.cleanup()</code>

Pada tabel 5.10 diatas merupakan potongan program yaitu untuk melakukan proses inialisai, *multiprocessing* secara *asynchronous*, dan untuk menutup semua program yang telah dilakukan. Baris ke-1 adalah jika nama sama adengan main maka akan menjalankan baris ke-3 sampai baris ke-24. Baris ke-3 sampai ke-7 merupakan fungsi untuk pemanggilan fungsi secara global, dimana menggunakan *mp.Manager().Namespace()* yaitu berguna utnuk memanage semua aktivitas yang berkaitan dengan nama *cam1* dan *cam2*. Baris ke-4 dan baris ke-5 yaitu variable global dengan variable *cam1* ketika memiliki nilai 0 maka *cam1* akan berjalan, jika variable global dengan variable *cam2* ketika memiliki nilai 0 maka *cam2* akan berjalan. Baris ke-6 sampai ke-7 yaitu merupakan jika *variable* global dengan *variable cam1* dan *cam2* ketika memiliki nilai *false* maka *cam 1* dan *cam2* tidak akan berjalan

Baris ke-9 yaitu merupakan fungsi untuk memanggil apa saja yang terintegrasi dengan *board Raspberry Pi*. Baris ke-11 samppai ke-12 merupakan fungsi untuk mengeset pin GPIO pada *board Raspberry Pi* yang terintegrasi dengan *Buzzer*. Baris ke-14 sampai ke-19 merupakan proses melakukan *multiprocessing* dengan terdapat 3 proses sekaligus dalam sistem ini, namun setiap proses memiliki sendiri proses jadi tidak ada proses yang saling menginterrupt antar proses 1 dengan yang lainnya. Baris ke-21 yaitu fungsi pada *library OpenCV* “*cv2.destroyAllWindows*” dimana untuk menghapus tampilan yang muncul pada monitor atau menghentikan seluruh program yang sedang berjalan. Sedangkan baris ke-23 merupakan konfigurasi pin GPIO pada *board Raspberry Pi* dengan *Buzzer*, pada baris ke-24 yaitu merupakan fungsi *cleanup* atau menghentikan pin GPIO pada *board Raspberry Pi* ketika pin GPIO tidak digunakan dan tidak terhubung.

## BAB VI

### PENGUJIAN DAN ANALISIS HASIL

Bab ini akan dibahas mengenai proses pengujian dan menganalisis hasil berkaitan dengan pengujian. Pengujian dilakukan agar mengetahui hasil dari implementasi yang dimana nantinya sesuai yang diharapkan oleh sistem. Dalam pengujian, pertama akan dilakukan pengujian yang akan diuji adalah fungsional dari sistem ini. Pada pengujian fungsional tersebut merupakan pengujian fungsi dari sistem, yang diharapkan adalah apakah sistem mampu bekerja dengan maksimal dan sesuai dengan spesifikasinya, proses pengujian tersebut meliputi pengujian terhadap keakurasian dalam mendeteksi api berdasarkan objek yang telah ditentukan, pengujian keakurasian pembacaan antara kamera pada setiap ruangan, pengujian terhadap hardware yang terintegrasi dengan *Raspberry Pi*, dan pengujian terakhir yaitu berapa lama waktu komputasi yang dibutuhkan sistem ini.

#### 6.1 Pengujian Keakurasian dalam Mendeteksi Titik Api Berdasarkan Objek

Pengujian keakurasian dalam mendeteksi api berdasarkan objek sangat penting, dikarenakan proses ini akan berperan penting dalam menentukan hasil dari *output* yang akan dikeluarkan oleh sistem ini. Pada pengujian ini dilakukan dengan melakukan pembacaan terhadap objek, objek itu sendiri terdiri dari 10 objek yang berpotensi menghasilkan titik api ketika terbakar pada setiap titik kamera yang terdapat pada setiap ruangan.

##### 6.1.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk mengetahui berapa tingkat keakurasian dari sistem ini dalam mendeteksi titik api yang terdapat pada objek yang telah ditentukan dengan menggunakan 10 objek berpotensi menghasilkan titik api ketika terbakar. Kemudian hasil dari keakurasian tersebut dihitung untuk mendapatkan nilai tingkat keakurasian pada sistem ini.

##### 6.1.2. Prosedur Pengujian

Dibawah merupakan prosedur yang akan digunakan untuk pengujian Keakurasian dalam Mendeteksi Titik Api Berdasarkan Objek :

1. Menghubungkan *power supply* dengan saluran listrik yang terhubung dengan *Raspberry Pi*.
2. Menyalakan *Raspberry Pi* yang telah terintegrasi dengan hardware Kamera *Logitech C525*, *Buzzer* dan *Modul SIM900A*.
3. Menghubungkan *Raspberry Pi* dengan monitor dengan kabel HDMI.
4. Menjalankan program deteksi titik api menggunakan *multiKamera* berbasis sms gateway pada *raspberry pi* pada terminal yang terdapat pada *raspberry pi*.

5. Mendeteksi dengan objek yang telah ditentukan, objek yang telah ditentukan yaitu 10 objek yang berpotensi menghasilkan titik api ketika terbakar.
6. Memberikan objek titik api agar nantinya kamera dapat membaca ketika objek titik api telah terdeteksi.
7. Hasil dari pengujian ini yaitu hasil keakurasian dalam mendeteksi titik api berdasarkan objek yang telah ditentukan, dengan melalui monitor yang terintegrasi dengan *raspberry pi* melalui kabel HDMI yang terpasang pada *raspberry pi*.
8. Menghitung hasil keakurasian dengan menggunakan rumus yang ditunjukkan pada **Persamaan (6.1)**

$$Akurasi = \frac{(a + d) - c}{a + b + c + d} \times 100\% \quad (6.1)$$

Keterangan :

$a$  = *True Positive* (Objek Titik Api, Sistem Mendeteksi Titik Api)

$b$  = *True Negative* (Objek Titik Api, Sistem Tidak Mendeteksi Titik Api)

$c$  = *False Positive* (Bukan Objek Titik Api, Sistem Mendeteksi Titik Api)

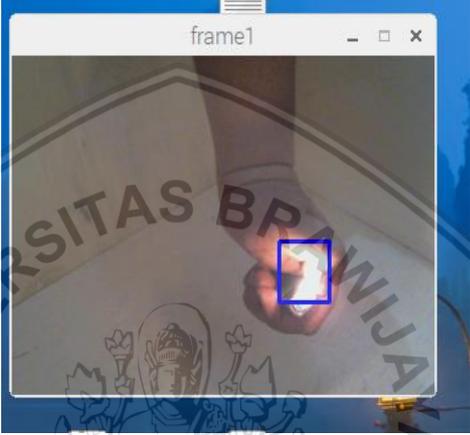
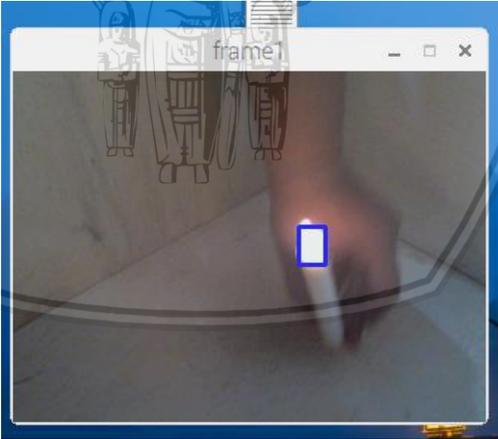
$d$  = *False Negative* (Bukan Objek Titik Api, Sistem Tidak Mendeteksi Titik Api)

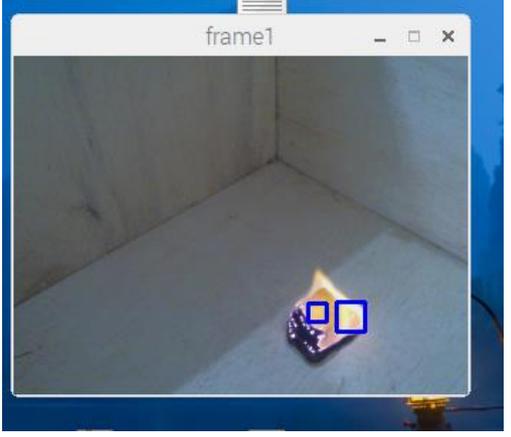
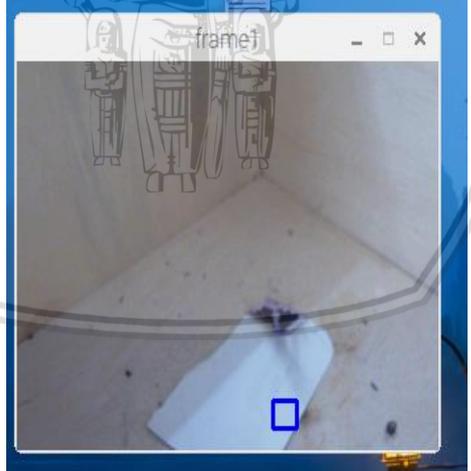
### 6.1.3 Pelaksanaan Pengujian

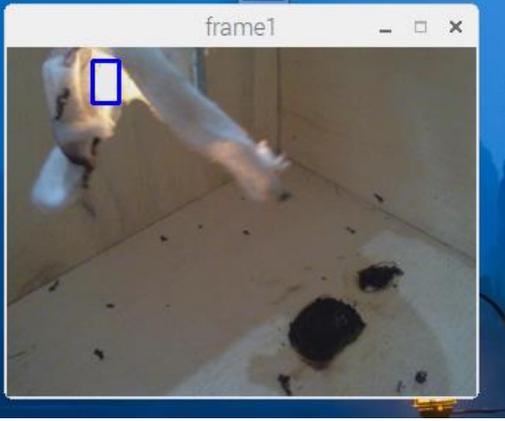
Pengujian ini dilakukan berdasarkan prosedur yang telah dijelaskan pada subbab sebelumnya. Pengujian ini untuk keakurasian mendeteksi titik api berdasarkan objek yang telah ditentukan yaitu 10 objek yang berpotensi menghasilkan titik api ketika terbakar dengan sekenarion kondisi *True Positive*, *Ture Negative*, *False Positive* dan *False Negative*.

### 6.1.4 Hasil dan Analisis Pengujian

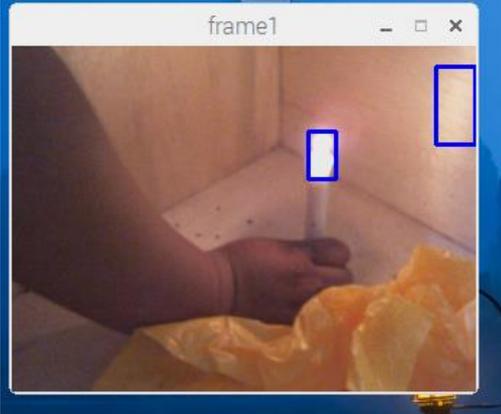
**Tabel 6.1 Data uji dan Hasil pengujian Mendeteksi Titik Api berdasarkan Objek yang telah ditentukan**

No	Objek yang berpotensi menghasilkan titik api ketika terbakar	Hasil deteksi	Skenario Kondisi
1.	Korek Api		<i>True Positive</i>
2.	Lilin		<i>True Positive</i>

3.	Kapas		<i>True Positive</i>
4.	Tisu		<i>True Positive</i>
5.	Kertas		<i>False Negative</i>

6.	Kain		<i>True Positive</i>
7.	Warna Mirip Api (Kuning)		<i>False Negative</i>
8.	Warna Mirip Api (Merah)		<i>False Negative</i>



9.	Warna Mirip Api (Kuning) dengan api		True Positive
10.	Warna Mirip Api (Merah) dengan api		True Positive

Berdasarkan hasil pengujian pada tabel 6.1 diatas, dapat dilihat pengujian dilakukan dengan menggunakan 10 objek yang berpotensi menyebabkan titik api ketika terbakar. Hasil pengujian tersebut akan ditampilkan ke dalam layar monitor, didapatkan hasil bahwa data yang ditampilkan ke dalam monitor dapat ditampilkan sesuai dengan harapan. Dari 10 objek yang telah dideteksi sebagai berikut :

1. Objek Titik Api, Sistem Mendeteksi Titik Api (*True Positive*) terdapat 7 objek yang sesuai dengan deteksi yaitu sistem mendeteksi titik api dengan sekenario ketika objek api terdeteksi maka sistem mendeteksi bahwa objek tersebut adalah titik api.
2. Bukan Objek Titik Api, Namun Sistem Mendeteksi Titik Api (*False Positive*) terdapat 1 objek yang tidak sesuai dengan deteksi titik api yaitu sistem mendeteksi titik api yang seharusnya objek tersebut bukanlah titik api yang harusnya dideteksi.
3. Bukan Objek Titik Api, Namun Sistem Tidak Mendeteksi Titik Api (*False Negative*) terdapat 2 objek bukan merupakan titik api dan sistem tidak mendeteksi objek tersebut sebagai titik api.

$$Akurasi = \frac{(a + d + c + d) - c}{a + b + c + d} \times 100\%$$

$$\begin{aligned} &= \frac{(7 + 2 + 0 + 1) - 1}{a + b + c + d} \times 100\% \\ &= \frac{10 - 1}{10} \times 100\% \\ &= \frac{9}{10} \times 100\% = 90\% \end{aligned}$$

## 6.2 Pengujian Keakurasi Pembacaan Antara Kamera pada setiap ruangan

Pengujian keakurasian pembacaan antara kamera pada setiap ruangan dalam mendeteksi api sangat penting, dikarenakan proses ini akan berperan penting dalam menentukan hasil dari *output* yang akan dikeluarkan oleh sistem ini. Pada pengujian ini dilakukan dengan melakukan pembacaan terhadap objek, objek itu sendiri terdiri dari 10 objek yang berpotensi menghasilkan titik api ketika terbakar pada setiap titik kamera yang terdapat pada setiap ruangan. pada setiap kamera yang terdapat pada setiap ruangan.

### 6.2.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk mengetahui berapa tingkat keakurasian dari sistem ini dalam pembacaan objek antar kamera yang terdapat pada setiap ruangan dalam mendeteksi titik api. Kemudian hasil dari keakurasian tersebut dihitung untuk mendapatkan nilai tingkat keakurasian pembacaan objek titik api oleh kamera yang terdapat pada setiap ruangan pada sistem ini.

### 6.2.2 Prosedur Pengujian

Dibawah merupakan prosedur yang akan digunakan untuk pengujian keakurasian pembacaan antara kamera pada setiap ruangan.

1. Menghubungkan *power supply* dengan saluran listrik yang terhubung dengan *Raspberry Pi*.
2. Menyalakan *Raspberry Pi* yang telah terintegrasi dengan hardware Kamera *Logitech C525*.
3. Menghubungkan *Raspberry Pi* dengan monitor dengan kabel menggunakan HDMI.
4. Menjalankan program deteksi titik api menggunakan Kamera berbasis sms *gateway* pada *raspberry pi* pada terminal yang terdapat pada *raspberry pi*.
5. Memberikan objek yang berpotensi menghasilkan titik api ketika terbakar agar nantinya kamera dapat membaca ketika objek titik api telah terdeteksi.
6. Mendeteksi titik api dengan objek yang terindikasi untuk terjadinya titik api yang telah ditentukan untuk pembacaan setiap kamera yang terdapat pada setiap ruangan untuk.

7. Hasil dari pengujian ini yaitu hasil pembacaan antara kamera pada setiap ruangan untuk deteksi titik api berdasarkan objek yang telah ditentukan melalui monitor yang terintegrasi dengan *raspberry pi* melalui kabel HDMI yang terpasang pada *raspberry pi*.
8. Menghitung hasil keakurasian dengan menggunakan rumus yang ditunjukkan pada **Persamaan (6.2)**

$$Akurasi = \frac{(a + d + c + d) - c}{a + b + c + d} \times 100\% \quad (6.2)$$

Keterangan :

- a* = *True Positive* (Objek Titik Api, Sistem Mendeteksi Titik Api)
- b* = *True Negative* (Objek Titik Api, Sistem Tidak Mendeteksi Titik Api)
- c* = *False Positive* (Bukan Objek Titik Api, Sistem Mendeteksi Titik Api)
- d* = *False Negative* (Bukan Objek Titik Api, Sistem Tidak Mendeteksi Titik Api)

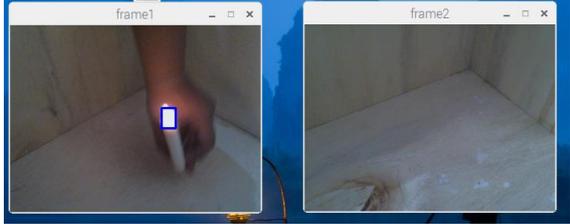
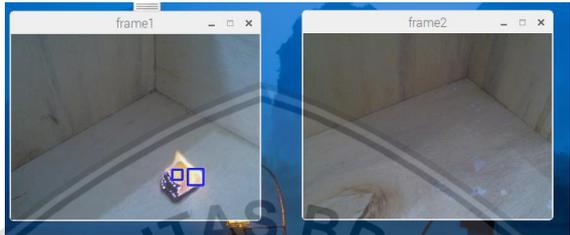
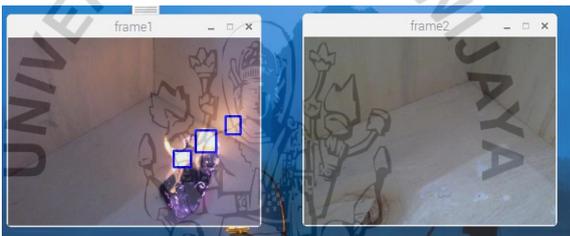
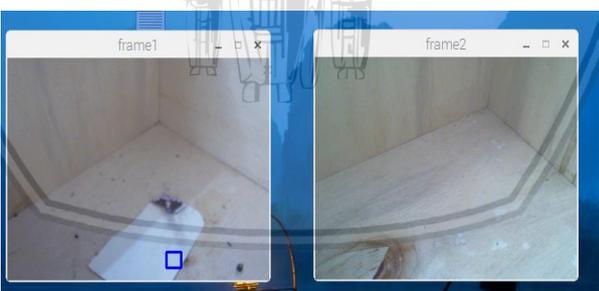
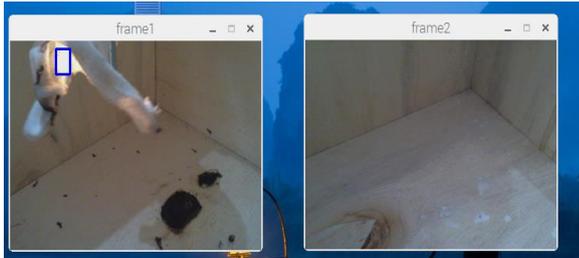
### 6.2.3 Pelaksanaan Pengujian

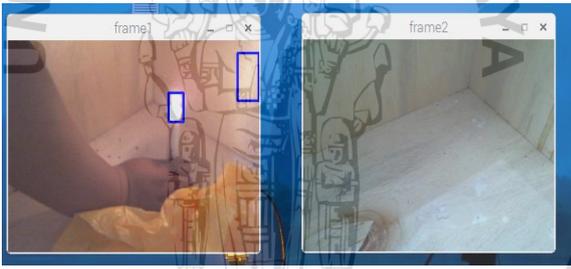
Pengujian ini dilakukan berdasarkan prosedur yang telah dijelaskan pada subbab sebelumnya. Pengujian ini untuk keakurasian pembacaan kamera pada setiap ruangan untuk mendeteksi objek yang berpotensi menghasilkan titik api ketika terbakar dengan berdasarkan objek yang telah ditentukan dengan sekenario kondisi *True Positive*, *True Negative*, *False Positive*, dan *False Negative*.

### 6.2.4 Hasil dan Analisis Pengujian

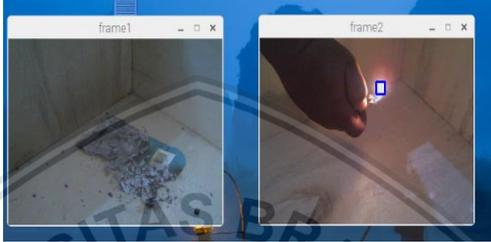
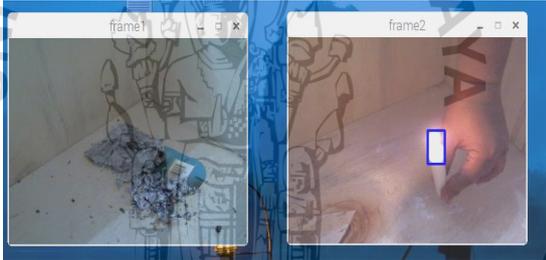
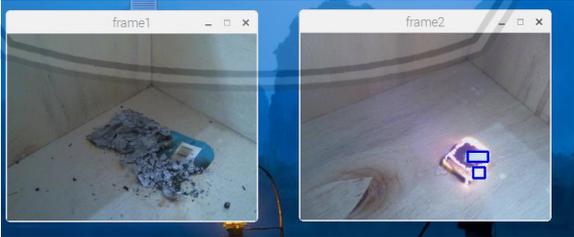
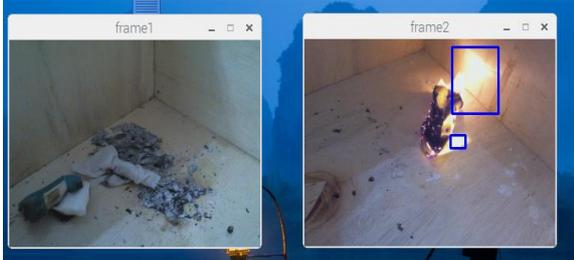
**Tabel 6.2 Keakurasi Pembacaan Antara Kamera pada ruangan 1**

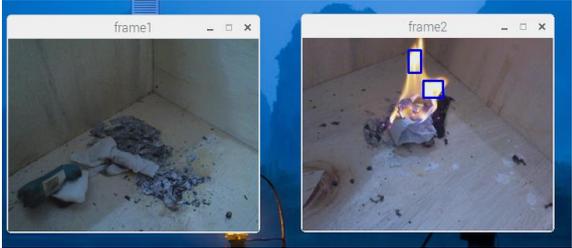
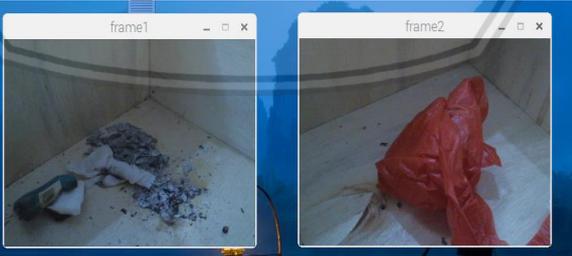
No	Objek yang berpotensi menghasilkan titik api ketika terbakar	Hasil deteksi pada ruang 1	Skenario Kondisi
1.	Korek Api		<i>True Positive</i>
2.	Lilin		<i>True Positive</i>

			
3.	Kapas		<i>True Positive</i>
4.	Tisu		<i>True Positive</i>
5.	Kertas		<i>False Positive</i>
6.	Kain		<i>True Positive</i>

<p>7.</p>	<p>Warna Mirip Api (Kuning)</p>		<p><i>False Negative</i></p>
<p>8.</p>	<p>Warna Mirip Api (Merah)</p>		<p><i>False Negative</i></p>
<p>9.</p>	<p>Warna Mirip Api (Kuning) dengan api</p>		<p><i>True Positive</i></p>
<p>10.</p>	<p>Warna Mirip Api (Merah) dengan api</p>		<p><i>True Positive</i></p>

**Tabel 6.3 Keakurasi Pembacaan Antara Kamera pada ruangan 2**

No	Objek yang berpotensi menghasilkan titik api ketika terbakar	Hasil deteksi pada ruang 2	Skenario Kondisi
1.	Korek Api		<i>True Positive</i>
2.	Lilin		<i>True Positive</i>
3.	Kapas		<i>True Positive</i>
4.	Tisu		<i>True Positive</i>

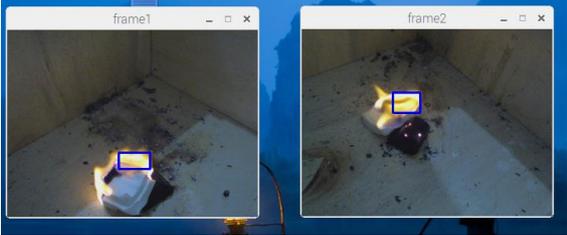
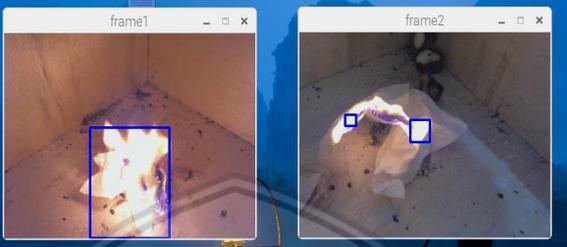
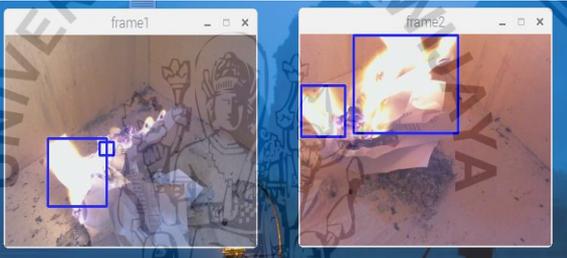
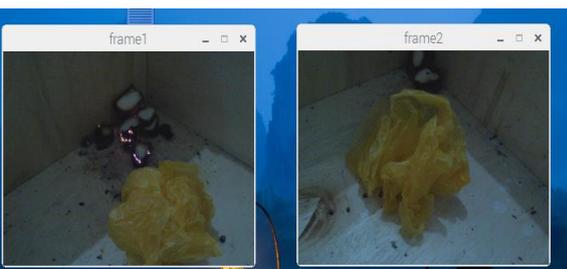
5.	Kertas		<i>True Positive</i>
6.	Kain		<i>True Positive</i>
7.	Warna Mirip Api (Kuning)		<i>False Negative</i>
8.	Warna Mirip Api (Merah)		<i>False Negative</i>
9.	Warna Mirip Api (Kuning) dengan api		<i>True Positive</i>



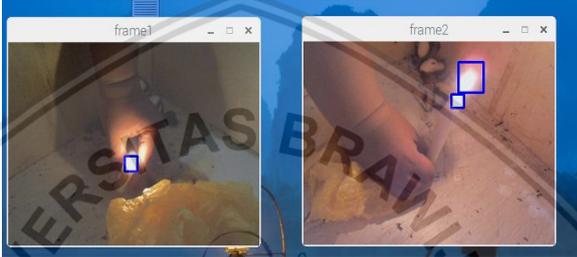
10.	Warna Mirip Api (Merah) dengan api		<i>True Positive</i>

**Tabel 6.4 Keakurasi Pembacaan Antara Kamera pada ruangan 1 dan ruangan 2**

No	Objek yang berpotensi menghasilkan titik api ketika terbakar	Hasil deteksi pada ruang 1 & 2	Skenario Kondisi
1.	Korek Api		<i>True Positive</i>
2.	Lilin		<i>True Positive</i>
3.	Kapas		<i>True Positive</i>

			
4.	Tisu		<i>True Positive</i>
5.	Kertas		<i>True Positive</i>
6.	Kain		<i>True Positive</i>
7.	Warna Mirip Api (Kuning)		<i>False Negative</i>



8.	Warna Mirip Api (Merah)		<i>False Negative</i>
9.	Warna Mirip Api (Kuning) dengan api		<i>True Positive</i>
10.	Warna Mirip Api (Merah) dengan api		<i>True Positive</i>

Berdasarkan tabel 6.2, tabel 6.3, dan tabel 6.3 diatas merupakan hasil dari pengujian dengan mendeteksi objek yang terindikasi titik api, dapat dilihat pada tabel diatas yaitu menggunakan jumlah pengujian sebanyak 30 data uji pada setiap kamera yang terdapat di ruangan

1. Objek Titik Api, Sistem Mendeteksi Titik Api (*True Positive*) terdapat 23 objek yang sesuai dengan deteksi yaitu sistem mendeteksi titik api dengan skenario ketika objek api terdeteksi maka sistem mendeteksi bahwa objek tersebut adalah titik api pada setiap ruangan yaitu ruang 1, ruang 2 , dan ruang 1&2.
2. Bukan Objek Titik Api, Namun Sistem Mendeteksi Titik Api (*False Positive*) terdapat 1 objek yang tidak sesuai dengan deteksi titik api yaitu pada ruang

- 1, sistem mendeteksi titik api yang seharusnya objek tersebut bukanlah titik api yang harusnya dideteksi.
3. Bukan Objek Titik Api, Namun Sistem Tidak Mendeteksi Titik Api (*False Negative*) terdapat 6 objek bukan merupakan titik api dan sistem tidak mendeteksi objek tersebut sebagai titik api pada masing-masing ruangan yaitu ruang 1, ruang 2, dan ruang 1&2.

Berdasarkan perhitungan pada **Persamaan (6.2)** yaitu sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{(a + b + c + d) - c}{a + b + c + d} \times 100\% \\
 &= \frac{(23 + 0 + 1 + 6) - 1}{30} \times 100\% \\
 &= \frac{30 - 1}{30} \times 100\% \\
 &= \frac{29}{30} \times 100\% = 96,66\%
 \end{aligned}$$

Sehingga didapatkan nilai akurasi sebesar 96,66%.

### 6.3 Pengujian Buzzer dan Modul SIM900A

*Buzzer dan Modul SIM900A* merupakan komponen yang berperan penting sebagai *output* dari sistem. *Buzzer dan Modul SIM900A* aktif berdasarkan perintah yang dikirimkan dari *Raspberry Pi* yang terintegrasi dengan *Buzzer* dan pada *Modul SIM900A* yang melalui komunikasi serial.

#### 6.3.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk mengetahui apakah hasil deteksi titik api berdasarkan objek yang terindikasi titik api sesuai dengan sistem yang dihasilkan sebagai *ouput* yang telah dirancang dan dihasilkan

#### 6.3.2. Prosedur Pengujian

Dibawah merupakan prosedur yang akan digunakan untuk pengujian Keakurasian dalam Mendeteksi Titik Api Berdasarkan Objek :

1. Menghubungkan *power supply* dengan saluran listrik yang terhubung dengan *Raspberry Pi*.
2. Menyalakan *Raspberry Pi* yang telah terintegrasi dengan hardware *Buzzer* dan *Modul SIM900A*.
3. Menghubungkan *Raspberry Pi* dengan monitor dengan kabel HDMI.
4. Menjalankan program deteksi titik api menggunakan *multiKamera* berbasis sms gateway pada *raspberry pi* pada terminal yang terdapat pada *raspberry pi*.
5. Mendeteksi dengan objek yang telah ditentukan, objek yang telah ditentukan yaitu 10 objek yang berpotensi menghasilkan titik api ketika terbakar.

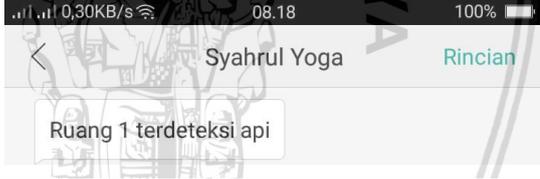
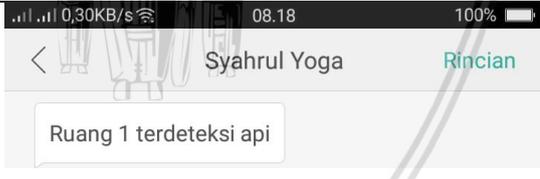
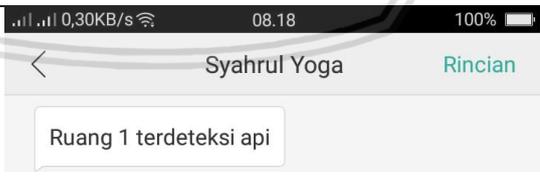
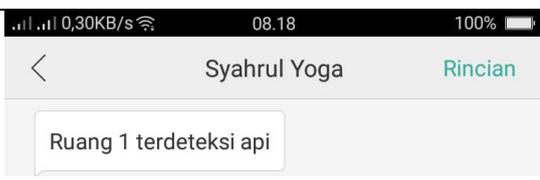
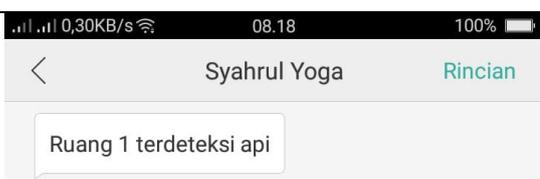
6. Memberikan objek titik api agar nantinya kamera dapat membaca ketika objek titik api telah terdeteksi.
7. Hasil dari pengujian ini yaitu apakah *Buzzer* dan *Modul SIM900A* yang terintegrasi dengan *Rapsberry Pi* apakah memberikan *output* yang sesuai dengan sistem yang sudah ditentukan oleh sistem yaitu ketika mendeteksi objek yang terindikasi titik api

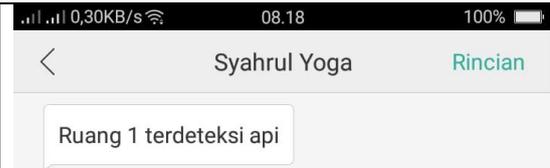
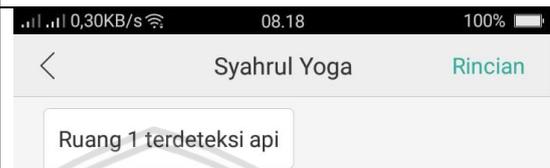
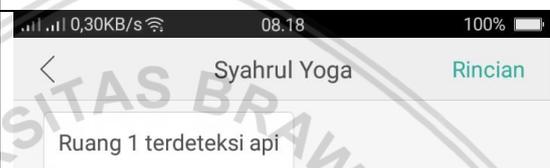
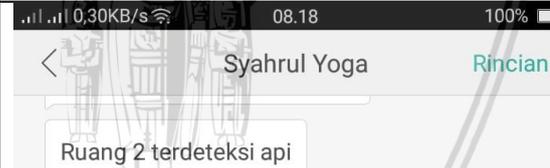
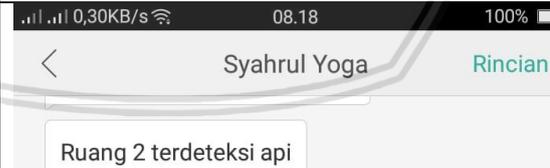
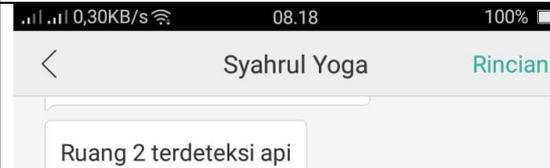
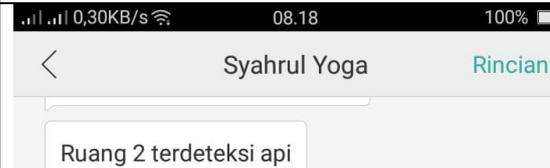
### 6.1.3 Pelaksanaan Pengujian

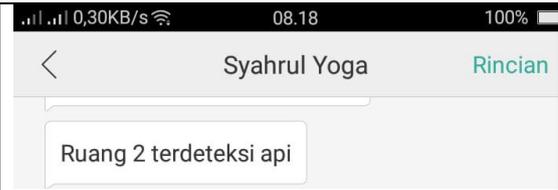
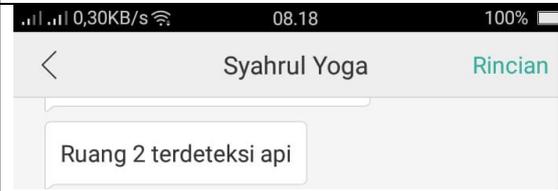
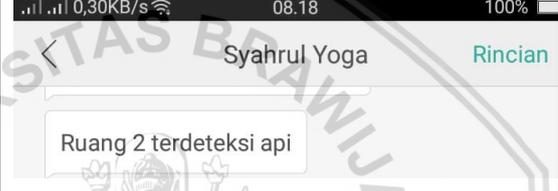
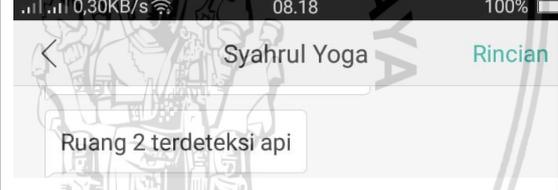
Pengujian ini dilakukan berdasarkan prosedur yang telah dijelaskan pada subbab sebelumnya. Pengujian ini untuk mengetahui hasil *ouput* ketika mendeteksi titik api berdasarkan objek yang telah ditentukan yaitu objek yang berpotensi menghasilkan titik api ketika terbakar.

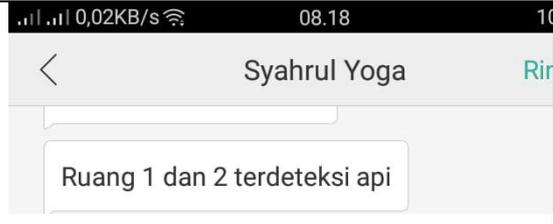
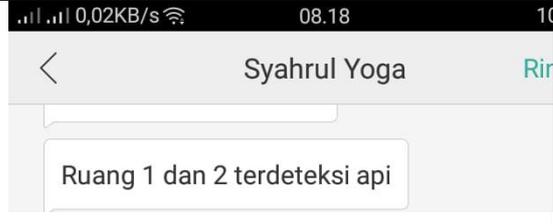
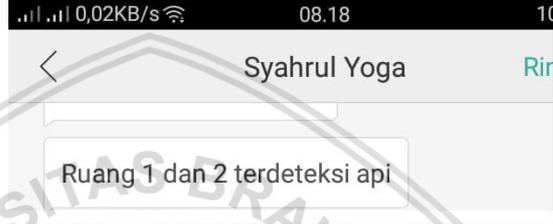
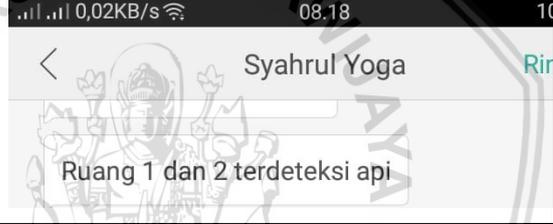
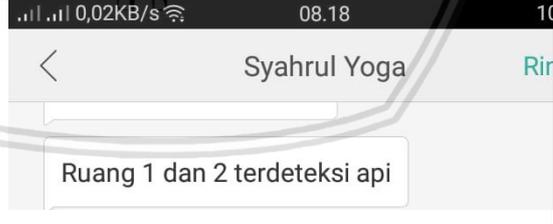
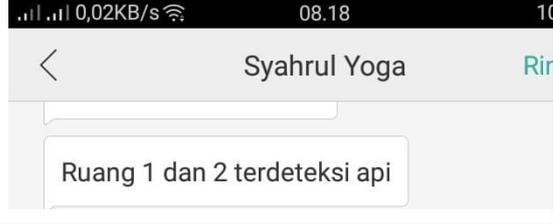
### 6.1.4 Hasil dan Analisis Pengujian

Tabel 6.5 Pengujian *Buzzer* dan *Modul SIM900A*

No	Pengujian Objek pada Ruang 1 Ke -	Buzzer	Modul SIM900A	Kesesuaian
1.	1	Aktif		Sesuai
2.	2	Aktif		Sesuai
3.	3	Aktif		Sesuai
4.	4	Aktif		Sesuai
5.	5	Aktif		Tidak Sesuai

6.	6	Aktif		Sesuai
7.	7	Tidak Aktif	Tidak Mengirim	Sesuai
8.	8	Tidak Aktif	Tidak Mengirim	Sesuai
9.	9	Aktif		Sesuai
10.	10	Aktif		Sesuai
No	Pengujian Objek pada Ruang 2 Ke -	Buzzer	UNIVERSITAS BRAWIJAYA Modul SIM900A	Kesesuaian
11.	11	Aktif		Sesuai
12.	12	Aktif		Sesuai
13.	13	Aktif		Sesuai
14.	14	Aktif		Sesuai

15.	15	Aktif		Sesuai
16.	16	Aktif		Sesuai
17.	17	Tidak Aktif	Tidak Mengirim	Sesuai
18.	18	Tidak Aktif	Tidak Mengirim	Sesuai
19.	19	Aktif		Sesuai
20.	20	Aktif		Sesuai
No	Pengujian Objek pada Ruang 1 dan Ruang 2 Ke -	Buzzer	Modul SIM900A	Kesesuaian
21.	21	Aktif		Sesuai
22.	22	Aktif		Sesuai

23.	23	Aktif		Sesuai
24.	24	Aktif		Sesuai
25.	25	Aktif		Sesuai
26.	26	Aktif		Sesuai
27.	27	Tidak Aktif	Tidak Mengirim	Sesuai
28.	28	Tidak Aktif	Tidak Mengirim	Sesuai
29.	29	Aktif		Sesuai
30.	30	Aktif		Sesuai

Berdasarkan tabel 6.5 diatas telah dilakukan pengujian, hasil pengujian dari 30 pengujian *Buzzer* dan *Modul SIM900A*. Terdapat 1 data uji yang tidak sesuai dengan hasil dari *output* dan keadaan sebenarnya dikarenakan seharusnya pada saat pengolahan citra digital seharusnya tidak mendeteksi titik api karena objek

tersebut bukanlah objek titik api, sehingga sistem tidak perlu mengaktifkan *Buzzer* serta tidak mengirim pesan melalui *Modul SIM900A*. Terdapat 23 data uji sesuai hasil dari *output* yang telah ditunjukkan pada tabel 6.5 bahwa *Buzzer* aktif sesuai dengan perintah yaitu sebagai notifikasi *alarm* dan *Modul SIM900A* aktif mengirimkan layanan pesan singkat (via sms) sesuai dengan yang diperintah oleh sistem.

## 6.4 Pengujian Waktu Komputasi Pemroses Citra

Pengujian waktu komputasi pemroses citra ini yaitu untuk mengetahui berapa lama waktu komputasi yang dibutuhkan untuk mendeteksi objek titik api.

### 6.4.1 Tujuan Pengujian

Tujuan dari pengujian ini yaitu untuk mengetahui hasil waktu komputasi yang dibutuhkan untuk deteksi titik api berdasarkan objek yang terindikasi titik api sesuai dengan sistem.

### 6.4.2. Prosedur Pengujian

Melakukan pengujian waktu komputasi pemrosesan citra dilakukan dengan cara menghitung waktu komputasi ketika program mengambil data dari objek titik api yang terdeteksi titik api dalam satu siklus dilakukan sebanyak 30 kali pengujian. Pengujian waktu komputasi ini diambil dari waktu komputasi akhir dikurangi komputasi awal. Dalam implementasinya program penghitung waktu komputasi pada sistem ini yaitu menggunakan fungsi "*time*" pada python berfungsi mendapatkan waktu dalam satuan 1 *milisecon (ms)*.

### 6.4.3 Pelaksanaan Pengujian

Pengujian ini dilakukan berdasarkan prosedur yang telah dijelaskan pada subbab sebelumnya. Pengujian ini untuk mengetahui hasil waktu komputasi pemroses citra ketika mendeteksi titik api berdasarkan objek yang telah ditentukan yaitu objek yang berpotensi menghasilkan titik api ketika terbakar.

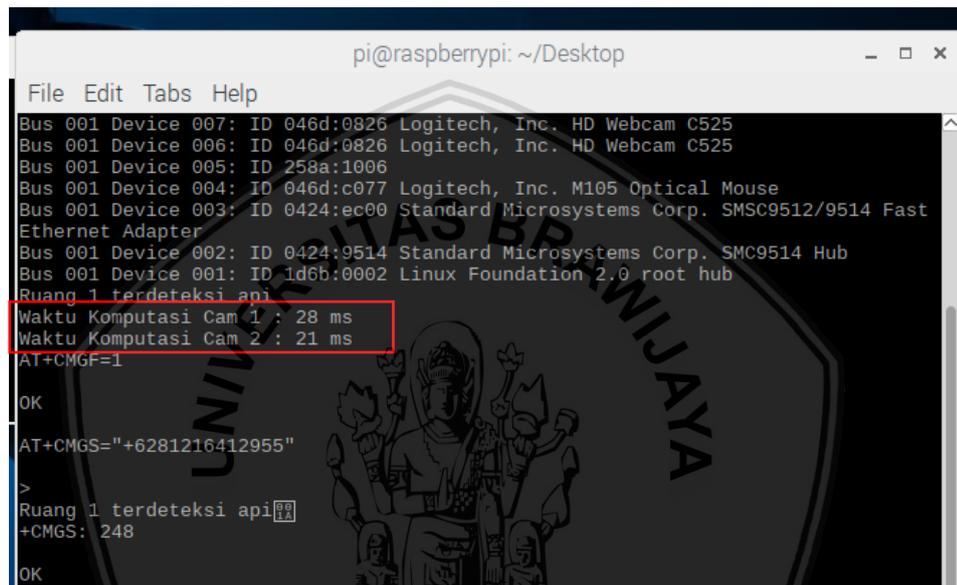
### 6.4.4 Hasil dan Analisis Pengujian

Tabel 6.6 Pengujian Waktu Komputasi Pemrosesan Citra

No	Pengujian Objek pada Ruang 1 Ke -	Waktu Komputasi ( <i>ms</i> )
1.	1	28 ms
2.	2	24 ms
3.	3	24 ms
4.	4	23 ms
5.	5	37 ms
6.	6	37 ms

7.	7	26 ms
8.	8	26 ms
9.	9	26 ms
10.	10	24 ms
Rata-Rata		27,5 ms
No	Pengujian Objek pada Ruang 2 Ke -	Waktu Komputasi (ms)
11.	11	28 ms
12.	12	22 ms
13.	13	21 ms
14.	14	21 ms
15.	15	20 ms
16.	16	21 ms
17.	17	21 ms
18.	18	21 ms
19.	19	23 ms
20.	20	24 ms
Rata-Rata		22,2 ms
No	Pengujian Objek pada Ruang 1 dan Ruang 2 Ke -	Waktu Komputasi (ms)
21.	21	30,5 ms
22.	22	26,5 ms
23.	23	31,5 ms
24.	24	25 ms
25.	25	62,5 ms
26.	26	53,5 ms
27.	27	23,5 ms
28.	28	34 ms
29.	29	34 ms
30.	30	41,5 ms
Rata-Rata		36,25 ms

Berdasarkan pengujian pada tabel 6.6 yang dilakukan sebanyak 30 kali percobaan, waktu komputasi sistem untuk melakukan pengambilan komputasi untuk mendeteksi objek titik api pada setiap kamera yaitu waktu komputasi pada ruang 1 yaitu rata-rata waktu komputasi adalah 27,5 ms pada saat pemrosesan sebuah citra, waktu komputasi pada ruang 2 yaitu rata-rata waktu komputasi adalah 22,2 ms pada saat pemrosesan sebuah citra, dan yang terakhir adalah waktu komputasi pada ruang 1 dan ruang 2 memiliki rata-rata waktu komputasi adalah 36,25 ms pada saat pemrosesan sebuah citra. Screenshot salah satu pengujian waktu komputasi pada python yang ditunjukkan pada gambar 6.1.



Gambar 6.1 Screenshot Hasil Pengujian Komputasi Pemrosesan Citra



## BAB VII

### PENUTUP

Pada bab ini merupakan kesimpulan dari proses yang telah dikerjakan sebelumnya dalam penelitian ini. Bab ini juga, peneliti membeirkan saran untuk digunakan pada pengembangan penelitian yang serupa supaya peneliatian selanjutnya lebih sesuai dengan harapan.

#### 7.1 Kesimpulan

Berdasarkan rumusan masalah yang telah dibuat pada awal penelitian dan hasil analisis dari pengujian yang dilakukan setelah perancangan dan implementasi sistem, didapatkan kesimpulan sebagai berikut :

1. Dalam penelitian ini telah diimplementasi deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms *gateway* pada *raspberry pi*. Perangkat utama yaitu *Raspberry Pi* yang diintegrasikan dengan Kamera *Logitech C525* , *Buzzer*, dan *Modul SIM900A*. Kedua yaitu diimplementasikan program utnuk melakukan proses deteksi titik api. Perangkat utama tersebut dipasang pada sebuah *box* yang menyerupai ruangan.
2. Berdasarkan rumusan masalah, citra digital mengenali objek antar titik api dengan objek lain disekitar titik api yaitu dengan melakukan proses pengolahan citra dengan melakukan *thresholding* pada objek titik api dan *morphological filtering* dengan menggunakan operasi *Opening* untuk membedakan objek antar titik api dengan objek lain disekitar titik api. Tingkat akurasi dari proses untuk mendeteksi objek titik api yang dihasilkan adalah sebanyak 10 data uji sebesar 90%.
3. Tingkat akurasi dari proses deteksi objek titik api pada setiap ruangan pada implementasi sistem deteksi titik api terpusat menggunakan kamera dengan notifikasi berbasis sms *gateway* pada *raspberry pi* pada setiap ruangan yaitu menggunakan 30 data uji disemua kondisi baik diruangan 1, diruangan 2, dan diruangan 1&2. Hasil dari deteksi tersebut dihasilkan keakurasian sebesar 96,66%.
4. Waktu komputasi yang dibutuhkan pada setiap pemrosesan citra digital dari 30 data uji yang diujikan yaitu waktu komputasi sistem untuk melakukan pengambilan komputasi untuk mendeteksi objek citra digital pada setiap kamera yaitu waktu komputasi pada ruang 1 yaitu rata-rata waktu komputasi adalah 27,5 ms pada saat pemrosesan sebuah citra. Waktu komputasi pada ruang 2 yaitu rata-rata waktu komputasi adalah 22,2 ms pada saat pemrosesan sebuah citra digital. Yang terakhir adalah waktu komputasi pada ruang 1 dan ruang 2 ketika sama-sama mendeteksi objek citra digital memiliki rata-rata waktu komputasi adalah 36,25 ms pada saat pemrosesan sebuah citra.

## 7.2 Saran

Adapun beberapa saran yang dapat dijadikan acuan untuk pengembangan penelitian ini ataupun penelitian yang serupa kedepannya adalah sebagai berikut :

1. Menggunakan metode pengolahan citra yang lain untuk membandingkan pengolahan citra manakah yang mempunyai performa yang lebih baik, serta memiliki keakuratan dalam mendeteksi titik api.
2. Menggunakan spesifikasi kamera yang mempunyai agar memperoleh keakuratan yang lebih tinggi dalam mendeteksi objek.
3. Menerapkan sistem serupa untuk memilih objek lain.



## DAFTAR PUSTAKA

- Denny, M. 2012. *Pengenalan Computer Vision dengan EmguCV di C#.Net*.  
<<http://bisakomputer.com/pengenalan-computer-vision-dengan-emgucv-di-c-net/>> [Diakses 02 Desember 2017]
- Derisma., 2017. *Faktor-Faktor yang Mempengaruhi Sistem Pengenalan Wajah Menggunakan Metode Eigenface pada Perangkat Mobile Berbasis Android*. Padang: Universitas Andalas.
- Fauzi, J.F, Herman Tolle, Ratih Kartika Dewi., 2017. *Implementasi Metode RGB To HSV pada Aplikasi Pengenalan Mata Uang Kertas Berbasis Android untuk Tuna Netra*. Malang: Universitas Brawijaya.
- Fitriyah, H., 2016. *Modul Ajar Komputasi Citra Dan Suara Digital*. Malang: Universitas Brawijaya.
- Gunawaardena, A.E, R.M.M. Ruwanthika, A.G.B.P. Jayasekara., 2016. *Computer Vision Base Fire Alarming System*, Srilanka: University of Moratuwa
- Hasan, Md.M., Razzak M.A., 2016. *An Automatic Fire Detection and Warning System Under Home Video Surveillance*. Bangladesh: Departement of Electronic Engineering, Independent University.
- Nurromianto, Imam., 2015. Rancangan Prototipe Pendeteksian Dini Lokasi Kebakaran Berbasis *Wireless* Dengan *Fuzzy Logic* Dan Pemberitahuan Via SMS. *Digital Repositoty*, Jember : Universitar Jember.
- Instructable, 2017 . *How to use a Buzzer Arduino Tutorial* [Online]  
<<http://www.instructables.com/id/How-to-use-a-Buzzer-Arduino-Tutorial/>>  
[Diakses 06 Maret 2018]
- Kelvin, Pram Eliyah Yuliana, dan Sri Rahayu.,2015. *Pemetaan Lokasi Kebakaran Berdasarkan Prinsip Segitiga Api Pada Industri Textile*. Surabaya: Sekolah Tinggi Teknik Surabaya.
- Kumaseh, M.R., Latumakulita, L., Naiggolan, N., 2013. Segmentasi Citra Digital Ikan Menggunakan Metode *Thresholding*. Manado : Universitas Sam Ratulangi.

- Logitech. *Logitech C525 Product Spesification*. [Online] <<https://www.logitech.com/id-id/product/hd-webcam-c525#specification-tabular>> [Diakses tanggal 06 Maret 2018].
- Pratama, R.A, Aqwam Rosadi Kardian., 2012. *Sensor Parkir Mobil Berbasis Mikrokontroler AT89S51 Dengan Bantuan Mini Kamera*. Jakarta: STMIK JAKARTA STI&K.
- Purnomo, R., Syauqy, D., & Hanafi, M., 2017. Implementasi Metode *Fuzzy Sugeno* Pada *Embedded System* Untuk Mendeteksi Kondisi Kebakaran Dalam Ruangan. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, vol.2, no. 4, p. 1428-1435.
- Pusdiklatkar., 2006. Modul Pelatihan: *Perilaku Api*. Jakarta.
- Putra, Darma., 2010. *Pengolahan Citra Digital*. Buku Teks. Penerbit ANDI. Yogyakarta. Indonesia.
- Raspberry Pi. 2 Model B. *Products, Raspberry Pi 2 Model B*. [Online] <<https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>> [Diakses tanggal 07 Januari 2017].
- Raharjo, Pambudi., 2014. *Deteksi Kebakaran Pada Video Berbasis Pengolahan Citra*, Yogyakarta : AMIKOM YOGAYAKARTA
- Seebamrungsat, J, Suphachai Praising, Panomkhawn Riyamongkol., 2014. *Fire Detection in the Buildings Using Image Processing*, Thailand: Naesuran University.
- Setiani, Y., 2015. *Pengendalian Bahaya Kebakaran Melalui Optimalisasi Tata Kelola Lahan Kawasan Perumahan di Wilayah Perkotaan*. p. 1.
- Sigana, T., 2017. *Pandan Bencana Kebakaran Rumah*. [Online] Available at: <http://sigana.web.id/index.php/panduan-bencana/kebakaran-rumah.html> [Diakses Tanggal 09 September 2017]
- Soberindo., 2017. *Teori Segitga Api*. [Online] at: <http://saberindo.co.id/2017/08/03/teori-segitiga-api.html> [Diakse Tanggal 02 Januari 2017]

Suryadi, Rivan Radian, Inung Wijayanto,S.T. , Angga Rusdinar, .S.T.,M.t.,Ph.D., 2017.

*Perancangan Dan Implementasi Sistem Pendeteksi Api Pada Robot Pemadam Api Dengan Menggunakan Sensor Api Dan Kamera.* Bandung: Universitas Telkom.

Swedia, Ericks R., Cahyanti, Margi., 2015. *Algoritma Transformasi Ruang Warna.*

Depon : Indie Publishing.

