

**SISTEM PENDETEKSI PENYAKIT DAUN BAWANG MERAH  
PROBOLINGGO MENGGUNAKAN METODE TEMPLATE  
MATCHING BERBASIS RASPBERRY PI**

**SKRIPSI**

Untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:

Moch. Zamroni

NIM: 145150301111010



**PROGRAM STUDI TEKNIK KOMPUTER**

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS BRAWIJAYA**

**MALANG**

**2018**

## PENGESAHAN

SISTEM PENDETEKSI PENYAKIT DAUN BAWANG MERAH PROBOLINGGO MENGGUNAKAN  
METODE TEMPLATE MATCHING BERBASIS RASBERRY PI

### SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan  
memperoleh gelar Sarjana Teknik

Disusun oleh:

Moch.Zamroni

NIM: 145150301111010

Skripsi ini telah diuji dan dinyatakan lulus pada  
27 Juli 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I

Hurriyatul Fitriyah, S.T., M.Sc  
NIP: 19851001 201504 2 003

Dosen Pembimbing II

Rizal Maulana, S.T., M.T., M.Sc  
NIK: 2016078 910091 001

Mengetahui

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, 27 Juli 2018



6000  
ENAM RIBURUPIAH

Moch. Zamroni

NIM: 145150301111010

## KATA PENGANTAR

Puji syukur kehadirat Allah SWT yang telah melimpahkan rahmat, taufik dan hidayah-Nya sehingga laporan skripsi yang berjudul “Sistem Pendeteksi penyakit daun bawang merah menggunakan metode template machine” ini dapat terselesaikan.

Penulis menyadari bahwa skripsi ini tidak akan berhasil tanpa bantuan dari beberapa pihak. Oleh karena itu, penulis ingin menyampaikan rasa hormat dan terima kasih kepada:

1. Kedua Orang Tua penulis, Bapak Asmito dan Ibu Saurib yang setiap saat selalu memberikan dukungan serta doa selama penyusunan skripsi ini.
2. Ibu Hurriyatul Fitriyah, S.T, M.Sc dan Bapak Rizal Maulana, S.T., M.T., M.Sc<sub>2</sub> selaku Dosen pembimbing skripsi yang telah dengan sabar membimbing dan mengarahkan penulis sehingga dapat menyelesaikan skripsi ini.
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika.
4. Seluruh dosen dan staff Fakultas Ilmu Komputer Universitas Brawijaya yang telah memberikan bekal ilmu yang luar biasa mulai dari proses pertama perkuliahan hingga akhir proses skripsi ini.
5. Seluruh mahasiswa Teknik Komputer Angkatan 2014.
6. Semua pihak yang tidak dapat penulis sebutkan satu-persatu. Terimakasih atas bantuannya.

Penulis menyadari bahwa dalam penyusunan skripsi ini masih banyak kekurangan, sehingga saran dan kritik yang membangun sangat penulis harapkan. Akhir kata penulis berharap skripsi ini dapat membawa manfaat bagi semua pihak yang menggunakannya.

Malang, 27 Juli 2018

Penulis

moch.zamroni96@gmail.com

## ABSTRAK

Dengan kemajuan teknologi sekarang ini semakin meningkat pula tuntutan akan sistem mikrokomputer yang mampu memberikan manfaat bagi kehidupan manusia. Salah satunya adalah kebutuhan embeded sistem yang dapat mengenali penyakit daun pada bawang merah. Bawang merah merupakan salah satu sayuran umbi yang penting bagi Indonesia. Kebutuhan bawang merah di Probolinggo setiap tahun terus meningkat dari tahun 2010 sampai 2014 rata-rata peningkatan produksi bawang merah di Indonesia selama 5 tahun terakhir sebesar 4.85persen per tahun. Namun beberapa tahun ini penghasilan tanaman bawang merah terus menurun. Para petani bawang merah merasa resah dengan kejadian ini. Sistem ini memiliki peranan penting untuk membantu manusia dalam melakukan kegiatan seperti mengetahui penyakit pada daun bawang merah, sehingga bermanfaat untuk orang awam dalam bidang pertanian penyakit daun bawang merah. *Embedded* sistem ini mempunyai input berupa *image capture* kamera dari obyek daun bawang yang akan dibandingkan dengan template daun bawang yang sudah tersedia dimikrokomputer, dimana akan dipilih nilai tertinggi dari perbandingan nilai grayscale template dan *image* yang *dicapture* selanjutnya nilai yang sudah terpilih akan diproses ke metode *Template Matching* untuk mendapatkan hasil yang lebih akurat dengan batasan *threshold* yang sudah ditentukan, dan akan memberikan output pada salah satu lampu LED yang sudah tersedia. Oleh karena itu, dirancanglah sebuah sistem pendeteksi penyakit daun bawang merah yang memudahkan mengetahui penyakit daun pada bawang merah dengan memanfaatkan teknologi tersebut. Hasil dari pengujian ini menunjukkan kamera Logitech C270 dapat mengambil gambar dari penyakit daun dengan rata-rata error kecil 0%. Dan Berdasarkan hasil pengujian Akurasi metode *Template Matching* yang terlihat bahwa dari jumlah 30 data terdapat 6 hasil dari sistem yang tidak sesuai dengan kelas sebenarnya. Sehingga akurasi yang diperoleh Sistem Pendeteksi Penyakit Daun dengan Metode *Template Matching* ini adalah sebesar 80%. Performasi waktu respon sistem penyakit daun bawang merah mempunyai nilai kecepatan waktu pemrosesan rata-rata sebesar 15.977 detik dari 30 kali pengujian. Pada penelitian ini telah dibuat sistem untuk mendeteksi penyakit daun bawang merah menggunakan template matching, dimana metode yang diterapkan mampu berjalan dengan apa yang diinginkan terbukti dengan sistem dapat mengklasifikasikan jenis penyakit daun menjadi penyakit daun bercak ungu, embun bulu serta Moler.

Kata kunci : Penyakit, Daun Bawang Merah, Template Matching.

## ABSTRACT

*By the progress of technology currently is also increasing demand for microcomputer systems that can provide benefits for human life. One of them is the need of Embedded system that can recognize leaf disease on the shallot. shallot is one of the important bulb vegetables for Indonesia. Every year, the need for shallot in Probolinggo continues to increase from 2010 to 2014 average increase in onion production in Indonesia for the last 5 years of 4.85 persen per year. But in recent years the production of shallot plants continues to decline. The onion farmers feel restless with this incident. This system has an important role to help humans, so it is useful for the layman in the field of shallot leaf agriculture. This Embedded system has a camera image capture input of a leek object that will be compared to templates already available in microcomputers, then the value is selected will be processed to the Template Matching method with the specified threshold limit, and will output one of the LED lights which is already available. Therefore, a system is designed to detect the disease of shallot to determine the leaf disease on the shallot. The results of this test show the Logitech C270 camera can take pictures. Based on the results of testing the accuracy of Template Matching method that the amount of 30 data there are 6 results from systems that does not match the actual class. So the accuracy is obtained by 80%. Performance of the system response time has a time value of 15.977 seconds from 30 tests. In this research, a system to detect the disease of shallot using template matching, where the applied method is able to run by what is desired proven by the system can classify the type of leaf disease into purple patch leaf disease, moisture and Moler.*

*Keywords : Disease, Shallot Leaves, Template Matching*

## DAFTAR ISI

<b>PENGESAHAN</b> .....	<b>Error! Bookmark not defined.</b>
<b>PERNYATAAN ORISINALITAS</b> .....	<b>Error! Bookmark not defined.</b>
<b>KATA PENGANTAR</b> .....	iii
<b>ABSTRAK</b> .....	iv
<b>ABSTRACT</b> .....	v
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR TABEL</b> .....	ix
<b>DAFTAR GAMBAR</b> .....	x
<b>BAB 1 PENDAHULUAN</b> .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan .....	3
1.4 Manfaat.....	3
1.5 Batasan Masalah.....	3
1.6 Sistematika pembahasan .....	3
<b>BAB 2 LANDASAN KEPUSTAKAAN</b> .....	6
2.1 Tinjauan Pustaka.....	6
2.2 Dasar Teori .....	8
2.2.1 Penyakit Tanaman Bawang Merah .....	8
2.2.2 <i>Grayscale</i> .....	10
2.2.3 <i>Template Matching</i> .....	10
2.2.4 Kamera <i>webcam</i> .....	11
2.2.5 Raspberry pi .....	12
2.2.6 Open CV.....	13
<b>BAB 3 METODOLOGI</b> .....	15
3.1 Alur metode penelitian .....	15
3.2 Studi Literatur .....	16
3.3 Analisa Kebutuhan .....	17
3.3.1 Kebutuhan Pengguna.....	17
3.3.2 Kebutuhan Sistem .....	17
3.3.3 Kebutuhan Perangkat Keras.....	17



3.3.4 Kebutuhan Perangkat Lunak.....	18
3.4 Perancangan Sistem.....	18
3.5 Implementasi Sistem.....	18
3.6 Pengujian dan Analisis .....	19
3.7 Kesimpulan.....	19
<b>BAB 4 REKAYASA KEBUTUHAN .....</b>	<b>20</b>
4.1 Gambaran Umum Sistem.....	20
4.2 Analisis Kebutuhan Sistem.....	20
4.2.1 Kebutuhan Fungsional .....	20
4.2.2 Kebutuhan Non Fungsional.....	21
4.3 Batasan Desain Sistem .....	22
<b>BAB 5 PERANCANGAN DAN IMPLEMENTASI .....</b>	<b>23</b>
5.1 Perancangan Sistem.....	23
5.1.1 Perancangan <i>Prototype</i> Alat Pendeteksi Penyakit Daun .....	23
5.1.2 Perancangan Perangkat Keras .....	24
5.1.3 Perancangan Perangkat Lunak.....	25
5.2 Implementasi Sistem .....	30
5.2.1 Implementasi <i>Prototype</i> Alat Pendeteksi Penyakit Daun Bawang Merah.....	31
5.2.2 Implementasi Perangkat Keras .....	32
5.2.3 Implementasi Perangkat Lunak .....	33
5.2.3.1 Implementasi Kode Program .....	33
<b>BAB 6 PENGUJIAN DAN ANALISIS .....</b>	<b>40</b>
6.1 Pengujian Fungsional Kamera.....	40
6.1.1 Tujuan Pengujian .....	40
6.1.2 Prosedur Pengujian.....	40
6.1.3 Hasil dan Analisis Pengujian.....	42
6.2 Pengujian Fungsional Tampilan pada LED .....	44
6.2.1 Tujuan Pengujian .....	45
6.2.2 Prosedur Pengujian.....	45
6.2.3 Hasil dan Analisis Pengujian.....	45
6.3 Pengujian Akurasi Hasil Klasifikasi <i>Template Matching</i> .....	46
6.3.1 Tujuan Pengujian .....	46
6.3.2 Prosedur Pengujian.....	46

6.3.3 Hasil dan Analisis Pengujian.....	47
6.4 Pengujian Waktu Komputasi Pemrosesan Sistem .....	52
6.4.1 Tujuan Pengujian .....	52
6.4.2 Prosedur Pengujian.....	52
6.4.3 Hasil dan Analisis Pengujian.....	53
BAB 7 PENUTUP .....	55
7.1 Kesimpulan .....	55
7.2 Saran .....	55
Daftar Pustaka.....	57



## DAFTAR TABEL

Tabel 2.1 Daftar beberapa paper yang mengenai penyakit daun bawang merah dan template matching .....	6
Tabel 5.1 keterangan koneksi pin raspberry pi GPIO .....	25
Tabel 5.2 kode pemograman inisialisasi <i>library</i> sistem pendeteksi penyakit daun .....	33
Tabel 5.3 kode pemograman inisialisasi pin GPIO dan Capture kamera .....	34
Tabel 5.4 kode pemograman pembuatan Template .....	37
Tabel 5.5 kode pemograman menggunakan Template Matching.....	38
Table 5.6 kode pemograman output Led .....	39
Tabel 6.1 hasil pengujian pembacaan kamera .....	42
Tabel 6.2 hasil pengujian lampu LED .....	45
Tabel 6.3 data uji dan hasil pengujian Template Matching.....	47
Tabel 6.4 hasil pengujian waktu komputasi .....	53



## DAFTAR GAMBAR

Gambar 2.1 penyakit <i>Alternaria porii</i> dan gejala serangan pada tanaman bawang merah .....	8
Gambar 2.2 gejala serangan penyakit embun bulu pada tanaman bawang merah .....	9
Gambar 2.3 gejala serangan moler pada tanaman bawang merah .....	10
Gambar 2.4 logitech C270 HD webcam .....	12
Gambar 2.5 raspberry pi .....	13
Gambar 2.6 logo open cv .....	14
Gambar 3.1 diagram alir penelitian .....	15
Gambar 3.2 gambaran umum sistem .....	18
Gambar 5.1 desain <i>prototype</i> alat pendeteksi daun bawang merah .....	23
Gambar 5.2 diagram skematik sistem .....	24
Gambar 5.3 flowchart utama .....	26
Gambar 5.4 diagram alir perancangan perangkat lunak pengambilan data obyek .....	27
Gambar 5.5 diagram alir perancangan pembuatan template .....	28
Gambar 5.6 diagram alir perancangan klasifikasi <i>Template Matching</i> .....	30
Gambar 5.7 implementasi <i>prototype</i> alat pendeteksi penyakit daun .....	31
Gambar 5.8 implementasi rangkaian raspberry pi dengan kamera .....	32
Gambar 5.9 implementasi rangkaian LED, jumper dan <i>push button</i> .....	32
Gambar 5.10 hasil gambar capture kamera .....	35
Gambar 5.11 hasil gambar template RGB Bercak Ungu .....	35
Gambar 5.12 hasil gambar template RGB Embun Bulu .....	36
Gambar 5.13 hasil gambar template RGB Moler .....	36



Gambar 5.14 hasil template Grayscale Bercak Ungu .....36

Gambar 5.15 hasil template Grayscale Embun Bulu .....37

Gambar 5.16 hasil template Grayscale Moler .....37

Gambar 5.17 hasil output Led penyakit daun bawang threshold  $\geq 80$  .....39

Gambar 5.18 hasil output Led penyakit daun bawang threshold  $\geq 90$  .....39



## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Bawang merah merupakan salah satu sayuran umbi yang penting bagi Indonesia. Kebutuhan bawang merah di Probolinggo setiap tahun terus meningkat dari tahun 2010 sampai 2014 rata-rata peningkatan produksi bawang merah di Indonesia selama 5 tahun terakhir sebesar 4.85 persen per tahun. Produksi bawang merah di Kabupaten Probolinggo hingga akhir September 2017 mencapai 60.399 ton atau 110% dari target produksi sebesar 54.880 ton dengan luas panen mencapai 5.967 hektar. Luas tanam bawang merah di Kabupaten Probolinggo mencapai 6.306 hektar atau 117,147% dari target luas tanam 5.383 hektar, Kepala Dinas Ketahanan Pangan dan Pertanian (DKPP) Kabupaten Probolinggo Ahmad Hasyim Ashari melalui Kasi Tanaman Hortikultura (Rahadiyanto, 2017). Namun beberapa tahun ini penghasilan tanaman bawang merah terus menurun. Para petani bawang merah merasa resah dengan kejadian ini.

Demikian juga dengan para petani tanaman bawang merah yang berada di Kecamatan Tegalsiwalan tempat penelitian yang saya lakukan, para petani mengalami berbagai permasalahan, mulai dari gejala, penyebab, hama dan penyakit daun bawang merah hingga penanganannya terhadap hama dan penyakit yang ada. Dimana pada saat ini banyak ditemukan dilapangan daun bawang merah yang terpelintir yang disebabkan oleh salah satu penyakit daun bawang merah yang membuat umbi bawang merah membusuk tidak normal seperti biasanya. Menurut Kepala BPP (Badan Pelaksana Penyuluhan) Banjarharjo dan Ketua Badan Pelaksana Penyuluhan Kabupaten Probolinggo, para petani yang mengeluhkan banyaknya hama dan penyakit yang menyerang tanaman bawang merah, sehingga para penyuluh merasa kesulitan dalam memberikan penyuluhan kepada para petani karena kurangnya pakar dalam mengatasi solusi terbaik dari permasalahan tersebut. Untuk itu diperlukan pendiagnosaan terhadap gangguan penyakit daun pada tanaman bawang merah. Tujuan penelitian ini adalah dihasilkannya perangkat keras yang dapat mendiagnosa penyakit daun bawang merah yang dialami tanaman bawang merah (Fadlil, 2013).

Pada saat ini, sering terjadi perubahan cuaca yang tidak menentu, seperti musim hujan yang tidak beraturan dan musim kemarau yang tidak sesuai dengan perkiraan cuaca. Hal itu berakibat juga pada sector pertanian. Contohnya pada petani bawang merah asal Probolinggo

yang mengalami penurunan kualitas hasil panen. Salah satu penyebab penurunan hasil panen adalah adanya penyakit daun bercak ungu /*Alternaria Pori* pada umumnya serangan ini ditandai dengan adanya noda putih atau kekuningan pada daun, kemudian melebar dan menyebabkan daun menjadi kering terutama pada bagian pucuk,selanjutnya menjalar dalam waktu 5-6 hari tanaman bawang merah akan mati. Penyakit Embun Bulu / *Peronospora destructor* pada kondisi ini ditandai dengan terlihat bulu-bulu halus berwarna ungu atau hitam yang menutupi daun bagian luar dan batang umbi gejala akan terlihat jika terkena embun, sehingga seluruh daun akan mengering, berkerut dan layu dan akan mengakibatkan umbi membusuk. Penyakit Moler/*Fusarium oxysporum* serangan ini ditandai dengan serangan bagian dasar umbi lapis sehingga daun menguning dan cenderung terpelintir(terputar). Oleh karena itu, perlu adanya sebuah sistem yang mampu melakukan pengamatan pada daun bawang merah secara berlanjut sehingga dapat memberikan informasi guna menjaga kondisi daun sesuai dengan yang diinginkan (budidaya, 2018).

Berdasarkan permasalahan yang disampaikan, pada penelitian ini penulis menggunakan kamera sebagai pengumpul data untuk mengkoleksi data penyakit daun bawang merah dan mikrokomputerr yang digunakan adalah *Raspberry Pi* serta kamera Logitech C270 dengan metode *Template Matching* untuk mengenali salah satu penyakit yang diantaranya penyakit Bercak Ungu, Penyakit Embun Bulu, dan Penyakit Moler.

## 1.2 Rumusan Masalah

Rumusan masalah yang dilakukan dalam skripsi ini adalah:

1. Bagaimana implementasi *Template Matching* pada *Raspberry Pi* untuk mengenali penyakit daun bawang merah?
2. Bagaimana membangun sistem pada *Raspberry Pi*?
3. Bagaimana hasil pengujian pencocokan *Template* pada Obyek menggunakan *Template Matching*?
4. Bagaimana waktu respon system terhadap perubahan yang terjadi pada lingkungan kamera?

### 1.3 Tujuan

Adapun tujuan dari penelitian ini adalah sebagai berikut:

1. Untuk mengetahui proses implementasi *Template Matching* pada *Raspberry Pi* dalam mengenali penyakit daun bawang merah
2. Untuk mengetahui bagaimana membangun sistem pada *Raspberry Pi*.
3. Untuk mengetahui bagaimana hasil pencocokan Template Obyek terhadap Metode Template Matching.
4. Untuk mengetahui bagaimana kemampuan sistem dalam menanggapi perubahan yang terjadi pada lingkungan kamera.

### 1.4 Manfaat

Hasil dari skripsi ini diharapkan dapat memberikan manfaat, diantaranya adalah sebagai berikut:

1. Skripsi ini diharapkan dapat mempermudah desa sumberbulu kota probolinggo dalam mengetahui suatu penyakit pada daun bawang merah .
2. Mendapatkan wawasan pengetahuan tentang sebuah sistem yang terdiri dari kamera dan sebuah mikrokomputer.

### 1.5 Batasan Masalah

Batasan masalah dalam skripsi ini adalah:

1. Menggunakan kamera Logitech c270 untuk pengambilan gambar
2. Menggunakan mikrokomputer *Raspberry pi* untuk pengolah gambar
3. Sistem ini dirancang untuk mendeteksi 3 penyakit daun bawang yaitu Bercak ungu, Embun bulu, dan Moler
4. Pembahasan difokuskan pada keakuratan hasil gambar yang dihasilkan oleh sistem
5. Obyek diletakkan pada sebuah kotak kecil yang ada didalam box
6. Semua daun memiliki penyakit bercak ungu, embun bulu, moler dan tidak ada yang normal

### 1.6 Sistematika pembahasan

Secara garis besar proposal ini terdiri dari 7(tujuh) bab dengan beberapa sub bab. Agar mendapat gambaran yang jelas mengenai keseluruhan hal yang tertulis maka sistematika pembahasan dari penyusunan skripsi adalah sebagai berikut.

## **BAB 1 Pendahuluan**

Bab I pendahuluan pada bab ini membahas beberapa sub bab berisikan pendahuluan yaitu tentang latar belakang mengapa penulis memilih masalah terkait untuk dijadikan penelitian, yang kedua adalah rumusan masalah yang berisi beberapa pertanyaan yang akan dijawab pada penelitian ini. Kemudian tujuan penelitian dan manfaat penelitian ini. Selanjutnya ada batasan masalah penelitian untuk membatasi ruang lingkup penelitian. Dan terakhir sistematika pembahasan pada penelitian ini.

## **BAB 2 Tinjauan Pustaka dan Dasar Teori**

Pada bab ini membahas tentang tinjauan pustaka yang menjadi referensi bagi penulis untuk melakukan penelitian ini. Teori dan kajian yang mendasari serta mendukung dalam menyelesaikan penelitian ini.

## **BAB 3 Metodologi Penelitian**

Pada bab ini membahas tentang bagaimana langkah-langkah untuk membangun sebuah sistem pengamatan pendeteksi daun dengan menggunakan kamera dengan mekanisme mikrokontroler. Bab ini juga membahas langkah-langkah implementasi, analisis dan pengujian sistem yang dibangun secara umum.

## **BAB 4 Rekayasa Kebutuhan**

Pada bab ini penulis menjelaskan bagaimana gambaran umum tentang semua kebutuhan sistem, serta kebutuhan fungsional dan non fungsional sistem. Perangkat keras dan perangkat lunak serta kebutuhan komunikasi yang dibutuhkan oleh sistem

## **BAB 5 Perancangan dan Implementasi**

Bab ini menjelaskan tentang perancangan sistem seperti diagram blok sistem, perancangan database sistem serta hasil implementasi perancangan sistem.

## **BAB 6 Pengujian dan Analisis**

Pengujian dilakukan apabila perancangan dan implementasi pada sistem telah dilakukan. Pengujian dilakukan untuk memperoleh data yang nantinya digunakan untuk proses analisis.

## **BAB 7 Penutup**

Pada bab ini penulis menjelaskan mengenai kesimpulan tentang penelitian ini dan saran dari hasil pembuatan sistem. Sehingga bab ini melakukan pengembangan mengenai penelitian ini tentang masa depan.



## BAB 2 LANDASAN KEPUSTAKAAN

Landasan kepastakaan berisi uraian dan pembahasan tentang teori, konsep, model, metode, atau sistem dari literatur ilmiah, yang berkaitan dengan tema, masalah, atau pertanyaan penelitian. Dalam landasan kepastakaan terdapat landasan teori dari berbagai sumber pustaka yang terkait dengan teori dan metode yang digunakan dalam penelitian. Jika dibutuhkan sesuai dengan karakteristik penelitiannya dan syarat kecukupan khusus keminatan tertentu, bisa juga terdapat kajian pustaka yang menjelaskan secara umum penelitian terdahulu yang berhubungan dengan topik skripsi dan menunjukkan persamaan dan perbedaan skripsi tersebut terhadap penelitian terdahulu yang dituliskan.

### 2.1 Tinjauan Pustaka

Dalam usaha penanaman bawang merah penyakit merupakan salah satu resiko yang harus dihadapi oleh petani. Oleh karena itu, mengenal gejala dari masing-masing penyakit bawang merupakan hal yang sangat penting bagi suksesnya usaha tanaman bawang merah. Sekitar 60 persen penduduk didaerah Kecamatan Maron Kabupaten Probolinggo bekerja pada sektor pertanian. Meskipun demikian, adanya permintaan dan kebutuhan bawang merah yang terus meningkat setiap tahunnya belum dapat diikuti oleh peningkatan produksinya. Hal ini disebabkan oleh keterbatasan dalam hal budidaya tanaman seperti keberagaman jenis tanah, pengendalian bawang, penyakit dan gulma, pemupukan serta penanganan pascapanennya.

Pada Tabel 2.1 berikut akan ditunjukkan beberapa paper yang menjadi acuan penulis yang membahas mengenai topik *Sistem pendeteksi penyakit daun bawang menggunakan Template Matching*.

**Tabel 2.1 Daftar Beberapa Paper yang Mengenai Penyakit daun bawang dan Template Matching**

No	Judul Paper	Tujuan	Metode	Hasil
1	SISTEM PAKAR IDENTIFIKASI PENYAKIT TANAMAN BAWANG MERAH MENGGUNAK	Dapat memberikan hasil diagnosa penyakit bawang merah berupa jenis penyakit, dan	TEOREMA BAYES	Sistem ini mampu mengidentifikasi penyakit bawang merah berdasarkan gejala yang

	AN METODE TEOREMA BAYES	cara pengobatanny a		dipilih serta dapat memberikan hasil kesimpulan.
2	IDENTIFIKASI JENIS POHON MANGGA BERDASARKAN DAUNNYA MENGGUNAK AN METODE <i>TEMPLATE MATCHING</i>	Sebagai alat untuk membantu mengenali jenis pohon manga berdasarkan daunnya, khususnya Mangga Manalagi, Mangga Apel dan Mangga Gadung	TEMPLATE MATCHIN G	Aplikasi identifikasi jenis pohon mangga berdasarkan daunnya ini berjalan dengan baik dan mampu mengidentifikas i jenis manga khususnya mangga manalagi, apel dan gadung .

Kesimpulan dari kedua paper diatas mempunyai kelebihan dan kekurangan masing masing diantaranya :

Sistem pakar identifikasi penyakit tanaman bawang merah menggunakan metode *Teorema Bayes* dimana mempunyai kelebihan untuk membantu petani bawang merah yang masih awam dalam penanganan dari berbagai macam penyakit serta pengobatannya, kekurangan dari paper ini kurang komplek data penyakit bawang merah yang lainnya (Ardhini warih utami, 2015). Sedangkan identifikasi jenis pohon manga berdasarkan daunnya menggunakan metode *Template Matching* paper ini mempunyai kelebihan untuk membantu mengenali jenis pohon mangga berdasarkan daunnya kususnya mangga manalagi, apel dan gadung. Kekurangan paper ini hanya bisa memberikan nama pada mangga tertentu saja yaitu mangga manalagi,apel dan gadung dan memiliki nilai akurasi 79,45% dalam penelitiannya (sari, 2017). Dan selanjutnya sistem pendeteksi penyakit daun bawang merah probolinggo menggunakan metode *Template Matching* berbasis raspberry pi mampu membedakan salah satu penyakit daun bawang merah bagi petani yang masih pemula yang diantaranya penyakit bercak ungu, penyakit embun bulu serta penyakit moler.

## 2.2 Dasar Teori

Pada bagian dasar teori menjelaskan tentang beberapa definisi tentang teori yang digunakan dalam pengolahan citra digital, penyakit daun bawang mengenai gejala, komponen yang digunakan untuk membangun sistem, dan aplikasi pendukung pemroses citra digital.

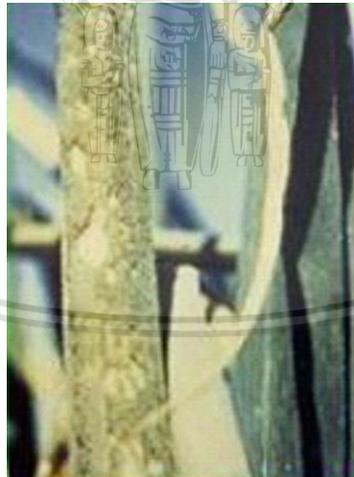
### 2.2.1 Penyakit Tanaman Bawang Merah

a. Penyakit trotol atau bercak ungu (Purple blotch) atau patogen cendawan *Alternaria porri* (Ell.) Cif merupakan penyakit yang gejala infeksi awal pada daun menimbulkan bercak berukuran kecil, melekok ke dalam, berwarna putih dengan pusat yang berwarna ungu (kelabu). Jika cuaca lembab, serangan berlanjut dengan cepat, bercak berkembang hingga menyerupai cincin dengan bagian tengah yang berwarna ungu dengan tepi yang kemerahan dikelilingi warna kuning yang dapat meluas ke bagian atas maupun bawah bercak. Ujung daun mengering, sehingga daun patah. Permukaan bercak tersebut akhirnya berwarna coklat kehitaman Gambar 2.1 Penyakit *Alternaria porri* dan gejala serangannya pada tanaman bawang merah (Foto : Jacobsen/Shurleff) Serangan dapat berlanjut ke umbi, yang menyebabkan umbi membusuk, berwarna kuning lalu merah kecoklatan. Semula umbi membusuk dan berair yang dimulai dari bagian leher, kemudian jaringan umbi yang terinfeksi mengering dan berwarna lebih gelap. Umbi tersebut dapat menjadi sumber infeksi untuk tanaman generasi berikutnya jika digunakan sebagai bibit (Bagus K.Udiarto, 2005).



Gambar 2.1 Penyakit *Alternaria porri* dan gejala serangannya pada tanaman bawang merah (Foto : Jacobsen/Shurleff)

b. Penyakit embun bulu atau tepung palsu (Downy mildew) atau pathogen cendawan *Peronospora destructor* (Berk.) Casp merupakan penyakit dimana pada kondisi yang lembab, berkabut atau curah hujan tinggi, cendawan akan membentuk masa spora yang sangat banyak, yang terlihat sebagai bulu-bulu halus berwarna ungu (violet) yang menutupi daun bagian luar dan batang (umbi) Gambar 2.2 Gejala serangan penyakit embun bulu pada tanaman bawang merah (Foto : Sherf) gejala kelihatan lebih jelas jika daun basah terkena embun. Gejala akibat infeksi cendawan ini dapat bersifat sistemik dan lokal. Jika infeksi terjadi pada awal pertumbuhan tanaman, dan tanaman mampu bertahan hidup, maka pertumbuhan tanaman terhambat dan daun berwarna hijau pucat. Bercak infeksi pada daun mampu menyebar ke bawah hingga mencapai umbi lapis, kemudian menjalar ke seluruh lapisan, Akibatnya, umbi menjadi berwarna coklat. Serangan lanjut akan mengakibatkan umbi membusuk, tetapi lapisan luarnya mengering dan berkerut, daun layu dan mengering, sering dijumpai anyaman miselia yang berwarna hitam. Gejala lokal biasanya merupakan akibat infeksi sekunder, yang mengakibatkan bercak pada daun yang berwarna pucat dan berbentuk lonjong, yang mampu menimbulkan gejala sistemik seperti tersebut di atas (Bagus K.Udiarto, 2005).



Gambar 2.2 Gejala serangan penyakit embun bulu pada tanaman bawang merah

(Foto : Sherf)

c. Penyakit moler atau layu Fusarium (Twisting Disease) atau Organisme cendawan *Fusarium oxysporum* (Hanz.) merupakan penyakit yang gejala sasaran serangan adalah bagian dasar umbi lapis. Akibatnya pertumbuhan akar maupun umbi terganggu. Gejala visual adalah daun yang menguning dan cenderung terpelintir (terputar). Tanaman sangat mudah tercabut karena pertumbuhan akar terganggu bahkan membusuk. Pada dasar umbi terlihat cendawan yang berwarna keputih-putihan, sedangkan jika umbi lapis dipotong membujur terlihat adanya pembusukan, yang berawal dari dasar umbi meluas ke atas maupun ke samping. Serangan lanjut akan mengakibatkan tanaman mati, yang dimulai dari ujung daun dan dengan cepat menjalar ke bagian bawahnya Gambar 2.3 Gejala serangan moler pada tanaman bawang merah (Bagus K.Udiarto, 2005).



Gambar 2.3 Gejala serangan moler pada tanaman bawang merah  
(Foto : Soetiarso)

### **2.2.2 Grayscale**

Secara umum untuk menghasilkan citra berwarna menjadi citra grayscale, konversi dilakukan dengan mengambil rata-rata dari nilai R, G, dan B sehingga menghasilkan nilai S sehingga nilai grayscale.

### **2.2.3 Template Matching**

Pengenalan citra penyakit daun bawang merah metode template matching adalah pengenalan dengan cara mencocokkan citra penyakit bawang merah yang di capture dengan menggunakan metode template matching, didalam prosesnya terdapat obyek yang siap dicapture serta template yang sudah tersedia dalam raspberry pi selanjutnya akan mencocokkan

nilai grayscale obyek dengan nilai *grayscale* template satu persatu dan kemudian akan membandingkan nilai *Array*, sehingga nantinya akan di cocokan dengan data yang ada dalam basis data, jika *record* dalam basis data terdapat informasi dari penyakit tersebut, maka basis data akan menampilkan informasi yang berkaitan dengan penyakit tersebut, maka sistem akan menghidupkan *output* salah satu lampu LED sesuai dengan penyakitnya. Contoh : Dalam melakukan proses klasifikasi menggunakan metode *Template Matching* terdapat beberapa tahap dimana yang menjadi masukan awal adalah hasil dari data obyek yang akan *digrayscale*. Hasil dari *Array* yang akan mempengaruhi penentuan jenis penyakit dari data *Template*. Proses dimulai dari data obyek yang akan *digrayscale*, hasil *grayscale* di *Array* selanjutnya akan memanggil satu persatu *Template* di *Matching*. Untuk menghitung kecocokan tiap piksel dapat menggunakan fungsi *CV\_TM\_CCOEFF\_NORMED* (OpenCV, *template\_matching*, 2018) yang tersedia dalam open cv dengan perhitungan:

$$R(x, y) = \frac{\sum_{x',y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x',y'} T'(x', y')^2 \cdot \sum_{x',y'} I'(x + x', y + y')^2}} \dots\dots\dots(1)$$

Keterangan :

T = Template

I = Image

X = Nilai Array koordinat X Template capture/ Obyek

Y = Nilai Array koordinat Y Template capture/ Obyek

X' = Nilai Array koordinat X' Template gambar

Y' = Nilai Array koordinat Y' Template gambar

Jika tingkat korelasi kecocokan diatas nilai *threshold* yang ditentukan maka citra yang diambil bisa dikatakan sesuai dengan templatnya. dan menentukan Hasil Tertinggi sehingga menghasilkan Hasil Klasifikasi Obyek.

### 2.2.4 Kamera *webcam*

*Webcam* adalah singkatan dari *web camera*. *Webcam* ini merupakan sebuah kamera digital yang bisa terhubung langsung ke *computer*. *Laptop* lazimnya sudah dilengkapi dengan



kamera *webcam* dan mikrofon, tetapi jika belum ada, kita dapat menambahkannya sendiri. Ada banyak macam jenis webcam, ada yang langsung ditancapkan di usb port pada *computer* dan ada juga melalui *wireless* (What is a webcam?, 2017)

Webcam yang digunakan di penelitian saya ini adalah webcam Logitech c270. Fitur di webcam ini yakni : dapat merekam video yang mempunyai kualitas HD 720p. webcam ini memiliki resolusi 3 megapixel dan memiliki mikrofon yang sudah include di dalamnya. Webcam ini memiliki fitur automatic light correction dan noise reduction. Port usbnya bertipe usb 2.0 (HD Webcam c270, 2010).



**Gambar 2.4** Logitech C270 HD Webcam

Sumber: Logitech (2010)

## 2.2.5 Raspberry pi

Raspberry pi adalah komputer mini berukuran cukup kecil kecil (hanya sebesar kartu kredit) dimana pada awalnya dirancang untuk edukasi. Sang pencipta raspberry pi yakni Eben Upton menyatakan bahwa tujuan dari pembuatan raspberry pi adalah untuk membuat atau menciptakan perangkat berbiaya rendah untuk meningkatkan skill pemrograman perangkat keras di tingkat pra-universitas. Karena ukuran Raspi yang kecil dan harganya yang terjangkau, produk ini sangat amat disukai oleh para developer elektronik yang proyeknya membutuhkan lebih dari mikrokontroller dasar. (What is a Raspberry Pi?, 2107)

Pada Penelitian saya , tipe raspberry pi yang saya gunakan adalah raspi 3 model b. Raspberry Pi 3 adalah generasi ketiga Raspberry Pi. Ia menggantikan Raspberry Pi 2 Model B pada bulan Februari 2016. Berikut ini adalah spesifikasi dari Raspberry Pi 3. (Pi3, 2018)

- A 1.2GHz 64-bit quad-core ARMv8 CPU
- 802.11n Wireless LAN

- Bluetooth 4.1
  - Bluetooth Low Energy (BLE)
  - 1GB RAM
  - 4 USB ports
  - 40 GPIO pins
  - Full HDMI port
  - Ethernet port
  - Combined 3.5mm audio jack and composite video
  - Camera interface (CSI)
  - Display interface (DSI)
  - Micro SD card slot (now push-pull rather than push-push)
  - VideoCore IV 3D graphics core
- (Raspberry Pi 3 Model B, 2016)



**Gambar 2.5** Raspberry Pi

Sumber: Raspberry 3 Model B (2016)

### 2.2.6 Open CV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah library atau pustaka computer vision dan machine learning yang bersifat open source. Library ini memiliki lebih dari 2500 algoritma yang telah dioptimasi. Algoritma OpenCV dapat digunakan untuk mendeteksi objek, tracking objek, face recognition dan lain lain. OpenCV dirilis dibawah lisensi BSD sehingga dapat digunakan secara bebas bahkan untuk komersil tanpa menulis kode sumbernya (team, 2017).

Versi OpenCV yang digunakan dalam penelitian ini adalah versi 2.4.13. versi ini merupakan cabang pengembangan dari versi 2.4 yang telah dilakukan bebarapa perbaikan *bug*. Contoh pada

penelitian Teknologi face recognition yang saat ini sudah sangat canggih dan diterapkan di hampir semua sistem mutakhir yang mengelola citra dan foto, contohnya situs-situs media sosial seperti Facebook dan Google Plus yang mampu mengenali wajah manusia dan bahkan memprediksi nama dari pemilik wajah manusia tersebut. Teknologi canggih itu dikembangkan dari beberapa pustaka-pustaka yang telah beredar secara umum. Contohnya OpenCV (Open-source Computer Vision).

Menurut [2], OpenCV atau Open-source Computer Vision adalah pustaka pemrograman yang digunakan untuk pemrosesan citra (image processing) secara real-time. Pustaka ini pertama kali dikembangkan di Intel Corporation tahun 2000 yang lalu sampai sekarang. Pustaka yang ditulis dalam bahasa pemrograman C++ ini telah tersedia dalam hampir semua bahasa pemrograman besar, seperti Python, Java dan MATLAB (Kandir, 2016).

OpenCV menggunakan algoritma Haar untuk melakukan analisis citra. Algoritma Haar sendiri merupakan salah satu dari sekian banyak algoritma matematis untuk menghitung sebuah bentuk dari suatu bagian citra yang diproses. Algoritma Haar sendiri berkembang dari teknologi face recognition pertama yang disebut Haar



**Gambar 2.6** Logo OpenCV

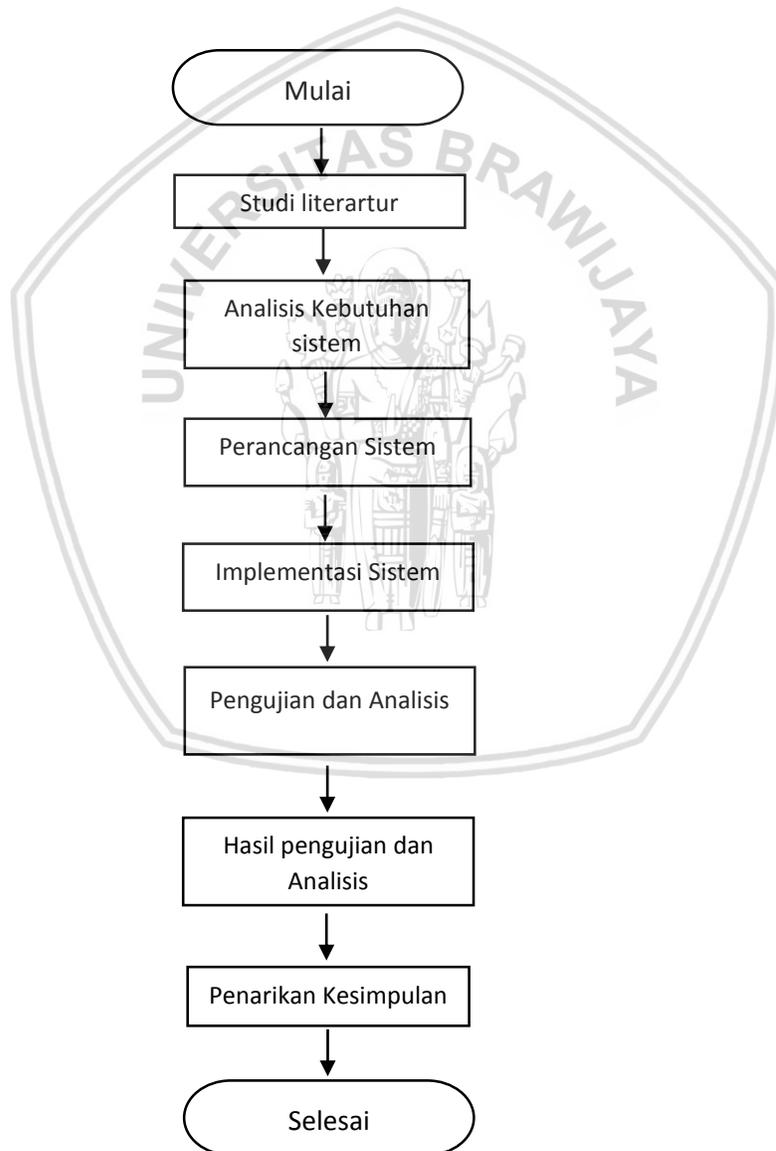
Sumber: (OpenCV, OpenCV, 2017)

## BAB 3 METODOLOGI

Bab ini menjelaskan metode yang digunakan dalam melakukan penelitian tentang sistem pendeteksi penyakit daun bawang merah pada citra hasil capture kamera menggunakan metode template matching. Tipe penelitian ini adalah implementatif dengan pendekatan perancangan melalui proses yang utuh mulai dari analisis perancangan hingga pengujiannya.

### 3.1 Alur metode penelitian

Alur metode penelitian yang dilakkan secara umum dapat dilihat dari diagram alir pada gambar.



Gambar 3.1 Diagram alir penelitian

Pada Gambar 3.1 merupakan diagram alir tahapan penulis dalam melakukan penelitian. Dimulai dari tahapan studi literatur, dimana pada tahap ini dilakukan pencarian sumber teori atau jurnal yang terkait dengan penelitian. Selanjutnya tahapan analisis kebutuhan, dimana pada tahapan ini dilakukan analisis terhadap apa saja kebutuhan fungsional maupun kebutuhan non-fungsional dari sistem yang dibuat. Selanjutnya tahap perancangan, yaitu pada tahap ini akan dilakukan perancangan sistem sesuai dengan dianalisis dan gambaran kebutuhan sistem pada tahap sebelumnya. Selanjutnya adalah tahapan implementasi, pada tahap ini dilakukan implementasi ke sistem yang telah dirancang, baik perangkat keras maupun perangkat lunak. Selanjutnya adalah tahapan pengujian dan analisis sistem yang telah selesai dibuat, pada tahap ini dilakukan beberapa skenario pengujian dan analisis sistem untuk mengetahui persentase keberhasilan dari hasil sistem yang dibuat. Hasil dari tahap pengujian ini akan dievaluasi apakah sistem sudah sesuai dengan rancangan dan kebutuhan yang diinginkan atau tidak. Jika memang tidak memenuhi maka proses penelitian akan diulang kembali ke tahap implementasi sistem dan selanjutnya akan diuji kembali dan seterusnya. Jika dirasa sistem sudah sesuai dengan rancangan dan kebutuhan yang diinginkan, maka dapat dilanjutkan ke tahap penarikan kesimpulan dan saran. Pada tahap ini berisi saran dan kesimpulan akhir mengenai dari hasil sistem yang dibuat, untuk masa depan penelitian selanjutnya.

### **3.2 Studi Literatur**

Studi literatur dilakukan untuk mencari referensi dari penelitian-penelitian sebelumnya sebagai pendukung penelitian yang akan dilakukan. Dari penelitian yang dibuat referensi dilakukan penyusunan teori-teori yang akan digunakan dalam penelitian dan perancangan sistem. Teori yang dapat dimanfaatkan, antara lain :

#### **1. Penyakit Daun Bawang Merah**

Studi literatur mengenai penyakit daun bawang merah dilakukan dengan mencari paper atau jurnal yang membahas penyakit daun bawang merah secara umum. Paper atau jurnal yang didapat bisa menjadi patokan bagi pengembangan teknologi yang terbaru, khususnya pada penelitian ini.

## 2. Mikrokontroler Raspberry Pi Pin GPIO

Studi literatur mengenai Mikrokontroler Raspberry Pi Pin GPIO dilakukan dengan mencari referensi dari web secara umum. Studi yang dilakukan bagaimana cara penggunaannya, dan bagaimana informasi mengenai pin-pin yang ada.

## 3. Open CV

Studi literatur mengenai Open CV juga dapat dilakukan dengan mencari referensi dari web resmi Raspberry Pi. Selain itu sebagai referensi tambahan terutama bagian pemrograman juga dapat dicari pada paper atau jurnal mengenai pemrograman dengan Python.

## 4. Raspberry Pi

Studi literatur mengenai Raspberry Pi dapat dilakukan dengan mencari referensi dari web resmi Raspberry Pi. Selain itu pada bagian pemrograman dan mengenai pin-pin raspberry Pi juga dapat dicari pada paper atau jurnal .

### 3.3 Analisa Kebutuhan

Analisa kebutuhan ditujukan untuk melakukan analisis pada beberapa kebutuhan yang diperlukan sistem pada penelitian. Analisa kebutuhan pada penelitian dibagi menjadi 2, yaitu kebutuhan Fungsional dan Non-fungsional.

#### 3.3.1 Kebutuhan Pengguna

Kebutuhan pengguna dalam penelitian ini adalah sebagai berikut:

- a) Sistem mampu menampilkan citra yang menampilkan penyakit daun bawang merah
- b) Sistem mampu menampilkan hasil pengenalan penyakit daun bawang merah dalam bentuk teks

#### 3.3.2 Kebutuhan Sistem

Kebutuhan sistem yang diperlukan untuk membangun sistem ini adalah sebagai berikut :

- a) Sistem dapat mengcapture citra yang mengandung penyakit daun bawang merah
- b) Sistem dapat mendeteksi penyakit daun bawang merah
- c) Sistem dapat mengkonversi penyakit daun bawang merah dari citra ke teks

#### 3.3.3 Kebutuhan Perangkat Keras

Kebutuhan Perangkat Keras yang diperlukan untuk membangun sistem ini adalah sebagai berikut:

- a) Perangkat Komputer/ Laptop
- b) Kamera Webcam
- c) Raspberry pi 3 model B
- d) Trigger
- e) Lampu LED 3 buah
- f) Lampu USB 1

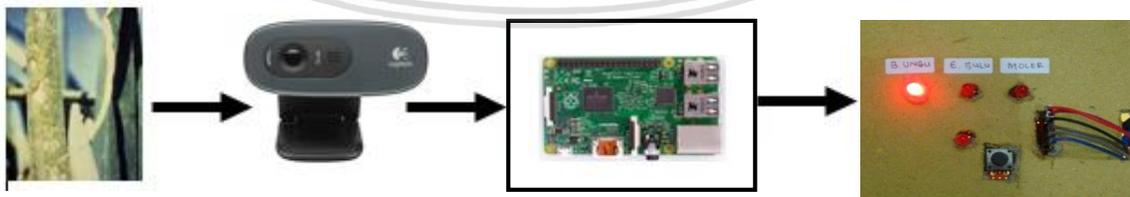
### 3.3.4 Kebutuhan Perangkat Lunak

Kebutuhan Perangkat lunak yang diperlukan untuk membangun sistem ini adalah sebagai berikut:

- a. Vnc Open CV
- b. OS Windows 8
- c. Bahasa Pemrograman Python

### 3.4 Perancangan Sistem

Perancangan sistem dilakukan apabila semua kebutuhan yang diperlukan oleh sistem sudah terpenuhi. Dalam sistem ini pengolahan data menggunakan raspberry pi . Citra penyakit daun bawang merah *dicapture* menggunakan kamera *webcam* dengan jarak ketinggian dan sudut yang telah ditentukan. Menggunakan kamera webcam supaya alat lebih mudah dibawa kemana saja daripada menggunakan kamera dari laptop. Dan hasil *capture* dikirim ke *raspberry pi* sebagai mikrokomputer yang akan melakukan proses pengolahan citra dan pencocokan template pada hasil capture. Kemudian hasilnya akan ditampilkan pada salah satu lampu LED .



Gambar 3.2 Gambaran umum sistem

### 3.5 Implementasi Sistem

Pada implementasi sistem, hal pertama yang dilakukan adalah pengerjaan perangkat keras. Setelah semua Kebutuhan kebutuhan perangkat keras terpenuhi, selanjutnya pengerjaan perangkat lunak untuk mengolah data yang telah dideteksi. Pada tahap akhir dilakukan integrasi

antara perangkat keras dan perangkat lunak dan dilakukan konfigurasi sistem sehingga bisa saling terhubung.

### 3.6 Pengujian dan Analisis

Pada tahapan pengujian sistem, parameter yang dilakukan dengan perancangan sistem. Beberapa perancangan untuk skenario pengujian adalah sebagai berikut:

- 1) Pengujian akurasi pendeteksian penyakit bawang merah dari hasil capture kamera dari beberapa sudut.
- 2) Pengujian terhadap akurasi ketepatan pengenalan pola karakter pada penyakit bawang merah
- 3) Pengujian terhadap performansi yang mampu dilakukan sistem dalam mengolah data

### 3.7 Kesimpulan

Kesimpulan sistem pada penelitian ini didapat apabila seluruh pengujian sistem dari perancangan telah dilakukan. Selanjutnya hasil dari pengujian sistem dari perancangan telah dilakukan. Selanjutnya hasil dari pengujian sistem secara garis besar dijadikan kesimpulan. Dan pada tahap ini juga diambil beberapa saran untuk memperbaiki penelitian ini dan sebagai dasar untuk orang yang akan melakukan penelitian serupa.

## BAB 4 REKAYASA KEBUTUHAN

Pada bab ini akan dijelaskan secara rinci terkait gambaran umum sistem, analisis kebutuhan fungsional, kebutuhan perangkat keras, kebutuhan perangkat lunak dan batasan desain sistem.

### 4.1 Gambaran Umum Sistem

Sistem Pendeteksi Penyakit Daun Bawang Merah Probolinggo Menggunakan *Template Matching* Berbasis *Raspberry Pi* merupakan suatu sistem yang dapat menentukan suatu penyakit bawang merah dengan parameter berupa pencocokan nilai grayscale obyek dan nilai grayscale *template* serta bentuk pada daun. Dibuat dengan perancangan sistem yang terdiri dari kamera *webcam C270*, *Raspberry Pi*, Lampu Led serta tombol Trigger untuk memberikan inputan. Adapun bentuk dan warna ini akan dibaca dengan *webcam C270*, dan akan diproses ke *raspberry pi 3 model B* Berdasarkan nilai dari kedua parameter tersebut maka dapat dilakukan klasifikasi hasilnya dengan menggunakan metode *Template Matching*. Sistem ini membantu menyelesaikan permasalahan penentuan tingkat suatu penyakit daun bawang merah secara otomatis, dimana selama ini faktor penentuan tingkat suatu penyakit daun bawang merah hanya dilakukan berdasarkan secara tanda-tanda yang ada. Dan hal itu menyebabkan tingkat objektifitas yang rendah. Selain itu penggunaan metode *Template Matching* menjadi metode yang tepat karena dapat menghasilkan tingkat akurasi yang tinggi sesuai dengan peluang jumlah fakta yang dianggap benar berdasarkan data yang sebenarnya. Hasil dari pengolahan sistem ini akan secara otomatis ditampilkan pada lampu Led serta dimonitor.

### 4.2 Analisis Kebutuhan Sistem

Analisis kebutuhan sistem dilakukan untuk menggali semua kebutuhan yang diperlukan untuk Sistem Pendeteksi Penyakit Daun Bawang Merah. Dalam melakukan analisis kebutuhan sistem terdiri atas beberapa kebutuhan yang perlu dijabarkan yakni kebutuhan fungsional dan kebutuhan non fungsional, dimana kebutuhan non fungsional terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak.

#### 4.2.1 Kebutuhan Fungsional

Berikut ini adalah kebutuhan fungsional yang harus mampu dilakukan oleh sistem:

1. Kamera Webcam C270 dapat mengcapture data yang dideteksi. Kamera Webcam c270 bertugas dapat mengcapture data yang dideteksi penyakit daun bawang merah.
2. Data dari raspberry pi dapat diolah dapat mengklasifikasikan hasilnya menjadi jenis penyakit daun bawang merah yang berbeda berdasarkan metode *Template Matching*. Pada fungsionalitas ini bertugas dapat melakukan proses klasifikasi dengan menggunakan metode *Template Matching* berdasarkan *input* yang diperoleh dari pembacaan webcam dan raspberry pi serta dari data latih yang ditanamkan pada pemrograman *Template Matching* tersendiri.
3. Sistem dapat menampilkan hasil klasifikasi jenis penyakit pada lampu Led serta monitor. Pada fungsionalitas ini bertugas untuk dapat menampilkan hasil *output* dari sistem yaitu hasil klasifikasi jenis penyakit pada salah satu lampu Led.

#### 4.2.2 Kebutuhan Non Fungsional

Kebutuhan non fungsional dari sistem ini terdiri dari kebutuhan perangkat keras dan kebutuhan perangkat lunak yang dijelaskan secara rinci dibawah ini.

##### 4.2.2.1 Kebutuhan Perangkat Keras

Guna mendukung implementasi pembuatan sistem dari sisi perangkat keras maka di perlukan beberapa alat yang dijelaskan berikut ini:

###### 1. Raspberry Pi 3 Model B

Raspberry Pi 3 Model B sebagai otak pengolahan informasi dalam perancangan sistem.

###### 2. Kamera Webcam C270

Kamera Webcam ini untuk membaca atau mengcapture penyakit daun bawang merah

###### 3. Kabel LAN

Kabel LAN pada sistem ini dibutuhkan untuk menghubungkan antara perangkat satu dengan perangkat lainnya.

###### 4. Laptop

Laptop dalam hal ini berfungsi sebagai media untuk membuat program untuk mikrokontroler selain itu berfungsi sumber daya dari sistem yang akan dibuat.

Adapun spesifikasi laptop yang digunakan antara lain :

- Model perangkat : TOSHIBA SATELLITE L740
- Prosesor : AMD A6-3420M, 1.50GHz
- Sistem Operasi : Window 8 64bit

#### **4.2.2.2 Kebutuhan Perangkat Lunak**

Pada bagian ini terdapat kebutuhan perangkat lunak yang dibutuhkan oleh sistem pada penelitian ini yaitu Open CV python yang merupakan IDE yang dibutuhkan untuk mengoperasikan mikrokontroler Raspberry Pi dan untuk menjalankan program lainnya yang dibutuhkan dalam pembuatan program dalam penelitian ini. Selain itu Open CV python ini terdapat fitur library yang berguna memudahkan dalam membuat program.

#### **4.2.2.3 Kebutuhan Prototype**

Pada kebutuhan prototype ini yang dibutuhkan oleh sistem pada penelitian sistem pendeteksi penyakit daun bawang merah berupa posisi jarak antara kamera webcam dengan obyek sejauh 28cm, sistem ini juga memerlukan Box supaya tidak terkena cahaya dari luar dikarenakan jika ada cahaya masuk dari luar maka sistem tidak akan berjalan sesuai dengan keinginan. Dan didalam Box terdapat kotak kecil sebagai tempat obyek yang akan diteliti.

### **4.3 Batasan Desain Sistem**

Dalam pembuatan Sistem Pendeteksi Dehidrasi ini terdapat beberapa batasan sehingga lingkup pembahasan, perancangan, maupun pengimplementasiannya tidak terlalu luas. Adapun batasan-batasan desain sistem ini adalah sebagai berikut :

1. Sistem melakukan klasifikasi terhadap 3 jenis penyakit pada daun bawang merah yaitu penyakit bercak ungu, embun bulu dan moler
2. Sebelum sistem digunakan, sistem harus diaktifkan terlebih dahulu untuk mengkalibrasi Raspberry Pi sebelum digunakan.

## BAB 5 PERANCANGAN DAN IMPLEMENTASI

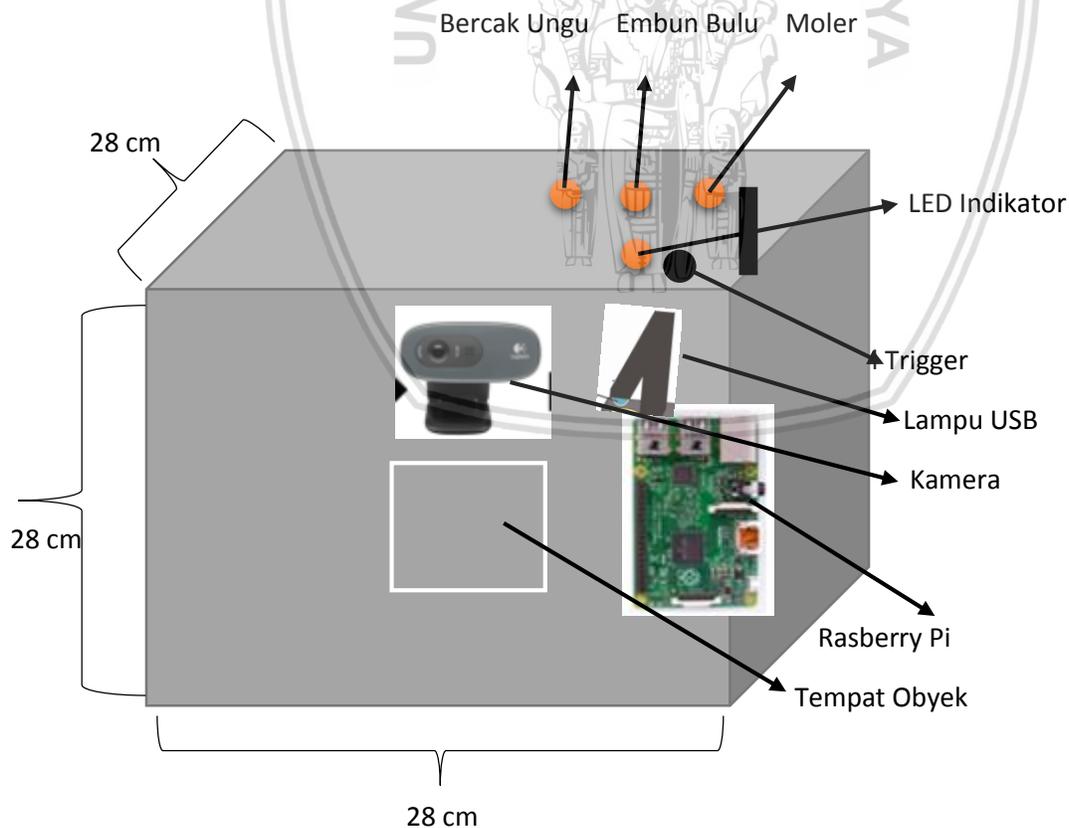
Bagian dari bab 5 ini menjelaskan mengenai proses perancangan dan implementasi sistem secara terperinci baik perancangan dan implementasi dari sisi perangkat keras maupun dari sisi perangkat lunak.

### 5.1 Perancangan Sistem

Dalam sub-bab ini akan menjabarkan cara perancangan sistem dimulai dari perancangan *prototype* alat, perancangan perangkat keras hingga perancangan perangkat lunak.

#### 5.1.1 Perancangan *Prototype* Alat Pendeteksi Penyakit Daun

Dalam melakukan desain *prototype* dari Sistem Pendeteksi Penyakit Daun Bawang Merah ini perlu diperhatikan peletak tiap-tiap komponen serta ukuran alat yang akan dikembangkan. Pembuatan desain sistem dirancang untuk menggambarkan bentuk *prototype* yakni berupa kotak hitam berukuran panjang 28cm, lebar 28 cm dan tinggi 28cm. Bentuk *prototype* alat ditunjukkan pada Gambar 5.1.

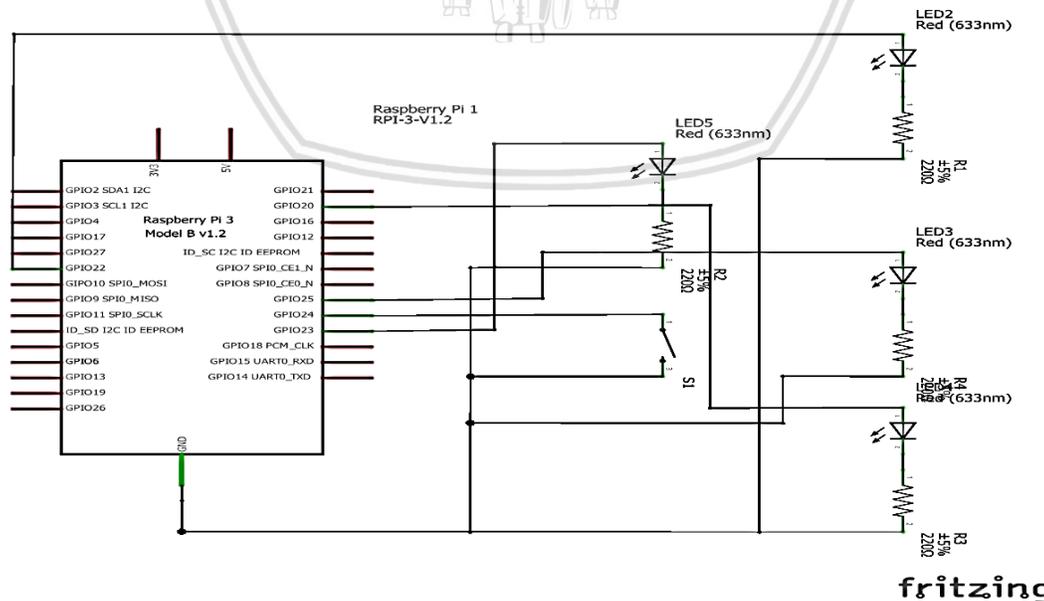


Gambar 5.1 Desain *Prototype* Alat Pendeteksi Daun Bawang Merah

Posisi kamera diletakkan di dalam kotak untuk menghindari interferensi cahaya lain dari objek yang akan dibaca dimana dalam hal ini adalah daun bawang. Lampu USB diletakkan didalam kotak sebelah kiri bersama Raspberry Pi dimana lampu sebagai alat penerangan untuk obyek yang dideteksi. Kamera ditempatkan didalam tidak jauh dengan raspberry, sebelah atas menghadap kebawah untuk deteksi obyek. LED output diletakkan dibagian luar atas untuk mempermudah pengguna melihat hasilnya. Selain itu di permukaan atas dari kotak juga diletakkan *push button* yang digunakan untuk men-*trigger* capture kamera yang akan diolah.

### 5.1.2 Perancangan Perangkat Keras

Perancangan perangkat keras dilakukan berdasarkan analisis kebutuhan perangkat keras serta spesifikasi dari masing-masing dari perangkat keras agar dapat membangun sistem sesuai dengan yang diharapkan. Berdasarkan Gambar 3.2 yang merupakan blok diagram sistem maka pada tahap perancangan perangkat keras ini menjabarkan secara rinci hubungan skematik pin-pin yang digunakan antara tiap komponen perangkat keras, dimana dalam hal ini kamera dan obyek daun bawang merah yang merupakan input dihubungkan dengan Mikrokontroler Raspberry Pi sebagai pengolah data sehingga hasil olahan datanya dapat ditampilkan melalui lampu LED dan tampilan dimonitor. Skematik diagram perancangan perangkat keras Sistem Pendeteksi Penyakit daun bawang merah ini ditunjukkan pada Gambar 5.2.



Gambar 5.2 Diagram Skematik Sistem

Pada diagram skematik diatas terlihat bahwa ada *push button* yang fungsinya telah dijelaskan pada deskripsi sistem. Rangkaian *push button* salah satu kakinya dihubungkan dengan pin GPIO 24 pada Rasberry Pi sedangkan pada kaki lainnya dihubungkan ke ground. Selanjutnya rangkaian LED dijelaskan terkait koneksi antar pinnya pada Tabel 5.1.

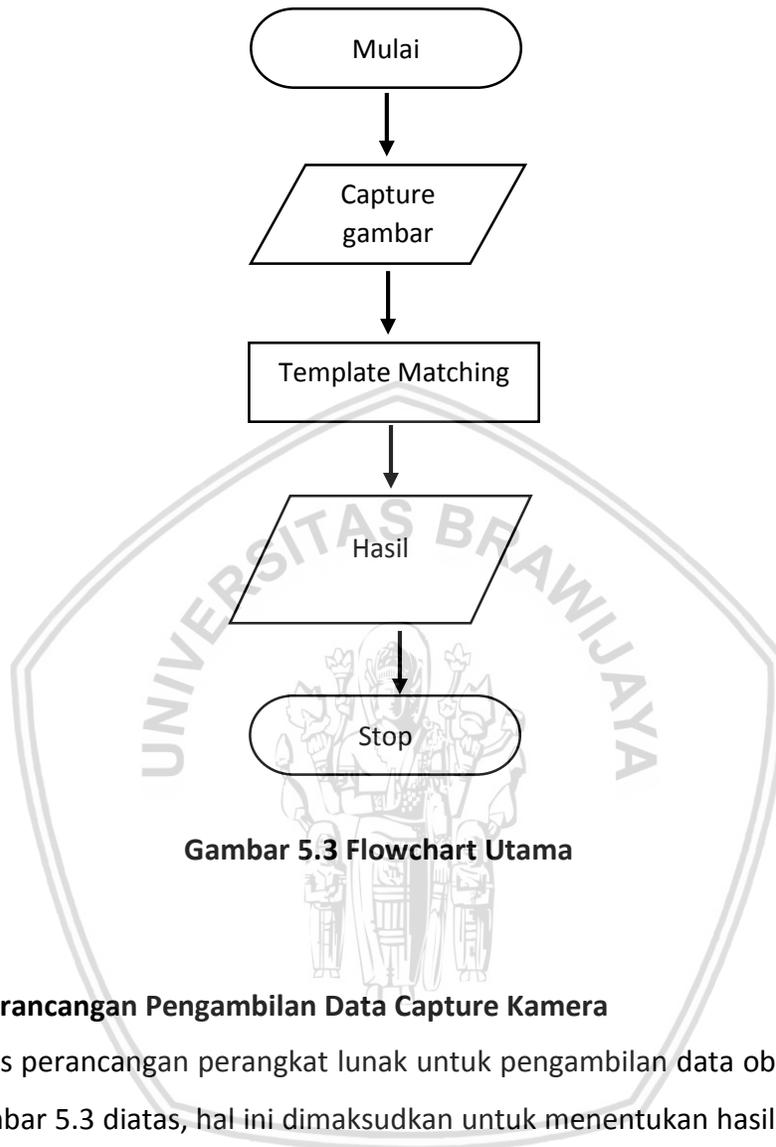
**Tabel 5.1 Keterangan koneksi pin Rasberry Pi GPIO**

Pin Rasberry Pi	Pin GPIO
6	GND
15	GPIO 22
16	GPIO 23
18	GPIO 24
22	GPIO 25
37	GPIO 26

Berdasarkan Tabel 5.1 tersebut terlihat bahwa dari 40 pin Rasberry Pi yang disambungkan secara langsung ke pin GPIO hanya 6 pin yaitu pin 6, 15, 16, 18, 22, dan 37. Untuk pin 6 dan 16 dan 18 dihungkan ke GND serta GPIO 23, GPIO 24 untuk mengaktifkan Triger dan lampu indicator Triger. Fungsi pin 15 disambungkan secara langsung ke GPIO 22 untuk mengaktifkan LED 1. Selanjutnya untuk fungsi pin 22 dan fungsi pin 37 untuk mengaktifkan lampu LED 2 beserta LED 3 untuk dapat menampilkan hasil dari output objek.

### 5.1.3 Perancangan Perangkat Lunak

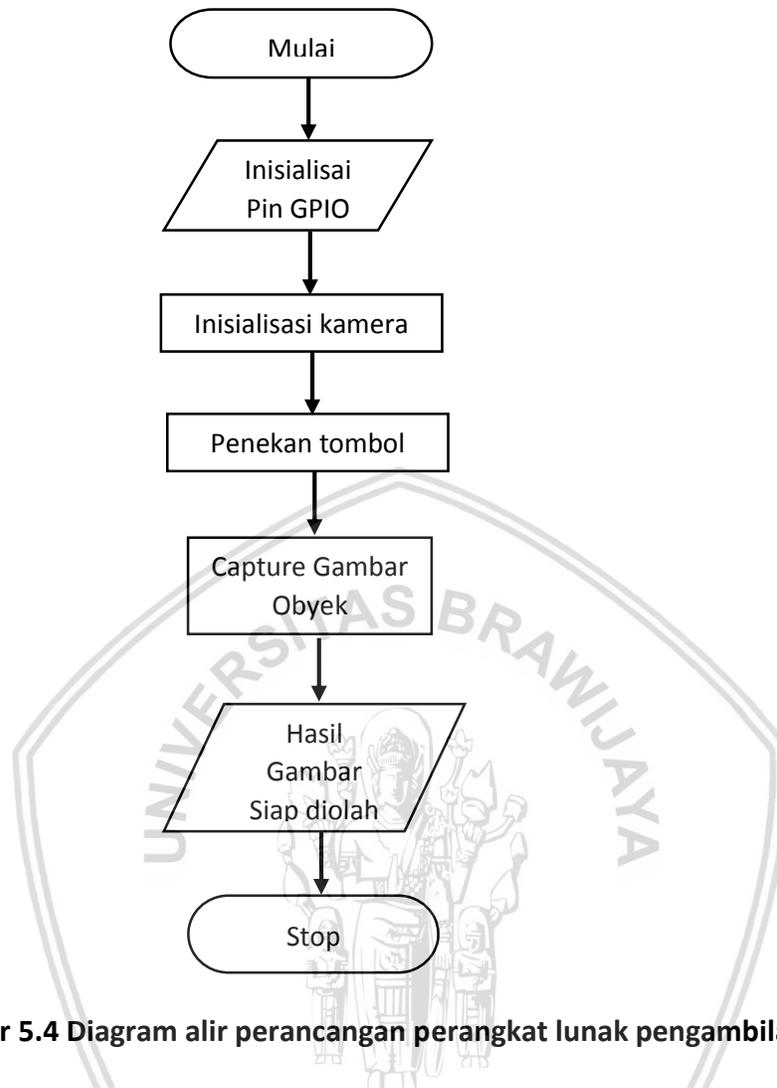
Pada sub bab perancangan perangkat lunak dibagi menjadi 2 pembahasan, yakni perancangan perangkat lunak pada mikrokontroler untuk mengambil data capture obyek yang akan diolah melalui pencocokan template yang diubah dari Grayscale untuk menentukan Array dari obyek dan akan dibandingkan dengan Array dari salah satu Template yang sudah tersedia di Rasberry Pi, dimana akan memilih satu dari 30 template yang tersedia untuk dijadikan output dengan nilai tertinggi hasil dari pencocokan nilai akhir Array dengan threshold 80%, sehingga jika melebihi 80% maka template tersebut yang akan dijadikan ouputan akhir. Sebaliknya apabila kurang dari 80% maka template tersebut tidak akan dijadikan outputan. Selanjutnya perancangan perangkat lunak untuk melakukan proses klasifikasi nilai akhir dengan menggunakan metode *Template Matching*, yang digunakan dalam penelitian ini.



Gambar 5.3 Flowchart Utama

#### 5.1.3.1 Perancangan Pengambilan Data Capture Kamera

Proses perancangan perangkat lunak untuk pengambilan data obyek ditunjukkan pada Gambar 5.3 diatas, hal ini dimaksudkan untuk menentukan hasil pembacaan data obyek yang akan diolah untuk dilakukan klasifikasi dengan menggunakan Template Matching. Sistem dimulai dengan melakukan inialisasi pin dari GPIO pada Rasberry Pi yang bertujuan untuk membedakan input dan output yang akan dibaca oleh masingmasing pin. Selanjutnya jika sudah dapat dibaca oleh mikrokontroler maka dilakukan proses pengecekan penekanan *push button*. Apabila terjadi penekanan *push button*, maka kamera akan mengcapture obyek sehingga akan digunakan untuk perhitungan Template Matching yang akan diuji.



**Gambar 5.4 Diagram alir perancangan perangkat lunak pengambilan data Obyek**

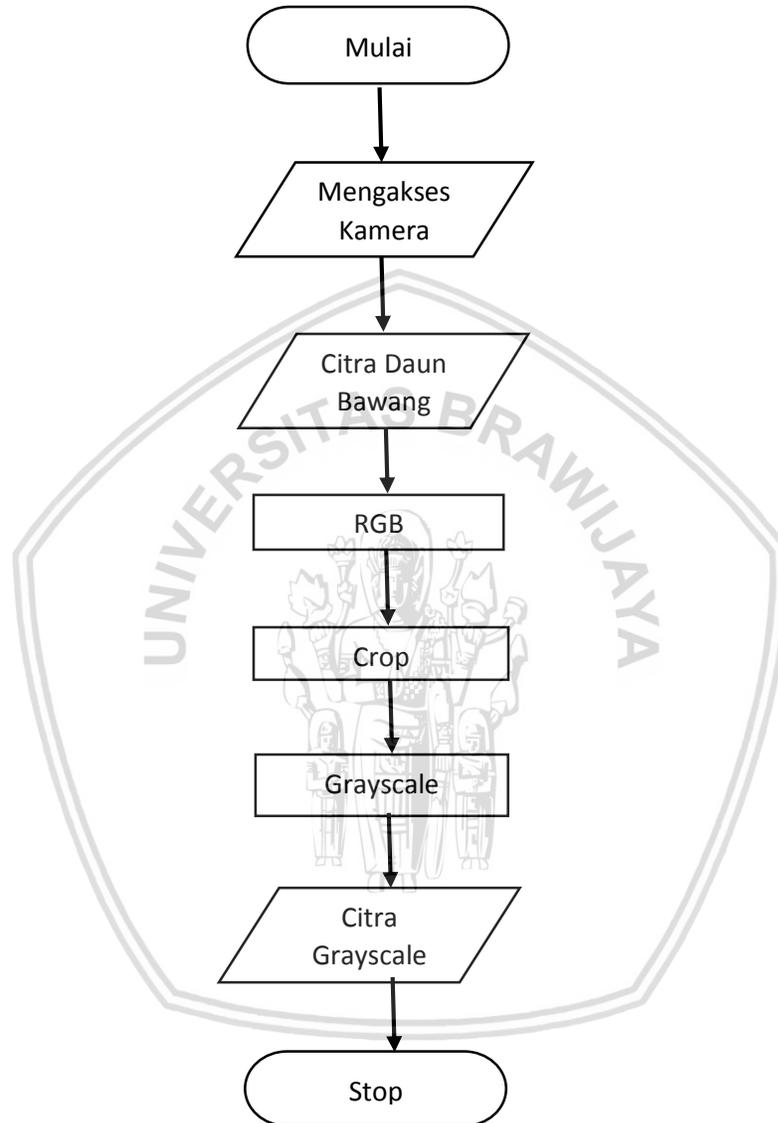
### 5.1.3.2 Perancangan Pembuatan Template

Pada gambar 5.4 merupakan proses pembuatan template. Proses pembuatan template dimulai dengan mengakses kamera kemudian mengambil citra penyakit daun bawang yaitu

1. Bercak Ungu sebanyak 20 template
2. Embun Bulu sebanyak 20 template
3. Moler sebanyak 20 template

Seperti yang sudah dijelaskan pada gambar 5.1. Setelah gambar didapat lalu citra diubah dari RGB dan grayscale. Citra RGB yang sudah di ubah menjadi grayscale akan disimpan sebagai database template. Jika masuk proses terakhir maka nantinya akan

mengeluarkan outputan pada salah satu Led yang akan menyala, sesuai dengan kelasnya.



**Gambar 5.5 Diagram alir perancangan Pembuatan Template**

### 5.1.3.2 Perancangan Klasifikasi Metode *Template Matching*

Dalam melakukan proses klasifikasi menggunakan metode *Template Matching* terdapat beberapa tahap sesuai dengan Gambar 5.5 dimana yang menjadi masukan awal adalah hasil dari data obyek yang akan digrayscale. Hasil dari Array yang akan mempengaruhi penentuan jenis penyakit dari data Template. Ada beberapa kelas penyakit yaitu template bercak ungu

dimana daun bercak ungu ujung daun mengkuning sampai pertengahan daun, embun bulu daun ini berwarna kuning dari ujung sampai kebawah, dan moler daunnya memiliki bentuk yang terpelintir. Proses dimulai dari data obyek yang akan digrayscale, hasil grayscale di Array selanjutnya akan memanggil satu persatu Template di Matching. Untuk menghitung kecocokan tiap piksel dapat menggunakan fungsi CV\_TM\_CCOEFF\_NORMED yang tersedia dalam open cv dengan perhitungan:

$$R(x, y) = \frac{\sum_{x',y'} (T'(x', y') \cdot I'(x + x', y + y'))}{\sqrt{\sum_{x',y'} T'(x', y')^2 \cdot \sum_{x',y'} I'(x + x', y + y')^2}} \dots\dots\dots(2)$$

Keterangan :

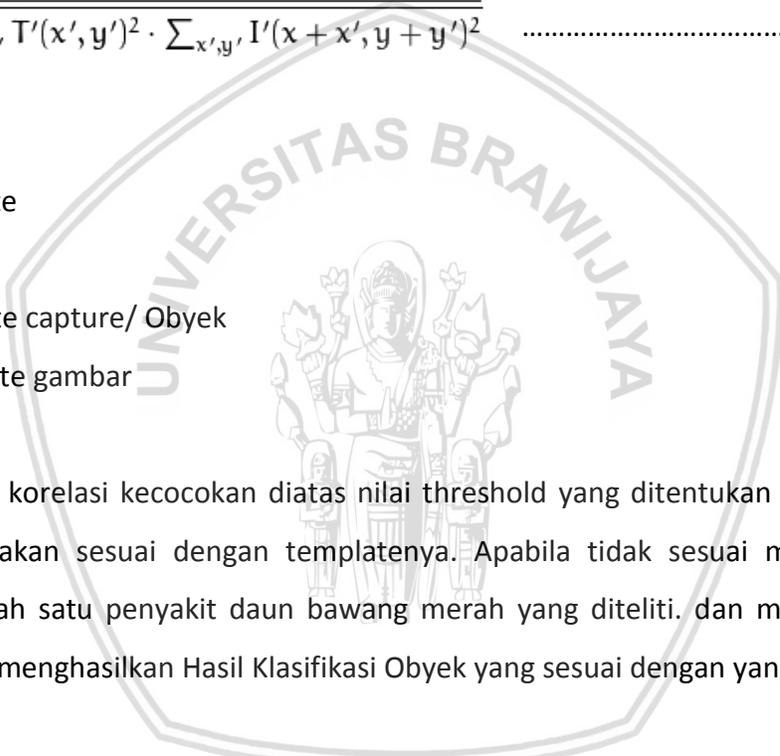
T = Template

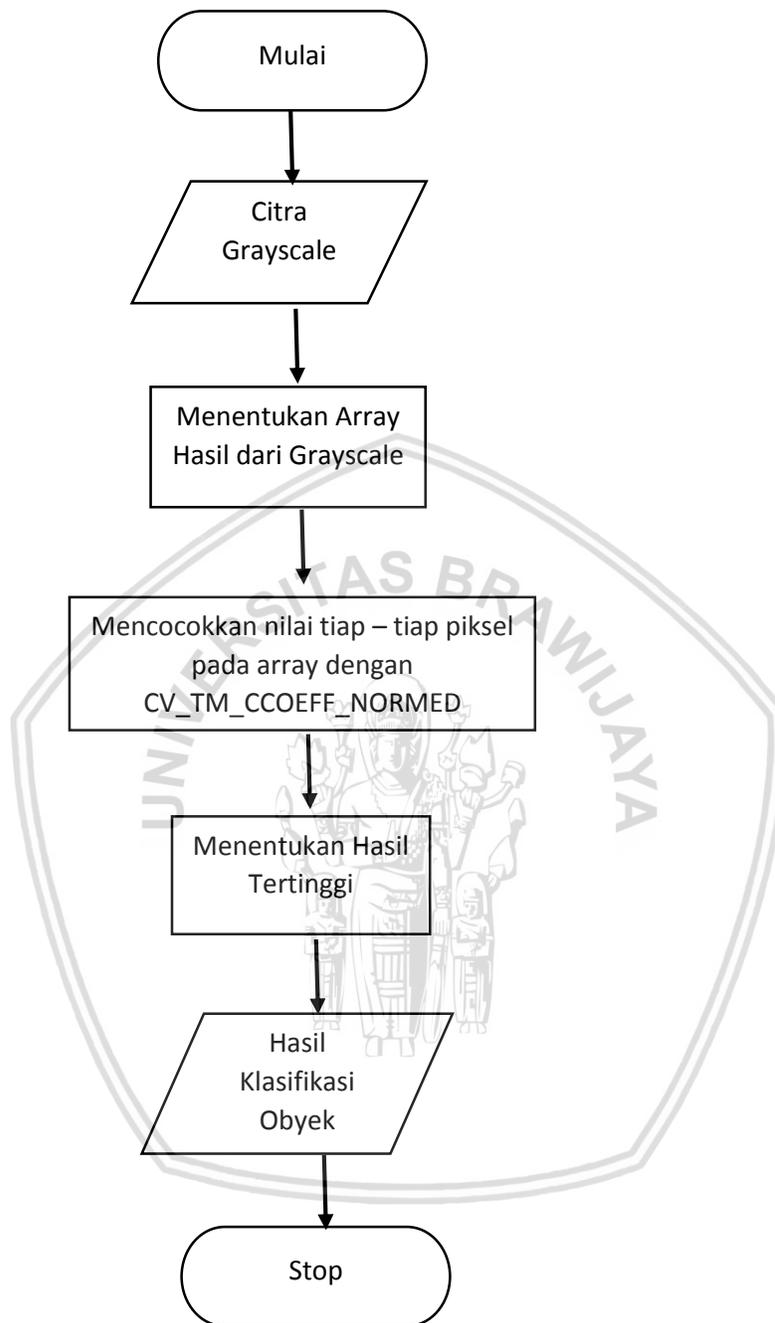
I = Image

X = Template capture/ Obyek

X' = Template gambar

Jika tingkat korelasi kecocokan diatas nilai threshold yang ditentukan maka citra yang diambil bisa dikatakan sesuai dengan templatnya. Apabila tidak sesuai maka tidak akan terdeteksi dari salah satu penyakit daun bawang merah yang diteliti. dan menentukan Hasil Tertinggi sehingga menghasilkan Hasil Klasifikasi Obyek yang sesuai dengan yang diinginkan.



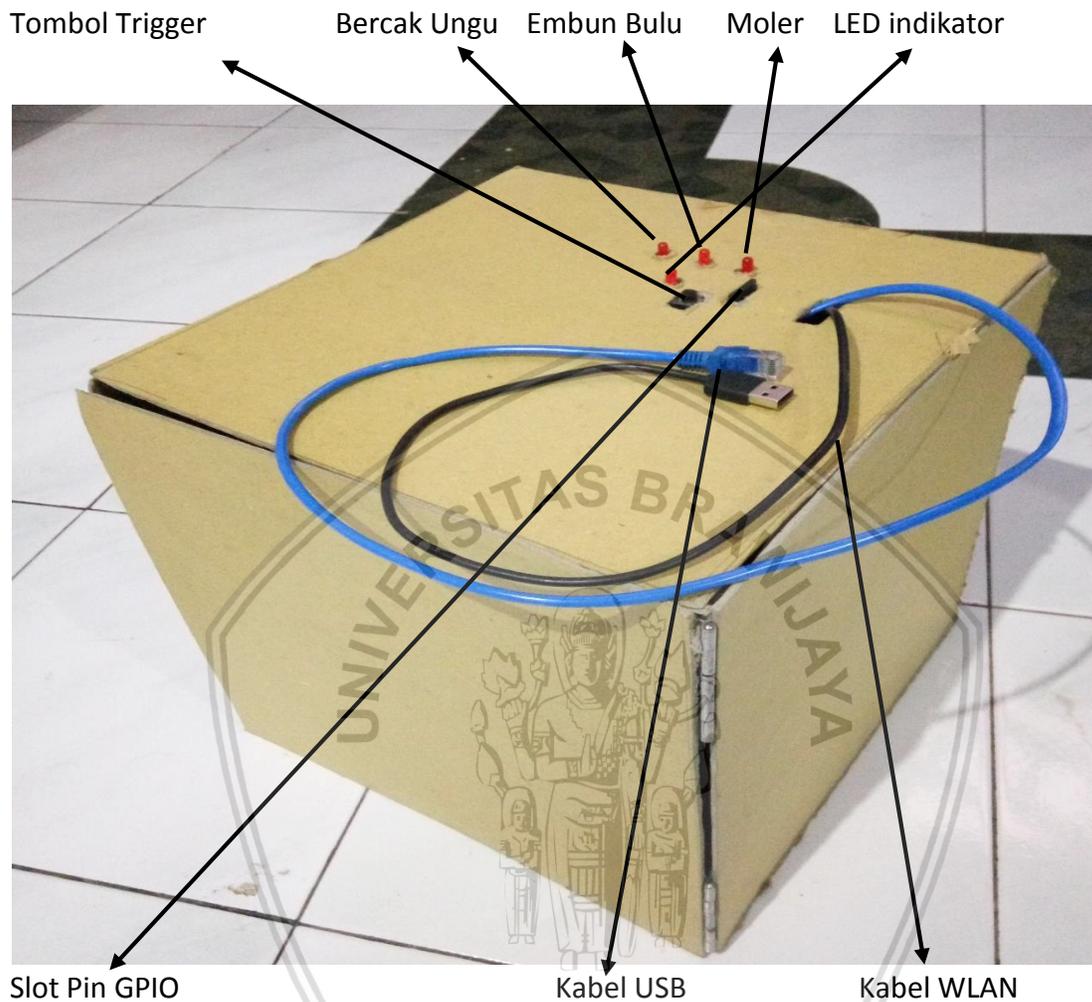


Gambar 5.6 Diagram alir perancangan klasifikasi *Template Matching*

## 5.2 Implementasi Sistem

Implementasi sistem merupakan tahap untuk merealisasikan pembuatan sistem berdasarkan semua perancangan yang telah dilakukan sebelumnya. Pada subbab ini menjelaskan

satu per satu secara rinci terkait implementasi *prototype*, implementasi perangkat keras serta implementasi perangkat lunak.



**Gambar 5.7 Implementasi Prototype Alat Pendeteksi Penyakit Daun**

### 5.2.1 Implementasi *Prototype* Alat Pendeteksi Penyakit Daun Bawang Merah

Dalam mengimplementasikan *prototype* alat pendeteksi daun bawang merah ini mengacu pada perancangan di subbab 5.1.1 yakni menggunakan kotak coklat berukuran  $28 \times 28 \times 28 \text{ cm}^3$ . Bahan yang digunakan untuk membuat kotak adalah karton coklat dengan ketebalan sebesar 2mm. Hasil implementasi *prototype* beserta peletakan komponen elektronik yang digunakan ditunjukkan pada Gambar 5.6 berikut.

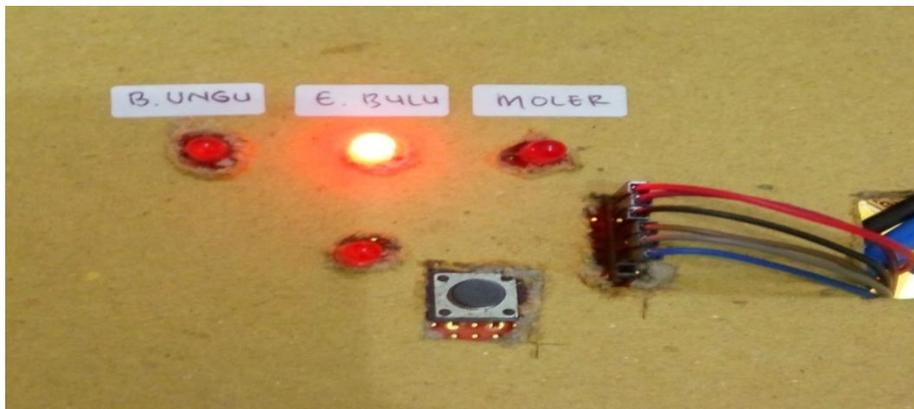
## 5.2.2 Implementasi Perangkat Keras

Tahap ini menjelaskan proses pengimplementasian perangkat keras yang mencakup komponen elektronik antara lain Raspberry Pi, kamera, *push button* dan LED. Keseluruhan komponen dirangkai menjadi satu dengan menggunakan PCB yang dihubungkan berdasarkan perancangan yang telah dijelaskan pada subbab 5.1.2



**Gambar 5.8 Implementasi rangkaian Raspberry Pi dengan kamera**

Pada Gambar 5.8 menunjukkan hasil implementasi kedua jenis kamera dan Raspberry Pi, dimana kamera dan raspberry pi yang telah dirangkai sesuai dengan perancangan diletakkan dibagian dalam dari kotak seperti gambar sebelah kiri, sedangkan led yang dirangkai pada PCB diletakkan dibagian atas yang dihubungkan melalui kabel *jumper*.



**Gambar 5.9 implementasi rangkaian LED, jumper dan *push button***

Berdasarkan perancangan perangkat keras LED dan *push button* yang telah dijelaskan pada subbab 5.1.2, hasil dari implementasinya ditunjukkan pada Gambar 5.9 diatas.

### 5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan proses realisasi program untuk Sistem Pendeteksi Penyakit Daun ini berdasarkan perancangan yang telah dilakukan pada subbab 5.1.3. Dalam melakukan implementasi perangkat lunak ini sepenuhnya proses pengkodean program dilakukan pada Raspberry Pi Open CV Python dimana diawal program dilakukan inialisasi *library* yang digunakan untuk mempermudah pemrograman beberapa fungsi tertentu. Pada Tabel 5.2 ditunjukkan pengimplementasian *library* pada sistem ini, diantaranya adalah *library* “Rpi.GPIO as GPIO” untuk penggunaan Pin GPIO, *library* “time” untuk penggunaan waktu delay, dan *library* “cv2” untuk mengaktifkan opencv2, *library* “os” untuk os open cv python, dan *library* numpy as np untuk perhitungan array dalam bentuk matrix.

**Tabel 5.2 Kode pemrograman inialisasi *library* Sistem Pendeteksi Penyakit Daun**

Baris	Kode Program
1	<code>import RPi.GPIO as GPIO</code>
2	<code>import time</code>
3	<code>import cv2</code>
4	<code>import os</code>
5	<code>import numpy as np</code>

#### 5.2.3.1 Implementasi Kode Program

Dalam melakukan implementasi untuk mengcapture obyek dari kamera tentunya perlu dilakukan inialisasi variabel dan konfigurasi pin sensor berdasarkan perancangan perangkat keras dan perancangan perangkat lunak yang telah dijelaskan pada subbab sebelum-sebelumnya. Tabel 5.3 menunjukkan pada baris ke-1 untuk inialisasi pin untuk setmode GPIO, baris ke-2 hingga baris ke-6 untuk inialisasi pin GPIO ke LED1, LED2, dan LED3, baris ke-9 hingga baris ke-11 untuk inialisasi pin push button, baris ke-12 hingga baris ke-18 untuk inialisasi pin LED indicator dengan filename hasil capture ke raspberry dengan nama img2.jpg, baris ke-20 hingga baris ke-27 untuk inialisasi pembacaan hasil capture yang kemudian digrayscale, selanjutnya akan memanggil template dalam bentuk array, dan masuk keperbandingan TemplateMatching sehingga menghasilkan nilai tertinggi.



Tabel 5.3 Kode pemrograman inisialisasi pin GPIO dan Capture Kamera

Baris	Kode Program
1	GPIO.setmode (GPIO.BCM)
2	GPIO.setup(23, GPIO.IN, pull_up_down=GPIO.PUD_UP)
3	GPIO.setup(24, GPIO.OUT)
4	GPIO.setup(25, GPIO.OUT)
5	GPIO.setup(22, GPIO.OUT)
6	GPIO.setup(26, GPIO.OUT)
7	GPIO.setwarnings (False)
8	
9	Try:
10	While True :
11	Button= GPIO.input (23)
12	If button == False:
13	GPIO.output (24, True)
14	Os.system("sudo rm img2.jpg")
15	Def main ():
16	cam= cv2.VideoCapture(0)
17	Frame=cam.read() [1]
18	Cv2.imwrite(filename= 'img2.jpg', img=frame)
19	
20	img = cv2.imread("img2.jpg")
21	gray_img = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
22	template = cv2.imread("k.jpg",Cv2.IMREAD_GRAYSCALE)
23	a,b = template.shape[:-1]
24	result=cv2.matchTemplate(gray_img,template,cv2.TM_ccoeff_NORMED)
25	loc=np.where(result>=0.8)
26	templatel=cv2.imread("1a.jpg", cv2.IMREAD_GRAYSCALE)
27	c,d=templatel.shape[:-1]



**Gambar 5.10 Hasil Gambar Capture Kamera**

Pada gambar 5.10 merupakan hasil gambar yang telah capture menggunakan webcam Logitech C270. Gambar tersebut masih bercitra RGB sehingga perlu diubah ke gray agar dapat diproses dengan menggunakan template matching

Pada tahap implementasi kode program pengambilan template ditujukan pada Tabel 5.11 bertujuan untuk merealisasikan kode program pengambilan template. Untuk membuat template ini kita harus mengumpulkan beberapa gambar dari penyakit daun bawang merah yang akan kita gunakan untuk mendeteksi jenis penyakit daun lalu mengubahnya kedalam citra grayscale. Jika diujikan dengan sudut obyek citra uji tidak sama dengan template maka tidak akan terdeteksi.



**Gambar 5.11 Hasil Gambar Template RGB Bercak Ungu**

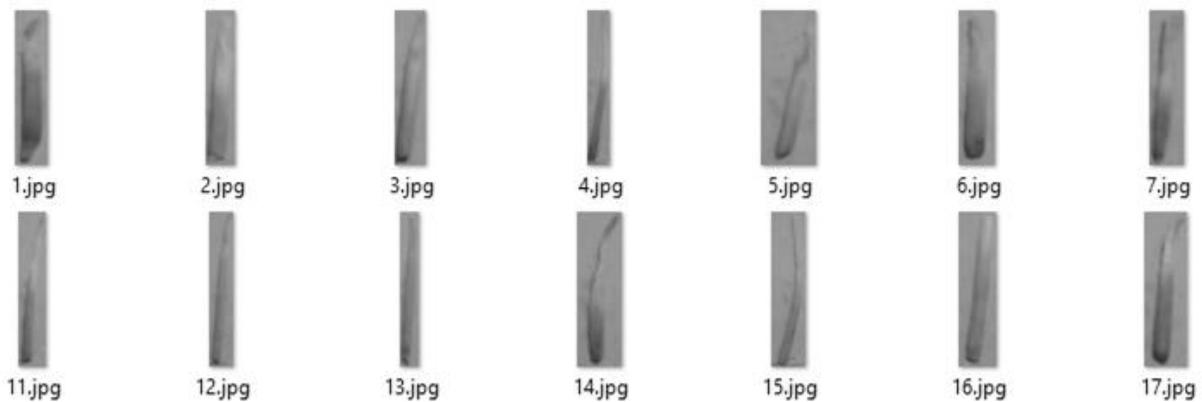


**Gambar 5.12 Hasil Gambar Template RGB Embun Bulu**

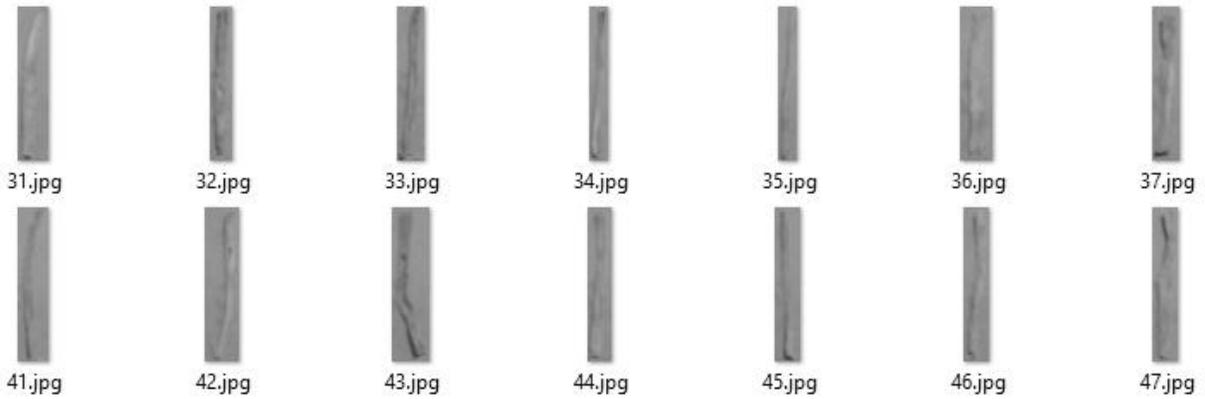


**Gambar 5.13 Hasil Gambar Template RGB Moler**

Pada gambar 5.11 merupakan gambar hasil dari template RGB dari citra Penyakit Daun Bawang Merah Bercak Ungu, gambar 5.12 gambar hasil dari template RGB dari citra Penyakit Daun Bawang Merah Embun Bulu, dan gambar 5.13 gambar hasil dari template RGB dari citra Penyakit Daun Bawang Merah Moler . Dan akan diubah ke citra grayscale agar mempermudah dalam proses menggunakan metode template matching



**Gambar 5.14 Hasil Template Grayscale Bercak Ungu**



**Gambar 5.15 Hasil Template Grayscale Embun Bulu**



**Gambar 5.16 Hasil Template Grayscale Moler**

Pada Gambar 5.14 merupakan hasil Template yang sudah di Grayscale dari citra Penyakit Daun Bawang Merah Bercak Ungu, gambar 5.15 merupakan hasil Template yang sudah di Grayscale dari citra Penyakit Daun Bawang Merah Embun Bulu, dan gambar 5.16 merupakan hasil Template yang sudah di Grayscale dari citra Penyakit Daun Bawang Merah Moler. Dan selanjutnya Template sudah bisa melakukan metode Template Matching.

**Tabel 5.4 Kode pemrograman Pembuatan Template**

Baris	Kode Program
1	while True:
2	if button == False:
3	GPIO.output(24, True)
4	cam = cv2.VideoCapture(0)
5	ret, frame=cam.read()
6	i=0
7	cv2.imwrite('a + str(i).jpg', frame)
8	img = cv2.imread("a + str(i).jpg")
9	gray_img = cv2.cvtColor(img,
10	

11	cv2.COLOR_BGR2GRAY)
12	i=i+1

Pada tahap implementasi kode program template Matching ini bertujuan untuk merealisasikan kode template Matching. Pada tabel 5.5 menjelaskan tentang proses template matching pada citra untuk mendeteksi penyakit daun bawang merah dengan menggunakan 1 gambar template untuk masing masing jenis. Pada baris 1-4, 6-9, 11-14 merupakan kode untuk membaca template masing masing penyakit daun bawang merah kemudian membaca gambar yang didapat dari kamera webcam. Setelah itu setiap nilai piksel pada masing – masing gambar dimasukkan kedalam sebuah array. Kemudian array akan mencocokkan antara gambar dan template dengan menggunakan fungsi cv2.TM\_CCoeff\_NORMED yang telah tersedia di dalam library Opencv. Jika kecocokan diatas 80% maka akan menjalankan fungsi for dibawahnya yang akan memberi gambar berbentuk persegi pada gambar salah satu penyakit daun bawang merah dan juga akan menyalakan LED sesuai dengan penyakit yang terdeteksi. Nilai threshold didapatkan  $\geq 0.78$  untuk mendapatkan nilai eror kecil, jika nilai threshold  $\geq 0.90$  maka nilai erro yang didapatkan sangat besar.

**Tabel 5.5 Kode pemrograman menggunakan Template Matching**

Baris	Kode Program
1	Templatea1 = cv2.imread("a1.jpg", cv2.IMREAD_GRAYSCALE)
2	a, b = templatea1.shape[::-1]
3	result = cv2.matchTemplate(gray_img, templatea1, cv2.TM_CCoeff_NORMED)
4	loc = np.where(result $\geq$ 0.78)
5	templateb1 = cv2.imread("b1.jpg", cv2.IMREAD_GRAYSCALE)
6	aa,bb = templateb1.shape[::-1]
7	result = cv2.matchTemplate(gray_img, templateb1, cv2.TM_CCoeff_NORMED)
8	lac = np.where(result $\geq$ 0.78)
9	
10	templatec1 = cv2.imread("c1.jpg", cv2.IMREAD_GRAYSCALE)
11	aaa, bbb = templatec1.shape[::-1]
12	result = cv2.matchTemplate(gray_img, templatec1, cv2.TM_CCoeff_NORMED)
13	lic = np.where(result $\geq$ 0.78)
14	



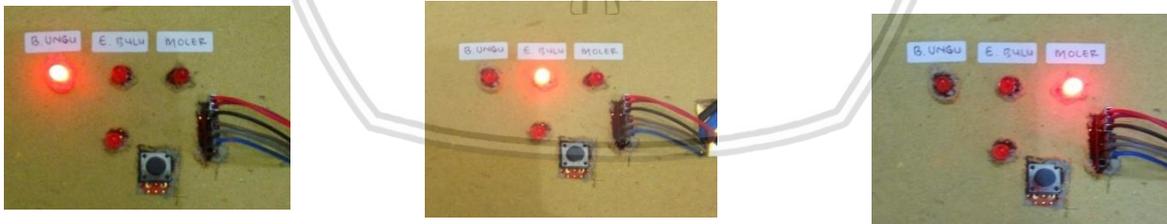
```

15 for pt in zip(*loc[::-1]):
16     cv2.rectangle(img, pt, (pt[0] + a, pt[1] +
17         b), (0, 255, 0), 3)
18         GPIO.output(25, True)
19
20
21 for pt in zip(*lac[::-1]):
22     cv2.rectangle(img, pt, (pt[0] + aa, pt[1] +
23         bb), (0, 255, 0), 3)
24         GPIO.output(25, True)
25
26 for pt in zip(*lic[::-1]):
27     cv2.rectangle(img, pt, (pt[0] + aaa, pt[1] +
28         bbb), (0, 255, 0), 3)
29         GPIO.output(25, True)
    
```

Pada tahap akhir implementasi ini bertujuan untuk memberikan outputan pada sistem ini. pada table 5.6 menjelaskan tentang proses pengeluaran sistem

**Tabel 5.6 Kode pemrograman Output LED**

Baris	Kode Program
1	Else
2	GPIO.output(24, False)
3	GPIO.output(25, False)
4	GPIO.output(22, False)
5	GPIO.output(26, False)



**Gambar 5.17 Hasil Output LED penyakit daun bawang Threshold  $\geq 80$**



**Gambar 5.18 Hasil Output LED penyakit daun bawang Threshold  $\geq 90$**

## BAB 6 PENGUJIAN DAN ANALISIS

Bab ini membahas proses pengujian serta menganalisis hasil dari pengujian yang dilakukan berdasarkan sistem yang telah dibuat. Adapun tujuan dilakukannya pengujian ada untuk mengetahui apakah semua kebutuhan yang diharapkan telah terpenuhi oleh sistem. Proses pengujian yang dilakukan yakni berupa pengujian fungsional, pengujian akurasi dan pengujian kecepatan pemrosesan sistem, dimana pengujian fungsional yakni menguji fungsi dari perangkat keras dalam hal ini berupa alat yang digunakan serta kamera apakah dapat bekerja spesifikasinya, pengujian akurasi yakni menguji seberapa akurat sistem yang telah dirancang dan diimplementasikan dibandingkan nilai atau hasil yang sebenarnya, sedangkan pengujian kecepatan pemrosesan sistem yakni untuk menguji waktu pemrosesan ketika sistem mulai di jalankan hingga menghasilkan jenis klasifikasi. Berikut dijelaskan beberapa skenario pengujian yang dilakukan untuk menguji sistem.

### 6.1 Pengujian Fungsional Kamera

Kamera merupakan alat yang mempunyai tugas utama dalam sistem ini untuk melakukan pembacaan terhadap obyek yang dideteksi. Pada pengujian ini dilakukan pengambilan gambar dari daun bawang merah, yang akan diproses untuk dijadikan template serta pengujian.

#### 6.1.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui fungsionalitas pembacaan kamera yang diterapkan apakah sesuai dengan karakteristik .

#### 6.1.2 Prosedur Pengujian

Berikut prosedur yang dilakukan untuk menguji kamera :

1. Meremote mikrokomputer Rasberry Pi dengan laptop.



2. Membuka aplikasi VNC Viewer.



3. Menghubungkan kamera dengan Raspberry Pi



4. Menghubungkan *Push Button* dengan pin GPIO



5. Menjalankan kode program untuk pengambilan gambar yang telah tersedia di Raspberry Pi.
6. Menekan tombol *Push Button*



7. Mengamati hasil pembacaan gambar obyek di monitor laptop untuk mengetahui tingkat keberhasilan dan kegagalan.

Adapun untuk mengukur jumlah keberhasilan dengan menggunakan **Persamaan (6.1)**

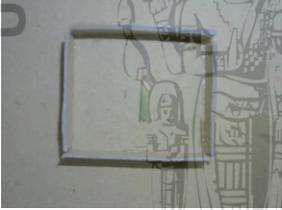
$$\text{Persentasi Keberhasilan} = \frac{\text{jumlah berhasil}}{\text{jumlah percobaan}} = 100 \% \quad (6.1)$$

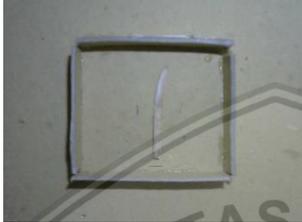
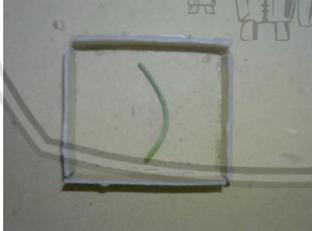
$$\text{Persentasi Error} = \frac{\text{jumlah gagal}}{\text{jumlah percobaan}} = 100 \% \quad (6.2)$$

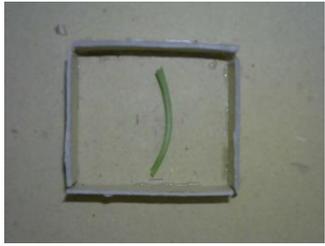
### 6.1.3 Hasil dan Analisis Pengujian

Hasil pengujian kamera ditunjukkan pada Tabel 6.1 , berdasarkan tabel tersebut terlihat tingkat keberhasilan pengambilan gambar beberapa macam penyakit daun bawang merah yang berbeda.

**Tabel 6.1 Hasil perngujian pembacaan Kamera**

Pengujian ke	hasil	keterangan
1		Berhasil
2		Berhasil
3		Berhasil
4		Berhasil

Pengujian ke	hasil	keterangan
5		Berhasil
6		Berhasil
7		Berhasil
8		Berhasil
9		Berhasil

Pengujian ke	hasil	keterangan
10		Berhasil

Berdasarkan **Persamaan (6.1)** untuk menentukan persentase *keberhasilan* pada setiap pengujian, diperoleh rata-rata *keberhasilan* sebesar 100% dan **Persamaan (6.2)** persentase *error* sebesar 0% . Contoh perhitungan persentase *keberhasilan* pada pengujian sebagai berikut :

Adapun untuk menghitung nilai rata-rata *keberhasilan* keseluruhan pengujian adalah sebagai berikut:

$$\begin{aligned} \text{Persentasi Keberhasilan} &= \frac{\text{jumlah berhasil}}{\text{jumlah percobaan}} \times 100 \% \\ &= \frac{10}{10} \times 100 \% \\ &= 100 \% \end{aligned}$$

$$\begin{aligned} \text{Persentasi error} &= \frac{\text{jumlah gagal}}{\text{jumlah percobaan}} = 100 \% \\ &= \frac{0}{10} \times 100 \% \\ &= 0 \% \end{aligned}$$

Rata-rata *keberhasilan* yang dihasilkan kamera sangat besar, sehingga dapat dikatakan bahwa akurasi pembacaan kamera sangat baik. Walaupun tidak terdapat sedikit *error* antara pembacaan kamera.

## 6.2 Pengujian Fungsional Tampilan pada LED

Lampu LED merupakan komponen untuk menampilkan salah satu penyakit pada daun bawang merah. Salah satu penyakit yang ditampilkan pada lampu LED pada umumnya berdasarkan apa yang telah ditentukan pada mikrokomputer, oleh karena itu pengujian ini dilakukan dengan melihat kesesuaian antara tampilan dengan rancangan yang ada.

### 6.2.1 Tujuan Pengujian

Pengujian Tampilan Lampu LED ini bertujuan untuk mengetahui apakah hasil pembacaan gambar yang berbeda dan hasil klasifikasi yang sesuai dengan yang diharapkan, Pada saat ditampilkan pada Lampu LED sesuai dengan kode program yang telah dirancang dan diimplementasikan.

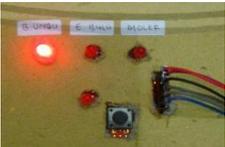
### 6.2.2 Prosedur Pengujian

Untuk melakukan pengujian tampilan pada Lampu LED dilakukan dengan prosedur sebagai berikut :

1. Menghubungkan Sistem Pendeteksi Penyakit daun yang telah dibuat dengan laptop.
2. Membuka aplikasi VNC Viewer.
3. Menghubungkan dengan pin GPIO
4. Menjalankan kode program untuk menampilkan lampu LED yang telah tersedia di Raspberry Pi.
5. Menekan tombol *Push Button*
6. Mengamati hasil pembacaan gambar obyek di monitor laptop untuk menegetahui tingkat keberhasilan dan kegagalan.
7. Mengamati hasil salah satu Lampu LED yang aktif

### 6.2.3 Hasil dan Analisis Pengujian

Tabel 6.2 Hasil pengujian Lampu LED

No	Kondisi	Gambar	Keterangan
1	PenyakitBercak Ungu		Berhasil
2	Penyakit Embun Bulu		Berhasil
3	Penyakit Moler		Berhasil

Berdasarkan hasil pengujian Lampu LED sesuai yang ditunjukkan pada table 6.2 terlihat bahwa Lampu LED dapat berjalan sesuai dengan yang diharapkan dengan kondisi-kondisi fungsi yang berbeda sekalipun.

### 6.3 Pengujian Akurasi Hasil Klasifikasi *Template Matching*

Sistem Pendeteksi Penyakit Daun Bawang Merah Probolinggo Menggunakan *Template Matching* Berbasis Raspberry Pi ini mempunyai tujuan utama untuk dapat mengklasifikasikan jenis penyakit yang diderita oleh daun bawang merah melalui pengamatan daun, oleh karena itu perlu diketahui tingkat keakuratan sistem dalam melakukan klasifikasi.

#### 6.3.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk menentukan nilai akurasi penggunaan metode *Template Matching* pada sistem pendeteksi penyakit daun yang telah dibuat.

#### 6.3.2 Prosedur Pengujian

Prosedur pengujian akurasi *Template Matching* dilakukan adanya masukan 1 gambar berdasarkan pembacaan kamera Logitech C270 yang menggunakan rasio perbandingan antara data latih dan data uji yaitu 2:1. Penentuan proporsi data latih dan data uji dengan rasio 2:1 dikarenakan untuk mendapatkan akurasi semakin tinggi maka jumlah data latih harus lebih banyak minimal 2 kali lipat dari data uji (Adhieputra, 2010). Dimana dari jumlah data set sebanyak 90 data, diantaranya dipilih secara acak untuk digunakan sebagai data latih yakni sebanyak 60 data dan yang lainnya digunakan sebagai data uji yaitu sebanyak 30 data. Untuk menentukan nilai akurasi dari sistem yakni dengan membandingkan hasil klasifikasi penyakit daun yang dilakukan oleh sistem dengan hasil status yang didapatkan lapangan pada proses pengumpulan data, rumus yang digunakan untuk menghitung akurasi ditunjukkan pada **Persamaan (6.3)**.

$$\text{Persentasi Keberhasilan} = \frac{\text{jumlah berhasil}}{\text{jumlah percobaan}} \times 100 \% \quad (6.3)$$

$$= \frac{10}{10} \times 100 \%$$

$$= 100 \%$$

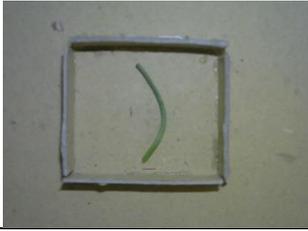
### 6.3.3 Hasil dan Analisis Pengujian

Tabel 6.3 Data uji dan hasil pengujian *Template Matching*

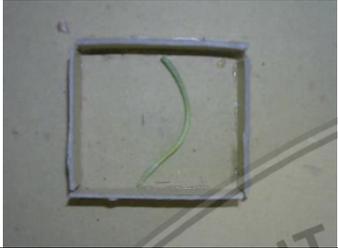
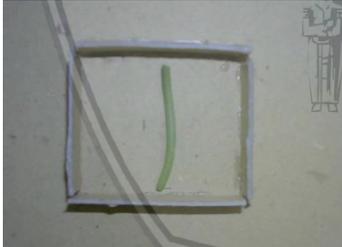
No	Gambar	Kelas	Hasil Sistem	Kesesuaian
1		Bercak Ungu	Bercak Ungu	Sesuai
2		Bercak Ungu	Bercak Ungu	Sesuai
3		Bercak Ungu	Bercak Ungu	Sesuai
4		Bercak Ungu	Bercak Ungu	Sesuai
5		Bercak Ungu	Bercak Ungu	Sesuai
6		Bercak Ungu	Bercak Ungu	Sesuai
7		Bercak Ungu	-	Tidak Sesuai

No	Gambar	Kelas	Hasil Sistem	Kesesuaian
8		Bercak Ungu	Bercak Ungu	Sesuai
9		Bercak Ungu	Bercak Ungu	Sesuai
10		Bercak Ungu	-	Tidak Sesuai
11		Embun Bulu	Embun Bulu	Sesuai
12		Embun Bulu	Embun Bulu	Sesuai
13		Embun Bulu	Embun Bulu	Sesuai
14		Embun Bulu	Embun Bulu	Sesuai

No	Gambar	Kelas	Hasil Sistem	Kesesuaian
15		Embun Bulu	Embun Bulu	Sesuai
16		Embun Bulu	Embun Bulu	Sesuai
17		Embun Bulu	Embun Bulu	Sesuai
18		Embun Bulu	-	Tidak Sesuai
19		Embun Bulu	-	Tidak Sesuai
20		Embun Bulu	-	Tidak Sesuai

No	Gambar	Kelas	Hasil Sistem	Kesesuaian
21		Moler	Moler	Sesuai
22		Moler	Moler	Sesuai
23		Moler	Moler	Sesuai
24		Moler	Moler	Sesuai
25		Moler	Moler	Sesuai
26		Moler	Moler	Sesuai



No	Gambar	Kelas	Hasil Sistem	Kesesuaian
27		Moler	Moler	Sesuai
28		Moler	Moler	Sesuai
29		Moler	Moler	Sesuai
30		Moler	-	Tidak Sesuai

Hasil Pengujian no 1 dengan penyakit Bercak Ungu yang menghasilkan hasil sistem (Bercak Ungu) dan hasil kesesuaian (Sesuai) menunjukkan alat pendeteksi penyakit daun bawang merah ini mendeteksi dengan baik. Dimana cara pendeteksiannya dengan mengambil salah satu obyek penyakit daun bawang yang dicapture melalui kamera yang akan dibandingkan dengan salah satu data template yang sudah tersedia di Raspberry Pi dengan nilai Array yang mempunyai kemiripan yang sama. Kemudian akan diproses di metode Template Matching, apakah ada kesamaan dengan template

jika nilai 80% keatas maka akan terdeteksi penyakit tersebut, apabila dibawah 80% maka tidak ada sama sekali kemiripan dengan template yang tersedia.

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.3 terlihat bahwa dari jumlah 30 data terdapat 6 hasil dari sistem yang tidak sesuai dengan kelas sebenarnya. Sehingga akurasi yang diperoleh Sistem Pendeteksi Penyakit Daun dengan Metode *Template Matching* ini berdasarkan **Persamaan (6.3)** adalah sebesar 80%. Proses perhitungan akurasinya sebagai berikut :

$$\begin{aligned}
 \text{Akurasi} &= \frac{\text{Total data} - \text{Data tidak sesuai}}{\text{Total data}} \times 100\% && \text{(6.3)} \\
 &= \frac{30-6}{30} \times 100\% \\
 &= \frac{24}{30} \times 100\% \\
 &= 80\%
 \end{aligned}$$

## 6.4 Pengujian Waktu Komputasi Pemrosesan Sistem

### 6.4.1 Tujuan Pengujian

Tujuan dilakukannya pengujian ini adalah untuk mengetahui waktu yang dibutuhkan sistem untuk melakukan proses pengklasifikasian jenis penyakit daun menggunakan metode *Template Matching*, hal ini diperlukan untuk mengetahui performansi dari sistem yang telah dibuat.

### 6.4.2 Prosedur Pengujian

Untuk melakukan pengujian waktu komputasi pemrosesan sistem dilakukan dengan cara mengukur waktu komputasi ketika program dimulai dan waktu komputasi ketika program selesai dalam satu siklus sebanyak 30 kali pengujian. Jika telah diketahui waktu komputasi awal dan akhir maka waktu yang dibutuhkan untuk sistem bekerja adalah nilai waktu komputasi akhir dikurangi dengan waktu komputasi awal. Prosedur pengujian ini dilakukan dengan menerapkan waktu pada program.

### 6.4.3 Hasil dan Analisis Pengujian

Tabel 6.4 Hasil pengujian waktu komputasi

Nama Pengujian	Waktu Komputasi (s)
Pengujian ke-1	15.139
Pengujian ke-2	16.435
Pengujian ke-3	15.728
Pengujian ke-4	16.771
Pengujian ke-5	16.855
Pengujian ke-6	16.672
Pengujian ke-7	16.845
Pengujian ke-8	14.896
Pengujian ke-9	16.263
Pengujian ke-10	17.004
Pengujian ke-11	16.285
Pengujian ke-12	14.660
Pengujian ke-13	16.568
Pengujian ke-14	14.778
Pengujian ke-15	16.325
Pengujian ke-16	16.156
Pengujian ke-17	14.551
Pengujian ke-18	16.381
Pengujian ke-19	16.444
Pengujian ke-20	14.566
Pengujian ke-21	16.972
Pengujian ke-22	16.159
Pengujian ke-23	15.934

Nama Pengujian	Waktu Komputasi (s)
Pengujian ke-24	16.356
Pengujian ke-25	16.392
Pengujian ke-26	14.921
Pengujian ke-27	13.551
Pengujian ke-28	16.621
Pengujian ke-29	16.514
Pengujian ke-30	16.579
<b>Rata-rata</b>	15.977

Berdasarkan hasil pengujian yang ditunjukkan pada Tabel 6.4 Hasil pengujian waktu komputasi pemrosesan sistem terlihat bahwa dari jumlah 30 data pengujian, yang dimulai dari pengujian ke-1 dengan waktu 15.139s dan pengujian ke-30 terakhir dengan waktu 16.579s. Sehingga rata-rata pengujian waktu dari 30 data yang diuji diperoleh sebesar 15.977s.

## BAB 7 PENUTUP

Bab ini memuat penarikan kesimpulan berdasarkan tahap-tahap yang telah dikerjakan sebelumnya. Selain itu pada bab ini pula peneliti menyampaikan saran yang diharapkan dapat digunakan untuk pengembangan penelitian yang serupa dengan penelitian ini selanjutnya.

### 7.1 Kesimpulan

Sesuai dengan rumusan masalah yang diajukan diawal penelitian serta berdasarkan hasil analisis dari pengujian yang dilakukan maka dapat ditarik beberapa kesimpulan sebagai berikut :

1. Implementasi Template Matching pada Raspberry Pi untuk mengenali penyakit daun bawang mempunyai inputan kamera untuk mengcapture gambar obyek akan membandingkan nilai obyek dengan 30 template daun bawang kemudian akan dikelompokkan dengan nilai persentasi yang tinggi .
2. Dibangun prototype terlebih dahulu dan gambar akan diambil dengan kamera setelah itu diproses dengan metode *template matching* dan akhirnya akan mempunyai outputan di Led.
3. Pengujian Akurasi metode *Template Matching* yang terlihat bahwa dari jumlah 30 data terdapat 6 hasil dari sistem yang tidak sesuai dengan kelas sebenarnya. Sehingga akurasi yang diperoleh Sistem Pendeteksi Penyakit Daun dengan Metode *Template Matching* ini adalah sebesar 80%.
4. Performansi waktu respon Sistem Penyakit Daun Bawang Merah Probolinggo Menggunakan Template Matching Berbasis Rasberry Pi mempunyai nilai kecepatan waktu pemrosesan rata-rata sebesar 15.977 detik dari 30 kali pengujian.

### 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 atau algoritma klasifikasi lain untuk membandingkan metode atau algoritma manakah yang mempunyai tingkat keakuratan lebih tinggi.

2. Menerapkan sistem serupa dengan tampilan antarmuka melalui *mobile apps* sehingga lebih memudahkan mobilitas penggunaan.



## Daftar Pustaka

- Ardhini warih utami, R. s. (2015). penyakit tanaman bawang merah. *Sistem pakar identifikasi penyakit tanaman bawang merah menggunakan metode teorema bayes*, 4-5.
- Bagus K.Udiarto, W. S. (2005). *OPT Bawang Merah Bercak Ungu atau Trotol(Purple Blotch)*. Bandung: Direktorat Perlindungan Holtikultura.
- budidaya, i. (2018, April 5). *cara merawat tanaman bawang merah*. Retrieved from cara-merawat-tanaman-bawang-merah-di-musim-hujan: [www.ilmubudidaya.com](http://www.ilmubudidaya.com)
- Fadlil, T. A. (2013). kepala BPP (Badan Pelaksana Penyuluhan). *sistem pakar untuk mendiagnosa hama dan penyakit tanaman bawang merah*, 1-2.
- HD Webcam c270. (2010). Retrieved from Logitech: <http://www.logitech.com/en-gb/product/hd-webcam-c270>
- Kandir, N. (2016, Desember 6). *Open CV*. Retrieved from Mengenal open cv dan python: <http://norkandirblog.wordpress.com>
- OpenCV. (2017). *OpenCV*. Retrieved from OpenCV: <https://opencv.org/>
- OpenCV. (2018, juni 22). *template\_matching*. Retrieved from [opencv/template\\_matching: http://docs.opencv.org/2.4/doc/tutorials/hisograms/template\\_matching/template\\_matching.html](http://docs.opencv.org/2.4/doc/tutorials/hisograms/template_matching/template_matching.html)
- Pi3, R. (2018, Februari 4). *Raspberrypi*. Retrieved from Raspberry-pi-3-model-b: <http://www.raspberrypi.org>
- Rahadiyanto, F. (2017). *Produksi Bawang Merah Capai 60 Ribu Ton*. probolinggo: [probolinggokab.go.id](http://probolinggokab.go.id).
- Raspberry Pi 2 Model B. (2015). Retrieved from Raspberry pi: <https://www.raspberrypi.org/products/raspberry-pi-2-model-b/>
- sari, R. (2017). Template Matching. *Identifikasi jenis pohon mangga berdasarkan daunnya menggunakan metode Template Matching*, 5-6.
- team, O. (2017). *OpenCV*. Retrieved from About OpenCV: <https://opencv.org/about.html>
- What is a Raspberry Pi? (2107). Retrieved from opensource.com: <https://opensource.com/resources/what-raspberry-pi>
- What is a webcam? (2017). Retrieved from Digital Unite: <https://www.digitalunite.com/guides/email-skype/webcams/what-webcam>